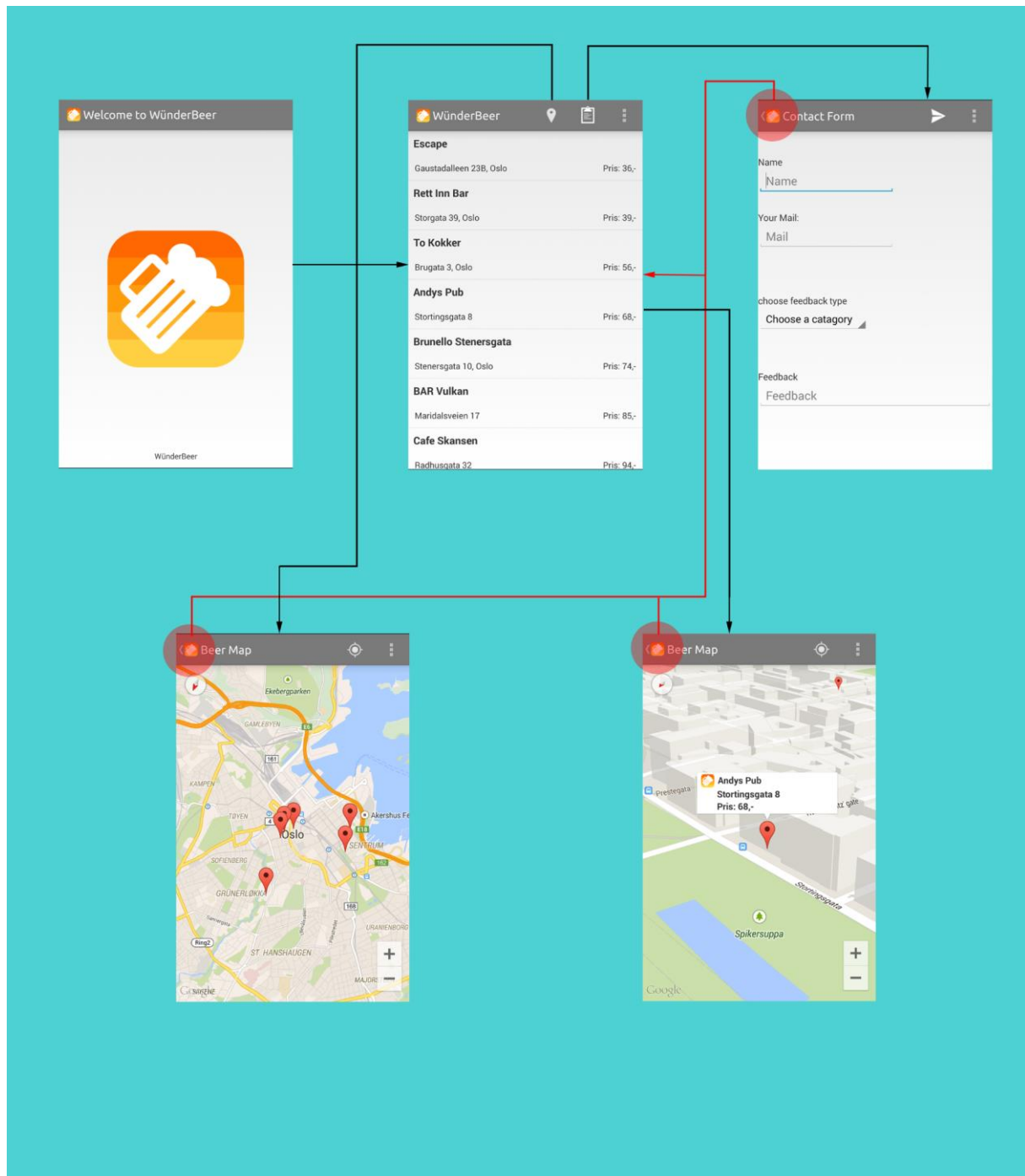


APPLIKASJONSUTVIKLING

3 mappeinnlevering

Audun Karlsrud Larsen /// S172860



Illustrasjon1: Navigasjonsflyt for funksjonalitetene i applikasjonen. Den svarte pilen indikerer flyt fremover i applikasjonen, mens den røde pilen illustrerer tilbake-navigasjon. En utfyllende forklaring rundt navigasjon finnes blir gjengitt i rapporten.

Readme:

Min applikasjon sentrer seg primært rundt funksjonalitet knyttet til GoogleMapsv2 APIet. Min utvikling har foregått på en Sony Xperia Z2 og ikke en emulator. Grunnlaget for dette valget er at emulator som utviklingsverktøy ikke støtter GoogleMaps funksjonalitet i like stabil grad som en fysisk enhet. Applikasjonen min er før innlevering testet på Emulator, der ble mine antagelser bekreftet da det kan forekomme en error «Fatal signal 11(SIGSEGV)» Etter undersøkelser på nettet har jeg avdekket at denne feilen hovedsakelig skyldes det interne minnet til emulatoren, og gjennspeiler ikke feil i programmet. Jeg vil derfor anbefale at testing foregår på en fysisk enhet.

Videre er det komplekst å ta i bruk GoogleMaps funksjonalitet grunnet nøkkelen som er knyttet opp mot pakkenavnet, jeg sender derfor med en debug.keystore fil. Ved å benytte seg av denne muliggjør det testing av applikasjonen på andre enheter uten å måtte opprette nye egen nøkkel. Jeg har selv testet denne funksjonaliteten på 2 andre enheter for å forsikre meg om at denne løsningen fungerer.

Antagelser og konseptstadiet

På bakgrunn av erfaring med innlevering av mappe1 og mappe2 var jeg klar over mengden tid som må dedikeres for å få en applikasjon man kan være tilfreds med, samt en applikasjon som måler grad av forståelse og kunnskaper rundt rammeverket. Jeg tok mye lærdom av de to foregående arbeidsprosessene og hentet ut erfaringer som tidsbruk, kontinuerlig testing og arbeid med utvikling av designelementer.

Spesielt for mappeinnlevering 3 var kunnskapen om den nødvendige arbeidsmengden kombinerte med en allerede hektisk eksamensperiode, som inneholdt 2 rapportinnlevering, er 2 ukers hjemmeeksamen samt en muntlig eksamen. Hvordan disse rammebetingelsene har preget oppgaven kommer frem i teksten samt i punktet «erfaringer».

På bakgrunn av feedback fra faglærer ble vi i denne helt frie oppgaven oppfordret til å velge en applikasjon, og tema som vi fant interessant. Jeg drøftet oppgaven med medstudenter på Universitetet i Oslo for å få innspill og forslag til mulige applikasjoner som kunne vise seg interessante samt lærerike og utvikle. Vi kom inn på studenter sine interesser og deres trange budsjett. Idemyldringer fortsatte i dette sporet før vi kom inn på øl, og viktigheten det har i en studenthverdag. (Jeg fastsatte også 2 andre konkrete ideer, men vil ikke gå nærmere inn på disse)

Min ide ble dermed å lage en applikasjon, som ved hjelp av de nyeste rammeverkene vi har lært i kurset. Applikasjonen skal informere og grafisk fremstille hvor, og hvordan en student kan finne og navigere seg frem til utestedet som har øl i laveste prisklasse.

Arbeidet startet ved at jeg hentet ut informasjon fra Aftenposten sin restaurantanmeldelses sider. Jeg satt igjen med JSON data for ca 150 utesteder i Oslo, med deres ølpriser, navn og åpnings/stengnings tid. Dokumentasjonen forteller videre hvordan jeg brukte dataen til å forme min applikasjon og arbeidet videre:

Design:

Jeg har utviklet mine designelementer basert på prinsippene fra *AndroidDeveloper*, samtidig som jeg har forsøkt å implementere mine egne design kunnskaper. Jeg har også forhørt meg med flere

testbrukere. Disse stegene har jeg tatt for å få feedback og øke validiteten rundt navigasjon og brukervennlighet.

Gjenkjennbare navigasjonsmønstre

Jeg har som et gjennomgående element valgt å implementere meny min applikasjon. Menyen opprettholdes som en konstant i alle skjermbildene. Dette valget er tatt bevist for å opprettholde et gjenkjennbart navigasjonsmønster internt i applikasjonen. Som det blir spesifisert på *AndroidDeveloper* «Simplyfy My Life» og «I should always know where I am» Den menyen inneholder en tilbakeknapp, som også er konstant på alle skjermbildene i applikasjonen. Tilbakeknappens posisjonering er i tråd med både *AndroidDeveloper* sine prinsipper, og mine undersøkelser på eksisterende og lignende applikasjoner på Android platformen. Dette er etter min oppfattelse også en konvensjon i Android-utvikling.

Navigasjonsmenyen innehar også et navigasjonselement på høyre side. Symboliserte i tråd med *AndroidDeveloper* sine retningslinjer med tre vertikale rundinger. Ved denne menyen får man tilgang til å avslutte applikasjonen uavhengig av hvilket skjermbildet man befinner seg i. Grunnlaget for dette valget er å øke navigasjonen til brukere og gi friheten til å avslutte applikasjonen uavhengig av hvor man befinner seg i applikasjonens livssyklus.

Som et konsekvent valg i min applikasjon har jeg ingen tekstbaserte knapper. Navigasjonsmønsteret i applikasjonen er ikonbaserte. Ikonene er hentet fra *AndroidDeveloper* sin offisielle ikonpakke. For valg av symboler har jeg foretatt enkle brukerundersøkelser for å teste symbolers mening, og har kommet frem til et utvalg av symboler som virket intuitive og klare i kontekst for mine testbrukere.



Illustrasjon 2: Viser den ikonbaserte navigasjonen.

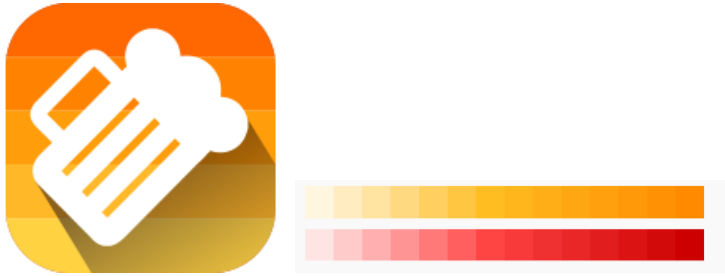
Jeg har også overholdt og fulgt prinsippene: «Only show what I need when I need it» Dette er løst ved at de ulike skjermbildene i applikasjonens livssyklus kun viser frem funksjonalitet som er relevant for det aktuelle skjermbildet. Eksempelvis i kart funksjonen er det muligheter for å hente min posisjon på kartet, mens i tilbake meldingsskjemaet har man muligheten å sende inn en tilbakemelding. For å klargjøre, all funksjonalitet som ikke er relevant for skjermbildet man operer i er fjernet for å minske risiko for bruker-genererte feil, og bedre navigasjonsflyten.

Applikasjonsikon og fargevalg

Jeg har hentet inspirasjon for tidligere prosjekter, og populære Android applikasjoner. På bakgrunn av kunnskapen jeg har opparbeidet meg har jeg implementert et minimalistisk design med lett tilgjengelige og intuitive ikoner for navigasjon.

Applikasjonsikon

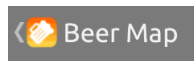
For å forklare fargevalget er det viktig å skjønne konteksten for applikasjonen. Jeg valgte farget pelleten min basert på et ønske om å skape assosiasjoner til følelser man får når man drikker øl. Løsningen på dette var å velge farget fra den lunere delen av fargeskalaen. Jeg har valgt å benytte meg av en kombinasjon av Orange og som heller mot rødt. Testing viser at logoen appellerer til det estetiske hos brukeren, og at den er intuitiv og enkel for å forstå det overordnede temaet til applikasjonen.



Illustrasjon 3: applikasjonsikon og farge pellet.

Fontvalg

Jeg har valgt vekk den standardiserte fonten som blir benyttet i Android rammeverket, valget har blitt tatt for å øke det estetiske bildet av applikasjonen basert på feedback fra tester. For å endre fronten har jeg hentet retningslinjer/hint fra *AndroidDeveloper* samt fontvalg tatt i andre populære applikasjoner. Jeg har valgt en font som støtter opp under *AndroidDeveloper sine prinsipper*:



«typographic tools such as scale, space, rhythm, and alignment with an underlying grid.

Successful deployment of these tools is essential to help users quickly understand a screen of information»

Illustrasjon 4: min valgte font.

Oppgavens Struktur:

Jeg har i oppgaven opprettet følgende struktur:

- 1) Hovedklassen MainActivity, denne klassen tar for seg all kommunikasjon med AsyncTask gjennom en intern klasse. Samtidig står klassen ansvarlig for parsing av JSON filen i assets mappen til prosjektet. Denne funksjonaliteten er gitt til MainActivity fordi den er essensiell for hele applikasjonens livssyklus. All videre funksjonalitet benytter seg av denne genererte dataen. Ved navigasjon kan man bli sendt til en av de to eksisterende barneaktivitetene til MainActivity
- 2) BeerMap er en av disse barneaktivitetene, denne klassen tar for seg all funksjonalitet knyttet til map fragmentet som blir definert i XML filen. Klassen oppretter markers, definerer current location, genererer informasjonsvinduer osv.
- 3) BeerContact er den siste barneklassen til MainActivity, denne klassen inneholder kontaktskjemaet som skal gi feedback til meg som utvikler. Klassen operer med en rekke tekstfelder med validering argumentert for i punktet inputt kontroll. Klassen setter opp en ny epost intent med informasjonen fra edittext boksene, og sender brukeren videre til ønsket mailklient, ved vellykket sending får bruker feedback og blir returnert til applikasjonens livssyklus.
- 4) Beer klassen er en standard POJO klasse med dedikerte variabler for hvert objekt i Java objektorientert ånd. Klassen implementerer gettere og settere for hvert attributt.
- 5) Klassen BeerListAdapter er klassen som fyller ListViewet med informasjon fra ArrayListen med objekter. **CustomArrayAdapter** er en tilpasning av ArrayAdapter som gir meg muligheten til å populære listViewet med beer-objekter.
- 6) MyApplication er en applikasjonsklasse som inneholder den globale arraylisten for øke tilgangen til den informasjonen lagret gjennom applikasjonens livssyklus.
- 7) TypefaceSpan er en hjelpeklasse som endrer ønsket font basert på otd filer i assets folderen.

Mine tekniske løsninger

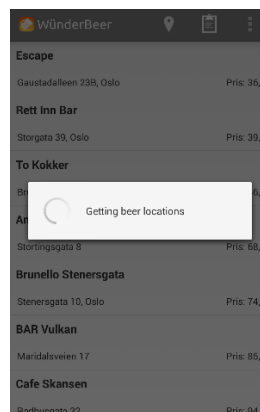
AsyncTask JSON

Informasjonen som tilrettelegger for funksjonaliteten til min applikasjon er strukturert som JSON objekter, som inneholder et JSONArray. Programmet leser manuelt inn objektene fra JSON filen ved og returnerer en String som inneholder informasjonen. Denne Stringen blir så splittet og jeg oppretter POJOs til de individuelle JSON objektene jeg har lest fra filen. Jeg har valgt å ikke benytte meg av eksisterende API som GSON. Grunnlaget for dette er at jeg ønsker å vise at jeg har forstått funksjonene som ligger bak dette APIet, og har selv valgt å manuelt opprette mine POJOs. Objektene blir så lest inn i et ArrayList. ArrayListet blir globalt gjennom opprettelsen av myApplication klassen.

AsyncTask

Jeg benytter meg av AsyncTask for å iterere over mine Java objekter i listen, og for hvert objekt som blir sendt til AsyncTask sender jeg med objektet og dens sin adresse. I retur får jeg en samling JSON objekter, jeg går inn og henter ut latitude og longitude, og oppdatere POJO klassen med de respektive variablene. Jeg har implementert en sjekk metode som gjør at AsyncTasken og ListViewet kun blir oppdatert en gang og ikke ved kjøring a onCreate metoden.

Utover den overnevnte informasjonen har jeg også implementert en «loading message» Dette er en av de mest kritiske funksjonalitetene i min applikasjon. Min AsyncTask må itererer over alle mine Beer Objects før de innehar koordinater til kartfunksjonaliteten. Hvis man forsøker å aksesserer kartfunksjonalitet før AsyncTasken er fullført vill applikasjonen krasje. Jeg har løst dette ved å implementere et Interface med en metode jeg kan fritt aksessere. Jeg tester på boolean verdier for å avgjøre når dataen er ferdig synkronisert, og fremviser beskjed til brukeren om prosessen ikke er fullført.



Illustrasjon 5: loadingscreen

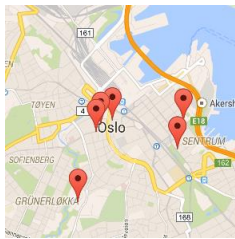
ListView

Jeg har valgt å grafisk fremstille min data i et listview, dette er for å gi brukeren enkel oversikt over utestedene applikasjonen innehar informasjon om. For å gjøre dette har jeg laget en CustomAdapter, som gjør det mulig å populære Listiewet fra min ArrayList med objekter. Objektene i listViewet blir også sortert etter pris og relevante attributter blir presentert. Prisen på ølen har blitt plassert mot høyre slik at det skal skille seg ut fra navn og adresse. Dette er et bevist valg, da hoved funksjonaliteten til applikasjonen skal være å vise frem billige enhetspriser.

Spannable

For å realisere mitt mål om å endre fonten som blir fremvist i min ActionBar har jeg implementert en klasse TypefaceSpan, ved et kall på denne metoden sendes ønsket streng og ønsket font inn til konstruktøren som variabler, og det returneres den innsendte strengen med en oppdatert drawstate, slik at den blir fremvist med den ønskede fonten.

GoogleMaps



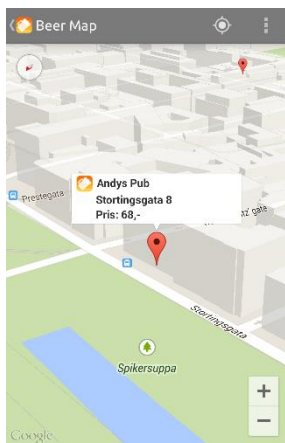
Min kartfunksjonalitet innehar flere funksjonaliteter, gjennom å trykke på ikonet for kart kommer man inn til et utsnitt av Oslo, som viser custom markers for vært individuelle utesteder. Jeg har utviklet en egen MarkerInfoAdapter som fremviser informasjonen om de individuelle utestedene. Videre har jeg skjult Google sin standardiserte lokalisasjonsknapp fra selve kartfragmentet og flyttet den opp til min ActionBar, dette er gjort i tråd med min visjon om å holde navigasjonen og interaksjonen med knapper konstant.

Illustrasjon 6: Utsnitt over Oslo.



Illustrasjon 7: min lokalisasjon knapp.

Videre har jeg implementert en onclicklistener i ListViewet, ved å trykke på ønsket utestedet blir man transportert direkte til utested sin lokalisasjon på karet. Her har jeg utnyttet funksjonalitet i Google Maps V2 og endret til en 3 dimensjonal fremvisning av det ønskedes utestedet. Jeg har løst dette ved å hente ut det markerte objektet sin informasjon basert på posisjon i ListViewet, jeg legger så denne



informasjonen inn i en bundle sammen med en nøkkel. I BeerMapActivity henter jeg ut informasjonen fra bundelen. Jeg henter videre ved en hjelpemetode det aktuelle objektet som innehar variablene jeg sendte ved. Jeg har tatt utgangspunkt i adresse da et utested kun kan ha en unik adresse i Oslo. Jeg arbeider ut fra denne tekstvariabelen, og endrer kartet sin zoom funksjonalitet til objektet man trykket på i ListViewet. Jeg har benyttet meg av en animasjon på 2000 ms for å gi brukeren et godt bilde på hvor det ønskede utestedet befinner seg på kartet, etter zoom er fullført kommer automatisk MarkerInfovinduet til det aktuelle utestedet frem, slik at brukeren har umiddelbar informasjon om det valgte utestedet. Jeg har valgt denne løsningen for å ta fullt utbytte av min nye kunnskap om GoogleMaps V2.

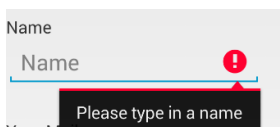
Illustrasjon 8: Detaljbildet over 3d funksjonalitet.

SplashScreen

Jeg har implementert en SplashScreen ved installering av min applikasjon, grunnlaget for dette valget er å vise frem min vise frem logoen og skape assosiasjoner til applikasjonens tema. Splash screenen vises kun ved innstallering av applikasjonen.

InputKontroll

BeerContact klassen er et kontaktskjema, kontaktskjemaet er har som hovedfunksjon og gjøre det mulig for brukere og gi feedback til utvikleren av applikasjonen, samt rapportere om de finner øl tilgjengelig på utesteder til en billigere penge. Siden informasjonen som blir innhentet gjennom kontaktskjema ikke inneholder noen sensitiv informasjon og skal implementeres i noen database, har jeg valgt å legge all validering til XML feltene, og skriver ut feilmelding i det individuelle feltet om det mangler innputt. Jeg fant det ikke nødvendig og implementer Regex basert



validering (Slik jeg gjorde i mappe2) da all data blir sendt til min personlige mail. Hvor jeg vil foreta en subjektiv evaluering om jeg benytter den gitte feedbacken til å forbedre applikasjonen min. Jeg ønsker allikevel at feltene skal fylles ut for at jeg enklere skal ha oversikt over tilsendt informasjon, og for eventuelle kontaktmuligheter i betastadiet.

Utfordringer og oppgavevalg

Jeg var klar over den generelle utfordringen av å ferdigstille løsningene mine innenfor den gitte tidsfristen, jeg hadde i perioden også et ekstremt arbeidspress med eksamener (som nevnt i innledningen) som bød på store tidsmessige utfordringer. Jeg har løst dette ved å dedikere store deler av fritiden min til applikasjonsutvikling. Grunnen til dette valget var at jeg kjente læringseffekten og mestrings-følelsen av å beherske et nytt rammeverk. Jeg ønsket med denne oppgaven å lage en applikasjon bygget rundt et interessefelt, samtidig som jeg ønsket å gjøre noe jeg syns var gøy. Et viktig valg jeg tok var at jeg med min mappeinnlevering 3 ønsket å benytte meg av rammeverk og løsninger jeg ikke tidligere hadde vist at jeg behersker. Jeg har gjennom de to tidligere oppgavene fått vist frem mine kunnskaper. Jeg ville med denne oppgaven vise at jeg også kunne tilegne meg kunnskap som ikke var gitt i en oppgavetekst.

En utfordring har vært å forstå mekanismen bak AsyncTask, og finne ut hvordan man henter ut den returnerte informasjonen. Løsningen ble etter mye debugging å innse at Beer objektet måtte bli sendt gjennom background tråden, og at resultatet fra onPostExecute måtte kalle på en metode gjengitt i UI tråden. For så å bli parset og oppdatere de ønskede koordinatene til det aktuelle objektet.

En annen utfordring har vært tiden det tok å sette seg inn i hele Google Maps APIet, og forstå funksjonalitet og metodene ved å opprette kart. En stor utfordring lå i og implementere Custom markers med et egendefinert informasjonsvindu med bilder.

Videre ønsket jeg å implementere den overnevnte OnClick funksjonaliteten i ListViewet for å bedre funksjonaliteten og øke kompleksiteten til min applikasjon, dette testet meg på å benytte opparbeidet kunnskap og kombinere dette med den nye funksjonaliteten i Google Maps v2.

Et viktig aspekt jeg har valgt å rette fokus mot i denne mappeinnleveringen er å ikke bruke tiden min på å implementere ekstrafunksjonalitet jeg har vist tidligere at jeg behersker. Jeg har derfor ikke utviklet en widget, eller støtte for flere språk. Jeg har istedet rettet fokus mot å utbedre funksjonaliteten til min eksisterende applikasjon ved å tilegne meg detaljert kunnskap om et nytt felt innenfor Android utvikling. Jeg føler at dette blir representert ved funksjonaliteten i min applikasjonen, og at det reflekterer hvor mye tid jeg har dedikert til dette faget.

Jeg ønsket også å implementere GeoFence kompatibilitet. Jeg satt meg inn i kompleksiteten til funksjonaliteten, og benyttet meg av preCode fra Googles hjemmesider. Dette var et tidskrevende og komplekst arbeid, det bestod av 6 underklasser som måtte kommunisere. Jo fikk funksjonaliteten oppe å gå, men den var for ustabil i testingen, jeg tok valget om å fjerne funksjonaliteten slik at faglærer ikke opplevde dårlig erfaring med applikasjonen grunnet en uferdig funksjonalitet. Dette blir heller et fokuspunkt ved videre utvikling.

Videre potensiale og muligheter

Jeg ønsker å fortsette arbeidet med min applikasjon og utvikle løsninger slik at denne kan bli min første publiserte applikasjon på Google Play Store. For å fullføre dette målet er det et par funksjonaliteter jeg ønsker å legge til programmet mitt. Potensiale for en søkefunksjon i ListViewet mitt kom frem da jeg utvidet applikasjonen og benyttet meg av den fulle JSON filen med innhentet

data. Dette er noe jeg ikke hadde mulighet til å implementere i den leverte løsningen men jeg har klart bildet av hvordan man løser dette ved hjelp av filtrering og to parallelle Arrayer med informasjon som blir fremvist i ListViewet.

Jeg ønsker også å implementere et stjernesystem, slik at man kan favorittseirer lokalisasjoner og barer man likte spesielt godt. I min innhentede data har jeg også tilgang til stengetider for ølsalget, jeg har ikke valgt å benytte meg av denne dataen aktivt i applikasjonen, men nærme innlevering innså jeg nytteverdien av denne dataen og jeg ønsker å implementere en boolean verdi som sjekker opp mot det nåværende tidspunktet og oppdatere ListViewet til å vise om det ønskede utestedet har sin servering åpen.

Erfaringer:

Denne utviklingsprosessen har gitt meg masse kunnskap om forskjellige APIer, den har også gitt meg innsikt i utvikling basert på et helt fri oppgaven og alle valgmulighetene dette medfører. Jeg fornøyd med at jeg har klart å utvikle alle elementer i applikasjonen alene, samtidig som jeg har klart å benytte meg av kunnskapen jeg har opparbeidet meg i kurset. Alt dette har skjedd innenfor en ekstremt tidspresst periode. Jeg føler at sluttproduktet reflekterer de dedikerte timene jeg har lagt inn i design, navigasjon, opparbeiding av kunnskap, programmering og brukertesting.