Utveckling av digitaliserat låssystem

ARIAN MALOKU OCH ANDRÉ NILSSON
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Utveckling av digitaliserat låssystem

Av

Arian Maloku och André Nilsson

Department of Electrical and Information
Technology Faculty of Engineering, LTH, Lund
University SE-221 00 Lund, Sweden

LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds universitet
Box 882
251 08 Helsingborg

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

Tryckt i Sverige
Lunds universitet
Lund 2020

Sammanfattning

Detta examensarbete utfördes på företaget Devex Mekatronik i Lund som under arbetets gång köptes upp av företaget Etteplan.
Målet med det här examensarbetet var att vidareutveckla och förbättra ett låssystem till ett medicinskåp med tillhörande databas. Devex Mekatronik hade påbörjat en egen lösning på problemet men ville utveckla ett fullständigt system som fungerade på det sätt de hade tänkt sig och som sedan skulle presenteras för potentiella kunder. Kortfattat gick arbetsprocessen ut på att utveckla en Android applikation som personer med behörighet kan logga in på och öppna olika skåp. Detta innebar då även att skapa en databas med tillhörande PHP-filer som applikationen kan kommunicera med samt att utveckla ett PCB-kort som för styrning av motorn i skåpet via appen. Resultatet blev ett säkrare och mer platseffektivt system.

Nyckelord: Android, SQL-databas, PCB, Server

Abstract

This thesis was conducted at the company Devex Mekatronik in Lund which along the way got sold to the company Etteplan.
The objective of this thesis was to further develop and improve a locking system for a medicine cabinet with appurtenant database. Devex Mekatronik had already started working on a solution for the problem but wanted to develop a complete system which would work in the way they wanted it to and could later be presented to potential customers. The summary of the working process was to develop an Android application that authorized people could log into and open different cabinets with. This also included creating a database with associated PHP-files which the application can communicate with and the development of a PCB-card for controlling the motor in the cabinet. This resulted in a safer and more space efficient system.

Keywords: Android, SQL-database, PCB, Server

Förord

Detta examensarbete är en avslutande del av vår utbildning på Elektroteknik med Automation på Campus Helsingborg som omfattar 22,5 högskolepoäng.

Vi vill först och främst börja med att tacka alla på Devex Mekatronik som hjälpt oss genom hela examensarbetet och framfört allt för att de gett oss möjligheten att utföra arbetet hos dem. Ett extra stort tack till våra handledare Thomas hellström som har varit där och stöttat och gett oss en bra handledning under examensarbetets gång.

Vi vill även tacka Pontus Holmberg för den tid du tog för att ha kunnat kolla igenom Eagle cad schemat men även för de olika samtals möten som vi har haft.

Vi vill även tacka vår handledare Mats Lilja och vår examinator Christian Nyberg för all hjälp och rådgivning som de har gett under examensarbetets gång

Innehållsförteckning

1. Inledning

1.1 Bakgrund

Företaget som examensarbetet utfördes på är ingenjörsbolaget Devex Mekatronik som har ett av sina kontor vid Ideon i Lund. Devex Mekatronik är ett konsultföretag grundat 1998 och har kompetens inom allt som har med elektronik att göra. Företaget strävar efter att utveckla idéer till tekniska helhetslösningar som skapar affärsnytta för deras kunder. Vid start av examensarbetet var Devex Mekatronik ett eget företag men är numera en del av den internationella koncernen Etteplan.

En kund till företaget Devex Mekatronik hade utvecklat ett skåp som ska vara till för att lagra olika typer av mediciner i. För att inte obehöriga ska få tillgång till innehållet i skåpet behövdes någon typ av lås som endast behöriga personer kan låsa upp. Devex Mekatronik hade påbörjat en lösning på det problemet vilket bestod av en Arduino (vilket är en programmerbar mikroprocessor skapad av företaget Arduino) som testenhet tillsammans med en enkel variant av en Android app som kunde användas för att scanna en QR-kod på skåpet vilket i sin tur var kopplat till en motor som öppnade skåpdörren. Problemet med det systemet var att det saknade många viktiga funktioner som behövdes samt att de lösningar som hade gjorts behövde förbättras. Vem som helst som har lösenord och användarnamn kan komma åt alla skåp och det finns inget sätt att administrera systemet utan att gå in i själva programmet för applikationen och ändra. Systemet använder kalkylark för att lagra informationen om användarna vilket inte heller är en säker lösning. Lösningen på det första problemet är att lagra namnen på personerna som ska få åtkomst till ett eller flera skåp i en databas. I databasen ska det anges vilka personer som ska få komma åt vilka skåp och på så sätt kan alla personer endast komma åt de skåp de har behörighet att öppna. Det ska även skapas ett Administratörskonto som kommer att ha tillgång till alla skåp samt att kunna registrera, redigera, ändra och bevaka personer. För att kunna göra systemet mindre i storlek är det tänkt att ett kretskort ska användas. Vilket sorts lösning som till slut kommer bli den som väljs ska undersökas senare i arbetet.

1.2 Syfte

Syftet med detta examensarbete är att utveckla ett låssystem till ett medicinskåp. Det förväntade resultatet är att få fram ett säkrare och mer platseffektivt system.

1.3 Målformulering

Examensarbetet ska först undersöka vilka lämpliga komponenter som ska användas till systemet baserat på pris, hållbarhet och kompabilitet. Därefter ska en plan för att utveckla systemet skapas och arbetet genomföras. Slutligen ska det färdigutvecklade systemet monteras in i medicinskåpet. Huvudmålet är att ta fram ett säkrare mer platseffektivt system.

1.4 Problemformulering

Innan arbetet ställdes följande frågeformuleringar:

1. Vilket system är mest lämpligt utifrån platsutrymme i skåpet?
2. Vilket system fungerar bäst för trådlös sammankoppling?
3. Vilka komponenter ska väljas till utvecklingen av det nya systemet?
4. Hur ska alla komponenter länkas samman med varandra för att skapa den färdiga produkten?
5. Vad kommer förbättras med det nya systemet ur säkerhetsaspekt?
6. Vilken typ av databas är mest lämplig att använda?

1.5 Avgränsningar

Som tidigare nämnts har en stor del av detta examensarbete inneburit att utveckla en app till Android. Dock utvecklades inte någon motsvarighet till IOS vilket är Apples operativsystem. Detta på grund av att ingen av författarna är det minsta bekant med utvecklingsmiljön som Apple använder sig av samt att det förmodades att ett sådant projekt skulle sträcka sig över tidsgränsen.

1.6 Resurser

Vid utvecklingen av en databas samt caddning av kretskort (och programmering av Arduino), behövdes det tillgång till dator med utvecklingsmiljöer som passade de komponenter som skulle användas. Ett kretskort behövdes även skaffas (samt en mindre variant av Arduino), vilket Devex Mekatronik stod för. Slutligen behövdes även en Android-telefon för att kunna testa hela systemet och en sådan fanns tillgänglig på företagets kontor.

Vid arbete med de saker författarna kunde mindre om såsom kretskort-caddning och databasutveckling kunde hjälp fås av anställda på Devex Mekatronik. Det fanns möjlighet att jobba på deras kontor och när något skulle programmeras i utvecklingsmiljöer författarna själva kunde skaffa fanns även möjlighet att jobba på skolan eller hemifrån.

2. Teknisk Bakgrund

Den här delen av rapporten beskriver vilka program, programmeringsspråk och komponenter som användes för genomförandet av examensarbetet.

2.2 SQL

SQL står för “Structured Query Language” och är ett programmeringsspråk som används för att skapa, editera samt hämta data från databaser av relationstypen. SQL utvecklades av forskare på det amerikanska företaget IBM under 1970-talet [1]. Nedan följer några exempel på hur SQL kan användas.

Låt oss säga att man vill skapa en tabell för alla anställda på ett företag och spara deras namn och telefonnummer. Tabellen får namnet “employee”. Vid skapandet av en tabell “employee” med namnet “name” och telefonnummer “nbr”
kan följande kod användas:

```
CREATE TABLE employee (
 name VARCHAR(20),
 nbr INTEGER
);
```

Talet 20 specificerar att namnet på den anställde inte får vara längre än 20 tecken. Integer innebär att variabeln nbr ska vara ett heltal.

Tabellen “employee” är nu skapad men för att sätta in värden i tabellen behöver man använda INSERT vilket visas i nedanstående kod:

```
INSERT INTO employee VALUES (“Peter”, 123456789);
```

Koden matar in personen “Peter” i tabellen med tillhörande telefonnummer “123456789”. Om man sedan vill hämta data från tabellen eller tabellerna så kan man göra det med funktionen SELECT. Man kan kombinera detta med funktionen WHERE för att sätta villkor för den data man vill hämta. Detta kan exempelvis användas på följande sätt:

```
SELECT name
FROM employee
WHERE nbr = 123456789;
```

Ovanstående kodstycke hämtar namnet på alla personer i tabellen med telefonnumret "123456789".

Det finns många andra funktioner i SQL men det är främst dessa som användes för databasen i examensarbetet.

2.3 Android Studio

För att utveckla själva applikationen till Android-telefonen valdes programmet Android Studio eftersom det kändes mest bekvämt då detta program tidigare hade använts till ett projekt i en kurs. Android Studio är det officiella utvecklingsprogrammet för Android och släpptes för första gången 2014 [2]. Programmet stödjer programmeringsspråken Java, Kotlin och C++. Applikationen som utvecklades till examensarbetet skrevs i Java eftersom författarna har goda kunskaper i detta programmeringsspråk.

2.3.1 Java

Java är ett programmeringsspråk på högnivå som utvecklades av företaget Sun Microsystems 1995 men ägs numera av Oracle [3]. Det är ett objektorienterat programmeringsspråk som baseras på klasser och som kan köras på det flesta plattformar (som exempelvis Windows, Mac, Linux). Några av fördelarna med att använda Java gentemot andra språk som exempelvis C++ är till exempel att det är mer användarvänligt och pålitligt. Den syntax som C++ använder kan lätt bli komplicerad vilket är något som förbättrats i Java [4].
Eftersom Java även är ett objektorienterat programmeringsspråk till skillnad från programmeringsspråk som till exempel C eller Pascal så minimerar det även risken att användare gör misstag som att upprepa koden igen då objektorienterad programmering hjälper användarna att hålla koll på sin kod [5].

Exempelvis kan följande kod användas för att skriva ut det klassiska "Hello World!":

```
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World!");
}}
```

När klassen HelloWorld körs kommer texten "Hello World!" att skrivas ut i konsolen.

2.4 PHP

PHP står för “Hypertext Preprocessor” och är ett skriptspråk utvecklat av Rasmus Lerdorf 1994 [7]. Ett skriptspråk är ett högnivåspråk som oftast utför mindre rutinuppgifter åt applikationer till skillnad från att programmeringsspråk som istället används för att utveckla fullständiga applikationer. Koden för skriptspråk utförs oftast utan att först kompileras och ligger i de flesta fall innesluten i andra mjukvarumiljöer. PHP är open source och används mycket vid webbutveckling för servrar med dynamiskt innehåll. En server med dynamiskt innehåll kan exempelvis vara en databasserver där innehållet genereras och förändras över tid [6]. På grund av bland annat detta passade PHP mycket bra att använda till detta examensarbete för kommunikation mellan Android-applikationen och SQL-servern.

Koden nedan visar ett program som skriver ut texten “Hello World!” skrivet i PHP:

```
<?php
    echo 'Hello World!';
?>
```

2.5 Google Sheets

Google Sheets är Googles motsvarighet till Microsoft Excel och är ett online-program som innehåller kalkylark. En fördel med att använda sig av Google Sheets jämfört med Excel är att Sheets lagras på Googles servrar och kan kommas åt varifrån som helst. Google Sheets har ett inbyggt skript editeringsprogram vilket kan användas för att utveckla applikationer till kalkylarken. Koden för skripteditingsprogrammet skrivs i JavaScript.

2.5.1 JavaScript

JavaScript är ett objektorienterat skriptspråk som används mycket för webbutveckling. Språkets syntax liknar till stor del den syntax Java använder men en stor skillnad är att i JavaScript utförs koden genom en webb-läsare medan koden i Java inte gör det. JavaScript är också mycket mer fritt än Java och typen för variablerna behöver ej anges i början av programmet utan kan deklareras vid användning [8].

Nedan följer ett exempel på en metod skriven i JavaScript:

```
function moveValue {
        var doc = SpreadsheetApp.getActiveSpreadsheet();
        var sheetName = "blad1";
        var sheet   = doc.getSheetByName(sheetName);
        var i = 1;
        var place1 = sheet.getRange(i, 1).getValue();
        var place2 = sheet.getRange(i, 2).getValue();
        var place3 = sheet.getRange(i, 3).getValue();
        var place4 = sheet.getRange(i, 4).getValue();
        sheet.getRange(i+1, 1).setValue(place1);
        sheet.getRange(i+1, 2).setValue(place2);
        sheet.getRange(i+1, 3).setValue(place3);
        sheet.getRange(i+1, 4).setValue(place4);
}
```

Metoden hämtar värden i kolumn 1, 2, 3 och 4 på raden i och placerar dessa på samma positioner i raden i+1 för kalkylarket “blad1” i Google Sheets.

2.6 PCB

PCB som på svenska även kallas för mönsterkort står för “Printed Circuit Board” och är en platta som med hjälp av spår av ledande material kopplar samman olika elektroniska komponenter. Själva plattan består av flera lager koppar laminerade med ett isolerande material för att undvika kontakt mellan komponenterna som i det flesta fall lödas på mönsterkortet. De ledande spåren etsas sedan in i kortet. Ett mönsterkort kan både skapas genom att komponenterna monteras på ytan av kortet (ytmontering) eller genom att placeras i hål i kortet (hålmontering). Ibland används en kombination av dem. [9]

PCB började utvecklas omkring år 1936 och innan det var färdigutvecklat var konstruktionsmetoden Point-to-Point mycket vanlig. Point-to-Point konstruktion innebär oftast att komponenter lödas fast på en platta och sedan förbinds med varandra genom manuell kabeldragning på baksidan av plattan. Detta gjorde dock plattan mycket stor och klumpig vilket är en av fördelarna att istället använda sig av ett mönsterkort. [10]

2.7 Arduino

En Arduino är ett litet mikrokontrollerkort med öppen källkod som kan läsa av insignaler och sedan omvandla dessa till utsignaler. Arduino utvecklades på Ivrea Interaction Design Institute i Italien och var tänkt att riktas mot studenter som studerat elektronik och programmering [11]. Vid examensarbetet användes en Arduino i kombination med ett kopplingsdäck för att testa kretsarna samt programmet för projektet. En stor fördel med att använda en arduino i kombination med ett kopplingsdäck för test av kretsar jämfört med ett mönsterkort är att man då snabbt och enkelt kan koppla om kretsen ifall något behöver ändras. Arduino finns i många olika varianter men den som användes till examensarbetet var en Arduino Uno.

Nedanstående kod visar ett exempel på hur koden ser ut i Arduino:

```
int button = 3;

// setup initializes serial and the button pin
void setup() {
  Serial.begin(9600);
  pinMode(button, INPUT);
}

void loop() {
  if (digitalRead(button) == HIGH) {
    Serial.write('Yes');
  }
  else {
    Serial.write('No');
  }
  delay(1000);
}
```

Koden loopar och kontrollerar ifall knappen trycks in. Om detta sker skriver den ut “Yes” annars “No”. Vid slutet av varje loop finns en delay som fördröjer loopen med 1 sekund.

2.8 EAGLE

EAGLE står för “Easy Applicable Graphical Layout Editor” och är ett program som används för utveckling av mönsterkort. Programmet utvecklades av företaget Cadsoft Computers och finns tillgängligt för flera olika plattformar. Många komponenter som är till för att monteras på mönsterkort har bibliotek som är skapade för dem och som kan laddas ned och köras i EAGLE. Det finns även möjlighet för användare att skapa sina egna komponenter och bibliotek. [12]

2.9 Notepad++

Notepad++ är ett editeringsprogram för kod som stödjer många olika syntaxer. Eftersom Notepad++ är gratis och kan användas för att skriva PHP-filer så passade detta mycket bra för examensarbetet.

2.10 phpMyAdmin

phpMyAdmin är ett program skrivet i PHP som används för att programmera SQL-servrar online och släpptes för första gången 2015 [13]. phpMyAdmin ger användaren möjlighet att hantera allting som är relevant för databaserna genom SQL-kod eller så kan phpMyAdmin sköta detta automatiskt för de som kan mindre om SQL-programmering. [14]

3. Metod

Följande del av rapporten kommer redovisa vilket tillvägagångssätt som användes vid utförandet av examensarbete.

Den första delen av arbetet var att under ett möte på Devex Mekatronik planera hur hela systemet skulle fungera. Androidtelefonen skulle kommunicera både med en databas och ett PCB-kort. En skiss gjordes på detta vilken kommer visas i nästa kapitel.

Nästa steg var undersöka vad som passade bäst att använda till systemet och bestod av att leta information om följande punkter:

- Vilken databas som kunde användas för bästa resultat
- Vilket program passade bäst att utveckla applikationen i
- Hur PHP fungerar

Efter detta kunde själva arbetet påbörjas. Regelbundna möten med Devex genomfördes under arbetets gång och ändringar i applikationen och databasen gjordes enligt överenskommelser. För att underlätta arbetet användes som sagt en Arduino Uno tillsammans med ett kopplingsdäck vid testandet av applikationen.

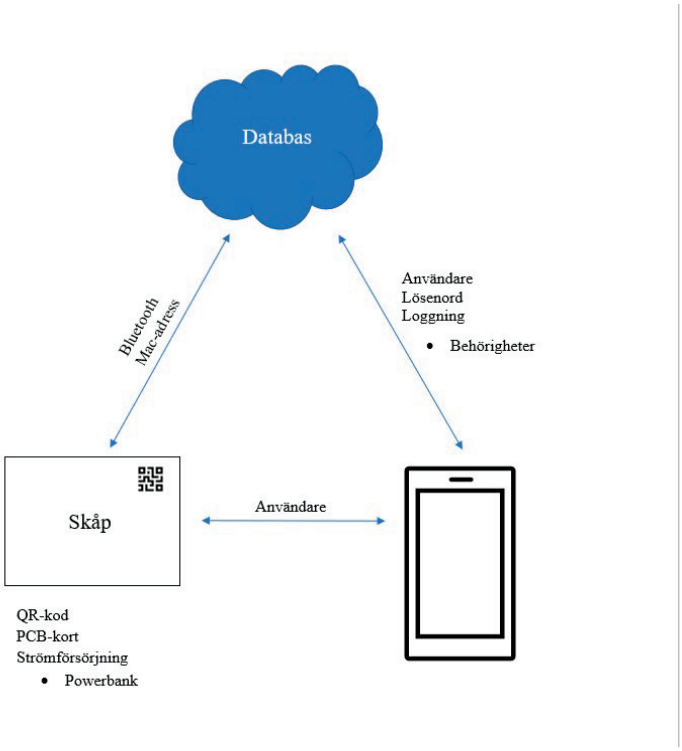
Då applikationens och databasens alla funktioner var färdiga påbörjades utvecklingen av ett PCB-kort. Det första steget i denna process var att skaffa mer kunskap om hur EAGLE fungerade. Därefter kunde ett schema för PCB-kortet konstrueras och sedan skickas till Devex för ett godkännande.

3.1 Källkritik

De källor som använts för examensarbetet har varit från antingen utvecklares och tillverkares egna webbsidor, olika lärosidor, artiklar i webbtidningar eller direkt från företaget. Källa [2], [3], [7], [11], [13] och [14] kan anses pålitliga eftersom de är källor som kommer direkt från de företag eller personer som äger produkterna. Samma sak förutsätts även givetvis om det som hämtats från de anställda på företaget. Källa [1] och [10] förutsätts vara korrekta och objektiva eftersom de är tagna från stora webbtidningar som inte marknadsför något på sina sidor. Källa [4], [5], [6], [8], [9] och [12] kan anses både pålitliga och objektiva då källorna är från stora lärosidor som inte förespråkar eller marknadsför något.

4. Genomförande

Innan arbetet kunde påbörjas var det nödvändigt att tillsammans med företaget skaffa sig en skiss av hur alla delar i systemet skulle hänga ihopa med varandra. Applikationen för Android skulle kunna kommunicera med en SQL-databas som skulle lagra användaruppgifter som namn, efternamn, lösenord, olika behörigheter o.s.v. Applikationen skulle även kunna kommunicera med ett pcb-kort via bluetooth som i sin tur skulle sitta i skåpet för att kunna styra den mekaniska delen d.v.s. öppna/stänga skåpet. På skåpet skulle även en QR-kod finnas med för att som användare kunna med hjälp av applikationen skanna koden och därefter öppna den ifall man har behörighet till skåpet. Figur 4.1 visar skissen som togs fram tillsammans med företaget.



Figur 4.1: Skiss av systemet

4.1 SQL-databas

Vid val av databas var det viktigt att databasen var gratis eftersom företaget först och främst ville ta fram en testvariant av en databas som de enkelt kunde visa för potentiella kunder. Man gjorde detta för att minimera kostnaden samt riskerna ifall det ej skulle finnas potentiella kunder. Efter en del undersökningar och tester av olika fria databaser kunde det konstateras att den bästa sidan som erbjöd databas-hosting gratis var www.000webhost.com.

4.2 Applikation för Android

Eftersom programmet Android Studio hade använts till tidigare projekt i andra kurser var det mest lämpligt att använda detta program för examensarbetet. För att göra arbetet enklare delades applikationens funktioner upp i mindre delar som sedan utfördes i små självständiga projekt. Genom att skapa flera mindre program istället för ett enda stort kunde man underlätta felsökningen i koden. Följande funktioner utvecklades i mindre projekt:

- Inloggning och registrering
- Anslutning med Bluetooth-enhet
- Uppladdning av bild till server
- Kontroll av behörighet till skåp
- Bildvisningsprogram

Punkterna visar huvudfunktionerna för de olika självständiga programmen men även mindre funktioner lades till i dem. Efter att varje mindre program fungerade, adderades det till ett större program som sedan blev den slutgiltiga applikationen.

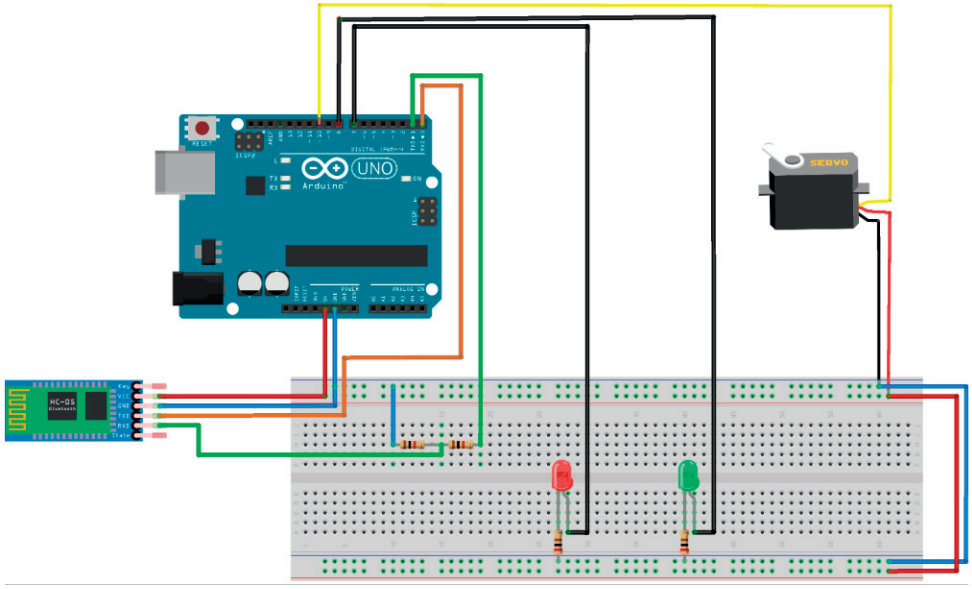
4.2.1 Kommunikation med databas

SQL-servern som används till examensarbetet har en filhanterare där användare kan ladda upp filer. För kommunikation mellan Android applikationen och SQL-servern används PHP-filer som placerades i filhanteraren. Filerna kunde sedan kommas åt genom respektive URL-adress från Android Studio.

4.2.2 Bluetooth-tester med Arduino

För att testa anslutningen mellan Android-applikationen och Bluetooth-enheten användes en Arduino Uno tillsammans med ett kopplingsdäck och en liten motor. Detta gjorde arbetet mycket enklare än test med hjälp av ett PCB-kort då ingenting behövde konstrueras eller beställas. Vid testningen anslöts Arduinon med motorn och Bluetooth-enheten och sedan kontrollerades det att Android-applikationen kunde styra motorn fram och tillbaka på korrekt

sätt. Figur 4.2 visar kopplingsschemat som togs fram där en servomotor, en HC-05 bluetooth-modul samt två lysdioder är kopplade till en arduino.



Figur 4.2: Figuren visar kopplingsschemat.

4.3 Lagring av bilder

Som en extra säkerhetsåtgärd för systemet skulle en funktion finnas, där användaren kan ta ett kort på skåpet innan denna kan stänga skåpet eftersom att detta underlättar för administratörerna att hålla koll på vad som finns kvar i varje skåp vid en viss tidpunkt. Efter lite undersökning om var det var mest lämpligt att lagra bilden, gjordes tester på två olika servrar för uppladdning av bild. Eftersom servern som användes för projektet var en gratis server med en begränsad mängd dataöverföring per månad verkade det som en bra idé att använda Google Sheets som ett alternativ för bilduppladdning då detta program ger användarna en obegränsad mängd dataöverföring. Nackdelen var dock givetvis att allt inte kommer lagras på samma server. För att undersöka vilket som passade bäst gjordes test på båda servrarna.

4.3.1 Google Sheets

I det första fallet testades Google Sheets i kombination med Google Drive som ett lagringsställe för bilderna och bildinformationen. Med hjälp av Googles skriptediteringsprogram kunde ett program skapas som tog emot bilder skickade från Android Studio, lagrade dem på Google Drive och sedan laddade in information samt länkar för alla bilder i det olika fälten i Google Sheets. Programmet sorterade även informationen i ordning med avseende på datum och tid. Det lagrade de tio senaste personerna som öppnat varje skåp i olika kalkylblad. Efter en del tester med detta program kunde det konstateras att det var ganska långsamt. Det tog ganska lång tid att skicka bilden och även att uppdatera listan i Google Sheets. Efter ett nytt möte med Devex bestämdes det att det andra alternativet skulle testas istället.

4.3.2 Lagring på 000webhost

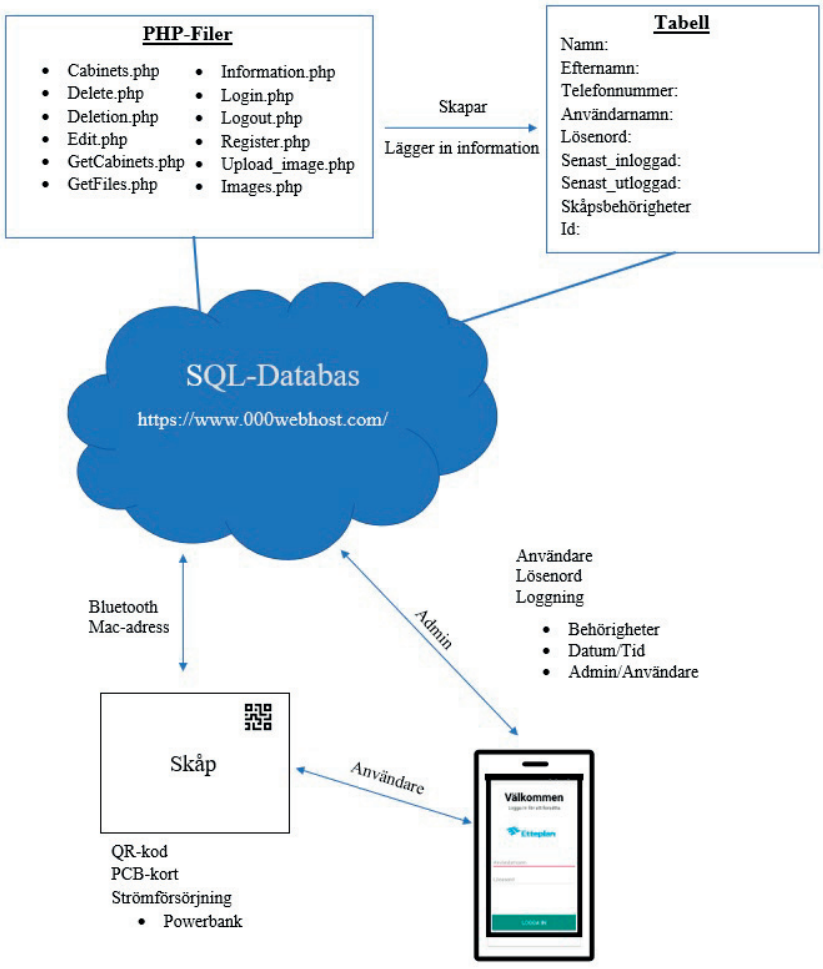
Eftersom filhanteraren på 000webhost även kan lagra bilder så testades detta som ett lämpligt alternativ till bilduppladdningen. En mapp skapades för detta och med hjälp av PHP-filer kunde bilderna skickas från Android Studio till mappen på 000webhost.
Detta alternativet visade sig vara mycket snabbare än det föregående och valdes därför som det slutgiltiga alternativet till projektet.

4.4 PCB och EAGLE

Vid start av caddning i EAGLE behövdes en del hjälp från Devex då detta program inte använts särskilt mycket innan. Ett möte med Devex genomfördes där de gav en introduktion till EAGLE samt förslag på vilka komponenter som skulle kunna passa bra till PCB-kortet. Efter en del undersökningar av komponenter kunde ett schema skapas i EAGLE. Detta schema skickades sedan vidare till företaget för godkännande.

5. Resultat

I detta kapitel kommer Resultatet för examensarbetet att redovisas.
Det schema som skapades i början av examensarbetet tillsammans med företaget (Se figur 4.1) följdes till stor del men en del ändringar och tillägg genomfördes. Figur 5.1 visar hur det slutliga systemet kom att se ut.



Figur 5.1: Figuren visar en överblick av det färdiga systemet

5.1 Androidapplikationen

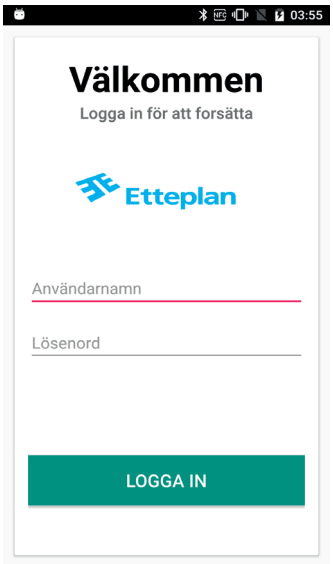
Androidapplikationen består av ett antal Java-klasser där en del klasser styr en viss layout medan andra endast är till för att hjälpa huvudklasserna. I följande delar av kapitlet kommer resultatet för applikationens huvudklasser att redovisas.

5.1.1 Inloggning

Vid start av applikationen körs klassen login.java. Denna klass tar emot data från användaren vid inloggning. Användaren ges möjlighet att mata in e-post och lösenord för sitt konto på servern. Om användaren anger e-post och lösenord som matchar något av de registrerade konton i databasen, startas nästa aktivitet och personen loggas in som antingen vanlig användare eller admin beroende på personens behörighet. Om uppgifterna ej stämmer överens med de i databasen, dyker en text upp och talar om detta för användaren. Mer om hur kommunikationen mellan Android och databasen fungerar tas upp i kapitel 5.4.

5.1.2.1 Inloggning - layout

Layouten för inloggningsskärmen visas i figur 5.2 och består av en inloggningssida med två textfält där användaren kan mata in e-post samt lösenord för sitt konto. Det finns även en knapp med texten “Logga In” som startar processen med att skicka uppgifterna till databasen då användaren trycker på den.



Figur 5.2: Figuren visar sidan för inloggning

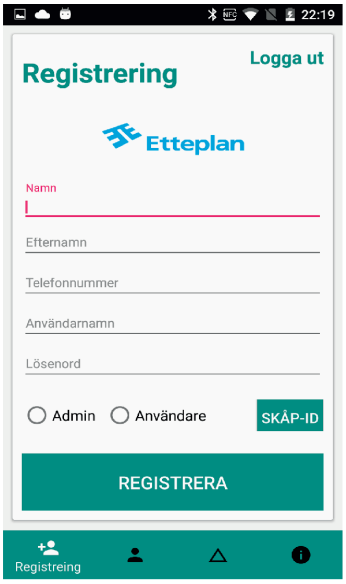
5.1.2 Registrering

För att kunna lägga till fler personer till databasen finns funktionen registrering med i applikationen. I det fall personen som loggar in är admin skickas denna vidare till sidan för registrering av nya personer i databasen (observera att detta inte sker om personen endast är en vanlig användare). Registreringssidan tillåter Administratörer att lägga till följande info om nya personer till databasen:

- **Namn**
- **Efternamn**
- **Telefonnummer**
- **Användarnamn**
- **Lösenord**
- **Administratör eller användare**
- **Skåpsbehörigheter (**De skåp som personen ska ha behörighet att öppna)

5.1.2.1 Registrering - layout

Layouten för registreringsfönstret visas i figur 5.3 och består av fem stycken textrutor med tips på vad som ska anges i respektive ruta. Administratör eller användare väljs under dessa rutor och vid tryck på knappen “SKÅP-ID” öppnas listan över vilka skåp som den nya personen ska ha behörighet att öppna. Längst nere finns en knapp för registrering som slutför registreringsprocessen och sänder all data till databasen.



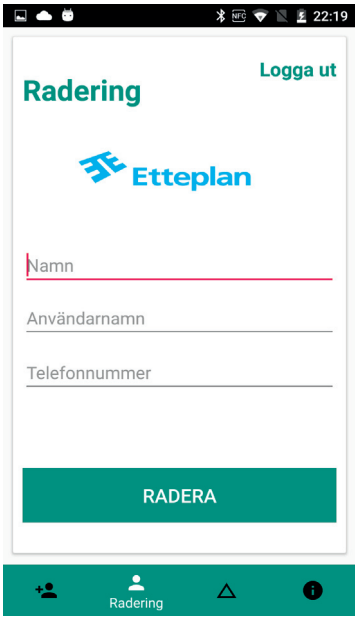
Figur 5.3: Figuren visar sidan för registrering

5.1.3 Borttagning och Redigering

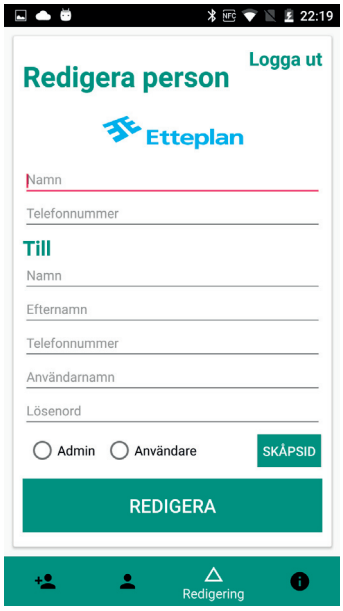
När någon är inloggad som administratör finns även möjligheten “Radering” vilket tillåter administratören att ta bort personer som finns i databasen. Administratören kan även ta bort sig själv med denna funktion. Vid borttagning anger administratören namn, användarnamn samt telefonnummer för personen som ska tas bort. Ifall man inte vill ta bort en person utan endast redigera personens uppgifter finns även funktionen “Redigera” som tar hand om detta. Vid redigering anges först namn och telefonnummer på personen som önskas redigeras. Därefter anger man vad man vill att personens uppgifter ska redigeras till. I dessa fält anges precis samma information som då en ny person ska registreras till databasen. Observera att även funktionerna borttagning och redigering endast finns tillgängliga för administratörer.

5.1.3.1 Borttagning och Redigering - layout

För borttagning av personer finns tre textfält i vilka informationen om personen ska anges för att rätt person ska tas bort från databasen. Knappen “RADERA” avslutar processen och tar bort personen. Samma gäller för knappen “REDIGERA” fast här uppdateras informationen om personen. Figur 5.4 visar sidan för radering och figur 5.5 visar sidan för redigering.



Figur 5.4: Figuren visar sidan för radering



Figur 5.5: Figuren visar sidan för redigering

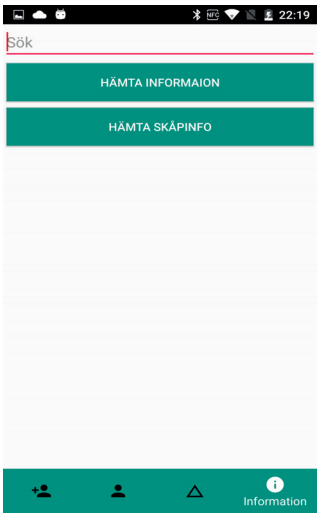
5.1.4 Hämta information

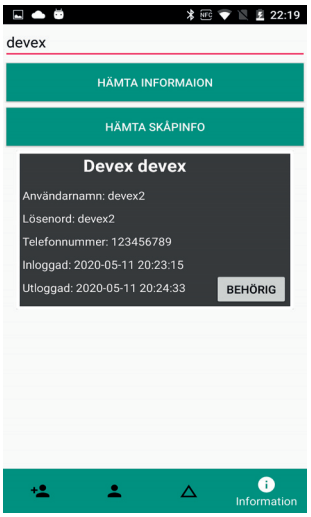
För att administratörer ska kunna se information om olika användare av applikationen samt de bilder på skåpen som laddats upp så finns en sida där den information kan hämtas. I ett textfält kan administratören söka på den information som vill hämtas. Detta visas i figur 5.6. Det finns sedan två möjliga alternativ att välja.

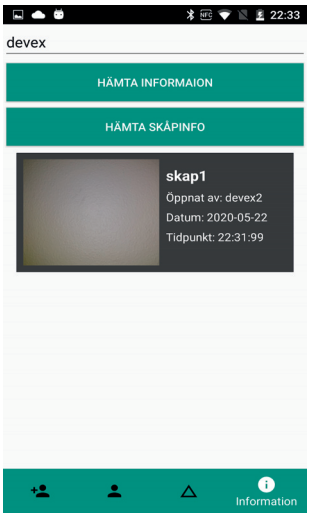
1. Antingen väljer man att hämta information om en användare. Man kan då söka på namnet eller delar av namnet för en person och därefter presenteras de personer vars namn matchar sökningen. Därifrån kan en person väljas och personens användarnamn, lösenord, telefonnummer, tid och datum för när personen senast loggade ut samt tid och datum för när personen senast loggade in visas. Detta visas i figur 5.7. Vill man sedan även veta vilka skåpsbehörigheter personen har kan man även välja att se detta. En ny lista presenteras då och personens behörigheter för skåpen visas.

2. I det andra fallet då man vill söka efter en bild kan man välja att söka på delar av eller hela personens namn, skåpsnummer, datum eller tid för då bilden togs. De bilder som matchar sökningen presenteras då samt ovanstående information om bilden. Detta visas i figur 5.8.

5.1.4.1 Hämta information - layout







Figur 5.6: Visar vyn för information **Figur 5.7 : Hämtar personinformation** **Figur 5.8: Hämtar information från skåp**

5.1.5 Main

Main-klassen är som namnet tyder på, själva huvudklassen i applikationen. Det är den klassen som startas efter att en användare har loggat in i applikationen. På denna huvudsida finns fem alternativ vilka är följande:

- **Anslutning till skåp**
- **Öppna skåpet**
- **Stänga skåpet**
- **Ta bild av skåpet**
- **Logga ut**

Till en början är endast det första alternativet tillgängligt men de resterande alternativen finns möjliga då användaren upprättat en anslutning med skåpet. Alternativen öppna och stänga skåpet vrider motorn fram och tillbaka vilket öppnar samt låser dörren till skåpet. Ta bild på skåpet är det som görs innan användaren ska logga ut från applikationen.

5.1.5.1 Main - layout

Figur 5.9 visar huvudsidan för användaren.

Figur 5.9: Figuren visar huvudsidan för användaren

5.1.5.2 QR-kod/Bluetooth-anslutning

Efter att en användare loggat in och trycker på knappen “Anslutning till skåp”startas klassen som hanterar scanning av QR-kod. Kameran startas automatiskt och användaren ges möjlighet att scanna QR-koden som finns på skåpet. När koden scannats och om användaren har behörighet till skåpet skickas användaren tillbaka till huvudsidan samtidigt som applikationen ansluter till Bluetooth-modulen. Om användaren ej är behörig till skåpet dyker ett felmeddelande upp och anslutning sker ej.

5.1.6 Uppladdning av bild

Då detta alternativ väljs startas kameran och användaren ges möjlighet att ta foto på skåpet insida innan denna loggar ut från applikationen. Denna bild skickas sedan till databasen och applikationen återvänder till huvudsidan.

5.2 SQL-Databasen

De flesta sidor som hostar SQL-databaser använder sig av phpMyAdmin för att administrera databaserna vilket även 000webhost gjorde. För examensarbetet behövdes endast två tabeller. Tabell 1 var den tabell som lagrade användarens uppgifter, de olika åtkomsten till skåpen samt när en användare loggar in och loggar ur applikation. I Tabell 2 lagrades en loggningshistorik på när en viss användare öppnar ett visst skåp i en viss tidpunkt. Tabell 5.1 och Tabell 5.2 visar de två tabellerna som är skapade i phpMyAdmin.

Namn	Efternamn	Telefonnummer	Anvandarnamn	Lösenord	last_login	last_logut	Id	skap0	skap1	skap2	skap3	skap4
Devex	devex	722855837	Devex1	devex1	2020-06-17 23:07:47	2020-06-17 23:07:49	1	0	0	0	0	0
Devex	devex	724565836	Devex2	devex2	2020-06-17 23:10:28	2020-06-17 21:27:05	0	1	0	1	0	1

Tabell 5.1: Tabellen visar användarens uppgifter

Namn	Anvandarnamn	Telefonnummer	skap0	skap1	skap2	skap3	skap4
Devex	Devex1	722855837	2020-06-14 20:31:26	2020-06-14 20:31:26	2020-06-14 20:31:26	2020-06-14 20:31:26	2020-06-14 20:31:26
Devex	Devex2	724565836	2020-06-17 23:10:39	2020-06-14 20:34:05	2020-06-17 23:12:03	2020-06-14 20:34:05	2020-06-16 20:08:33

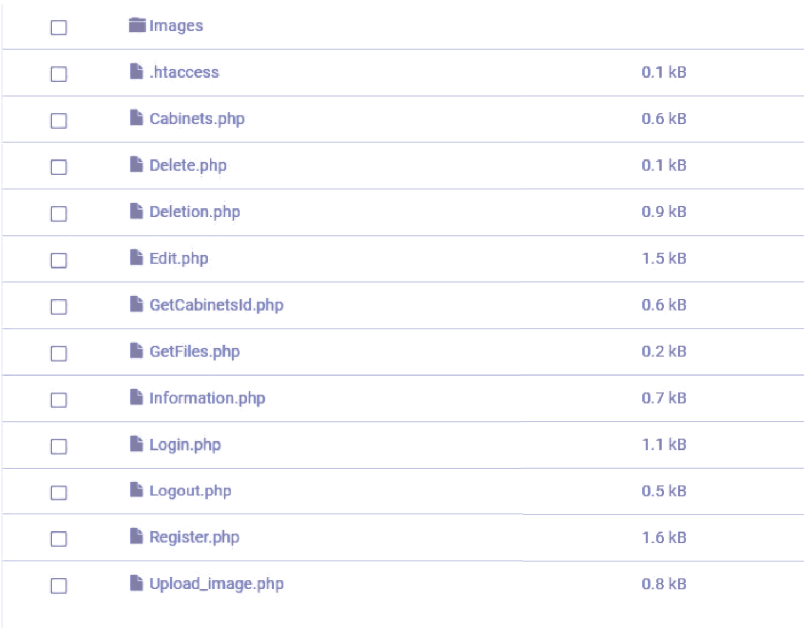
Tabell 5.2: Tabellen visar när användaren har öppnat ett skåp vid ett visst tidpunkt

Namn, användarnamn och lösenord för den registrerade användaren och är av typen VARCHAR med en maxlängd på 50 tecken. **id** är av typen INTEGER och anger ifall en person bara är en vanlig användare eller en administratör. **id** kan innehålla värdet 1 eller 0, där 1 innebär en administratör och 0 en användare. **skap0** upp till **skap10** är också av typen INTEGER och kan även dom endast

anta värdena 1 eller 0. I detta fallet anger värdet 1 att personen har behörighet att öppna skåpet och värdet 0 att personen inte har det.

5.3 Applikation till server - kommunikation

Kommunikationen mellan Android-applikationen och SQL-servern sköts med hjälp PHP-filer som lagras i databasens filhanterare. I filhanteraren finns även mappen “Images” i vilken alla bilder tagna av användarna lagras. Filhanteraren visas i figur 5.10.



Figur 5.10: Figuren visar filhanteraren på 000webhost

5.3.1 Inloggning och registrering

```php
<?php

date_default_timezone_set("Europe/Stockholm");

$date_add= date("Y-m-d G:i:s");

 $con=mysqli_connect("localhost","id12375893_arian23","arian123","id12375893_arian");

 $anvandarnamn = $_POST["Anvandarnamn"];
 $password = $_POST["Lösenord"];

echo"";

 $sql = "SELECT * FROM User WHERE Anvandarnamn = '$anvandarnamn'
 AND Lösenord= '$password'";
 $sql2 = "SELECT * FROM User WHERE Anvandarnamn='$anvandarnamn'
 AND Lösenord= '$password' AND Id= 1 " ;

$result2 = mysqli_query($con,$sql2);
$result =mysqli_query($con,$sql);

if($result2->num_rows > 0 ){

  echo "Du är nu inloggad som Admin";
  $sql123 = "UPDATE User SET last_login = '$date_add'
  WHERE Anvandarnamn='$anvandarnamn' AND Lösenord= '$password' AND Id= 1";
  mysqli_query($con,$sql123);

}

else if($result->num_rows > 0){

    echo "Du är nu inloggad som Användare";
    $sql1234 = "UPDATE User SET last_login= '$date_add'
    WHERE Anvandarnamn= '$anvandarnamn' AND Lösenord= '$password'";
    mysqli_query($con,$sql1234);

}

else{
    echo "Ingen användare hittades var vänlig och försök igen";
}

?>
```

Figur 5.11: Figuren visar PHP-filen “Login.php”.

Filen upprättar anslutning med SQL-databasen samt tar emot användarnamn och lösenord från användaren och jämför dessa med de personer som finns registrerade i databasen. If- och else-satserna i slutet av koden tar hand om fallen då en administratör loggar in, en användare loggar in samt då ingen användare med angivna inloggningsuppgifter finns registrerad. Datum och tid för inloggning registreras här med hjälp av variabeln $date_add vilken används för att uppdatera senaste inloggning för personen och sätta in värdet i databasen. Hela filen visas i figur 5.11.

```php
<?php

date_default_timezone_set("Europe/Stockholm");
$date_add= date("Y-m-d G:i:s");

 $con=mysqli_connect("localhost","id12375893_arian23","arian123","id12375893_arian");

 $name = $_POST["Namn"];
 $Efternamn = $_POST["Efternamn"];
 $anvandarnamn = $_POST["Anvandarnamn"];
 $password = $_POST["Lösenord"];
 $telefonnummer = $_POST["Telefonnummer"];
 $id=$_POST["Id"];
 $skap0 = $_POST["skap0"];
 $skap1 = $_POST["skap1"];
 $skap2 = $_POST["skap2"];
 $skap3 = $_POST["skap3"];
 $skap4 = $_POST["skap4"];
 $skap5 = $_POST["skap5"];
 $skap6 = $_POST["skap6"];
 $skap7 = $_POST["skap7"];
 $skap8 = $_POST["skap8"];
 $skap9 = $_POST["skap9"];
 $skap10 = $_POST["skap10"];



$sql = "INSERT INTO User(Namn,Efternamn,Telefonnummer,Anvandarnamn,Lösenord,Id,
skap0,skap1,skap2,skap3,skap4,skap5,skap6,skap7,skap8,skap9,skap10) VALUES
('$name','$Efternamn','$telefonnummer','$anvandarnamn','$password','$id','$skap0','$skap1',
'$skap2','$skap3','$skap4','$skap5','$skap6','$skap7','$skap8','$skap9','$skap10')";

$sql2 = "INSERT INTO Detaljer(Namn,Anvandarnamn,skap0,skap1,skap2,skap3,
skap4,skap5,skap6,skap7,skap8,skap9,skap10) VALUES
('$name','$anvandarnamn','$date_add','$date_add','$date_add','$date_add','$date_add',
'$date_add','$date_add','$date_add','$date_add','$date_add','$date_add')";


 $result =  mysqli_query($con,$sql);

 $result2 =  mysqli_query($con,$sql2);

if($result && $result2  ){
    echo "Registeringen genomfördes korrekt";
}
else{
    echo "Registreringen kunde inte genomföras";
}

?>
```

Figur 5.12: Figuren visar PHP-filen “Register.php”.

Även denna fil (som visas i figur 5.12) upprättar först en uppkoppling med databasen med angivna värden för namn och lösenord. Filen tar sedan emot de värden som administratören specificerat på de olika parametrarna i applikationen för den person som ska registreras och sätter med hjälp av SQL-kod in dessa i de båda tabellerna. variabeln $sql1 gör detta för användaruppgifterna och $sql2 för skåpsvärdena. if- och else-satserna skickar tillbaka bekräftelser till administratören om registreringen genomfördes eller inte.

5.3.2 Borttagning och redigering

```php
<?php

$con = mysqli_connect("localhost","id12375893_arian23","arian123","id12375893_arian");

$name =$_POST["namn"];

$anvandarnamn=$_POST["anvandarnamn"];

$telefonnummer=$_POST["telefonnummer"];

$sql2 = "SELECT * FROM User WHERE Namn ='$name'
AND Telefonnummer ='$telefonnummer' AND Anvandarnamn ='$anvandarnamn'";
$sql = "DELETE FROM User WHERE Namn ='$name'
AND Telefonnummer ='$telefonnummer' AND Anvandarnamn ='$anvandarnamn' " ;
$sql3 = "DELETE FROM Detaljer WHERE Namn='$name'
AND Anvandarnamn ='$anvandarnamn' " ;

$result2 = mysqli_query($con,$sql2);
$result = mysqli_query($con,$sql);
$result3 = mysqli_query($con,$sql3);


if(($result2->num_rows > 0)&&$result3&&$result){

echo "Personen har tagits bort";
}

else{

echo "Kunde inte ta bort personen var vänligen och försök igen";
}

  mysqli_close($con);
?>
```

Figur 5.13: Figuren visar PHP-filen “Deletion.php”.

Filen ansluter först till servern med uppgifter via $con. Därefter tar den emot namn, användarnamn samt telefonnummer som administratören matat in i applikationen för personen som ska tas bort. Sedan tas först personen bort från den första tabellen (Se Tabell 5.1) och efter det från den andra tabellen (Se Tabell 5.2). Slutligen skickas en bekräftelse till administratören ifall personen togs bort eller ej. Hela filen visas i figur 5.13.

```php
<?php

$con=mysqli_connect("localhost","id12375893_arian23","arian123","id12375893_arian");

 $old_name=$_POST["old_name"];
 $old_telefonnummer=$_POST["old_telefonnummer"];
 $name = $_POST["Namn"];
 $efternamn =  $_POST["Efternamn"];
 $anvandarnamn = $_POST["Anvandarnamn"];
 $lösenord = $_POST["Lösenord"];
 $telefonnummer =  $_POST["Telefonnummer"];
 $id=$_POST["Id"];
 $skap0 = $_POST["skap0"];
 $skap1 = $_POST["skap1"];
 $skap2 = $_POST["skap2"];
 $skap3 = $_POST["skap3"];
 $skap4 = $_POST["skap4"];
 $skap5 = $_POST["skap5"];
 $skap6 = $_POST["skap6"];
 $skap7 = $_POST["skap7"];
 $skap8 = $_POST["skap8"];
 $skap9 = $_POST["skap9"];
 $skap10 = $_POST["skap10"];

$sql = "UPDATE User SET Namn = '$name',Efternamn ='$efternamn',
Telefonnummer= '$telefonnummer',Anvandarnamn='$anvandarnamn',
Lösenord = '$lösenord',Id ='$id',skap0='$skap0',skap1='$skap1',
skap2='$skap2',skap3='$skap3',skap4='$skap4',skap5='$skap5',
skap6='$skap6',skap7='$skap7',skap8='$skap8',skap9= '$skap9',
skap10='$skap10' WHERE Namn='$old_name' AND '$old_telefonnummer'=Telefonnummer  ";

$result = mysqli_query($con,$sql);

if($result){
echo "Redigering utav användaren $name genomfördes korrekt ";

}
else{

echo "Redigeringen kunde inte genomföras";
}
?>
```

Figur 5.14: Figuren visar PHP-filen “Edit.php”.

Filen upprättar anslutning med databasen och tar därefter emot namn och telefonnummer som administratören angivit i applikationen för personen som ska redigeras samt all information om vad personens data ska ändras till. Detta skrivs sedan i SQL-kod som utför och uppdaterar informationen om personen. Slutligen ges även här en respons på om redigeringen har genomförts. Hela filen visas i figur 5.14.

5.3.3 Hämta information

```php
<?php

$con=mysqli_connect("localhost","id12375893_arian23","arian123","id12375893_arian");

$namn  = $_GET['Namn'];

$sql = "select * from User where Namn = '$namn'";

$res = mysqli_query($con,$sql);
$result = array();

if($res->num_rows > 0 ){

while($row = mysqli_fetch_array($res)){

array_push($result,array('Id'=>$row[7],'last_logut'=>$row[6],
'last_login'=>$row[5],'Lösenord'=>$row[4],

'Anvandarnamn'=>$row[3],
'Telefonnummer'=>$row[2],
'Efternamn'=>$row[1],
'Namn'=>$row[0]

));
}

echo json_encode(array("result"=>$result));
mysqli_close($con);
}
else{
    echo "Hittar ej personen";

mysqli_close($con);
}

?>
```

Figur 5.15: Figuren visar PHP-filen “Information.php”.

Denna PHP-fil ansluter till databasen och hämtar med hjälp av en while-loop all information om alla personer vars namn motsvarar det som angivits av administratören i applikationen. Datan returneras tillbaka till applikationen och administratören meddelas om någon person ej hittats. Hela filen visas i figur 5.15.

```php
<?php
$search = $_POST["ser"];
$allfiles = scandir('Images/');
foreach ($allfiles as $file) {
    if (strstr($file, $search)) {
        //unlink ("Images/$file");
        echo "https://arian23.000webhostapp.com/Images/$file ";
    }
}
?>
```

Figur 5.16: Figuren visar PHP-filen "GetFiles.php".

Filen tar emot ett sökord angivet av administratören i applikationen och specificerar en katalog var den ska söka. Detta sker med variablerna $search och $allfiles. Med hjälp av en foreach-sats söks hela katalogen innehållande bilderna igenom och en sträng returneras som innehåller URL-adresser till alla bilder som hittades. Anledningen till att en URL-adress returneras är för att göra det enklare då bilderna ska presenteras i applikationen. Hela filen visas i figur 5.16.

Efter att en bild hämtats av administratören i applikationen kan denna bild även tas bort av administratören. För detta används PHP-koden som visas i figur 5.17.

```php
<?php
$search = $_POST["delete"];
    unlink ("Images/$search");
    echo "bild borttagen";
?>
```

Figur 5.17: Figuren visar PHP-filen "Delete.php".

Denna fil tar emot URL-adressen för den bild som ska tas bort och tar därefter bort bilden från katalogen som finns i filhanteraren i databasen.

```php
<?php

$con=mysqli_connect("localhost","id12375893_arian23","arian123","id12375893_arian");

 $name = $_POST["name"];

 $sql = "SELECT * FROM User WHERE Anvandarnamn = '$name'";

 $result =  mysqli_query($con,$sql);

while ($row = $result->fetch_assoc()) {
    echo $row['skap0']." ";
    echo $row['skap1']." ";
    echo $row['skap2']." ";
    echo $row['skap3']." ";
    echo $row['skap4']." ";
    echo $row['skap5']." ";
    echo $row['skap6']." ";
    echo $row['skap7']." ";
    echo $row['skap8']." ";
    echo $row['skap9']." ";
    echo $row['skap10']." ";
}

?>
```

Figur 5.18: Figuren visar PHP-filen “Cabinets.php”.

Även den här filen upprättar först en uppkoppling med databasen och tar sedan emot namnet på den användare som man vill se behörigheterna för. En while-loop används därefter för att hämta behörigheterna på varje skåp. Hela filen visas i figur 5.18.

5.3.4 Scanning av QR-kod

```php
<?php

date_default_timezone_set("Europe/Stockholm");
$date_add= date("Y-m-d G:i:s");

$con=mysqli_connect("localhost","id12375893_arian23","arian123","id12375893_arian");

$name = $_POST["names"];
$IDS = $_POST["SKAP"];

$sq = "SELECT * FROM User WHERE  Anvandarnamn='$name' AND $IDS = 1 ";

$result =  mysqli_query($con,$sq);

 if($result->num_rows > 0 ){

 echo "Du kommer nu att anslutas";

 $sql234 = "UPDATE Detaljer SET $IDS = '$date_add', senast_inloggad = '$date_add' WHERE Anvandarnamn = '$name'
 mysqli_query($con,$sql234);

 }
 else{

    echo "Du har ej behörighet";
}

?>
```

Figur 5.19: Figuren visar PHP-filen “GetCabinetsId.php”.

Filen upprättar anslutning med databasen och kontrollerar ifall användaren har behörighet att öppna skåpet. Om användaren har behörighet kommer denna att anslutas och databasen uppdateras med den nya senaste tiden som användaren loggade in. Hela filen visas i figur 5.19.

5.3.5 Bilduppladdning och Utloggning

```php
<?php

$target_dir = "Images/";
$target_file_name = $target_dir .basename($_FILES["file"]["name"]);
$response = array();

// Check if image file is a actual image or fake image
if (isset($_FILES["file"])) {

 if (move_uploaded_file($_FILES["file"]["tmp_name"], $target_file_name)) {

  $success = true;
  $message = "Uppladning genomfördes";

  $response["success"] = $success;
  $response["message"] = $message;
 }
 else {

  $success = false;
  $message = "Fel under uppladdningen";

  $response["success"] = $success;
  $response["message"] = $message;
 }
}
else {

 $success = false;
 $message = "Misslyckades med att hitta bild";

 $response["success"] = $success;
 $response["message"] = $message;
}
echo json_encode($response);

?>
```

Figur 5.20: Figuren visar PHP-filen “Upload_image.php”.

Filen innehåller först variabeln $target_dir som används för att sätta en specifik katalog. Den kontrollerar sedan att bildfilen verkligen är en fil och därefter laddar den upp bilden. Ifall något blir fel skickas respons för detta till applikationen där ett felmeddelande dyker upp. Hela filen visas i figur 5.20.

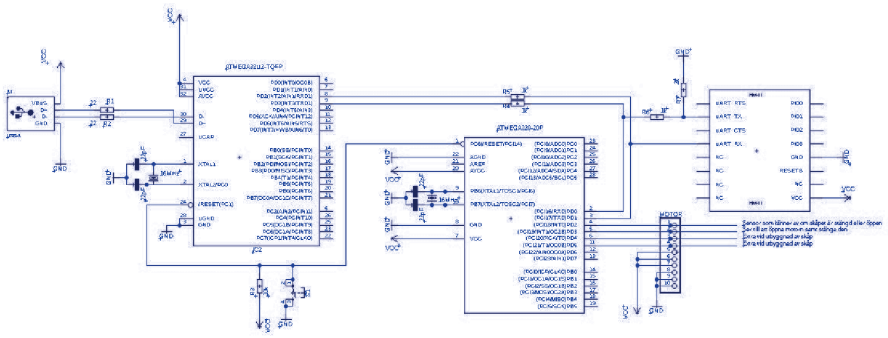
```php
<?php

date_default_timezone_set("Europe/Stockholm");
$date_add= date("Y-m-d G:i:s");

$con=mysqli_connect("localhost","id12375893_arian23","arian123","id12375893_arian");
$namn  = $_POST['Namn'];

$sql = "select * from User where Anvandarnamn = '$namn'";
$result = mysqli_query($con,$sql);

if($result->num_rows > 0){

 $sql234 = "UPDATE User SET last_logut = '$date_add' where Anvandarnamn = '$namn'";

 mysqli_query($con,$sql234);

 echo "Utloggad";

}
?>
```

Figur 5.21: Figuren visar PHP-filen “Logout.php”.

Här lagras datum och tid för utloggning i variabler och därefter skapas en anslutning med databasen. Tabellen “User” (Se Tabell 5.1) uppdateras sedan med den senaste utloggningstiden för personen och en respons “Utloggad” ges i applikationen. Hela filen visas i figur 5.21.

5.4 PCB-kortet

PCB-kortet som användes för systemet utvecklades i EAGLE och såg ut på följande sätt:



Figur 5.22: Figuren visar schemat för PCB-kortet.

Komponenterna på kortet sett från vänster i figur 5.22 består av en USB-ingång typ A kopplad till mikrokontrollern ATMEGA32U2 som endast används för hämtning av Bootloader via Arduinos programvara. Mikrokontrollern är sedan kopplad till en andra mikrokontroller med beteckningen ATMEGA328 som är huvudkomponenten för PCB-kortet där alla systemfunktioner körs. I denna huvudkomponent kopplas en Bluetooth-modul med beteckning HM-10. ATMEGA328 är även sammanlänkad med en kopplingsplint som används för att strömförsörja motorn till låsmekanismen och även en givare som kontrollerar om skåpet är öppet/stängt. Figur 5.23 visar komponenterna för PCB-kortet.

Part	Value	Device	Package	Description	MF	MPN	OC_FARNELL	OC_NEWARK	POPULARITY	PROD_ID	SPICEPREFIX
C1		C-EUC0805	C0805	CAPACITOR, European symbol				88	C		
C2		C-EUC0805	C0805	CAPACITOR, European symbol				88	C		
C3		C-EUC0805	C0805	CAPACITOR, European symbol				88	C		
C4		C-EUC0805	C0805	CAPACITOR, European symbol				88	C		
IC1	ATMEGA328-20P	ATMEGA328-20P	DIL28-3	MICROCONTROLLER							
IC2	ATMEGA32U2-TQFP	ATMEGA32U2-TQFP	TQFP32-08	ATmega32u2							
MOTOR		PINHD-1X8-3.5MM	1X08-3.5MM	PIN HEADER							
Q1		CRYSTALHC49U-H	HC49U-H	CRYSTAL			1666973	unknown			
Q2		CRYSTALHC49U-H	HC49U-H	CRYSTAL			1666973	unknown			
R1	22	R-EU_R0805	R0805	RESISTOR, European symbol				86	R		
R2	22	R-EU_R0805	R0805	RESISTOR, European symbol				86	R		
S1		MOMENTARY-SWITCH-SPST-2-SMD-4.6X2.8MM	TACTILE_SWITCH_SMD_4.6X2.8MM	Momentary Switch (Pushbutton) - SPST - Two Circuits							SWCH-13065
U$1	HM-10	HM-10	HM-10	JNHuaMao HM-11 Bluetooth 4.0 Transceiver							
X1	USB-A	USB-A	MOLEX_480371000	USB type A 'plug'							

Figur 5.23: Figuren visar komponentlistan för PCB-kortet.

6. Systemändringar och vidareutveckling

Detta kapitel kommer redovisa ur administratörssynpunkt vad som ska justeras och ändras på då man vill vidareutveckla systemet eller göra systemändringar beroende på kunders krav och behov.

6.1 Ändra antal skåp

Eftersom systemet bygger på att man vid installation har ett förvalt antal skåp behövs ändringar göras i koderna för applikationen och PHP-filerna för att kunna lägga till fler skåp i systemet. Tabellerna i databasen måste också ändras för att anpassas till det önskade systemet. Denna del av kapitlet kommer visa vad som ska ändras i databasen och i de olika kodfilerna för att man ska kunna lägga till eller ta bort skåp i systemet.

6.1.1 Databasen

Som det tidigare redovisats i rapporten lagras all information i två tabeller i databasen. Eftersom båda tabellerna lagrar information om skåpen så måste även båda tabellerna ändras när man ska lägga till eller ta bort skåp. Genom att använda funktionen ”redigera” i phpmyadmin kan man ändra vilka kolumner som finns med i tabellerna. Man kan då manuellt lägga till eller ta bort skåp (genom inmatning i SQL-kod). Om man istället vill att phpmyadmin ska sköta SQL-koden åt en så kan man välja funktionen ”struktur” vilket tillåter att man lägger till fler kolumner ur en meny.

6.1.2 PHP-filerna

En del av PHP-filerna använder namnen för de olika skåpen för att utföra olika uppgifter i databasen bland annat vid exempelvis sökfunktionen i applikationen och vid registrering av en ny person. De PHP-filerna som behöver editeras är *Cabinets.php, Edit.php* och *Register.php.*

Cabinets.php innehåller en while-loop där information om vilka skåp en specifik användare har åtkomst till. While-loopen visas i figur 6.1.

```
11  while ($row = $result->fetch_assoc()) {
12      echo $row['skap0']." ";
13      echo $row['skap1']." ";
14      echo $row['skap2']." ";
15      echo $row['skap3']." ";
16      echo $row['skap4']." ";
17      echo $row['skap5']." ";
18      echo $row['skap6']." ";
19      echo $row['skap7']." ";
20      echo $row['skap8']." ";
21      echo $row['skap9']." ";
22      echo $row['skap10']." ";
23  }
```

Figur 6.1: Figuren visar while-loopen i PHP-filen Cabinets.php.

Det enda som behövs justeras i den här filen är att lägga till ett echo för nästa skåp på en ny rad efter skåp nummer 10 enligt **echo $row['skap11']." ";** för ett tolfte skåp.

I filen Edit.php behöver man göra ändringar på två ställen i filen. Det första stället är i början av filen där variablerna deklareras. För att lägga till fler skåp måste dessa deklareras efter de föregående skåpen vilket visas i figur 6.2.

```
14  $skap0 = $_POST["skap0"];
15  $skap1 = $_POST["skap1"];
16  $skap2 = $_POST["skap2"];
17  $skap3 = $_POST["skap3"];
18  $skap4 = $_POST["skap4"];
19  $skap5 = $_POST["skap5"];
20  $skap6 = $_POST["skap6"];
21  $skap7 = $_POST["skap7"];
22  $skap8 = $_POST["skap8"];
23  $skap9 = $_POST["skap9"];
24  $skap10 = $_POST["skap10"];
25  $skap11 = $_POST["skap11"];
```

Figur 6.2: Figuren visar den överstrukna koden som ska läggas till vid tilläggning av ett nytt skåp i deklarationen.

Nästa ställe som behöver ändras är den SQL-sträng som uppdaterar användaruppgifter i databasen vid redigering. Detta visas i figur 6.3.

```
28
29  $sql = "UPDATE User SET Namn = '$name',Efternamn ='$efternamn',
30          Telefonnummer= '$telefonnummer',Anvandarnamn='$anvandarnamn',
31          Lösenord = '$lösenord',Id ='$id',skap0='$skap0',skap1='$skap1',skap2='$skap2',skap3='$skap3',skap4='$skap4',
32          skap5='$skap5',skap6='$skap6',skap7='$skap7',skap8='$skap8',skap9= '$skap9',skap10='$skap10', skap11='$skap11'
33          WHERE Namn='$old_name' AND '$old_telefonnummer'=Telefonnummer  ";
34
35
```

Figur 6.3: Figuren visar den överstrukna koden som ska läggas till vid tilläggning av ett nytt skåp i filen Edit.php.

Den sista PHP-filen som måste ändras är *Register.php* och det behövs göras på tre ställen eftersom skåpen finns med i både deklaration och vid insättning i de båda tabellerna. För att lägga till ett nytt skåp i deklarationen får man göra på samma sätt som visas i figur 6.2. De andra två ställena i *Register.php* där kod ska läggas till är överstrukna och visas i figur 6.4.

```
28  $sql = "INSERT INTO User(Namn,Efternamn,Telefonnummer,Anvandarnamn,Lösenord,Id,
29  skap0,skap1,skap2,skap3,skap4,skap5,skap6,skap7,skap8,skap9,skap10,skap11) VALUES
30  ('$name','$Efternamn','$telefonnummer','$anvandarnamn','$password','$id','$skap0','$skap1',
31  '$skap2','$skap3','$skap4','$skap5','$skap6','$skap7','$skap8','$skap9','$skap10','$skap11')";
32
33  $sql2 = "INSERT INTO Detaljer(Namn,Anvandarnamn,Telefonnummer,
34  skap0,skap1,skap2,skap3,skap4,skap5,skap6,skap7,skap8,skap9,skap10,skap11) VALUES
35  ('$name','$anvandarnamn','$telefonnummer','$date_add','$date_add','$date_add','$date_add','$date_add',
36  '$date_add','$date_add','$date_add','$date_add','$date_add','$date_add','$date_add')";
37
```

Figur 6.4: Figuren visar den överstrukna koden som behövs läggas till på de andra två ställena i Register.php.

6.2 Ändra användaruppgifter

Om man vill ändra uppgifter för användare och administratörer av systemet krävs det även där en del ändringar av kodfilerna. Även här måste databasen justeras och ändringar i kodfilerna genomföras. De nedanstående delkapitlen kommer redogöra för hur detta fungerar.

6.2.1 Databasen

Det enda som behöver justeras här är att i phpmyadmin ändra de uppgifter som lagras i Tabell 5.1. I det fall namn, användarnamn eller telefonnummer ska ändras så måste detta också göras i Tabell 5.2 eftersom även denna tabell innehåller dessa uppgifter.

6.2.2 PHP-filerna

De flesta PHP-filerna till systemet innehåller kod för att hantera någon information om användaruppgifterna men hur många filer som behöver justeras beror givetvis på vad man vill ändra på. Om man vill lägga till fler användaruppgifter behöver man ändra PHP-filerna *Register.php*, *Edit.php* och *Information.php*.

I filen *Register.php* läggs önskade uppgifter till i deklarationen. Beroende på vilka uppgifter man vill spara får man ändra SQL-förfrågningarna *$sql* eller *$sql2*. Den första förfrågningen lägger in värdena i Tabell 5.1 och den andra lägger in i Tabell 5.2. Exempelvis är personnummer mest lämpligt att spara i Tabell 5.1 vilket innebär en ändring i *$sql.* Om man istället vill till exempel

lägga till en variabel som talar om hur gånger en användare har varit inloggad så passar det bäst att lagra denna i Tabell 5.2 vilket innebär en ändring i *$sql2*.

Därefter måste filen *Edit.php* ändras och detta görs på precis samma sätt som *Register.php*.

I filen *Information.php* hanteras den information som ska visas för en användare då man söker efter personen. Om man vill att något av det man lagt till ska presenteras i applikationen måste detta ändras i denna PHP-fil. Detta ska göras i den vektor som lagrar värdena och för att göra detta får man lägga till ytterligare en plats i vektorn som visas i figur 6.5.

```
18  array_push($result,array('Id'=>$row[7],'last_logut'=>$row[6],'last_login'=>$row[5],'Lösenord'=>$row[4],
19
20  'Anvandarnamn'=>$row[3],
21  'Telefonnummer'=>$row[2],
22  'Efternamn'=>$row[1],
23  'Namn'=>$row[0]
24
25  ));
26  }
```

Figur 6.5: Figuren visar vektorn som ska ändras i filen Information.php.

6.2.3 Xml-filerna

Vid den här systemändringen behöver även en del ändras i Android Studio för att programmet ska fungera korrekt. För att all data om användarna och administratörerna av systemet ska kunna läggas till vid registrering måste Xml-filen *activity_register.xml* ändras. Om några nya användaruppgifter lagts till i databasen som administratören vill kunna ange vid registrering av ny användare behöver ett nytt fält av typen EditText skapas och läggas in i *activity_register.xml*. Vid kommentar 1 i figur 6.6 väljs ett id för EditText fältet med liknande namn som det attribut som lagts till i databasen för att underlätta programmeringen. Vid kommentar 2 väljs i figur 6.6 vad som ska visas i applikationen för vad administratören ska mata in.

```
<EditText
    android:id="@+id/ed_telefonnummer"     //1
    android:layout_width="match_parent"
    android:layout_height="35dp"
    android:layout_marginTop="10dp"
    android:hint="Telefonnummer"           //2
    android:inputType="number"
    android:textSize="14sp" />
```

Figur 6.6: Figuren visar ett exempel på ett EditText fält.

Om man vill göra det möjligt att editera de nya användaruppgifterna måste fält för EditText läggas in även i xml-filen för editering av användaruppgifter vilket är *activity_edit.xml*. Detta görs på samma sätt som i figur 6.6.

Slutligen finns även xml-filen *activity_viewadapter1.xml* som presenterar information om en person då man söker efter personen. Om man vill att den nya informationen ska presenteras här måste man lägga till en TextView. Vid kommentar 1 i figur 6.7 väljs ett liknande namn som det attribut som lagts till i databasen.

```
<TextView
    android:id="@+id/Telefonnummer"        //1
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:textColor="#FDFAFA"
    android:layout_gravity="left"
    android:layout_marginLeft="3dp"
    android:layout_marginTop="10dp"
    android:textAppearance="@style/TextAppearance.AppCompat" />
```

Figur 6.7: Figuren visar ett exempel på ett TextView fält.

6.2.4 Java-filerna

När Xml-filerna är ändrade måste även java-filerna anpassas för den nya informationen. För detta måste ändringar göras i filerna *Register.java* och *Edit.java.* Alternativt även *Information.java* och *ViewAdapter1.java* då man vill presentera den nya informationen vid sökning. I *Register.java* måste först objekt skapas av typen EditText samt String. Detta görs efter de befintliga objekten i deklarationen som visas i figur 6.8.

```
37      private String url = "https://arian23.000webhostapp.com/Register.php";
38      private EditText Ed_username, Ed_name, Ed_lastname, Ed_telephonenumber, Ed_password;
39      private String str_name, str_username, str_password, str_lastname, str_telephonenumber;
```

Figur 6.8: Figuren visar deklarationen av variabler i filen Register.java.

I metoden *InitiateVariables* deklareras vilket EditText-objekt som motsvarar ett visst EditText-fält i *activity_register.xml.* Det nya EditText-fältet måste initieras här. Detta visas i figur 6.9.

```
62      // Initierar variablerna
63      private void InitiateVariables() {
64          navigation = (BottomNavigationView) findViewById(R.id.navigation);
65          Ed_name = findViewById(R.id.ed_namn);
66          Ed_lastname = findViewById(R.id.ed_efternamn);
67          Ed_username = findViewById(R.id.ed_email);
68          Ed_telephonenumber = findViewById(R.id.ed_telefonnummer);
69          Ed_password = findViewById(R.id.ed_lösenord);
70          Ed_Group = (RadioGroup) findViewById(R.id.ed_group);
71          Ed_admin = (RadioButton) findViewById(R.id.ed_adm);
72          Ed_user = (RadioButton) findViewById(R.id.ed_anv);
73          BtnCabinets = (Button) findViewById(R.id.ed_skåp);
74      }
```

Figur 6.9: Figuren visar initiering av variabler i filen Register.java

I listan av Toasts som meddelar administratören om att viss information inte har angetts kan det vara lämpligt att lägga till en ny Toast för den nya informationen som ska matas in vid registrering. Figur 6.10 visar var detta ska läggas till.

```
177     if (Ed_name.getText().toString().equals("")) {
178         Toast.makeText(this, "Glömt mata in Namn", Toast.LENGTH_SHORT).show();
179     } else if (Ed_lastname.getText().toString().equals("")) {
180         Toast.makeText(this, "Glömt mata in Efternamn", Toast.LENGTH_SHORT).show();
181     } else if (Ed_telephonenumber.getText().toString().equals("")) {
182         Toast.makeText(this, "Glömt mata in Telefonnummer", Toast.LENGTH_SHORT).show();
183     } else if (Ed_username.getText().toString().equals("")) {
184         Toast.makeText(this, "Glömt mata in Användarnamn", Toast.LENGTH_SHORT).show();
185     } else if (Ed_password.getText().toString().equals("")) {
186         Toast.makeText(this, "Glömt mata in lösenord", Toast.LENGTH_SHORT).show();
187     } else if (Ed_Group.getCheckedRadioButtonId() == -1) {
188         Toast.makeText(this, "Glömt mata in konto", Toast.LENGTH_SHORT).show();
189     } else {
190
191         progressDialog.show();
192         str_name = Ed_name.getText().toString().trim();
193         str_lastname = Ed_lastname.getText().toString().trim();
194         str_telephonenumber = Ed_telephonenumber.getText().toString().trim();
195         str_username = Ed_username.getText().toString().trim();
196         str_password = Ed_password.getText().toString().trim();
```

Figur 6.10: Figuren visar listan av toasts i filen Register.java.

När all information slutligen ska skickas till php-filen som kommunicerar med databasen görs detta genom att alla värden lagras i en Map. Den sträng som finns i objektet Map måste stämma överens med namnet på variabeln i php-filen *Register.php.* Exempelvis måste “Namn” som visas i figur 6.11 stämma överens med det som enligt figur 5.12 finns deklarerat på rad 8 som $_POST[“Namn”].

```
225                 params.put("Namn", str_name);
226                 params.put("Efternamn", str_lastname);
227                 params.put("Telefonnummer", str_telephonenumber);
228                 params.put("Anvandarnamn", str_username);
229                 params.put("Lösenord", str_password);
```

Figur 6.11: Figuren visar objektet Map i filen Register.java.

I filen *Edit.java* görs ändringar på samma sätt som i *Register.java.* Det enda som skiljer sig åt är radnumren.

För att presentera den nya informationen som man vill lägga till i applikationen behöver man ändra på filerna *Information.java* samt *ViewAdapter1.java.* I filen *Information.java* som visas i figur 6.12 måste en ny sträng läggas till.

```
38
39      private static final String url = "https://arian23.000webhostapp.com/GetFiles.php";
40      private static final String DATA_URL = "https://arian23.000webhostapp.com/Information.php?Namn=";
41      private static final String Namn = "Namn";
42      private static final String Efternamn = "Efternamn";
43      private static final String Användarnamn = "Anvandarnamn";
44      private static final String Telefonnummer = "Telefonnummer";
45      private static final String Lösenord = "Lösenord";
46      private static final String Senast_inloggad = "last_login";
47      private static final String Senast_utloggad = "last_logut";
```

Figur 6.12: Figuren visar deklarationen i filen Information.java.

När strängen har lagts till måste man även ändra Void-metoden showJSON som finns i *Information.java* vilken hämtar information om en användare från PHP-filen *Information.php* och lagrar detta i en Map. Objektet Map lagras sedan i en Arraylist. Detta visas i figur 6.13.

```
private void showJSON(String response) {

    ArrayList<HashMap<String, String>> list = new ArrayList<~>();
    try {
        JSONObject jsonObject = new JSONObject(response);
        JSONArray result = jsonObject.getJSONArray(JSON_ARRAY);

        for (int i = 0; i < result.length(); i++) {
            JSONObject jo = result.getJSONObject(i);
            String id = jo.getString(ID);

            if (!(id.compareTo(String.valueOf(1)) == 0)) {

                String Namn = jo.getString(Information.Namn);
                String Efternamn = jo.getString(Information.Efternamn);
                String Telefonnummer = jo.getString(Information.Telefonnummer)
                String Anvandarnamn = jo.getString(Användarnamn);
                String Lösenord = jo.getString(Information.Lösenord);
                String lastlogin = jo.getString(Senast_inloggad);
                String lastlogut = jo.getString(Senast_utloggad);

                final HashMap<String, String> employees = new HashMap<>();
                employees.put(Information.Namn, Namn);
                employees.put(Information.Efternamn, Efternamn);
                employees.put(Information.Telefonnummer, Telefonnummer);
                employees.put(Information.Användarnamn, Anvandarnamn);
                employees.put(Information.Lösenord, Lösenord);
                employees.put(Information.Senast_inloggad, lastlogin);
                employees.put(Information.Senast_utloggad, lastlogut);

                list.add(employees);
            }
```

Figur 6.13: Figuren visar metoden showJSON i filen Information.java.

Slutligen ska ViewAdapter1 definieras i filen *Information.java* och här ska den information som ska visas i applikationen finnas med. Figur 6.14 visar deklarationsdelen.

```
216         ViewAdapter1 myadapter = new ViewAdapter1(this, new String[][Namn, Efternamn, Telefonnummer, Användarnamn, Lösenord, Senast_inloggad, Senast_utloggad], list);
217         recyclerView.setLayoutManager(new LinearLayoutManager(this));
218         recyclerView.setAdapter(myadapter);
```

Figur 6.14: Figuren visar deklarationen av ViewAdapter1 i filen Information.java.

När ovanstående steg är gjorde måste till sist filen *ViewAdapter1.java* editeras. I denna fil deklareras alla värden som String och det som ska läggas till måste finnas med här. Detta visas i figur 6.15.

```
20      private static final String Name = "Namn";
21      private static final String Lastname = "Efternamn";
22      private static final String Username = "Anvandarnamn";
23      private static final String Telephonenumber = "Telefonnummer";
24      private static final String Password = "Lösenord";
25      private static final String Lastlogin = "last_login";
26      private static final String Lastlogout = "last_logut";
```

Figur 6.15: Figuren visar deklarationen av ViewAdapter1 i filen ViewAdapter1.java.

Metoden onBindViewHolder vilken visas i figur 6.16 hämtar värden från listan list och lägger in dem på de olika variablerna i vyn. De nya värdena läggs till här med de valda namnen och parametrarna.

```
49      @Override
50      public void onBindViewHolder(@NonNull ViewAdapter1.MyViewHolder holder, final int position) {
51          holder.Name.setText(list.get(position).get(Name));
52          holder.Lastname.setText(list.get(position).get(Lastname));
53          holder.Telephonenumber.setText("Telefonnummer: " + list.get(position).get(Telephonenumber));
54          holder.Username.setText("Användarnamn: " + list.get(position).get(Username));
55          holder.Password.setText("Lösenord: " + list.get(position).get(Password));
56          holder.Lastlogin.setText("Inloggad: " + list.get(position).get(Lastlogin));
57          holder.Lastlogout.setText("Utloggad: " + list.get(position).get(Lastlogout));
```

Figur 6.16: Figuren visar metoden onBindViewHolder i filen ViewAdapter1.java.

Det sista som behöver ändras i denna java-fil är att deklarera TextView-objekten i klassen MyViewHolder samt att deklarera vilket objekt som hör ihop med vilket Id från xml-filen *activity_viewadapter1.xml.* Deklarationen av TextView visas i figur 6.17.

```
76      public class MyViewHolder extends RecyclerView.ViewHolder {
77
78          TextView Name, Lastname, Telephonenumber, Password, Lastlogin, Lastlogout, Username;
79          Button BtnCompetence;
80
81          public MyViewHolder(@NonNull View itemView) {
82              super(itemView);
83
84              Name = itemView.findViewById(R.id.Namn);
85              Lastname = itemView.findViewById(R.id.Efternamn);
86              Telephonenumber = itemView.findViewById(R.id.Telefonnummer);
87              Username = itemView.findViewById(R.id.date);
88              Password = itemView.findViewById(R.id.data);
89              Lastlogin = itemView.findViewById(R.id.tvid);
90              Lastlogout = itemView.findViewById(R.id.utloggad);
91              BtnCompetence = itemView.findViewById(R.id.btnbehörig);
```

Figur 6.17: Figuren visar deklarationen av TextView i filen ViewAdapter1.java.

6.3 Framtida utveckling av systemet

Detta kapitel kommer redovisa några exempel på framtida utvecklingar som skulle kunna genomföras för att optimera systemet. Det finns en del saker man skulle kunna ändra på och lägga till för att göra systemet ännu bättre och smidigare men på grund av en tidsbegränsning kunde inte allt detta genomföras.

Ett förslag på vidareutveckling som diskuterades med företaget var att lägga till fler behörighetsnivåer för systemet. För att applikationen ska veta om en person är en användare eller en administratör finns det en Id-kolumn i databasen som anger detta. Id-kolumnen är av typen integer och i det nuvarande systemet finns det endast två lägen på denna. Värdet kan anta antingen en nolla eller en etta. En nolla innebär att en person är en användare medan en etta innebär en administratör. Om man vill utöka detta skulle man kunna lägga till fler värden som kan innebära exempelvis en gäst i systemet. Detta är dock inte något som ingick i arbetsprocessen och kommer inte att finnas med i examensarbetet.

En annan punkt som togs upp i ett möte med företaget är att det hade varit smidigt om man kunde ändrat antalet skåp direkt genom applikationen istället för att göra detta i databasen. Denna funktion hade underlättat ifall de som köpt systemet har planer på att förändra antalet installerade skåp ofta då de slipper göra några ändringar i databasen eller php-filerna. Eftersom det blir ganska många förändringar som måste göras i detta system för att en sådan funktion ska bli möjlig bestämdes det att detta inte skulle genomföras därför att detta skulle göra att arbetet sträcker sig utanför tidsplanen.

Till sist hade det även varit bra att utveckla en applikation till Apples operativsystem IOS vilket är något som företaget kan komma att göra i framtiden.

7. Manual

För att göra det enkelt att komma åt databasen och filhanteraren kommer denna del att vara en kortfattad manual för just detta.

Som nämnts tidigare lagras databasen på sidan 000webhost.com och url-adressen till denna sida är https://www.000webhost.com/

Efter ett tryck på ”Sign in” uppe till höger ska e-postadress samt lösenord matas in. För examensarbetet har följande uppgifter använts:

E-post: arian_maloku@hotmail.com

Lösenord: *****

Efter inloggning visas en sida med texten ”My Websites”. Under denna finns hemsidan ”arian23” med en knapp ”Manage Website” i mitten. Efter ett tryck på denna visas en ny sida och till vänster finns en lista med olika funktioner. För att komma till filhanteraren välj ”Tools” och därefter ”File Manager”. Tryck sedan på ”Upload Files” och filhanteraren öppnas. I mappen ”public_html” finns alla php-filer samt mappen ”Images” där de uppladdade bilderna lagras.

För att komma till databashanteraren välj ”Database Manager” istället för ”File Manager”. En ny sida visar databasen med en flik ”Manage”. Välj i denna flik ”PhpMyAdmin” och sidan för denna öppnas. Uppe till vänster syns databasen ”id12375893_arian” och om denna expanderas med plustecknet syns tabellerna ”Detaljer” och ”User”.

8. Slutsats

I detta examensarbete har ett låssystem för medicinskåp utvecklats. All information om användare och administratörer av systemet lagras i två tabeller i en SQL-databas. Systemet använder sig av Android, ett PCB-kort med tillhörande powerbank, en servomotor och en SQL-databas.
I Android-applikationen finns möjligheten för administratörer att registrera, editera eller helt ta bort användare av systemet. Administratörer kan även söka på en användare och se all information om denna samt vilka skåp personen senast loggat in på. En användare loggar in i applikationen med sitt användarnamn och lösenord och kan därefter öppna och stänga dörren till de medicinskåp som denna har tillgång till. Innan användaren loggar ut finns möjligheten att ta en bild av skåpets innehåll vilken sedan skickas till databasen. Dessa bilder kan sedan visas för administratörerna i applikationen.

8.1 Reflektion över problemformuleringen

Följande frågor ställdes i början av examensarbetet:

- Vilket system är mest lämpligt utifrån platsutrymme i skåpet?
- Vilket system fungerar bäst för trådlös sammankoppling?
- Vilka komponenter ska väljas till utvecklingen av det nya systemet?
- Hur ska alla komponenter länkas samman med varandra för att skapa den färdiga produkten?
- Vad kommer förbättras med det nya systemet ur säkerhetsaspekt?
- Vilken typ av databas är mest lämplig att använda?

I examensarbetet kunde det konstateras att det system som var mest lämpligt utifrån utrymmessynpunkt var att använda ett PCB-kort med en powerbank som strömkälla. Det PCB-kort som konstruerades till arbetet blev betydligt mindre än den arduino som befann sig i skåpet från början. Svaret på den andra frågan som ställdes blev att det system som visade sig passa bäst för trådlös sammankoppling var att använda en Android-telefon tillsammans med en SQL-databas och en HM-10 bluetooth-modul monterad på ett PCB-kort. Detta svar baseras endast på den information som erhölls vid möten och diskussioner med företaget då någon annan trådlös sammankoppling inte testades. Komponenterna som valdes till det nya systemet var en Android-telefon, ett PCB-kort och en powerbank och de länkas samman på så sätt att Android-applikationen kommunicerar med SQL-databasen och även med PCB-kortet och databasen har därför ingen direkt koppling till PCB-kortet utan all data emellan går igenom applikationen. Det nya systemet som utvecklades i examensarbetet är säkrare eftersom det lagrar informationen i en SQL-databas istället för ett kalkylark på Google sheets. Det finns även funktioner som att användarna tar bilder på innehållet i

skåpen innan de loggar ut samt att administratörerna kan se tid för utloggning för varje användare vilket förbättrar säkerheten för systemet. Vid efterforskning i början av examensarbetet samt i möten med företaget kunde det konstateras att en SQL-databas var mest lämplig att använda för systemet.

8.2 Reflektion över etiska aspekter

Eftersom allt fler använder sig av det digitala systemet i arbetslivet så vill man ta fram olika framtida lösningar som underlättar för personalen på ett företag. Man vill underlätta arbetet så att personen i sig fokuserar mer på det specifika arbetet den är till för och inte lägger allt för mycket av sin tid till administrativt arbete. I examensarbetet har detta tagits hänsyn till då det utvecklade systemet har skapats för att göra det enklare och säkrare att administrera ett medicinskåp till vården. Detta leder till att personalen i vården kan lägga mer resurser på vården och mindre till administration. Det utvecklade systemet har även minimerat risken för borttappade nycklar samt att läkemedel kommer på avvägar. Det kommer även underlätta en hel del för personalen som inte behöver springa runt med många nycklar på sig och hitta rätt nyckel till de olika medicinskåp som finns på exempelvis sjukhus eller vårdcentral. De kan nu enkelt öppna skåpet via Androidapplikationen vilket kommer medföra en väldigt stor kostnadsreducering.

9. Terminologi

Android – Mobilt operativsystem

Caddning – Konstruktionsutveckling i datorprogram

SQL-databas – Databas som använder SQL som programspråk

Bluetooth – Trådlös kommunikation

URL-adress – Uniform Resource Locator, teckensträng som identifierar webbsida

Google Drive – Molntjänst för lagring av filer

QR-kod – Quick Response Code, rutkod som kan användas som identifierare

Bootloader – Programvara som laddar in ett operativsystem

10. Referenser

[1] Books, Chad. Business News Daily. *What is SQL?*. 2014-01-21
https://www.businessnewsdaily.com/5804-what-is-sql.html
(Hämtad 2020-03-10)

[2] Android Studio. *Android Studio Release Notes*. 2020-05-29.
https://developer.android.com/studio/releases/index.html
(Hämtad 2020-03-12)

[3] Java.com. *General Questions.*
https://java.com/en/download/faq/whatis_java.xml
(Hämtad 2020-03-12)

[4] Leahy, Paul. ThoughtCo. *What Is Java?*. 2019-07-03.
https://www.thoughtco.com/what-is-java-2034117
(Hämtad 2020-03-12)

[5] w3schools.com. *Java OOP.*
https://www.w3schools.com/java/java_oop.asp
(Hämtad 2020-03-12)

[6] Guru99.com. *What is PHP?*
https://www.guru99.com/what-is-php-first-php-program.html#2
(Hämtad 2020-03-12)

[7] Php.net. *History of PHP.*
https://www.php.net/manual/en/history.php.php
(Hämtad 2020-03-13)

[8] Guru99.com. *Java vs JavaScript: Most Important Differences You Must Know.*
https://www.guru99.com/difference-between-java-and-javascript.html
(Hämtad 2020-03-13)

[9] Kearney, Georgina. MINT-TEK. *The History of Printed Circuit Boards*. 2016-02-19.
https://mint-tek.com/the-history-of-printed-circuit-boards-2/
(Hämtad 2020-03-14)

[10] Guppy, Nick. MusicRadar. *Handwired vs PCB amps: what's the difference?*. 2016-02-01.
https://www.musicradar.com/news/guitars/handwired-vs-pcb-amps-whats-the-difference-633137
(Hämtad 2020-03-14)

[11] arduino.cc. *What is Arduino?*

https://www.arduino.cc/en/Guide/Introduction
(Hämtad 2020-03-16)

[12] Heble, Soumil. maxEmbedded.com. *PCB Design using EAGLE – Part 1: Introduction to EAGLE and Software Environment.* 2014-06-11.
http://maxembedded.com/2014/06/pcb-design-eagle-part-1-introduction-eagle-software-environment/
(Hämtad 2020-03-16)

[13] phpMyAdmin.net. *Team.*
https://www.phpmyadmin.net/team/
(Hämtad 2020-03-19)

[14] phpMyAdmin.net. ***About.***
https://www.phpmyadmin.net/
(Hämtad 2020-03-19)

11. Appendix

11.1 Java-filer

11.1.1 Login.java

```
1.  package com.example.arian;
2.  import android.app.Activity;
3.  import android.app.ProgressDialog;
4.  import android.content.Intent;
5.  import android.os.Bundle;
6.  import android.view.View;
7.  import android.view.inputmethod.InputMethodManager;
8.  import android.widget.EditText;
9.  import android.widget.Toast;
10. import androidx.appcompat.app.AppCompatActivity;
11. import com.android.volley.AuthFailureError;
12. import com.android.volley.Request;
13. import com.android.volley.RequestQueue;
14. import com.android.volley.Response;
15. import com.android.volley.VolleyError;
16. import com.android.volley.toolbox.StringRequest;
17. import com.android.volley.toolbox.Volley;
18. import java.util.HashMap;
19. import java.util.Map;
20.
21. public class Login extends AppCompatActivity {
22.
23.   private static final String url = "https://arian23.000webhostapp.com/Login.php";
24.   private EditText Ed_Anvandarnamn, Ed_Lösenord;
25.   private String Anvandarnamn, Lösenord;
26.   public static String Username;
27.
28.   //Metoden är till för att starta en activity i sitt första steg samt initiera variabler
29.   @Override
30.   protected void onCreate(Bundle savedInstanceState) {
31.       super.onCreate(savedInstanceState);
32.       setContentView(R.layout.activity_login);
33.
34.       Ed_Anvandarnamn = findViewById(R.id.ed_email);
35.       Ed_Lösenord = findViewById(R.id.ed_lösenord);
36.   }
37.
38.   //Metoden är till för tillbaka knappen på telefonen
39.   public void onBackPressed() {
40.   }
41.
42.  //Metoden är till då användaren vill dra ner tangentbordet automatisk via ett klick på fönstret
43.   public void HideKeyboard(View view) {
44.       InputMethodManager m = (InputMethodManager) getSystemService((Activity.INPUT_METHOD_SERVICE));
45.       m.hideSoftInputFromWindow(view.getWindowToken(), 0);
46.   }
47.   // Metoden är till för inloggning av användaren
48.   public void LogIn(View view) {
49.
```

```
50.      if (Ed_Anvandarnamn.getText().toString().equals("")) {
51.         Toast.makeText(this, "Glömt mata in Användarnamn", Toast.LENGTH_SHORT).show();
52.      } else if (Ed_Lösenord.getText().toString().equals("")) {
53.         Toast.makeText(this, "Glömt mata in Lösenordet", Toast.LENGTH_SHORT).show();
54.      } else {
55.
56.         final ProgressDialog progressDialog = new ProgressDialog(this);
57.         progressDialog.setMessage("Väntar på svar");
58.
59.         progressDialog.show();
60.
61.         Anvandarnamn = Ed_Anvandarnamn.getText().toString().trim();
62.         Lösenord = Ed_Lösenord.getText().toString().trim();
63.
64.         StringRequest request = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
65.            @Override
66.            public void onResponse(String response) {
67.
68.               if (response.equalsIgnoreCase("Du är nu inloggad som Admin")) {
69.
70.                  Ed_Anvandarnamn.setText("");
71.                  Ed_Lösenord.setText("");
72.                  startActivity(new Intent(getApplicationContext(), Register.class));
73.                  Toast.makeText(Login.this, response, Toast.LENGTH_SHORT).show();
74.
75.               } else if (response.equalsIgnoreCase("Du är nu inloggad som Användare")) {
76.
77.                  Ed_Anvandarnamn.setText("");
78.                  Ed_Lösenord.setText("");
79.                  Intent t1 = new Intent(getApplicationContext(), ScanCode.class);
80.                  startActivity(t1);
81.                  Username = Anvandarnamn;
82.                  Toast.makeText(Login.this, response, Toast.LENGTH_SHORT).show();
83.
84.               } else {
85.
86.                  progressDialog.dismiss();
87.                  Toast.makeText(Login.this, "Ingen Användare hittades", Toast.LENGTH_SHORT).show();
88.               }
89.            }
90.         }, new Response.ErrorListener() {
91.
92.            @Override
93.            public void onErrorResponse(VolleyError error) {
94.               progressDialog.dismiss();
95.               Toast.makeText(Login.this, "Konfigurera Internet anslutningen", Toast.LENGTH_SHORT).show();
96.            }
97.         }
98.
99.
100.        ) {
101.           @Override
102.           protected Map<String, String> getParams() throws AuthFailureError {
103.              Map<String, String> params = new HashMap<String, String>();
104.              params.put("Anvandarnamn", Anvandarnamn);
105.              params.put("Lösenord", Lösenord);
106.              return params;
107.
108.           }
109.        };
110.
111.        RequestQueue requestQueue = Volley.newRequestQueue(Login.this);
112.        requestQueue.add(request);
113.     }
114.  }}
115.
```

11.1.2 Register.java

```
1.  package com.example.arian;
2.  import android.app.Activity;
3.  import android.app.AlertDialog;
4.  import android.app.ProgressDialog;
5.  import android.content.DialogInterface;
6.  import android.content.Intent;
7.  import android.os.Bundle;
8.  import android.view.MenuItem;
9.  import android.view.View;
10. import android.view.inputmethod.InputMethodManager;
11. import android.widget.Button;
12. import android.widget.EditText;
13. import android.widget.RadioButton;
14. import android.widget.RadioGroup;
15. import android.widget.Toast;
16. import androidx.annotation.NonNull;
17. import androidx.appcompat.app.AppCompatActivity;
18. import com.android.volley.AuthFailureError;
19. import com.android.volley.Request;
20. import com.android.volley.RequestQueue;
21. import com.android.volley.Response;
22. import com.android.volley.VolleyError;
23. import com.android.volley.toolbox.StringRequest;
24. import com.android.volley.toolbox.Volley;
25. import com.google.android.material.bottomnavigation.BottomNavigationView;
26. import java.util.ArrayList;
27. import java.util.HashMap;
28. import java.util.Map;
29.
30. public class Register extends AppCompatActivity implements View.OnClickListener {
31.
32.     private String url = "https://arian23.000webhostapp.com/Register.php";
33.     private EditText Ed_username, Ed_name, Ed_lastname, Ed_telephonenumber, Ed_password;
34.     private String str_name, str_username, str_password, str_lastname, str_telephonenumber;
35.     private RadioButton Ed_user, Ed_admin;
36.     private RadioGroup Ed_Group;
37.     private Button BtnCabinets;
38.     private BottomNavigationView navigation;
39.     private String[] listItems;
40.     private boolean[] checkedItems;
41.     private ArrayList<Integer> mUserItems;
42.
43.     //Metoden är till för att starta en activity i sitt första steg samt initiera variabler
44.     @Override
45.     protected void onCreate(Bundle savedInstanceState) {
46.         super.onCreate(savedInstanceState);
47.         setContentView(R.layout.activity_register);
48.
```

```
49.     mUserItems = new ArrayList<>();
50.     listItems = getResources().getStringArray(R.array.id_item);
51.     checkedItems = new boolean[listItems.length];
52.     InitiateVariables();
53.     BottomNavigationView();
54.     BtnCabinets.setOnClickListener(this);
55.  }
56.
57.  // Initierar variablerna
58.  private void InitiateVariables() {
59.     navigation = (BottomNavigationView) findViewById(R.id.navigation);
60.     Ed_name = findViewById(R.id.ed_namn);
61.     Ed_lastname = findViewById(R.id.ed_efternamn);
62.     Ed_username = findViewById(R.id.ed_email);
63.     Ed_telephonenumber = findViewById(R.id.ed_telefonnummer);
64.     Ed_password = findViewById(R.id.ed_lösenord);
65.     Ed_Group = (RadioGroup) findViewById(R.id.ed_group);
66.     Ed_admin = (RadioButton) findViewById(R.id.ed_adm);
67.     Ed_user = (RadioButton) findViewById(R.id.ed_anv);
68.     BtnCabinets = (Button) findViewById(R.id.ed_skåp);
69.  }
70.
71.  //Knapptryckningen för att välja olika skåpsbehörigeter
72.  @Override
73.  public void onClick(View v) {
74.
75.     switch (v.getId()) {
76.
77.        case R.id.ed_skåp:
78.           AlertDialog.Builder mBuilder = new AlertDialog.Builder(Register.this);
79.           mBuilder.setTitle("Välj Skåpsbehörigheter för användare");
80.           mBuilder.setMultiChoiceItems(listItems, checkedItems, new DialogInterface.OnMultiChoiceClickListener() {
81.              @Override
82.              public void onClick(DialogInterface dialog, int pos, boolean isChecked) {
83.
84.                 if (isChecked) {
85.                    if (!mUserItems.contains(pos)) {
86.                       mUserItems.add(pos);
87.                    } else {
88.                       mUserItems.remove(pos);
89.                    }
90.                 }
91.              }
92.           });
93.           mBuilder.setCancelable(false);
94.           mBuilder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
95.              @Override
96.              public void onClick(DialogInterface dialog, int which) {
97.                 String item = "";
98.                 for (int i = 0; i < mUserItems.size(); i++) {
99.                    item = item + listItems[mUserItems.get(i)];
100.                }
101.             }
102.          });
103.          mBuilder.setNeutralButton("Radera allt", new DialogInterface.OnClickListener() {
104.             @Override
105.             public void onClick(DialogInterface dialog, int which) {
106.                for (int i = 0; i < checkedItems.length; i++) {
107.                   checkedItems[i] = false;
108.                   mUserItems.clear();
109.                }
110.             }
111.          });
112.          AlertDialog mDialog = mBuilder.create();
113.          mDialog.show();
```

```
            break;
        }
    }

    //De olika knapparna Registrering,Ändring,Borttagning och Information som finns nere för att byta activtyn
    private void BottomNavigationView() {
        navigation.setSelectedItemId(R.id.navigation_Reg);
        navigation.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                switch (item.getItemId()) {

                    case R.id.navigation_Reg:
                        return true;

                    case R.id.navigation_ändra:
                        startActivity(new Intent(getApplicationContext(), Edit.class));
                        overridePendingTransition(0, 0);
                        return true;

                    case R.id.navigation_info:
                        startActivity(new Intent(getApplicationContext(), Information.class));
                        overridePendingTransition(0, 0);
                        return true;

                    case R.id.navigation_tabort:
                        startActivity(new Intent(getApplicationContext(), Deletion.class));
                        overridePendingTransition(0, 0);
                        return true;
                }
                return false;
            }
        });
    }

    //Metoden ser till att hoppa till en ny klass vilket är Inloggning
    public void ToLogin(View view) {
        startActivity(new Intent(getApplicationContext(), Login.class));
        finish();
    }

    //Metoden är till då användaren vill dra ner tangentbordet automatisk via ett klick på fönstret
    public void HideKeyboard(View view) {
        InputMethodManager m = (InputMethodManager) getSystemService((Activity.INPUT_METHOD_SERVICE));
        m.hideSoftInputFromWindow(view.getWindowToken(), 0);
    }

    //Metoden är till för tillbaka knappen på telefonen
    public void onBackPressed() {

    }

    //Metoden ser till att Registrera personuppgifter
    public void Register(final View view) {

        final ProgressDialog progressDialog = new ProgressDialog(this);
        progressDialog.setMessage("Väntar..");

        if (Ed_name.getText().toString().equals("")) {
            Toast.makeText(this, "Glömt mata in Namn", Toast.LENGTH_SHORT).show();
        } else if (Ed_lastname.getText().toString().equals("")) {
            Toast.makeText(this, "Glömt mata in Efternamn", Toast.LENGTH_SHORT).show();
        } else if (Ed_telephonenumber.getText().toString().equals("")) {
            Toast.makeText(this, "Glömt mata in Telefonnummer", Toast.LENGTH_SHORT).show();
        } else if (Ed_username.getText().toString().equals("")) {
            Toast.makeText(this, "Glömt mata in Användarnamn", Toast.LENGTH_SHORT).show();
```

```
180.    } else if (Ed_password.getText().toString().equals("")) {
181.        Toast.makeText(this, "Glömt mata in lösenord", Toast.LENGTH_SHORT).show();
182.    } else if (Ed_Group.getCheckedRadioButtonId() == -1) {
183.        Toast.makeText(this, "Glömt mata in konto", Toast.LENGTH_SHORT).show();
184.    } else {
185.        progressDialog.show();
186.        str_name = Ed_name.getText().toString().trim();
187.        str_lastname = Ed_lastname.getText().toString().trim();
188.        str_telephonenumber = Ed_telephonenumber.getText().toString().trim();
189.        str_username = Ed_username.getText().toString().trim();
190.        str_password = Ed_password.getText().toString().trim();
191.
192.        StringRequest request = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
193.
194.            @Override
195.            public void onResponse(String response) {
196.
197.                progressDialog.dismiss();
198.                Ed_name.setText("");
199.                Ed_lastname.setText("");
200.                Ed_telephonenumber.setText("");
201.                Ed_username.setText("");
202.                Ed_password.setText("");
203.
204.                Toast.makeText(Register.this, response, Toast.LENGTH_SHORT).show();
205.            }
206.        }, new Response.ErrorListener() {
207.
208.            @Override
209.            public void onErrorResponse(VolleyError error) {
210.                progressDialog.dismiss();
211.                Toast.makeText(Register.this, error.getMessage().toString(), Toast.LENGTH_SHORT).show();
212.            }
213.        }
214.        ) {
215.            @Override
216.            protected Map<String, String> getParams() throws AuthFailureError {
217.                Map<String, String> params = new HashMap<String, String>();
218.
219.                params.put("Namn", str_name);
220.                params.put("Efternamn", str_lastname);
221.                params.put("Telefonnummer", str_telephonenumber);
222.                params.put("Anvandarnamn", str_username);
223.                params.put("Lösenord", str_password);
224.
225.                if (Ed_admin.isChecked()) {
226.                    params.put("Id", 1 + "");
227.                } else if (Ed_user.isChecked()) {
228.                    params.put("Id", 0 + "");
229.                }
230.
231.                for (int i = 0; i < checkedItems.length; i++) {
232.                    if (checkedItems[i]) {
233.                        params.put("skap" + i, 1 + "");
234.                    } else {
235.                        params.put("skap" + i, 0 + "");
236.                    }
237.                }
238.                return params;
239.            }
240.        };
241.        RequestQueue requestQueue = Volley.newRequestQueue(Register.this);
242.        requestQueue.add(request);
243.    }
244. }
245.}
```

11.1.3 Deletion.java

```
1.  package com.example.arian;
2.  import androidx.annotation.NonNull;
3.  import androidx.appcompat.app.AppCompatActivity;
4.  import android.app.Activity;
5.  import android.app.ProgressDialog;
6.  import android.content.Intent;
7.  import android.os.Bundle;
8.  import android.view.MenuItem;
9.  import android.view.View;
10. import android.view.inputmethod.InputMethodManager;
11. import android.widget.EditText;
12. import android.widget.Toast;
13. import com.android.volley.AuthFailureError;
14. import com.android.volley.Request;
15. import com.android.volley.RequestQueue;
16. import com.android.volley.Response;
17. import com.android.volley.VolleyError;
18. import com.android.volley.toolbox.StringRequest;
19. import com.android.volley.toolbox.Volley;
20. import com.google.android.material.bottomnavigation.BottomNavigationView;
21. import java.util.HashMap;
22. import java.util.Map;
23.
24. public class Deletion extends AppCompatActivity {
25.
26.   private String url = "https://arian23.000webhostapp.com/Deletion.php";
27.   private String Namn, Användarnamn, Telefonnummer;
28.   private EditText ED_Namn, ED_Användarnamn, ED_Telefonnummer;
29.   private BottomNavigationView navigation;
30.
31.   //Metoden är till för att starta en activity i sitt första steg samt initiera variabler
32.   @Override
33.   protected void onCreate(Bundle savedInstanceState) {
34.     super.onCreate(savedInstanceState);
35.     setContentView(R.layout.activity_deletion);
36.
37.     ED_Namn = findViewById(R.id.DED_namn);
38.     ED_Användarnamn = findViewById(R.id.DED_användarnamn);
39.     ED_Telefonnummer = findViewById(R.id.DED_telefonnummer);
40.     navigation = (BottomNavigationView) findViewById(R.id.navigation);
41.     BottomNavigationView();
42.   }
43.
44.   //De olika knapparna Registrering,Ändring,Borttagning och Information som finns nere för att byta activtyn
45.   private void BottomNavigationView() {
46.     navigation.setSelectedItemId(R.id.navigation_tabort);
47.     navigation.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
48.
49.       @Override
50.       public boolean onNavigationItemSelected(@NonNull MenuItem item) {
51.         switch (item.getItemId()) {
52.
53.           case R.id.navigation_Reg:
54.             startActivity(new Intent(getApplicationContext(), Register.class));
55.             overridePendingTransition(0, 0);
56.             return true;
57.
58.           case R.id.navigation_ändra:
59.             startActivity(new Intent(getApplicationContext(), Edit.class));
```

```
                overridePendingTransition(0, 0);
                return true;

            case R.id.navigation_info:
                startActivity(new Intent(getApplicationContext(), Information.class));
                overridePendingTransition(0, 0);
                return true;

            case R.id.navigation_tabort:
                return true;
        }
        return false;
      }
    });
  }

  //Metoden är till för tillbaka knappen på telefonen
  public void onBackPressed() {
  }

  //Metoden är till då användaren vill dra ner tangentbordet automatisk via ett klick på fönstret
  public void HideKeyboard(View view) {
    InputMethodManager m = (InputMethodManager) getSystemService((Activity.INPUT_METHOD_SERVICE));
    m.hideSoftInputFromWindow(view.getWindowToken(), 0);
  }

  //Metoden ser till att hoppa till en ny klass vilket är Inloggning
  public void ToLogin(View view) {
    startActivity(new Intent(getApplicationContext(), Login.class));
    finish();
  }

  //Metoden är till för borttagning av person
  public void BtnDelete(final View view) {

    final ProgressDialog progressDialog = new ProgressDialog(this);
    progressDialog.setMessage("Väntar..");

    if (ED_Namn.getText().toString().equals("")) {
      Toast.makeText(this, "Glömt mata in Personen du vill ändra", Toast.LENGTH_SHORT).show();
    } else if (ED_Användarnamn.getText().toString().equals("")) {
      Toast.makeText(this, "Glömt mata in Användare du vill ändra", Toast.LENGTH_SHORT).show();
    } else if (ED_Telefonnummer.getText().toString().equals("")) {
      Toast.makeText(this, "Glömt mata in Telefonnummer du vill ändra", Toast.LENGTH_SHORT).show();
    } else {

      progressDialog.show();

      Telefonnummer = ED_Telefonnummer.getText().toString().trim();
      Namn = ED_Namn.getText().toString().trim();
      Användarnamn = ED_Användarnamn.getText().toString().trim();

      StringRequest request = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {

        @Override
        public void onResponse(String response) {
          progressDialog.dismiss();

          ED_Telefonnummer.setText("");
          ED_Namn.setText("");
          ED_Användarnamn.setText("");

          Toast.makeText(Deletion.this, response, Toast.LENGTH_SHORT).show();
        }
      }, new Response.ErrorListener() {

```

```
126.             @Override
127.             public void onErrorResponse(VolleyError error) {
128.                 progressDialog.dismiss();
129.                 Toast.makeText(Deletion.this, error.getMessage().toString(), Toast.LENGTH_SHORT).show();
130.             }
131.         }
132.
133.         ) {
134.             @Override
135.             protected Map<String, String> getParams() throws AuthFailureError {
136.                 Map<String, String> params = new HashMap<String, String>();
137.                 params.put("namn",Namn);
138.                 params.put("anvandarnamn",Användarnamn);
139.                 params.put("telefonnummer",Telefonnummer);
140.
141.                 return params;
142.             }
143.         };
144.         RequestQueue requestQueue = Volley.newRequestQueue(Deletion.this);
145.         requestQueue.add(request);
146.     }
147. }
148. }
```

11.1.4 Edit.java

```
1.  package com.example.arian;
2.  import androidx.annotation.NonNull;
3.  import androidx.appcompat.app.AppCompatActivity;
4.  import android.app.Activity;
5.  import android.app.AlertDialog;
6.  import android.app.ProgressDialog;
7.  import android.content.DialogInterface;
8.  import android.content.Intent;
9.  import android.os.Bundle;
10. import android.view.MenuItem;
11. import android.view.View;
12. import android.view.inputmethod.InputMethodManager;
13. import android.widget.Button;
14. import android.widget.EditText;
15. import android.widget.RadioButton;
16. import android.widget.RadioGroup;
17. import android.widget.Toast;
18. import com.android.volley.AuthFailureError;
19. import com.android.volley.Request;
20. import com.android.volley.RequestQueue;
21. import com.android.volley.Response;
22. import com.android.volley.VolleyError;
23. import com.android.volley.toolbox.StringRequest;
24. import com.android.volley.toolbox.Volley;
25. import com.google.android.material.bottomnavigation.BottomNavigationView;
26. import java.util.ArrayList;
27. import java.util.HashMap;
28. import java.util.Map;
29.
30. public class Edit extends AppCompatActivity implements View.OnClickListener {
31.
32.   private String url = "https://arian23.000webhostapp.com/Edit.php";
33.   private EditText Ed_Name, Ed_UserName, Ed_Password, Ed_LastName, Ed_TelephoneNumber, Old_Name,
    Old_TelephoneNumber;
34.   private String str_Name, str_Username, str_password, str_Lastname, str_Oldnamn, str_telephonenumber, str_Oldtelephonenumber;
35.   private RadioButton R_user, R_admin;
36.   private BottomNavigationView navigation;
37.   private RadioGroup R_Group;
38.   private Button BtnCabinets;
39.   private String[] listItems;
40.   private boolean[] checkedItems;
41.   private ArrayList<Integer> UserItems;
42.
43.   //Metoden är till för att starta en activity i sitt första steg samt initiera variabler
44.   @Override
45.   protected void onCreate(Bundle savedInstanceState) {
46.     super.onCreate(savedInstanceState);
47.     setContentView(R.layout.activity_edit);
48.     InitiateVariables();
49.     listItems = getResources().getStringArray(R.array.id_item);
50.     checkedItems = new boolean[listItems.length];
51.     UserItems = new ArrayList<>();
52.     BottomNavigationView();
53.     BtnCabinets.setOnClickListener(this);
54.
55.   }
56.
57.   // Initierar variablerna
58.   private void InitiateVariables() {
59.     navigation = (BottomNavigationView) findViewById(R.id.navigation);
60.     Old_Name = findViewById(R.id.edr_namn);
61.     Old_TelephoneNumber = findViewById(R.id.edr_telefonnummer);
```

```
62.      Ed_Name = findViewById(R.id.ed_namn);
63.      Ed_LastName = findViewById(R.id.ed_efternamn);
64.      Ed_UserName = findViewById(R.id.ed_email);
65.      Ed_TelephoneNumber = findViewById(R.id.ed_telefonnummer);
66.      Ed_Password = findViewById(R.id.ed_lösenord);
67.      R_Group = (RadioGroup) findViewById(R.id.ed_group);
68.      R_admin = (RadioButton) findViewById(R.id.ed_adm);
69.      R_user = (RadioButton) findViewById(R.id.ed_anv);
70.      BtnCabinets = (Button) findViewById(R.id.ed_skåp);
71.  }
72.
73.  //Knapptryckningen för att välja olika skåpsbehörigeter
74.  @Override
75.  public void onClick(View v) {
76.
77.      switch (v.getId()) {
78.
79.          case R.id.ed_skåp:
80.              AlertDialog.Builder mBuilder = new AlertDialog.Builder(Edit.this);
81.              mBuilder.setTitle("Välj Skåpsbehörigheter för användare");
82.              mBuilder.setMultiChoiceItems(listItems, checkedItems, new DialogInterface.OnMultiChoiceClickListener() {
83.                  @Override
84.                  public void onClick(DialogInterface dialog, int pos, boolean isChecked) {
85.
86.                      if (isChecked) {
87.                          if (!UserItems.contains(pos)) {
88.                              UserItems.add(pos);
89.                          } else {
90.                              UserItems.remove(pos);
91.                          }
92.                      }
93.                  }
94.              });
95.
96.              mBuilder.setCancelable(false);
97.              mBuilder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
98.                  @Override
99.                  public void onClick(DialogInterface dialog, int which) {
100.                     String item = "";
101.
102.                     for (int i = 0; i < UserItems.size(); i++) {
103.                         item = item + listItems[UserItems.get(i)];
104.                     }
105.                 }
106.             });
107.             mBuilder.setNeutralButton("Radera allt", new DialogInterface.OnClickListener() {
108.                 @Override
109.                 public void onClick(DialogInterface dialog, int which) {
110.                     for (int i = 0; i < checkedItems.length; i++) {
111.                         checkedItems[i] = false;
112.                         UserItems.clear();
113.                     }
114.                 }
115.             });
116.             AlertDialog mDialog = mBuilder.create();
117.             mDialog.show();
118.             break;
119.     }
120. }
121.
122. //De olika knapparna Registrering,Ändring,Borttagning och Information som finns nere för att byta activtyn
123. private void BottomNavigationView() {
124.     navigation.setSelectedItemId(R.id.navigation_ändra);
125.     navigation.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
126.         @Override
127.         public boolean onNavigationItemSelected(@NonNull MenuItem item) {
```

```
            switch (item.getItemId()) {

                case R.id.navigation_Reg:
                    startActivity(new Intent(getApplicationContext(), Register.class));
                    overridePendingTransition(0, 0);
                    return true;

                case R.id.navigation_ändra:
                    return true;

                case R.id.navigation_info:
                    startActivity(new Intent(getApplicationContext(), Information.class));
                    overridePendingTransition(0, 0);
                    return true;

                case R.id.navigation_tabort:
                    startActivity(new Intent(getApplicationContext(), Deletion.class));
                    overridePendingTransition(0, 0);
                    return true;
            }

            return false;
        }
    });
}

//Metoden ser till att hoppa till en ny klass vilket är Inloggning
public void LogIn(View view) {
    startActivity(new Intent(getApplicationContext(), Login.class));
    finish();
}

//Metoden är till för tillbcka kanppen på telefonen
public void onBackPressed() {

}

//Metoden är till då användaren vill dra ner tangentbordet automatisk via ett click på fönstret
public void HideKeyboard(View view) {
    InputMethodManager m = (InputMethodManager) getSystemService((Activity.INPUT_METHOD_SERVICE));
    m.hideSoftInputFromWindow(view.getWindowToken(), 0);
}

//Metoden ser till att uppdatera personuppgifter
public void BtnUpdate(final View view) {

    final ProgressDialog progressDialog = new ProgressDialog(this);
    progressDialog.setMessage("Väntar..");

    if (Old_Name.getText().toString().equals("")) {
        Toast.makeText(this, "Glömt mata in Personen du vill ändra", Toast.LENGTH_SHORT).show();
    } else if (Old_TelephoneNumber.getText().toString().equals("")) {
        Toast.makeText(this, "Glömt mata in Telefonnummer", Toast.LENGTH_SHORT).show();
    } else if (Ed_Name.getText().toString().equals("")) {
        Toast.makeText(this, "Glömt mata in Namn", Toast.LENGTH_SHORT).show();
    } else if (Ed_LastName.getText().toString().equals("")) {
        Toast.makeText(this, "Glömt mata in Efternamn", Toast.LENGTH_SHORT).show();
    } else if (Ed_UserName.getText().toString().equals("")) {
        Toast.makeText(this, "Glömt mata in Användarnamn", Toast.LENGTH_SHORT).show();
    } else if (Ed_TelephoneNumber.getText().toString().equals("")) {
        Toast.makeText(this, "Glömt mata in Telefonnummer", Toast.LENGTH_SHORT).show();
    } else if (Ed_Password.getText().toString().equals("")) {
        Toast.makeText(this, "Glömt mata in lösenord", Toast.LENGTH_SHORT).show();
    } else if (R_Group.getCheckedRadioButtonId() == -1) {
        Toast.makeText(this, "Glömt mata in konto", Toast.LENGTH_SHORT).show();
    } else {
```

```
194.
195.        progressDialog.show();
196.
197.        str_Name = Ed_Name.getText().toString().trim();
198.        str_Lastname = Ed_LastName.getText().toString().trim();
199.        str_telephonenumber = Ed_TelephoneNumber.getText().toString().trim();
200.        str_Username = Ed_UserName.getText().toString().trim();
201.        str_password = Ed_Password.getText().toString().trim();
202.
203.        str_Oldnamn = Old_Name.getText().toString().trim();
204.        str_Oldtelephonenumber = Old_TelephoneNumber.getText().toString().trim();
205.
206.        StringRequest request = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
207.
208.            @Override
209.            public void onResponse(String response) {
210.                progressDialog.dismiss();
211.                Ed_Name.setText("");
212.                Ed_LastName.setText("");
213.                Ed_Password.setText("");
214.                Ed_UserName.setText("");
215.                Ed_TelephoneNumber.setText("");
216.                Old_Name.setText("");
217.                Old_TelephoneNumber.setText("");
218.
219.                Toast.makeText(Edit.this, response, Toast.LENGTH_SHORT).show();
220.            }
221.        }, new Response.ErrorListener() {
222.
223.            @Override
224.            public void onErrorResponse(VolleyError error) {
225.                progressDialog.dismiss();
226.                Toast.makeText(Edit.this, error.getMessage().toString(), Toast.LENGTH_SHORT).show();
227.            }
228.        }
229.
230.        ) {
231.            @Override
232.            protected Map<String, String> getParams() throws AuthFailureError {
233.                Map<String, String> params = new HashMap<String, String>();
234.
235.                params.put("old_name", str_Oldnamn);
236.                params.put("old_telefonnummer", str_Oldtelephonenumber);
237.                params.put("Namn", str_Name);
238.                params.put("Efternamn", str_Lastname);
239.                params.put("Telefonnummer", str_telephonenumber);
240.                params.put("Anvandarnamn", str_Username);
241.                params.put("Lösenord", str_password);
242.
243.                if (R_admin.isChecked()) {
244.                    params.put("Id", 1 + "");
245.                } else if (R_user.isChecked()) {
246.                    params.put("Id", 0 + "");
247.                }
248.
249.                for (int i = 0; i < checkedItems.length; i++) {
250.
251.                    if (checkedItems[i]) {
252.                        params.put("skap" + i, 1 + "");
253.                    } else {
254.                        params.put("skap" + i, 0 + "");
255.                    }
256.                }
257.                return params;
258.            }
259.        };
```

```
260.            RequestQueue requestQueue = Volley.newRequestQueue(Edit.this);
261.            requestQueue.add(request);
262.        }
263.
264.    }
265. }
```

11.1.5 ScanCode.java

```
1.  package com.example.arian;
2.  import android.app.AlertDialog;
3.  import android.content.Intent;
4.  import android.os.Bundle;
5.  import android.os.Handler;
6.  import android.widget.Toast;
7.  import androidx.appcompat.app.AppCompatActivity;
8.  import com.android.volley.AuthFailureError;
9.  import com.android.volley.Request;
10. import com.android.volley.RequestQueue;
11. import com.android.volley.Response;
12. import com.android.volley.VolleyError;
13. import com.android.volley.toolbox.StringRequest;
14. import com.android.volley.toolbox.Volley;
15. import com.google.zxing.Result;
16. import java.util.HashMap;
17. import java.util.Map;
18. import me.dm7.barcodescanner.zxing.ZXingScannerView;
19.
20. public class ScanCode extends AppCompatActivity implements ZXingScannerView.ResultHandler {
21.
22.    private String url = "http://arian23.000webhostapp.com/GetCabinetsId.php";
23.    public static String CabinetsId;
24.    private ZXingScannerView scannerview;
25.    private String Result;
26.
27.    //Metoden är till för att starta en activity i sitt första steg samt initiera variabler
28.    @Override
29.    protected void onCreate(Bundle savedInstanceState) {
30.        super.onCreate(savedInstanceState);
31.        scannerview = new ZXingScannerView(ScanCode.this);
32.        setContentView(scannerview);
33.    }
34.
35.    //Metoden hämtar resultatet från qr skanningen
36.    @Override
37.    public void handleResult(Result result) {
38.
39.      Result = result.getText().toString();
40.     CabinetsId = result.getText().toString();
41.
42.        StringRequest request1 = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
43.
44.            @Override
45.            public void onResponse(String response) {
46.
47.                if (response.equalsIgnoreCase("Du kommer nu att anslutas")) {
48.                    startActivity(new Intent(getApplicationContext(), MainActivity.class));
49.                } else if (response.equalsIgnoreCase("Du har ej behörighet")) {
50.
51.                    AlertDialog.Builder builder = new AlertDialog.Builder((ScanCode.this));
```

```
52.                 builder.setTitle("Meddelande");
53.                 builder.setMessage("Du har ej behörighet till detta skåp");
54.                 builder.setPositiveButton("ok", null);
55.                 builder.show();
56.
57.                 new Handler().postDelayed(new Runnable() {
58.                     @Override
59.                     public void run() {
60.                         Intent i = new Intent(ScanCode.this, Login.class);
61.                         startActivity(i);
62.                         finish();
63.                     }
64.                 }, 4000);
65.             }
66.         }
67.     }, new Response.ErrorListener() {
68.
69.         @Override
70.         public void onErrorResponse(VolleyError error) {
71.             Toast.makeText(ScanCode.this, error.getMessage().toString(), Toast.LENGTH_SHORT).show();
72.         }
73.     }
74.     ) {
75.         @Override
76.         protected Map<String, String> getParams() throws AuthFailureError {
77.             Map<String, String> params = new HashMap<String, String>();
78.             params.put("names", Login.Username);// namnet på användaren
79.             //params.put("SKAP",BleName);
80.
81.             params.put("SKAP", Result);
82.
83.             return params;
84.         }
85.     };
86.     RequestQueue requestQueue = Volley.newRequestQueue(ScanCode.this);
87.     requestQueue.add(request1);
88. }
89.
90. //När activity är i pausläge
91. @Override
92. protected void onPause() {
93.     super.onPause();
94.     scannerview.stopCamera();
95. }
96.
97. //När activity kommer tillbaka då den har varit gömd från viewn
98. @Override
99. protected void onResume() {
100.     super.onResume();
101.     scannerview.setResultHandler(this);
102.     scannerview.startCamera();
103. }
104. }
105.
```

11.1.6 MainActivity.java

```
1.  package com.example.arian;
2.  import androidx.appcompat.app.AppCompatActivity;
3.  import androidx.core.content.FileProvider;
4.  import android.app.Activity;
5.  import android.app.AlertDialog;
6.  import android.app.ProgressDialog;
7.  import android.content.Intent;
8.  import android.content.pm.PackageManager;
9.  import android.net.Uri;
10. import android.os.Build;
11. import android.os.Bundle;
12. import android.os.Environment;
13. import android.provider.MediaStore;
14. import android.view.View;
15. import android.widget.Button;
16. import android.widget.ImageView;
17. import android.widget.TextView;
18. import android.widget.Toast;
19. import com.android.volley.AuthFailureError;
20. import com.android.volley.Request;
21. import com.android.volley.RequestQueue;
22. import com.android.volley.Response;
23. import com.android.volley.VolleyError;
24. import com.android.volley.toolbox.StringRequest;
25. import com.android.volley.toolbox.Volley;
26. import com.example.arian.networking.ApiConfig;
27. import com.example.arian.networking.AppConfig;
28. import com.example.arian.networking.ServerResponse;
29. import java.io.File;
30. import java.io.IOException;
31. import java.text.SimpleDateFormat;
32. import java.util.Date;
33. import java.util.HashMap;
34. import java.util.Map;
35. import app.akexorcist.bluetotohspp.library.BluetoothSPP;
36. import app.akexorcist.bluetotohspp.library.BluetoothState;
37. import okhttp3.MediaType;
38. import okhttp3.RequestBody;
39. import retrofit2.Call;
40. import retrofit2.Callback;
41. public class MainActivity extends AppCompatActivity implements View.OnClickListener {
42.
43.    private static final String url = "http://arian23.000webhostapp.com/Logout.php";
44.    private BluetoothSPP bluetooth;
45.    private static final String ON = "1";
46.    private static final String OFF = "0";
47.    private static final int CAMERA_PIC_REQUEST = 11;
48.    private String ImagePath = "";
49.    private Button openCabinets, closeCabinets, Connect, takePicture;
50.    private String BleNamn, UserName, postPath;
51.    private ImageView LockImage, OpenImage;
52.    private TextView CabinetsId;
53.    private int count = 0;
54.    private Boolean bildtagning, bild, öppen;
55.    private ProgressDialog pDialog;
56.
57.    //Metoden är till för att starta en activity i sitt första steg samt initiera variabler
58.    @Override
59.    protected void onCreate(Bundle savedInstanceState) {
60.        super.onCreate(savedInstanceState);
61.        setContentView(R.layout.activity_main);
62.
```

```
63.         InitiateVariables();
64.         TurnOnBluetooth();
65.         initDialog();
66.
67.         takePicture.setOnClickListener(this);
68.         Connect.setOnClickListener(this);
69.         openCabinets.setOnClickListener(this);
70.         closeCabinets.setOnClickListener(this);
71.
72.     }
73.
74.     // Initierar variablerna
75.     private void InitiateVariables() {
76.         takePicture = (Button) findViewById(R.id.image);
77.         bluetooth = new BluetoothSPP(this);
78.         Connect = (Button) findViewById(R.id.connect);
79.         openCabinets = (Button) findViewById(R.id.on);
80.         closeCabinets = (Button) findViewById(R.id.off);
81.         LockImage = (ImageView) findViewById(R.id.lockimage);
82.         OpenImage = (ImageView) findViewById(R.id.ble);
83.         CabinetsId = (TextView) findViewById(R.id.skapid);
84.         CabinetsId.setText(ScanCode.CabinetsId);
85.         UserName = Login.Username;
86.         TextView tex = findViewById(R.id.txtname);
87.         tex.setText(UserName);
88.         bildtagning = false;
89.         bild = false;
90.         öppen = true;
91.     }
92.
93.     // Olika knapptryckningar för respektive funktion
94.     @Override
95.     public void onClick(View v) {
96.
97.         switch (v.getId()) {
98.
99.             case R.id.image:
100.                if (bildtagning == true && count < 1) {
101.                    takePicture();
102.                }
103.                break;
104.
105.            case R.id.connect:
106.                Intent intent = new Intent(getApplicationContext(), DeviceList.class);
107.                startActivityForResult(intent, BluetoothState.REQUEST_CONNECT_DEVICE);
108.                break;
109.
110.            case R.id.on:
111.                if (bluetooth.getServiceState() > 1) {
112.                    bluetooth.send(ON, true);
113.                    öppen = false;
114.                    bildtagning = true;
115.                    LockImage.setImageResource(R.drawable.ic_lock_open_black_24dp);
116.                }
117.                break;
118.
119.            case R.id.off:
120.                if (bluetooth.getServiceState() > 1 && bild == true) {
121.                    bluetooth.send(OFF, true);
122.                    öppen = true;
123.                    bildtagning = false;
124.                    LockImage.setImageResource(R.drawable.ic_lock_outline_black_24dp);
125.                    bild = false;
126.                    count = 0;
127.                } else if (bildtagning == true) {
128.                    AlertDialog.Builder builder = new AlertDialog.Builder((MainActivity.this));
```

```
129.                builder.setTitle("Meddelande");
130.                builder.setMessage("Du måste ta bild för att kunna stänga skåpet");
131.                builder.setPositiveButton("ok", null);
132.                builder.show();
133.
134.            }
135.            break;
136.        }
137.    }
138.
139.    //Metoden används för att kolla om olika tillstånd för bildtagning har accepterats av användaren
140.    @Override
141.    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
142.        if (requestCode == 0) {
143.            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED
144.                    && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
145.                takePicture.setEnabled(true);
146.            }
147.        }
148.    }
149.
150.    //Metoden används för att ta bild och skicka iväg den till databasen
151.    private void takePicture() {
152.
153.        if (Build.VERSION.SDK_INT > 22) {
154.
155.            Intent callCameraApplicationIntent = new Intent();
156.            callCameraApplicationIntent.setAction(MediaStore.ACTION_IMAGE_CAPTURE);
157.            File ImageFile = null;
158.
159.            try {
160.                ImageFile = FileCreateImage();
161.
162.            } catch (IOException e) {
163.
164.                e.printStackTrace();
165.            }
166.
167.            Uri outputUri = FileProvider.getUriForFile(
168.                    this,
169.                    BuildConfig.APPLICATION_ID + ".provider",
170.                    ImageFile);
171.
172.            callCameraApplicationIntent.putExtra(MediaStore.EXTRA_OUTPUT, outputUri);
173.            callCameraApplicationIntent.addFlags(Intent.FLAG_GRANT_WRITE_URI_PERMISSION |
    Intent.FLAG_GRANT_READ_URI_PERMISSION);
174.            startActivityForResult(callCameraApplicationIntent, CAMERA_PIC_REQUEST);
175.        }
176.    }
177.
178.    //Metoden används för att skapa fram ett namn till bildfilen
179.    private File FileCreateImage() throws IOException {
180.
181.        String timeStamp = new SimpleDateFormat("yyyy-MM-dd_HH:mm:SS").format(new Date());
182.        String imageFileName = ScanCode.CabinetsId + "_" + UserName + "_" + timeStamp;
183.        File storageDirectory = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES);
184.
185.        if (!storageDirectory.exists()) storageDirectory.mkdir();
186.
187.        File image = new File(storageDirectory, imageFileName + ".jpg");
188.
189.        ImagePath = image.getAbsolutePath();
190.
191.        return image;
192.    }
193.
```

```
194.  //Metoden är till för tillbaka knappen på telefonen
195.  public void onBackPressed() {
196.
197.  }
198.
199.  //Metoden används till att sätt igång bluetoothen
200.  public void TurnOnBluetooth() {
201.
202.      if (!bluetooth.isBluetoothAvailable()) {
203.          Toast.makeText(getApplicationContext(), "Bluetooth är inte tillgänglig", Toast.LENGTH_SHORT).show();
204.          finish();
205.      }
206.
207.      bluetooth.setBluetoothConnectionListener(new BluetoothSPP.BluetoothConnectionListener() {
208.          public void onDeviceConnected(String name, String address) {
209.              Connect.setText("Ansluten till " + name);
210.              BleNamn = name;
211.              OpenImage.setImageResource(R.drawable.ic_bluetooth_black_24dp);
212.          }
213.
214.          public void onDeviceDisconnected() {
215.
216.              Connect.setText("Anslutningen bröts");
217.              OpenImage.setImageResource(R.drawable.ic_bluetooth_disabled_black_24dp);
218.          }
219.
220.          public void onDeviceConnectionFailed() {
221.              Connect.setText("Anslutningen misslyckades");
222.              OpenImage.setImageResource(R.drawable.ic_bluetooth_disabled_black_24dp);
223.          }
224.      });
225.  }
226.
227.  //Då knappen logga ut trycks vill man återgå till loginActivityn
228.  public void LoginActivity(View view) {
229.
230.      if (öppen == true) {
231.          StringRequest request2 = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
232.
233.              @Override
234.              public void onResponse(String response) {
235.                  Toast.makeText(MainActivity.this, response, Toast.LENGTH_SHORT).show();
236.
237.              }
238.          }, new Response.ErrorListener() {
239.
240.              @Override
241.              public void onErrorResponse(VolleyError error) {
242.                  Toast.makeText(MainActivity.this, error.getMessage().toString(), Toast.LENGTH_SHORT).show();
243.              }
244.          }
245.          ) {
246.              @Override
247.              protected Map<String, String> getParams() throws AuthFailureError {
248.                  Map<String, String> params = new HashMap<String, String>();
249.                  params.put("Namn", UserName);// namnet på användaren
250.
251.                  return params;
252.              }
253.          };
254.          RequestQueue requestQueue = Volley.newRequestQueue(MainActivity.this);
255.          requestQueue.add(request2);
256.          startActivity(new Intent(getApplicationContext(), Login.class));
257.          finish();
258.
259.      } else if (öppen == false) {
```

```
260.         AlertDialog.Builder builder = new AlertDialog.Builder((MainActivity.this));
261.         builder.setTitle("Meddelande");
262.         builder.setMessage("Stäng skåpet för att logga ut");
263.         builder.setPositiveButton("ok", null);
264.         builder.show();
265.     }
266. }
267.
268. // Utförs vid start
269. public void onStart() {
270.     super.onStart();
271.     if (!bluetooth.isBluetoothEnabled()) {
272.         bluetooth.enable();
273.     } else {
274.         if (!bluetooth.isServiceAvailable()) {
275.             bluetooth.setupService();
276.             bluetooth.startService(BluetoothState.DEVICE_OTHER);
277.         }
278.     }
279. }
280.
281. // Då activityn avbryts
282. public void onDestroy() {
283.     super.onDestroy();
284.     bluetooth.stopService();
285. }
286.
287. //Metoden är till för att få fram ett resultat från en annan activity som startade i denna activityn
288. public void onActivityResult(int requestCode, int resultCode, Intent data) {
289.
290.     super.onActivityResult(requestCode, resultCode, data);
291.     if (requestCode == BluetoothState.REQUEST_CONNECT_DEVICE) {
292.         if (resultCode == Activity.RESULT_OK) {
293.             bluetooth.connect(data);
294.         }
295.     }
296.     if (resultCode == RESULT_OK) {
297.         if (requestCode == CAMERA_PIC_REQUEST) {
298.             if (Build.VERSION.SDK_INT > 21) {
299.                 postPath = ImagePath;
300.                 LaddaUppFil();
301.                 count = 1;
302.             }
303.         }
304.     } else if (requestCode == BluetoothState.REQUEST_ENABLE_BT) {
305.
306.         if (resultCode == Activity.RESULT_OK) {
307.             bluetooth.setupService();
308.         } else {
309.             Toast.makeText(getApplicationContext(), "Bluetooth är inte tillgänglig", Toast.LENGTH_SHORT).show();
310.             finish();
311.         }
312.     }
313. }
314.
315. //Metoden skapar en Dialogruta
316. protected void initDialog() {
317.     pDialog = new ProgressDialog(this);
318.     pDialog.setMessage("Laddar...");
319.     pDialog.setCancelable(true);
320. }
321.
322. //Metoden visar Dialogrutan
323. protected void showpDialog() {
324.     if (!pDialog.isShowing()) {
325.         pDialog.show();
```

```
326.     }
327.  }
328.
329.  // Metoden gömmer Dialogrutan
330.  protected void hidepDialog() {
331.     if (pDialog.isShowing()){
332.        pDialog.dismiss();
333.     }
334.  }
335.
336.  //Metoden är till för att Ladda upp bilden som tas till databasen
337.  private void LaddaUppFil() {
338.
339.     if (postPath == null || postPath.equals("")) {
340.        Toast.makeText(this, "Något blev fel försök igen ", Toast.LENGTH_LONG).show();
341.        return;
342.     } else {
343.        showpDialog();
344.        Map<String, RequestBody> map = new HashMap<>();
345.        File file = new File(postPath);
346.        RequestBody requestBody = RequestBody.create(MediaType.parse("*/*"), file);
347.        map.put("file\"; filename=\"" + file.getName() + "\"", requestBody);
348.        ApiConfig getResponse = AppConfig.getRetrofit().create(ApiConfig.class);
349.        Call<ServerResponse> call = getResponse.upload("token", map);
350.        call.enqueue(new Callback<ServerResponse>() {
351.
352.           @Override
353.           public void onResponse(Call<ServerResponse> call, retrofit2.Response<ServerResponse> response) {
354.
355.              if (response.isSuccessful()) {
356.                 if (response.body() != null) {
357.                    hidepDialog();
358.                    ServerResponse serverResponse = response.body();
359.                    Toast.makeText(MainActivity.this, serverResponse.getMessage(), Toast.LENGTH_SHORT).show();
360.                    bild = true;
361.                 }
362.              } else {
363.                 hidepDialog();
364.                 Toast.makeText(getApplicationContext(), "Problem med att ladda upp bild försök igen",
    Toast.LENGTH_SHORT).show();
365.              }
366.           }
367.           @Override
368.           public void onFailure(Call<ServerResponse> call, Throwable t) {
369.              hidepDialog();
370.           }
371.        });
372.     }
373.  }
374.}
```

11.1.7 LoadingDialog.java

```
1.  package com.example.arian;
2.  import android.app.Activity;
3.  import android.app.AlertDialog;
4.  import android.view.LayoutInflater;
5.
6.  public class LoadingDialog {
7.
8.    private Activity activity;
9.    private AlertDialog dialog;
10.
11.   /** En konstruktor som har parametren myActivity för att välja activity som man vill visa sin dialogruta */
12.   public LoadingDialog(Activity myActivity) {
13.       activity = myActivity;
14.   }
15.
16.   /** Startar sin Dialogruta i den activity som man bestämmer att den ska starta */
17.   public void startLoadingActvity() {
18.
19.       AlertDialog.Builder builder = new AlertDialog.Builder(activity);
20.       LayoutInflater inflater = activity.getLayoutInflater();
21.       builder.setView(inflater.inflate(R.layout.customdialog, null));
22.       builder.setCancelable(false);
23.       dialog = builder.create();
24.       dialog.show();
25.   }
26.
27.   /**Metoden är till för att stänga ner Dialogrutan */
28.   public void dismissDialog() {
29.       dialog.dismiss();
30.   }
31. }
```

11.1.8 Listview.java

```
1.  package com.example.arian;
2.  import android.app.Activity;
3.  import android.os.Bundle;
4.  import android.view.View;
5.  import android.view.inputmethod.InputMethodManager;
6.  import androidx.appcompat.app.AppCompatActivity;
7.
8.  public class Listview extends AppCompatActivity {
9.
10.   /**Metoden är till för att starta en activity i sitt första steg samt initiera variabler*/
11.   @Override
12.   protected void onCreate(Bundle savedInstanceState) {
13.       super.onCreate(savedInstanceState);
14.       setContentView(R.layout.activity_viewadapter1);
15.   }
16.
17.   /**Metoden är till då användaren vill dra ner tangentbordet automatisk via ett klick på fönstret  */
18.   public void Gömtangentbord(View view) {
19.       InputMethodManager m = (InputMethodManager) getSystemService((Activity.INPUT_METHOD_SERVICE));
20.       m.hideSoftInputFromWindow(view.getWindowToken(), 0);
21.   }
22. }
23.
```

11.1.9 Information.java

```
1.  package com.example.arian;
2.  import androidx.annotation.NonNull;
3.  import androidx.appcompat.app.AppCompatActivity;
4.  import androidx.recyclerview.widget.LinearLayoutManager;
5.  import androidx.recyclerview.widget.RecyclerView;
6.  import android.app.Activity;
7.  import android.content.Intent;
8.  import android.os.Bundle;
9.  import android.view.MenuItem;
10. import android.view.View;
11. import android.view.inputmethod.InputMethodManager;
12. import android.widget.Button;
13. import android.widget.EditText;
14. import android.widget.ListView;
15. import android.widget.Toast;
16. import com.android.volley.AuthFailureError;
17. import com.android.volley.Request;
18. import com.android.volley.RequestQueue;
19. import com.android.volley.Response;
20. import com.android.volley.VolleyError;
21. import com.android.volley.toolbox.StringRequest;
22. import com.android.volley.toolbox.Volley;
23. import com.google.android.material.bottomnavigation.BottomNavigationView;
24. import org.json.JSONArray;
25. import org.json.JSONException;
26. import org.json.JSONObject;
27. import java.util.ArrayList;
28. import java.util.HashMap;
29. import java.util.List;
30. import java.util.Map;
31.
32. public class Information extends AppCompatActivity implements View.OnClickListener {
33.
34.     private static final String url = "https://arian23.000webhostapp.com/GetFiles.php";
35.     private static final String DATA_URL = "https://arian23.000webhostapp.com/Information.php?Namn=";
36.     private static final String Namn = "Namn";
37.     private static final String Efternamn = "Efternamn";
38.     private static final String Användarnamn = "Anvandarnamn";
39.     private static final String Telefonnummer = "Telefonnummer";
40.     private static final String Lösenord = "Lösenord";
41.     private static final String Senast_inloggad = "last_login";
42.     private static final String Senast_utloggad = "last_logut";
43.     private static final String ID = "Id";
44.     private static final String JSON_ARRAY = "result";
45.     private Button btnInformation, btnskapInfo;
46.     private BottomNavigationView navigation;
47.     private RecyclerView recyclerView;
48.     private String img1, sökord;
49.     private EditText txtvärde;
50.     private List<Image> list;
51.
52.
53.     //Metoden är till för att starta en activity i sitt första steg samt initiera variabler
54.     @Override
55.     protected void onCreate(Bundle savedInstanceState) {
56.         super.onCreate(savedInstanceState);
57.         setContentView(R.layout.activity_information);
58.
59.         list = new ArrayList<>();
60.
61.         txtvärde = (EditText) findViewById(R.id.editText);
62.         Intent t = new Intent(getApplicationContext(), ListView.class);
```

```
        t.putExtra("namn", txtvärde.getText().toString().trim());

        btnInformation = (Button) findViewById(R.id.buttonInfo);
        btnskapInfo = (Button) findViewById(R.id.buttonSkap);

        recyclerView = findViewById(R.id.recyclerviewid);

        navigation = (BottomNavigationView) findViewById(R.id.navigation);

        BottomNavigationView();

        btnInformation.setOnClickListener(this);
        btnskapInfo.setOnClickListener(this);

    }

    //De olika knapptryckningarna
    @Override
    public void onClick(View v) {

        switch (v.getId()) {

            case R.id.buttonInfo:
                GetInfo();
                InputMethodManager m = (InputMethodManager) getSystemService((Activity.INPUT_METHOD_SERVICE));
                m.hideSoftInputFromWindow(v.getWindowToken(), 0);
                break;

            case R.id.buttonSkap:
                SkåpInfo();
                InputMethodManager m1 = (InputMethodManager) getSystemService((Activity.INPUT_METHOD_SERVICE));
                m1.hideSoftInputFromWindow(v.getWindowToken(), 0);
                break;
        }
    }

    //De olika knapparna Registrering,Ändring,Borttagning och Information som finns nere för att byta activtyn
    private void BottomNavigationView() {
        navigation.setSelectedItemId(R.id.navigation_info);
        navigation.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                switch (item.getItemId()) {
                    case R.id.navigation_Reg:
                        startActivity(new Intent(getApplicationContext(), Register.class));
                        overridePendingTransition(0, 0);
                        return true;

                    case R.id.navigation_ändra:
                        startActivity(new Intent(getApplicationContext(), Edit.class));
                        overridePendingTransition(0, 0);
                        return true;

                    case R.id.navigation_tabort:
                        startActivity(new Intent(getApplicationContext(), Deletion.class));
                        overridePendingTransition(0, 0);
                        return true;

                    case R.id.navigation_info:
                        return true;
                }
                return false;
            }
        });
    }

```

```
129.  //Metoden är till för tillbaka knappen på telefonen
130.  public void onBackPressed() {
131.
132.  }
133.
134.  //Metoden är till då användaren vill dra ner tangentbordet automatisk via ett klick på fönstret
135.  public void HideKeyboard(View views) {
136.      InputMethodManager m = (InputMethodManager) getSystemService((Activity.INPUT_METHOD_SERVICE));
137.      m.hideSoftInputFromWindow(views.getWindowToken(), 0);
138.  }
139.
140.  //Metoden hämtar activity_information om användaren
141.  private void GetInfo() {
142.
143.      String value = txtvärde.getText().toString().trim();
144.
145.      if (value.equals("")) {
146.          Toast.makeText(this, "Inget sökord angivet", Toast.LENGTH_LONG).show();
147.          return;
148.      }
149.
150.      String url = DATA_URL + txtvärde.getText().toString().trim();
151.
152.      StringRequest stringRequest = new StringRequest(url, new Response.Listener<String>() {
153.          @Override
154.          public void onResponse(String response) {
155.
156.              if (response.equalsIgnoreCase("Kunde inte hitta personen")) {
157.                  Toast.makeText(Information.this, response, Toast.LENGTH_SHORT).show();
158.              } else
159.
160.                  showJSON(response);
161.          }
162.      },
163.              new Response.ErrorListener() {
164.                  @Override
165.                  public void onErrorResponse(VolleyError error) {
166.                      Toast.makeText(Information.this, "Kunde inte hitta användaren", Toast.LENGTH_LONG).show();
167.                  }
168.              });
169.
170.      RequestQueue requestQueue = Volley.newRequestQueue(this);
171.      requestQueue.add(stringRequest);
172.  }
173.
174.  //Metoden Hämtar activity_information från en vektorn som innehåller activity_information om användarna som
      lagras sedan i arraylisten
175.  private void showJSON(String response) {
176.
177.      ArrayList<HashMap<String, String>> list = new ArrayList<HashMap<String, String>>();
178.      try {
179.          JSONObject jsonObject = new JSONObject(response);
180.          JSONArray result = jsonObject.getJSONArray(JSON_ARRAY);
181.
182.          for (int i = 0; i < result.length(); i++) {
183.              JSONObject jo = result.getJSONObject(i);
184.              String id = jo.getString(ID);
185.
186.              if (!(id.compareTo(String.valueOf(1)) == 0)) {
187.
188.                  String Namn = jo.getString(Information.Namn);
189.                  String Efternamn = jo.getString(Information.Efternamn);
190.                  String Telefonnummer = jo.getString(Information.Telefonnummer);
191.                  String Anvandarnamn = jo.getString(Användarnamn);
192.                  String Lösenord = jo.getString(Information.Lösenord);
193.                  String lastlogin = jo.getString(Senast_inloggad);
```

```
                String lastlogut = jo.getString(Senast_utloggad);

                final HashMap<String, String> employees = new HashMap<>();
                employees.put(Information.Namn, Namn);
                employees.put(Information.Efternamn, Efternamn);
                employees.put(Information.Telefonnummer, Telefonnummer);
                employees.put(Information.Användarnamn, Anvandarnamn);
                employees.put(Information.Lösenord, Lösenord);
                employees.put(Information.Senast_inloggad, lastlogin);
                employees.put(Information.Senast_utloggad, lastlogut);

                list.add(employees);
            }
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
    ViewAdapter1 myadapter = new ViewAdapter1(this, new String[]{Namn, Efternamn, Telefonnummer, Användarnamn,
Lösenord, Senast_inloggad, Senast_utloggad}, list);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));
    recyclerView.setAdapter(myadapter);
}

//Metoden ser till att hämta activity_information om olika skåp
public void SkåpInfo() {

    if (txtvärde.getText().toString().equals("")) {
        Toast.makeText(this, "Inget sökord angivet", Toast.LENGTH_SHORT).show();
    } else {
        sökord = txtvärde.getText().toString();
        list.clear();
        final String search = sökord;
        StringRequest request = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                String currentString = response;
                if (currentString.isEmpty()) {
                    setuprecyclerview(list);
                    return;
                }
                String[] separated = currentString.split(" ");
                String remove = "https://arian23.000webhostapp.com/Images/";

                for (int i = 0; i < separated.length; i++) {

                    img1 = separated[i];
                    String splitter = separated[i].replace(remove, "");
                    String tripleSplitter[] = splitter.split("_");
                    String jpgReplacer = tripleSplitter[3].replace(".jpg", "");

                    Image image = new Image();
                    image.setName(tripleSplitter[0]);
                    image.setDeletedPerson(splitter);
                    image.setDate("Datum: " + tripleSplitter[2]);
                    image.setOpenBy("Öppnat av: " + tripleSplitter[1]);
                    image.setNb_episode(2);
                    image.setTime("Tidpunkt: " + jpgReplacer);
                    image.setImage_url(img1);
                    list.add(image);
                    setuprecyclerview(list);
                }
            }
        }, new Response.ErrorListener() {

            @Override
            public void onErrorResponse(VolleyError error) {
```

```
                Toast.makeText(Information.this, error.getMessage().toString(), Toast.LENGTH_SHORT).show();
            }
        }

        ) {
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
                Map<String, String> params = new HashMap<String, String>();
                params.put("ser", search);

                return params;
            }
        };
        RequestQueue requestQueue = Volley.newRequestQueue(Information.this);
        requestQueue.add(request);
    }
}

//Metoden hämtar från listan lstAnime för att sedan spotta ut de i en recyclerView
public void setuprecyclerview(List<Image> lstAnime) {

    ViewAdapter2 myadapter = new ViewAdapter2(this, lstAnime);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));
    recyclerView.setAdapter(myadapter);
}
}
```

11.1.10 Image.java

```
package com.example.arian;

public class Image {

    private String name;
    private String deletedPerson;
    private String Date;
    private int nb_episode;
    private String OpenBy;
    private String Time;
    private String image_url;

    //Tom konstruktor
    public Image() {
    }

    //Skapar en konstruktor med parametrarna
    public Image(String name, String deletePerson, String Date, int nb_episode, String openBy, String time, String image_url) {

        this.name = name;
        this.deletedPerson = deletePerson;
        this.Date = Date;
        this.nb_episode = nb_episode;
        this.OpenBy = openBy;
        this.Time = time;
        this.image_url = image_url;
    }

```

```
29.  // Returnerar namnet
30.  public String getName() {
31.      return name;
32.  }
33.
34.  //Returnerar deletedPerson
35.  public String getDeletedPerson() {
36.      return deletedPerson;
37.  }
38.
39.  //Returnerar datumet
40.  public String getDate() {
41.      return Date;
42.  }
43.
44.  public int getNb_episode() {
45.      return nb_episode;
46.  }
47.
48.  //Returnerar personen som skåpet har öppnats av
49.  public String getOpenBy() {
50.      return OpenBy;
51.  }
52.
53.  //Returnerar tiden
54.  public String getTime() {
55.      return Time;
56.  }
57.
58.  //Returnerar bilden
59.  public String getImage_url() {
60.      return image_url;
61.  }
62.
63.  //Sätter name till name
64.  public void setName(String name) {
65.      this.name = name;
66.  }
67.
68.  //Sätter deletedPerson till deletedPerson
69.  public void setDeletedPerson(String deletedPerson) {
70.      this.deletedPerson = deletedPerson;
71.  }
72.
73.  //Sätter Date till date
74.  public void setDate(String date) {
75.      this.Date = date;
76.  }
77.
78.  //Sätter
79.  public void setNb_episode(int nb_episode) {
80.      this.nb_episode = nb_episode;
81.  }
82.
83.  //Sätter openBy till openBy
84.  public void setOpenBy(String openBy) {
85.      this.OpenBy = openBy;
86.  }
87.
88.  //Sätter time till time
89.  public void setTime(String time) {
90.      this.Time = time;
91.  }
92.
93.  //Sätter image_url till image_url
94.  public void setImage_url(String image_url) {
```

```
95.         this.image_url = image_url;
96.     }
97. }
```

11.1.11 DeleteCabinetInfo.java

```
1.  package com.example.arian;
2.  import android.os.Bundle;
3.  import android.view.View;
4.  import android.view.animation.Animation;
5.  import android.view.animation.AnimationUtils;
6.  import android.widget.Button;
7.  import android.widget.ImageView;
8.  import android.widget.Toast;
9.  import androidx.appcompat.app.AppCompatActivity;
10. import com.android.volley.AuthFailureError;
11. import com.android.volley.Request;
12. import com.android.volley.RequestQueue;
13. import com.android.volley.Response;
14. import com.android.volley.VolleyError;
15. import com.android.volley.toolbox.StringRequest;
16. import com.android.volley.toolbox.Volley;
17. import com.bumptech.glide.Glide;
18. import com.bumptech.glide.request.RequestOptions;
19. import java.util.HashMap;
20. import java.util.Map;
21.
22. public class DeleteCabinetInfo extends AppCompatActivity implements View.OnClickListener {
23.
24.     private String url = "https://arian23.000webhostapp.com/Delete.php";
25.     private String DeletedPerson, Image;
26.     private Button taBort;
27.
28.     //Metoden är till för att starta en activity i sitt första steg samt initiera variabler
29.     @Override
30.     protected void onCreate(Bundle savedInstanceState) {
31.         super.onCreate(savedInstanceState);
32.         setContentView(R.layout.activity_deletecabinetinfo);
33.
34.         taBort = findViewById(R.id.tabort);
35.         taBort.setOnClickListener(this);
36.
37.         RecieveData();
38.
39.         ImageView img = findViewById(R.id.aa_thumbnail);
40.
41.         RequestOptions requestOptions = new
    RequestOptions().centerCrop().placeholder(R.drawable.loading_shape).error(R.drawable.loading_shape);
42.
43.         Glide.with(this).load(Image).apply(requestOptions).into(img);
44.
45.     }
46.
47.     //Hämtar Data från Intent
48.     private void RecieveData() {
49.
50.         DeletedPerson = getIntent().getExtras().getString("DeletedPerson");
51.         Image = getIntent().getExtras().getString("Image");
52.     }
53.
```

```
54.    //Metod för knapptryckning
55.    @Override
56.    public void onClick(View view) {
57.        if (view == taBort) {
58.            delete();
59.            finish();
60.        }
61.    }
62.
63.    //Tar bort bilden från databasen samt även i Recyclerviewn
64.    public void delete() {
65.        StringRequest request = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
66.            @Override
67.            public void onResponse(String response) {
68.                String currentString = response;
69.            }
70.        }, new Response.ErrorListener() {
71.
72.            @Override
73.            public void onErrorResponse(VolleyError error) {
74.                Toast.makeText(DeleteCabinetInfo.this, error.getMessage().toString(), Toast.LENGTH_SHORT).show();
75.            }
76.        }
77.        ) {
78.            @Override
79.            protected Map<String, String> getParams() throws AuthFailureError {
80.                Map<String, String> params = new HashMap<String, String>();
81.                params.put("delete", DeletedPerson);
82.
83.                return params;
84.            }
85.        };
86.        RequestQueue requestQueue = Volley.newRequestQueue(DeleteCabinetInfo.this);
87.        requestQueue.add(request);
88.    }
89. }
```

11.1.12 DeviceList.java

```
1.  package com.example.arian;
2.  import android.Manifest;
3.  import android.annotation.SuppressLint;
4.  import android.app.Activity;
5.  import android.bluetooth.BluetoothAdapter;
6.  import android.bluetooth.BluetoothDevice;
7.  import android.content.BroadcastReceiver;
8.  import android.content.Context;
9.  import android.content.Intent;
10. import android.content.IntentFilter;
11. import android.os.Build;
12. import android.os.Bundle;
13. import android.os.Handler;
14. import android.view.View;
15. import android.widget.AdapterView;
16. import android.widget.AdapterView.OnItemClickListener;
17. import android.widget.ArrayAdapter;
18. import android.widget.Button;
19. import android.widget.ListView;
20. import android.widget.TextView;
21. import java.util.HashSet;
22. import java.util.Set;
23. import app.akexorcist.bluetotohspp.library.BluetoothState;
24.
25. public class DeviceList extends Activity implements View.OnClickListener{
26.
27.   private BluetoothAdapter BtAdapter;
28.   private ArrayAdapter<String> PairedDevicesArrayAdapter;
29.   private Set<BluetoothDevice> PairedDevicesList;
30.   private Set<BluetoothDevice> DeviceList;
31.   private Button scanButton;
32.   private LoadingDialog loadingDialog;
33.
34.   //Metoden är till för att starta en activity i sitt första steg samt initiera variabler
35.   protected void onCreate(Bundle savedInstanceState)  {
36.      super.onCreate(savedInstanceState);
37.      setContentView(R.layout.activity_devicelist);
38.      loadingDialog = new LoadingDialog(DeviceList.this);
39.      DeviceList = new HashSet<BluetoothDevice>();
40.      PairedDevicesArrayAdapter = new ArrayAdapter<String>(this, R.layout.device_name);
41.       // Hitta och lägg upp parkopplade enheter i listViewn
42.      ListView pairedListView = (ListView) findViewById(R.id.list_devices);
43.      pairedListView.setAdapter(PairedDevicesArrayAdapter);
44.      pairedListView.setOnItemClickListener(mDeviceClickListener);
45.
46.      // Registrera i broadcasten när en enhet har upptäckts
47.      IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
48.      this.registerReceiver(mReceiver, filter);
49.
50.      // Registrera i broadcasten när en enhets uppteckningen har avslutats
51.      filter = new IntentFilter(BluetoothAdapter.ACTION_DISCOVERY_FINISHED);
52.      this.registerReceiver(mReceiver, filter);
53.
54.      // Hämta lokala bluetooth adaptern
55.      BtAdapter = BluetoothAdapter.getDefaultAdapter();
56.
57.      // Hämtar de parkopplade enheter
58.      PairedDevicesList = BtAdapter.getBondedDevices();
59.
60.      scanButton = (Button) findViewById(R.id.button_scan);
61.      scanButton.setText("Sök efter nya enheter");
62.      scanButton.setOnClickListener(this);
```

```
63.
64.     checkForPairedDevices();
65.   }
66.
67.   // Knapptryckning för att söka fram ny enheter
68.   @Override
69.   public void onClick(View v) {
70.     switch(v.getId()){
71.
72.       case R.id.button_scan:
73.         loadingDialog.startLoadingActvity();
74.         doDiscovery();
75.         new Handler().postDelayed(new Runnable() {
76.           @Override
77.           public void run() {
78.             BtAdapter.cancelDiscovery();
79.
80.             loadingDialog.dismissDialog();
81.             for (BluetoothDevice device : DeviceList) {
82.               if (device.getName() != null) {
83.                 PairedDevicesArrayAdapter.add(device.getName() + "\n" + device.getAddress());
84.               } else {
85.                 PairedDevicesArrayAdapter.add("" + "\n" + device.getAddress());
86.               }
87.             }
88.
89.           }
90.         }, 10000);
91.     }
92.   }
93.
94.   //Kollar igenom ifall de finns några parkopplade enheter
95.   private void checkForPairedDevices() {
96.
97.     if (PairedDevicesList.size() > 0) {
98.       for (BluetoothDevice device : PairedDevicesList) {
99.         PairedDevicesArrayAdapter.add(device.getName() + "\n" + device.getAddress());
100.      }
101.    } else {
102.      String noDevices = "Inga parkopplade enheter finns";
103.      PairedDevicesArrayAdapter.add(noDevices);
104.    }
105.  }
106.
107.  //Då activtyn avbryts
108.  protected void onDestroy() {
109.    super.onDestroy();
110.
111.    if (BtAdapter != null) {
112.      BtAdapter.cancelDiscovery();
113.    }
114.    // Stänger ner Receivern
115.    this.unregisterReceiver(mReceiver);
116.    this.finish();
117.  }
118.
119.  //Startar sökningen efter bluetooth enheter med hjälp av BluetoothAdapter
120.  private void doDiscovery() {
121.
122.    if (BtAdapter.isDiscovering()) {
123.      BtAdapter.cancelDiscovery();
124.    }
125.    checkBTPermissions();
126.    PairedDevicesArrayAdapter.clear();
127.    checkForPairedDevices();
128.    BtAdapter.startDiscovery();
```

```
129.    }
130.
131.    // Vid tryckning av en enhet i listviewn
132.    private OnItemClickListener mDeviceClickListener = new OnItemClickListener() {
133.        public void onItemClick(AdapterView<?> av, View v, int arg2, long arg3) {
134.
135.            // Stoppar sökningen utav enheter då det kostar för mycket att ha igån den
136.            if (BtAdapter.isDiscovering()) {
137.                BtAdapter.cancelDiscovery();
138.            }
139.
140.            if (!((TextView) v).getText().toString().equals("Inga enheter finns")) {
141.
142.                //Hämta enhetens MAC adress som är 17 bokstäver lång i vyn
143.                String info = ((TextView) v).getText().toString();
144.                String address = info.substring(info.length() - 17);
145.
146.                //Lägger in MAC adressen i en intent för att sedan verifiera bluetoothens uuid
147.                Intent intent = new Intent();
148.                intent.putExtra(BluetoothState.EXTRA_DEVICE_ADDRESS, address);
149.
150.                //Lögger in resultatet och avlutar denna activity
151.                setResult(Activity.RESULT_OK, intent);
152.                finish();
153.            }
154.        }
155.    };
156.
157.    // Lägger in olika enheter i DeviceList som BroadcastReceiver upptäcker
158.    private final BroadcastReceiver mReceiver = new BroadcastReceiver() {
159.        public void onReceive(Context context, Intent intent) {
160.            String action = intent.getAction();
161.
162.            // När en enhet upptäcks
163.            if (BluetoothDevice.ACTION_FOUND.equals(action)) {
164.
165.                // Hämtar olika Bluetooth enheter från intent
166.                BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
167.
168.                // Om enheten är redan parkopplade så hoppar vi över den och söker efter ny enhet
169.                if (device.getBondState() != BluetoothDevice.BOND_BONDED) {
170.
171.                    if (PairedDevicesArrayAdapter.getItem(0).equals("Inga enheter finns")) {
172.                        PairedDevicesArrayAdapter.remove("Inga enheter finns");
173.                    }
174.                    DeviceList.add(device);
175.                }
176.            }
177.        }
178.    };
179.
180.    //Kollar igenom olika tillstånden ifall de finns i manifest
181.    private void checkBTPermissions() {
182.        if (Build.VERSION.SDK_INT > Build.VERSION_CODES.LOLLIPOP) {
183.            int permissionCheck =
    this.checkSelfPermission("Manifest.activity_viewadapter3.ACCESS_FINE_LOCATION");
184.            permissionCheck +=
    this.checkSelfPermission("Manifest.activity_viewadapter3.ACCESS_COARSE_LOCATION");
185.            if (permissionCheck != 0) {
186.                this.requestPermissions(new String[]{Manifest.permission.ACCESS_FINE_LOCATION,
    Manifest.permission.ACCESS_COARSE_LOCATION}, 1001); //Any number
187.            }
188.        }
189.    }
190. }
191.
```

11.1.13 CabinetsPermission.java

```
1.  package com.example.arian;
2.  import android.os.Bundle;
3.  import android.widget.Toast;
4.  import androidx.appcompat.app.AppCompatActivity;
5.  import androidx.recyclerview.widget.LinearLayoutManager;
6.  import androidx.recyclerview.widget.RecyclerView;
7.  import com.android.volley.AuthFailureError;
8.  import com.android.volley.Request;
9.  import com.android.volley.RequestQueue;
10. import com.android.volley.Response;
11. import com.android.volley.VolleyError;
12. import com.android.volley.toolbox.StringRequest;
13. import com.android.volley.toolbox.Volley;
14. import java.util.ArrayList;
15. import java.util.HashMap;
16. import java.util.List;
17. import java.util.Map;
18.
19. public class CabinetsPermission extends AppCompatActivity {
20.
21.   private String url = "https://arian23.000webhostapp.com/Cabinets.php";
22.   private String Username;
23.   private List<Image> list;
24.   private RecyclerView recyclerView;
25.
26.   //Metoden är till för att starta en activity i sitt första steg samt initiera variabler
27.   @Override
28.   protected void onCreate(Bundle savedInstanceState) {
29.     super.onCreate(savedInstanceState);
30.     setContentView(R.layout.cabinets);
31.     list = new ArrayList<>();
32.     recyclerView = findViewById(R.id.recyclerviewid2);
33.     Username = getIntent().getExtras().getString("Username");
34.     getCabinetsPermission();
35.   }
36.
37.   //Hämtar tillstånden för olika skåpen för specifik Användare
38.   public void getCabinetsPermission() {
39.     StringRequest request = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
40.       @Override
41.       public void onResponse(String response) {
42.         String currentString = response;
43.         String[] separated = currentString.split(" ");
44.         String bool;
45.         for (int i = 0; i < separated.length; i++) {
46.           if (separated[i].equals("1")) {
47.             bool = "JA";
48.           } else {
```

```
                bool = "NEJ";
            }
            Image image = new Image();
            image.setName("Skåp " + i);
            image.setDate("Åtkomst: " + bool);
            list.add(image);
            setuprecyclerview(list);
        }
      }
    }, new Response.ErrorListener() {

      @Override
      public void onErrorResponse(VolleyError error) {
        Toast.makeText(CabinetsPermission.this, error.getMessage().toString(), Toast.LENGTH_SHORT).show();
      }
    }

    ) {
      @Override
      protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<String, String>();
        params.put("name", Username);
        return params;

      }
    };
    RequestQueue requestQueue = Volley.newRequestQueue(CabinetsPermission.this);
    requestQueue.add(request);
  }

  //Lägger in värden från listan lstAnime in i Recyclerviewn
  private void setuprecyclerview(List<Image> lstAnime) {

    ViewAdapter3 myadapter2 = new ViewAdapter3(this, lstAnime);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));
    recyclerView.setAdapter(myadapter2);
  }
}
```

11.1.14 ViewAdapter1.java

```
1.  package com.example.arian;
2.  import android.content.Context;
3.  import android.content.Intent;
4.  import android.view.LayoutInflater;
5.  import android.view.View;
6.  import android.view.ViewGroup;
7.  import android.widget.Button;
8.  import android.widget.TextView;
9.  import androidx.annotation.NonNull;
10. import androidx.recyclerview.widget.RecyclerView;
11. import java.util.ArrayList;
12. import java.util.HashMap;
13.
14. public class ViewAdapter1 extends RecyclerView.Adapter<ViewAdapter1.MyViewHolder> {
15.
16.   private static final String Name = "Namn";
17.   private static final String Lastname = "Efternamn";
18.   private static final String Username = "Anvandarnamn";
19.   private static final String Telephonenumber = "Telefonnummer";
20.   private static final String Password = "Lösenord";
21.   private static final String Lastlogin = "last_login";
22.   private static final String Lastlogout = "last_logut";
23.   private String s1[];
24.   private ArrayList<HashMap<String, String>> list;
25.   private Context ct;
26.
27.   //Konstruktor med olika parametrar
28.   public ViewAdapter1(Context ct, String s1[], ArrayList<HashMap<String, String>> list) {
29.     this.ct = ct;
30.     this.s1 = s1;
31.     this.list = list;
32.   }
33.
34.   //Metoden bestämmer vilken view som ska användas
35.   @NonNull
36.   @Override
37.   public ViewAdapter1.MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
38.     LayoutInflater inflater = LayoutInflater.from(ct);
39.     View view = inflater.inflate(R.layout.activity_viewadapter1, parent, false);
40.     return new MyViewHolder(view);
41.   }
42.
43.   // Metoden hämtar olika värden från list och lägger in dom i viewns olika variabler
44.   @Override
45.   public void onBindViewHolder(@NonNull ViewAdapter1.MyViewHolder holder, final int position) {
46.     holder.Name.setText(list.get(position).get(Name));
47.     holder.Lastname.setText(list.get(position).get(Lastname));
48.     holder.Telephonenumber.setText("Telefonnummer: " + list.get(position).get(Telephonenumber));
49.     holder.Username.setText("Användarnamn: " + list.get(position).get(Username));
50.     holder.Password.setText("Lösenord: " + list.get(position).get(Password));
51.     holder.Lastlogin.setText("Inloggad: " + list.get(position).get(Lastlogin));
52.     holder.Lastlogout.setText("Utloggad: " + list.get(position).get(Lastlogout));
53.     holder.BtnCompetence.setOnClickListener(new View.OnClickListener() {
54.       @Override
55.       public void onClick(View v) {
56.         Intent t1 = new Intent(ct, CabinetsPermission.class);
57.         t1.putExtra("Username", list.get(position).get(Username));
58.         ct.startActivity(t1);
59.       }
60.     });
61.   }
62.
```

```
63.    //Hämtar listans längd
64.    @Override
65.    public int getItemCount() {
66.        return list.size();
67.    }
68.
69.
70.    // En klass som ärver RecyclerView där den  hämtar de olika variabler som är skapad i layouten
71.    public class MyViewHolder extends RecyclerView.ViewHolder {
72.
73.        TextView Name, Lastname, Telephonenumber, Password, Lastlogin, Lastlogout, Username;
74.        Button BtnCompetence;
75.
76.        public MyViewHolder(@NonNull View itemView) {
77.            super(itemView);
78.
79.            Name = itemView.findViewById(R.id.Namn);
80.            Lastname = itemView.findViewById(R.id.Efternamn);
81.            Telephonenumber = itemView.findViewById(R.id.Telefonnummer);
82.            Username = itemView.findViewById(R.id.date);
83.            Password = itemView.findViewById(R.id.data);
84.            Lastlogin = itemView.findViewById(R.id.tvid);
85.            Lastlogout = itemView.findViewById(R.id.utloggad);
86.            BtnCompetence = itemView.findViewById(R.id.btnbehörig);
87.
88.        }
89.    }
90. }
91.
```

11.1.15 ViewAdapter2.java

```
1.  package com.example.arian;
2.  import android.content.Context;
3.  import android.content.Intent;
4.  import android.view.LayoutInflater;
5.  import android.view.View;
6.  import android.view.ViewGroup;
7.  import android.widget.ImageView;
8.  import android.widget.LinearLayout;
9.  import android.widget.TextView;
10. import androidx.recyclerview.widget.RecyclerView;
11. import com.bumptech.glide.Glide;
12. import com.bumptech.glide.request.RequestOptions;
13. import java.util.List;
14.
15. public class ViewAdapter2 extends RecyclerView.Adapter<ViewAdapter2.MyViewHolder> {
16.
17.    private Context mContext;
18.    private List<Image> mData;
19.    private RequestOptions option;
20.
21.    //Konstruktor med olika parametrar
22.    public ViewAdapter2(Context mContext, List<Image> mData) {
23.
24.        this.mContext = mContext;
25.        this.mData = mData;
26.        option = new RequestOptions().centerCrop().placeholder(R.drawable.loading_shape).error(R.drawable.loading_shape);
27.    }
28.
```

```
29.    //Metoden bestämmer vilken view som ska användas
30.    @Override
31.    public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
32.
33.        View view;
34.        LayoutInflater inflater = LayoutInflater.from(mContext);
35.        view = inflater.inflate(R.layout.activity_viewadapter2, parent, false);
36.
37.        final MyViewHolder viewHolder = new MyViewHolder(view);
38.        viewHolder.view_container.setOnClickListener(new View.OnClickListener() {
39.            @Override
40.            public void onClick(View v) {
41.
42.                Intent i = new Intent(mContext, DeleteCabinetInfo.class);
43.                i.putExtra("DeletedPerson", mData.get(viewHolder.getAdapterPosition()).getDeletedPerson());
44.                i.putExtra("Image", mData.get(viewHolder.getAdapterPosition()).getImage_url());
45.
46.                mContext.startActivity(i);
47.            }
48.        });
49.
50.        return viewHolder;
51.    }
52.
53.    // Metoden hämtar olika värden från list och lägger in dom i viewns olika variabler
54.    @Override
55.    public void onBindViewHolder(MyViewHolder holder, int position) {
56.
57.        holder.CabinetsName.setText(mData.get(position).getName());
58.        holder.Date.setText(mData.get(position).getDate());
59.        holder.Time.setText(mData.get(position).getTime());
60.        holder.Username.setText(mData.get(position).getOpenBy());
61.        Glide.with(mContext).load(mData.get(position).getImage_url()).apply(option).into(holder.Image);
62.
63.    }
64.
65.    //Hämtar listans längd
66.    @Override
67.    public int getItemCount() {
68.
69.        return mData.size();
70.    }
71.
72.    // En klass som ärver RecyclerView där den  hämtar de olika variabler som är skapad i layouten
73.    public static class MyViewHolder extends RecyclerView.ViewHolder {
74.
75.        TextView CabinetsName;
76.        TextView Date;
77.        TextView Time;
78.        TextView Username;
79.        ImageView Image;
80.        LinearLayout view_container;
81.
82.        public MyViewHolder(View itemView) {
83.            super(itemView);
84.
85.            view_container = itemView.findViewById(R.id.container);
86.            CabinetsName = itemView.findViewById(R.id.Cabinets_Name);
87.            Username = itemView.findViewById(R.id.Username);
88.            Date = itemView.findViewById(R.id.Date);
89.            Time = itemView.findViewById(R.id.Time);
90.            Image = itemView.findViewById(R.id.image);
91.
92.        }
93.    }
94. }
```

11.1.16 ViewAdapter3.java

```
1.  package com.example.arian;
2.  import android.content.Context;
3.  import android.view.LayoutInflater;
4.  import android.view.View;
5.  import android.view.ViewGroup;
6.  import android.widget.LinearLayout;
7.  import android.widget.TextView;
8.  import androidx.recyclerview.widget.RecyclerView;
9.  import com.bumptech.glide.request.RequestOptions;
10. import java.util.List;
11.
12. public class ViewAdapter3 extends RecyclerView.Adapter<ViewAdapter3.MyViewHolder> {
13.
14.   private Context mContext;
15.   private List<Image> list;
16.   private RequestOptions option;
17.
18.   //Konstruktor med olika parametrar
19.   public ViewAdapter3(Context mContext, List<Image> list) {
20.
21.     this.mContext = mContext;
22.     this.list = list;
23.     option = new RequestOptions().centerCrop().placeholder(R.drawable.loading_shape).error(R.drawable.loading_shape);
24.   }
25.
26.   //Metoden bestämmer vilken view som ska användas
27.   @Override
28.   public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
29.
30.     View view;
31.     LayoutInflater inflater = LayoutInflater.from(mContext);
32.     view = inflater.inflate(R.layout.activity_viewadapter3, parent, false);
33.     final MyViewHolder viewHolder = new MyViewHolder(view);
34.
35.     return viewHolder;
36.   }
37.
38.   // Metoden hämtar olika värden från list och lägger in dom i viewns olika variabler
39.   @Override
40.   public void onBindViewHolder(MyViewHolder holder, int position) {
41.
42.     holder.CabinetsName.setText(list.get(position).getName());
43.     holder.Permission.setText(list.get(position).getDate());
44.   }
45.
46.   //Hämtar listans längd
47.   @Override
48.   public int getItemCount() {
49.
50.     return list.size();
51.   }
52.
53.   // En klass som ärver RecyclerView där den  hämtar de olika variabler som är skapad i layouten
54.   public static class MyViewHolder extends RecyclerView.ViewHolder {
55.
56.     TextView CabinetsName;
57.     TextView Permission;
58.     LinearLayout view_container;
59.
60.     public MyViewHolder(View itemView) {
61.       super(itemView);
62.       view_container = itemView.findViewById(R.id.container2);
```

```
            CabinetsName = itemView.findViewById(R.id.cabinetsname);
            Permission = itemView.findViewById(R.id.permission);
        }
    }
}
```

11.1.17 ApiConfig.java

```
package com.example.arian.networking;
import java.util.Map;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.http.Header;
import retrofit2.http.Multipart;
import retrofit2.http.POST;
import retrofit2.http.PartMap;

public interface ApiConfig {

    @Multipart
    @POST("images/Upload_image.php")
    Call<ServerResponse> upload(
            @Header("Authorization") String authorization,
            @PartMap Map<String, RequestBody> map
    );
}
```

11.1.18 AppConfig.java

```
package com.example.arian.networking;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class AppConfig {

    public static String BASE_URL = "https://arian23.000webhostapp.com/Upload_image.php/";

    public static Retrofit getRetrofit() {

        return new Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .build();
    }
}
```

11.1.19 AppConfig.java

```
1.  package com.example.arian.networking;
2.  import com.google.gson.annotations.SerializedName;
3.
4.  public class ServerResponse {
5.
6.    @SerializedName("success")
7.    boolean success;
8.    @SerializedName("message")
9.    String message;
10.
11.   public String getMessage() {
12.     return message;
13.   }
14.
15.   public boolean getSuccess() {
16.     return success;
17.   }
18. }
```

11.2 Xml-filer

11.2.1 activity_deletecabinetinfo.xml

```
1.  <?xml version="1.0" encoding="utf-8"?>
2.  <androidx.coordinatorlayout.widget.CoordinatorLayout
3.      xmlns:android="http://schemas.android.com/apk/res/android"
4.      xmlns:app="http://schemas.android.com/apk/res-auto"
5.      xmlns:tools="http://schemas.android.com/tools"
6.      android:layout_width="match_parent"
7.      android:layout_height="match_parent"
8.      android:background="#303030"
9.      tools:context="com.example.arian.DeleteCabinetInfo"
10.     tools:ignore="MissingDefaultResource">
11.
12.     <androidx.appcompat.widget.Toolbar
13.         android:layout_width="match_parent"
14.         android:layout_height="?attr/actionBarSize"
15.         android:theme="@style/ThemeOverlay.AppCompat.Light"
16.         app:layout_collapseMode="pin">
17.     </androidx.appcompat.widget.Toolbar>
18.
19.     <LinearLayout
20.         android:layout_width="match_parent"
21.         android:layout_height="match_parent"
22.         android:orientation="horizontal">
23.
24.         <com.github.chrisbanes.photoview.PhotoView
25.             android:id="@+id/aa_thumbnail"
26.             android:layout_width="match_parent"
27.             android:layout_height="match_parent"
28.             android:layout_marginTop="50dp"
29.             android:background="@drawable/loading_shape" />
30.     </LinearLayout>
31.
32.     <Button
33.         android:id="@+id/tabort"
34.         android:layout_width="wrap_content"
35.         android:layout_height="wrap_content"
36.         android:layout_weight="1"
37.         android:text="Ta Bort" />
38. </androidx.coordinatorlayout.widget.CoordinatorLayout>
```

11.2.2 activity_deletion.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  tools:context=".Deletion"
  tools:ignore="MissingDefaultResource">

  <androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="540dp"
    android:layout_margin="10dp"
    android:layout_marginTop="10dp"
    android:onClick="HideKeyboard">

    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:layout_marginTop="2dp"
      android:orientation="vertical">

      <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp">

        <TextView
          android:id="@+id/textView"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_marginTop="15dp"
          android:text="Radering"
          android:textColor="@color/colorPrimary"
          android:textSize="30sp"
          android:textStyle="bold" />

        <TextView
          android:id="@+id/txtsignup"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_alignParentRight="true"
          android:onClick="ToLogin"
          android:text="Logga ut"
          android:textColor="@color/colorPrimary"
          android:textSize="20sp"
          android:textStyle="bold" />

      </RelativeLayout>

      <ImageView
        android:id="@+id/imageView"
        android:layout_width="189dp"
        android:layout_height="131dp"
        android:layout_marginStart="50dp"
        android:gravity="center"
        app:srcCompat="@drawable/logo" />

      <EditText
```

```
            android:id="@+id/DED_namn"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:ems="10"
            android:hint="Namn"
            android:inputType="text" />

        <EditText
            android:id="@+id/DED_användarnamn"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:ems="10"
            android:hint="Användarnamn"
            android:inputType="text" />

        <EditText
            android:id="@+id/DED_telefonnummer"
            android:layout_width="match_parent"
            android:layout_height="42dp"
            android:layout_marginTop="15dp"
            android:ems="10"
            android:hint="Telefonnummer"
            android:inputType="number" />

        <Button
            android:id="@+id/btnTabort"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="start"
            android:layout_marginTop="77dp"
            android:background="@color/colorPrimary"
            android:foregroundGravity="center"
            android:gravity="center"
            android:onClick="BtnDelete"
            android:padding="15dp"
            android:text="RADERA"
            android:textColor="@android:color/white"
            android:textSize="20dp" />

    </LinearLayout>

  </androidx.cardview.widget.CardView>

  <com.google.android.material.bottomnavigation.BottomNavigationView
      android:id="@+id/navigation"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginTop="2dp"
      app:itemBackground="@color/colorPrimary"
      app:itemIconTint="@drawable/selector"
      app:itemTextColor="@drawable/selector"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="1.0"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:menu="@menu/bottom_nav_menu" />

</LinearLayout>
```

11.2.3 activity_devicelist.xml

```
1.  <?xml version="1.0" encoding="utf-8"?>
2.  <RelativeLayout
3.   xmlns:android="http://schemas.android.com/apk/res/android"
4.    android:layout_width="fill_parent"
5.    android:layout_height="fill_parent"
6.    android:background="#FFFFFF" >
7.
8.    <TextView
9.       android:id="@+id/text1"
10.      android:layout_width="match_parent"
11.      android:layout_height="50dp"
12.      android:background="@android:drawable/editbox_dropdown_dark_frame"
13.      android:textColor="#F8F0F0"
14.      android:textSize="16dp"
15.      android:textStyle="bold">
16.   </TextView>
17.
18.   <ListView
19.      android:id="@+id/list_devices"
20.      android:layout_width="match_parent"
21.      android:layout_height="wrap_content"
22.      android:layout_alignParentTop="true"
23.      android:layout_above="@+id/button_scan"
24.      android:layout_marginLeft="10dp"
25.      android:layout_marginRight="10dp"
26.      android:layout_marginTop="55dp"
27.      android:smoothScrollbar="true" />
28.
29.   <Button
30.      android:id="@+id/button_scan"
31.      android:layout_width="wrap_content"
32.      android:layout_height="wrap_content"
33.      android:layout_alignParentBottom="true"
34.      android:layout_alignParentLeft="true"
35.      android:layout_alignParentRight="true"
36.      android:textSize="14dp"
37.      android:text="Sök efter nya enheter" />
38.
39. </RelativeLayout>
```

11.2.4 activity_edit.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:orientation="vertical"
  android:layout_width="match_parent"
  tools:context=".Edit"
  android:layout_height="match_parent">

  <androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="543dp"
    android:layout_marginTop="10dp"
    android:layout_margin="10dp"
    android:onClick="HideKeyboard">

    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:orientation="vertical">

      <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:layout_alignParentEnd="true"
          android:layout_marginTop="12dp"
          android:layout_marginEnd="0dp"
          android:text="Redigera person"
          android:textColor="@color/colorPrimary"
          android:textSize="30sp"
          android:textStyle="bold" />

        <TextView
          android:id="@+id/txtsignup"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_alignParentRight="true"
          android:onClick="LogIn"
          android:text="Logga ut"
          android:textColor="@color/colorPrimary"
          android:textSize="20sp"
          android:textStyle="bold" />
      </RelativeLayout>

      <ImageView
        android:id="@+id/imageView2"
        android:layout_width="162dp"
        android:layout_height="65dp"
        android:layout_gravity="center"
        android:layout_marginTop="8dp"
        app:srcCompat="@drawable/logo" />

      <EditText
        android:id="@+id/edr_namn"
        android:layout_width="match_parent"
        android:layout_height="36dp"
```

```
            android:layout_marginTop="1dp"
            android:layout_marginBottom="0dp"
            android:hint="Namn"
            android:inputType="textPersonName"
            android:textSize="14sp" />

        <EditText
            android:id="@+id/edr_telefonnummer"
            android:layout_width="match_parent"
            android:layout_height="36dp"
            android:layout_marginBottom="0dp"
            android:hint="Telefonnummer"
            android:inputType="number"
            android:textSize="14sp" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="26dp"
            android:layout_marginTop="2dp"
            android:text="Till"
            android:textColor="@color/colorPrimary"
            android:textSize="11pt"
            android:textStyle="bold" />

        <EditText
            android:id="@+id/ed_namn"
            android:layout_marginTop="2dp"
            android:layout_width="match_parent"
            android:layout_height="36dp"
            android:layout_marginBottom="0dp"
            android:hint="Namn"
            android:inputType="textPersonName"
            android:textSize="14sp" />

        <EditText
            android:id="@+id/ed_efternamn"
            android:layout_width="match_parent"
            android:layout_height="36dp"
            android:hint="Efternamn"
            android:inputType="textPersonName"
            android:textSize="14sp" />

        <EditText
            android:id="@+id/ed_telefonnummer"
            android:layout_width="match_parent"
            android:layout_height="36dp"
            android:hint="Telefonnummer"
            android:inputType="number"
            android:textSize="14sp" />

        <EditText
            android:id="@+id/ed_email"
            android:layout_width="match_parent"
            android:layout_height="36dp"
            android:hint="Användarnamn"
            android:inputType="textEmailAddress"
            android:textSize="14sp" />

        <EditText
            android:id="@+id/ed_lösenord"
            android:layout_width="match_parent"
            android:layout_height="36dp"
            android:hint="Lösenord"
            android:inputType="textPassword"
            android:textSize="14sp" />

```

```
129.        <RadioGroup
130.            android:id="@+id/ed_group"
131.            android:layout_width="match_parent"
132.            android:layout_height="36dp"
133.            android:layout_marginTop="4dp"
134.            android:orientation="horizontal"
135.            android:textSize="14sp">
136.
137.            <RadioButton
138.                android:id="@+id/ed_adm"
139.                android:layout_width="wrap_content"
140.                android:layout_height="match_parent"
141.                android:layout_alignParentStart="true"
142.                android:layout_marginStart="5dp"
143.                android:checked="false"
144.                android:text="Admin"
145.                android:textSize="14sp" />
146.
147.            <RadioButton
148.                android:id="@+id/ed_anv"
149.                android:layout_width="wrap_content"
150.                android:layout_height="match_parent"
151.                android:layout_alignParentStart="true"
152.                android:layout_marginStart="10dp"
153.                android:checked="false"
154.                android:text="Användare"
155.                android:textSize="14sp" />
156.
157.            <RelativeLayout
158.                android:layout_width="186dp"
159.                android:layout_height="match_parent">
160.
161.                <Button
162.                    android:id="@+id/ed_skåp"
163.                    android:layout_width="68dp"
164.                    android:layout_height="match_parent"
165.                    android:layout_alignParentStart="true"
166.                    android:layout_marginStart="65dp"
167.                    android:layout_marginTop="1dp"
168.                    android:background="@color/colorPrimary"
169.                    android:gravity="center"
170.                    android:text="SkåpsID"
171.                    android:textAlignment="center"
172.                    android:textColor="@android:color/white"
173.                    android:textColorHint="#00FCF7F7"
174.                    android:textSize="14sp" />
175.
176.            </RelativeLayout>
177.        </RadioGroup>
178.
179.        <Button
180.            android:id="@+id/btnUppdatera"
181.            android:layout_width="match_parent"
182.            android:layout_height="wrap_content"
183.            android:layout_alignParentTop="true"
184.            android:layout_alignParentEnd="true"
185.            android:layout_gravity="start"
186.            android:layout_marginTop="12dp"
187.            android:layout_marginEnd="0dp"
188.            android:background="@color/colorPrimary"
189.            android:foregroundGravity="center"
190.            android:gravity="center"
191.            android:onClick="BtnUpdate"
192.            android:padding="15dp"
193.            android:text="Redigera"
194.            android:textColor="@android:color/white"
```

```
            android:textSize="20dp" />
        </LinearLayout>
    </androidx.cardview.widget.CardView>

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/navigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        app:itemBackground="@color/colorPrimary"
        app:itemIconTint="@drawable/selector"
        app:itemTextColor="@drawable/selector"
        app:menu="@menu/bottom_nav_menu" />
</LinearLayout>
```

11.2.5 activity_information.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:onClick="HideKeyboard"
    android:orientation="vertical"
    tools:context=".Information">

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Sök"
        android:inputType="textPersonName"
        tools:ignore="MissingConstraints" />

    <Button
        android:id="@+id/buttonInfo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="4dp"
        android:layout_marginStart="20dp"
        android:background="@color/colorPrimary"
        android:gravity="center"
        android:onClick="closeKeyboard"
        android:text="Hämta Informaion"
        android:textAlignment="gravity"
        android:textColor="#FCFAFA"
        tools:ignore="MissingConstraints,OnClick" />

    <Button
        android:id="@+id/buttonSkap"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="4dp"
        android:layout_marginStart="20dp"
        android:background="@color/colorPrimary"
        android:gravity="center"
        android:onClick="closeKeyboard"
        android:text="Hämta Skåpinfo"
        android:textAlignment="gravity"
        android:textColor="#FCFAFA"
        tools:ignore="OnClick" />

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerviewid"
        android:layout_width="match_parent"
        android:layout_height="397dp"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp">

    </androidx.recyclerview.widget.RecyclerView>

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/navigation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="6dp"
```

```
63.         app:itemBackground="@color/colorPrimary"
64.         app:itemIconTint="@drawable/selector"
65.         app:itemTextColor="@drawable/selector"
66.         app:layout_constraintBottom_toBottomOf="parent"
67.         app:layout_constraintEnd_toEndOf="parent"
68.         app:layout_constraintHorizontal_bias="1.0"
69.         app:layout_constraintLeft_toLeftOf="parent"
70.         app:layout_constraintRight_toRightOf="parent"
71.         app:menu="@menu/bottom_nav_menu" />
72.
73.  </LinearLayout>
74.
```

11.2.6 activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:layout_margin="10dp"
  android:orientation="vertical"
  tools:context=".Login">

  <androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#007A7878"
    android:backgroundTint="@color/cardview_light_background"
    android:baselineAligned="false"
    android:elevation="4dp"
    app:cardUseCompatPadding="true">

    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_margin="15dp"
      android:onClick="HideKeyboard"
      android:orientation="vertical">

      <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:gravity="center">

        <TextView
          android:id="@+id/textView"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_gravity="right"
          android:text="Välkommen"
          android:textColor="@android:color/black"
          android:textSize="40sp"
          android:textStyle="bold" />

      </RelativeLayout>

      <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:fontFamily="sans-serif-light"
        android:text="Logga in för att forsätta"
        android:textSize="18sp"
        android:textStyle="bold" />

      <ImageView
        android:id="@+id/imageView"
        android:layout_width="190dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="45dp"
        app:srcCompat="@drawable/logo" />

      <EditText
```

```
            android:id="@+id/ed_email"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:ems="10"
            android:hint="Email"
            android:inputType="text" />

        <EditText
            android:id="@+id/ed_lösenord"
            android:layout_width="match_parent"
            android:layout_height="42dp"
            android:layout_marginTop="20dp"
            android:ems="10"
            android:hint="Lösenord"
            android:inputType="textPassword" />

        <Button
            android:id="@+id/btn_login"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="105dp"
            android:background="@color/colorPrimary"
            android:foregroundGravity="center"
            android:gravity="center"
            android:onClick="LogIn"
            android:padding="15dp"
            android:text="Logga in"
            android:textColor="@android:color/white"
            android:textSize="20dp" />

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="40dp">

        </com.google.android.material.textfield.TextInputLayout>

    </LinearLayout>
  </androidx.cardview.widget.CardView>
</LinearLayout>

```

11.2.7 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:layout_margin="10dp"
  android:orientation="vertical"
  tools:context=".MainActivity">

 <androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#007A7878"
    android:backgroundTint="@color/cardview_light_background"
    android:baselineAligned="false"
    android:elevation="4dp"
    app:cardUseCompatPadding="true">

    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_margin="15dp"
      android:orientation="vertical">

      <ImageView
        android:id="@+id/ble"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/ic_bluetooth_disabled_black_24dp" />

      <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:text="Välkommen"
        android:textColor="@android:color/black"
        android:textSize="34sp"
        android:textStyle="bold" />

      <TextView
        android:id="@+id/txtname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="2dp"
        android:gravity="center"
        android:text="TextView"
        android:textColor="@android:color/black"
        android:textSize="20sp"
        android:textStyle="bold" />

      <ImageView
        android:id="@+id/lockimage"
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:layout_gravity="center"
        android:layout_marginTop="20dp"
        app:srcCompat="@drawable/ic_lock_outline_black_24dp" />
```

```
63.
64.         <TextView
65.             android:id="@+id/skapid"
66.             android:layout_width="wrap_content"
67.             android:layout_height="wrap_content"
68.             android:layout_gravity="center"
69.             android:text="TextView"
70.             android:textColor="@android:color/black"
71.             android:textSize="12dp"
72.             android:textStyle="bold">
73.         </TextView>
74.
75.         <Button
76.             android:id="@+id/connect"
77.             android:layout_width="355dp"
78.             android:layout_height="50dp"
79.             android:layout_gravity="center"
80.             android:layout_marginTop="10dp"
81.             android:text="Anslutninng till skåp" />
82.
83.         <Button
84.             android:id="@+id/on"
85.             android:layout_width="355dp"
86.             android:layout_height="50dp"
87.             android:layout_alignParentTop="true"
88.             android:layout_gravity="center"
89.             android:layout_marginTop="10dp"
90.             android:text="Öppna skåp" />
91.
92.         <Button
93.             android:id="@+id/off"
94.             android:layout_width="355dp"
95.             android:layout_height="50dp"
96.             android:layout_gravity="center"
97.             android:layout_marginTop="10dp"
98.             android:text="Stäng skåp" />
99.
100.        <Button
101.            android:id="@+id/image"
102.            android:layout_width="355dp"
103.            android:layout_height="50dp"
104.            android:layout_alignParentTop="true"
105.            android:layout_gravity="center"
106.            android:layout_marginTop="10dp"
107.            android:text="Ta Bild" />
108.
109.        <Button
110.            android:id="@+id/button"
111.            android:layout_width="355dp"
112.            android:layout_height="66dp"
113.            android:layout_gravity="center"
114.            android:layout_marginLeft="5dp"
115.            android:layout_marginTop="25dp"
116.            android:layout_marginRight="5dp"
117.            android:background="@color/colorPrimary"
118.            android:gravity="center"
119.            android:onClick="LoginActivity"
120.            android:text="Logga ut"
121.            android:textColor="@android:color/white"
122.            android:textSize="18sp" />
123.    </LinearLayout>
124. </androidx.cardview.widget.CardView></LinearLayout>
```

11.2.8 activity_register.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".Register"
  tools:ignore="MissingDefaultResource">

  <androidx.cardview.widget.CardView
    android:id="@+id/cardView"
    android:layout_width="match_parent"
    android:layout_height="540dp"
    android:layout_margin="10dp"
    android:layout_marginTop="15dp"
    android:onClick="HideKeyboard"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    tools:ignore="MissingConstraints"
    tools:layout_editor_absoluteY="29dp">

    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:layout_marginTop="10dp"
      android:orientation="vertical">

      <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="2dp">

        <TextView
          android:id="@+id/textView"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_marginTop="10dp"

          android:text="Registrering"
          android:textColor="@color/colorPrimary"
          android:textSize="30sp"
          android:textStyle="bold" />

        <TextView
          android:id="@+id/txtsignup"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_alignParentRight="true"
          android:onClick="ToLogin"
          android:text="Logga ut"
          android:textColor="@color/colorPrimary"
          android:textSize="20sp"
          android:textStyle="bold" />

      </RelativeLayout>

      <ImageView
        android:id="@+id/imageView2"
        android:layout_width="162dp"
```

```
63.             android:layout_height="85dp"
64.             android:layout_gravity="center"
65.             android:layout_marginTop="12dp"
66.             app:srcCompat="@drawable/logo" />
67.
68.         <com.google.android.material.textfield.TextInputLayout
69.             android:layout_width="match_parent"
70.             android:layout_height="42dp"
71.             android:layout_marginTop="5dp">
72.
73.             <EditText
74.                 android:id="@+id/ed_namn"
75.                 android:layout_width="match_parent"
76.                 android:layout_height="35dp"
77.                 android:layout_marginTop="20dp"
78.                 android:hint="Namn"
79.                 android:inputType="textPersonName"
80.                 android:textSize="14sp" />
81.         </com.google.android.material.textfield.TextInputLayout>
82.
83.         <EditText
84.             android:id="@+id/ed_efternamn"
85.             android:layout_width="match_parent"
86.             android:layout_height="35dp"
87.             android:layout_marginTop="10dp"
88.             android:layout_marginBottom="0dp"
89.             android:hint="Efternamn"
90.             android:inputType="textPersonName"
91.             android:textSize="14sp" />
92.
93.         <EditText
94.             android:id="@+id/ed_telefonnummer"
95.             android:layout_width="match_parent"
96.             android:layout_height="35dp"
97.             android:layout_marginTop="10dp"
98.             android:hint="Telefonnummer"
99.             android:inputType="number"
100.            android:textSize="14sp" />
101.
102.        <EditText
103.            android:id="@+id/ed_email"
104.            android:layout_width="match_parent"
105.            android:layout_height="35dp"
106.            android:layout_marginTop="10dp"
107.            android:layout_marginBottom="0dp"
108.            android:hint="Användarnamn"
109.            android:inputType="textPersonName"
110.            android:textSize="14sp" />
111.
112.        <EditText
113.            android:id="@+id/ed_lösenord"
114.            android:layout_width="match_parent"
115.            android:layout_height="35dp"
116.            android:layout_marginTop="10dp"
117.            android:ems="10"
118.            android:hint="Lösenord"
119.            android:inputType="textPassword"
120.            android:textSize="14sp" />
121.
122.        <RadioGroup
123.            android:id="@+id/ed_group"
124.            android:layout_width="match_parent"
125.            android:layout_height="35dp"
126.            android:layout_marginTop="20dp"
127.            android:orientation="horizontal">
128.
```

```
129.            <RadioButton
130.               android:id="@+id/ed_adm"
131.               android:layout_width="wrap_content"
132.               android:layout_height="match_parent"
133.               android:layout_marginRight="10dp"
134.               android:textSize="16sp"
135.               android:checked="false"
136.               android:text="Admin" />
137.
138.            <RadioButton
139.               android:id="@+id/ed_anv"
140.               android:layout_width="wrap_content"
141.               android:layout_height="match_parent"
142.               android:textSize="16sp"
143.               android:checked="false"
144.               android:text="Användare" />
145.
146.            <RelativeLayout
147.               android:layout_width="186dp"
148.               android:layout_height="50dp"
149.               android:layout_gravity="center">
150.
151.               <Button
152.                  android:id="@+id/ed_skåp"
153.                  android:layout_width="70dp"
154.                  android:layout_height="70dp"
155.                  android:layout_alignParentStart="true"
156.                  android:layout_marginStart="54dp"
157.                  android:layout_marginLeft="10dp"
158.                  android:layout_marginTop="5dp"
159.                  android:background="@color/colorPrimary"
160.                  android:gravity="center"
161.                  android:text="Skåp-ID"
162.                  android:textAlignment="center"
163.                  android:textColor="@android:color/white"
164.                  android:textColorHint="#00FCF7F7"
165.                  android:textSize="16sp" />
166.
167.            </RelativeLayout>
168.         </RadioGroup>
169.
170.         <RelativeLayout
171.            android:layout_width="match_parent"
172.            android:layout_height="wrap_content">
173.
174.            <Button
175.               android:id="@+id/btn_register"
176.               android:layout_width="match_parent"
177.               android:layout_height="64dp"
178.               android:layout_alignParentTop="true"
179.               android:layout_alignParentEnd="true"
180.               android:layout_marginTop="25dp"
181.               android:layout_marginEnd="0dp"
182.               android:background="@color/colorPrimary"
183.               android:onClick="Register"
184.               android:text="Registrera"
185.               android:textColor="@android:color/white"
186.               android:textSize="20dp" />
187.
188.         </RelativeLayout>
189.      </LinearLayout>
190.
191.   </androidx.cardview.widget.CardView>
192.
193.   <com.google.android.material.bottomnavigation.BottomNavigationView
194.      android:id="@+id/navigation"
```

```
195.     android:layout_width="match_parent"
196.     android:layout_height="wrap_content"
197.     android:layout_marginTop="2dp"
198.     app:itemBackground="@color/colorPrimary"
199.     app:itemIconTint="@drawable/selector"
200.     app:itemTextColor="@drawable/selector"
201.     app:layout_constraintBottom_toBottomOf="parent"
202.     app:layout_constraintEnd_toEndOf="parent"
203.     app:layout_constraintHorizontal_bias="1.0"
204.     app:layout_constraintLeft_toLeftOf="parent"
205.     app:layout_constraintRight_toRightOf="parent"
206.     app:menu="@menu/bottom_nav_menu" />
207.
208. </LinearLayout>
209.
```

11.2.9 activity_viewadapter1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_marginTop="2dp"
  android:id="@+id/container"
  android:layout_height="200dp"
  android:orientation="vertical"
  tools:context=".Listview">

  <androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:onClick="HideKeyboard">

    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginTop="4dp"
      android:layout_marginLeft="4dp"
      android:layout_marginRight="4dp"
      android:layout_marginBottom="4dp"
      android:background="@drawable/bgg"
      android:orientation="vertical"
      tools:ignore="MissingConstraints">

      <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="35dp">
        <TextView
          android:id="@+id/Namn"
          android:layout_width="wrap_content"
          android:layout_height="match_parent"
          android:layout_marginStart="75dp"
          android:textAppearance="@style/TextAppearance.AppCompat"
          android:textColor="#FDFAFA"
          android:textSize="11pt"
          android:textStyle="bold" />

        <TextView
          android:id="@+id/Efternamn"
          android:layout_width="wrap_content"
          android:layout_height="match_parent"
          android:layout_marginStart="5dp"
          android:textColor="#FDFAFA"
          android:layout_marginLeft="6dp"
          android:layout_marginTop="0dp"
          android:layout_toEndOf="@id/Namn"
          android:layout_toRightOf="@id/Namn"
          android:lineSpacingExtra="1dp"
          android:textAppearance="@style/TextAppearance.AppCompat"
          android:textSize="11pt"
          android:textStyle="bold" />
      </RelativeLayout>

      <TextView
        android:id="@+id/date"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:textColor="#FDFAFA"
```

```
63.             android:layout_gravity="left"
64.             android:layout_marginLeft="3dp"
65.             android:layout_marginTop="8dp"
66.             android:textAppearance="@style/TextAppearance.AppCompat" />
67.
68.         <TextView
69.             android:id="@+id/data"
70.             android:layout_width="wrap_content"
71.             android:layout_height="match_parent"
72.             android:textColor="#FDFAFA"
73.             android:layout_gravity="left"
74.             android:layout_marginLeft="3dp"
75.             android:layout_marginTop="10dp"
76.             android:gravity="center"
77.             android:textAppearance="@style/TextAppearance.AppCompat" />
78.
79.         <TextView
80.             android:id="@+id/Telefonnummer"
81.             android:layout_width="wrap_content"
82.             android:layout_height="match_parent"
83.             android:textColor="#FDFAFA"
84.             android:layout_gravity="left"
85.             android:layout_marginLeft="3dp"
86.             android:layout_marginTop="10dp"
87.             android:textAppearance="@style/TextAppearance.AppCompat" />
88.
89.         <TextView
90.             android:id="@+id/tvid"
91.             android:layout_width="wrap_content"
92.             android:layout_height="wrap_content"
93.             android:layout_gravity="left"
94.             android:textColor="#FDFAFA"
95.             android:layout_marginLeft="3dp"
96.             android:layout_marginTop="10dp"
97.             android:textAppearance="@style/TextAppearance.AppCompat" />
98.
99.         <RelativeLayout
100.            android:layout_width="match_parent"
101.            android:layout_height="wrap_content">
102.
103.        <TextView
104.            android:id="@+id/utloggad"
105.            android:layout_width="wrap_content"
106.            android:layout_height="wrap_content"
107.            android:layout_gravity="left"
108.            android:layout_marginLeft="3dp"
109.            android:textColor="#FDFAFA"
110.            android:layout_marginTop="10dp"
111.            android:layout_marginBottom="8dp"
112.            android:textAppearance="@style/TextAppearance.AppCompat" />
113.
114.            <Button
115.                android:id="@+id/btnbehörig"
116.                android:layout_width="wrap_content"
117.                android:layout_height="match_parent"
118.                android:layout_marginStart="230dp"
119.                android:text="Behörig">
120.            </Button>
121.        </RelativeLayout>
122.    </LinearLayout>
123.  </androidx.cardview.widget.CardView>
124. </LinearLayout>
```

11.2.10 activity_viewadapter2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/container"
  android:layout_width="match_parent"
  android:layout_height="150dp"
  android:layout_marginTop="5dp"
  android:background="@drawable/bgg"
  android:orientation="horizontal"
  android:padding="8dp">

  <ImageView
    android:id="@+id/image"
    android:layout_width="161dp"
    android:layout_height="match_parent"
    android:background="@drawable/loading_shape" />

  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="130dp"
    android:layout_margin="8dp"
    android:orientation="vertical">

    <TextView
      android:id="@+id/Cabinets_Name"
      android:layout_width="119dp"
      android:layout_height="wrap_content"
      android:text="Skåp"
      android:textColor="#fff"
      android:textSize="18sp"
      android:textStyle="bold" />

    <TextView
      android:id="@+id/Username"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="6dp"
      android:text="Användare"
      android:textColor="#fff" />

    <TextView
      android:id="@+id/Date"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="6dp"
      android:text="Datum"
      android:textColor="#fff" />

    <TextView
      android:id="@+id/Time"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="6dp"
      android:text="Tidpunkt"
      android:textColor="#fff" />

  </LinearLayout>
</LinearLayout>

```

11.2.11 activity_viewadapter3.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  android:id="@+id/container2"
  android:layout_width="match_parent"
  android:layout_height="80dp"
  android:background="@color/common_google_signin_btn_text_light_disabled"
  android:orientation="vertical">

  <androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#007A7878"
    android:backgroundTint="@color/cardview_light_background"
    android:baselineAligned="false"
    android:elevation="4dp"
    app:cardUseCompatPadding="true">

    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:orientation="vertical">

      <TextView
        android:id="@+id/cabinetsname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:text="Användare"
        android:textColor="#0B0A0A"
        android:textSize="18sp"
        android:textStyle="bold" />

      <TextView
        android:id="@+id/permission"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:text="Användare"
        android:textColor="#0B0A0A"
        android:textSize="18sp"
        android:textStyle="bold" />
    </LinearLayout>

  </androidx.cardview.widget.CardView>

</LinearLayout>

```

11.2.12 cabinets.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayou
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.arian.DeleteCabinetInfo"
    android:orientation="vertical">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerviewid2"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="#656565">
    </androidx.recyclerview.widget.RecyclerView>

</LinearLayout>
```

11.2.13 customdialog.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:padding="50dp"  >

  <ProgressBar
    android:id="@+id/progressBar"
    style="?android:attr/progressBarStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="Laddar..."
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/progressBar" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

11.2.14 device_name.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TextView
   xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:textColor="#222222"
  android:textSize="14dp"
  android:text="Text" />
```

11.3 Android Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  package="com.example.arian">

  <uses-permission android:name="android.permission.BLUETOOTH" />
  <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-feature android:name="android.hardware.camera" android:required="true" />
  <uses-permission android:name="android.permission.CAMERA" />
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
  <uses-permission android:name="android.permission.WRITE_INTERNAL_STORAGE" />
  <uses-permission android:name="android.permission.READ_INTERNAL_STORAGE" />


  <application

    android:allowBackup="true"
    android:icon="@drawable/inloggning"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme"
    tools:ignore="GoogleAppIndexingWarning">
    <activity android:name=".Login">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <activity android:name=".ScanCode" />
    <activity android:name=".Deletion" />
    <activity android:name=".Edit" />
    <activity android:name=".Image" />
    <activity android:name=".DeviceList" />
    <activity android:name=".Listview" />
    <activity android:name=".Information" />
    <activity android:name=".DeleteCabinetInfo"/>
    <activity android:name=".CabinetsPermission"/>
    <activity android:name=".MainActivity" />
    <activity android:name=".Register" />

    <provider
      android:name="androidx.core.content.FileProvider"
      android:authorities="${applicationId}.provider"
      android:exported="false"
      android:grantUriPermissions="true">
      <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/provider_paths" />
    </provider>

  </application>
</manifest>
```

Printed by Tryckeriet i E-huset, Lund 2020

LUND
UNIVERSITY

Series of Bachelor´s theses
Department of Electrical and Information Technology
LU/LTH-EIT 2020-787
http://www.eit.lth.se