Chess club system.

(Picture 1: See source in bottom of document)

Group: Andreas Lind, Andreas Rømer, Oliver, Jonas.
Class: Dat18c.

Content

Chess System

This assignment involves a fictive chess club which needs a system to track and keep count of how many chess players that are active and are in the chess club.

Boundaries

We need to make a chess system which has several functionalities which include things such as logging in, creating members, managing members, and keeping track of people's' chess ranks in different categories. If everything goes well we will be able to do all of the above things, but else we will at least be able to CRUD members, ranks, and in smaller parts manage people's memberships and payments.

This was the idea at the start of the project. See end of the document to see how things turned out.

Abstract

This project is about making a database system for a chess club with CRUD in mind. We started our project with some group contracts and a SWOT analysis, and then we moved on to our vision, requirements and a GANTT-chart. After setting up requirements we made a glossary which we continuously filled with important principles from our project. Then we made our use cases from which we made a use case diagram and a domain model where we got a better idea of how our program could look in classes. Furthermore we made a system- and sequence system diagram, in which we defined some processes in our program. Finally we made a class diagram, from which we constructed the program.

SWOT-Analyse

	Positive	Negative
Internal conditions:	**Strengths:** - Good social environment - Tournaments, ELO-rating - Possible to choose your prefered membership	**Weaknesses:** - Not optimized administrationssystem - Dependency on junior members for money
External conditions:	**Opportunities:** - Get a new optimized administrationssystem - Try to appeal more to junior players - Social events, tournaments	**Threats:** - Downhill trend in members. Less junior members means less money. - Online chess services

Vision

Making management of members easier for employees, so these employees can spend more time on their members, ultimately leading to more members of the club.

Mission

Our mission is to produce an intuitive database program which can handle a large amount of chess players' personal information, ratings and subscriptions.

Requirements

Functional requirements regarding member management.

ID	Requirement	Comment
REQ101	The member management system will only be accessed and managed by the chairman.	
REQ102	The system will handle members accordingly to CRUD. (create, read, update, delete)	
REQ103	The system will manage the personal information about members. ● Name ● Address ● Age ● Gender ● CPR-number ● Phone Number ● Email address. ● Date of creation	The CPR-number has to be encrypted if stored in System.
REQ104	For any given member a subscription can be set to passive or active.	
REQ105	A member can have one of the following memberships. ● Junior (0-17) ● Senior (18 - 60) ● Senior (61+) ● Passive Membership	
REQ106		
REQ107		

Functional requirements regarding subscription management

ID	Requirement	Comment
REQ201	The subscription management system will only be accessed and managed by the cashier.	
REQ202	Subscription is on a yearly basis and price is dependant on membership and age. For subscriptions the follow rules apply: ● Junior (0-17) - 200 kr ● Senior (18 - 60) - 600 kr ● Senior (61+) - 600 kr minus 25% ● Passive membership - 100 kr	
REQ203	The system can print a list of members who owes subscription money.	
REQ204	The system shall print a list of Junior members. This list should contain the following: ● CPR-number ● Name ● Address	This list is printed once a year for Københavns kommune to get Subsidy payment.

Functional requirements regarding Player rating:

ID	Requirement	Comment
REQ301	Tournament players are connected with a team leader.	
REQ302	The system will be able to create and print a strength list for teamleaders to print to setup teams. See REQ303 for specification	
REQ303	**Strength list:** Passive and “hyggespillere” is not a part of the list. The list contains ratings of tournament players in 4 chess disciplines.(see comment) The list contains the following information for a tournament player: ● Rating ● Date achieved ● Tournament name Ratings are given accordingly to the ELO-System	**Disciplines are:** Lightning chess Fast chess Standard tournament chess Correspondence chess
REQ304	Player ratings can be printed by all(without login.)	
REQ305	Player rating update system can only be accessed and managed by Teamleaders.	

Functional requirements regarding Management of System:

ID	Requirement	Comment
REQ401	System can be managed by an admin.	
REQ402	Admin can create new user with one of the roles: Admin, Chairman, Cashier, TeamLeader	
REQ403	Admin can delete an old user	

Non-functional requirements for system:

ID	Requirement	Comment
NFR001	Easy maintenance	Structure easy to update
NFR002	Reliable system	
NFR003	Usability & User-friendliness	The system shall allow for easy, intuitive interaction with the user.
NFR004	Efficiency	The system shall handle requests with a low response time.

Glossary

Member	Is registered in database as active / passive player and pays yearly subscription fee.
Club database	Database of chess member informations.
Passive player	A member who pays a lesser amount.
“Hyggespiller”	A player that does not play in tournaments.
Tournament player	A player who's playing in tournaments.
Junior	A type of chess member.
Senior	A type of chess member.
Senior(60+)	A type of chess member.
Encryption	Self Made Simple Encryption (SMSE)
Search	Search for 1 member in the database.
Read	Reads information about members from from Database and is printed. The information printed in the list is depending on what role is printing.
Active member.	A member who pays the full amount.
Passive member	
Cashier	Role in system
Chairman	Role in system
Team Leader	Role in system
Memberships	Junior(0-17) Senior(18-60) Senior(61+)
Chairman menu	Menu presented to chairman after login
Team leader menu	Menu presented to Team leader after login
Cashier menu	Menu presented to Cashier after login
Members list.	Managed by Chairman. Holds personal information.
Ranking list	Managed by Team Leaders

MGMT	Management
FileHandler	
MMFH	
CashierFH	
TeamLeaderFH	

Gantt

AKTIVITET	STARTDATO FOR PLAN	VARIGHED FOR PLAN	FAKTISK STARTDATO	FAKTISK VARIGHED	PROCENTDEL FULDFØRT
Requirements	23-nov	23-nov	23-nov	23-nov	100%
SWOT-analyse	23-nov	24-nov	23-nov	23-nov	100%
Use Case	23-nov	29-nov	23-nov	26-nov	100%
Use Case Dia.	23-nov	29-nov	23-nov	26-nov	100%
Domain Model	26-nov	29-nov	23-nov	26-nov	100%
SSD	26-nov	29-nov	26-nov	26-nov	100%
SD	26-nov	29-nov	26-nov	27-nov	100%
Class Diagram	27-nov	29-nov	26-nov	27-nov	100%
Kodning	29-nov	04-dec	28-nov	05-dec	100%
Test	04-dec	06-dec	05-dec	06-dec	100%

Use cases

Use case accordingly to membership management: (Chairman)

- Use case 1: Create new member. (Fully dressed)
- Use case 2: Printing members list.
- Use case 3: Update member information.
- Use case 4: Delete member.

Use cases accordingly to subscription management: (Cashier)
Vælg en fully dressed.

- Use case 5: Print a list of Junior members (Fully dressed)
- Use case 6: Print subscription list of members.
- Use case 7: Print subscription list of non-paying members.
- Use case 8: Update subscription list with payments from members.

Use cases accordingly to ranking management: (Team Leader)

- Use case 9: Create new line in strength-list of players
- Use case 10: Delete a line of information in strength list.

Other Use cases:

- Use case 11: Print strength-list of players
- Use case 12: Login to system.

Use case 1. (Fully dressed)	**Description**
Title:	**Create new member.**
Scope:	Database
Level:	User goal
Primary Actor:	Chairman
Precondition:	Use Case 12 (Login)
Stakeholders and interests:	Chess management, developers, users
Main success scenario:	1. Chairman chooses **create new member** from the Chairman menu. 2. Chairman is asked by the system to input the information for the new member (See REQ103). 3. Chairman inputs personal information, 1 information at a time. 4. System generates new member number. 5. System validates input data and presents data to Chairman. 6. Chairman confirms new member. 7. System saves input data
Extensions:	**1a. Chairman can exit to menu at any given point by writing "Exit".** **4a. CPR-number already in database.** 1. System prints an error message 2. Chairman is presented by the chairman menu. **5a. System rejects input** 1. System presents the Chairman of error. 2. Chairman can enter information again until system approves or he exits to menu. If exited system deletes all entered information. **6a. Chairman rejects new member** 1. System deletes information and member-number. 2. Chairman is presented with the Chairman menu.

Use case 2: (Casual)	**Description:**
Title:	**Printing members list**
Primary Actor:	Chairman
Precondition:	Use Case 12 (Login)
Main success scenario:	1. Chairman chooses **Print member lists** from Chairman menu. 2. Chairman is presented with a menu with the types of memberships. ● All members (full list, and count). ● Junior ● Senior ● Senior+ 3. Chairman chooses a membership. 4. System prints a list of the of all members according to chosen membership.
Extensions:	**4a. There are no memberships of chosen type.** 1. System prints that there are no members of this type.

Use case 3 (Casual)	Description
Title:	**Update member information**
Primary Actor:	Chairman
Precondition:	Use Case 12 (Login)
Main success scenario:	1. Chairman chooses **Update member information** from the Chairman menu. 2. System prints a list of all members. 3. System asks Chairman of member to update. 4. Chairman picks a member to update. 5. System presents editable member information. 6. Chairman picks what information to update. 7. Chairman enters new information. 8. System validates information. 9. Chairman gets presented with updated member information. 10. Chairman confirms update. 11. System saves the information.
Extensions:	**1a. Chairman can exit to menu at any given point by writing "Exit".** **2a. There are no users in list.** 1. System prints a message that there are no members to update. 2. Chairman is presented with his menu. **8a. System rejects input.** 1. System presents an error message.. 2. Chairman can enter information again until system approves or he exits to menu. **10a. Chairman rejects update** 1. System deletes entered data and exits to menu.

Use case 4 (Casual)	**Description**
Title:	**Deleting a member.**
Primary Actor:	Chairman.
Precondition:	Use Case 12 (Login)
Main success scenario:	1. Chairman chooses **Delete member** from the “Chairman menu”. 2. System prints a list of all members. 3. System asks Chairman of member to delete. 4. Chairman picks a member to delete. 5. System presents chairman of all member information. 6. Chairman confirms deletion. 7. System deletes all information of the member in member list and strengthlist..
Extensions:	**6a. Chairman rejects deletion.** 1. System does nothing but go back to menu.

Use case 5. (Fully dressed)	**Description**
Title:	**Print a list of Junior members**
Scope:	Database
Level:	User goal
Primary Actor:	Cashier
Precondition:	Use Case 12 (Login)
Stakeholders and interests:	Københavns Kommune.
Main success scenario:	1. Cashier chooses "**Print List of Junior members**" from the Cashier menu. 2. System prints a list of all active Junior memberships. The list contains following information, (see REQ204): ○ CPR-number ○ Name ○ Address 3. The cashier sends the list manually by mail to Københavns Kommune.
Extensions:	2a. There are no active Junior members. 1. System prints that there are no Junior members. 2. Cashier is presented with cashiers menu.

Use case 6. (Brief)	**Description**
Title:	**Print subscription list of members.**
Primary Actor:	Cashier
Main success scenario:	1. Cashier chooses "Print subscription list" from cashiers menu. 2. System prints a list of all members with information about payment status.

Use case 7. (Brief)	Description
Title:	**Print subscription list of non-paying members.**
Primary Actor:	Cashier
Main success scenario:	1. Cashier chooses “Print subscription list” from cashiers menu. 2. System prints a list of non-paying members.

Use case 8 (Brief)	Description
Title:	**Update subscription list with payments from members.**
Primary Actor:	Cashier
Main success scenario:	1. Cashier retrieves payment from member via bank/mobilepay/cash. 2. Cashier updates a members payment status. 3. System saves data..

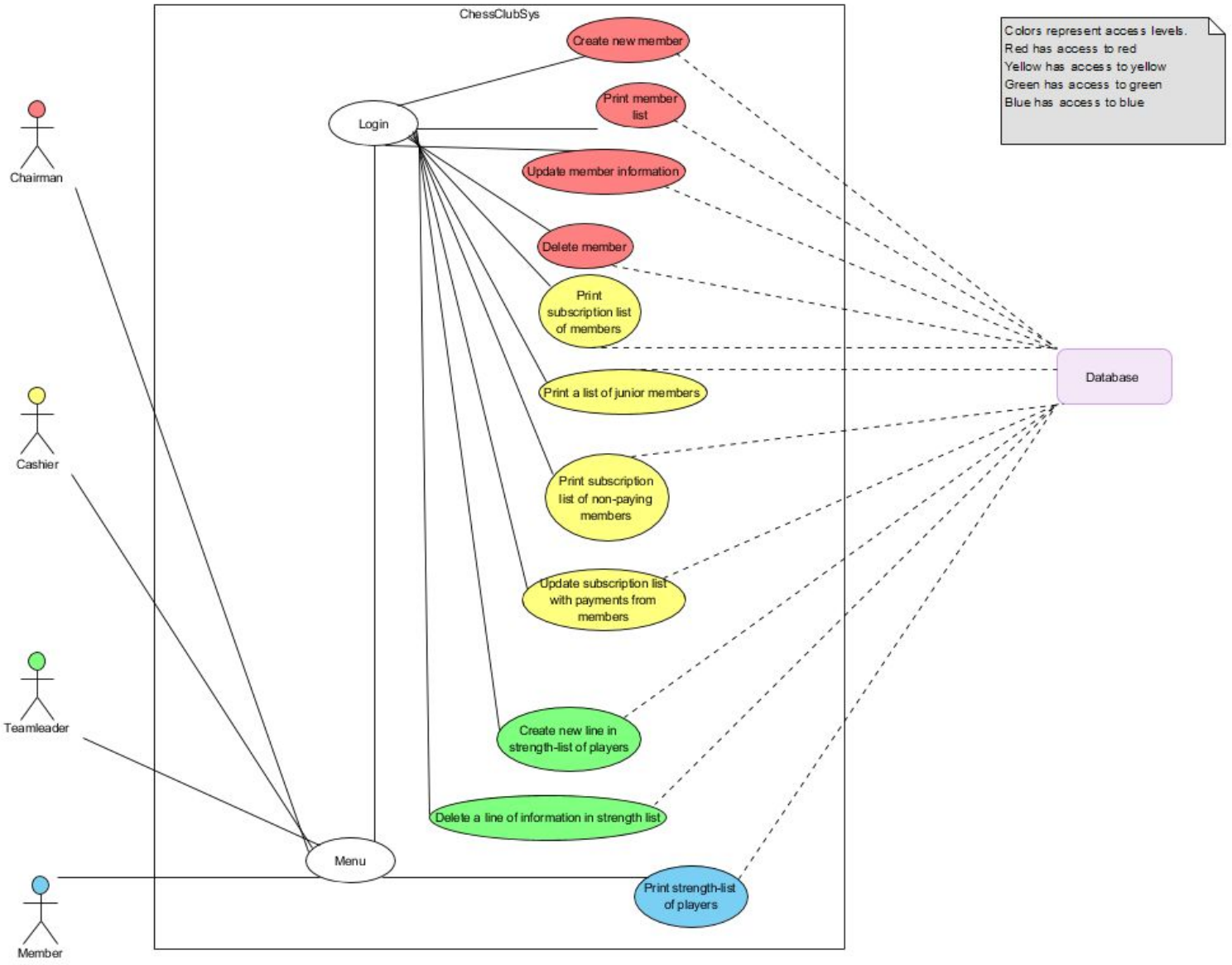
Use case 9 (Brief)	Description
Title:	Create new line in strength-list of players
Primary Actor:	Team Leader
Main success scenario:	1. Team leader chooses create new line in strength-list from menu. 2. Team leader enters player number, rating, date achieved and tournament name. 3. Team leader confirms new entry.

Use case 10 (Brief)	Description
Title:	Delete a line of information in strength list.
Primary Actor:	Team Leader
Main success scenario:	1. Team leader chooses delete new line in strength-list from the team leader menu. 2. Team leader chooses a line to delete. 3. Team leader confirms deletion.

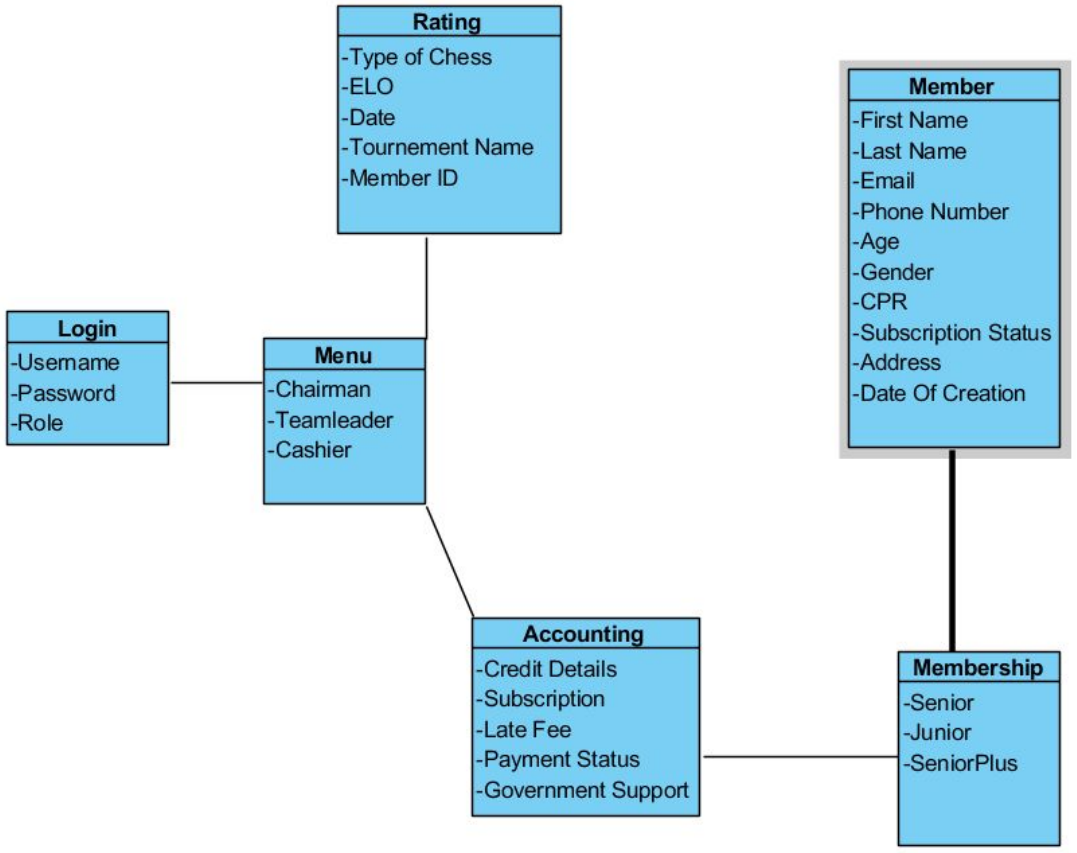
Use case 11 (Brief)	Description
Title:	Print strength-list of players
Primary Actor:	Chairman, Cashier, Team Leader + Members
Main success scenario:	1. Actors chooses print strength-list 2. System generates the strength-list and prints.

Use case 12 (Brief)	Description
Title:	Login to system.
Primary Actor:	Chairman, Cashier, Team Leader
Main success scenario:	1. The actor starts the system and system asks for username and password 2. System validates login credentials. 3. User is logged in and is presented with a menu.

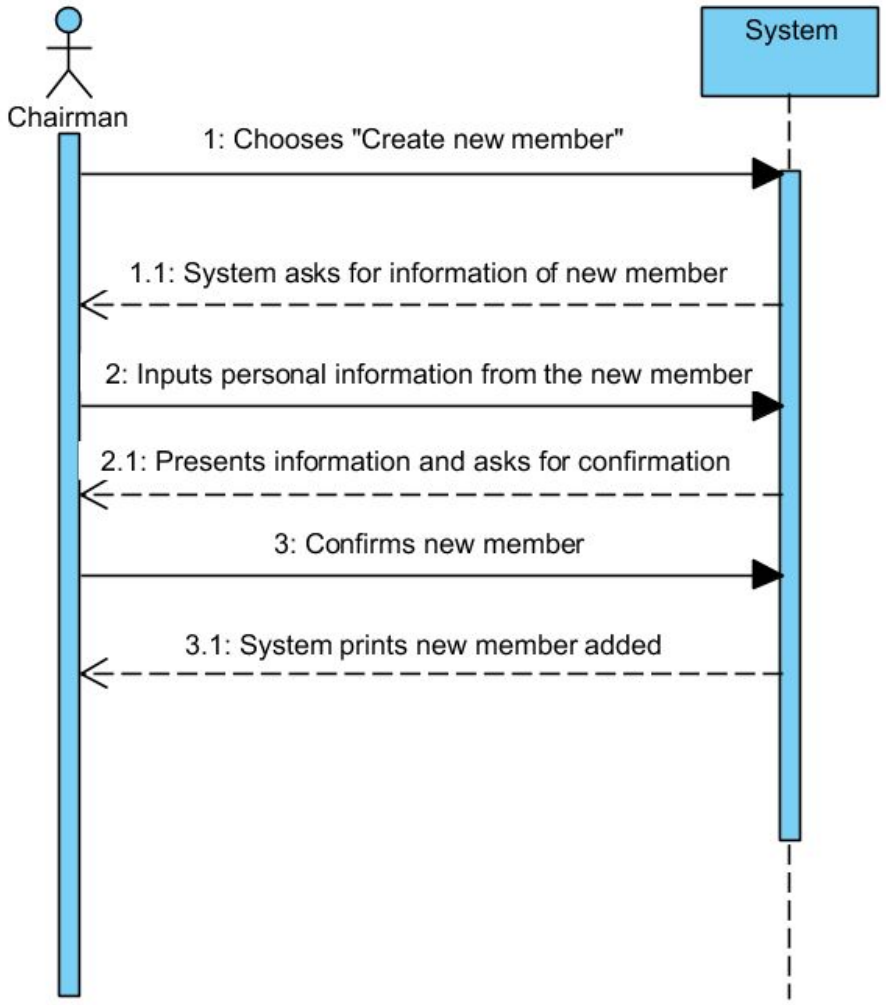
Use case diagram



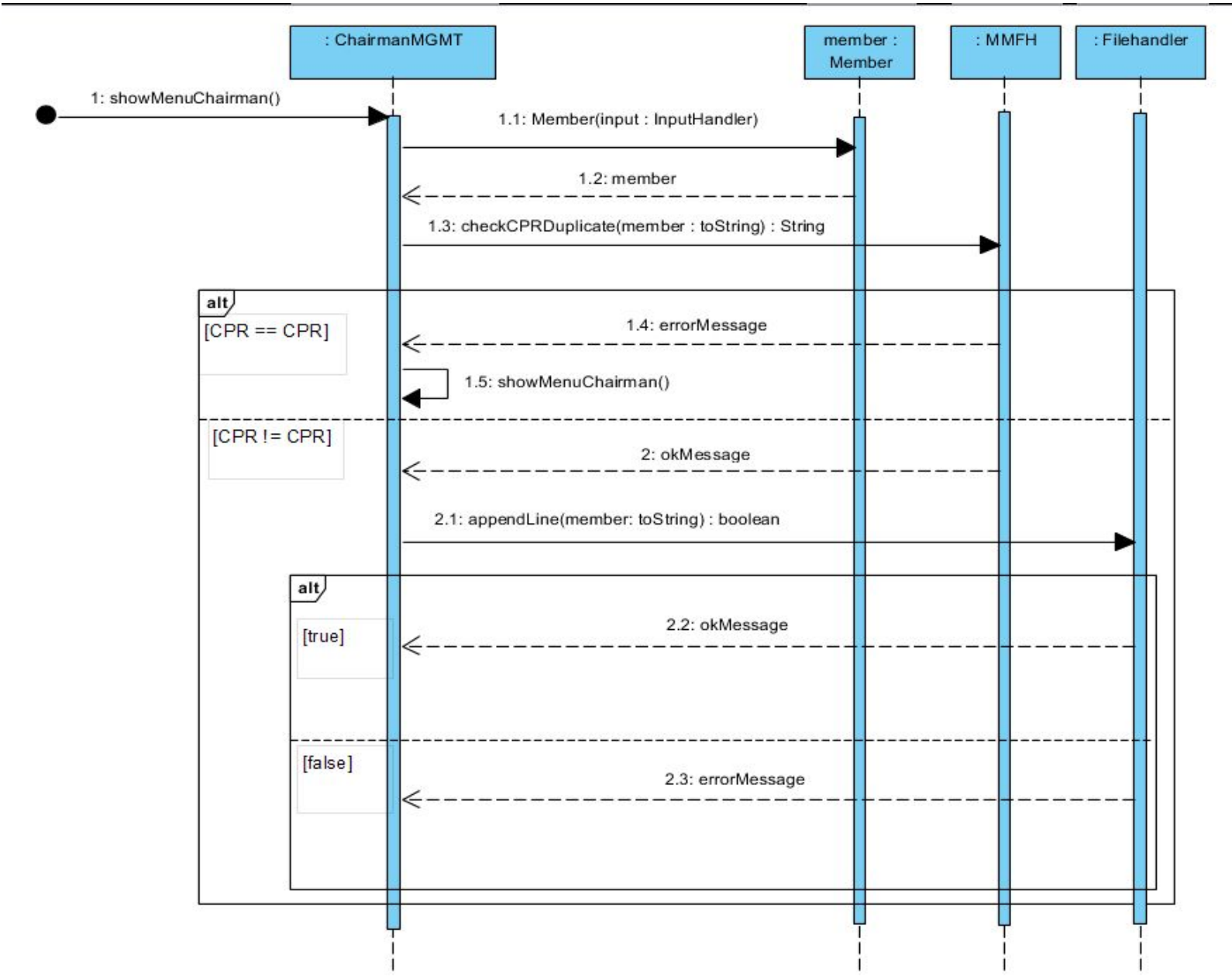
Domain model



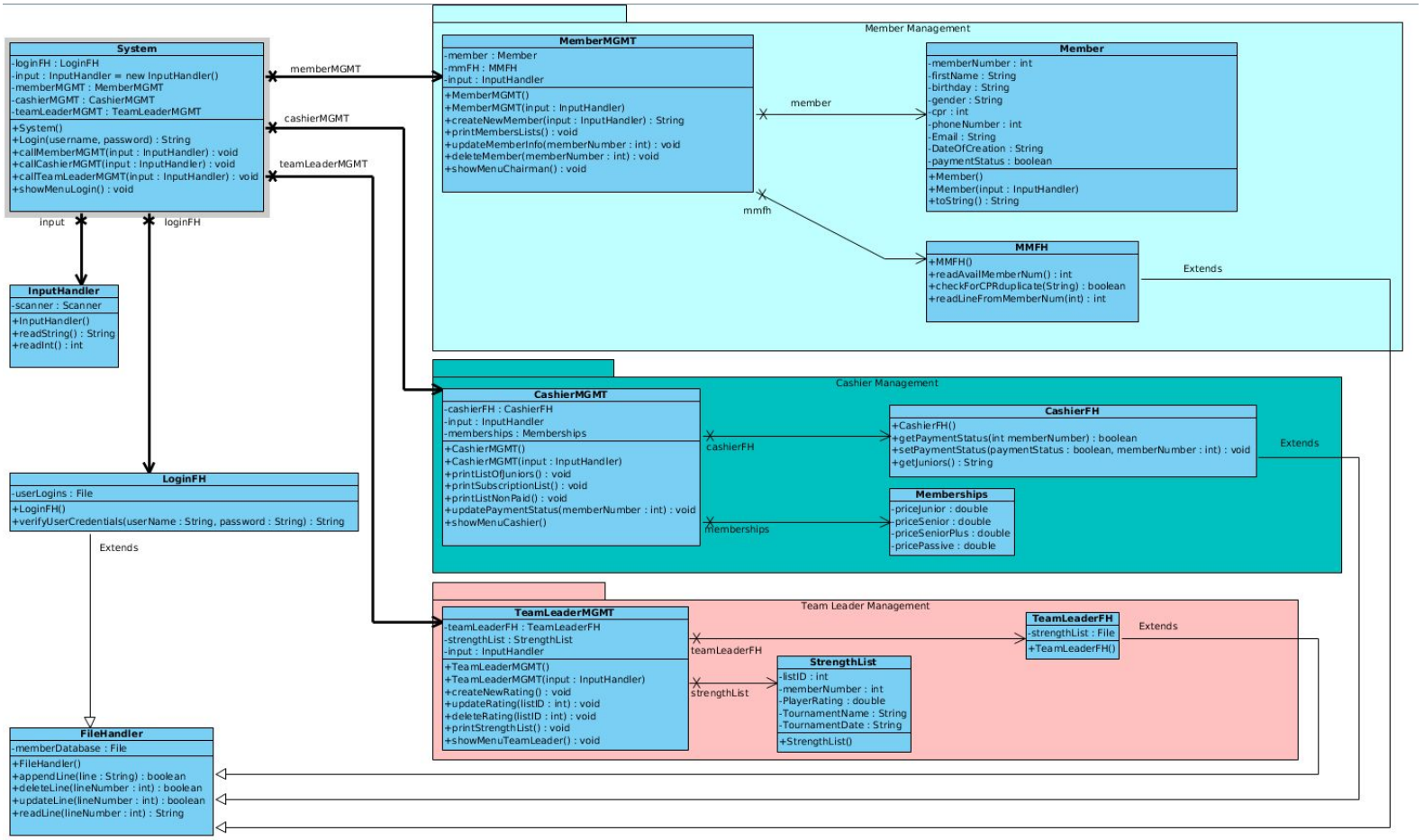
SSD


Chairman
System
1: Chooses "Create new member"
1.1: System asks for information of new member
2: Inputs personal information from the new member
2.1: Presents information and asks for confirmation
3: Confirms new member
3.1: System prints new member added

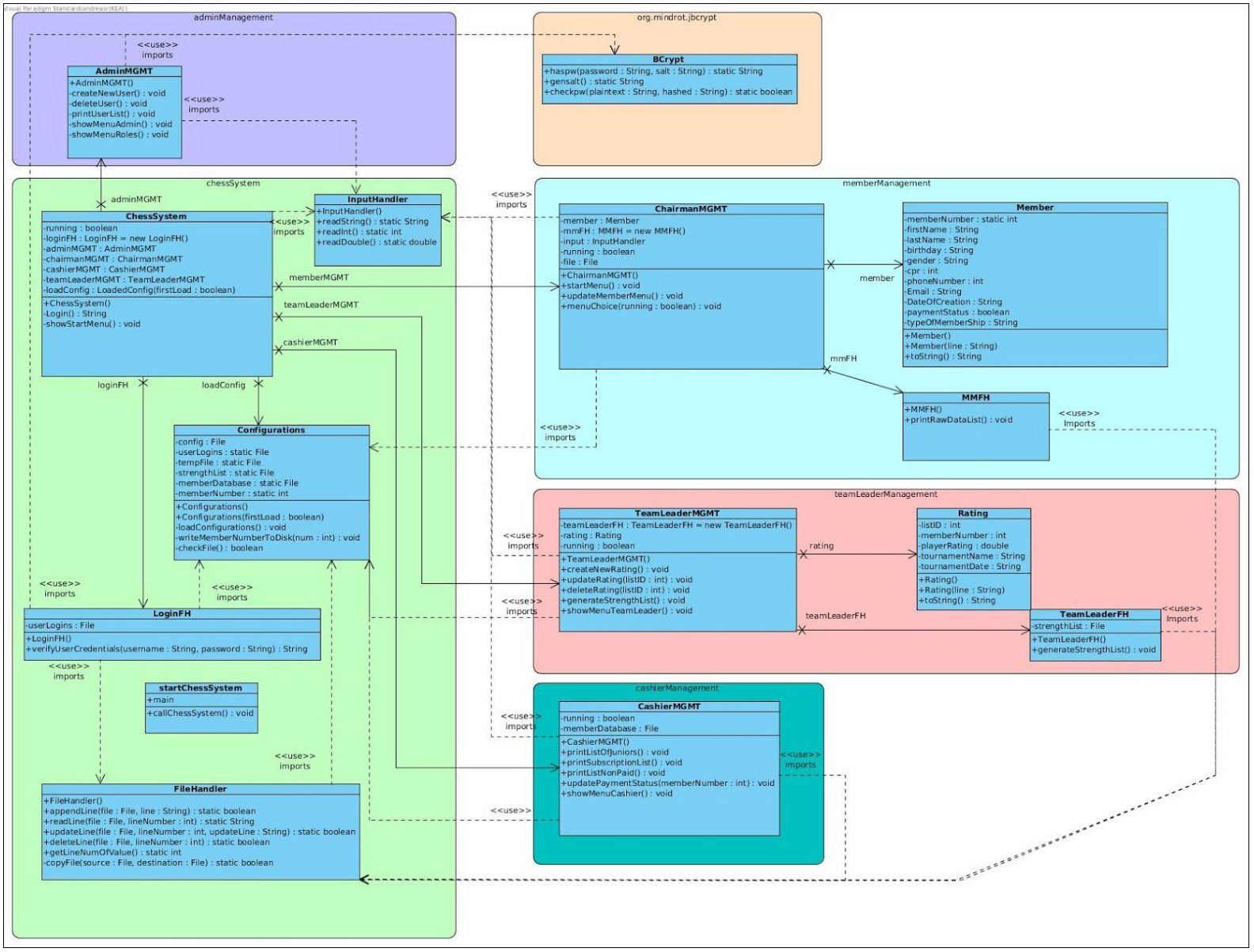
SD


: ChairmanMGMT
member : Member
: MMFH
: Filehandler
1: showMenuChairman()
1.1: Member(input : InputHandler)
1.2: member
1.3: checkCPRDuplicate(member : toString) : String
alt
[CPR == CPR]
1.4: errorMessage
1.5: showMenuChairman()
[CPR != CPR]
2: okMessage
2.1: appendLine(member: toString) : boolean
alt
[true]
2.2: okMessage
[false]
2.3: errorMessage

Class Diagram (1. udkast).



Class Diagram (Opdateres løbende).



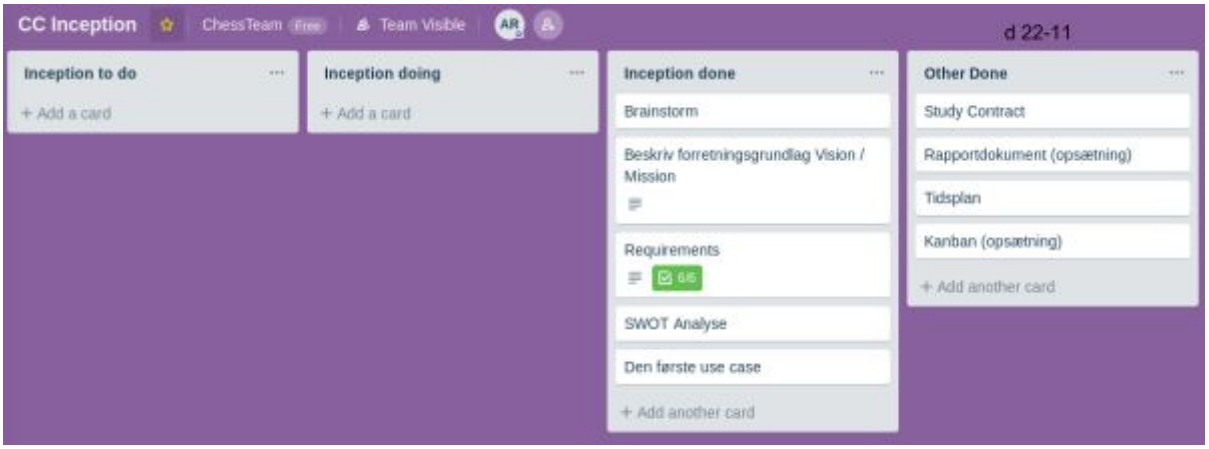
Conclusion

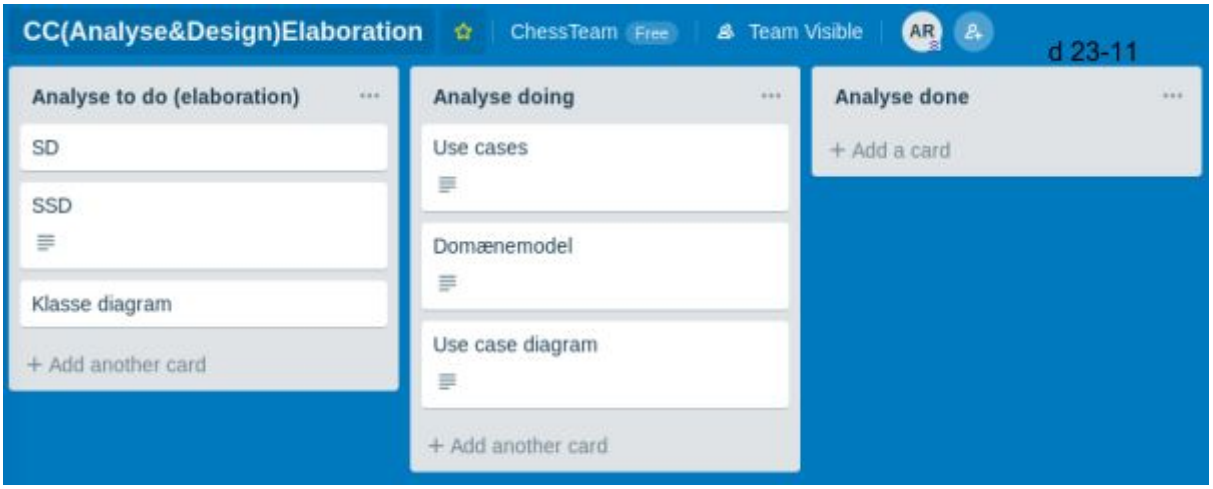
Java doc

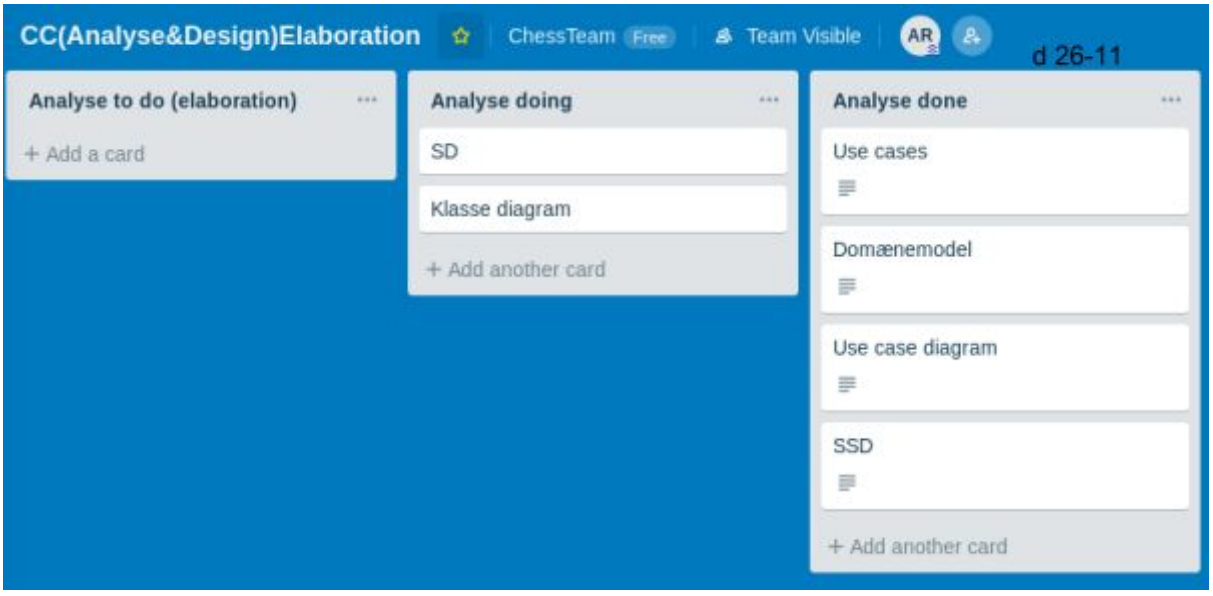
Uploaded til GIT - zip.

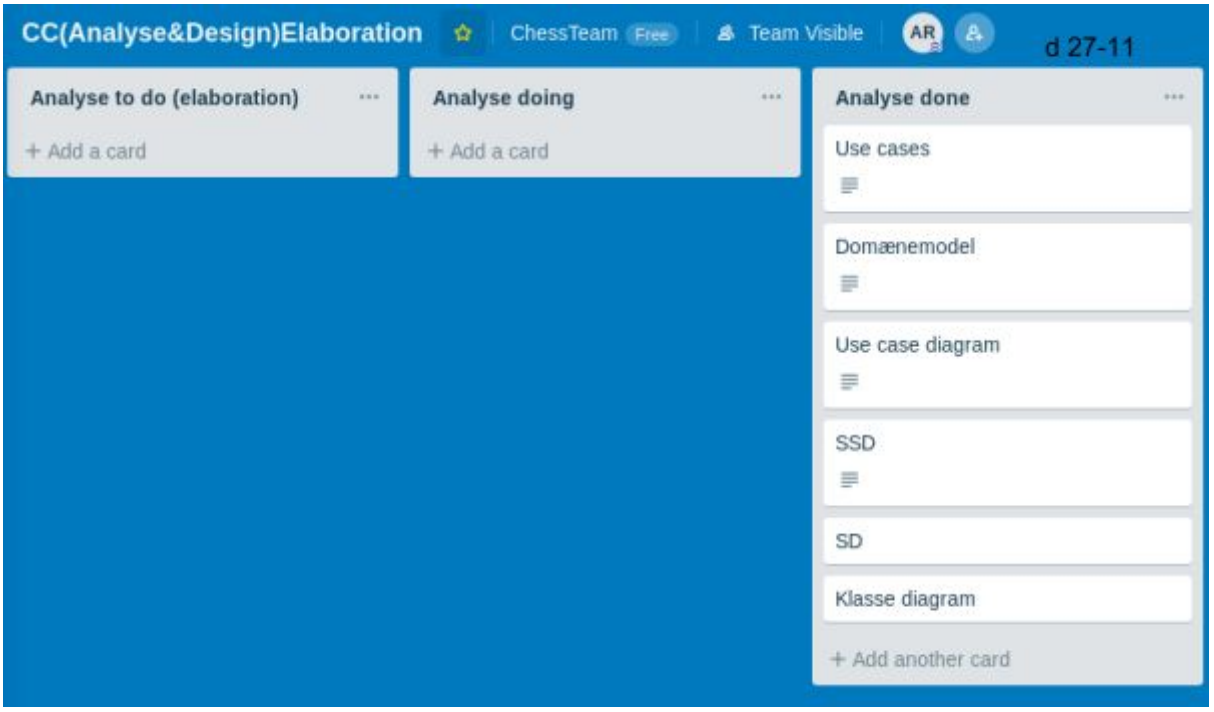
KANBAN

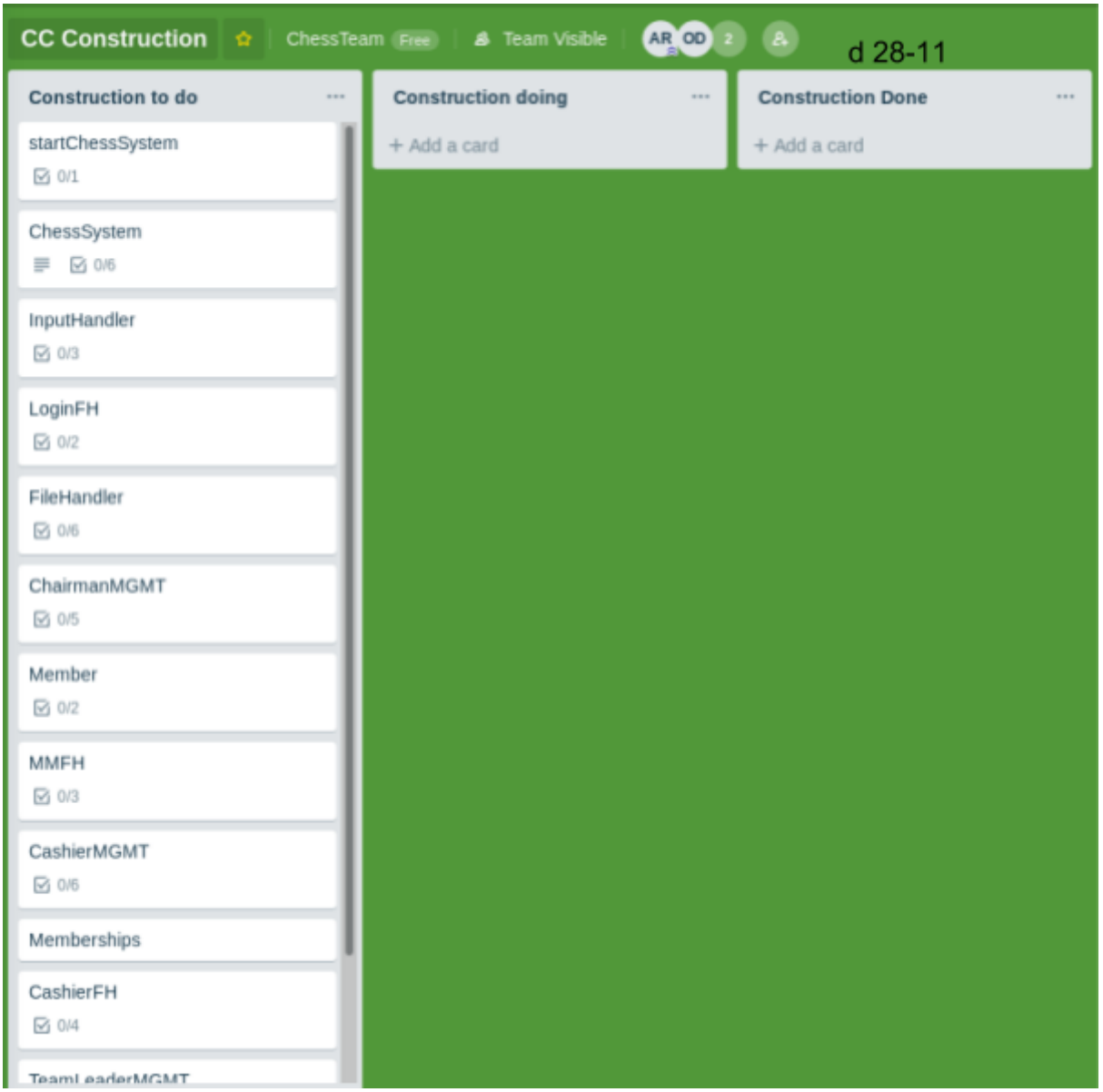
We wanted to encrypt the social security numbers and other personal parameters, such as phone number, email etc. in the program, but did not find the time to do this. Furthermore an optimization of the ELO-system could be done, calculating the ELO you win and lose, depending on the ELO of the players, who are facing each other. We also wanted to calculate the type of membership based on birthdays, and then calculate the price of subscription based on that, but didn't have the time for it. Furthermore passive and "hyggespillere" is not a part of the strength list, which could've been implemented if we were given more time.

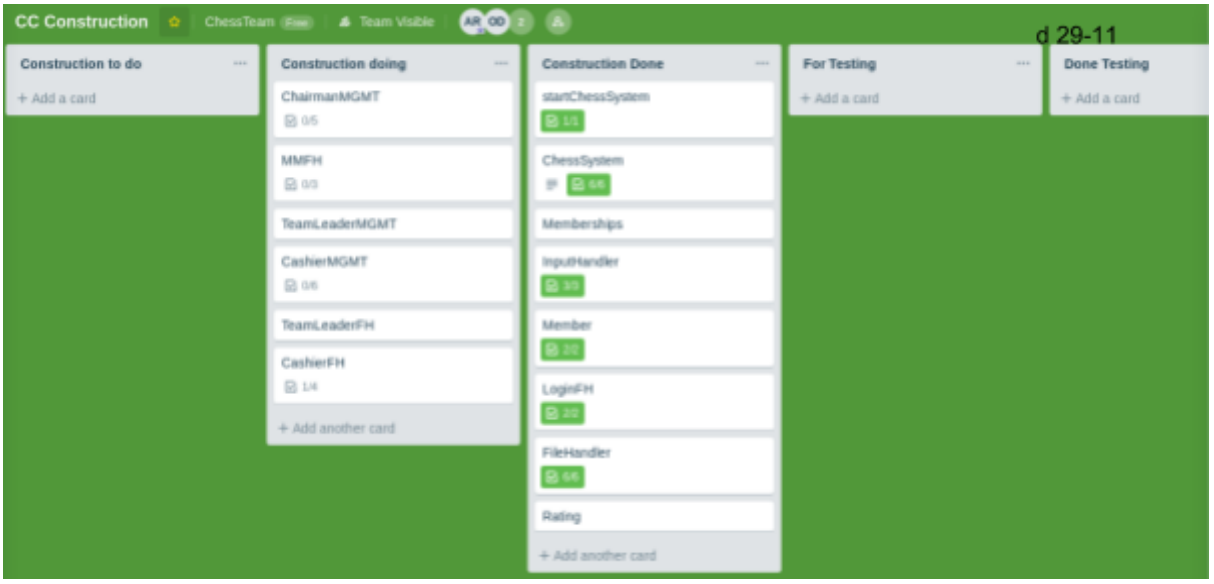

CC Inception
ChessTeam Free
Team Visible
AR
d 22-11
Inception to do
+ Add a card
Inception doing
+ Add a card
Inception done
Brainstorm
Beskriv forretningsgrundlag Vision / Mission
Requirements
6/6
SWOT Analyse
Den første use case
+ Add another card
Other Done
Study Contract
Rapportdokument (opsætning)
Tidsplan
Kanban (opsætning)
+ Add another card

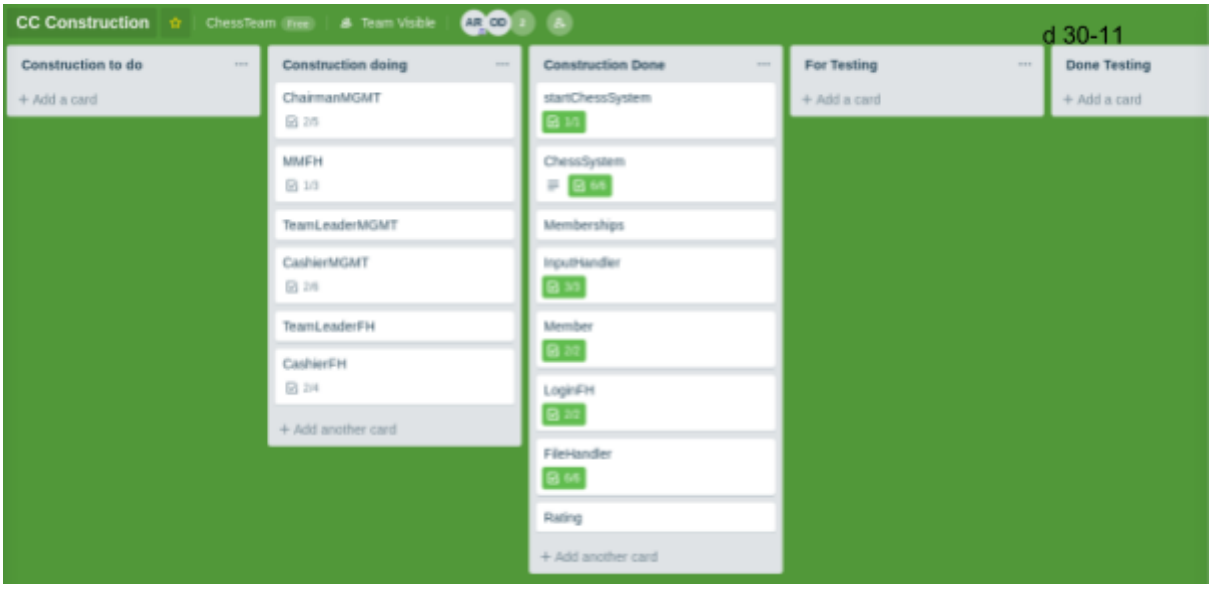

CC(Analyse&Design)Elaboration
ChessTeam Free
Team Visible
AR
d 23-11
Analyse to do (elaboration)
SD
SSD
Klasse diagram
+ Add another card
Analyse doing
Use cases
Domænemodel
Use case diagram
+ Add another card
Analyse done
+ Add a card

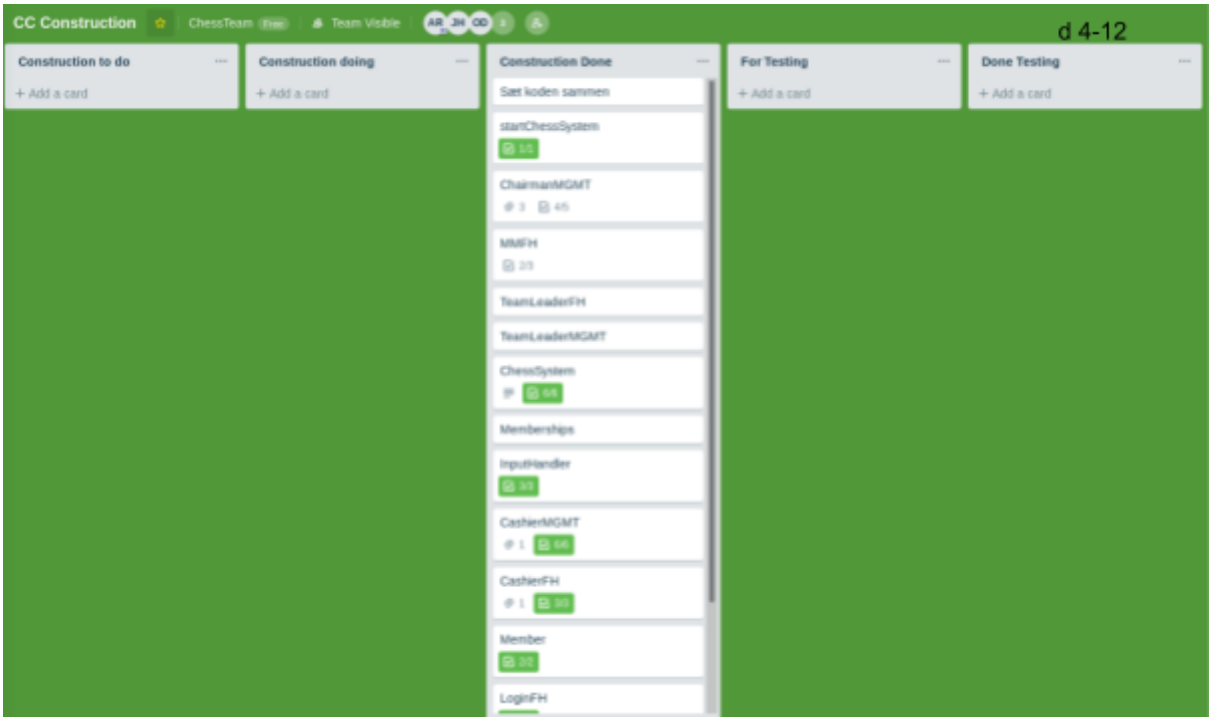

CC(Analyse&Design)Elaboration
ChessTeam Free
Team Visible
AR
d 26-11
Analyse to do (elaboration)
+ Add a card
Analyse doing
SD
Klasse diagram
+ Add another card
Analyse done
Use cases
Domænemodel
Use case diagram
SSD
+ Add another card

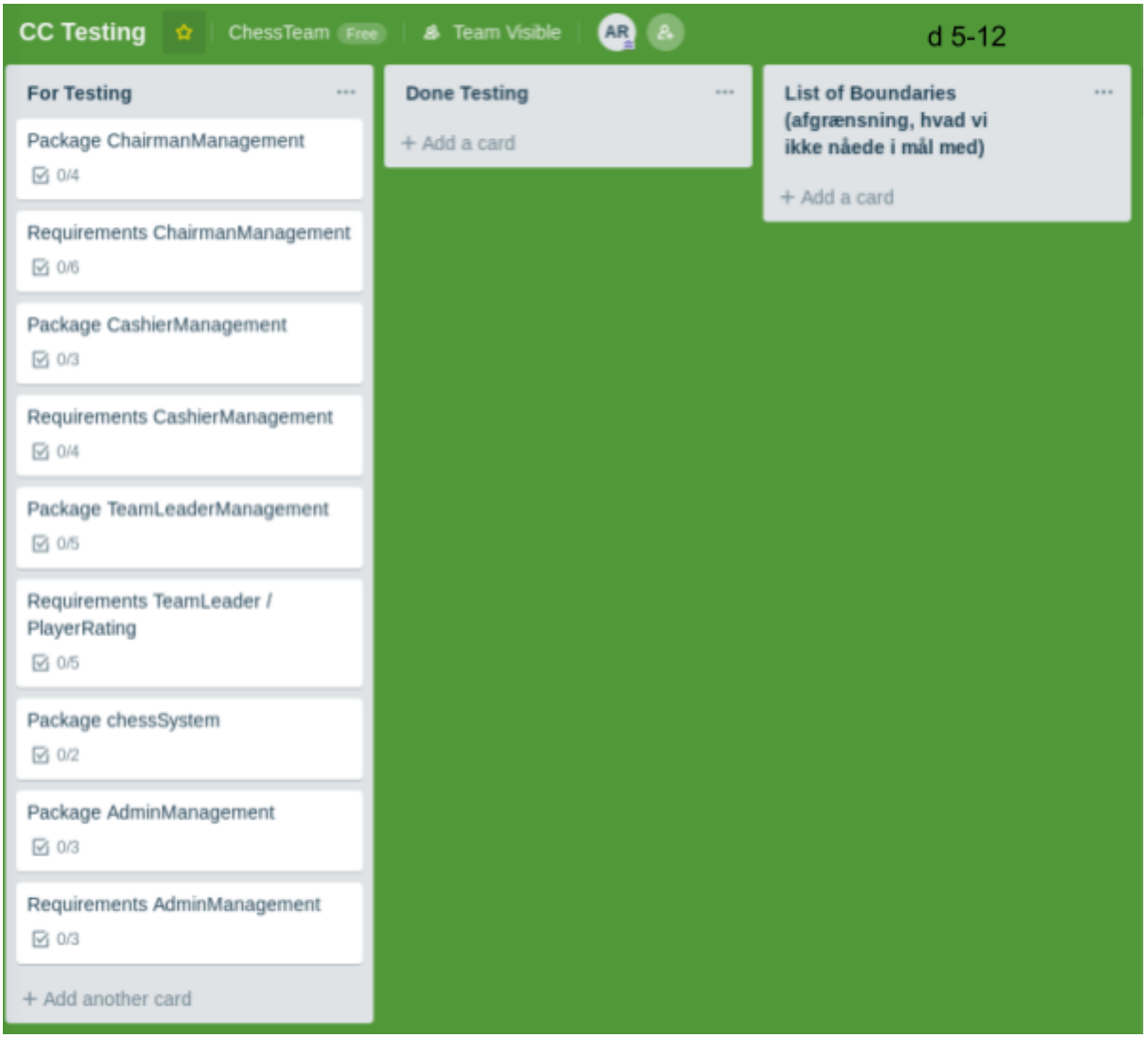

CC(Analyse&Design)Elaboration
ChessTeam
Free
Team Visible
AR
d 27-11
Analyse to do (elaboration)
+ Add a card
Analyse doing
+ Add a card
Analyse done
Use cases
Domænemodel
Use case diagram
SSD
SD
Klasse diagram
+ Add another card

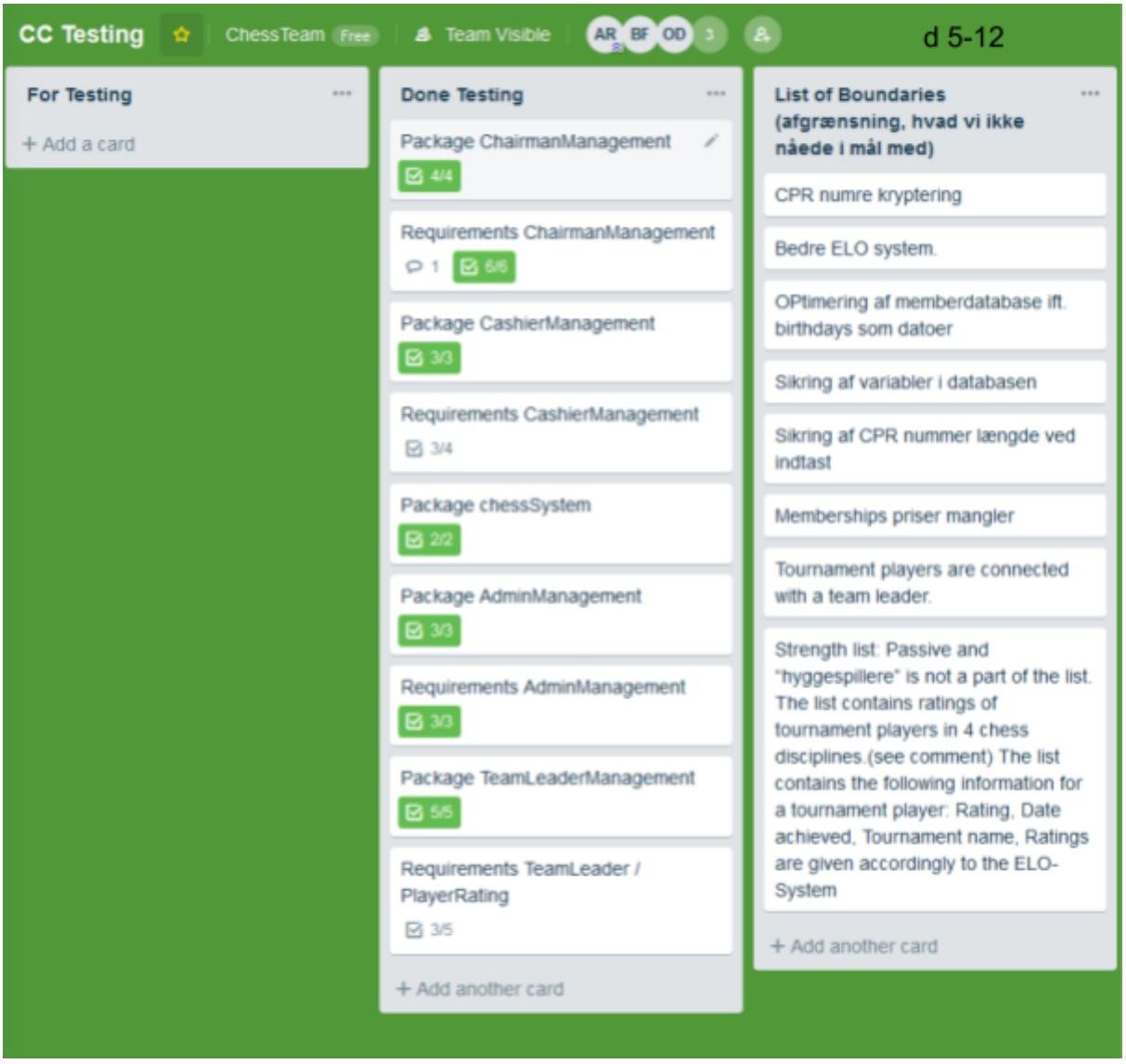

CC Construction
ChessTeam
Free
Team Visible
AR
OD
2
d 28-11
Construction to do
startChessSystem
0/1
ChessSystem
0/6
InputHandler
0/3
LoginFH
0/2
FileHandler
0/6
ChairmanMGMT
0/5
Member
0/2
MMFH
0/3
CashierMGMT
0/6
Memberships
CashierFH
0/4
Construction doing
+ Add a card
Construction Done
+ Add a card


CC Construction
ChessTeam
Team Visible
d 29-11
Construction to do
+ Add a card
Construction doing
ChairmanMGMT
0/5
MMFH
0/3
TeamLeaderMGMT
CashierMGMT
0/6
TeamLeaderFH
CashierFH
1/4
+ Add another card
Construction Done
startChessSystem
1/1
ChessSystem
Memberships
InputHandler
3/3
Member
2/2
LoginFH
2/2
FileHandler
6/6
Rating
+ Add another card
For Testing
+ Add a card
Done Testing
+ Add a card


CC Construction
ChessTeam
Team Visible
d 30-11
Construction to do
+ Add a card
Construction doing
ChairmanMGMT
2/5
MMFH
1/3
TeamLeaderMGMT
CashierMGMT
2/6
TeamLeaderFH
CashierFH
2/4
+ Add another card
Construction Done
startChessSystem
1/1
ChessSystem
Memberships
InputHandler
3/3
Member
2/2
LoginFH
2/2
FileHandler
6/6
Rating
+ Add another card
For Testing
+ Add a card
Done Testing
+ Add a card


CC Construction
ChessTeam
Team Visible
d 4-12
Construction to do
+ Add a card
Construction doing
+ Add a card
Construction Done
Sæt koden sammen
startChessSystem
ChairmanMGMT
MMFH
TeamLeaderFH
TeamLeaderMGMT
ChessSystem
Memberships
InputHandler
CashierMGMT
CashierFH
Member
LoginFH
For Testing
+ Add a card
Done Testing
+ Add a card


CC Testing
ChessTeam
Free
Team Visible
AR
d 5-12
For Testing
Package ChairmanManagement
0/4
Requirements ChairmanManagement
0/6
Package CashierManagement
0/3
Requirements CashierManagement
0/4
Package TeamLeaderManagement
0/5
Requirements TeamLeader / PlayerRating
0/5
Package chessSystem
0/2
Package AdminManagement
0/3
Requirements AdminManagement
0/3
+ Add another card
Done Testing
+ Add a card
List of Boundaries (afgrænsning, hvad vi ikke nåede i mål med)
+ Add a card


CC Testing
ChessTeam
Free
Team Visible
AR
BF
OD
3
d 5-12
For Testing
+ Add a card
Done Testing
Package ChairmanManagement
4/4
Requirements ChairmanManagement
1
6/6
Package CashierManagement
3/3
Requirements CashierManagement
3/4
Package chessSystem
2/2
Package AdminManagement
3/3
Requirements AdminManagement
3/3
Package TeamLeaderManagement
5/5
Requirements TeamLeader / PlayerRating
3/5
+ Add another card
List of Boundaries (afgrænsning, hvad vi ikke nåede i mål med)
CPR numre kryptering
Bedre ELO system.
OPtimering af memberdatabase ift. birthdays som datoer
Sikring af variabler i databasen
Sikring af CPR nummer længde ved indtast
Memberships priser mangler
Tournament players are connected with a team leader.
Strength list: Passive and "hyggespillere" is not a part of the list. The list contains ratings of tournament players in 4 chess disciplines.(see comment) The list contains the following information for a tournament player: Rating, Date achieved, Tournament name, Ratings are given accordingly to the ELO-System
+ Add another card