CHALMERS
UNIVERSITY OF TECHNOLOGY

Ett virtuellt provrum

Implementation av en applikation för virtuell klädprovning

Examensarbete inom Data- och Informationsteknik

KARIN PERSTORPER

Institutionen för data- och informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg 2020

EXAMENSARBETE

Ett virtuellt provrum

Implementation av en applikation för virtuell klädprovning

KARIN PERSTORPER

Institutionen för data- och informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg 2020

Ett virtuellt provrum

Implementation av en applikation för virtuell klädprovning

KARIN PERSTORPER

Examinator: Jonas Duregård

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

Institutionen för Data- och Informationsteknik

Göteborg 2020

SAMMANFATTNING

Det här examensarbetet är en del i teknikföretaget Semcons löpande filantropiska arbete. Det övergripande målet har varit att ta fram en lösning för virtuell klädprovning. I förlängningen kan det fungera som en plats att informera klädkonsumenter om de arbetsvillkor kläderna tillverkats under, och på så vis påverka vilka kläder som konsumeras. Om konsumenten dessutom informeras om huruvida ett plagg passar eller inte minskar sannolikt onödiga returer av plagg med dålig passform.

Utvecklingen har resulterat i en prototyp för en applikation som visar på möjligheten att uppnå virtuell klädprovning med hjälp av förstärkt verklighet, AR. I applikationen ges möjlighet att prova huvudbonader på en 3D-modell av en person som antas likna användaren. Flera möjligheter till vidareutveckling finns. Att låta användaren prova mer än huvudbonader är en av dem.

Nyckelord: Androidutveckling, Augmented Reality, Java, 3D, klädprovning.

ABSTRACT

This thesis project is a part of the IT company Semcon’s ongoing charity work. The overall goal has been to create a solution for virtual clothing fittings. This could serve as a space to communicate to customers information about the working conditions of workers in the garment industry, which in turn could have an impact on which clothes are consumed. Furthermore, if the consumer is informed about whether a garment fits or not, it is likely to reduce unnecessary returns of garments with poor fit.

The development has resulted in a prototype of an application that demonstrates the possibility of achieving virtual clothing fitting using augmented reality, AR. The application gives the opportunity to try headgear on a 3D model of a person who is assumed to be similar to the user. There are several opportunities for further development. Allowing the user to try more than headgear is one of them.

Keywords: Android development, Augmented Reality, Java, 3D.

FÖRORD

Examensarbetet har genomförts vid företaget Semcon i Göteborg. Arbetet omfattar 15 högskolepoäng och är ett avslutande moment för utbildningen Datateknik 180 högskolepoäng på Chalmers tekniska högskola.

Tack till Anna Funke och Jenny Forsberg för möjligheten att arbeta med ett inspirerande projekt på Semcon. Ett varmt tack till Johannes Magnusson för värdefull teknisk rådgivning under projektets gång. Tack också till Einar Sundgren för uppmuntran under projektets senare hälft.

Ett särskilt tack riktas till min handledare Joachim von Hacht för ett effektivt samarbete och god vägledning vid utformningen av rapporten.

Begrepp och förkortningar

API Application Programming Interface.
AR Augmented Reality (sv. förstärkt verklighet).
CSR Corporate Social Responsibility (sv. företagens sociala ansvar).
IDE Integrated Development Environment.
JSON JavaScript Object Notation. Textbaserat dataformat.
Scrum Arbetsmetod för agila team.
SDK Software Development Kit. Uppsättning utvecklingsverktyg.
Sprint Begrepp för arbetsperiod som används inom Scrum.
UI User Interface (sv. användargränssnitt).
UX User Experience (sv. användarupplevelse).
UX-design Avser arbetet med att designa produkter och tjänster med fokus på användarens upplevelse.

Innehåll

1. Inledning

1.1 Bakgrund

Examensarbetet genomförs i samarbete med Semcon, ett internationellt teknikföretag specialiserat på produktutveckling med huvudkontor i Göteborg.

Semcon bedriver löpande CSR-arbete med fokus på samhällsnytta i form av att dedikera ett antal arbetstimmar per anställd till att stödja utvalda organisationer och engagemang. En av dessa är Clean Clothes Campaign, ett globalt nätverk av kvinno-, solidaritets-, ungdoms- och fackliga organisationer med den gemensamma visionen att förbättra arbetsvillkoren för arbetare inom konfektions- och textilindustrin [1]. Fotot till höger är taget i samband med öppningen av klädbutiken Uniqlo i Köpenhamn 2019.

Figur 1. Uniqlo-arbetare [2].

Att göra klädkonsumenter medvetna om de förhållanden kläderna de köper tillverkas under kan bidra till medvetna val och en förändring i vilka kläder som konsumeras. Om konsumenten får veta att ett plagg tillverkats under förhållanden där arbetarna får en skälig lön, arbetar under kollektivavtal och med rätt att organisera sig, kan sannolikheten öka att konsumenten väljer att köpa det plagget framför ett annat som tillverkats under sämre arbetsförhållanden, eller där sådan information saknas.

En förutsättning för att konsumenten ska ta till sig information om ett klädesplaggs tillverkning är att den presenteras på ett genomtänkt sätt och placeras på en plats där den är lättillgänglig. En sådan plats skulle kunna vara en etikett på plagget eller, vad som kommer att bli examensarbetets fokus, en tjänst för virtuell klädprovning.

Är det möjligt att upplysa en konsument om ett klädesplaggs passform utan att hen har det i sin hand? Med hjälp av virtuell klädprovning vill Semcon testa detta. Förhoppningen är att kunna presentera för olika aktörer inom

klädindustrin ett förslag till ett nytt sätt att prova kläder som kan ha stor inverkan på hur mycket, och vilka, kläder som konsumeras.

1.2 Syfte

Syftet med examensarbetet är att, som en del av Semcons arbete med Clean Clothes Campaign, ta fram en lösning för virtuell klädprovning.

1.3 Mål

Examensarbetets syfte ska uppfyllas genom implementation av en prototyp för en mjukvara som gör det möjligt för användaren att prova kläder virtuellt. Mjukvaran ska leverera önskad funktionalitet genom att låta användaren placera en digital version av ett klädesplagg på en digital version av sig själv.

1.4 Avgränsningar

Examensarbetet kommer inte att innefatta insamling av någon information som Clean Clothes Campaign tillhandahåller, utan endast fokuseras på ett mjukvaruprojekt (framöver “projektet”) där utveckling av en prototyp med funktionalitet för virtuell klädprovning är målet.

Att tillgodose användaren med någon hög trovärdighet beträffande passform är för projektet sekundärt. Vi kommer att anse att målet är uppfyllt om det bedöms möjligt för användaren att göra en grov uppskattning av ett klädesplaggs passform.

Examensarbetet kommer endast att beakta de enheter och storleksmått för kläder som är standardiserade på den svenska marknaden.

2. Metod

I projektets inledande fas hålls möten tillsammans med teknisk handledare och övergripande projektansvarig på Semcon. Den primära avsikten är att diskutera möjliga tillvägagångssätt för att uppnå målet med projektet.

Sedan utforskas 3D-modellering och användbara verktyg för att konstruera digitala avbilder av fysiska objekt. Parallellt studeras diverse standarder för klädmått.

Därefter bestäms kravspecifikation, plattformar och utvecklingsmiljöer som ska användas. Handledare på Semcon kontaktar representanter på Clean Clothes Campaign för att samla in eventuella önskemål rörande mjukvarans funktion och/eller utseende.

Projektet övergår sedan i en implementationsfas där en agil/iterativ utvecklingsprocess ska användas med regelbundna möten för handledning. Projektets tillämpningsområde är från början outforskat av projektets deltagare, och därför appliceras så kallat learning-by-doing. Inspiration och assistans från teknisk handledare på Semcon fungerar som vägledning vid beslutsfattande om nödvändiga tekniker och koncept för applikationsutvecklingen.

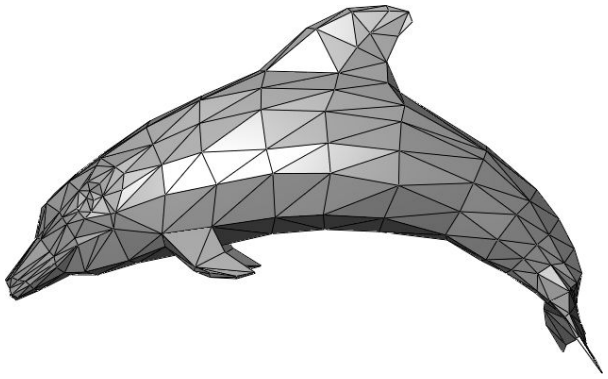
3. Teoretisk bakgrund

Följande avsnitt beskriver teoretiska begrepp relevanta för projektet.

3.1 3D-modellering och polygoner

Begreppet 3D-modellering syftar till skapandet av en matematisk representation av en yta för ett objekt i tre dimensioner [3]. Det kan vara en avbild av ett fysiskt föremål eller levande organism. Slutprodukten är en 3D-modell som kan renderas (konverteras till en tvådimensionell bild eller animering) eller skrivas ut med hjälp av en 3D-skrivare.

Ett sätt att skapa 3D-modeller är att bygga dem genom ihopsättning av ett antal polygoner i ett så kallat polygon-mesh. Ofta används polygoner i form av trianglar då det förenklar rendering [4], men mer komplexa polygon-meshar förekommer. Antalet polygoner i ett polygon-mesh bestämmer dess upplösning, se figur 2.



Figur 2. Ett lågupplöst polygon-mesh av trianglar föreställande en delfin [5].

3.2 Fotogrammetri

Fotogrammetri är tekniken att utifrån fotografier av ett fysiskt objekt göra mätningar av dess tredimensionella egenskaper, vilka sedan kan användas för att generera exempelvis en 3D-modell av objektet [4].

3.3 Augmented reality - AR

Augmented reality (sv. förstärkt verklighet) är en interaktiv upplevelse av den fysiska världen, förstärkt eller kompletterad med hjälp av datorgenererade sinnesintryck i form av ljud, grafik, video och GPS-data [7].

3.4 ISO 18825-1:2016 och ISO 18831:2016

Den internationella standardiseringsorganisationen, ISO, har publicerat ett antal standarder för vokabulär och terminologi för storleksmärkning, kroppsmått, virtuella kläder, den virtuella mänskliga kroppen och digital provning. Utdrag ur de olika standarderna finns tillgängliga gratis på engelska och franska. Nedan följer ett litet antal exempel på termer som definieras. Svenska termer som används i rapporten och projektet är i möjligaste mån direktöversättningar.

Ur ISO 18825-1:2016 Clothing - Digital fittings - Part 1: Vocabulary and terminology used for the virtual human body [8]

virtual human body
: virtual human model for digital fitting in the apparel industry, including information such as size, shape, cross section, body texture and skeletal structure

parametric human body
: virtual human model with changeable parameters such as size and shape, etc.

virtual clone
: virtual human body that is created by forming three-dimensional surface data from a 3D body scanned point cloud (...)

virtual twin
: morphed virtual human body that is applied body dimensions acquired either through manual or automatic measurements

Ur ISO 18831:2016 Cloting - Digital fittings - Attributes of virtual garments [9]

virtual garment items
: virtual garments worn on the virtual human body for digital fitting

virtual hat
: virtual garment worn on the head of the virtual human body

texture maps

bitmap images used for the realistic visualization of virtual garments

3.5 EN 13402-3:2017

Det svenska institutet för standarder, SIS, har fastställt Europastandarden med nummer EN 13402-3:2017 som svensk standard för storleksmärkning av kläder. Dokumentet innehåller figurer och tabeller som beskriver hur olika kroppsmått anges, se exempel nedan.

Ur EN 13402-3:2017 Size designation of clothes - Part 3: Size labelling based on body measurements and intervals [10]

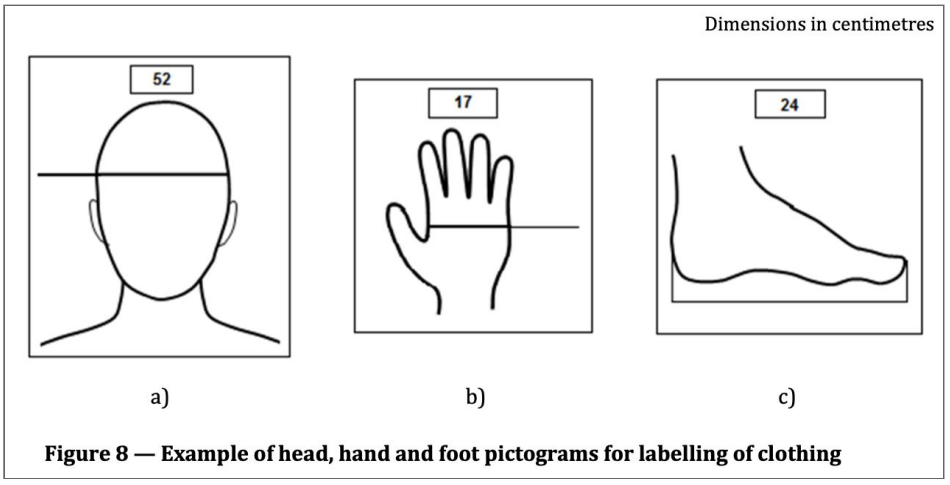


Figure 8 — Example of head, hand and foot pictograms for labelling of clothing

Figur 3. En figur i EN 13402-3:2017 som visar exempel på ett antal kroppsmått och hur de anges.

Table A.21 — Example of head girth measurements for headwear

Dimensions in centimetres

Head girth																
↑	33		34		35		36		37		38		39		40	
Range	32,5	33,5	33,5	34,5	34,5	35,5	35,5	36,5	36,5	37,5	37,5	38,5	38,5	39,5	39,5	40,5
	41		42		43		44		45		46		47		48	
Range	40,5	41,5	41,5	42,5	42,5	43,5	43,5	44,5	44,5	45,5	45,5	46,5	46,5	47,5	47,5	48,5
	49		50		51		52		53		54		55		56	
Range	48,5	49,5	49,5	50,5	50,5	51,5	51,5	52,5	52,5	53,5	53,5	54,5	54,5	55,5	55,5	56,5
↓	57		58		59		60		61		62		63		64	
Range	56,5	57,5	57,5	58,5	58,5	59,5	59,5	60,5	60,5	61,5	61,5	62,5	62,5	63,5	63,5	64,5
Intervals	←	*1*		*1*		*1*		*1*		*1*		*1*		*1*		→

Figur 4. En tabell i EN 13402-3:2017 som beskriver olika huvudmått som används för storlekar på huvudbonader.

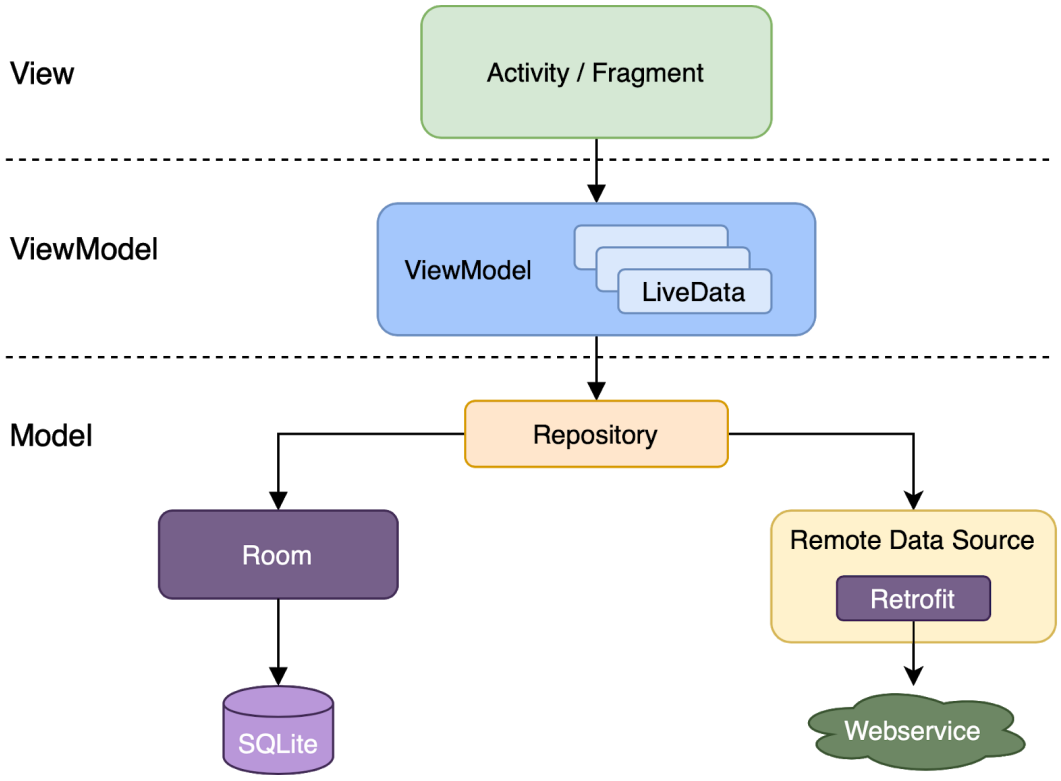
4. Teknisk bakgrund

Nedan följer en genomgång av tekniker som använts i examensarbetet.

4.1 Android, MVVM och Android Architecture Components

Android är ett Linux-baserat operativsystem som i huvudsak används i mobila enheter [11]. Vid applikationsutveckling för Android rekommenderas användning av Android Architecture Components samt mjukvaruarkitekturen MVVM - Model View ViewModel [12].

Architecture Components är en samling bibliotek att använda i MVVM-arkitekturen och finns fritt tillgängliga genom Googles Maven repository. För att använda dem behöver det inkluderas i programmeringsprojektet. Figur 5 visar en översikt av den grundläggande strukturen hos en applikation med delar av Architecture Components, enligt MVVM.



Figur 5. Rekommenderad applikationsstruktur med Android Architecture Components och mjukvaruarkitekturen MVVM.

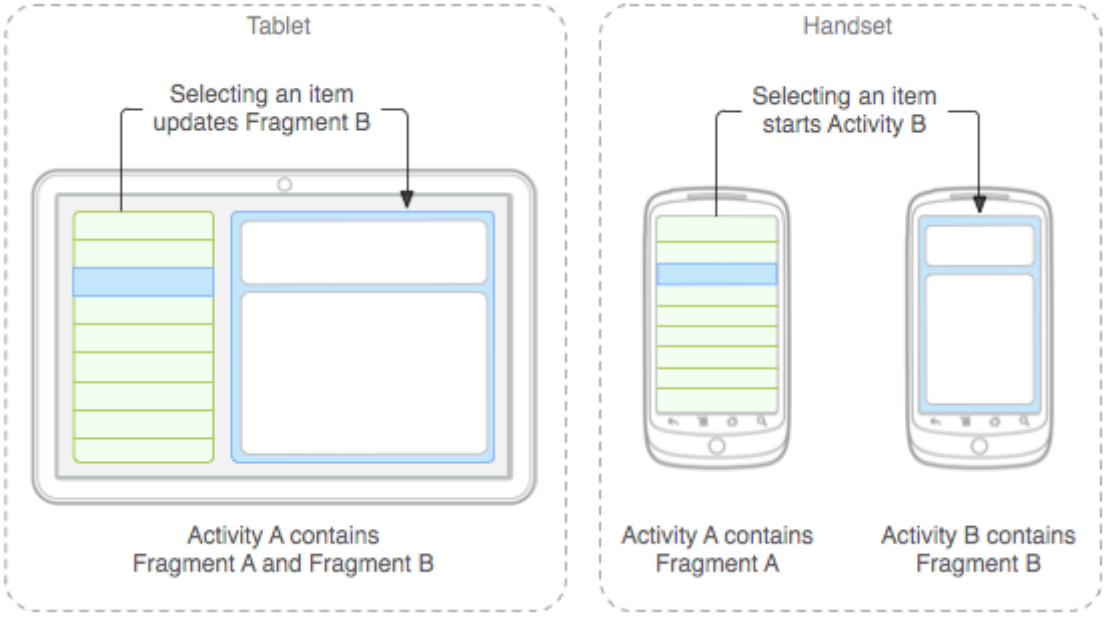
Model representerar data. View är komponenter för användargränssnitt och informerar ViewModel om användarinteraktioner. ViewModel hämtar och lagrar data från Model och exponerar den för View.

4.2 View

4.2.1 Aktiviteter och fragment

En Androidapplikation använder vad som kallas för aktiviteter och fragment. En aktivitet representerar ett användningsfall och är ofta kopplad till en viss skärmbild [13]. En aktivitet kan kan starta andra aktiviteter och kan innehålla ett eller flera fragment som representerar delar av aktivitetens användningsfall.

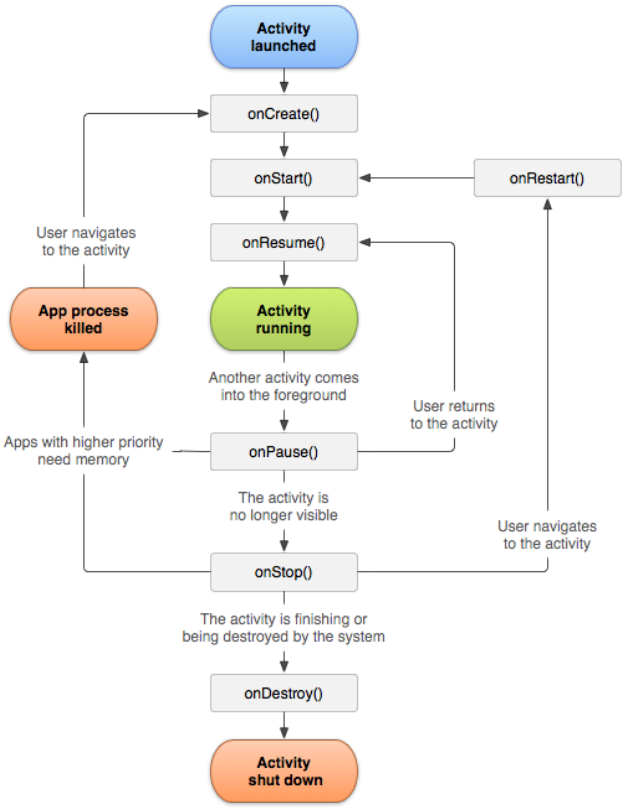
Fragment gör det möjligt att bestämma utseendet på en applikation utifrån enhetens skärmstorlek. Skärmen hos en surfplatta kan exempelvis rymma två fragment samtidigt, medan samma applikation installerad på en mobiltelefon med mindre skärm endast tillåts visa ett fragment åt gången [14], se figur 6.



Figur 6. Exempel på hur två användargränssnitt definierade av fragment kan kombineras i en aktivitet för surfplattor, men separeras i mobiltelefoner [14].

4.2.1.1 Activity Lifecycle

Varje aktivitet har en livscykel. När en användare navigerar genom applikationen passerar aktiviteten olika tillstånd i livscykeln och triggar anrop till callback-metoder [15], se figur 7.



Figur 7. En aktivitets livscykel [15].

När aktivitetens skärmbild, exempelvis en startvy, visas i applikationen befinner sig aktiviteten i exekveringstillståndet som visas i grönt i figuren – aktiviteten körs. Om exempelvis ett knapptryck någonstans i startvyn startar en annan aktivitet, sker ett anrop till onPause(). Om användaren återgår till startvyn anropas onResume() och aktiviteten befinner sig åter i exekveringstillståndet.

4.2.1.2 Fragment Lifecycle

I likhet med aktiviteter har ett fragment sin egen livscykel. Då fragment alltid har en värdaktivitet (*host activity)* påverkas fragmentets livscykel av värdaktivitetens livscykel [14]. När aktiviteten pausats, pausas fragmentet. När aktiviteten avslutas, avslutas fragmentet.

4.3 ViewModel

Del i arkitekturen för att hämta data från Model och göra den tillgänglig för View [12]. ViewModel exponerar data i form av LiveData som kan observeras under en given livscykel. ViewModel hanterar också användarinteraktion i View och skickar uppdateringar till Model.

4.3.1 LiveData

LiveData är en observerbar komponent i ViewModel som möjliggör att förändring av innehållet i viss data i ett gränssnitt syns direkt. När LiveData ändras så notifieras Observers direkt och på så vis är gränssnittet alltid uppdaterat [16]. LiveData är dock livscykelmedveten, vilket innebär att endast de Observers i View-komponenter som är i aktivt tillstånd uppdateras.

4.4 Model

Model-skiktet hanterar applikationens data.

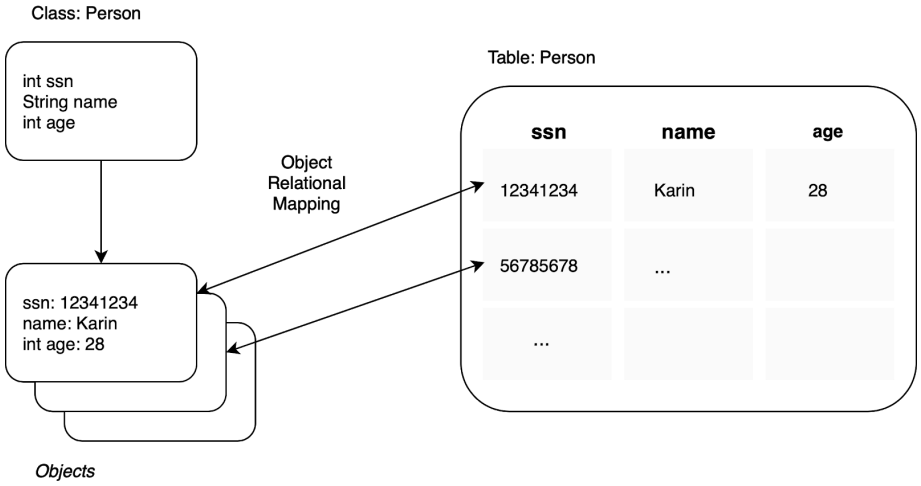
4.4.1 Repository

För att hantera data som potentiellt kommer från olika typer av källor rekommenderas att abstrahera källorna i ett så kallat repository, en komponent med metoder som ViewModel använder för att få tillgång till data [12]. På så vis behöver inte ViewModel bli beroende av var data kommer ifrån.

Android kommer med en förinstallerad SQLite-databas, en relationsdatabas som används i flertalet operativsystem [17]. Figur 5 ovan visar strukturen hos en applikation där Repository hanterar data från två olika källor, remote via en webbtjänst och lokalt med hjälp av SQLite och Room. Det rekommenderas att ett repository implementeras, även i de fall där en applikation endast hanterar data från en källa [12].

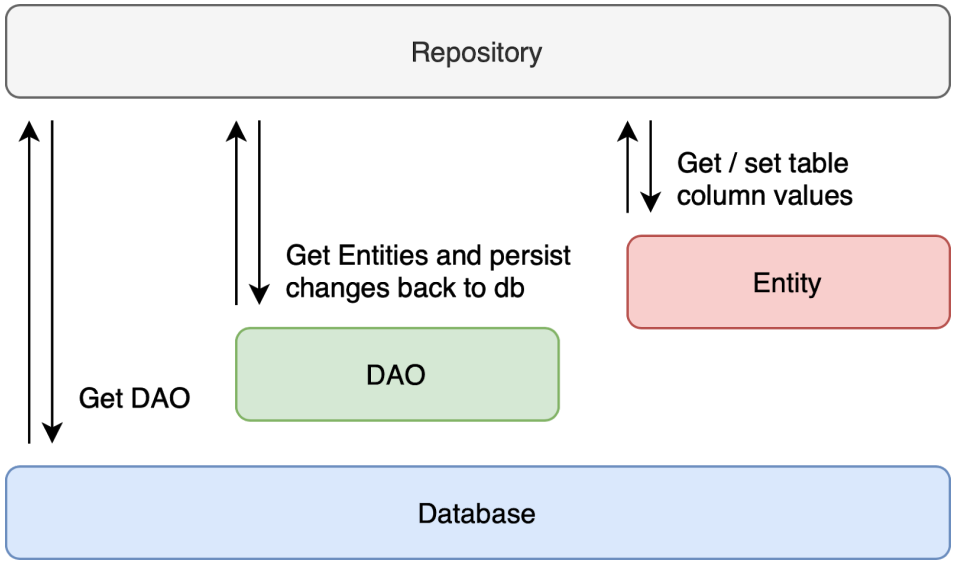
4.4.2 Room

Room Persistence Library är ett bibliotek i Android Architecture Components. Mer specifikt är Room ett ORM-bibliotek (objekt-relationell mappning), vilket innebär att databastabeller representeras som klasser, tabellrader till och från objekt och celler till objekt-attribut [18], se figur 8.



Figur 8. Object relational mapping.

Med ett antal klasser som beskrivs nedan konstrueras ett databaslager i applikationen för att få en övergång mellan SQLite och själva programmet, se figur 9.



Figur 9. Struktur av ett databaslager konstruerad med Room Persistence Library.

Entity

Beskriver en tabell i databasen. Skapas med hjälp av en klass som annoteras med @Entity. Ett objekt av klassen motsvarar en rad i tabellen.

DAO

Data Access Object. Översätter metoder till databasförfrågningar (SQL queries). Tillhandahåller Repository med data från databasen. Skapas med en klass som annoteras med @Dao.

Database

Abstrakt databas-klass, annoterad med @Database, som ärver från klassen RoomDatabase. Fungerar som ett lager ovanpå och en accesspunkt till underliggande SQLite-databas. Använder DAO för att skicka förfrågningar till SQLite.

5. Applikationsutveckling för Android

Följande avsnitt ger först en introduktion till applikationsutveckling för Android, och beskriver sedan hur AR kan implementeras.

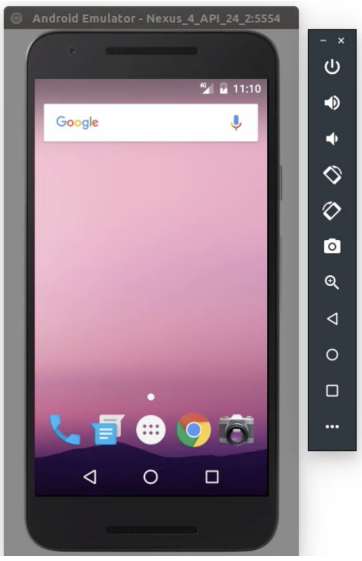
5.1 Språkalternativ

Applikationer för Android kan skrivas i en rad olika språk, bland andra Java, Kotlin, C/C++ och C#. Sedan maj 2019 är Kotlin det språk som Google rekommenderar [19].

5.2 Android Studio

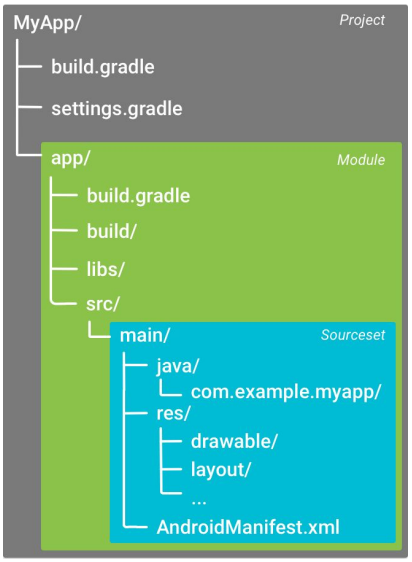
Android Studio är en integrerad utvecklingsmiljö (Integrated Development Environment, IDE) framtagen av Google för utveckling av applikationer för operativsystemet Android [20]. Att Android Studio är ett IDE innebär att det är enkelt att använda olika verktyg, exempelvis plugins och API:er, för att underlätta utvecklingen. Ett par sådana verktyg som använts i programmeringsprojektet nämns nedan.

För att testköra applikationen under utvecklingens gång kan den installeras på en riktig Android-enhet kopplad med USB-sladd till utvecklingsmaskinen, eller en så kallad emulator [21]. För att använda en emulator finns Android Virtual Device (AVD) Manager inbyggd i Android Studio, som gör det möjligt att simulera en specifik enhet från ett urval av färdiga hårdvaruprofiler [22]. Det är också möjligt att skapa egna hårdvaruprofiler eller konfigurera existerande. I figur 10 visas ett exempel på en emulator med hårdvaruprofilen Nexus 4.



Figur 10. Emulator med AVD.

Android Studio använder det fristående byggverktyget Gradle, som ger möjlighet att konfigurera ett projekts bygg- och kompileringsinställningar [23]. Sådana konfigurationer görs i så kallade .gradle-filer. I figuren till höger visas en översikt av en katalog i ett projekt i Android Studio. Rotkatalogen innehåller två .gradle-filer, där settings.gradle specificerar vilka moduler som ska ingå när projektet byggs, och build.gradle definierar konfigurationer som appliceras på alla ingående moduler. Konfigurationer på modulnivå är också möjliga, och görs i respektive moduls build.gradle-fil.



Figur 11. Projektkatalog i Android Studio [23].

5.3 ARCore

ARCore är ett fristående SDK skapat av Google som används inom AR för att modellera den fysiska världen. ARCore använder ett antal API:er för att genom beräkningar på ljusförhållanden och ytor (exempelvis väggar och golv) samt rörelsedetektering göra det möjligt för en mobil enhet att detektera plan och förstå hur dess kringliggande miljö ser ut. Därefter kan virtuella objekt integreras på ett verklighetstroget sätt [24].

För att ARCore ska kunna nyttjas krävs åtkomst till den mobila enhetens kamera, samt att enheten har applikationen Google Play Services for AR installerad. ARCore kan inkluderas i ett projekt i Android Studio, och stöds av Android version 7.0 och högre.

5.4 Sceneform

Sceneform är ett SDK för rendering av 3D-grafik i en applikation. Sceneform inkluderar ett API och ett plugin för Android Studio, version 3.1 eller senare [25]. Pluginet används för att importera, visa och bygga så kallade assets att använda i en 3D-scen. API:et inkluderas i applikationens build.gradle-fil, på modulnivå.

Sceneform tillåter import av 3D-objekt (assets) i tre olika filformat: .OBJ, .FBX och .gLTF. Baserat på ett sådant 3D-objekt och tillhörande specifikationer för texturer och färgläggning, så kallat *material*, genererar Sceneform två nya filer i formaten .sfa (Sceneform asset definition) och .sfb (Sceneform binary asset). Sfb-filen kan därefter användas för att rendera 3D-objektet i en 3D-scen, och modifieras med hjälp av .sfa-filen, som till skillnad från den binära filen kan läsas och tolkas av en människa.

Sceneform ger tillsammans med ARCore möjligheter för utvecklare att modellera 3D-modeller i en AR-miljö.

6. Projektorganisation, grafisk design och konstruktion av 3D-modeller

6.1 Adobe XD

Adobe XD är ett verktyg för UX-design utvecklat av Adobe Systems som möjliggör snabb framtagning av GUI-skisser för webb- och mobilapplikationer [26]. Att snabbt kunna skissa upp olika designidéer för layouter som sedan kan användas som referens eller inspiration underlättar mycket när sådana gränssnitt ska implementeras i en applikation.

6.2 Display.land

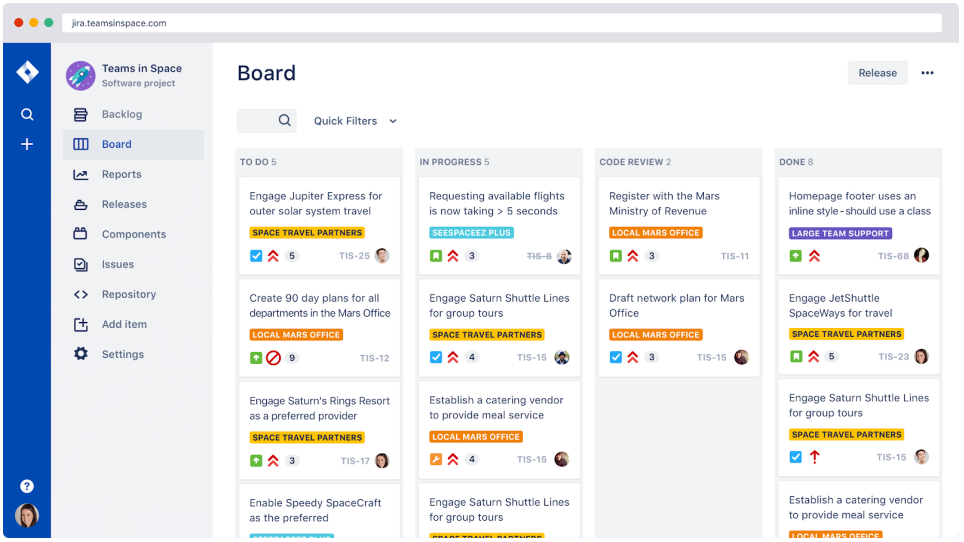
Display.land är en gratis mobilapplikation för generering av 3D-modeller med hjälp av fotogrammetri [27].

6.3 Blender

Program som används för att skapa tredimensionella modeller. Blender har stöd för flertalet tekniker, så som modellering, rendering, animering och simulering [28].

6.4 Jira

Ett verktyg för projektledning och problemlösning som används för att strukturera arbetet i agila team [29]. En så kallad product backlog fylls på med uppgifter som ska utföras, vilka sedan väljs ut till att ingå i en kommande arbetsperiod. För att få en överblick över arbetet listas uppgifterna i en form av enskilda kort i en tavla som förflyttas mellan olika kolumner för att indikera olika faser i arbetsprocessen, se exempel i figur 12.



Figur 12. En uppgiftstavla i Jira [29].

6.5 Git

Git är ett verktyg som används för distribuerad versionshantering av programkod. Verktyget tillåter att flera versioner av en kodbas vidareutvecklas parallellt och lokalt på respektive användares maskin, för att sammanfogas i ett senare skede [30]. Det underlättar samarbete i projekt där flera utvecklare arbetar tillsammans.

6.6 Slack

Ett molnbaserat kommunikationsverktyg för team som behöver en samlad plats att föra diskussioner och fatta beslut [31]. I ett arbetsutrymme i Slack kan exempelvis gruppkonversationer och olika kanaler skapas beroende på diskussionsämne eller deltagare.

6.7 Microsoft Teams

Microsoft Teams är ett samarbetsverktyg där team kan chatta, dela filer, delta i videosamtal och hålla onlinemöten, [32].

7. Genomförande

Följande avsnitt går igenom vad som gjordes för att uppfylla målet med examensarbetet. Först beskrivs utformningen av projektets kravspecifikation och ett antal användarfall. Efter det beskrivs utvecklingsprocessen, använda utvecklingsverktyg, projektets systemdesign och slutligen projektets implementation.

7.1 Kravspecifikation

Projektet inleds med att föra diskussioner med syfte att bestämma en rimlig omfattning. Det står redan klart att projektet kommer att omfatta applikationsutveckling men vidare avgränsningar återstår.

En aspekt som genomsyrar diskussionerna är vikten av en god användarupplevelse. Med utgångspunkt i vad för lösning mötesdeltagarna själva kan tänka sig använda undersöks exempelvis användning av kroppsstrumpor med sensorer för haptisk[1] feedback. Istället bestäms att applikationen ska låta användaren bläddra mellan olika klädesplagg, addera en 3D-modell av ett klädesplagg till en 3D-modell av sig själv, placerad någonstans i den kringliggande miljön med hjälp av AR. Applikationen ska även vara, för projektets deltagare, estetiskt tilltalande. För att knyta an till Clean Clothes Campaign ska information om organisationen finnas i applikationen.

Redan existerande mobilapplikationer som angränsar till projektets tillämpningsområde utforskas, exempelvis applikationer för AR, 3D-modellering och fotogrammetri. Det konstateras att det inte finns någon applikation på marknaden idag som uppfyller målet för projektet på ett önskvärt sätt.

Applikationen ska primärt inte innehålla funktionalitet för att ta fram en 3D-modell av användaren, utan en färdig modell som antas efterlikna användarens kroppsbyggnad ska användas. Om tid finns senare under projektet kan 3D-modellering av användaren utforskas närmare. Tekniker för att generera 3D-modeller av klädesplagg ska dock undersökas och testas.

Ett initialt mål med applikationen är att den ska gå att använda i mobiltelefoner som kör både iOS och Android, vilket förutsätter att applikationen utvecklas i en så kallad cross-platform-miljö. Som en möjlig lösning diskuteras utveckling i Unity, en plattform för utveckling av spel för flera operativsystem, däribland Android [34]. Från Unity kan färdiga projekt exporteras för flera olika operativsystem vid ett tillfälle. Viss expertis och erfarenhet av verktyget finns dessutom tillhanda hos projektets tekniska handledare. En nackdel med Unity är dock att det kostar pengar över en viss nivå av ett företags omsättning [35].

[1] Haptik är läran om effekterna av beröring och kroppsrörelser. [33]

Applikationen är endast tänkt att vara en prototyp, ett koncepttest utan lanseringsdatum, och det beslutas att utvecklingen ska fokuseras på ett operativsystem. Valet faller på Android. Andra operativsystem ska inte tas hänsyn till. Applikationen ska skrivas i Java, som länge varit det officiella språket för Android-utveckling och är väldokumenterat [36]. Vidare ska nödvändig data lagras lokalt.

Sammanfattning av kravlista:

- Utveckla applikation för Android
- Skriv applikationen i Java
- Lagra data lokalt
- Låt användaren bläddra mellan plagg
- Inkludera AR-funktionalitet
- Använd färdig 3D-modell av människa
- Skapa 3D-modell av ett klädesplagg
- Inkludera information om Clean Clothes Campaign
- Skapa ett estetiskt tilltalande UI

7.2 Användarfall

För att hitta funktionaliteten för applikationen skapades ett antal användarfall, se tabell 1 nedan.

Namn	**Händelse**
Bläddra	Användaren bläddrar mellan olika plagg, väljer ut ett eller flera att lägga till i ett slags provrum.
Prova	Användaren navigerar till sitt provrum och väljer ett plagg att prova i applikationens AR-funktion.
Info	Användaren letar upp information om Clean Clothes Campaign.

Tabell 1. Användarfall.

7.3 Agil utvecklingsprocess

Under projektets gång genomförs arbetet agilt, inspirerat av konceptet Scrum [37], med sprintar om oftast en vecka, med reservation för att det i vissa fall behövs mer tid för en given sprint.

Varje vecka, mot slutet av varje sprint, hålls ett avstämningsmöte med teknisk handledare och övergripande projektansvarig på Semcon. Under sådana möten presenteras utfört arbete för sprinten som passerat och en plan för kommande sprint tas fram. Nya uppgifter läggs till i Jira och en ny sprint sätts igång. Ytterligare ett möte hålls med teknisk handledare varje vecka, när halva sprinten passerat. På grund av rådande omständigheter i samhället hålls möten digitalt via Microsoft Teams. Utöver avstämningsmöten håller projektets deltagare och teknisk handledare löpande kontakt via Slack.

7.4 Utvecklingsmiljö och hjälpmedel

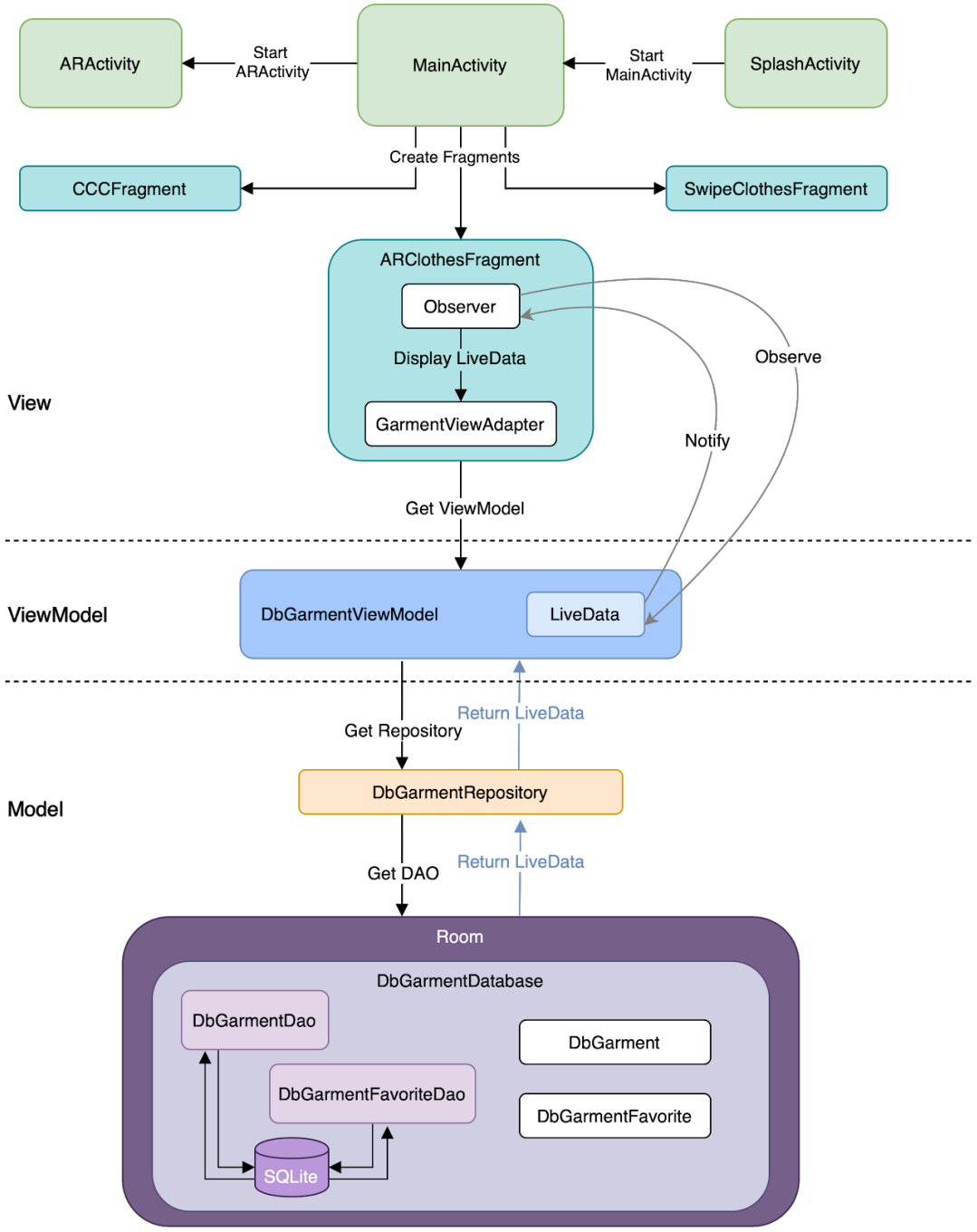
Som utvecklingsmiljö valdes Android Studio version 3.5.3. Den laddas ned och installeras på en maskin med macOS Catalina, version 10.15. En mobiltelefon av modell Google Pixel 4 med Android version 10 installerat kopplas med USB-C-kabel till datorn, se figur 13. I Android Studio skapas ett nytt projekt för att bygga en applikation. ARCore och Sceneform inkluderas i applikationens build.gradle-fil, på modulnivå, och Googles Maven repository inkluderas projektets build.gradle-fil[2]. Till mobiltelefonen installeras applikationen Google Play Services for AR.

Figur 13. En mobiltelefon av modell Google Pixel 4 kopplad med USB-C-kabel till en MacBook Pro.

[2] I Android Studio version 3.0 och senare inkluderas Googles Maven repository i varje nytt projekt.

7.5 Systemdesign

Applikationen har följt MVVM-arkitekturen och använt delar av biblioteken i Architecture Components. Figur 14 visar den övergripande strukturen.



Figur 14. Applikationens övergripande struktur.

7.5.1 View

View innehåller tre aktiviteter: SplashActivity, MainActivity och ARActivity. Applikationen startar i SplashActivity och visar applikationens logotyp under tiden det tar att ladda innehåll i databasen. SplashActivity startar sedan MainActivity och stängs ned.

MainActivity skapar tre fragment: SwipeClothesFragment, ARClothesFragment och CCCFragment, med tillhörande gränssnitt som användaren navigerar mellan. SwipeClothesFragment låter användaren bläddra mellan och favoritmarkera ett antal plagg. När ett plagg markeras som favorit läggs det till i databastabellen DbGarmentFavorite, via DbGarmentViewModel.

I ARClothesFragment visas de favoritmarkerade plaggen i en lista. Listan fylls på med hjälp av GarmentViewAdapter och en Observer som observerar innehållet i databasen med hjälp av LiveData. När ett element lagts till i DbGarmentFavorite notifieras Observern och listan uppdateras. Om användaren tar bort ett element ur listan uppdateras DbGarmentFavorite, LiveData och Observern notifieras.

Från listan av favoritmarkerade plagg i ARClothesFragment kan användaren välja ett att prova. När användaren väljer ett plagg skickas ett anrop i MainActivity att starta ARActivity med det valda plagget, och MainActivity pausas. När ARActivity avslutas fortsätter MainActivity.

CCCFragment är en enkel vy med ett textelement med information om Clean Clothes Campaign.

7.5.2 ViewModel

DbGarmentViewModel hämtar och uppdaterar data i DbGarmentDatabase via DbGarmentRepository och uppdaterar LiveData som observeras i View-delen.

7.5.3 Model

DbGarmentRepository skickar förfrågningar (SQL-queries) till DbGarmentDatabase med hjälp av DAO, som returnerar LiveData.

DbGarmentDatabase innehåller två tabeller (entities), DbGarment och DbGarmentFavorite med respektive DAO.

7.6 Implementation

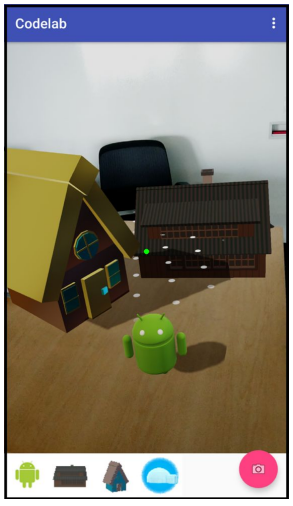
7.6.1 Codelab Tutorial

För att så snabbt som möjligt komma igång med applikationsutvecklingen och utforska möjligheten att inkludera AR-funktionalitet följs en tutorial (lektion) given av Google Codelabs, ett kursbibliotek med ett stort utbud av lektioner för verktyg utvecklade av Google [38]. Mer specifikt följs en tutorial som ämnar introducera utveckling av AR-applikationer med hjälp av verktyget Sceneform. Ett tomt projekt skapas i Android Studio, Sceneform version 1.15 laddas ned och Sceneform plugin för Android Studio installeras. Andra grundförutsättningar är en mobiltelefon kompatibel med ARCore, en USB-kabel att koppla till datorn samt internetåtkomst.

Då Sceneform baseras på Java version 8, behöver kompilering av koden enligt Java 8 säkerställas, se figur 15. I figur 16 syns resultatet av genomgången tutorial [39]. En enkel applikation har skapats som låter användaren placera ut olika 3D-objekt i den fysiska världen.

```
compileOptions {
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
}
```

Figur 15. Kompileringsinställningar i applikationens build.gradle-fil (modulnivå).



Figur 16. Slutprodukt efter genomgången tutorial [39].

7.6.2 Vidareutveckling av resultat från tutorial

Den enkla applikation som Codelab-lektionen resulterade i anses vara ett bra utgångsläge för vidareutveckling och påbyggnad av önskad funktionalitet för

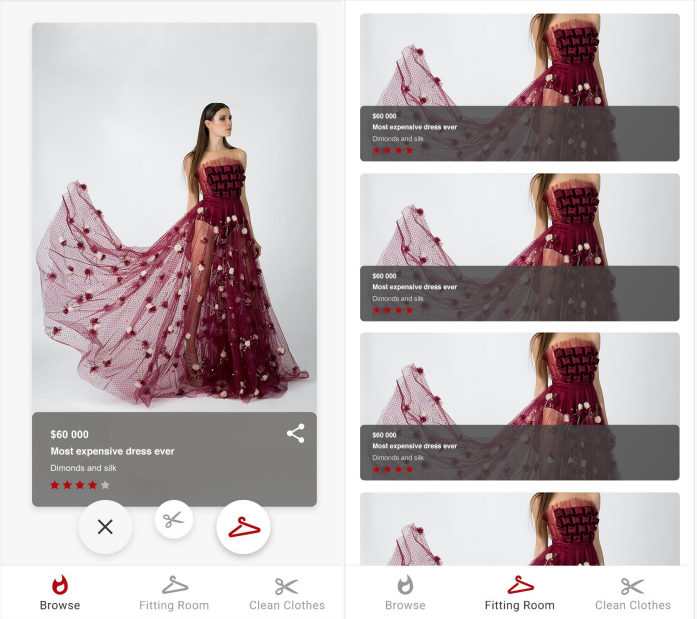
projektet. Så här långt består applikationen av endast en aktivitet (en klass som ärver från klassen Activity) och startar i AR-läget direkt. En första uppgift blir därför att addera en enkel startsida innehållande en knapp som ska starta AR-läget. Att en aktivitet kan starta andra aktiviteter utnyttjas när startsidans aktivitet läggs till.

7.6.3 Design av användargränssnitt

Implementationen av funktionalitet för att kunna bläddra mellan och favoritmarkera plagg hämtade inspiration från applikationer som låter användaren svepa en bild åt höger eller vänster för att indikera ”ja” eller ”nej”.

För att skapa en bra användarupplevelse behövs ett mer sofistikerat användargränssnitt. En skiss att utgå ifrån tas fram med hjälp av Adobe XD. Skissen inkluderar en meny längs nedre kanten av gränssnittet med tre menyval. De tre menyvalen representerar varsin vy, som i sin tur representerar ett fragment.

I den första vyn, “Browse”, presenteras användaren med en uppsättning plagg som visas i en hög av kort. Genom att svepa ett kort åt höger kan användaren lägga till det i sitt virtuella provrum som visas i den andra vyn, “Fitting Room”. Se figur 17 och 18 nedan. Den tredje vyn specificeras inte i detalj, men är avsedd för presentation av Clean Clothes Campaigns arbete.



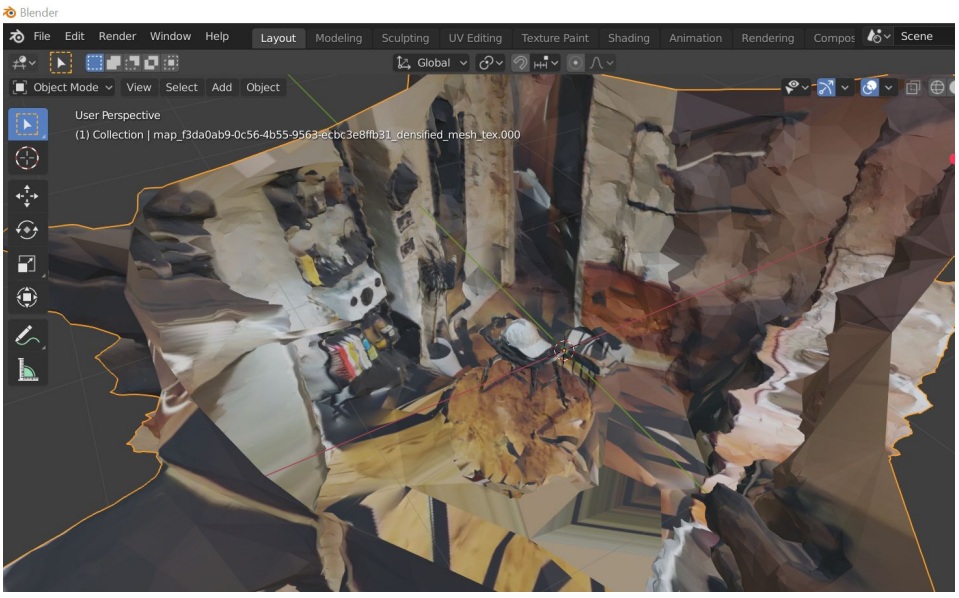
Figur 17 och 18. Skisser av användargränssnitt skapade med Adobe XD. Bild på klänning från [40]. Unsplash License.

7.6.4 Skapa 3D-modell av en keps

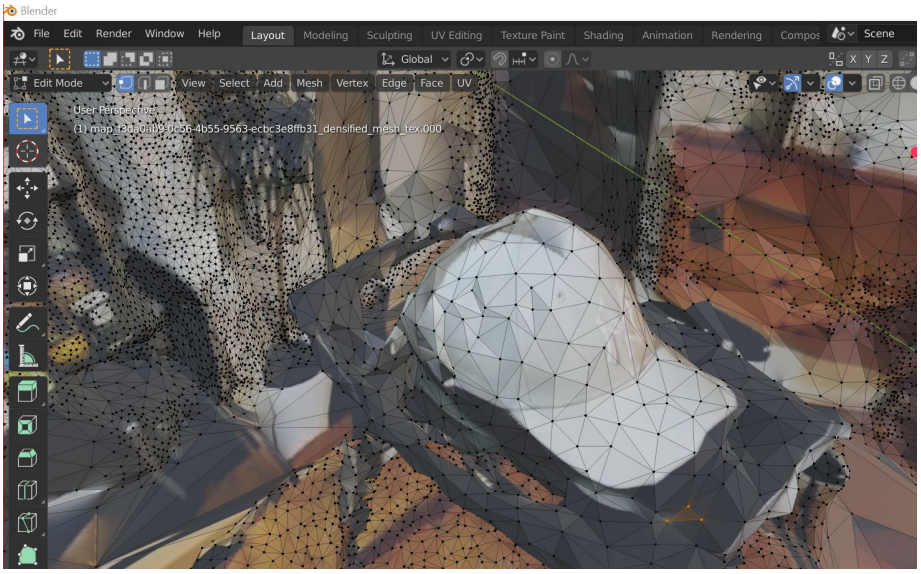
En punkt i kravlistan är att skapa en 3D-modell av ett klädesplagg. I avsnitt 7.6.5 beskrivs hur funktionalitet läggs till för att prova huvudbonader, och för att kunna använda koden som skrivits för det testas att modellera en keps.

Det finns olika metoder att följa när det kommer till 3D-modellering av fysiska objekt. Exempelvis finns det på marknaden gratis mobilapplikationer som tillåter användaren att spela in en film av ett objekt med mobilkameran för att sedan exportera ett 3D-objekt i ett filformat som kan användas i utvecklingsmiljöer som Android Studio. Enligt samma princip finns program som genom att läsa in ett stort antal fotografier av ett objekt från olika vinklar genererar 3D-objekt. Båda alternativen kräver dock ofta en mängd efterbehandling, då spill i form av miljön i objektets omedelbara närhet är svår att undvika och därmed måste skäras bort i efterhand. Ett verktyg som kan användas för att göra sådana korrektioner är Blender.

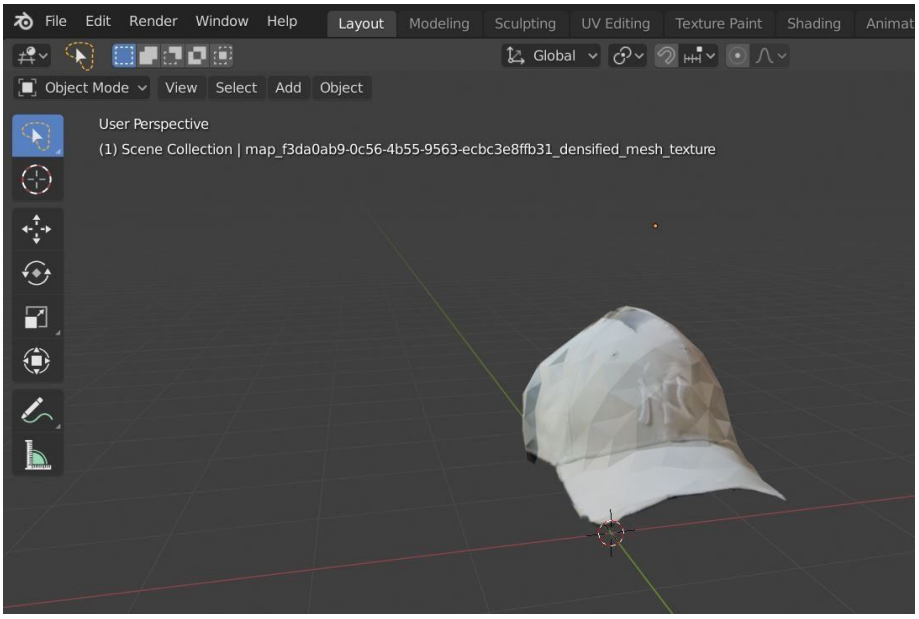
För att läsa in en keps och skapa en 3D-modell av den används mobilapplikationen display.land. När inscanning är färdig exporteras objektet i form av en .OBJ-fil. I figurerna nedan syns skärmdumpar från efterföljande korrektioner i Blender. Figur 19 visar objektet i sitt initiala utförande, med en stor mängd spill, figur 20 visar objektets polygon-mesh och figur 21 visar kepsen efter att korrektioner gjorts.



Figur 19. 3D-objekt med spill i Blender.



Figur 20. Polygon-mesh av 3D-objekt med spill.



Figur 21. 3D-objekt utan spill.

7.6.5 Addera en 3D-hatt till en 3D-person

För att skapa funktionalitet för att addera ett plagg till ett 3D-objekt föreställande en person hämtades inspiration från ett exempel i Googles AR-repository på GitHub. Exemplet använder ARCore och Sceneform och visar hur ett 3D-objekt i form av en keps kan adderas till huvudet på Androids gröna maskot Andy.

När ett virtuellt objekt ska placeras ut måste ett ankare (*Anchor*) definieras så att ARCore kan spåra objektets position över tid [22]. En nod innehållande objektet (*renderable)* kan sedan fästas i ankaret. I kodsnutten i figur 22 skapas ankaret med hjälp av ett *HitResult* innehållande (x,y)-koordinater för en punkt på skärmen användaren tryckt på. I ankaret skapas därefter en nod och till den noden adderas *humanNode* och *hatNode.*

```
Anchor anchor = hitResult.createAnchor();
anchorNode = new AnchorNode(anchor);
anchorNode.setParent(arFragment.getArSceneView().getScene());

humanNode = new SkeletonNode();
humanNode.setParent(anchorNode);
humanNode.setRenderable(humanRenderable);

Node boneNode = new Node();
boneNode.setParent(humanNode);
humanNode.setBoneAttachment(HAT_BONE_NAME, boneNode);

hatNode = new Node();
hatNode.setRenderable(hatRenderable);
hatNode.setParent(boneNode);
```

Figur 22. Kodsnutt ur ARActivity.

ARActivity startar från MainActivity när användaren klickar på ett av plaggen i provrummet. För att AR-aktiviteten ska veta vilket plagg som ska adderas till huvudet skickas ett extra meddelande med när aktiviteten startas, se figur 23.

```
Intent intent = getIntent();
String model = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);

modelLoader = new ModelLoader( owner: this);
modelLoader.loadModel(HUMAN_RENDERABLE, Uri.parse("claudia.sfb"));
modelLoader.loadModel(HAT_RENDERABLE, Uri.parse(model));
```

Figur 23. Kodsnutt ut ARActivity

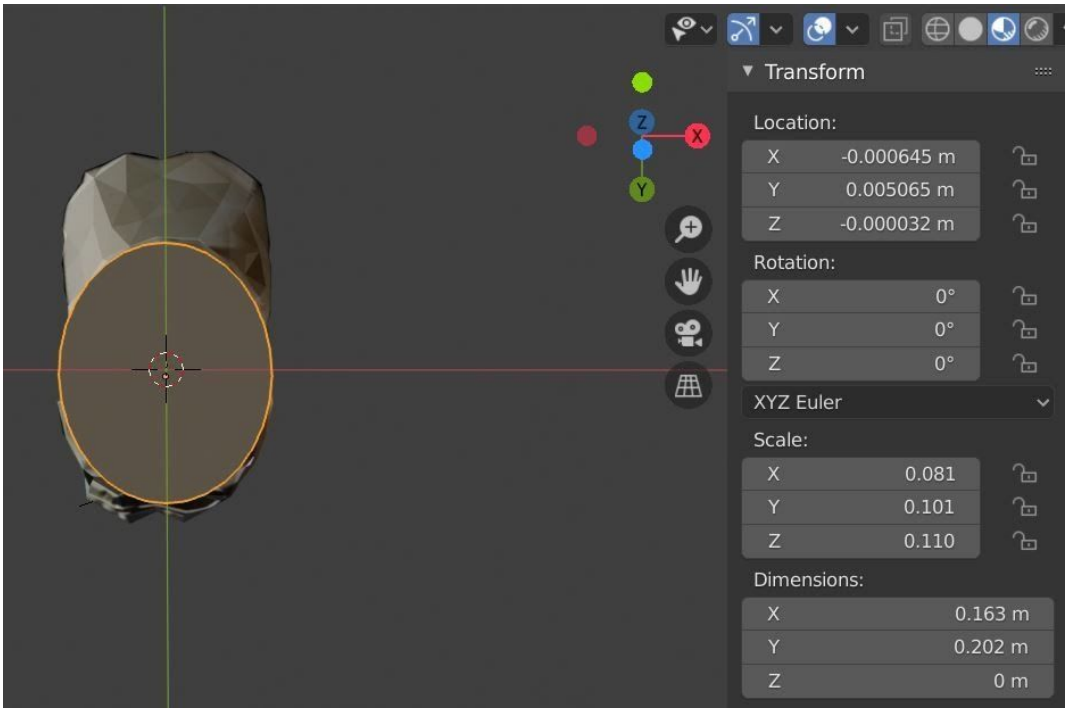
“Claudia.sfb” är en 3D-modell av en kvinna hämtad från FlippedNormals, en webbsida med ett stort utbud av färdiga 3D-modeller [41]. Den inlästa kepsen (7.6.4) samt ett antal 3D-modeller av olika hattar och andra huvudbonader hämtas för att testa möjligheten att prova olika plagg.

7.6.6 Bedömning av passform

Det görs inga justeringar av 3D-modellerna gällande skala. Både den inlästa kepsen och kvinnan, “Claudia.sfb”, är i skala 1:1. För den inlästa 3D-kepsen finns exakta mått att tillgå men mått för huvudomkrets saknas för 3D-modellen av kvinnan. Detta gör det svårt att säkerställa att kepsen faktiskt passar, även

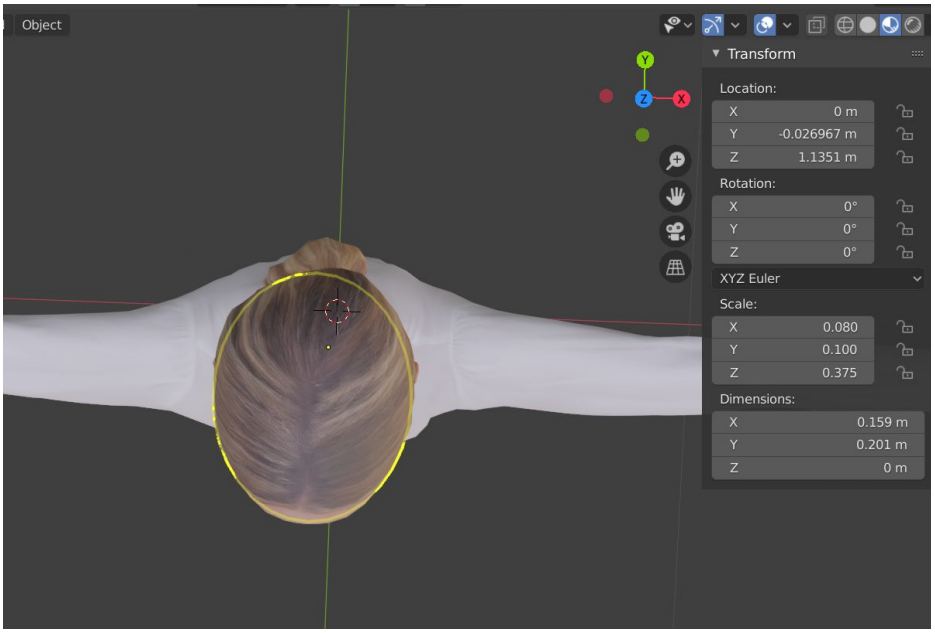
om den ser ut att göra det. Eftersom målet med projektet är att ta fram en prototyp saknar det större betydelse, men det tycks ändå vara intressant att försöka ta reda på om det som visas i AR-funktionen kan stämma. För att ta reda på det görs mätningar av 3D-modellerna med hjälp av Blender.

Kepsens verkliga uppmätta omkrets är 58 cm. Figur 24 nedan visar 3D-modellen av kepsen i Blender, underifrån och med en ellips placerad inuti som tangerar kepsens innerkant där omkretsen är störst. Ellipsens omkrets ska användas för att uppskatta omkretsen av 3D-modellen av kepsen. För att räkna ut ellipsens omkrets används dimensionerna som läses av längst ned till höger i figuren. En ellips med en storaxel om 20,2 cm och en lillaxel om 16,3 cm har en omkrets om ungefär 57,7 cm.



Figur 24. 3D-modellen av kepsen i Blender, underifrån, med en ellips placerad inuti.

På samma sätt räknas kvinnans huvudomkrets ut. En ellips placeras i höjd med huvudets största omkrets, se figur 25. De uppmätta dimensionerna indikerar en omkrets om ungefär 56,9 cm, vilket är mindre än kepsens 57,7 cm. Enligt ovan beräkningar passar kepsen med en marginal om 0,8 cm.



Figur 25. 3D-modellen av kvinnan i Blender, med en ellips placerad i höjd med huvudets största omkrets.

7.6.7 Databas

Information om plaggen i Browse-läget samt provrummet hanteras med en lokal databas med hjälp av Room. Två tabeller skapas: DbGarment och DbGarmentFavorite.

Ett DbGarment beskriver ett plagg och har ett autogenererat id, en url, ett namn, ett pris och ett namn på 3D-objektet som föreställer plagget, figur 26. När applikationen startar första gången skapas ett antal DbGarment-objekt baserat på innehållet i en JSON-fil och används för att fylla på högen av kort som visas i Browse-vyn.

```
@Entity(tableName = "garmenttable")
public class DbGarment {

    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "g_id")
    public int id;

    public String url;

    public String garmentName;

    public String sfb;

    public int priceInSek;

    public DbGarment (@NonNull String url, @NonNull String garmentName, int priceInSek, @NonNull String sfb){
        this.url = url;
        this.garmentName = garmentName;
        this.priceInSek = priceInSek;
        this.sfb = sfb;
    }
}
```

Figur 26. DbGarment.

När användaren favoritmarkerar plagg skapas objekt av DbGarmentFavorite, figur 27. Initialt har sådana objekt endast ett eget (autogenererat) id, och en referens till motsvarande DbGarment id.

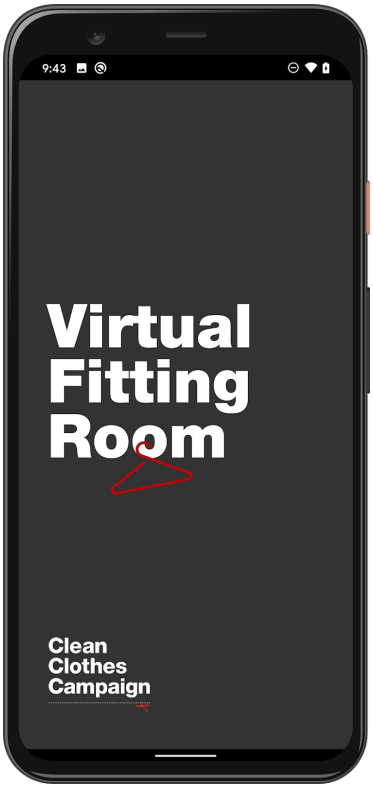
```
@Entity(tableName = "garmentfavoritetable",
        foreignKeys = {@ForeignKey(entity = DbGarment.class,
                                   parentColumns = "g_id",
                                   childColumns = "garmentId",
                                   onDelete = ForeignKey.CASCADE)}
       )
public class DbGarmentFavorite {

    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "f_id")
    public int id;

    @ColumnInfo(name = "garmentId")
    public int garmentId;

    public DbGarmentFavorite(int garmentId){
        this.garmentId = garmentId;
    }
}
```

Figur 27. DbGarmentFavorite.

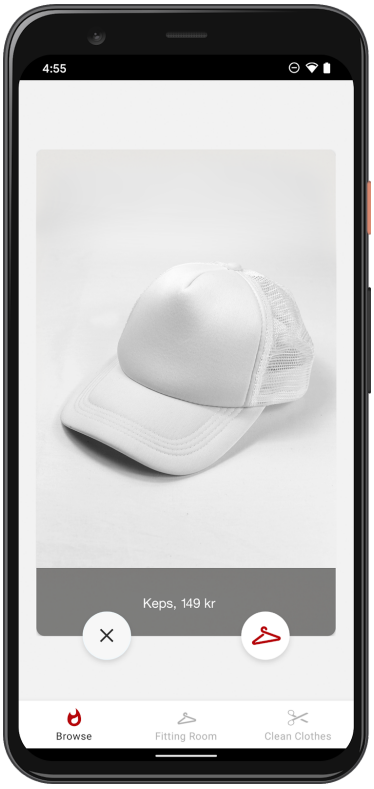
8. Resultat

Applikationen heter Virtual Fitting Room. När applikationen startar visas en Splash screen medan innehåll laddas, se figur 28. Den röda galgen som hänger i ordet “Room” i logotypen symboliserar provrummet och återkommer i applikationen. När Splash screen släckts visas Browse-läget, första menyvalet i bottenmenyn.

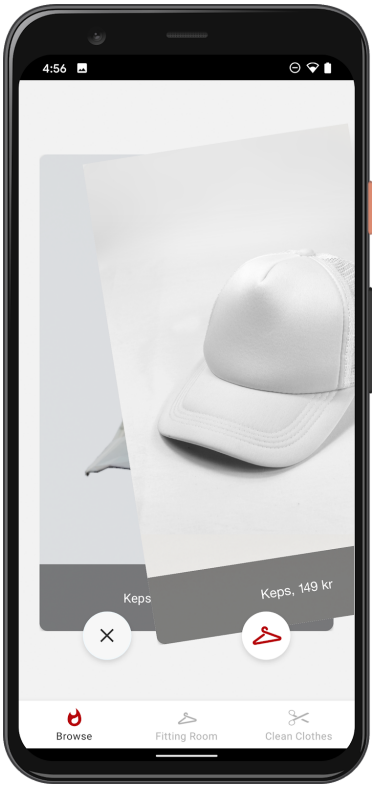
Användaren presenteras med en hög av kort där varje kort är ett klädesplagg, se figur 29. I denna prototyp används hattar. Användaren kan välja att favoritmarkera ett plagg genom att svepa kortet åt höger eller ta bort ett plagg genom att svepa kortet åt vänster. Alternativt kan någon av knapparna användas, se figur 30. Knappen med den röda galgen eller svepning åt höger lägger till plagget i provrummet.



Figur 28. Splash screen

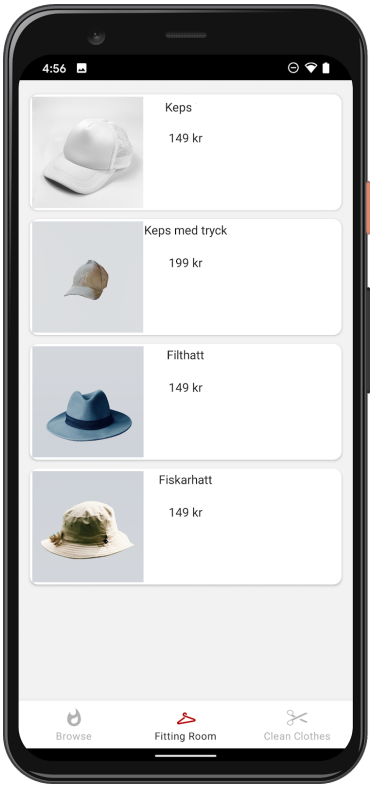


Figur 29. Browse-läget. Bild på vit keps från [42]. Unsplash License.

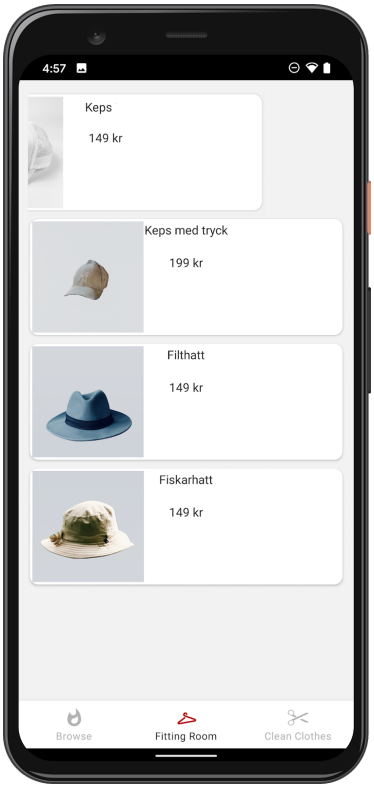


Figur 30. Favoritmarkering. Bild på vit keps från [42]. Unsplash License.

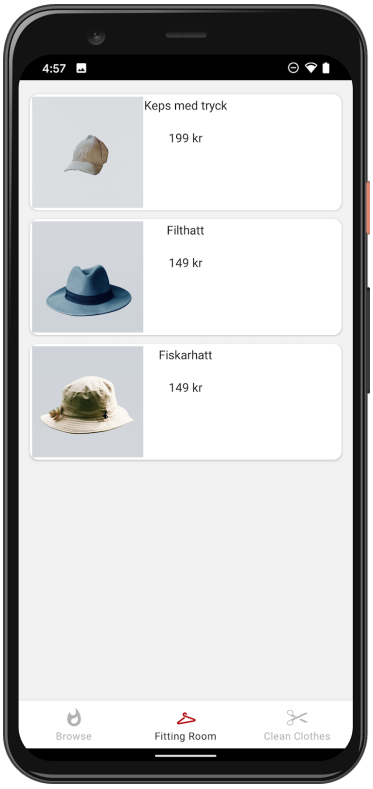
Provrummet nås med den mittersta knappen i bottenmenyn. Innehållet är en lista med de favoritmarkerade plaggen, se figur 31. Användaren kan ta bort ett element i listan genom att svepa det åt vänster, se figur 32 och 33.



Figur 31. Provrummet. Bild på vit keps från [42], filthatt från [43] och fiskarhatt från [44]. Unsplash License.



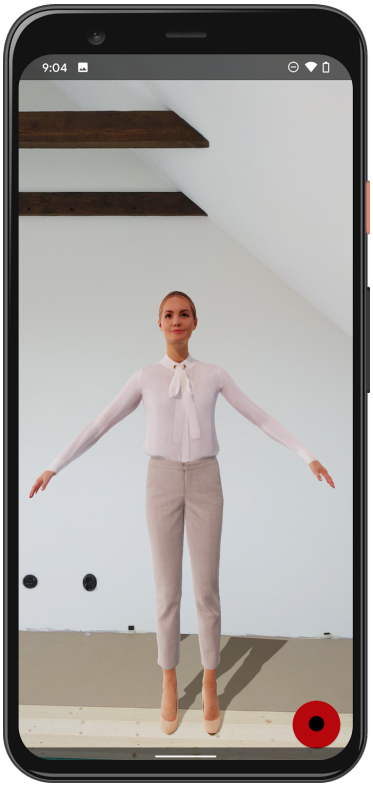
Figur 32. Ta bort plagg från provrummet. Bild på vit keps från [42], filthatt från [43] och fiskarhatt från [44]. Unsplash License.



Figur 33. Provrummet efter att ett plagg tagits bort. Bild på filthatt från [43] och fiskarhatt från [44]. Unsplash License.

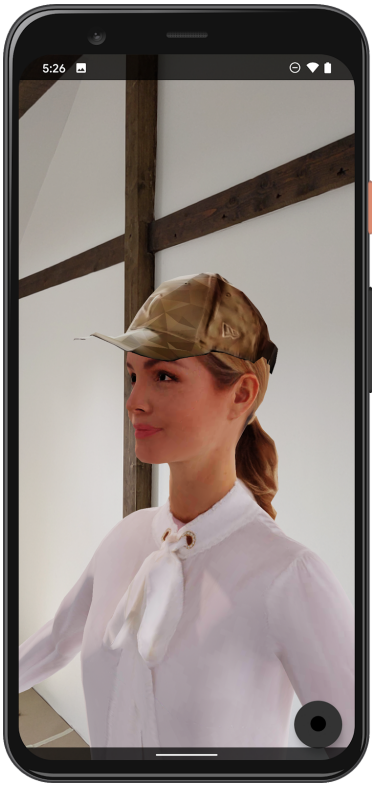
I provrummet kan användaren välja ett plagg att prova, vilket görs genom att klicka på ett av elementen i listan. När användaren gjort det startar AR-funktionen. Figurerna 34, 35 och 36 nedan visar applikationen med AR-funktionen igång då användaren valt att prova en keps (elementet överst i listan i figur 33). Användaren placerar ut 3D-modellen av sig själv, i det här fallet en kvinna som antas likna användaren, någonstans i rummet och kan därefter med hjälp av knappen nere till höger prova kepsen. Samma knapp används också för att ta av kepsen om så önskas.

Genom att svepa åt vänster från högerkanten av skärmen kan användaren återgå till provrummet och välja ett annat plagg att prova.



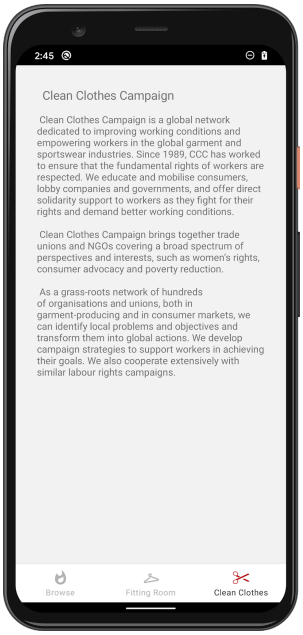
Figur 34. Användaren har placerat ut sin 3D-modell.

Figur 35. Användaren provar kepsen.



Figur 36. Användaren provar kepsen.

Den sista vyn är en ren textsida med information om Clean Clothes Campaign. Valet av en röd sax som ikon för menyvalet inspirerades av Clean Clothes Campaigns logotyp som innehåller en röd sax.



9. Diskussion och slutsatser

Examensarbetets deltagare tog sig an arbetet med stor optimism om att hitta en duglig lösning för virtuell klädprovning. Det stod sedan klart att det skulle handla om utveckling av en prototyp, vilket sänkte förväntningarna något. Resultatet uppfyller de specificerade kraven på prototypen, med reservation för att typen av plagg som användaren kan prova begränsats till huvudbonader och att mer utförlig information om Clean Clothes Campaign hade kunnat inkluderas.

Det finns stora utvecklingsmöjligheter med applikationen. För att en mer trovärdig uppfattning om ett klädesplaggs passform ska uppnås behövs en förfinad metod för framtagning av en 3D-modell av plagget. Den applikation som testades för att läsa in en egen keps, display.land, var tillräckligt sofistikerad för projektets omfattning, men genererade 3D-modeller av tämligen låg kvalitet. En keps eller hatt har en relativt simpel konstruktion med få detaljer. Inläsning av ett klädesplagg som exempelvis en skjorta med ärmar, krage och manschetter hade ställt större krav på detaljrikedom i den genererade 3D-modellen. Display.land användes även vid ett tillfälle för att testa att skapa en 3D-modell av projektets deltagare, med bristande resultat.

Vid en vidareutveckling av applikationen bör tillgång till högkvalitativa 3D-modeller av plagg säkerställas. Funktionalitet för att skapa en 3D-modell av användaren bör också implementeras, exempelvis med hjälp av att låta användaren “bygga” sig själv genom att ange ett antal mått, eller med ett externt, mer avancerat verktyg.

Nedan följer en lista med övriga tänkbara tilläggsfunktioner.

- Möjlighet att ange preferenser gällande passform, till exempel att användaren föredrar tröjor med oversize-passform.
- Implementera maskininlärning. Presentera i Browse-läget plagg baserat på tidigare val. Om användaren favoritmarkerat många plagg i färgen blå eller som har en viss typ av passform, visa fler sådana plagg.
- Låt användaren ange plagg med definitivt bra passform. Användaren kanske har ett par jeans i sin garderob som hen tycker sitter perfekt. Implementera en funktion som låter användaren ange tillverkare, modell/artikelnummer och storlek för de jeansen. Föreslå plagg baserat på sådan information.

9.1 Hållbar utveckling och samhällsnytta

Att utveckla en prototyp som, om den vidareutvecklas, kan få positiva samhälleliga effekter var motiverande. Om applikationen får spridning bland klädkonsumenter kan en möjlig effekt vara en minskning av onödiga returer av plagg som inte passar. En annan potentiell effekt är att konsumtion av kläder som tillverkas under tvivelaktiga arbetsvillkor minskar. Att informera konsumenter om klädesplaggens tillverkning kan påverka vilka kläder som konsumeras. Det i sin tur kan leda till förbättrade arbetsförhållanden för arbetare inom konfektions- och textilindustrin.

En tänkbar negativ följdeffekt av att konsumenter provar plagg virtuellt kan vara att behovet av fysiska butiker minskar och butiksanställda därmed förlorar sina jobb.

9.2 Slutsatser

Syftet med Examensarbetet var att ta fram en lösning för virtuell klädprovning, vilket skulle realiseras genom att implementera en prototyp som låter användaren placera en digital version av ett klädesplagg på en digital version av sig själv. Det beslutades senare att en färdig 3D-modell av en människa skulle användas istället för en 3D-modell av den faktiska användaren.

Ett antal användningsfall bestämdes för att specificera funktionella krav närmare. Det första kravet var att användaren skulle kunna bläddra mellan olika plagg, vilket anses vara uppfyllt genom Browse-läget som beskrivs i resultatdelen. Användaren skulle också kunna navigera till ett slags provrum och välja ett plagg att prova. Också det anses vara uppfyllt genom provrummet och AR-funktionen. Att kunna hitta information om Clean Clothes Campaign var även det ett krav och bedöms vara delvis uppfyllt.

Referenser

[1] Clean Clothes Campaign, “Who we are”. [Online]. Tillgänglig: https://cleanclothes.org/about, hämtad: 2020-03-17.

[2] Clean Clothes Campaign, “Former Uniqlo garment workers attend flagship store opening in Denmark to highlight Uniqlo’s wage-theft”, 2019. [Online]. Tillgänglig: https://cleanclothes.org/news/2019/former-uniqlo-garment-workers-attend-flagship-store-opening-in-denmark-to-highlight-uniqlos-wage-theft, hämtad: 2020-03-17.

[3] Wikipedia, “3D modeling”, 2020. [Online]. Tillgänglig: https://en.wikipedia.org/wiki/3D_modeling, hämtad: 2020-04-22.

[4] A. Chopine, "Polygons: How 2D Becomes 3D," i *3D Art Essentials: The Fundamentals of 3D Modeling, Texturing and Animation*. Burlington, MA, USA: Elsevier Inc., 2011, kap. 3, ss. 21-24. [Online]. Tillgänglig: https://www.sciencedirect.com/science/article/pii/B9780240814711100037, hämtad: 2020-05-28.

[5] Wikimedia-användaren Chrschn ”The polygon mesh of dolphin” 27 mars 2007. [Online]. Tillgänglig: https://commons.wikimedia.org/wiki/File:Dolphin_triangle_mesh.png#mw-jump-to-license, hämtad: 2020-04-22.

[6] Wikipedia, “Fotogrammetri”, 2018, [Online]. Tillgänglig: https://sv.wikipedia.org/wiki/Fotogrammetri, hämtad: 2020-04-22.

[7] Wikipedia, “Förstärkt verklighet”, 2019, [Online]. Tillgänglig: https://sv.wikipedia.org/wiki/Förstärkt_verklighet, hämtad: 2020-04-22.

[8] Clothing — Digital fittings — Part 1: Vocabulary and terminology used for the virtual human body, ISO 18825-1:2016, 2016-07.

[9] Clothing — Digital fittings — Attributes of virtual garments, ISO 18831:2016, 2016-04.

[10] Size designation of clothes – Part 3: Size labelling based on body measurements and intervals, EN 13402-3:2017, 2017.

[11] Wikipedia, “Android (operativsystem)”, 2020, [Online]. Tillgänglig: https://sv.wikipedia.org/wiki/Android_(operativsystem), hämtad: 2020-04-29.

[12] Google Developers, “Guide to app architecture”, 2019, [Online]. Tillgänglig: https://developer.android.com/jetpack/docs/guide, hämtad: 2020-05-02.

[13] Google Developers, “Activity”, 2020, [Online]. Tillgänglig: https://developer.android.com/reference/android/app/Activity, hämtad: 2020-05-02.

[14] Google Developers, “Fragments”, 2019, [Online]. Tillgänglig: https://developer.android.com/guide/components/fragments, hämtad: 2020-05-02.

[15] Google Developers, “Understand the Activity Lifecycle”, 2020, [Online]. Tillgänglig: https://developer.android.com/guide/components/activities/activity-lifecycle, hämtad: 2020-05-02

[16] Google Developers, “LiveData Overview”, 2020, [Online]. Tillgänglig: https://developer.android.com/topic/libraries/architecture/livedata, hämtad: 2020-05-05.

[17] SQLite, “Most Widely Deployed and Used Database Engine“, [Online]. Tillgänglig: https://www.sqlite.org/mostdeployed.html, hämtad: 2020-05-28.

[18] R. Meier, I. Lake, "Creating and Using Databases," i *Professional Android*, 4 uppl. Indianapolis, IN, USA: John Wiley & Sons, Inc., 2018, kap. 9, ss. 282-284. [Online]. Tillgänglig: https://books.google.se/books?id=Xu5qDwAAQBAJ&lpg=PP1&hl=sv&pg=PA281#v=onepage&q&f=false, hämtad: 2020-05-28.

[19] Android Developers Blog, “Android’s commitment to Kotlin”, [Online]. Tillgänglig:

https://android-developers.googleblog.com/2019/12/androids-commitment-to-kotlin.html, hämtad: 2020-05-28.

[20] Google Developers, “Android Studio”, 2020, [Online]. Tillgänglig: https://developer.android.com/studio, hämtad: 2020-05-07.

[21] Google Developers, “Run apps on the Android Emulator”, 2020, [Online]. Tillgänglig: https://developer.android.com/studio/run/emulator, hämtad: 2020-05-19.

[22] Google Developers, “Create and manage virtual devices, 2019, [Online]. Tillgänglig: https://developer.android.com/studio/run/managing-avds, hämtad: 2020-05-07.

[23] Google Developers, “Configure your build”, 2020, [Online]. Tillgänglig: https://developer.android.com/studio/build, hämtad: 2020-05-20.

[24] Google Developers, “Fundamental concepts”, 2020, [Online]. Tillgänglig: https://developers.google.com/ar/discover/concepts, hämtad: 2020-05-28.

[25] Google Developers. “Sceneform overview”, 2020, [Online]. Tillgänglig: https://developers.google.com/sceneform/develop, hämtad: 2020-05-28.

[26] Adobe XD, “Let’s XD together”, 2020, [Online]. Tillgänglig: https://www.adobe.com/se/products/xd.html, hämtad: 2020-05-26.

[27] Ubiquity6, “Display.land 101”, 2020, [Online]. Tillgänglig: https://learn.display.land/display-land-101, hämtad: 2020-05-05.

[28] Blender, “The Software”, 2020, [Online]. Tillgänglig: https://www.blender.org/about/, hämtad: 2020-05-05.

[29] Atlassian, “The #1 software development tool used by agile teams”, 2020, [Online]. Tillgänglig: https://www.atlassian.com/software/jira, hämtad: 2020-05-28.

[30] Atlassian, “What is Git”, 2020, [Online]. Tillgänglig: https://www.atlassian.com/git/tutorials/what-is-git, hämtad: 2020-05-28.

[31] Slack Technologies, Inc., “The collaboration software that moves work forward”, 2020, [Online]. Tillgänglig: https://slack.com/intl/en-se/features, hämtad: 2020-04-07.

[32] Microsoft, “Navet för samarbete i Microsoft 365”, 2020, [Online]. Tillgänglig: https://www.microsoft.com/sv-se/microsoft-365/microsoft-teams/group-chat-software, hämtad: 2020-04-07.

[33] Wikipedia, “Haptik”, 2015, [Online]. Tillgänglig: https://sv.wikipedia.org/wiki/Haptik, hämtad: 2020-05-28.

[34] Unity, “Unity for all”, [Online]. Tillgänglig: https://unity.com, hämtad: 2020-05-28.

[35] Unity, “Plans and pricing”, [Online]. Tillgänglig: https://store.unity.com, hämtad: 2020-05-28

[36] Android Authority, “I want to develop Android apps — What languages should I learn?”, 2019, [Online]. Tillgänglig: https://www.androidauthority.com/develop-android-apps-languages-learn-391008/, hämtad: 2020-05-28.

[37] Wikipedia, “Scrum”, 2019, [Online]. Tillgänglig: https://sv.wikipedia.org/wiki/Scrum, hämtad: 2020-05-07.

[38] Codelabs, “Welcome to Codelabs!”, [Online]. Tillgänglig: https://codelabs.developers.google.com, hämtad: 2020-05-28.

[39] Codelabs, “Introduction to Sceneform”, 2019, [Online]. Tillgänglig: https://codelabs.developers.google.com/codelabs/sceneform-intro/index.html?index=..%2F..index#0, hämtad: 2020-05-28.

[40] Khaled Ghareeb, 12 september 2019. [Online]. Tillgänglig: https://unsplash.com/photos/-NyPn9up_7o, hämtad: 2020-06-05.

[41] FlippedNormals, “Claudia Rigged 002 – 3D Business Woman”, [Online]. Tillgänglig: https://flippednormals.com/downloads/claudia-rigged-002-3d-business-woman/, hämtad: 2020-05-11.

[42] Unsplash-användaren Mediamodifier, 7 maj 2020. [Online]. Tillgänglig: https://unsplash.com/photos/t8HiP3e5abg, hämtad: 2020-06-05.

[43] Celine Ruiz, 26 december 2017. [Online]. Tillgänglig: https://unsplash.com/photos/rr4bawLxOjc, hämtad: 2020-06-05.

[44] Chris Boese, 17 december 2019. [Online]. Tillgänglig: https://unsplash.com/photos/wE1cIk0skNw, hämtad: 2020-06-05.