

Delfinen

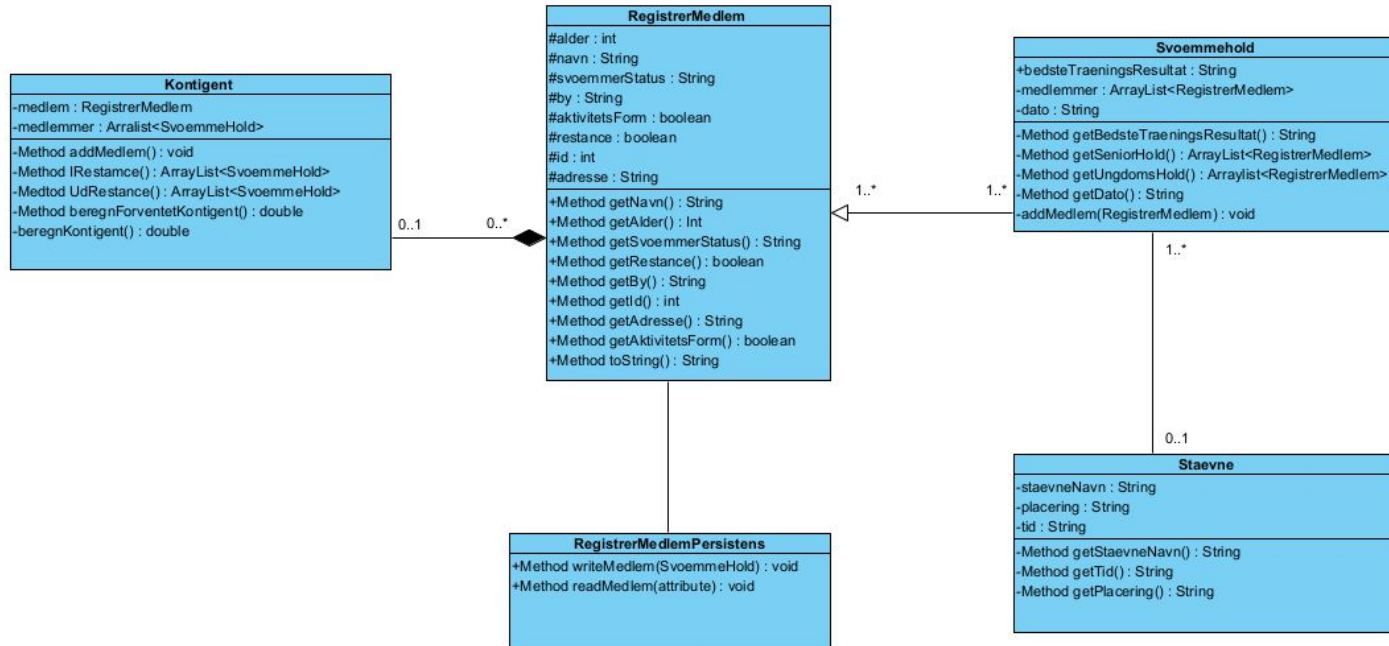
Gruppe 10

Anton(BBlazee), Andy (Aypn69),
Asger (askau2003), Simon (Sighgen)
og Chris (Granroyal)

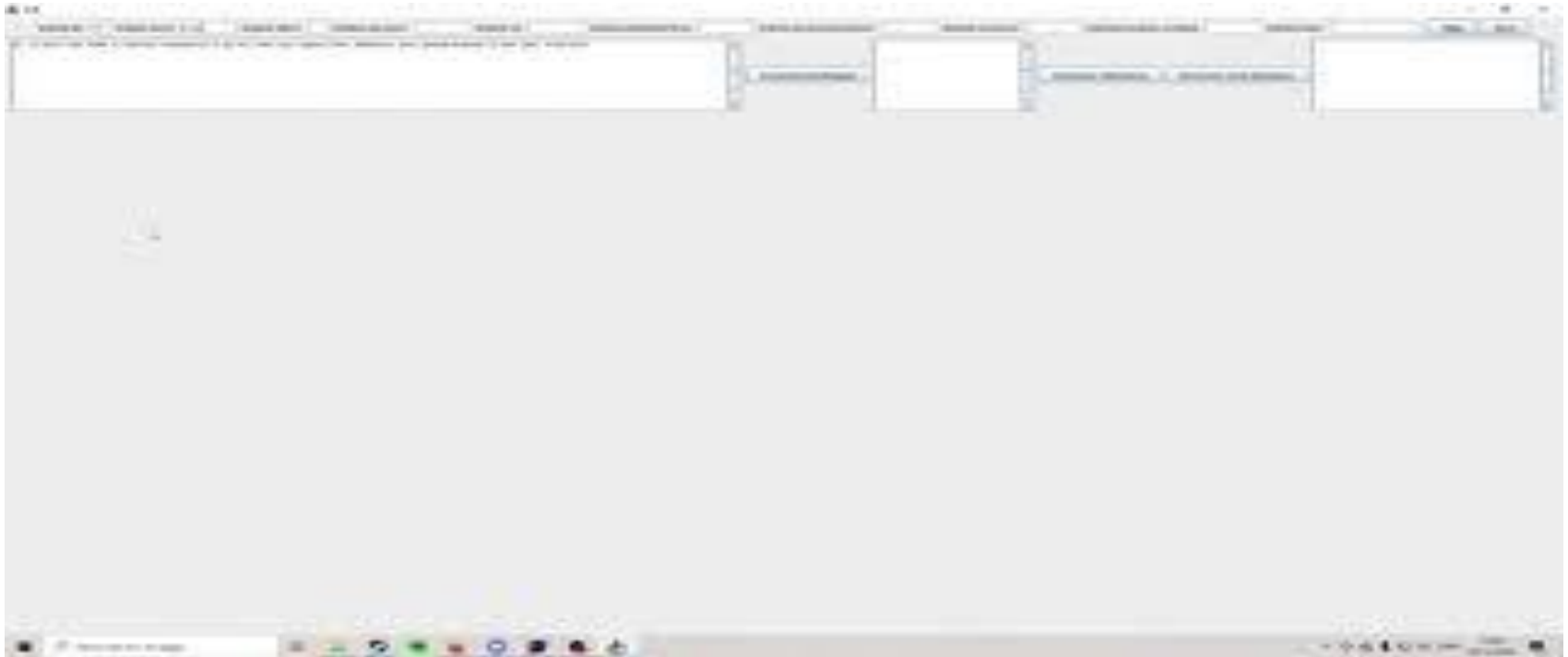


<https://github.com/askau2003/Delfinen>

Klassediagram



Introduktion af Demo



User Stories

Medlemsregistrering

1. Som formand ønsker jeg **at kunne registrere nye medlemmer** med deres stamoplysninger, så vi kan holde styr på medlemskabet.
2. Som medlem ønsker jeg **at kunne vælge mellem aktivt eller passivt medlemskab og angive alder**, så mit medlemskab kan tilpasses.

Kontingentadministration

3. Som kasserer ønsker jeg **at kunne beregne og registrere kontingent baseret på medlemskabsstatus og alder**, så jeg kan holde styr på betalinger.
4. Som kasserer ønsker jeg **en samlet oversigt over forventede kontingentindtægter**, så jeg kan planlægge klubbens økonomi.
5. Som kasserer ønsker jeg **en liste over medlemmer i restance**, så jeg kan følge op på manglende betalinger.

Konkurrenceadministration

6. Som træner ønsker jeg **at kunne registrere og opdatere træningsresultater for hver svømmedisziplin**, så jeg kan vurdere svømmernes udvikling.
7. Som træner ønsker jeg **en oversigt over klubbens top 5 svømmere pr. disciplin fordelt på junior og senior**, så jeg kan udvælge deltagere til konkurrencer.

Systemfunktionalitet

8. Som bruger ønsker jeg **en simpel grafisk brugergrænseflade**, så jeg kan interagere med systemet på en enkel måde.
9. Som udvikler ønsker jeg, **at data gemmes i filer**, så information kan bevares mellem sessioner.

SCRUM-tasks + Story Points

Medlemsregistrering

1. Design og implementer funktion til at registrere nye medlemmer. —————> 5 points
2. Implementer valg af medlemskabstype (aktiv/passiv). —————> 2 points
3. Tilføj validering for alder og medlemskabstype. —————> 3 points

Kontingentadministration

4. Udvikl kontingentberegningsfunktion baseret på medlemsdata. —————> 5 points
5. Implementer funktion for samlet kontingentindtægt. —————> 5 points
6. Udvikl listefunktion for medlemmer i restance. —————> 3 points

Konkurrenceadministration

7. Implementer registrering af træningsresultater pr. disciplin. —————> 8 points
8. Udvikl oversigtsfunktion for top 5 svømmere pr. disciplin. —————> 8 points

Systemfunktionalitet

9. Design grafisk brugergrænseflade. —————> 5 points
10. Implementer simpel filbaseret datalagring i .txt format. —————> 3 points

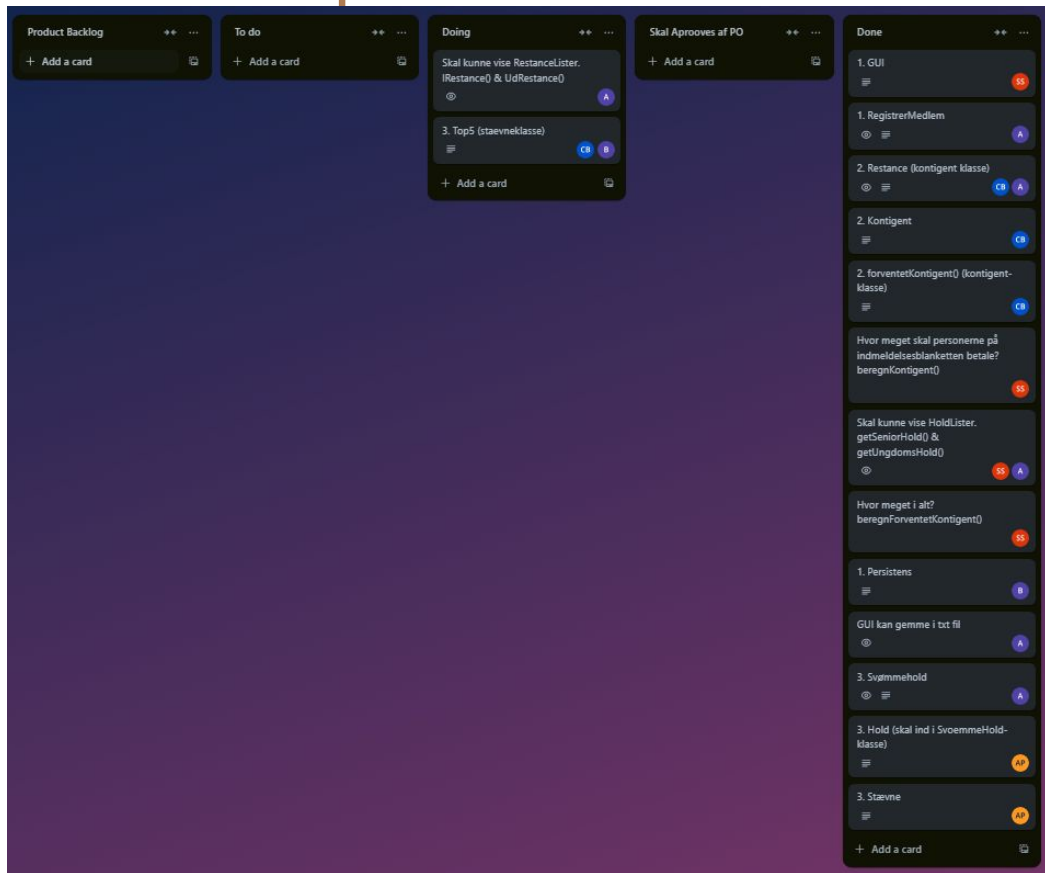
Story Points i alt: —————> = 47 points

Trello Board after Sprint 1

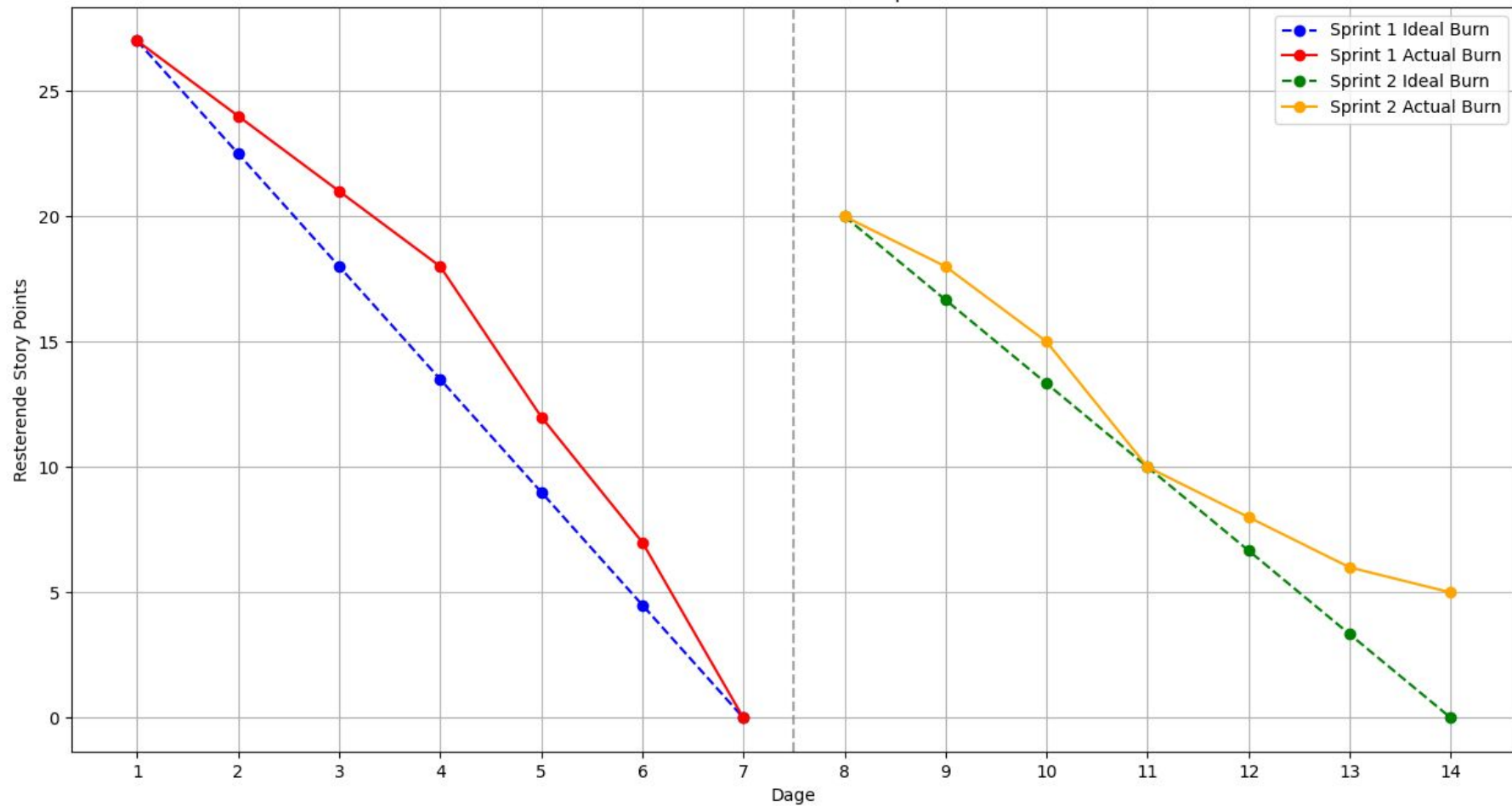
The Trello board is organized into five columns, each representing a stage in the sprint cycle. The cards are as follows:

- Product Backlog**
 - Skal kunne vise RestanceLister. IRestance() & UdRestance() (Label: A)
 - 3. Top5 (staevneklasse) (Labels: CB, B)
 - Skal kunne vise HoldLister. getSeniorHold() & getUngdomsHold() (Labels: SS, A)
 - GUI kan gemme i txt fil (Label: A)
- To do**
 - + Add a card
- Doing**
 - 1. GUI (Label: SS)
 - Hvor meget skal personerne på indmeldelsesblanketten betale? beregnKontigent() (Label: SS)
 - Hvor meget i alt? beregnForventetKontigent() (Label: SS)
 - 1. Persistens (Label: B)
- Skal Approves af PO**
 - + Add a card
- Done**
 - 1. RegistrerMedlem (Label: A)
 - 2. Restance (kontigent klasse) (Labels: CB, A)
 - 2. Kontigent (Label: CB)
 - 2. forventetKontigent() (kontigent-klasse) (Label: CB)
 - 3. Svømmehold (Label: A)
 - 3. Hold (skal ind i SvoemmeHold-klasse) (Label: AP)
 - 3. Stævne (Label: AP)

Trello Board after Sprint 2



Burndown Chart Over 2 Sprints



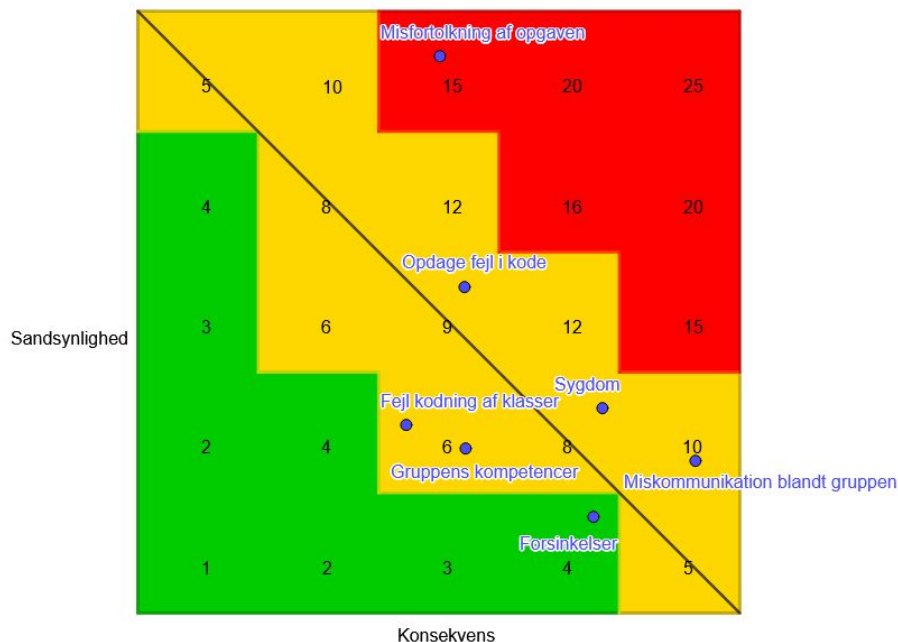
Interessentanalyse

Interessent (hvem)	Hvad og hvornår	Hvordan	Ansvarlig
Ledelse	<ul style="list-style-type: none"> - Bedre service til deres Medlem - Spare tid/Spare penge - Sikker betaling - Bedre overblik over økonomien 	<ul style="list-style-type: none"> - Kommunikere med statusrapporter med opdateringer om projektets fremdrift, risici og succeser. - Afholde præsentationer for at demonstrere forbedringer eller besparelser, som projektet har skabt. 	<ul style="list-style-type: none"> - Projektlederen, som sikrer, at alle nødvendige detaljer deles med ledelsen.
Medlemmer	<ul style="list-style-type: none"> - Mere convenient og effektivt system (Så medlemmer har bedre styr på, hvad de skylder og hvornår). 	<ul style="list-style-type: none"> - Udsendelse af nyhedsbreve med opdateringer om nye funktioner eller forbedringer i systemet. - Afholdelse af informationsmøder, hvor medlemmer kan stille spørgsmål og give feedback. 	<ul style="list-style-type: none"> - Produktowner i samarbejde med kommunikationsafdelingen.
Kasserer	<ul style="list-style-type: none"> - Mere automatiseret system angående kontingenter 	<ul style="list-style-type: none"> - Implementere et dashboard for at give løbende overblik over automatiseret kontingenthåndtering. - Tilbyde træning i brugen af det nye system via workshops. 	<ul style="list-style-type: none"> - Projektlederen for opsætning, og supportteamet for efterfølgende hjælp.
Konkurrencesvømmere	<ul style="list-style-type: none"> - At kunne registrere trænings resultater og dato - Mere convenient og effektivt system 	<ul style="list-style-type: none"> - Bruge hjemmesiden til at registrere og tracke træningsresultater. - Gennemføre feedbacksessions for at sikre, at systemet opfylder deres behov. 	<ul style="list-style-type: none"> - Produktowner og udviklingsteamet.
Træner	<ul style="list-style-type: none"> - Bedre oversigt over top 5 svømmediscipliner 	<ul style="list-style-type: none"> - Tilbyde detaljerede analyser af svømmernes præstationer via et rapporteringsværktøj. - Holde workshops for at demonstrere, hvordan data kan bruges til træningsoptimering. 	<ul style="list-style-type: none"> - Produktowner og teknisk træner
Kommunen	<ul style="list-style-type: none"> - Budget - Kontingent - Effektivt system 	<ul style="list-style-type: none"> - Fremsende kvartalsvise rapporter om projektets bidrag til budget, kontingentsstyring og systemets effektivitet. - Invitere til fremvisninger af systemet for at styrke samarbejdet. 	<ul style="list-style-type: none"> - Projektlederen i tæt samarbejde med økonomiteamet.

Risikoanalyse

Risiko	Sandsynlighed	Konsekvens	Score	Håndtering
Forsinkelser	4	1	4	<p>Risikohåndtering: Hvis en forsinkelse opstår i form af at man har sovet for længe, man skal til lægen eller lignende har vi muligheden for at benytte discord som portal til kommunikation. Derudover har vi også muligheden for at udskyde møder til senere tidspunkter.</p> <p>Mitigerig: Omtrent 20-21 skrives der sammen om folk har nogle planer i løbet af næste dag + der skal huskes at sættes ur.</p>
Sygdom	4	2	8	<p>Risikohåndtering: Ved opståelse af sygdom benytter vi os af Discord som portal for kommunikation. Ved akut sygdom eller sygdom som der påvirker ens arbejdsevne i en højere grad aftales det internt i gruppen angående fordelingen af arbejdet, samt opsummering af fremgangen.</p>
Gruppens Kompetencer	2	3	6	<p>Risikohåndtering: Hvis kompetencerne i gruppen forhindrer os i at fortsætte på opgaven vil vi som udgangspunkt håndtere problemet i følgende rækkefølge:</p> <ol style="list-style-type: none"> 1. Spørge internt i gruppen om hjælp, hvor vi herfra deler vores individuelle erfaring. 2. Lave en google søgning i form af forskellige hjemmesider såsom: w3schools 3. Spørge en lærer.
Opdage fejl i koden som gør den mindre robust	3	3	9	<p>Risikohåndtering: Vi er så begavede at product owneren kan finde ud af at kode, og vi bruger derfor ham som fejlnet.</p> <p>Mitigerig: Til de daglige scrum møder har man mulighed for at spørge de andre i gruppen om hjælp hvis man er i tvivl.</p>
Miskommunikation blandt gruppen	5	2	10	<p>Risikohåndtering:</p> <ol style="list-style-type: none"> 1. Finde én overensstemmelse og gå ud fra dette 2. Hvis (1.) ikke er muligt vil vi derefter benytte os af vores lærere. <p>Mitigerig: Sørge for at spørge ind hvis man er i tvivl om det mindste.</p>
Misfortolkning af Opgaven	3	5	15	<p>Risikohåndtering: Vi sætter os ned og kigger det igennem som gruppe.</p> <p>Mitigerig: Sørge for alle har læst opgaven grundigt, så der er større chance for at vi undgår mistolkninger.</p>
Fejl kodning af klasser	2	3	6	<p>Risikohåndtering: Hvis kompetencerne i gruppen forhindrer os i at fortsætte på opgaven vil vi som udgangspunkt håndtere problemet i følgende rækkefølge:</p> <ol style="list-style-type: none"> 1. Spørge internt i gruppen om hjælp, hvor vi herfra deler vores individuelle erfaring. 2. Lave en google søgning i form af forskellige hjemmesider såsom: w3schools 3. Spørge en lærer.

Tilhørende risikomatrix



Højrisikoområdet (rød zone):

"Misfortolkning af opgaven" skal adresseres hurtigt og proaktivt for at minimere problemer.

Moderate risici (gul zone): Fejl i kode, kommunikation og sygdom kræver forebyggende tiltag og planlægning for at mindske deres påvirkning.

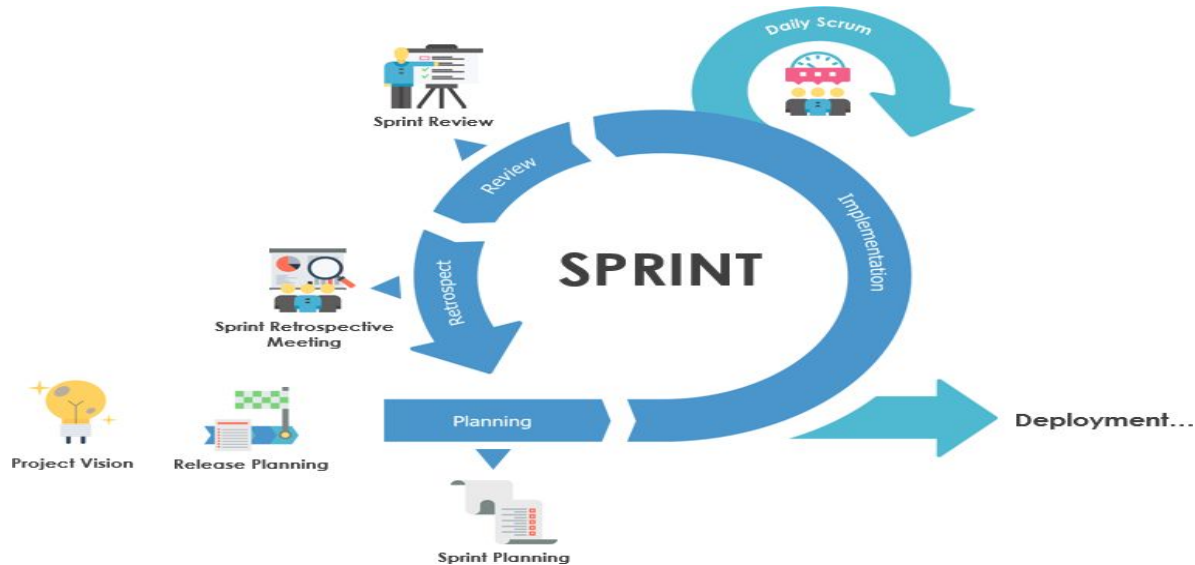
Lavrisikoområdet (grøn zone):

Kompetencer og forsinkelser kræver minimal opmærksomhed, men bør overvåges løbende

Udviklingsproces

Anton

- Scrum master - Chris
- Product Owner - Asger



Konklusion

På baggrund af dette projekt kan vi konkludere, at der har været mere kode at skrive, som skulle integreres og arbejde sammen. Vi valgte en tilgang, hvor teammedlemmer kunne tage opgaver fra Trello-boardet efter behov. Desuden har vi prioriteret at udvikle en GUI.

Vi kan også konkludere, at projektet krævede, at deltagerne sikrede, at de relevante attributter og metoder blev skrevet korrekt. Dette var afgørende for, at de forskellige klasser kunne fungere sammen og blive integreret problemfrit, når de blev merged via GitHub.

Andre vigtige erfaringer fra projektet:

- Tillid til projektgruppen.
- Øget selvstændighed.
- Sparring og samarbejde.

Med hensyn til sprints valgte vi at have et sprint i 2 dele i stedet for at have et sprint hver dag (som i sidste projekt). Denne tilgang fungerede godt, da den gav mere fleksibilitet og øget effektivitet, eftersom gruppemedlemmer kunne arbejde, når det passede dem bedst. Vi erfarede dog, at vi ikke fik opdelt vores tasks nok til at sikre en ligelig arbejdsfordeling og et glidende tempo.