

Gruppe 2

# Premier league Projekt

12-9-2015

## Content

Gruppemedlemmer .....	1
Indledning.....	2
Use-cases .....	2
Brief use-cases .....	2
Fully Dressed Use Case: Compare Salary and Effectiveness .....	3
Supplerende kravliste .....	4
Valg truffet undervejs.....	4
Særligt interessante dele af koden.....	4
Problemer .....	4
Beskrivelse af systemafprøvelse.....	5
Ansvarsområder for programmet: .....	5
Faseplan.....	5
SWD Diagrammer .....	6
Use case Diagram .....	6
Class diagram.....	7
Sekvensdiagrammer .....	7
Kommunikationsdiagrammer .....	9
Klassediagram.....	11
Kildeliste .....	12

## Gruppemedlemmer

Frederik Blixt	120594
Daniel Krell Refn	100895
Jonas Tvede Henriksen	260495
Fredrik Bjørklund	080787

## Indledning

(Daniel)

I denne opgave laver vi et program, dette program skal kunne se om en høj løn hænger sammen med en høj målscore, ud over dette skal programmet også kunne check hvilken lineups der virker bedste mod diverse hold i premier league og se de tre spillere på holdet der har den største indflydelse på spillet.

## Use-cases

(Frederik Blix og Jonas)

### Check the team roster:

User wants to check the roster of a team as well as the salary and statistics of each player (including data such as how many wins and losses the player has participated in).

### Check team schedule:

User wants to check the schedule of the team (including previous matches).

### Check match statistics:

User wants to check statistics of a match.

### Compare salary and effectiveness:

User wants to examine the connection between the salary and performance of a player.

### Positive Reinforcement:

User wants to see a top 3 of players who have participated in the most wins of the team

### Line-up planning:

User wants to see (based on match results) which line-up is the best against specific teams.

## Brief use-cases

(Frederik Blixt)

### Check the team roster:

- User opens the program, selects the roster menu. User will then get a list of all the players the list can be sorted by name or position on the field. User can then click on a player to get a salary and the statistics of the player. If the desired player is the goal keeper, his saving statistics will be displayed.

### Check team schedule:

- User opens the program and selects schedule. A diagram showing all the matches of the season will appear, making it possible also to check previous matches.

#### Check match statistics:

- User opens the program and selects match statistics. All matches played will be displayed along with a number. The user can then dial in this number to get the statistics of the desired match.

#### Positive Reinforcement:

- The user opens the program and selects reinforcing statistics. The user selects top 3 winning players of this season. The user gets an overview of the win ratio of the players of the team.

#### Line-up planning:

- The user opens the program and selects reinforcing statistics. The user selects line-up planning. The user gets an overview of the matches played versus opposing teams along
- with the end result and the lineup used in set match.

## Fully Dressed Use Case: Compare Salary and Effectiveness

(Fredrik Bjørklund)

Primary Actor: Manager

#### Stakeholder&Interests:

Manager: Needs easy access to the team's salary, the players salary and players statistics data, along with tools to calculate upon these data to further enhance the effectiveness of the team.

#### Preconditions:

The team has played a match and the statistics have been recorded.

#### Main Success Scenario:

1. Manager opens up the system.
2. He gets prompted with available players.
3. From list he then selects players he wants to compare.
4. Systems gives him feedback and key performance measurements of selected players.
5. Manager exits system when he is done analysing.

#### Extensions:

##### 1a. System is offline

1. Manager contacts IT-Support.
2. IT-Support restores system.

##### 2a. Manager compares a player to itself

1. Manager is prompted with an error message.
2. System returns to selection.

##### 3a. Manager attempts a comparison using a player without any statistics yet

1. Manager is informed about the lacking data on the player

## 2. System returns him to selection screen

### Special Requirements:

- Usability: Manager needs to be able to access the system and use it as a first time user.
- Stability: 99% uptime.

## Supplerende kravliste

(Jonas)

- At oprette et login til programmet kunne være en mulig komponent at lave, da det sikre at andre ikke ville kunne ændre fx. de tekst filer som programmet i høj grad er afhængig af.
- En interessant usecase at tilføje kunne være en funktion der viser modstandernes top spillere så man på den måde som bruger af systemet kan få information om hvilke af modstanderholdets spillere man skal være særligt opmærksom på.
- Ud fra hvilke spillere der score flest mål mod hvem kunne der laves en funktion der på baggrund af de data beregner den bedste hold sammensætning mod hvem.

## Valg truffet undervejs

(Frederik Blixt)

Vi som gruppe har valgt at ikke implementere en stor del Frederik Blixts kodning ind i det endelige program. Dette betyder ikke, at det ikke er der, men at det er lagt sammen i nogle andre metoder såsom "printplayers" i klassen PlayerAnalytics. Vi har valgt at gøre det på denne måde da vi ikke syntes at det var relevant at bare kunne printe alle spillere i truppen samt at det ikke var nødvendigt at dedikere en hel klasse til noget en funktionelt nærliggende metode kunne ordne. Ud over delen der nu ligger i PlayerAnalytics lavede han også en Schedule klasse. Denne klasse har vi valgt ikke at arbejde videre med på nuværende tidspunkt men det vil bestemt være en muligt at tilføje den til programmet på et senere tidspunkt.

## Særligt interessante dele af koden

(Jonas og Fredrik Bjørklund)

Da det meste af programmet er bygget op omkring streaming af data fra en fil ind i systemet, er klassen FileOperations og klassen lavet på baggrund af denne, FileOperationsLineUps, særligt interessante. Det er disse der står for at læse data fra tekstfiler som så streames ind i en ArrayList som så bruges i programmets metoder og variabler. Klasserne skilder teksten fra tekstfilen ved kommaer, så hver streng mellem kommaerne kan sættes ind i en konstruktør for et objekt. På den måde kan programmets data styres af brugeren som kan skrive et ønsket datasæt ind i tekstfilen og derved bruge det i programmet, uden at skulle programmere noget ind i det selv.

## Problemer

(Jonas)

Vores "Line Up Analytics" funktion endte med ikke at kunne checke alle instanser af vores holds strategi mod modstandernes hold strategi. Som det er nu, løser programmet gennem en fil indeholdende spillede kampe, og finder de kampe, hvor slutresultatet var en sejr til vores hold samt at strategien for begge hold

var offensiv. Derudover er der også kun lavet metoder for 3 modstanderholds navne og ikke alle Premier Leagues deltagende hold. Det er ikke et problem vi ikke kunne løse, men nærmere et problem vi følte at vi løste forkert. Da tiden var knap, valgte vi at fokusere anderledes og i stedet assimilere hvordan denne funktion af programmet fungerer. Vi kunne nemlig godt have masse kopieret de metoder vi bruger nu til at kunne tælle alle de mulige strategi kombinationer mod alle de mulige modstandere. Dette gjorde vi ikke, da vi fandt det ineffektivt at kopiere de næsten identiske metoder igen og igen og derved lave vores classes meget store.

## Beskrivelse af systemafprøvelse

(Daniel)

Undervejs i udviklingen af softwaren er der løbende blevet tjekket op på metodernes udførelse. Altså har det oftest aldrig været nok at kunne compilere en klasse, men klassens metoder er blevet afprøvet således at metodernes funktioner altid har virket korrekt, før en ny metode med reference til den gamle påbegyndes. Det er f.x ved 'Tre mest betydningsfulde spillere' blevet afprøvet ved at oprette en midlertidig main metode, der sørger for at metoden har relevant input at blive afprøvet med.

## Ansvarsområder for programmet:

(Daniel)

Klasser

- Controllers
  - FileOperations (Fredrik Bjørklund)
  - FileOperationsLineUps (Fredrik Bjørklund, redigeret af Jonas)
  - LineUpAnalytics (Jonas)
  - PlayerAnalytics (Daniel)
  - SalaryAnalytics (Frederik Blixt)
- Main
  - Run (Frederik Bjørklund)
- Model
  - Keeper (Daniel)
  - Player (Daniel)
  - Match (Jonas)
- View
  - InputReader (Fredrik Bjørklund, Jonas)

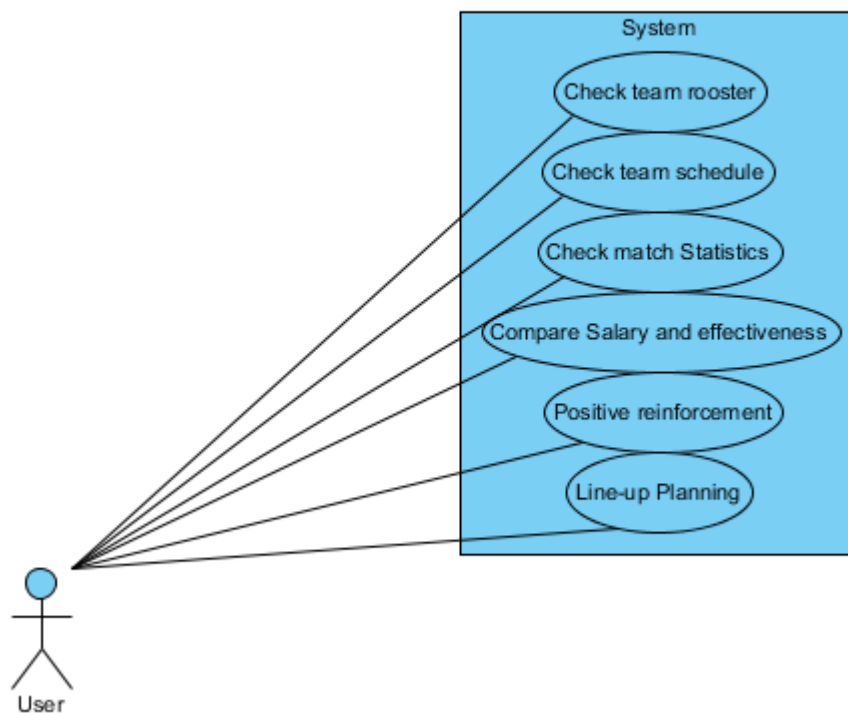
## Faseplan

- Til fredag d. 20 november regner vi med at være færdige med System Sequence Diagrams og Use Cases.

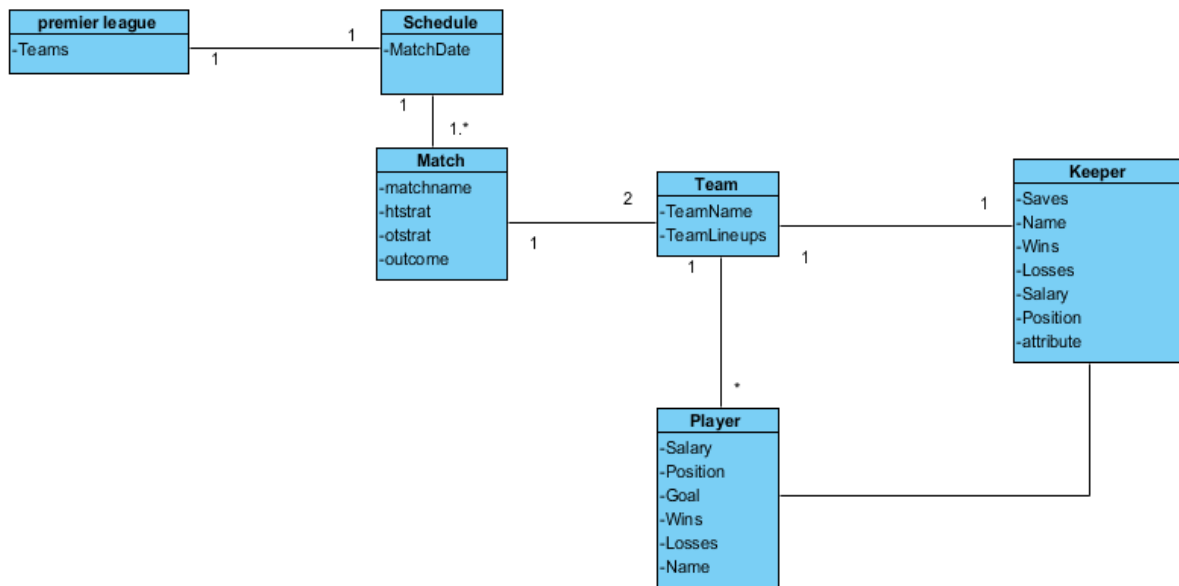
- Til onsdag d. 25 november regner vi med at kunne komme i gang med at lave kode delen af opgaven, da det er der et af vores gruppe medlemmer kommer hjem fra ferie.
- Tirsdag d. 1 december regner vi med at finpudse SWD delen.
- Til fredag d. 4 december regner vi med at være færdige med vores individuelle dele af koden og derefter kunne sætte det sammen til et program.
- Resten af tiden til onsdag d. 9 december skal bruges på at finpudse projektet som helhed.

## SWD Diagrammer

### Use case Diagram

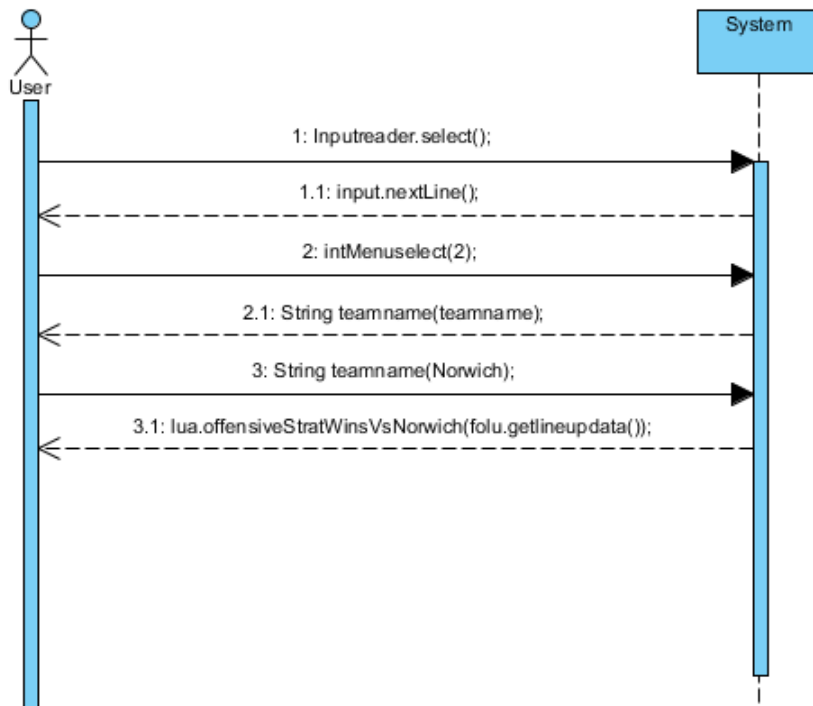


## Class diagram



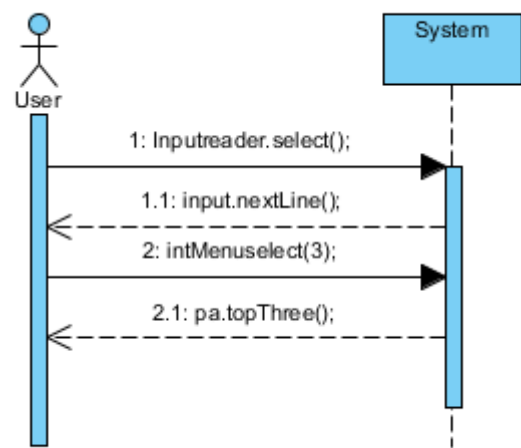
## Sekvensdiagrammer

Check MatchStats

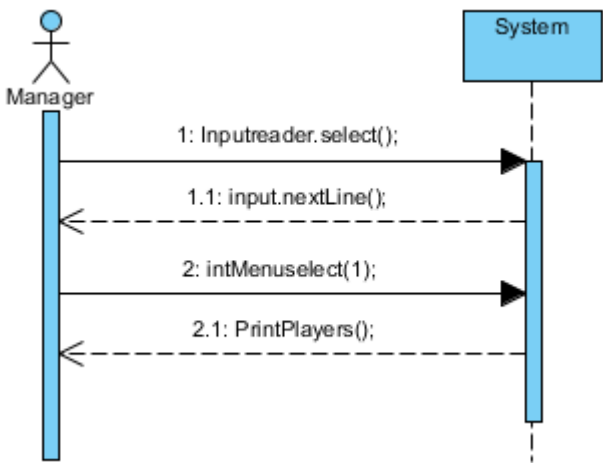




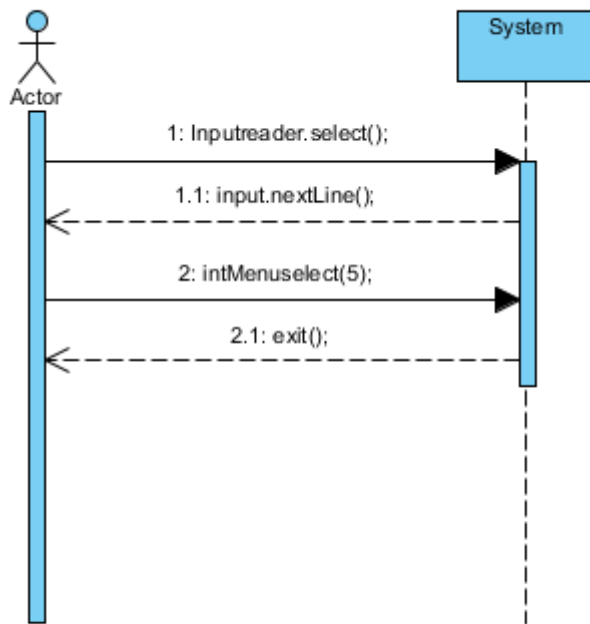
Check Topthree



SalaryAnalytics

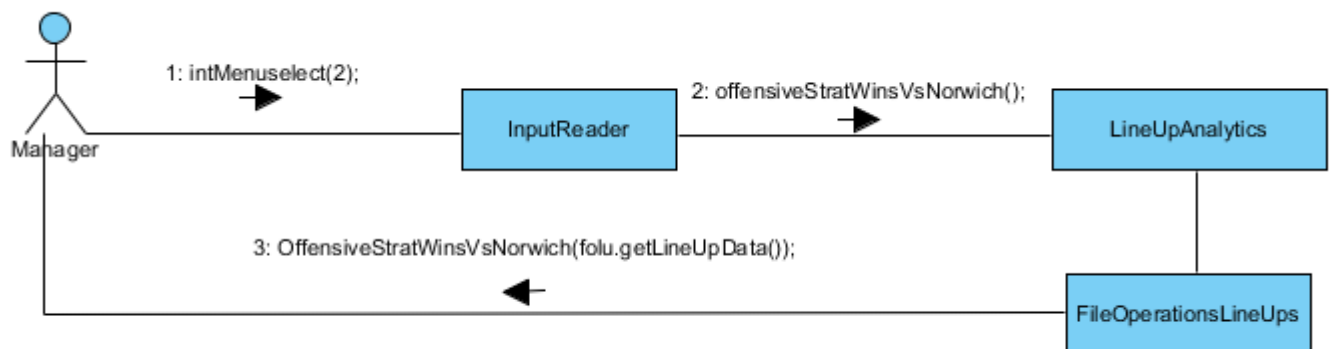


Exit program

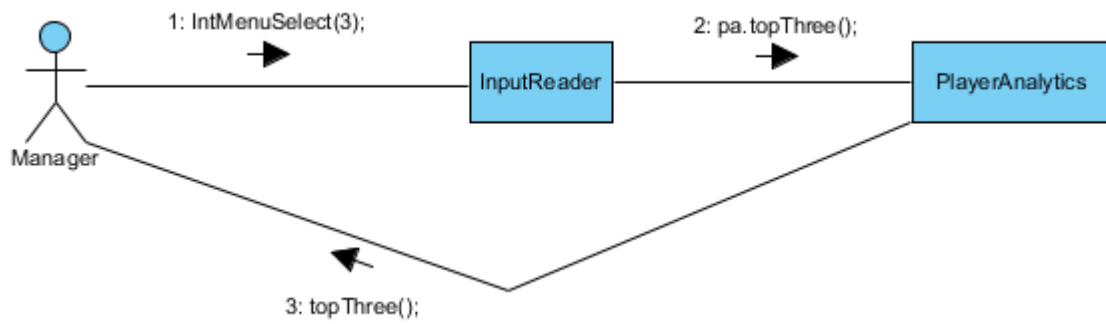


## Kommunikationsdiagrammer

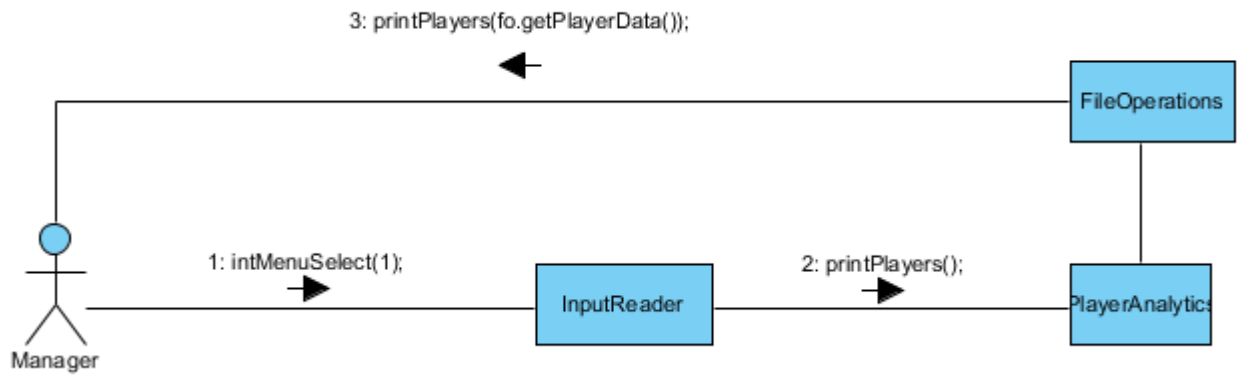
Check MatchStats



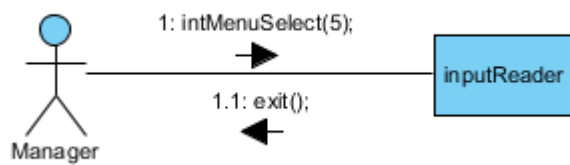
### Check Topthree

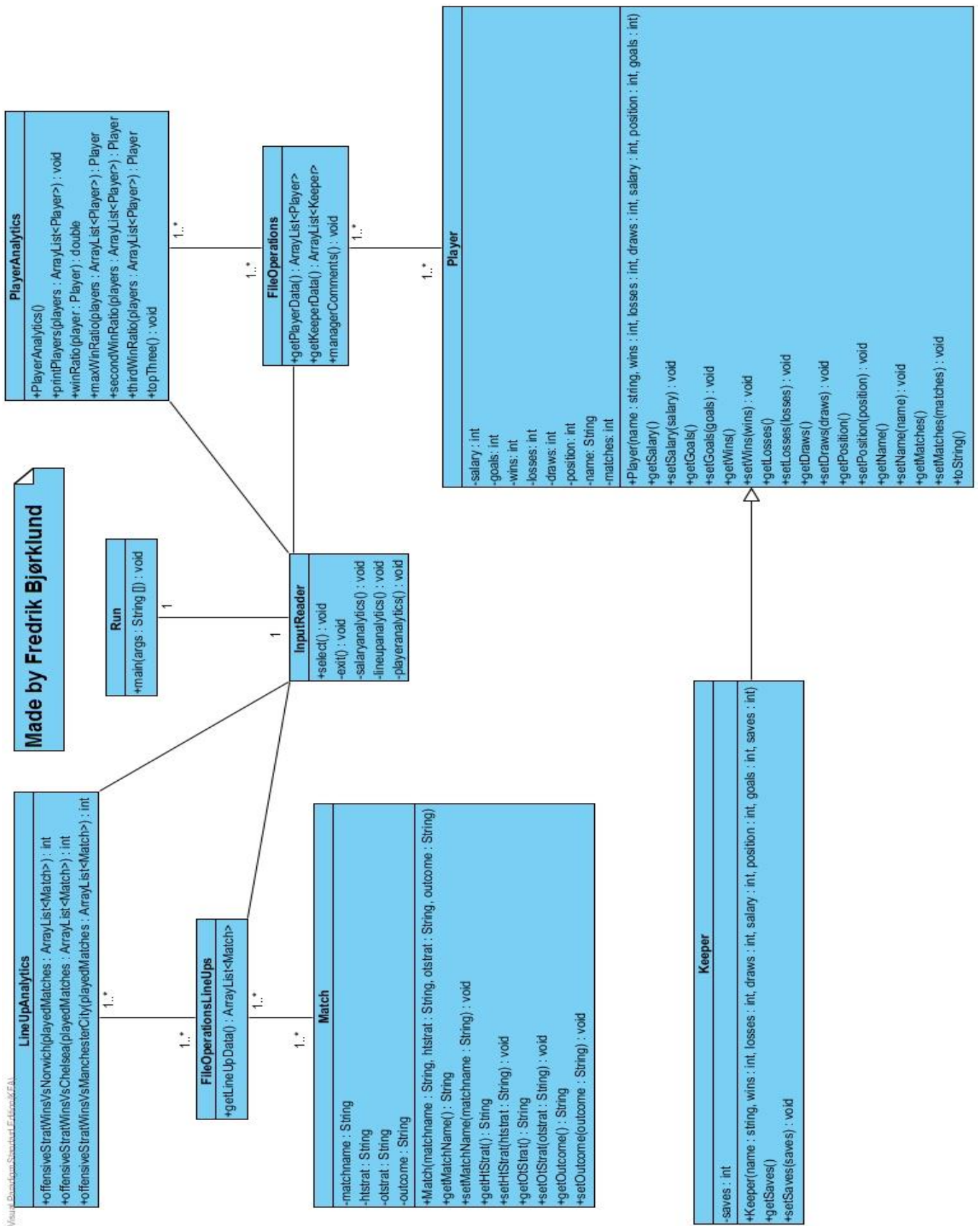


### SalaryAnalytics



### Exit program





## Kildeliste

- Craig Larman – Applying UML and patterns
- Diverse gamle opgaver