

BACHELOROPPGAVE

Utvidet virkelighet for visualisering av vann- og avløpsrør

Augmented Reality for visualizing water- and sewer pipes

Arisa Seljestokken, Fredrik Garseg Mørk og Simen Osland Romstad

DAT190 Bacheloroppgave – Dataingeniør og informasjonsteknologi

Fakultet for ingeniør- og naturvitenskap

Institutt for datateknologi, elektroteknologi og realfag

Veileder: Adrian Rutle

Innleveringsdato: 2. juni 2020

TITTELSIDE FOR HOVEDPROSJEKT

<i>Rapportens tittel:</i> Utvidet virkelighet for visualisering av vann- og avløpsrør	<i>Dato:</i> 02.06.2020
<i>Forfattere:</i> Arisa Seljestokken, Fredrik Garseg Mørk og Simen Osland Romstad	<i>Antall sider uten vedlegg: 57</i>
	<i>Antall sider vedlegg: 5</i>
	<i>Antall sider totalt: 62</i>
<i>Studieretning:</i> Dataingeniør	<i>Antall disketter/CD-er: ingen</i>
<i>Kontaktperson ved studieretning:</i> Adrian Rutle	<i>Gradering: ingen</i>
<i>Merknader:</i> ingen	

<i>Oppdragsgiver:</i> Bouvet Norge AS	<i>Oppdragsgivers referanse:</i>
<i>Oppdragsgiver kontaktperson:</i> Øyvind Litlere	<i>Telefon:</i> +47 915 32 684

<i>Sammendrag:</i> <p>Bachelorprosjektet gikk ut på å utvikle en mobilapplikasjon for fremstilling av vann- og avløpsrør i utvidet virkelighet. Applikasjonen kan bli brukt for arbeidere som unngår disse rørene når de graver eller bygger nye bygninger. Dagens løsningen er foreløpig i 2D. Prosjektet gir et forslag for en 3D løsning som gjør det enklere å tolke rørdatabasen til vann- og avløpsetaten.</p>

Stikkord:

Mobilapplikasjon Utvidet virkelighet	Unity Engine Kartdata	.NET Microsoft Azure Spatial Anchors
---	--------------------------	---

FORORD

Denne rapporten dokumenterer bakgrunnen og arbeidet bak bachelorprosjektet *Utvidet virkelighet for visualisering av vann- og avløpsrør*. Rapporten beskriver prosessen å utvikle en applikasjon for fremstilling av vann og avløpsrør (VA-rør) under bakken i utvidet virkelighet. Prosjektet ble utført våren 2020 av datastudentene Arisa Seljestokken, Fredrik Garseg Mørk og Simen Osland Romstad som det avsluttende emnet i bachelorutdanningen.

Vi ønsker å rette en stor takk til Bouvet for en utrolig spennende og meningsfull bacheloroppgave å jobbe med. Spesielt takk til Fredrik Christoffer Chrislock for veiledning og oppfølging i prosjektet. Takk til vann- og avløpsetaten i Bergen kommune som ga oss rørdatabasene som vi benyttet til å utvikle applikasjonen.

En siste takk til vår interne veileder ved Høgskulen på Vestlandet, Adrian Rutle, for gode innspill og veiledning gjennom prosjektperioden.

INNHALDSFORTEGNELSE

FORORD	3
FORKORTELSER	9
1 INNLEDNING	10
1.1 Mål og motivasjon	10
1.2 Kontekst	11
1.3 Avgrensning	11
1.4 Ressurser	11
1.5 Oppbygging av rapporten	12
2 PROSJEKTBESKRIVELSE	13
2.1 Praktisk bakgrunn	13
2.1.1 <i>Prosjekteier</i>	13
2.1.2 <i>Tidligere arbeid</i>	13
2.1.3 <i>Metode</i>	14
2.1.4 <i>Initielle krav</i>	14
2.1.5 <i>Initiell løsnings-idé</i>	14
2.2 Relatert arbeid	15
2.2.1 <i>Prosjekt: WaterGo</i>	15
2.2.2 <i>Universitetet av Nottingham</i>	15
2.2.3 <i>REDUCT</i>	15
2.2.4 <i>ZoArg</i>	16
2.2.5 <i>vGIS</i>	16
2.2.6 <i>One-call</i>	17
2.2.7 <i>Alec Pestov</i>	17
2.2.8 <i>Diskusjon</i>	18
3 ANALYSE OG TILNÆRMING	18
3.1 Utvidet virkelighet	18
3.1.1 <i>Hva er utvidet virkelighet?</i>	18
3.1.2 <i>Komponentene i utvidet virkelighet</i>	19
3.1.3 <i>Ulike typer utvidet virkelighet</i>	20
<i>Projiserende kartlegging</i>	20
<i>Superimosisjon</i>	20
3.2 Forslag til løsning	21
3.2.1 <i>Unity med Azure Spatial Anchors for kryssplattform</i>	21
3.2.2 <i>Unity med andre rammeverk for smartmobil</i>	21
3.2.3 <i>Unity med Hololens 2</i>	21
3.2.4 <i>Adobe Aero for smartmobil</i>	22
3.2.5 <i>Diskusjon av alternativene</i>	22
3.2.6 <i>Valgt løsning</i>	23
3.3 Valg av verktøy	23

3.3.1 Unity	23
3.3.2 C#	24
3.3.3 Visual Studio	24
3.3.4 Microsoft Azure Spatial Anchor	24
3.3.5 AR Foundation	24
3.3.6 Google ARCore	24
3.3.7 Apple ARKit	25
3.4 Valg av hjelpemidler	25
3.4.1 Git	25
3.4.2 Google Disk	25
3.4.3 Slack	25
3.5 Valg av bibliotek	26
3.5.1 CoordinateSharp	26
3.5.2 Newtonsoft	26
3.5.3 ESRI Shapefile-importer	26
3.5.4 Google Elevation API	26
3.6 Metodikk i prosjektet	27
3.6.1 Utviklingsmetodikk	27
3.6.2 Designprinsippene	28
3.6.3 Prosjektplan	28
Kommunikasjon	28
3.6.4 Risikovurdering	29
Kobling mellom AR-verden og GPS-koordinater	29
Bruk av GPS-koordinater	29
Ressursleveranse	29
Manglende kompetanse	31
Kommunikasjon	31
Valg av rammeverk og biblioteker	31
Balansering av prosjekt og rapport	31
Forståelsen av oppgaven	32
Tidsbruk	32
Bruk av utstyr	32
3.7 Evalueringsplan	33
4 DESIGN OG ARKITEKTUR	33
4.1 Design av applikasjon	33
4.1.1 Produktdesign	33
4.1.2 Utfordringer	34
4.2 Arbeidsflyt	34
4.3 Programvarearkitektur	35
4.4 Utformingsmodeller	36
4.4.1 Brukstilfelle	36

<i>Skanne og se rør</i>	36
<i>Se tidligere anker</i>	37
<i>Laste ned data</i>	37
<i>Manuell kalibrere</i>	37
4.4.2 <i>Klassediagram</i>	38
4.4.3 <i>Sekvensdiagram</i>	38
4.5 <i>Mappestruktur</i>	40
<i>Plugins</i>	40
<i>Prefabs</i>	40
<i>Scripts</i>	40
<i>Resources</i>	41
4.6 <i>Bruk av teknologi</i>	41
4.6.1 <i>Importering av shapefiler</i>	41
4.6.2 <i>Spatial Anchor - skanning av området</i>	41
4.6.3 <i>AR Camera</i>	41
4.6.4 <i>Compass API - synkronisering av kompass-retning og Unity's z-retning</i>	42
4.6.5 <i>Plassering av anker og rør</i>	42
4.7 <i>Database</i>	42
4.7.1 <i>Metoder for ankeret i skyen</i>	43
<i>CreateAnchor</i>	43
<i>SaveAnchorAsync</i>	43
<i>PlaceAnchorOnDevice</i>	43
<i>PlacePipes</i>	43
4.7.2 <i>Livssyklus til et Spatial Anchor</i>	43
4.8 <i>Sikkerhet og personvern</i>	44
5 EVALUERING OG RESULTAT	44
5.1 <i>Evalueringsmetode</i>	44
5.1.1 <i>Semi-strukturert intervju</i>	45
5.1.2 <i>Kravspesifikasjoner</i>	45
5.1.3 <i>Funksjonell testing</i>	45
5.1.4 <i>Testing med VA-etaten</i>	45
5.1.5 <i>Etter-evaluering</i>	46
<i>GitHub</i>	46
<i>Arbeid</i>	47
<i>Kommunikasjon</i>	47
5.2 <i>Evalueringsresultat</i>	47
5.3 <i>Prosjekresultater</i>	48
6 DISKUSJON	49
6.1 <i>Konsekvenser</i>	49
6.2 <i>Refleksjon</i>	50

7 KONKLUSJON OG VIDERE ARBEID	50
7.1 Konklusjon	50
7.2 Videre arbeid	51
7.2.1 Manuell kalibrering	51
7.2.2 Hente shapefilene direkte inn fra VA-etatens database	52
7.2.3 Skape og gjenkjenne ankre	52
7.2.4 Informasjon om rørene	52
7.2.5 VA-sensorer	52
7.2.6 Rørdesign basert på rørtype	53
8 LITTERATURLISTE	54
9 VEDLEGG	58
9.1 Risikoliste	58
9.2 Testing av applikasjonen	59
9.3 Brukermanual	60
9.4 Brev fra ekstern veileder	62
FIGURLISTE	
1.1 VA-etatens logo	10
1.2 En illustrasjon av dagens løsning av VA-kartet i Gemini Portal	10
2.1 Spytheunderground 2016	16
2.2 Alec Pestov sin løsning fra vGIS	17
3.1 Sofa blir plassert i AR	19
3.2 Bruk av AR innenfor medisin	20
3.3 Unity game engine 2020	23
3.4 Scrum-iterasjon	27
3.5 Trello kanban-board sprint 1	32
4.1 Applikasjonens logo på Android	33
4.2 Applikasjonens logo på iOS	33
4.3 Arbeidsflyt	34
4.4 Arbeidsflyt, utvikler-utkast på iOS	35
4.5 Generell arkitektur for programvaren	36
4.6 Brukstilfellene	37
4.7 Klassediagram av applikasjonen	38
4.8 Sekvensdiagram av applikasjonen	39
4.9 Mappedstrukturen i prosjektet	40
4.10 Mappedstrukturen i prosjektet	41
4.11 Anker som en nord-pil	42
4.12 Spatial Anchors oversikt i Azure-portalen	43
5.1 Ali Mekki og Simen O. Romstad tester applikasjonen	46
7.1 Bachelorgruppens sammensetning av Google maps og rørdatabasen som ble tilsendt	51
7.2 WaterGo, sommerprosjektet til Norkart 2017	52
7.3 Oversikt over alle type rør	53
7.4 Feilhåndtering	54

TABELLISTE

0.1 Forkortelser	8
3.1 Gantt-diagram	30
9.2 Risikoliste	58

FORKORTELSER

Forkortelse	Ord
AR	Augmented Reality
API	Application Programming Interface
DD	Decimal Degrees (desimalgrader)
DMS	Degree-minutes-seconds
HVL	Høgskulen på Vestlandet
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
GPS	Global Positioning System
SAS	Spatial Anchor Services
UTM	Universal Transverse Mercator
VA	Vann- og avløp
VR	Virtual Reality

Tabell 0.1 Forkortelser

1 INNLEDNING

I denne delen av rapporten blir bakgrunnen for prosjektet introdusert. Delkapitlene under vil gi et overblikk over prosjektets mål, motivasjon, kontekst, avgrensninger og ressurser.

1.1 Mål og motivasjon

Vann- og avløpsetaten i Bergen har ansvaret for å forsyne byens befolkning med rent drikkevann hele året. Etaten bidrar gjennom utstrakt avløpsrensing til en ren byfjord og et godt nærmiljø. VA-etaten arbeider kontinuerlig med å forbedre vannkvaliteten i Bergen, og har et mål om å være et ledende fagmiljø på vann- og avløp i Norge. Håndtering av vannforsyning og avløpsrensing er et stort ansvarsområde med en rekke utfordringer. Nye risikoområder som kontinuerlig oppstår setter krav om en kontinuerlig utvikling, der nye teknologier tas i bruk (Bergen kommune, 2019).



Vann- og avløpsetaten

Figur 1.1 VA-etatens logo (hentet fra Vann- og avløpsetaten, u.å.).

Vann- og avløpskartet blir stort sett benyttet av alle som jobber i VA-etaten og av virksomheter som samarbeider med VA-etaten i ulike prosjekter. «Kartet blir benyttet av saksbehandlere som skal godkjenne nye utbygginger, driftsoperatører som skal registrere utført arbeid på objekter rundt i Bergen kommune, og rørleggere som trenger data for å kunne legge planer for sitt arbeid for private kunder» (Kraft, 2020).

Med dagens løsning i Gemini Portal har de som benytter VA-kartet muligheten til å ta med seg tilstrekkelige mengder data ut i felt. Løsningen er et kartbasert brukergrensesnitt med en nettbasert applikasjon som kan benyttes på mobil, nettbrett eller PC (Powel, 2020). Alle rørdatabasene er vist på et 2D kart. VA-etaten mener at denne løsningen utfører jobben tilstrekkelig, men at det er rom for forbedringer.



Figur 1.2 En illustrasjon av dagens løsning av VA-kartet i Gemini Portal (hentet fra VAnytt, 2016).

Målet med dette bachelorprosjektet var å utvikle en applikasjon som fremstiller vann- og avløpsrør i utvidet virkelighet. Det vil si at applikasjonen kombinerer både det virkelige og virtuelle, at applikasjonen er interaktivt i sanntid og at grafikk-modellene blir vist i 3D. Applikasjonen skulle bidra til en enklere hverdag for alle som benyttet VA-kartet i arbeidsdagen sin. Spesielt gjaldt dette for de som utførte gravearbeid, vedlikehold og fornying av rør. Motivasjonen i prosjektet var å utvikle en løsning på VA-kartet gjennom å tenke innovativt og å bruke tilgjengelige teknologier i markedet.

1.2 Kontekst

Før prosjektets start fantes det små ulemper med løsningen for VA-kartet. Vann og avløpsetaten tenkte at fokuset i bachelorprosjektet ikke handlet om å fikse på de små ulempene, men å heller lage en helt ny løsning. I den nye løsningen skulle gruppen tenke nytt og innovativt, og bruke ressurser som allerede var tilgjengelig i markedet. Den nye løsningen skulle gjøre det mulig å se rør i sanntid gjennom kamera-applikasjonen. Dette ville gjøre arbeidsoppgavene for de som brukte VA-kartet mer behagelige.

1.3 Avgrensning

I et prosjekt som dette fantes det naturligvis noen avgrensninger som kunne ha påvirkning på prosjektets resultat. Disse avgrensningene var definert av tidsperioden, ressursene og omfanget til prosjektet.

Prosjektgruppen hadde lite relevant erfaring med data og verktøy som ble brukt i prosjektet. Gruppen hadde ikke arbeidet med kartdata eller utvidet virkelighet tidligere. Dermed måtte gruppen bruke en del arbeidstimer på å bli kjent med dataene, og verktøyene som ble brukt i prosjektet.

Siden prosjektets endelige resultat var påvirket av tilgjengelige ressurser og arbeidsomfang, måtte gruppen på forhånd gjøre en vurdering på hva nøkkel-funksjonaliteten skulle være. Dette ble til å visualisere vann- og avløpsrør i utvidet virkelighet. Med dette målet kunne gruppen levere en brukbar prototype til sluttdatoen. Andre funksjoner ville bli tilleggsfunksjoner som kunne blitt implementert, om det var tid til overs.

1.4 Ressurser

Gjennomføring av prosjektet krevde en del ressurser og verktøy. Heldigvis for prosjektgruppen stilte oppdragsgiveren, Bouvet, med kontor. I tillegg hadde gruppen tett samarbeid med en ekstern veileder, Fredrik Christopher Chrislock, som har erfaring innenfor spillutvikling i Unity. Prosjektgruppen ble tildelt kartdata fra VA-etaten i området rundt Høgskulen på Vestlandet (HVL) som ble benyttet i prosjektet. Gruppen stilte derfor sterkt da det gjaldt tilgjengelige ressurser i

form av kunnskap og material. Prosjektgruppen hadde også en tett kommunikasjon med veilederen, Adrian Rutle, som ga gruppen en god oppfølging i prosjektet og rapporten.

Prosjektet krevde at gruppen hadde tilgang til forskjellige smartmobiler og nettbrett. Både Bouvet og HVL tilbydde dette. Som verktøy til selve applikasjonsutviklingen benyttet gruppen Unity sammen med programmeringsspråket C#. Visual Studio var kode-editoren gruppen brukte til dette. For deling av kode brukte gruppen versjonskontrollsystemet Git. Google Disk ble brukt for deling av dokumenter og filer. Valget av disse verktøyene, bibliotekene og hjelpemidlene blir gjennomgått mer detaljert i kapittel 3.

1.5 Oppbygging av rapporten

Denne rapporten beskriver løsningen som prosjektgruppen presenterte for VA-etaten, på vegne av Bouvet. Rapporten er delt opp i ni kapitler, der de ulike kapitlene har hvert sitt fokusområde.

Kapittel 1 : En kort introduksjon til prosjektet med oversikt over avgrensninger og ressurser.

Kapittel 2 : En detaljert beskrivelse av prosjektet, samtidig som initielle krav og løsningsidé.

Kapittel 3 : Diskusjon rundt alternative løsninger og teknologier, sammen med prosjektets arbeidsmetode og evalueringsplan.

Kapittel 4 : En detaljert beskrivelse av produktdesign, teknologibruk og programvarearkitektur.

Kapittel 5 : Evaluering av prosjektet og resultatene fra dette. Prosjektets resultat basert på krav fra oppdragsgiver og VA-etaten.

Kapittel 6 : Diskusjoner og konsekvenser rundt prosjektets resultat.

Kapittel 7 : Prosjektets konklusjon og videre arbeid.

Kapittel 8 : En liste over kilder og referanser som brukes i rapporten.

Kapittel 9 : Et vedleggskapittel, en liste over risikoer i prosjektet og agenda for testingen, brukermanual for applikasjonen og brev fra ekstern veileder.

2 PROSJEKTBEKRIVELSE

Dette kapittelet beskriver prosjektet sin opprinnelse og utførelse. Det blir gitt svar på hvilke veier gruppen har valgt å gå, og hvorfor valgene har passet best til problemet oppgaven skal løse. Det blir også drøftet tidligere relevant arbeid.

2.1 Praktisk bakgrunn

2.1.1 Prosjekteier

Bouvet har ikke arbeidet med denne idéen før oppstart av dette bachelorprosjektet. Det var VA-etaten som kontaktet Bouvet og spurte om det var mulig å lage en løsning på problemet deres. Bouvet har hatt et internt ønske om å bruke mer ny teknologi i sine prosjekt og så muligheten til å gjøre det i dette prosjektet. Siden problemet ikke var stort og det ikke var hastverk med å få en løsning, gjorde Bouvet dette til en bacheloroppgave som prosjektgruppen fikk. Bouvet var både oppdragsgiver og prosjekteier. Selskapet stod derfor ansvarlig for utførelsen av oppdraget.

Siden VA-etaten har hatt samarbeid med Bouvet tidligere, gikk de til dem med problemet. Det at Bouvet ville bruke utvidet virkelighet (AR) i dette prosjektet, kom fra Bouvet sitt ønske om å bli bedre kjent med denne teknologien. Med alle dataene om vann- og avløpsrør som var tilgjengelig så Bouvet mange muligheter.

For VA-etaten var det viktig å kunne gjøre arbeidet med å lokalisere rør i bakken lettere. Før oppstart av prosjektet hadde VA-etaten brukt 2D-modeller og kart til å finne rør. Dette var tidkrevende og bestod ofte av å måtte grave med spade. Med store og unøyaktige gravemaskiner var det lett for å ødelegge ledningene og å lage lekkasje i rørnettet.

2.1.2 Tidligere arbeid

Bachelorgruppen var de første som skulle jobbe med en AR-løsning til VA-etaten i Bergen, og måtte derfor starte fra bunnen av. For å unngå å bruke tid på å lage grunnleggende funksjonalitet selv, brukte gruppen tid på å skaffe seg så mye informasjon som mulig fra andre lignende prosjekt, og om annet relatert arbeid som kunne være nyttig.

I arbeidet rundt å konvertere VA-etaten sin database om rør-informasjon, hadde gruppen også tatt i bruk kode fra ESRI (Internasjonal leverandør av programvare for geografisk informasjon) sitt bibliotek for å importere shapefile til Unity (Csharpdoc, 2019). En shape-fil er en fil som inneholder data og informasjon om rør under bakken som koordinater, dimensjoner og rørtyper. I tillegg brukte gruppen dokumentasjon fra ESRI til å utarbeide egne metoder i kildekoden (ESRI, 1998).

2.1.3 Metode

I dette prosjektet ble en mulighetsstudie (feasibility study) brukt til testingen som ble utført sammen med VA-etaten. Målene til en mulighetsstudie er å grundig forstå alle aspekter ved et prosjekt. Det innebærer å være klar over potensielle problemer som kan oppstå under implementering av prosjektet, og å avgjøre om prosjektet er verdt å gjennomføre. En metode for å evaluere en mulighetsstudie er et semi-strukturert intervju. Det er et intervju der spørsmålene er bestemt på forhånd. Disse spørsmålene er ikke nøyaktig formulert, men i form av en beskrivelse. Det er viktig å liste opp fordeler og ulemper i forkant av en slik gjennomførelse (Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., & Wesslén, A., 2012). Testingen vil bli gjennomgått mer detaljert i kapittel 5.

2.1.4 Initielle krav

Produkteier var Bouvet, men kravene kom fra VA-etaten. Kravene var løst definert og dermed kunne gruppen spesifisere kravene selv. Å bruke AR-teknologi var det riktige valget for dette prosjektet. Gruppen har brukt tid på å samle informasjon om hva gode krav for oppgaven kunne være og kom frem til noen få konkrete punkter:

- Et proof of concept for at mobilapplikasjonen skal kunne bli brukt av arbeiderne i felt for å lettere finne rør under bakken. Applikasjonen skal kunne gjenkjenne hvor brukeren er og plassere grafiske rør i riktig posisjon.
- Utvikler-versjonen av applikasjonen skal være tilgjengelig for autoriserte brukere på iOS, iPadOS og Android.
- Applikasjonen skal inneholde shapefiler som blir visualisert.

2.1.5 Initiell løsnings-idé

Allerede den første idéen var en AR-løsning, der en skulle kunne se visualiserte rør som lå under bakken. Løsnings-ideen var en applikasjon på en smarttelefon eller nettbrett. Denne applikasjonen skulle arbeiderne kunne bruke ved lokalisering av rør rundt i Bergen sine gater.

Applikasjonen skulle bruke lokasjonsdata fra GPS til å hente ned nærliggende rør fra en database. Databasen skulle inneholde rørene sine koordinater sammen med andre rørddata. Brukeren skulle holde mobilen foran seg med kamera pekende ned mot veien, og skulle kunne se grafiske modeller av rørene. De grafiske rørene skulle ha farge som symboliserte hvilke type rør de var. For at det skulle bli lettere å se dybden ble det anbefalt å bevege på kamera for å få riktig dybdefølelse. Det kunne bli lagt på en tekst eller en annen form for visualisering som fortalte hvor langt rørene lå under bakken.

VA-etaten sin database av rør-informasjon er en lukket kilde. Bare godkjente personer kan få tilgang siden databasen inneholder sensitiv informasjon som allmennheten ikke skal kunne se. Det første steget i løsningen var å bruke databasen med informasjon om vann- og avløpsrør og

overføre dette inn i applikasjonen. Dette skulle skje automatisk ved oppstart av appen. Hvilken informasjon som skulle bli overført var bestemt ved bruk av GPS og innlogging.

Hvilken plattform applikasjonen skulle være tilgjengelig på var ikke bestemt. Det stod mellom Microsoft Hololens eller smartmobil og nettbrett. Med Hololens skulle skjermen som var innebygd i brillene bli brukt, mens på smartmobiler skulle kameraet bli brukt. Basert på flere grunnlag viste det seg tidlig for gruppen at det ene alternativet var bedre enn det andre. Gruppen valgte derfor å gå vekk fra Hololens og heller satse på smartmobil og nettbrett. Med en smartmobil kan en bruke GPS til å lokalisere relativt nøyaktig hvor enheten befinner seg. Deretter skulle brukeren kalibrere rørene i forhold til kumlokk. Idéen om kalibrering til kumlokk vil bli nøyere diskutert i kapittel 8.

2.2 Relatert arbeid

Litteratur som gruppen hadde brukt til kilder i dette bachelorprosjektet var ulike nettsteder, digitale leksikon og noen digitale bøker. Det fantes lite litteratur om lignende prosjekt, men problemet med å finne vannrør under bakken har lenge eksistert. Dette problemet kan en gjenkjenne i utallige artikler og tidligere forskning. Utvidet virkelighet har blitt mye brukt før, for eksempel i Pokémon Go, IKEA Place og Snapchat. Dette gjorde det enklere å finne ressurser som kunne være til hjelp for utvikling.

2.2.1 Prosjekt: WaterGo

Sommeren 2017 hadde et team i Norkart, BorderGo, sammen med sommerstudenter laget en Android-applikasjon med Google Tango, der de viste rør under bakken i Lillehammer. I 2017 var Google Tango en ny teknologi, men ble i senere tid erstattet av etterfølgeren ARCore. Norkart skrev om at de hadde problemer med å koble sammen GPS-verdenen mot AR-verdenen. Deretter la de til en kalibrerings-funksjon for brukeren. Denne funksjonen skulle gi mer nøyaktig posisjon på rørene. I tillegg til 3D-verdenen hadde de med 2D-kart av rørene som ga mer oversikt (Norkart, 2017).

2.2.2 Universitetet av Nottingham

For å ta et eksempel fra 2002, før utvidet virkelighet var et kjent begrep for de fleste, jobbet et forskerlag ved Universitetet av Nottingham på en mulig løsning for visualisering av rør under bakken. «Et forskerlag ved University of Nottingham har utviklet et prototype system som legger et sanntids omgivelsesbilde sammen med et databilde av det som ligger nedgravd» (TU, 2002). Dette var relevant, men ikke noe gruppen hadde hatt bruk for i dette prosjektet.

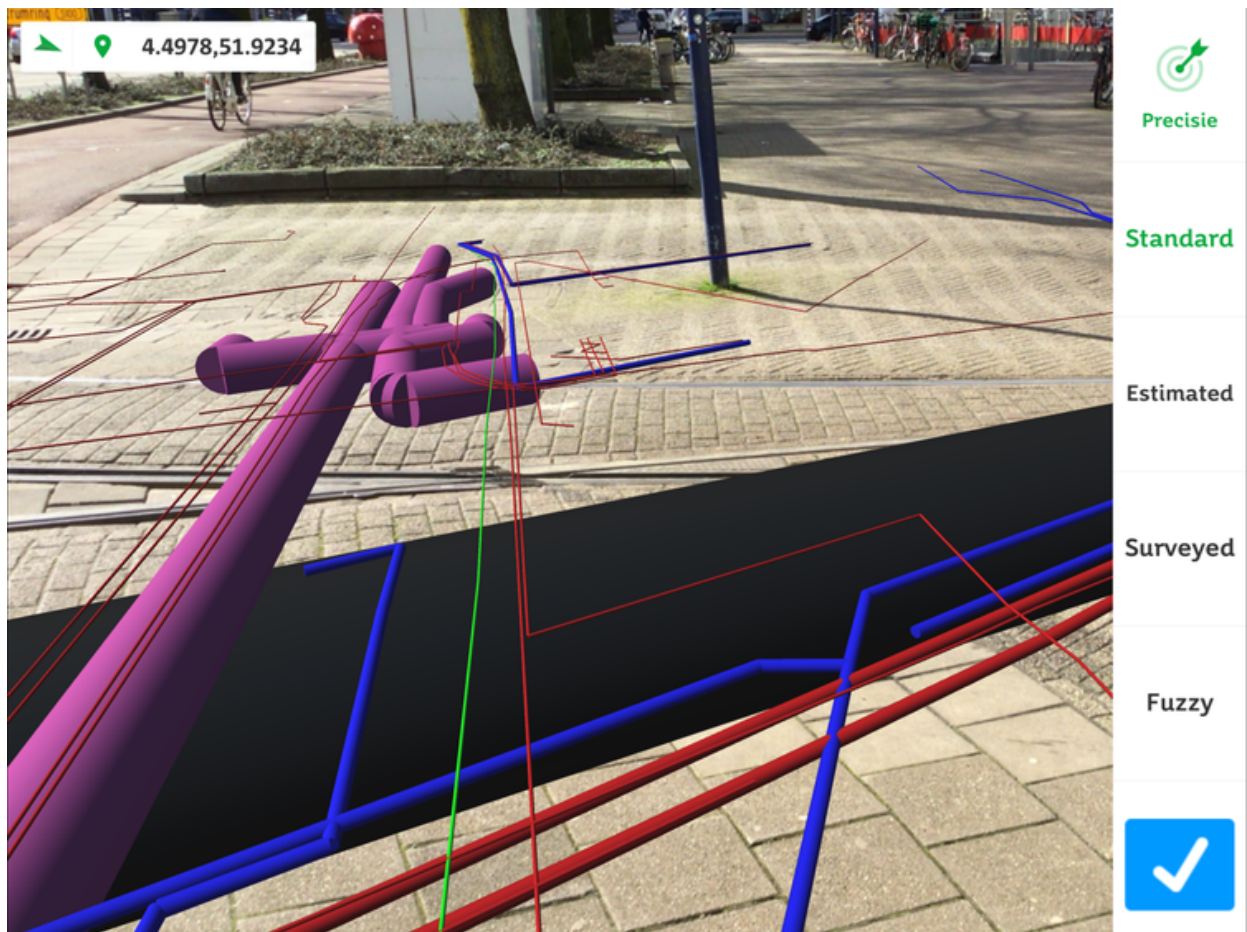
2.2.3 REDUCT

REDUCT er et selskap som jobber med å få inn rørteknologi i Norge. Selskapet utviklet et instrument som nøyaktig kan måle posisjonen av rør liggende under bakken, uten å måtte grave. Instrumentet var et annet eksempel på relevant arbeid, men det hadde et motsatt problem; å

kunne lokalisere rør som allerede eksisterer for å unngå skade på eksisterende infrastruktur mens nye rør utbygges (VA Nytt, 2016).

2.2.4 ZoArg

ZoArg arbeidet med et lignende prosjekt i 2016 og 2017. Deres mål var å lage en applikasjon på iPad som fremstilte alle typer rør og ledninger under bakken i AR. Applikasjonen skulle bli brukt til alle typer arbeid under bakken, og ikke kun for arbeidet med vann- og avløpsrør. ZoArg la vekt på å kunne estimere dybde, standard dybde, og faktisk målt dybde. Figur 2.1 er en skjermdump fra applikasjonen til ZoArg. På bildet er det flere typer rør og ledninger som blir visualisert med ulike farger og størrelser.



Figur 2.1 Spytheunderground 2016 (hentet fra ZoARG, 2018).

2.2.5 vGIS

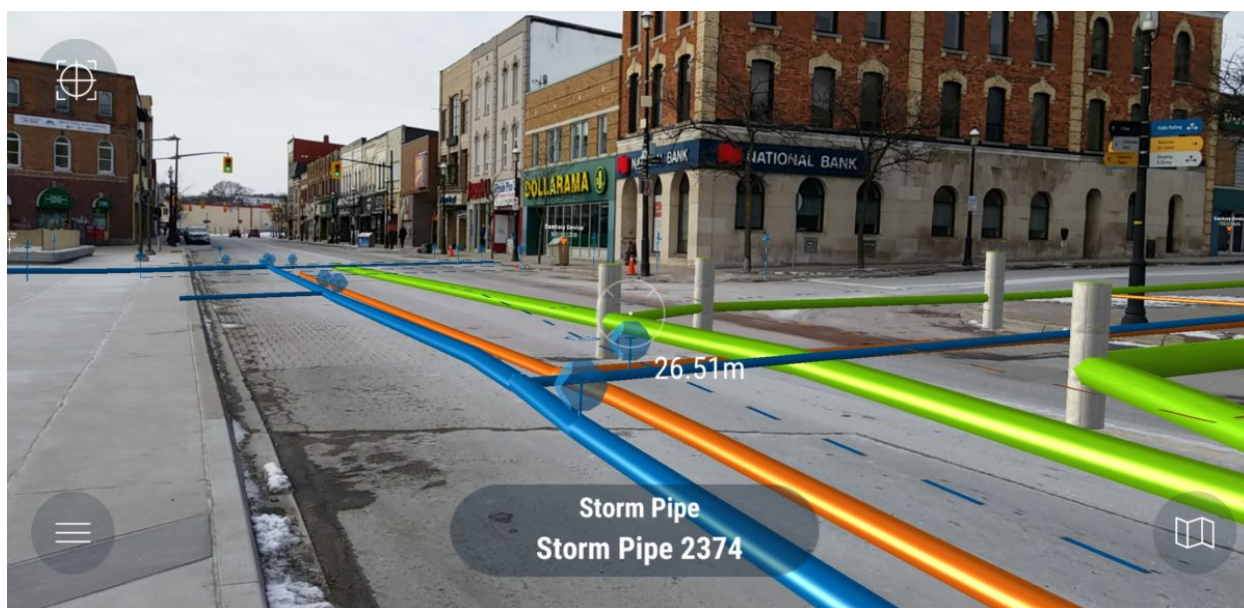
vGIS har utviklet et AR system for kommuner og offentlige sektorer. vGIS Utilities konverterer 2D kart om til synlige hologrammer av rør under bakken. Systemet skulle gi brukerne synlig infrastruktur på en naturlig måte. De påstod at det kunne spare aktive brukere opp til 12-20 timer i måneden (vGIS, 2020).

2.2.6 One-call

Applikasjonen deres skildret tydelig hva som var under bakken med bilder av fargekodede rør og kabler, kanaler og mangler. Applikasjonen viste deres nøyaktige plasseringer i forhold til sanntidsbildet av overflaten. For å forholde seg til sanntidsvisningen fra kameraet måtte kartdataene være i et digitalt format i databasen. For at 3D-objekter skulle bli vist på skjermen måtte det være en tjenesteleverandør og en database tilstede. En tjenesteleverandør er en virksomhet som tilbyr databaserte tjenester til kunder over et nettverk. Ved hjelp av denne sammen med en database ble 3D-objektene vist i utvidet virkelighet. I følge One-Call ga forholdet mellom strukturer over bakken og infrastruktur under bakken flere fordeler for arbeidere som ingeniører, planleggere, entreprenører, eiendomsbesitter og beredskapspersonell (Griffin, J., 2015).

2.2.7 Alec Pestov

Alec Pestov jobber for vGIS og har presentert en løsning som bruker data samlet inn av DGT Associates. Løsningen skapte et tydelig 3D-bilde ved å bruke Microsofts Hololens-teknologi. Bildet skulle hjelpe eiendomsbesittere og designere å ta informerte beslutninger. Løsningen tilbyr brukeren muligheten til å se planer fra kontoret sitt, eller gå rundt på det faktiske stedet. Pestov brukte Google maps, Bing, Esri og OpenStreetMaps kombinert i det nye systemet. Dette prosjektet startet som de aller fleste prosjekter, og måtte hente inn nøyaktig kartdata fra kommunen. Deretter har vGIS-systemet konvertert dataene til et meningsfullt bilde for Hololens og en smartmobil. Applikasjonen fungerte slik at brukeren kan ta på Hololens eller bruke en smarttelefon i området. Brukeren skulle se ned mot veien og skulle få opp en fargekodet projeksjon av de faktiske rørene under bakken (DGT Associates, 2020).



Figur 2.2 Alec Pestov sin løsning fra vGIS (hentet fra DGT Associates, 2020).

2.2.8 Diskusjon

Kapitlene ovenfor viser at det finnes mange eksisterende prosjekter som ligner på gruppens prosjekt. Noen av disse prosjektene har blitt brukt som inspirasjon i prosjektet, til blant annet brukergrensesnittet på applikasjonen. Gruppen har vært i dialog med Norkart, og det var de som gjorde gruppen oppmerksom på problemet med å koble sammen GPS-verdenen mot AR-verdenen, som vedlagt i 9.1, Risikoliste. Siden gruppen var oppmerksom på dette, var gruppen forberedt til å bruke en del arbeidstimer på problemet.

Det var flere ulikheter som skilte bachelorprosjektet fra relaterte arbeid som er nevnt ovenfor. Blant annet teknologien og verktøyene som ble valgt til bachelorprosjektet. Valget av disse verktøyene vil bli gjennomgått mer detaljert i kapittel 3. Bachelorprosjektet har hatt begrenset tilgang på ressurser siden det ble utviklet av studenter. De aller fleste teknologiene, bibliotekene og verktøyene som ble brukt i prosjektet var gratis. Det som ikke var gratis ble dekket av Bouvet. I de fleste relaterte arbeidene var prosjektene utviklet av erfarne utviklere, som hadde fått hjelp av eksperter i fagområdet. Disse utviklerne har hatt et enklere arbeidsmiljø, med mer ressurser som de kan bruke på prosjektet. Dette bachelorprosjektet var også kun et *proof of concept* i en mulighetsstudie, som ikke skulle bli solgt eller være helt klar til bruk ved endt prosjekt. Det skulle kun være en fungerende prototype.

3 ANALYSE OG TILNÆRMING

I dette kapitlet blir det gitt en grundig introduksjon til utvidet virkelighet. Det blir forklart hva teknologien består av, ulike typer av den og hvordan den har blitt en del av hverdagen til mange mennesker. Senere blir det diskutert ulike løsninger til prosjektet, valg av verktøy, fremgangsmåte og evalueringsplan.

3.1 Utvidet virkelighet

3.1.1 Hva er utvidet virkelighet?

Utvidet virkelighet er en levende kopi av synet på den fysiske og ekte verden. Elementene i synet blir utvidet med datagenererte sensoriske inndata som lyd, video eller grafikk (Crystalinks, 2014). Teknologien gjør det mulig å legge på ekstra lag med informasjon på virkeligheten, ved å supplere med de sensoriske inndataene. AR kan sees på som en hybrid mellom virkeligheten og virtuell realitet (VR). AR og VR er to lignende konsepter, men AR skiller seg fra VR ved at det er interaktivt i sanntid i ekte verden. VR baserer seg kun på en datakonstruert virkelighet (Deloitte, u.å.).



Figur 3.1 Sofa blir plassert i AR (hentet fra Wikipedia Commons, 2016).

Med AR har en muligheten til å visuelt endre på objekter og personer på en enkel måte, ved å rette kameraet mot dem. På figur 3.1 ser en hvordan AR kan bli brukt i varehandelen. På figuren er det en virtuell sofa på nettbrettet, uten at sofaen faktisk eksisterer i virkeligheten. Dette kan være nyttig for en person som skal kjøpe seg en sofa, og kunne tenkt seg å se hvordan sofaen vil se ut i stuen før personen bestemmer seg for å kjøpe den.

Et annet populær bruksområde for AR er på mobilapplikasjonen Snapchat. Mobilapplikasjonen bruker AR-teknologi til å lage filtre og effekter gjennom brukernes kamera i sanntid (Deloitte, u.å.).

3.1.2 Komponentene i utvidet virkelighet

I AR finnes det to hovedkomponenter som blir brukt for å fremstille objekter. Disse er kamera og skjerm. I innmatingen fra kameraet er det viktig å identifisere unike punkter i omgivelsen som er lett å kjenne igjen, men samtidig er unike. Gjenkjenning av rom eller objekter er viktig for at AR-programvaren skal få romforståelse. Skjermen som brukeren ser på, enten mobilen, nettbrettet, dataen eller hololensen, gir brukeren et sammensatt syn på den virtuelle og den virkelige verden.

Programvaren består av flere deler. Den viktigste delen er datasynet. Dette er en vitenskapelig del som viser til datamaskinens forståelse av digitale bilder og videoer. Kameraet gir programvaren en strøm med bilder av omgivelsene, som må bli tolket om til en 3D-forståelse. Nevrale nettverk er den mest effektive måten å gjøre dette på. Datasynet anskaffer, bearbeider, analyserer og forstår disse bildene, trekker ut dataene sine og gjør dette om til tall og symboler. Dette bruker eksempelvis Snapchat og Facebook når de skal gjenkjenne personer på bilder/kamera.

3.1.3 Ulike typer utvidet virkelighet

Markørbasert and markørløs

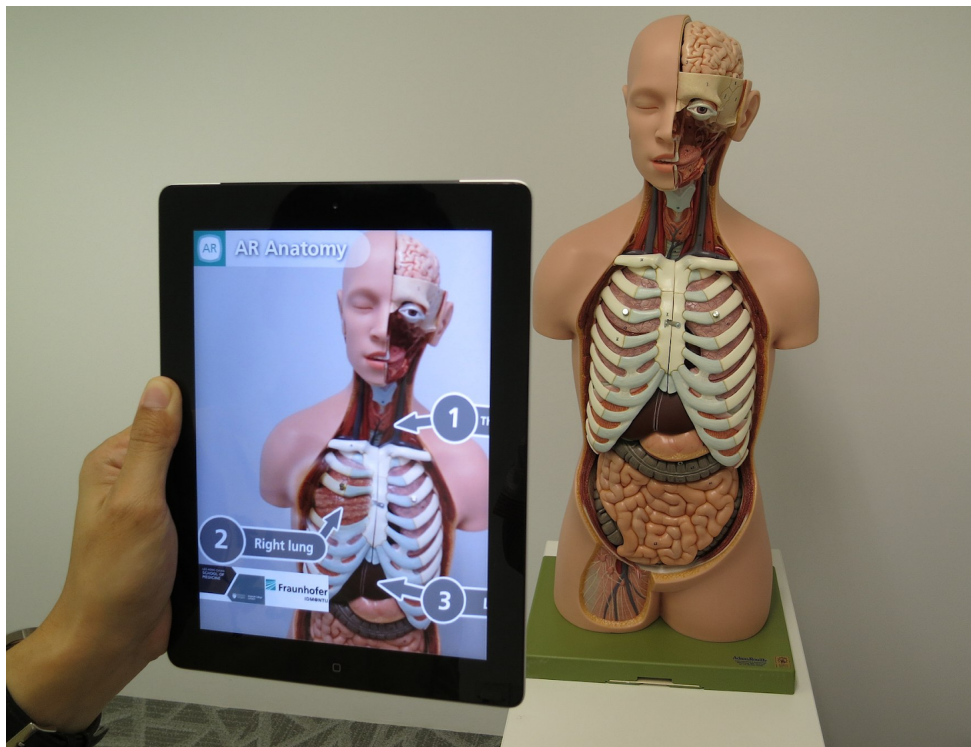
Det går an å skille AR i to typer, markørbasert og markørløs. Markørbasert AR går ut på å bruke en markør i omgivelsen som en aktivator, mens markørløs ikke bruker dette. I markørbasert AR kan markøren enten være et 2D-bilde med visuelle funksjoner eller naturlige gjenstander i det virkelige miljøet. Markørløs AR bruker GPS og sensorteknologi for å kontrollere det relative posisjonsforholdet mellom de virkelige og de virtuelle objektene (Cheng, J. & C., Weiwei, K. & C., 2017).

Projiserende kartlegging

Projiserende kartlegging er en undertype av utvidet virkelighet, som benytter seg av datasyn-teknologi, og kombinerer 3D-sensorsystemer som dybdekamera. Det gjør det mulig å benytte seg av projisert kartlegging. Der blir et projisert bilde kartlagt på de fysiske objektet og skaper direkte digitale lag (White, G., 2018).

Superimposisjon

AR basert på superimposisjon gir et annet syn på objekter, enten ved å erstatte objektene en ser, eller ved å bytte ut objektet med et forstørret bilde. Objektgjenkjenning er viktig når en skal bruke denne teknikken. (Cheng, J. & C., Weiwei, K. & C., 2017).



Figur 3.2 Bruk av AR innenfor medisins (Wikipedia Commons, 2012)

3.2 Forslag til løsning

Det første steget var å velge riktig utviklingsmiljø, rammeverk og biblioteker for oppdraget. For fremstilling av 3D-objekter i en utvidet virkelighet fantes det ulike verktøy. Eksempelvis Spark AR Studio, Unity Real-time development platform, AR scapes, ARCore og flere andre. Valget ble vurdert ut i fra hvilke verktøy som best mulig grad kunne ta i mot koordinater og vise dette som grafiske rør. Tilslutt stod det mellom fire alternative løsninger. Disse blir diskutert i dette delkapitlet.

3.2.1 Unity med Azure Spatial Anchors for kryssplattform

Den første løsningen var å bruke Unity og Azure Spatial Anchors gjennom ARCore og ARKit for iOS og Android. ARKit og ARCore er program-utviklingssett som gjør det mulig å visualisere 3D-objekter for iOS og Android-enheter. Fordelen med denne løsningen var at både Unity, ARCore og ARKit var gratis tilleggspakker i Unity. Unity hadde god støtte for AR, med flere millioner brukere (Unity, 2020). Script-filer i Unity ble skrevet i C#, et programmeringsspråk som ligner på Java. ARKit og ARCore er relativt nytt og kunne bli en utfordring å lære på så kort tid. Det var likevel lett å få tak i hjelp både på nettet og hos Bouvet. Spatial Anchors hadde derimot få kilder og hjelpemiddel på nettet, med kun Microsoft sin dokumentasjon tilgjengelig for utviklere.

3.2.2 Unity med andre rammeverk for smartmobil

Flere eksempler på rammeverk er Vuforia og Wikitude. Wikitude er lokasjon-basert apputvikling for AR, og ble kåret til beste utviklingsverktøyet i 2017 (Wikitude, 2017). Ulempen med denne løsningen var prisen. Det kostet 499 Euro for demopakke i 30 dager, og langt mer for full tilgang. Vuforia er designet for bildegjenkjenning, og er ikke like nyttig for dette prosjektet.

3.2.3 Unity med Hololens 2

Utvidet virkelighet kunne bli vist på nettbrett, mobil og Hololens. Hololens er et AR-headsett som tilbyr interaksjon med AR-verdenen ved hjelp av fingre, syn, bevegelse og sted. Hololens har ikke innebygget GPS og kan ikke bruke stedstjeneste. Det kunne være vanskelig og ressurskrevende ute, dersom gruppen skulle finne ut hvordan dette kunne muliggjøres (Microsoft, 2020). Hololens er noe de færreste har eller bruker, mye på grunn av prisen. I tillegg måtte sluttbrukeren fått opplæring til både å bruke de grunnleggende egenskapene til brillene, og hvordan applikasjonen skulle bli brukt.

Hololens er designet for å bli brukt hovedsakelig innendørs. Dette er fordi Hololens bruker Spatial Anchoring til å gjenkjenne hvor den er i et rom, for å så plassere grafiske objekter der de er programmert til å være (Microsoft Azure, 2019). Dette er optimalt til bruk innendørs, men er også med litt større utfordring mulig ute. Da ville prosjektet vært avhengig av eksterne plug-ins som løser GPS-problemet. Omgivelsene utendørs er i mye større grad i endring enn det er innendørs.

3.2.4 Adobe Aero for smartmobil

Ved valget av hvilke utviklingsmiljø som skulle bli tatt i bruk, ble også Adobe Aero vurdert. Dette var kun i betaversjon for pc-brukere (desktop-mode) og hadde derfor langt færre brukere enn Unity (Adobe, 2020). Adobe Aero går ut på å skanne et område og deretter legge ut et 3D-objekt. Det kunne blitt ressurskrevende forarbeid i uteområdet og ble sterkt anbefalt til å bli brukt inne.

3.2.5 Diskusjon av alternativene

En smartmobil er noe de fleste har tilgjengelig til enhver tid. Arbeiderne, som graver rør i Bergen, har som regel en smartmobil lett tilgjengelig. Dessuten er en mobil enkel å ta i bruk. Alle bruker som regel en smartmobil i sin hverdag til å surfe på internett, snakke i telefon eller kommunisere på andre måter. Majoriteten har en grunnleggende forståelse på hvordan en skal bruke mobilen, og trenger derfor kun en innføring i hvordan applikasjonen skal brukes. Dette gjorde at valget lente seg mot å bruke smartmobil.

Hololens-løsningen falt fort vekk da det ikke var enkelt å bruke GPS i teknologien. Løsningen er avhengig av GPS for å plassere rørene. Hvis oppdraget var å visualisere 3D-objekter innendørs, hadde løsningen falt mer mot Hololens 2.

Adobe Aero er et nytt verktøy som er i betaversjon, noe som betyr at det ikke er ferdig utviklet. Det hadde ikke så mange brukere, sammenlignet med Unity på dette tidspunktet. Adobe Aero tilbydde svært lite muligheter til å lage skript til 3D-objektene. Dette gjorde det enklere å håndtere, men begrenset mulighetene for utviklingen. Gruppen valgte ikke Adobe Aero fordi det var ikke mulig å oppgi hvor objekter skulle bli plassert med koordinater.

Unity er populært blant utviklere, dermed er det lettere å få hjelp på nettet og enkelt å stille spørsmål på spesialiserte forumer. I tillegg fikk gruppen hjelp av ekstern veileder i Bouvet som hadde god erfaring med verktøyet. Det ga prosjektet et godt utgangspunkt, og forsikret at prosjektgruppen ville klare å gå videre dersom gruppen trengte hjelp. Unity hadde i tillegg mange biblioteker som kunne bli benyttet i prosjektet. Eksempler på dette er ARCore og ARKit, som er utviklet av Google og Apple.

Microsoft sin Azure Spatial Anchors er et nytt verktøy for utvidet virkelighet, der en kan feste et anker til et punkt i et rom eller et koordinat. Ankeret blir værende og det går an å legge til objekter på ankeret, eller i nærheten av ankeret. Dette kan eksempelvis være et sylinder fra punkt til punkt, som representerer VA-rør. Disse ankrene og objektene blir lagret i en Microsoft Cloud, som kan aksesserer av flere brukere som har tilgang på skyen. Dette verktøyet passet best til hva gruppen trengte.

3.2.6 Valgt løsning

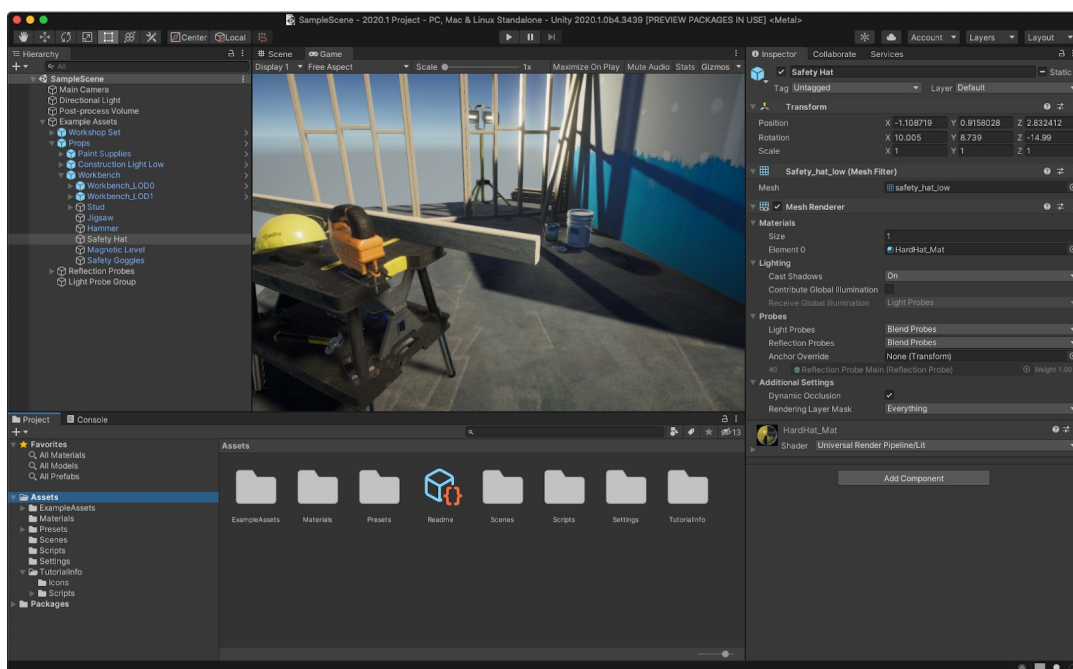
Etter å ha sett på alle fordelene og ulempene med de ulike alternative løsningene, valgte prosjektgruppen tilslutt å gå for løsning nummer 1: Unity med Azure Spatial Anchors gjennom ARKit for iOS-enheter og ARCore for Android-enheter.

3.3 Valg av verktøy

Til prosjektet ble utviklingsmiljøet Unity valgt sammen med språket C#. Prosjektgruppen benyttet Visual Studio IDE som kode-editor, og Git som versjonskontrollsystem. Rapporten og alle dokumenter ble lagret i Google Disk. Kommunikasjonen i gruppen sammen med veiledere foregikk gjennom direktemeldingsplattformen Slack. Teknologien som tok seg av AR på mobil var Microsoft sin Azure Spatial Anchor, som brukte ARKit-pakken for kamera interagering for iOS og ARCore for Android. Løsningen var en felles kodebase i C#, slik at programmeringen av språkene Swift og Kotlin kunne unngås. Den valgte løsning var mindre tidkrevende i forhold til de andre mulige løsningene.

3.3.1 Unity

Unity er en verdensledende 3D-utviklingsplattform i sanntid, og har verktøyene til å lage spill og publisere dem til mange forskjellige enheter. Unitys 3D-utviklingsplattform lar designere og utviklere samarbeide for å skape interaktive opplevelser (Unity, 2020). Unity ble anbefalt av ekstern veileder, som hadde god erfaring med verktøyet og kunne stille til hjelp dersom gruppen stod fast i en knipe. Dette sparte prosjektgruppen masse tid på, og gjorde arbeidet lettere.



Figur 3.3 Unity game engine 2020 (hentet fra Unity, 2020).

3.3.2 C#

C# (Uttales C sharp) er programmeringsspråket som brukes i Unity. Dette er et objektorientert språk fra Microsoft, utviklet for deres .NET plattform. .NET er en samling av teknologier utviklet av Microsoft med mål om å lettere utvikle større sammensatte systemer. Unity er i senere tid blitt en del av dette Microsoft-universet, med bruk av for eksempel C#. C# er sett på som en hybrid versjon av det beste C og C++ kan gjøre for et modernisert språk. C# har flere hjelpetrekk som f.eks. *Garbage collection*, som gjenopptar minne fra ubrukte objekter. Det type-sikre designet av språket gjør det vanskeligere å lese uinitierte variabler til indekserte tabeller utenfor deres grenser. Alle de primitive variablene arver fra én rot: "objekt-typen" (Microsoft, 2017).

Ved valget av Unity fulgte C# naturlig med. Heldigvis har C# vært gjennomgått på bachelorstudiet tidligere, slik at det ikke var helt ukjent for gruppen. I tillegg har C# mange lignende trekk som Java, et annet objektorientert språk prosjektgruppen har god erfaring med fra studiet.

3.3.3 Visual Studio

Microsoft Visual Studio er en fullstendig IDE for å kode, debugge, teste og deploye til hvilken som helst plattform. Dette verktøyet blir både brukt på MacOS og Windows, og er kompatibelt for kryssplattform (Microsoft, 2020). IDE-en ble valgt fordi den har støtte til .NET teknologi. Siden skripter i Unity blir skrevet i C#, passet kode-editoren godt til prosjektet.

3.3.4 Microsoft Azure Spatial Anchor

Microsoft Azure Spatial Anchor blir brukt til å utvikle realitets-applikasjoner. Disse applikasjonene blir brukt til å kartlegge, tilskrive og innhente presise interessepunkter. Applikasjonene kunne bli benyttet på flere plattformer: HoloLens, iOS og Android-enheter. Istandsatt orienterings-muligheter på tvers av områder for å hjelpe brukere å samarbeide mer effektivt. (Microsoft Azure, 2019). Denne teknologien ble anbefalt av gruppens ekstern veileder. Til gruppens fordel var teknologien lettvinnt å sette opp og ta i bruk.

3.3.5 AR Foundation

AR Foundation lar utviklere jobbe med AR-plattformer på en kryssplattform måte i Unity. Denne pakken presenterer et grensesnitt for Unity-utviklere å bruke, men implementerer ikke noen AR-funksjoner i seg selv. For å bruke AR Foundation på en målenhet, trenger brukeren også en egen pakke for den plattformen. I dette prosjektets tilfelle, ARKit og ARCore (Unity, 2020). AR Foundation var essensielt å ha med for å kunne bruke AR-interaksjoner i applikasjonen for mobilene.

3.3.6 Google ARCore

ARCore er Googles plattform for å bygge AR-opplevelser. Ved å bruke forskjellige API-er gjør ARCore at en mobil kan gjenkjenne sitt miljø, forstå verden og samhandle med informasjon. Noen

av API-ene er tilgjengelige på Android og iOS for å aktivere delte AR-opplevelser (Google developer, 2020). Googles ARCore ble valgt fordi dette gjorde det mulig å lage en AR-applikasjon for Android-enheter og det var mange som brukte dette fra før av.

3.3.7 Apple ARKit

Apple ARKit er den største AR-plattformen for iOS og iPadOS i verden. Den beskriver brukeropplevelser som legger til 2D- eller 3D-elementer til live-visningen fra kameraets enhet på en måte som gjør at elementene ser ut til å befinne seg i den virkelige verden. ARKit kombinerer bevegelse av enhetsbevegelser, kameraopptak, avansert scenebehandling og visnings-fasiliteter for å forenkle oppgaven med å bygge en AR-opplevelse (Apple, 2020). Apples ARKit ble valgt fordi plattformen gjorde det mulig å lage en AR-applikasjon for iOS-enheter, og det var mange som brukte dette fra før av.

3.4 Valg av hjelpemidler

3.4.1 Git

Git er et distribuert versjonskontrollsystem som lar utviklere loggføre filendringer under utvikling, slik at en utvikler kan gå tilbake til en tidligere versjon. Dette blir brukt sammen med Github, som brukes til oppbevaringssted for koden. Git er kjent for sitt gren-system for ikke-lineær utvikling. Forgrening betyr å avvike fra hovedlinjen i utviklingen, og fortsetter å jobbe på en ny gren uten å røre hovedlinjen (Chacon, S. og Straub B. 2014). Gjennom deltakelse på bedriftspresentasjoner har gruppen erfart at Git er mer brukt enn andre versjonskontrollsystemer, som Subversion og Mercurial. I tillegg hadde bachelorgruppen noe erfaring med Git fra før av. Det var en enkel og sikker måte å lagre koden og prosjektet på.

3.4.2 Google Disk

Google Disk er Google sin lagring og fildelingssystem på nett. Google Disk gjør det mulig å få tak i filene hvor som helst så lenge en har nettilgang. Systemet er dessuten enkelt og raskt, og har ingen begrensninger for filtyper. Gruppen benyttet denne løsningen fordi den tilbydde en oversiktlig måte å dele filer på. Ytterligere var systemet noe gruppen var godt kjent med.

3.4.3 Slack

Slack er en direktemeldingsplattform der samtaler mellom brukere skjer i kanaler. Kanalene er som regel organisert etter tema, prosjekt, team eller hva som er fornuftig for gruppene. Med disse kanalene blir ikke de rette personene utelatt og informasjon går ikke tapt. Det er også mulig å bruke tredjeparts applikasjoner for en bedre arbeidsflyt. Disse kan for eksempel være Google Disk og Trello (Slack, 2020). Det var enkelt å opprette nye kanaler, dersom det var behov for det, og like enkelt å slette utdaterte. Gruppen valgte å benytte Slack på grunn av programmets brukervennlighet, og positive erfaringer fra tidligere bruk.

3.5 Valg av bibliotek

3.5.1 CoordinateSharp

CoordinateSharp er et enkelt .NET Standard-bibliotek designet for å hjelpe med geografiske koordinater konvertering, formatering og stedsbaserte himmel-beregninger. CoordinateSharp har muligheten til å konvertere forskjellige typer koordinater frem og tilbake. Eksempelvis fra bredde- og lengdegrad til Decimal Degrees (DD), Degree-minutes-seconds (DMS) eller universal transverse mercator (UTM) (CoordinateSharp, 2020). I starten brukte gruppen en programmert matematisk beregning, men på grunn av unøyaktig kalkulering trengte gruppen et bibliotek som gjorde konverteringen nøyaktig på centimeter-nivå. Biblioteket ble brukt i prosjektet fordi det var lett tilgjengelig og var det eneste biblioteket som var gratis.

3.5.2 Newtonsoft

Newtonsoft er et JSON (JavaScript Object Notation) rammeverk i .NET som lar utviklere analysere JSON-filer og bruke den informasjonen i koden (Newtonsoft, u.å.). JSON er en enkelt tekstbasert måte å formatere dokumenter og data på. Den skal representere enkle datastrukturer, og være lett å lese for både mennesker og datamaskiner. Rørdataen bortsett fra koordinatene, måtte bli lest som en JSON-fil. For å kunne lese denne JSON-filen, trengte gruppen et rammeverk som oversatt dataen til informasjon som kunne bli lagret i en liste av rørene i applikasjonen. Rammeverket passet godt til dette formålet og var i tillegg brukt av mange utviklere i verden.

3.5.3 ESRI Shapefile-importer

ESRI shapefile-importer er et bibliotek skrevet for å kunne importere shapefiler til Unity (Csharpdoc, 2019). ESRI biblioteket ble valgt for å unngå å måtte konvertere til andre filformater. Dette biblioteket ble nyttig brukt i prosjektet for å importere viktige filer til applikasjonsbruket. Biblioteket var ikke ferdig til dette prosjektets bruk, dermed måtte gruppen utvide biblioteket og skrive egne metoder som var nødvendige for prosjektet.

3.5.4 Google Elevation API

Google Elevation API (Application Programming Interface) er et enkelt grensesnitt til å finne høyde på et gitt punkt på jorden (Elevation på engelsk). Høyden er antall meter over havet, og er utgangspunkt til posisjonene til rørene. Dette er Google sitt API som blir brukt i Google Maps (Google Developer, 2020). Google Elevation API ble brukt i dette prosjektet fordi Unity sin tilsvarende løsning var mindre nøyaktig enn nødvendig. Google Elevation API ga mer nøyaktige data og ble viktig for enhetens posisjon i forhold til rørene.

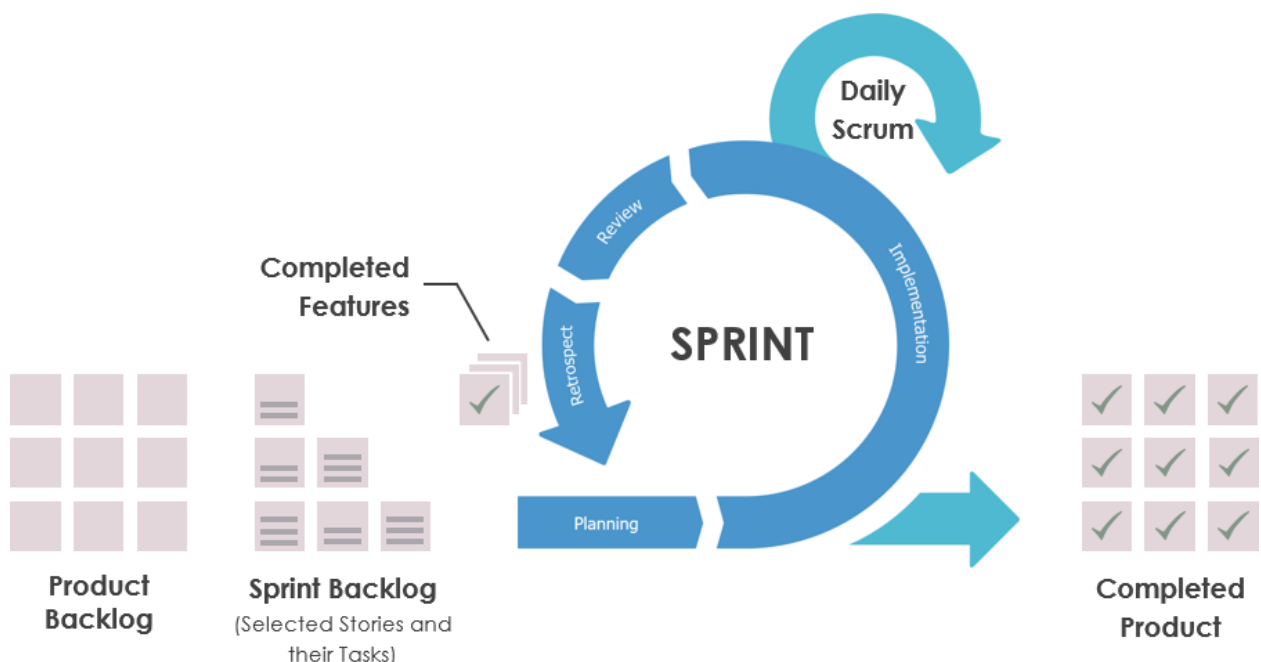
3.6 Metodikk i prosjektet

Det finnes mange smidige metoder som gir et team med utviklere en god arbeidsflyt. Disse metodene gir utviklere evnen til å kunne respondere raskt på endringer og takle usikkerheter umiddelbart. Metodene har som mål å forbedre kvaliteten på produktet, samtidig som de tilbyr en god arbeidsflyt for teamet.

3.6.1 Utviklingsmetodikk

Den smidige utviklingsmetoden, Scrum, ble valgt for dette prosjektet siden gruppen har tidligere erfaring med Scrum fra et fag på studiet, *DAT109 Systemutvikling*. Scrum-utviklerne har definert Scrum som: «Et rammeverk der mennesker kan adressere komplekse adaptive problemer, mens de produktivt og kreativt leverer produkter med høyest mulig verdi» (Schwaber, K. og Sutherland, J. 2017).

Scrum består av iterasjoner ved bruk av sprinter som foregår i normalt 2 ukers periode. Alle oppgaver som skal gjøres i nåværende sprint, havner i Backloggen på kanban-tavlen. En Kanban-tavle består av oppgaver som skal gjøres i sprinten, med en oversikt av *product backlog* (hvilke oppgaver produktet mangler), *sprint backlog* (hva som skal bli gjort i nåværende sprint), *todo* (det neste som skal gjøres), *on-going* (hva som blir gjort nå og av hvem) og *done* (hva som er gjort så langt i prosjektet). Hver oppgave ble kategorisert i farger hva oppgaven gjaldt, dette blir nærmere forklart i kapittel 3.6.4, Balansering av prosjekt og rapport. I hver sprint blir rollen Scrummaster, byttet på. *Scrummaster* sitt ansvar er å sørge for at oppgavene blir gjort, og at passende oppgaver havner i backloggen. Iterasjonene i Scrum skal sørge for at problemene i prosjektet blir tidlig oppdaget.



Figur 3.4 Scrum-iterasjon (Visual Paradigm, 2020).

3.6.2 Designprinsippene

Designprinsippene GRASP ble brukt i utviklingen av applikasjonen til å håndtere ansvarsfordelingen mellom klassene og objektene. Disse prinsippene skulle gjøre designet mer forståelig, fleksibelt og lett å vedlikeholde. De ulike reglene som er definert i GRASP ledet avgjørelsene for opprettelsen av klassene i applikasjonen på hver sin måte.

Her blir klassen *Anchor Manager* brukt som et eksempel for å forklare hvordan gruppen brukte GRASP. Klassene i applikasjonen blir grundigere gjennomgått senere i kapittel 4.4.2. Gruppen valgte at denne klassen skulle være skaperen i applikasjonen, siden den har ansvaret for å lage objektene. Samtidig er denne klassen også definert som kontrolleren. Den styrer hovedoppgavene i applikasjonen, og hvordan underklasser utfører sine ansvar.

For å få lav kobling i applikasjonen har gruppen hatt fokus på at klassene skal være lite avhengige av hverandre. Lav kobling bidrar til at klassene og objektene i applikasjonen kan bli gjenbrukt og endret på, med få konsekvenser.

Sammen med lav kobling er høy kohesjon viktig for leseligheten og gjenbruken av applikasjonen. Gruppen har prøvd å fokusere på denne regelen, men har prioritert lav kobling for å ha tid til å bli ferdig med applikasjonen. Høy kohesjon fokuserer på å minimalisere ansvaret for hver klasse, og fordeler gjerne store ansvarsoppgaver ut til flere klasser.

Reglene om polymorfisme, beskyttet variasjon og fabrikasjon har ikke vært behov for i denne applikasjonen og er derfor ikke blitt benyttet.

3.6.3 Prosjektplan

Prosjektplanen ble fremstilt ved hjelp av et Gantt-diagram som er vist under, i tabell 3.1. Utviklingsmetodikken Scrum baserte seg på sprinter som varte i 2 ukers intervaller. Disse sprintene startet i uke 11, og avsluttet i uke 23. I hver sprint var det forskjellige oppgaver knyttet til prosjektet. Dette sikret for fremgang i prosjektet. Gantt-diagrammet ble delt opp i tre deler: planlegging, rapport og prosjekt. Planleggingsfasen foregikk før eksamensuken i uke 10. Da gikk arbeidet på å forstå hva som skulle gjøres og forberedes til prosjektstart i uke 11. Gruppen arbeidet med prosjektet og skrev rapporten samtidig, for å ikke glemme ut arbeidet som skulle bli dokumentert i rapporten i ettertid. Oppgavene ble fordelt slik at gruppen kunne jobbe jevnt mellom prosjektet og rapporten.

Kommunikasjon

Hver sprint varte i 2 uker og foregikk fra mandag til mandag. Det var daglig oppdatering på hva som hadde blitt gjort siden sist, hva gruppemedlemmene holdt på med, og hva planen videre skulle bli. Dette var den vanlige *stand up*-en i Scrum-metodikk. På denne måten fikk hele gruppen

status på det som ble gjort, hver dag. I slutten av en sprint ble det holdt et møte der teamet diskuterte hva som ble gjort i sprinten, og hvilke nye oppgaver som skulle være i neste sprint.

Ukentlig kommunikasjon mellom gruppen og oppdragsgiver var avgjørende for å få riktige ressurser i forbindelse med prosjektet. For å vise framgang ga gruppen oppdateringer i form av bilder og videoer. Disse oppdateringene ble sendt til VA-etaten, Bouvet og HVL-veileder. Testingen av applikasjonen ble gjort i starten av mai. Møter med veileder ble holdt annenhver uke eller ved behov for å få svar på spørsmål rundt rapporten og arbeidet som ble gjort.

3.6.4 Risikovurdering

Risikohåndteringen er noe av det første som skal bli vurdert hver gang et nytt prosjekt opprettes. Dette ble gjort for å gjøres kjent med risikoer som kan oppstå under dette prosjektet. Vedlagt i kapittel 9.1 ligger risikolisten. Denne består av aktiviteter med en fare og årsak, med konsekvens og sannsynlighet i tallform fra en til fem, der en er lav og fem er høy. Dette utgjorde sammen (multiplisert) risikoen for faren i området 1 - 25. I tillegg ble det tatt et tiltak for å dempe risiko-tallene som ble evaluert på nytt.

Kobling mellom AR-verden og GPS-koordinater

Likt som tidligere relevante prosjekt, hadde dette prosjektet også utfordring med sammenkoblingen av AR-verdenen og den ekte verdens GPS-koordinater (Norkart, 2019). Norkart skriver at denne utfordringen var en av de største utfordringene i deres prosjekt. Risikoen for at det samme skulle skje i dette prosjektet var derfor stort.

Bruk av GPS-koordinater

Kartdataene levert av VA-etaten hadde en nøyaktighet på centimeter-nivå. Den samme nøyaktigheten har ikke smartmobiler. Under åpen himmel er unøyaktigheten for GPS-signaler på smartmobiler opptil 4.9 meter. Det fins flere grunner til at det også kan være enda verre; blokkering av signal, forstyrrende frekvenser, solstormer, satellitt vedlikehold og flere grunner nevnt i artikkelen om GPS-nøyaktighet (U.S. government, 2017).

Ressursleveranse

I uke 11, da prosjektet begynte for fullt, fikk bachelorgruppen kartdataene for Kronstad-området. Alle rørdataben i Bergen er unntatt offentligheten og dataen mottatt var kun for å vise at det var mulig å visualisere rørene i 3D. Grunnen til dette var sikkerheten om at ikke all data ble eksponert. Problemet var at det ikke var lett å tolke dataene. Flere mailer ble sendt frem og tilbake, og det tok flere dager før gruppen forsto hvordan dataene skulle bli brukt.

Dato: 20.03.2020

Ukenummer

Hovedoppgaver	Før	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Frister/ Komm entarer
Planlegging	Febr					Mar	s			April					Mai			Juni		
Danne gruppe																				1.12.19
Velge oppgave fra bedrift																				1.12.19
Kontakte personer																				
Skaffe veileder																				
Rapport																				2.6.20
Forside																				18.3.20
Forord																				
Innledning																				
Prosjektbeskrivelse																				
Design av prosjektet																				
Detaljert design																				
Evaluerings																				
Resultater																				
Diskusjon																				
Konklusjon og videre arbeid																				
Referanser																				
Appendix																				
Rettskriving																				
Prosjekt																				
Velge verktøy																				
Bli kjent med Unity																				
Bli kjent med shape-filer																				
Skrive import-script for shp																				
Bibliotek for tegning av 3D objekter																				
Koble gps og AR-koordinater sammen																				
Sette sammen komponentene																				
Testing																				

Tabell 3.1 Gantt-diagram

Manglende kompetanse

Ved prosjektstart hadde ingen i gruppen brukt Unity før. Dette ble løst ved introduksjon til verktøyet med gruppens eksterne veileder i Bouvet. Gruppen fikk to hele dager med kurs i grunnleggende Unity, og fikk dermed en god og innholdsrik intro til Unity. Som tidligere nevnt i kapittel 3.3 skrives skripten i Unity i språket C#. Dette hadde 2 av 3 medlemmer erfaring med fra tidligere. Sammen med kurs og egen research gikk det greit å benytte Unity som verktøy.

Kommunikasjon

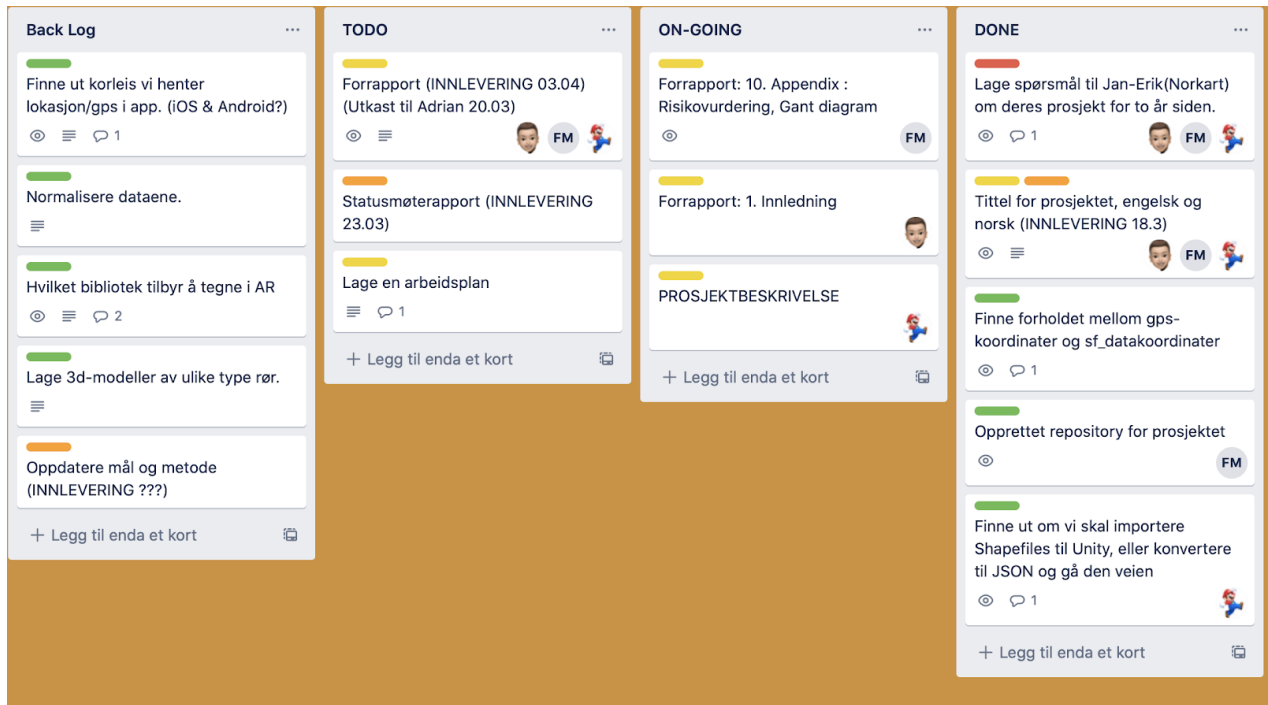
I uke 11 kom det unntakstilstander i Norge som varte ut siste semesteret i utdanningsløpet (Regjeringen, 2020). Dette forsinket arbeidet betraktelig, siden arbeidet ikke lenger kunne foregå på Bouvet sine kontorer, eller på HVL sine klasserom. Hjemmekontor ble løsningen og det var ikke like enkelt å samarbeide på bachelorprosjektet. Gruppen hadde et par sprint-møter før påsken. Da foregikk kommunikasjonen over videosamtaler og vanlige lydsamtaler. Etter påsken jobbet gruppen godt sammen hos hverandre.

Valg av rammeverk og biblioteker

Utvidet virkelighet er relativt nytt for smartmobiler. Det var ikke mulig for iOS-enheter å bruke ARKit før i 2017, da det ble tilgjengelig for iOS 11 (Apple, 2020). Det kunne vært vanskelig å finne et bibliotek for dette prosjektet, siden det ikke var mange som hadde gjort noe lignende tidligere. Det fantes noen få biblioteker, men de aller fleste var ikke gratis. Det ble dermed brukt en del tid på å finne gratis biblioteker.

Balansering av prosjekt og rapport

Gruppen synes at prosjektet var spennende å jobbe med. Å bruke for mye tid på prosjektet, og for lite på rapporten var dermed noe gruppen lett kunne endt opp med å gjøre. Kanban-tavlen på Trello viste hvilke oppgaver som måtte gjøres og hvilken prioritet (fra topp til bunn) oppgavene hadde. Tavlen hjalp gruppen med å balansere arbeidet jevnt mellom rapporten og prosjektet. Gruppen delte inn i fargene grønt for prosjektarbeid, gul for rapportarbeid, rødt for diverse og oransje for innleveringer for å lettere kunne se hva enkelte oppgave handlet om.



Figur 3.5 Trello kanban-board sprint 1

Forståelsen av oppgaven

Kommunikasjonen mellom kunden og arbeidstaker er svært viktig med tanke på prosjektet en utfører. Det er viktig å oppdatere kunden på prosjektet og stille spørsmål om det er slik utviklerne tenker, er det samme som kunden tenkte resultatet skulle bli. Ukentlig kommunikasjon mellom oppdragsgiver og utviklerne var derfor nødvendig ved gjennomføring av prosjektet.

Tidsbruk

Mennesker underestimerer bruken av arbeidsoppgaver. Forutsigelser om hvor mye tid som trengs for å fullføre en fremtidig oppgave, viser en optimisme-skjevhet og undervurderer tiden som trengs (Kahneman, D., Tversky, A., 1977). Det var derfor bruken av smidige metoden Scrum nevnt i kapittel 3.6.1 ble tatt i bruk. Metoden sørget for at gruppen arbeidet på en iterativ måte.

Bruk av utstyr

Dersom Unity programvaren ble oppdatert til en nyere versjon kunne prosjektet bli utdatert og ikke eksekverbar. Prosjektet er avhengig av riktig versjon av Unity for å kunne fungere optimalt. Det går an å refaktorere prosjektet til en gitt versjon, men det skal vurderes om det er nyttig å oppdatere til nyeste versjon. Prosjektet ble vist ved hjelp av ARKit for Apple og ARCore for Google (Android-brukere) nevnt i kapittel 3.3. ARCore kunne eksempelvis ikke bli brukt på en iOS-enhet og omvendt. Det var dermed viktig at prosjektet ble testet på flere enheter, for å sikre at det ble eksekvert riktig. Denne testen blir nøyere beskrevet i kapittel 5, Evaluering og resultat.

3.7 Evalueringsplan

Planen for testing foregikk på Kronstad-området der dataene befant seg, og av utviklerne selv, med og uten VA-etaten. Testing av applikasjonen ble filmet og tatt skjermopptak av for å analysere og dokumentere. På denne måten fikk prosjektgruppen en tilbakemelding på hva som måtte endres på, etter VA-etaten sine ønsker. Før testen, ble en agenda for møtedagen laget og gjennomført. Der fikk VA-etaten selv prøve applikasjonen og kunne prøve seg frem uten instruksjoner. Deretter ble VA-etaten fortalt hvordan applikasjonen hadde blitt laget, og hva som var tenkt videre. Agendaen for testingen ligger vedlagt i 9.2.

Foreløpig var det planlagt at testen skulle utføres i overgangen fra april til mai. Testen foregikk på en iPhone, iPad og en Android mobil. På denne måten ble det testet om applikasjonen fungerte som normalt eller om det var feil og mangler basert på enhet og operativsystem. Testingen og resultatet blir nøyere gjennomgått i kapittel 5 evaluering og resultat.

4 DESIGN OG ARKITEKTUR

Dette kapitlet gir en detaljert beskrivelse av applikasjonens design og programvarearkitektur.

4.1 Design av applikasjon

4.1.1 Produktdesign

Da det kom til produktdesign hadde verken oppdragsgiveren, Bouvet, eller VA-etaten noen kravspesifikasjoner på designet til løsningen. Gruppen fikk aldri sett hvordan dagens løsning på VA-kartet så ut. Dermed, ble applikasjonens design bestemt ut i fra gruppens egne idéer. Dette gjaldt alt fra applikasjonens navn, logo og brukergrensesnitt. Gruppen valgte å gi applikasjonen navnet *Virtual Pipes*, og brukte VA-etaten sin logo. I figur 4.1 ser en logoen på Android, og i figur 4.2 på iOS. Siden gruppen hadde lite erfaring da det kom til utvikling av mobilapplikasjon design var dette litt utfordrende. Gruppen var enige om å utvikle et brukervennlig design, som skulle være så intuitiv som mulig. Likevel var gruppen bevisste på at selv om designet var intuitivt for gruppen, betydde det ikke at det også var intuitivt for brukere av applikasjonen.



Figur 4.1 Applikasjonens logo på Android



Figur 4.2 Applikasjonens logo på iOS

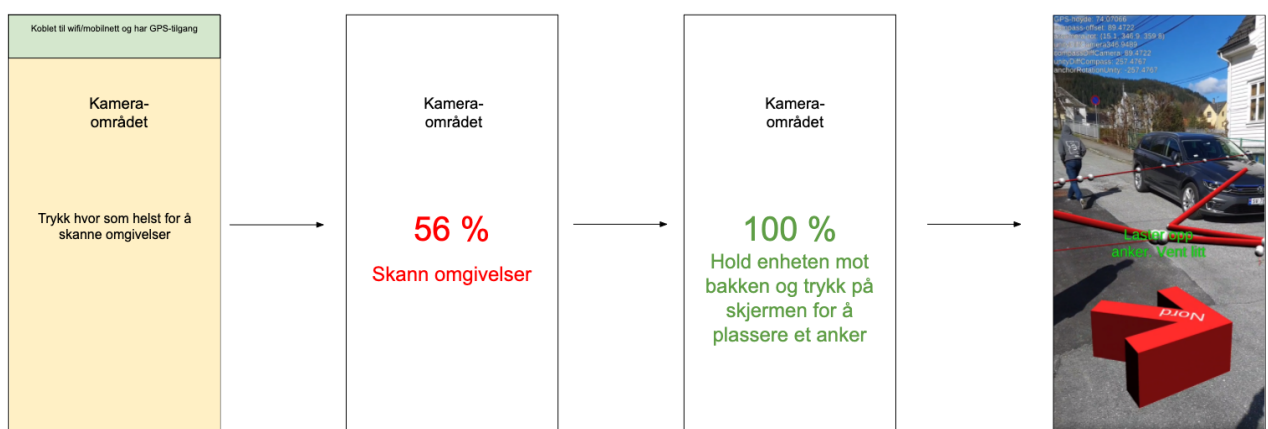
4.1.2 Utfordringer

Siden applikasjonen bruker AR-teknologi, og kombinerer det virtuelle med den virkelige verden, fantes det en rekke utfordringer under utvikling. Nedenfor ligger utfordringene punktvis og gir en beskrivelse av hvordan utfordringene har påvirket gruppens valg på applikasjonens design.

- Fremstillingen av vann- og avløpsrørene er virtuelle objekter. Hvordan skulle de se mest realistiske ut med tanke på form og størrelse?
Gruppen valgte å bruke sylindere til å visualisere rørene. Størrelsen på rørene var bestemt av diameteren som er hentet fra VA-etaten sine rørdatabaser.
- Hvordan skulle brukergrensesnittet være for at brukeren lett skulle ta i bruk applikasjonens hovedfunksjonalitet?
For at brukeren skulle kunne se den ekte verden kombinert med virtuelle rør i sanntid, var det viktig at fremstillingen var så elegant som mulig. Gruppen har valgt å designe applikasjonen slik at minst mulig område av skjermen bestod av tekst og andre UI. Hovedsakelig skulle skjermen kun bestå av AR.
- Hvordan skulle utviklerne få brukeren til å forstå at rørene lå under bakkenivå?
Dette kunne være litt utfordrende for en bruker som ikke var kjent med AR-teknologi. Gruppen valgte å plassere en liten tekst på røret som pekte mot enhetens kamera. Teksten informerte brukeren om avstanden i meter fra røret til enheten. Det ble også plassert små fargepunkter rundt på planer, for å lettere forstå at rørene ligger under bakken.

4.2 Arbeidsflyt

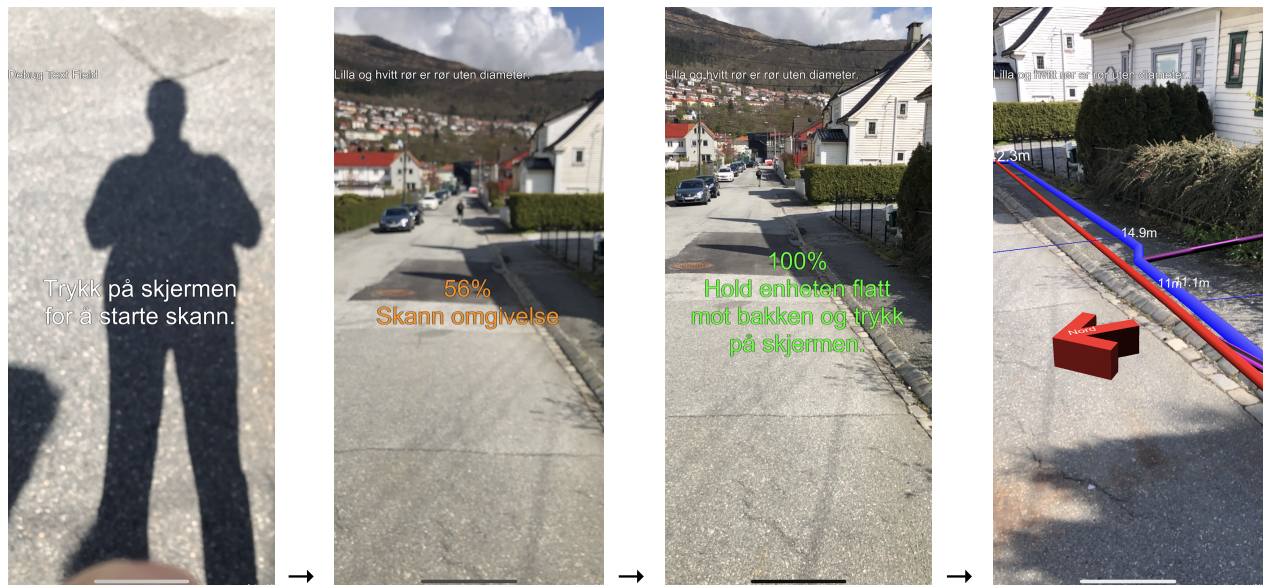
Hvis brukeren har tilkoblet GPS, internett og blåtann, skulle brukeren kunne gå gjennom en rekke handlinger for å kunne vise rørene gjennom kamera. I første steg står det en tekst som sier at brukeren kan trykke et sted på skjermen for å begynne skanningen av området.



Figur 4.3 Arbeidsflyt

For å skanne området skal brukeren bevege kameraet rundt. Da får Spatial Anchor Services (SAS) nok informasjon fra sensorene på enheten til å tolke området. Etter området blir tolket, legger SAS ut et anker som blir faststående i rommet. Når SAS har fått nok informasjon, skal teksten

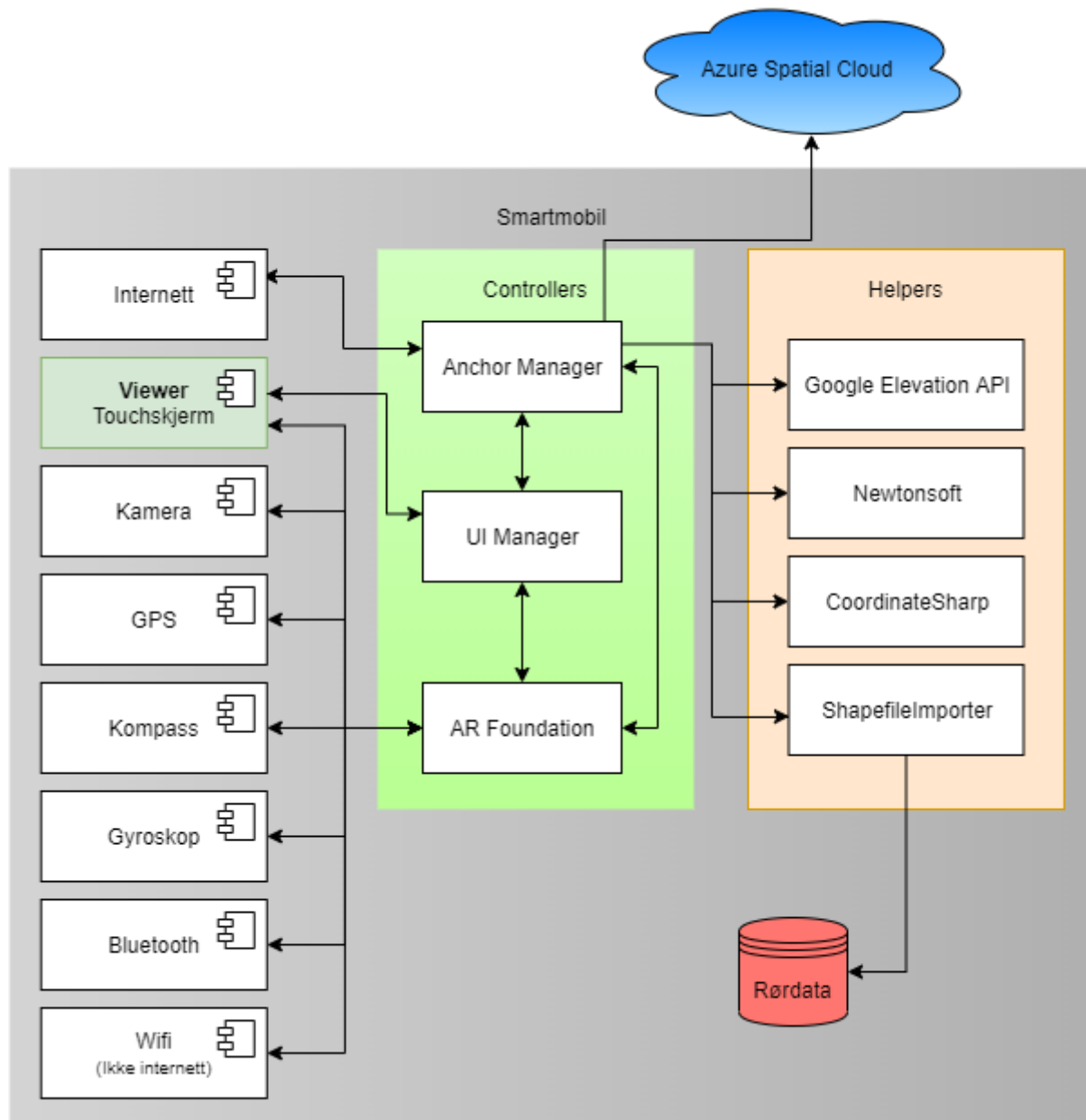
endre seg til: «Hold enheten mot bakken for å plassere et anker». Ankeret er en rød pil med teksten «Nord» på. Da vil ankeret bli plassert i rommet pekende nord. Basert på hvor ankeret er, vil applikasjonen finne ut hvor rørene er i forhold til ankeret i den ekte verdenen. Kort tid etter trykket kan brukeren se rørene rundt ankeret. Brukeren kan da gå rundt og utforske området for rør-data. I tillegg dukket det opp informasjon om hvor langt unna rørene er fra enheten.



Figur 4.4 Arbeidsflyt, utvikler-utkast på iOS

4.3 Programvarearkitektur

Det brukeren ser og interagerer med er vist i boksene under i figur 4.5. De fleste boksene er sensorer som gir indirekte data fra brukeren, om for eksempel posisjon og omgivelse. Den grønne boksen *Touchskjerm* er det brukeren ser og direkte interagerer med. Dette er skjermen på smartmobilen og viser frontend for brukeren. I den grønne boksen er kontrollere til applikasjonen. Det er her det meste av logikken til applikasjonen ligger. Hovedkontrolleren *Anchor Manager* er ansvarlig for skyen (som tar vare på ankrene), og tar i bruk hjelperne som er markert med oransje til høyre i figuren. I denne oransje boksen ligger de ulike bibliotekene som gruppen har tatt i bruk. Rørdataene blir eksempelvis lest og tolket av biblioteket *ShapefileImporter*.



Figur 4.5 Generell arkitektur for programvaren

4.4 Utformingsmodeller

4.4.1 Brukstilfelle

Brukstilfelle er hvilke oppgaver brukeren har i applikasjonen (OMG, 2017). Det er planlagt fire forskjellige brukstilfeller som applikasjonen skal ta for seg.

Skanne og se rør

Den første brukstilfellen er at brukeren starter applikasjonen ved å trykke på skjermen for å begynne å skanne omgivelsene. Når det er gjort skal brukeren trykke på skjermen for å opprette et anker. Deretter viser rørene automatisk og brukeren kan se rørene gjennom kameraet.

Se tidligere anker

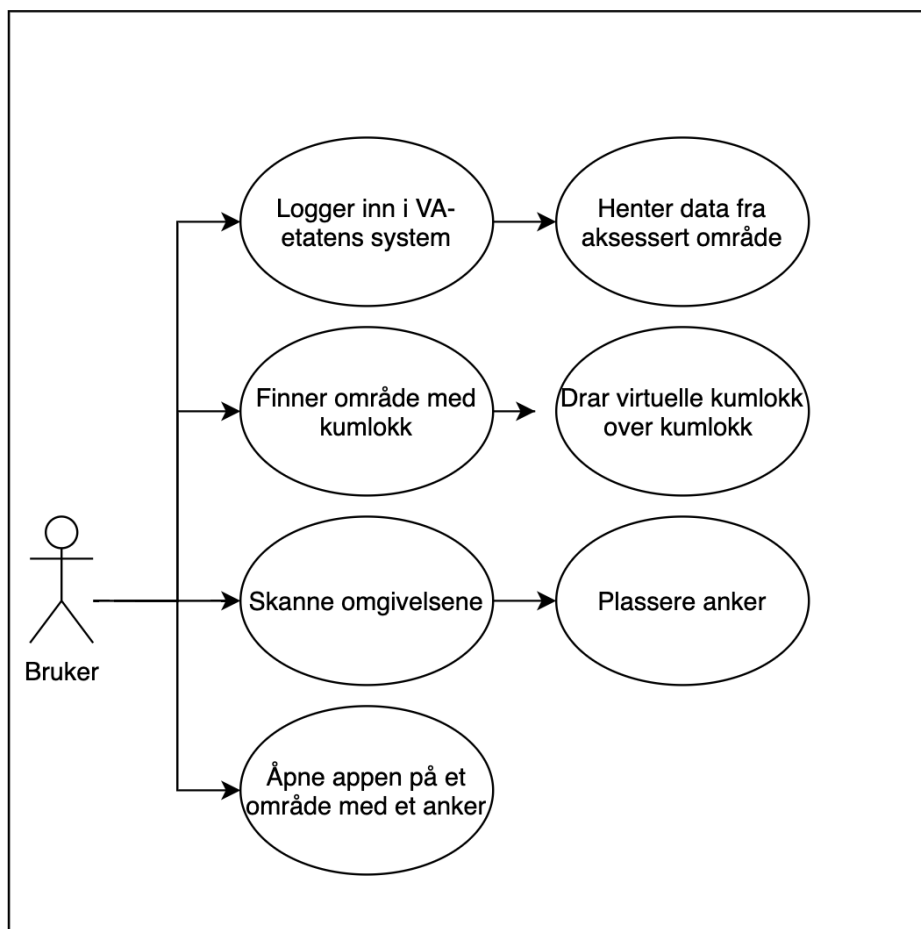
Et annet tenkt brukstilfelle er at brukeren skal kunne se rørene allerede ved start av applikasjonen. Denne brukstilfellen er aktuelt når brukeren starter applikasjonen på et område som har blitt skannet tidligere og et anker har allerede blitt plassert. Da skal rørene dukke opp når ankeret er funnet av applikasjonen.

Laste ned data

Brukeren skal kunne logge seg inn i VA-etatens systemer og få tilgang til et ønsket område for kartdata. Etter det skal kartdataene være tilgjengelig på enheten. Deretter kan brukeren begynne å skanne omgivelsene og plassere et anker for å se rørene.

Manuell kalibrere

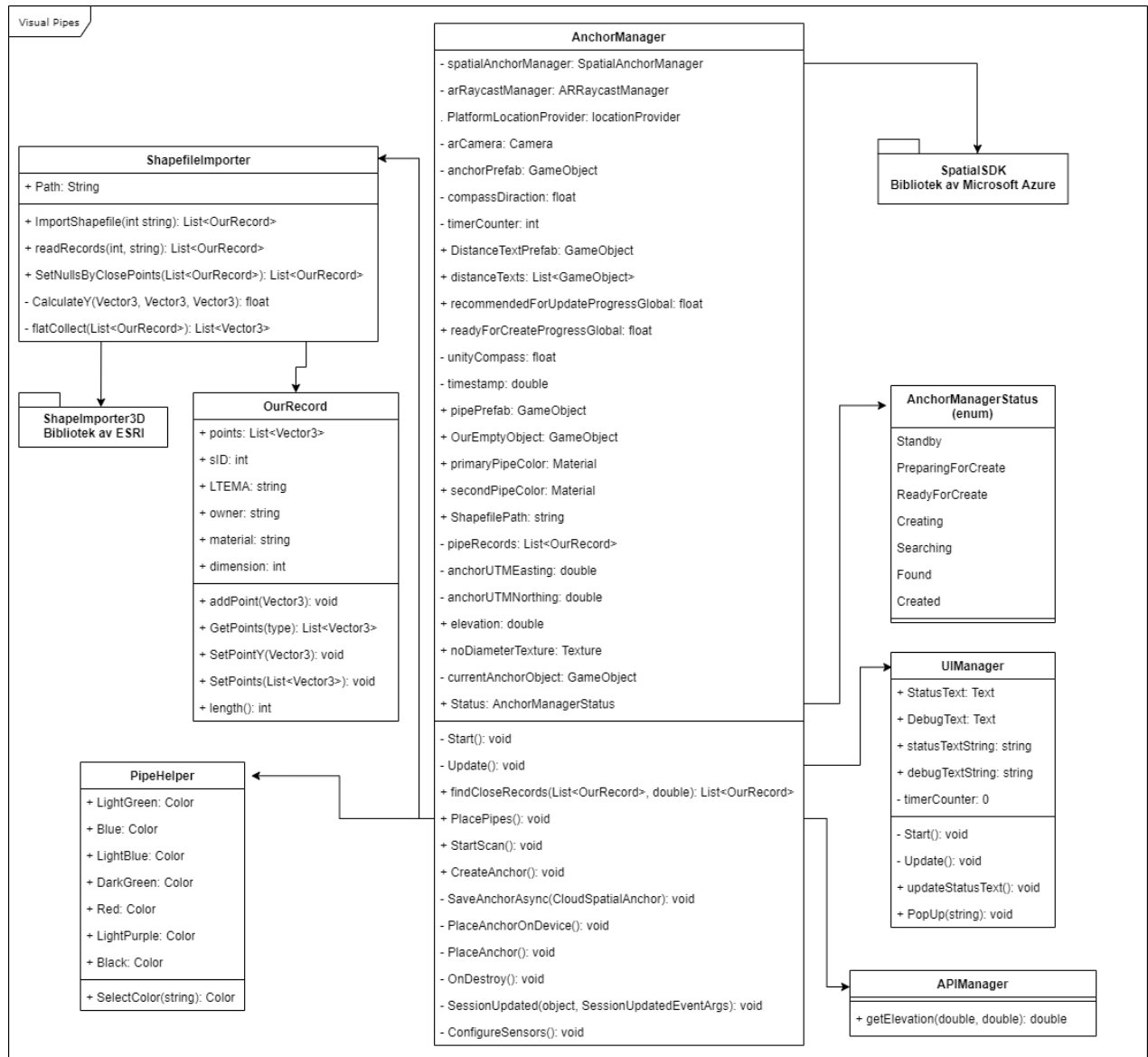
Det siste brukstilfelle for brukeren i applikasjonen er å finne et område med ekte verden kumlokk og dra et virtuelt kumlokk over denne. Rørene vil følge etter og ligger i mer nøyaktighet posisjon.



Figur 4.6 Brukstilfellene

4.4.2 Klassediagram

Klassediagrammet viser klasser som gruppen har laget for applikasjonen. Kontrolleren er *AnchorManager*. Denne klassen håndterer ulike type sesjoner, og bruker Unity sine metoder (*Start* og *Update*) for at applikasjonen skal starte og holde seg eksekvert. For at programmet skulle bli fleksibelt og vedlikeholdbart, ble GRASP brukt under utvikling. GRASP ble også brukt til å holde en oversiktlig ansvarsfordeling gjennom utviklingen og for å gjøre videreutviklingen enklere, som forklart i kapittel 3.6.1 utviklingsmetodikk.

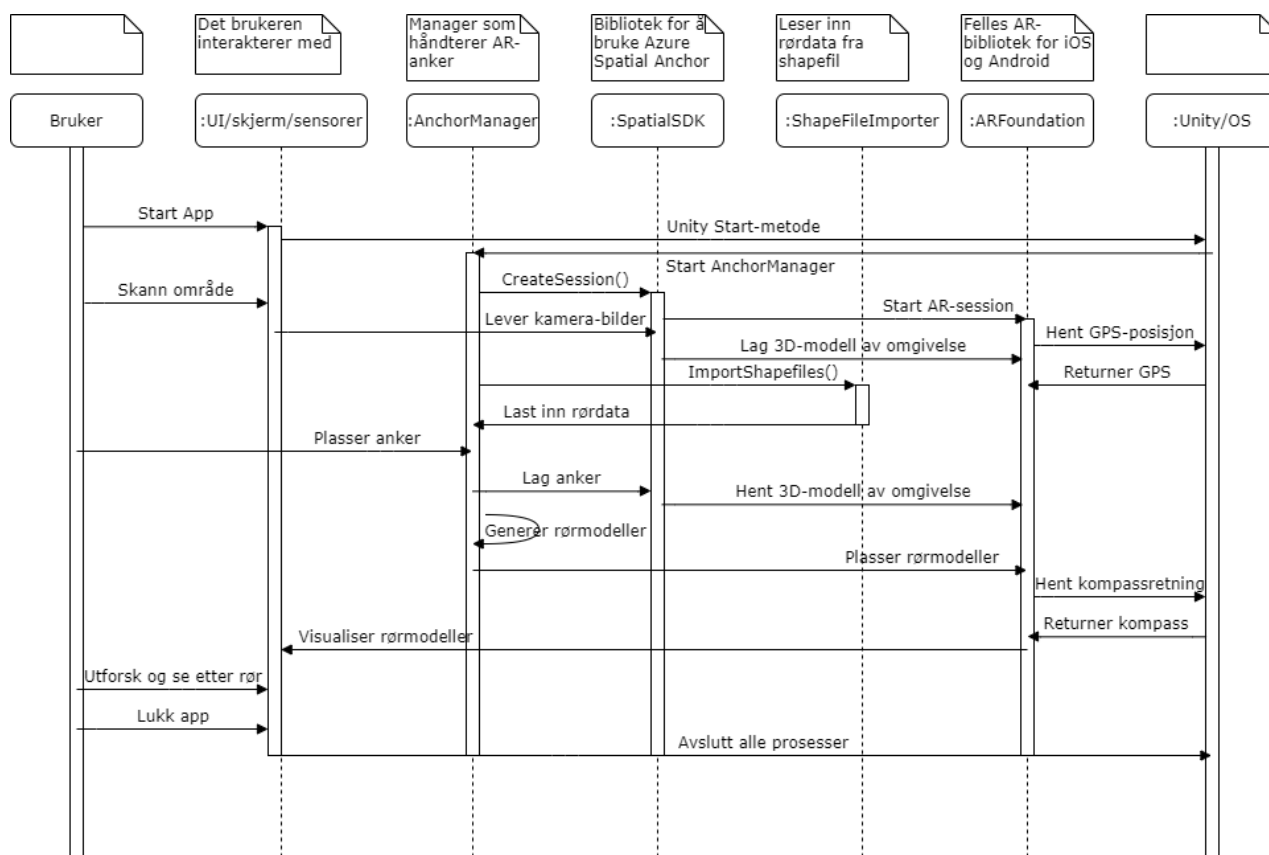


Figur 4.7 Klassediagram av applikasjonen

4.4.3 Sekvensdiagram

Brukeren av applikasjonen er den som starter og lukker applikasjonen. Siden det er en mobilapplikasjon, starter sekvensene med at brukeren starter applikasjonen fra hjemskjermen.

Etter dette vil applikasjon selv starte flere underprosesser som gjør applikasjonen klar for AR. Etter at de rette prosessene er oppe og går, skanner brukeren omgivelsene ved å snu rundt på mobilkameraet. Bilder av omgivelsene blir levert til SpatialSDK og ARFoundation. Med bildene finner SpatialSDK særtrekk ved lokasjonen, sånn at ARFoundation kan lage en 3D-modell av omgivelsen. 3D-modellen gjør at programmet får en romforståelse.



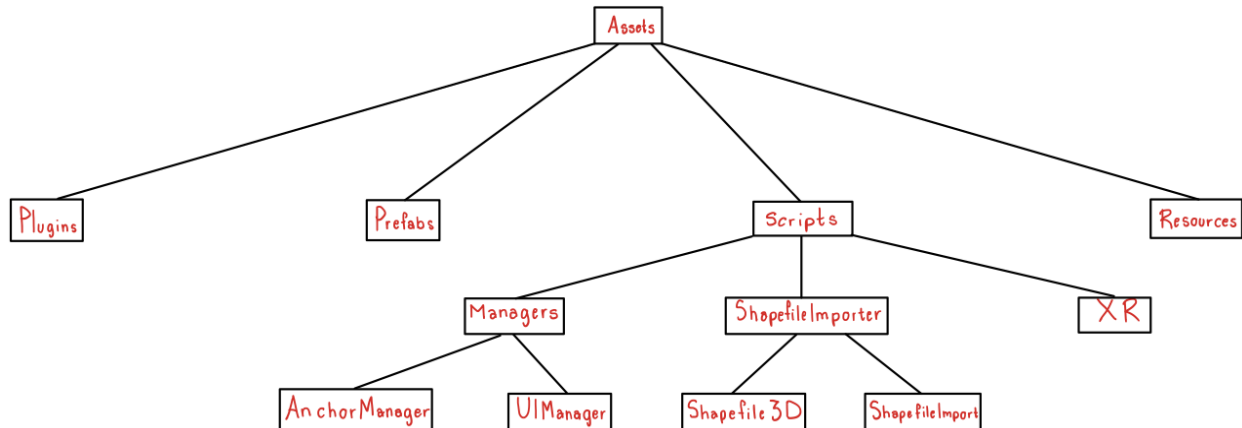
Figur 4.8 Sekvensdiagram av applikasjonen

Brukeren interagerer kun med *UI/skjerm/sensor* (med unntak av ett tilfelle). *UI/skjerm/sensor* fungerer som et lag som viderefremidler informasjon mellom den fysiske enheten og programvaren som ligger under. Kamera, GPS, blåtann og Wifi er eksempel på sensorer som er i denne klassen. Selv om dette egentlig ikke oppfyller kravene til hva en klasse skal være, har gruppen for enkelhetens skyld valgt å definere det som én klasse. Etter at skanningen og modellen av omgivelsen er ferdig, blir rørddataene importert med klassen *ShapeFileImporter*. Dataene blir filtrert basert på GPS-posisjonen til enheten og det blir lagd modeller ut av dataene. Modellene blir til slutt visualisert i riktig posisjon definert av både mobilen sin GPS-posisjon og retningen mobilkompasset peker på.

Sekvensdiagrammet er redusert ned til syv hovedgrener (klasser), men består egentlig av flere underklasser. Grunnen til reduksjon er abstraksjonsnivåer som er lagt inn for å lettere forstå sammenhengen og oppbygningen i applikasjonen. Den grunnleggende implementasjonen av AR og digital grafikk, har blitt lagt under ulike abstraksjonsnivåer i Unity og SpatialSDK, og ville vært overflødig å tatt med i dette sekvensdiagrammet.

4.5 Mappestruktur

Gruppens prosjekt, *SpatialServicesClient*, inneholder mange mapper, undermapper og filer. Dette er vanlig for de fleste Unity prosjekter. Noen mapper har gruppen selv laget for å ha oversikt. Andre mapper ble laget av Unity da prosjektet ble opprettet. Undermappene som er viktige å nevne i dette prosjektet er Plugins, Prefabs, Scripts og Resources som ligger inne i Assets-mappen. Assets-mappen er hovedmappen i prosjektet. Denne mappen inneholder den lokale databasen, metoder til applikasjonen, tilleggsbiblioteker og objekter som prosjektet bruker.



Figur 4.9 Mappestrukturen i prosjektet

Plugins

Plugins-mappen inneholder biblioteker av kode for ulike miljøer, for eksempel Android og iOS. Disse bibliotekene er som oftest skrevet i C, C++ eller Objective-C (Unity 3D, 2019). Plugins gjør det mulig for kildekoden i prosjektet å kalle på funksjoner i de ulike bibliotekene. Bibliotekene som benyttes i prosjektet er Android og CoordinateSharp.

Prefabs

Inne i Prefabs-mappen ligger alle spill-objekter som blir brukt i prosjektet. En prefab kan bli sett på som en forhåndslaget Unity-komponent, som blir brukt til å lage, konfigurere og lagre et komplett GameObject. Et GameObject kan ha egne komponenter, egenskaper og andre GameObject under seg (Unity 3D, 2019). Gruppens prosjekt har mange prefabs, *NordPil.prefab* og *Cylinder.prefab* er to eksempler.

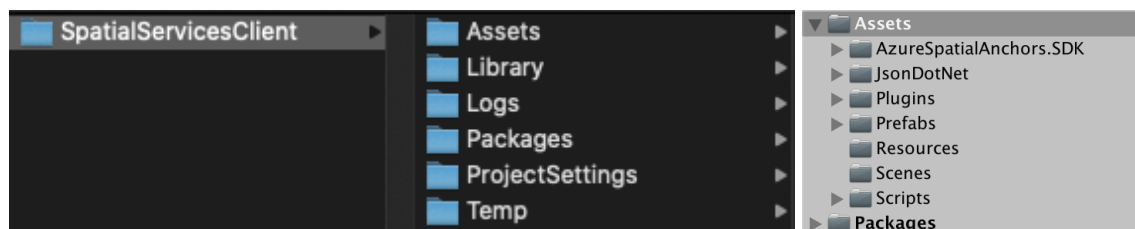
Scripts

I mappen *Scripts* ligger tre undermapper: *Managers*, *ShapefileImporter* og *XR*. I *Managers* ligger klassene *AnchorManager.cs* og *UIManager.cs*. Det er i disse to klassene mesteparten av prosjektets funksjonalitet ligger. Klassen *AnchorManager.cs* inneholder metoder som er relatert til Microsoft Azure Spatial Anchors, og er klassen som setter i gang all funksjonalitet i applikasjonen. Klassen *UIManager.cs* håndterer alt som har med brukergrensesnittet å gjøre.

Undermappen *ShapefileImporter* inneholder klassene *Shapefile3D.cs* og *ShapefileImport.cs*. Disse klassene inneholder kode fra ESRI sitt bibliotek, i tillegg til metoder som gruppen har selv implementert for importering og lesing av shapefiler. *XR* inneholder komponenter fra Microsoft Azure Spatial Anchors som blir brukt i prosjektet, for eksempel *XRCameraPicker*.

Resources

Resources mappen er laget av Unity, og det er her viktige filer skal bli lagt med når applikasjonen bygges. I denne mappen har gruppen lagt til rørddata fra VA-etaten, i form av shapefile og JSON.



Figur 4.10 Mappestrukturen i prosjektet

4.6 Bruk av teknologi

4.6.1 Importering av shapefiler

Ved eksekvering av applikasjonen vil importering av shapefilene bli gjort i *Start*-metoden. Shapefilene er i *Resource*-mappen i applikasjonen og blir importert til Unity av et bibliotek fra ESRI (Csharpdoc, 2019), som gruppen har videreutviklet på. Når ankeret blir utplassert, henter applikasjonen rørene i nærheten av ankeret som forklart i delkapittel 4.6.5.

4.6.2 Spatial Anchor - skanning av området

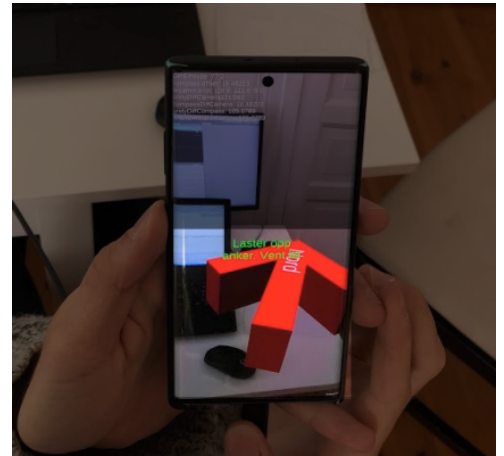
Som nevnt i kapittel 4.2, starter skanningen av området etter brukerens trykk på skjermen hvis statusen i manageren er *Standby*. Da blir metoden *StartScan()*, i *AnchorManager.cs*, kalt på. Denne metoden leter etter Spatial data i omgivelsene for å opprette et anker. Først blir statusen endret til *PreparingForCreate* som endrer teksten på skjermen. Så blir *location provideren* opprettet og startet. Denne sørger for å hente enhetens posisjon. Deretter blir sensor-data som WiFi, GPS og Blåtann konfigurert til å bli tatt i bruk, slik at skanne-prosessen kan starte.

4.6.3 AR Camera

ARFoundation tilbyr flere AR-komponenter som en utvikler kan ha i sin AR-scene i Unity. For å bruke AR-funksjoner i Unity trengs det et AR-kamera. Denne er kompatibel med ARKit og ARCore. Det betyr at raycasting (strålesending) er mulig for brukeren. Raycasting lar brukeren bestemme hvor en stråle skjærer med objektet og sender informasjonen til ARFoundation, hvor et objekt kan bli opprettet.

4.6.4 Compass API - synkronisering av kompass-retning og Unity's z-retning

Under update metoden i Unity, blir Unity's Compass API brukt for å finne retningen enheten peker mot, oppgitt i grader. Denne kompass-retningen blir brukt sammen med kameraets retning for å kunne plassere ankeret, Nord-pilen, i riktig retning. Hvis enheten peker mot bakken, oppdateres kompasset for mer nøyaktig data. Når brukeren trykker på skjermen igjen, ved status lik *ReadyForCreate*, opprettes ankeret med samme retning som kompassretning.



Figur 4.11 Ankeret som en nord-pil

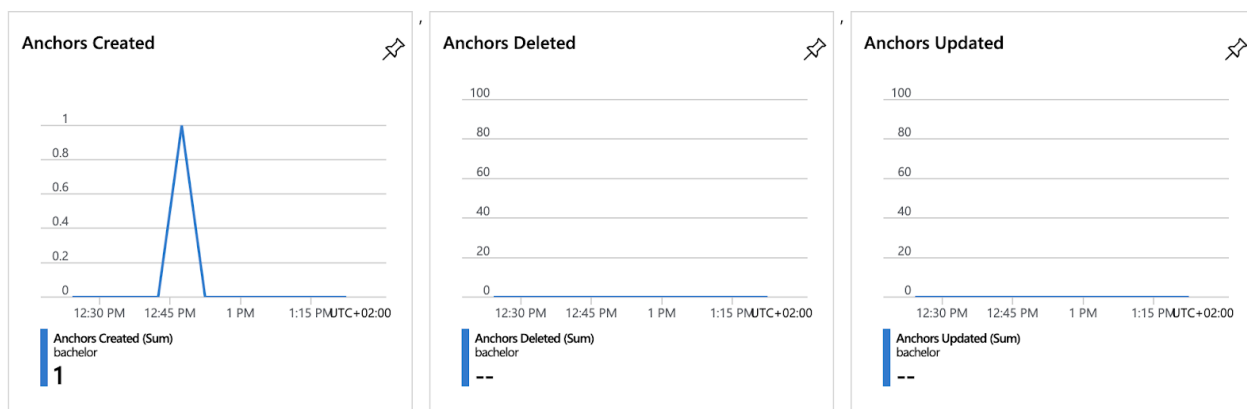
4.6.5 Plassering av anker og rør

Når nord er funnet, og brukeren trykker på skjermen for å plassere anker, vil enhetens posisjon ved hjelp av stedstjeneste bli tilgjengelig i applikasjonen. Koordinatene er oppgitt i DD som blir konvertert til UTM ved hjelp av biblioteket *CoordinateSharp*. Før konverteringen blir Googles Elevation API brukt for å finne den riktige høyden for enheten. For å bruke API-et, blir det sendt inn en spørring med bredde- og lengdegrad for å få riktig høyde tilbake i JSON-objekt format. Kartdataene er i UTM sone 32V. Verden er delt opp i 60 soner med 6 grader som utgjør 360 grader rundt jorden. Fra C til X som er fra sør til nord, med unntak av bokstavene I og O (Dick, Øystein B. , 2020). Ved hjelp av UTM koordinater er det lettere å regne ut avstanden fra ankeret til rørene. Denne avstanden er brukt for å plassere rørene med riktig avstand til ankeret. Ved kun ett anker, er avviket større desto lenger avstand fra ankeret brukeren er. Derfor er radiusen til rør fra ankeret satt til 20 meter for å få mest mulig nøyaktighet.

4.7 Database

Med tillatelse og hjelp fra Bouvet, har gruppen fått en Spatial Anchor-konto i Microsoft Azure sin portal. Dette gir oversikt over gruppens direkte bruk av Spatial Anchors. Her går det an å monitorere alt fra siste timen til siste 30 dagene om ankre har blitt opprettet, slettet eller oppdatert.

Monitoring

Show data for last **1 hour** 6 hours 12 hours 1 day 7 days 30 days


Figur 4.12 Spatial Anchors oversikt i Azure-portalen

4.7.1 Metoder for ankeret i skyen

CreateAnchor

Når nok data eksisterer til å opprette et anker, vil denne metoden lage ankeret og vedvare det. I denne metoden blir objektet lagret opp i skyen med metoden *SaveAnchorAsync* som blir synlig for oversikten i Spatial Anchor kontoen. Gruppen kan sette på tidsinnstilt slik at ankeret forsvinner etter en gitt tid.

SaveAnchorAsync

Når denne metoden blir kalt på, eksekverer manager klassen i Spatial Anchor biblioteket, *CreateAnchorAsync*. Den oppretter et nytt vedvarende anker fra de spesifiserte lokale anker- og streng egenskapene. Til slutt endrer *SaveAnchorAsync*, statusen til manager-klassen til *Created*.

PlaceAnchorOnDevice

Statusen *ReadyForCreate* starter *PlaceAnchorOnDevice* metoden som sjekker om stedstjenesten gir geolocation og om Spatial Anchor sin manager er klar for oppretting. Etter det vil den opprette et anker-objekt, som peker mot nord.

PlacePipes

Så lenge ankeret eksisterer og er opprettet, kan plasseringen av rørene legges ut ved hjelp av ankerets posisjon, som forklart i kapittel 4.6.5.

4.7.2 Livssyklus til et Spatial Anchor

Når et anker blir opprettet, vedvarer deres plassering på tvers av enheter over tid. Så lenge ankeret eksisterer, vil rørene vises frem. Gruppen har foreløpig satt på en tidsbegrensning til et anker for sin eksistens, for å ikke overbelaste databasen under startfasen av applikasjonen. Tidsbegrensningen er foreløpig satt på tre minutter. Når tiden går ut forsvinner ankeret sammen

med rørene. Hvis brukeren åpner applikasjonen på nytt, må brukeren opprette et nytt anker på samme måte som tidligere.

4.8 Sikkerhet og personvern

I utgangspunktet er ikke sikkerhet noe gruppen har tatt stor hensyn til i prosjektet, siden applikasjonen skulle kun være en *proof of concept*. Dersom applikasjonen skal komme i produksjon, må sikkerheten bli et fokusområde. Da må gruppen sørge for at applikasjonen er troverdig og trygg å bruke. Applikasjonen skulle hovedsakelig bli brukt av autoriserte personer, siden den skulle inneholde sensitive data. Dersom personer med dårlige hensikter skulle få tilgang, kunne det føre til at mange mennesker blir rammet på kort tid.

Gruppen har tenkt at et innloggingssystem i applikasjonen ville vært aktuelt for å øke sikkerheten. Med et slikt system ville brukeren kunne laste opp data som skulle bli brukt i applikasjonen, og ha denne dataen tilgjengelig for kun seg selv. Med dette implementert ville arkitekturen sett annerledes ut. Applikasjonen måtte eksempelvis hatt en innloggingsskjerm, med ulike klasser som håndterte dette. Som følger av å ha flere aktiviteter i applikasjonen, hadde det også vært naturlig å gi applikasjonen en hovedmeny for å gjøre den mer oversiktlig.

Det vil alltid være noen som tar sikkerhet på alvor mer enn andre. Disse vil kanskje være mer skeptiske til å gi applikasjonen tillatelse for kamera- og GPS-tilgang. Dette vil være en utfordring, siden applikasjonen bruker kamera til å vise rørene og GPS-signalene for å øke nøyaktigheten til rørposisjonene. Applikasjonen skulle i utgangspunktet kun bli brukt i felt og ikke på privat eiendom. Bachelorgruppen har ingen intensjoner om å samle inn brukerdata, og dette burde nevnes klart og tydelig til brukeren.

5 EVALUERING OG RESULTAT

Dette kapitlet handler om evaluering av applikasjonen. Det blir gjennomgått hvilken metode som ble brukt i evalueringen, og hva VA-etaten synes om applikasjonen gruppen utviklet.

5.1 Evalueringsmetode

I slutten og starten av hver Scrum-sprint, ble det holdt et sprint-møte der gruppen evaluerte siste sprint. Det ble sett på hva som ble gjort i tidligere sprinter og hva som skulle bli gjort senere i prosjektet. Både *product backlog* og Gantt-skjema ble sett på ved hver evaluering og planlegging. Alt dette ble vurdert opp mot kravspesifikasjonene. Under utviklingen ble hele applikasjonen testet flere ganger for å kunne se de minste endringene. Applikasjonen ble testet med en mulighetsstudie sammen med VA-etaten. Dette var fordi en liten endring i koden kunne påvirke store deler av applikasjonen. Det var dermed mer fornuftig å velge funksjonell testing til applikasjonen, framfor å skrive testmetoder.

5.1.1 Semi-strukturert intervju

Ved testingen til prototypen av applikasjonen hadde gruppen definert noen spørsmål på forhånd som skulle bli svart på. Fordelen med dette var tilbakemeldinger som kunne bli brukt til videreutviklingen av prototypen. Ulempen med denne metoden var at svarene gruppen fikk tilbake, var kun i forhold til spørsmålene som var definert på forhånd. Spørsmål som gruppen ikke var oppmerksom på forhånd, gjorde at gruppen gikk glipp av tilbakemeldinger som kunne vært nyttig for videreutviklingen. Listen av spørsmålene var nøye planlagt og ga forholdsvis riktig tilbakemelding til prosjektet. Denne listen er lagt til som vedlegg 9.2 Testing av applikasjonen.

5.1.2 Kravspesifikasjoner

For hver oppgave som ble laget til prosjektet, ble oppgavene målt opp mot kravspesifikasjonen, og de ble evaluert mot kravspesifikasjonen etter de var ferdige. Kravspesifikasjonene var ganske løse før gruppen fikk beskjed at de selv kunne være med på å bestemme kravene. Ønskene fra Bouvet og VA-etaten ble tatt i betraktning, men kravspesifikasjonene ble bestemt ut i fra prosjektets tidsbegrensning. Evalueringen av de nye kravene tok utgangspunkt i dette.

5.1.3 Funksjonell testing

Funksjonelle tester utføres for å evaluere samsvars av et system eller en komponent med spesifiserte funksjonelle krav. Funksjonell testing beskriver vanligvis hva systemet gjør. Det som er viktig med funksjonell testing er at en deler opp applikasjonen i riktige biter for å kunne se om riktige komponenter jobber riktig med hverandre. Funksjonell testing skal gi inntrykk på en forventet kvalitet på applikasjonen. Det som også er viktig med den funksjonelle testen, er at det skal vise den faktiske bruken av applikasjonen.

I dette prosjektet ble store deler av testingen gjort ved hjelp av en debug tekst, som ble printet til konsollen. Debug teksten ga informasjon om kamera, GPS, koordinater og posisjon til smartmobilen. Det ble også brukt debug log i Unity for hver eksekvering, for å printe ut viktige verdier og informasjon som ikke er synlig for brukeren, men viktig for at systemet skal fungere på riktig måte. Til slutt måtte gruppen sjekke om applikasjonen fungerte etter VA-etatens behov.

5.1.4 Testing med VA-etaten

Før testingen med VA-etaten, ble det på samme dag kontrollert at applikasjonen fungerte på iOS, Android og iPadOS. Resultatet på iOS og Android var nærmest helt likt, bortsett fra at brukergrensesnittet var bedre tilpasset til Android. Ellers var røre informasjonen, nord-pilen og rørene helt like. iPaden gruppen testet applikasjonen på ga unøyaktig posisjon på rørene. Dette var på grunn av selve lokasjonen på iPaden. iPaden målte sin lokasjon til å være omtrent 30 meter unna der den egentlig var fysisk. Årsaken til dette var at iPad-er uten mobildata baserte sin lokasjon på WiFi-signaler, og hadde ikke innebygget lokasjonstjeneste.

Agendaen vedlagt i 9.2 ble gjennomgått under utførelsen av testingen på Kronstad, sammen med VA-etaten den 7. mai. Både Ali Mekki, spesialkonsulent, og Tommas Kraft, fagansvarlig for kartinfo, fikk prøve applikasjonen uten instruksjoner. Dette var for å kunne teste brukervennligheten til applikasjonen. Kort tid etter testingen fikk de vite hva som var fremgangsmåten til utbyggingen av applikasjonen og hvordan teknologiene bak fungerte. Gruppen informerte også VA-etaten om unøyaktigheten applikasjonen hadde, men dette var forventet fra deres side. Senere fikk gruppen innspill fra VA-etaten om brukergrensesnittet. Gruppen spurte om hva forventningene til applikasjonen var, i forhold til hva som ble oppnådd. Til slutt ble det diskutert tilleggsarbeid til applikasjonen.



Figur 5.1 Ali Mekki og Simen Romstad tester applikasjonen.

5.1.5 Etter-evaluering

Testing og evalueringen med VA-etaten var viktig for prosjektet. Dette ga nyttig informasjon på om det var potensial for at applikasjonen var mer brukervennlig enn deres nåværende løsning. Tilbakemeldingene fra VA-etaten og Bouvet om gruppens arbeid påvirket videre arbeid for applikasjonen. Videre arbeid blir nøyere gjennomgått i kapittel 8. Evalueringen med VA-etaten ble gjort for å forsikre om at utseende, brukeropplevelsen og brukergrensesnittet var god nok. VA-etaten fikk sett på den nye teknologien Spatial Anchors, som gjorde det mulig å utvikle til flere enheter og operativsystemer med samme kodebase.

GitHub

Det var viktig å teste applikasjonen hver gang en ny funksjon ble implementert. Dette var fordi en liten endring kunne forårsake store ødeleggelser. Disse ødeleggelsene kunne i verste fall føre til at applikasjonen sluttet å fungere. Gruppen måtte dermed sørge for at det alltid fantes en fungerende versjon av applikasjonen. For dette formålet ble oppbevaringsstedet GitHub brukt.

Arbeid

Gruppen valgte å jobbe faste tidspunkter og faste dager gjennom hele bachelorprosjektet, for å få en mest mulig jevn flyt på arbeidet. Mye av tiden gikk til å finne ut hvordan gruppen skulle løse problemet. Ved etter-evaluering så gruppen at det lønnet seg med mye arbeid i prosjektperioden. Det førte til at gruppen fikk god tid til både rapportskrivning, retting og promo-video til prosjektet i mai.

Kommunikasjon

For å sikre høy kvalitet for kunden var det viktig med hyppig kommunikasjon mellom arbeidsgiver, kunde og veileder. Dette sikret også for mest mulig fremgang i prosjektet. Tett kommunikasjon med alle involverte reduserte misforståelser og forsinkelser i prosjektet.

5.2 Evalueringsresultat

Det ble avklart tidlig at prosjektet hadde store muligheter, derfor var det viktig å avgrense kravspesifikasjonen. Hovedkravet var at applikasjonen skulle fremstille rør i AR gjennom kameraet. Ytterligere oppgaver kunne gruppen selv komme med. Fra VA-etaten var det ingen spesifikke ønsker, men fra Bouvet kom det et par tilleggsoppgaver som VA-sensorer og vise mer informasjon enn bare rørene. Dette falt bort fordi VA-etaten ikke ville prioritere dette, men heller vise rørene der de var. Det var klart at oppdragsgiver forventet minimumskravene for oppgaven - å fremstille rørene. All ekstra tid kunne gå til tilleggsfunksjoner. Oppdragsgiver hadde et par forslag til dette, samtidig som gruppen hadde egne ideer. Noen få av disse ideene ble gjennomført, andre ble lagt i product backlogen og i videre arbeid.

- Applikasjonen var som forventet med tanke på GPS-nøyaktighet. VA-etaten hadde GPS-instrument for å øke nøyaktigheten til centimeter nivå. De likte ideen om manuell kalibrering med kumlokk. Dette blir forklart i kapittel 7.2.1.
- Sikkerhet var eneste tilleggsfunksjon VA-etaten ville legge til i applikasjonen, som nevnt i kapittel 4.8.
- Brukeren kunne trenge en begrunnelse på hvorfor brukeren skulle skanne omgivelsene for enheten. Ellers var dette noe VA-etaten kunne tenkt seg å prøve.

Gjennomføringen av testingen med VA-etaten ga oss gode tilbakemeldinger på applikasjonen. Det viktigste som ble notert i testingen, var sikkerheten rundt hvem som skulle få tilgang til all rørdatabasen. Andre bemerkelsesverdige punkter var brukergrensesnittet og manuell kalibrering. For å få tilgang på VA-etaten sin database om rørene, vil applikasjonen trenge en trygg innlogging, som nevnt i kapittel 4.8 Sikkerhet. En bruker skulle heller ikke kunne få tilgang på all data, men kun der arbeidet foregikk. I brukergrensesnittet manglet det litt informasjon om hvorfor brukeren måtte skanne omgivelsene. VA-etaten ble forklart at dette var for å gi Spatial Anchors nok data til å vite hvor den kunne plassere ankeret. VA-etaten likte til slutt ideen om manuell kalibrering, der

brukeren kan se et virtuelt kumlokk og dra det bort til det ekte kumlokket for å øke nøyaktigheten.

5.3 Prosjekresultater

VA-etaten hadde ingen spesifikke ønsker om applikasjonen, men ønsket å se hvordan en 3D-løsning kunne blitt i forhold til deres nåværende løsning i 2D kart.

VA-etaten mente at arbeidet som var gjort for å tolke rørdatabasene var ekstra imponerende. Selv om AR-delen av oppgaven også var bemerkelsesverdig, la de vekt på at en stor del av oppgaven var hvordan gruppen skulle tolke ukjent data. Dette var en god erfaring for gruppen, som måtte tilpasse arbeidet etter kundens ressurser. Da gruppen presenterte resultatet til VA-etaten, la de merke til at høyden til rørene og bakken ikke stemte helt overens. Det var ingen store reaksjoner på at posisjonene ikke var så nøyaktig, for det var en enkelt løsning med å bruke presisjonsinstrumenter av GPS-posisjon til rørene. Høyden er noe de har hatt problemer med på selv de dyreste instrumentene. Det at gruppen ikke fikk vist helt riktig høyde var noe de forventet i forkant.

Etter testingen som var utført sammen med VA-etaten på Kronstad arbeidet ikke gruppen videre med applikasjonen. Dette var fordi gruppen måtte fokusere på rapporten like mye som selve prosjektet. Gruppen var fornøyd med evalueringsresultatene da applikasjonen i sin helhet møtte forventningene til VA-etaten. Tilbakemeldingen gruppen fikk fra VA-etaten vil være inspirasjon til hva videre arbeid kan være.

Gruppen oppnådde minstekravene til prosjektet og fikk til å visualisere rørene ved hjelp av AR-teknologi, basert på lokasjon til bruker og fremstille rørene der brukeren er. Kravene var at applikasjonen skulle bruke shapefiler direkte ved bruk av applikasjonen, noe som fungerte. I tillegg var funksjoner som avstand til rørene, informasjon om feil ved diameter på rør, farge for type rør og diameter på rørene lagt til. Etter kommunikasjon med VA-etaten og Bouvet var dette en fin start på prosjektet og de var fornøyd med resultatet. Begge parter kunne se for seg at prosjektet ble videreutviklet og etterhvert kunne bli tatt i bruk av VA-etaten. Det var derfor blitt et *proof of concept* for arbeidere eller andre som kan få tilgang på rør-databasene. Applikasjonen fungerte på Android, iOS og iPadOS slik som kravspesifikasjonen nevnte.

Evalueringsresultater tilsier at applikasjonen er noe som VA-etaten og Bouvet vil satse på i fremtiden. Siden applikasjonen ikke er et ferdig produkt, må den bli mer utviklet før den skal tas i bruk. Eksempelvis må nøyaktigheten bli bedre. Om prosjektet skal fortsettes på er mye avhengig av om Bergen kommune mener dette er økonomisk lurt å satse på.

Mulighetsstudien handlet om å finne ut om produktet kunne skape verdier for VA-etaten eller lignende kunder. VA-etaten nevnte at produktet gruppen har utviklet kan skape verdier hos Bergen kommune og andre kommuner som benytter samme løsning som de i dag.

6 DISKUSJON

Gjennom prosjektet har gruppen tatt flere viktige valg. Dette kapitlet handler om hvordan de ulike valgene som ble tatt har påvirket prosjektets resultat. Det vil også bli diskutert hva som ville bli gjort annerledes, dersom prosjektet skulle blitt gjennomført på nytt.

6.1 Konsekvenser

Målet i prosjektet var å utvikle en applikasjon som fremstilte vann- og avløpsrør i utvidet virkelighet. For å oppnå dette målet gikk mye av tiden på å undersøke ulike teknologier og verktøy som skulle bli brukt i prosjektet. Gruppen visste ikke ved prosjektstart at Microsoft Spatial Anchor var den rette teknologien til å bruke i prosjektet. Spatial Anchors ble anbefalt av gruppens eksterne veileder i Bouvet da prosjektet allerede var godt i gang. Det hadde gått mindre tid til undersøkelse av andre teknologier om gruppen hadde visst at Spatial Anchors var den rette teknologien i prosjektet. Da hadde gruppen fått mer tid til utvikling av applikasjonen.

Selv om gruppen sent kom i gang med Spatial Anchors sparte gruppen mye tid på å bruke denne teknologien. Dette var fordi Spatial Anchors sammen med Unity og ARFoundation støttet utvikling av kryssplattform applikasjon. Dersom andre teknologier ble valgt for prosjektet hadde kanskje applikasjonen fungert kun på den ene plattformen. Da måtte gruppen skrive to separerte koder for plattformene iOS og Android. Den valgte teknologien sparte gruppen for mye tid da den samme kildekoden kunne bygges til to ulike miljøer.

Uten .NET-biblioteket, *CoordinateSharp*, måtte gruppen ha implementert egne metoder for å konvertere fra lengdegrader og breddegrader til UTM-koordinater. Eller så måtte gruppen ta i bruk andre eksisterende konverterings metoder som ga upresise tall, i forhold til *CoordinateSharp* sine metoder.

Det ble bestemt at versjonskontrollsystemet, Git, skulle bli brukt fra starten. Som nevnt i kapittel 3.4.1 hadde gruppen litt erfaring med Git fra før. Git har hjulpet gruppen til å samarbeide med koden. Å oppbevare koden på GitHub gjorde det mulig å gå tilbake til sist fungerende versjon av koden, dersom det tok for lang tid å debugge koden.

Den smidige arbeidsmetodikken, Scrum, har fungert godt til prosjektet og ga struktur i arbeidet. I tillegg hadde arbeidsmetodikken Scrum hjulpet gruppen til å samarbeide på prosjektet på en rask og effektiv måte. Metodikken gjorde det mulig å dele opp et stort prosjekt som dette i flere oppgaver på flere sprints. Den daglige kommunikasjonen har hjulpet gruppen til å arbeide raskere og få raskere oppklart usikkerheter rundt prosjektet.

6.2 Refleksjon

Spatial Anchors kan bli brukt til å lage virtuelle omgivelser og gjenkjenne dem. For dette prosjektet betyr det at brukeren må legge inn rørledning i systemet kun én gang. Neste person som skal se på det samme røret, i de samme omgivelsene, slipper å skanne området på nytt. Dette er fordi alle som har applikasjonen vil kunne gjenkjenne omgivelsene og plassere ut rørene umiddelbart.

Denne funksjonaliteten til Spatial Anchor var helt avhengig av en database som kunne ta vare på disse ankrene. I tillegg måtte det finnes et system for å finne tilbake til riktig anker i databasen, basert på omgivelsene brukeren stod i. På grunn av prosjektets tidsbegrensning har gruppen valgt å sette denne funksjonaliteten som videre arbeid. Denne funksjonaliteten står øverst på listen av videre arbeid, siden den hadde gjort applikasjonen mye mer brukervennlig.

Siden prosjektet ikke lagret ankrene i en database var bruken av Spatial Anchors i prosjektet overflødig. Grunnen til at Spatial Anchors ble valgt til prosjektet var for at ulike enheter skal kunne gjenkjenne samme grafiske objekter i AR samtidig. Siden prosjektet ikke tok i bruk denne funksjonaliteten av Spatial Anchors, kunne gruppen ha gjennomført prosjektet annerledes ved en omstart av prosjektet. Gruppen kunne vurdert å kun bruke ARFoundation og ikke Spatial Anchors. Det hadde vært enklere å utvikle applikasjonen, men mulighetene for videre arbeid ville vært begrenset.

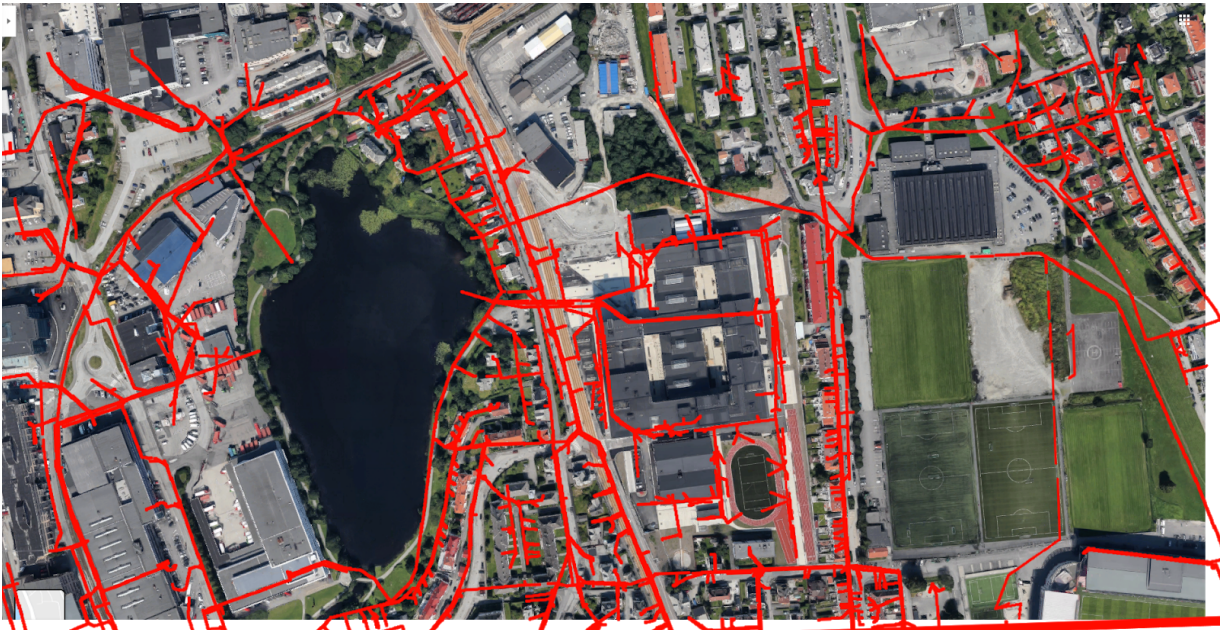
Det er uten tvil at hvis gruppen hadde visst at Spatial Anchors var den rette teknologien å bruke i prosjektet tidligere, hadde resultatet vært mye bedre. Gruppen hadde fått tid til å implementere flere viktige funksjonaliteter i applikasjonen. Rørene hadde kanskje også fått mer presise posisjoner. Mindre tid hadde blitt brukt til å undersøke potensielle teknologier, og mer tid hadde blitt brukt på utvikling av applikasjonen. Gruppen har lært mye av prosessen om å tolke ukjent data og å finne egnet teknologi som kan hjelpe til med å løse problemstillingen.

7 KONKLUSJON OG VIDERE ARBEID

7.1 Konklusjon

Hovedkravet til dette prosjektet var å utvikle en AR-applikasjon for å visualisere vann- og avløpsrør for Bergen kommune. I følge resultatene fra evalueringen ble minimumskravene fullført. Applikasjonen bruker enhetens kamera for å hente inn nok informasjon til å legge ut et anker som igjen la ut rørene basert på posisjon på Kronstad-området. Applikasjonen vil kunne bli brukt av utvalgte fra VA-etaten. Dette er fordi kartdataene som applikasjonen bruker er unntatt offentligheten. Applikasjonen bruker kun kartdata fra et utvalgt område rundt Kronstad, som vises på figur 7.1. Det er stor sannsynlighet for at VA-etaten kommer til å ta applikasjonen i bruk, siden de omtalte applikasjonen som «når den kommer i produksjon». Det var tenkt at brukeren

skulle lett åpne applikasjonen og få opp rørene dersom et anker tidligere var laget på et område. Etterhvert kunne applikasjonen potensielt inneholde alle rørene i hele Bergen.



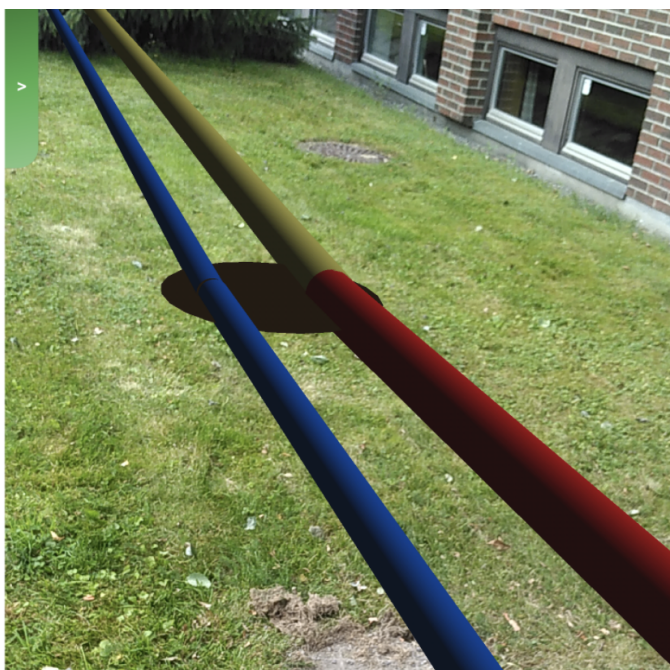
Figur 7.1 Bachelorgruppens sammensetning av Google maps og rørdatabasen som ble tilsendt (Avvik kan forekomme).

7.2 Videre arbeid

Applikasjonen er ikke helt klart til bruk og mangler noen funksjonaliteter. Den krever tilgang til kartdata som gruppen ikke har. Gruppen måtte derfor ha en avtale med VA-etaten for ytterligere forbedringer og arbeid med applikasjonen, gjennom Bouvet. Dataen til rørene inneholdt ikke bare koordinater, størrelse og type, men også sensorer som kunne gi informasjon om trykk, hastighet, lekkasje og mer. Bouvet kom derfor med et nytt forslag som handlet om å utnytte tilleggs-dataene til å utvide produktet. Produktet skulle ikke kun vise rør, men også bruke disse sensorene til å forutse rør-sprekker og å få informasjon om hvor dette kom fra. Dette bachelorprosjektet hadde som mål kun å lage en løsning for å vise rørene i utvidet virkelighet. Å vise ekstra sensordata ble noe gruppen kunne jobbe med senere, dersom prosjektet hadde mer tid.

7.2.1 Manuell kalibrering

Bruken av kumlokk i applikasjonen kan være svært nyttig for å ha en referanse til hvor rørene faktisk er. Nåværende løsning tilbyr kun rør. Det går an å se på et ekte kumlokk og se omtrent hvor rørene ligger. Det er enda bedre å kunne dra de virtuelle kumlokkene til de ekte kumlokkene for å øke nøyaktigheten til rørene. Dette er den viktigste funksjonaliteten for å gjøre det mulig at VA-etaten skal ta i bruk applikasjonen. I figur 7.2 under, har Norkart gjort dette.



Figur 7.2 WaterGo, sommerprosjektet til Norkart 2017 (Norkart, 2017).

7.2.2 Hente shapefilene direkte inn fra VA-etatens database

For at det skal bli enkelt for VA-etaten å bruke applikasjonen vil de kunne hente data direkte fra deres kartdatabase. Denne dataen skal som dagens løsning bli behandlet på samme måte som nevnt i kapittel 4. Kartdataene til VA-etaten blir stadig oppdatert og blir lagt til nye data. For at applikasjonen ikke må bygges på nytt hver gang det skal bli lagt til data, har gruppen en løsning. Det er å hente data fra deres database automatisk. Tilgang til databasen må derfor bli godkjent.

7.2.3 Skape og gjenkjenne ankre

For at brukerne skal slippe å skanne hver gang applikasjonen brukes på et område, må ankrene hentes fra Spatial Anchors databasen og fremstilt ved start av applikasjonen. Dette er ikke tidkrevende arbeid, og kan være viktig for et brukervennlig grensesnitt for brukere av applikasjonen.

7.2.4 Informasjon om rørene

Rørene ble visualisert gjennom kameraet som var definert i kravspesifikasjonen. Noen forslag til framtidig arbeid var følgende: en grei funksjon er å legge til mer informasjon om rørene. Et lite vindu øverst til venstre med informasjon om hvor store og lange rørene er, hvilken type røret er og etterhvert kunne legge til eventuelle lekkasjer og trykk på rørene og hvor det kommer fra.

7.2.5 VA-sensorer

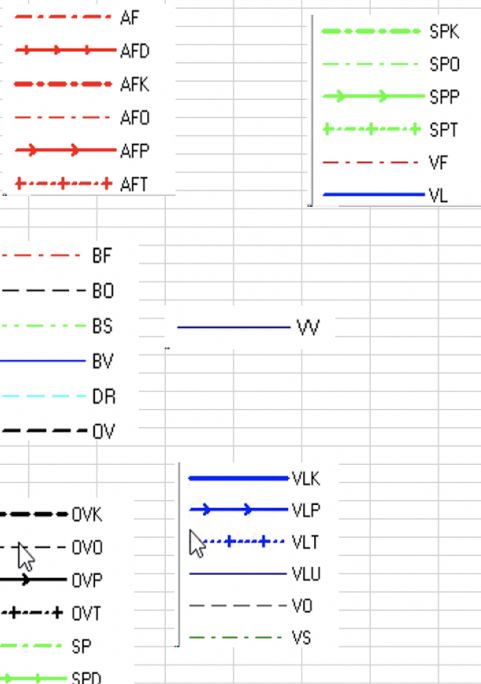
I tidligere samtaler med både VA-etaten og Bouvet fikk gruppen høre om VA-sensorer som er utplassert over vann- og avløpsrørene. Disse sensorene kan måle trykk på rørene, om eventuelle lekkasjer og hvor lekkasjene er. For en framtidig versjon av applikasjonen kan det bli planlagt å

legge til VA-sensor for ekstra informasjon. Dette kan bidra til at applikasjonen blir en plattform VA-etaten har et stort behov for.

7.2.6 Rørdesign basert på rørtype

VA-etaten har en oversikt over hvordan de forskjellige rørene skal se ut i 2D løsning. En videre utvikling av denne applikasjonen vil være å se på oversikten i figur 7.3 under og designe 3D objekter med flere detaljer for å bruke dette i applikasjonen.

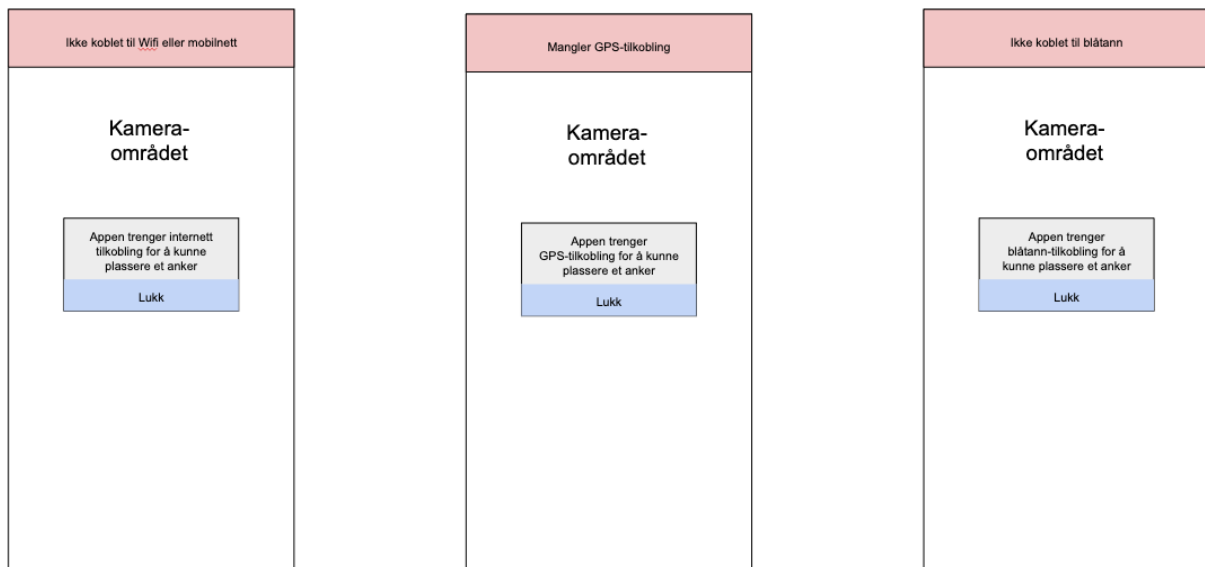
Tema	Forklaring	Antall treff i databasen	Kommentar
AF	Avløp felles ledning	17604	
AFD	Avløp felles dykkerledning	85	
AFK	Avløp felles kanal	199	
AFO	Avløp felles overløp	540	
AFP	Avløp felles pumpeledning	186	
AFT	Avløp felles tunnel	339	
AFX	Avløp felles kopling	2	
BF	Borrehull, avløp felles	13	
BO	Borrehull, overvann	43	
BS	Borrehull, spillvann	592	
BV	Borrehull, vann	507	
DR	Drensledning	1355	
IZ	Sigevannsledning	3	
OV	Overvann	43121	
OVF	Overvann fordrøyning	3	
OVK	Overvann kanal	475	
OVO	Overvann overløp	75	
OVP	Overvann pumpeledning	54	
OVR	Overvann renne	88	
OVT	Overvann tunnel	269	
SP	Spillvannsledning	90183	
SPD	Spillvann dykkerledning	87	
SPK	Spillvann kanal	37	
SPO	Spillvann overløp	327	
SPP	Spillvann pumpeledning	1604	
SPS	Spillvann sugeledning	2	
SPT	Spillvann tunnel	211	
SPX	Spillvann kopling	1	
VF	Varerør avløp felles	8	
VL	Vannledning	76356	
VLK	Vann kanal	43	
VLT	Vann tunnel	494	
VLU	Vann utspyler/spyleledning	103	
VO	Varerør overvann	38	
VS	Varerør spillvann	55	
VV	Varerør vann	79	
XF	Fjernvarme	2	
XG	Gassledning	5	



Figur 7.3 Oversikt over alle type rør

7.2.7 Kobling av mobilenheten til internett, GPS og blåtann

Internett-, GPS- og blåtann-tilkobling er nødvendige tilkoblinger for applikasjonens funksjonalitet. Det er viktig at smartmobilen har disse tilkoblingene mens applikasjonen er i bruk. Gruppen tenker at det kan vært nyttig med en funksjonalitet som gir brukeren tilbakemeldinger, dersom en av disse tilkoblingene mangler. Brukeren skal få en av feilmeldingene som vises i figur 7.4, dersom mobilen mangler de gitte tilkoblingene. Feilmeldingen skal bli vist i et rødt område øverst på skjermen, og forsvinner når brukeren har gitt tilgang til det som var nødvendig. Det skal også dukke opp et pop-up vindu midt på skjermen, som viser brukeren hvilken tilkoblinger applikasjonen mangler. I tillegg skal samme informasjon bli vist i en popup midt på skjermen.



Figur 7.4 Feilhåndtering

8 LITTERATURLISTE

Adobe (2020), *Utvidet virkelighet. Nå en realitet*. Tilgjengelig fra:

<https://www.adobe.com/no/products/aero.html> (Hentet: 18. mars 2020).

Apple (2020), *ARKit*. Tilgjengelig fra: <https://developer.apple.com/documentation/arkit/> (Hentet: 26. mars 2020).

Bergen Kommune (2019), *Om oss*. Tilgjengelig fra:

<https://www.bergen.kommune.no/omkommunen/avdelinger/vannog-avlopsetaten/9080/article-9285> (Hentet: 17. mars 2020).

Chacon, S. og Straub B. (2014) *Pro Git*. Apress.

Cheng, J. & C., Weiwei, K. & C., (2017). *Comparison of marker-based AR and markerless AR: A case study on indoor decoration system*. 10.24928/JC3-2017/0231.

CoordinateSharp (2020), *Developer Guide*. Tilgjengelig fra:

<https://coordinatesharp.com/DeveloperGuide> (Hentet: 30. april 2020).

Crystalinks (2014), *Augmented Reality*. Tilgjengelig fra: <https://www.crystalinks.com/ar.html> (Hentet: 26. mai 2020).

Csharpdoc (2019), *C# Class TestShapeFile.ShapeFile*. Tilgjengelig fra:

<https://csharpdoc.hotexamples.com/class/TestShapeFile/ShapeFile> (Hentet: 1. februar 2020).

Deloitte (u.å.) *5 ganger du har brukt Augmented Reality uten å vite det*. Tilgjengelig fra: <https://www2.deloitte.com/no/no/pages/technology-media-and-telecommunications/articles/augmented-reality-bruk.html> (Hentet: 26. mai 2020).

DGT Associates (2020), *Augmented Reality and 3D imaging: An exciting new dimension*. Tilgjengelig fra: <https://dgtassociates.com/augmented-reality-and-3d-imaging-an-exciting-new-dimension/> (Hentet: 5. mai 2020).

Dick, Øystein B. (2020), *UTM*. Tilgjengelig fra: <https://snl.no/UTM> (Hentet: 30. mai 2020).

ESRI (1998), *ESRI Shapefile Technical Description*. White Paper. Sted: New York, USA. Tilgjengelig fra: <https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf> (Hentet: 23. april 2020).

Google developer (2020), *ARCore overview*. Tilgjengelig fra: <https://developers.google.com/ar/discover> (Hentet: 26. mars 2020).

Google developer (2020), *Google Disk*. Tilgjengelig fra: <https://developers.google.com/maps/documentation/elevation/intro?hl=nb> (Hentet: 19. mars 2020).

Google developer (2020), *Elevation API*. Tilgjengelig fra: <https://developers.google.com/maps/documentation/elevation/intro?hl=nb> (Hentet: 30. april 2020).

Griffin, J. (2015), *Locate Infrastructure Faster and Prevent Errors*. Tilgjengelig fra: <https://www.ecmag.com/section/your-business/underground-report-augmented-reality> (Hentet 5. april 2020).

Kahneman, D., Tversky, A. (1977) *INTUITIVE PREDICTION: BIASES AND CORRECTIVE PROCEDURES*. Advanced Decision Technology.

Kommuner i hundre over Gemini Portal (2016). [VAnytt]. Tilgjengelig fra: <https://www.vanytt.no/2016/01/29/kommuner-i-hundre-over-gemini-portal/> (Hentet: 5. mai 2020).

Microsoft Azure (2019), *Spatial Anchors*. Tilgjengelig fra: <https://azure.microsoft.com/nb-no/services/spatial-anchors/> (Hentet: 26. mars 2020).

Microsoft (2020), *Hololens 2 tech specs*. Tilgjengelig fra: <https://www.microsoft.com/en-us/hololens/hardware> (Hentet: 18. mars 2020).

Microsoft (2017), *Introduction - C# language specification* / *Microsoft Docs*. Tilgjengelig fra:
<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/introduction> (Hentet: 19. mars 2020).

Microsoft (2020), *Visual Studio 2019 for Mac*. Tilgjengelig fra:
<https://visualstudio.microsoft.com/vs/mac/> (Hentet: 19. mars 2020).

Newtonsoft (u.å.), *JSON.NET Documentation*. Tilgjengelig fra:
<https://www.newtonsoft.com/json/help/html/Introduction.htm> (Hentet: 23. april 2020).

Norkart (2017), *Sommer 2017: Prosjekt WaterGo*. Tilgjengelig fra:
<https://www.norkart.no/2017/08/04/sommer-2017-prosjekt-watergo/> (Hentet: 20. januar 2020).

OMG (2017), *Unified Modeling Language, v2.5.1*. Tilgjengelig fra:
<https://www.omg.org/spec/UML/2.5.1/PDF> (Hentet: 25. mai 2020).

Powel (2020), *Gemini Portal*. Tilgjengelig fra:
<https://www.powel.no/vann-og-miljo/drift-og-vedlikehold/gemini-portal> (Hentet: 24. mars 2020).

Regjeringen (2020), *Tiltakene mot koronavirus videreføres*. Tilgjengelig fra:
<https://www.regjeringen.no/no/aktuelt/tiltakene-mot-koronavirus-viderefores/id2694682/>
(Hentet: 24. mars 2020).

Schwaber, K. og Sutherland, J. (2017), *The Scrum Guide*. Tilgjengelig fra:
<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>
(Hentet: 19. mars 2020).

Slack (2020), *Slack is where work happens*. Tilgjengelig fra:
https://slack.com/intl/en-no/lp/three?utm_medium=ppc&utm_source=google&utm_campaign=d_ppc_google_northern-europe_en_brand-hv-exp-messaging&utm_term=slack&ds_rl=1249094&gclid=EAIaIQobChMIyfPI_6Wm6AIVA6qaCh2kTAvZEAAYASAAEgIlo_D_BwE&gclsrc=aw.ds
(Hentet: 19. mars 2020).

Teknisk Ukeblad (2002), *Se rør i bakken uten graving*. Tilgjengelig fra:
<https://www.tu.no/artikler/se-ror-i-bakken-uten-graving/219794> (Hentet: 26. mars 2020).

Unity (2020), *About AR Foundation*. Tilgjengelig fra:
<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@3.1/manual/index.html> (Hentet: 29. april 2020).

Unity (2020), *Native plug-ins*. Tilgjengelig fra:
<https://docs.unity3d.com/Manual/NativePlugins.html> (Hentet: 30. april 2020).

Unity (2020), *Prefabs*. Tilgjengelig fra: <https://docs.unity3d.com/Manual/Prefabs.html> (Hentet: 30. april 2020).

Unity (2020), *Unity 2020.1 beta*. Tilgjengelig fra: <https://unity3d.com/beta/2020.1b> (Hentet: 3. april 2020).

Unity (2020), *Unity Core Platform*. Tilgjengelig fra: <https://unity.com/products/core-platform> (Hentet: 19. mars 2020).

Unity (2020), *Who we are*. Tilgjengelig fra: <https://unity.com/our-company> (Hentet: 18. mars 2020).

U.S. government (2017), *GPS Accuracy*. Tilgjengelig fra: <https://www.gps.gov/systems/gps/performance/accuracy/> (Hentet: 24. mars 2020).

Vann- og avløpsetaten (u. å.), *Vann i Bergen*. Tilgjengelig fra: <https://www.facebook.com/bergenvann/> (Hentet: 5. mai 2020).

VAnytt (2016), *Presis lokalisering av rør i bakken med nytt instrument*. Tilgjengelig fra: <https://www.vanytt.no/2016/12/14/presis-lokalisering-av-ror-i-bakken-med-nytt-instrument/> (Hentet: 26. mars 2020).

vGIS (2020), *Locate Infrastructure Faster and Prevent Errors*. Tilgjengelig fra: <https://www.vgis.io/esri-augmented-reality-gis-ar-for-utilities-municipalities-locate-and-municipal-service-companies/> (Hentet: 5. april 2020).

Visual Paradigm (2020), *What is a Sprint in Scrum?*. Tilgjengelig fra: <https://www.visual-paradigm.com/scrum/what-is-sprint-in-scrum/> (Hentet: 26. mars 2020).

White, G. (2018), *Projected Augmented Reality*. Tilgjengelig fra: <https://cantina.co/projected-augmented-reality/> (Hentet: 25. mai 2020).

Wikimedia Commons (2012), *Augmented-reality-1957411 1920*. Tilgjengelig fra: https://commons.wikimedia.org/w/index.php?sort=relevance&search=augmented+reality&title=Special:Search&profile=advanced&fulltext=1&advancedSearch-current=%7B%7D&ns0=1&ns6=1&ns12=1&ns14=1&ns100=1&ns106=1#/media/File:Augmented-reality-1957411_1920.jpg (Hentet: 26. mai 2020).

Wikipedia (2016), *Augmented Reality for eCommerce*. Tilgjengelig fra: https://commons.wikimedia.org/wiki/File:Augmented_Reality_for_eCommerce.jpg (Hentet: 26. mai 2020)

Wikitude (2017), *Wikitude Elected Best Augmented Tool*. Tilgjengelig fra:
<https://www.wikitude.com/blog-wikitude-winner-auggie-awards-2017/> (Hentet: 26. mars 2020).

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., & Wesslén, A. (2012). *Experimentation in Software Engineering* (2012 ed., Vol. 9783642290442). Heidelberg: Springer Berlin Heidelberg.

ZoARG (2018), *Spyunderground2016*. Tilgjengelig fra:
<http://www.zoarg.com/spytheunderground2016> (Hentet: 20. mars 2020).

9 VEDLEGG

9.1 Risikoliste

K = konsekvens for at hendelsen inntreffer. S = sannsynlighet for at hendelsen inntreffer. R = Risikofaktor. 1 - 5.

Nr:	Hovedaktivitet	Fare:	Årsak:	K :(1)	S :(2)	R :(3)	Tiltak	K:	S:	R:
1	Kobling mellom AR-verden og GPS-koordinater	Koblingen går ikke. Kan miste synligheten av det som blir laget.	To forskjellige systemer. AR-verdenen bruker meter, mens GPS har longitude og latitude.	4	2	8	Høre med tidligere lignende prosjekter hvordan det ble løst, og bruke kunnskapen til oppdragsgiver om teknologien.	3	2	6
2	Bruk av GPS-koordinater	I verste fall vil det bli en unøyaktighet på 10-20 meter radius.	GPS er ikke så nøyaktig.	3	3	9	Manuell kalibrering til et fast punkt som for eksempel et kumlokk	2	1	2
3	Leveranse av ressurser	Prosjektet får ikke ressursene til avtalt tid. Utsetter derfor arbeidet. Får ikke riktig informasjon om ressursene prosjektet har fått utlevert.		4	2	8	Holde kommunikasjonen mellom oppdragstakere og VA-etaten.	2	1	2
4	Manglende kompetanse	Kommer ikke i mål med prosjektet.	Ny teknologi.	4	3	12	Bruke mye tid på å lære den nye teknologien.	2	2	4
5	Kommunikasjon	Bruker lang tid/ikke bli ferdig med prosjektet. For lite kommunikasjon. Gjør dobbelt arbeid/ feil fordelte oppgaver. Dårlig samarbeid.	Covid-19 virus, og må ha hjemmekontor.	3	3	9	Chatte ofte på Slack og ta daglige samtaler.	3	1	3
6	Valg av rammeverk og biblioteker	Det kan koste penger, vanskelig å bruke, ikke riktig for prosjektet. Kan hende det ikke er riktig bibliotek for det som skal gjøres i prosjektet.	Undersøke hva som er blitt brukt tidligere, hva som er enkelt og bra å bruke.	2	3	6	Rådføring hos Bouvet for hvilket utviklingsmiljø de anbefaler og se på tidligere arbeid av lignende prosjekt.	2	1	2

7	Balansering av prosjekt og rapport	Bruke for mye tid på prosjektet og for lite tid til rapporten.	Utviklerne feilberegner hva som tar hvor lang tid. Går altfor dypt inn i prosjektet.	3	2	6	Passe på at frister blir holdt, og at det hele tiden skrives litt på rapporten.	2	1	2
8	Forståelsen av oppgaven	Utviklerne gir ikke ønsket produkt til arbeidsgiver.	Mangel av kommunikasjon og testing.	5	2	10	Hyppig kommunikasjon mellom oppdragsgiver oppdragstakere.	3	1	3
9	Tidsbruk	Komme på etterskudd.	Undervurdering av tiden det tar å gjøre forskjellige ting.	2	3	6	Bruk av agile metoden Scrum gjennom prosjektet for å hele tiden tenke fremover og ha nye oppgaver som skal bli gjort.	1	2	2
10	Bruk av utstyr	Ulike komponenter, utviklingsmiljø eller rammeverk blir utilgjengelig for prosjektet under arbeid.	Oppdateringer som gjør ting	3	2	6	Vurdering av oppdatering: kan det unngås? Trengs denne oppdateringen for å videreutvikle? Vil oppdateringen ødelegge fremgangen?	2	1	2

Tabell 9.1 Risikoliste

9.2 Testing av applikasjonen

Utført: Torsdag 7. mai 2020 kl. 14:00-14:30

Sted: Høgskulen på Vestlandet

Tilstede: Simen Romstad, Arisa Seljestokken, Fredrik Mørk, Ali Mekki og Tommas Kraft.

Agenda

1. Spør om tillatelse til å filme testingen (for å bruke til videre arbeid, dokumentasjon i rapporten og promo-video).
2. La VA-etaten prøve applikasjonen uten forklaring, deretter forklare hvordan gruppen tenker bruker-interaksjonen skal foregå.
3. Fortelle om hva gruppen har klart på den gitte tidsperioden, og videre arbeid. Opplyse om unøyaktighet og hvordan dette kan forbedres.
4. Høre på innspill om hva de syntes om brukergrensesnittet.
5. Spørre om hva de så for seg, deres forventninger til prosjektet ift. hva gruppen oppnådde.
6. Spørre om hvilke ekstra funksjoner de kan se for seg i fremtiden (til videre arbeid).

Resultat

For testing av nåværende app møtte bachelorgruppen to personer fra VA-etaten, Ali og Tommas. Disse to representerer VA-etaten i denne testen. Under utvikling og planlegging av appen, har bachelorgruppen hatt jevnlig kontakt med begge. Ali var først ut med testing, og etterpå Tommas. Her er en oppsummering av møtet:

- Det var greit at gruppen tok videoopptak av hendelsen. VA-etaten ville gjerne se den endelige promo-videoen som opptakene bla. skulle bli brukt til. Gruppen ble enige med VA-etaten om å sende lenke til videoen når den er blitt ferdig.
- Første steg i applikasjonen var litt forvirrende. Appen legger opp til at brukeren skal skanne omgivelsene, men dette var litt for lite tydelig å forstå for brukeren. Test-brukeren snudde

telefonen sakte (nesten i ro), og skanningen gikk mye saktere enn det som egentlig er mulig. Forslag fra VA-etaten var å gjøre dette steget mer tydelig, for eksempel med en mer presis tekstbeskrivelse.

- Retningen på kompasset viste rett veg. Ingen kommentar, alt var rett. Skulle gjerne sett hva som hadde vært reaksjon om kompasset viste feil veg.
- Plasseringen av rør var omtrent to til tre meter feil. Dette mente VA-etaten var forståelig. De kom med et forslag om å bruke en ekstern antenne som kan finne posisjon og kompass med centimeter presisjon.
- Vi foreslo å bruke kumlokk til å kalibrere rørene. Dette forslaget likte VA-etaten godt. Å la brukeren kalibrere selv, er enda bedre enn å bruke ekstern antenne, sa VA-etaten.
- Applikasjonen var slik som VA hadde sett for seg. Tommas sa også at han var imponert over hva gruppen hadde fått til. Han mente dessuten at gruppen i sluttrapporten måtte legge vekt på at en stor del av oppgaven handlet om å tolke og lese rørdatabasene.

9.3 Brukermanual

Visualizing pipes in Augmented Reality

This bachelor project was to develop a mobile application for the production of water and sewer pipes in Augmented Reality. The application can be used for workers who avoid these pipes when digging or building new buildings. Today's solution is currently in 2D. The project provides a proposition for a 3D solution that makes it easier to interpret the pipeline data for the water and wastewater treatment plant.

Getting Started

These instructions will get you a copy of the project up and running on your local machine for development and testing purposes. See deployment for notes on how to deploy the project on a live system.

Prerequisites

This is what you need to install the application on your device.

```
Unity  
Android SDK  
AR Foundation  
AR Kit  
AR Core
```

Installing

How to get the development env running. Step by step:

```
0. In terminal/cmd type in: git clone https://github.com/fredrikmark/Visualizing_pipes-AR.git (Make sure you have git installed)
1. Make sure you have the Unity version 2019.2.21f1
2. Make sure to have Android SDK build
In Unity:
3. Download ARFoundation version 2.0.2

4. IF IOS : Download ARKit
4. IF ANDROID: Download AR Core

5. To build the project you have to check the "Developer" checkbox and then name the build to "Android" or "iOS"
6. Plug in an Android-device to the computer and run.
```

If you develop on an iPhone, you also need to have:

```
7. Xcode minimum version 11.4.1 (Don't open Xcode)
8. After build in Unity, you need to go in the "iOS" folder and type the command in the terminal window: pod and then: open ./Unity-iPhone.xcworkspace
9. Go to "Signing & Capabilities" and select a Team. If you don't have one, make one.
10. In the info.plist file: add a row, Privacy Bluetooth always usage description : allow
11. Build and the run.
```

This should work for both devices.

Built With

- [Unity](#) - Unity Engine
- [Spatial Anchors](#) - Microsoft Azure Spatial Anchors
- [AR Foundation](#) - AR interactions

Versioning

1.0.1

Authors

- Arisa Seljestokken - *Initial work* - [Arisa](#)
- Simen Osland Romstad - *Initial work* - [Simen](#)
- Fredrik Garseg Mørk - *Initial work*

Acknowledgments

- Many thanks to our supervisor at Bouvet, [Fredrik Chrislock](#)

9.4 Brev fra ekstern veileder

Til de det måtte angå,

Jeg, Fredrik Chrislock, har vært assisterende veileder for Fredrik Garseg Mørk, Arisa Seljestokken og Simen Osland Romstad som i de siste månedene utviklet et «romlig» brukergrensesnitt basert på utvidet virkelighet. Målet har vært å skape et støttesystem for arbeidere som lar dem visualisere innebygde strukturer som ellers ville vært usynlige. Dette bidraget har en verdi fordi arbeiderne får en mer presis situasjonsforståelse før de tar viktige beslutninger og iverksetter kostbare tiltak. Som en del av arbeidet har teamet fokusert på å lage en god brukeropplevelse og et produkt som er versatilt nok til å brukes på forskjellige enheter. Teamet har jobbet systematisk, målrettet og demonstrert en genuin interesse for å levere et produkt av god kvalitet. Løsningen som teamet har endt opp med demonstrerer potensialet som ligger i utvidet virkelighet og de spennende mulighetene dette åpner for i næringslivet.



Fredrik Chrislock,
Konsulent, Bouvet ASA