



# CHALMERS

---

## Positionsbaserad quiz i ett hälsofrämjande spel för Android

Examensarbete inom Data- och Informationsteknik

ELENA MARZI

EXAMENSARBETE

**Positionsbaserad quiz i ett  
hälsofrämjande spel för Android**

ELENA MARZI

Institutionen för Data- och Informationsteknik  
CHALMERS TEKNISKA HÖGSKOLA  
GÖTEBORGS UNIVERSITET

Göteborg 2019



**Positionsbaserad quiz i ett hälsofrämjande spel för Android**  
ELENA MARZI

© ELENA MARZI, 2019

Examinator: Peter Lundin  
Handledare: Jan Jonsson

Institutionen för Data- och Informationsteknik  
Chalmers Tekniska Högskola / Göteborgs Universitet  
412 96 Göteborg  
Telefon: 031-772 1000

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Institutionen för Data- och Informationsteknik  
Göteborg 2019

## SAMMANFATTNING

Att vidareutveckla en Android-baserad prototyp, vars syfte är att främja fysisk motion hos människor i Lindängen, Malmö, har varit det övergripande målet med detta examensarbete. Utvecklingen har resulterat i en digital tipspromenad där spelarna kan svara på frågor medan de motionerar utomhus.

Under spelets gång ser spelaren sin egen samt nästa frågas positioner på en karta så att frågan går att nå och besvaras. Användaren kan också skapa quizzar vars frågor är knutna till geografiska positioner, genom att rita en promenadslinga på en karta som visas på skärmen. Frågorna placeras ut av systemet jämnt utspridda längs linjen och går sedan att flytta. En kopieringsfunktionalitet har också adderats till prototypen, där användaren kan kopiera en redan existerande quiz, inklusive promenadslinga och frågor.

Prototypen, som är skriven i Java, har realiserats i utvecklingsmiljön Android Studio. Ramverket ArcGIS Runtime SDK har använts för att visualisera och behandla geografisk information och har varit en avgörande faktor för arbetets framgång. Lösningen har också i viss utsträckning berört redan existerande databas och API driftsatta i molntjänsten Microsoft Azure.

**Nyckelord:** Android, Android Studio, ArcGIS, Esri, geokodning, GIS, Google Maps, hälsa, Java, motion, positionering, spel, tipspromenad, quiz



## ABSTRACT

The overall goal of this thesis project has been to develop an Android-based prototype, whose purpose is to promote physical activity for the people of Lindängen, Malmö. The implementation has resulted in a digital quiz walk game, where players answer questions while exercising outdoors.

During the game, the positions of the player and the next question are displayed on a map, so that the player can reach the question and answer it. In addition, users can create new quizzes with questions linked to geographical positions. A walking trail can easily be drawn on a map, the questions are placed automatically by the system evenly spread out along the trail, and can then be moved. A copy functionality has also been added to the prototype, so that users can make copies of existing quizzes, including the walking trail and the questions attached to it.

The prototype, which is developed in Java, has been realized using the Android Studio development environment. The ArcGIS Runtime SDK framework has played an important role for handling and visualizing geographical information. Moreover, the solution has included modifications to a database and an API, already deployed in the Microsoft Azure cloud service.

**Keywords:** Android, Android Studio, ArcGIS, Esri, exercise, game, geocoding, GIS, Google Maps, health, Java, positioning, quiz, quiz walk





## **FÖRORD**

Tack till Jenny Forsberg, Anna Funke och Mehdi Khayyami på Semcon för att ha gett mig möjlighet att delta i ett samhällsviktigt projekt. Ett varmt tack även till min handledare Jan Jonsson för uppmuntran, vägledning och tålamod under projektets gång.

Författaren, Göteborg, 29 maj 2019



## FÖRKORTNINGAR

<b>API</b>	Application Programming Interface
<b>APK</b>	Android Package
<b>Esri</b>	Environmental Systems Research Institute
<b>GIS</b>	Geographic Information System
<b>GNSS</b>	Global Navigation Satellite System
<b>GPS</b>	Global Positioning System
<b>HTTP</b>	HyperText Transfer Protocol
<b>JDBC</b>	Java DataBase Connectivity
<b>JSON</b>	JavaScript Object Notation
<b>REST</b>	REpresentational State Transfer
<b>SDK</b>	Software Development Kit
<b>SQL</b>	Structured Query Language
<b>UX</b>	User eXperience
<b>XML</b>	eXtensible Markup Language

## BEGREPPSLISTA

**Adobe XD** – Verktyg för att skapa prototyper och design av användargränssnitt [1].

**Android** – Linux-baserat operativsystem, främst för mobila enheter [2].

**Android Studio** – Utvecklingsmiljö som används för att skapa applikationer för det mobila operativsystemet Android [3].

**APK Analyzer** – analysverktyg, inbyggt i Android Studio, för att granska installationsfiler för Android-applikationer [4].

**Application Programming Interface (API)** – Gränssnitt som gör möjligt kommunikation mellan olika delar av ett större system, som till exempel en databas och ett användargränssnitt [5].

**ArcGIS Runtime SDK** – utvecklingsramverk från Esri för att visualisera och behandla geografisk information [6].

**Azure** – molntjänst från Microsoft för att skapa servertjänster som automatiskt kan installeras i Microsofts serverparker [7].

**Azure SQL** – en särskild molnbaserad variant av databasmotorn Microsoft SQL Server [8].

**Backend** – delar av ett större mjukvarusystem som lagrar och hanterar data och som endast indirekt påverkas av användares aktioner.

**Frontend** – den del av ett större mjukvarusystem som användare direkt interagerar med.

**Geokodning** – omvandling av en adress eller en namngiven plats till geografiska koordinater [9].

**GIS** – datoriserade informationssystem med funktioner för insamling, lagring, bearbetning, analys och visualisering av geografiska data [10].

**Git** – öppen programvara som används för att hantera olika versioner av programkod och som underlättar samarbete mellan flera utvecklare [11].

**Google Maps SDK** – utvecklingsramverk från Google för att visualisera och hantera kartor samt använda API:er från Google [12].

**Google Play Services** – programpaket från Google som kan användas i utveckling av Android Applikationer [13].

**HTTP** – ett applikationsprotokoll för klient-server-lösningar som bygger på requests och responses som klienten och server skickar till varandra. Det är grunden för all data-kommunikation över World Wide Web [14].

**Java** – objektorienterat språk för alla typer av mjukvaruutveckling som till exempel klient-server-lösningar, inbyggd mjukvara och Android-applikationer [15].

**JDBC** – ramverk för att ansluta till SQL-databaser genom en uppsättning standardiserade Java-API:er [16].

**Microsoft SQL Server Management Studio** –utvecklingsplattform från Microsoft som används för att hantera olika databastjänster, till exempel i molntjänsten Azure [17].

**REST** – HTTP-baserat protokoll för att hämta, uppdatera data eller radera data via ett API [18].

**Software development kit** – uppsättning av programbibliotek och utvecklingsverktyg som möjliggör utveckling av mjukvara för en specifik plattform.

**SQL** – språk som används för att skapa, uppdatera och läsa ut data från tabeller i relationsdatabaser. Språket finns i flera olika varianter, till exempel Transact-SQL som används i Microsofts databasprodukter.

**Trello** – webbaserad tjänst för att hantera projekt och organisera uppgifter [19].

**Wireframe** – enklare prototypskiss som används för att strukturera innehållet av en web or mobil applikation.

# Innehåll

<b>Sammanfattning .....</b>	<b>iv</b>
<b>Abstract .....</b>	<b>vi</b>
<b>Förord.....</b>	<b>viii</b>
<b>Förkortningar .....</b>	<b>x</b>
<b>Begreppslista .....</b>	<b>xi</b>
<b>1. Inledning.....</b>	<b>2</b>
1.1 Projektbeskrivning .....	3
1.2 Examensarbetets förutsättningar .....	4
1.3 Syfte .....	6
1.4 Mål .....	7
1.5 Avgränsningar .....	8
<b>2. Metod .....</b>	<b>10</b>
2.1 Arbetsverktyg.....	10
2.2 Arbetsmetod .....	12
<b>3. Genomförande .....</b>	<b>15</b>
3.1 Positionsbaserade quizzar .....	15
3.1.1 Undersökning av olika ramverk .....	15
3.1.2 Komplettering av databasen .....	18
3.1.3 Implementation av att skapa quiz .....	19
3.1.4 Implementation av att spela quiz .....	26
3.2 Kopiering av quiz.....	29
3.2.1 Implementation.....	30
3.2.2 Förändring av databas.....	31
<b>4. Resultat.....</b>	<b>32</b>
4.1 Positionsbaserade quizzar .....	32
4.2 Kopiering av quiz.....	35
<b>5. Diskussion.....</b>	<b>37</b>
5.1 Positionsbaserade quizzar .....	37
5.2 Kopiering av quiz.....	38
5.3 Framtida möjligheter till vidareutveckling .....	39
5.4 Samhällsnytta.....	40
<b>6. Slutsatser .....</b>	<b>41</b>
<b>7. Referenser .....</b>	<b>42</b>

# 1. Inledning

Sambandet mellan socioekonomiska förhållanden och hälsa är väl etablerat i sociologiska studier [20]. Världshälsoorganisationen (WHO) har identifierat samhällsklass, utbildningsnivå, samt ekonomiskt och socialt välbefinnande som direkta eller indirekta hälsomarkörer på global, nationell och lokal nivå [21].

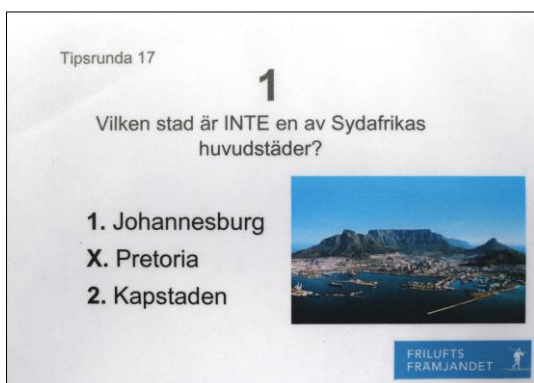
En undersökning utförd på uppdrag av Göteborgs Stad har tydligt visat på geografiskt relaterade hälsoskillnader mellan olika stadsdelar: invånare i stadsdelar med sämre socioekonomiska faktorer rapporterar i högre grad en dålig självupplevd hälsa än i övriga stadsdelar [22]. Motsvarande studie genomförd i Malmö visar på liknande resultat mellan Malmös olika stadsdelar, där till exempel 17 % av invånarna i den särskilt utsatta stadsdelen Rosengård anser sig ha dålig hälsa, medan samma siffra för det mer välbeställda området Limhamn är 4 % [23].

Hälsoklyftor mellan olika stadsdelar i Malmö har uppmärksammats av Malmö Universitet, som år 2017 sökte finansiering från Sveriges innovationspolitiska myndighet Vinnova för ett hälsofrämjande projekt. Projektet heter "Hälsofrämjande innovation i samverkan" och går ut på att etablera en plattform för innovation och samarbete mellan invånare i det särskilt utsatta området Lindängen, universitetet, samt offentlig och privat sektor [24].

Inom ramarna för projektet har ett antal studiegrupper organiserats i Malmö, vilka har fokuserat på olika hälsoaspekter [25]. "Trygghet i omgivningen" och "Utegymp / Fitness justice" syftar till att förhöja jämlikhet och jämställdhet, öka trygghet och kunskap om det egna området samt främja motion, bland annat genom att använda digital och mobil teknik som hjälpmedel [26].

## 1.1 Projektbeskrivning

En av de privata aktörerna som är involverade i projektet är teknikföretaget Semcon. En del av deras uppdrag består av att utveckla en prototyp av en Android-baserad applikation, som skall likna en tipspromenad<sup>1</sup>.



Figur 1 Exempel på tipspromenadfråga på papper [27].

Applikationen, som går under arbetsnamnet *Hälsoslingan*, är tänkt att vara en elektronisk variant av en klassisk tipspromenad (se Figur 1) som i första hand riktar sig till barn i skolåldern. Frågor ska placeras ut längs en digital promenadslinga, visualiseras på spelarens mobiltelefon och besvaras genom att välja mellan olika svarsalternativ.

Enligt den ursprungliga idén skulle spelaren som svarar rätt samla virtuella belöningar, som till exempel diamanter, stjärnor eller poäng. Tack vare detta spelliknande belönings-system skulle målgruppen på så sätt uppmuntras att röra på sig mer och förbättra sin fysiska hälsa, samt öka sin allmänbildning och kunskap om området.

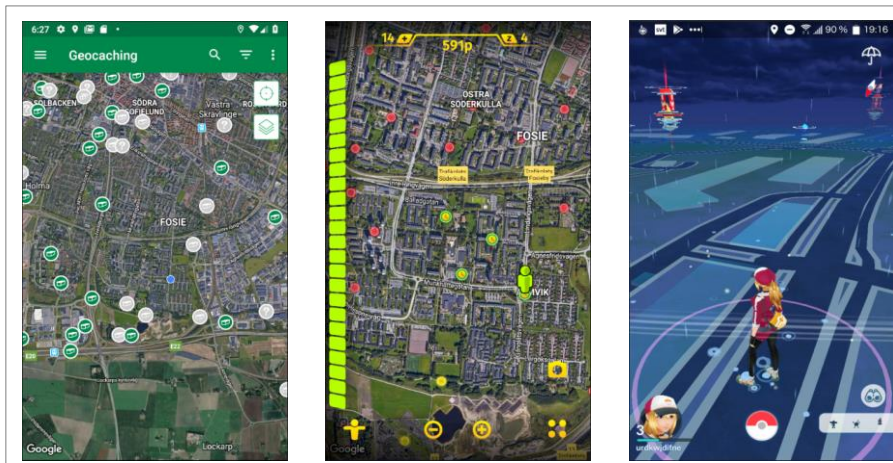
Att använda staden som spelplats är inget nytt koncept [28] och många liknande exempel kan nämnas. *Geocaching*, som har funnits sedan år 2000, är en spelliknande aktivitet där deltagare världen över kan leta efter verkliga "skatter", som oftast är små obetydliga saker, och dela med sig av sina fynd med andra likasinnade [29].

---

<sup>1</sup> Enligt Svenska Akademin Ordbok (SAOB) är en tipspromenad en "tävling i form av promenad längs (snitslad [eller] på annat sätt utmärkt) bana med utplacerade frågor som besvaras på medförd blankett" [56]



Ett svenskt exempel är det platsbaserade Android-spelet *Turf* från 2010 som låter spelare kämpa om att ta över virtuella zoner genom att besöka dem i den fysiska verkligheten [30]. Det populära spelet *Pokémon GO* såg dagens ljus 2016 och bygger på samma princip: spelaren måste ta sig till olika områden där det finns monster (Pokémon) för att fånga dem [31].



Figur 2 Skärmbilder av spelen Geocaching, Turf och Pokémon GO.

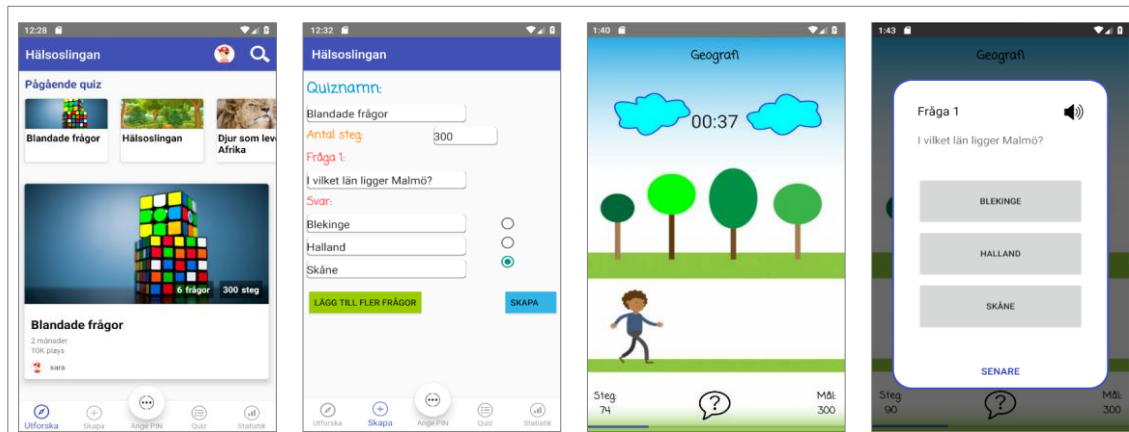
Undersökningar har nyligen visat på positiva fysiska hälsoeffekter denna typ av mobila spel har på individ- och samhällsnivå [32] [33]. De individer som har spelat Pokémon GO, till exempel, har ökat sin dagliga fysiska motion, ökat sin exponering för solljus samt höjt sin grad av interaktioner utomhus med andra människor [34]. Dessutom har en studie som följt ett stort antal individer under flera år visat att även en liten ökning av den fysiska aktiviteten bidrar till mätbara förbättringar av allmänhetens hälsonivå [35].

Hälsoslingan skulle på samma sätt kunna bidra till en förbättring av livskvaliteten hos Lindängens befolkning. För att bekräfta grundtanken ska prototypen testas med olika målgrupper, bland annat i skolor i Lindängen, och i en senare fas tjäna som konceptvalidering med målgruppen inför framtida utveckling.

## 1.2 Examensarbetets förutsättningar

Vid examensarbetets start var viss funktionalitet i prototypen redan implementerad. En första variant av en promenadslinga byggde på att frågor dök upp efter ett visst antal steg,

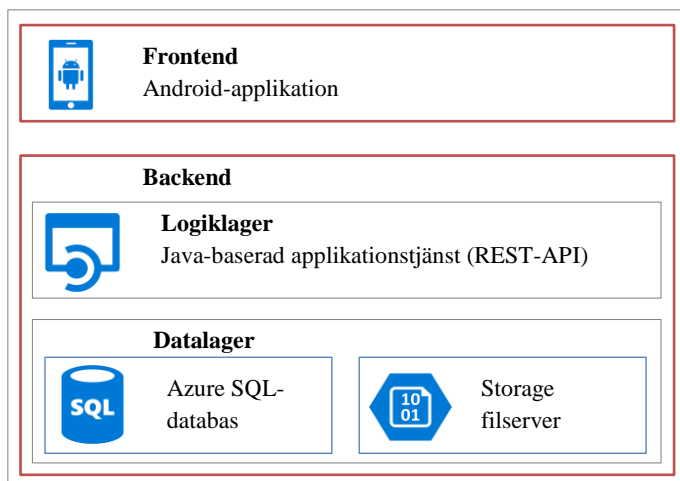
som detekterades av stegräknarsensorn som finns inbyggd i operativsystemet Android [36].



Figur 3 Skärmbilder av existerande funktionalitet vid examensarbetets start: startsidan, vyn för att skapa quizzar med antal steg, vyn för att spela en quiz och vyn som visar frågorna.

Som visas i Figur 3 kunde användare alltså skapa en quiz med ett totalt antal steg att gå och spela den under en promenad. Frågorna visades sedan med ett visst antal stegs mellanrum och i fall av felaktigt svar, hade spelaren alltid möjlighet att göra ett nytt försök. En startsida fanns också redan implementerad och visade en lista med alla quizzar som hade skapats hittills.

Applikationen kommunicerade med en databas och en lagringsyta för bilder via en API-server. När en quiz skapades, lagrades den i databasen och när quizzen spelades, hämtades dess frågor och svarsalternativ tillsammans med quizzens bakgrundsbild för att visas i användargränssnittet. Lösningens hela backend, det vill säga databas, lagringsyta och API, fanns också installerade i Microsofts molnlösning Azure.



Figur 4 Systemets övergripande arkitektur.

När examensarbetet påbörjades saknades i prototypen funktionalitet för att kunna skapa positionsbaserade quizzar tillsammans med annat, som exempelvis möjligheten att kopiera en redan existerande quiz, vilket uttryckts som en önskan från målgruppen.

### 1.3 Syfte

Eftersom fysisk aktivitet och motion utomhus är den centrala aspekten av detta projekt [25], är syftet med detta examensarbete att utöka den redan existerande prototypen för att kunna stimulera målgruppen att röra på sig mer.

Följaktligen består den första och viktigaste uppgiften för examensarbetet av att undersöka och implementera en positionsbaserad quiz som ska likna en tipspromenad, utöver den redan existerande stegräknarbaserade mekanismen. Den andra uppgiften för examensarbetet är att implementera en kopieringsfunktionalitet, som skall ge möjlighet för användaren att utgå från andras quizzar och anpassa dem för egna behov. Till exempel skulle en lärare kunna återanvända en kollegas quiz för sina egna elever.

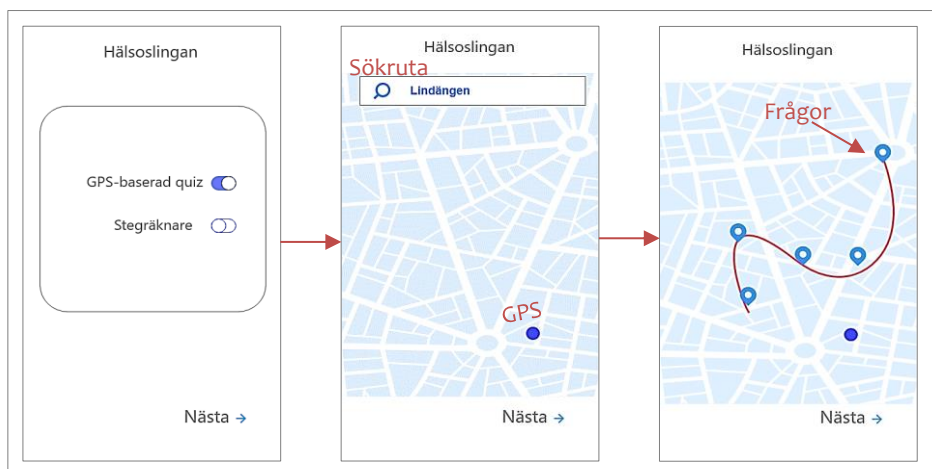
## 1.4 Mål

Målet med examensarbetet är alltså att undersöka, utveckla och leverera två olika delar av prototypen.

### Positionsbaserade quizzar

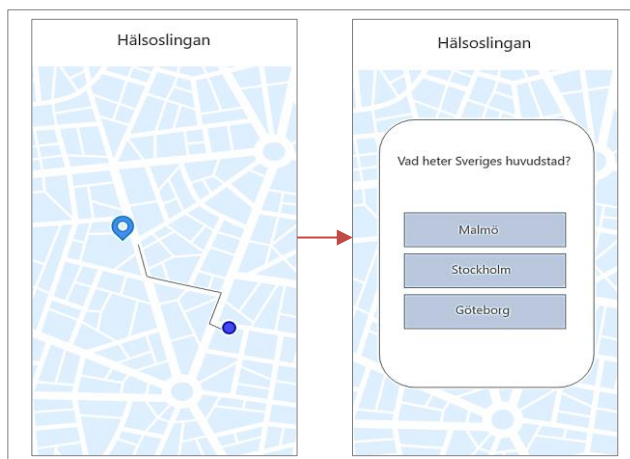
En digital positionsbaserad promenadslinga ska utvecklas. Användaren som skapar en quiz ska först kunna välja mellan att använda stegräknaren eller den positionsbaserade varianten. Om det senare alternativet väljs ska användaren kunna utgå från sin nuvarande position eller söka efter en plats på kartan för att kunna identifiera området där promenadslingan ska placeras.

Sedan ska en linje kunna ritas på kartbilden och frågor ska förläggas av systemet jämnt utspridda längs linjen knutna till geografiska positioner. Därefter ska användaren kunna se frågorna på kartan, flytta dem manuellt längs linjen, inspektera innehållet samt radera dem. Ett exempel av en sådan mekanism visas i Figur 5.



Figur 5 Wireframes som visar ett exempel på den ursprungliga idén, där användaren kan välja mellan positionsbaserad quiz och stegräknare, söka efter en plats, se sin egen position, rita en slinga med frågor som jämnt sprids ut längs den ritade linjen.

Användare ska också kunna spela positionsbaserade quizzar. En vy med en kartbild ska då visualiseras där spelarens och nästa frågas position ska visas som i Figur 6. Kartbilden ska uppdateras kontinuerligt efter hand som spelaren flyttar på sig och när spelaren tar sig fram till nästa frågas position ska den presenteras med sina svarsalternativ. De frågor som besvarats felaktigt ska presenteras på nytt i slutet av promenadslingan.



Figur 6 Wireframes som visar vad som ska hända när spelaren spelar. Spelaren ska kunna ta sig till en fråga och besvara den.

För att kunna realisera det beskrivna målet ska olika ramverk för utveckling av kartbaserade användargränssnitt för Android-plattformen analyseras noggrant. De ramverk som ska jämföras är *Google Maps SDK* från *Google* och *ArcGIS Runtime SDK* från *Esri*. Resultatet av undersökningen ska dokumenteras och det mest lämpliga alternativet ska användas för implementationen.

### Kopiering av quiz

Användaren ska ges möjlighet att skapa kopior av redan existerande quizzar. När en quiz kopieras ska den visas i en vy för att skapa quiz, men med alla delar ifyllda, så att användaren kan välja mellan att behålla exakt samma frågor och svar eller förändra dem. Funktionaliteten ska ge möjlighet att kopiera en quiz som skapats med stegräknare eller med positionsbaserade frågor. I det senare fallet ska applikationen kopiera även positionen och låta användaren redigera frågor och flytta frågorna till andra positioner.

## 1.5 Avgränsningar

I examensarbetet ingår inte:

- att implementera inloggning eller användarregistrering i applikationen, inklusive vyer knutna till inloggad användare
- att göra det möjligt att lägga till bilder till quizzar
- att samla in, behandla eller implementera vyer för att visa statistik
- att implementera olika sätt för att visualisera frågor

- att utveckla ett webbgränssnitt för att administrera, skapa eller spela quiz
- att utforska eller lösa eventuella säkerhetsproblem (kryptering, signering eller andra säkerhetsaspekter)
- att göra datainsamling, användartester eller intervjuer på plats i Malmö stad
- att implementera vyer utöver de som nämnts specifikt

## 2. Metod

Under examensarbetet har vissa verktyg använts för att kunna skapa och utveckla de olika delarna av systemet och dessutom har projektet följt så kallade agila principer för det praktiska genomförandet.

### 2.1 Arbetsverktyg

Eftersom utvecklingen av prototypen redan hade påbörjats vid examensarbetets start var vissa tekniska verktyg redan fastställda på applikationsnivå såväl som på API- och databasnivå.

#### Java

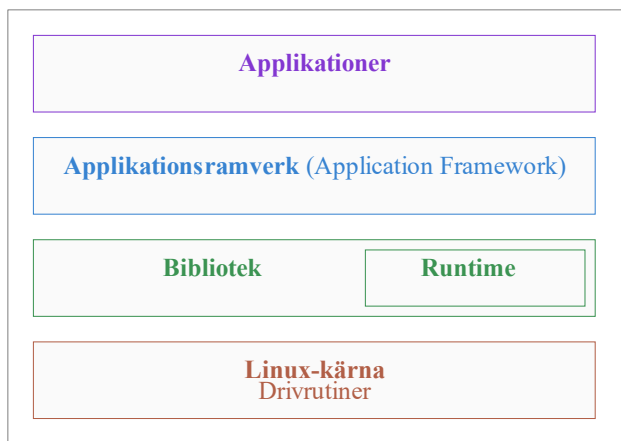
Både API:et och applikationen var från början skrivna i Java, vilket är ett objektorienterat programspråk som vanligtvis är plattformsoberoende [37]. Detta betyder att program skrivna i Java kan exekveras på olika kombinationer av operativsystem och hårdvara, så länge en virtuell Java-maskin finns tillgänglig [38]. I projektet har API:et utvecklats i operativsystemet Windows för att sedan driftsättas i en Linux-miljö i Azure, medan applikationen har implementerats för operativsystemet Android.

#### Android

Android är dock mer än bara ett operativsystem. Det är en komplex plattform för mobila enheter med olika abstraktionsnivåer som lager efter lager bygger upp en stack [39] som illustreras i Figur 7. Den lägsta nivån består av en variant av Linux-kärnan inklusive drivrutiner<sup>2</sup>.

---

<sup>2</sup> Drivrutiner är mjukvara som kommunicerar direkt med ett datorsystems hårdvara. Drivrutiner är modulära, vilket betyder att tillverkaren av en enhet kan välja vilka drivrutiner som ska inkluderas i kärnan på varje typ av enhet. [57]



Figur 7 Visuell beskrivning av hur Android-plattformen är uppbyggd.

I detta examensarbete har vissa drivrutiner varit nödvändiga för utveckling av den geografiskt baserade funktionaliteten. Drivrutiner för GNSS-mottagare, trådlöst nätverk och mobilt nätverk har använts för positionsbestämning, samt magnetometer och accelerometer för beräkning av kompassriktning. Drivrutinerna har inte kontaktats direkt från applikationens programkod, utan via ett gränssnitt i ett högre abstraktionslager (*Application Framework*) [40].

För att kunna utveckla applikationen har *Android SDK* använts. Detta innehåller nödvändiga arbetsverktyg för att kompilera och felsöka programkoden samt simulera en Android-baserad mobil enhet [41].

Applikationer i Android bygger på ett samspel mellan bland annat aktiviteter (*activities*), tjänster (*services*) och mottagare (*broadcast receivers*). Varje aktivitet motsvarar konceptuellt ett användningsfall, som till exempel "skapa en quiz" eller "spela en quiz", är fristående från övriga aktiviteter och implementeras i programkod med klasser som ärver från *Activity*. Varje aktivitet kan starta andra aktiviteter vid behov och kan i sin tur innehålla ett eller flera fragment som identifierar delar av ett användningsfall [42]. Varje fragment representeras med en klass som ärver från klassen *Fragment*.

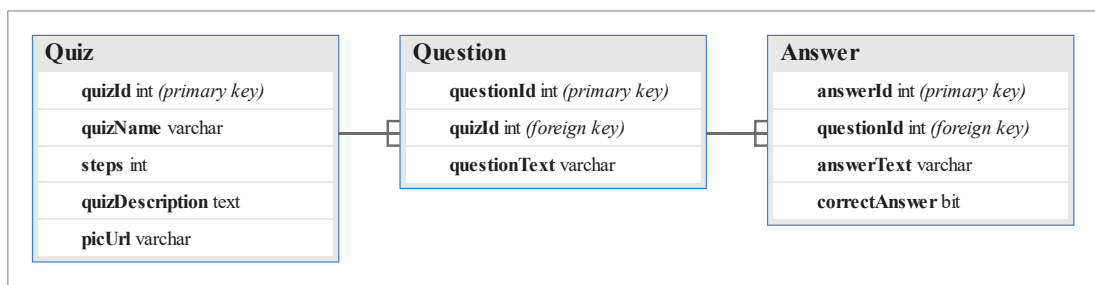
Utan att gå in för djupt i detalj har samspelet mellan *Activity* och *Fragment* varit viktigt för att kunna skapa applikationens användargränssnitt. Exempelvis fanns ursprungligen i kodbasen en aktivitet ansvarig för att visa den första vyn som möter användaren när applikationen startar, och motsvarar användningsfallet "visa startmenyn med en lista med quiz". Denna aktivitet startade i sin tur en annan aktivitet vilken representerar



användningsfallet "skapa quiz" som ursprungligen innehöll ett enda fragment, där användaren kunde mata in all information för en ny quiz.

## Databas

När det gäller databasen var den också redan implementerad och, tillsammans med API:et och en filserver för att lagra bilder, driftsatt i molntjänsten Microsoft Azure. I databasen fanns all information om samtliga spelbara quizzar, uppdelade i tre tabeller: Quiz, Question och Answer. Tabellerna för databasen hade skrivits i SQL med hjälp av verktyget Microsoft SQL Server Management Studio.



Figur 8 Databasen i startläget.

Slutligen är det värt att nämna att för att kunna skapa de ursprungliga ritningarna av applikationens användargränssnitt (*wireframes*) har programmet *Adobe XD* använts. Detta har underlättat förståelsen för hur applikationen till slut skulle se ut, samt har varit nödvändigt i kommunikationen med UX-experter på företaget och övriga programmerare i projektet.

## 2.2 Arbetsmetod

Under arbetets gång har utvecklingen brutits ner i två distinkta väldefinierade delar, som var och en för sig har utgjort ett separat delprojekt inom ramarna för prototypens vidareutveckling. Den första delen, som har bestått av positionsbaserade quizspel, var ett absolut krav från företagets sida för att de över huvud taget skulle kunna leverera ett värde till projektet, medan den andra delen bestämts skulle utföras i mån av tid och var mindre väsentlig.

De två delarna har behandlats sekventiellt och en viss mängd förarbete har genomförts för att kunna lägga grunden för utveckling av varje del. Olika utvecklingsramverk har utvärderats samt en mindre utökning av databasen och API:et har genomförts.

Agila arbetssätt har tillämpats under examensarbetets gång. Varje vecka bestod av en så kallad *sprint*, vilket är en tidsperiod där ett lag samarbetar för att leverera förbättringar eller utveckling av ny funktionalitet i ett leveransfärdigt tillstånd [43]. I detta fall har sprintar huvudsakligen inneburit veckovisa avstämningsmöten för att diskutera veckans framsteg, dock för att bestämma målet inför nästa vecka.

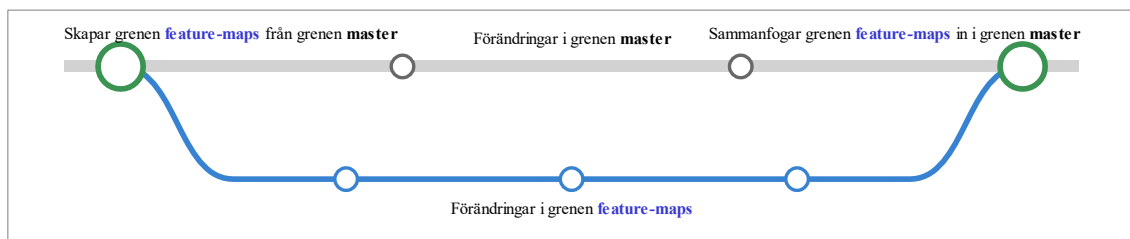
En digital uppgiftstavla från tjänsten Trello har använts för att organisera arbetet (se Figur 9). I kolumnen kallad *backlog* har övergripande uppgifter placerats som kallas *epic* och som motsvarar de två huvudområdena för examensarbetet och har varsin färgkod. I den andra kolumnen har varje epic brutits ner i mindre delar kallade *user stories*. En user story beskriver på en icke-teknisk nivå vilket värde som ska levereras till slutanvändarna under en sprint [44].



Figur 9 Uppgiftstavla i tjänsten Trello.

Varje user story har i sin tur brutits ner i mindre uppgifter (*tasks*), som blivit tilldelade samma färgmarkering och placerats i den tredje kolumnen i väntan att påbörjas. Tasks och user stories har flyttats framåt i olika kolumner för att indikera framtida, pågående och avslutade uppgifter. När alla user stories tillhörande samma epic hade flyttats till sista kolumnen, kunde även den övergripande epic ses som färdig och försvinna från backloggen. På detta sätt har det varit enkelt att organisera arbetet på ett strukturerat sätt.

I linje med den agila arbetsprocessen har programmet Git används för att hantera olika versioner av källkoden. I början av projektet har koden lagrats i en huvudgren (*master branch*) som har agerat grund för alla andra kodvarianter, vilka har gett upphov till egna funktionsgrenar (*feature branches*). När en funktionsgren har testats och ansetts färdig, har det varit praxis för inblandade programmerare i projektet att sammanfoga den med huvudgrenen.



Figur 10 Exempel på hur funktionsgrenar används.

Denna process har gjort det möjligt att upprätthålla en historik av alla förändringar i programkoden och, vid behov, återställa programkod till ett tidigare tillstånd. Detta har varit en nödvändighet eftersom flera utvecklare har samarbetat i projektet och bidragit med olika separata funktioner.

### 3. Genomförande

De två olika uppgifterna som examensarbetet har bestått av har inneburit undersökning samt implementationsarbete av varierande omfattning. Gemensam förberedelse för båda uppgifterna har varit att aktiviteten "skapa quiz", som ursprungligen bara bestod av ett enda fragment, har delats upp i flera fragment: ett för att ange quizzens namn, ett för att mata in frågor med svarsalternativ och ett för att välja mellan stegräknarbaserad eller positionsbaserad quiz.

#### 3.1 Positionsbaserade quizzar

Implementationen av positionsbaserade quizspel resulterade i den största mängden arbete. Först har två möjliga ramverk jämförts för att kunna identifiera den mest lämpliga tekniska lösningen att använda, sedan har databasen och det redan existerande API:et utökats så att de kunde behandla positioner och slutligen har den valda lösningen implementerats i kodbasen i Android-applikationen.

##### 3.1.1 Undersökning av olika ramverk

Som förberedelse för den efterföljande utvecklingen har två mycket enkla applikationer implementerats fristående från den prototyp som examensarbetet ämnar förverkliga. Applikationernas enda funktionalitet har varit att visa en kartvy med möjlighet att zooma och förflytta kartbilden med hjälp av den mobila enhetens pekskärm, samt söka efter adresser. Dessa har tjänat som konceptvalidering för att testa den tekniska genomförbarheten, jämföra svårighetsnivån för utveckling samt utvärdera vilket ramverk som innehåller mest funktionalitet för att utföra de beskrivna uppgifterna. Ramverken som undersöktes var *Google Maps SDK*, och *ArcGIS Runtime SDK*.

##### Google Maps SDK

Google Maps SDK är inbyggt i *Google Play Services* och kommer förinstallerat på de flesta Android-enheter. Detta innebär att programkod för att visualisera kartor och utföra vissa enkla geografiska operationer redan finns tillgängligt för Android-applikationer och därmed inte behöver laddas ner och installeras [13].

För att skapa denna applikation har steg-för-steg-instruktioner på webbsidan för Google Maps SDK följts [45]. Först har en SDK för Google Play Services installerats i Android Studio genom att en referens till Googles nedladdningsserver adderats i en konfigurationsfil (*build.gradle*). Därefter har en förinstallerad mall som visar en karta (*Google Maps Activity*) valts som huvudsakligt användargränssnitt från de mallar som presenteras för utvecklaren när ett nytt projekt skapas. En API-nyckel har sedan registrerats på administrationsgränssnittet för Googles molnplattform. Nyckeln har använts för att kunna nå och använda tjänster på Google Maps servrar. En enkel kartvy har på detta sätt kunnat visas i en mobil enhet. En förutbestämd position var redan implementerad i programkoden, så att när applikationen startades, panorerade kartvyn dit. Kartan kunde zoomas och roteras med fingrarna.

Att söka efter en plats innebär att koordinater som identifierar platsen begärs av applikationen från en av Googles molntjänster. Först har en textruta och en sökknapp lagts till i användargränssnittet. Därefter har den sökta adressen skickats över nätet via en HTTP-förfrågan och svaret från tjänsten har tolkats för att kunna centrera kartvyn på den sökta positionen. En stor mängd egen programkod har behövt skrivas, samt två ytterligare externa ramverk inkluderas i projektet, för att kunna kommunicera med tjänsten och tolka svaret, eftersom ingen sådan funktionalitet fanns inbyggd i ramverket Google Maps.

### **ArcGIS Runtime SDK**

Den andra applikationen utvecklades med ArcGIS Runtime SDK som inte är inbyggt i Android, utan måste laddas ner från Esri och bäddas in i applikationen vid kompilering [46].

Även för ArcGIS följdes en steg-för-steg-guide, från Esris webbsida [47]. För att göra det möjligt att använda ramverket, har Esris nedladdningsserver lagts till i två av projektets konfigurationsfiler. I detta fall har ingen API-nyckel behövt registreras för att använda Esris molntjänster. Däremot har en ny kartvy från Esri behövt läggas till manuellt i en layout-fil, på grund av att endast Googles egna mallar finns inbyggda i Android Studio. Dessutom har en viss mängd programkod behövt skrivas för att kunna använda den nya kartvyn som huvudsakligt användargränssnitt. Som i fallet med Google Maps, var det då möjligt att zooma, panorera och rotera kartbilden med fingrarna.

Även om mekanismen för att söka efter en plats i stora drag liknar det som beskrivits för Google Maps, har platssökning inneburit mycket mindre programkod, eftersom klasser för att kommunicera med Esris servrar för geokodning finns inbyggt i ramverket [48].

```
// Skicka förfrågan till Esris geokodningsserver
locator = new LocatorTask(GEOCODE_SERVER_URL);
geocodeFuture = locator.geocodeAsync(addressSearched);

// Ta emot resultatet från servern
geocodeFuture.addDoneListener(() -> {
    List<GeocodeResult> geocodeResults = geocodeFuture.get();
    GeocodeResult topResult = geocodeResults.get(0);

    // Centrera kartvyn på den sökta platsen
    mapView.setViewpointAsync(topResult.getDisplayLocation());
});
```

Figur 11 Pseudokod som illustrerar hur geokodning fungerar med Esri ArcGIS Runtime SDK. Klasserna *LocatorTask*, *GeocodeResult* är inbyggda i ramverket. *LocatorTask* sköter kommunikation över HTTP medan *GeocodeResult* översätter svaret till en objektstruktur som kan användas i programkod.

## Jämförelse och val av ramverk

Tack vare den stora mängd information som finns tillgänglig på nätet har båda prototyperna varit relativt enkla att utföra. Visuellt har dessutom prototyperna varit mycket lika. För att välja ut rätt ramverk har dock även andra aspekter varit viktiga, som ramverkens storlek, mängden tillgänglig funktionalitet, samt den ekonomiska aspekten.

Efter kompilering av respektive applikation mättes storleken av dess installationspaket (APK-fil). Som visas i Tabell 1 blev den ArcGIS-baserade applikationen 28 766 kiB större än applikationen som utvecklats med Google Maps SDK, vilket är till fördel för Google Maps SDK.

Dessutom användes verktyget *APK Analyzer* för att granska antalet metoder som definierats i samtliga klasser i de båda ramverken. Totalt innehåller ArcGIS Runtime SDK 10 078 fler metoder än Google Maps SDK. Även om antalet metoder inte ger en perfekt bild av tillgänglig funktionalitet, kan det härledas att ArcGIS Runtime SDK är det ramverk som ger större möjlighet att utveckla komplexa geografiska tillämpningar i en Android-applikation.

Tabell 1 Jämförelse mellan två ramverk för kartfunktionalitet.

Ramverk	Storlek APK-fil	Antal definierade metoder	
Google Maps SDK	5 169 kiB	com.google.android.gms.maps	<b>1 920</b>
ArcGIS Runtime SDK	33 935 kiB	com.esri.core	9 101
		com.esri.android	2 897
		<b>Totalt:</b>	<b>11 998</b>

Slutligen har den ekonomiska aspekten analyserats. Google Maps SDK medför vissa kostnader under utvecklingstiden, eftersom adresssökningar är knutna till en API-nyckel och kostar ungefär 0,05 SEK per sökning. I fallet med ArcGIS Runtime SDK finns ingen sådan utgift under utvecklingstiden, eftersom ramverket tillåter upp till en miljon adresssökningar per månad utan kostnad. Däremot visas ett så kallat vattenmärke i form av texten *"Licensed for Developer Use Only"* på kartbilden för att förhindra att kostnadsfria utvecklingslicenser används i kommersiella applikationer.

Tabell 2 Ekonomisk jämförelse mellan två ramverk för kartfunktionalitet.

Ramverk	Kostnad per kartvisning	Kostnad per geokodning
Google Maps SDK	US\$ 0,00 <sup>3</sup>	US\$ 0,005 <sup>4</sup> (ungefär 0,05 SEK)
ArcGIS Runtime SDK	Gratis under utvecklingstiden, med ett vattenmärke <sup>5</sup>	1 000 000 sökningar per månad är gratis <sup>6</sup>

Under detta examensarbete vägde den ekonomiska aspekten, och den begränsade funktionaliteten, av Google Maps SDK negativt. Även om ArcGIS Runtime SDK visade ett vattenmärke har den stora mängden funktionalitet bedömts mer användbar under prototypens utvecklingstid. Valet föll därför på det senare.

### 3.1.2 Komplettering av databasen

Den ursprungliga databasen bestod av de tre tabellerna Quiz, Question och Answer. För att kunna implementera positionsbaserade quizzar har två attribut lagts till i tabellen Question för lagring av varje frågas latitud och longitud. Latituder och longituder formar ett rutnät runt jorden, används för att definiera geografiska positioner och representeras av numeriska värden [49]. Exempelvis har statyn av Poseidon i Göteborg longitud

<sup>3</sup> <https://developers.google.com/maps/billing/understanding-cost-of-use#mobile-static>

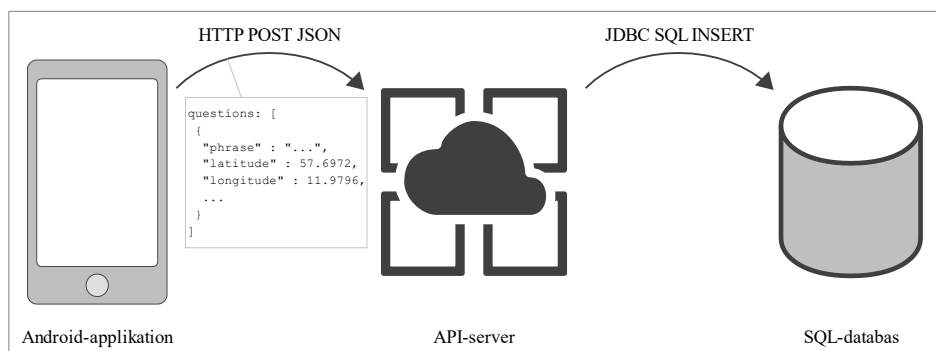
<sup>4</sup> <https://developers.google.com/maps/billing/understanding-cost-of-use#geocoding>

<sup>5</sup> <https://developers.arcgis.com/android/latest/guide/license-your-app.htm>

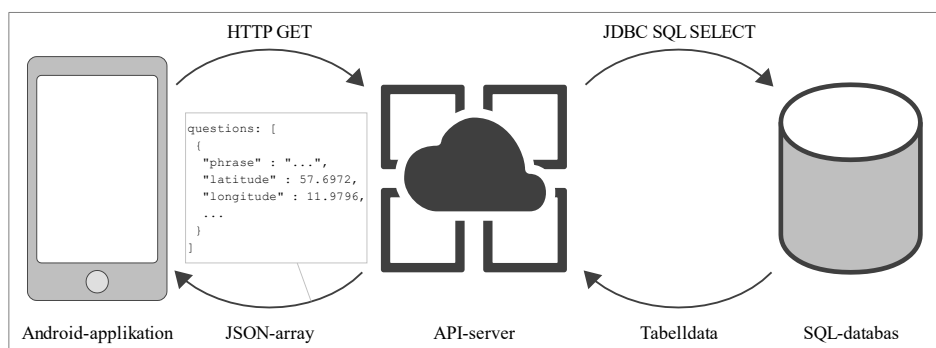
<sup>6</sup> <https://developers.arcgis.com/features/geocoding/>

11,9796 och latitud 57,6972. Givet latituden och longituden är det möjligt att hitta en viss plats på en karta.

I API-servern har latitud och longitud adderats i form av två flyttal, så att när en quiz skapas kan API:et ta emot frågor med bestämd position från klient-sidan och skicka den till databasen som i Figur 12, samt när quizzar hämtas kommer informationen om alla frågor inklusive deras latitud och longitud från databasen till klienten och sparas i applikationsprocessens minne som illustreras i Figur 13.



Figur 12 En ny quiz, inklusive frågors positioner, sparas i databasen via ett anrop till systemets API-server.



Figur 13 En lista med quizzar, inklusive frågors positioner, hämtas från databasen via systemets API-server.

### 3.1.3 Implementation av att skapa quiz

Ett av de första delmålen för att implementera positionsbaserade quizzar har varit att skapa quizzar med frågor knutna till geografiska positioner. Som första steg har en kartvy från ArcGIS adderats som ett nytt fragment till den ursprungliga "skapa quiz"-aktiviteten genom att följa samma steg som i den ovan nämnda undersökningsapplikationen. Dessutom har en stor mängd funktionalitet implementerats som i beskrivningen nedan.



## Visualisering av ArcGIS-karta

För att kunna visualisera en karta i applikationen har en ny layout introducerats. Layouten har bestått av en XML-fil, vilken illustreras i Figur 14, som producerar en grafisk visualisering av en karta (*MapView*) [50]. Att kartan täcker hela skärmen görs möjligt tack vare attributen *android:layout\_width* och *android:layout\_height*. Attributet *android:id* har använts för att matcha kartvyn med ett *MapView*-objekt i Java i aktiviteten *MapActivity* som i Figur 15.

```
<com.esri.arcgisruntime.mapping.view.MapView
    android:id="@+id/mapView"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</com.esri.arcgisruntime.mapping.view.MapView>
```

Figur 14 Element i en XML-baserad layout som används för att skapa en kartvy med hjälp av ArcGIS Runtime SDK från Esri.

```
class MapActivity extends Activity {
    private CreateMapQuestionsHelper mapHelper;
    ...

    void setupMapView() {
        MapView map = findViewById(R.id.mapView);
        this.mapHelper = new CreateMapQuestionsHelper(quiz, map);
        ...
    }
}
```

Figur 15 Programkod i Java för att initialisera kartvyn med hjälp av klassen *CreateMapQuestionsHelper*.

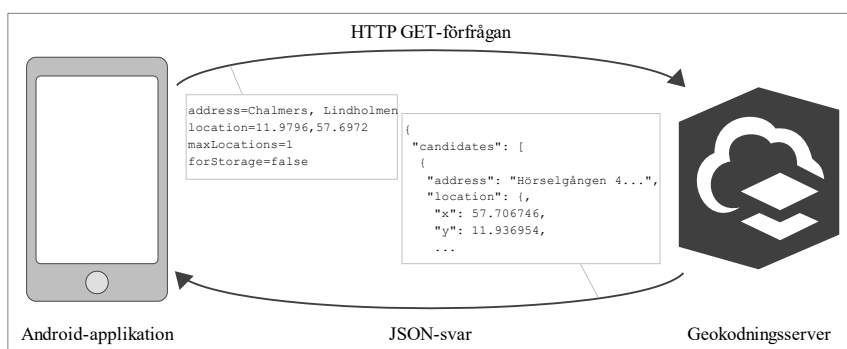
Enhetens aktuella position begärs från operativsystemet och, om denna är tillgänglig, centreras kartbilden på den plats där användaren befinner sig. Ifall positionen saknas, används istället en standardposition som är Lindängen Centrum. Med hjälp av en inbyggd funktionalitet från ArcGIS SDK kan kartan zoomas in och ut, samt panoreras, alltså flyttas så att andra områden visas.

## Sökning av en plats

För att ge användaren möjlighet att skapa en quiz på valfri plats har en sökfunktionalitet implementerats på kartvyn. Eftersom sökning bara ska ske när användaren trycker på en knapp med en särskild symbol, har en inmatningsruta adderats till kartlayouten men hålls dold tills knappen trycks. Detta har kunnat göras med hjälp av ett attribut i layouten

(*android:visibility="gone"*). När knappen trycks förändras synligheten reaktivt för inmatningsrutan samt tangentbordet aktiveras.

Användaren kan mata in en adress eller namnet på en plats, som till exempel "Chalmers, Lindholmen, Göteborg". Adressen geokodas, vilket betyder att den sökta adressen i form av en sträng, skickas som parameter till ett REST-API på en geokodningsserver hos Esri. På serversidan konverteras adressen till latitud och longitud och resultatet skickas tillbaka till applikationen som använder data för att centrera kartvyn på den sökta positionen. [51]



Figur 16 Applikationen hämtar latitud och longitud för en adress från Esris geokodningsserver.

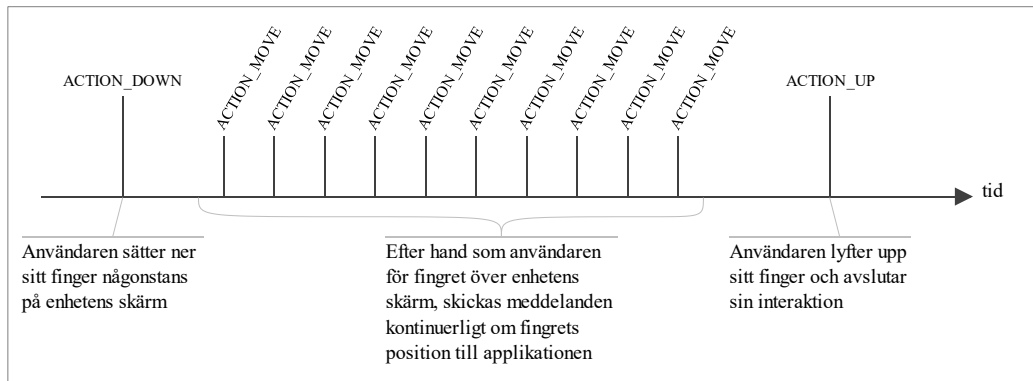
Tillsammans med söksträngen skickas också en parameter som anger vilken del av världen som ska föredras (*location*), så att om adressen som användaren har matat in påträffas på flera olika platser i världen, är det den plats som ligger närmast vad kartvyn visar som används. Om användaren exempelvis söker efter Paris och kartvyn visar någon plats i Europa, kommer sökningen att ge Paris i Frankrike och inte Paris i Texas, USA.

### Rita en slinga på kartan

Ritmekanismen har valts efter att ha utrett olika möjligheter. En första möjlighet var att låta användaren trycka på en sekvens av punkter på skärmen som systemet skulle förbinda med linjer. Den här lösningen bedömdes ha låg användarvänlighet och valdes därför bort. Den andra lösningen innebar att låta användare rita en linje på fri hand på skärmen. Den senare lösningen uppskattades ge mer frihet till användaren.

Androids *touch*-funktionalitet gör det möjligt att interagera med telefonens pekskärm. Förenklat detekteras och analyseras användarens rörelser på pekskärmen av enhetens sensorer och skickas som parameter till metoder i applikationen, som lyssnar på informationen och reagerar på den [52]. I det här fallet finns tre olika typer av information

som är relevanta: den som representerar en händelse där användaren sätter ner ett finger på skärmen (ACTION\_DOWN), den som motsvarar att användaren flyttar fingret (ACTION\_MOVE) samt den som innebär att fingret lyfts upp (ACTION\_UP), vilket visas i Figur 17.



Figur 17 Tidslinje som visar hur användarens interaktioner tolkas som händelser i Android-applikationen.

Vid en ACTION\_DOWN-händelse påbörjas ett polygontåg<sup>7</sup> (*Polyline*) som hanteras av en datastruktur i ArcGIS som heter *PolylineBuilder* och som innehåller en lista av punkter som i sin tur består av en latitud och en longitud. Efter hand som användaren drar fingret längs skärmen, upprepas en ACTION\_MOVE-händelse och polygontåget byggs på med flera punkter som sammanfogas till ett slutligt polygontåg, vilket gör att linjen växer. Händelsen ACTION\_UP är en signal till applikationen att slingan är färdigritad. Figur 18 visar hur PolylineBuilder används för att skapa, bygga och visualisera ett polygontåg.

<sup>7</sup> Enligt Svenska Akademiens Ordbok är ett polygontåg: "Serie av (terräng-)punkter som ligga på längre [eller] kortare avstånd från varandra i en bruten linje".

```

void drawingLineActionDown(Point point) {
    // Påbörja ett polygontåg genom att skapa ett PolylineBuilder-objekt
    trailBuilder = new PolylineBuilder();
    trailBuilder.add(point);

    // Spara det nya polygontåget som ett Polyline-objekt
    trail = trailBuilder.toGeometry();

    // Visa polygontåget på skärmen
    trailGraphic.setGeometry(trail);
}

void drawingLineActionMove(Point point) {
    // Lägg till en ny punkt i slutet av polygontåget
    trailBuilder.add(point);
    trail = trailBuilder.toGeometry();

    // Uppdatera grafiken på skärmen
    trailGraphic.setGeometry(trail);
}

void drawingLineActionUp() {
    // Visa en raderingsknapp
    showDeleteTrailButton();
}

```

Figur 18 Pseudokod som visar hur pekskärmshändelser används för att skapa en promenadslinga. I metoden *drawingLineActionDown* består polygontåget av endast en punkt, medan *drawingLineActionMove* lägger till en punkt i taget i slingan.

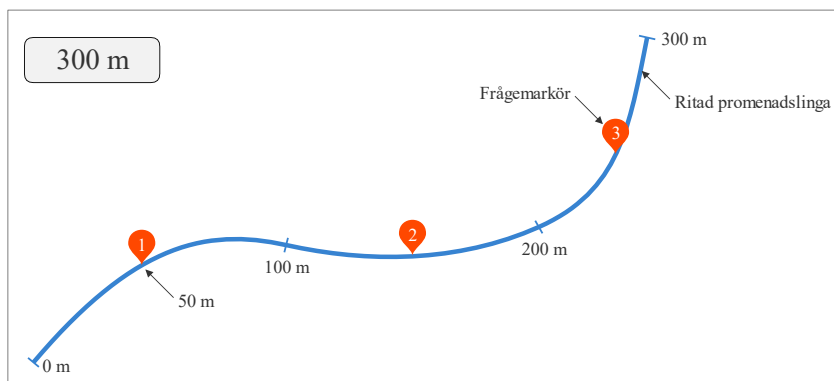
För att undvika att kartan under ritningen fortfarande kan zoomas och panoreras, vilket hade gjort ritning nästan omöjligt, är dessa funktioner låsta under tiden användaren ritat slingan. För varje ACTION-händelse ska ett booleskt returvärde från *touch*-metoden signalera att händelsen är fullständigt hanterad.

Slutligen visas den totala längden på slingan kontinuerligt. Detta är möjligt eftersom vid varje ACTION\_MOVE beräknas slingans längd på nytt och den visas i en textruta på skärmen. Längden på slingan är begränsad nedåt till minst 50 meter per fråga. Om längden är inte tillräckligt lång är nästa steg omöjligt att utföra. Funktionaliteten har gjorts möjlig tack vare en klass i ArcGIS (*GeometryEngine*), som innehåller metoder för att korrekt beräkna längden av ett polygontåg.

## Positionera frågor

Längs den färdigritade slingan placeras frågor i form av röda markörer. De består av två olika delar: en yttre röd figur ovanpå vilken en textmarkör som innehåller frågans ordningsnummer läggs till så att det formas en sammansatt symbol.

För att kunna placera frågorna på rätt plats längs slingan, har hänsyn tagits till längden av slingan som dividerats med antalet frågor. I synnerhet delas slingan in i lika många delar som antal frågor och i mitten av varje sådan del placeras en fråga. Exempelvis, om antalet frågor som skapats är 3 och slingan är 300 meter lång, kommer varje del att vara  $300 / 3 = 100$  meter lång och frågorna placeras 50, 150 respektive 250 meter från slingans start.



Figur 19 Frågorna placeras jämnt utspridda längs slingan.

## Flytta på frågor

Användaren kan välja att flytta på frågorna längs slingan genom att peka på dem och dra dem till den önskade positionen.

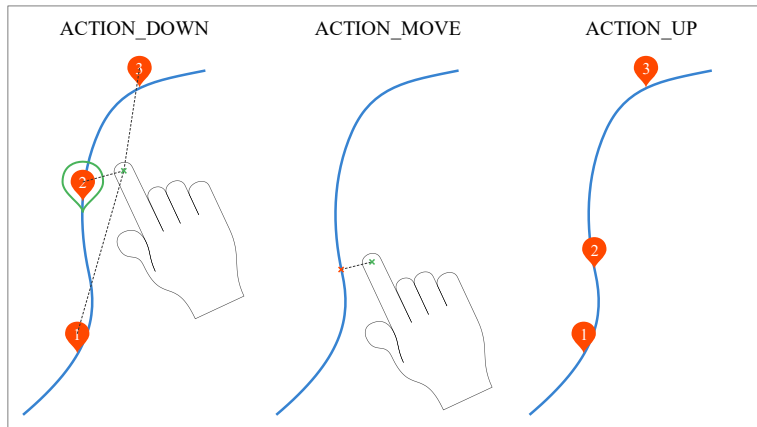
Även i detta fall används touch-funktionalitet för att detektera vilken fråga som ska flyttas. Eftersom en genomsnittlig fingerspets täcker flera hundra pixlar,<sup>8</sup> på en modern mobil enhets skärm och en frågemarkör är av mindre storlek, hade det kunnat vara nästan omöjligt för användaren att placera sin fingerspets precis på den önskade frågan. Lösningen har varit att tillåta användaren att peka på ett något större område, dock så nära frågan som möjligt.

För att bestämma vilken fråga användaren vill flytta har därför en algoritm för att hitta minimum använts, vilket identifierar den fråga som ligger närmast fingret. Alla frågor undersöks och avståndet från fingret till frågan mäts. Den fråga som har kortast avstånd

---

<sup>8</sup> En muspekare på en datorskärm har istället pixelprecision och kan låta användaren peka på en position med en väldigt hög noggrannhet.

till fingret väljs ut för att flyttas. När en fråga flyttas är dess position låst till den ritade slingan och kan inte placeras helt fritt på kartan. För detta har en liknande mekanism implementerats för att identifiera vilken punkt i polygontåget som har kortaste avståndet till fingret. När användare släpper upp fingret från skärmen, placeras frågan på denna punkt.



Figur 20 Visuell representation av algoritmer för att välja och flytta en fråga.

## Förhandsgranskning av frågor

En förhandsgranskning av frågorna och dess svarsalternativ har gjorts möjlig när användaren trycker på en av frågorna. Om en ACTION\_DOWN och ACTION\_UP inträffar i sekvens utan en ACTION\_MOVE emellan, tolkas detta som ett klick och inte som en flytt av en fråga. Klicket startar en modal vy<sup>9</sup> som visar information om den valda frågan.

För att kunna radera frågan måste den grafiska representation av frågan, det vill säga den röda markören, tillsammans med det objekt som motsvarar frågan i quizzens datastruktur raderas. Frågorna som är kvar måste dessutom numreras om. Exempelvis om det totala antalet frågor är 5 och fråga 3 raderas, måste fråga 4 och 5 bli de nya 3 och 4. Kodavsnittet i Figur 21 visar hur alla grafiska objekt som representerar frågor undersöks för att kunna identifiera dels den fråga som ska raderas, dels de som får nya ordningsnummer.

<sup>9</sup> En modal vy visas tillfälligt ovanpå applikationens huvudvy. Användaren måste interagera med den modala vyn för att kunna gå tillbaka till huvudvyn. [58]

```

void removeQuestionGraphic(int questionIndex) {
    // iterera över alla grafiska objekt
    graphicsLayer.getGraphics().forEach(g -> {

        // hämta det grafiska objektets frågeindex
        int index = (Integer)g.getAttributes().get(QUESTION_INDEX_KEY);

        if (questionIndex == index) {
            // detta grafiska objekt ska raderas
            graphicsLayer.getGraphics().remove(g);
        }
        else if (index > questionIndex) {
            // detta grafiska objekt ska numreras om
            g.getAttributes().put(QUESTION_INDEX_KEY, index - 1);
            ...
        }
    }
}

```

Figur 21 Pseudokod som visar hur grafiska objekt raderas och numreras om vid radering av en fråga.

### 3.1.4 Implementation av att spela quiz

De tekniska aspekter som har varit avgörande för implementationen av denna del av spelet har varit bestämning och visualisering av spelarens position, kompassriktning och nästa frågas position, samt säkerställande av att all sådan information alltid är närvarande på skärmen.

#### Positionsbaserad spelmekanism

Grunden för att spela en quiz med positionsbaserade frågor är att avgöra när spelaren befinner sig på den plats där en fråga finns placerad. För att uppnå detta måste avståndet mellan spelarens position och nästa fråga uppskattas kontinuerligt. En bakgrundstjänst har utvecklats som prenumererar på positionsuppdateringar från operativsystemet. Varje gång en ny position erhålls, mäts avståndet. För avståndsmätning har inbyggd funktionalitet (metoden *distanceGeodetic*) i ArcGIS använts. När avståndet är tillräckligt litet, skickas en signal till spelmotortjänsten om att nästa fråga ska visas.

#### Spelarens position: bestämning och visualisering

Positionen av en mobil enhet kan bestämmas på flera olika sätt. Det vanligaste bygger på GNSS, *Global Navigation Satellite System*<sup>10</sup>, som använder information från satelliter runt jorden och hårdvarubaserade mottagare i mobila enheter. I optimala fall är

---

<sup>10</sup> Den mest kända varianten är *Global Positioning System* (GPS) som är ett amerikanskt system.

noggrannheten på satellitbaserad positionering mellan 5 och 15 meter, men mycket beror på störningar från exempelvis väderförhållanden, höjdskillnader och närvaro av höga byggnader [53]. Satellitteknik fungerar dessutom endast utomhus och det är därför mobila enheter behöver kompletteras med andra mekanismer för att kunna bestämma sin position inomhus eller i tätbebyggda områden.

Trådlösa nätverk kan användas som komplettering till satellitnavigation. Genom att mäta signalstyrkan hos flera Wifi-accesspunkter, kan enhetens position uppskattas med hög noggrannhet. Om varken GNSS- eller Wifi-signaler finns tillgängliga är det i sista hand möjligt att grovt uppskatta positionen med hjälp av mobila basstationer, vilket är en del av infrastrukturen för mobiltelefoni [54].

För att kunna använda alla dessa tekniker tillsammans och på så sätt få den noggrannaste positionen som är möjlig vid varje tillfälle, har en *FusedLocationProviderClient* använts. Detta är en klass inbyggd i Android som, mycket förenklat, kommunicerar med sensorer på hårdvarunivå för positionering och på så sätt prenumererar på information om enhetens position [55].

### **Visualisering av positionsnoggrannhet**

Positionens noggrannhet bestäms av information från *FusedLocationProviderClient* och visas för användaren i form av en halvtransparent cirkel som växer eller krymper beroende på positionens noggrannhet. För att implementera positionsmarkören har ett inbyggt symbolobjekt från ArcGIS använts (klassen *SimpleMarkerSymbol*), medan för att visualisera noggrannhetssymbolen har en approximering av en cirkel skapats i form av en polygon (klassen *Polygon*) med ett stort antal punkter, vars centrumpunkt är spelarens position och radie är positionens noggrannhet. Punkterna i denna polygon har beräknats med hjälp av sinus och cosinus.



```

static final int DEGREES_FULL_CIRCLE = 360;
static final int DEGREES_PER_POINT = 6;

void updateAccuracyMarker() {
    PointCollection points = new PointCollection();
    for (int i = 0; i < DEGREES_FULL_CIRCLE; i += DEGREES_PER_POINT) {
        points.add(pos.getX() + Math.sin(i * Math.PI / 180) * accuracy,
                  pos.getY() + Math.cos(i * Math.PI / 180) * accuracy);
    }

    userAccuracyGraphic.setGeometry(new Polygon(points));
}

```

Figur 22 Exempel på programkod i Java för att skapa en cirkelliknande polygon omkring spelarens position.

## Kompassriktning

För att visualisera kompassriktningen, måste applikationen beräkna den mobila enhetens orientering. För detta används de två hårdvaruenheterna geomagnetisk fältsensor och accelerometer. Den första mäter styrkan och riktningen på jordens magnetfält i förhållande till den mobila enheten och enheten kan på så sätt hålla reda på riktningen till den magnetiska nordpolen. För att kunna visualisera kompassriktningen oavsett om enheten är liggande eller stående, måste applikationen dessutom ta reda på gravitationens riktning, vilket utförs med hjälp av accelerometern [56].

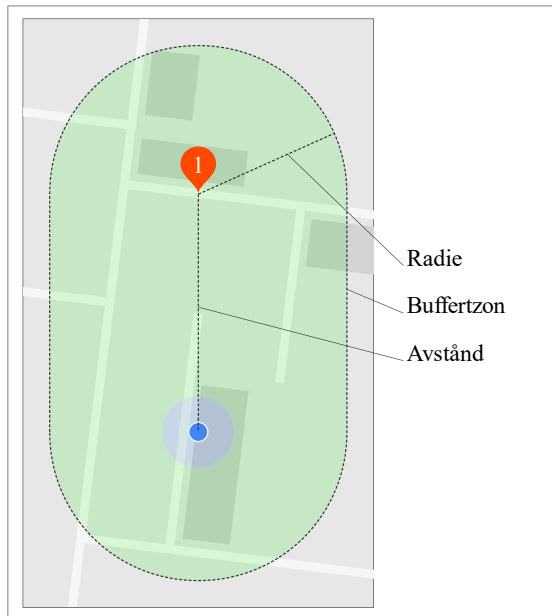
En särskild klass i kodbasen är ansvarig för att lyssna på information från de två sensorerna och, med hjälp av inbyggda metoder i Android, utföra de nödvändiga matematiska operationerna, bland annat korsprodukt av matriser, för att kombinera sensordata till en kompassriktning. Riktningen visualiseras i form av en blå vinkelindikator.

## Kartrotation

Efter hand som spelaren går, vrids och zoomas kartbilden så att spelarens position alltid visas längst ner, medan frågans position visas högst upp. På detta sätt är det spelet som tar ansvar för att visa relevant information på kartan och underlätta spelarens jagande av frågor. Kartan behöver alltså aldrig manipuleras av spelaren under spelets gång.

För att säkerställa att markören med spelarens position och nästa fråga alltid visas på skärmen, har en buffertzona runt de två objekten skapats. Kartvyn instrueras att hela buffertzonen, vars radie är proportionell mot avståndet mellan spelaren och frågan, måste vara synlig.

Medan spelaren går, vrids också kartvyn så att spelarens position alltid syns rakt under frågan. Avståndet mellan spelaren och frågan mäts kontinuerligt, inte bara för att kunna skapa buffertzonen ovan, utan också för att beräkna bäringen, alltså kompassriktningen från spelaren till frågan. Kartvyn vrids slutligen med en vinkel som motsvarar den uppmätta bäringen.



Figur 23 Kartvyn instrueras att rotera och passa in buffertzonen.

### 3.2 Kopiering av quiz

Implementationen av kopieringsfunktionaliteten har återanvänt mycket av den redan existerande kodstrukturen för att skapa nya quizzar. Som förut visas flera fragment med inmatningsfält i följd som steg i en och samma aktivitet. Samtidigt är den enda skillnaden på applikationsnivå att vid kopiering är all information förifylld med data från den kopierade quizzen. På databasnivå har dessutom mer data om varje quiz behövt sparas för att kunna återanvända all information om tidigare skapade quizzar.

### 3.2.1 Implementation

En quiz representeras i minnet av en datastruktur i form av objekt av klassen Quiz, som innehåller ett antal objekt av klassen Question vilka representerar de ingående frågorna, och som i sin tur innehåller tre objekt vardera av klassen Answer. När en quiz kopieras, startas samma aktivitet som vid skapande av en ny quiz. Datastrukturen som representerar den valda quizzen serialiseras<sup>11</sup> med hjälp av det inbyggda Java-interfacet Serializable och skickas in som en parameter med namnet "copy" till den startade aktiviteten, som Figur 24 illustrerar.

```
void copyQuiz() {
    Intent intent = new Intent(this, CreateQuizActivity.class);

    // Parametern serialiseras och skickas in
    intent.putExtra("copy", quiz);
    startActivity(intent);
}
```

Figur 24 Pseudokod som illustrerar hur en kopia av en quiz skapas.

Aktiviteten som startas kontrollerar om en sådan parameter existerar och i så fall deserialiseras<sup>12</sup> parameterns värde, vilket ger upphov till en exakt kopia av den ursprungliga quizzen. Kopian ges en ny tidsstämpel och texten "(kopia)" läggs till i slutet av dess titel.

```
// Kontrollera om parametern "copy" existerar
if (intent.hasExtra("copy")) {
    quiz = (Quiz)intent.getSerializableExtra("copy");

    // Uppdatera tidsstämpeln för kopian
    quiz.setCreated(Calendar.getInstance().getTime());

    // Lägg till texten "(kopia)" sist i titeln
    quiz.setTitle(quiz.getTitle() + " (kopia)");
}
else {
    // Skapa en ny quiz
    quiz = new Quiz();
}
```

Figur 25 I aktiviteten för att skapa en quiz kontrolleras om en kopia eller en ny quiz ska skapas.

---

<sup>11</sup> Serialisering innebär att översätta data från datastrukturer till ett textbaserat eller binärt format som kan skickas som parameter till en metod.

<sup>12</sup> Deserialisering är det motsatta, att ett text- eller binärt format översätts tillbaka till den ursprungliga datastrukturen.

Informationen om en quiz skickas vidare till de olika fragmenten, som vart och ett har ansvar för att låta användaren redigera olika delar av en quiz. Till det första fragmentet skickas quizzens titel så att det visas i användargränssnittet. Till det andra fragmentet skickas de olika frågorna med sina svarsalternativ och information om vilka alternativ som är korrekta. Till det tredje fragmentet skickas information om ifall quizzen är stegräknarbaserad eller positionsbaserad. För positionsbaserade quizzar skickas också information om frågornas respektive positioner, samt den ritade promenadslingans sträckning. När användaren går vidare till nästa fragment, uppdateras den kopierade datastrukturen med ny information.

### 3.2.2 Förändring av databas

I databastabellen Quiz har också attributet *polyline* adderats, vilket lagrar latitud och longitud för punkterna i polygontåget som formar en promenadslinga, detta för att kunna grafiskt visa promenadslingan när användaren väljer att kopiera en positionsbaserad quiz. Polygontåget sparas som en sträng i databasen i ett JSON-baserat format. En fördel med att använda detta format är att stöd för det finns inbyggt i ArcGIS, vilket innebär att även komplexa geografiska strukturer, som till exempel polygontåg, kan lagras i form av en enkel sträng. Det geografiska objektet översätts till en JSON-sträng och tvärtom med hjälp av två metoder i ramverket.

```
{
  "paths" : [
    [ 13.0148038, 55.5610618 ],
    [ 13.0131826, 55.5615463 ],
    [ 13.0134754, 55.5627593 ],
    [ 13.0144357, 55.5634918 ],
    [ 13.0150098, 55.5626525 ],
    [ 13.0161790, 55.5620269 ]
  ],
  "spatialReference" : {
    "wkid" : 4326
  }
}
```

Figur 26 Exempel på en promenadslinga i JSON-format.

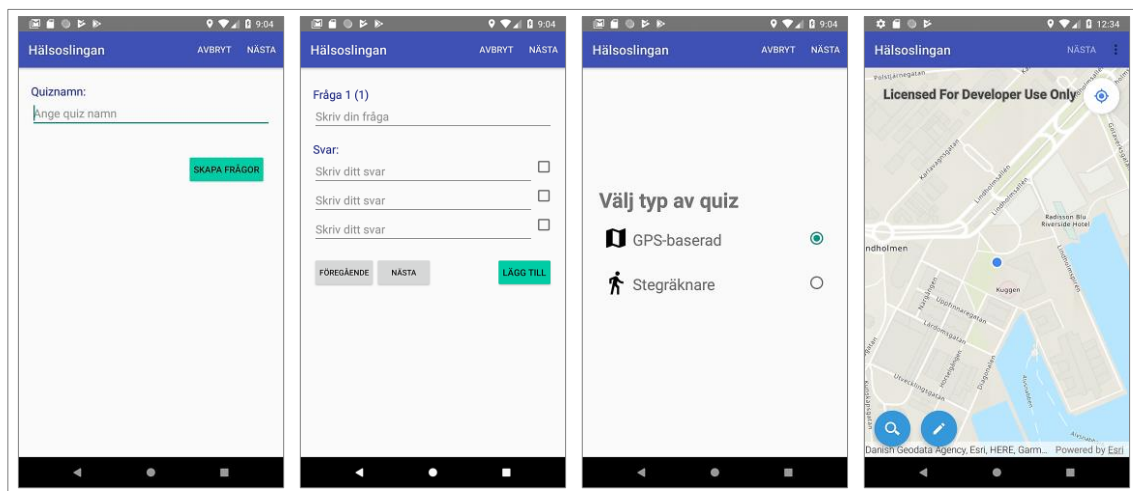
## 4. Resultat

Under examensarbetet har den planerade funktionaliteten adderats till prototypen med resultat som har uppfyllt de förväntade målen i avsnitt 1.4.

### 4.1 Positionsbaserade quizzar

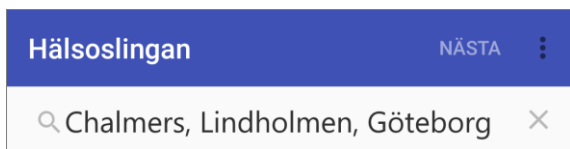
Denna uppgift har haft som mål dels att ge möjligheten för en användare att skapa digitala tipspromenadslingor, dels att göra dessa spelbara. Dessa mål anses uppnådda i alla sina delar.

Som visas i Figur 27, har användaren möjlighet att skapa en quiz, ange quizzens namn, mata in frågor och svarsalternativ i textrutor, samt välja mellan stegräknar- eller positionsbaserad quiz. Användaren kan bläddra mellan frågorna och redigera dem vid behov. Om stegräknare väljs, måste användaren ange totalt antal steg, medan om det andra alternativet väljs, visualiseras en kartbild.



Figur 27 Steg för att skapa positionsbaserade spel. Användaren väljer quizzens titel, matar in ett antal frågor och väljer typ av quiz.

På kartbilden ser användaren sin egen position från början i form av en blå punkt. Användaren kan också välja en annan plats där slingan ska ritas genom att trycka på en knapp längst ner till vänster som symboliserar ett förstoringsglas. Vid knapptryckning visas en ruta längst upp på skärmen och enhetens tangentbord aktiveras.

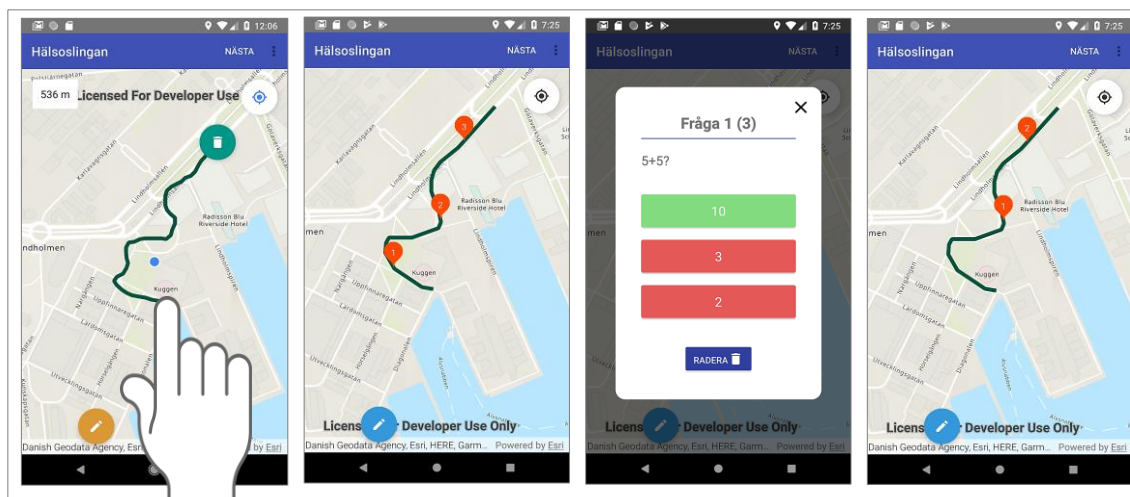


Figur 28 Inmatningsfält för att söka adresser eller platser.

Att panorera och vrida på kartan är också möjligt. Genom att trycka på en annan knapp, som identifieras med en penna, kan en slinga ritas på kartbilden. Ritning sker på fri hand med fingret utan att vara begränsad till vägnät eller geometriska former. Användaren kan bestämma att avvisa den ritade linjen genom att klicka på symbolen av en soptunna, som dyker upp vid den färdigritade slingan, eller gå vidare till nästa steg (Figur 29). I så fall placeras frågorna jämnt utspridda längs linjen. Genom att peka på frågorna och dra dem längs linjen kan användaren välja en annan positionering av frågor.

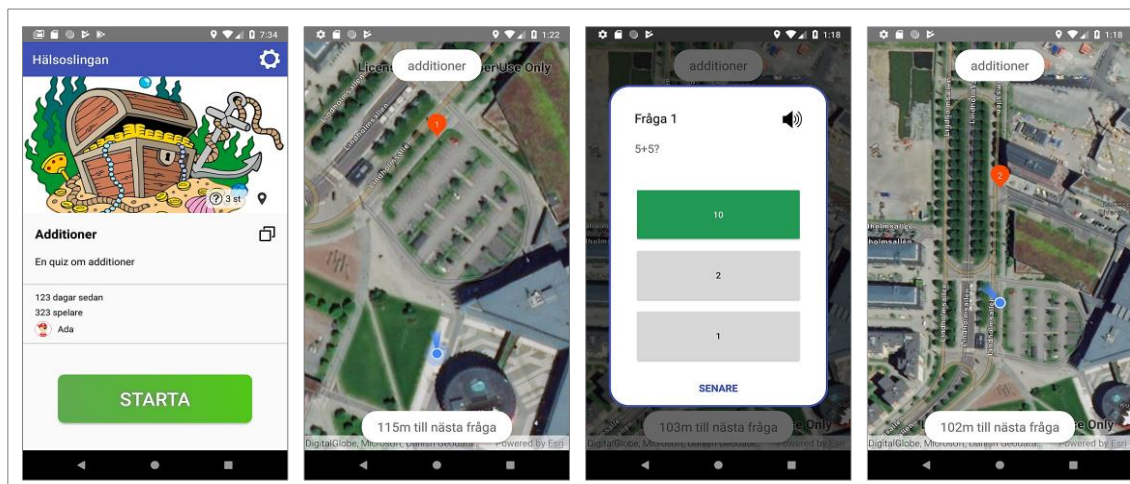
Användaren kan också se en förhandsgranskning av frågorna genom att klicka på dem. Denna överblick över frågorna ger möjlighet att utföra en sista granskning innan quizzen kan sparas. Frågans ordningsnummer, dess titel och de tillgängliga svaren presenteras då, som visas i Figur 29. Svaren som har markerats som rätt visas med en grön bakgrund medan de som har identifierats som fel markeras med en röd bakgrund. Användaren kan antingen stänga fönstret och behålla frågan som den är eller radera den. När en fråga raderas byter de resterande frågorna nummer. Exempelvis, om fråga 1 raderas, numreras frågorna med nummer 2 och 3 om och blir de nya nummer 1 och 2. När sista frågan raderats kan användaren mata in frågor på nytt.

När användaren känner sig nöjd med frågorna, kan hen gå vidare till nästa steg genom att trycka på knappen "nästa" längst upp på skärmen och antingen spela quizzen omedelbart eller gå tillbaka till applikationens startsida, där nu den nya quizzen finns sparad.



Figur 29 Användaren ritar en slinga med fingret, placerar frågor, förhandsgranskar dem och kan radera dem.

När en positionsbaserad quiz startar, kan spelaren alltid se sin position på en kartbild i form av en blå punkt längst ner på skärmen, omgiven av en blå kompassmarkör som anger enhetens riktning, samt en ljusblå cirkel som visar positionens noggrannhet. Nästa frågas position visualiseras högst upp på skärmen i form av en röd markör som innehåller frågans ordningsnummer.



Figur 30 Användaren spelar en quiz. Hen ser sin position i form av en blå punkt samt nästa fråga att besvara.

När spelaren flyttar på sig för att kunna nå frågan, zoomas kartan in och följer automatiskt spelarens position tills spelaren når frågan och besvarar den. När spelaren kommer fram till en fråga, dyker frågetexten och de tillgängliga svarsalternativen upp och spelaren kan välja att ge ett svar eller att hoppa över frågan. När spelaren har nått och besvarat quizzens

sista fråga, ges möjlighet till nya försök med de frågor som hoppats över eller besvarats felaktigt. Spelet avslutas först när spelaren angivit rätt svar på samtliga frågor.

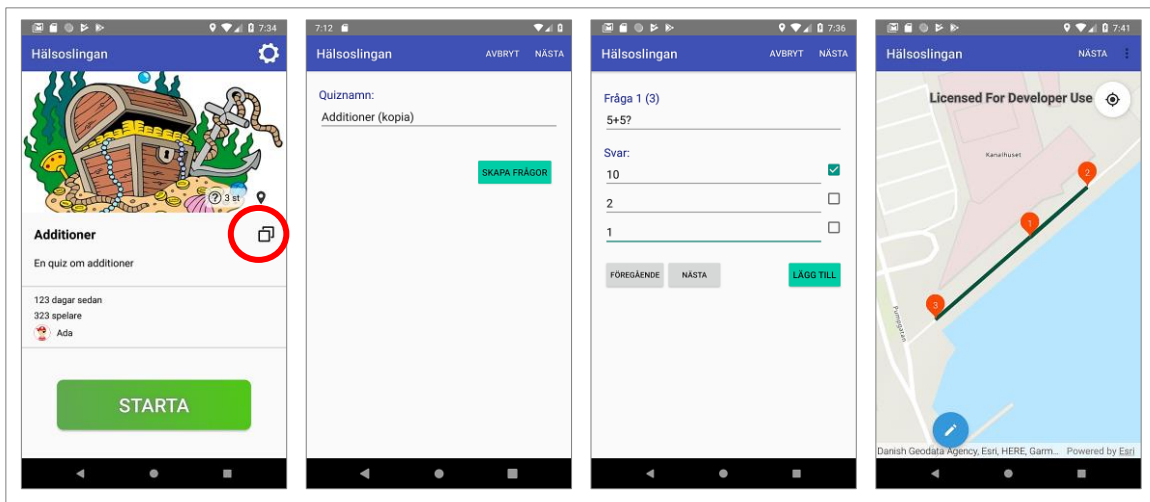
## **4.2 Kopiering av quiz**

Målet med den här uppgiften var att ge användaren möjlighet att återanvända en redan existerande quiz för sitt eget bruk. Att modifiera frågorna, addera nya, eller kasta bort frågor tillhörande den kopierade quizen var den centrala funktionaliteten som önskades. Detta mål har nåtts på ett tillfredställande sätt.

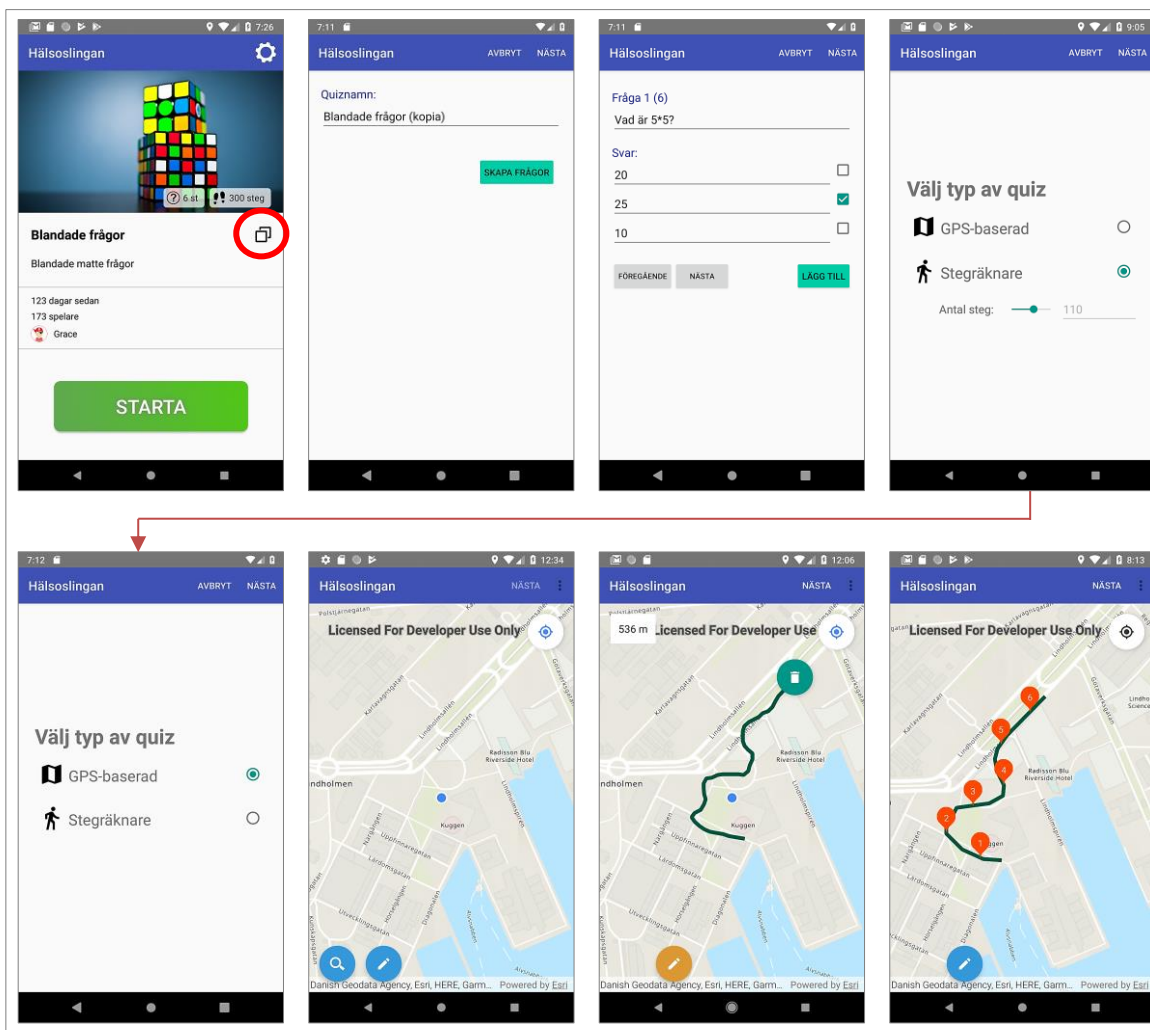
Användaren kan välja en quiz från listan i applikationens startvy och kopiera den genom att trycka på en kopieringsknapp. Vyn för att skapa en quiz presenteras förifylld med information från den kopierade quizen, det vill säga med quizzens titel, dess frågor och svarsalternativ. Användaren kan bläddra mellan frågor genom att trycka på knapparna "föregående" och "nästa" som i Figur 31.

Såvida en positionsbaserad quiz väljs för kopiering, kan användaren välja att modifiera quizen så att den blir stegbaserad med ett antal steg att gå, eller att behålla det ursprungliga valet och i så fall visas slingan med frågor som kan omplaceras efter önskemål, som Figur 31 förtydligar. Ifall en quiz baserad på stegräknare väljs för kopiering, kan användaren antingen välja ett nytt antal steg att gå eller att skapa en positionsbaserad quiz (Figur 32). I detta fall visas en kartvy och användaren kan rita en ny slinga.





Figur 31 Exempel på kopiering av en positionsbaserad quiz. Användaren kan välja att modifiera titeln och frågorna och välja typen av quiz som förut. Om positionsbaserad quiz är vald presenteras slingan och frågor för användaren.



Figur 32 Exempel på kopiering av en quiz som ursprungligen var skapad för att vara stegbaserad. Användaren kan välja att använda samma frågor för att skapa en positionsbaserad quiz istället.

## 5. Diskussion

Att främja fysisk motion hos boende i Lindängen har varit det övergripande syftet med projektet "Hälsofrämjande innovation i samverkan". Som del av detta ambitiösa projekt har en Android-prototyp tagits fram och två olika delar av denna har utvecklats under examensarbetet. Båda uppgifterna har inneburit flera olika lösningsalternativ samt presenterat både tekniska och designmässiga utmaningar.

### 5.1 Positionsbaserade quizzar

Valet av ArcGIS har visat sig vara fördelaktigt. Ur ett ekonomiskt perspektiv har den totala kostnaden kunnat hållas nere till noll under utvecklingstiden. API-nyckel har inte behövt läggas till i programkoden och sökfunktionalitet har varit kostnadsfri.

Ur ett tekniskt perspektiv har ramverket varit till betydande hjälp eftersom vissa algoritmer, som annars vore komplexa att utveckla, har kunnat utnyttja den stora mängd inbyggda funktionalitet som ArcGIS gör tillgänglig. Exempelvis har mätning av avstånd mellan olika geografiska objekt underlättats vid olika tillfällen. För att säkerställa en tvingande minsta längd har först slingan enkelt kunnat mätas under tiden användaren ritar den på skärmen. Dessutom har avståndet mellan användarens position och nästa fråga kunnat mätas kontinuerligt så att systemet kunnat veta när en fråga ska visas under spelets gång.

En annan inbyggd funktion i ArcGIS har hjälpt till att placera frågor på exakt rätt positioner längs en ritad slinga. I detta fall hade annars en algoritm behövt skrivas för att hitta rätt koordinater längs polygontåget (slingan) genom att följa dess punkter framåt och identifiera den del av polygontåget som innehåller den önskade distansen. Att upprepat applicera den räta linjens ekvation hade varit en tillräcklig lösning, om det använda koordinatsystemet hade varit ortogonalt och inte sfäriskt, alltså uttryckt som latitud och longitud. ArcGIS sköter automatiskt alla nödvändiga beräkningar av koordinatprojektioner vilket har underlättat arbetet avsevärt.

Däremot kan tveksamheter uppstå angående användarupplevelsen i samband med att knyta frågor till geografiska positioner. Den utvecklade prototypen utgår från att

användaren förstår att en slinga först måste ritas innan frågorna kan placeras. Möjligen hade applikationen kunnat handleda användaren steg för steg, till exempel genom en animerad instruktion, eller också hade användaren helt enkelt kunnat placera ut varje fråga på kartan punkt för punkt, genom att upprepat peka på skärmen istället för att först rita en slinga. Ytterligare användartester hade kunnat genomföras för att undersöka flera olika alternativ före implementationen för att identifiera den bästa lösningen.

Dessutom hade möjligheten att låsa slingan till vägnätet och undvika byggnader kunnat utforskas genom att använda öppna geografiska informationsdatabaser och maskin-inläring. Å andra sidan hade detta kunnat begränsa användarens frihet att placera frågor på öppna ytor, till exempel på fotbollsplaner, i skogar eller på skolgårdar.

Slutligen hade också spelmekanismen kunnat utredas mer noggrant före implementation. Som det är nu är spelaren inte tvungen att följa den ritade promenadslingan för att kunna svara på frågorna. Det enda som visas på skärmen är nästa frågas position, vilket gör att spelaren kan jaga frågor genom att gå kortaste vägen från fråga till fråga. Möjligheten att få spelaren att följa hela promenadslingan för att nå frågorna hade kunnat utforskas vidare.

## **5.2 Kopiering av quiz**

Den största utmaningen i denna uppgift har varit att låta användaren kopiera positionsbaserade quizzar. De enskilda frågornas positioner fanns redan sparade i databasen, som ett delresultat av implementationen av positionsbaserade quizzar. Däremot saknades i databasen information om den ritade promenadslingans sträckning.

Detta hade som konsekvens att när användaren hade kopierat en quiz och kommit fram till sista steget, alltså placering av frågor på kartan, hade den kopierade slingan inte kunnat visualiseras, utan bara frågornas position. Eftersom mekanismen för att flytta frågor på skärmen kretsat kring att frågorna är knutna till en slinga, har flyttning av frågor i en kopierad quiz varit omöjlig.

Olika lösningar på detta problem hade kunnat vara möjliga. En möjlig lösning hade kunnat vara att, endast i fallet med en kopierad quiz, låta användaren placera frågor helt fritt på kartan, utan att låsa positionerna till en promenadslinga, eftersom en sådan inte

skulle existera. Den lösningen valdes bort eftersom det vore inkonsekvent med mekanismen för att skapa quiz och det hade kunnat förhindra återanvändning av en stor del av lösningens programkod.

En annan lösning hade kunnat vara att låta applikationen approximera en promenadslinga genom att interpolera ett polygontåg med utgångspunkt i frågornas positioner. Även denna lösning valdes bort, eftersom det dels skulle medföra en orimlig mängd arbete, dels aldrig skulle kunna generera en promenadslinga som garanterat liknar den ursprungliga.

Lösningen som till slut valdes var att förändra informationen som lagras i databasen via API:et så att även promenadslingans polygontåg sparas, knutet till den aktuella quizzen, så att även denna inkluderas vid kopiering.

### **5.3 Framtida möjligheter till vidareutveckling**

Prototypen har blivit en god utgångspunkt för framtida utveckling, även om mycket fortfarande återstår. Den ursprungliga planen innehöll flera ytterligare utvecklingsområden, däribland en sökfunktionalitet. I nuläget visas på applikationens startsida en sökruta utom bakomliggande implementation. Denna hade kunnat användas för att söka efter de quizzar som användaren önskat spela genom att mata in nyckelord.

I en första variant av en sådan funktion hade alla quizzar som innehåller det sökta nyckelordet i titel eller beskrivning kunnat ingå i sökresultatet. I en mer avancerad variant hade nyckelord kunnat matchas även mot namnet på den som skapat en quiz, en viss kategori eller en kombination av flera sökkriterier. Exempelvis hade en elev kunnat söka efter alla matematikquizzar skapade av sin lärare.

För en ännu mer sofistikerad sökfunktionalitet hade också användarens egenskaper, såsom ålder, kunskapsnivå eller preferenser, tagits hänsyn till. Denna avancerade funktionalitet hade varit beroende av att användare kunnat logga in i applikationen och ange viss information om sig själv. En inloggad elev i femte klass hade till exempel kunnat söka efter quizzar med ämnet matematik och automatiskt få ett sökresultat med quizzar som motsvarar matematik på femteklassnivå. Förslag på quizzar hade möjligen också kunnat introduceras, liknande mekanismen som används för att rekommendera videoklipp på YouTube eller riktad reklam.

## **5.4 Samhällsnytta**

Syftet med prototypen har varit att tjäna som uppmuntran för befolkningen i särskilt utsatta områden att röra på sig mer utomhus och på så vis främja en sundare livsstil. Förhoppningsvis kommer en färdig applikation att födas ur denna prototyp och bli ett av de viktiga resultaten av hela det hälsofrämjande projektet.

För att klargöra ifall applikationen uppfyller sitt hälsofrämjande syfte skulle flera olika typer av användartester och informationsinsamling behöva genomföras i framtiden. Till exempel skulle användningsfrekvens och -mönster kunna samlas in från applikationen under en lång period och kompletteras med studiegrupper och intervjuer om invånarnas upplevda hälsa. Förändringar över tid skulle också kunna monitoreras ständigt och analyseras i ett geografisk informationssystem som kan ge en tydlig överblick av hälsoläget i samhället och ligga till grund för ytterligare studier samt riktade politiska insatser.

## 6. Slutsatser

Examensarbetet har varit en utmanande uppgift som har inneburit tekniska och användbarhetsmässiga aspekter. Prototypen har mest tjänat som konceptvalidering under hela processen och kommer att ligga till grund för framtida utveckling av en färdig applikation, som troligen ska användas för att kunna jämföra ut hälsoklyftan mellan stadsdelar med olika socioekonomiska förutsättningar. Samhällsnyttan, som är den bakomliggande motivationen under hela projektet, kommer förhoppningsvis att uttryckas i form av förbättrad hälsa för de boende i Lindängen.

Delaktighet i projektet har bidragit till en bättre förståelse för hur mjukvara utvecklas, testas och produceras, samt det lagarbete som ligger bakom produktion av en fullstack-applikation. Frontend, det vill säga Android-applikationen, har varit den största delen av detta examensarbete och har inkluderat aspekter som användarvänlighet, förberedelsearbete med skisser, brainstorming-sessioner och val av ramverk. Backend, alltså API-server och databas, har också berörts i viss utsträckning av examensarbetet.

Även om projektet utan tvekan har varit lärorikt och användbart för företaget, skulle vissa aspekter kunnat ha genomförts annorlunda. En djupare förståelse för Android-arkitekturen och olika komponenter hade behövt undersökas noggrannare, till exempel genom att skapa separata sidoprojekt av mycket mindre komplexitet, för att på ett djupare sätt förstå hur komponenter hänger ihop. Dessutom hade ett designdokument kunnat tas fram i ett mycket tidigare stadium, som hade kunnat underlätta vid vidareutvecklingen och även hjälpa övriga utvecklare att ha en arkitektonisk överblick. Förfinande av programkoden hade också behövt göras oftare och iterativt, samt hade användartester kunnat förberedas i förväg så att designdriven utveckling kunnat bedrivas.

I en nära framtid skulle prototypen berikas med andra aspekter som är viktiga för att uppmuntra motion och lärande, såsom spelifiering och augmenterad verklighet, så att spelet utvecklas från en tipspromenadslinga till en mer mogen och genomtänkt spelupplevelse som kan engagera människor av alla åldrar och bakgrunder.

## 7. Referenser

- [1] "Get to know Adobe XD," Adobe, 14 maj 2018. [Online]. Hämtad från: <https://helpx.adobe.com/xd/how-to/what-is-xd.html>. [Använd 22 mars 2019].
- [2] Ş. Kocakoyun, "Developing of Android Mobile Application Using Java and Eclipse: An Application," *International Journal of Electronics, Mechanical and Mechatronics Engineering*, vol. 7, nr 1, s. 1335-1354, 2017.
- [3] Google Developers, "Meet Android Studio," [Online]. Hämtad från: <https://developer.android.com/studio/intro/>. [Använd 26 november 2018].
- [4] Google Developers, "Analyze your build with APK Analyzer," Google, 2019. [Online]. Hämtad från: <https://developer.android.com/studio/build/apk-analyzer>. [Använd 16 april 2019].
- [5] B. De, API Managment: an Architect's Guide to Developing and Managing APIs for Your Organisation, Bangalore: Apress, 2017, s. 1.
- [6] Esri, "ArcGIS Runtime SDK for Android," 17 oktober 2018. [Online]. Hämtad från: <https://developers.arcgis.com/android/latest/>. [Använd 20 februari 2019].
- [7] Microsoft, "What is Azure?," [Online]. Hämtad från: <https://azure.microsoft.com/en-us/overview/what-is-azure/>. [Använd 26 november 2018].
- [8] Microsoft, "Azure SQL Database Documentation," [Online]. Hämtad från: <https://docs.microsoft.com/en-us/azure/sql-database/>. [Använd 26 november 2018].
- [9] ArcGIS Resource Center, "What is geocoding?," Esri, 18 augusti 2010. [Online]. Hämtad från: <http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/002500000001000000.htm>. [Använd 29 maj 2019].
- [10] L. Harrie, Geografisk informationsbehandling: teori, metoder och tillämpningar, Lund: Studentlitteratur AB, 2013.

- [11] R. Preißel och B. Stachmann, *Git : Distributed Version Control--Fundamentals and Workflows*, Heidelberg: Dpunkt Verlag, 2014, s. 33.
- [12] Google, "Maps SDK for Android," 19 december 2018. [Online]. Hämtad från: <https://developers.google.com/maps/documentation/android-sdk/intro>. [Använd 20 februari 2019].
- [13] Google Developers, "Overview of Google Play Services," Google, 14 Februari 2019. [Online]. Hämtad från: <https://developers.google.com/android/guides/overview>. [Använd 25 Mars 2019].
- [14] W. Stallings, *Data and Computer Communications*, 10:e red., Pearson, 2014, s. 826-827.
- [15] J. Gosling och H. McGilton, "The Java Language Environment," maj 1996. [Online]. Hämtad från: <https://www.oracle.com/technetwork/java/intro-141325.html#334>. [Använd 26 november 2018].
- [16] Oracle, "Java SE Technologies - Database," [Online]. Hämtad från: <https://www.oracle.com/technetwork/java/javase/jdbc/index.html>. [Använd 26 november 2018].
- [17] Microsoft, "SQL Server Management Studio," 16 oktober 2018. [Online]. Hämtad från: <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017>. [Använd 08 december 2018].
- [18] IBM, "RESTful web services," 13 oktober 2018. [Online]. Hämtad från: [https://www.ibm.com/support/knowledgecenter/en/SSGMCP\\_5.2.0/com.ibm.cics.ts.webservices.doc/concepts/concepts\\_restful.html](https://www.ibm.com/support/knowledgecenter/en/SSGMCP_5.2.0/com.ibm.cics.ts.webservices.doc/concepts/concepts_restful.html). [Använd 26 november 2018].
- [19] Trello, "Om Trello - allt bakom tavlorna," [Online]. Hämtad från: <https://trello.com/about>. [Använd 26 november 2018].
- [20] I. T. Elo, "Social Class Differentials in Health and Mortality: Patterns and Explanations in Comparative Perspective," *Annual Review of Sociology*, vol. 35, s. 553-572, 2009.
- [21] O. Solar och A. Irwin, "A conceptual framework for action on the social determinants of health. Social Determinants of Health Discussion Paper 2 (Policy and Practice)," WHO Library, Genève, 2010.



- [22] Jämlikt Göteborg, Hela staden socialt hållbar, ”Skillnader i livsvillkor och hälsa i Göteborg,” Göteborgs stad, Göteborg, 2014.
- [23] Kommission för ett socialt hållbart Malmö, ”Malmöns väg mot en hållbar framtid,” Malmö Stad, Malmö, 2013.
- [24] Malmö Universitet, ”Hälsofrämjande Innovation i samverkan,” 2017. [Online]. Hämtad från: <https://www.mah.se/Forskning/Sok-pagaende-forskning/Halsoframjande-Innovation-i-samverkan1/>. [Använd 17 februari 2019].
- [25] Jämlik hälsa, ”Hälsofrämjande innovation i samverkan,” [Online]. Hämtad från: <https://jamlikhalsa.se/>. [Använd 9 mars 2019].
- [26] Jämlik hälsa, ”Trygghet i omgivningen,” [Online]. Hämtad från: <https://jamlikhalsa.se/verkstader/trygghet-i-omgivningen/>. [Använd 9 mars 2019].
- [27] Wikimedia-användaren Jorchr, ”Bild på tipspromenadfråga,” 5 maj 2018. [Online]. Hämtad från: <https://sv.wikipedia.org/wiki/Tipspromenad#/media/File:Tipspromenadfr%C3%A5ga.jpg>. [Använd 9 mars 2019].
- [28] A. Nijholt, Playable Cities : The City as a Digital Playground, Singapore: Springer Science+Business Media, 2017.
- [29] A. Nijholt, Playable Cities: The City as a Digital Playground, Singapore: Springer Science+Business Media, 2017, s. 115.
- [30] Andrimon AB, ”Turfgame,” [Online]. Hämtad från: <https://turfgame.com/info>. [Använd 9 mars 2019].
- [31] D. Wang, W. Tingmin, W. Sheng, L. Donghai, X. Yang, Z. Wanlei, H. Hassan och A. Alelaiwi, ”Pokémon GO in Melbourne CBD: A case study of the cyber-physical symbiotic social networks,” *Journal of Computational Science*, vol. 26, nr Maj 2018, s. 456-467, 2018.
- [32] L. D. Kaczmarek, M. Misiak, M. Behnke, M. Dziekan och P. Guzik, ”The Pikachu effect: Social and health gaming motivations lead to greater benefits of Pokémon GO us,” *Computers in Human Behaviour*, vol. 75, s. 356-363, oktober 2017.
- [33] D. R. Broom och S. W. Flint, ”Gotta Catch ‘Em All: Impact of Pokémon GO on Physical Activity, Sitting Time, and Perceptions of Physical Activity and Health at

Baseline and Three-Month Follow-Up,” *Games for Health Journal*, vol. 7, s. 401-408, 2018.

- [34] L. Kogan, P. Hellyer, C. Duncan och R. Schoenfeld-Tacher, ”A pilot investigation of the physical and psychological benefits of playing Pokémon GO for dog owners,” *Computers in Human Behaviour*, vol. 76, s. 431-437, november 2017.
- [35] U. Ekelund, ”Physical activity and all-cause mortality across levels of overall and abdominal adiposity in European men and women: the European Prospective Investigation into Cancer and Nutrition Study (EPIC),” *The American Journal of Clinical Nutrition*, vol. 101, nr 3, s. 613-621, 1 mars 2015.
- [36] Google Developers, ”Motion sensors,” [Online]. Hämtad från: [https://developer.android.com/guide/topics/sensors/sensors\\_motion](https://developer.android.com/guide/topics/sensors/sensors_motion). [Använd 3 mars 2019].
- [37] H. Schildt, *Java: The Complete Reference*, Tenth Edition red., Oracle Press, 2018.
- [38] J. Skansholm, ”Vad är Java?,” i *Java - steg för steg*, Lund, Studentlitteratur AB, 2012, s. 9-10.
- [39] O. Cinar, *Android Quick APIs Reference*, New York: Springer Science+Business Media, 2015, s. 41.
- [40] O. Cinar, *Android Quick APIs Reference*, New York: Springer Science+Business Media, 2015, s. 3.
- [41] Google Developers, ”SDK Platform release notes,” Google, augusti 2018. [Online]. Hämtad från: <https://developer.android.com/studio/releases/platforms>. [Använd 7 april 2019].
- [42] Google Developers, ”Application Fundamentals,” Google, 2019. [Online]. Hämtad från: <https://developer.android.com/guide/components/fundamentals>. [Använd 30 mars 2019].
- [43] K. Rubin, *Essential Scrum - A practical guide to the most popular agile process*, Addison-Wesley, 2013, s. 20-28.
- [44] H. Kniberg, *Scrum and XP from the trenches*, USA: C4Media, 2015, s. 6-7.
- [45] Google Developers, ”Maps SDK for Android: Get Started,” Google, 19 december 2018. [Online]. Hämtad från:

<https://developers.google.com/maps/documentation/android-sdk/start>. [Använd 21 april 2019].

- [46] Esri, "ArcGIS Runtime SDK for Android," 2019. [Online]. Hämtad från: <https://developers.arcgis.com/android/latest/guide/guide.htm>. [Använd 25 Mars 2019].
- [47] Esri, "ArcGIS Tutorials: Create a Starter App," Esri, 2018. [Online]. Hämtad från: <https://developers.arcgis.com/labs/android/create-a-starter-app/>. [Använd 21 april 2019].
- [48] Esri, "Search for an address," Esri, 2018. [Online]. Hämtad från: <https://developers.arcgis.com/labs/android/search-for-an-address/>. [Använd 21 april 2019].
- [49] L. Harrie, Geografisk Informationsbehandling: teori, metoder och tillämpningar, Lund: Studentlitteratur AB, 2013, s. 85.
- [50] Esri, "Develop your first map app with Java," Esri, 2019. [Online]. Hämtad från: <https://developers.arcgis.com/android/latest/guide/develop-your-first-map-app.htm>. [Använd 6 april 2019].
- [51] Esri, "ArcGIS REST API: geocodeAddresses," Esri, 2019. [Online]. Hämtad från: <https://developers.arcgis.com/rest/geocode/api-reference/geocoding-geocode-addresses.htm>. [Använd 30 mars 2019].
- [52] Android Developers, "Track touch and pointer movements," Google, 2019. [Online]. Hämtad från: 03. [Använd 30 mars 2019].
- [53] Understanding GPS/GNSS - Principles and Applications, Artech House, 2017, s. 604.
- [54] Google Developers, "Location strategies," Google, 2019. [Online]. Hämtad från: <https://developer.android.com/guide/topics/location/strategies>. [Använd 6 april 2019].
- [55] Google Developers, "FusedLocationProviderClient," Google, 21 mars 2018. [Online]. Hämtad från: <https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient>. [Använd 6 april 2019].

- [56] Android Developers, "Position sensors," Google, 2019. [Online]. Hämtad från: [https://developer.android.com/guide/topics/sensors/sensors\\_position](https://developer.android.com/guide/topics/sensors/sensors_position). [Använd 31 mars 2019].
- [57] Svenska Akademiens Ordbok, "Tipspromenad," [Online]. Hämtad från: [https://www.saob.se/artikel/?seek=tipspromenad&pz=1#U\\_T1499\\_130411](https://www.saob.se/artikel/?seek=tipspromenad&pz=1#U_T1499_130411). [Använd 9 mars 2019].
- [58] M. L. Jangir, Linux Kernel and Device Driver Programming, Laxmi Publications, 2014.
- [59] J. Tidwell, Designing Interfaces, Sebastopol: O'Reilly Media, 2011, s. 97.