



COPENHAGEN SCHOOL OF  
DESIGN AND TECHNOLOGY

*Date: 01-06-2022*

# Projekt Rapport *Bilabonnement*

***DAT21d***

**Group: N/A**

## **Project Participants:**

Sofia Emilia Abrahamsen – 25/10/1996

*Mathias Eliot Nielsen – 6/10/1996*

*Tobias Hammeken Arboe – 14/06/1997*

*Lasse Bøgeskov-Jensen – 23/06/2000*

## Document Template History

Date	Version	Author	Description
201x-xx-xx	00.01.00	All	Initialized

## Document Version

/\* TO BE REMOVED...

- Who has last updated and for what reason?
- What has changed?

\*/

Date	Version	Author	Description
2022-05-03	00.01.00	All	Initialized

Document Template History .....	2
Document Version .....	2
Contents .....	4
Problemstilling .....	5
Problemformulering .....	5
Interessent analyse .....	5
Feasibility Study .....	7
Economic feasibility .....	7
Technical feasibility .....	7
Operational feasibility .....	8
Schedule feasibility .....	8
Alternative software-løsninger .....	8
Konklusion .....	8
Vision .....	9
Business Rules / Domain Rules .....	10
Requirements .....	11
Supplementary Specification .....	11
FURPS .....	12
.....	12
.....	12
.....	12
.....	12
.....	12
(FURPS)+ .....	14
.....	14
.....	14
.....	14
.....	14
Domain Model .....	15
Use Cases .....	15
Use Case Diagram .....	15
Use Case .....	15
Login .....	16
Logout .....	16
Create user .....	17
Edit user .....	17
Delete user .....	19
View users .....	19

System Sequence Diagram .....	20
Operation Contracts .....	21
Package Diagram.....	21
Design Class Diagram .....	21
Sequence Diagram .....	21
GRASP.....	22
State Machine diagram.....	22
“Optional” – Other Information? <sup>[OBJ]</sup> .....	22
Glossary.....	23
Business .....	23
Unified Process - Iteration Plan .....	24
REVIEW.....	26
Test Cases.....	27
Litteraturliste .....	28

## Contents

/\* TO BE REMOVED...

**Remember** to edit or delete all comment sections marked /\* \*/

**“Optional” sections** – Do you need them?

**Remember** to update Contents.

\*/

# Problemstilling

Af Sofia, Lasse og Mathias

Vi er blevet stillet til opgave at lave en prototype af et system til biludlejningsfirmaet Bilabonnement.dk. De er en virksomhed som arbejder indenfor udlejning og leasing af personbiler til privatpersoner. De har efterspurgt et system der kan medvirke til at deres service bliver bedre for deres kunder, og som hjælper firmaets ansatte med at registrere og administrere firmaets data. Firmaet efterspørger en applikation med et interface for hver af virksomhedens afdelinger.

Dataregistrering ønsker at kunne registrere nye lejeaftaler. Skade- og udbedring ønsker at kunne registrere fejl- og mangler samt prisen pr. Fejl. Forretningsudviklerne ønsker at kunne se hvor mange biler der er lejet ud samt den samlede pris for udlejede biler.

## Problemformulering

Af Sofia, Lasse og Mathias

Ud fra vores problemstilling har vi valgt at formulere følgende spørgsmål til vores problemformulering:

1: Hvordan skal systemet udvikles?

- Hvordan vil vi tilgå udarbejdelsen af systemet? (UP, UML, GRASP, FURPS+)
- Hvordan undersøger vi om projektet kan betale sig? (feasability study)
- Hvilken udviklingsmetode bruger vi? (TDD - test driven development)

2: Hvilke tekniske overvejelser skal vi gøre os?

- Hvilke frameworks skal vi bruge og hvorfor? (Spring)
- Hvordan skal databasen sætte op? (MySQL workbench)
- Hvordan kan vi hoste en løsning hos en cloud udbyder? (teknologi krav - offline)

## Interessent analyse

Af Sofia, Lasse og Mathias

Vi har fundet disse interessenter i projektet:

**Bestyrelsesformand:** Kenneth Keller Hansen (K.W. Bruun, 2022)

**Direktør:** Carsten Stephensen (Find-virksomhed.dk, 2022)

**Medarbejdere:**

- Dataregistrering-medarbejdere
- Skade- og udbedrings-medarbejdere
- Forretningsudviklere

**Samarbejdspartnere:** (Bilabonnement, 2022)

- Jyske Finans A/S
- TopDanmark Forsikring A/S
- FDM
- Offentlige myndigheder
- m.fl.

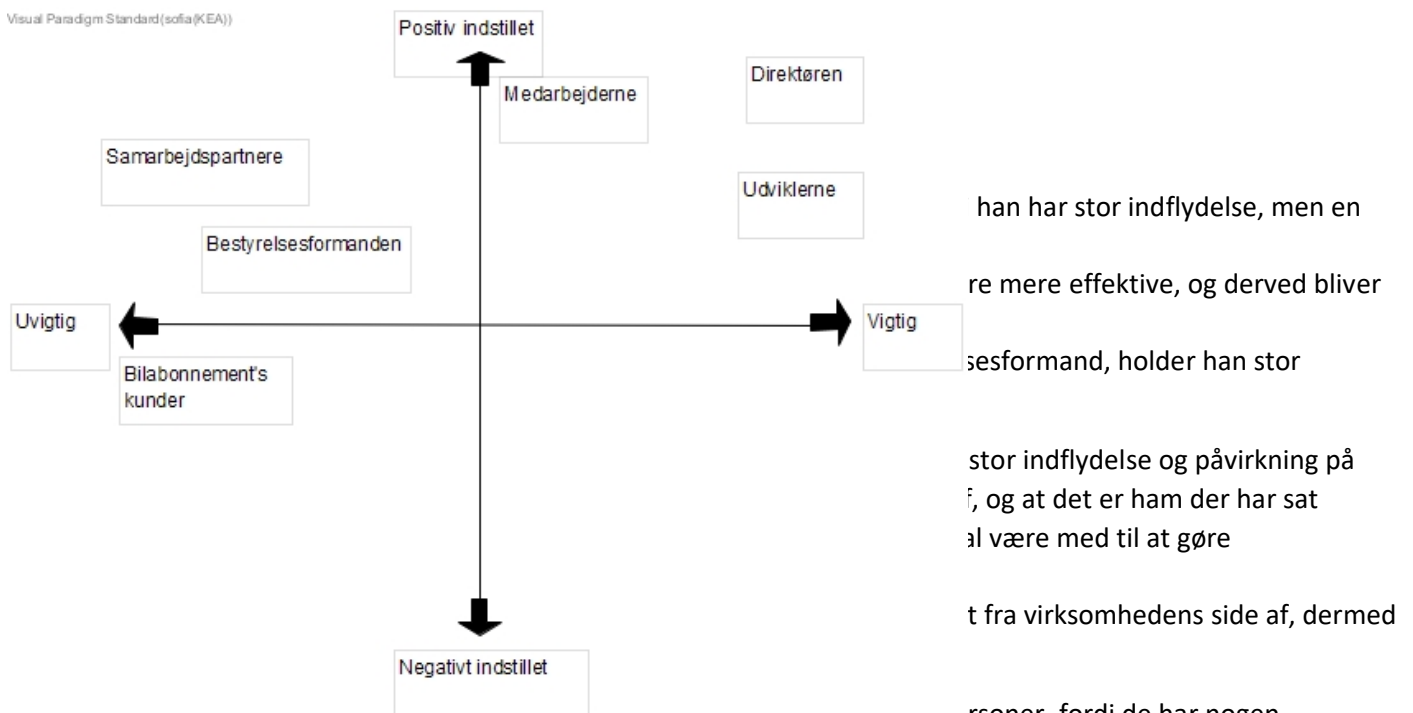
**Udviklere:**

- Sofia Emilia Abrahamsen
- Mathias Eliot Nielsen

- Tobias Hammeken Arboe
- Lasse Bøgeskov-Jensen

Nedenstående billede er en model med kategorisering af interessenterne:

Visual Paradigm Standard (sofia@KEA)



Samarbejdspartnere, bestyrelsesformanden og bilabonnements kunder er eksterne interessenter, fordi de har nogen indflydelse på projektet, da det er dem som kommer til at være primærbrugere på systemet, og derfor kommer til at indgå i udviklingen af systemet. Dog er de heller ikke naturlige ressourcepersoner, da de ikke er direkte indblandet i projektet, til dagligt.

Vi formoder, at deres krav til projektet er at gøre deres hverdag nemmere ved at holde data gemt ét sted. Da de bruger Excel ark på nuværende tidspunkt, er et databasesystem også en bedre løsning ift. at minimere fejl fra deres side af.

Så er der os, udviklerne. Vi er ressourcepersoner, da vi har stor betydning og påvirkning af dette projekt, da det er os der udvikler det. Vi har til formål at levere et godt produkt, opfylde kundens krav, samt at blive betalt for det arbejde vi laver.

Vi har undersøgt og fundet flere samarbejdspartnere af bilabonnemnet.dk's, vi sætter dem under eksterne interessenter, da de ikke har indflydelse på projektet og bliver minimalt påvirket.

Deres primære interesse er reklamation fra bilabonnemnet.dk, og derfor er påvirkningen hos dem afhængig af bilabonnemnet.dk's succes og deraf succes af projektet.

Bilabonnemnet's kunder er ligesom samarbejdspartnerne eksterne interessenter, som ikke har indflydelse på projektet og de bliver minimalt påvirket. Kundernes påvirkning kommer i form af eventuel hurtigere service, hvis systemet forbedrer medarbejdernes effektivitet. Der er dog en chance for at de kan blive negativ påvirket, hvis implementeringen af systemet medfører ringere service i overgangsperioden.

Til sidst har vi lavet en prioriteringsrækkefølge af vigtigheden af interessenter:

1. Direktør/bestyrelsesformand
2. Medarbejderne
3. Samarbejdspartnere, Bilabonnemnet's kunder

Vi har valgt at prioritere interessenterne, således, fordi vi tænker at direktøren og bestyrelsesformanden har størst betydning for projektet, tæt efterfulgt af medarbejderne, da de bliver primærbrugere. Til sidst har vi samarbejdspartnere og bilabonnemnet's kunder, da de kun er sekundært afhængige af systemet.

# Feasibility Study

Af Tobias

## Economic feasibility

Vi er stadig tidligt i udviklingsprocessen, så mange af de økonomiske fordele er "ikke håndgribelige" lige nu, omend de vil være længere henne i processen.

De ikke håndgribelige fordele involverer:

- "Organisational efficiency" ; systemet vil på sigt forøge effektiviteten for de ansatte
- "Improvement of management planning and control"\* ; systemet vil gøre det nemmere at få overblik over biler, om de skal repareres, og ville idealt involvere notifikationer om mangler, hvilket vil hjælpe management
- "Increased speed of activity"\* ; på same måde som overblik vil hjælpe management kommer det også til at hjælpe både "Skade og udbredning", samt "Forretningsudvikler" afdelingerne med at gennemfører nogle af deres arbejdsaktiviteter langt hurtigere.

Punkterne med \* vil længere i processen potentielt kunne rykkes ned til håndgribelige fordele når man får en ide om hvor meget tid og penge de reelt bespare.

På den anden side har vi nogle omkostninger; her er der primært tale om håndgribelige omkostninger. De bliver heri opgjort delvist i kroner, og delvist i mandetimer, henholdsvis pr. ansat – altså hvor mange mandetimer en kategori ville koste en ansat – og pr. opgave.

Tilbagevendende omkostninger:

- Drift og vedligeholdelse af database (20.000 årligt) **NOTE: UDREGN**
  - o Betaling for at holde databasen kørende, samt serviceaftale om sikret antal service-timer pr. måned

Engangsomkostninger:

- "Træning" i brug af systemet (1 time pr. ansat)
  - o De ansatte skal vises systemet og bruge tid på at sætte sig ind i det
- Overførsel af data (3-4 timer total)
  - o Der er en masse data der skal overføres fra det gamle system (eg. kunder, biler mm.) til det nye system
- "Udviklingsomkostninger"
  - o Da vi er en gruppe studerende, er vi som sådan gratis arbejde; havde vi været et professionelt team ville løn ligge her; ligeledes ville der være en kategori som hed "timer mistet på samarbejde" hvor vi, som udviklere, skulle ud og snakke med folk i virksomheden.

## Technical feasibility

Når man snakker om "technical feasibility" snakker man om man har de tekniske evner til at udføre projektet. Dette "opgøres" i risiko – altså hvor mange risici der er ved at udføre projektet; dette kan opgøres i fire overordnede kategorier:

- Projektets størrelse
- Projektets struktur
- Udviklergruppen
- Brugergruppen

Hver kategori kan have emner som forøger eller formindsker risikoen for en given kategori, og hvert emne vægter selvfølgelig ikke nødvendigvis ens.

Projektets størrelse:

- Vi er en relativ lille projektgruppe;
- Har relativ lang tid til at lave et minimum-viable product
- Alt sker "in-house"
- Og fleksible arbejdstider til at opnå alle arbejdsmaal.

#### Projektets struktur:

- Nyt system
- Høj commitment fra management
- Systemet fikser eksisterende problemer som brugeren efterspørger

#### Udviklingsgruppen:

- Nogen familiaritet med hardwaren, høj overordnet familiaritet med softwaren og værktøjerne til udvikling
- Nogen til høj erfaring med at lave systemer af lignende størrelse og kompleksitet

#### Brugergruppen

- Brugergruppen har nogle erfaringer med lignende systemer
  - o Omen ikke nødvendigvis ret meget
- Der har formodentlig været nogen form for indflydelse fra brugergruppen/rne omkring hvordan systemet skal udvikles

Overordnet set er der relativ lav risiko forbundet med projektet; der er klare og strukturerede mål at arbejde efter, projektstørrelsen er overskuelig og der bliver ikke brugt noget teknologi fra vores (eg. udviklingsgruppen) som vi ikke har arbejdet med før.

#### Operational feasibility

"Operational feasibility" har stort fokus på om man opnår de mål man sætter sig for med projektet; i det her tilfælde er der, som tidligere nævnt, meget faste og tydelige mål at arbejde ud efter.

Det må derfor antages at, såfremt de mål mødes, antages projektet som en succes.

Dog er det også en fair antagelse et system der er mere effektivt end det nuværende potentielt kan påvirke firmaets struktur og deres processer.

#### Schedule feasibility

Tidsmæssigt er projektet meget overskueligt; en måned til at nå et "minimum viable produkt" er meget gøreligt, der er tid inkluderet for testing og selv om der er udvidede system-krav for projektet så er det ikke mandatoriske; hvis vi vælger at forfølge dem kan de blive planlagt for.

#### Alternative software-løsninger

Vi har kigget på en række programmer som potentielt kunne løse behovet;

(indsæt liste)

Vi vil anbefale firmaet at undersøge om disse muligheder løser deres softwarebehov. Ikke desto mindre er dette en eksamensopgave, og vi vil derfor stadig fortsætte med udviklingen.

#### Konklusion

Alt i alt er projektet meget gennemførligt; der er nogle få barrierer og risici som skal tages højde for, men det kan planlægges omkring.

På den anden side har vi et produkt som løser problemer for firmaet og som de individuelle afdelinger virker til at have haft indflydelse på, hvilket gør at modstanden "in-house" er mindre for omskiftning til det nye software.

Ligeledes har vi fornuftig tid til at færdiggøre et projektet der har et overskueligt "scope".



## Vision

Af Sofia, Lasse og Mathias

Vores vision med projektet er at strømline arbejdsprocesser for de ansatte, så de slipper for at lave unødigt og besværligt arbejde og kan fokusere på det arbejde der giver værdi for dem og firmaet.

Bilabonnement ønsker et system til at håndtere arbejdsopgaver, som før blev lavet i flere excel ark, i et samlet system.

### Opsummering af systemets funktioner:

Systemet håndterer bilens rejse fra afhentningssted til skadesregistrering.

De vigtige funktioner i systemet inkluderer:

- Registrering af nye lejeaftaler
- Registrering af fejl og mangler, når en bil er afleveret og har overstået leje periode
- Overvågning af biler, status og lejekontrakter
- Rapportering af statistik
- Web-baseret
- Flerbruger system
- MySQL database

/\* TO BE REMOVED...

### The GOAL!

It defines the stakeholders view of the product/service to be developed, specified in terms of the stakeholders key needs and features. Containing an outline of the envisioned core requirements.

### Summary of System Features

#### Definition

Features are behavioral functions a system can do. They should pass this linguistic test:  
The system does <feature X>.

For example:

The system does payment authorization.

The major features include:

- POS services
- Inventory management
- Web-based shopping
- ...

### Other Requirements in the Vision?

In the Vision, system features briefly summarize functional requirements often detailed in the use cases. Likewise, the Vision can summarize other requirements (for example, reliability and usability) that are detailed in the Supplementary Specification. But be careful to avoid going down the path of repeating yourself.

### See Larman for more information:

Chapter 7.6. NextGen Example: (Partial) Vision

Chapter 7.7. Commentary: Vision

\*/

## Business Rules / Domain Rules

/\* TO BE REMOVED...

### **Can be part of the Supplementary Specification (SS)**

Typically describe requirements or policies that transcend one software project—they are required in the domain or business, and many applications may need to conform to them. An excellent example is government tax laws. Domain rule details may be recorded in the Supplementary Specification, but because they are usually more enduring and applicable than for one software project, placing them in a central Business Rules artifact (shared by all analysts of the company) makes for better reuse of the analysis effort.

### **See Larman for more information:**

Chapter 5.5. How are Requirements Organized in UP Artifacts?

Chapter 7.10. NextGen Example: Business Rules (Domain Rules)

Chapter 7.11. Commentary: Domain Rules

\*/

## Requirements

Af Sofia, Mathias og Tobias

De informationer vi har fået omkring krav til systemet er:

Req001 - Nye lejeaftaler skal kunne registreres af Dataregistreringsmedarbejdere.

Req002 - Fejl og mangler, samt pris pr. fejl, på biler der er blevet tilbageleveret og har overstået lejeperioden, skal kunne registreres af Skade- og udbedringsmedarbejdere.

Req003 - Forretningsudviklerne vil gerne kunne se hvor mange biler der er lejet ud, samt hvad den sammenlagte pris er på nuværende lejede biler.

## Supplementary Specification

/\* TO BE REMOVED...

Elements of the Supplementary Specification include:

- reports ---- ??????
- hardware and software constraints (operating and networking systems, ...)
- development constraints (for example, process or development tools)
- other design and implementation constraints
- internationalization concerns (units, languages)
- documentation (user, installation, administration) and help
- licensing and other legal concerns
- packaging
- standards (technical, safety, quality)
- physical environment concerns (for example, heat or vibration)
- operational concerns (for example, how do errors get handled, or how often should backups be done?)
- application-specific domain rules
- information in domains of interest (for example, what is the entire cycle of credit payment handling?)

### Use Either FURPS+ or Functional/Non-Functional

basically, everything not in the use cases. This artifact is primarily for all non-functional requirements, such as performance or licensing. It is also the place to record functional features not expressed (or expressible) as use cases; for example, a report generation.

### See Larman for more information:

Chapter 5.4. What are the Types and Categories of Requirements?

Chapter 5.5. How are Requirements Organized in UP Artifacts?

\*/

## FURPS

### Functional

*/\* To be removed — features, capabilities, security. \*/*

Id	Requirement	Comment
FR001	Login	
FR002	Logout	
FR003	CRUD User	
FR004	CRUD RentalAgreement	
FR005	CRUD DamageReport	
FR006	CRUD Car	
FR007	View amount of rented cars	
FR008	View total price of rented cars	
FR009	View dashboard	
FR010		

### Usability

*/\* To be removed — human factors, help, documentation. \*/*

Id	Requirement	Comment
UR001	Readability over looks	Big text and clear buttons are more important than clean looks
UR002	As much as possible, have consistent layout	
UR003	When filling forms (and so forth) keep temporary backups of the data in the cells	

### Reliability

*/\* To be removed — frequency of failure, recoverability, predictability. \*/*

Id	Requirement	Comment
RR001	Keep alive function	
RR002	Enforce datatypes in forms	
PR003		

### Performance

*/\* To be removed — response times, throughput, accuracy, availability, resource usage. \*/*

Id	Requirement	Comment
PR001	Faster than doing it by hand in excel	
PR002	...	

### Supportability

*/\* To be removed — adaptability, maintainability, internationalization, configurability. \*/*

Id	Requirement	Comment
SR001	Well documented code	
SR002	...	



## (FURPS)+ Implementation

/\* To be removed — resource limitations, languages and tools, hardware, ... \*/

Id	Requirement	Comment
IMPR001	Java	
IMPR002	HTML	
IMPR003	CSS	
IMPR004	Thymeleaf	
IMPR005	Spring	

## Interface

/\* To be removed — constraints imposed by interfacing with external systems. \*/

Id	Requirement	Comment
INTR001	MySQL - JDBC	

## Operations

/\* To be removed — system management in its operational setting. \*/

Id	Requirement	Comment
OR001	Tomcat	

## Packaging

/\* To be removed — for example, a physical box. \*/

Id	Requirement	Comment
PACR001	Git-repo	

## Legal

/\* To be removed — licensing and so forth. \*/

Id	Requirement	Comment
LR001	MySQL	
LR002	GDPR loven	

## Domain Model

/\* TO BE REMOVED...

A domain model is the most important—and classic—model in OO analysis.

Motivation: Lower Representational Gap with OO Modeling

### Objectives

- Identify conceptual classes related to the current iteration.
- Create an initial domain model.
- Model appropriate attributes and associations.

**See Larman for more information:**

Chapter 9. Domain Models

\*/

## Use Cases

### Use Case Diagram

/\* TO BE REMOVED...

Insert your Use Case Diagram here.

\*/

### Use Case

/\* TO BE REMOVED...

- Description of each Actor and Goals  
**See Larman for more information:** Chapter 6.14. Guideline: Take an Actor and Actor-Goal Perspective
- Short Use Case Model description
- Use Case Diagram(s)
- Use Case 1 text – Brief/Casual/Fully Dressed
- Use Case 2 text – Brief/Casual/Fully Dressed
- Use Case 3 text – Brief/Casual/Fully Dressed
- ...

### Remember to add

“: INCLUDE *UCxx IncludedUseCaseName.*” in main use case step where INCLUDE is used.

Supporting **Use Case sections** template created by Alistair Cockburn

Use Case Section	Comment
Use Case Name	Start with a verb.
Scope	The system under design.
Level	“user-goal” or “subfunction”
Primary Actor	Calls on the system to deliver its services.
Stakeholders and Interests	Who cares about this use case, and what do they want?
Preconditions	What must be true on start, and worth telling the reader?
Success Guarantee	What must be true on successful completion, and worth telling the reader.
Main Success Scenario	A typical, unconditional happy path scenario of success.
Extensions	Alternate scenarios of success or failure.
Special Requirements	Related non-functional requirements.

Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Influences investigation, testing, and timing of implementation.
Miscellaneous	Such as open issues.

## Login

Use Case Section	Comment
Use Case Name	Login
Scope	Bilabonnement system
Level	Subfunction
Primary Actor	Alle brugere: System administrator, dataregistrering, skade- og udbedring, forretningsudviklere
Stakeholders and Interests	Alle brugere der skal bruge systemet, er tvunget til at logge ind først, derfor er alle brugere stakeholders.
Preconditions	Ingen?
Success Guarantee	Brugeren har et gyldigt login til siden. Brugeren bliver fremvist den korrekte side svarende til deres afdeling.
Main Success Scenario	<ul style="list-style-type: none"> <li>- Brugeren besøger siden</li> <li>- Bliver præsenteret med en loginside.</li> <li>- Brugeren indtaster det personlige brugernavn og kodeord</li> <li>- Brugeren bliver derefter mødt med den side der repræsenterer deres afdeling i systemet.</li> </ul>
Extensions	Et alternativt flow sker hvis brugeren indtaster et forkert brugernavn eller kodeord. Brugeren vil så blive mødt med en advarsel om at det indtastede kodeord eller brugernavn er forkert.
Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Login bliver brugt hver eneste gang en bruger vil tilgå systemet
Miscellaneous	Such as open issues.

## Logout

Use Case Section	Comment
Use Case Name	Logout
Scope	Bilabonnements system
Level	subfunction
Primary Actor	Alle brugere: System administrator, dataregistrering, skade- og udbedring, forretningsudviklere
Stakeholders and Interests	Alle brugere af systemet er interesseret i at logge ud af systemet, for at undgå at andre kan tilgå deres data.
Preconditions	Brugeren er logget ind på system siden
Success Guarantee	Brugeren er blevet logget ud af systemet
Main Success Scenario	<ul style="list-style-type: none"> <li>- Brugeren er logget ind med deres personlige login</li> <li>- Brugeren trykker på log ud knappen på siden</li> <li>- Brugeren bliver logget ud af systemet</li> <li>- Brugeren bliver vist en bekræftelse på log ud</li> </ul>



Extensions	Ingen
Special Requirements	Mus, computeren skal have adgang til databasen.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Hver gang en bruger er færdig med at bruge systemet og trykker på logud.
Miscellaneous	Such as open issues.

## Create user

Af Mathias

Use Case Section	Comment
Use Case Name	Create user
Scope	Bilabonnements system
Level	Subfunction
Primary Actor	Systemadministrator
Stakeholders and Interests	Systemadministratoren har interesse i at oprette brugere til de ansatte. De ansatte har interesse i at modtage en personlig bruger med personligt brugernavn og kodeord
Preconditions	Systemadministratoren er i forvejen logget ind med et gyldigt login der giver adgang til systemadministrator funktioner.
Success Guarantee	En gyldig bruger med et givet funktionsniveau er oprettet med et originalt brugernavn og et tildelt kodeord
Main Success Scenario	<ul style="list-style-type: none"> <li>- Brugeren tilgår bilabonnement systemet</li> <li>- SYSTEM: Returnere login-side</li> <li>- Brugeren logger ind med et gyldigt systemadministratorlogin</li> <li>- SYSTEM: Returnere admin-dashboard</li> <li>- Brugeren vælger funktionen "opret ny bruger"</li> <li>- SYSTEM: Returnere opret-bruger-side</li> <li>- Brugeren indtaster et originalt brugernavn og kodeord</li> <li>- SYSTEM: tjekker om det er tilgængeligt</li> <li>- SYSTEM: opretter bruger</li> <li>- SYSTEM: Returnere side med brugerens info</li> </ul>
Extensions	<ul style="list-style-type: none"> <li>- Brugeren indtaster et brugernavn der ikke er tilgængeligt</li> </ul>
Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Periodisk hvis nyt personale bliver ansat
Miscellaneous	Such as open issues.

## Edit user

Use Case Section	Comment
Use Case Name	Edit user
Scope	Bilabonnements system
Level	Subfunction
Primary Actor	Systemadministrator

Stakeholders and Interests	Systemadministratoren har interesse i at redigere en eksisterende bruger. De ansatte har interesse i at kunne få redigeret deres bruger, heri brugernavn, kodeord eller funktion.
Preconditions	Systemadministratoren er i forvejen logget ind med et gyldigt login der giver adgang til systemadministrator funktioner.
Success Guarantee	En eksisterende brugers brugernavn, kodeord eller funktionsniveau er blevet redigeret og gemt.
Main Success Scenario	<ul style="list-style-type: none"> <li>- Brugeren tilgår bilabonement systemet</li> <li>- Brugeren logger ind med et gyldigt systemadministratorlogin</li> <li>- Brugeren klikker på se brugere</li> <li>- Brugeren vælger funktionen "Rediger", ud fra den bruger som der skal redigeres.</li> <li>- Brugeren redigerer den eksisterende brugers brugernavn, kodeord eller funktionsniveau</li> <li>- Brugeren gemmer redigeringen</li> <li>- Brugeren logger eventuelt ud af systemet</li> </ul>
Extensions	<ul style="list-style-type: none"> <li>- Brugeren indtaster et brugernavn der ikke er tilgængeligt</li> </ul>
Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Periodisk hvis ansatte for eksempel bliver forfremmet, eller skifter afdeling
Miscellaneous	Such as open issues.

Use Case Section	Comment
Use Case Name	Edit user
Scope	Bilabonnements system
Level	Subfunction
Primary Actor	Systemadministrator
Stakeholders and Interests	Systemadministratoren har interesse i at redigere en eksisterende bruger. De ansatte har interesse i at kunne få redigeret deres bruger, heri brugernavn, kodeord eller funktion.
Preconditions	Systemadministratoren er i forvejen logget ind med et gyldigt login der giver adgang til systemadministrator funktioner.
Success Guarantee	En eksisterende brugers brugernavn, kodeord eller funktionsniveau er blevet redigeret og gemt.
Main Success Scenario	<ul style="list-style-type: none"> <li>- Brugeren tilgår bilabonement systemet</li> <li>- Brugeren logger ind med et gyldigt systemadministratorlogin</li> <li>- Brugeren vælger funktionen "rediger bruger"</li> <li>- Brugeren redigerer den eksisterende brugers brugernavn, kodeord eller funktionsniveau</li> <li>- Brugeren gemmer redigeringen</li> <li>- Brugeren logger eventuelt ud af systemet</li> </ul>
Extensions	<ul style="list-style-type: none"> <li>- Brugeren indtaster et brugernavn der ikke er tilgængeligt</li> </ul>

Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Periodisk hvis ansatte for eksempel bliver forfremmet, eller skifter afdeling
Miscellaneous	Such as open issues.

## Delete user

Use Case Section	Comment
Use Case Name	Delete user
Scope	Bilabonnements system
Level	Subfunction
Primary Actor	Systemadministrator
Stakeholders and Interests	Systemadministratoren har interesse i at slette en eksisterende bruger. De ansatte/virksomheden har interesse i at kunne få slettet deres bruger, heri brugernavn, kodeord eller funktion.
Preconditions	Systemadministratoren er i forvejen logget ind med et gyldigt login der giver adgang til systemadministrator funktioner.
Success Guarantee	En eksisterende brugers brugernavn, kodeord eller funktionsniveau er blevet slettet
Main Success Scenario	<ul style="list-style-type: none"> <li>- Brugeren tilgår bilabonnement systemet</li> <li>- Brugeren logger ind med et gyldigt systemadministratorlogin</li> <li>- Brugeren vælger funktionen "slet bruger"</li> <li>- Brugeren sletter den eksisterende bruger</li> <li>- Brugeren bliver slettet</li> <li>- Brugeren logger eventuelt ud af systemet</li> </ul>
Extensions	Hvis en bruger der ikke har rolle som systemadministrator får adgang til systemadministrator-siden, og forsøger at slette en bruger vil brugeren blive logget ud og redirected til startsiden
Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Periodisk hvis ansatte for eksempel bliver fyret eller siger op
Miscellaneous	Such as open issues.

## View users

Use Case Section	Comment
Use Case Name	View users
Scope	Bilabonnements system
Level	Subfunction
Primary Actor	Systemadministrator
Stakeholders and Interests	Systemadministratoren har interesse i at se alle oprettede brugere i systemet, herunder brugernavne og funktionsniveau

Preconditions	Systemadministratoren er i forvejen logget ind med et gyldigt login der giver adgang til systemadministrator funktioner.
Success Guarantee	Brugeren bliver fremvist en liste over alle brugere heri deres brugernavn og funktionsniveau
Main Success Scenario	<ul style="list-style-type: none"> <li>- Brugeren tilgår bilabonnement systemet</li> <li>- Brugeren logger ind med et gyldigt systemadministratorlogin</li> <li>- Brugeren vælger funktionen "se alle brugere"</li> <li>- Brugeren logger eventuelt ud af systemet</li> </ul>
Extensions	
Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Periodisk
Miscellaneous	Such as open issues.

## View dashboards

Use Case Section	Comment
Use Case Name	View dashboards
Scope	Bilabonnements system
Level	Subfunction
Primary Actor	Alle brugere
Stakeholders and Interests	De forskellige typer af brugere skal kunne tilgå hver deres side med funktioner der tilhører deres afdeling.
Preconditions	Alle de forskellige brugere skal have en bruger, som er koblet til den afdeling de arbejder i, samt de skal være logget ind i systemet
Success Guarantee	De forskellige bruger får det rigtige dashboard vist dvs. den side der har med deres afdeling at gøre.
Main Success Scenario	<ul style="list-style-type: none"> <li>- Brugerne tilgår bilabonnement systemet</li> <li>- Brugerne logger ind med et gyldigt systemlogin</li> <li>- Brugerne bliver dirigeret videre til den side med deres afdeling</li> </ul>
Extensions	
Special Requirements	Related non-functional requirements.
Technology and Data Variations List	Varying I/O methods and data formats.
Frequency of Occurrence	Sker hver gang en bruger har logget ind
Miscellaneous	Such as open issues.

See Larman for more information: Chapter 6.9. What do the Sections Mean?

See Larman for more information:  
Chapter 6. Use Cases

\*/

## System Sequence Diagram

/\* TO BE REMOVED...

Objective

- Identify system events.

- Create system sequence diagrams for use case scenarios.

**See Larman for more information:**

Chapter 10. System Sequence Diagrams

\*/

## Operation Contracts

/\* TO BE REMOVED...

Definition: What are the Sections of a Contract?

A description of each section in a contract is shown in the following schema.

Operation	Name of operation, and parameters
Cross References	Use cases this operation can occur within
Preconditions	Noteworthy assumptions about the state of the system or objects in the Domain Model before execution of the operation. These are non-trivial assumptions the reader should be told.
Postconditions	This is the most important section. The state of objects in the Domain Model after completion of the operation.

**See Larman for examples:** 11.9. Example: NextGen POS Contracts

**See Larman for more information:**

Chapter 11. Operation Contracts

\*/

## Package Diagram

/\* TO BE REMOVED...

\*/

## Design Class Diagram

/\* TO BE REMOVED...

The UML includes class diagrams to illustrate classes, interfaces, and their associations. They are used for static object modeling.

**See Larman for more information:**

Chapter 16. UML Class Diagrams

\*/

## Sequence Diagram

/\* TO BE REMOVED...

### Guideline

Spend time doing dynamic object modeling with interaction diagrams, not just static object modelling with class diagrams.

**See Larman for more information:**  
Chapter 15. UML Interaction Diagrams

\*/

## GRASP

/\* TO BE REMOVED...

### **How GRASP was used?**

Explain how following patterns was considered and used in the application

- Creator
- Information Expert
- Low Coupling
- Controller
- High Cohesion
- Polymorphism
- Indirection
- Pure Fabrication
- (Protected Variations)

### **See Larman for more information:**

Chapter 17. GRASP: Designing Objects with Responsibilities

Chapter 18. Object Design Examples with GRASP

Chapter 25. GRASP: More Objects with Responsibilities

\*/

## State Machine diagram

/\* TO BE REMOVED...

Shows the lifecycle of an object: what events it experiences, its transitions, and the states it is in between these events.

As with activity diagrams, UML state diagrams show a dynamic view. The UML includes notation to illustrate the events and states of things—transactions, use cases, people, and so forth.

### **See Larman for more information:**

Chapter 29. UML State Machine Diagrams and Modeling

\*/

## “Optional” – Other Information?

/\* TO BE REMOVED...

Other informative text/diagram to include?

\*/

# Glossary

## Business

Term	Alias	Beskrivelse
Dataregistrering		registrering af nye lejeaftaler og biler
Skade og udbedring		Afdeling som tager sig af registrering af fejl/mangler efter lejeperiode

/\* TO BE REMOVED...

**Use either 1 Glossary with 2 section or 2 Glossaries:**

One for Domain/Business

One for Technical/Data dictionary

**See Larman for more information:**

Chapter 7.8. NextGen Example: A (Partial) Glossary for more information

Chapter 7.9. Commentary: Glossary (Data Dictionary)

\*/

## Unified Process - Iteration Plan

Iteration	Disciplines	Artifact	Reflection (After each iteration)
I1 - Inception Start: d.03/05 Slut: d.05/05	Business Modeling	Domæne model	De logiske processer for at lave den gav god mening for os, det var ikke de store udfordringer.
	Requirements	Use case diagram Vision statement	Use case diagrammet var lidt kringlet, fordi vi var i tvivl om hvilke use cases vi skulle have på. Minimumskrav eller mere specificerende use cases? Vi har valgt at lave udkast til minimumskraven, da det er dem vi vil starte med at implementere når vi skal kode. Tilføjer vi mere end minimumskraven tilføjes disse use cases også til diagrammet.  Vision statement kræver mere arbejde, men der er blevet lavet et udkast til det.
	Analysis and Design	Interessent analyse Feasibility Study Risikoplan/analyse Requirement analyse(FURPS)	Interessentanalysen, feasibility study og risikoplan/analyse blev lavet uden de store problemer. Dog skal de stadig ses igennem og evt. rettes lidt i til sidst i projektet.  Requirement analysen er ikke blevet færdig, men der er blevet lavet lidt til den.
E1 - Elaboration Start: d.06/05 Slut: d.10/05	Business Modeling		
	Requirements	Use case elaboration (descriptions) Glossary problemformulering	Glossary is well underway, but is missing the technical part Problemformulering is done
	Analysis and Design	Udkast til DCD Package diagram EER SSD og SD	DCD is created and has been updated with what we have currently made Package diagram is done EER is well underway SSD and SD has only been made for Create User
	Implementation	Basic user functions - CRUD - Login - Logout - Role specific dashboard DDL and DML for users	DDL and DML has had users and location added to it Role specific dashboard did we start on, but still need a few things. Create user has been done, but missing styling in html
	Testing	Test cases for implemented functions Unit tests	Test case for create is done Basic unit tests for create user and login made  Overall, well overestimated our capabilities to make the amount of use cases listed
	Deployment		
C1 - Construction	Business Modeling		



Start: d.11/05 Slut: d.13/05	Requirements	Use case elaboration per use case (descriptions)	Have been made to every use case.
	Analysis and Design	SSD og SD per use case	Have been made to every use case.
	Implementation	Basic user functions - RUD - Login - Logout - Finish role specific dashboard	All the basic user functions have been made. Login and logout are working and the specific dashboard for the roles have been made except for business developer. We are waiting with business developer because we don't have the functionality for it, and we want that before making it.
	Testing	Test cases for implemented functions Unit tests	We have been unit testing all the methods we could.  Test cases have been made for all use cases.
	Deployment		
C2 - Construction Start: d.16/05 Slut: d.18/05	Business Modeling		
	Requirements	Use case elaboration per use case (descriptions)	
	Analysis and Design	SSD og SD per use case	
	Implementation	CRUD for Cars DDL and DML for cars	
	Testing	Test cases for implemented functions Unit tests	
	Deployment		
C3 - Construction Start: d.xx/x Slut: d.xx/x	Business Modeling		
	Requirements		
	Analysis and Design		
	Implementation		
	Testing		
	Deployment		
xx - x Start: d.xx/x Slut: d.xx/x	Business Modeling		
	Requirements		
	Analysis and Design		
	Implementation		
	Testing		
	Deployment		

## REVIEW

/\* TO BE REMOVED...

1. Producer-group selects artifacts to be reviewed, and fill in first 3 columns: **Review ID**, **Artifact name** and **Focus on**
2. Producer-group hands-over review document and belonging artifacts to a review-group
3. Review-group examine all artifacts, and fill in **Review note**
4. Review-group returns review-document to the producer-group
5. Producer-group decides what to happen with each reviewed artifact, and fill in **Correction/conclusion after review**

\*/

Review ID	Artifact name (step 1):	Focus on (step 1):	Review note (step 2):	Correction/conclusion after review (step 3):

## Test Cases

/\* TO BE REMOVED...

Insert Test Cases Table here - Find and check out different templates on google: <https://www.google.com/search?q=test+case+templates>

\*/

Module name:	Create user
Created by:	Mathias
Date of creation:	09-05-2022
Date of review:	13-05-2022
Reviewer:	Mathias

TEST CASE ID	TEST SENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULTS	POST CONDITION	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_CREATE_USER_001	Verify the creation of a user	Enter valid Username and valid Password	Needs to be logged in as an admin	Enter Username	<Valid Data>	Successful Creation of user	Open the users info page	Opened the user info page	PASS
				Enter Password	<Valid Data>				
				Select Role	<Valid Data>				
				Select location	<Valid Data>				
				Click "Opret Bruger"					
TC_CREATE_USER_002	Verify the creation of a user	Enter invalid Username and valid Password	Needs to be logged in as an admin	Enter Username	<No Input>	Prompt user to input a username		Set the username input field in focus	PASS
				Enter Password	<Valid Data>				
				Select Role	<Valid Data>				
				Select location	<Valid Data>				
				Click "Opret Bruger"					
TC_CREATE_USER_003		Enter valid Username		Enter Username	<Valid Data>	Prompt user to		Set the password input field in focus	PASS
				Enter Password	<No Input>				

	Verify the creation of a user	and invalid Password	Needs to be logged in as an admin	Select Role	<Valid Data>	input a password			
				Select location	<Valid Data>				
				Click "Opret Bruger"	<Valid Data>				

Module name:	View users
Created by:	Mathias
Date of creation:	09-05-2022
Date of review:	13-05-2022
Reviewer:	Sofia

TEST CASE ID	TEST SENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULTS	POST CONDITION	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_VIEW_USER_001	Verify the view of all the users	View all users when there are user in the database	Needs to be logged in as an admin	Enter Username	<Users in database>	Successful View of all users in database		The admin can see all the users in the system. He can edit and delete them with bottoms.	PASS
TC_VIEW_USER_002	Verify the view of all the users	View all users when there is no user in the database	Needs to be logged in as an admin	Enter Password	<No Users in database >	Opens the page, but has "No users found" written on the page		the page said "no users found", but the html and css has to make it look more professional.	PASS

Module name:	Edit user
Created by:	Mathias
Date of creation:	09-05-2022
Date of review:	13-05-2022
Reviewer:	Arboe

TEST CASE ID	TEST SENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULTS	POST CONDITION	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_EDIT_USER_001	Verify the edit function of a user	Edit a user by only changing the username	Needs to be logged in as an admin	Change Username	<Valid data>	Successful change of username	Reload the users page	Name was updated	Pass
				Click "Rediger bruger"					
TC_EDIT_USER_002	Verify the edit function of a user	Edit a user by only changing the role	Needs to be logged in as an admin	Select different role	<Selection of different UserRole>	Successful change of role	Reload the users page	Role was updated	Pass
				Click "Rediger bruger"					
TC_EDIT_USER_003	Verify the edit function of a user	Edit a user by only changing the location	Needs to be logged in as an admin	Select different location	<Selection of different Location>	Successful change of role	Reload the users page	Location was updated	Pass
				Click "Rediger bruger"					
TC_EDIT_USER_004	Verify the edit function of a user	Edit a user by inputting a new password	Needs to be logged in as an admin	Input password	<Valid password>	Successful change of password and login	Logged in	Logged in	Pass
				Click "Rediger bruger"					
				Click "log out"					
				Log in as edited user with new password	<Valid username and new password>				
TC_EDIT_USER_005	Verify the edit function of a user	Edit a user by only changing the username	Needs to be logged in as an admin	Change Username	<Delete what is written>	Prompt to input username		User was prompted to write a username	Pass
				Click "Rediger bruger"					

Module name:	Delete user
Created by:	Lasse
Date of creation:	12-05-2022
Date of review:	13-05-2022
Reviewer:	Lasse

TEST CASE ID	TEST SENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULTS	POST CONDITION	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_DELETE_USER_001	Deleting a user from the system	System administrator deletes a user through the system	Needs to be logged in as system administrator	Access system	<Valid userID>	The desired user is deleted from the page and the system		The user has been deleted from the database and is no longer visible in the system	Pass
				Log in with a valid system administrator user					
				Press "se alle brugere"					
				Select user and press "rediger"					
				Press delete user					
TC_DELETE_USER_002	Deleting a user from the system	User without system administrator role deletes a user	Needs to be logged in as a user without system administrator role	Access system	<Valid userID>	The desired user is not deleted, and the user of the system is immediately logged out and redirected to landingpage		The user is immediately logget out.	Pass
				Log in with a valid user without system administrator role					
				Press delete user					

Module name:	Logout
Created by:	Sofia og Lasse
Date of creation:	12-05-2022
Date of review:	13-05-2022
Reviewer: [OBJ] [OBJ]	[OBJ] [OBJ] Sofia

TEST CASE ID	TEST SENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULTS	POST CONDITION	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_LOG_OUT_001	Verify the successful log out of a user	Log out a valid user from the system	Needs to be logged in as a valid user	Access system		Successfully log out the user	Redirects to the landingpage	User is logged out and redirected to the landingpage.	PASS
				Login with a valid username and password	<Valid Data>				
				Press the designated log out button					
				User is now logged out of the system					

				User will be redirected to the landingpage					
--	--	--	--	---	--	--	--	--	--



Module name:	View dashboards
Created by:	Sofia og Lasse
Date of creation:	12-05-2022
Date of review:	
Reviewer:	

TEST CASE ID	TEST SENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULTS	POST CONDITION	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_VIEW_DASHBOARD_001	Direct the user to the correct dashboard belonging to their department.	Access admin dashboard	Needs to be logged in as an admin	Enter valid admin Username and password		Successful View of all the correct dashboard		Being directed to the admin page	Pass
				Observe if the correct URL and dashboard has been reached					
TC_VIEW_DASHBOARD_002	Direct the user to the correct dashboard belonging to their department.	Access business developer dashboard	Needs to be logged in as a business developer	Enter valid username and password		Successful View of all the correct dashboard		Being directed to the business developer page	Pass
				Observe if the correct URL and dashboard has been reached					
TC_VIEW_DASHBOARD_003			Needs to be logged in as	Enter valid admin		Successful View of all			Pass

	Direct the user to the correct dashboard belonging to their department.	Access data registration dashboard	data registration [OBJ] [OBJ]	username and password Observe if the correct URL and dashboard has been reached		the correct dashboard		Being directed to the data registration page	
TC_VIEW_DASHBOARD_004	Direct the user to the correct dashboard belonging to their department.	Access damage and rectification dashboard	Needs to be logged in as a damage and rectification user [OBJ]	Enter valid admin username and password Observe if the correct URL and dashboard has been reached		Successful View of all the correct dashboard		Being directed to the damage and rectification page	Pass

## Litteraturliste

- K.W. Bruun & Co., *K.W. Bruun Koncernen*, n.d., <https://www.kwbruun.com/da/kwbruun-koncernen>. [Lokaliseret d. 09.05.2022]
- Bilabonnement, *Samarbejdspartnere*, n.d., <https://bilabonnement.dk/samarbejdspartnere>. [Lokaliseret d. 09.05.2022]
- Find-Virksomhed.dk, *Bilabonnement A/S*, n.d., <https://find-virksomhed.dk/firma/bilabonnement-a-s-37856827>. [Lokaliseret d. 09.05.2022]