

Utveckling av ett uppföljningsverktyg i Microsoft Access

Rasmus Lindholm

Examensarbete för ingenjör (YH)-examen

Utbildningsprogrammet för produktionsekonomi

Vasa 2023

EXAMENSARBETE

Författare: Rasmus Lindholm

Utbildning och ort: Produktionsekonomi, Vasa

Handledare: Mikael Ehrs, Magnus Jakobsson

Titel: Utveckling av ett uppföljningsverktyg i Microsoft Access

Datum: 19.4.2023 Sidantal: 34

Abstrakt

Examensarbetet är gjort för att underlätta uppföljningen av anmälningar vid Österbottens polisinrättning. Österbottens polisinrättningen var i behov ett verktyg, som var bättre än de befintliga verktygen och vilket kunde förbättra uppföljningen av anmälningar. Verktöget skulle klara av tusentals datainmatningar och tiotals användare, som använder verktyget samtidigt. Verktögen måste vara godkänt av IT-administratörerna. Utseendemässigt fick verktyget se ut som utvecklaren ville.

Arbetet utfördes som en systemutveckling enligt systemutvecklingens livscykel. Tre olika metoder inom systemutvecklingen undersöks, vattenfallsmetoden, den snabba applikationsutvecklingen och den flexibla metoden. Av dessa kommer utvecklingen ske enligt den metod som lämpar sig bäst. Utöver det användes en utvecklingsmetod med namnet Full stack, som är typisk inom webb- och applikationsutveckling där man delar upp utvecklingen i en gränssnitt- och en serverdel.

Verktöget utvecklades i databasprogrammet Microsoft Access, eftersom det var godkänt av IT-administratörerna och det är ett program som kan ha tiotals användare samtidigt och klarar av tusentals datainmatningar. I arbetet utnyttjades Access hela funktionalitet från tabeller till rapportering.

Resultatet blev ett uppföljningsverktyg som är uppdelat i två delar, en visuell- och ett kalkylbladsdel. Den visuella delen är uppdelad i en ärendespecifik del och i en tilläggsdel. I den ärendespecifika delen finns det funktioner som ändrar beroende på den information som matas in. Tilläggsdelen är gjord som tilläggsinformation och är inte obligatorisk för att man skall få en överblick över ärendet men som ändå kan hjälpa användaren. Kalkylbladsdelen ser ut som ett typisk kalkylblad och ärenden presenteras lättförståeligt och smidigt.

Språk: svenska

Nyckelord: systemutveckling, Microsoft Access, uppföljningsverktyg

OPINNÄYTETYÖ

Tekijä: Rasmus Lindholm

Koulutus ja paikkakunta: Tuotantotalous, Vaasa

Ohjaaja(t): Mikael Ehrs, Magnus Jakobsson

Nimike: Juttuseuranta työkalun kehittäminen Microsoft Accessissa

Päivämäärä 19.4.2023 Sivumäärä 34

Tiivistelmä

Opinnäytetyö on tehty helpottaakseen ilmoitusten seuranta Pohjanmaan poliisilaitoksella. Poliisilaitos tarvitsi paremman työkalu, kuin jo käytössä olevat työkalut ja työkalun avulla pystyisi parantamaan ilmoitusten seuranta. Työkalun piti pystyä hallitsemaan tuhansia syöttöjä, kymmeniä yhtäaikaista käyttäjiä ja se on oltava käyttöjärjestelmä valvojan hyväksymä. Kehittäjä itse sai päättää miltä työkalu näyttää.

Työ on toteutettu ohjelmakehityksenä ohjelmakehitys elinkaarin mukaisesti. Kolme eri menetelmää on tutkittu, vesiputousmenetelmä, nopea sovelluskehitys ja taipuisa menetelmä. Näistä kolmesta menetelmästä valitaan sopivimman. Tämän lisäksi työssä on käytetty Full stack kehitystä mitä yleensä käytetään nettisivujen ja applikaatioiden kehityksessä, missä jaetaan kehitystä kahteen osaan, rajapintaan ja palvelimeen.

Työkalu valittiin kehitettäväksi Microsoft Accessissa, tämä sen takia että se oli järjestelmävalvojan hyväksymä, siitä pystyy käyttämään kymmeniä yhtä aikaa ja se pystyy hallitsemaan tuhansia yhtäaikaista syöttöjä. Työssä on käytetty hyväksi koko Accessin toiminnallisuus.

Tuloksena oli seuranta työkalu, mikä on jaettu kahteen osaan, visuaaliseen- ja laskentataulukko osaan. Visuaalinen osa on jaettuna kahteen osaan asiaerityiseen ja lisäosaan. asiaerityisessä osassa on toimintoja mitkä muuttuvat lisätyn tiedon mukaan ja lisäosa on tehty lisätiedolle mikä ei ole pakollista olla laskentataulukossa mutta voi auttaa käyttäjää. Laskentataulukko on tyyppillisen laskentataulukon näköinen ja asiat ovat esillä helposti havaittavasti ja sujuvasti.

Kieli: ruotsi

Avainsanat: ohjelmistokehitys, Microsoft Access, seurantatyökalu

BACHELOR'S THESIS

Author: Rasmus Lindholm

Degree Programme: Industrial management, Vasa

Supervisor(s): Mikael Ehrs, Magnus Jakobsson

Title: Development of a Follow-up Tool in Microsoft Access

Date 19.4.2023 Number of pages 34

Abstract

The thesis is made for aiding the follow-up of reports at Ostrobothnia police department. The department needed a tool which was better than the already existing ones and it could improve the report follow-up. The tool had to be able to handle over 1000 inputs, tens of simultaneous users, and the tool had to be approved by the system admin. The developer was allowed to visually make the tool as he pleased.

The development was made as a system development according to the system development lifecycle. Three different methods were studied, the waterfall method, the rapid development, and the agile method. Of these three one method is chosen and the development will be made according to that method. A full stack method is also used to determine which is common in website and application development with a front-end and a back-end development.

The chosen system was Microsoft Access, which was approved by the system admin and is a program that handles thousand inputs and tens of simultaneous users. The whole broad spectrum of Access is used in the finished tool.

The result was a follow-up tool that is divided into two parts, a visual part, and a spreadsheet part. The visual part is divided into two parts, a report specific and an additional information's part. In the report specific part there are functions that are dependent on the inputted value and the additional information's part is additional information that isn't necessary but can help the user. The spreadsheet part is a typical spreadsheet, and the reports are presented understandably and conveniently.

Language: Swedish

Key words: system development, follow-up tool, Microsoft Access

Innehållsförteckning

1	Inledning.....	1
1.1	Polisen i Finland.....	2
1.1.1	Polisinrättningen i Österbotten	2
1.1.2	Kust Österbottens brottsstatistik 2021	2
1.2	Avgränsning.....	3
2	Systemutveckling och metoder.....	5
2.1	Systemutvecklingslivscykel.....	5
2.1.1	Vattenfallsmetoden.....	6
2.1.2	Rapid Application Development.....	7
2.1.3	Flexibel utveckling	8
3	Full Stack-utveckling.....	10
3.1	Gränssnitt.....	10
3.1.1	Verkställande informationssystem.....	10
3.1.2	Polisens hemsida	11
3.2	Serverdel.....	12
4	Microsoft Access	14
4.1	Tabeller	14
4.2	Formulär och rapporter.....	15
4.3	Frågor, makron och moduler	16
5	Utförande.....	17
5.1	Systemutvecklingen	17
5.1.1	Påbörjande av projektet.....	18
5.1.2	Planeringen.....	18
5.1.3	Analyseringen	19
5.1.4	Design.....	20
5.1.5	Implementering.....	21
5.2	Full stack-utvecklingen	23
5.2.1	Gränssnitt	24
5.2.2	Serverdel.....	24
6	Resultatet.....	25
6.1	Visuell del	25
6.2	Kalkylbladsdelen	27
6.3	Sammanfattning.....	28
7	Diskussion	29
8	Referenser.....	31

1 Inledning

I arbetet har jag som målsättning att utveckla ett informationssystem till Kust-Österbottens brottsundersökning vid Österbottens polisinrättning. Systemet utvecklas för att förenkla uppföljningen av brottsanmälningar och är främst ämnat för undersökare och undersökningsledare men kommer också att kunna användas för rapportering. Systemet skall inte innehålla sekretessbelagd information. Systemet kommer att förvaras på ett internt nätverk och kommer att vara krypterat så att inte obehöriga skall ha tillgång till innehållet i systemet. Det finns för tillfället inte något specifikt uppföljningssystem för brottsundersökningen utan uppföljningen görs på varierande sätt. Med systemet finns det möjligheten att kunna standardisera vissa tillvägagångssätt vid uppföljningen.

För att ett informationssystem skall tas i bruk måste det tillfredsställa behoven. Det skall innehålla samma funktioner som de nuvarande systemen samt kunna erbjuda ytterligare funktioner som underlättar för användaren. Användaren kommer främst vara personer som arbetar som poliser inom Kust-Österbottens område. För att kunna utveckla ett bättre system än de nuvarande undersöks det nuvarande systemen. På grund av sekretessen i Lag om offentlighet i myndigheternas verksamhet (621/1999), 24 kapitel 1 moment 5 paragraf kommer arbetet inte innehålla närmare beskrivning av de befintliga systemen inom polisinrättningen. Ett befintligt program för uppföljning som polisen använder är Microsoft Excel. Funktionaliteten med Excel är begränsad och för att kunna utnyttja alla funktioner i Excel kan endast en person åt gången modifiera ett dokument. Det finns en Excel online version som gör att flera personer kan modifiera samma Excel-dokument samtidigt men då är funktionerna begränsade. (Lag om offentlighet i myndigheternas verksamhet 21.5.1999/621).

Programmet skall klara upp till 60 simultana användare utan att prestandan försämras. Om programmet inte klarar av belastningen kommer det resultera i att programmet inte används. Det skall vara så gott som underhållsfritt och klara flera tusentals datainmatningar.

1.1 Polisen i Finland

I Finland fanns det år 2021 7 415 poliser i tjänst. I Finland är polisen uppdelad i 11 polisinspektörer som har ansvar för respektive områden (Verksamhet i hela Finland, 16.3.2023). Polisens ansvar och befogenheter är skrivna i lag och vilken polisens uppgift i Finland är står skriven i Polislagens 1 kapitel 1 paragraf (Polislag 872/2011).

Polisens uppgift är att trygga rätts- och samhällsordningen, skydda den nationella säkerheten, upprätthålla allmän ordning och säkerhet samt att förebygga, avslöja och utreda brott och föra brott till åtalsprövning. Polisen ska upprätthålla säkerheten i samarbete med andra myndigheter samt med sammanslutningar och invånarna och sköta det internationella samarbete som hör till dess uppgifter. (26.4.2019/581)

Polisen ska dessutom sköta uppgifter i samband med tillståndsförvaltningen och andra uppgifter som uttryckligen föreskrivs i lag samt inom ramen för sina uppgifter ge var och en den hjälp som han eller hon behöver. Om det finns grundad anledning att anta att en person har försvunnit eller råkat ut för en olycka ska polisen vidta de åtgärder som behövs för att finna personen.

I förundersökningslagen (805/2011) föreskrivs om förundersökning av brott och i tvångsmedelslagen (806/2011) om tvångsmedel i samband med förundersökning av brott. (Polislag 872/2011).

Till polisens grunduppgifter hör att förebygga, avslöja och utreda brott och föra brott till åtalsprövning. När polisen utreder brott sker det genom en förundersökning. En förundersökning skall utföras enligt parametrarna i Förundersökningslagen. År 2021 kom det till polisens kännedom 482 593 strafflagsbrott (Inrikesministeriet, 16.3.2023).

1.1.1 Polisinspektören i Österbotten

Polisen i Österbotten består av landskapen Österbotten, Södra Österbotten och Mellersta Österbotten. Regionen har cirka 443 000 invånare. Polisinspektören är uppdelad i tre områden, Kust-Österbotten, Mellersta Österbotten och Södra Österbotten. Till Kust-Österbottens område hör Kaskö, Korsnäs, Korsholm, Kristinestad, Laihela, Malax, Närpes, Storkyro, Vasa och Vörå. (Polisen, Polisinspektören i Österbotten, 16.3.2023b).

1.1.2 Kust Österbottens brottsstatistik 2021

Enligt Statistikcentralen kom det till en förundersökningsmyndighet (Polis, Tull och Gränsbevakning) kännedom om 8656 strafflagsbrott och av dessa upplades 6418 år 2021. (Statistikcentralen, 16.3.2023). På Kust Österbottens område kom det enligt polisens statistik år 2021 totalt 8643 strafflagsbrott till polisens kännedom (Polisen,

Verksamhetsmiljö och statistik, 16.3.2023c). Av detta går det att uppskatta att av dessa brott upplärdades mellan 6300–6400 brott på Kust Österbottens område.

1.2 Avgränsning

För att systemet inte skall överskrida gränsen mellan användarvänlighet och hur komplicerat det är, kommer det att avgränsas. Det skall endast innehålla väsentlig information för användaren. Systemet skall heller inte användas som ett personregister utan som ett informationssystem för brottsundersökningar. I Personregisterlagens definieras ett personregister enligt följande:

Personregister en datamängd som innehåller personuppgifter och som behandlas med automatisk databehandling samt sådana förteckningar och kartotek, eller en på motsvarande sätt ordnad datamängd, som innehåller personuppgifter och ur vilka information om en bestämd person med lätthet och utan oskäligen kostnader kan erhållas. (Personregisterlag 471/1987).

Vid valet av informationssystem ställdes det krav av Österbottens polisinrättning. Tre av kraven var att systemet skall vara godkänt av IT-administratörerna, programmet skall inte få kosta något och programmet skall vara enkelt att använda. Systemet måste också klara av flera tusen datainmatningar. För att kunna uppfylla dessa krav undersöks först de befintliga programmen som polisinrättningen hade till sitt förfogande. Bland de befintliga programmen fanns Microsoft Access. Microsoft Access är ett databashanteringssystem utvecklat av Microsoft (Microsoft, Databasgrunder, 16.3.2023).

En brottsanmälan innehåller en stor mängd information, som är väsentlig för brottsutredning. Informationen är dock inte lika väsentlig i ett uppföljningsverktyg för brottsanmälningar. Exempelvis är det inte nödvändigt att verktyget innehåller information om hur många gånger en misstänkt slog en målsägande eller hur mycket pengar en person stal från kassan vid ett inbrott. Enligt polisens egen hemsida när en brottsanmälan görs skall bland annat meddelas följande saker, vad som har skett, hur gärning eller försummelsen har skett, var och när gärningen har skett och vem som begått gärningen. (Polisen, Gör brottsanmälan, 16.3.2023a).

Redan i ett tidigt skede av planeringen blev det klart att för att systemet skall klara av flera tusentals datainmatningar måste de vara eller ha en databas, där informationen sparas. MS

Access är som program gjort för att klara av stora datamassor. I Access går det att göra ett grafiskt gränssnitt som gör att användningen av programmet blir enklare och smidigare. MS Access kan också ha flera samtidiga användare, vilket var ett av de ställda kraven. På marknaden finns andra databasprogram som kunde lämpa sig bättre för detta ändamål, men eftersom Access redan fanns bland de befintliga programmen vid Polisinrättningen kommer det att användas.

2 Systemutveckling och metoder

Vid utvecklande av ett informationssystem finns det många olika metodologier. För att välja den rätta metodologin granskas flera metodologier och de mest passande används.

Satzinger, Jackson & Burd (2016) skriver att ett informationssystem är en uppsättning av sammanhängande komponenter som samlar in, bearbetar, lagrar och förser företag med information för att utföra uppgifter. En annan term för informationssystem ett datorprogram. Ett datorprogram är en programvara som utförs på en dator, eller motsvarande apparat, för att utföra specifika funktioner eller funktioner som är relaterade till varandra. (Satzinger, Jackson, & Burd , 2016, p. 4).

2.1 Systemutvecklingslivscykel

Ofta vid systemutveckling används en så kallade systemutvecklingslivscykel, (Systems development life cycle, SDLC) som är en process där man försöker förstå hur ett informationssystem kan stöda företagets behov genom att designa ett system, utveckla det och leverera det till kunden. (Dennis , Wixom, & Tegarden, 2015, p. 1). Schwalbe (2011) skriver att systemutvecklingslivscykeln är stommen för utveckling och underhåll av informationssystem (Schwalbe , 2011, p. 60).

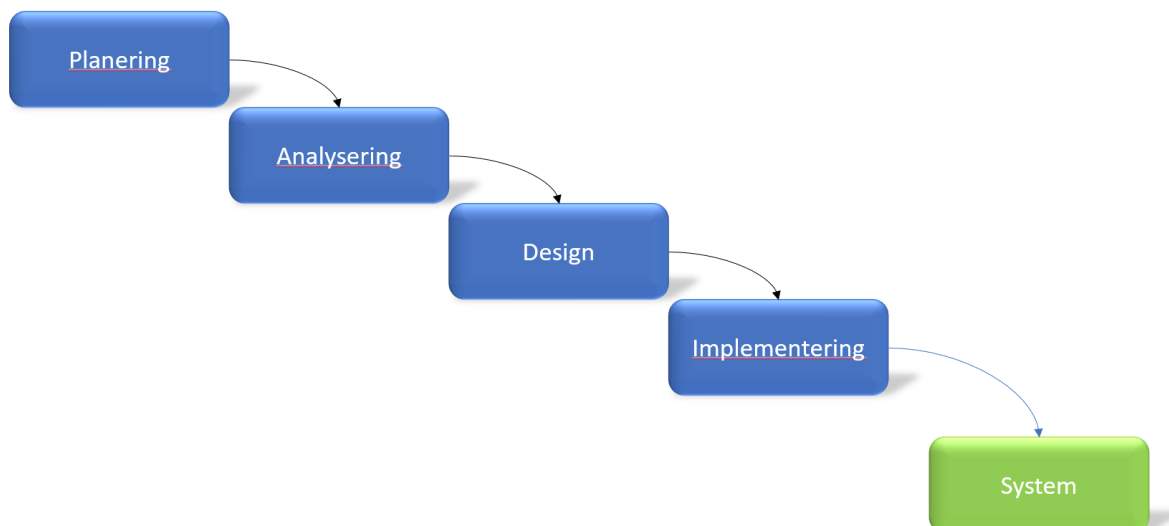
På många sätt är systemutvecklingslivscykeln jämförbara med att bygga ett hus. Det inleds med en idé som följs upp av en enkel ritning som tillsammans med kunden utformas enligt kundens önskemål för att till slut bilda en slutlig ritning. Efter det görs det en detaljritning över huset och efter det inleds byggandet av huset. Systemutvecklingslivscykeln använder för de olika stegen termerna planering, analysering, design och utförande. (Dennis , Wixom, & Tegarden, 2015, p. 2). Den förutsägbara utvecklingen är enligt Satzinger, Jackson & Burd (2016) när ett utvecklingsprojekt kan bli planerat och organiserat och det nya informationssystemet kan bli utvecklat enligt planen. Detta sätt är användbart för system som är klart uppfattade och definierade. Den innehåller sex skilda grupperingar, påbörjande av projektet, planeringen, analyseringen, design, implementering och utförande. (Satzinger, Jackson, & Burd , 2016, p. 298).

Schwalbe (2011) skriver att inom systemutvecklingslivscykeln används det flera olika sorters metoder och dessa är bland annat Vattenfallsmetoden, Rapid Application

Development (RAD) och flexibel utveckling. De två förstnämnda är exempel på metoder som är del av en förutsägbar livscykel, som betyder att projektets mål kan formuleras klart och tydligt och tidtabellen och kostnaderna kan räknas noggrant. Flexibel utveckling hör till en anpassningsbar utveckling som används när kraven inte klart och tydligt kan påvisas i ett tidigt skede av projektet. Den anpassningsbara utvecklingen ger rum för mera frihet än den förutsägbara utvecklingen. (Schwalbe , 2011, pp. 60-61). Satzinger, Jackson & Burd (2016) skriver att i den anpassningsbara utvecklingen används det i stället för att systematisk gå från sekvens till sekvens, av upprepningar som i form av flera mindre projekt fokuserar på mindre delar av applikationen och när ett mindre projekt är färdigt flyttar man vidare till följande mindre projekt. I varje projekt följer man samma sekvenser analysering, design, implementation och testning. Genom användning av upprepningar kan projektet anpassa sig till förändringar under projektets gång. (Satzinger, Jackson, & Burd , 2016, p. 299).

2.1.1 Vattenfallsmetoden

Vattenfallsmetoden är en av de först metoderna inom systemutveckling som började användas och används än idag. Den går ut på att systematiskt utföra varje sekvens efter varandra. Först när en sekvens är färdig, dokumenterad och godkänd går man vidare till följande sekvens. Efter att man flyttat vidare till följande sekvens går man inte tillbaka till föregående sekvens. För att vattenfallsmetoden skall fungera är planeringsskedet ytterst viktigt och tidskrävande. Det negativa med metoden är att implementeringen kommer i ett väldigt sent skede av utvecklingen och kunden kommer inte att se en fungerande version innan det. (Filipova & Vilão, 2018, pp. 27-28).



Figur 1. Vattenfallsmetoden.

Det positiva med vattenfallsmetoden är att den redan i ett tidigt skede identifierar systemkrav, redan innan programmeringen har inletts och minimerar förändring av kraven samtidigt som projektet fortskrider. Negativa är att designen måste vara fullständigt färdig innan programmeringen kan börja och det tar långt tid mellan analyseringssekvensen och det färdiga systemet. (Dennis , Wixom, & Tegarden, 2015, p. 7). Det har vidareutvecklats modeller av vattenfallsmodellen som möjliggör att man går tillbaka till en tidigare sekvens (Zaigham, 2017, p. 369).

2.1.2 Rapid Application Development

Rapid Application Development kan översättas till snabb applikationsutveckling. Det är en metod som kombinerar analysering- och designsekvenser genom användning av prototyper och användning av applikationslogik i process specifikationen. (Langer, 2016, p. 397). Metoden utvecklades som svar till de besvärliga och icke flexibla vattenfallsmetoderna, men används inte mera som en fristående metod utan det har utvecklats andra flexibla metoder. (Zaigham, 2017, p. 369). Schmidt skriver att i den snabba applikationsutvecklings metoden utvecklar och planerar man mjukvaruprodukten under tiden man programmerar produkten och således har man möjlighet att snabbare ändra kraven som ställs på produkten. (Schmidt, 2013, p. 104). Enligt Dennis, Wixom & Tegarden (2015) försöker man snabbt få en fungerande version till användarna så att de kan bättre

förstå systemet och de kan ge respons så att systemet blir så som man vill ha det. ((Dennis , Wixom, & Tegarden, 2015)

McConnell skriver att när man använder sig av en snabb utvecklingsmetod finns de några klara misstag som man skall undvika att göra, dessa är bland annat orealistisk förväntan, brist av användarfeedback, över ambitiös tidsplanering och otillräcklig planering. Av dessa hör de två första till motivations fasen och de två andra till process fasen. Motivationsfasen innebär vilken motivation som personerna som håller på att utveckla produkten har. Om personerna inte har tillräcklig motivation kommer det inte gå att utnyttja sig av den snabba utvecklingsmetoden och motivationen är en av de viktigaste aspekterna i den snabba utvecklingen. I processfasen inom mjukvaruutveckling ingår både lednings och tekniska metodologier. De går lättare att räkna ut hur processen påverkar utvecklingstidtabellen än hur personer påverkar den. (McConnell, 1996, pp. 14, 42, 44).

2.1.3 Flexibel utveckling

Flexibel utveckling är en metodologi inom systemutvecklingslivscykeln som är den modernaste av de tre presenterad. Schwalbe definierar flexibel utveckling som "en metod för mjukvaruutveckling som utnyttjar nya tillvägagångssätt, fokuserar på nära samarbete mellan programmerarna och företagsexperterna." (Schwalbe , 2011, p. 74). Enligt Dennis et. al, (2015) finns det i flexibel utveckling tolv principer. Utifrån dessa tolv principer fokuserar metodologin på att effektivera systemutvecklings processen genom att eliminera tiden som det tar för att designa och dokumentera. (Dennis , Wixom, & Tegarden, 2015, p. 12). Den flexibla utvecklingen kan delas upp i 12 punkter:

- Mjukvaran levereras tidigt och kontinuerligt under utvecklingsprocessen, för att göra kunden nöjd
- Kravsförändringar tas emot oberoende i vilket skede av utvecklingen de kommer
- Ett fungerande system levereras kontinuerligt till kunden
- Kunder och utvecklare arbetar tillsammans för att lösa verksamhetsproblem
- Motiverade individer kommer på lösningar, ge dem verktygen och omgivningen de behöver, och de kommer att leverera
- Kommunicera i person med utvecklingsteamet är den effektivaste och smidigaste metoden för att klarställa kraven
- Det primära sättet att mäta framsteg är genom att arbeta, utföra programmering
- Både kunden och utvecklare skall arbeta i en takt som är hållbar. Mängden arbete skall bibehållas utan att någon blir utbränd
- Flexibiliteten ökar genom att fokusera på teknisk excellens och design
- Det är viktigt att hålla saker enkelt, för att undvika onödigt arbete.
- Självorganiserade team utvecklar den bästa arkitekturen, kraven och designen
- Utvecklingsteam reflekterar regelbundet över hur de skall förbättra utvecklingsprocessen

(Dennis , Wixom, & Tegarden, 2015, p. 12)

Enligt Northwood (2018) och Filipova & Vilão (2018) finns det ett manifest som fungerar som grunden för den flexibla utvecklingen. (Northwood , 2018, p. 12). (Filipova & Vilão, 2018, pp. 28-29). I *Agile manifesto* finns det fyra grundprinciper som representerar grunden till den flexibla utvecklingen. Dessa är att värdesätta att anpassa sig till förändring framöver att följa en plan, värdesätta individer och växelverkan över processer och verktyg, värdera att arbeta på programmering framöver omfattande dokumentering och värdesätta samarbete med kunden framöver kontraktsförhandling. (Beck, et al., 2001)

3 Full Stack-utveckling

Vid utveckling av en applikation eller en webbsida används ofta en så kallad Full Stack-utveckling. Denna utveckling baserar sig på att man delar utvecklingen av en applikation i två delar, ett gränssnitt och en serverdel. Gränssnitt kallas inom full stack-utveckling för front end och omfattar det som är synligt för användaren av applikationen. Serverdelen kallas för back end och den delen ansvarar bland annat för datainnehållet och för att utföra begäran som kommer från användaren av applikationen.

Det finns många orsaker varför man delar upp en applikation i två delar. Dessa är bland annat att för att programmeraren inte skall behöva hantera för många programmeringsspråk, varje programmeringsspråk har styrkor och svagheter, det är lättare att underhålla delarna när det är delade och det har bättre skydd mot hackers. Vid implementeringsskedet är det ibland lättare om man inte har applikationen delad eftersom det är lättare att testa funktionerna, optimeringen och skalbarheten i applikationen. (McCarron, 11.4.2023).

3.1 Gränssnitt

Ett gränssnitt är det som slutanvändaren kommer att använda. Det skall ge en positiv upplevelse så att användaren kommer att vilja använda det dagligen. Webbsidor är typexempel där det grafiska gränssnittet spelar stor roll. Webbsidor skall vara attraktiva och utstråla tillit som gör att besökaren kommer att besöka sidan på nytt. (Xanthidis, Nicholas, & Argyrides, 2010, p. 120).

3.1.1 Verkställande informationssystem

Ett grafiskt gränssnitt som används inom den privata sektorn är ett så kallat verkställande informationssystem, Executive informationssystem EIS, som är ämnat för företagsledningen. Informationen i systemet skall var så tillräcklig att företagsledningen kan göra beslut utgående från det. Gary P. Moynihan (2011) skriver om funktionaliteten att det skall vara bland annat, ett lätt att använda och enkelt att underhålla grafiskt användargränssnitt, integrerade möjligheter för data åtkomst, analysering och kontroll, sofistikerade verktyg för systemnavigering, funktioner för dataanalys och skildring av kritiska faktorer för framgång för organisationen. Verktuget skall vara lätt att använda,

hanterbart och innehålla de väsentliga delar för analysering och kontroll. (Moynihan, 2011, p. 235).

Med systemet kan företagsledningen granska specifika delar av företaget och utifrån informationen göra beräknade beslut. Genom att välja ut specifika delar av systemet kan det också utnyttjas i detta system. I de officiella systemen kan all data finnas men i det utvecklade systemet begränsas data till endast den nödvändigaste.

3.1.2 Polisens hemsida

Polisen uppdaterade sin hemsida poliisi.fi till 2022 (figur 3). Den nya hemsidan präglas med polisens kännsypaka färger vitt och blått. *Kumar och Sareen (2010)* skriver att en hemsida skall kunna identifieras med den organisation som den tillhör. En hemsida skall innehålla för organisationen kännsypaka logon, färger, layout etcetera. (Kumar & Sareen, 2010, p. 1199).

Programmet kommer inte att vara en hemsida och kommer inte att fungera på internet men eftersom det grafiska användargränssnittet för organisationen är så kännsypakt kommer samma färger och typsnitt att användas. Också placeringen av flikar och den grafiska designen kommer att utnyttjas.



Figur 2. Polisens hemsida poliisi.fi.

Färgschemat (Figur 3) som används i den blåa färgen är, enligt hexadecimal numreringsystemet, #06377b och vita färgen är #fff. Hexadecimal numreringsystemet är ett 16-bas system som kräver endast sex karaktärer för att bestämma en RGB (Röd, Grön, Blå) färg. På RGB skalan baserar sig på hur mycket av vilken grund färg (röd, grön eller blå) används. Skalan är mellan 0–255 för varje enskild färg. Exempelvis en färg som innehåller enbart röd färg har RGB kombinationen 255,0,0. (Robbins , 2006, p. 356). Typsnittet som används är Roboto som är designat av Christian Robertson för Google, togs i bruk 2011 och är gränssnittet för Android-plattformen. (frwiki, 16.3.2023).

3.2 Serverdel

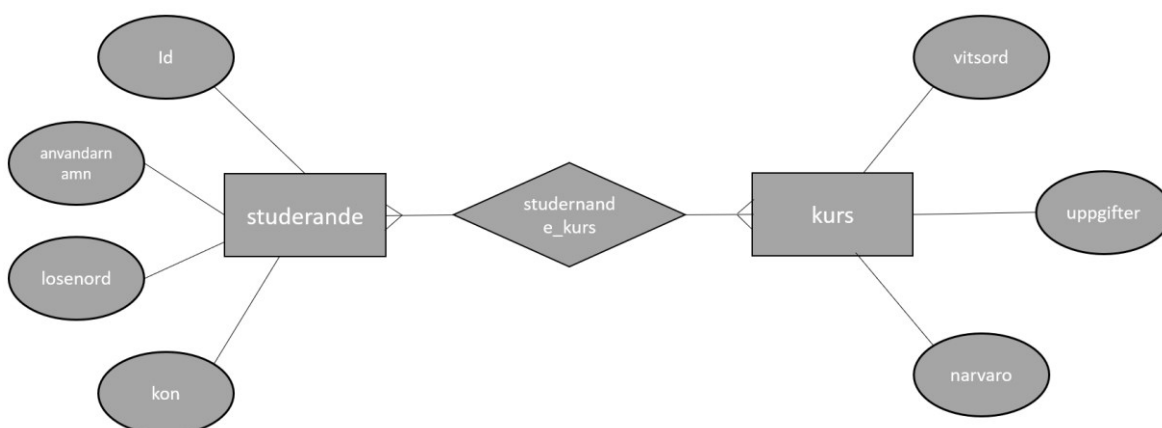
Serverdelen är platsen där väsentlig information kommer att sparas i en databas. Enligt Zdrożny, et al. (2009) är en databas en samling av beständig data. I en databas är data format enligt en databasmodell. Modellen definierar hur data är strukturerat, hur integriteten och säkerheten är begränsad och hur data beter sig. (Zdrożny, de Tré, de Caluwe, & Kacprzyk, 2009, p. 155). Eftersom systemet skall klara av att spara över 1000 datainmatningar måste allting sparas på ett sätt så att det är lättåtkomligt. En databas är enligt Lutkevich (2023) information som är enkel att komma åt, hantera och uppdatera. Han menar också att databaser som finns på datorer är ofta sammanställningar av datainmatningar eller filer som innehåller information, dessa är exempelvis kunddata, transaktioner och produktinformation. (Lutkevich, 2023).

När en databas skall tas i bruk måste man först beakta faktorer som påverkar lagringskapaciteten. Dessa är platsen där data skall förvaras, hur länge data skall lagras, återställning av borttagen data, återställning av utrymme efter borttagen data och periodiskt krav att mata in ny data. Faktorerna påverkar lagringen som kommer att påverka programvaran och databashanteringssystemet och påverkar direkt konfigurationen av datasystemet. (Schmidt, 2013, p. 197).

När en databas utvecklas skall man utgå från en konceptuell beskrivning som utvecklar ett detaljerat databasschema som kan direkt implementeras med ett databashanteringssystem. Schemat innehåller utveckling av tabeller, nyckel- och indexidentifiering, attributtyper och andra beslut, som effektiverar processen. Dessa görs i det här skedet. Utvecklingen kan var en invecklad och avancerad process. Först utformas

den huvudsakliga strukturen av systemet genom att identifiera delsystem och kontakten till andra system. Inom varje delsystem görs det sedan ett grafiskt användargränssnitt, affärslogik och databasåtkomst. (Satzinger, Jackson, & Burd , 2016, p. 22).

För att förverkliga databasschema använder man sig ofta av en relationsdatabas. Enligt Filipova och Vilão (2018) är en relationsdatabas det som oftast används och fördelarna med den är att den har strukturerad data som klargör att det inte uppstår några duplikationer, begränsar data som gör att det förekommer mindre felaktig data, relationer mellan data kan ha begränsningar. Begränsningarna gör att det är säkert att utforma relationer och säkerställa att de faktiskt är möjliga. Nackdelar med en relationsdatabas är att data normaliseras vilket gör att man ibland för att samla enkel information måste göra komplexa frågor och kopplingar vilka begränsar prestandan och det går inte heller att göra komplexa objekt eftersom de främst fungerar med enkla attributtyper som numror, strängar, datum etcetera. (Filipova & Vilão, 2018, p. 109).



Figur 3. Ett databasschema.

Ett databasschema (Figur 4) över en relationsdatabas för en kurs i en skola. Relationen mellan de skilda tabellerna är planerad så att en studerande kan delta i många kurser och för att det skall gå måste det skapas en skild tabell mellan "studerande" och "kurs" tabellerna en "studerande_kurs" tabell som innehåller information för hur många kurser en och samma studerande har tagit. Inom tabellerna "studerande" och "kurs" finns det primärnycklar som är unika för varje tabell och upptar en rad i tabellerna. Det är primärnycklarna som identifierar en tabell. Dessa underlättar att identifiera exempelvis vilken studerande har vilket användarnamn, lösenord och kön. Attributtyperna varierar beroende på innehållet i raden, exempelvis är ofta primärnyckeln ett nummer och ett användarnamn är en sträng som innehåller text.

4 Microsoft Access

Microsoft Access är ett databasprogram som har utvecklats av Microsoft. Den första versionen av Access kom redan 1992 och efter det har det kommit många versioner och den nyaste är Access 2019. Microsoft Access kombinerar Microsofts relations Jet Database Engine (sv. databasmotor) med ett mjukvaruutvecklingsprogram och ett gränssnitt. (Terra, 17.2.2023)

Microsoft marknadsför Microsoft Access som ett databasprogram som man kan skapa och dela appar utan att vara en utvecklare. Med Access kan du ansluta till flera datakällor och integrera data mellan Access och andra verksamhetsspecifika applikationer. Det går enkelt att lagra data till exempelvis en SQL-server. (Microsoft, Utnyttja dina data, 10.4.2023f).

I Access finns det flera olika delar som är gjorda för att man skall kunna bygga en applikation med en databas och ett gränssnitt. Dessa är tabeller, formulär, rapporter, frågor, makron och moduler. En funktion som också finns i Access är att man kan dela upp applikationen så att man skiljer databasen och gränssnittet. Detta kallas i Access att göra en delad databas. På det sättet får man samma kontroll som med en databas på en server och man behöver inte vara orolig att någon i misstag raderar databasen. Gränssnittet är ett skilt Access program som kan skickas exempelvis via email eller som finns på en server som man kan ladda ner till sin egen dator. Kontakten mellan gränssnittet och databasen är uppgjord i samband med uppdelningen. (Microsoft, Split an Access database, 11.4.2023)

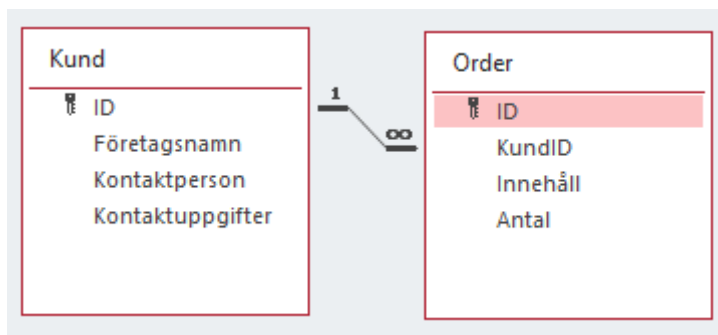
4.1 Tabeller

Den kanske mest väsentliga funktionen i en relationsdatabas är platsen där data kommer att sparas och i Access heter det tabeller. I en tabell kan man spara data i rader och kolumner. När man börjar utforma en databas i Access börjar man med att skapa tabellerna och först efter det de övriga objekten. En tabell delas upp i två delar, rader och kolumner. I en rad sparas exempelvis information om en kund så som företagets namn, kontaktperson kontaktuppgifter etcetera. En kolumn innehåller information om tabellens ämne, exempelvis alla företagsnamn som finns inmatade. En rad går i horisontell riktning och en kolumn i vertikal. (Microsoft, Introduktion till tabeller, 10.4.2023e)

ID	Företagsnamn	Kontaktperson	Kontaktuppgifter
1	Företag 1	Kontaktperson 1	Kontakuppgift 1
2	Företag 2	Kontaktperson 2	Kontakuppgift 2
3	Företag 3	Kontaktperson 3	Kontakuppgift 3
4	Företag 4	Kontaktperson 4	Kontakuppgift 4

Figur 4. Exempel på en databastabell.

I en databas lagras data i fält och varje fält har en attributtyp och med dessa typer kan man styra vilken typ av data som lagras i fältet. I figur 5 är den data som är sparad i ID-kolumnen ett automatiskt nummer och de övriga kolumnerna är sparade som text. Raden automatiskt nummer är hur databasen identifierar varje rad i tabellen och är oftast obligatoriskt för att databasen skall hållas klar och hanterbar. Det automatiska numret fungerar ofta som primärnyckel som krävs när man börja bygga relationer mellan tabeller. En tabell kan endast ha en primär nyckel. En tabell kan innehålla en sekundär nyckel som motsvarar en annan tabells primär nyckel. Genom det går det att utforma en relation mellan två tabeller. Detta görs för att exempelvis hantera vilken kund har beställt vad. Relationen mellan tabellerna binder kunden till beställningen. Den primära nyckeln identifieras i Access med en nyckelikon innan namnet. (Microsoft, Introduktion till tabeller, 10.4.2023e)



Figur 5. Exempel på en relation mellan två tabeller.

4.2 Formulär och rapporter

Gränssnittet som används inom Access kallas för formulär. I formuläret går det att modifiera hur man vill att data skall presenteras från exempelvis en tabell. Det går att göra ett formulär som inte är bunden till någon datakälla utan det kan ha funktioner som kan bidra till att man kan ha kommandoknappar eller andra kontroller som styr till olika formulär. Genom att göra ett formulär som är bundet till en datakälla kan man styra vad som presenteras i formuläret. Det går att välja vilka rader från en tabell som är synliga i

formuläret och på så vis göra det enklare att förstå innehållet. (Microsoft, Introduktion till formulär, 10.4.2023a).

Ett formulär kan innehålla många olika funktioner. Det kan vara en kommandoknapp, som har en inprogrammerad funktion som aktiveras vid tryck. En kryssruta, som vid tryck kryssas för eller ett underformulär, som betyder att man kan lägga till ett formulär i ett formulär. Utöver det kan man dela ett formulär så att på en del finns ett kalkylblad och på den andra delen finns funktionerna i formuläret. (Microsoft, Introduktion till formulär, 10.4.2023a).

Men hjälp av rapporterings delen i Access går det att göra rapporter som visar diverse data sammanfattningar exempelvis hur mycket som har sålts av en viss produkt vid ett visst datum. Rapporterings delens utseende går att modifiera enligt egna målsättningar. Det går också att göra ett formulär som fungerar som styrmodul för rapporteringen. Det går exempelvis att lägga in en kombinationsruta med vilken man kan bestämma mellan vilka datum man vill se sålda produkter. (Microsoft, Introduktion till rapporter i Access, 10.4.2023d).

4.3 Frågor, makron och moduler

De mest avancerade delarna av en Access-applikation är frågor, makron och moduler. Med frågor kan man kombinera flera relaterade tabeller tillsammans. Med frågor kan man specificera det som man vill att skall visas i ett formulär. Om man till exempel vill granska hur mycket av en produkt har sålts till ett visst område i en stad. Med frågor kan man också göra beräkningar baserade på data från tabeller och det går att lägga till eller uppdatera data från en bunden fil eller databas. (Microsoft, Introduktion till frågor, 10.4.2023b).

Makro och modul delen av Access använder sig av Microsoft egna programmeringsspråk VBA-kod. Genom att programmera kan man göra mångfacetterade funktioner som kan underlätta exempelvis underhåll av databasen. När en funktion läggs till på en kommandoknapp kan man använda sig av VBA-kod eller utnyttja den visuella programmeringshjälpen. (Microsoft, Introduktion till programmering i Access, 10.4.2023c).

5 Utförande

Inledningsvis granskades det givna uppdraget. Verktuget skall bland annat vara lätt att använda, kunna hantera en stor massa information och programmet som används ska vara ett av de befintliga programmen som finns till förfogande. Det skall klara av upp till 60 samtidiga användare. Den första versionen av programmet kommer att vara på finska men efter det kommer det troligtvis utvecklas en version på svenska.

5.1 Systemutvecklingen

Redan i ett tidigt skede av systemutvecklingen blev det klart att användningen av vattenfallsmetoden som metod kommer vara en utmaning. Eftersom uppgiftsgivaren gav mycket frihet till utvecklaren var det näst intill omöjligt att använda sig av vattenfallsmetoden. Detta eftersom den kräver redan i ett initialt skede att allting planeras in i minsta detalj och att efter att man flyttat till följande sekvens inte längre kan gå tillbaka till föregående sekvens.

Då återstår den snabba applikationsutvecklingsmetoden och den flexibla utvecklingsmetoden. Den snabba applikationsutvecklingsmetoden kan anses som ändamålsenlig eftersom produkten utvecklas under tiden den programmeras och därför kan man snabbare reagera på om det ställs nya krav (se kapitel 2.1.2). Den flexibla utvecklingen handlar om att man skall följa fyra grundprinciper, värdesätta förändring framöver att följa en plan, värdesätta individer och växelverkan över processer och verktyg, värdera programmering framöver dokumentering och värdesätta kunden framöver kontraktsförhandlingar (se kapitel 2.1.3). Eftersom utvecklingen är på småskalig nivå och planeringsmöten är få kommer det inte att ske så många förändringar. Förändringarna som kommer att göras sker först i det skedet som programmet kommer i användning. Det är främst utvecklarens motivation som styr när programmet blir färdigt eftersom uppdragsgivaren inte gav någon specifik deadline.

Utvecklingen av produkten utförs enbart av en person. Uppdragsgivaren ställde några krav som måste uppfyllas och utöver det hölls några möten under utvecklingens gång. Uppdragsgivaren hade inte heller någon klar vision hur verktuget skulle se ut vilket gav utvecklaren möjligheten att utveckla något som ser bra ut visuellt och något som är väl konstruerat. Eftersom utvecklingen av produkten utförs av en person kommer den snabba

utvecklingsmetoden vara den bästa. Utvecklaren har flera års erfarenhet inom området som verktyget skall utvecklas för. Kraven som uppgiftsgivaren ställde begränsar inte heller användningen av metoden. Eftersom skribenten arbetade på sektorn där verktyget behövdes fanns det en personlig motivation att utveckla ett verktyg och därav vara valet av den snabba utvecklingsmetoden ett bra alternativ.

5.1.1 Påbörjande av projektet

Inom systemutveckling finns de skilda faser och den första av dessa faser är påbörjande av projektet. Projektet påbörjades egentligen vid en kaffebordsdiskussion där det framkom att det skulle vara bra att få ett uppföljningsverktyg som alla använder och som är användarvänligt. De befintliga programmen som användes följde en viss struktur men skilde sig från användare till användare. Det att flera personer kunde använda programmet samtidigt sågs som en positiv egenskap.

Vid möte med uppdragsgivaren framkom det några krav som måste uppfyllas gällande det utvecklade verktyget. Detta ställde högre krav på utvecklaren eftersom verktyget måste uppfylla dessa krav men att det samtidigt inte fanns någon klar design hur uppdragsgivaren ville att verktyget skulle se ut.

Kraven var klara, systemet skall vara godkänt av IT-administratörerna, programmet får inte kosta något, programmet skall vara enkelt att använda och det skall klara av flera tusen datainmatningar. Någon specifik deadline för när programmet skulle vara färdigt gavs inte men det skulle gärna vara färdigt tills våren 2023.

5.1.2 Planeringen

Det första som gjordes i planeringsskedet vara att de befintliga programmen granskades och deras möjligheter och begränsningar kartlades. För att användningen skall vara så enkel som möjligt måste det vara ett befintligt program som används nästan dagligen eller ett program som är lätt att använda.

I ett tidigt skede framkom det under möten med uppdragsgivaren att programmet måste ha en databas som sparar all data för att klara av den relativt stora datamassan. Eftersom databaser ofta inte har något enkelt gränssnitt måste det utvecklas ett sådant. För att det skall gå måste det användas ett annat program eller ett program som innehåller båda

funktionerna. Av de befintliga programmet är Microsoft Access det mest lämpade. I Microsoft Access finns det en databas och det går att göra ett enkelt gränssnitt som uppfyller kraven.

I det här skedet började det planeras vilken information som är väsentlig för användaren av verktyget. Vilken information som programmet måste innehålla för att det skall vara så effektivt som möjligt. I figur 7 presenteras den första planeringen av vad som verktyget skall innehålla.

- Ärendenummer.
- Vem som ansvarar för ärendet.
- Information om ärendet
 - Rubricering.
 - När den misstänkta gärningen inträffat.
 - När den misstänkta gräningen har anmälts.
- I vilket skede är undersökningen
 - Ex. färdig, väntar, icke påbörjad, på hälft.
- Dagar sedan händelsen samt dagar sedan anmälts.
- När det misstänkta brottet skriberas.
- Utförda åtgärder.
- Utförda förhör.

Figur 6. Inledande planeringspunkter.

5.1.3 Analyseringen

Inledningsvis av analyseringskedet definieras kraven som krävs av programmet. Kraven skrivs ofta upp på en lista som sedan följs. Kravformuleringen delas upp i två skilda delar, icke funktionella och funktionella krav. De icke funktionella kraven är funktioner så som prestanda och användbarhet och de funktionella kraven är hurdan process som de skall utföra eller hurdan information som programmet skall upprätthålla. (Dennis , Wixom, & Tegarden, 2015, p. 87).

Icke funktionella krav:

1. Operativa krav
 - Systemet skall fungera på arbetsdatorerna.
 - Systemet skall ha kontakt med internserver.
 - Systemet skall kunna göra en säkerhetskopiering
2. Prestanda krav
 - Systemet skall uppdateras direkt efter inmatad data.
 - Systemets funktioner skall fungera efter inmatad/ändrad data.
 - Systemet skall spara befintlig session
3. Säkerhetskrav
 - Systemet skall vara krypterat så att inte obehöriga kan använda det.
 - Databasen skall vara krypterad så att informationen inte går att raderas.
 - Endast inmatade personer i systemet kan tilldelas uppgifter.
 - Ett anmälningnummer får endast finnas, inga duplikationer.

Funktionella krav:

1. Modifiering av data
 - Användaren skall kunna mata in nya data, ändra data och radera data.
 - Filtrering av information.
 - Systemet ändras enligt inmatad data
2. Rapportering
 - Användaren skall kunna köra en rapport över varje ärende.
3. Modifiering av gränssnitt
 - Användaren skall kunna modifiera gränssnittet enligt ramen för programmet.

Figur 7. Icke funktionella och funktionella krav.

Eftersom utvecklaren arbetar inom området som verktyget utvecklas för hade utvecklaren en inblick över vad som krävs av systemet. Det hölls möten med uppdragsgivaren var kraven diskuterades och klargjordes. Ofta vid analyseringsskedet brukar man intervjua de blivande användarna eller använda sig av någon form av frågeformulär som användarna svara på. Men eftersom utvecklaren hade tidigare uppföljningsverktyg till förfogande och egen erfarenhet ansågs det inte nödvändigt att intervjua eller skicka ut frågeformulär. (Satzinger, Jackson, & Burd , 2016, pp. 50,54).

Kraven som uppdragsgivaren ställde gav en del frihet till utvecklaren. Det konstaterades i ett tidigt skede av analyseringen att de mest tidskrävande kommer att vara programmeringen av programmet. När det finns en färdigt system kan det vid behov skickas ut frågeformulär till användarna och utifrån svaren kan systemet förbättras. Detta är en metod som används inom den snabba applikationsutvecklingen (se kapitel 2.1.2).

5.1.4 Design

Designskede i systemutvecklingscykeln är när man bestämmer hur system kommer att fungera med tanke på hårdvaran, mjukvaran och nätverkets infrastruktur. Det delas upp i fyra punkter, Design strategin, som följs av det arkitektoniska och gränssnitts designen. Den arkitektoniska designen ingår hårdvaran, mjukvaran och nätverkets infrastruktur och hur det kommer att fungera i systemet. Gränssnitts designen fokuserar på den visuella delen och hur det kommer att se ut för användaren och hur man manövrerar sig igenom

systemet. Den tredje punkten är databasen och specificeringen av filer som kommer att finnas i systemet. Den fjärde punkten är designen av programmet vilket innebär att man gör upp en plan över programmen och vad vilket program skall göra. (Dennis , Wixom, & Tegarden, 2015, p. 4).

Den arkitektoniska designen var från en uppdragsgivarens sida begränsad och den kunde inte modifieras eller ändras. Det som kunde undersökas var nätverkets infrastruktur. Eftersom uppdragsgivaren hade Access som ett program fanns det andra Microsoft Office program som kunde användas. I Access finns det en möjlighet att använda ett Access program som databasprogram eller så går det att använda Microsoft SharePoint listor som databas.

5.1.5 Implementering

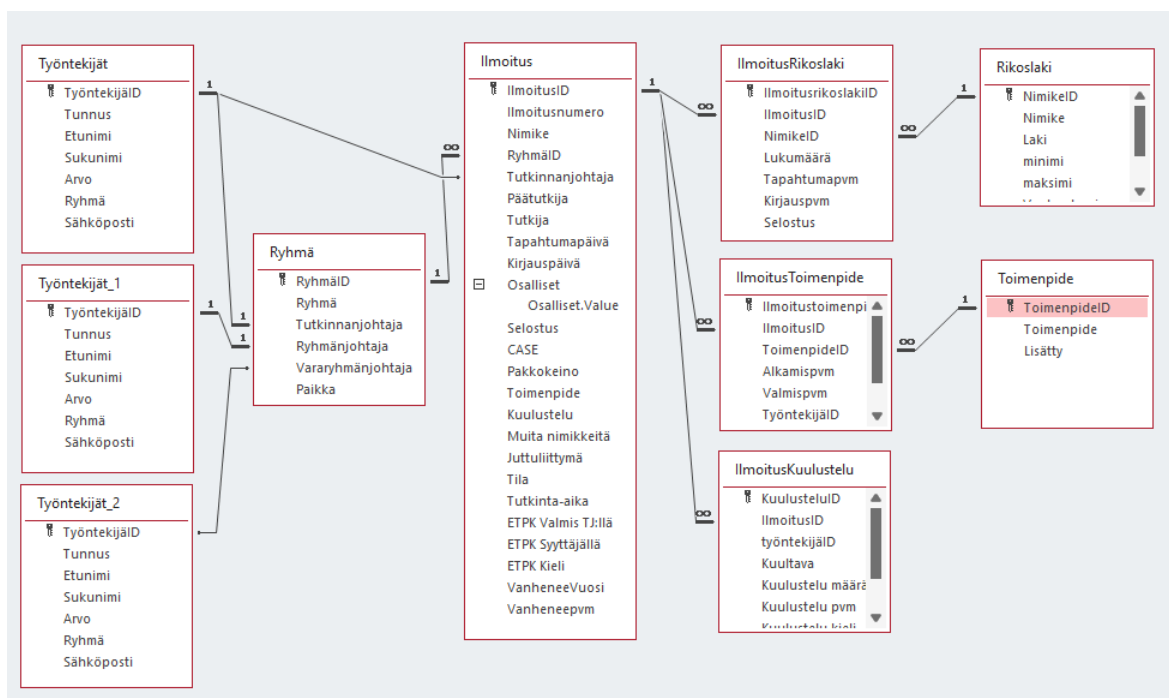
Inledningsvis i implementeringsskedet gällde det att konstruera tabellerna, relationerna mellan dem och göra upp ett databasschema hur databasen kommer att se ut. Det inleddes med tabellen som innehåller information om själva ärendet, vilken brottsrubricering, datum för händelsen, ansvarsperson och när det anmäldes. Den färdiga tabellen ändrade från planeringskedet till implementeringskedet.

	Fältnamn	Datotyp
	IlmoitusID	Räknare
	Ilmoitusnumero	Kort text
	Nimike	Tal
	RyhmäID	Tal
	TutkinnanjohtajaID	Tal
	Päätutkija	Tal
	Tutkija	Tal
	Tapahtumapäivä	Datum/tid
	Kirjauspäivä	Datum/tid
	Osalliset	Kort text
	Selostus	Lång text
	CASE	Kort text
	Toimenpide	Ja/Nej
	Kuulustelu	Ja/Nej
	Muita nimikkeitä	Ja/Nej
	Juttuliittymä	Ja/Nej
	Tila	Kort text
	ETPK Valmis TJIllä	Ja/Nej
	ETPK Syyttäjällä	Ja/Nej
	ETPK Kieli	Kort text

Figur 8. Ärendetabell.

I ärendetabellen (Figur 9) visas den första tabellen som gjordes. Tabellen kommer att fungera som grunden för hela programmet. Den primära nyckel i tabellen är tilldelad till raden "IlmoitusID" och dess datatyp är räknare. Typen räknare tilldelar varje ny rad ett eget nummer och fungerar som identifikationsnummer och detta görs för att inte korrumpiera databasen. Följande rad "Ilmoitusnumero" innehåller det själva ärendenumret. Ärendenumret är unikt för varje ärende och kunde ha fungerat som primärnyckel för tabellen men för enkelhetens skull tilldelar programmet numret. För raden "Ilmoitusnumero" går det att ställa in att informationen som matas in inte kan var duplikation med redan inmatad information.

De rader som har sista bokstäverna "ID" fungerar som så kallade främmande nycklar och har primär nyckel i en annan tabell. Det betyder att informationen för det fältet egentligen kommer från en annan tabell och mellan tabellerna bildas det relationer. Relationernas typer beror på vilken funktion de har i tabellen. De flesta relationer i databasschemat (Figur 10) är bundna till "IlmoitusID" raden för att säkerställa att informationen i tabellerna hålls bundet till rätt ärende.



Figur 9. Databasschema.

Mellan tabellerna finns linjer som är antingen försedda med ingenting, en 1a eller ett evighetstecken. Tecken påvisar vilken relation som tabellerna har mellan varandra, exempelvis har tabellerna Ilmoitus och IlmoitusRikoslaki en relation som på Ilmoitus

tabellens sida har en 1a och på IlmoitusRikoslaki ett evighetstecken. Detta betyder att ett IlmoitusID kan innehålla många IlmoitusRikoslakiID, som betyder att ett ärende kan ha många brottsrubriceringar. Från IlmoitusRikoslaki går det en likadan relation till Rikoslaki tabellen och den är likadan som relationen mellan Ilmoitus och IlmoitusRikoslaki. Tabellerna är utformade på det här sättet av databastekniska skäl. Tanken är att ett ärende kan innehålla många brottsrubriceringar. Brottsrubriceringarna finns i Rikoslaki tabellen. Tabellen IlmoitusRikoslaki kallas för en kopplingstabell. IlmoitusToimenpide och Toimenpide tabellerna fungera på samma vis.

På den andra sidan av tabellen Ilmoitus finns arbetarnas och gruppens tabell. Varje undersökare som är inmatad i system måste höra till en grupp och varje grupp måste ha någon som leder gruppen. Utöver det måste varje ärende höra till en grupp, en undersökningsledare och en huvudundersökare. Det är möjligt att lägga till en andra undersökare till varje ärende.

Efter att databasschemat (Figur 10) var färdigt utvecklades gränssnittet. Access hade ett inbyggt uppdelat databasformulär där två formulär kombineras till ett. Den första delen var den visuella delen och den andra en kalkylbladsdel. I den visuella delen kombinerades Ilmoitus tabellens alla delar på ett visuellt klart och tydligt sätt. Där läggs till fält som räknar ut hur många dagar som det har gått sedan det misstänkta brottet inträffade och när det anmäldes. Det läggs också till en funktion som räknar ut när brottet preskriberas. Den andra delen av programmet fungera som ett kalkylblad där alla ärende är sparade i ett kalkylblad.

Under tiden som programmet utvecklades följdes den snabba utvecklingsfasen genom att först programmera en del, sedan testa det, programmera följande del och testa. Sedan slogs delarna ihop och testades och till slut bildades det ett fungerande system som levererades till uppdragsgivaren.

5.2 Full stack-utvecklingen

Samtidigt som systemutvecklingen utfördes, så utfördes också full stack-utvecklingen. Full stack delas i två delar, ett gränssnitt och en serverdel. Full stack-utvecklingen utfördes för att enklare skilja åt de två delarna gränssnitt och serverdel. Hela programmet kunde ha

utförts som ett program men då kunde endast en person ha använt programmet åt gången, vilket inte var ändamålsenligt.

5.2.1 Gränssnitt

Vid utveckling av webbsidor och applikationer måste man ta i beaktande hur olika resolutioner påverkar hur det ser ut och hur informationen framhävs. Ofta på en webbsida finns det två versioner, den ena är gjord för mobiltelefonen och den andra är gjord för datorer. Microsoft Access är ett dator baserat program och finns inte att få till en mobiltelefon. Detta underlättade utformningen för programmet. De flesta datorskärmar har resolutionen 1920x1080 och den kommer att användas som utgångspunkt. Programmet går att använda i andra resolutioner men utseendet är optimerat till en resolution. Vid användning i andra resolutioner kommer det finnas rullgardiner som med hjälp av de går att förflytta sig.

Programmet är gjort så att de går att modifiera kalkylbladsdelen så att det är enligt eget tycke. Det går exempelvis att flytta kolumner, ändra kolumnernas storlekar, filtrera kolumnerna och annat likande.

5.2.2 Serverdel

Förvarandet av serverdelen kunde ha gjorts på två sätt, genom en intern server eller som en SharePoint lista. Vid prestandatestning blev det klart att den interna servern fungerade aningen snabbare och vara lättare att underhålla eftersom databasen är sparad i ett Access program och inte som en SharePoint lista. Eftersom Access programmet gjordes som en uppdelad databas betydde det att det fanns två program. Det ena var databasprogrammet och det andra var gränssnittsprogram. För enkelhetens skull sparades båda programmen i närheten av varandra men med olika kryptering. Administratören för servern har begränsat rättigheterna för användare så att man inte kan radera en fil som man inte är ägare till. I och med detta går det att hålla databasen synlig för användare fast de inte kan använda eller radera den.

6 Resultatet

I resultatavsnittet presenteras programmet (Figur 11) med figurer och funktionerna i programmet presenteras skriftligen. Avsnitten är uppdelade i den visuella delen och kalkylbladsdelen. Utöver de två delarna har programmet en rubrik som har polisen blå färg som bakgrund och till höger har den sex kommandoknappar. Dessa knappar är Avustajat Vaasa, Yhteenveto, Tallenna, Päivitä, Takaisin och Lopeta. Knapparna är programmerade att göra vissa funktioner, exempelvis öppnar Avustajat Vaasa knappen en flik över rättegångsbiträden som finns i Österbotten och Yhteenveto knappen ger en sammanfattning på ärendet som är markerat. Övriga knappar har mera programtekniska funktioner som antingen, sparar, uppdaterar, går tillbaka eller stänger programmet. Under rubriken kommer den visuella delen och under den kommer kalkylbladet.

The screenshot shows the 'Juttuseuranta' application interface. At the top, there are navigation buttons: Avustajat Vaasa, Yhteenveto, Tallenna, Päivitä, Takaisin, and Lopeta. The main area is divided into a left sidebar with filters and a main content area. The sidebar includes fields for Ilmoitus (Brott6), Ysikkö (TJ), Päättökija (Petter), Tutkija (Minni), Päänimike (Petos), Tapahtumapvm (30/03/2020), Kirjauspvm (06/04/2022), Osalliset (Kiire), Tila (CASE), Aikaa tapahtumasta (1109), Tutkinta-aika (372), and Vanhenee (30/03/2025). The main content area has a search bar and a table with columns: Nimikkeet, Toimenpiteitä, Kuulustelu, Määrä, Tapahtumapvm, Kirjauspvm, and Selostus. Below the interface is a detailed data table.

	Ilmoitus	Päänimike	Ysikkö	TJ	Päättökija	Selostus	Tapahtumapvm	Kirjauspvm	Osalliset	CASE	Tila	Vanhenee	Aikaa	Tutkinta
Brott6	Petos	Grupp2	Petter	Minni			30/03/2020	06/04/2022			Kiire	30/03/2025	1109	372
Brott3	Pahoitteleminen	Grupp1	Joakim	Tjätte			27/09/2020	10/04/2023			Kiire	27/09/2025	928	3
Brott5	Dieväs pahoitteleminen	Grupp1	Joakim	Tjätte			03/04/2021	03/05/2022			Valmis	03/04/2023	740	345
Brott4	Varkaus	Grupp3	Kajsa	Minni			23/09/2021	20/05/2022			Odottaa	23/09/2026	569	328
Brott2	Varkaus	Grupp2	Petter	Minni			27/03/2023	07/04/2023			Jaettu	27/03/2028	17	6
Brott8	Näpistys	Grupp1	Joakim	Tjätte			27/03/2023	02/04/2023			Valmis	27/03/2025	17	11
Brott9	Tärkeä varkaus	Grupp2	Kajsa	Knatte			28/03/2023	29/03/2023			Kesken	28/03/2033	16	15
Brott1	Näpistys	Grupp1	Joakim	Tjätte			30/03/2023	06/04/2023			Jakaamatta	30/03/2025	14	7
Brott7	Maksuvälitepetos	Grupp1	Joakim	Tjätte			30/03/2023	04/04/2023			Odottaa	30/03/2028	14	9
Brott10	Pahoitteleminen	Grupp3	Petter	Micki			04/04/2023	12/04/2023			Aloitamatta	04/04/2025	9	1

Figur 10. Uppföljningsprogrammet.

6.1 Visuell del

Den visuella delen består av två delar, den ärendespecifika och tilläggsdelen. Den ärendespecifika delen består av information om ärendet, som är markerat i kalkylbladet, och tilläggsdelen består av information som underlättar uppföljningen av ärendet.

Ilmoitus	Brott6		Muita nimikkeitä	<input type="checkbox"/>
Yksikkö	TJ	Päätutkija	Toimenpiteitä kesken	<input type="checkbox"/>
Grupp2	Petter	Minni	Kuulustelut valmiina	<input type="checkbox"/>
Päänimike	Petos	5	ETPK Valmis TJ:llä	<input type="checkbox"/>
	Tapahtumapvm	Kirjauspvm	ETPK Syyttäjällä	<input type="checkbox"/>
	30/03/2020	06/04/2022	ETPK Kieli	<input type="text"/>
Osalliset	<input type="text"/>			
Tila	Kiire			
CASE	<input type="text"/>			
Aikaa tapahtumasta	1109			
Tutkinta-aika	372			
Vanhenee	30/03/2025			
	Selostus			
	<input type="text"/>			

Figur 21. Ärendespecifik del.

Från den ärendespecifika delen (Figur 12) får man fram information om själva ärendet. Från den översta fältet kan man se vilket ärende som det är fråga om, Brott6. Övriga fält innehåller information om ärendet b.la. vem är undersökare, vilken grupp, när det inträffat etcetera. Till fältet Tila har det programmerats färgfunktioner som ändra färg enligt vilket stadie ärendet är i. Ärendet är märkt som ett brådskande ärende och då har fältens färg ändrat till rött. Varje stadie är tilldelat en färg som hjälper uppfatta i vilket stadie ärendet är i. Till exempel färgen grönt visar att ärendet är färdigt. De tre fält som är längst ner har funktioner som påverkas av inmatad information. De två första fälten räknar hur många dagar det har gått sedan händelsen inträffade respektive anmäldes. Dessa är programmerade så att om dagar överskrider en viss gräns så ändras färgerna i fälten. Det är gjort av den anledningen att man lättare kan hålla kolla på vilka ärenden som är brådskande.

På den högra sidan finns Ja/nej alternativ som underlättar att kontrollera om det finns exempelvis någon åtgärd som är på hälft. Det stora fältet är reserverat för text. Texten som får skrivas här är enbart text som underlättar undersökaren med identifieringen av ärendet.

Nimikkeet		Toimenpiteitä		Kuulustelu	
Toimenpide	Alkamispvm	Valmispvm	Suorittaja	Selostus	
Utfört åtgärd 1	13/04/2023	13/04/2023	Micki		
Utfört åtgärd 10	12/04/2023	12/04/2023	Micki		
Utfört åtgärd 2	08/04/2023	10/04/2023	Petter		
Utfört åtgärd 6	08/04/2023	09/04/2023	Minni		
Utfört åtgärd 4	10/04/2023	11/04/2023	Micki		
	13/04/2023				

Figur 32. Tilläggsdelen.

Ett ärende kan innehålla många tilläggshändelser som inte klassas som nödvändiga för att finnas på kalkylbladet men som kan underlätta uppföljningen. När man exempelvis påbörjar en åtgärd men inte får den färdig genast eller tilldelar en uppgift någon annan kan man skriva in det i programmet. Man kan då senare granska vem som gjorde vilken åtgärd och när. Det har lagts till en motsvarande flik för förhör och för andra rubriceringar. Tabellen Ilmoitus kunde på grund av databas tekniska orsaker endast spara en rubricering åt gången och därför lades andra rubriceringar i fliken Nimikkeet.

6.2 Kalkylbladsdelen

För att man så snabbt som möjligt skall få en uppfattning över en stor mängd information måste de presenteras på ett sätt så att man uppfattar det. För att göra detta använder man sig ofta av ett kalkylblad som klart och tydligt framställer informationen. Ett kalkylblad (Figur 14) användes i detta program för att framställa en stor mängd information på ett så smidigt och lättförståeligt sätt som möjligt.

	Ilmoitus	Päänimike	Yksikkö	TJ	Päättökija	Selostus	Tapahtumapvm	Kirjauspvm	Tila	Vanhenee	Aikaa	Tutkinta
+	Brott6	Petos	Grupp2	Petter	Minni		30/03/2020	06/04/2022	Kiire	30/03/2025	1109	372
+	Brott3	Pahoinpitely	Grupp1	Joakim	Tjatte		27/09/2020	10/04/2023	Kiire	27/09/2025	928	3
+	Brott5	Lievä pahoinpitely	Grupp1	Joakim	Tjatte		03/04/2021	03/05/2022	Valmis	03/04/2023	740	345
+	Brott4	Varkaus	Grupp3	Kajsa	Minni		23/09/2021	20/05/2022	Odottaa	23/09/2026	567	328
+	Brott2	Varkaus	Grupp2	Petter	Minni		27/03/2023	07/04/2023	Jaettu	27/03/2028	17	6
+	Brott8	Näpistys	Grupp1	Joakim	Tjatte		27/03/2023	02/04/2023	Valmis	27/03/2025	17	11
+	Brott9	Törkeä varkaus	Grupp2	Kajsa	Knatte		28/03/2023	29/03/2023	Kesken	28/03/2033	16	15
+	Brott1	Näpistys	Grupp1	Joakim	Tjatte		30/03/2023	06/04/2023	Jakaamatta	30/03/2025	14	7
+	Brott7	Maksuvälinepetos	Grupp1	Joakim	Tjatte		30/03/2023	04/04/2023	Odottaa	30/03/2028	14	9
+	Brott10	Pahoinpitely	Grupp3	Petter	Micki		04/04/2023	12/04/2023	Aloittamatta	04/04/2025	9	1
*										#Typ!		

Figur 43. Kalkylbladsdelen.

Stilen på kalkylbladet är en väldigt typisk stil. Informationen är uppdelad i rader och informationen går att sortera och filtrera enligt kolumnerna, exempelvis är figur 14 sorterad enligt den äldsta händelsetidpunkten. Det går att ändra storlek och flytta plats på kolumnerna. Färgprogrammeringen för "Tila" kolumnen framkommer här på ett lättuppfattligt sätt.

6.3 Sammanfattning

En tilläggfunktion som lades till i programmet var en sammanfattnings funktion som sammanfattar det ärende som är valt. I sammanfattningen kommer all den information som finns om ärendet i kalkylbladet och i tilläggsdelen. Sammanfattningen går att spara som en PDF eller printa ut.

Rikosilmoitus							Brott6	
Ilmoitusnumero	RyhmäID	Tutkinnanjohtaja	Päätutkija	Tutkija	ETPK Kieli	PK Tila	ETPK TJ:llä	Syyttäjällä
Brott6	Grupp2		Minni			<input checked="" type="checkbox"/> Kiire	<input type="checkbox"/>	<input type="checkbox"/>
Päänimike		Tapahtumapäivä	Kirjauspäivä	Osalliset	Selostus			
Petos		30/03/2020	06/04/2022					
Muut nimikkeet								
Nimike		KPL	Tapahtumapvm	Kirjauspvm	Selostus			
Petoksen yritys		1	30/03/2020	13/04/2023				
Kavalluksen yritys		1	30/03/2020	13/04/2023				
Kuulustelut								
Kuulustelija	Kuultava	K. nro	Kuulustelu pvm	Kieli	Selostus			
Micki	AO1	1	13/04/2020	Ruotsi				
Minni	RE1	1	07/12/2021	Suomi				
Toimenpiteet								
Toimenpide		Alkamispvm	Valmispvm	Selostus				
Utfört åtgärd 4		10/04/2023	11/04/2023					
Utfört åtgärd 6		08/04/2023	09/04/2023					
Utfört åtgärd 2		08/04/2023	10/04/2023					
Utfört åtgärd 10		12/04/2023	12/04/2023					
Utfört åtgärd 1		13/04/2023	13/04/2023					

Figur 54. Sammanfattning.

7 Diskussion

Redan i ett tidigt skede av designprocess blev det klart att programmet kommer att kräva en hel del tid för att den skall uppnå de krav som utvecklaren ställde på programmet. Eftersom uppdragsgivaren inte hade större krav på utseendet eller funktionerna i verktyget skulle antagligen en mycket enklare version uppfyllt kraven. Programmet kunde optimeras ytterligare och olika scheman kunde förbättras men programmet uppfyller redan som sådant sin funktion.

Utvecklingsprocessen var i sig en utmaning och några gånger kom det en känsla av förtvivlan som gjorde att hela idén ifrågasattes. Utvecklarens motivation att utveckla ett fungerande program överrumplade förtvivlan och i april 2023 har verktyget varit i användning i tre månader. Det har gjorts små förbättringar i användningen och det har lagts till rader enligt användarnas önskemål. Som ett tillägg har det utvecklats ett inmatningsprogram, i Access, som används när ett nytt ärende matas in i programmet av någon som inte har rättighet till programmet.

Hela utvecklingsprocessen var också en inlärningsperiod eftersom utvecklaren endast hade grundläggande kunskaper i Access. Inledningsvis tog det tiotals timmar att förstå hur olika funktioner fungerade i Access. Vid inledningen av utvecklingen fanns det inte heller någon visuell plan hur verktyget skulle se ut men efter undersökning av olika program och webbsidor kunde det utformas en initial tanke över hur gränssnittet skall se ut och vad det skall innehålla.

Microsoft Access är ett program som är lätt att använda men programmet tar en del genvägar som gör en möjlig förflyttning av databasen till ett program som är gjord för databaser, exempelvis MySQL, komplicerad. Några funktioner som finns i Access gör att databasen blir korrumpert och kan inte användas någon annanstans än i Access. Om man skall göra en enkel databas över ett fabrikslager kan dessa funktioner hjälpa mycket men när man har tusentals rader data kommer man att få problem om man tagit genvägar vid uppbyggnaden av databasen. I detta uppföljningsverktyg användes inte dessa funktioner utan det utvecklades så att om man i ett senare skede vill flytta databasen till ett dedikerat databasprogram skall det vara möjligt.

Utöver utveckling av programmet överfördes en stor mängd data in i programmet för att programmet skulle vara uppdaterat för användaren. Till det användes Excel. För databastekniska skäl måste data som fördes in i Access förvandlas till siffror som programmet identifierade och ändrade till förståeliga.

Programmet har med hållna utbildningar visat sig vara tämligen enkelt att använda och är dagligen i bruk. En klar utmaning som uppstod var introduktionen av Access. Större delen av användarna var inte bekanta med Access sedan tidigare och därför fanns det en viss ovisshet och rädsla. En sak som underlättade ibruktagningen markant var att utvecklaren fanns tillförfogande på plats och fick vetskap om problemen direkt de uppstod. Detta bidrog till att problemen snabbt kunde åtgärdas och användaren fick den hjälp den behövde.

För vidareutveckling kunde programmet införas till de två återstående områdena vid Österbottens polisinrättning och varför inte andra polisinrättningar runtom i Finland. Programmet kan utvecklas och det finns redan förbättringsförslag för ytterligare utveckling. Programmet kunde med några modifieringar ändras så att det kunde användas som ett uppföljningsverktyg för en projektledare.

Slutligen vill jag tacka Magnus Jakobsson som fungerat som min handledare vid Österbottens polisinrättning och för hans hjälp gällande utvecklingsarbetet.

8 Referenser

- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas Dave. (2001). *Manifesto for Agile Software Development*. Hämtat från <http://agilemanifesto.org/iso/en/manifesto.html> den 28 March 2023
- Dennis, A., Wixom, B. H., & Tegarden, D. (2015). *System Analysis & Design - An Object-Oriented Approach with UML* (5th uppl.). New Jersey: John Wiley & Sons, Inc.
- Filipova, O., & Vilão, R. (2018). *Software Development From A to Z, A Deep Dive into all the Roles Involved in the Creation of Software* (1st uppl.). Berlin: Apress Media LLC.
- frwiki. (16.3.2023). *frwiki*. Hämtat från Roboto (typsnitt): https://sv.frwiki.wiki/wiki/Roboto_%28police_d%27%C3%A9criture%29
- Inrikesministeriet. (16.3.2023). *Brottslighet i Finland*. Hämtat från inrikesministeriet: <https://intermin.fi/sv/polisvasendet/brottslighet-i-finland>
- Kumar, M., & Sareen, M. (2010). Building Trust in E-Commerce through Web Interface. i A. Tatnall (Red.), *Web Technologies, Concepts, Methodologies, Tools, and Application* (ss. 1195-1205). Hershey: Information Science Reference (an imprint of IGI Global).
- Lag om offentlighet i myndigheternas verksamhet 21.5.1999/621. (2023). Hämtat från Finlex: <https://www.finlex.fi/sv/laki/ajantasa/1999/19990621>
- Langer, A. M. (2016). *Guide to Software Development - Designing and Managing the Life Cycle* (2nd uppl.). New York: Springer-Verlag London Ltd.
- Lutkevich, B. (2023). *Tech Target*. Hämtat från <https://www.techtarget.com/searchdatamanagement/definition/database> den 8 April 2023
- McCarron, C. (11.4.2023). *7 Splendid Reasons to Keep Frontend and Backend Separate*. Hämtat från IT Chronicles: <https://itchronicles.com/software-development/7-splendid-reasons-to-keep-frontend-and-backend-separate/>
- McConnell, S. (1996). *Rapid Development, Taming Wild Software Schedules* (1st uppl.). Redmond: Microsoft press.
- Microsoft. (10.4.2023a). *Introduktion till formulär*. Hämtat från Microsoft: <https://support.microsoft.com/sv-se/office/introduktion-till-formul%C3%A4r-e8d47343-c937-44e8-a80f-b6a83a1fa3ae>
- Microsoft. (10.4.2023b). *Introduktion till frågor*. Hämtat från Microsoft: <https://support.microsoft.com/sv-se/office/introduktion-till-fr%C3%A5gor-a9739a09-d3ff-4f36-8ac3-5760249fb65c>
- Microsoft. (10.4.2023c). *Introduktion till programmering i Access*. Hämtat från Microsoft: <https://support.microsoft.com/sv-se/office/introduktion-till-programmering-i-access-92eb616b-3204-4121-9277-70649e33be4f>

- Microsoft. (10.4.2023d). *Introduktion till rapporter i Access*. Hämtat från Microsoft: <https://support.microsoft.com/sv-se/office/introduktion-till-rapporter-i-access-e0869f59-7536-4d19-8e05-7158dcd3681c>
- Microsoft. (10.4.2023e). *Introduktion till tabeller*. Hämtat från Microsoft: <https://support.microsoft.com/sv-se/office/introduktion-till-tabeller-78ff21ea-2f76-4fb0-8af6-c318d1ee0ea7>
- Microsoft. (10.4.2023f). *Utnyttja dina data*. Hämtat från Microsoft: <https://www.microsoft.com/sv-se/microsoft-365/access>
- Microsoft. (11.4.2023). *Split an Access database*. Hämtat från Microsoft: <https://support.microsoft.com/en-us/office/split-an-access-database-3015ad18-a3a1-4e9c-a7f3-51b1d73498cc>
- Microsoft. (16.3.2023). *Databasgrunder*. Hämtat från Microsoft: <https://support.microsoft.com/sv-se/office/databasgrunder-a849ac16-07c7-4a31-9948-3c8c94a7c204>
- Moynihan, G. P. (2011). *An Overview of Executive Information Systems*. i I. R. USA, *Enterprise Information Systems: Concepts, Methodologies, Tools, and Applications* (ss. 235-249). Hershey: Information Science Reference (an imprint of IGI Global).
- Northwood, C. (2018). *The Full Stack Developer: Your essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer* (1st uppl.). New York: Apress Media LLC.
- Pawlak, E. J., & Vinter, R. D. (2004). *Designing and Planning Programs for Nonprofit and Government Organizations* (1st uppl.). San Francisco: A Wiley Imprint .
- Personregisterlag 471/1987. (2023). Hämtat från Finlex: <https://finlex.fi/sv/laki/alkup/1987/19870471>
- Polisen. (16.3.2023a). *Gör brottsanmälan*. Hämtat från Polisen: <https://poliisi.fi/sv/gor-en-brottsanmalan>
- Polisen. (16.3.2023b). *Polisinrättningen i Österbotten*. Hämtat från Polisen: <https://poliisi.fi/sv/polisinrattningen-i-osterbotten>
- Polisen. (16.3.2023c). *Verksamhetsmiljö och statistik*. Hämtat från Polisen: <https://poliisi.fi/sv/operativ-miljo-och-statistik>
- Polislag 872/2011. (2023). Hämtat från Finlex: <https://finlex.fi/sv/laki/alkup/2011/20110872?search%5Btype%5D=pika&search%5Bpika%5D=polislag>
- Robbins, J. N. (2006). *Web Design in a Nutshell*. Sebastopol: O'Reilly Media, Inc.
- Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2016). *Systems Analysis and Design in a Changing World* (7th uppl.). Boston: Cengage learning .
- Schmidt, R. F. (2013). *Software Engineering, Architecture-Driven Software Development* (1st uppl.). Waltham: Elsevier Inc.

- Schwalbe , K. (2011). *Information Technology - Project management* (6th uppl.). Boston: Course Technology, Cengage Learning .
- Statistikcentralen. (16.3.2023). *13ex -- Brott som kommit till kännedom och utredning av dem efter brottskategori, kommunen där brottet skett och anmälningår, 1980-2021*. Hämtat från Statistikcentralen: https://pxdata.stat.fi/PxWeb/pxweb/sv/StatFin/StatFin_rpk/statfin_rpk_pxt_13ex.px/
- Terra, J. (17.2.2023). *Simplilearn*. Hämtat från What is Microsoft Access? An Introductory Guide: <https://www.simplilearn.com/what-is-microsoft-access-article>
- Verksamhet i hela Finland*. (16.3.2023). Hämtat från Polisen: <https://poliisi.fi/sv/vi-verkar-overallt-i-finland>
- Xanthidis, D., Nicholas, D., & Argyrides, P. (2010). A Proposed Template for the Evaluation of Web Design Strategies. i A. Tatnall, *Web Technologies: Concepts, Methodologies, Tools, and Applications* (ss. 119-144). Hershey: Information Science Reference (an imprint of IGI Global).
- Zadrożny, S., de Tré, G., de Caluwe, R., & Kacprzyk, J. (2009). An Overview of Fuzzy Approaches to Flexible Database Querying. i J. Erickson (Red.), *Database Techonologies: Concepts, Methodologies, Tools, And Applications* (ss. 135-156). Hershey: Information Science Reference (an imprint of IGI Global).
- Zaigham, M. (2017). *Software Project Management for Distributed Computing - Life-Cycle Methods for Deveolping Scalable and Reliable Tools* . Springer International Publishing AG.