UNIVERSITETET I AGDER

IS-304, 2021

Explainable Artificial Intelligence (XAI)

Emnekode:	IS-304
Emnenavn:	Bacheloroppgave i informasjonssystemer
Emneansvarlig:	Hallgeir Nilsen
Veileder:	Hallgeir Nilsen
Oppdragsgiver:	TietoEVRY

Studenter:

Etternavn	Fornavn
Befring	Hedda
Garborg	Sander Kvale
Gulliksen	Henrik
Hjermann	Jacob Mario
Vereide	Sander

Jeg/vi bekrefter at vi ikke siterer eller på annen måte bruker andres arbeid uten at dette er oppgitt, og at alle referanser er oppgitt i litteraturlisten.	JA X	NEI
Kan besvarelsen brukes til undervisningsformål?	JA X	NEI
Vi bekrefter at alle i gruppa har bidratt til besvarelsen	JA X	NEI

Forord

Denne rapporten er hovedsakelig skrevet for personer med kjennskap til tekniske fagbegreper innen systemutvikling og det agile rammeverket Scrum.

Vi vil benytte denne anledningen til å uttrykke vår takknemlighet til alle de personene som har vært involvert i vårt bachelorprosjekt. Det har vært en lærerik prosess hvor kunnskap og tips fra de involverte har vært til stor hjelp.

Takk til Hallgeir Nilsen for god oppfølging og veiledning på prosjektet. Du har gitt oss gode råd knyttet til prosjektstyring og planlegging. Du har også vært tilgjengelig alle gangene vi stod fast og trengte veiledning.

Takk til Roar Engen og Elisabeth Cederberg Øvensen fra TietoEVRY avdeling Kristiansand. Dere tok oss godt imot fra dag en og har bidratt til at alle på gruppen har følt seg velkommen.

Takk til Nils Fredrik Iselvmo Bjerk og Karl Vetle Huvestad Rødbakk som også har vært til god hjelp gjennom hele vår periode hos TietoEVRY. Det har vært hjelpsomt å ha dere som veiledere da dere også har gjennomført samme type bachelorprosjekt.

Til slutt vil vi også takke Duncan Irving og Kolbjørn Flaarønning for gode tilbakemeldinger og hjelp relatert til AI.

Hedda Befring

Henrik Gulliksen

Jacob Mario Hjermann

Sander Kvale Garborg

Sander Vereide

Sammendrag

Denne rapporten er skrevet i forbindelse med bacheloroppgaven i emnet IS-304 våren 2021. Rapporten er skrevet av Internet Explorers, som er en gruppe bestående av 5 studenter fra Universitetet i Agder. Oppgaven er skrevet i samarbeid med TietoEVRY og går ut på å teste ut kundetilfredsheten på beslutninger tatt av *Explainable Artificial Intelligence* (XAI) versus *Artificial Intelligence* (AI).

Problemstillingen resulterte i en Flask webapplikasjon hvor brukere kan legge inn egenskaper ved en bolig, og deretter få oppgitt et prisestimat med en tilhørende forklaring. Estimeringene ble gjort ved hjelp av en maskinlæringsmodell fra Google Cloud AutoML som ble trent opp med historisk salgsdata på boliger i Oslo. For å teste ut kundetilfredsheten har to testgrupper blitt benyttet hvor den ene fikk oppgitt estimatet med tilhørende forklaring, mens den andre kun fikk estimatet. Deretter ga testobjektene en tilbakemelding på sin opplevelse med systemet.

Scrum rammeverket har blitt benyttet for å sikre agilitet gjennom hele prosjektløpet. Scrum metodikken er godt utbredt i arbeidslivet, og har også blitt brukt gjennom studieløpet. Ettersom tanken var at gruppen skulle utforske ny teknologi, og utfordringer kunne oppstå, var det et bevisst valg å benytte Scrum. På denne måten kunne eventuelle utfordringer bli håndtert, slik at kontinuitet ble sikret underveis i prosjektet. I tillegg har rammeverket blitt hyppig brukt hos oppdragsgiver.

Fra prosjektets start har gruppen vært påvirket av den pågående covid-19 pandemien. Dette har ført til begrensninger på fysisk oppmøte. Likevel har samarbeidet og kommunikasjonen vært god, da gruppen tidlig etablerte gode rutiner med blant annet faste arbeidstider. Microsoft Teams har vært en avgjørende kommunikasjonsplattform så vel som dokumentlagringstjeneste for gruppen, og har bidratt til kontroll og oversikt gjennom prosjektet.

Etter å ha testet den endelige applikasjonen på testgrupper, var resultatene som fremkom noe overraskende. Testgruppen som hadde fått forklaring på sitt estimat hadde noe mer tillit til beslutningen, men selve tilfredsheten økte ikke i takt med dette.

Innholdsfortegnelse

Figurliste

1 Introduksjon

Internet Explorers er en gruppe bestående av 5 studenter fra Universitetet i Agder som ble etablert høsten 2020. Fra tidligere hadde samtlige gruppemedlemmer gode bekjentskaper til hverandre, og flere hadde jobbet sammen i lignende prosjektoppgaver tidligere. Dette var en av hovedgrunnene til at deltakerne valgte å opprette en bachelorgruppe sammen.

Etter at gruppen ble etablert ble forventninger og ambisjonsnivå diskutert. Her ble det bestemt av samtlige medlemmer å ha ambisjonsnivå tilsvarende en toppkarakter. Høsten 2020 kom gruppen i kontakt med TietoEVRY og dannet senere et samarbeid med bedriften. Hovedgrunnen til at gruppen bestemte seg for å søke om et samarbeid med TietoEVRY var fordi de hadde et spennende og ambisiøst prosjektforslag, hvor gruppen kunne fordype seg i den fremtidsrettede teknologien *Explainable Artificial Intelligence,* også referert til som XAI. I tillegg til dette var gruppen også kjent med organisasjonens gode rykte blant IT-bedriftene i Norden (Charlesen, 2020), noe som gjorde det ekstra spennende å inngå et samarbeid med organisasjonen.

Gjennom dette semesteret har gruppen jobbet målrettet og kontinuerlig med bachelorprosjektet, og god planlegging og kommunikasjon har vært en nøkkelfaktor. Det som har vært spesielt spennende er TietoEVRY sin satsing på den relativt nye teknologien XAI. XAI baserer seg på grunnleggende AI, men skiller seg ut i den form at XAI også vil gi brukeren en forklaring på det valget som ble tatt. Vanligvis er brukeren uvitende om hva som skjer i bakgrunnen når en AI produserer et svar, og det kan derfor ofte være vanskelig å stole fullt og helt på de valgene som blir tatt. XAI sin hensikt er å gjøre svarene mer troverdig og menneskelignende, nettopp ved å gi sluttbrukeren en forklaring.

Prosjektet går ut på å lage en webapplikasjon som estimerer prisen på en bolig ved hjelp av kunstig intelligens. Brukeren av systemet skriver inn informasjon om sin bolig, og deretter vil det bli gitt et estimat basert på historisk salgsdata fra 2020/2021. Problemstillingen i prosjektet er å teste om brukere blir mer fornøyd dersom det blir gitt en forklaring til prisestimatet enn om det ikke blir gitt en forklaring. XAI er et svært sentralt tema ettersom artikkel 15(h) i Personopplysningsloven krever at brukere skal få innsyn i logikken bak automatiserte avgjørelser som er gjort med vedkommende sine personopplysninger (Personopplysningsloven, 2018). Resultatene som fremkommer fra prosjektet, kan da bli benyttet ved utviklingen av lignende systemer fremover.

2 Produktet

For å visualisere produktet har gruppen laget en video, hvor prosessen for en prisestimering blir gjennomgått samt andre funksjonaliteter i webapplikasjonen. Under ligger lenken til videoen:

- https://www.youtube.com/watch?v=c8-Yiq2uLfw&ab_channel=HeddaBefring

3 Sentrale avgjørelser og erfaringer

I denne seksjonen vil sentrale avgjørelser rundt blant annet teknologi, prosjektstyringsverktøy og kommunikasjonsplattform bli lagt frem og drøftet.

3.1 Vinkling av prosjektoppgave

Til å begynne med ønsket gruppen, etter veiledning og forslag fra TietoEVRY, å lage en webapplikasjon som tok for seg lånesøknadsprosessen i bank. Tanken var å gjøre lånesøknadsprosessen enklere ved å implementere XAI, slik at brukeren/kunden umiddelbart kunne få avslag eller aksept på lånesøknaden, samt en forklaring av utfallet. Denne oppgaven ble påbegynt i første sprint hvor gruppen startet med å undersøke hva som krevdes i en slik prosess. Her ble intervjuer foretatt med forskjellige banker, samt innad i TietoEVRY med ansatte som har erfaring fra lignende oppgaver. Etter intervjuene fikk gruppen sin første indikasjon på at dette ville vært vanskeligere å gjennomføre enn først antatt.

En av de største hindringene gruppen møtte på var anskaffelse av aktuelt og relevant datasett. Et godt datasett var essensielt i oppgaven ettersom dette er noe AI-en trenger for å læres opp og ta avgjørelser ut ifra (Algorithmia, 2020). Etter dialog med en ansatt i Financial Services Solutions i TietoEVRY ble det avtalt at de kunne ordne et datasett som kunne benyttes til opptreningen av AI-tjenesten. Etter hvert viste dette seg å ta lengre tid enn først antatt, da det var mye sensitiv informasjon som måtte maskeres. Dermed tok gruppen en avgjørelse sammen med produkteier om å vinkle på oppgaven, slik at den omhandlet prisestimering på boliger i stedet for lånesøknader.

3.2 Kunstig intelligens

Før utviklingssprintene startet hadde gruppen flere sprinter med planlegging, research og design. Her la gruppen blant annet opp til utforsking og diskusjoner rundt valg av AI-leverandør. Da hovedformålet med prosjektet var basert på analyse og tolkninger fra en AI-tjeneste, ble det bestemt at det skulle settes av litt ekstra tid til dette. Det var flere leverandører

som tilbød AI-tjenester som støttet forklaringselementet, hvor Google og IBM var blant de største aktørene. Tjenestene som ville være aktuelle for gruppen var Google AutoML og IBM Watson AutoAI.

Google AutoML er en Machine Learning as a Service (MLaaS) tjeneste fra Google som gjør det mulig å trene opp maskinlæringsmodeller av høy kvalitet, med begrenset eller ingen erfaring innen feltet. Tjenesten kan brukes til å lage maskinlæringsmodeller for strukturert data, bilder, videoer, dokumenter, etc. (Google Cloud, u.å.). I dette prosjektet var det strukturert data som skulle benyttes i opptreningen av modellen, og Google sin AutoML Tables var derfor en sterk kandidat ettersom den støtter nettopp denne typen data (Google Cloud, u.å).

IBM Watson AutoAI er en MLaaS tjeneste fra IBM som, i likhet med Google AutoML, skal automatisere blant annet modellutvikling og hyperparameter optimalisering (IBM, u.å). Denne tjenesten har mange av de samme mulighetene som Google sin tjeneste. En sammenligning utført av G2 viste at tjenestene scoret ganske likt med tanke på brukertilfredshet, men at Google sin tjeneste scoret noe høyere på brukervennlighet (G2, u.å.). Dette var veldig viktig for gruppen, ettersom ingen av medlemmene hadde vært borti denne typen tjenester tidligere.

Basert på den overnevnte informasjonen samt anbefalinger fra veiledere i TietoEVRY bestemte gruppen seg for å benytte Google AutoML tjenesten i prosjektet. Hovedgrunnene til dette var andre brukere sin erfaring med tjenestene og lite krav til forkunnskap rundt fagfeltet AI.

I etterkant er dette valget noe gruppen er fornøyd med, selv om opplæringen rundt bruken av tjenesten tok lenger tid enn forventet. Dokumentasjonen var noe abstrakt og det var få til ingen eksempler på hvordan Google AutoML Tables kunne implementeres sammen med en Flask webapplikasjon. Etter noen dager med utforsking, prøving, og feiling ble det omsider opprettet en sammenkobling. En grunn til at dette tok noe lengre tid enn planlagt var at teknologien var relativt ny og uberørt, som igjen førte til at det fantes få eksempler på nett som forklarte hvordan dette skulle gjøres.

3.3 Programmeringsspråk

Tidlig i planleggingsfasen diskuterte gruppen valg av programmeringsspråk. Gruppen rådførte seg med TietoEVRY og undersøkte hvilket programmeringsspråk som ville egne seg

best til utførelsen av prosjektet. Etter grundigere undersøkelser bestemte gruppen seg for å benytte Python som programmeringsspråk.

Python har hatt stor oppslutning de siste årene og er det programmeringsspråket som er i størst vekst for øyeblikket. Python anses å være oppe i toppen blant de mest brukte programmeringsspråkene sammen med Java og JavaScript (Eastwood, 2020). Ved siden av dette tilbyr Python også mange tilleggspakker som enkelt kan importeres fra nettet. Det brede tilbudet skyldes at mye av kolleksjonen og kodesamlingen er åpen kildekode, som betyr at den er åpen til disposisjon for offentligheten. Dette gjør den helhetlige opplevelsen godt egnet for uerfarne brukerne (Patel, 2018).

Via veilederne i TietoEVRY kom gruppen i kontakt med Vincent Aardalsbakke som på dette tidspunktet var Head of Automation and Security hos TietoEVRY. Vincent anbefalte gruppen å gå for Python da han mente at dette ville passe prosjektet godt ettersom det ofte ble benyttet i maskinlæringsprosjekter. En annen betydelig faktor for at gruppen valgte Python som programmeringsspråk var fordi Python støttet sammenkoblingen av Google AutoML Tables i Flask prosjekter. Kort sagt gjorde dette det mulig å importere Google sine AI-funksjoner til webapplikasjonen.

Det ble også sett på andre typer programmeringsspråk. Java var et annet alternativ og en god kandidat til Python da alle på gruppen tidligere hadde vært gjennom to emner med objektorientert programmering, samt et programmeringsprosjekt i tredje semester. Ulempen med å velge Java var at gruppen ville gå glipp av muligheten til å fordype seg i et nytt og spennende programmeringsspråk. Gruppen benyttet derfor denne anledningen til å lære Python.

Ifølge industrieksperter fra Springboard finnes det ikke et konkret svar på hvilke programmeringsspråk som passer best til maskinlæring da det er flere faktorer å ta hensyn til. Det nevnes derimot at noen språk er bedre egnet til maskinlæringsoppgaver enn andre. Flere maskinlæringsingeniører velger et maskinlæringsspråk basert på hvilket jobbproblem de har. Python egner seg godt til analyse, mens Java egner seg bedre til sikkerhet og trusseloppdagelse (Springboard, 2020).

3.4 Web-rammeverk

Da valget kom til hvilket web-rammeverk gruppen skulle bruke var dette noe som var relativt nytt for alle. Gruppen gikk dermed gjennom hvilke alternativer som var tilgjengelig

og vurderte deretter hvilket rammeverk som kunne passe best mtp. kunnskapsnivå og behov. Etter en del undersøkelse stod valget til slutt mellom to kjente koderammeverk for Python; Django og Flask.

Django hadde god dokumentasjon som gjorde det gunstig for nye brukere å skaffe seg en overordnet forståelse. I tillegg til dette hadde rammeverket mange aktive brukere som anbefalte det videre. Ulempen med Django var at det er ment for store og komplekse web applikasjoner (Mindfiresolutions, 2018), og kunne muligens ha blitt noe overveldende for gruppen.

Flask hadde flere av de positive egenskapene som Django hadde, men skilte seg fra Django ved at det var bedre egnet til mindre prosjekter og krevde mindre forkunnskap for å ta i bruk (Singh, 2021). Dokumentasjonen til Flask var også veldig ryddig og oversiktlig, og som nye brukere til rammeverket verdsatte gruppen nettopp dette.

Etter at begge web-rammeverkene var satt opp mot hverandre, valgte gruppen å gå for Flask. Denne avgjørelsen ble hovedsakelig tatt basert på dokumentasjonen og brukergrensesnittet til Flask. I ettertid er gruppen fornøyd med valget som ble tatt fordi web-rammeverket viste seg å være intuitivt og lite tidkrevende. Dette gjorde at gruppen kunne rette fokuset mot hovedessensen i prosjektet og ikke bruke mer tid enn nødvendig på opplæring av et web-rammeverk.

3.5 Prosjektstyringsverktøy

Før gruppen bestemte seg for hvilket verktøy som skulle brukes til prosjektstyring ble det diskutert innad i gruppen, og senere sammen med TietoEVRY, om de valgmulighetene som var til stede. Gruppen ble introdusert for Azure DevOps og Jira som gode alternativer til prosjektstyringsplattformer. Ett av gruppemedlemmene hadde tidligere jobbet med prosjektstyringsverktøyet Jira, og hadde gode erfaringer med dette. Jira er også verktøyet produkteierne i TietoEVRY bruker i mange av sine prosjekter. I tillegg hadde Jira gode muligheter når det gjaldt timeføring og oppdeling av oppgaver, som var essensielt da det ville gjøre det lettere å ha kontroll i prosjektet. Valget falt dermed naturlig på Jira.

I tillegg til å velge et verktøy for selve prosjektstyringen trengte gruppen også et versjonskontrollsystem for å sikre god standard for programvareutviklingsprosessen. Gruppen har brukt Github som hosting plattform for kode gjennom forskjellige fag i tidligere semestre. Dette verktøyet var derfor i utgangspunktet et naturlig valg helt til gruppen ble introdusert for

Bitbucket, en hosting plattform som allerede var integrert i Jira. I tillegg hadde også et av gruppemedlemmene jobbet med dette tidligere og kunne derfor gi resten av gruppen en introduksjon på hvordan dette fungerte. Med disse faktorene tatt i betraktning landet gruppen på Bitbucket og har ikke angret på valget siden.

3.6 Kommunikasjonsplattform

Det ble tidlig bestemt hvilken kommunikasjonsplattform gruppen ville ta i bruk. Dette ble gjort for å raskest mulig kunne sette fokuset over på selve prosjektet. Flere plattformer ble vurdert, og valget stod mellom Discord, Zoom og Microsoft Teams. Ettersom Microsoft Teams var den plattformen som ble brukt av TietoEVRY, var det naturlig at dette ble valget også for gruppen. Dette gjorde det lettere da daily standup, sprint review og generell kommunikasjon med bedriften også foregikk her.

Totalt sett er gruppen veldig fornøyd med valget som ble tatt da Microsoft Teams viste seg å være veldig intuitivt. Prosessen med å da lære seg å bruke Teams var veldig enkel. Teams ga også en oversiktlig mappestruktur til oppbevaring av viktige dokumenter og filer. Det var også viktig å ha et bra kommunikasjonsverktøy ettersom at store deler av prosjektet måtte gjennomføres hjemmefra grunnet covid-19 restriksjoner.

Etter å ha brukt Teams noen måneder ser gruppen at det er noen mangler med plattformen. Det er ikke mulig å åpne flere enn ett dokument når man er inne på applikasjonen. Dette bydde på enkelte problemer ettersom mange av dokumentene var lagret der. Utenom dette har Teams fungert utmerket som en kommunikasjonsplattform både innad i gruppen, men også sammen med TietoEVRY.

4 Prosjektstyring

Prosjektstyring er en viktig del i alle typer prosjekter og omhandler “applikasjonen av kunnskap, ferdigheter, verktøy og teknikker for å møte et mål eller et prosjektkrav” (Schwalbe, 2015). Ved hjelp av god prosjektstyring har gruppen holdt oversikt over oppgaver i backloggen, samt timeoversikt, noe som definitivt har vært viktig i bachelorprosjektet, og har sikret godt samarbeid og riktig ressursfordeling. Det ble derfor tatt en beslutning om å ha et eget kapittel om nettopp dette, som tar for seg viktige beslutninger rundt prosjektstyring.

4.1 Prosjektstyringsrammeverk

Det ble bestemt at gruppen skulle benytte Scrum. Dette var først og fremst etter ønske fra arbeidsgiver, samt at det har vært fokus på dette rammeverket gjennom hele studieløpet. TietoEVRY benytter Scrum i flere av sine utviklingsprosjekter, noe som gjorde at gruppen så på dette som en gyllen mulighet for å lære mer om hvordan dette rammeverket blir brukt i praksis i større organisasjoner.

Noen av de mest brukte agile rammeverkene for prosjektstyring er Scrum og Kanban (Lamelas, 2018). Gruppen så nærmere på hvilke av de to metodene som passet best til bachelorprosjektet. Kanban egner seg godt til prosjekter der det ikke er en stor backlog som utviklerne skal igjennom. Her er målet å komme seg fort gjennom oppgavene og planlegging er mindre sentralt (Tranter, 2016).

Scrum brukes av software-utviklings teams for å strukturere og styre arbeid gjennom en gitt periode (Drumond, u.å.). Når prosjektet har mål og milepæler vil Scrum være det beste valget. Rammeverket deles opp i forskjellige sprinter der man tar med lærdom og erfaringer inn i neste sprint (Tranter, 2016). Temaet XAI var relativt nytt for alle gruppemedlemmene, noe som gjorde at Scrum sin agile metodikk egnet seg bra med tanke på at mye kunne endre seg i løpet av prosjektet.

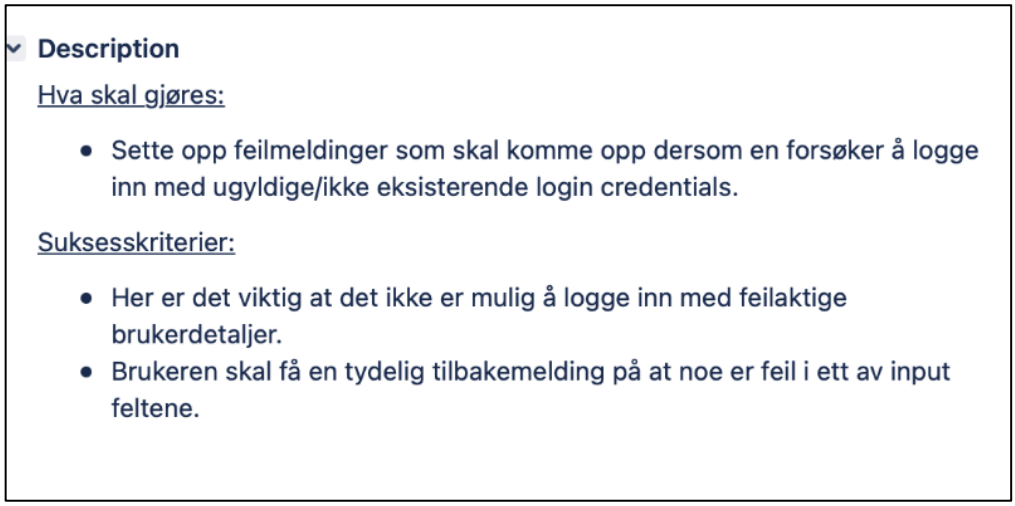
4.2 Sprinter

Et av hovedaspektene med Scrum er sprinter. Hele prosjektet blir delt opp i sprinter på maksimum én måned, og gjerne kortere. Hver sprint kan sees på som et lite prosjekt. Det skal leveres noe produkt av kvalitet for å sikre fremgang i prosjektet frem mot leveranse (Scrum.org, u.å.). Gruppen valgte å dele sprintene opp i 2 ukers intervaller, da det ikke ville bli for lang tid mellom sprintene, som ville ført til sjeldnere tilbakemeldinger fra produkteierne. Gruppen ville også unngå å ha kortere sprinter enn 2 uker, da status ikke ville ha endret seg nok mellom hver sprint til at gruppen kunne fått brukbare tilbakemeldinger fra produkteier.

4.3 Sprint planning

Før hver sprint ble det gjennomført et sprint planning møte. Her ble det planlagt hvilke oppgaver fra product backloggen som skulle være med inn i den kommende sprinten. Det var varierende hvor mange timer gruppen hadde tilgjengelig for hver sprint, grunnet forelesninger og andre forpliktelser. Derfor ble det satt opp et regneark hvor det ble estimert antall effektive

arbeidstimer tilgjengelig pr. dag i den følgende sprinten. Ved å gjøre dette fikk gruppen en bedre indikasjon på hvor mange oppgaver som skulle flyttes til sprint backloggen. Etter at dette var gjort måtte alle oppgavene estimeres basert på tidligere erfaringer og intuisjon, samt tilbakemeldinger fra veiledere. Hver oppgave fikk også egendefinerte suksesskriterier som kort skulle beskrive hva som måtte til for å tilfredsstille kravene for oppgaven. Eksempel på dette vises i figur 1.



Figur 1 - Illustrasjon på en oppgavebeskrivelse med suksesskriterier

Hvis en oppgave var mer omfattende enn ett dagsverk, ble det opprettet såkalte subtasks som er mindre underoppgaver knyttet til oppgaven. Disse underoppgavene gjorde det lettere og mer oversiktlig å loggføre og delegere arbeid til flere personer på samme oppgave. På denne måten sikret gruppen at oppgavene ikke ble for store og tidkrevende, og at hele gruppen var i stand til å forstå oppgavens innhold og fremgangsmåte. Gruppen prøvde å unngå estimeringer på over ett dagsverk, men det forekom noen avvik dersom flere medlemmer jobbet på samme oppgave. Ett dagsverk tilsvarte 5,5t for gruppen ettersom arbeidstiden var satt fra 08:30 – 14:30.

For å prioritere oppgaver i backloggen, ble Jira sin integrerte prioriteringsskala brukt. Denne formen for prioritering lignet veldig på MoSCoW metoden, som gruppen er kjent med fra studiet. Jira sin prioriteringsskala bestod av: *blocker, critical, major, medium, trivial* og *minor.* Ved å prioritere oppgavene vil det hjelpe alle (kunder, prosjektleder, designer og utviklere) å forstå de viktigste oppgavene, samt rekkefølgen de skal gjennomføres i (Haughey, 2021).

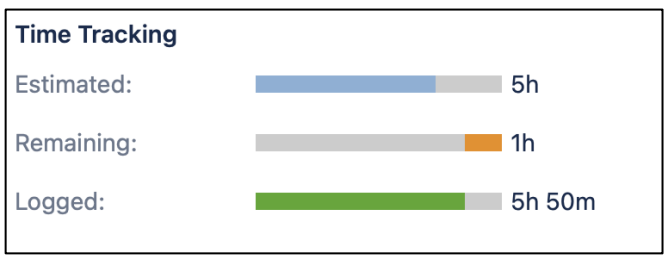
Icon and name	Description
Blocker	Blocks development and/or testing work, production could not run.
Critical	Crashes, loss of data, severe memory leak.
Major	Major loss of function.
Minor	Minor loss of function, or other problem where easy workaround is present.
Trivial	Cosmetic problem like misspelt words or misaligned text.
Medium	Created by JIM during import process

Figur 2 - Jira prioriteringsskala

Etter at gruppen hadde forberedt planen for neste sprint, ble det gjennomført et møte med produkteiere fra TietoEVRY. Gjennom møtet diskuterte produkteierne og utviklerne hvilke elementer og oppgaver som skulle inkluderes i kommende sprint (Scrum.org, u,å). Videre ble det forklart hva som var de viktigste oppgavene, samt hva som skulle forbedres til neste sprint.

4.4 Tidsestimering

En viktig del av prosjektet har vært tidsestimering. Tidsestimering er svært krevende, ettersom det er vanskelig å forutse hvor lang tid en oppgave vil ta. Det er spesielt krevende dersom den ansvarlige gjennomfører oppgaver som er ukjente og nye. Som tidligere nevnt brukte gruppen Jira som prosjektstyringsverktøy. En funksjon som ble hyppig brukt var "Remaining". Denne funksjonen ga en god oversikt over hvor mange timer som gjenstod av oppgaven. I tillegg til "Remaining" hadde Jira også et felt kalt "Logged" som viste hvor mange timer som hadde blitt logget på en oppgave til enhver tid. Feltet var meget nyttig ettersom det gjorde det lett å sammenligne estimatet med realisert arbeidstid. Differansen mellom disse tallene ble kontinuerlig benyttet for å forbedre tidsestimatene i kommende sprinter, som førte til mer nøyaktige estimater utover i prosjektløpet.



Figur 3 - Illustrasjon av tidsestimering, samt gjenværende tid på en gitt oppgave

For å vite hvor mange oppgaver som kunne være med i de forskjellige sprintene, beregnet gruppen tilgjengelig tid i den kommende sprinten. På en ideell arbeidsdag ville hvert

gruppemedlem ha 5.5 timer effektiv jobbing tilgjengelig. Noen dager ville det komme møter eller forelesninger i andre fag som gjorde at tilgjengelige timer ble noe redusert. Dette ble også tatt høyde for når tilgjengelige timer for sprinten ble satt opp.

Ved en aktiv bruk av "Remaining" funksjonen ble også tidsestimeringen mer og mer nøyaktig for hvert re-estimat, ettersom det ble mer tydelig hva som gjenstod på den gitte oppgaven. Dette gjorde at gruppen fikk en god oversikt over hvordan man lå an i forhold til sprintmålet.

4.5 Standup

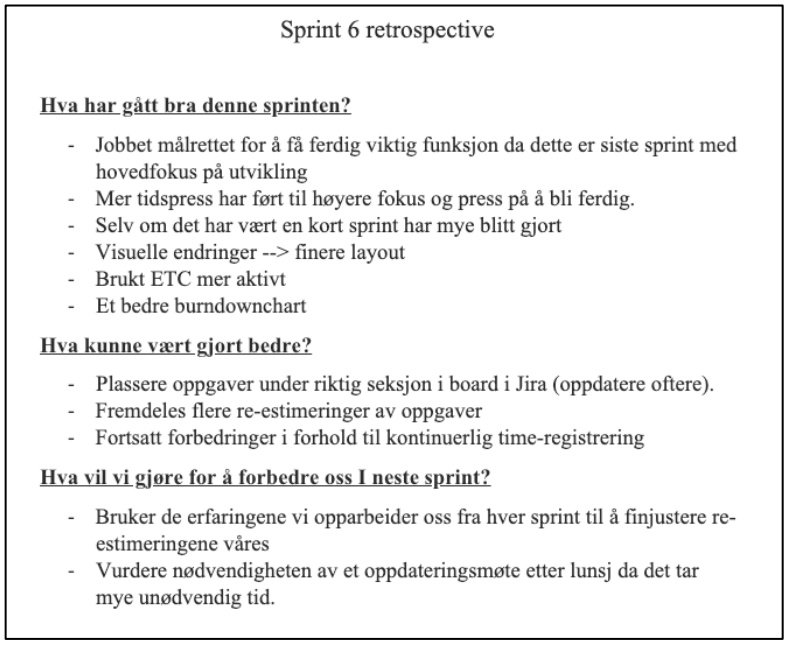
Gjennom hele bachelorprosjektet har gruppen gjennomført daily standup hver morgen klokken 08:30. Daily standup er et 15 minutters langt møte for utviklerne i et Scrum team hvor deltakerne skal diskutere fremgangen og hva som skal gjøres (Scrum,org, u.å.). Spørsmålene som ble besvart i gruppen sitt tilfelle var; *Hva gjorde du i går? Hva skal du gjøre i dag?* Og, *har du noen utfordringer?* Daily standup har vært nyttig for å holde orden på fremgangen i prosjektet, ettersom gruppemedlemmene hver dag var oppdatert på hvilke oppgaver som var i fokus, og hvem som jobbet med hva. Utfordringer knytet til arbeidsoppgaver ble også adressert, noe som førte til at problemer og utfordringer raskt ble tatt tak i for å forhindre usikkerhet over lenger tid.

Figur 4 – Utdrag fra daily standup.

I tillegg til muntlig gjennomgang ble daily standup skrevet ned og sendt til veiledere i TietoEVRY i en egen Microsoft Teams kanal. På denne måten hadde produkteiere mulighet til å følge med på hvordan arbeidet utfoldet seg gjennom hele prosjektet, og ikke bare på sprint-møte hver andre uke.

4.6 Sprint retrospective og sprint review

Sprint retrospective ble avholdt den siste dagen i hver sprint. Hensikten med sprint retrospective er å planlegge hvordan gruppen kan forbedre kvaliteten og effektiviteten i arbeidet (Scrum.org, u.å). Her gikk gruppen gjennom hva som hadde blitt gjort i den aktuelle sprinten, og adresserte følgende punkter; *Hva har gått bra denne sprinten? Hva kunne gått bedre? Hva kan vi gjøre for å forbedre oss i neste sprint?* (Scrum.org, u.å.). Sprint retrospective gjorde det lettere å holde fokus og være observant på hvilke tiltak som skulle til for å forbedre arbeidet. Tingene som ble gjennomgått i gruppens sprint retrospective ble notert ned i egne dokumenter og deretter presentert for produkteierne på sprint review som ble gjennomført den påfølgende dagen.



Sprint 6 retrospective

Hva har gått bra denne sprinten?

- Jobbet målrettet for å få ferdig viktig funksjon da dette er siste sprint med hovedfokus på utvikling
- Mer tidspress har ført til høyere fokus og press på å bli ferdig.
- Selv om det har vært en kort sprint har mye blitt gjort
- Visuelle endringer --> finere layout
- Brukt ETC mer aktivt
- Et bedre burndownchart

Hva kunne vært gjort bedre?

- Plassere oppgaver under riktig seksjon i board i Jira (oppdatere oftere).
- Fremdeles flere re-estimeringer av oppgaver
- Fortsatt forbedringer i forhold til kontinuerlig time-registrering

Hva vil vi gjøre for å forbedre oss I neste sprint?

- Bruker de erfaringene vi opparbeider oss fra hver sprint til å finjustere re-estimeringene våres
- Vurdere nødvendigheten av et oppdateringsmøte etter lunsj da det tar mye unødvendig tid.

Figur 5 - Sprint retrospective eksempel

Etter hver sprint ble det avholdt et sprint review møte med produkteiere samt resterende veiledere i TietoEVRY. En sprint review skal vare i maks fire timer dersom en sprint varer i en måned, men for kortere sprinter vil tiden bli redusert (Scrum.org, u.å,). I dette prosjektet der sprintene varte i to uker, var 45 minutter satt av til hver sprint review. I disse møtene gikk gruppen gjennom hva som hadde blitt gjort den aktuelle sprinten, adresserte eventuelle problemer eller utfordringer, og fikk tilbakemeldinger fra produkteierne. I tillegg ble det gjort justeringer på product backloggen dersom dette var nødvendig, samt diskutert i hvilken grad tidsestimeringen tilsvarte den faktiske arbeidstiden. Sprint reviews hver andre

uke førte til hyppige tilbakemeldinger fra produkteier hvor det ble gitt mange gode innspill for hvordan neste sprint skulle gjennomføres.

4.7 Rollefordeling

I gruppearbeid er det naturlig at hvert enkelt individ har forskjellige egenskaper. Dermed er det en fordel å kjenne hverandres styrker og svakheter. Innad i gruppen er det forskjellig kunnskapsnivå relatert til forskjellige aspekter ved prosjektet. På bakgrunn av dette ble det gjort en rollefordeling under planleggingsfasen slik at tilretteleggingen av ansvar og arbeidsoppgaver ble tilpasset hvert enkelt gruppemedlem på best mulig måte.

4.7.1 Scrum Product Owner

Produkteier i dette bachelorprosjektet har vært Roar Engen og Elisabeth Cederberg Øvensen fra TietoEVRY avdeling Kristiansand. De har ledet an prosjektet og hatt ansvaret for innkalling til sprint møter samt generell oppfølging. I tillegg har gruppen også fått tildelt noen av produkteier oppgavene, som for eksempel prioritering.

4.7.2 Scrum master

Hovedansvaret til Scrum Masteren er å sørge for at alle gruppemedlemmer er innforstått med Scrum teori og praksis. I tillegg til dette er Scrum Master ansvarlig for effektivitet innad i gruppen (Scrum.org, u.å.). Scrum Masteren har hatt det overordnede ansvaret med å gjennomføre daily standup, sprint planning, sprint reviews, og statusoppdateringer. Kommunikasjonen med produkteier har også stort sett gått gjennom Scrum Master. Alle på gruppen hadde et relativt likt erfaringsgrunnlag, så Scrum Master tittelen ble tildelt til Hedda Befring basert på eget ønske.

4.7.3 Git master

Arbeidsoppgavene til Git-masteren bestod av å holde orden og skaffe seg ekspertise på det som gikk på remote oppdatering og endring av mappe- og filstruktur i prosjektet. Da et av gruppemedlemmene hadde mer erfaring når det kom til versjonskontrollsystemet Git, ble ansvaret gitt til Sander Vereide.

4.7.4 Frontend

Frontend-laget bestod av Jacob Mario Hjermann, Henrik Gulliksen, Hedda Befring og Sander Vereide. Dette laget hadde ansvaret for den helhetlige brukeropplevelsen og utseende til webapplikasjonen. Flere av gruppemedlemmene som jobbet med frontend hadde også jobbet med alt fra sketches til prototype tidligere i prosjektet. Dette så gruppen på som en

fordel da disse medlemmene hadde best forståelse av hvordan det endelige utseende til webapplikasjonen skulle bli.

4.7.5 Backend

Backend-laget bestod hovedsakelig av Sander Kvale Garborg og Sander Vereide, men Henrik Gulliksen og Jacob Mario Hjermann var også med på enkelte av oppgavene. Sander Kvale Garborg hadde tidligere erfaring med Python som gjorde det naturlig at han tok styringen her.

4.8 Risikoregister

En viktig del av et prosjekt er å ha oversikt over potensielle risikoer som kan oppstå. Dette gjelder både risikoer knyttet til det tekniske aspekter, og personlige hendelser. For å komme i gang med risikoanalysen fikk gruppen en mal som TietoEVRY bruker i sine prosjekter. Dette var et godt utgangspunkt, der hver risiko skulle inneholde en beskrivelse av konsekvens, type tiltak, resultat av tiltak, ansvarlig for tiltak og status. Ved å ha en detaljert beskrivelse, og en hovedansvarlig for hver risiko var det enkelt å ha kontroll på de forskjellige risikoene som måtte tas hensyn til. Ifølge Digitaliseringsdirektoratet inneholder en risikovurdering tre steg:

1. **Risikoidentifisering**

Risikoidentifisering er det første steget i en risikovurdering. Her blir det laget en liste med alle mulige potensielle risikoer (Digitaliseringsdirektoratet, u.å.). Gruppen satt sammen og diskuterte forskjellige risikoer som kunne oppstå. Listen ble delt opp i tre deler der den første delen inneholdt risikoer knyttet til teknologien som ble brukt, den andre delen inneholder menneskelige risikoer, og den siste delen inneholder andre risikoer knyttet til blant annet TietoEVRY som bedrift og veilederne.

2. **Risikoanalyse**

Det andre steget i en risikovurdering er å foreta en risikoanalyse. Her fastslås den mulige konsekvensen og tilhørende sannsynlighet (Digitaliseringsdirektoratet, u.å.). Etter å ha gjennomført en risikoanalyse vil det være tydelig hvilke risikonivå de forskjellige risikoene befinner seg på. For å vurdere sannsynlighet ble det brukt en skala fra 1- 4, der 1 = Usannsynlig, 2 = Mindre sannsynlig, 3 = Mulig og 4 = Sannsynlig. For å vurdere konsekvens av risikoen ble det også brukt en skala fra 1- 4 der 1 = Ubetydelig, 2 = Moderat, 3 = Alvorlig, 4 = Kritisk. Konsekvensen ble vurdert ut ifra hvordan risikoen ville skapt forsinkelse, samt

påvirket kvaliteten av prosjektet. Vanligvis ville også kostnader spilt en rolle i hvor alvorlig konsekvensene er, men med tanke på at det ikke er noen kostnader involvert i dette prosjektet ble denne faktoren ikke vurdert. Figur 6 viser et utdrag fra risikoanalysen. Denne inneholder flere rader og kolonner, men av hensyn til lesbarhet er den fullstendige risikoanalysen lagt til i vedlegg 7.

#	Beskrivelse av risiko	Kommentar og konsekvens	S	K	R
	Teknisk risiko				
1	Teknisk svikt av fysisk utstyr	Svikt av teknisk utstyr kan føre til at én eller flere pc'er blir satt ut av spill, og vi vil dermed miste arbeidskraft frem til dette blir fikset.	2	2	4
2	Mister tilgang til Google AutoML Tables (Beta)	Ettersom Google AutoML Tables fortsatt er i Beta fasen vil en risiko være at vi mister tilgang til tjenesten underveis i prosjektet. Dette vil kunne føre til et stort set-back i prosjektet vårt da vi eventuelt må gå over til en ny tjeneste og endre på store deler av koden.	2	2	4
3	Mister tilgang til filer i Microsoft Teams	Dersom serverene til Microsoft skulle krasje vil vi kunne risikere å miste tilgang til essensielle filer og dokumenter. Dette vil i verste fall føre til permanent tap av verdifull data.	1	3	3

Figur 6 - Utdrag fra risikoanalysen

3. Risikoevaluering

Det tredje og siste steget er risikoevaluering. Før hvert sprint-møte med TietoEVRY hadde gruppen en gjennomgang av risikovurderingen. Her ble hver enkelt risiko gjennomgått, og sannsynligheten og/eller konsekvensen ble justert dersom risikobildet hadde endret seg siden forrige møte. På sprint-møte ble de eventuelle endringene presentert for veilederne og dersom det var noen problemer ble disse adressert. En risikomatrise (se vedlegg 8) ble opprettet basert på sannsynligheten og konsekvensen av hvert risikoelement. Risikomatrisen bestod av en tabell for gjeldende risikobilde, og en tabell som illustrerte det initielle risikobildet. På denne måten hadde gruppen oversikt over hvilke risikoer som hadde endret seg siden starten av prosjektet.

5 Kvalitet og kvalitetssikring

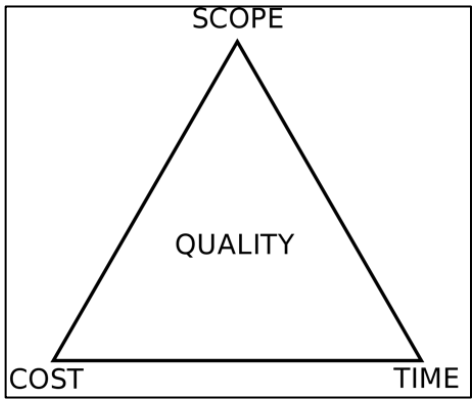
Kvalitet og kvalitetssikring er noe som har blitt høyt prioritert gjennom hele prosjektet. Det er vanskelig å definere kvalitet, men på generell basis kan kvalitet innen programvareutvikling defineres som; å lage noe som gjør det kunden ønsker, slik at programmet faktisk blir benyttet (Cohn, 2014). Det handler altså ikke bare utelukkende om å få bort "bugs" i koden, men også om den helhetlige opplevelsen til brukeren. Det har ingen hensikt å lage et feilfritt program dersom det ikke gjør det brukeren ønsker. Kvalitetssikring i gruppens prosjekt har dermed bestått av flere ulike aktiviteter som blant annet par-programmering, kodegjennomganger, Scrum, testing og valg av teknologier.

5.1 Kvalitet i gruppens prosjekt

Som nevnt over var det et tydelig fokus på å etablere klare rammer og rutiner for å sikre god kvalitet i prosjektet. Det ble blant annet opprettet maler og dokumenter som ga gruppen en felles forståelse for hvordan kvaliteten i prosjektet skulle sikres (se vedlegg 3-6). Det viktigste for gruppen var at alle var på samme bølgelengde og hadde like forventninger til hvordan kvaliteten skulle være. Det ble gitt en veldig åpen oppgave fra TietoEVRY, noe som medførte at ansvaret hovedsakelig lå på gruppen når det kom til å definere selve oppgaven samt kvaliteten i denne. Kvaliteten i dette prosjektet har en klar sammenheng med systemets enkle og intuitive brukeropplevelse. En "ren" og minimalistisk webapplikasjon uten unødvendige ekstrafunksjonaliteter tilrettela for en enklere læringskurve blant nye brukere. Dette var viktig ettersom systemet ikke nødvendigvis ville bli brukt kontinuerlig eller over en lengre periode, og at brukerne skulle unngå å bruke unødvendig tid på å sette seg inn i systemet for hver gang det skulle benyttes. Samtidig var det viktig for kvaliteten at estimatene ga en realistisk gjenspeiling av boligmarkedet, ettersom dette var et av de mest sentrale aspektene i hele webapplikasjonen.

Ifølge den vitenskapelige utgiveren Emerald Insight er den såkalte "Iron Triangle" eller "Project Management Triangle" et fundamentalt aspekt med hvordan en skal forstå suksess i prosjekter. Triangelet er en representasjon av hvordan de grunnleggende kriteriene i suksess blir målt i henhold til; om prosjektet blir levert innen tidsfristen, innenfor budsjettet, og til avtalt nivå av kvalitet eller ytelse (Pollack, Helm & Adler, 2018). Som figur 7 viser, kan tre faktorer påvirke kvaliteten til et produkt. En av faktorene som har påvirket kvaliteten i dette prosjektet er tidsfristen. I dette bachelorprosjektet har gruppen hatt 4-5 måneder på å utvikle et produkt. Videre har målet vært å oppnå et ferdig produkt med fungerende funksjonalitet.

Innenfor bachelorprosjektets tidsramme har gruppen klart oppnå målet med en webapplikasjon som estimerer boligverdi basert på boligegenskaper ved bruk av XAI. Det ble ikke satt noen spesifikke krav til ytelse fra oppdragsgiver utenom at webapplikasjonen skulle håndtere estimeringene til begge testgrupper, og gjerne håndtere flere estimeringer samtidig uten å krasje. Når det kommer til kostnader knyttet til prosjektets kvalitet, så har de naturligvis vært tilnærmet lik 0,-. Grunnen til dette er fordi gruppen ikke har hatt noen utgifter, da dette har vært et bachelorprosjekt i frivillig samarbeid med TietoEVRY.



Figur 7 - Project Management Triangle (Wikipedia, 2021)

5.2 Scrum

Som tidligere nevnt er prosjektstyringsrammeverket Scrum benyttet i prosjektet. Dette var i stor grad et kvalitetssikringstiltak som sikret fremgang og gjennomsiktighet i prosjektet. Dette gjorde det mulig for gruppen å være omstillingsdyktige gjennom hele prosjektet slik at sluttproduktet ble som ønsket. I tillegg til dette var det flere andre elementer i Scrum som sikret kvalitet i prosjektet, deriblant retrospective, daily standup, burndown chart og tidsestimering for å nevne noen. Retrospective gjorde at gruppen kunne reflektere over egen innsats i sprinten som hadde vært, og dermed lære av dette og kontinuerlig forbedre seg i de påfølgende sprintene. Dette førte til at den gjennomgående kvaliteten i prosjektet økte gjennom hele prosjektperioden. Daily standup, burndown chart og tidsestimering, inkludert ETC, gjorde det enklere for gruppen å holde kontroll i eget prosjekt, og dermed levere i henhold til tid, som er et av punktene i "Project Management Triangle" som illustrert i figur 7.

5.3 Kodestandard

Som et kvalitetssikringstiltak bestemte gruppen seg tidlig for å benytte seg av en satt kodestandard for all kode som ble skrevet i utviklingsfasen. Grunnen til dette var for å sikre at all koden i prosjektet skulle være konsistent og følge best practice. Dette førte til at koden var

mer sammenhengende, og gjorde det dermed mindre synlig at programmet ble laget av flere forskjellige utviklere.

For Python koden bestemte gruppen seg for å benytte PEP-8 standarden. PEP-8 er en stilguide som inneholder kodekonvensjoner for koden som ligger til grunn for standardbiblioteket i Python (Van Rossum, Warsaw & Coghlan, 2013). Ettersom PEP-8 stilguiden er noe omfattende, ble det bestemt at det skulle settes opp en sjekkliste som inneholdt de mest sentrale punktene fra standarden (se vedlegg 3). Denne sjekklisten ville også ligge til grunn for kodegjennomganger før kode ble pushet til master.

Sjekkliste - PEP8	***Kryss av***
IKKE mellomrom inntil parenteser, kolon, semikolon og komma	
Bruker kun doble hermetegn	
Dersom "=" brukes ved keyword argument, så skal det ikke være mellomrom før og etter. For eksempel: t=test ✅ t = test ❌	
Ved exceptions er det brukt spesifikke exceptions, og ikke bare en *except* klausul	
Naming conventions	
Klasser har alltid CapWords convention.	
Variabler og klasser har liten forbokstav (evt skilles med _)	
Dunders skal alltid plasseres etter docstring og før import statements.	
Konstanter er navngitt med ALL CAPITAL bokstaver, hvor flere ord er separert med _	

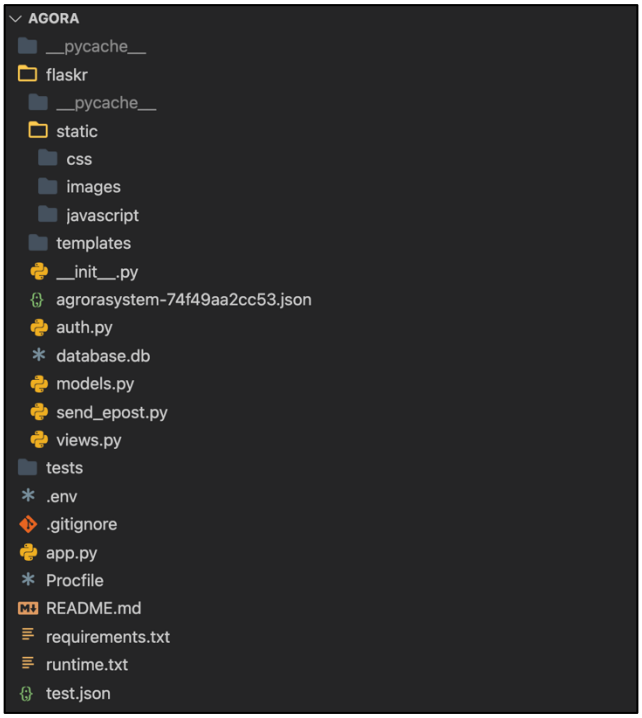
Figur 8 - Utdrag kodestandard - Python

Gruppen bestemte også at kodestandarder for HTML og CSS koden ville være hensiktsmessig for å sikre kvalitet og bedre samkjøring mellom medlemmene. Derfor ble det også laget sjekklister med standarder for denne koden (se vedlegg 4 og 5). Grunnlaget for disse sjekklistene var Google sin egen kodestandard samt noen elementer fra W3Schools.com. Google sin standard definerer formattering og stilguider for både HTML og CSS, og har som mål å bedre blant annet samarbeid og kvalitet (Google Developers, u.å). På lik linje med Python sjekklisten lå disse også til grunn ved kodegjennomganger, som blir forklart nærmere i kapittel 5.7.

5.4 Coupling og Cohesion

Ifølge en artikkel fra blogg- og nettpubliseringsplattformen Medium, er nøkkelen til en vedlikeholdbar og oversiktlig kode basert på prinsippene om lav coupling og høy cohesion (Montiel, 2018). Dette kan utdypes med at lav coupling refererer til selvstendige moduler som ikke avhenger av eller påvirker andre moduler (Montiel, 2018). Med høy cohesion menes det at elementer innenfor en modul er plassert sammen fordi de hører sammen. Relatert kode skal være nært hverandre for å gi koden et høyt samhold og forhindre kodeduplikasjon mellom moduler (Montiel, 2018).

I utviklingssprintene hadde gruppen fokus på prinsippene nevnt over for å unngå uoversiktlig kode. Figur 9 viser et utdrag fra det integrerte utviklingsmiljøet (IDE) Visual Studio Code, hvor gruppen har modularisert kode i forskjellige mapper. De gråfargede mappene representerer uåpnede mapper, mens de gule mappene viser mapper som er åpnet. Hver enkelt mappe inneholder også tilhørende relevant kode. Det var viktig for gruppen at lav coupling ble benyttet, noe som ble gjort ved at hver funksjonalitet i prosjektet ble delt opp i egne funksjoner. For eksempel, så inneholder *send_epost.py* en funksjon som kun har som oppgave å sende en e-post til brukeren. Denne blir kun kalt på der den trengs, og dersom det forekommer en endring i funksjonen, vil ikke dette påvirke andre deler av programmet. Dette ble gjort for å gjøre det lettere for TietoEVRY å videreutvikle prosjektet senere.



Figur 9 - Mappestruktur VS Code

5.5 GIT veiledning

Ettersom det var noe varierende kunnskapsnivå blant gruppemedlemmene når det kom til versjonskontrollsystemet Git, bestemte gruppen seg for å utarbeide en Git veiledning som inneholdt steg-for-steg instruksjoner for grunnleggende Git kommandoer. Det som ble inkludert i denne veiledningen var hovedsakelig de absolutt nødvendige kommandoene brukt til kloning, henting og oppdatering av kode. Dette var et sentralt kvalitetssikringstiltak ettersom det hindret brukerfeil ved oppdatering av koden, som kunne ført til forsinkelser i prosjektet.

5.6 Testing

For å sikre god kvalitet i prosjektet etablerte gruppen klare rammer for testing av funksjonalitet samt kjøring av kode. Ettersom testing var et tema med lite fokus på UiA ble gruppen selv nødt til å utforske dette temaet dypere. Som et kritisk kvalitetssikringstiltak ble det bestemt å ha spesielt fokus på testing, både manuell brukertesting og automatiserte tester av koden. Dette ble bestemt tidlig i prosjektet da oppdragsgiver ga uttrykk for at testing var noe de i stor grad brukte som et kvalitetssikringstiltak.

I en av de første sprintene benyttet gruppen seg av sine kontakter i TietoEVRY og ble henvist til en erfaren tester innad i organisasjonen. Her fikk gruppen en rask introduksjon til testing i Python og fikk gode tips til kjente test-rammeverk og testmetoder som gruppen videre kunne utforske på egenhånd. Fra kurset fikk medlemmene et godt grunnlag for å sette opp en egen plan for testing (se vedlegg 6). Denne planen gir en god oversikt over hvordan en skal tilrettelegge for Pytest rammeverket i utviklingsmiljøet.

5.6.1 Manuelle tester

For å sikre at systemet fungerte som det skulle ble det gjennomført manuelle tester. Disse ble gjennomført for å finne potensielle feil og problemer i koden (Guru99, u.å.). Ettersom gruppen hadde lite erfaring med unit-testing, var manuelle tester et godt sted å starte. For å gjennomføre disse testene ble det opprettet test-caser med en beskrivelse av testen, fremgangsmåte for å utføre testen, samt et forventet resultat. Dersom systemet oppførte seg som det skulle, kunne testeren bekrefte dette. Hvis testeren oppdaget feil, eller fikk et annet resultat enn det som var forventet, ble dette formidlet til utviklerne og feilen ble rettet opp i.

Manuelle tester egnet seg godt til å teste funksjoner som innlogging, registrering og estimering da det var viktig at skjemaene kun responderte dersom feltene var fylt inn riktig. Ettersom at dette er en vanlig måte å gjennomføre manuelle tester på, var det mye informasjon tilgjengelig med forslag til test-caser relatert til for eksempel innlogging og registrering.

Brukerregistrering						
Opprettet av: Hedda Befring						
Dato: 17.03.21						
Beskrivelse av testen	Test ID	Test nr	Beskrivelse	Utfør test	Forventet resultat	
En bruker skal kunne registrere seg i systemet	Brukerregistrering_1	1	Det skal ikke være mulig å registrere seg dersom ikke alle feltene er fylt inn	Trykk på "Registrer" uten å ha skrevet noe inn i feltene	Det skal komme feilmelding som forteller at du må fylle ut de gjeldene feltene	
	Brukerregistrering_2	2	Det skal være mulig å registrere seg	Fyll inn alle nødvendige felter og trykk på registrer	En profil skal bli opprettet og du skal få en melding som indikerer dette.	
	Brukerregistrering_3	3	Felt for epost skal inneholde vanlig format for epost (@gmail.com/ @hotmail.com osv)	Registrer deg med følgende eposter: eksempelAtgmail.com eksempel.gmail.com eksempel@gmail eksempel@gmailcom	Brukeren vil ikke bli registrert, og en melding om feil epost-format skal vises	

Figur 10 - Eksempel på en test-case relatert til brukerregistrering

5.6.2 Automatiserte tester

Som et annet viktig kvalitetssikringstiltak ble det, i tillegg til de manuelle testene, satt opp flere automatiserte tester ved hjelp av Pytest. Pytest er et testrammeverk for Python hvor små og enkle tester kan bli gjennomført, samtidig som det er skalerbart til mer komplekse tester (Pytest, u.å.). Det er et av de mest brukte testrammeverkene i Python (Lyubinsky, 2020), og i tillegg er det dette rammeverket som ligger til grunn for testing i Flask rammeverket (Flask, u.å.). Dette var altså hovedgrunnene til at Pytest rammeverket ble benyttet i akkurat dette prosjektet.

Testene som ble opprettet ved hjelp av Pytest omhandlet i hovedsak "routing" mellom sider i programmet. Disse testene ble kjørt hver gang før kode ble pushet for å forsikre at hele programmet lastet inn som planlagt, dvs. at alle sidene førte til der de skulle samt inneholdt den informasjonen som var forventet.

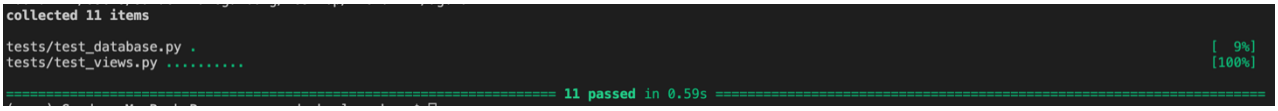
```python
# Tester at index siden laster som normalt
def test_index_route(client):
    response = client.get('/')
    assert b"Velkommen!" in response.data
    assert response.status_code == 200

# Tester at om oss siden laster som normalt
def test_omoss_route(client):
    response = client.get('/omoss')
    assert b'Om Oss' in response.data
    assert response.status_code == 200

# Tester at kalkuler siden laster som normalt
def test_kalk_route(client):
    response = client.get('/kalkuler')
    assert b"Redirecting" in response.data
    assert response.status_code == 302
```

Figur 11 - Utdrag fra tester i Pytest

Alle testene lå i egne mapper som raskt kunne kjøres ved å skrive inn kommandoen *pytest* i utviklingsmiljøets terminal. Dersom alle testene kjørte feilfritt ble dette vist i terminalen som illustrert i figur 12. Dersom det derimot forelå noen feil i koden var tilbakemeldingen noe mer omfattende, som vist i figur 13, hvor brukeren blir opplyst om hvilken test som feiler samt hva feilen er.



Figur 12 - Tilbakemeldinger ved suksessfulle tester

Figur 13 - Tilbakemelding ved feil i test

Disse testene var et svært sentralt kvalitetssikringsaspekt i prosjektet ettersom det hindret at "ødelagt" kode ble pushet til ulike brancher, deriblant masterbranchen, som potensielt kunne ha skapt store forsinkelser i prosjektets tidslinje.

5.7 Kodegjennomgang

Et annet sentralt kvalitetssikringstiltak som har blitt benyttet kontinuerlig gjennom prosjektet var kodegjennomganger ved oppdatering eller endringer i koden. Da endringer og oppdateringer fant sted i koden, benyttet gruppen seg av kildekodelagringstjenesten Bitbucket for å pushe lokale endringer opp til remote. Bruken av versjonskontroll sikret god kvalitet ved å spore hver enkelt endring i tillegg til å fange opp eventuelle konflikter (Atlassian, u.å.). Alle oppdateringer gikk gjennom en såkalt "pull request". En pull request er ifølge GitKraken en forespørsel til andre kodesjekkere om å se over endringer i en branch før den blir merget (flettet) inn i en annen (GitKraken, u.å). Før en pull request fant sted var det helt essensielt at webapplikasjonen kjørte feilfritt lokalt. For å teste ut dette startet man webapplikasjonen, kjørte gjennom de automatiserte testene, og sjekket at ingen feilmeldinger dukket opp. Rammeverket Flask og Pytest ga tydelig beskjed under kjøring hvis noe var feil i koden.

Når en pull request ble opprettet, ble minimum to andre gruppemedlemmer som ikke hadde jobbet med den oppdaterte koden lagt til som reviewers. Dette betydde ikke at kodesjekkerne nødvendigvis måtte forstå alle kodeendringene, men de skulle sørge for å sjekke at koden samsvarte med sjekklistene som nevnt i kapittel 5.2. Hvis en kodeundersøker fant ut at den nye koden ikke holdt standarden, ble pull requesten satt til "ikke godkjent" med en beskrivende begrunnelse, slik at en ny og oppdatert pull request kunne bli opprettet. Takket være gode sjekklister og tydelige rammeverk innad i gruppen hendte det at pull requests ble avslått. Dette ser gruppen tilbake på som et godt tegn, fordi det viser at kvalitetssikringen hindret at mindre eller større kodefeil ble flettet inn i masterbranchen.

6 Prosjektgjennomføring

I denne delen vil gjennomføringen av prosjektet bli presentert. Her vil gruppen drøfte rundt arbeidet i sin helhet, problemer som har oppstått underveis og hvordan disse har blitt håndtert og behandlet. Gjennom hele prosessen har gruppen tatt i bruk agil arbeidsmetodikk noe som gjenspeiles gjennom strukturen på rapporten. Hver enkelt sprint vil også bli gjennomgått.

6.1 Analyse

Analyse er den prosessen der man setter sammen kunnskapen man har på et område basert på undersøkelser og innsamlet materiale (UiO, u.å.). I dette prosjektet har gruppen gjort visse valg for å forsikre seg om at prosjektet står til forventingene, både for produkteierne og brukerne. Det var derfor viktig å hente informasjon fra ulike aspekter og perspektiver. I startfasen av informasjonsinnhentingen ble det utført intervjuer fra bransjeansatte som kunne gi en pekepinn på hvilken retning applikasjon burde gå i. I tillegg ble det også laget personas og brukerhistorier for å gi god innsikt i hvordan målgruppen vil at applikasjonen skal fungere.

6.1.1 Målgruppe

En målgruppe representerer den type mennesker du tror har lyst til å kjøpe produktet ditt (Torgersen, 2018). Etter at det ble besluttet å vinkle problemstillingen ble målgruppen en annen enn først planlagt, og i dette tilfellet er målgruppen de brukerne som kommer til å bruke systemet, og ha lyst til å estimere pris på en bolig. Ved hjelp av intervjuer og samtaler med produkteier ble det bestemt at en passende målgruppe ville være "mannen i gata". Hensikten med boligestimeringen var at hvem som helst kunne estimere prisen på egen bolig.

Dette vil være et lavterskeltilbud slik at huseiere som kun er nysgjerrig på verdi av egen bolig kan finne ut av dette uten å måtte kontakte en megler. Etter at gruppen bestemte at dette ville være en passende målgruppe, ble det opprettet brukerhistorier og personas som representerer en tenkt bruker.

6.1.2 Intervjuer

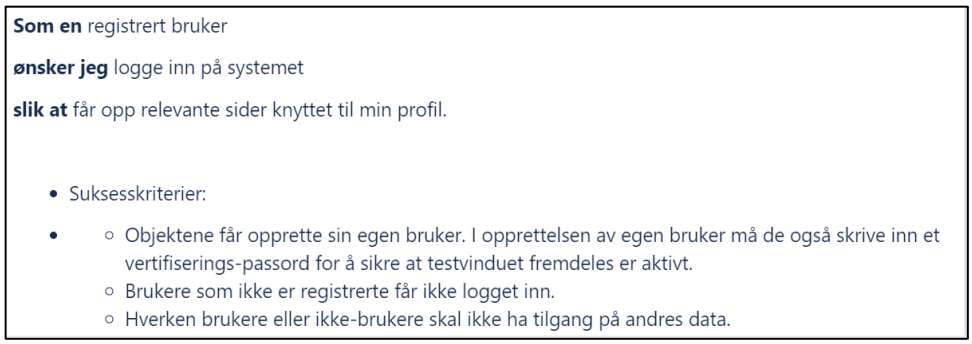
Tidlig i prosjektet ble det gjennomført intervjuer relatert til låneprosessen i bank. Da gruppen skulle samle inn informasjon til dette, ble det foretatt flere intervjuer med både bankansatte og kunder/brukere i en lånesøknadsprosess. Ved å intervjue begge parter av prosessen fikk gruppen et oversiktlig og grundig informasjonsgrunnlag som igjen kunne brukes til å lage produktet. Intervjuene viste seg å være effektive som informasjonsgrunnlag hovedsakelig grunnet diskusjonene som kom som et resultat av den forhåndsdefinerte intervjumalen gruppen hadde opprettet. Gjennom samtalene fikk gruppen fyldige og velformulerte svar som ble analysert i etterkant.

Intervjuene som er foretatt av gruppen er et godt eksempel på kvalitativ metode. Her gikk gruppen for "det åpne individuelle intervjuet". Dette er en datainnsamlingsmetode som kjennetegnes av en samtale mellom undersøker og respondent (Jacobsen, 2015, s.146). Under intervjuene ble det tatt notater som senere ble korrekturlest og skrevet om til referater.

Da problemstillingen ble vinklet falt hensikten med de allerede gjennomførte intervjuene bort. Det ble da besluttet at det var nødvendig å foreta nye intervjuer. På grunn av knapp tid ble det kun gjennomført ett intervju relatert til den nye vinklingen. Dette var med en eiendomsmegler fra DNB. Formålet med intervjuet var å få et innblikk i hvordan taksering foregår på boligmarkedet i dag, og å få et innblikk i viktige aspekter relatert til prissetting. Gjennom intervjuet (se vedlegg 9) fikk gruppen innsikt i hvilke faktorer som påvirket prisen i størst grad og hva slags systemer for dette som allerede finnes. Informanten uttalte også at ideen til prosjektet virket spennende.

6.1.3 Brukerhistorier

I planleggingsfasen av prosjektet ble det opprettet brukerhistorier som forklarte de mest sentrale funksjonalitetene til applikasjonen, samt subtasks tilhørende disse. En brukerhistorie er et format for å beskrive oppgaver som vektlegger oppgavens betydning og funksjon (ENTUR, u.å.). Formatet disse brukerhistoriene ble satt opp i var; selve historien og suksesskriteriene. På denne måten vil oppsettet av brukerhistorien utgjøre at misforståelser, informasjonstap og dobbeltarbeid uteblir slik at ressursene kunne utnyttes bedre.



Figur 14 - Eksempel på brukerhistorie for oppgaven "Bruker login" hentet fra backloggen

6.1.4 Personas

Målet med personas er å bli kjent med målgruppen. Ofte kan en utvikler overse at produktet skal være enkelt å forstå og ikke minst bruke, derfor ble det opprettet personas slik at dette kunne ses fra en brukers perspektiv. Det ble bestemt at det skulle opprettes to forskjellige personas; en lærer samt en økonom som ønsket å se den estimerte prisen på boligen sin, uten å gå gjennom en eiendomsmegler (se vedlegg 10). Både basert på brukerbehov og interesser fikk gruppen en bedre forståelse om hva en bruker ønsket, og dermed et bedre syn på hvordan systemet skulle fungere.

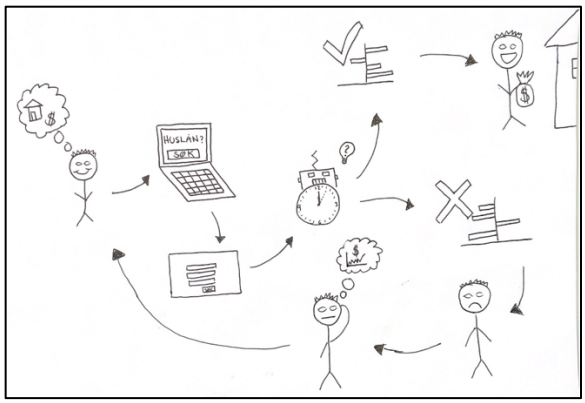
6.2 Design

I sprint 2 var design av webapplikasjonen hovedfokuset. Det var viktig å ha et intuitivt og enkelt system der det er tydelig for brukeren hva som foregår. Intuitivt og enkelt betyr her at systemet var lite tidkrevende å forstå. Designfasen startet med idémyldring innad i gruppen. Dette var en kreativ og effektiv måte å dele individuelle tanker i plenum. Rikt bilde var det første som ble opprettet i prosjektet. Gruppen fikk da et visuelt bilde av hvordan kundereisen så ut. Videre ble det utarbeidet sketches og wireframes som gjorde det mer tydelig hvordan webapplikasjonen skulle se ut. Etter at det grove designet var på plass, sto navigation map og prototype for tur.

Designfasen ble gjennomført før vinklingen av oppgaven fant sted. Dette hadde ikke stor påvirkning på designet, da problemstillingen i hovedsak fortsatt var den samme, og tanken om hvordan systemet skulle se ut ikke endret seg underveis. Det ble derimot gjort endringer på innholdet på webapplikasjonen, så designelementene er noe misvisende når det gjelder overskrifter og innhold. Vinklingen krevde relativt lite endringer mtp. design, noe som gjorde at gruppen bestemte at det ikke var hensiktsmessig å starte denne prosessen på nytt, og dermed ble de originale beholdt.

6.2.1 Rikt bilde

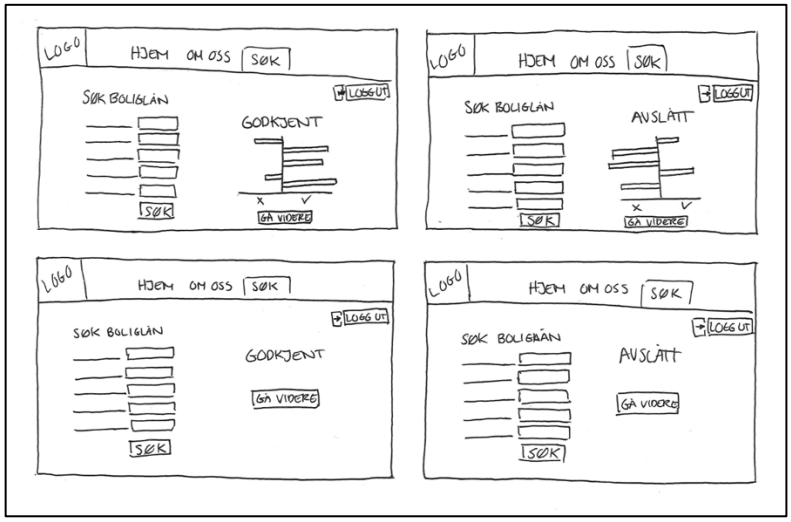
"Basert på det rike bildet ser vi hvem som er involvert og hvilket syn de har på situasjonen" (Bratteteig & Herstad, 2018). Gruppen så på rikt bilde som en viktig start på designfasen, så tre av gruppemedlemmene satt seg ned og tegnet kundereisen hver for seg. Her kom det frem at gruppen hadde en lik visjon om hvordan systemet skulle fungere. Det ble opprettet et endelig utkast av rikt bilde, der elementer fra de tre individuelle ble trukket sammen.



Figur 15 - Rikt bilde

6.2.2 Sketches

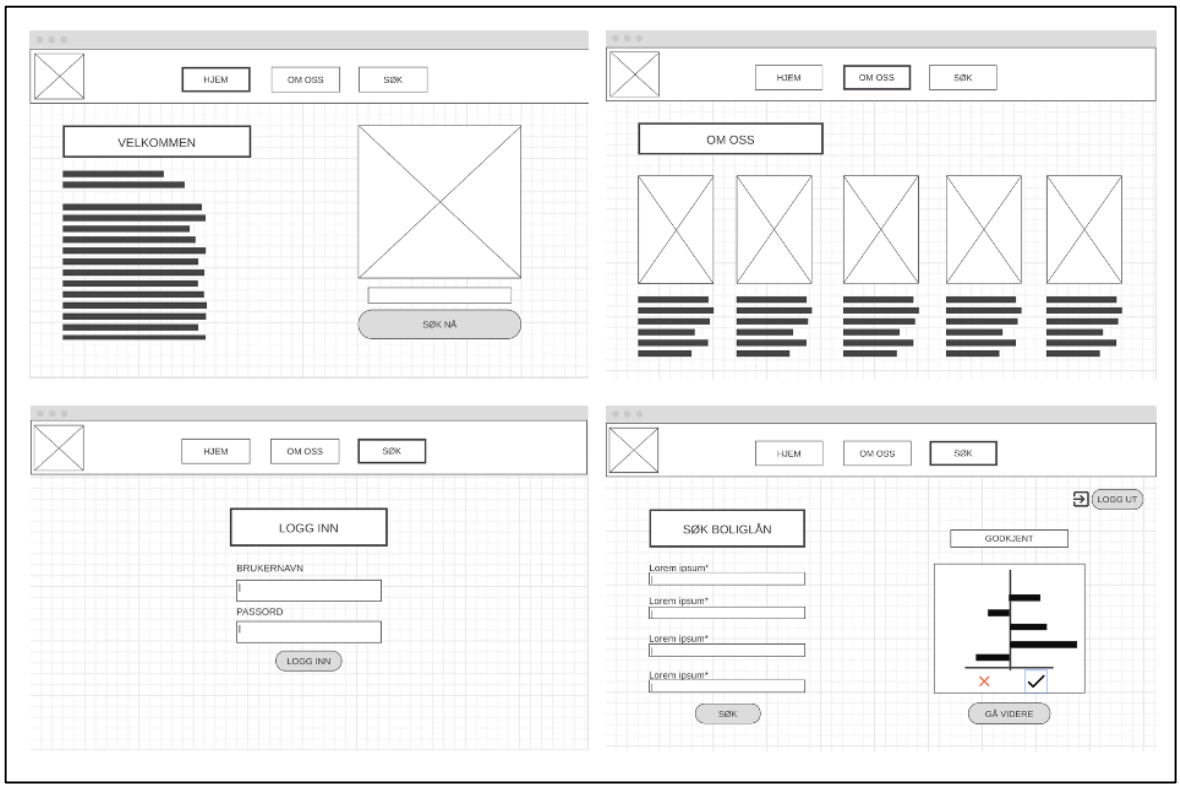
Målet med sketches var å få en grov oversikt over hvordan systemet skulle se ut, samt å danne en enighet angående designet. Penn og papir ble benyttet for at elementer enkelt kunne legges til og fjernes. Her tegnet tre av gruppemedlemmene raskt en sketch for så å sammenligne med de andre. Til slutt tegnet et av medlemmene en mer nøyaktig sketch som gruppen skulle bruke videre i designfasen. Figur 16 presenterer noen av sketchene i prosessen.



Figur 16 – Sketches utdrag

6.2.3 Wireframes

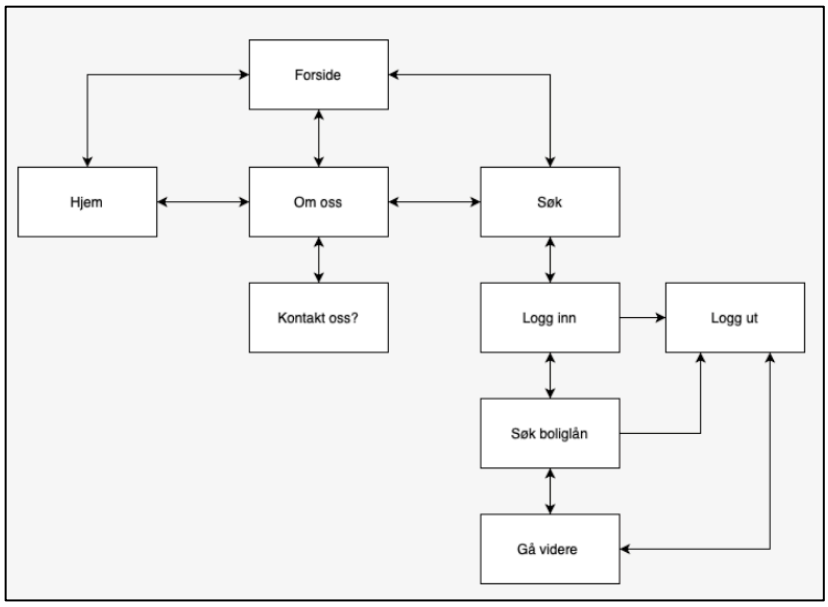
Wireframes brukes tidlig i utviklingsprosessen for å etablere den grunnleggende strukturen på en side før visuellt design og innhold legges til (Experienceux, u.å). Wireframes er ikke detaljerte, men gir en god indikasjon på hvor elementer skal plasseres slik at jobben for utviklerne blir lettere. Her ble en nettside med navn wireframe.cc benyttet for å utarbeide wireframes for både web browser, nettbrett og mobil. Ut ifra sketchene fikk gruppen en bedre oversikt over hvor de forskjellige elementene skulle plasseres.



Figur 17 – Wireframes utdrag

6.2.4 Navigation map

Det ble også bestemt at det skulle utarbeides et navigation map. Dette ga en oversikt over sammenhengen mellom de ulike sidene, slik at gruppen fikk et overordnet blikk om hvordan koblingen skulle se ut. Dette hindret også usikkerhet rundt webapplikasjonens oppbygning og struktur. Til sammen var det åtte sider som skulle kobles sammen, der forsiden var utgangspunktet. Etter dette var det "Om oss" og "Søk/kalkuler" sidene som bygget videre fra forsiden.



Figur 18 - Navigation map

6.2.5 Prototype

En prototype er en klikkbar visning av hvordan webapplikasjonen skal se ut (Experienceux, u.å.). Etter utforsking av hvilke prototype-tjenester som fantes, ble det senere bestemt at prototypen skulle lages i proto.io (se vedlegg 11). Her var det lett og oversiktlig å opprette en prototype, og nettsiden gjorde det også enkelt å legge inn standard elementer som menyer, søkefelt osv. For å skape et godt design på webapplikasjonen ble flere av valgene basert på Benyons 12 designprinsipper, hvor blant annet visibility, consistency og navigation har vært høyt prioritert. Visibility dreier seg om at det skal være tydelig for brukerne hvor på siden de befinner seg. Consistency går ut på konsekvent bruk av farger, fonter og plasseringer, mens navigation omhandler en enkel navigasjon inne på webapplikasjonen (Benyon, 2014, s. 86-87).

Font: Når gruppen skulle komme fram til hvilken type font som skulle brukes var det noen kriterier som ble spesielt mer vektlagt enn andre. Disse kriteriene var hovedsakelig at de var lesbare for brukerne og at de passet inn i designet av prototypen som ble laget. Tidlig bestemte gruppen seg for å gå for en kombinasjon av to fonter, en font til overskrifter og en annen til innholdet på de forskjellige sidene.

Gruppen besluttet å bruke fonten Open Sans til overskriftene og Montserrat for innholdet. Open Sans er en Google font som er optimalisert for web- og mobilgrensesnitt. Fonten utmerker seg med tanke på god lesbarhet (Google Fonts, u.å.). Alt innhold på webapplikasjonen er skrevet med fonten Montserrat. Dette er også en Google Font som passer

til nettsider grunnet den gode lesbarheten. Ifølge Google Font, passer Montserrat godt sammen med Open Sans (Google Fonts, u.å.). På dette grunnlaget falt valget på denne sammensettingen av fonter.

Farger: Gjennom startfasen av prosjektet kom gruppen raskt fram til at det skulle utarbeides et minimalistisk design. Denne beslutningen ble, på lik linje som valg av font, basert på tidligere erfaringer samt hva som er trenden på nettsider i 2021 (Smith, 2020). På bakgrunn av dette ble det også besluttet å bruke fargen hvit, og deretter finne den fargen som passet best. Etter en del utforsking og testing landet valget på en lyseblå farge som gruppen mente komplimenterte hvitfargen på best mulig måte. Den blå fargen er lett på øyet og er ofte assosiert med profesjonalitet da logoer som Skype, Zoom og Facebook tar i bruk fargen (Canva, 2021).

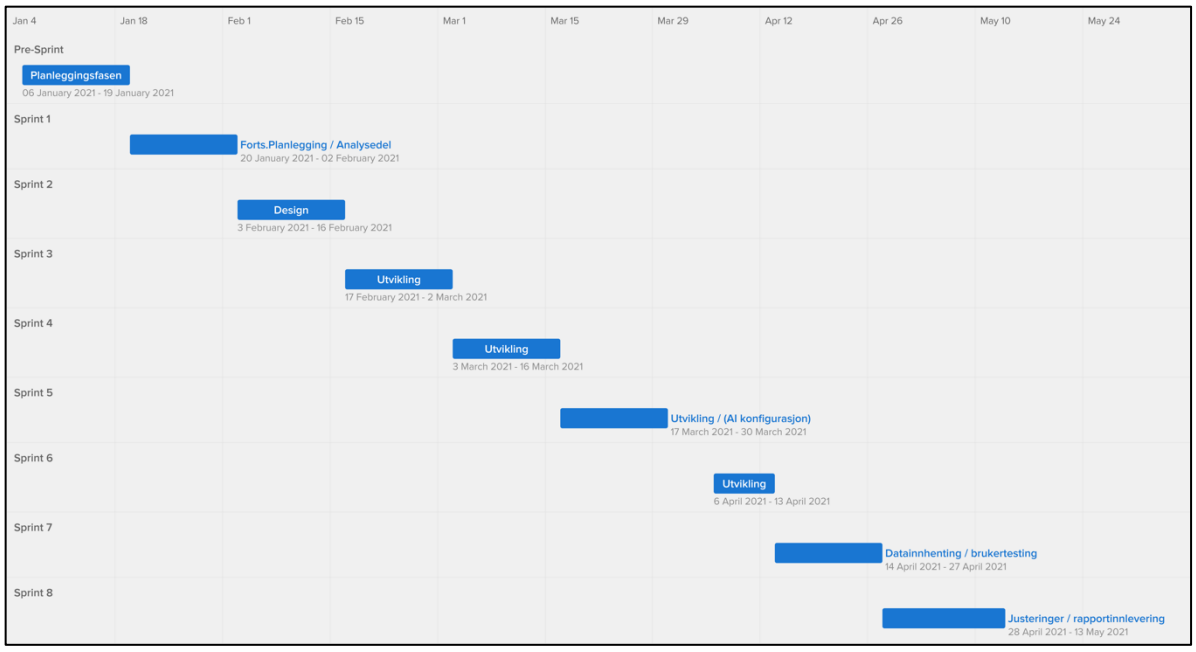
6.2.6 Universell utforming

Universell utforming går ut på at flest mulig skal ha tilgang og mulighet til å bruke ulike tjenester uavhengig av funksjonsevne (Uutilsynet, u.å.). Prinsippet er at istedenfor tilrettelagte tilbud for de med nedsatt funksjonsevne, skal den originale tjenesten være tilgjengelig/brukbar for alle (Uutilsynet, u.å.).

Universell utforming har blitt lovpålagt, og fra 1. januar 2021 er det et krav at alle IKT tjenester følger kravene for universell utforming (Tilsynet for universell utforming av ikt, 2020, 0:05). Derfor prioriterte gruppen universell utforming da webapplikasjonen skulle utvikles. Dette innebærer god lesbarhet, klar tekst, og farger som møter kravene. I tillegg er det viktig å ha mulighet til å navigere med tastaturet (Uutilsynet, u.å.).

6.3 Sprinter og implementeringer

I denne seksjonen vil hovedmålene i hver sprint bli presentert, samt en kort refleksjon rundt hva som har gått bra, og hva som kunne forbedres. Det ble utarbeidet en tidslinje som viser planen for når de forskjellige sprintene skulle gjennomføres og hva hovedfokuset skulle være. Figur 19 viser kun en grov oversikt over alle sprintene og illustrerer tidsaspektet i prosjektet. Mer utdypende figurer over hva som er blitt gjort i de ulike sprintene finnes i kapittel 6.3.1 - 6.3.9.



Figur 19 - Prosjektets tidslinje

6.3.1 Pre-sprint – Planleggingsfasen (6. januar - 19. januar)

Figur 20 - Pre-sprint tidsrom og hovedfokus for sprinten

Før gruppen gikk i gang med sprint 1 var det en felles forståelse for å sette av en egen presprint. Denne presprinten gikk ut på å få en oversikt over målet med prosjektet samt å planlegge de kommende sprintene. Dette var svært nyttig fordi det bidro til at alle gruppemedlemmene var på samme bølgelengde og ikke hadde forskjellige forestillinger av hvordan prosjektet skulle gjennomføres.

Videre satte gruppen seg ned for å planlegge og diskutere arbeidsmengde, prosjektstyring og ledelse av prosjektet. Gjennom denne perioden ble det satt målsettinger om hvordan gruppen skulle planlegge de ulike sprintene og hvor lenge de skulle vare.

I tillegg til overnevnte ble det også satt av tid til å gjøre seg kjent med forskjellige systemer og teknologier som potensielt ville bli brukt i prosjektet. Gruppen var produktive tross hjemmekontor og Teams-møter. Samarbeidet innad i gruppen fungerte bra, og gruppen var positive til tiden fremover sammen. På det første sprint-møte med produkteiere fra

TietoEVRY hadde ikke gruppen planlagt hvilke oppgaver som skulle inn i sprint 1 før møtet, noe som burde ha blitt gjort. Grunnen til dette var en misforståelse, da gruppen trodde at dette var en felles oppgave som produkteiere også skulle ta del i. Denne lærdommen tok gruppen med seg videre til de kommende sprint møtene. Det ble senere bestemt at gruppen selv skulle prioritere oppgavene før sprint møtene, og dermed få litt erfaring som produkteiere.

6.3.2 Sprint 1 - Fortsettelse av planlegging og analyse (20. januar - 3. februar)

Figur 21 - Sprint 1 tidsrom og hovedfokus for sprinten

I sprint 1 gikk mesteparten av tiden til å strukturere og organisere oppgavene som til slutt skulle utgjøre backloggen. Oppgaver fra backloggen ble deretter flyttet inn i den kommende sprint backloggen. Etter pre-sprinten lærte gruppen viktigheten av å sette av spesifikke datoer for oppgavene og planlegge disse på en bedre måte. I tillegg ble det også gjennomført intervjuer, satt opp krav for universell utforming og foretatt et valg av AI-leverandør. Etter undersøkelser og samtaler med TietoEVRY stod valget til slutt mellom AI-tjenesten fra Google eller IBM. Det viste seg at begge selskapene hadde relativt like tjenester, men AI-veilederne i TietoEVRY hadde på generell basis bedre erfaringer med dokumentasjon til Google rammeverket. Valget falt dermed på Google sin tjeneste som nevnt i kapittel 3.

Gruppen valgte å gjennomføre fire intervjuer for å få et bredere perspektiv på dagens lånesøknadsprosess. Det ble dermed besluttet å intervju to bankansatte og to kunder som nylig hadde søkt om boliglån. På denne måten forsikret gruppen seg om å få forståelse av ulike aspekter og perspektiver i dagens søknadsprosess fra begge sider av prosessen.

Lærdommen i denne sprinten ble i all hovedsak knyttet til tidsestimering. Dette var noe gruppen hadde lite erfaring med fra tidligere, og som følge av dette ble de fleste oppgavene underestimert. Dette førte igjen til at noen oppgaver måtte gå over til neste sprint. I tillegg måtte gruppen også bli bedre til å gjennomføre daily standup, slik at alle gruppemedlemmene hadde en bedre oversikt på starten av dagen. Bedre gjennomføring

innebar at alle gruppemedlemmer måtte bli flinkere på å holde en muntlig dialog gående under daily standup, i tillegg til å få notert ned hva som ble sagt.

I slutten av denne sprinten ble gruppen enige, etter oppfordring fra veileder, om å opprette en risikomatrise som skulle brukes til å analysere ulike typer risiko som kunne påvirke arbeidet. Planen da var å se på denne før starten av hver eneste sprint slik at gruppen skulle være forberedt dersom noe skulle oppstå. Denne risikomatrisen var dynamisk, slik at gruppen kunne redigere matrisen etter forholdene.

6.3.3 Sprint 2 – Design (3. februar – 16. februar)

Figur 22 - Sprint 2 tidsrom og hovedfokus for sprinten

Hovedfokuset i denne sprinten var design. Planen var å utarbeide rich picture, sketches, wireframes, prototype og navigation map. Gruppen valgte å dele seg opp for å fokusere på de forskjellige punktene. Som et tiltak for å forbedre tidsestimeringen ble det bestemt at alle oppgavene skulle stykkes opp i mindre underoppgaver. Dette gjorde det enklere å estimere timer mer nøyaktig. Det ble også gitt tilbakemelding om at en oppgave ikke skulle vare lenger enn ett dagsverk.

Videre var mer dialog under daily standup et forbedringspotensial etter denne sprinten. Samtlige gruppemedlemmer var flinke på å notere ned egne arbeidsoppgaver for dagen, men hadde forbedringspotensiale når det kom til å presentere det muntlig for de andre gruppemedlemmene. I tillegg til dette ble det bestemt at logging av arbeidstimer i Jira skulle skje etter daily standup, før lunsj, og på slutten av dagen. Hyppigere loggføring ville gjøre det mer oversiktlig for en selv og andre gruppemedlemmer, slik at faktiske arbeidstimer ble loggført og ikke glemt bort.

6.3.4 Sprint 3 - Frontend (17. februar - 2. mars)

Figur 23 - Sprint 3 tidsrom og hovedfokus for sprinten

I sprint 3 var planen å starte med frontend utvikling samt enkelte backend elementer. AI og API konfigurering var også på agendaen for denne sprinten, men ble ikke startet grunnet underestimering av oppgaver i tidligere sprinter. Etter å ha gjennomført både planlegging og design var grunnlaget lagt for å begynne å utvikle. I denne sprinten ble det brukt mye tid på å bli kjent med kildekodelagringstjenesten Bitbucket for å sette en god standard fra start. I tillegg til å opprette og style de forskjellige HTML-sidene ble det laget ER- og RM-diagram til databasen.

Enkelte backend oppgaver ble også startet på i denne sprinten. Dette innebar blant annet routing av sider som var nødvendig for å sørge for riktig URL/HTTP forespørsler. Dette var derimot en kortfattet prosess. Hovedfokuset var som sagt oppsett av sider på webapplikasjonen.

På grunn av lite erfaring var mange av oppgavene i sprint backloggen fremdeles større enn de burde ha vært. Gruppen hadde forbedret seg på dette siden sist, men hadde fremdeles en vei å gå.

6.3.5 Sprint 4 - Frontend (3. mars – 16.mars)

Figur 24 - Sprint 4 tidsrom og hovedfokus for sprinten

I sprint 4 fortsatte gruppen med frontend utvikling. Mye av tiden gikk også til å prøve å finne relevante datasett for å kunne trene opp maskinlæringsmodellen, men på grunn av personvernsopplysninger var dette vanskeligere enn først antatt. Etter et møte med veilederne fra TietoEVRY ble det besluttet at det mest hensiktsmessige ville være å vinkle problemstillingen slik at det ville være mulig å få reelle data. Dermed ble lånesøknad byttet ut med prisestimering av bolig, og gruppen satt i gang innhenting av datasett. Denne tilsynelatende store endringen utgjorde ingen stor utfordring for gruppen, da hensikten fortsatt var den samme; å teste kundetilfredsheten til brukeren ved hjelp av Explainable AI versus AI.

Etter vinklingen av problemstillingen, startet letingen etter relevante og gode datasett. Det var hovedsakelig to av gruppemedlemmene som hadde ansvaret for denne oppgaven, men flere ble involvert senere ettersom det var vanskelig å finne gode datasett med nok attributter, som i dette tilfellet var boligegenskaper. Det var viktig for oppgaven sin del at datasettet inneholdt nok attributter, ettersom prisestimeringen ville bli mer presis som følge av dette. Gruppen var inne på tanken av å bruke en metode kalt *web scraping* til innhenting av boligdata. Web scraping er en prosess hvor en såkalt bot blir brukt til å innhente innhold og data fra en nettside (Imperva, u.å.). Dette viste seg dog å være mer avansert enn først antatt, så det ble dermed bestemt at innhentingen skulle gjøres manuelt. Den manuelle innhentingen av data skjedde via nettsiden Virdi.no, fordi nettsiden presenterte informasjon av høy kvalitet. Denne tjenesten blir også benyttet av meglere, banker og takstmenn (Boligmappa, 2020).

Det var mye som gikk bra i denne sprinten selv med en endring i problemstillingen. Gruppen tok med seg erfaringene fra sprint 3 og delte opp store oppgaver til mindre. Dette gjorde det lettere å fordele arbeid og ga gruppen en bedre oversikt. I tillegg til dette ble det gjort endringer i hvordan gruppen estimerte arbeidstimer. Nå ble absolutt alt arbeid loggført. Tidligere hadde gruppen utelatt daily standup, møter, og andre oppgaver som ikke hadde direkte påvirkning på prosjektet. Det var først i sprint review 3 at gruppen sammen med produkteiere innså at det ble estimert for få timer. Dette ble endret på fordi timeestimatet ikke reflekterte gruppens faktiske arbeidstid.

6.3.6 Sprint 5 – Frontend og backend (17. mars - 30. mars)

Figur 25 - Sprint 5 tidsrom og hovedfokus for sprinten

I denne sprinten skulle arbeidet rundt opptrening og implementering av AI fullføres. Ingen på gruppen hadde tidligere erfaring med maskinlæring. Dermed var det helt naturlig at gruppen ville bruke en AI-tjenestene tilbudt av en av de store teknologiselskapene. Etter grundig undersøkelse i første del av denne sprinten fant gruppen ut at XAI var et relativt nytt felt som det stod lite informasjon om på nett.

Selve opptreningen av modellen gikk veldig bra. Google AutoML Tables krevde et datasett på minimum 1000 rader, og en treningstid på minimum én time. Google Cloud Platform var komplisert ved første øyekast, men etter et par timer inne på plattformen var det greit å manøvrere seg frem.

Gruppen endte til slutt opp med et datasett på rundt 1100 rader. Grunnet manuell datainnhenting sa gruppen seg fornøyd med å fylle minstekravet til antall rader i datasettet. Gruppen var klar over at dette ville føre til en mindre presis AI, men ville ikke la for mye tid gå bort på denne oppgaven da det enda var mye arbeid som gjenstod.

Det ble også opprettet en administratorside i denne sprinten. På administratorsiden skulle det være mulig å få oppgitt data knyttet til bruken av systemet, brukere samt tilbakemeldinger fra hver enkelt bruker. Denne oppgaven gikk relativt greit for seg da det kun var behov for noen enkle databasekall for å få informasjonen som trengtes. Det som tok lengst tid var tilgangsstyring for å unngå at brukere som ikke hadde administratorstatus fikk tilgang til personopplysninger.

På grunn av lite erfaring med unittesting fra tidligere, ble det i denne sprinten opprettet og utført manuelle tester. Disse bestod av test-caser som gjorde det enkelt for testeren å gjennomføre testene, uten noe særlig krav til forkunnskaper. De manuelle testene fungerte etter sin hensikt, og det ble oppdaget småfeil som raskt kunne rettes opp i før systemet ble lansert.

Det som kunne blitt gjort bedre i denne sprinten var å jevnlig oppdatere gjenværende tid på arbeidsoppgavene i Jira. Ofte viste det seg at en oppgave tok lengre eller kortere tid enn forventet. Dermed var det å kunne oppdatere gjenværende tid underveis viktig for å få et korrekt estimat på oppgavene.

6.3.7 Sprint 6 – Frontend og backend (6. april - 13 april)

Figur 26 - Sprint 6 tidsrom og hovedfokus for sprinten

Sprint 6 var den siste sprinten som var satt av til utvikling, og målet var her å ferdigstille webapplikasjonens hovedfunksjoner. Fra tidligere av hadde gruppen en database som ikke egnet seg i et produksjonsmiljø, så denne måtte derfor endres før webapplikasjonen kunne bli lansert. Med tanke på at lanseringsdato nærmet seg, merket gruppen at det var noe mer tidspress enn normalt for å få fullført oppgavene tilhørende sprinten. I tillegg var denne sprinten kortere enn vanlig på grunn av helligdager i forbindelse med påsken. Gruppen ble dermed enige om å også jobbe med prosjektet enkelte kvelder under denne sprinten.

Til tross for en kortere sprint enn normalt, fikk gruppen ferdigstilt alt det planlagte arbeidet, herav webapplikasjonens hoved funksjonalitet. Tilbakemeldingene fra veilederne var positive. Et viktig punkt som ble tatt med til neste sprint var å bruke ETC mer aktivt, ettersom det ga en bedre indikasjon på hvor lang tid som gjenstod på de ulike oppgavene. Videre måtte gruppen vurdere nødvendigheten av møter rett etter lunsj og eventuelt droppe dette dersom det tar bort mye arbeidstid.

6.3.8 Sprint 7 – Brukertesting og datainnhenting (14. april – 27. april)

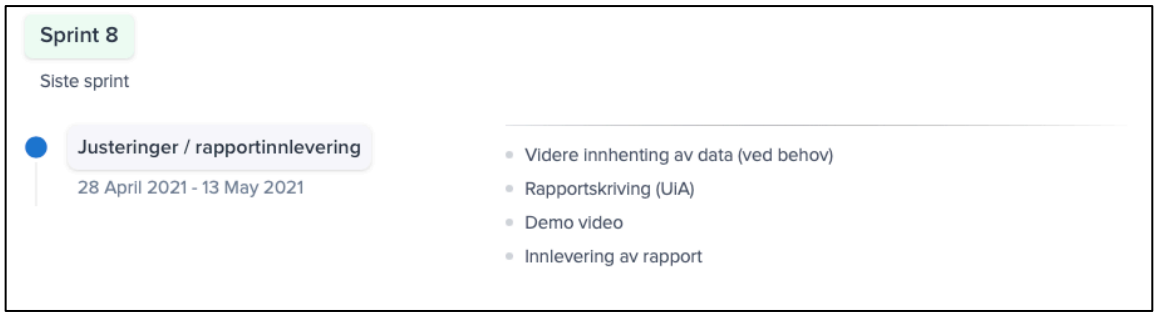
Figur 27 - Sprint 7 tidsrom og hovedfokus for sprinten

Sprint 7 var nest siste sprint, og her hadde tiden kommet for å teste systemet på ulike testgrupper. I starten av sprinten jobbet gruppen med selve lanseringsprosessen for webapplikasjonen. Før dette ble gjort ble det gjennomført noen forbedringer/justeringer på webapplikasjonen basert på tilbakemeldinger fra produkteier.

Etter møtet ble noen få endringer utført i tillegg til en stress-test av systemet. Under stress-testen fant gruppen ut at webapplikasjonen hadde en betydelig feil. Prisestimeringen ble ikke korrekt dersom systemet ble overbelastet, grunnet hostingtjenestens håndtering av variabler. Dette ble ordnet ved å sende prisestimatene samt forklaring til databasen, for så å kalle på disse verdiene når de skulle bli vist for kunden. Verdiene behøvde dermed ikke å sendes gjennom systemet som variabler, noe som var hovedroten til problemene. Denne feilen gjorde at den endelige lanseringen av webapplikasjonen ble utsatt med to dager. Heldigvis ble feilen oppdaget og rettet opp i før testgruppene skulle teste systemet.

Alt i alt gikk denne sprinten veldig bra. Gruppen fikk hentet inn nødvendig data fra testgruppene og opprettholdt en god dialog med testpersonene og protukteierne. Forbedringspotensialet fra denne sprinten, som vil bli jobbet videre med i neste sprint, er gruppens samkjøring på rapportskriving.

6.3.9 Sprint 8 – Justeringer og rapportskriving (28. april – 13. mai)



Figur 28 - Sprint 8 tidsrom og hovedfokus for sprinten

Sprint 8 var den siste sprinten i prosjektet. I denne sprinten lå hovedfokuset på rapportskriving samt eventuelle minimale justeringer på webapplikasjonen. Selve arbeidet med rapporten startet tidligere, men det har ikke vært et hovedfokus før denne sprinten.

Noe annet som også ble gjort denne sprinten var å opprette en videodemonstrasjon av selve systemet. Videoen vil være en del av rapporten og beskriver kortfattet prisestimeringsprosessen fra start til slutt.

I backloggen for sprint 8 ble det opprettet oppgaver til hver seksjon av rapporten. Dette gjorde at gruppen hadde god oversikt over hvilke deler av rapporten som var påbegynt, og ferdig. Det var noe krevende å estimere hvor mye tid som var nødvendig til hver seksjon, men ved hjelp av hyppig re-estimering fungerte dette bra. Det ble estimert for lite tid på korrekturlesing, og gruppen jobbet flere timer enn planlagt med dette.

7 A/B testing

I dette kapittelet vil funnene fra brukertestingen bli presentert, samt hvordan gruppen har kommet frem til disse. For å teste kundetilfredsheten ble forskningsmetodikken A/B testing benyttet. A/B testing omhandler prosessen om å sammenligne to versjoner av en nettside mot hverandre for å se hvilken versjon som presterer best (Optimizely, u.å.). Videre vil resultatene mellom begge testgrupper bli sammenlignet og drøftet.

7.1 Metodisk tilnærming

Metode er et begrep som beskriver prosessen for å samle inn data om virkeligheten, eller det som kalles empiri på fagspråket (Jacobsen, 2015, s.21). I dette prosjektet er det brukt to former for metode ved innhenting av data; kvalitativ og kvantitativ. “Den kvantitative

tilnærmingen har som et grunnleggende utgangspunkt at den sosiale virkeligheten kan måles ved hjelp av instrumenter som kan gi oss informasjon i form av tall" (Jacobsen, 2015, s.24). I bunn og grunn består denne formen for metode av innhenting av tallverdier og koding (konvertering) av disse. Videre tar den kvalitative tilnærmingen et utgangspunkt i at "virkeligheten er for kompleks til å reduseres til tall, og at en derfor må samle inn informasjon i form av ord som åpner for mer nyanserikdom" (Jacobsen, 2015, s.24).

I denne delen av prosjektet ble kvantitativ metode benyttet for å innhente og analysere tilbakemeldingene fra testgruppene. Tilbakemeldingene ble lagret som tall i gruppens database, og senere kodet om til betydningsfulle verdier (mer om dette under kap. 7.4 - Resultater).

7.2 Innhenting av data

For å teste kundetilfredsheten var det nødvendig å komme i kontakt med testpersoner. Ettersom prisestimeringen baserte seg på data om boliger i Oslo var det nødvendig at testobjektene var bosatt eller eide bolig i Oslo. Sammen med veilederne i TietoEVRY ble det bestemt at Roar Engen skulle publisere en oppfordring til å delta internt på Workplace, som er en intern kommunikasjonskanal som blir benyttet i TietoEVRY. I tillegg er tre av gruppemedlemmene fra Oslo området, og det ble sendt ut forespørsel om å delta til bekjente.

For å ha kontroll på testobjektene ble det først sendt ut en lenke til et interesseskjema (se vedlegg 12). Her ble ideen rundt prosjektet, samt hva det betyr å delta presentert. Dersom personene fant dette interessant og ønsket å delta, la de igjen navn, epost og hvilken bydel boligen befinner seg i.

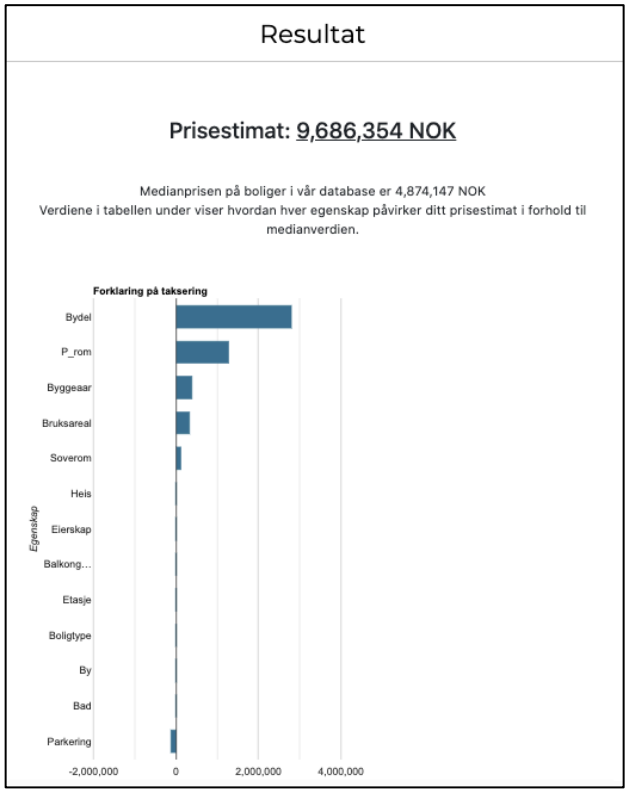
Etter at 29 personer hadde meldt sin interesse ble det opprettet en bruker til hver enkelt person. Brukerne ble opprettet av gruppen ettersom det gjorde det enklere å holde styr på testgruppene, samt unngå at andre personer utenom testobjektene opprettet brukere og skapte rot i testingen. Deretter ble lenken til webapplikasjonen samt innloggingsinformasjon send ut, og testingen av systemet kunne starte.

7.3 Testgrupper

For å kunne teste kundetilfredsheten ble gruppen enige med produkteier om at det mest hensiktsmessige var å dele opp testobjektene i to grupper, der gruppe A ville få med forklaring og gruppe B uten. På denne måten ville de to gruppene være uvitende om at det var to forskjellige fremstillinger av resultatet, og tilfredsheten ville bli målt uten påvirkning av

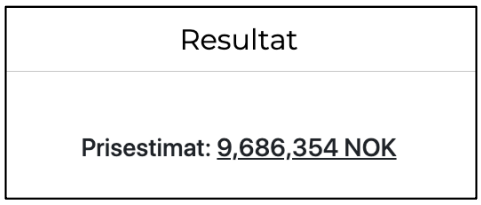
dette. For å ha to forskjellige fremstillinger av resultatene ble det lansert to uavhengige nettsider med egne databaser for å motta resultatene.

Testgruppe A: Testgruppe A bestod av 17 personer som fikk både et prisestimat og en grafisk forklaring som begrunnet prisen.



Figur 29 - Estimat med forklaring

Testgruppe B: Testgruppe B bestod av 17 personer som testet samme system, men uten en forklaring for prisestimatet.



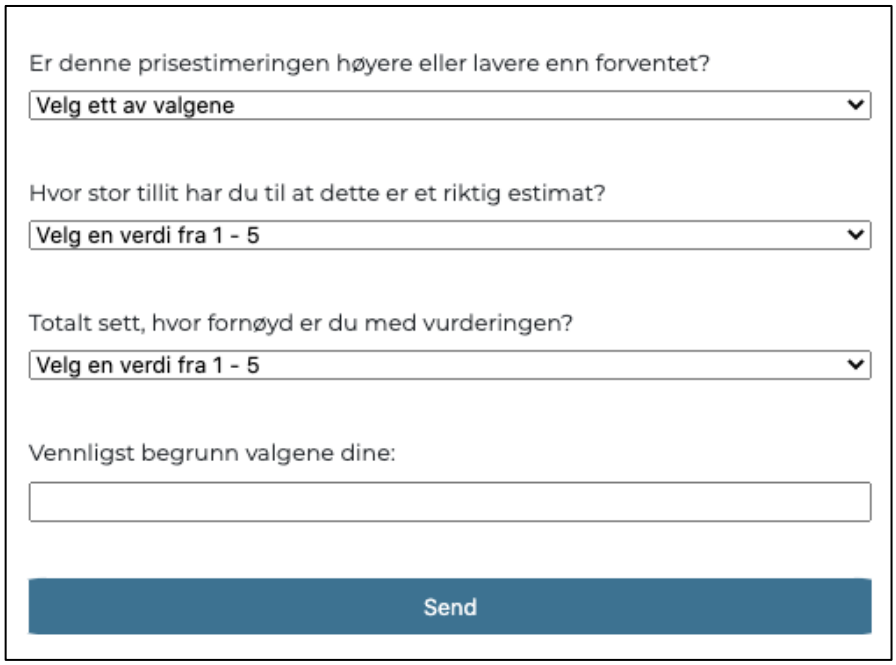
Figur 30 - Estimat uten forklaring

7.4 Resultater

Det ble satt en frist for testbrukerne slik at gruppen skulle ha god nok tid til å tolke resultatene. Da fristen hadde gått ut var det ikke alle som hadde gjennomført testingen, så det ble sendt ut en påminnelse på e-post til disse. Etter at 85% av interessentene (29 personer) hadde testet systemet ble det besluttet at tolkningen av funnene kunne starte, da det var stor

sannsynlighet for at de resterende 4 personene ikke kom til å delta. Fra testgruppe A var det 16 av 17 personer som hadde svart på undersøkelsen, mens i testgruppe B var det 13 av 17.

De tre spørsmålene testbrukerne ble stilt var:



Figur 31 - Spørsmål til testbrukere

Resultatene som fremkom var:

Testgruppe A - XAI			
Testobjekt ID	Prisforventning	Tillit	Tilfredshet
1	Lavere	2	2
2	Lavere	2	1
3	Lavere	2	2
4	Lavere	4	4
5	Lavere	3	4
6	Lavere	1	3
7	Lavere	3	3
8	Som forventet	5	5
9	Som forventet	4	3
10	Som forventet	4	4
11	Som forventet	4	5
12	Som forventet	4	4
13	Høyere	5	5
14	Høyere	2	2
15	Høyere	5	5
16	Høyere	2	2

	Tillit	Tilfredshet
Gjennomsnitt	3.25	3.38
Lavere	2.75	2.71
Som forventet	4.20	4.20
Høyere	3.5	3.5

Testgruppe B - AI			
Testobjekt ID	Prisforventning	Tillit	Tilfredshet
17	Lavere	2	3
18	Lavere	2	2
19	Lavere	4	2
20	Lavere	4	3
21	Som forventet	4	4
22	Som forventet	4	4
23	Som forventet	4	4
24	Som forventet	4	4
25	Høyere	1	2
26	Høyere	4	4
27	Høyere	1	3
28	Høyere	3	5
29	Høyere	4	4

	Tillit	Tilfredshet
Gjennomsnitt	3.15	3.38
Lavere	3.00	2.50
Som forventet	4.00	4.00
Høyere	2.60	3.60

Figur 32 - Resultater av A/B-testing (XAI vs AI)

Testgruppe A (XAI):

I testgruppe A kommer følgende resultater frem: 7 uttrykte at prisestimatet var lavere enn forventet, 4 personer sa det var høyere enn forventet, og 5 personer syntes prisen var som forventet. Gjennomsnittet på tilliten var 3.25 der skalaen gikk fra "1- svært lite tillit" til "5- stor tillit". Tilfredsheten var noe høyere, der gjennomsnittet var 3.38.

Testgruppe B (AI):

I testgruppe B kommer følgende resultater frem: 4 uttrykte at prisestimatet var lavere enn forventet, 5 personer sa det var høyere enn forventet, og 4 personer synes prisen var som forventet. Gjennomsnittet på tilliten var 3,15 og gjennomsnittet på tilfredsheten var 3,38.

7.5 Drøfting av resultater

Etter å ha gjennomført testingen viste det seg at testgruppe A og testgruppe B hadde en tilnærmet lik grad av tilfredshet og tillit til systemet. Gjennomsnittet på tillit i testgruppe A var noe høyere enn hos testgruppe B, mens graden av tilfredshet var lik i begge testgruppene.

En utfordring relatert til testingen av tilfredshet var hvor mye selve prisestimatet påvirket svarene. Ved hjelp av kommentarene brukerne la igjen kommer det frem at de fleste som oppga en lav grad av tillit eller tilfredshet enten mente de hadde fått et for lavt eller for høyt prisestimat. Det var en tydelig sammenheng der personer som har oppgitt at prisen var som forventet også har gitt en høy grad av tillit og tilfredshet. Under arbeid med å utforme spørsmålene til testbrukerne var dette et tema som ble lagt mye fokus på. I samarbeid med både veileder og produkteiere ble det diskutert hvordan spørsmålene skulle formuleres slik at de var minst mulig avhengig av prisestimatet brukeren fikk opplyst, samtidig som de ikke skulle styre testobjektene i en gitt retning.

Selv om det viste seg at mange av brukerne baserte tillitten og tilfredsheten på prisestimatet, var det også noen av brukerne i testgruppe A som poengterte at forklaringsgrafen var god. Et eksempel: "*Kortfattet og enkel å fylle ut og graf med oversiktlige forklaringer.*". I testgruppe B lød den ene kommentaren "*Kunne ønske meg en god begrunnelse for prisestimatet som ble gitt.*". I tillegg ble det observert at noen av testbrukerne i gruppe A var mindre fornøyd fordi de ikke var enige med forklaringsgrafen.

I testgruppe A er det tre personer som har valgt "5 - stor tillit" og fire som har valgt "5 - svært fornøyd". I testgruppe B er det kun oppgitt én toppscore. Dette indikerer at selv om

gjennomsnittet på de to gruppene er tilnærmet lik, er det flere i testgruppe A som er svært fornøyde og har stor tillit til estimatet.

Det kan diskuteres om testingen er generaliserbar da det kun var 29 personer som testet systemet. For å få et mer nøyaktig resultat burde systemet ha blitt testet av flere brukere, men det var ikke mulig nå på grunn av mangel på tid og ressurser.

8 Refleksjoner

I dette kapitlet vil gruppen reflektere rundt ulike aspekter som har hatt positive innvirkninger på arbeidsprosessen samt faktorer som har gitt gruppen utfordringer underveis i prosessen. Her vil det reflekteres rundt fire sentrale deler av prosjektet ble løst ved hjelp av godt samarbeid og god kommunikasjon innad i gruppen. Disse temaene inkluderer vinkling av prosjektoppgave, bruk av Scrum, videre utvikling av prosjektet, og utfordringer som har oppstått gjennom prosjektet.

8.1 Vinkling av prosjektoppgave

Vinklingen av prosjektoppgaven står sentralt i prosjektet. Gruppen hadde kommet godt i gang med prosjektet da det ble besluttet å vinkle oppgaven på grunn av manglende historisk data. Den agile arbeidsmetodikken gruppen jobbet etter medførte derimot ikke noen store forsinkelser eller problemer.

Valget om å vinkle problemstillingen viste seg å være fordelaktig. Dersom vinklingen ikke hadde blitt gjennomført kunne det ført til to mulige scenarioer. Enten ville gruppen ha fått tak i et datasett, men på grunn av lang ventetid ville dette påvirket sluttresultatet og ført til et langt større tidspress. Et annet utfall kunne vært at gruppen måtte brukt et datasett uten reelle verdier som ville påvirket kvaliteten av sluttproduktet. Alt i alt førte vinklingen til at gruppen klarte å levere et produkt med god kvalitet og gode data innenfor tidsrammen som var satt.

8.2 Scrum

Gjennom prosjektet har Scrum vært et viktig rammeverk for å sikre god kvalitet, målrettet fremgang i prosjektet og god oversikt over den gitte perioden. Med daily standup har alle til enhver tid visst hva de skulle jobbe med og hatt oversikt over hva de resterende gruppemedlemmene jobber med. Det ble også benyttet et ekstra møte på slutten av dagen for

å gjennomgå status på oppgavene samt utfordringer. Bakgrunnen for dette var at gruppen ville opprettholde hyppig dialog i en periode med mye hjemmekontor. Sprint planning har vært en avgjørende faktor for å gjennomføre gode sprinter ettersom det har gitt både gruppen og produkteierne en oversikt over målene for hver sprint. Ved å ha sprint review og retrospective har gruppen omstilt seg og tatt med viktige erfaringer inn i neste sprint, noe som har ført til et stort læringsutbytte gjennom hele prosjektet. Bruken av Scrum i bachelorprosjektet har gitt gruppen en bedre forståelse for metodikken som brukes og hvordan det er å jobbe agilt.

8.3 Videre utvikling

Etter produktet var ferdig utviklet etter produkteiers ønsker for funksjonalitet, sitter gruppen igjen med et godt fungerende "proof of concept". Potensialet for prosjektet videre er stort, noe som ble bekreftet av sjefen for kundeinnsikt i DNB, som gruppen var så heldige å få en ha samtale med i sluttfasen av prosjektet. Et godt poeng som ble trukket frem her var at forklaringen kunne ha blitt forbedret noe ved å kun inkludere de attributtene som trekker prisen mest opp og ned, i stedet for å vise alle attributtene. Dette kunne redusere risikoen for misforståelser rundt forklaringen.

Noe annet som burde blitt gjort ved en eventuell videreutvikling er å inkludere mer historisk data i opptrening av maskinlæringsmodellen. Prosjektet som gruppen har utviklet tar ikke for seg prisutvikling i markedet, men gir bare grove estimater for hvordan prisene er i 2020/2021. Dersom mer historisk data hadde blitt innhentet kunne modellen funnet mønstre i prisutviklingen fra år til år og måned til måned, som igjen kunne gjort applikasjonen mer brukbar i et langsiktig perspektiv.

8.4 Utfordringer

Gjennom semesteret har gruppen møtt på ulike utfordringer som med varierende konsekvenser. Gjennom dette delkapitlet vil gruppen ta for seg hvilke utfordringer som oppsto i løpet av prosjektet og hvordan disse ble håndtert og løst.

8.4.1 Covid-19

Covid-19 har bydd på en rekke utfordringer. Konsekvensene var at det ble noe utfordrende arbeidsforhold, da mesteparten av arbeidet foregikk på hjemmekontor. Da smittetallene var lavere i Kristiansand var det en periode hvor gruppen fikk lov til å sitte på kontoret å jobbe med prosjektet. Dette var veldig inspirerende da gruppen fikk tildelt plass i et fint og stort lokale. Dessverre tok det ikke lang tid før smittetrenden snudde slik at gruppen

måtte trekke seg tilbake på hjemmekontor. På tross av dette har gruppen klart å holde motivasjonen oppe med faste oppmøtetider på Teams hver dag innad i gruppen, og faste møtetidspunkter med veiledere i TietoEVRY. I tillegg har veilederne alltid vært tilgjengelig via Teams dersom gruppen hadde noen utfordringer, noe som har vært til meget stor hjelp. Til syvende og sist har gruppen, og veilederne, håndtert utfordringene på en god måte ved å opprettholde en tett dialog gjennom hele prosjektet.

8.4.2 Tidsestimering

Ettersom estimering er noe som er vanskelig for uerfarne så vel som erfarne, var dette en stor utfordring for gruppen, spesielt i startfasen av prosjektet. Hovedårsaken til dette var generelt sett lite erfaring med omfattende prosjekter fra tidligere, samt mye ukjent fagstoff.

Til å begynne med ble kun oppgaver som hadde direkte tilknytting til applikasjonen estimert og ført ned, mens alle de indirekte oppgavene som daily standup og diverse møter uteble. Etter at dette ble påpekt av produkteier i begynnelsen av sprint 4 begynte gruppen å inkludere alle oppgaver i estimeringene, både de som direkte kunne knyttes til produktet, samt de som var noe mer indirekte. Dette gjorde at gruppen fikk en langt bedre oversikt i prosjektet slik at estimatene reflekterte det faktiske arbeidet.

Etter hvert fant gruppen ut at estimering var noe som ble bedre og bedre med tid, ettersom god logging av timer og retrospective gjorde det mulig å ta lærdom fra erfaringer. Gruppen fikk også dette bekreftet av ansatte i TietoEVRY med lang arbeidserfaring i IT bransjen.

9 Konklusjon

I dette bachelorprosjektet har gruppen jobbet med å svare på problemstillingen som angår kundetilfredshet basert på kunstig intelligens med og uten forklaring. For å få til dette ble det utviklet en Flask webapplikasjon som estimerer boligpriser ved hjelp av Google Cloud AutoML. Maskinlæringsmodellen som ble trent opp i Google Cloud baserte seg på reelle salgsdata fra boliger i Oslo slik at estimatene skulle bli så virkelighetsnære som mulig.

For å svare på problemstillingen er det ingen markant forskjell i tilfredsheten blant de som fikk forklaring versus de som ikke fikk, men det var noe høyere grad av tillit blant gruppen med forklaring. Det er flere faktorer som spiller inn på funnene, som for eksempel at den testgruppen som fikk et prisestimat med en tilhørende forklaringsgraf fokuserte mer på selve estimatet enn den gitte forklaringen. I tillegg viste det seg i noen tilfeller at forklaringen førte til lavere tilfredshet dersom brukeren var uenig i noen av faktorene som hadde trukket prisen opp eller ned. Begge gruppene var stort sett fornøyd dersom de fikk et estimat som de selv syntes reflekterte boligens faktiske verdi ("som forventet"), noe som kan påvirke den positive responsen på tilbakemeldingene uavhengig av forklaringen.

Planlegging og prosjektgjennomføring har vært svært sentrale elementer gjennom hele prosjektet. Fra start til slutt har gruppen tilegnet seg mye kunnskap, og det har vært en stor lærdom å anvende teori i praksis. Samarbeidet med TietoEVRY har gitt gruppen god innsikt i hvordan prosjektstyring foregår i en stor IT bedrift, og dette er verdifull erfaring som vil bli tatt med videre ut i arbeidslivet. Den tette kommunikasjonen og planleggingen har gjort det mulig for alle prosjektmedlemmer å hele tiden ha god oversikt i prosjektet, samt vite hva resterende gruppemedlemmer jobber med. Dette har sørget for god kontinuerlig fremgang, samt sikret kvalitet i sluttproduktet.

Alt i alt er gruppen meget fornøyd med produktet som har blitt utviklet. Temaet var relativt nytt og veldig spennende, men også utfordrende å jobbe med. Webapplikasjonen som er blitt utviklet er mer et "proof of concept" enn et ferdig produkt, men det har vist seg å fungere meget godt når det ble lansert for testgruppene. I etterkant av utviklingen har gruppen også fått gode tilbakemeldinger fra sjef for kundeinnsikt i DNB med tips om hvordan applikasjonen kunne blitt forbedret før en eventuell offentlig lansering. Dette, i tillegg til tilbakemeldinger fra veiledere, har gitt en god indikasjon på hva som kunne blitt jobbet videre med ved en eventuell offentlig lansering.

10 Referanser

Algorithmia. (2020). The importance of machine learning data. Hentet fra https://algorithmia.com/blog/the-importance-of-machine-learning-data

Atlassian. (u.å.). What is version control?. Hentet fra https://www.atlassian.com/git/tutorials/what-is-version-control

Benyon. D. (2014). Designing Interactive Systems (3. utg.). Harlow, United Kingdom: Pearson Education Limited.

Boligmappa. (2020. 17. november). DERFOR ER VIRDI.NO VERDT ET BESØK. Hentet fra https://boligmappa.no/derfor-er-virdi-no-verdt-et-besok/

Bratteteig, T. Herstad, J. (2018). Rike bilder. Hentet fra https://www.uio.no/studier/emner/matnat/ifi/IN1030/v18/undervisningsmateriale/rike-bildernotat.pdf

Canva. (2021). Everything about the color Light Blue. Hentet fra https://www.canva.com/colors/color-meanings/light-blue/

Charlesen, H. (2020). IT-selskapene tjente 39 milliarder. Hentet fra https://finansavisen.no/nyheter/teknologi/2020/09/20/7568509/her-er-listen-over-norges-500-storste-it-selskaper

Cohn, M. (2014, 07. oktober). What Is Quality? [Blogginnlegg]. Hentet fra https://www.mountaingoatsoftware.com/blog/what-is-quality

Digitaliseringsdirektoratet. (u.å.) Hva er risikovurdering? Hentet fra https://internkontroll-infosikkerhet.difi.no/risikostyring/risikovurdering

Drumond, C. (u.å) What is Scrum?. *Atlassian Agile Coach.* Hentet fra https://www.atlassian.com/agile/scrum

Eastwood, B. (2020). THE 10 MOST POPULAR PROGRAMMING LANGUAGES TO LEARN IN 2021. *Northeastern University.* Hentet fra https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/

ENTUR. (u.å.). Brukerhistorier. Hentet fra https://design.entur.org/kom-i-gang/for-designere/brukerhistorier

Experienceux. (u.å.). What is a website prototype?. Hentet fra https://www.experienceux.co.uk/faqs/what-is-a-website-prototype/

Flask. (u.å.). Testing Flask Applications. Hentet fra https://flask.palletsprojects.com/en/1.1.x/testing/

G2. (u.å.) Compare Google Cloud AutoML and IBM Watson Machine Learning. Hentet fra https://www.g2.com/compare/google-cloud-automl-vs-ibm-watson-machine-learning

GitKraken. (u.å.). Pull Requests. Hentet fra https://support.gitkraken.com/working-with-repositories/pull-requests/

Google Cloud. (u.å.). AutoML Tables. Hentet fra https://cloud.google.com/automl-tables

Google Cloud. (u.å.). Cloud AutoML. Hentet fra https://cloud.google.com/automl

Google Developers. (u.å.). Google HTML/CSS Style Guide. Hentet fra https://google.github.io/styleguide/htmlcssguide.html

Google Fonts. (u.å.). Montserrat. Hentet fra https://fonts.google.com/specimen/Montserrat#standard-styles

Google Fonts. (u.å.). Open Sans. Hentet fra https://fonts.google.com/specimen/Open+Sans#standard-styles

Guru99. (u.å.). Manual Testing Tutorial: What is, Concepts, Types & Tool. Hentet fra https://www.guru99.com/manual-testing.html

Haughey, D. (2020). MoSCoW Method. *Projectsmart.* Hentet fra https://www.projectsmart.co.uk/moscow-method.php

IBM. (u.å). AutoAI with IBM Watson Studio. Hentet fra https://www.ibm.com/cloud/watson-studio/autoai

Imperva. (u.å.). Web Scraping. Hentet fra https://www.imperva.com/learn/application-security/web-scraping-attack/

Jacobsen, D. I. (2015). Hvordan gjennomføre undersøkelser? Innføring I samfunnsvitenskapelig metode (3. Utg.) Cappelen Damm Akademisk

K, Schwalbe. (2015). Introduction to project management. *Academia.edu, WHAT IS PROJECT MANAGEMENT, 8-13*. Hentet fra https://d1wqtxts1xzle7.cloudfront.net/44752493/5e-ch-1.pdf?1460705206=&response-content-disposition=inline%3B+filename%3DAn_Introduction_to_Project_Management_Fi.pdf&Expires=1619779872&Signature=d~0EmRSoeUJzpmNwDnWIuHPnrGuOjd2RKPEJc8vs0H1kO9Sd4c54rtpwDRfh6VEnqeri6Gu3vUSVbYrGuUUjZ3~99LvAqOjufeROG7lQOk5~6BGjPg4x6TTgCJcwvaZ2dS~y-~Wj0Ka9SXPJsJyoHQ9za8A-0oBK1ubAFd94QFKtMUUWIzZINBiOa2cy47J-Xkd7j5CBItKbYnDUNYHE9NcsrC8IL7jY~a0ZboAo5nXQRj9Lr7aXLjO7eaDYHNCV1s0FQKNYFehK-~j~FJzJO52vfu28-HM2C4xHujEmAUYg1x-pgJcmc8l-J7BQfj7MA485uE-lBjrpIWvtXpZkhg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

Lamelas, A. (2018) Top main Agile methodologies: advantages and disadvanteges. *Xpadit.* Hentet fra https://www.xpand-it.com/blog/top-5-agile-methodologies/

Lyubinsky, Sapir. (2020, 27. oktober). Top Python Testing Frameworks [Blogginnlegg]. Hentet fra https://blog.testproject.io/2020/10/27/top-python-testing-frameworks/

Mindfiresolutions. (2018). Flask vs Django. Hentet fra http://www.mindfiresolutions.com/blog/2018/05/flask-vs-django/

Montiel, I. (2018, 17. september). Low Coupling, High Cohesion. *Medium*. Hentet fra https://medium.com/clarityhub/low-coupling-high-cohesion-3610e35ac4a6

Optimizely. (u.å.). A/B Testing. Hentet fra https://www.optimizely.com/optimization-glossary/ab-testing/

Patel, P. (2018). Why Python is the most popular programming language used for Machine Learning?. Hentet fra https://medium.com/@UdacityINDIA/why-use-python-for-machine-learning-e4b0b4457a77

Personopplysningsloven. (2018). Lov om behandling av personopplysninger (LOV-2018-12-20-116). Hentet fra https://lovdata.no/lov/2018-06-15-38

Pollack, J., Helm, J., Adler, D. (2018). What is the Iron Triangle, and how has it changed?. *Emerald insight.* Hentet fra https://www.emerald.com/insight/content/doi/10.1108/IJMPB-09-2017-0107/full/html?casa_token=rNCQnEsYEloAAAAA:utFk9QQ8dinZECCPFnuFSMv0r6voB0dY1EsdOdZhgfxeL9o9Qwet5IxtAyGy8FWlw24FmsarIlbOI8fhKAB4uTloanfokyNx-ZxU-OsyRI91SK05

Pytest. (u.å.). Pytest: helps you write better programs. Hentet fra https://docs.pytest.org/en/6.2.x/

Scrum.org. (u.å.). What is a Daily Scrum?. Hentet fra https://www.scrum.org/resources/what-is-a-daily-scrum

Scrum.org. (u.å.). What is a Sprint in Scrum?. Hentet fra https://www.scrum.org/resources/what-is-a-sprint-in-scrum

Scrum.org. (u.å.). What is a Sprint Retrospective?. Hentet fra https://www.scrum.org/resources/what-is-a-sprint-retrospective

Scrum.org. (u.å.). What is a Sprint Review?. Hentet fra https://www.scrum.org/resources/what-is-a-sprint-review

Singh, V. (2021, 7. januar). Flask vs Django in 2021: Which Framework to Choose?. Hentet fra https://hackr.io/blog/flask-vs-django

Smith, A. (2020). Less Is Still More: The Importance Of The Minimalist Approach To Web Design. *UsabilityGeek*. Hentet fra https://usabilitygeek.com/less-is-more-importance-minimalist-web-design/

Springboard. (2020. 31. august). Best language for Machine Learning: Which Programming Language to Learn [Blogginnlegg]. Hentet fra https://in.springboard.com/blog/best-language-for-machine-learning/

Tilsynet for universell utforming av ikt. (2020, 18. desember). Film på 1 minutt om tilgjenglegheitserklæringa [videoklipp]. Hentet fra https://www.youtube.com/watch?v=mg-XtLTqhHg

Torgersen, O. (2018, 21. mars). Marked og målgrupper. Hentet fra https://ndla.no/nb/subject:12/topic:1:88173/topic:1:183807/resource:1:88174?filters=urn:filter:6b35c125-5a82-4a30-9d60-0646c31dce32

Tranter, L. (2016). When to use kanban vs scrum. Hentet fra https://www.extremeuncertainty.com/when-to-use-kanban-vs-scrum/

UiO. (u.å.). Hva er analyse?. Hentet fra https://www.uio.no/studier/emner/matnat/ifi/INF3700/v12/undervisningsmateriale/%C3%98ving%2015.mars%202012.pdf

Uutilsynet. (u.å.). Kvifor universell utforming av ikt?. Hentet fra https://www.uutilsynet.no/veiledning/kvifor-universell-utforming-av-ikt/240

Uutilsynet. (u.å.). Løsningsforslag for nettsider. Hentet fra https://www.uutilsynet.no/wcag-standarden/losningsforslag-nettsider/36

Van Rossum, G., Warsaw, B., Coghlan, Nick. (2013). Style Guide for Python Code. Hentet fra https://www.python.org/dev/peps/pep-0008/

Wikipedia. (2021). Project Management Triangle. Hentet fra https://en.wikipedia.org/wiki/Project_management_triangle

11 Vedlegg

Vedlegg 1: Uttalelse fra oppdragsgiver

EVRY Norge AS
Roar Engen

Final v1.0
2021-05-12

Bachelor prosjekt for TietoEVRY, 2021

TietoEVRY har lange tradisjoner med gjennomføring av Bachelor prosjekter sammen med studenter fra UiA, hvor det hele veien har vært oppnådd svært gode resultater. Bachelor prosjektet i 2021 med gruppen Internet Explorers var ikke vært noe unntak. Under RefreshIT i fjor høst presenterte vi en uspesifisert oppgave under temaet Explainable Artificial Inteligence (XAI). I tiden etter RefreshIT intervjuet vi til sammen 4 studentgrupper før vi valgte gruppe 9 Internet Explorers.

TietoEVRY ønsker å jobbe med studenter som ikke går av veien for en utfordring, vi legger derfor minst mulig føringer på prosjektgruppen når oppgaven skal spesifiseres. Studentgruppen har valgt å måle hvordan bruk av AI oppleves for sluttbrukere med eller uten en forklaring på hvordan den kunstige intelligensen har tatt sine valg. Dette er et viktig tema ettersom den nye personvernforordningen (GDPR) gir den registrerte rett til å få innsyn i hvordan/hvorfor en automatisert beslutning er tatt.

For å måle dette har studentgruppen laget en webapplikasjon basert på Python, Flask og Google AI rammeverk. Løsningen benytter kunstig intelligens for å verdivurdere boliger i Oslo basert på en rekke kriterier (eks. bydel, alder, areal, parkering osv.) som kan trekke prisvurderingen opp eller ned. Deretter har man målt kundetilfredsheten og sammenlignet testgrupper som fikk forklaring ved hjelp av Explainable AI, mot dem som bare fikk en prisvurdering basert på AI uten forklaring.

Vi i TietoEVRY er imponert over hvordan gruppen har jobbet seg gjennom flere relativt store utfordringer og likevel kommet i mål med et meget godt produkt og masse god læring. Eksempelvis kan vi nevne:

- Åpen problemstilling hvor studentene selv måtte komme opp med sin egen oppgave. TietoEVRY vurderer problemstillingen gruppen valgte som meget god og veldig aktuell.
- Valg av AI rammeverk som p.t. er meget ny teknologi uten mengder av dokumentasjon.
- Gruppen måtte bytte teknisk løsning fra boliglånssøknad til verdivurdering av bolig, da datasett for opplæring av AI ble umulig å få tak i innenfor boliglånsdomenet.
- Hjemmekontor og bare digital kontakt gjennom praktisk talt hele prosjektperioden.

Studentgruppen har imponert oss med sin evne til å sette seg inn i komplekse problemstilling og ny teknologi, men aller mest har det imponert oss hvor raskt de har tatt til seg råd og tilbakemeldinger fra veilederteamet og forbedret både den tekniske løsningen og prosjektgjennomføringen. Vi har inntrykk av at studentene har hatt stort læringsutbytte av de tilbakemeldingene de har fått gjennom hele prosjektet.

TietoEVRY har hatt stort fokus på gjennomføringsmetodikken i prosjektet, i sprint reviewene og sprint planleggingen har vi jobbet tett med studentene for å sikre en god prosess. Studentgruppen har vist stor modenhet ifht metodikk. De har for eksempel vist stor evne til å reflektere over eget arbeid, spurt om råd og hele veien implementert endringer for å optimalisere prosessen. Noen gode eksempler er at de gjennomførte løpende oppdatering av risikooversikt/plan, og de hadde i løpet av prosjektet hatt meget stor fremgang innen estimering og oppfølging av fremgang i sprintene. I tillegg gledet det veilederteamet at gruppen hadde et bra fokus på kodekvalitet og testing.

tieto EVRY

TietoEVRY er glade for å få jobbe sammen med gruppen Internet Explorers og er meget godt fornøyd med resultatet studentene har oppnådd. Vi tror den læringen studentene har fått gjennom dette prosjektet vil gi dem verdifull erfaring og gjøre dem bedre forberedt til arbeidslivet.

Takk for en strålende innsats og gratulerer med et godt gjennomført prosjekt og et flott produkt!

Med vennlig hilsen veilederteamet i TietoEVRY som har bestått av:

Roar Engen, sjefskonsulent og leder av Application Innovation Agder

Nils Fredrik Iselvmo Bjerk, utvikler konsulent og tidligere UiA student

Vetle Rødbakk, utvikler konsulent med AI fordypning

Elisabeth Cederberg Øvensen, senior prosjektleder og konsulent

Duncan Irwing, leder for Analytics, AI & Automation, Consulting Principal

Kolbjørn Flaarønning, Data Scientist

Kristiansand 12.05.2021

Roar Engen
TietoEVRY

Vedlegg 2: Selvevaluering

Gruppeevaluering

Alt i alt er vi veldig fornøyde med hvordan prosjektet endte opp. Det ble tidlig avklart hvilke forventninger gruppemedlemmene hadde til prosjektet, noe som har ført til en god samkjøring gjennom hele prosjekt. Samarbeidet har fungert veldig bra, og alle gruppemedlemmene har bidratt i like stor grad. Ved å motivere hverandre og utnytte hverandres styrker sitter vi alle igjen med en god følelse, og stolthet knyttet til prosjektet.

Hedda Befring

Bachelorprosjektet har vært utrolig lærerikt og spennende å jobbe med. Det jeg synes har vært mest interessant er hele prosessen med prosjektgjennomføring og prosjektstyring. Som Scrum Master har jeg hatt hovedansvaret for det administrative arbeidet, som å gjennomføre statusoppdateringer og sprint-møter, opprettholde kontakt med TietoEVRY, og å ha en helhetlig oversikt over prosjektet. Jeg har vært innom mange områder som har vært både utfordrende og lærerike gjennom dette prosjektet som frontend utvikling, designprosesser og brukertesting. Dette har gjort at jeg har tilegnet meg mye verdifull kunnskap som jeg vil ta med meg videre ut i arbeidslivet.

Sander Kvale Garborg

Gjennom prosjektet har jeg hovedsakelig jobbet med backend elementer, som blant annet database og API tilkobling til Google sin AutoML tjeneste som ble benyttet til å foreta prisestimeringer. Fra tidligere hadde jeg noe erfaring med Python, som viste seg å være en fordel ettersom dette ble valgt som programmeringsspråk gjennom prosjektet. Selv med noe tidligere erfaring med programmeringsspråket, var majoriteten av teknologiene som ble benyttet nytt for meg. Det ble derfor brukt en del tid til å sette seg inn i disse teknologiene gjennom hele prosjektløpet. Alt i alt har det vært et utfordrende, men samtidig veldig lærerikt semester, og erfaringene som er tilegnet er noe som garantert vil bli tatt med videre ut i arbeidslivet.

Henrik Gulliksen

Dette bachelorprosjektet har vært spennende, interessant og ikke minst lærerikt. Gjennom prosjektet har jeg opparbeidet meg kunnskap i form av koding i tillegg til erfaringer med både prosjektstyring og gjennomføring. Gjennom denne prosessen har jeg jobbet med

både utvikling og prosjektstyring. Med utvikling har jeg jobbet med både frontend og backend mens jeg også har hatt ansvar for datainnsamling i form av intervjuer. Alt i alt har dette vært et utfordrende og spennende semester som har gitt meg mye ny kunnskap og erfaringer som vil komme godt med i arbeidslivet.

Jacob Mario Hjermann

Mye tid og planlegging har blitt lagt ned i dette bachelorprosjektet. Etter snart 5 måneder sitter jeg igjen med uvurderlig kunnskap som jeg vil ta med meg videre. Videre har jeg opparbeidet erfaringer innen prosjektstyring og mer tekniske oppgaver innenfor programmering. Gjennom prosjektet har min hovedrolle vært utvikler, utenom det har jeg jobbet med designprosesser. Som utvikler har jobbet jeg både med frontend og backend, og fordypet meg i innlogging/registering av brukere og Admin bruker. Til slutt vil jeg nevne at det har vært utfordrerne, men med flinke medstudenter og veiledere har veien blitt lettere.

Sander Vereide

Dette bachelorprosjektet har vært veldig lærerikt og interessant. Jeg har opparbeidet meg bedre teknisk kunnskap samt erfaringer med prosjektstyring og prosjektgjennomføring. I denne perioden har jeg jobbet med både frontend og backend. Vi har tatt i bruk nye teknologier som Google AutoML Tables til trening av vår maskinlæringsmodell. Det er vært en periode fylt med mange arbeidsoppgaver og mye selvstudie. I likhet med gruppen min har jeg møtt opp til gruppens faste arbeidstid. Til slutt sitter jeg igjen med uvurderlig kunnskap og mer erfaring rundt nye teknologier og rammeverk.

Vedlegg 3: Python Stilguide

Sjekkliste Python	Kryss av
IKKE mellomrom inntil parenteser, kolon, semikolon og komma	
Bruker kun doble hermetegn	
Dersom "=" brukes ved keyword argument, så skal det ikke være mellomrom før og etter. For eksempel: t=test ✅ t = test ❌	
Ved exceptions er det brukt spesifikke exceptions, og ikke bare en *except* klausul	
Naming conventions	
Klasser har alltid CapWords convention.	
Variabler og klasser har liten forbokstav (evt skilles med _)	
Dunders skal alltid plasseres etter docstring og før import statements.	
Konstanter er navngitt med ALL CAPITAL bokstaver, hvor flere ord er separert med _	
Variabler med én bokstav skal ikke bestå av liten L (l), stor O (O) eller stor I (I).	
Moduler har korte navn med kun lowercase. Lowercase brukes ved flere ord, men skal helst unngås.	
Code layout	
Max 79 tegn per linje (72 tegn for kommentar)	
Bruk mellomromstasten 4 ganger, IKKE TAB	
To blanke linjer før og etter functions og klassedefinisjoner	
En blank linje før og etter metodedefinisjoner inne i en klasse	
Importerte filer ligger i toppen av filen. Dersom det er flere, skal disse ligge på separate linjer (med mindre de hentes fra samme bibliotek- da kan de ligge på samme linje)	
Linjeskift før binære operatører (+, -)	
Comments	
Kommentarer er tydelige og lette å forstå. Alltid oppdater kommentarene dersom det er nødvendig etter endring i koden	
Ikke unødvendige selvforklarende kommentarer	
Blokk kommentarer er på samme indent level som koden den refererer til	
Docstrings står etter *def* line.	
""" står på egen linje dersom docstringen består av flere linjer	
""" står på samme linje dersom doctringen kun består av én linje	

Vedlegg 4: HTML Stilguide

Sjekkliste HTML	Kryss av
Fra Google (https://google.github.io/styleguide/htmlcssguide.html)	
Definerer meta charset- `<meta charset="UTF-8">`	
Alltid <!DOCTYPE HTML> I toppen av koden	
Alle elementene har en end tag (** **)	
Elementer er brukt til det som er det originale formålet med dem.	
Bilder: `alt`, `width` og `height` er spesifisert	
Tydelig skille mellom struktur (markup), presentasjon (styling) og oppførsel (scripting). CSS skal i CSS fil og ikke bakes inn i HTML.	
Entities er ikke brukt, men de faktiske tegnene som skal benyttes. Skriv € i stedet for “&eur;”	
`Type` attributt er ikke brukt	
Ny linje for hver block, liste eller table element. Hvert child element skal ha innrykk.	
Doble anførselstegn (" ") rundt attribute values.	
Fra W3Schools (https://www.w3schools.com/html/html5_syntax.asp)	
Filnavn har liten forbokstav	
Elementnavn har små bokstaver (** **)	
Attributter har små bokstaverer (<a **href**)	
Attributtverdier er I hermetegn <table class = **"striped"**>	
Ikke mellomrom før eller etter likhetstegn (=)	
Bruk <title> elementet	
Bruker *lang* elementet inne i HTMLtaggen for å spesifisere hvilket språk websiden har `<html lang="en-us">`	
Inkluder denne i alle filer slik at nettsiden tilpasser seg skjermen den sees på `<meta name="viewport" content="width=device-width, initial-scale=1.0">`	
Korte kommentarer er skrevet på denne måten <!-- Kommentar -->	
Lengere kommentarer er skrevet på denne måten: <!-- Kommentar her Kommentar fortsetter Kommentar fortsetter -->	
Bruker en enkel syntax for å linke til stylesheets. `<link rel="stylesheet" href="styles.css">`	
HTML fil har .html extensions (forside.**html**)	

Vedlegg 5: CSS Stilguide

Sjekkliste CSS	Kryss av
Alle ID'er, klasser og attributter er skrevet med små bokstaver og bindestrek er brukt dersom det er doble ord. Så korte som mulig, men så lange som nødvendig	
Presise kommentarer der koden ikke sier seg selv (/* kommentar */)	
Formatering	
Bruker alltid enkle anførselstegn(' ')	
Ikke anførselstegn i URL verdier (url())	
Ikke units etter 0 med mindre de er required (for eksempel ikke skriv 0px, skriv 0,)	
Ikke 0 foran verdier eller lengder mellom –1 og 1. eks: skriv ,8 istedenfor 0,8	
Bruk #000, for eksempel ved farger	
Declarations er listet nedover alfabetisk	
Innrykk på hver nye blokk	
Whitespace mellom property og verdi. Eks. • Font-weight:bold; ❌ • Font-weight: bold; ✅	

Opening brace er på samme linje som selektor. Eks.
`video {`
`  margin-top: 1em;}`	

Alltid ny linje for hver selektor og deklarasjon:
`h1,`
`h2,`
`h3 {`
`  font-weight: normal;`
`  line-height: 1.2;`
`}`	

Alltid ny linje for nye rules. Eks: `html { background: #fff;}MELLOMROM HERbody { margin: auto; width: 50%;}`	

Seksjoner gruppert ved kommentar dersom dette er mulig
`/* Header */`
`#adw-header {}/* Footer */#adw-footer {}/* Gallery */`
`.adw-gallery {}`	

Vedlegg 6: Testkrav

Testing

I dette prosjektet benytter gruppen Pytest rammeverket for å teste all Python-koden som er skrevet. Hovedgrunnen til dette ettersom rammeverket ligger til grunn for testing segmentet i Flask dokumentasjonen.

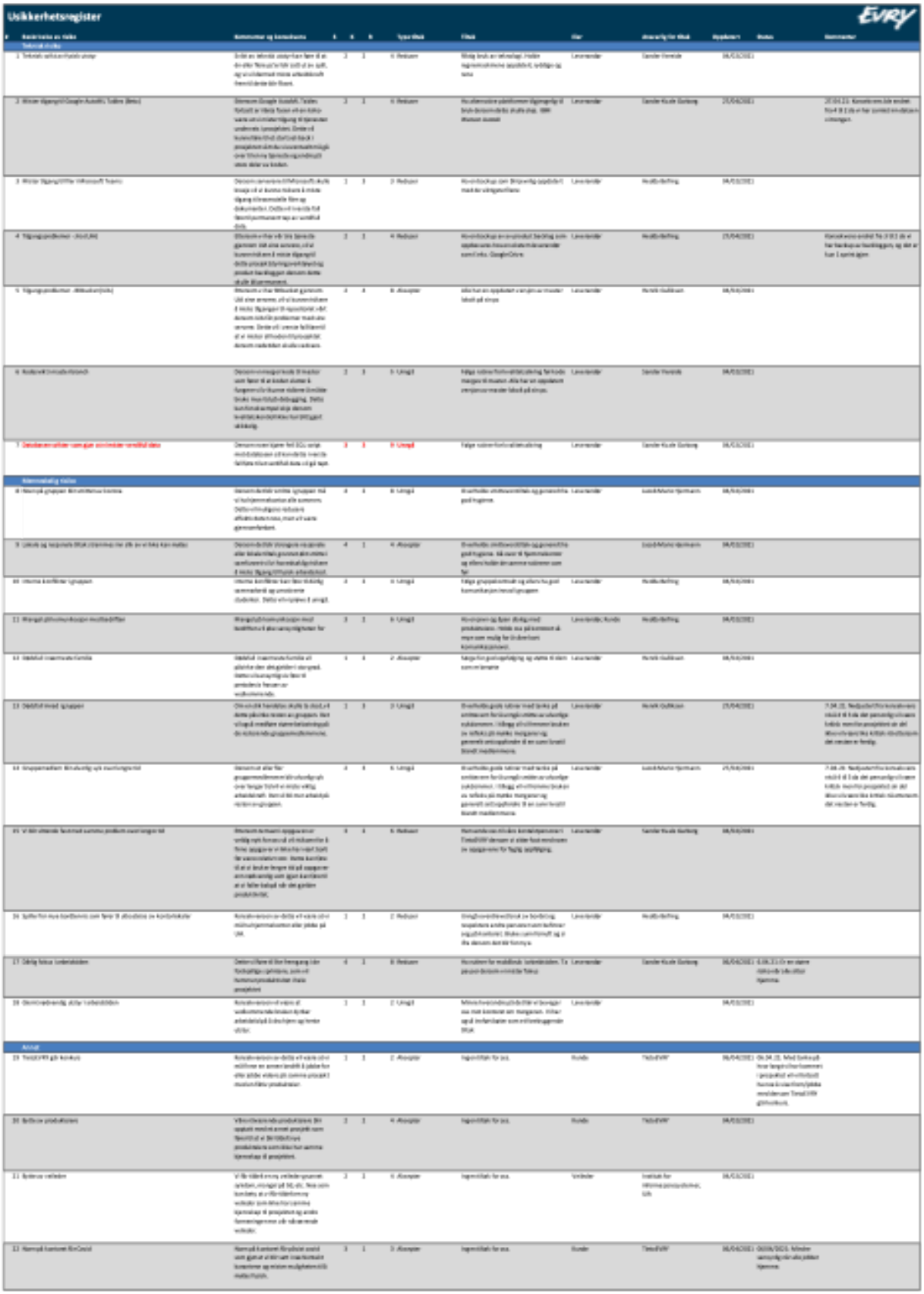
Testklasser legges inn i et directory som heter *tests* hvor hver testfil navngis *test_*.py.* Dersom denne prefiksen ikke følges vil ikke pytest registrere disse som tester og de vil dermed ikke kjøres automatisk.

Før en oppgave kan settes til done og kode kan merges til master så er noen krav nødt til å bli oppfylt:

- All kode skal ha tilhørende unit tester som må bestå før kode kan pushes til master.
    - Unit testene skrives av personen som har utviklet den tilhørende koden.
- Manuell brukertesting skal også gjennomføres etter unit testene har bestått.
    - Skal gjennomføres av en person som ikke har utviklet koden
    - *Positiv testing*: Kjøre gjennom akseptansekriteriene for en gitt oppgave/brukerhistorie for å sikre at alt fungerer som det skal.
    - *Negativ testing*: Bevisst prøve å "ødelegge" koden for å belyse eventuelle svakheter.

Dersom en kode ikke består kravene over så skal den flagges og settes tilbake til "to do" kolonnen i Jira slik at den kan modifiseres. Testeren skal da legge inn kommentarer i oppgaven/brukerhistorien hvor det beskrives hvilke tester som feilet og hva som må fikses.

Vedlegg 7: Risikoregister



Usikkerhetsregister

EVRY

#	Beskrivelse av risiko	Konsekvens og konsekvens	S	K	R	Type tiltak	Tiltak	Eier	Ansvarlig for tiltak	Oppdatert	Status	Kommentar
	Teknisk risiko											
1	Teknisk løsning feiler	Dette er en teknisk usikkerhet som kan oppstå, og vil kunne føre til at vi ikke får levert løsningen som forventet. Dette kan føre til forsinkelser og økte kostnader for prosjektet.	3	3	9	Reduser	Tidlig test av teknologi, risiko vurdering og valg av alternativ	Leverandør	Sander Hvende	04/03/2021		
2	Mister tilgang til Google Autodesk, Tableau (Beta)	Dersom Google Autodesk, Tableau (Beta) ikke fungerer som forventet, kan det føre til at vi ikke får tilgang til nødvendige verktøy for å gjennomføre prosjektet. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	3	3	9	Reduser	Vurdere alternative plattformer tilgjengelig til bruk dersom dette skulle skje. Ikke mulig å gjøre noe	Leverandør	Sander Kvale Gotberg	20/04/2021		20.04.21: Konsekvens blir endret fra 4 til 3 da vi har samlet inn tilbakemeldinger.
3	Mister tilgang til Microsoft Teams	Dersom vi mister tilgang til Microsoft Teams, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	3	3	9	Reduser	Vurdere alternative plattformer tilgjengelig til bruk dersom dette skulle skje. Ikke mulig å gjøre noe	Leverandør	Heidi Berling	04/03/2021		
4	Tilgangsproblemer - Azure AD	Dersom vi har et problem med tilgang til Azure AD, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	3	3	9	Reduser	Vurdere alternative plattformer tilgjengelig til bruk dersom dette skulle skje. Ikke mulig å gjøre noe	Leverandør	Heidi Berling	15/04/2021		Konsekvens endret fra 3 til 2 da vi har backup av backloggen, og det er kun 1 sprint igjen
5	Tilgangsproblemer - Miksukurs (nye)	Dersom vi har et problem med tilgang til Miksukurs (nye), kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	4	4	16	Akseptér	Ikke noe en oppdatert vårt plan av muske/ bleid på sirup.	Leverandør	Henrik Gulbrun	04/03/2021		
6	Redskrift innsatsforsinkelse	Dersom vi mangler kode til noe av det som fører til at koden stopper å fungerer, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	3	3	9	Unngå	Følge rutiner for kvalitetssikring før kode mergen til master. Alle har en oppdatert versjon av master lokalt på sirup.	Leverandør	Sander Hvende	04/03/2021		
7	Datalekkasje/utilsiktet varsling av sensitive verdifulle data	Dersom noen hjemler feil SQL script, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	3	3	9	Unngå	Følge rutiner for kvalitetssikring	Leverandør	Sander Kvale Gotberg	04/03/2021		
	Menneskelig risiko											
8	Mennyå gruppen blir midlertidig borte	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	4	4	16	Unngå	Overholde arbeidsmiljølov og generell fornuftig god hygiene.	Leverandør	Lucie Marie Hjelmaas	04/03/2021		
9	Lekkasje og begrenset tilgang til en av vi ikke kan møte	Dersom det blir sykmeldinger på gruppen, eller dersom det er vi ikke kan møte, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	4	1	4	Akseptér	Overholde arbeidsmiljølov og generell fornuftig god hygiene. Må være til hjemmekontor og ellers holde den samme rutinene som før.		Lucie Marie Hjelmaas	04/03/2021		
10	Interne konflikter i gruppen	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	4	4	16	Unngå	Følge gruppekontrakt og ellers ha god kommunikasjon med gruppen.	Leverandør	Heidi Berling	04/03/2021		
11	Manglet tilhørighet/følelse av fellesskap	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	3	3	9	Unngå	Kvalitetssikring av kjøre dialog med produksjon. Holde oss på kommet så mye som mulig for å opprettholde kommunikasjonen.	Leverandør, Kunde	Heidi Berling	04/03/2021		
12	Ubalanse i arbeidsmengde	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	4	4	16	Akseptér	Mange forskjellige oppfølging og støtte til dem som trenger det	Leverandør	Henrik Gulbrun	04/03/2021		
13	Dårlig med kjæppen	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	3	3	9	Unngå	Overholdelse av lover med tanke på arbeidsmiljø for å unngå å måtte ta utenlige sykdommer. Tilbygg av tilbud som bruker og ellers på måte mangler og generelt velfungerende. Et av som bredt blandt medlemmene.	Leverandør	Henrik Gulbrun	15/04/2021		7.04.21: Nedjustert fra konsekvens nivå 4 fra det personlig tilsvare bilde mentale prosjektet at det ikke er så farlig. Alltids tilstrekkelig det neste er fordig.
14	Gruppemedlem blir alvorlig syk over lengre tid	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	4	4	16	Unngå	Overholde arbeidsmiljølov og generell fornuftig god hygiene. Må være til hjemmekontor og ellers holde den samme rutinene som før.	Leverandør	Lucie Marie Hjelmaas	15/04/2021		7.04.21: Nedjustert fra konsekvens nivå 4 fra det personlig tilsvare bilde mentale prosjektet at det ikke er så farlig. Alltids tilstrekkelig det neste er fordig.
15	Vi blir utbrente fra med samme problem over lengre tid	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	4	4	16	Reduser	Vurdere alternative plattformer tilgjengelig til bruk dersom dette skulle skje. Ikke mulig å gjøre noe	Leverandør	Sander Kvale Gotberg	04/03/2021		
16	Spiller for mye bordtennis som fører til alkoholisk av kjelleforhold	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	3	3	9	Reduser	Vurdere alternative plattformer tilgjengelig til bruk dersom dette skulle skje. Ikke mulig å gjøre noe	Leverandør	Heidi Berling	04/03/2021		
17	Dårlig fokus i arbeidstiden	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	4	3	12	Reduser	Vurdere alternative plattformer tilgjengelig til bruk dersom dette skulle skje. Ikke mulig å gjøre noe	Leverandør	Sander Kvale Gotberg	04/03/2021		6.04.21: Er en større risiko enn vi ikke ønsker å ha. Må være til hjemmekontor.
18	Økonomiavhengig dårlig i arbeidstiden	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	3	3	9	Unngå	Vurdere alternative plattformer tilgjengelig til bruk dersom dette skulle skje. Ikke mulig å gjøre noe	Leverandør		04/03/2021		
	Annet											
19	TestEVRY gir konkurs	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	3	3	9	Akseptér	Ingen tiltak for oss.	Kunde	TestEVRY	04/03/2021		06.04.21: Med bakgrunn i hvor langt vi har kommet i prosjektet vil vi tilbudt kunne kvalifisere/påbegynne en ny tilbud gjennomført.
20	Syttende pandemien	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	3	3	9	Akseptér	Ingen tiltak for oss.	Kunde	TestEVRY	04/03/2021		
21	Sjokkrus veiledere	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	3	3	9	Akseptér	Ingen tiltak for oss.	Veileder	Institutt for informasjonsteknologi, UiA	04/03/2021		
22	Mennyå kontoret fin Covid	Dersom det blir sykmeldinger på gruppen, kan det føre til at vi ikke får tilgang til nødvendige filer og dokumenter. Dette kan føre til forsinkelser og økte kostnader for prosjektet. Det er derfor viktig å ha en plan for hvordan vi skal håndtere dette dersom det skulle oppstå.	3	3	9	Akseptér	Ingen tiltak for oss.	Kunde	TestEVRY	04/03/2021		06.04.21: Alleder vanskelig å kvalifisere, påbegynne.

Vedlegg 8: Risikomatrise

Risikomatrise

Gjeldende Risikobilde

Sannsynlighet						
	4	Sannsynlig	9	8, 17		
	3	Mulig	22	11, 15	7	
	2	Mindre sannsynlig		1, 10, 20, 21	4,6	2, 5, 14
	1	Usannsynlig		12, 16, 18, 19	3	13
			Ubetydelig	Moderat	Alvorlig	Kritisk
			1	2	3	4
			Konsekvens			

Initielt risikobilde

Sannsynlighet						
	4	Sannsynlig				
	3	Mulig	9	8, 15, 17	7	
	2	Mindre sannsynlig		1, 10, 11, 20, 21	4,6	2, 5, 14
	1	Usannsynlig		12, 16, 18	3, 19	13
			Ubetydelig	Moderat	Alvorlig	Kritisk
			1	2	3	4
			Konsekvens			

Vedlegg 9: Intervju

Intervju med eiendomsmegler fra DNB

Estimert lengde på intervju:

- Rundt 30 min

Innledningsspørsmål

- **Hvordan foregår en salgsprosess?**

Svar noteres her:

Hvordan salgsprosessen foregår avhenger av firma. Jeg kan ta DNB som utgangspunkt, ettersom det er der jeg jobber.

Det starter med at kunden booker oss inn for en befaring. Da kommer vi til eiendommen og ser på den. Dersom det viser seg at kunden ønsker å selge, er det din oppgave som eiendomsmegler å selge deg inn og få kunden til å velge deg.

Dersom du blir valgt ut til å ta på deg oppgaven blir man enig om en pris kunden skal betale for tjenesten. Deretter skjer alt av formaliteter som befaring, salgsoppgaver, foto, innblanding av takstmann osv. før boligen blir lagt ut for salg.

- **Hvilke egenskaper ved en bolig pleier å trekke opp og ned prisen?**

Dette varierer ofte, da alle boliger er unike, og folk har forskjellig syn på for eksempel plassering, tilstand og så videre.

Trekker opp:

Det som trekker prisen desidert mest opp er en høy standard på boligen. Det er takstmenn som vurderer standarden, på en skala fra 1-3 også er det med å påvirke prisantydningen. Andre ting som ofte trekker opp prisen er:

- *Heis*
- *Balkong*
- *Utsikt*
- *Parkering*
- *Plassering*

Dersom du kan sjekke av alle disse boksene, og boligen er i god stand vil dette påvirke prisen mye.

Trekker ned:

Det som trekker mest ned er dårlig standard på boligen. Dersom det er mye som må fikses, eller det er et oppussingsobjekt vil prisen synke. Selv om trenden viser at oppussingsobjekter

blir solgt for lavere pris har det vært en endring under koronasituasjonen. Det er mange som er interessert i å tjene penger, og hvis de har mye tid til overs investerer flere i oppussingsobjekter. Dette er derimot ikke så vanlig.

- *Dårlig standard*
- *Mye som må gjøres med boligen*
- *Oppussingsprosjekt*
- *Ikke like relevant på generell basis, men i koronatiden har oppussingsobjekter gått for høyere priser for å tjene penger og har tid til overs. Men dette er egentlig sjeldent.*

- **Finnes det noen digitale løsninger for prisestimering av bolig, eller går det alltid gjennom en takstmann?**

Ja, vi bruker "Eiendomsverdi". Det er noe jeg tror alle meglerforetak bruker. Det er et stort program som genererer et prisestimat basert på forskjellige inputs. Ved å for eksempel legge inn adressen på boligen vil vi få opp alt som er solgt i området. Programmet estimerer verdien basert på forskjellige inputs, verdien i nåværende marked, og hva andre boliger er solgt for. Den tar ikke høyde for alt, men det er et veldig godt verktøy. Dersom for eksempel "Eiendomsverdi" ikke har fått med seg at boligen er totalrenovert må vi justere prisen deretter. Dette programmet er gull verdt og gjør så å si hele jobben for vår del når det kommer til prisantydninger.

- **Hadde du hatt tillit til at et slikt system hadde estimert korrekt beløp på boliger?**

"Eiendomsverdi" har også en egen funksjon som genererer e-takster. Her får du begrunnelse for hvorfor prisen er som det er. Verdivurdering = hvorfor er prisen som den er. Her vises det for eksempel "scorer så og så høyt pga. dette og dette". Både vi og kundene må ha tillitt til dette systemet. Det er godt utviklet og har vært på markedet lenge så vi har ingen grunn til å tvile på dette.

- **Er du tilfreds med hvordan prosessen med takstmann og verdivurdering på bolig fungerer per dags dato?**

Ja, takstmann er veldig straight forward. Megler – takstmann er det eneste leddet som er uavhengig i salg av bolig. Verdivurdering fungerer ekstremt bra. Jeg vet ikke om takstmenn har tilgang til "eiendomsverdi". Takstmannens oppgave er å vurdere boligen sin tilsand. Dette skjer på en skala fra 1-3. Det er vi meglere som foretar prisvurderingen basert på tilbakemeldingen fra takstmannen og ved bruk av "Eiendomsverdi".

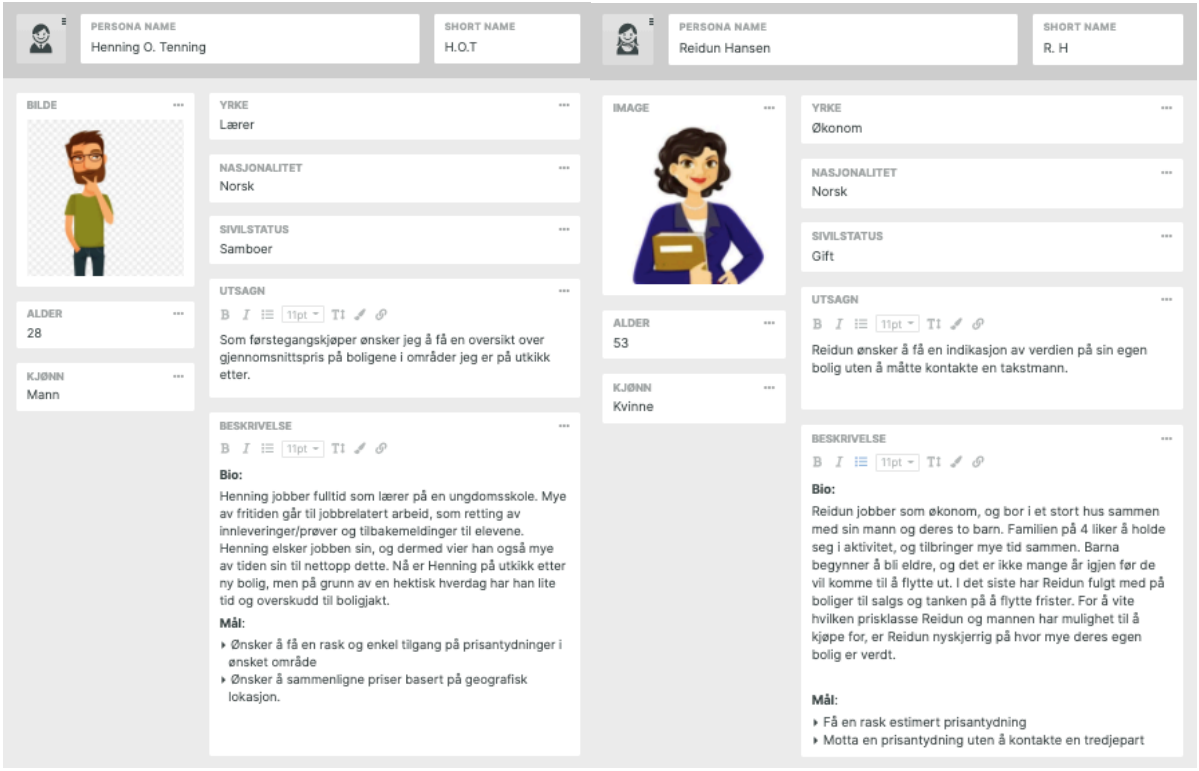
- **Tanker om vårt prosjekt?**

Prosjektet høres ganske likt ut som "Eiendomsverdi". Ettersom at dette programmet er så veletablert og brukt av eiendomsmeglere i hele landet har vi tillitt til det som allerede finnes. Da det fungerer så bra ville det vært vanskelig å utkonkurrere "Eiendomsverdi".

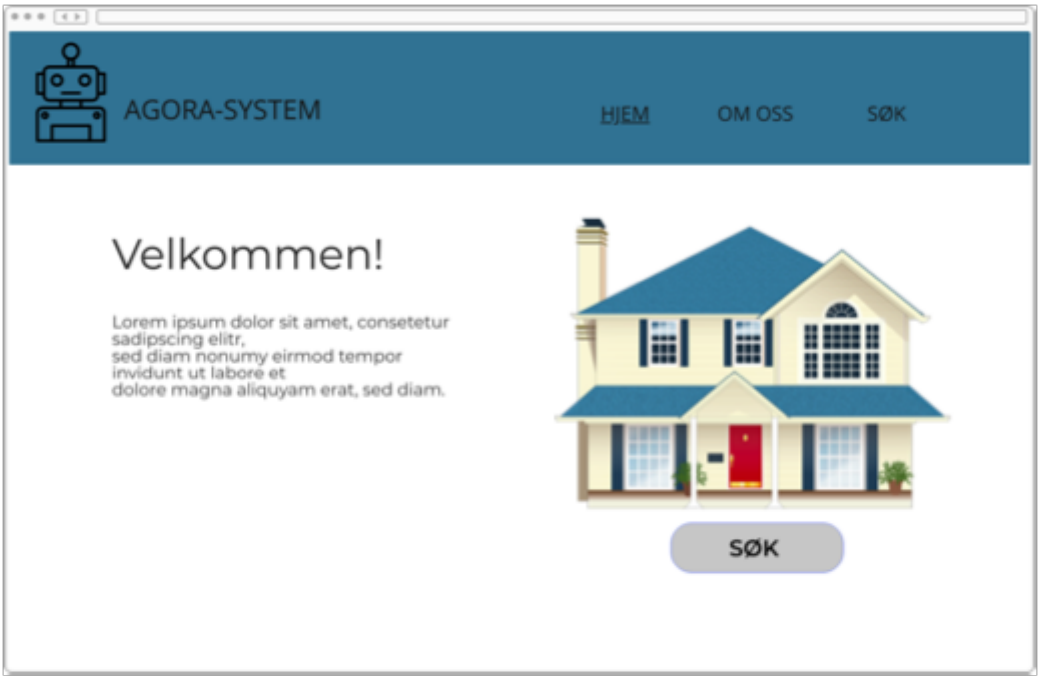
At dere skal rette systemet mot "mannen i gata", som ikke har tilgang til "eiendomsverdi" høres veldig smart ut. Jeg vet ikke om noe annet system som allerede finnes for folk som ikke er meglere. Det er ofte når jeg kommer på befaring at kunden er mest interessert i å finne ut hvor mye boligen deres er verdt. Det er per i dag både dyrt og et noenlunde stort steg å kontakte en eiendomsmegler bare fordi du er interessert i hvor mye boligen din er verdt. Det hadde vært gøy for kundene å kunne sjekke dette på egenhånd.

Smart å vinkle webapplikasjonen mot de som ikke er meglere. Jeg vet ikke om noe annet system som ligger tilgjengelig ute. Konseptet deres virker veldig kult, og jeg har hørt mange som vil sjekke prisen. Som regel er det dyrt og høyterskel. Hadde vært gøy for kundene å bare sjekke prisen enkelt og greit.

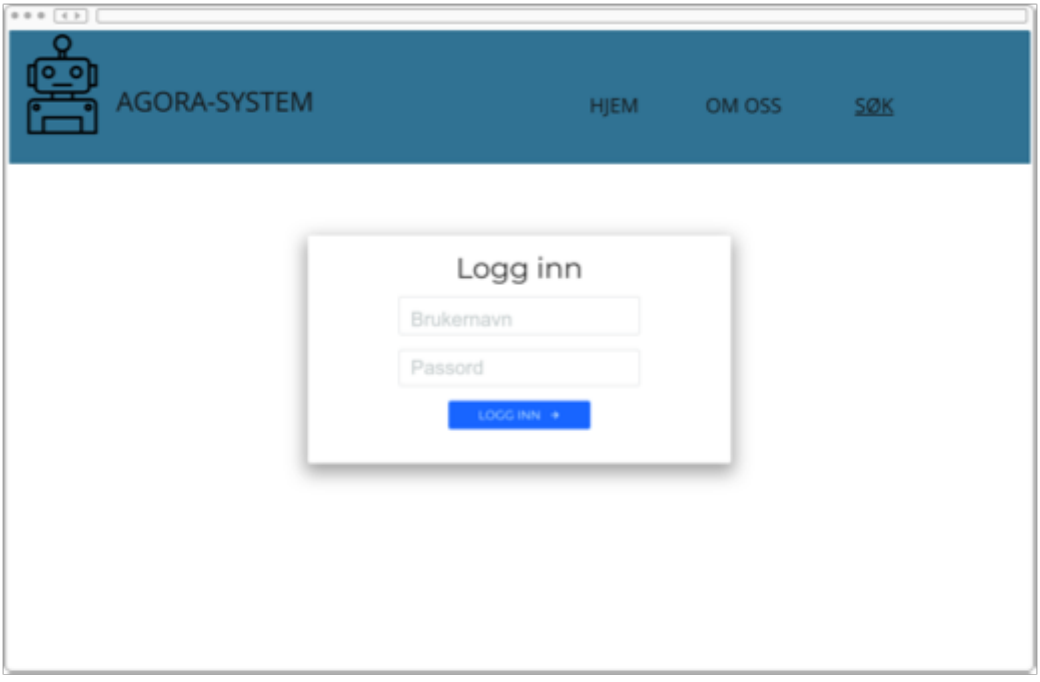
Vedlegg 10: Personas

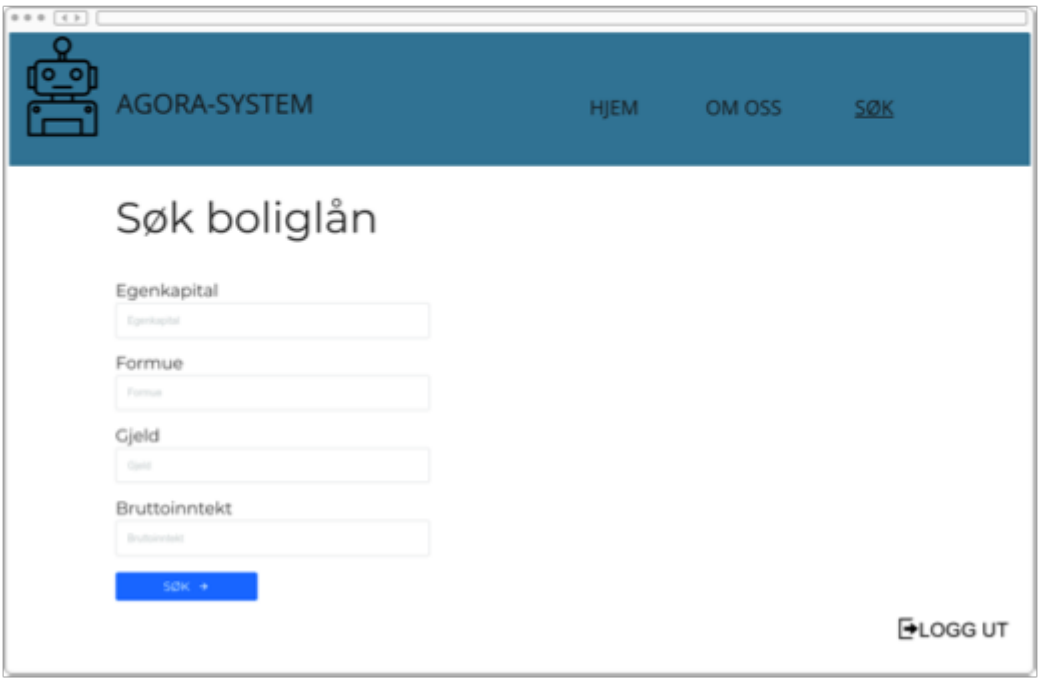


Vedlegg 11: Prototype

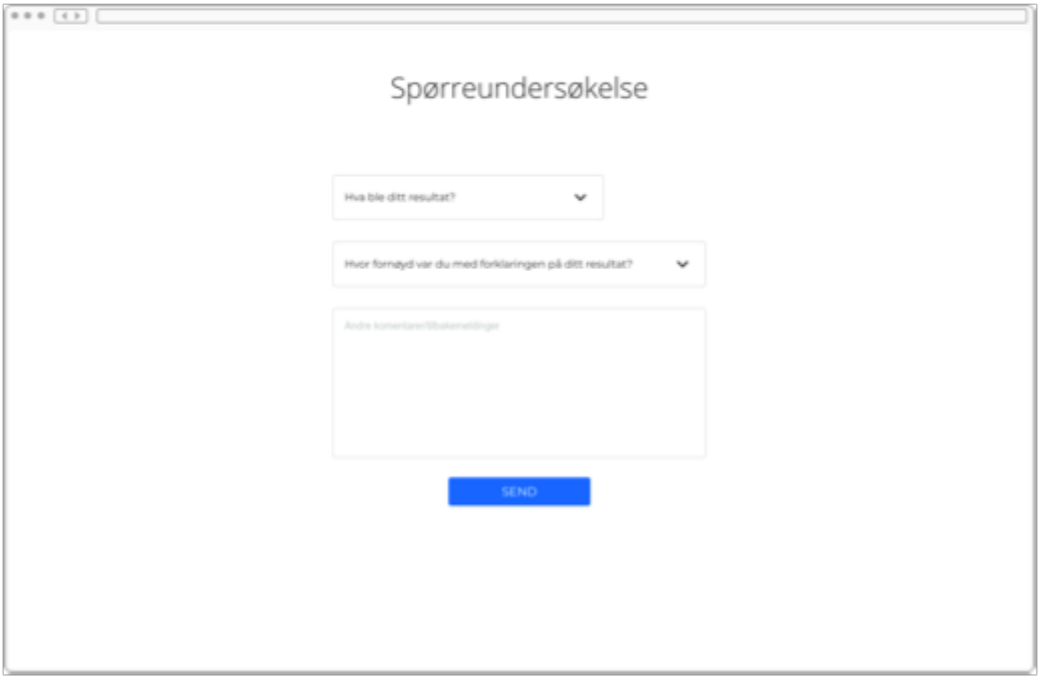


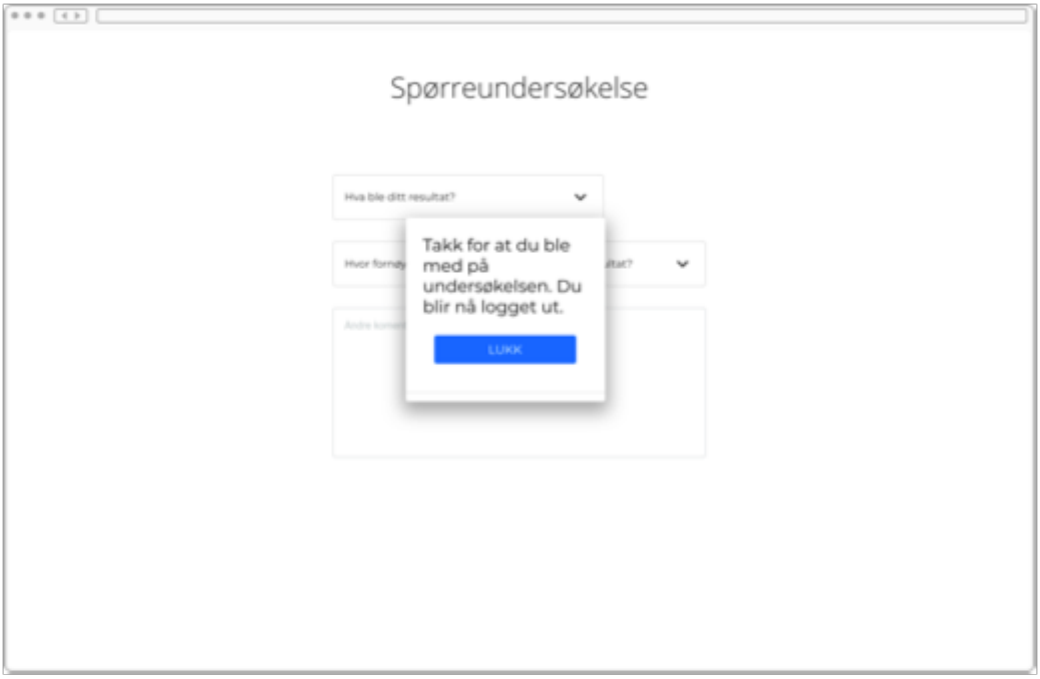

AGORA-SYSTEM
HJEM
OM OSS
SØK
OM OSS
Liten tekst om oss


AGORA-SYSTEM
HJEM
OM OSS
SØK
Logg inn
Brukernavn
Passord
LOGG INN


AGORA-SYSTEM
HJEM
OM OSS
SØK
Søk boliglån
Egenkapital
Formue
Gjeld
Bruttoinntekt
SØK
LOGG UT


AGORA-SYSTEM
HJEM
OM OSS
SØK
Søk boliglån
Egenkapital
600 000
Formue
1 500 000
Gjeld
500 000
Bruttoinntekt
600 000
GODKJENT
GÅ VIDERE
LOGG UT


Spørreundersøkelse
Hva ble ditt resultat?
Hvor fornøyd var du med forklaringen på ditt resultat?
SEND


Spørreundersøkelse
Hva ble ditt resultat?
Takk for at du ble med på undersøkelsen. Du blir nå logget ut.
LUKK

Vedlegg 12: Interesseskjema