

Optimalt lärande: Automatisering av entalsaritmetik för barn med hjälp av spelifiering och visualisering i en mobilapplikation

Optimal education: Automation of singular arithmetic for children using gamification and visualization in a mobile application

Mabest Amin
Odin Dahlström
Emil Holmstedt
Martin Hultgren
Casper Erik Nerf Kanefall
Eric Van Nunen
William Puranen
Yadgar Suleiman

Handledare : Lena Buffoni
Examinator : Kristian Sandahl

Upphovsrätt

Detta dokument hålls tillgängligt på Internet - eller dess framtida ersättare - under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida <http://www.ep.liu.se/>.

Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a period of 25 years starting from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: <http://www.ep.liu.se/>.

Mabest Amin
Odin Dahlström
Emil Holmstedt
© Martin Hultgren
Casper Erik Nerf Kanefall
Eric Van Nunen
William Puranen
Yadgar Suleiman

Sammanfattning

TODO

Tillkännagivanden

TODO

Innehåll

Sammanfattning	iii
Författarens tack	iv
Innehåll	v
Figurer	vii
1 Introduktion	1
1.1 Motivering	1
1.2 Syfte	1
1.3 Frågeställning	2
1.4 Avgränsningar	2
1.5 Kontext	2
2 Bakgrund	3
2.1 Kundens bakgrund och vision	3
2.2 Kundens optimeringsalgoritm	3
2.3 Gruppens tidigare erfarenheter	4
2.4 Tidigare arbete	4
3 Teori	6
3.1 Spelifiering	6
3.2 Entalsaritmetik	6
3.3 Strategier för entalsaritmetik	6
3.4 Programmeringsspråk	8
3.5 Ramverk	8
3.6 Verktyg	8
3.7 Kommunikationskanaler	9
3.8 Utvecklingsmetod	10
4 Metod	11
4.1 Skapa värde för kunden	11
4.2 Erfarenhetsdokumentation	11
4.3 Stöd av systemanatomi	12
4.4 Design av grafiskt gränssnitt	12
4.5 Design av användarprofilsystem	12
5 Resultat	14
5.1 Systembeskrivning	14
5.2 Processbeskrivning	18
5.3 Gemensamma erfarenheter	20
5.4 Användartester	20

6	Diskussion	21
6.1	Resultat	21
6.2	Metod	22
6.3	Arbetet i ett vidare sammanhang	22
7	Slutsatser	25
	Litteratur	26

Figurer

2.1	Mobilapplikationen Optimalt Lärande innan grafiska ändringar	5
5.1	Figma-prototyp för mobilapplikationen Optimalt Lärande	15
5.2	Övergripande arkitektur för mobilapplikationen Optimalt Lärande	17
5.3	Systemanatomi för mobilapplikationen Optimalt Lärande	18
6.1	SusAF: Hållbarhetsanalys-diagram	22



1 Introduktion

I den här rapporten behandlas ett projektarbete utfört av en grupp studenter vid Linköpings universitet. Rapportens fokus är att beskriva utvecklingsprocessen relaterad till vidareutvecklingen av en mobilapplikation. Applikationens syfte är att underlätta inlärandet av entalsaritmetik för barn. För att uppnå syftet har en ny visuell design utvecklats till applikationen. Designen använder spelifieringsprinciper för att förbättra användarens upplevelse och engagemang [1]. Applikationen tar även hjälp av visualiseringsstrategier för att främja lärandet för barn.

1.1 Motivering

Grundläggande matematikkunskaper är en otroligt viktig förutsättning för framgång inom fortsatta studier samt fortsättningsvis i framtida karriär. Dessvärre är det idag väldigt vanligt med bristande matematikkunskaper bland människor i många länder [2]. Det är framförallt förekommande i länder med bristande ekonomiskt stöd för utbildningssystem [2]. Den ökande tillgången till internet och mobiltelefoner runtom i världen ger dock nya och intressanta möjligheter till utbildning på andra sätt än den traditionella skolan. Målet med detta projekt är att använda dessa nya möjligheter genom att släppa en interaktiv mobilapplikation som kan användas över hela världen. Global tillgänglighet är extra lämpligt inom specifikt matematik då det matematiska språket endast kräver bekantskap med arabiska siffror. Studier visar även på att matematiska kunskaper tidigt i livet är en av de bästa indikationerna på framtida akademisk framgång [3].

1.2 Syfte

Syftet med den här rapporten är framför allt att beskriva projektets arbetsprocess och dess resultat. Projektet som beskrivs är en vidareutveckling av en existerande applikation utvecklad som tidigare kandidatprojekt. Projektet innehåller en optimeringsalgoritm vars syfte är att ta fram optimala sekvenser av entalsaritmetikuppgifter. Denna algoritm lägger grunden till projektet och möjliggör optimalt lärande. Till följd av denna möjlighet är målet med detta projekt att vidareutveckla en mobilapplikation som kan fånga användarens intresse för att uppnå teknikens påvisade potential. Projektet använder tekniker som spelifiering och visualisering

som möjliggör för matematiken att läras ut på ett roligare och effektivare sätt. Rapporten avser att dokumentera och beskriva lärdomar som skapats från de utvecklingsprocesser som tillämpats under projektets gång.

1.3 Frågeställning

1. Hur kan mobilapplikationen Optimalt Lärande implementeras så att man skapar värde för kunden?
2. Vilka erfarenheter kan dokumenteras från mobilapplikationen Optimalt Lärande som kan vara intressanta för framtida projekt?
3. Vilket stöd kan man få genom att skapa och följa upp en systemanatomi?
4. Hur designar man ett intuitivt grafiskt gränssnitt utan användandet av text och språk?
5. Hur designar man ett intuitivt användarprofilsystem för barn med stöd för offlineanvändning, anonym datainsamling samt flera användare per enhet och e-postadress?

1.4 Avgränsningar

Projektet är en vidareutveckling av en existerande applikation vilket gjorde att val av utvecklingsverktyg, databas, och andra verktyg var förutbestämda då det inte var möjligt att ändra på dessa för att kunna slutföra projektet inom de tidsramar som satts upp.

Då applikationen är ämnad att kunna användas oavsett vilken bakgrund och förkunskaper användaren har, inkluderat vilket språk användaren pratar, är det också ett krav att applikationen inte ska använda sig av språk för att förmedla saker till användaren. Detta bidrar till stora begränsningar eftersom det finns en gräns för hur mycket en applikation kan förmedla med bara bilder och färger.

Eftersom applikationen ska kunna användas av så många användare som möjligt innebär det att den ska fungera även på äldre enheter. Då det finns många personer i världen som inte har tillgång till en nyare mobiltelefon är det viktigt att applikationen fungerar på äldre enheter också. Detta begränsar vilka funktioner som kan användas samt hur mycket prestanda applikationen kan kräva av enheten.

1.5 Kontext

Projektet har genomförts i kontexten av kursen TDDD96 *Kandidatprojekt i programvaruutveckling* vid Linköpings Universitet. Detta medför att gruppmedlemmarnas tid på projektet samt dokumentation är begränsad till 400 timmar per person. Då gruppen består av 8 personer är den totala budgeten att spendera på planering, utveckling, dokumentation, seminarium, föreläsningar och andra relevanta områden 3200 timmar. Kursen medför även högre krav på dokumentskrivning, inlämningar och andra processer som har påverkat projektets utförande.



2 Bakgrund

Undervisning av matematik för barn har under lång tid sett ut på samma sätt. Den lärs ut på samma sätt för alla barn oavsett kunskapsnivåer. Detta har gjort att utbildning inom matematik har varit ineffektiv och många barn går miste om kunskaper som de annars hade kunnat få. Att istället lära ut matematik till barn med hjälp av interaktiv mjukvara som lär känna barnens kunskapsnivåer och sedan anpassar sig efter det kan på många olika sätt ha positiva effekter, inte minst att barnen blir bättre på matematik.

Applikationen kommer i huvudsak användas i utbildande syfte för barn men kommer även användas i forskningssyfte där man kan forska på olika metoder som används i applikationen och hur barn interagerar med dessa metoder.

2.1 Kundens bakgrund och vision

Kunden har en bakgrund inom forskning av optimeringslära vid Institutionen för ekonomisk och industriell utveckling på Linköpings universitet. Kunden har tidigare arbetat ihop med kandidatgrupper på Linköpings universitet för att utveckla denna applikation med mål att den ska publiceras på de två största distributionsplattformarna för mobilapplikationer, *App Store* och *Play Store*. Kunden vill på så sätt kunna fortsätta forska på hur barn lär sig matematik på ett optimalt sätt samt hjälpa barn i hela världen att bli bättre på matematik.

2.2 Kundens optimeringsalgoritm

Kunden har inom ramen för sin forskning utvecklat en optimeringsbaserad algoritm vars syfte är att generera en optimal sekvens av uppgifter, med avseende på inläring och automatisering av entalsaritmetik för lågstadieelever. Som indata använder algoritmen olika data som samlas in av en mobilapplikation medan användaren löser de uppgifter som genereras av algoritmen, till exempel svar på uppgifter (rätt eller fel) samt hur lång tid det tog att lösa uppgifterna. Optimeringsalgoritmen bestämmer på så vis vilka uppgifter som användaren behöver repetera och väljer ut dessa innan användaren kan fortsätta med nya uppgifter och räknesätt. Kundens optimeringsalgoritm kan generera uppgifter inom entalsaritmetik för sex

olika räknesätt: addition, addition med okänd, subtraktion, multiplikation, multiplikation med okänd samt division.

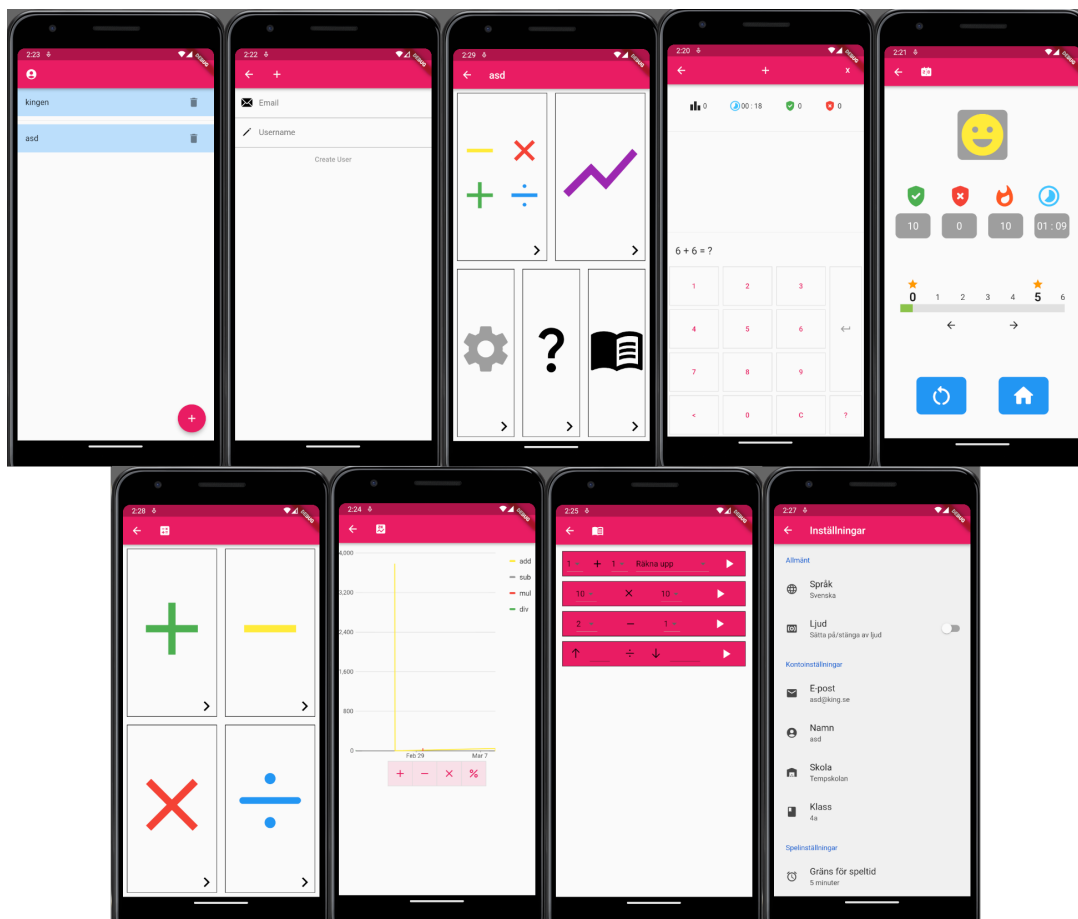
2.3 Gruppens tidigare erfarenheter

Gruppen består av 8 civilingenjörsstudenter på Linköpings universitet, av dessa pluggar 5 civilingenjör i datateknik och 3 civilingenjör i mjukvaruteknik. Utöver de många erfarenheter som studenterna fått genom sina nästan 3 år på utbildningen finns det spridd kompetens inom gruppen för utveckling av applikationer i *Flutter* samt databasutveckling. För att kunna utveckla denna applikation har gruppen anordnat workshops för att utbilda hela gruppen i olika verktyg som har varit viktiga för projektets fortskridande.

Kompetenser inom programutvecklingsmetodik, såsom agila metoder och versionshantering, har samtliga studenter i gruppen då dessa kompetenser är en del av utbildningen. Alla inom gruppen har även tidigare erfarenhet av att jobba i grupp då utbildningarna innehåller flera olika grupparbeten.

2.4 Tidigare arbete

Systemet som projektet berör är från början utvecklad av en tidigare kandidatgrupp som under våren 2022 utvecklade grunden till applikationen. Denna grupp utvecklade grundläggande funktionalitet för räkning av uppgifter samt koppling mot en databas[4]. Systemet har sedan blivit ytterligare vidareutvecklat av en kandidatgrupp som under våren 2023 implementerade funktionalitet för visualisering av de fyra räknesätten. Denna grupp utforskade även hur spelifieringselement kunde implementeras för att öka barns intresse av att använda applikationen [5]. Utseendet av applikationen innan någon vidareutveckling gjordes presenteras i figur 2.1.



Figur 2.1: Mobilapplikationen Optimalt Lärande innan grafiska ändringar



3 Teori

Detta kapitel behandlar teori som är relevant inom ramen för detta projekt för att ge läsaren en bättre förståelse för rapporten.

3.1 Spelifiering

Spelifiering är användandet av spelkomponenter i icke-spelkontexter [1]. Exempel på detta kan vara poängsystem, topplistor, och utmärkelser. Poängsystemet kan användas till att dela ut poäng baserat på hur spelaren presterar, topplistor används ofta till för att jämföra spelarens prestation med andra spelares prestationer, och utmärkelser brukar användas för att ge spelaren extra uppmärksamhet för att ha klarat av något som kan anses vara extra svårt [6]. Syftet med detta är att göra någonting som annars kan upplevas tråkigt eller långsamt mer intressant eller roligt.

3.2 Entalsaritmetik

Entalsaritmetik syftar på matematik som enbart berör uppgifter med enbart en operator och där operanderna är ental. Entalsaritmetik kommer i denna rapport, utöver de fyra grundläggande räknesätten, inkludera två av kunden definierade räknesätt. Addition med okänd variabel samt multiplikation med okänd variabel. Dessa räknesätt innebär att den som räknar uppgiften får se svaret och en av operanderna. Lösningen på uppgiften blir då istället vad den andra operanden behöver vara för att få det önskade svaret.

3.3 Strategier för entalsaritmetik

Addition är det mest grundliga räknesättet. Alla andra räknesätt vi behandlar kan på något sätt kopplas tillbaka till addition. Till följd av detta är det väsentligt att additionsräkningen automatiseras. På grund av detta finns det ett flertal strategier för addition. Varje strategi har en talmängd A där strategin kan appliceras.

3.3.1 Tiokompisar

Tiokompisar är baserat på att försöka summera upp x och y till en tia och sedan addera på resten. Talmängden A då denna strategi kan användas är följande:

$$A = \{x, y \mid x + y > 10\}$$

Ett exempel av strategin tiokompisar är:

$$7 + 4 \rightarrow 10 + 1 \rightarrow 11$$

3.3.2 Räkna upp

Räkna upp är baserat på att föra över allt värde från den lägre siffran till den andra. Talmängden A då denna strategi kan användas är följande:

$$A = \{x, y \mid x + y \leq 10\}$$

Ett exempel av strategin räkna upp är:

$$2 + 4 \rightarrow 0 + 6 \rightarrow 6$$

3.3.3 Dubblar

Dubblar är en grundsten för senare strategier där kunskapen om vad lägre tal dubblas till används. Talmängden A då denna strategi kan användas är följande:

$$A = \{x, y \mid x = y\}$$

Om x skall dubblas så används kunskapen att $n + 1 = x$ för att räkna ut att $x + x \rightarrow n + n + 2$. Ett exempel av strategin räkna upp är:

$$6 + 6 \rightarrow (5 + 1) + (5 + 1) \rightarrow (5 + 5) + (1 + 1) \rightarrow 10 + 2 \rightarrow 12$$

3.3.4 Nästan dubblar I

Nästan dubblar I baseras på strategin dubblar och används när siffrorna har en differens av ett. Talmängden A då denna strategi kan användas är följande:

$$A = \{x, y \mid \text{abs}(x - y) = 1\}$$

Här används kunskapen att uppgiften kan skrivas som $x + (x + 1) \rightarrow (x + x) + 1$ för att räkna ut talet med hjälp av strategin dubblar och sedan addera ett. Ett exempel på detta är:

$$(8 + 9) \rightarrow (8 + 8) + 1 \rightarrow \text{dubblar}(8) + 1 \rightarrow 16 + 1$$

3.3.5 Nästan dubblar II

Nästan dubblar II baseras på strategin dubblar och används när siffrorna har en differens av två. Talmängden A då denna strategi kan användas är följande:

$$A = \{x, y \mid \text{abs}(x - y) = 2\}$$

Här används kunskapen att uppgiften kan skrivas som $(x + 1) + (x - 1) \rightarrow x + x$ för att räkna ut talet med hjälp av strategin dubblar. Ett exempel på detta är:

$$(7 + 9) \rightarrow (8 + 8) \rightarrow \text{dubblar}(8) \rightarrow 16$$

3.4 Programmeringsspråk

Dessa programmeringsspråk är relevanta för projektet.

3.4.1 Dart

Dart är ett programmeringsspråk på hög abstraktionsnivå utvecklat av Google som främst används för att bygga plattformsoberoende applikationer med grafiskt användargränssnitt i ramverket Flutter [7].

3.4.2 Python

Python är ett mycket populärt programmeringsspråk på hög abstraktionsnivå som fokuserar på läsbarhet och enkelhet. Python har ett brett användningsområde såsom webbutveckling, maskininlärning, och artificiell intelligens. Detta språk används i mobilapplikationens tillhörande backend-system [8].

3.4.3 C++

C++ är ett programmeringsspråk på låg abstraktionsnivå med fokus på prestanda och flexibilitet. Det är en övermängd till programmeringsspråket C och erbjuder alternativ för både imperativ och objektorienterad programmering. Språket används i mobilapplikationens plattformsspecifika bryggkod mot kundens optimeringsalgoritm, eftersom denna är skriven i C++ [9].

3.5 Ramverk

Detta avsnitt beskriver ramverken som använts under projektet.

3.5.1 Flutter

Flutter är ett ramverk med öppen källkod utvecklat av Google som används för att kunna bygga plattformsoberoende applikationer med grafiska användargränssnitt från en enda kodbas. Programmeringsspråket som används är Dart, vilket innebär att applikationen kan kompileras till binärkod som kan exekveras av Dart VM, eller till maskinkod för ARM eller x86 [10].

3.5.2 Django

Django är ett ramverk för webbutveckling i språket python. ramverket har öppen källkod och är utvecklat för att göra det så enkelt att börja och köra direkt utan större konfiguration [11].

3.6 Verktyg

Följande verktyg har använts under projektets gång.

3.6.1 Gitlab

Gitlab är en webbtjänst för mjukvaruhantering. Den erbjuder versionshantering av koden med bug spårning och pipeline för testning. Alla medlemmar i ett projekt har tillgång till hela kodbasen och kan ladda upp nya versioner [12].

3.6.2 CI pipeline

CI pipeline är ett verktyg som finns integrerat i Gitlab. Det tillåter utvecklare att definiera kod som utvecklarna vill ska köras vid specifika tillfällen. Till exempel varje gång någon lägger till ny kod i projektet [12].

3.6.3 Android Studio

Android Studio är en integrerad utvecklingsmiljö specialiserad på Androidutveckling skapad av Google. Den erbjuder stöd av Flutter och Dart med integrering av *Android Software Development Kit*. Därmed kan Android-enheter emuleras på datorer [13].

3.6.4 Visual Studio Code

Visual Studio Code är en textredigerare med stöd för integration med flera ramverk och verktyg skapat av Microsoft [14].

3.6.5 Container

En mjukvaru-container är ett portabelt paket som innehåller den miljö som krävs för att köra specifik mjukvara [15].

3.6.6 Docker

Docker är ett verktyg för att hantera och packetera mjukvaru-containerar. Dessa containerar ser till att miljön runt mjukvaran inte varierar. Detta resulterar i att det bara krävs en konfiguration [15].

3.6.7 Overleaf

Overleaf är en webbtjänst som tillåter samarbete från flera parter i skapandet av dokument i LaTeX format [16].

3.6.8 Google Drive

Google Drive är en molntjänst för förvaring och samarbete av dokument skapat av Google [17].

3.6.9 Figma

Figma är ett verktyg för att skapa designprototyper av användargränssnitt. Prototyperna ger en överblick av designen och begränsad funktionalitet av att navigera mellan olika sidor. Ingen underliggande funktionalitet implementeras men de går därmed mycket snabbare att skapa prototypen [18].

3.7 Kommunikationskanaler

Följande kommunikationskanaler användes under projektets gång.

3.7.1 Discord

Discord är ett verktyg för röst-, text- och videokommunikation. Med hjälp av discord delas ämnen upp i olika kanaler för att enkelt kunna hitta rätt information [19].

3.7.2 Microsoft Teams

Microsoft Teams är en molnbaserad plattform för kommunikation och delning av filer i realtid [20].

3.8 Utvecklingsmetod

I detta avsnitt beskrivs utvecklingsmetoderna som användes under projektets gång.

3.8.1 Scrum

Scrum är ett agilt ramverk för problemlösning och projektledning. Meningen med Scrum är att iterativt och responsivt utveckla en produkt samtidigt som utvecklingen är responsiv till variabler [21]. Sättet detta görs på är genom att utveckla i tidsepoker som kallas *sprints*. Varje sprint pågår i 2 till 4 veckor och har ett delmål som hela gruppen skall jobba med där delmålet är en funktionell komponent av det slutgiltiga målet, produkten. Detta medför att produkten alltid går att använda. I början av en sprint planeras vad som ska göras för att uppnå delmålet, detta representeras oftast med en *Scrumtavla*. I slutet av en sprint genomförs en utvärdering som kallas *sprintåterblick*, i denna utvärdering diskuteras hur arbetet planerades, vad som gick bra, vad som gick dåligt och vad som bör ändras i planeringen inför nästa sprint. Planeringen och utvärderingen leds av en person med titeln *Scrum Master*. [21]



4 Metod

Detta kapitel beskriver vilka metoder som har använts för att kunna svara på frågeställningarna som ställdes upp tidigare i rapporten.

4.1 Skapa värde för kunden

För att veta vad gruppen kunde implementera för att skapa värde för kunden användes framförallt en kravspecifikation. Denna kravspecifikation skrevs i samråd med kunden och inkluderade alla de krav som applikationen behövde uppfylla för att kundens behov skulle vara fyllda. Exempel på saker som ingick i kravspecifikationen var vilken data som skulle lagras i en databas, vilka funktioner som applikationen skulle ha samt en specifikation vilken typ av enheter applikationen skulle fungera på.

Gruppen använde sig också av en kvalitetsplan för att säkerställa att projektet skulle ha ett framtida värde för kunden. För att koden som utvecklats i projektet skulle fortsätta ha värde i framtiden var det viktigt att denna kod var dokumenterad och kommenterad. Detta säkerställer att framtida utvecklare kan få en inblick i hur applikationen fungerar vid eventuella behov av vidareutveckling.

4.2 Erfarenhetsdokumentation

För detta projekt finns det många erfarenheter som kan föras vidare till framtida projekt. För att dessa erfarenheter ska dokumenteras och uppmärksammas fanns det ett antal metoder som gruppen använde sig av.

4.2.1 Sprintåterblick och veckomöte

För att ta vara på erfarenheter från varje sprint höll gruppen ett sprintåterblicksmöte två gånger per sprint. Vid varje möte fick gruppmedlemmarna berätta vad de hade gjort under veckan samt om det fanns någonting som hindrade dem från att utföra sitt arbete eller om de behövde hjälp med någonting. I början av projektet avslutades möten med att gruppen gemensamt diskuterade om det var någonting under veckan som hade fungerat mer eller mindre bra. Detta dokumenterades sedan i mötets anteckningar. Senare i projektet beslutade

gruppen att mer fokus bör läggas på erfarenhetsfångst under dessa möten och därför byttes ordningen i agendan. Mötet inleddes då istället med att diskutera vad som hade fungerat mer eller mindre bra och därefter diskuterades vad som gjorts under veckan.

4.2.2 Workshops

Gruppen höll ett antal workshops för att de gruppmedlemmar som fördjupat sig inom något specifikt ämne relevant för projektet, eller de med relevanta förkunskaper, skulle kunna förmedla denna kunskap till övriga gruppmedlemmar. Efter dessa workshops kunde gruppen komma överens om hur dessa specifika områden skulle behandlas i projektet. Dessa beslut noterades sedan i det dokument som berörde den punkten. Bestämmelser kring hur en *merge request* skulle gå till dokumenterades till exempel i kvalitetsplanen.

4.3 Stöd av systemanatomi

För att söka svar på frågeställning 3 deltog gruppen på en workshop om systemanatomier som hölls av Kristian Sandahl, för att skapa en systemanatomi för mobilapplikationen Optimalt Lärande. Först identifierades anatomer i systemet (viktiga funktioner och beståndsdelar) med hjälp av gemensam brainstorming. Sedan diskuterades resultatet gemensamt inom gruppen, för att värdera och välja ut de anatomer som skulle ingå i systemanatomien. Slutligen producerade gruppen en systemanatomi genom att identifiera beroenden mellan anatomerna och länka samman de framtagna anatomerna baserat på dessa beroenden.

4.4 Design av grafiskt gränssnitt

Eftersom det grafiska gränssnittet inte fick innehålla någon text var det extra viktigt att gruppen verifierade att gränssnittet var intuitivt för användare.

4.4.1 Designprototyp

Gruppen skapade två prototyper för hur applikationen skulle kunna se ut. Dessa diskuterades sedan på ett internt möte mellan gruppmedlemmarna. Efter detta möte finslipades prototyperna lite till innan de visades för kund. Efter möte med kunden kunde de delar från varje prototyp som ansågs lämpligast föras över till den slutgiltiga designen. Dessa prototyper skapades i Figma, se avsnitt 5.2

4.4.2 Användartester

För att verifiera att designen var intuitiv och användarvänlig utfördes användartester på skolbarn. TODO: Detta är inte utfört än.

4.5 Design av användarprofilsystem

Användarprofilsystemet som gruppen skapade till applikationen behövde både vara intuitivt och uppfylla alla de krav för funktionallitet som kunden önskade. Gruppen använde sig av ett antal metoder för att undersöka hur detta kunde genomföras.

4.5.1 Arkitekturdiagram

Arkitekturansvarig i samråd med utvecklingsledare utvecklade ett diagram för att måla upp hur användarprofilsystemet skulle fungera. Detta diagram kunde sedan användas vid diskussion med kund för att säkerställa att den design som tagits fram uppfyllde alla krav från kunden.

4.5.2 Användartest

TODO: Användartest har ännu inte utförts.



5 Resultat

I detta avsnitt ges en översiktlig beskrivning av resultat gällande levererad produkt, arkitektur, testning, processer samt gemensamma erfarenheter.

5.1 Systembeskrivning

Dett kapitel beskriver systemet vilket är uppdelat i två separata delar, *frontend* och *backend*. Dessa delar representerar den delen användarna interagerar med, mobilapplikationen, respektive en webserver och databas. Det fokus gruppen har lagt på frontend är grafisk design och det fokus gruppen har lagt på backend är kontobaserat. Utöver detta har gruppen utvecklat dokument för att färdigställa processer, arkitektur och planering.

5.1.1 Dokument

I detta avsnitt följer en beskrivning av de dokument som levererats i samband med produkten.

Projektplan

Projektplanens huvudsakliga syfte är att både beskriva projektet som helhet samt dess struktur. Projektplanen gav även en övergripande plan på hur projektet skulle genomföras.

Kravspecifikation

Kravspecifikationen togs fram i syfte att komma överens med kunden om vilka krav systemet ska uppfylla. Detta dokumentet definierar även krav på vilka dokument som skulle vara del av leveranser samt definierar hur utvecklingen av systemet skulle prioriteras. Kravspecifikationen speglar en överenskommelse mellan kunden och projektgruppen i relation till hur den slutgiltiga produkten skulle se ut.

Kvalitetsplan

Kvalitetsplanen utvecklades för att säkerställa att kvaliteten hålls på en hög nivå i produkt och processer. Dokumentet definierar vilka processer gruppen skall följa under utvecklingen

gång samt hur dessa processer mäts och utvärderas. Kärnan av denna bit diskuteras ytterligare i kapitel 5.2.1. Utöver detta beskrivs vilka mätbara krav som ställs på produkten och hur de mäts. Detta redovisas i kapitel **TODO: Ref när vi har kvalitetskrav med.**

Arkitekturdokument

Arkitekturdokumentet togs fram i syfte om att samla all arkitektonisk information relaterad till projektet på ett och samma ställe. Målet är att ge läsaren en god överblick på systemets arkitektur.

Testplan

Testplanens togs fram i syfte att dokumentera hur systemet skulle testas. Testplanen beskriver hur alla former av testning skulle gå till samt hur samtliga krav från kravspecifikation skulle valideras.

5.1.2 Grafisk design

Designprototypen som togs fram för att kunna ge applikationen en ny visuell design presenteras i figur 5.1.

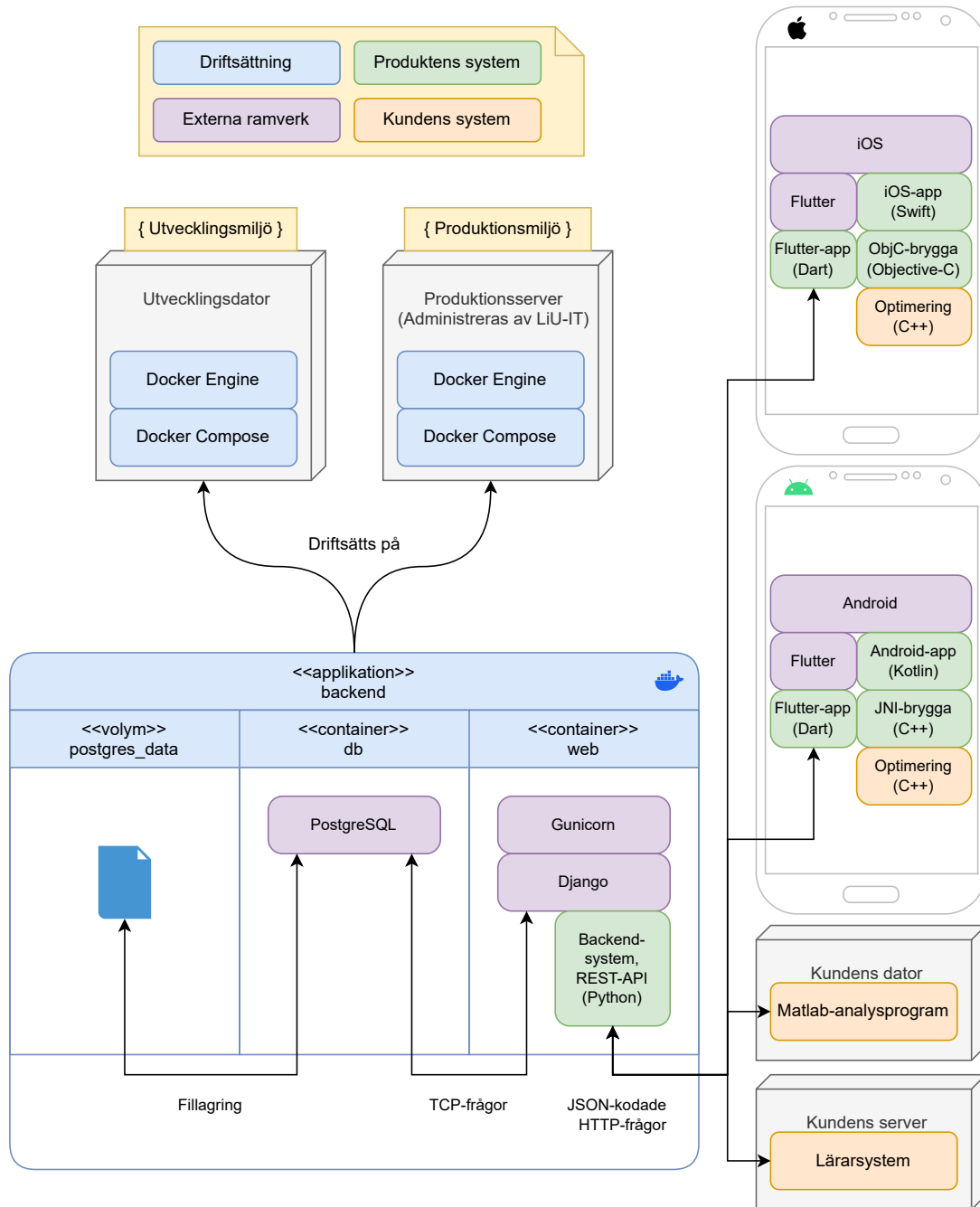


Figur 5.1: Figma-prototyp för mobilapplikationen Optimalt Lärande

5.1.3 Systemarkitektur

Systemets övergripande arkitektur beskrivs av arkitekturdiagrammet i figur 5.2. Komponenterna i arkitekturdiagrammet har färgkodats (se diagramförklaringen) för att tydliggöra och visa relationen mellan de system som utvecklas inom ramen för produkten (mobilapplikationen Optimalt Lärande), respektive system som utvecklas av kunden, samt externa ramverk och driftsättningsverktyg. När en komponent i diagrammet ligger under en annan ska det tolkas som att den undre komponenten bygger på eller använder den övre komponenten.

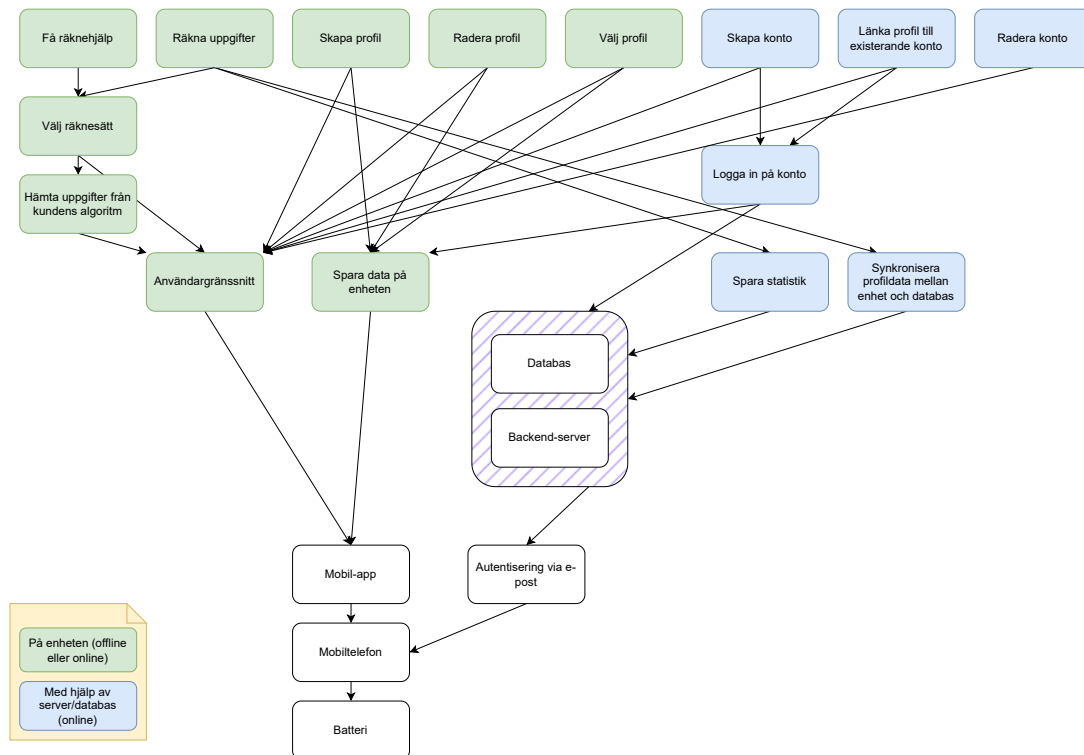
Produkten består, som figur 5.2 visar, av två mobilapplikationer, en för respektive plattform (Android och IOS), som båda kopplas samman med en Flutter-applikation. Denna Flutter-applikation är plattformsoberoende och utgör majoriteten av respektive mobilapplikation, vilket underlättar utveckling av produkten genom att möjliggöra återanvändning av källkod och kodbibliotek samt ökar enhetligheten mellan respektive mobilapplikation. Trots detta krävs fortfarande plattformsoberoende kod, som till största delen består av kundens optimeringsalgoritm samt bryggkod för att möjliggöra kommunikation mellan denna optimeringsalgoritm och den plattformsoberoende delen av mobilapplikationen.



Figur 5.2: Övergripande arkitektur för mobilapplikationen Optimalt Lärande

5.1.4 Systemanatomi

Resultatet av arbetet med att ta fram systemanatomin visas i figur 5.3. De användarnära funktionerna i systemet visas högst upp, och i systemanatomin kan sedan se hur dessa funktioner är beroende av andra, mer grundläggande funktioner. Man kan även se ur färgkodningen i systemanatomin att det finns två huvuddelar i systemet: en applikation som körs på en mobil enhet, och ett backend-system bestående av en databas och en webserver.



Figur 5.3: Systemanatomi för mobilapplikationen Optimalt Lärande

5.1.5 Kravuppfyllnad

TODO: Beskriv vilka krav som är uppfyllda. Fast kanske inte alla krav, bara typ alla krav är uppfyllda förutom".

5.2 Processbeskrivning

För att producera det system som uppfyllde alla krav som gruppen tagit fram använde sig gruppen av ett antal processer.

5.2.1 Process i fokus: Scrum

Scrum som definierat i kapitel 3.8.1 passar tyvärr inte detta projekts utveckling på alla punkter. Därför är några av grunddelarna i Scrum modifierade i detta projekt för att passa vidareutvecklingen av en produkt där det oftast är flera fast mindre uppgifter att göra. Utöver ändringar behövde också några bestämmelser göras. De bestämmelser och modifikationer som gruppen har gjort på Scrum är:

- Gruppen indelas i arbetslag under sprintplaneringen varav det ska vara minst två personer per grupp. Dessa arbetslag jobbar mot varsitt mål vilket betyder att gruppen kan ha flera mål igång samtidigt. Detta ger en mellangrund där gruppen inte behöver hitta ett stort mål som kanske inte finns samtidigt som gruppen inte får för många mål som kan resultera i splittrad utveckling.
- Gruppen genomför sprintåterblick två gånger under varje sprint. Detta betyder att gruppen utvärderar arbetet veckovis och inte sprintvis vilket resulterar i att problem i processen och planeringen potentiellt kan identifieras tidigare.

- Längden av en sprint är 2 veckor. Vilket ger en bra balans mellan responsiv planering och tid att arbeta mot delmålen.
- Gruppens Teamledare är även Scrum Master. Detta inföll naturligt då gruppens Teamledare ledsagat veckomöterna innan beslutet togs att använda Scrum.

För att utvärdera och förbättra denna process behövs mätpunkter. Dessa mätpunkter är till för att identifiera problem i gruppens utvecklingprocess och hur den kan förbättras. De mätpunkter som valts är:

1. Hur många issues blev över efter sprinten?
2. Varför var dessa issues kvar?
3. Hur många timmar hade gruppen kvar i arbete för sprinten när alla issues var klara.
4. Vilka övriga frågor togs upp på sprintåterblicken.

Dessa mätpunkter ska hjälpa till att identifiera hur fel tidsplaneringen är genom punkt 1 och 3 samt varför dessa fel skedde genom punkt 2 och 4. Detta bör underlätta nästkommande planering genom data att basera förbättringar på.

5.2.2 Designprocess

För att utveckla designprototypen presenterad i figur 5.1 har en designprocess följts. Processen har delats upp i 8 delar med inspiration från Arvola som följande [22]:

1. Konkurrensundersökning
2. Framställning av designskisser
3. Intern granskning av designskisser
4. Framställning av datorprototyper i Figma
5. Intern granskning av datorprototyper
6. Vidareutveckling av datorprototyper i Figma
7. Presentation och granskning av datorprototyper med kunden
8. Slutgiltiga justeringar enligt återkoppling

5.2.3 Kontinuerlig integrering

Det räcker inte att bara sätta upp bestämmelser för hur någonting ska göras. Gruppen var även tvungen att ha processer för att se till att de bestämdes faktiskt efterföljdes. För att åstadkomma detta använde sig gruppen av kontinuerlig integrering. Detta innebar att gruppen skapade en GitLab CI pipeline med ett antal integrationsprocesser. Dessa såg till att varje gång någon i gruppen pushade ny kod till GitLab kördes alla de tester som gruppen definierat. Dessa inkluderade statisk kodanalys, byggande av själva applikationen, samt de specifika enhetstester som gruppen skapat för att verifiera att koden fungerade som den skulle. Om något av dessa tester inte gick igenom fick gruppen en varning och kunde åtgärda det.

5.2.4 Systemtester

Hela systemet utvecklades för att uppfylla de krav som definierades i gruppens kravspecifikation. För att verifiera att kraven var uppfyllda definierades i början av projektet ett antal systemtester som hade som mål att verifiera huruvida ett eller flera specifika krav var uppfyllda.

5.3 Gemensamma erfarenheter

I detta avsnitt diskuteras gruppens gemensamma erfarenheter som gruppen upplevt, såsom samarbete i grupp, kundkontakt, och olika arbetsprocesser.

5.3.1 Samarbete i grupp

Gruppen jobbade konsekvent genom projektet i grupp för att stödja varandra. Gruppen hade ofta gemensamma möten för hela gruppen där man stämde av hur arbetet har gått och om någon behöver hjälp med något.

För att möjliggöra samarbete ihop oftare bestämde gruppen regelbundet tider där man kunde sitta på distans med varandra i kommunikationsverktyget Discord

5.3.2 Kundkontakt

Gruppen hade en analysansvarig som hanterade all kommunikation med kund. Den huvudsakliga kundkontakten skedde på inplanerade möten. Gruppen upplevde att det lönade sig att ha många möten tidigt i projektet eftersom det gjorde att kravspecifikationen kunde verifieras ordentligt och att gruppen kände till och förstod alla kundens krav och önskemål på projektet.

5.3.3 Kommunikation

Gruppen kommunicerade kontinuerligt via Discord. Detta gjorde att gruppen inte var låst vid specifika tider att arbeta på utan kunde arbeta när möjlighet fanns. Det fanns också ett krav att se om det fanns några nya meddelanden att ta del av minst en gång varje dygn. Meningen med detta var att om gruppen behövde kommunicera om någonting viktigt skulle det inte ta för lång tid.

5.3.4 Arbetsgrupper

Varje sprint delades gruppen in i mindre arbetsgrupper. Dessa grupper blev sedan tilldelade ett antal arbetsuppgifter att utföra under sprinten. Detta minskade risken att någon inte visste vad de skulle göra eller hur det skulle göras.

5.3.5 Lärdomar till framtida projekt

TODO: Detta läggs till när arbetet är slutfört

5.4 Användartester

Nedan följer resultat från de användartester som utförts för att värdesätta det omarbete som gjorts på det grafiska gränssnittet. TODO: Dessa har inte utförts än.



6 Diskussion

I detta kapitel diskuterar ni era resultat. Här kan ni bygga resonemang på antaganden, dvs. man måste inte underbygga allt med fakta eller referenser till annat arbete. Det är fullt möjligt att vara kreativ här.

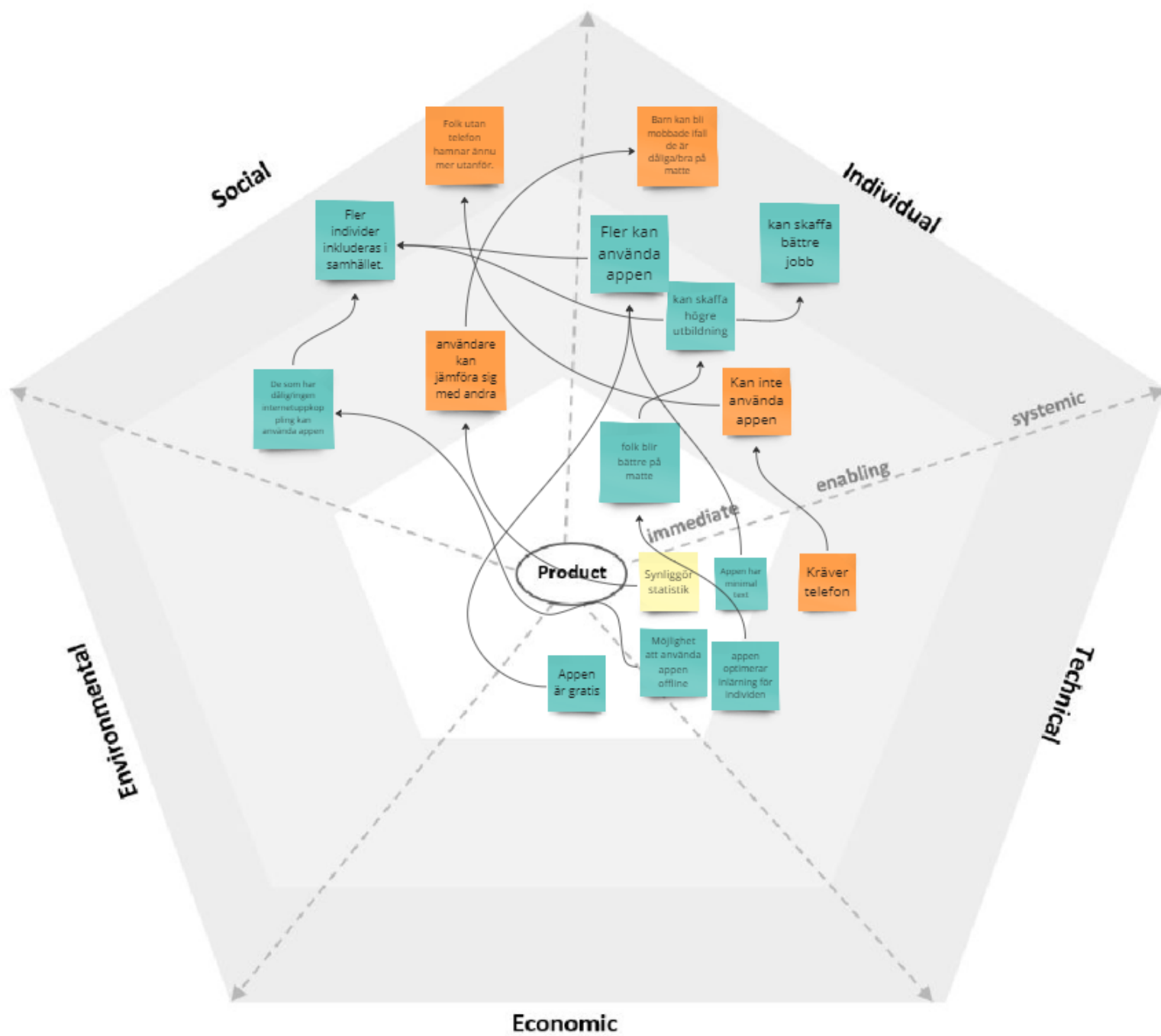
6.1 Resultat

6.1.1 Alternativ implementation

TODO: skriv nåt om alternativa verktyg, ramverk osv / implementationsstrategier

6.2 Metod

6.3 Arbetet i ett vidare sammanhang



Figur 6.1: SusAF: Hållbarhetsanalys-diagram

Under projektets gång hölls det ett seminarium där gruppen fick tillfälle att diskutera projektets påverkan på sin omvärld och vad man möjligtvis bör tänka på i hur man hanterar projektet. Dessa frågor diskuterades i form av fem olika perspektiv; social, individuell, teknisk, ekonomisk, och klimatpåverkan. Följande stycke diskuterar vad som diskuterades under seminariet och vad man kan ta med sig från detta. Analyserna baserades på SusAF, *Sustainability awareness framework* [23]. Resultatet av hållbarhetsanalysen presenteras i figur 6.1.

6.3.1 Social

Eftersom applikationen som utvecklas under projektet är ämnad för barn är de sociala konsekvenserna främst inriktade på barns individuella liv och deras upplevelser i till exempel skolan med sina kompisar. Då applikationen gärna inriktar sig på länder där utbildningssystemet inte är bra och extra utbildning behövs, vilket ofta är i utvecklingsländer, finns det risk att inte alla har en telefon. Detta såg gruppen som en risk när det gäller det sociala då dessa personer inte har möjlighet att använda applikationen och kan då hamna utanför, i både sin kompisgrupp, men även i samhället om dess matematikkunskaper inte utvecklas till en bra nivå.

Det finns även positiva aspekter som diskuterades under seminariet, att applikationen är utvecklad så att det går att använda den utan internet gör att fler personer har potential att kunna använda den, vilket gör att fler personer kan inkluderas i samhället.

6.3.2 Individuell

Att den utvecklade applikationen är optimerad för att lära ut matematik på bästa sätt för individen som använder den gör att den som använder den blir bättre på matematik än vad den annars hade blivit. På grund av att applikationen inriktar sig mot yngre barn så är huvudtanken med den att dessa barn ska bli mer intresserade av matematik och få en bättre grundläggande kunskap. Förhoppningen är att detta ger barnen en större chans att kunna söka sig till högre utbildning i sina framtida liv. Detta har en stor påverkan på individen, dessa kan vara att de kan få en högre lön i framtiden eller att de kan få ett bättre jobb med fler friheter.

På seminariet diskuterades det även negativa aspekter. En negativ aspekt som togs upp var kopplad till det sociala. En av applikationens största utmaningar är att den ska kunna användas av så många som möjligt och en individuell begränsning till denna utmaning är att om den inte går att användas för att man inte har en telefon så kan man hamna utanför.

6.3.3 Teknisk

Under den tekniska aspekten finns det flera olika saker som diskuterades. Till exempel finns det samma begränsning som har diskuterats tidigare i detta dokument vilket är att det kräver en telefon. Denna begränsning kan leda till en del andra sociala och individuella konsekvenser.

Att applikationen har som mål att använda så lite text som möjligt, vilket är en teknisk kompetens inom applikationen, är en aspekt som leder till flera positiva individuella och sociala aspekter. Främst att fler personer kan använda applikationen. Att den dessutom är utvecklad så att den går att använda utan tillgång till internet samt att den är optimerad för inlärning av individen leder även de till olika sociala och individuella konsekvenser.

När applikationen används sparas data i databasen som kan användas som forskningsunderlag för vidareutveckling av applikationer i liknande slag, och undervisning av matematik rent generellt. Denna forskning

6.3.4 Ekonomisk

Ur det ekonomiska perspektivet har applikationen potential att bidra med en mer resurseffektiv undervisning i matematik genom applikationens algoritm. Den skräddarsydda undervisningen för eleverna sparar på pengar i form av framtida resurser om nu eleven inte skulle lära sig automatiseringen helt och hållet, vilket på sikt kan leda till en lidande matematisk kompetens som kan gå ut över andra ämnen och senare matematikstudier.

Dessutom sparar applikationen både på fysiskt material i form av papper, pennor och dylikt. att applikationen även är gratis är ett stort steg i att erbjuda en likvärd, kvalitativ matematisk undervisning till elever oavsett socioekonomisk bakgrund vilket bidrar till en hållbar samhällsutveckling.

6.3.5 Klimatpåverkan

Att undervisningen sker digitalt i smartphones och tablets är något som är bra för miljön och naturen eftersom att dessa elektroniska enheter kan laddas med förnybart el och på så sätt lämna ett minimalt avtryck på miljön. Klassiska stenciler med papper och penna har kollektivt, över hela landet betydligt mer påverkan på miljön genom nödvändigheten att avverka träd för att producera pappersmassa, transport till fabrik och sedan till butik. På sikt kan det ha en positiv klimatpåverkan att använda applikationen för den matematiska undervisningen.

Eftersom denna applikation är mobil så kräver den någon slags elektronisk enhet för att kunna användas. Att detta sker är bra för miljön men har även en del baksidor. Elektroniskt avfall är dåligt för miljön och många länder vet inte vad de ska göra med allt avfall som produceras varje år. Detta gör att många industriländer betalar utvecklingsländer för att få dumpa sitt avfall hos dem. Industriländer skeppar varje år hundratusentals ton elektroniskt avfall och detta är någonting man bör ha koll på när man jobbar mot digitalisering av undervisning. [24]

6.3.6 Sammanställning

Med tanke på dessa aspekter finns det mycket olika positiva konsekvenser som denna applikation kan bidra med om den används i tillräckligt hög utsträckning. Främst gäller dessa aspekter att barn kan lära sig bättre matematik och på det sättet tillföra nytta och värde till samhället när de blir äldre och kommer in i sina framtida utbildningar med bättre matematikkunskaper.

De negativa aspekterna som detta projekt kan ha som påverkan syftar främst till saker som begränsar människor att använda denna applikation. Saker som att människor talar olika språk och att inte alla har tillgång till en telefon är begränsande för användandet av applikationen.



7

Slutsatser



Litteratur

- [1] Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O'Hara och Dan Dixon. "Gamification. using game-design elements in non-gaming contexts". I: *CHI '11 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '11. Vancouver, BC, Canada: Association for Computing Machinery, 2011, s. 2425–2428. ISBN: 9781450302685. DOI: 10.1145/1979742.1979575. URL: <https://doi.org/10.1145/1979742.1979575>.
- [2] Karolina Muhrman. "Inget klöver utan matematik: En studie av matematik i yrkesutbildning och yrkesliv". Diss. Linköping universitet, 2016.
- [3] Amy Claessens och Mimi Engel. "How Important is Where you Start? Early Mathematics Knowledge and Later School Success". I: *Teachers College Record* 115.6 (2013), s. 1–29. DOI: 10.1177/016146811311500603.
- [4] Mattias Sjögren, Simon Malm, Rebecka Hedin, Agnes Wendt, Anton Jacobaeus, Anton Sars Oom, Gustav Andersson och Luka Jankovic. "Optimalt lärande: Utvecklandet av en matematikapplikation för grundläggande aritmetik". sv. I: (2022).
- [5] Benjamin Sundvall, Oscar Wahllöf, Sebastian Olsson, Joakim Andersson, Simon Gisels-son, Amanda Sandberg, Michelle Galin, Parisa Khedri, Ture Wramner och Ludvig Jealmo. "Optimalt lärande : Spelifiering och visualisering av entalsaritmetik för barn". sv. I: (2023).
- [6] Ming-Shiou Kuo, Tsung-Yen Chuang, Yen-Ning Su, Tzu-Ying Chen och Bo-Yau Cheng. "What are the Better Gamification Tools for Elementary School Teachers?" I: *2018 7th International Congress on Advanced Applied Informatics (IIAI-AAI)*. 2018, s. 346–349. DOI: 10.1109/IIAI-AAI.2018.00074.
- [7] Google. *Paint your UI to life*. Hämtad 2024-03-05. URL: <https://dart.dev>.
- [8] Python software foundation. *Python*. Hämtad 2024-03-05. URL: <https://www.python.org>.
- [9] International Organization for Standardization (ISO). C++. Hämtad 2024-03-05. URL: <https://isocpp.org>.
- [10] Google. *Build apps for any screen*. Hämtad 2024-03-01. URL: <https://flutter.dev>.
- [11] Django software foundation. *Django*. Hämtad 2024-03-01. URL: <https://www.djangoproject.com>.

-
- [12] Gitlab. *Software. Faster*. Hämtad 2024-03-01. URL: <https://about.gitlab.com>.
 - [13] Google. *Android Studio*. Hämtad 2024-03-01. URL: <https://developer.android.com/studio>.
 - [14] Microsoft. *Visual Studio Code*. Hämtad 2024-03-08. URL: <https://code.visualstudio.com/docs>.
 - [15] Docker. *Docker*. Hämtad 2024-03-08. URL: <https://www.docker.com/resources/what-container/>.
 - [16] Overleaf. *Overleaf*. Hämtad 2024-03-08. URL: <https://www.overleaf.com/about/why-latex>.
 - [17] Google. *Google Drive*. Hämtad 2024-03-08. URL: <https://www.google.com/drive/>.
 - [18] Figma. *Figma*. Hämtad 2024-03-08. URL: <https://www.figma.com/design-overview/>.
 - [19] Discord Inc. *Discord*. Hämtad 2024-03-08. URL: <https://discord.com/company>.
 - [20] Microsoft. *Microsoft Teams*. Hämtad 2024-03-08. URL: <https://www.microsoft.com/sv-se/microsoft-teams/group-chat-software>.
 - [21] Ken Schwaber och Jeff Sutherland. *The Scrum Guide*. <https://scrumguides.org/scrum-guide.html>. [Online; refererad februari 15, 2024]. 2020.
 - [22] Mattias Arvola. *Interaktionsdesign och UX: om att skapa en god användarupplevelse*. Studentlitteratur AB, 2014.
 - [23] Leticia Duboc, Birgit Penzenstadler, Jari Porras, Sedef Akinli Kocak, Stefanie Betz, Ruzanna Chitchyan, Ola Leifler, Norbert Seyff och Colin Venters. "Requirements engineering for sustainability: an awareness framework for designing software systems for a better tomorrow". I: *Requirements Engineering* 25 (dec. 2020). DOI: 10.1007/s00766-020-00336-y.
 - [24] Utrikesmagasinet. *E-skrot både miljöbov och inkomstkälla*. Hämtad 2024-03-07. URL: <https://www.ui.se/utrikesmagasinet/analyser/2022/juli/e-skrot-bade-miljobov-och-inkomstkalla/>.