Digitalisering av evenemangshantering

LUNDS UNIVERSITET
Campus Helsingborg

Lunds Tekniska Högskola vid Campus Helsingborg
Institutionen för datavetenskap

Muhammed Ar
EDAL05 - Examensarbete i datavetenskap

LTH 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 2026

Sammanfattning

En verksamhet för evenemang som tidigare drevs osystematiskt och med hjälp av många osammanhängande manuella eller analoga rutiner analyseras. Verksamheten hyrde ut produkter och fullständiga evenemang.

Utifrån analysen görs en plan för digitaliseringen. Ett ramverk utvärderas och sedan byggs ett digitalt system som täcker de flesta tidigare manuella/analoga rutiner. Det nya systemet hanterar allt från kundens bokningsprocess och uppsättning av evenemangen, till lagerhantering, säkerhetskontroller och uppföljning av evenemangen.

Genom det nya systemet förbättras processen väldigt mycket. Bland annat så minskas säkerhetsriskerna och lagret behålls välhanterat. Tack vare det så drivs verksamheten mer säkert och med ordentlig hantering av lagret vilket förhindrar att produkter och redskap tappas bort och att onödiga kontroller undviks.

Examensarbetet testar praktiskt en digitaliseringsprocess och resulterar i en lyckad metod för digitalisering som även kan användas för andra digitaliseringsprojekt.

Nyckelord: digitalisering, modellering, evenemang, säkerhet, prototyputveckling

Abstract

A business that was previously run unsystematically and with the help of many incoherent manual or analog routines is analyzed. The business rented out products and complete events.

Through the analysis, a plan for digitization is made. A framework is evaluated and subsequently a digital system that covers most of the previously manual/analog routines is built. The new system manages everything from the customer's booking process and setup of the events, to storage management, security controls, and follow-up of the events.

Through the new system, the process is vastly improved, and among other things, security risks are lowered and the inventory is kept well-managed. Thanks to that, the business is run safer and with proper controls of the inventory which avoids loss of products and items and also avoids unnecessary controls.

The thesis tests a digitization process practically and results in a successful method for digitization which can be applied for other digitization projects as well.

Keywords: digitization, modeling, events, security, prototyping

Förord

Jag vill tacka min handledare Roger Henriksson för all hjälp han gett mig och hans stöd genom hela arbetet.

Jag vill också tacka min examinator Mathias Haage för att ha varit så flexibel och tillmötesgående genom kampen att få till en bra problemformulering.

Dessutom vill jag tacka Acke Salem från Edument AB för att ha kommit på projektidén och gett mig en chans att utföra den.

Slutligen vill jag tacka min familj och våra bekanta för att ha stöttat mig och hjälpt mig hitta ett examensarbete.

Innehållsförteckning

1 Inledning....8
1.1 Bakgrund....8
1.2 Syfte....8
1.3 Målformulering....8
1.4 Problemformulering....9
1.5 Motivering av examensarbetet....9
1.6 Avgränsningar....10
1.6.1 Ingen rekrytering och automatisk passplanering....10
1.6.2 Betallösning integreras inte....10
1.6.3 Databashanterare utvärderas inte....10
2 Teknisk bakgrund....11
2.1 Databaser....11
2.2 Spring....11
2.3 Vaadin....12
2.4 Säkerhet....12
3 Metod....13
3.1 Förstudie....13
3.2 Testimplementation....13
3.3 Implementation av prototypen....14
3.4 Testning och utvärdering av prototypen....14
3.5 Källkritik....14
4 Analys....16
4.1 Tidigare process....16
4.2 Initial datamodellering....18
4.3 Utvärdering av lämplig teknikstack....19
4.4 Slutlig datamodell....20
4.5 Risker med digitalisering....21
4.6 Autentisering....21
4.7 Åtkomstkontroll....22
5 Resultat....24
5.1 Prototypen....24
5.2 Den nya processen....24
5.3 Feedback från utvärdering av prototypen....28
5.4 Nya datapunkter....28
5.5 Diskussion....28
6 Slutsats....30
6.1 Sammanfattning....30

6.2 Svar på problemformuleringen.....30
6.3 Etiska aspekter.....31
6.4 Framtida utvecklingsmöjligheter.....31
7 Källförteckning.....32
8 Bilagor.....33

1 Inledning

Detta kapitel beskriver arbetets inledning.

1.1 Bakgrund

Examensarbetet görs på uppdrag av Edument AB. Edument är ett programvaruföretag som grundades år 2010 som ett utvecklingsbolag för teknikhus i Sverige som förflyttade sig från licensierad programvara till att arbeta med öppen källkod. Företaget utvecklar proof of concept för stora och små företag samt investerar i små start-uppar både tekniskt och finansiellt. Den verkställande direktören för Edument var en av de ansvariga för föregångaren till Jumpster.

Analysen och arbetet görs för Jumpster, som är ett evenemangsbolag som hyr ut både produkter och fullständiga evenemang som de ifall kunden önskar det även sätter upp och driver evenemanget. Tidigare hanterades bokning, resursplanering, personalplanering, utrustningshantering, schemaläggning och uppföljning via manuella eller analoga flöden.

Problemen som uppstod var bland annat att utrustning kom bort, att skador inte dokumenterades väl vilket ledde till att överflödiga kontroller krävdes på all utrustning, att evenemangs lönsamhet inte kunde analyseras och slutligen hög ineffektivitet.

Examensarbetet löser dessa problem genom att digitalisera processen och utveckla ett enhetligt system som används av alla tre grupper: administratörerna, evenemangspersonalen och kunderna. Genom att alla tre användningsområden är integrerade och hanteras i ett och samma system löses den undermåliga processen.

1.2 Syfte

Undersöka arbetsprocesser, rutiner och system för en befintlig verksamhet som drivits genom manuella eller analoga rutiner. Utifrån undersökningen modellera och designa ett fullständigt och digitalt system för hela processen för evenemangshantering.

1.3 Målformulering

Processerna ska digitaliseras och en enhetlig plattform för alla tre användargrupper ska designas, modelleras och implementeras. Systemet ska innehålla dessa funktioner:

För kunder:

- Bokning via webb
- Bokningsbekräftelse via e-post

För evenemangspersonalen:

- Förberedelse och avslutning av evenemang
  - Hantering av nödvändig utrustning
  - Rapportering av skador på utrustningen
- Kalender

För administratörer:

- Lagerhantering och inventering
- Uppsättning av färdiga paket
- Hantering av personalens konton

Det är dessa tre användargrupper som finns och varje användargrupp får sin egen vy & del i systemet.

Målens uppfyllelse mäts genom en jämförelse av den tidigare processens och det nya systemets effektivitet, detta görs genom testning av det nya systemet följt av intervju med personer som arbetat med den tidigare processen.

Uppfyllelsen mäts vidare genom en jämförelse av mängden datapunkter som lagras. Detta innebär att om datapunkter och detaljer som tidigare inte lagrades nu lagras så är det en förbättring.

1.4 Problemformulering

Övergripande fråga: Hur kan ett enhetligt digitalt system byggas för separata manuella eller analoga rutiner och processer?

1. Vilka rutiner och processer finns just nu?
2. Hur går processen till, från början till slut?
3. Vilka relationer till varandra har de olika rutinerna?
4. Vilka brister och risker finns med ett digitalt system? (t.ex. säkerhetsaspekter)
5. Hur ser en digitaliserad process ut? Vad skiljer sig mot den manuella, vilka vinster uppnås?

1.5 Motivering av examensarbetet

Att genom digitalisering öka spårbarhet och användarvänlighet och uppnå en systematisk process för evenemangshantering.

Ett nuvarande problem är att utrustningen inte dokumenteras väl. Flera olika enheter av samma produkt separeras inte från varandra, och deras skick dokumenteras inte. Det leder till att utrustning ibland kommer bort och att det inte finns en tydlig bild över lagerhållningen.

Den nuvarande processen är inte transparant och spårbar. Hanteringen av pengarna görs manuellt och därför händer det att pengar kommer bort. Eftersom lagret inte hanteras

väl är översikten mycket bristande och saknad utrustning märks endast vid en kontroll som görs på slutet av varje år.

Inom evenemangsuthyrning (inte endast denna verksamhet) utförs spårning och hantering av skador på produkter ofta inte systematiskt. Genom digitaliseringen ska spårbarheten för skador ökas och detta leder till en möjlig implementation för det. På detta vis gynnas både andra företag men också omgivande samhälle genom att evenemang kan drivas säkrare och med ordentliga kontroller som även förhindrar överflödiga kostnader i form av onödiga kontroller (som tidigare gjordes eftersom det inte var tydligt vad som hade kontrollerats eller inte).

Examensarbetet bidrar till kunskapsutvecklingen genom att en metod för digitalisering beskrivs och testas. Från början till slut utvecklas och genomförs en utförlig metod för digitalisering som åter kan genomföras för andra digitaliseringsprojekt.

1.6 Avgränsningar

1.6.1 Ingen rekrytering och automatisk passplanering

Automatisk personal-/passplanering och rekrytering utvecklas inte. Administratörer tilldelar manuellt evenemangspersonal till evenemang och rekryteringsprocessen hanteras helt utanför systemet. Detta eftersom ett externt system har hand om dessa uppgifter, och det är det som – om det behövs – kommer att kunna integreras med detta system (framtida utvecklingsmöjligheter).

1.6.2 Betallösning integreras inte

En betallösning integreras inte, i prototypen görs bokningar utan att användaren behöver betala och inga betaluppgifter kan matas in. Detta lämnas utanför prototypen och är en del av vad som kan vidareutvecklas efteråt. Detta avgränsar också hantering av moms. Anledningen till att detta avgränsas är att det är utanför projektets omfattning och implementeras vanligtvis med hjälp av externa tjänster.

1.6.3 Databashanterare utvärderas inte

Databashanterare (DBMS) utvärderas inte. Ett bibliotek som stöder de flesta databashanterarna används och valet lämnas åt klienten. Det finns många SQL-databashanterare och en stor del av dem stöder JDBC, så genom att endast använda standard SQL blir det flexibelt vilken DBMS som ska användas.

2 Teknisk bakgrund

Detta kapitel beskriver tekniker och metoder som använts under examensarbetet.

2.1 Databaser

En datamodell eller databasmodell är en specifikation som beskriver hur ett objekt/record/entitet ser ut. Inom SQL-databaser kallas dessa objekt för tabeller och instanser av objekten kallas för rader. Datamodeller beskriver vilka fält (attribut / egenskaper / kolumner) objekten innehåller (och ibland deras typ) och objekts relation till varandra. Datamodeller brukar visas med hjälp av diagram.

Javas standardbibliotek innehåller API:n JDBC (Java Database Connectivity) som fungerar som ett standardgränssnitt för att använda en mängd databashanterare. När JDBC används med SQL-databaser anges SQL-sökfrågorna och värdena matas in. JDBC fungerar som ett tunt lager över SQL-databasen och erbjuder därför inte serialisering och deserialisering av egna Java-objekt.

För att förenkla användningen av databaser kan JPA (Java Persistence API / Jakarta Persistence) användas. Det är en specifikation med flera implementationer, och den förenklar användningen av databaser. Med JPA skrivs Java-klasser för tabeller, och implementationen tar hand om att konvertera mellan Java-objektet och SQL-representationen [1].

Databashanterare (DBMS, database management system) är programvaror för att hantera databaser. Det är dessa programvaror som tar emot sökfrågor och andra kommandon, och som genom dem lagrar och slår upp databaser. Vissa databashanterare fungerar som servrar som tar emot anslutningar från applikationer, medan andra fungerar som inbäddade bibliotek och lagrar data direkt i en eller flera filer utan en extern serverprocess. Ibland kallas databashanterare kort för databas, men med databas menas mer exakt en samling data.

2.2 Spring

Spring Framework är ett ramverk för Java-applikationer. Ramverket erbjuder en komplett modell för programmering och konfigurering av moderna Java-applikationer [2].

Spring Framework används ofta med Spring Boot, som bygger på Spring Framework och automatiserar konfigureringen av olika delar av Spring eller specifika Spring-bibliotek, för att förenkla utvecklingen av Spring-baserade applikationer [3].

Spring erbjuder ett stort antal bibliotek för utveckling av applikationer, ett av dessa är Spring Data. Spring Data är en familj av bibliotek som erbjuder konsistenta

programmeringsmodeller för hantering av data [4]. Ett av biblioteken ur denna familj är Spring Data JPA som är en implementation av JPA [5].

Ett annat Spring-bibliotek är Spring Security, som används för att hantera bland annat autentisering och åtkomstkontroll [6]. Spring Security erbjuder flera olika autentiseringsmetoder, till exempel lösenordbaserad autentisering och One-Time Token (OTT, engångstoken/kod), och även möjligheten att implementera nya metoder [7].

2.3 Vaadin

Vaadin är ett ramverk för utveckling av Java-webbappar. Vaadin används för att implementera både back-end och front-end och webbapparna kan skrivas fullständigt i Java när Vaadin Flow används. Klienten och servern integreras tätt med Vaadin Flow, användargränsnittets kod skrivs i Java och skickas automatiskt över till klienten (webbläsaren), sedan kommunicerar klienten och servern med varandra för att hantera uppdateringar och händelser [8]. Vaadin brukar användas med Spring Boot men kan också användas självständigt.

2.4 Säkerhet

Inom säkerhet kallas entiteter (t.ex. användare, processer eller enheter) som begär åtkomst till och som hanterar resurser för subjekt. Resurserna som hanteras kallas i sin tur för objekt.

Åtkomstkontroller är olika metoder som används för att kontrollera vilka subjekt som kan komma åt vilka objekt och vad de kan göra med dem.

Två relevanta metoder för åtkomstkontroll är följande:

Attributbaserad åtkomstkontroll (ABAC) är en metod där subjekten eller objekten (eller båda) har attribut som används i en algoritm för att bestämma rättigheter.

Rollbaserad åtkomstkontroll (RBAC) är en metod där subjekten har en eller flera roller, och sedan har rollerna åtkomst till olika objekt (åtkomsten kan också bestå av olika rättigheter, t.ex. läsa, skriva, radera) [9].

3 Metod

Detta är en kvalitativ och deduktiv studie. I detta kapitel beskrivs genomförandet av examensarbetet.

3.1 Förstudie

För att samla in information om den tidigare processen och krav för det nya systemet så användes tre metoder.

Den första metoden som användes var att intervjua. Detta gjordes genom att frågor förbereddes inför intervjun och sedan gjordes en fysisk intervju med röstinspelning. För att erhålla en tillräcklig mängd information ställdes både generella och specifika frågor om processen och den typ av intervju som gjordes var semistrukturerad. En semistrukturerad intervju går ut på att det finns ett antal frågor som utgångspunkt, och att det sedan ställs följdfrågor och fördjupande frågor. Eftersom inte mycket var känt om den tidigare processen behövdes ett sådant flexibelt sätt att intervjua.

Den andra metoden som användes var att frågor skrevs och skickades in för att få svar. Detta gjordes eftersom kontaktpersonen inte hade möjlighet och tid för intervju tillräckligt ofta.

Den tredje metoden som användes var analys av gamla dokument som hade använts i verksamheten. Eftersom mängden historik som kvarstod från tidigare arbete var låg så kunde inte denna metod användas mycket.

På grund av att verksamheten inte är aktiv just nu kunde inte observationer göras för att samla in information om den tidigare processen, utan dessa tre metoder fick användas.

Efter förstudierna gjordes en analys av hur processen kunde digitaliseras. Sedan utvecklades en prototyp som utvärderades genom testning och intrevju.

3.2 Testimplementation

För att testa teknikstacker och datamodellen gjordes en enkel och grundläggande implementation av systemet, där endast en adminpanel ingick. Adminpanelen valdes eftersom det är den den delen av systemet där mest funktioner är koncentrerade.

Datamodellen för testimplementationen behölls avsiktlig väldigt enkel och innehöll inte allt som skulle behövas för systemet. Den innehöll endast databasobjekt för användare, evenemang, produkter, produktenheter, media och certifieringar.

Det hade specificerats att implementationen skulle vara en webbapp och utifrån det kravet utvärderades ramverk, tekniker och bibliotek.

Testimplementationen gjordes med Vaadin och hade ingen autentisering. Vaadins funktioner utvärderades för att kontrollera dess lämplighet för systemet. Till en början digitaliserades lagerhanteringen med hantering av produkter och produktenheter, för att testa formulär, anslutning till databas, och filuppladdningar.

3.3 Implementation av prototypen

Efter analys och utvärdering av testimplementationen beslutades det att prototypen skulle vidareutvecklas från testimplementationen. Prototypen är en mer komplett implementation av projektet och är det som blir examensarbetets resultat.

Först slutfördes datamodellen och efter det utvecklades adminpanelen vidare. När adminpanelen nådde en punkt där lagerhantering och kontohantering var fullständigt implementerade (tilläggning och redigering av produkter, produktenheter och konton för personal) började den kundinriktade delen implementeras. Till en början implementerades sidor/vyer för att se alla tillgängliga produkter, och sedan implementerades en bokningssida. Slutligen så implementerades den tredje delen, nämligen panelen för evenemangspersonal.

3.4 Testning och utvärdering av prototypen

Eftersom det inte går att mäta och jämföra den tidigare processen som gjordes utan systemet och den nya processen som utförs med systemet då de är så olika och då verksamheten inte är aktiv just nu, så mättes målens uppfyllelse genom två metoder. Den första metoden var testning och utvärdering av protoypen av en tidigare ansvarig för verksamheten. Detta gjordes vid två tillfällen.

En testning gjordes med testpersonen, som fick testa allt i applikationen. Testpersonen fick testa att använda ett adminkonto, ett evenemangspersonalskonto, och slutligen använda tjänsten utan konto (den kundinriktade delen). Under testförsöket förklarades minimalt med saker som inte ombetts att förklaras, för att kunna observera en naturlig första användning och upptäcka hur enkel eller svårt det var att förstå systemet.

Den andra metoden som användes var att jämföra vilka datapunkter som lagrades. Baslinjen var att samma mängd data skulle lagras, och allt utöver baslinjen räknades som en förbättring.

3.5 Källkritik

För [1]: Detta är webbsidan där specifikationen för JPA beskrivs och är därför den huvudsakliga källan för information.

För [2], [3], [4], [5], [6]: Spring.io är den officiella webbsidan för Spring och hanteras av de ansvariga för Spring. Eftersom en av målgrupperna är potentiella utvecklare kan den möjligtvis vara osaklig angående subjektiv information, men objektiv fakta om Spring kan anses vara pålitligt.

För [7]: Docs.spring.io är den officiella dokumentationen för Spring. Samma resonemang som för Spring.io följer.

För [8]: Vaadin.com är den officiella webbsidan för Vaadin och hanteras av de ansvariga, därför är den en pålitlig källa när det kommer till fakta och tekniska detaljer angående Vaadin. Eftersom det är deras egen produkt kan källan möjligtvis vara partisk när det kommer till subjektiv information, men här används endast objektiv information.

För [9]: Denna bok används som kursbok för kursen Säkerhet (EITF55) på Lunds Tekniska Högskola. Eftersom boken används inom undervisning på universitetet för programmen högskoleingenjör i datateknik och högskoleingenjör i elektroteknik kan den anses vara pålitlig.

För [10]: Detta är den nya standarden som byter ut den gamla RFC 4122 från 2005. Eftersom det är denna standard (och tidigare RFC 4122) som beskriver UUIDv4 är den den auktoritativa källan för ämnet och kan litas på fullständigt.

För [11]: Artikeln är en trovärdig källa eftersom den skrevs av en framstående matematiker och lade grunden för informationsteorin. Fastän artikeln är gammal används fortfarande dess teori och terminologi i modern kryptografi och informationssäkerhet.

4 Analys

Detta kapitel beskriver analysen som gjordes genom examensarbetet.

4.1 Tidigare process

Verksamheten är just nu inaktiv så därför beskrivs detta i preteritum, men denna process var den som användes sist.

Processen var helt manuell och gjordes utan några system. Bokning gjordes manuellt via e-post, personalhantering via en extern tjänst, tidrapportering med hjälp av papper och penna, försäljning på plats med manuella kvitton. Utrustning reviderades manuellt en gång om året, och det var vid det tillfället som besiktning och kontroll av skador och skick gjordes. Lagersaldon hanterades med ett kalkylark, där varje artikelnummer endast hade ett antal (inga enskilda identiteter för artiklarna).

Ett exempel från början till slut hade sett ut som följande, notera att det kunde skilja sig eftersom olika typer av evenemang hanteras på olika vis, och kunder kunde själv välja om de ville hämta produkterna eller om de skulle bli levererade och hanterade.

1. Kunden bokar ett evenemang via e-post
2. Företaget skriver in detta i en kalender
3. Kunden får en bokningsbekräftelse som skrivs för hand och betalas via bankgiro
4. När tiden kommer så väljs godtyckliga produktenheter för de önskade produkterna, och de transporteras till adressen
5. Evenemangspersonalen sätter upp utrustningen
6. Vid slutet på evenemanget transporteras produktenheterna tillbaka till lagret

För öppna/offentliga evenemang som sätts upp av företaget och som är ämnade för allmänheten att delta i, så togs betalning (kontanter) på plats och kvittona gjordes manuellt.

Eftersom bokningen gjordes via e-post ledde det till att flera e-posttrådar behöver hållas koll på. När ett evenemang skulle sättas upp så registrerades det inte vilka produktenheter som hade använts, och eftersom detta inte heller speglades i lagersaldot så var det otydligt om hur många produktenheter som var tillgängliga vid en tidpunkt (lagersaldot visade endast hur många produkter som fanns i företagets ägo).

Ifall en produkt slets eller skadades på ett annat vis fanns det inget standardiserat och systematiskt sätt att dokumentera det på, utan de ansvariga evenemangspersonalen gjorde på det sätt de tyckte passade bäst (t.ex. skicka ett sms). Men även då, eftersom produktenheterna inte hade en identitet, var det problematiskt att hålla koll på vilken produktenhet det handlade om.

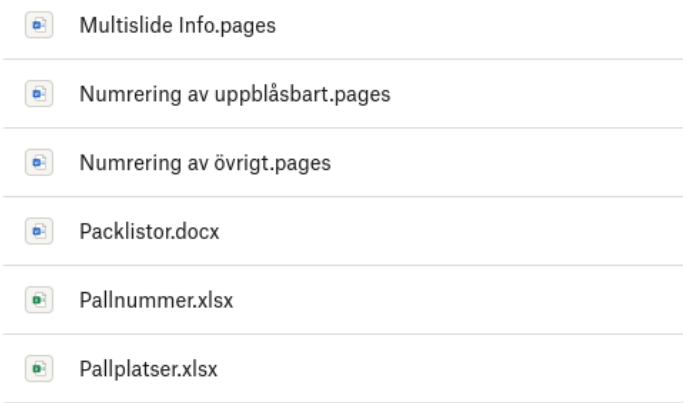
Inventeringslista

	Åstorp
Pooler	
5x8 Pool	2
8x8 Pool	1
11x8 Pool	1
11d pool (rund)	1
Plattform	
Plattformskit	
Waterz	
- *hela*	
- *söndriga*	

Figur 1. Ett utdrag från en gammal inventeringslista

Eftersom kalkylarken för inventeringslistorna (se Figur 1) inte specificerade vad som vid det tillfället var utanför lagret (på ett evenemang) skedde det inga systematiska kontroller om att allt var tillbaka på lagret efter evenemangen. Figuren visar också att skador inte dokumenterades väl, utan de dokumenterades endast som ett tal för produkter i god skick och ett fal för produkter i dåligt skick.

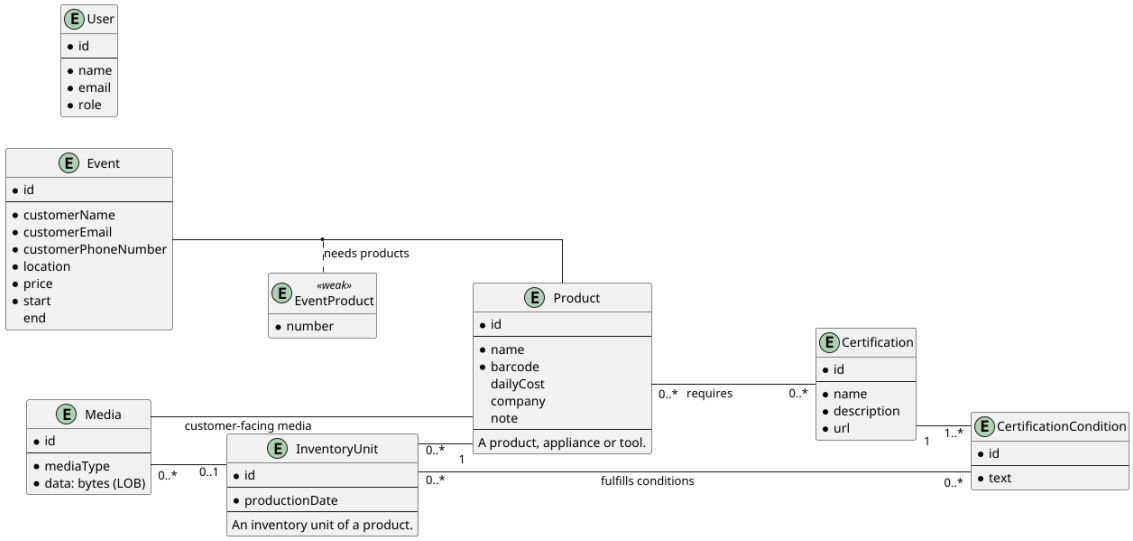
Enligt en tidigare ansvarig på verksamheten är det även så att det just nu är otydligt om vad som finns på lager (p.g.a. den bristfälliga lagerhanteringen) och därför kommer en fullständig inventering att ske när det nya systemet är på plats.



Figur 2. Ett antal filer från Dropbox-molnlagringen.

Dokumenten hanterades via Dropbox och molnutrymmet delades med ansvarig personal. Se Figur 2 för ett utdrag från en mapp.

4.2 Initial datamodellering



Figur 3. Initial datamodell.

För testimplementationen skapades en initial datamodell (se Figur 3).

I denna modell representerar *Product* en produktmodell, t.ex. "Blå Studsmatta", medan *InventoryUnit* representerar en enskild enhet av en produkt. *Media* används för både bilder och videor, och datan lagras i databasen som en Large Object-kolumn. I diagrammet så är kolumner som måste ha ett värde markerade med en rund ifylld cirkel, medan kolumner som kan vara null inte har något till vänster om sig.

En viktig del av datamodellen är att produktenheter har sin egen identitet. Detta för att kunna spåra var en produktenhet har använts, när den har producerats, och kunna knyta anmälningar om skador till en specifik enhet.

En certifiering (*Certification*) är en standard (t.ex. ISO-standard) som produkter måste uppfylla för att anses vara säker. Detta görs genom att ett antal krav/villkor (*CertificationCondition*) definieras för certifieringen, sedan definieras vilka certifieringar som en produkt måste uppfylla. Slutligen så markeras de olika krav/villkoren som uppfyllda för produktenheterna och varje villkor för en certifiering måste uppfyllas för att certifieringen ska räknas som uppfylld.

Istället för att vid behov beräkna den totala kostnaden via produkterna länkade till ett *Event*-objekt så lagras den totala kostnaden i objektet. Detta för att ändringar på produkternas pris inte ska påverka historiken.

Denna grundläggande modell användes för att implementera en testversion av projektet med hjälp av teknikstacken, för att utvärdera stackens lämplighet innan vidareutveckling.

4.3 Utvärdering av lämplig teknikstack

Eftersom mycket inom evenemangshanteringsprocessen handlar om data, prioriterades ramverk som möjliggör enkel interaktion och kommunikation mellan back-end och front-end. Detta för att undvika att ha två beskrivningar av varje objekts struktur på vardera sida, och för att undvika att manuellt serialisera och deserialisera objekt när de skickas från servern till klienten eller tvärtom. Utöver det kan det undvikas att behöva utforma en API, eftersom ramverket hanterar kommunikationen. En naturlig konsekvens av sådana ramverk är att oftast så används samma programmeringsspråk genom hela applikationen, vilket var gynnsamt eftersom ett av önskemålen var att endast använda ett programmeringsspråk.

Det var inte ett mål att jämföra olika teknikstacker, utan målet var endast att hitta en teknikstack som lämpar sig för implementering av projektet.

Följande teknikstack valdes:

- **Programmeringsspråk:** Java
- **Ramverk:** Spring Boot (4.0.6) & Vaadin Flow (25.0.7)
- **Databashantering:** Spring Data JPA
- **Säkerhet:** Spring Security

För implementationen utvärderades ramverket Vaadin. Med Vaadin Flow medföljer kravet att klienten alltid är online, men eftersom alla delar av systemet ändå kräver kommunikation med servern så finns det inget som skulle fungera offline oavsett, därför påverkas inte användarupplevelsen förutom att på väldigt långsamma anslutningar kan det dröja lite lång tid att utföra vissa handlingar.

Spring Boot utvärderades inte, utan den valdes eftersom Vaadins "komma igång"-guide använder den. Vaadin och Spring Boot används i detta fall som ett paket och de används tillsammans.

Enligt avgränsning 1.6.3 och eftersom Spring Data JPA kan användas på samma sätt för databashanterare som erbjuder JDBC-stöd utvärderades inte en databashanterare. Under utvecklingsprocessen användes en godtyckligt vald databashanterare och för att förenkla processen valdes en inbäddad sådan, nämligen H2.

För att testa teknikstacken så gjordes en testimplementation av en liten del av systemet.

Testimplementationen lyckades och teknikstacken bedömdes som lämplig eftersom ramverkets modell passade projektets ändamål. Att hantera databasen var enkelt och det var lätt att via kod i servern visa information till och ta emot information inmatad från användaren från klienten (webbläsaren).

På grund av det testades inga fler ramverk eller ändringar på teknikstacken, eftersom den uppfyllde kraven. Därefter användes testimplementationen som en grund för prototypen, prototypen är en vidareutveckling av samma kodbas.

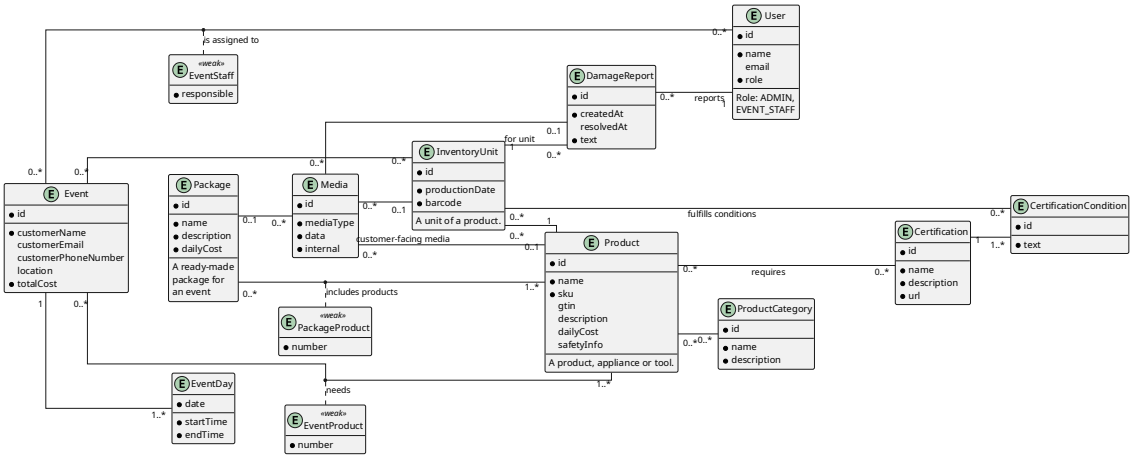
4.4 Slutlig datamodell

Efter testningen slutfördes datamodellen. Detta gjordes genom att informationen från förstudien (intervjuerna och frågorna) kontrollerades mot den initala datamodellen, och följande krav noterades:

- Skaderapporter för produktenheter.
- Ett objekt för att representera färdiga paket av aktiviteter.
- Åtkomstkontroll för media, för att skydda interna bilder från obehöriga.
- Associering av evenemangspersonal till evenemang.
- Associering av produktenheter till evenemang.
- Möjlighet för tiderna att variera för evenemang som sträcker sig över flera dagar.

Genom analys av etiska aspekter noterades ett ytterligare krav:

- Möjlighet att radera personliga uppgifter utan att radera objekten *Event* & *User*.



Figur 4. Slutlig datamodell. Figuren finns även som bilaga i större storlek, se Bilaga 1.

Ur dessa krav utvecklades en ny datamodell (se Figur 4), som är den som använts i prototypens implementation.

För att olika evenemangspersonal skulle kunna ha olika grad av ansvar för ett evenemang introducerades en tabell som var en svag entitet och som länkade personal och evenemang. Denna tabell innehöll ett attribut som specificerade om den anställde var ansvarig i detta evenemang eller inte.

För att kunna radera kundernas och personalens personliga uppgifter efter att de inte längre behövdes så togs kravet på att ett värde ska finnas bort från flera kolumner. Kravet togs bort från evenemangobjektets fält för kundens e-postadress och telefonnummer, och från fältet för e-postadressen på konton för personalen. Detta för att

kunna endast radera attributens värden när de inte längre behövs (och inte behöva radera hela objektet), utan att påverka spårbarheten och historiken.

En begränsning för relationerna för *Media* är att endast den endast kan användas i en relation åt gången. Om ett *Media*-objekt används för en produkt kan den inte användas för ett färdigt paket (*Package*).

4.5 Risker med digitalisering

En naturlig egenskap hos analoga system, som fungerar både som ett skyddslager och som en begränsning, är att de endast kan kommas åt via fysisk åtkomst.

Digitala system behöver skyddas mot obehöriga och angripare. Detta kan göras genom att kräva autentisering för åtkomst till intern data.

4.6 Autentisering

Det hade specificerats att endast e-postadresser skulle användas för inloggning. Vid inmatning av e-postadressen skulle ett e-post skickas till adressen med en kod eller länk för att komma in på kontot.

Av de inbyggda autentiseringsmetoderna i Spring Security testades initialt OTT-baserad inloggning. OTT-stödet är implementerat på så vis att en unik kod (token) genereras för att kunna logga in. Denna kod kan både användas i en särskild URL för automatisk inloggning, men den kan också matas in manuellt eller via inklistring.

Spring Securitys inbyggda OTT-stöd genererar en UUID (version 4 – UUIDv4) som inloggningskod, vilket är ett 128-bitars värde som representeras i hexadecimal form med 4 bindestreck [10]. Denna sträng består av totalt 36 tecken och är därför inte praktisk att matas in manuellt, vilket försvårar för användaren att t.ex. kan logga in på en dator, när denne har sin e-postinkorg på sin mobiltelefon.

Initialt försöktes kodgenereringen anpassas, men eftersom inget specifikt stöd fanns för det behövdes hela klassen som hanterar koderna kopieras och modifieras. Då upptäcktes det även att koden inte är knuten till användarens session. Istället implementerades en ny autentiseringsmetod för Spring Security.

För att utvärdera säkerheten av slumpmässigt genererade koder finns det bland andra, två metoder att använda. Den första metoden är att beräkna det totala antalet möjliga koder som kan genereras, men eftersom dessa tal kan bli väldigt stora är det inte lätt att jämföra dem och dra slutsatser. Därför kan istället entropi användas.

Det önskade formatet på koden bestod av 6 st bokstäver ur mängden M = {B, C, D, F, G, H, J, K, L, M, N, P, Q, R, S, T, V, W, X, Z}, vilket är ISO latinska alfabetet förutom de svenska vokalerna (för att minska risken att ord uppstår).

För att beräkna entropin för det önskade formatet kan Claude Shannons teori användas.

Denna formel kan användas:
$$H(X) = -\sum_{i} p_i \log_2(p_i)$$

Där $H(X)$ är entropin i bitar och p_i är sannolikheten för utfallet x_i [11]. Mängden M innehåller 20 element. Eftersom varje bokstav är slumpmässigt utvald så är sannolikheten likadan, vilket innebär att:
$$p_i = \frac{1}{|M|} = \frac{1}{20} \qquad i = 1, 2, 3, \ldots, 20$$

Entropin för en bokstav blir:
$$H_1 = -\sum_{i=1}^{20} \frac{1}{20} \log_2\left(\frac{1}{20}\right) \approx 4{,}32193$$

Entropin för en sträng med 6 bokstäver blir:
$$H_{kod} = \sum_{i=1}^{6} H_1 = 6 * H_1 \approx 26$$

För att fastställa entropin av en UUIDv4 behöver bitarna i den förtydligas, då inte alla bitar är slumpmässiga. Vissa visar versionen och varianten på UUID-värdet. Antalet slumpmässiga bitar i en UUIDv4 är 122 och sannolikheten för varje värde är lika stor, därmed är entropin för UUIDv4 lika med 122 bitar [10].

Resultatet är att entropin för det nya kodformatet är ungefär 26, medan entropin för UUIDv4 är 122. Detta är ett mycket mindre värde, men det nya kodformatet ger ändå 64 miljoner möjliga kombinationer ($20^6 = 64000000$).

För att ändå inte minska säkerheten knöts inloggningskoderna i den nya autentiseringsmetoden till sessionen som påbörjade inloggningsprocessen. På så sätt kan endast webbläsaren som påbörjade inloggningsförsöket mata in koden för att logga in. Utöver det gjordes det så att efter 5 misslyckade försök så ogiltigförklarades koderna.

4.7 Åtkomstkontroll

För åtkomstkontroll valdes en kombination av RBAC och ABAC. Två roller definierades: *administratör* och *evenemangspersonal*. Besökare utan konton (kunder) räknas som utan roll, och med en alternativ benämning, anonyma.

Alla besökare har åtkomst till den offentliga, kundinriktade sektionen. Endast administratörer har åtkomst till adminpanelen, och endast evenemangspersonal har åtkomst till panelen för evenemangspersonal. Inom panele för evenemangspersonalen används vidare attributbaserad åtkomstkontroll (ABAC) för att endast tillåta evenemangspersonalen öppna och redigera evenemang som de är tillsatta.

För media (bilder, videor och dokument) krävdes mer utvärdering, eftersom viss media endast är privat och används internt medan annan media är offentlig och används både internt (panelerna) och externt (den kundinriktade delen). Två alternativ utvärderades,

det ena var att ingen autentisering skulle krävas, men att medieobjekten skulle identifieras med en UUID för att endast kunna hittas ifall UUID:n redan kändes till. Detta alternativ användes inte eftersom en missuppfattning om medians privathet kunde leda till att en priviligierad användare offentligt delade en länk med förväntningen att endast privilegierade användare skulle kunna komma åt den.

Istället användes attributbaserad åtkomstkontroll på så sätt att varje medieobjekt innehåller ett binärt attribut som specificerar om det är privat. Beroende på var medieobjektet skapas får den sant värde på attributet automatiskt, t.ex. om den skapas genom uppladdning av en bild för en produkt så blir värdet falskt (offentligt), men om den skapas genom uppladdning av en bild för en enskild produktenhet eller från en rapportering om en skada på en produktenhet så blir värdet sant (privat). Eftersom medieobjekt endast kan användas av exakt ett annat objekt och inte kan flyttas mellan olika objekt så är detta en lämplig lösning.

Media för enskilda produktenheter är inte ämnade för kunder att ses och är endast intern data. Produktenheternas media kan också vara saker som inte är bilder på enheten, t.ex. kvitto eller liknande. Media för produkterna är det som visas för kunderna. Eftersom privat media skyddas mot obehöriga identifieras media med ett sekventiellt ID, om de inte skyddades så hade sekventiella ID:n varit en säkerhetsbrist då obehöriga hade kunnat upptäcka intern media genom att ändra på talet (t.ex. bläddra genom alla ID:n från 0 till 1000).

5 Resultat

5.1 Prototypen

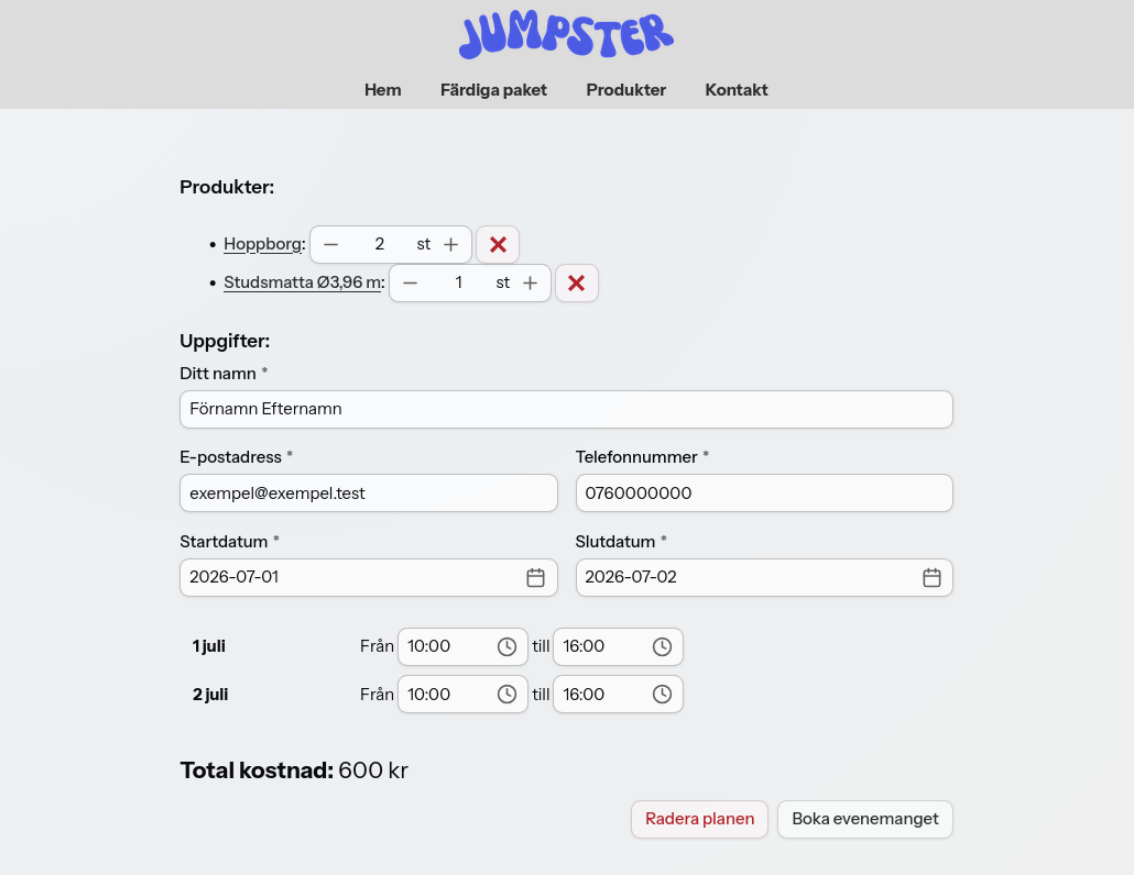
Prototypen är en implementation av projektet. Den färdiga prototypen inkluderar alla tre planerade sektioner, nämligen den kundinriktade offentliga sektionen, panelen för evenemangspersonalen, och adminpanelen.

5.2 Den nya processen

Den nya processen är som följande med prototypen, i detta exempel är det ett evenemang som kunden vill ha uppsatt och hanterat av personalen:

1. Kunden bokar via webbsidan och får en bokningsbekräftelse till sin e-postinkorg (Betalning hanteras inte i prototypen, men det kommer att vara så att kunden betalar på webbsidan innan bokningen slutförs)
2. Bokningen lagras i databasen och kan ses i en kalender på adminpanelen.
3. Administratör öppnar evenemanget och tilldelar evenemangspersonal till det (både ansvariga och de som inte är det).
4. Evenemangspersonalen som tilldelats evenemanget får upp det i sin kalender (på panelen för evenemangspersonalen).
5. Ansvarig evenemangspersonal får upp en lista på produkter som behövs för evenemanget och kort innan evenemanget så skannar de streckkoderna för produktenheterna som de valt att använda.
6. När korrekt antal produktenheter skannats visas detta på båda panelerna.
7. Personalen transporterar och sätter upp produkterna.
8. Om en produkt slits eller skadas på något sätt kan ansvarig evenemangspersonal skicka in en anmälan om skada på produkten.
9. Denna rapport visas tydligt under produktenheten på adminpanelen och kan när lämpligt markeras som löst av administratörer.
10. När evenemanget är slut finns det en tydlig lista på vilka produktenheter som hade använts, och personalen kollar med listan så att allt är tillbaka på plats.

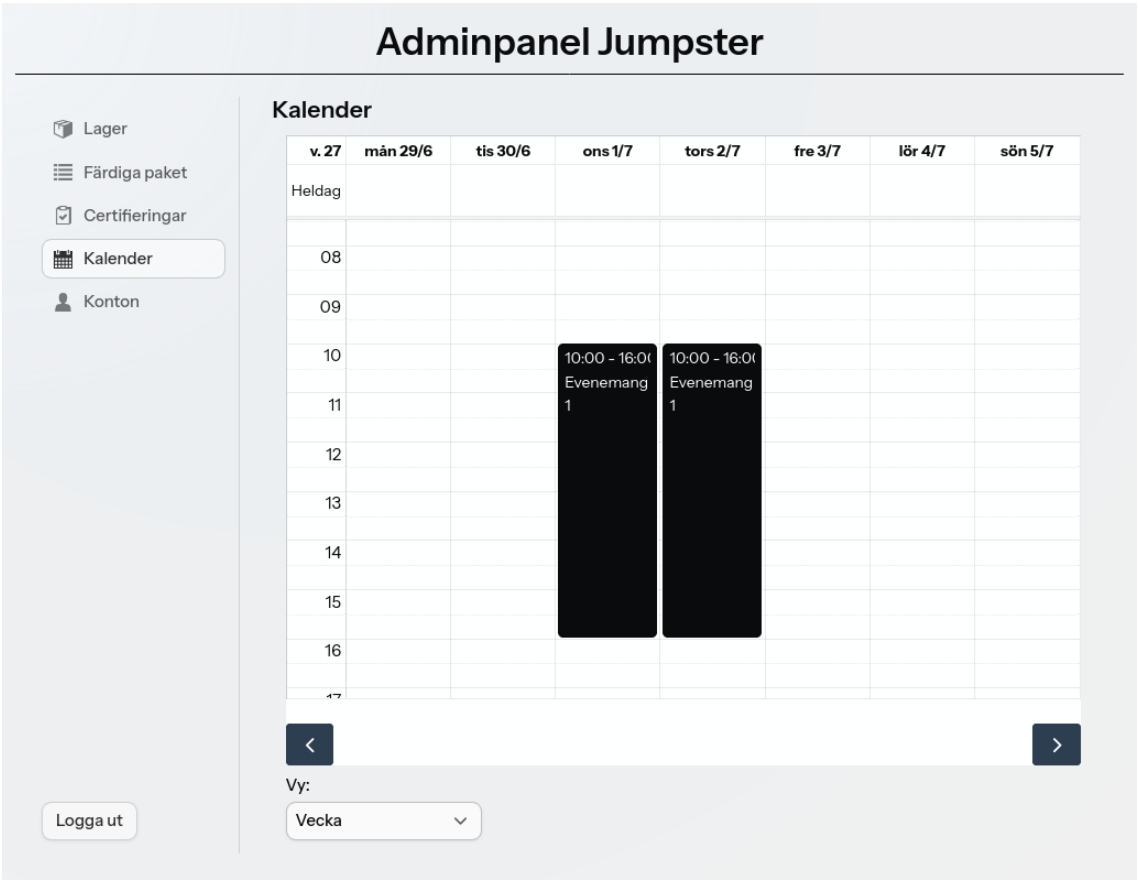
Skärmdumpar från processen:



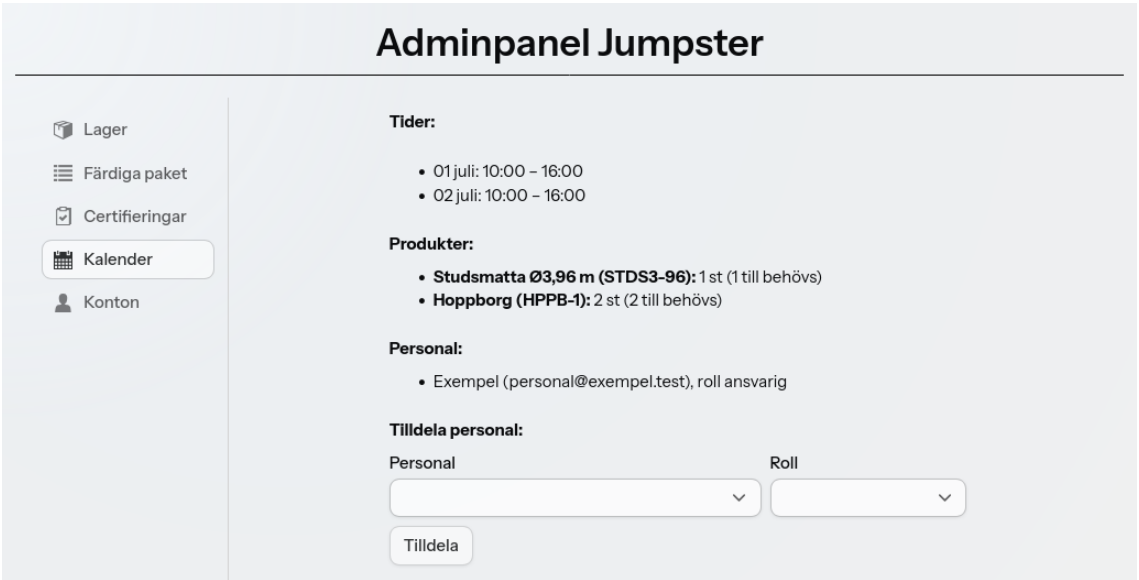
Figur 5. Bokningsformulär

För att boka ett evenemang så använder kunden ett formulär (se Figur 5). Från början är formuläret inte tillgängligt, utan knappen för att öppna formuläret kommer upp efter att kunden valt ett färdigt paket eller minst en produkt. Formulärets information lagras i sessionen, så om kunden stänger fliken eller laddar om sidan förloras inte datan.

För att tillåta evenemanget att ha olika tider på olika dagar presenteras en lista på alla dagar mellan start- och slutdatumet och kunden kan därifrån välja start- och sluttiden för varje dag.



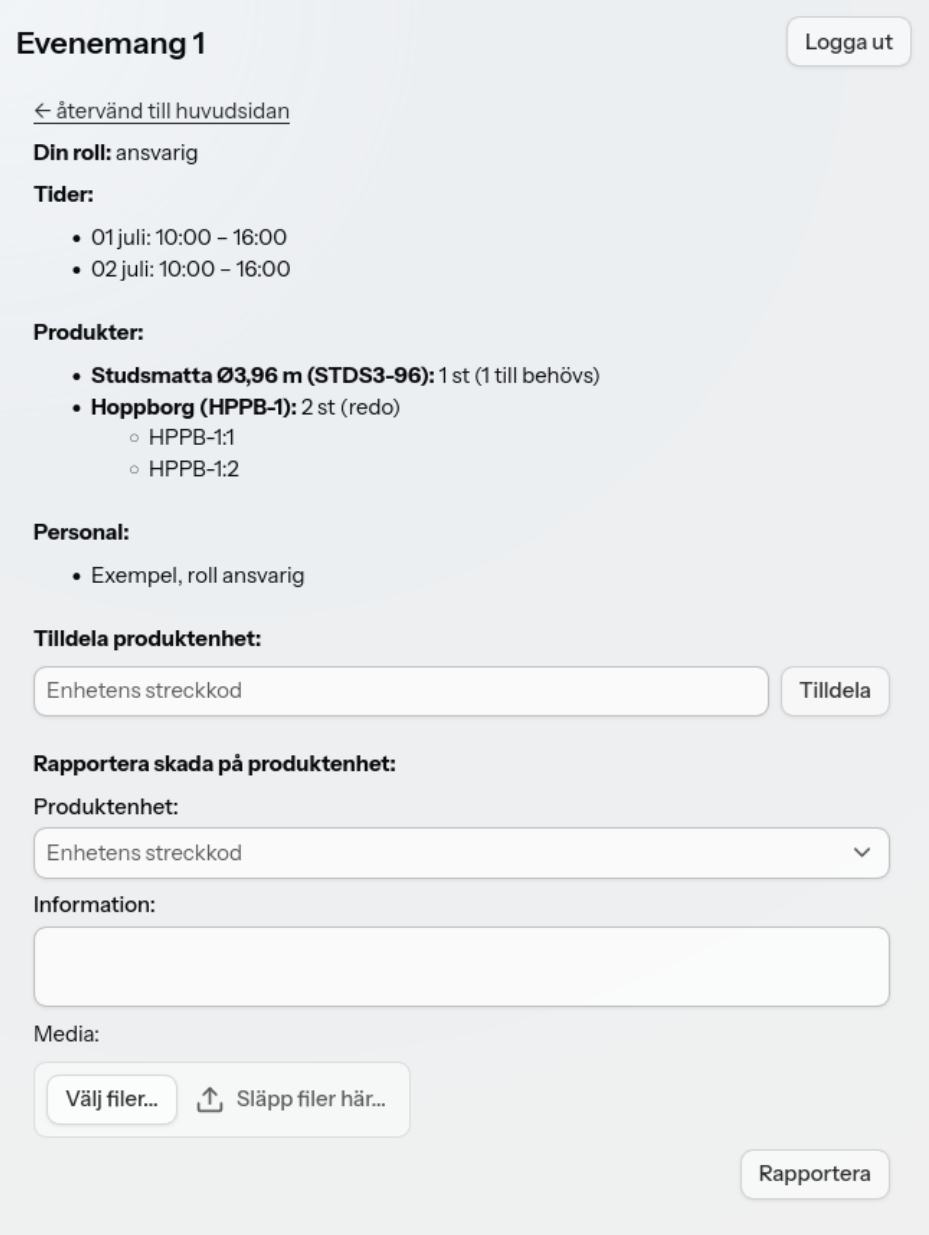
Figur 6. Adminpanelens kalender



Figur 7. Adminpanelens vy för evenemang

Från adminpanelens kalender (se Figur 6) kan administratörer komma åt evenemang och hantera dem. Sedan tilldelar administratören personal via vyn för ett evenemang (se Figur 7).

Huvudsidan för panelen för evenemangspersonalen är en likadan kalender som administratörernas (se Figur 6), med skillnaden att endast evenemang som de är tilldelade visas.



Figur 8. Evenemangspersonalens vy för ett evenemang

Ansvarig evenemangspersonal tilldelar produktenheter till evenemang genom att skanna eller mata in deras streckkoder (se Figur 8). Ansvarig evenemangspersonal kan också rapportera skador på produktenheter och bifoga bilder och videor och skriva en kommentar. Rapporteringen skickas till administratörerna och ses i adminpanelen på sidan för produktenheten.

5.3 Feedback från utvärdering av prototypen

För att testa målets uppfyllelse hade en testning och utvärdering av prototypen planerats. Personen som testade och utvärderade prototypen var en tidigare ansvarig för verksamheten.

Enligt personen som testade och utvärderade prototypen är detta ett optimerat system som samlar ihop flera rutiner till ett och samma samma system. Tidigare var det olika system och separata rutiner som användes, till exempel så fanns det separata filer för olika produkter.

Det fanns ingen koppling mellan bokning & leverans och webbsidan, utan allt gjordes via e-postkontakt. Detta är nu integrerat och bokningen görs direkt på webbsidan och produkter reserveras direkt vilket förhindrar att överlappande evenemang bokas som hade behövt produkter som inte är tillgängliga. Därmed är detta också en optimering av flödet.

Enligt personen så är hela processen samlad nu och det finns en slags versionshantering där det syns vilka produktenheter som använts i vilket evenemang och när, och att på så sätt minskar risken att produktenheter tappas bort.

5.4 Nya datapunkter

Dessa nya datapunkter lagras i systemet och som inte lagrades i den tidigare processen:

- En identitet för produktenheter. Det är en streckkod som skrivs ut och klistras på produktenheten för att identifiera den.
- Rapporteringar för skador och slitage på produkter.
- Lagersaldon i realtid som visar ifall en produktenhet är reserverad och ifall den används just nu.
- Bilder och media för enskilda produktenheter.

5.5 Diskussion

Designen av datamodellen gjordes med hjälp av intervjumaterialet och svaren på de inskickade frågorna. Detta gjordes genom att intervjuerna och frågorna gjordes utkast för datamodellen gjordes och sedan ställdes förtydligande frågor för att justera datamodellen. Detta var en välfungerande process där intervjupersonen inte behövde förstå eller tänka på datamodellen, utan endast tänka på vad som hänger ihop med vad och vilka attribut och egenskaper som behöver lagras.

För att undersöka målets uppfyllelse gjordes testning av systemet, men dessa tester användes inte för att bygga gränssnittet. Hela gränssnittet utformades på egen hand förutom den kundinriktade delen som också designades med hjälp av en anställd på företaget (som delade bilder av en grundläggande design). För implementationen av gränssnittet användes intervjuerna men endast för att bestämma vad gränssnittet skulle innehålla, inte hur det skulle se ut.

Eftersom en fullständig specifikation för projektet inte hade gjorts behövdes all information inhämtas via intervju, frågor och analys av gamla dokument. Detta ledde till att många frågor behövdes ställas och flera av dem fick vara generella, men vissa oklarheter fanns som tog lång tid att klaras upp och en liten del av systemet som velats ha hade inte kommunicerats. Detta ledde i slutändan till att denna lilla del inte kunde vara med i systemet eftersom det kom väldigt sent i processen, vilket skedde på grund av att kommunikationen med företaget var bristande då de inte hade mycket tid. Denna lilla del (offentliga evenemang) beskrivs i framtida utvecklingsmöjligheter men räknas inte med som en del av examensarbetet eftersom det inte kommunicerades i tid, vilket även företaget håller med om.

Intervjumaterialet som visade sig vara mest viktigt var en inspelad intervju där en fråga ställdes om att berätta den tidigare processen från start till slut (från kundens bokning ända till slutet där evenemanget avslutas och följs upp). Frågor som kräver utförlig beskrivning av processen var de som var mest viktiga eftersom intervjupersonen då behövde tänka igenom allting och berätta saker som intervjuaren möjligtvis inte alls kände till.

All intervju och testning gjordes med samma person, en tidigare ansvarig för Jumpsters föregångare. Verksamheten kommer startas upp på nytt och det är också nya anställda och ansvariga, därför fanns det inte möjlighet att till exempel testa med evenemangspersonal då det inte fanns några tillgängliga. Att samma person testade alla tre delar kan tänkas vara problematiskt eftersom det under vanlig användning kommer vara tre olika användargrupper som kommer att använda endast en del av de tre delarna var. Eftersom testpersonen hade mycket koll över verksamheten dömdes det som lämpligt att denne skulle testa allt, men personen var särskilt lämplig för de interna delarna just eftersom denne var en tidigare ansvarig. Därför kunde värdet av digitaliseringen ha testats ytterligare med en tidigare kund som hade använt det gamla sättet att boka (via e-post) jämfört med det nya sättet. Verksamheten hade varit inaktiv under en period och därför fanns det inga kunder som inom närtid bokat evenemang, vilket var anledningen till att det inte gjordes en sådan testning. Värdet av en sådan ytterligare testning är att det kunde ha resulterat i möjliga förenklingar och förbättringar eftersom det skulle varit ett perspektiv från en användargrupp som inte har något att göra med de interna delarna.

6 Slutsats

Detta kapitel beskriver det viktigaste i resultatet och besvarar de ursprungliga frågorna. Utöver det analyseras etiska aspekter och framtida utvecklingsmöjligheter.

6.1 Sammanfattning

Verksamheten drevs tidigare genom separata manuella eller analoga rutiner. E-post, dokument, papper och kalkylark användes och det saknades ett systematiskt arbetssätt. Det fanns ingen spårbarhet, administreringen var ineffektiv och det var svårt att få en korrekt överblick över verksamheten.

I detta examensarbete undersöktes ett tidigare arbetssätt och en plan för digitalisering formulerades och genomfördes. Genom att undersöka arbetssättet modellerades en databas och ett digitalt system utformades för hela processen och de tre användargrupperna.

Den tidigare processen digitaliserades med framgång. Det nya systemet uppfyller alla ursprungliga mål i målformuleringen och resultatet visar att syftet uppnåtts genom utvecklingen av en fungerande prototyp som samlar kunder, administratörer och evenemangspersonal i ett gemensamt system som inkluderar:

- En kundinriktad del för bokning och utforskning av produkter och paket
- En panel för evenemangspersonal för att sätta upp och hantera evenemang och rapportera skador på produkter
- En panel för administratörer för att hantera lager, evenemang, konton, kategorier och certifieringar

Resultatet bidrar till syftets uppfyllelse genom att verksamheten nu har en modell och prototyp för ett systematiskt digitalt arbetssätt. Systemet ger bättre kontroll över lager och utrustning, integrerar spårbarhet mellan evenemang och produktenheter och ger bättre förutsättningar för uppföljning av evenemang.

Resultatet kommer att användas som grund för verksamhetens framtida system och prototypen visar att de identifierade rutinerna kan digitaliseras och fungera samtidigt.

6.2 Svar på problemformuleringen

Den övergripande frågan i problemformuleringen var: "Hur kan ett enhetligt digitalt system byggas för separata manuella eller analoga rutiner och processer?"

Ett enhetligt digitalt system kan byggas genom en utförlig analys av den befintliga processen för att sedan utvärdera en lämplig teknikstack. För att utvärdera teknikstacken kan en mindre testimplementation göras som ifall den lyckas kan bli grunden till ett slutligt system. Det slutliga systemet byggs genom kartläggning av befintliga

arbetsflöden, identifiering av relationer mellan rutinerna som följs av modellering av en gemensam datamodell.

En digitaliserad process skiljer sig från den tidigare processen genom att allt är systematiskt och spårbart. Alla tre användargrupper använder sig av samma databas och historiken för evenemang sparas utan att kompromissa på personlig integritet.

Med digitalisering kommer dock nya säkerhetskrav eftersom datan inte längre skyddas av fysisk åtkomst, och detsamma gäller för överföring av hanteringen från molnlagring till det nya systemet då den datan inte längre skyddas av molntjänsten och nu måste skyddas av systemet.

6.3 Etiska aspekter

I systemet ingår lagring av kund- och personaluppgifter. Kunder som bokar evenemang anger sitt namn, e-postadress, telefonnummer och om önskat adress (plats). Efter ett evenemangs avslut används inte längre personuppgifterna, men eftersom de kan behövas för uppföljning, klagomål eller anmälningar så behöver de lagras en tid efter slutet också.

För konton (evenemangspersonalens och administratörernas) lagras namn och e-postadress. När konton är aktiverade behövs denna information för att tillhandahålla tjänsten. Konton raderas inte från systemet för att behålla en spårbar historik, vilket innebär att konton kvarblir i systemet även när anställningen avslutats. Istället för att raderas så avaktiveras konton, och då kan de inte längre användas.

För att kundernas e-postadress och telefonnummer, samt före detta personals e-postadress inte ska kvarbli i systemet efter att de inte längre behövs så raderas de automatiskt 30 dagar efter evenemanget för kunder, och för före detta personal så raderas deras e-postadress efter att deras kontot varit avaktiverat i 30 dagar. Antalen dagar är konfigurerbara. Detta är en form av anonymisering, dock är det inte fullständig anonymisering eftersom namn kvarstår.

6.4 Framtida utvecklingsmöjligheter

För att möjliggöra extern integration med systemet kan en HTTP API implementeras. På så sätt kan externa verktyg och plattformar anropa metoder för att hämta och ändra data som lagras i systemet utan att behöva lagra egen data, alternativt så kan de lagra egen data externt (data som systemet inte hanterar) och länka det till objekten i systemets databas. Användningsområden för ett sådant API kan bland annat vara att automatisera beräkningen av lön för personal och förenkla bokföring.

För att stöda offentliga evenemang som är öppna för allmänheten (detta var inte en del av examensarbetet) kan en ny typ av evenemang implementeras. Då kan administratörer skapa sådana evenemang som allmänheten via webbsidan kan betala för och få en biljett som även lagras i databasen. Genom att jämföra kostnaden att sätta upp evenemanget mot intäkterna från billjettförsäljning kan lönsamheten automatiskt beräknas.

7 Källförteckning

[1] “Jakarta Persistence Specification Project.” Accessed: Jun. 02, 2026. [Online]. Available: https://jakartaee.github.io/persistence/

[2] Spring, “Spring Framework.” Accessed: Jun. 02, 2026. [Online]. Available: https://spring.io/projects/spring-framework

[3] Spring, “Spring Boot.” Accessed: May 21, 2026. [Online]. Available: https://spring.io/projects/spring-boot

[4] Spring, “Spring Data.” Accessed: May 21, 2026. [Online]. Available: https://spring.io/projects/spring-data

[5] Spring, “Spring Data JPA.” Accessed: Jun. 02, 2026. [Online]. Available: https://spring.io/projects/spring-data-jpa

[6] Spring, “Spring Security.” Accessed: May 18, 2026. [Online]. Available: https://spring.io/projects/spring-security

[7] Spring, “Authentication.” Accessed: May 18, 2026. [Online]. Available: https://docs.spring.io/spring-security/reference/servlet/authentication/index.html

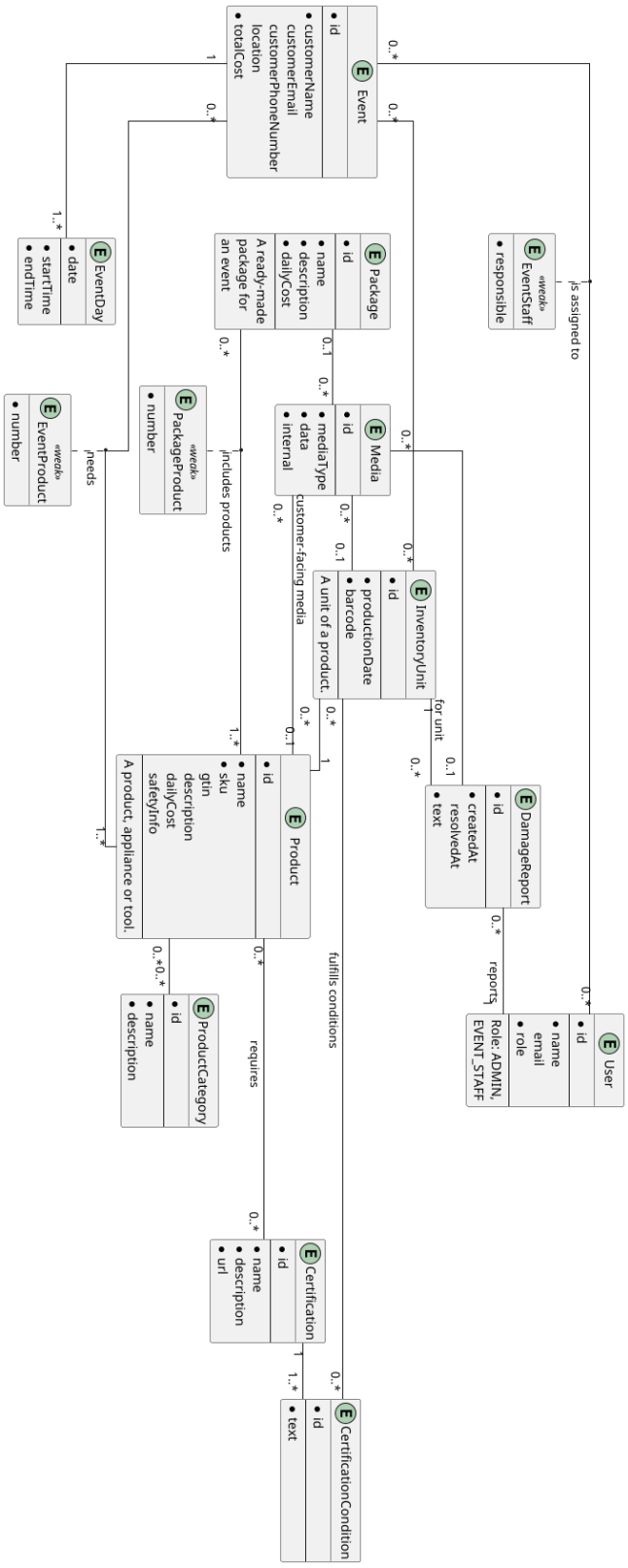
[8] Vaadin, “What is Flow?” Accessed: May 18, 2026. [Online]. Available: https://vaadin.com/docs/latest/flow/what-is-flow

[9] W. Stallings and L. Brown, *Computer Security: Principles and Practice*, Fifth Edition. Pearson, 2024.

[10] K. R. Davis, B. Peabody, and P. Leach, “Universally Unique IDentifiers (UUIDs),” May 2024, *RFC Editor*: 9562. doi: 10.17487/RFC9562.

[11] C. E. Shannon, “A Mathematical Theory of Communication,” *The Bell System Technical Journal*, 1948, [Online]. Available: https://people.math.harvard.edu/%7Ectm/home/text/others/shannon/entropy/entropy.pdf

8 Bilagor



Bilaga 1. Figur 4 (Slutlig datamodell) i större storlek.