

# DICOMplicate

Daga Indebetou (BME23), Virginia Piehl (BME23)

**Sammanfattning**—Breast cancer is the most prevalent form of cancer among women in Sweden, with 9 000 diagnoses during 2023. While the five-year survival rate reaches 92.5% due to national screening programs, transitioning from 2D Digital Mammography to 3D Digital Breast Tomosynthesis (DBT) could further improve early detection. However, the high volumes of data increases the examination times, preventing it from becoming the clinical standard. While AI models offer a potential to speed up the process, their development is limited by both the need of large datasets to ensure unbiased and accurate models, and the need of pseudonymization to follow patient integrity laws.

The M-BIG database, containing 450 000 examinations, could be a significant resource but lacks a scalable tool for data pseudonymization. This project addresses this need by developing DICOMplicate, a graphical user interface built with Streamlit. The tool allows the user to perform SQL queries and export DICOM files where all patient-identifying metadata is automatically removed. By ensuring safe pseudonymization while enabling easy data extraction, DICOMplicate makes it possible for more researchers to access the data. Performance evaluations show that the tool can reliably export 5 000 images in 30 minutes, demonstrating its potential to become a valuable tool in breast cancer research.

## I. INTRODUKTION

### A. Bakgrundsteckning

**B**RÖSTCANCER är den vanligaste cancerformen bland kvinnor och drabbar i genomsnitt var tionde kvinna under sin livstid. Under 2023 diagnostiserades över 9 000 nya fall, där orsakerna främst kan härledas till en kombination av genetiska, hormonella och livsstilsrelaterade faktorer. Tack vare etablerade screeningprogram och framsteg inom medicinsk forskning kan tumörer idag detekteras i ett tidigt skede, vilket har bidragit till en femårsöverlevnad på 92,5% [1]. Enligt beslut från Socialstyrelsen erbjuds kvinnor i åldersspannet 40 till 74 år avgiftsfri screening, i syfte att sänka mortaliteten genom tidig detektion [2].

Standardmetoden vid bröstcancerscreening i Sverige är digital mammografi (DM), vilket genererar tvådimensionella projektioner av bröstet från två vinklar, kraniokaudal (CC) och mediolateral snedprojektion (MLO). En begränsning med DM är den så kallade maskeringseffekten, där tät bröstvävnad kan täcka och dölja tumörer, vilket kan resultera i ett falskt negativt resultat. På senare tid har digital brösttomosyntes (DBT) utvecklats som svar på DMs brister. Till skillnad från DM, där en enskild bild tas i varje position, rör sig röntgenröret

i en båge och samlar in bilder från flera olika vinklar. Dessa bilder rekonstrueras sedan till tunna skiktbilder som bildar en tredimensionell volym (ofta kallad 2,5 D eftersom bilderna samlas in i en begränsad båge på 15°–50° istället för 360°). Detta gör det möjligt för radiologen att granska bröstet snitt för snitt och på så sätt komma runt maskeringseffekten [3].

I studien Malmö Breast Tomosynthesis Screening Trial undersöktes just detta, om DBT har en större klinisk nytta än DM. I studien påvisades att cancerdetektionen var 8,7 per 1 000 kvinnor undersökta med DBT, kontra 6,5 per 1 000 kvinnor med DM. Detta innebar att cancerdetektionen ökade med 34% med DBT jämfört med DM [4]. Dessa resultat återspeglas även i europeiska riktlinjer från ECIBC, där DBT rekommenderas som alternativ till DM inom screening, speciellt för kvinnor med hög bröstdensitet [5].

Trots fördelarna med DBT är det fortfarande inte standardmetoden för screening i Sverige. Detta beror på att tekniken är betydligt mer resurskrävande än DM. Både bildtagningen och rekonstruktionen tar längre tid än konventionell DM, men den mest kritiska aspekten är granskningen av bilden. Radiologen måste granska uppemot 50 bilder per bröst och undersökning, vilket gör det till en tidskrävande process som i sin tur belastar en redan ansträngd vård ännu mer [3]. I Socialstyrelsens rekommendation hänvisar de även till att det fortfarande är osäkert om DBT-screening minskar dödligheten, samtidigt som det skulle kräva stora resurstillskott att införa DBT i det nationella screeningprogrammet [6].

AI har under den senaste tiden uppvisat lovande resultat inom diagnostik, identifiering av biomarkörer och prognos av sjukdomsförlopp [7]. I en omfattande studie utförd vid Lunds Universitet (MASAI-studien) konstaterades att AI-stöttat beslutsstöd kunde detektera 29% fler tumörer vid screening [8], samtidigt som arbetsbördan för radiologerna minskade med 44% [9]. Studien påvisade även en signifikant ökning av sensitiviteten, 80,5% för gruppen med AI-beslutsstöd jämfört med 73,8% för kontrollgruppen, utan att påverka specificiteten [8].

De modeller som används för medicinsk bildanalys är ofta baserade på konvolutionella neurala nätverk (CNN), en typ av Deep Learning (DL). Uppbyggnaden av dessa nätverk är inspirerade av biologiska neurala strukturer och tränas genom att lagervis filtrera ut och viktta särdrag i bilddata. Detta gör att dessa modeller kan urskilja komplexa mönster som inte hade kunnat uppfattas av det mänskliga ögat [7], [10]. För att utveckla en bra modell krävs dock stora mängder av högkvalitativ träningsdata, vilket har visat sig vara en av svårigheterna med AI implementering. Dessutom krävs även heterogen data för att modellen ska fungera bra. Om ett dataset är för homogent riskerar modellen att utveckla ett bias där träffsäkerheten sjunker för underrepresenterade grupper [10]. Tack vare de olika screeningprogram världen över har forskare inom bröstcancerforskningen en unik möjlighet att uppnå ett

Inlämnat den 6 juni 2026

Emailadress: {da4874in-s@student.lu.se, vi7540pi-s@student.lu.se}

Huvudhandledare: Magnus Dustler, magnus.dustler@med.lu.se, Institutionen för translationell medicin

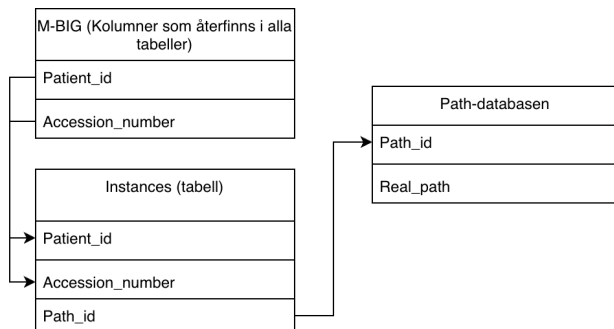
Biträdande handledare: Daniel Förnvik, daniel.fornvik@med.lu.se, Institutionen för translationell medicin

Svensk titel: DICOMplicate

Engelsk titel: DICOMplicate

högkvalitativt och heterogent dataset.

I Sverige finns en av de främsta databaserna med mammo-grafiundersökningar, nämligen Malmö Breast Imaging Database (M-BIG). Databasen omfattar ca 450 000 undersökningar från över 100 000 kvinnor, insamlade mellan 2004 och 2020. M-BIG innehåller både DM- och DBT-bilder med tillhörande radiologutlåtanden och registerdata [11]. Åtkomsten till databasen sker via ett PostgreSQL-gränssnitt där metadata är strukturerade i olika tabeller. Kopplingen mellan tabellerna med metadata och de faktiska bildfilerna sker via ett `path_id` som återfinns i tabellen `instances` som i sin tur kan kopplas till en ytterligare databas med den faktiska sökvägen, `path-databasen`, se Figur 1.



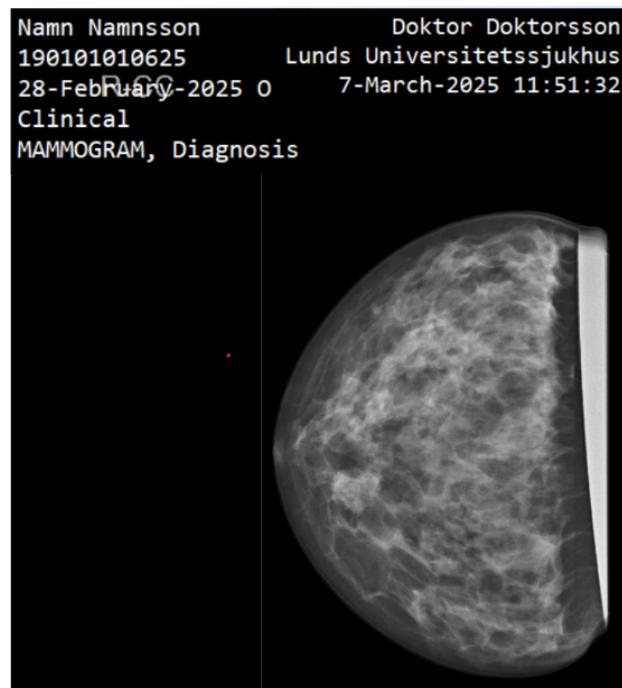
Figur 1. Entity-Relationship-diagram över M-BIG-databasen.

Det finns ett stort intresse för delning M-BIG-datan, för att möjliggöra utvecklingen av fler bra AI-modeller. Den största utmaningen för att göra detta ligger i att dela tillräckligt detaljerad data för att träna en AI-modell utan att bryta mot Patientdatalagen och GDPR. Även om tabellerna är pseudonymiserade är själva bildformatet som undersökningarna är sparade i det största hindret. Bildformatet, DICOM (Digital Imaging and Communications in Medicine), bakar ihop metadata och bilddata från undersökningen. Bilderna får då en "header" som innehåller känslig metadata, inklusive namn och personnummer direkt i filen [11]. Ett exempel på hur detta kan se ut visas i Figur 2, där känslig metadata visas direkt i bilden.

För att möjliggöra datadelning som kan kompletteras med ytterligare data i ett senare skede krävs därför en process för pseudonymisering snarare än total anonymisering. Vid anonymisering raderas all patientdata, vilket skulle göra det omöjligt att genomföra longitudinella studier, då kopplingen mellan en patients undersökningar kommer att försvinna. Genom pseudonymisering ersätts istället den känsliga metadata med unika koder, vilket gör att forskare och läkare kan titta på alla undersökningar från en patient utan att röja dennes identitet. Då det hittills saknats ett skalbart och säkert pseudonymisering verktyg har datan varit begränsad till ett fåtal personer i Malmö [11].

## B. Test

Med 9 000 personer som drabbas av bröstcancer varje år är behovet av ny forskning stort. Att utnyttja databaser för att träna och utveckla AI modeller har visat sig vara en stor tillgång för forskning. M-BIG-databasen, med över



Figur 2. Exempelbild på hur metadata bakas in i en DICOM-fil

450 000 bröstundersökningar, har därför stor potential men har i nuläget signifikanta begränsningar. Begränsningarna omfattar både höga tekniska trösklar för de faktiska användarna och även problematik kring att ta fram bilderna pseudonymiserat, vilket är en grund för att uppfylla integritetskrav och följa regelverk som GDPR.

Vi utvecklade därför DICOMplicate. En webbsida som inte bara är användarvänlig utan även automatiskt pseudonymiserar bilderna. På så sätt underlättar vi för databasen att användas till sin fulla potential och möjliggör vidare forskning för att fortsätta kampen mot bröstcancer.

## C. Agenda

Metoden beskriver varför och hur webbsidan utvecklades. Webbsidans användargränssnitt och effektivitet visas i resultatet, vilket ligger till grund för den efterföljande diskussionen. Diskussionen omfattar även lösningens begränsningar och utvecklingspotential samt aspekter som hållbar utveckling och etik.

## II. METOD

### A. Förarbete och val av teknisk lösning

1) *Förstudie*: Projektet inleddes med en litteraturstudie för att skapa en tydlig förståelse för projektet och dess omfattning. Särskild vikt lades vid att sätta sig in i databasen M-BIG, vilket var den databas som projektet skulle utgå ifrån. Studien fokuserade även på att förstå hur data lagras i DICOM-filer och hur SQL, det databasspråk som används för att kommunicera med M-BIG, är uppbyggt och används.

2) *Kravspecifikation:* Genom diskussion med användare av databasen togs en kravspecifikation fram. Kravspecifikationen uppdaterades under projektets gång och resulterade slutligen i tre huvudsakliga krav.

Det första kravet var att automatisera både framtagandet av data och framtagandet och pseudonymiseringen av bilder. Metadata utgör ofta en viktig del i forskningsundersökningarna vilket motiverar behovet av en exporterbar tabell med datan. Framtagandet och pseudonymiseringen av bilder var tidigare onödigt tidskrävande, bland annat på grund av olika sökvägar beroende på vilken data och vilka bilder som efterfrågades, tillsammans med avsaknaden av en tydlig koppling mellan metadata och DICOM-filer. Eftersom pseudonymiseringen tidigare utfördes manuellt i flera steg, bidrog detta ytterligare till ineffektiviteten.

Det andra kravet var att användaren inte skulle få direkt tillgång till path-databasen, utan att kopplingen skulle hanteras i bakgrunden. Detta krav tillkom på grund av behovet att minimera exponering av känslig information och på så sätt även minimera risken för intergritetsläckor.

Det tredje kravet var att lösningen skulle vara användarvänlig. Eftersom användningen av databasen tidigare har krävt teknisk expertis och programmeringskunskaper har den varit svår för forskarna att använda självständigt. Kravet blev därför att lösningen skulle vara intuitiv och lättanvänd för att undvika behovet av extern hjälp.

För att uppfylla dessa tre krav drogs slutsatsen att lösningen borde vara ett grafiskt användargränssnitt (GUI). Genom ett GUI kan komplexa processer, med hjälp av rätt kod, hanteras och automatiseras i bakgrunden. Vidare kan sekretessbelagd information hållas dold, samtidigt som användaren får tillgång till pseudonymiserad data. Slutligen möjliggör ett GUI, genom en genomtänkt utformning, en lösning som är både användarvänlig och intuitiv.

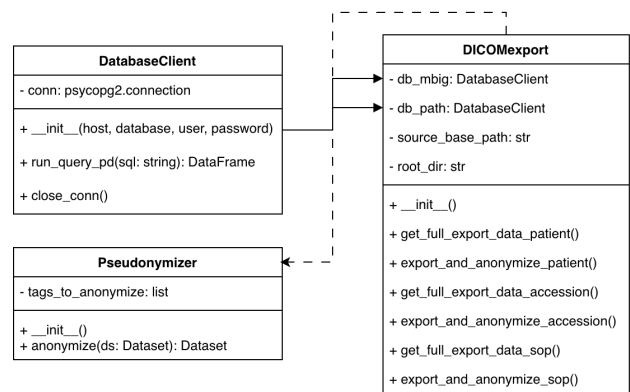
3) *Val av verktyg:* Python valdes som programmeringsspråk då de är lättanvänt och mångsidigt, samt erfarenhet av Python från tidigare kurser. Valet stärktes ytterligare av att Python erbjuder `pydicom`, vilket är ett välutvecklat bibliotek för hantering av DICOM-filer.

För val av verktyg till GUI undersöktes alternativ som Tkinter, PyQt och Streamlit. Då målet med projektet var att implementera funktioner och inte skapa ett GUI från grunden valdes Streamlit. Streamlit har inbyggda funktioner för exempelvis knappar och filhantering vilket gör det både lättanvänt och tidseffektivt för att bygga ett GUI.

## B. Systemarkitektur, Design och översiktlig beskrivning av klasserna

Vid programutformningen valdes en modular arkitektur, vilket innebär att programmet delas upp i mindre självständiga delar. Detta gjordes dels för att förenkla felsökning, dels för att öka återanvändbarheten. Problemet bröts upp i mindre delproblem, vilka identifierades som databaskoppling, export och pseudonymisering. Dessa blev i sin tur tre klasser med var sitt ansvarsområde baserat på dessa delproblem, se Figur 3.

Den första klassen, `DatabaseClient` utformades för att upprätta all kommunikation till databasen, både själva



Figur 3. UML-klassdiagram över programmet arkitektur som beskriver ansvarsfördelningen mellan databaskoppling, pseudonymisering och export

kopplingen och extraktionen av data. Klassen `DICOMexport` utformades för att hitta bildfilerna och strukturera dem i nya exportmappar. Slutligen implementerades `Pseudonymizer` vars uppgift var att anonymisera eller pseudonymisera metadata i DICOM-filerna.

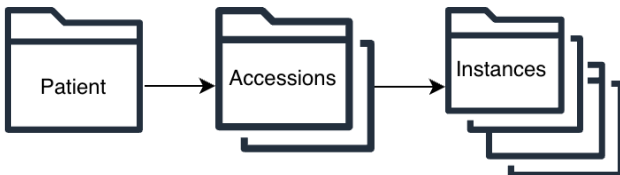
1) *DatabaseClient:* För att upprätta själva kopplingen implementerades en konstruktor som tog emot inloggningsuppgifter och andra parametrar för att hitta databasen. Därefter implementerades den centrala metoden `run_query_pd` som ställde SQL-frågor till databasen och returnerade resultat som en `Pandas DataFrame`, ett format som är enklare att hantera och möjliggör enklare hantering av data i de andra delarna av programmet.

2) *DICOMexport:* `DICOMexport` baserades främst på två centrala metoder, `get_full_export_data` och `export_and_anonymize`. Dessa metoder anpassades för att hantera export på nivåerna `patient_id`, `accession_number` och `sop_instance_uid`, beroende på användarens val i gränssnittet. Detta för att möjliggöra för användaren att filtrera ut olika typer av data, om användaren antingen är intresserad av en enskild bild, undersökning eller patient. Som attribut tog klassen emot kopplingar till både M-BIG-databasen och path-databasen i form av två `DatabaseClient`-objekt.

Metoden `get_full_export_data` utformades för att ta emot en `DataFrame` som parameter. Utifrån denna extraherades samtliga matchande rader från tabellen `instances`. Därefter hämtades tillhörande `path_id` för att lokalisera de fysiska filsökvägarna i path-databasen i en ny `DataFrame`. Dessa två `DataFrames` slogs samman med metoden `merge` för att skapa en slutgiltig tabell innehållande tre kolumner: `användarval`, `path_id` och `real_path` (där `användarval` exempelvis är `accession_number`).

Metoden `export_and_anonymize` tog den sammanställda tabellen, ett `Pseudonymizer`-objekt och namnet på exportmappen som parameter. Vid exekvering skapades först en mappstruktur baserad på användarens val, varpå systemet itererade genom tabellens rader. Med hjälp av biblioteket `Pydicom` och funktionen `dcmread` öppnades bildfilerna direkt i primärminnet, där de pseudonymiserades via `Pseudonymizer`-klassens `anonymize`. Slutligen generera-

des unika filnamn baserat på attributen för bildvinkeln och tidpunkten vid bildtagning (*AcquisitionTime*). Beroende på användarens val sparades bilderna i olika hierarkier. Vid valet *patient\_id* exporteras alla undersökningar och bilder som tillhör varje *patient\_id* som sökningen resulterar i. Vid valet *accession\_number* sparas istället alla undersökningar från resultatet med tillhörande bildfiler. Slutligen sparas alla bildfiler vid export baserad på *sop\_instance\_uid*. I alla tre fall är det samma mapphierarki, där *patient\_id* är huvudmapp, där under *accession\_number* och slutligen bildfilerna (se Figur 4).



Figur 4. Exporthierarki

3) *Pseudonymizer*: Klassen *Pseudonymizer* utformades för att hantera den faktiska pseudonymiseringen av metadata i DICOM-filerna. Genom metoden *anonymize* bearbetades varje bild utifrån en förutbestämd lista av DICOM-taggar. Denna metod utformades för att antingen nollställa attribut eller ersätta dem med pseudonymiserad data.

Valet mellan pseudonymisering och nollställning baserades på attributens känslighet och framtida användningsområde. Taggar som bedömdes utgöra en direkt säkerhetsrisk utan kliniskt värde nollställdes helt. För att möjliggöra longitudinella studier, där data från olika tidpunkter ska kunna kopplas till samma patient, valdes pseudonymisering, där bland annat *patient\_id* användes. Eftersom projektets databas innehöll pseudonymiserad data, programmerades metoden att extrahera och applicera dessa befintliga värden på de relevanta taggarna i bildfilen. Besluten rörande vilka taggar som behövde nollställas eller pseudonymiseras fattades i samråd med projektets handledare samt ansvarig för databasen, Victor Dahlblom, för att säkerställa att lösningen varken raderade viktig klinisk data eller stred mot juridiska krav.

### C. Flödeshantering

Genomgående för hela lösningen sköttes all datahantering i primärminnet (RAM). Detta val motiverades främst av två aspekter. För det första optimerades prestandan, genom att bearbeta datan i primärminnet behövde filerna aldrig laddas ner på datorn förrän de var färdigbehandlade och redo för export. För det andra eliminerades risken för att känslig information tillfälligt skulle lagras på datorns hårddisk. Den oskyddade data som skulle kunna äventyra patientens integritet existerade därmed bara temporärt och kunde därför inte lämna några spår efter att programmet avslutats.

### D. Felhantering

För att undvika programavslut implementerades felhantering med *try-except*-block. Metoden fungerar som en säkerhetsåtgärd, om ett fel uppstår under exekvering av koden i

*try*-blocket fångas felet upp och skickas vidare till *except*-blocket som kan ta emot det utan att hela applikationen avslutas oväntat.

Denna metod var särskilt viktig vid implementationen av klassen *DICOMexport* där funktionen *dcmread* användes. Då databasen innehöll filer som i vissa fall saknade innehåll skulle en inläsning av sådan fil resulterat i att funktionen *dcmread* hade stoppat programmet. Genom att implementera *dcmread* i ett *try-except*-block istället, kunde programmet göra ett undantag (exception) och sedan gå vidare till nästa fil. Denna metod användes genomgående i programmet för att öka applikationens stabilitet.

Utöver hantering av exekveringsfel implementerades även några spärrar för att skydda databasens innehåll. Då vissa kommandon i SQL kan förändra eller radera data var det nödvändigt att förhindra exekveringen av sådana kommandon. Detta löstes genom att göra en lista över alla otillåtna kommandon (t.ex. *DELETE*, *DROP* och *MERGE*). Programmet kontrollerar då om den inmatade frågan innehåller något av dessa ord och förhindrar exekveringen i sådana fall. Det genereras även ett varningsmeddelande som informerar användaren om att ett sådant kommando inte får användas. Då det är svårt att upptäcka alla riskabla kommandon, är det om möjligt bra om användaren endast har läsbehörighet till databasen.

### E. Implementering

Med hjälp av biblioteket *Streamlit* och klassen *DatabaseClient* skapades en webbsida. Alla visuella delar av webbsidan, såsom knappar och sökfält, skapades med hjälp av *Streamlits* metoder. Eftersom projektets huvudsakliga fokus låg på att utveckla den bakomliggande logiken för webbsidan beskrivs inte implementeringen av de grafiska funktionerna i detalj i rapporten.

Därefter utvecklades funktionen för att välja vilken data som skulle hämtas. För att uppfylla kravet om användarvänlighet övervägdes inledningsvis en lösning med färdiga alternativ, där användaren skulle kunna välja att sortera data utifrån exempelvis datum, ålder eller undersökningstyp. Ett problem med detta alternativ var dock att det skulle bli svårt att skapa tillräckligt många alternativ för att möjliggöra så specifika frågor som vanligtvis behövs vid användningen av databasen. Efter diskussion och utvärdering med databasanvändare drogs slutsatsen att färdiga alternativ skulle begränsa mer än underlätta användningen. Därför beslutades istället att implementera ett sökfält där användaren kan formulera frågor i SQL. Detta gjordes med hjälp av metoden *run\_query\_pd* från *databaseClients* där SQL-frågan blev indatan till *run\_query\_pd*. Funktionen designades så att datan som hämtats från M-BIG, i form av en *DataFrame*, presenterades i en tabell under sökfältet.

För att samtidigt bibehålla användarvänligheten inkluderades även en "fusklapp" ovanför sökfältet i form av ett expanderbart fönster med vanliga SQL-kommandon. På så sätt krävs endast grundläggande kunskap i SQL för att kunna formulera både specifika och avancerade frågor.

För att implementera export-funktionerna skapades ett sökfält för att välja sökvägen till mappen. Det gav

användaren möjlighet att både bestämma var mappen skulle sparas och vad den skulle heta. Sökvägen sattes till variabeln `user_folder_name`. Tre olika knappar implementerades i användargränssnittet för att kunna exportera baserat på patient (`patient_id`), undersökning (`accession_number`) och fil (`sop_instance_uid`). Beroende på vilket nivå-val som gjordes användes metoden `get_full_export_data_[nivå]` som sattes till variabeln `final_df`. Slutligen användes metoden `export_and_anonymize_[nivå]` med både `final_df` och `user_folder_name` som indata för att skapa mappen med de pseudonymiserade filerna.

En inloggningssida utvecklades med fält för användarnamn och lösenord till databasen, med krav för att kunna använda webbsidan. Dessa uppgifter valdes som indata till klassen `DatabaseClient`.

Slutligen skapades en så kallad batch-fil för att underlätta uppstarten av webbsidan och för att uppfylla kravet om att användaren inte ska ha tillgång till path-databasen. Innan batch-filen skapades behövde webbsidan startas via utvecklingsmiljön, där både anslutningsuppgifter till path-databasen och all kod var exponerade för användaren. Genom att öppna batchfilen, kopplad till pythonfilen med koden, öppnas Streamlit-applikationen automatiskt utan att användaren går via utvecklingsmiljön. Det gör lösningen både säkrare och mer användarvänlig.

Ett genomgående problem i utvecklingen av webbsidan var att ge korrekta fel-kommandon och säkerställa att sidan inte skulle kunna krascha. Precis som i utvecklingen av klasserna användes `try-except`-block även i utvecklingen av webbsidan varpå felmeddelande returnerades till webbsidan. Ett problem för `run_query` var att sidan kraschade om man inte ställde en fungerande SQL-fråga. Detta löstes genom en `rollback`-funktion som gjorde att om en fråga inte gick att genomföra återställdes sidan till hur den var innan frågan ställdes och ett felmeddelande returnerades.

### F. Testning

För att säkerställa funktionen av lösningen gjordes tester kontinuerligt under utvecklingsprocessen. Genom regelbunden testning kunde fel upptäckas tidigt, vilket underlättade att både hitta och åtgärda dem.

Tester utfördes bland annat för att kontrollera att den framtagna datan var korrekt, genom att jämföra med den ursprungliga datan. Vidare verifierades pseudonymiseringen genom att inspektera "DICOM-headers" och funktionen av felkommandon säkerställdes genom att simulera olika tänkbara felfall. Slutligen genomfördes prestandatester på olika stora dataset (50, 100, 1000 och 5000 bilder) för att undersöka lösningens effektivitet och hur lång tid extraktionerna skulle ta i praktiken.

### G. Användning av AI

Under projektets gång har artificiell intelligens använts som stödverktyg, främst genom språkmodellen Gemini (Google). Verket har använts som ett bollplank för att utvärdera

metoder och tekniker, samt som tekniskt stöd under programmeringsfasen. Inom denna fas har modellen främst använts för stöd vid felsökning. För att säkerställa modellens tillförlitlighet har allt som AI-genererats granskats noggrant och kontrollerats manuellt.

## III. RESULTAT

1) *Resultat av användargränssnitt:* Resultatet av projektet är applikationen DICOMplicate, ett användargränssnitt uppbyggt i Streamlit. Gränssnittets utformning vid inloggning visas i Figur 5, medan vyn efter lyckad inloggning visas i Figur 6.

### DICOMplicate

Figur 5. Grafik innan inloggning

### DICOMplicate

Figur 6. Grafik efter inloggning

Överst i vyn återfinns en "fusklapp" (se Figur 7) som fungerar som stöd för användaren vid formulering av SQL-frågor. Under "fusklappen" finns ett textfält där användaren anger sin SQL-fråga, följt av knappen "Run query" för att exekvera sökningen (se Figur 6).

| SQL Cheat Sheet - Quick Guide |                                                                          |
|-------------------------------|--------------------------------------------------------------------------|
| Goal                          | SQL Code (Example)                                                       |
| Fetch all records             | <code>SELECT * FROM [table]</code>                                       |
| Specific Patient              | <code>WHERE patient_id = '[patient_id]'</code>                           |
| Date Range                    | <code>WHERE [parameter] BETWEEN '[xxxx-xx-xx]' AND '[yyyy-yy-yy]'</code> |
| Specific Equipment            | <code>WHERE manufacturer LIKE '%Siemens%'</code>                         |
| Combine Requirements          | <code>WHERE ([parameter 1] = 'X' AND [parameter 2] &gt; 'Y')</code>      |
| Limit Results                 | <code>LIMIT 10 (Shows the first 10 rows)</code>                          |

Figur 7. Grafik över fusklappen

När en sökning har genomförts presenteras resultatet i form av en tabell (se Figur 8). I detta skede uppkommer även ett skrivfält där användaren kan välja vart exportmappen ska sparas. Som standard föreslås en mapp på användarens

skrivbord med namnet My\_Dicom\_Export. Utöver destinationen för exportmappen visas även tre olika exportknappar: `sop_instance_uid`, `accession_number` och `patient_id`. När exporten slutförts uppkommer en grön ruta där filvägen för exportmappen syns samt antalet exporterade DICOM-filer. De exporterade filerna kommer att sorteras på det sätt som beskrivs i (Avsnitt II-B2). Huvudmappen namnges efter Patient ID, följt av undermappar namngivna med undersökningsdatum och Accession number. DICOM-filnamnen baseras på bildvin- kel och acquisition time. Dessutom är det möjligt att ladda ner själva resultattabellen.

#### Results

|   | media_storage_sop_instance_uid                                   | sop_instance_uid                                                 | study_instance_uid  |
|---|------------------------------------------------------------------|------------------------------------------------------------------|---------------------|
| 0 | 1.3.6.1.4.1.9590.100.1.2.250833360313870995335801560783380443601 | 1.3.6.1.4.1.9590.100.1.2.250833360313870995335801560783380443601 | 1.3.6.1.4.1.9590.10 |
| 1 | 1.3.6.1.4.1.9590.100.1.2.348470167611601185326152070661695480029 | 1.3.6.1.4.1.9590.100.1.2.348470167611601185326152070661695480029 | 1.3.6.1.4.1.9590.10 |
| 2 | 1.3.6.1.4.1.9590.100.1.2.36033261412263211418807103031469863810  | 1.3.6.1.4.1.9590.100.1.2.36033261412263211418807103031469863810  | 1.3.6.1.4.1.9590.10 |
| 3 | 1.3.6.1.4.1.9590.100.1.2.7725749312473821728473445131054777521   | 1.3.6.1.4.1.9590.100.1.2.7725749312473821728473445131054777521   | 1.3.6.1.4.1.9590.10 |
| 4 | 1.3.6.1.4.1.9590.100.1.2.393694225710141280511254591170229049464 | 1.3.6.1.4.1.9590.100.1.2.393694225710141280511254591170229049464 | 1.3.6.1.4.1.9590.10 |

Enter path

C:\Users\Marvin\Desktop\My\_DICOM\_Export

Start export based on sop\_instance\_uid

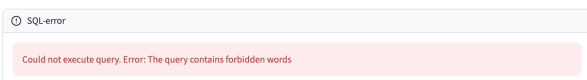
Success! The file is stored in: (C:\Users\Marvin\Desktop\My\_DICOM\_Export\5)

Start export based on accession\_number

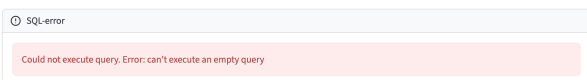
Start export based on patient\_id

Figur 8. Grafik när resultatet visas

2) *Felhantering*: Systemets felhantering som beskrivs i (Avsnitt II-D) förhindrar inmatning av ogiltiga värden eller otillåtna SQL-kommandon. Om användaren försöker använda förbjudna ord, såsom “delete” visas ett varningsmeddelande (se Figur 9). Om användaren istället försöker köra en tom fråga genereras ett felmeddelande som informerar användaren om att en fråga saknas (se Figur 10)



Figur 9. Felmeddelande när ett förbjudet ord använts

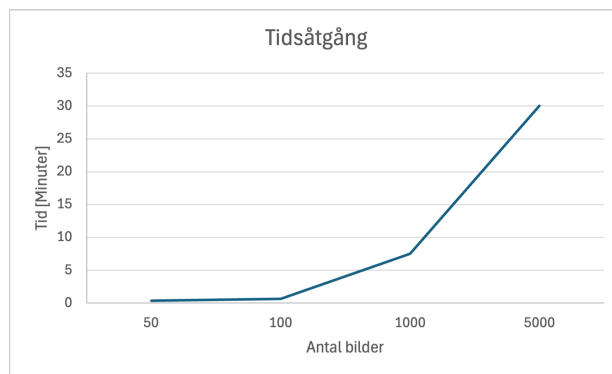


Figur 10. Felmeddelande när skrivfältet lämnas tomt

3) *Effektivitet och exporthastighet*: Utvärderingen av systemets exporthastighet baserades på tidsåtgången för att exportera ett specifikt antal bilder via export baserad på `sop_instance_uid` enligt metodavsnitt. I Tabell I och Figur 11 presenteras export av 50, 100, 1000 respektive 5000 bilder. Resultatet visar på en stabil och skalbar prestanda där tidsåtgången ökar linjärt i förhållande till datamängden.

Tabell I  
TIDSÅTGÅNG FÖR EXPORT AV DICOM-BILDER

| Antal bilder | Tid [min] |
|--------------|-----------|
| 50           | 0,33      |
| 100          | 0,66      |
| 1000         | 7,5       |
| 5000         | 30        |



Figur 11. Diagram över tidsåtgång för export av DICOM-bilder

## IV. DISKUSSION

### A. Utvärdering av lösningen

Jämfört med tidigare lösningar för att pseudonymisera M-BIGs DICOM-filer är DICOMplicate en betydligt mer effektiv lösning. Användaren behöver varken växla mellan olika program, eller ha kunskap i hur metadatan och bilderna är kopplade. Dessutom behövs ingen ny kod för varje pseudonymiserings-process.

Lösningen uppfyller även till viss del kravspecifikationen. Den sköter pseudonymiseringen automatiskt och användaren får inte direkt tillgång till path-databasen. Dessutom är den relativt användarvänlig med tydliga knappar, fält och statusmeddelanden (se i figur 5 - 10). Ytterligare är webbsidan relativt snabb vilket kan ses i resultatet från prestandatesterna i tabell I och figur 11. Exempelvis tar pseudonymisering och export av 5000 filer 30 minuter vilket motsvarar cirka 0.36 sekunder per fil.

Lösningen har dock fortfarande en del begränsningar. För att ta fram data och bilder krävs det att användaren skriver frågan i SQL. Eftersom de tänkta användarna inte är utbildade i SQL gör det lösningen mer svårhanterlig. Även om “fusklappen” kan underlätta en del så krävs fortfarande att användaren fått någon form av introduktion till SQL. Möjliga lösningar och vidareutvecklingsmöjligheter diskuteras i Framtida potential.

En annan begränsning är att för att användaren ska kunna ta fram ett specifikt dataset måste den specificera från vilken tabell och ofta vilka kolumner datan ska komma ifrån. I det grafiska gränssnittet finns ingen sådan information, utan det förutsätts att användaren redan vet. Eftersom de tänkta användarna ofta redan är insatta i databasen bör det inte vara ett omfattande problem. En möjlig lösning för vidare utveckling hade kunnat vara att addera ett kompendium med information om de olika tabellerna och kolumnerna.

## B. Hållbar utveckling

DICOMplicate har potential att bidra till ekonomisk, social och teknisk hållbarhet.

Ur ett ekonomiskt perspektiv kan lösningen bidra till ökad resurseffektivitet. Med hjälp av automatiserad pseudonymisering kan lösningen effektivisera det tidigare manuella arbetet avsevärt. Det kan i sin tur frigöra tid och resurser som istället kan användas för exempelvis forskning och analys.

Ur ett socialt perspektiv kan lösningen, genom att minska risken för mänskliga fel, bidra till ökad datasäkerhet. Eftersom databasen innehåller känslig medicinsk information är det av stor vikt att identifierbar information pseudonymiseras korrekt. Manuell pseudonymisering ökar risken för exempelvis slarvfel, vilket kan leda till att känslig information exponeras. Genom automatisering hanteras data mer konsekvent och, förutsatt att koden är korrekt konfigurerad, minskar risken för fel avsevärt.

Ytterligare kan lösningen anses vara tekniskt hållbar. På grund av den modulära kodstrukturen är det tydligt hur systemet är uppbyggt. Det underlättar vidareutveckling och exempelvis anpassning till andra databaser.

## C. Etik

Vidare berör projektet flera etiska punkter.

En central etisk aspekt är att databasen innehåller mycket känslig information och det är därför avgörande att lösningen fungerar som avsett och inte innehåller brister, exempelvis i pseudonymiseringen. För att säkerställa detta krävs ytterligare och mer omfattande tester som komplement till den granskning och de tester som redan genomförts, se avsnitt II-F. I dagsläget krävs det att personen som ska ta fram de pseudonymiserade bilderna har tillgång till path-databasen, som innehåller kopplingen mellan personlig identifierare och personnummer. Denna information är särskilt känslig och kan, om den exponeras, äventyra patienternas integritet. Om lösningen fungerar som avsett behövs inte längre direkt åtkomst till denna databas, vilket ytterligare skyddar denna känsliga information.

En annan etisk aspekt gäller risken kopplad till felaktigt formulerade frågeställningar, eftersom de tänkta användarna inte nödvändigtvis är vana vid att arbeta med SQL. Detta i sin tur skulle kunna leda till analyser av felaktig eller ofullständig data och resultera i missvisande slutsatser. Därför är det viktigt att användarna har förståelse för databaser och SQL innan lösningen används.

## D. Framtida potential

Även om lösningen uppfyller de mest grundläggande kraven som identifierades i projektets början finns det fortsatt utvecklingspotential.

En möjlig vidareutveckling är att förbättra stödmaterialet som lösningen tillhandahåller för att underlätta användningen av systemet. Exempelvis skulle, som tidigare nämnt, ett kompendium med information om databasens olika tabeller och kolumner kunna bidra, då det i dagsläget inte finns något sådant stödmateriäl. Dessutom skulle den nuvarande "SQL-fusklappen" kunna utvecklas då den idag är högst grundläggande.

En annan relativt enkel, men värdefull vidareutveckling, skulle vara att begränsa åtkomsten till mappen som innehåller programmeringskoden genom lösenordsskydd. Detta skulle gjort lösningen säkrare då koden och dess känsliga innehåll skulle vara mer skyddad från obehörig åtkomst och förändringar.

Vidare skulle lösningen bli betydligt mer användarvänlig om användaren inte själv behöver formulera frågorna i SQL. En möjlig lösning skulle kunna vara att integrera en AI-baserad funktion som kan omformulera vardagligt språk till SQL-frågeställningar. En sådan lösning skulle dock innebära flera risker och utmaningar, exempelvis kopplade till blackbox fenomen, vilket leder till svårigheter att både förstå och verifiera hur AI-systemet genererar sina svar. Med fortsatt utveckling och mer tillförlitliga AI-system är det fortfarande en möjlig framtida utveckling som skulle kunna sänka tröskeln för användning av systemet ytterligare.

Slutligen finns potential att anpassa lösningen för att användas till fler databaser. Det är möjligt att det finns databaser som liknar M-BIG och som har liknande svårigheter i användningen av den, där en liknande lösning skulle kunna vara till nytta. Tack vare den modulära arkitekturen av koden är den relativt lätt att förstå och vidareutveckla. Därför skulle denna lösning förhoppningsvis även kunna anpassas till andra databaser.

## V. SLUTSATSER

Projektet resulterade i ett grafiskt användargränssnitt, DICOMplicate, med syfte att underlätta användningen av M-BIG-databasen och forskningen den möjliggör. Lösningen har vissa begränsningar, exempelvis att användarna fortfarande behöver ha viss programmeringskunskap för att formulera frågor i SQL. Trots det kan lösningen anses vara både användarvänlig och effektiv, då den automatiserar tidigare tidskrävande processer som pseudonymisering och framtagningen av sökvägar.

## VI. EFTERORD

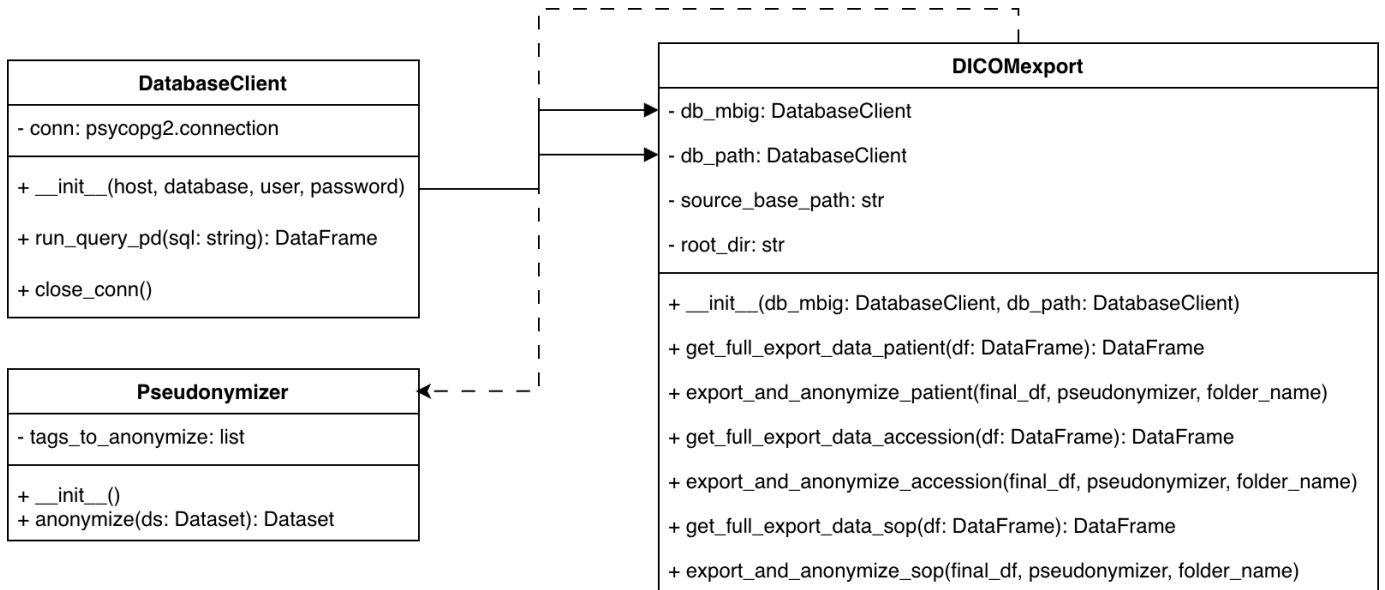
Vi vill rikta ett stort tack till våra handledare, Magnus Dustler och Daniel Förnvik, för vägledning och stöd under projektets gång. Vi vill även tacka Victor Dahlblom, ansvarig för databasen, för värdefulla synpunkter kring lösningen och stöd i arbetet med databasen. Slutligen vill vi tacka Ester Daniel Ytterbrink som bidrog till en bra start på projektet och gav oss grunden för förståelse av hur databaser och SQL fungerar. Arbetet under projektet och skrivandet av rapporten har varit jämt uppdelat mellan författarna.

## REFERENSER

- [1] Internetmedicin, "Bröstcancer, utredning,"[Online]. Tillgänglig: <https://www.internetmedicin.se/onkologi/brustcancer-utredning>. [Hämtad: 24-april-2026].
- [2] Socialstyrelsen, "Bröstcancer – screening med mammografi,"[Online]. Tillgänglig: <https://www.socialstyrelsen.se/kunskapsstod-och-regler/regler-och-riktlinjer/nationella-screeningprogram/slutliga-rekommendationer/brustcancer/>. [Hämtad: 24-april-2026].

- [3] SBU, "Bröstcancerscreening med 3D-mammografi – digital brösttomosyntes med Selenia Dimensions," Statens beredning för social och medicinsk utvärdering, SBU kommenterar nr 2019\_02, 2019.[Online]. Tillgänglig: [https://www.sbu.se/contentassets/28b19ffd589a4518bdb27c32d2073fa9/brostcancerscreening\\_3d\\_mammografi\\_digital\\_brosttomosyntes\\_selenia\\_dimensions.pdf](https://www.sbu.se/contentassets/28b19ffd589a4518bdb27c32d2073fa9/brostcancerscreening_3d_mammografi_digital_brosttomosyntes_selenia_dimensions.pdf). [Hämtad: 24-april-2026].
- [4] S. Zackrisson et al., "One-view breast tomosynthesis versus two-view mammography in the Malmö Breast Tomosynthesis Screening Trial (MBTST): a prospective, population-based, diagnostic accuracy study," *Lancet Oncol.*, vol. 19, no. 11, pp. 1493–1503, Nov. 2018. doi: 10.1016/S1470-2045(18)30521-7.
- [5] European Commission Initiative on Breast Cancer (ECIBC), "European breast cancer guidelines – Screening with tomosynthesis," Joint Research Centre, European Commission. [Online]. Tillgänglig: <https://cancer-screening-and-care.jrc.ec.europa.eu/en/ecibc/european-breast-cancer-guidelines?topic=65&updatef2=0&usertype=60#guideline>. [Hämtad: 19-maj-2026].
- [6] Socialstyrelsen, "Screening för bröstcancer – Rekommendation och bedömningsunderlag," Socialstyrelsen, Stockholm, Artnr. 2023-5-8564, maj 2023. [Online]. Tillgänglig: <https://www.socialstyrelsen.se/contentassets/a74c09ee052b4c8aa4a01dc75490ba33/2023-5-8564.pdf>. [Hämtad: 24-april-2026].
- [7] J. S. Ahn et al., "Artificial Intelligence in Breast Cancer Diagnosis and Personalized Medicine," *J Breast Cancer*, vol. 26, no. 5, pp. 405–435, Oct. 2023. doi: 10.4048/jbc.2023.26.e45
- [8] J. Gommers et al., "Interval cancer, sensitivity, and specificity comparing AI-supported mammography screening with standard double reading without AI in the MASAI study: a randomised, controlled, non-inferiority, single-blinded, population-based, screening-accuracy trial," *Lancet*, vol. 407, no. 10527, pp. 505–514, Jan. 2026. doi: 10.1016/S0140-6736(25)02464-X.
- [9] K. Lång et al., "Artificial intelligence-supported versus breast cancer screening with digital mammography in the Mammography Screening with Artificial Intelligence trial (MASAI): a prospective, population-based, randomised, controlled, non-inferiority, single-blinded, screening accuracy trial," *Lancet Oncol.*, vol. 24, no. 9, pp. 936–944, Sep. 2023. doi: 10.1016/S1470-2045(23)00298-X.
- [10] J. L. Cross, M. A. Choma, and J. A. Onofrey, "Bias in medical AI: Implications for clinical decision-making," *PLOS Digit. Health*, vol. 3, no. 11, p. e0000651, Nov. 2024. doi: 10.1371/journal.pdig.0000651.
- [11] V. Dahlblom, "Improving breast cancer screening with artificial intelligence," Doctoral Thesis (compilation), Department of Translational Medicine. Lund University, Faculty of Medicine, 2024. [Online]. Tillgänglig: <https://portal.research.lu.se/en/publications/improving-breast-cancer-screening-with-artificial-intelligence/>

## BILAGA



Figur 12. Komplet UML-digagram