



CDIO Final

Monopoly

Gruppe nr: 12

Afleveringfrist: 22/01/2018

02312 Indledende programmering E18

Gruppemedlemmer:



Rasmus Traub Nielsen
s185101



Jens Daniel Kramhøft
s175445



Michael Lund Jarberg
s185091



Nicolai Nisbeth
s175565



Tanja Sølvsten
s185089



Marina Schmidt Malling
s185104

https://github.com/jarberg/12_3uger

Indholdsfortegnelse	
Timeregnskab	3
Resume	5
Indledning	5
1. Analyse	6
1.1 Vision	6
1.2 Kravspecifikation	6
MoSCoW	6
Must have	6
Should have	7
Could have	8
Won't have (this time)	8
1.3 Use cases	9
1.3.1 Use case diagram	9
1.3.3 Fully-dressed use case beskrivelser	9
1.4 System sekvens diagram	13
1.5 Domænemodel	16
2. Design	17
2.1 Design klassediagrammer	17
2.2 Sekvensdiagrammer	18
2.3 Arkitektur	21
2.4 GRASP	23
2.4.1 Low Coupling	23
2.4.2 High Cohesion	23
2.4.3 Creator	23
2.4.4 Controller	23
2.4.5 Information Expert	23
2.4.6 Polymorphism	23
3. Implementering	24
4. Test	25
4.1 Struktur	25
4.2 Testmetoder	25
4.3 Test cases	25
4.4 Kode dækning	25
4.5 Traceability Matrix	25
5. Konfiguration	27
5.1 Versionsstyring	27
6. Projektforløb	28

7. Konklusion	29
8. Bilag	30
8.1 Supplerende specifikation	30
8.2 Use cases	30
8.3	36

Timeregnskab

Jens Daniel Kramhøft	Analyse	Design	Implementation	Dokumentation
7/1/2019	4½	2		1
8/1/2019	5	1		1
9/1/2019		2	6	
10/1/2019		3	4	
11/1/2019		2	4	
12/1/2019				
13/1/2019			1	
14/1/2019		3	3	1
15/1/2019		1	5	

Michael Lund Jarberg	Analyse	Design	Implementation	Dokumentation
7/1/2019	5	2		2
8/1/2019	4	1		2
9/1/2019		1	5-6	1
10/1/2019			7	
11/1/2019			7	
12/1/2019				
13/1/2019				
14/1/2019			7	
15/1/2019			7	

Marina Schmidt Malling	Analyse	Design	Implementation	Dokumentation
7/1/2019	4½	2		1
8/1/2019	4	1		2
9/1/2019		2	5	
10/1/2019			6	
11/1/2019	2		4	1
12/1/2019			2	
13/1/2019				
14/1/2019			6½	
15/1/2019			5	1

Nicolai Nisbeth	Analyse	Design	Implementation	Dokumentation
7/1/2019	5	2		2
8/1/2019	4	1		2
9/1/2019		2	5	
10/1/2019		3	5	
11/1/2019	2	1	2	1
12/1/2019			5	
13/1/2019			2	
14/1/2019				2
15/1/2019			5	

Tanja Sølvsten	Analyse	Design	Implementation	Dokumentation
7/1/2019	4½	2		1
8/1/2019	4	1		2
9/1/2019		2	5	
10/1/2019			6	
11/1/2019	2		4	1
12/1/2019				
13/1/2019				
14/1/2019		1½	4	
15/1/2019			5	1

Rasmus Traub Nielsen	Analyse	Design	Implementation	Dokumentation
7/1/2019	4½	2		1
8/1/2019	4	1		2
9/1/2019		2	5	
10/1/2019	1	1	6	
11/1/2019			4	
12/1/2019				
13/1/2019				
14/1/2019			5	
15/1/2019				

Resume

Indledning

1. Analyse

I følgende analyse tages der udgangspunkt i kundens vision, som i enighed med udviklerne bliver yderligere specificeret i kravspecifikationen.

1.1 Vision

Kundens vision er, at der skal udvikles en digital version af det fulde Matadorspil. Reglerne hentes fra det originale brætspil. Kunden har overladt os en vurderingen af, hvad der er de vigtigste elementer i spillet.

1.2 Kravspecifikation

Kravene prioriteres alt efter vigtigheden af deres implementering, da det ud fra kundens vision ikke er nødvendigt for spillets funktionalitet at have samtlige krav/regler implementeret. Til prioriteringen anvendes MoSCoW metoden. Alle ikke-funktionelle krav prioriteres ikke efter MoSCoW metoden og fremgår i stedet i projektets supplerende specifikation, xxx(ref til bilag).

MoSCoW

Must have

Liste over krav der **skal** implementeres.

M1: Systemet skal spilles af 3-6 spillere.

M2: Systemet skal lade spillerne vælge mellem 6 køretøjer.

M3: Systemet skal lade spillernes figurer starte på "START"- feltet.

M4: Systemet skal give spillerne en startkapital på 1500 kr.

M5: Systemet skal lade spilleren kaste med terningen og få en sum.

M6: Systemet skal kun rykke spilleren med uret det antal felter ternings sum giver.

M7: Systemet skal lade spilleren modtage 200 kr. fra banken hvis denne spiller lander på/passere start-feltet.

M8: Systemet skal lade ejendoms grunde starte uden ejer.

M9: Systemet skal lade spilleren købe et ledigt felt når spilleren lander på feltet.

M10: Systemet skal kræve husleje, hvis en spiller lander på et felt ejet af en anden spiller.

M11: Systemet skal indikere hvilke spillere der ejer grundene.

M12: Systemet skal lade en spiller rykke ud fra fængsel hvis spilleren betaler 50 kr.

M13: Systemet skal eliminere en spiller, hvis spilleren skylder flere penge end spilleren har mulighed for at frembringe fra sine værdier. **Resterende grunde sættes på auktion.**

M14: Systemet skal vælge vinderen som den sidste spiller tilbage når alle andre er gået fallit.

Should have

Liste over krav der **burde** implementeres.

S1: Systemet skal give en spiller en ekstra tur hvis spilleren slår to ens.

S2: Systemet skal give spillerne mulighed for at udveksle fire huse til et hotel der koster fem gange så meget som et hus.

S3: Systemet skal indikere om en grund er pantsat.

S4: Systemet skal kræve dobbelt leje fra en grund hvis ejeren af grunden ejer alle grunde i samme farve.

S5: Systemet skal lade spillerne pantsætte ubebyggede grunde til banken.

S6: Systemet skal give spillerne mulighed for at købe deres pantsatte grunde tilbage med 10 procent i rente.

S7: Systemet skal ikke kræve leje af pantsatte grunde når andre spillere lander på dem.

S8: Systemet skal undlade at give spillere penge for at passere start når de sættes i fængsel.

S9: Systemet skal lade en spiller rykke ud fra fængsel hvis spilleren har et friheds-kort.

S10: Systemet skal lade en spiller rykke ud fra fængsel hvis spilleren slår 2 ens og får også et ekstra kast.

S11: Systemet skal lade en spiller rykke ud fra fængsel hvis spilleren har været i fængsel over 3 omgange. Ved tredje kast der ikke er 2 ens skal spilleren betale 50 kr. i bøde samt rykke antal øjne.

S12: Systemet skal lade spilleren trække det øverste kort i 'prøv lykke' bunken, hvorefter teksten på kortet udføres af spilleren.

S13: Systemet skal lægge kortet som er blevet trukket tilbage nederst i bunken.

S14: Systemet skal blande bunken med chancekort ved spillets start.

S15: Systemet skal kræve enten 200 kr penge plus 10% af spillerens værdier fra spilleren når spilleren lander på indkomstskat feltet, efter valg af spilleren.

S16: Systemet skal lade spillere bygge huse på deres ejede felter, hvor det så er givet at spilleren ejer alle felter af samme type, og at de andre felter af den type har en eller færre antal huse forskel fra de andre felter.

S17: Systemet skal lade spillere bygge hoteller for 5*huspris på deres ejede felter, givet af spilleren ejer alle felter af den type, og at de andre felter af samme type har 4 huse eller et hotel.

Could have

Liste over krav der **kunne** implementeres.

C1: System skal smide en spiller i fængsel hvis spilleren har slået 2 ens 3 gange i træk.

C2: Systemet skal lade en grund gå på auktion, hvis en spiller vælger ikke at købe den.

C3: Systemet skal give spillere mulighed for at sælge bygninger til banken.

C4: Systemet skal lade spillere sælge deres bygninger til halv pris.

Won't have (this time)

Liste over krav der **ikke** implementeres.

W1:

W2:

W3:

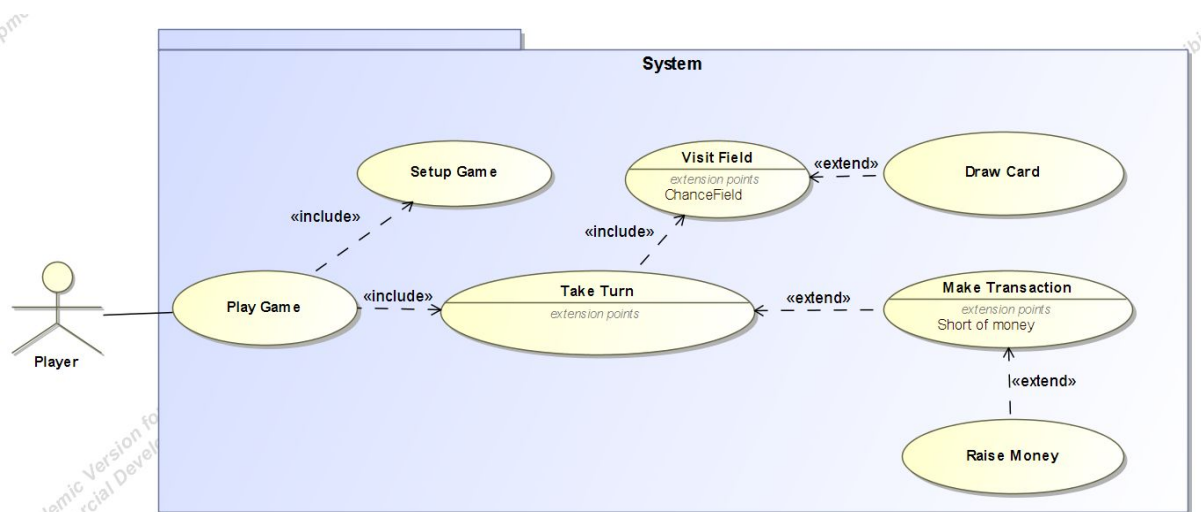
Alle implementeret krav bliver dokumenteret ved opstilling- og kørsel af tilhørende test cases i test afsnittet.

1.3 Use cases

Ud fra kundens vision og den udarbejdede kravspecifikation, bliver systemets use cases defineret.

1.3.1 Use case diagram

Spillets primær- og eneste aktør er “Player” som kun har relationen til systemets kritiske use case “Play Game”, se figur xxx. “Play Game” indeholder sub-use cases som hver er ansvarlig for mindre dele af systemets funktionalitet. Fordelen ved at opdele “Play Game” i sub-use cases er en bedre fordeling af ansvar som højner samhørigheden.



figur xxx

1.3.3 Fully-dressed use case beskrivelser

Samtlige use cases er beskrevet fully-dressed. Use casene “Play Game”, “Setup Game” og “Take Turn” fremgår nedenstående i samme rækkefølge som de er opremset. De resterende use cases er i bilag xxx.

Use casen “**Play Game**” er ansvarlig for eksekveringen af spillet.

Use case: Play Game
ID: UC1
Kort beskrivelse: Spillet spilles indtil vinder findes

Primær aktør: Spiller
Precondition: 3-6 spillere ønsker at spille spillet
Main flow: 1. include(Setup Game). 2. Så længe spillet har mere end én spiller tilbage <ul style="list-style-type: none"> 2.1 Systemet beder spiller om at starte sin tur 2.2 include(Take Turn) 3. Systemet kårer vinderen
Postcondition: En vinder er fundet og spillet er afsluttet
Alternative flows:

Use casen “**Setup Game**” er ansvarlig for at klargøre spillet.

Use case: Setup Game
ID: UC2
Kort beskrivelse: Spiller starter spillet og spillet klargøres
Primær aktør: Spiller
Precondition: Systemet er startet.
Main flow: 1. Systemet beder Spiller om at vælge sprog. 2. Spiller vælger spillets sprog. 3. Systemet beder Spiller om at angive antal af spillere. 4. Spiller vælger antallet af spillere. 5. For hver spiller <ul style="list-style-type: none"> 5.1 Systemet beder spiller om at indtaste deres navn. 5.2 Spiller indtaster deres navn. 5.3 Systemet beder spiller om at vælge spillebrik. 5.4 Spiller vælger spillebrik. 6. Systemet opretter spillepladen med spillere.
Postcondition: Spillet er klargjort
Alternative flows:

Use casen “**Take Turn**” er ansvarlig for eksekveringen af en spillers tur.

Use case: Take Turn

ID: UC3
Kort beskrivelse: Den aktive spiller kaster med terningerne. Systemet rykker brikken og effekten af det pågældende felt udføres.
Primær aktør: Spiller
Precondition: Det er en spillers tur
<p>Main flow:</p> <ol style="list-style-type: none"> 1. Spiller kaster med terningerne. 2. Systemet rykker spilleren til et felt ud fra terningens værdi. 3. Hvis Spiller har passeret start så <ol style="list-style-type: none"> 3.1 Systemet giver Spiller 200 kr. (Make Transaction) 4. include(Visit Field) 5. Systemet præsenterer aktioner 6. Spilleren vælger aktionen "End turn" 7. Turen slutter.
Postcondition: Ny spillers tur.
<p>Alternative flows:</p> <ol style="list-style-type: none"> 1a. Spiller er i fængsel <ol style="list-style-type: none"> 1. Systemet viser tre valgmuligheder. <ol style="list-style-type: none"> 1.1 Spiller udløser et 'Get out of Jail' kort. 1.2. Spiller betaler 50 kr. 1.3. Spiller slår med terninger. <ol style="list-style-type: none"> 1.3.1. Spiller slår to ens. 1.3.2 Spiller slår ikke to ens <ol style="list-style-type: none"> 1.3.2.1 Spillers tur er slut. 1.3.2.2 Spiller har "ikke slået to ens" tre gange i træk <ol style="list-style-type: none"> 1.3.2.2.1 Spiller betaler 50 kr. <ol style="list-style-type: none"> 1.3.2.2.1.1. Spiller har ikke nok penge <ol style="list-style-type: none"> 1.3.2.2.1.1.1 include(Raise Money) 2. Spiller slår terninger.

6.a Spiller vælger aktionen "Sell"

1. Systemet viser to valgmuligheder.

1.1 Spiller vælger jail card

1.1.1 Spiller vælger den spiller de vil sælge til

1.1.1.1 Den valgte spiller vælger ja

1.1.1.1.1 (Make Transaction)

1.1.1.2 Den valgte spiller vælger nej

1.1.1.2.1 Systemet annullerer handlen

1.2 Spiller vælger building

1.2.1 Systemet præsenterer mulige properties med buildings

1.2.1.1 Spiller vælger en property

1.2.1.1.1 (Make Transaction)

6.b Spiller vælger aktionen "Trade Property"

1. Spiller vælger property de vil bytte.

1.1 Spiller vælger den spiller de vil bytte med.

1.1.1 Systemet informerer den valgte spiller om handlen

1.1.1.1 Den valgte spiller vælger ja

1.1.1.1.1 Den valgte spiller vælger det felt de vil bytte med.

1.1.1.1.2 (Make Transaction)

1.1.1.2 Den valgte spiller vælger nej

1.1.1.2.1 Systemet annullerer handlen

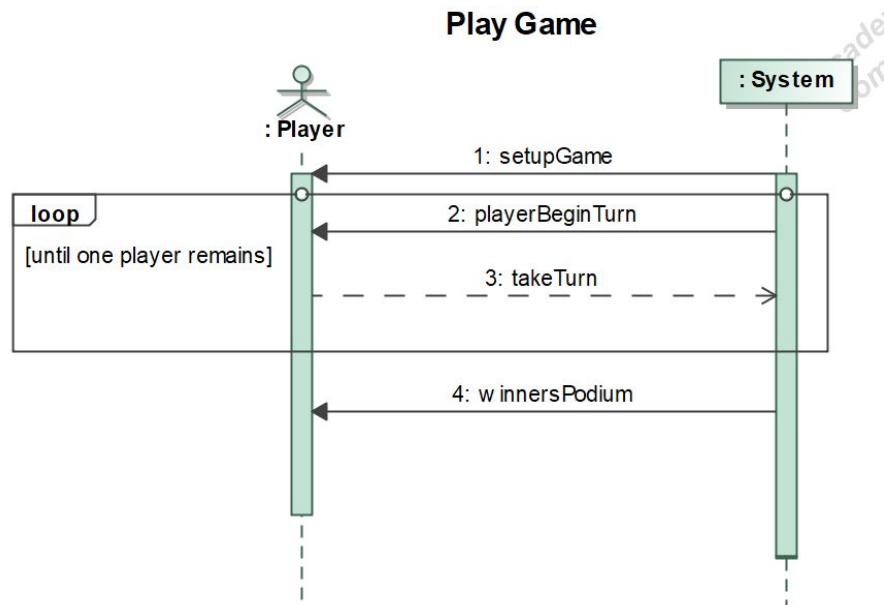
6.c Spiller vælger aktionen "Pantsæt Property"

1 Spiller vælger det felt de vil pantsætte

1.1 (Make Transaction)

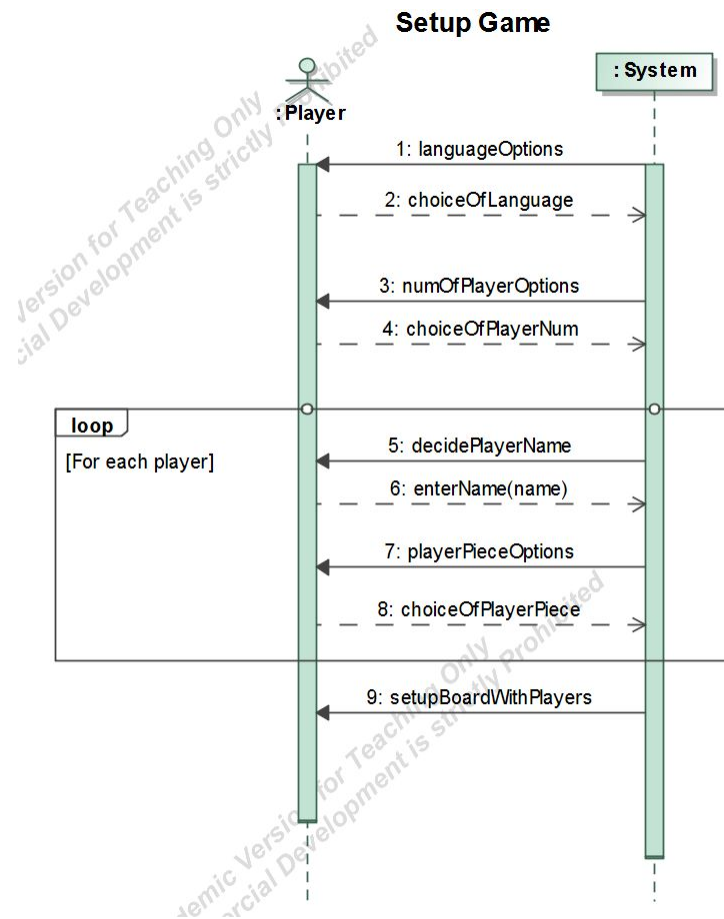
1.4 System sekvens diagram

I **“Play Game”** beder systemet spillerne om at fuldføre **“Setup Game”** og skiftevis **“Take Turn”** indtil der kun er én spiller tilbage i spillet, se figur xxx. Afslutningsvis kåres den tilbageblivende spiller som Matador.



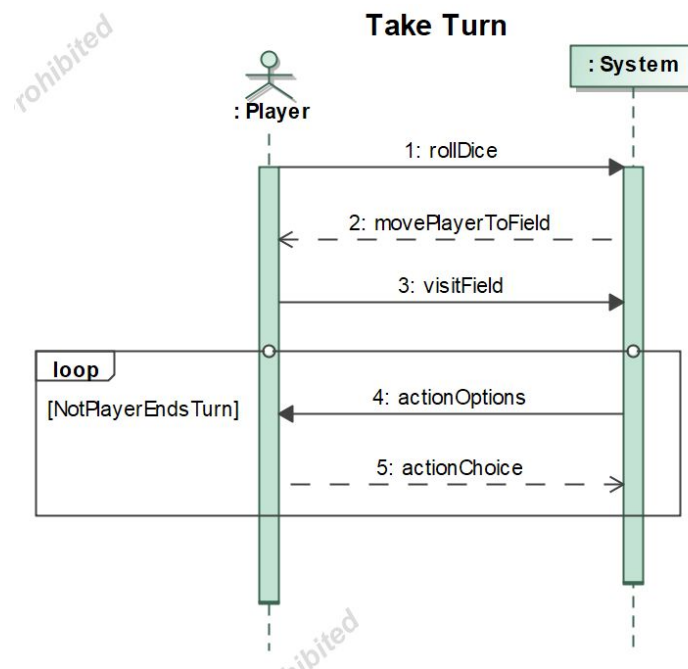
figur xxx

I “**Setup Game**” beder systemet spillerne om at specificere hvilke sprog de ønsker spillet skal være i og antallet af deltagende spillere, se figur xxx. For hver spiller angives spillerens navn og valg af spillebrik. Afslutningsvis opstiller systemet spillebrættet og placerer spillerne på startfeltet.



figur xxx

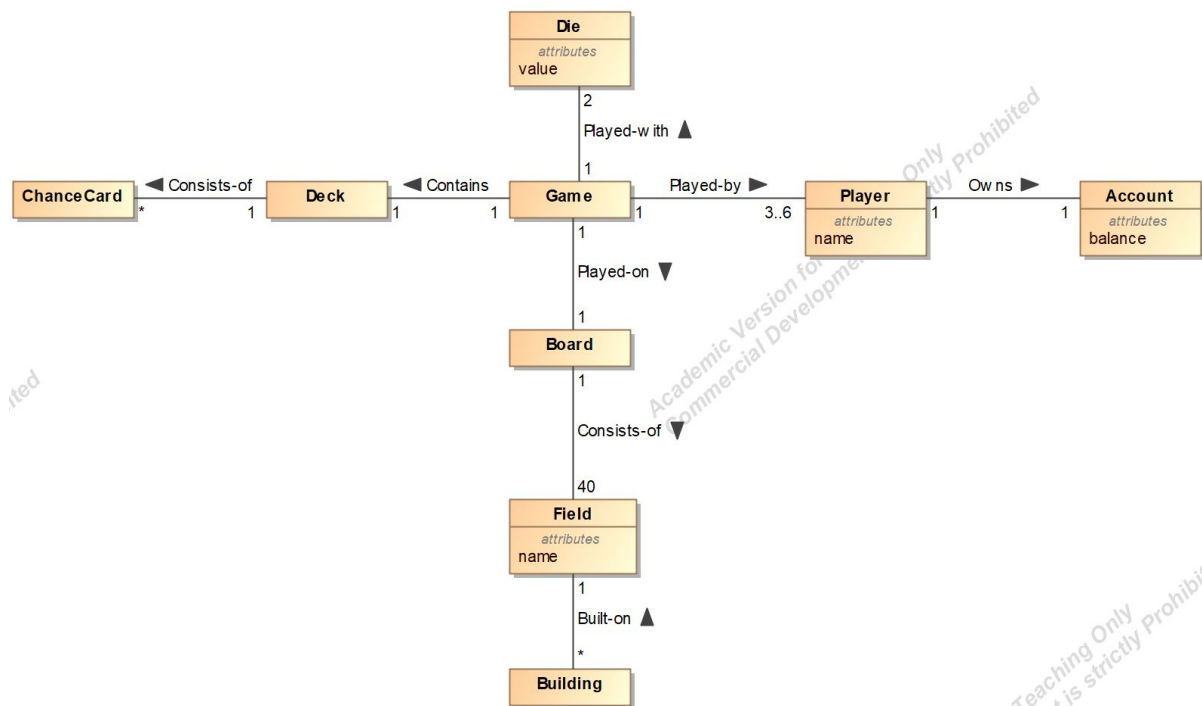
I **“Take Turn”** beder systemet spilleren om at slå med terningerne og terningernes sum angiver, hvor mange felter spilleren rykkes frem, se figur xxx. Effekten af feltet, der landes på, udføres og spillerens balance påvirkes. Ved turens afslutning bliver spilleren præsenteret for følgende aktioner, “sell property”, “trade property”, “pantsæt/pawn property” og “end turn”. Når spilleren ønsker at afslutte sin tur vælges “end turn”.



figur xxx

1.5 Domænemodel

Domænemodellen består af koncepter og deres indbyrdes relationer i den virkelige, ikke-software orienterede, verden. Koncepterne er blevet identificeret ud fra navneord i use case beskrivelserne og har til formål at agere som inspiration til designet af software arkitekturen.



2. Design

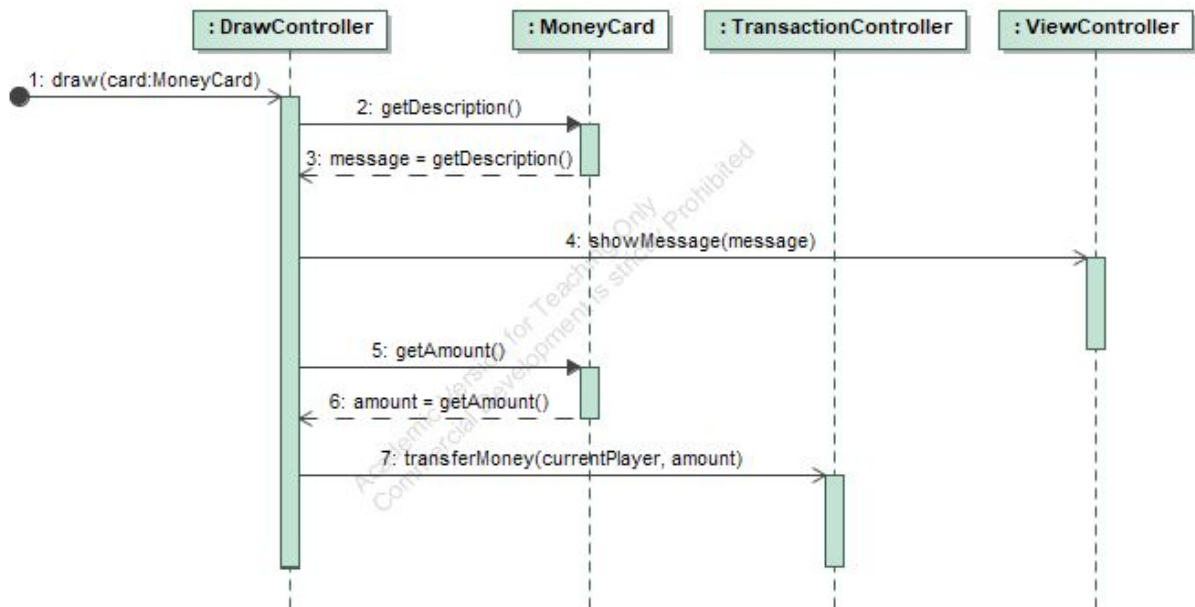
Vores design er opbygget ud fra analysen, GRASP-mønstre, design patterns og møder med kunden og ekstern projektleder.

2.1 Design klassediagrammer

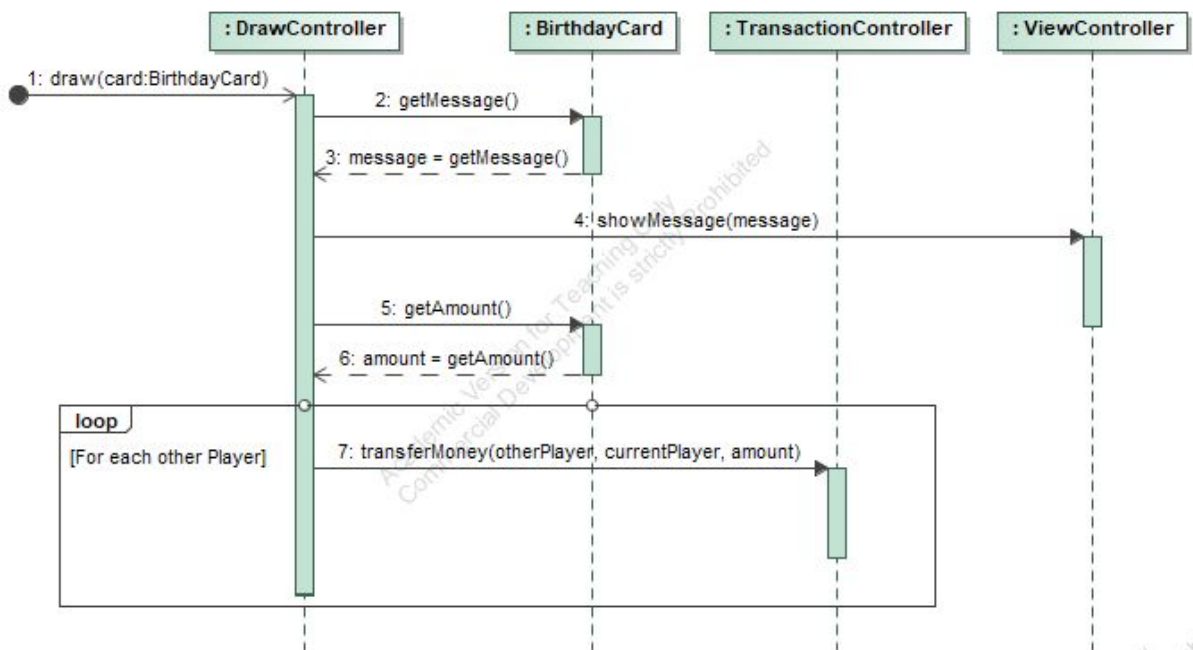
Text

2.2 Sekvensdiagrammer

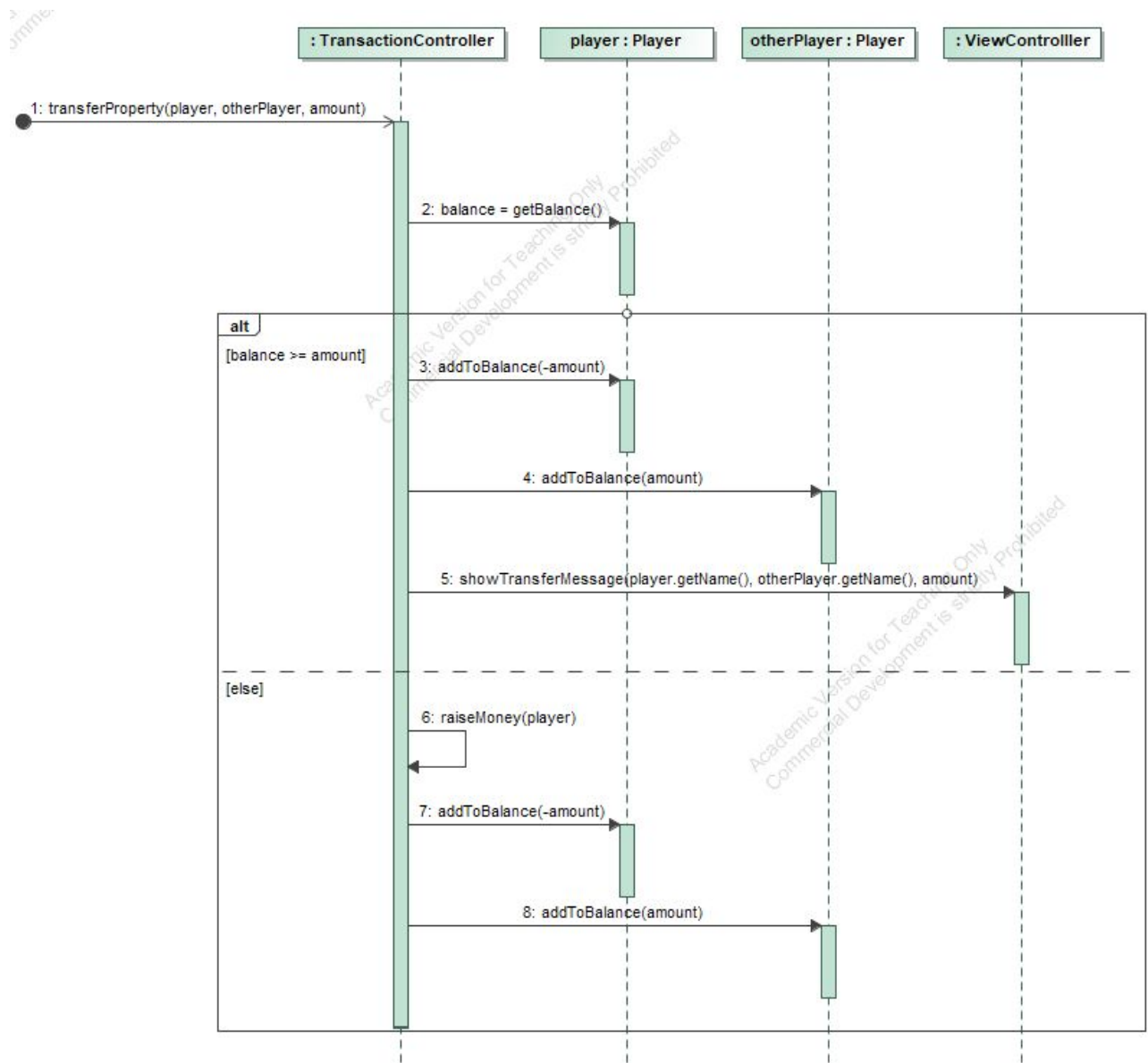
Draw Card



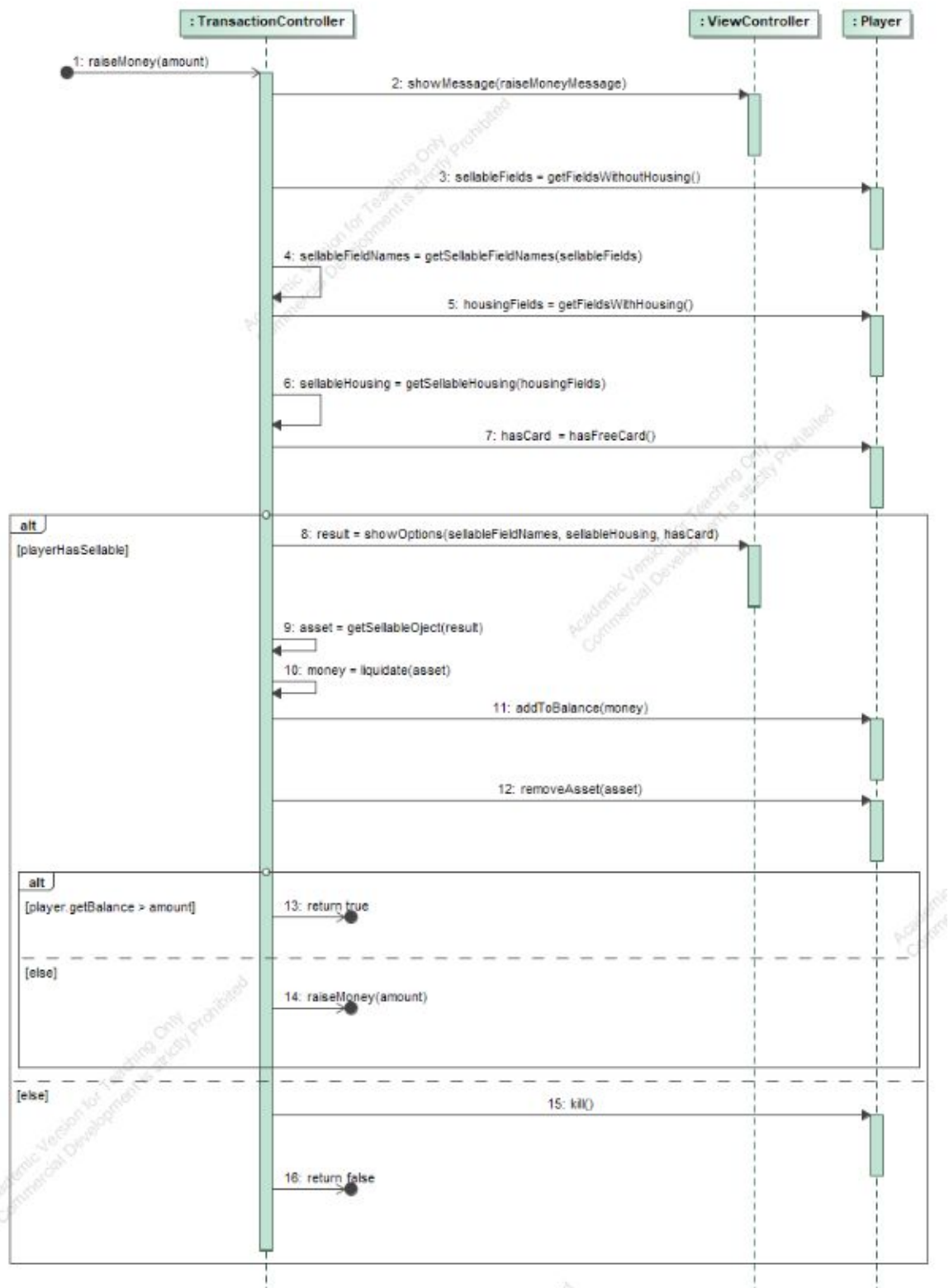
Draw Card



Make Transaction



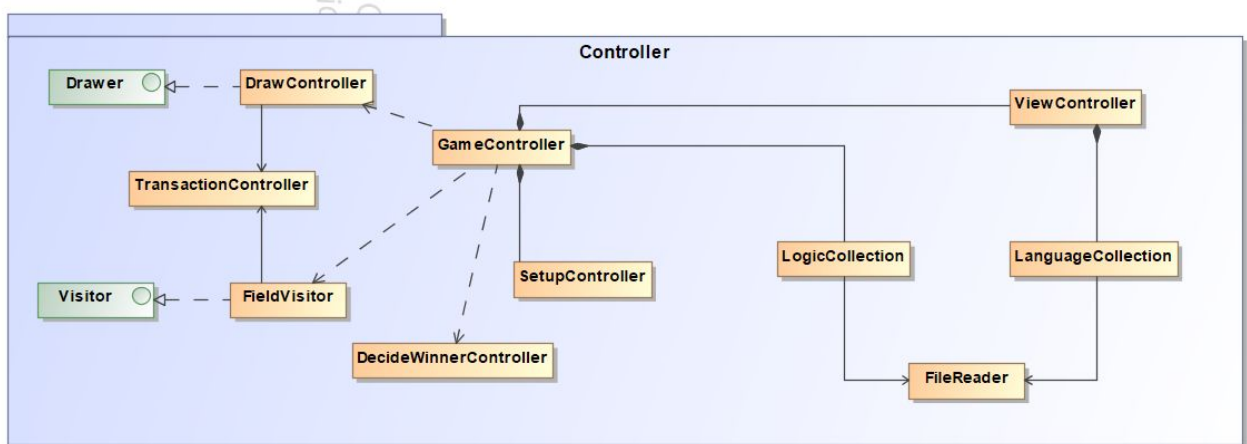
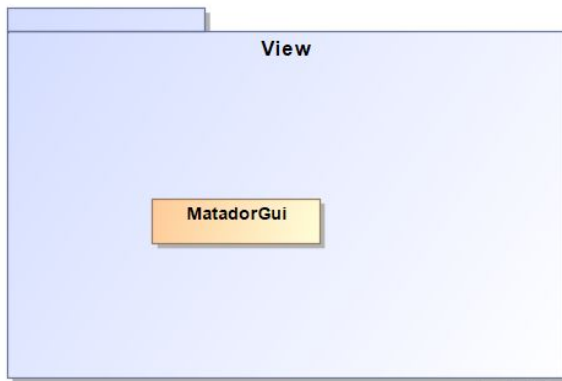
Raise Money

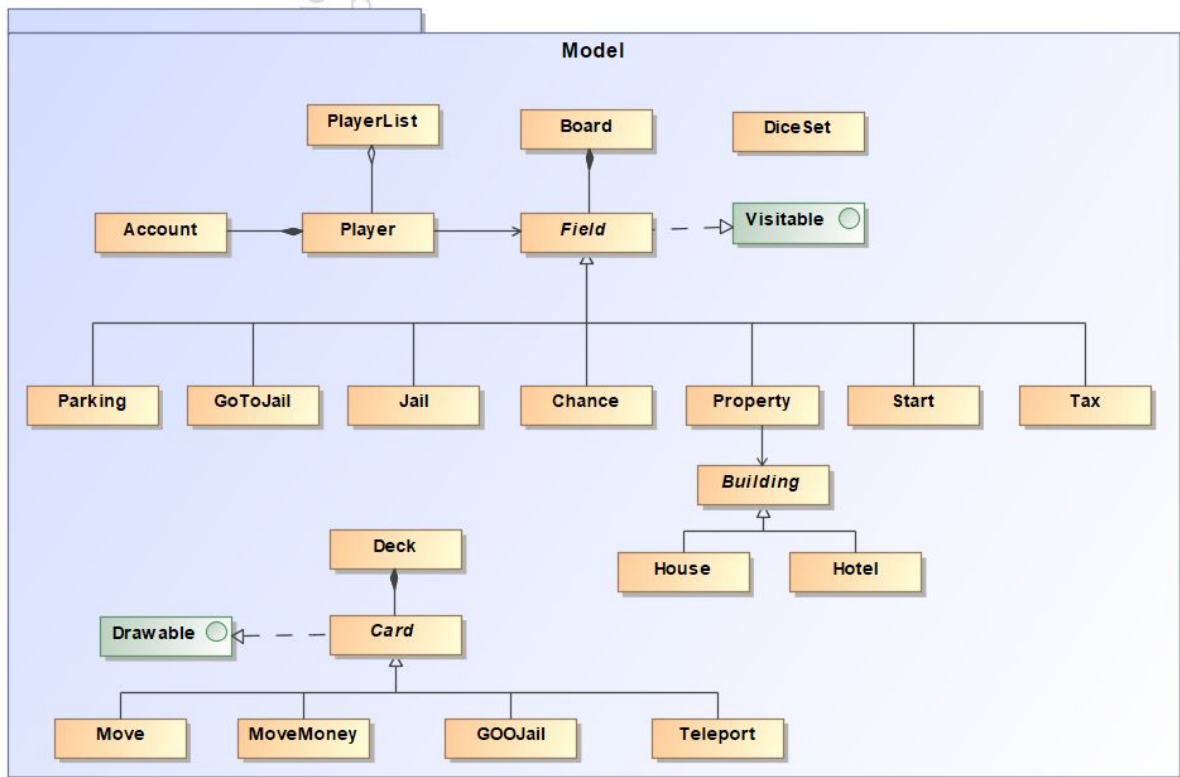


2.3 Arkitektur

MVC

Text





2.4 GRASP

Beskriv gerne hvordan i anvender GRASP, FURPS etc, men ingen forklaringer af koncepter. Betragt det som en rapport til en ligemand

2.4.1 Low Coupling

Text

2.4.2 High Cohesion

Text

2.4.3 Creator

Text

2.4.4 Controller

Text

2.4.5 Information Expert

Text

2.4.6 Polymorphism

Text

3. Implementering

Text (Indeholder både forklaring i tekst og i diagramform)

Udvalgte komplekse kode dele forklares

4. Test

4.1 Struktur

@Before, setUp()

@After, tearDown()

4.2 Testmetoder

Text

4.3 Test cases

Text

4.4 Kode dækning

Vurdering af systemets status kvalitetsmæssigt

Text

4.5 Traceability Matrix

Must have - Matrix

TC	M1	M2	M3	M4	M5	M6	M7	M8	M9	M1 9	M1 1	M1 2	M1 3	M1 4
TC 1	x	x	x	x	x	x								
TC 2														
TC 3														
TC														

4														
TC 5														
TC 6														
TC 7														
TC 8														

Should have - Matrix

Could have - Matrix

5. Konfiguration

Text

5.1 Versionsstyling

Text

6. Projektforløb

Text

7. Konklusion

Text

8. Bilag

8.1 Supplerende specifikation

Ikke-funktionelle krav

IF1: Systemet skal udvikles i Java.

IF2: Systemet skal udvikles i IntelliJ.

IF3: Systemet skal kunne spilles på Windows på databarene på DTU.

IF4: Systemet skal kunne spilles uden bemærkelsesværdige forsinkelser.

IF5: Systemet skal være let oversat.

IF6: Systemet skal være digitalt og dermed ignoreres proceduren for spillets fysiske opsætning samt restriktioner.

8.2 Use cases

Use case: Make Transaction
ID: UC4
Kort beskrivelse: En handel mellem spiller og en anden part.
Primær aktør: Spiller
Precondition: Det er en spillers tur
Main flow: 1. Systemet informerer spiller om handlen. 2. Spiller betaler 200 kr til banken (Raise Money)
Postcondition: Transaktionen er gennemført
Alternative flows: 2.a Spiller indgår i en JailCard handel

1. Spiller har valgt at sælge jailCard
1.1 Systemet overfører jailCard til den modtagende spiller
1.1.1 Systemet overfører penge til den sælgende spiller
2.b Spiller indgår i en Building handel
1. Spiller har valgt at købe building på property
1.1 Systemet trækker building værdien fra spillerens balance
1.1.1 Systemet tilføjer building til spillerens property
2.c Spiller indgår i en Property handel
1. Spiller har valgt at bytte property
1.1 Systemet overfører property til den modtagende spiller
1.1.1 Systemet overfører property til den sælgende spiller
2. Spiller har valgt at købe ledig property
2.1 Systemet trækker property værdien fra spillerens balance
2.1.1 Systemet gør spiller ejer af property
3. Spiller har valgt at sælge property
3.1 Systemet tilføjer property værdien til spillerens balance
3.1.1 Systemet gør banken til ejer af property

Use case: Raise Money
ID: UC5
Kort beskrivelse: Spiller anskaffer penge ved salg af ejendomme.
Primær aktør: Spiller
Precondition: Spiller deltager i en handel hvori Spiller ikke har nok penge.
Main flow: <ol style="list-style-type: none"> 1. Systemet giver spilleren salgsmuligheder 2. Spilleren vælger at sælge ejendomme 3. Spilleren vælger en ejendom at sælge 4. Systemet giver Spilleren penge for ejendommen ud fra dens værdi(Make Transaction) <ol style="list-style-type: none"> 5.1 Spilleren har nok penge

5.1.1 Systemet gennemfører handlen 5.2 Spilleren har ikke nok penge 5.1.1 gentag fra 3.
Postcondition: Spiller har nok penge
Alternative flows: 2.a Spilleren har ingen ejendomme før Spilleren har nok penge 2.a.1 Spilleren overdrager alle sine assets til den Spiller de var i handel med (Make Transaction) 2.a.1.1 Spiller erklæret fallit 2.a.2 Spilleren overdrager alle sine assets til banken(Make Transaction) 2.a.1.2 Spiller erklæret fallit

Use case: Visit Field
ID: UC6
Kort beskrivelse: Spiller lander på felt og feltets effekt udføres.
Primær aktør: Spiller
Precondition: Spiller er landet på et felt
Main flow: 1. Systemet informerer Spiller om feltet 2. Systemet spørger spilleren om de vil købe feltet 3. Spiller køber felt fra banken
Postcondition: Spilleren har fri tur og mulighed for at give turen videre
Alternative flows: 2a Spiller lander på startfelt 1. Spiller modtager kr. 200 fra banken(Make Transaction) 2b Spiller lander på property felt 1. Property er ledig 1.1 Spiller køber property fra banken(Make Transaction) 1.1.1 Spiller har ikke nok penge 1.1.1.1 include(Raise Money) 1.2 Spiller køber ikke property fra banken

<p>1.2.1 Property sættes på auktion</p> <p>2. Property er ejet</p> <p>2.1 Spiller betaler leje til ejer(Make Transaction)</p> <p>2.1.1 Spiller har ikke nok penge</p> <p>2.1.1.1 include(Raise Money)</p> <p>2c Spiller lander på chance flet</p> <p>1. include(Draw Card)</p> <p>2d Spiller lander på fængsels felt</p> <p>1. Spiller er på besøg i fængslet og intet sker i denne runde</p> <p>2e Spiller lander på "gå i fængsel" felt</p> <p>1. Spiller bliver rykket til fængsels felt</p> <p>2. Systemet sætter spiller i fængsel</p> <p>2f Spiller lander på parkering felt</p> <p>1. Spiller holder på parkeringspladsen og intet sker i denne runde</p> <p>2g Spiller lander på tax felt</p> <p>1. Spiller betaler tax/skat til banken(Make Transaction)</p> <p>1.1 Spiller har ikke nok penge</p> <p>1.1.1 include(Raise Money)</p>

Use case: Draw Card
ID: UC7
Kort beskrivelse: Spiller trækker et chancekort og dets effekt udføres..
Primær aktør: Spiller
Precondition: Spiller har landet på et "Prøv Lykken" felt.
Main flow:

1. Systemet informerer Spiller om kortet.
2. Spiller modtager 50,00 kr. fra banken.

Postcondition: Spilleren har fri tur og mulighed for at give turen videre

Alternative flows:

2a. GetOutOfJailCard

- 1.1 Spiller vælger at beholde kort
 - 1.1.1 Systemet tildeler kortet til spilleren
- 1.2 Spiller vælger sælge kort til en anden spiller
 - 1.2.1 Hver gang det er spillerens tur, kan spilleren vælge at sælge kortet
 - 1.2.1.1 Spilleren sælger til en bestemt spiller
 - 1.2.1.1.1 Spilleren sætter en bestemt pris
 - 1.2.1.1.1.1 Den anden spiller køber
 - 1.2.1.1.1.2 Den anden spiller køber ikke
 - 1.2.1.1.1.2.1 kortet bliver ikke solgt og spiller kan-prøve at sælge igen i næste runde
 - 1.2.1.2 Spilleren sætter kortet på auktion
 - 1.2.1.2.1 Alle andre spillere indskriver en pris
 - 1.2.1.2.1.1 Spilleren som har angivet den højeste pris-køber automatisk feltet

2b MonopolyJackpotCard

- 1 Systemet undersøger om Spillers formue er over kr. 750.
 - 1.1 Spillers formue er over 750 kr.
 - 1.1.1 Spiller modtager ikke Matador-legatet
 - 1.1.1.2 Spiller modtager ikke de 2000 kr.
 - 1.2 Spillers formue er under 750 kr.
 - 1.2.1 Spiller modtager Matador-legatet
 - 1.2.1.2 Spiller modtager de 2000 kr. fra banken
(Make Transaction)

2c MoveCard

- 1.1 Spillers brik bliver rykket til Grønningen feltet
 - 1.1.1 Spiller passere ikke start
 - 1.1.1.1 Spiller modtager ikke 200 kr.
 - 1.1.2 Spiller passere start
 - 1.1.2.1 Spiller modtager 200 kr. (Make Transaction)
- 1.2 Spillers brik bliver rykket til næste Øresund færgefelt
 - 1.1.1 Spiller passere ikke start
 - 1.1.1.1 Spiller modtager ikke 200 kr.
 - 1.1.2 Spiller passere start
 - 4.1.2.1 Spiller modtager 200 kr. (Make Transaction)
- 1.3 Spillers brik bliver rykket 3 felter tilbage

1.4 Spillers brik bliver rykket frem til start

2d PayForBuildingsCard

1.1 Spiller skal betale ekstra udgift pr hus eller hotel de ejer, da kul-og kokspriserne er steget

1.1.1 Spiller har ikke bygninger på nogle af sine felter

1.1.1.1 Spiller skal ikke betale

1.1.2 Spiller har bygninger på sine felter

1.1.2.1 Spiller skal betale 25 kr. per hus

1.1.2.1.1 Spiller skal betale 125 kr. per hotel

1.2 Spiller skal betale ekstra udgift pr hus eller hotel de ejer, da ejendomsskatterne er steget

1.2.1 Spiller har ikke bygninger på nogle af sine felter

1.2.1.1 Spiller skal ikke betale

1.2.2 Spiller har bygninger på sine felter

1.2.2.1 Spiller skal betale 50 kr. per hus

1.2.2.1.1 Spiller skal betale 125 kr. per hotel

2e TeleportAndPayDoubleCard

1.1 Spiller bliver rykket frem til det nærmeste dampskibsselskab

1.1.1 Systemet tjekker om feltet er ejet af en anden spiller

1.1.1.1 Spiller betaler ejeren to gange den leje, han ellers er berettiget til

1.1.1.2 Spiller kan købe feltet af banken

2f TeleportCard

1.1 Spiller bliver rykket direkte i fængsel

1.1.1 Spiller modtager ikke 200 kr. hvis de passere start

1.2 Spiller bliver rykket ind til Rådhuspladsen

2g Birthday Card

1. Spiller modtager 25 kr. fra de andre spillere

1.1 include(Make Transaction)

2h MoneyCard

1.1 Spiller skal modtage

1.1.1 Spiller modtager 200 kr. fra banken (Make Transaction)

1.1.2 Spiller modtager 25 kr. fra banken (Make Transaction)

1.1.3 Spiller modtager 20 kr. fra banken (Make Transaction)

1.1.4 Spiller modtager 1 kr. fra banken (Make Transaction)

1.1.5 Spiller modtager 108 kr. fra banken (Make Transaction)

1.1.6 Spiller modtager 100 kr. fra banken (Make Transaction)

1.2 Spiller skal betale

1.2.1 Spiller betaler 20 kr. til banken (Make Transaction)

<p>1.2.2 Spiller betaler 100 kr. til banken (Make Transaction)</p> <p>1.2.3 Spiller betaler 10 kr. til banken (Make Transaction)</p>
--

8.3