**Utvikling for teateret ‘New Wave’ sitt bestillingsprogram**

Gjennom utviklingen av programmet som skal sørge for bestillinger hos teateret ‘New Wave’ har vi gått gjennom en rekke sprinter, hvor vi har laget flere utkast av programmet. Vi har støtt på problemer og endret store deler av koden fra sprint til sprint for å sørge for et mer intuitivt program, både for bruker og for utvikler.

**Sprint 1 (førsteutkast):**

Hoveddelen av sprint 1 gikk ut på å lage et ‘skjelett’ til videre utviklingen. Noe som resten av våre sprinter kunne basere seg ut på. Klasser ble lagd, funksjoner, datalagring, input, output, feilsjekking og mye mer. Gjennom utviklingen av sprint 1 benyttet vi både JSON og Excel ark som skulle sørge for datalagring. Dette var noe som var litt ugunstig til tider. Her er eksempelvis noe av det vi utviklet fra sprint 1, dette er datalagring ved hjelp JSON, hvor informasjonen skrives inn i en separat fil som et objekt, hvor det deretter kan tas ut igjen og benyttes permanent.

A computer screen shot of text

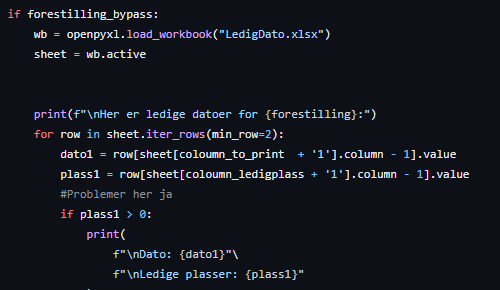
Description automatically generated

Mye av de viktigste grunnsteinene ble utviklet i sprint 1, og vi fikk virkelig satt godt i gang med kodingen.

**Sprint 2 (andreutkast):**

Sprint nr. 2 var et stort steg fra tidligere kode. Under sprint 2 endret vi datalagringssystemet fullstendig fra JSON til bare lagring via ‘openpyxl’ altså lagring i Excel-ark. Dette gjorde all lagring av bestilling og personer svært mye mer enkel. Dette førte til at både vi som utviklet programmet kunne lettere ha kontroll på lagret informasjon, og at bruker kan lettere ha tilgang til sine egne bestillinger. Vi utviklet et system som lot bruker hente ut sine egne bestillinger som er lagret, og bestille flere ganger ved å logge på sin tidligere lagde bruker.

**Eksempel**, her har vi skrevet kode som åpner informasjonen i regnearket som følger med koden, ‘LedigDato.xlsx’. Dette arket har kontroll på dato, ledige forestillinger, antall plasser, osv. Her printer vi ut alle de ledige datoene som ligger inne i excel-arket, de datoene som er utsolgt, vil ikke printes ut.



Samtidig, under sprint 2 støtte vi på et av de aller største problemene hittil, som var lagringen av flere bestillinger innenfor et og samme excel-ark, som hver bruker får tildelt når de lager en bruker. Alt annet fungerte svært godt ved slutten av sprint 2, utenom dette lille problemet.

**Sprint 3 (tredjeutkast):**

I løpet av sprint 3 gjorde vi ikke svært mange endringer, hoveddelen av sprint 3 gikk ut på å finne løsninger for det lille problemet med datalagring som vi støttet på i sprint 2. Vi fant ingen løsning og flyttet dette videre til sprint 4. Noe annet som skjedde i sprint 3 var at vi utviklet et system som beregnet profitt, og om profitten var stor nok, ville skuespillet ‘Small Talk’ også bli lagt til i tillegg til de originale to, ‘Vildanden’ og ‘De Elendige’. Under er vedlagt en del av koden fra sprint 3, som la til ‘Small Talk’ om profitten ble stor nok

A computer screen with white text

Description automatically generated

Denne biten av kode sier at om profitten er tørre eller lik 5000, vil det føre til at ‘Small Talk’ legges til, dette defineres lengere nede i koden, utenfor denne biten med kode.

Det eneste problemet som gjensto etter sprint 3 var å fikse det problemet med flere bestillinger per bruker, som vi støttet på tidlig, allerede i sprint 2. Dette problemet lå hovedsakelig i at når vi skulle lese av regnearket hadde vi et bestillingsnummer, dette bestillingsnummeret klarte ikke koden å lese og dermed ble det returnert en NoneType. Dette førte til at bestillingsnummeret forble som om det var brukeren sin første bestilling og gjorde til at man ikke kunne legge til en til bestilling, men at koden skrev over den gamle bestillingen. Under er koden vi hadde problemer med vedlagt

A computer screen shot of a program

Description automatically generated

Vi valgte å ta dette problemet videre til sprint 4.

**Sprint 4 (fjerdeutkast/endelig program):**

Denne delen av utviklingen gikk hovedsakelig ut på å rydde opp i det problemet vi hadde støtt på siden sprint 2, med datalagringen for flere bestillinger. Dette problemet klarte vi ikke å fikse og dermed måtte ta et valg om å la det være slik det er, hvor hver bruker kan bare bestille en gang, og etter det så vil neste bestilling bare overskrive den tidligere. Vi skulle gjerne ha klart å utvikle en bedre løsning, men dette problemet tok oss såpass mye tid at vi ikke klarte å tenke ut noe annet, dermed endte vi opp med en litt verre løsning enn planlagt.