# Refleksjon Oddvar prosjekt TDT4100:

Appen jeg har utviklet, basert på 'higher or lower' konseptet med NRK-data, oppfyller ikke bare kriteriene for appen, men bidrar også til forståelsen av flere pensumområder. Delegeringen av oppgaver er nøye gjennomtenkt, hvor hver klasse har sin spesifikke funksjon, og metodene innad i klassene fungerer som byggesteiner for å skape mer komplekse funksjonaliteter. I tillegg har jeg implementert to interfaces som benyttes i kombinasjon med Java streams, noe som har forbedret min forståelse og bruk av Java streams betraktelig.

Jeg la også inn observere for å starte en runde i spillet med en gang fillen var klar samt arv der jeg lagde er generell scraper først og utvidete til spesielle nettsider. Jeg kunne ha utforsket måte å gi GUI-en bedre respons med flere tråder under lasting, men GUI for meg var ikke en prioritet dette prosjektet.

I utviklingen av appen støtte jeg på utfordringer med å identifisere navn korrekt ved å bruke regulære uttrykk (regex), siden regex ikke effektivt kunne skille ut navn fra andre ord som begynner med stor bokstav. Etter å ha utforsket alternative løsninger, oppdaget jeg Natural Language Processing (NLP) og spesifikt CoreNLP-biblioteket, som jeg valgte å integrere i appen for mer presis navngjenkjenning.

Videre var jeg allerede kjent med Jsoup-biblioteket for parsing av HTML, som jeg vurderte å bruke på grunn av dets kraftige parsing-funksjoner. Imidlertid oppdaget jeg gjennom undersøkelser at nettverksforbindelsen via Jsoup kan være relativt treg. Jeg utforsket alternativer og fant ut at OKHTTP3-biblioteket tilbyr en raskere forbindelse, med betydelig lavere ping ved oppkobling. Dette ledet meg til å vurdere en kombinert tilnærming der OKHTTP3 brukes for nettverksforbindelsen, mens Jsoup fortsatt brukes for HTML-parsing, gitt dens effektivitet på dette området. Dette ble imitelrtid ikke gjort da det å lese seg opp på CoreNLP var tidkrevende nok. Derfor er hele scraping delen av appen laget med jsoupe som jeg er kjent med. Må sies at det var en bug på CoreNLP og modell interfacet som ikke ble fikset før i siste versjon, dette tok langt tid å finne ut av og jeg måtte grave dypt inne på forumer noe som var utrolig irriterende.

Når jeg reflekterer over Model-View-Controller-prinsippet, kan jeg si at databehandlingen i appen, spesielt i klassene Scraper og TestNLP samt Predicates, etter min mening, er utmerket utført. Imidlertid observerte jeg at da jeg programmerte HigherOrLower-klassen, begynte dens funksjonalitet å overlappe litt med kontrollerklassen. Det indikerer at kontrolleren kanskje har påtatt seg for mye ansvar, hvor jeg heller kunne ha overført mer av spillets logikk til HigherOrLower-klassen selv. Når det gjelder selve visningen av appen, investerte jeg ikke særlig mye arbeid i designet. Jeg la til noen bilder og pop-up-elementer, men for å være ærlig ser appen ut som om den kunne vært fra 2005. På den positive siden inneholder hovedapplikasjonen min ingen logikk, så visningsdelen er godt skilt ut fra resten

Testene ble utviklet på en inngripende måte. Her måtte jeg gå tilbake og endre deler av koden for å gjøre den mer testbar, noe som var svært lærerikt. Det viste seg at det var enda bedre med mindre funksjoner som bygger på hverandre enn det jeg opprinnelig trodde. Tidligere hadde jeg bygget for store funksjoner. Når det kom til å lage tester for scraping, og siden et testmiljø ikke skal påvirkes utenfra, valgte jeg å laste ned sidene jeg skrapet. Dette sikret at jeg alltid fikk samme resultat og kunne teste alle metoder bortsett fra connect-metoden, da den er statisk, dette fikk jeg ifra en Stackowerflow tråd. Det er viktig å designe et program med tanke på testbarhet mens man skriver det.

Under testingen av scraping-funksjonaliteten bekreftet jeg at den kunne identifisere elementene på nettsiden jeg valgte å skrape fra. Selv om jeg ikke kan garantere at absolutt alle sider følger samme struktur, har jeg i løpet av utviklingen av denne appen skrapet innhold fra en rekke sider på NRK og Wikipedia. Derfor velger jeg å anta at sidestrukturen generelt er lik. Dette kan imidlertid føre til et "svart svane"-problem, siden jeg ikke kan utelukke at det finnes unntak. Testene for NLP-delen er spesielle ettersom de involverer maskinlæring, og jeg har ikke trent en egen modell. Dette kan potensielt svekke nøyaktigheten. Jeg valgte å fokusere på spesifikke lette setninger for å teste de private metodene, som for eksempel createHashMap. Utfordringen med CORENLP er hvordan jeg kan verifisere at den gir tilfredsstillende resultater basert på de kriteriene jeg har satt. Siden jeg ikke har utviklet datasettet selv, blir det vanskeligere å måle nøyaktigheten eksakt.

Jeg må virkelig si at jeg har kost meg med dette prosjektet, ettersom det har vært utrolig gøy å kunne skape noe selv som faktisk kan brukes. Selv om appen min ikke er et utrolig spennende spill eller lignende, er det likevel tilfredsstillende å se resultatet.