**Movie World System**

**Taha Mohamed Alzain (269055)**

**Muhammad Nadeem (266704)**

**Fadi Atia Dasus (266265)**

**Oskars Arajs (266534)**

**Balkis Ibrahim (260092)**

**Supervisors**

**Ole Ilsgaard Hougaard**

**Christian Flinker Sandbeck**

**ICT Engineering, VIA University College, Horsens**

**3rd Semester**

*Abstract*

The purpose of this project

**Table of content**

[1.Introduction 2](#_1fob9te)

[2 User Stories and Requirements 3](#_3znysh7)

[2.1 User Stories 3](#_2et92p0)

[2.2 Functional Requirements 3](#_tyjcwt)

[2.2 Non-Functional Requirements 3](#_3dy6vkm)

[3 Analysis 3](#_1t3h5sf)

[3.1 Scenarios 3](#_17dp8vu)

[3.2 Use case diagram 4](#_26in1rg)

[3.3 Use case description 4](#_lnxbz9)

[3.4 Activity diagram 4](#_35nkun2)

[3.5 Conceptual diagram 5](#_1ksv4uv)

[3.6 Database 5](#_44sinio)

[4 Design 5](#_2jxsxqh)

[4.1 Sequence Diagram 5](#_z337ya)

[4.2 Class Diagram 6](#_1y810tw)

[4.3 GUI 6](#_2xcytpi)

[4.4 Client/ Server (RMI) 7](#_3whwml4)

[5 Implementation 7](#_2bn6wsx)

[6 Testing 7](#_qsh70q)

[7 Conclusion 8](#_3as4poj)

[8 References 8](#_1pxezwc)

**List of Figures**

# 1.Introduction

# 2 User Stories and Requirements

In this chapter the purpose will be to draw the user stories, that can help to create a simplified requirements. The requirements are one of the major tasks in designing systems development (Alsaleh.S and Haroon.H., 2016). It is very critical to determine the right requirements for the system otherwise the system won’t have all the functionality needed. So first we start writing the user stories then requirements. And requirements can be made and separated into two groups: functional and non-functional requirements.

## 2.1 User Stories

As the name suggests. it describe how the user require the software that is particularly helpful for customer. User should be the one who need the functioning software and customers can visit the software.

1. As an User I want to be able to add movie to the system, so that the Customer can see it.
2. As an User I want to be able to remove movie from the system, so that I can delete old movies .
3. As a Customer I want to be able to select a movie, so that I can check all the details about it .
4. As a Customer I want to be able to book a ticket for a certain movie, so that I can reserve a place for that movie.
5. As a Customer I want to be able to stream movie, so that I can watch movie at home.
6. As a Customer I want to be able to pay using electronic  payment method for my reservation or streaming, so that I can have a receipt.

## 2.2 Functional Requirements

Based on the information above the following requirements have been created.

1. The system must allow the User to be able to add movie.
2. The system must allow the User to be able to remove movie.
3. The system must allow the Customer to be able to see a list of movies.
4. The system must allow the Customer to be able to get movie information.
5. The system must allow the Customer to be able to book a cinema  ticket for the selected movie.
6. The system must allow the Customer to be able to purchase the ticket electronically for cinema.
7. The system must allow the Customer to be able to purchase the ticket electronically for streaming.
8. The system must allow the Customer to be able to get a receipt for his payment.
9. The system must allow the Customer to be able to stream movie.

## 2.2 Non-Functional Requirements

1. The system must be implemented as a heterogeneous system (C# and Java).
2. The system must be 3-tier architecture .
3. The must use web services
4. The system must use a protocols for sockets and RMI
5. The system must include GUI for each client.

# 

# 

# 

# 3 Analysis

Analysis should be consist of scenarios, use case diagram and use case description. The **use case diagram** is the depiction of what a system can do for a user. Use case diagram is based on the scenarios so these two are connected to each other. A **scenario** is the representation of what is going to happen when someone uses the system. Furthermore, **use case descriptions** have been made for each use case of the actors which participate in this flow.

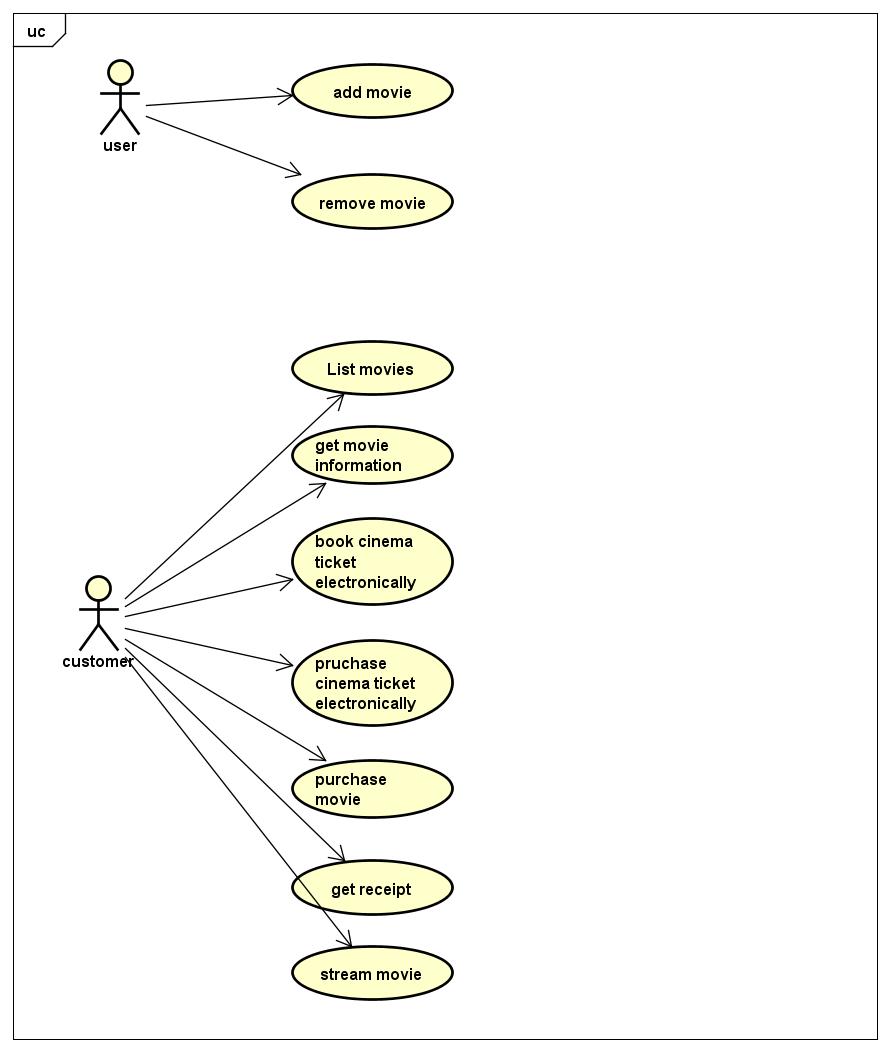
## 3.1 Scenarios

Here only one scenario is presented and that is “add an appointment” scenario of Clinical Management system

## 3.2 Use case diagram

The use case diagram is a graphical representation of the users that are interacting with the system while performing certain tasks. Based on the scenarios it’s possible to create as many use cases as the system needs and these cases are presented in

***Figure 3.1 Use case diagram***



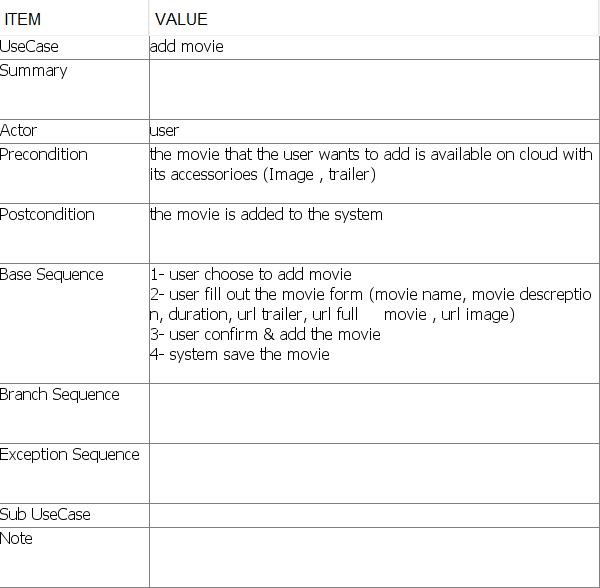
***Figure 3.1 Use case diagram***

## 3.3 Use case description

Based on the use case diagram shown above, several use case descriptions were made for this subchapter. Only one-use case description will be shown as an example while the rest can be seen in ***Appendix 2.***

The use case description is a collection of values, preconditions, postconditions and base sequences which form a detailed view of the actor’s actions while interacting with the system.

***Figure 3 2 Add a Movie use case description***



***Figure 3 2 Add a Movie use case description***

## 3.4 Activity diagram

The activity diagram describes the flow between the action of adding an appointment and what consequences some interactions will produced if executed by the actor.

***Figure 3.3 activity diagram***

*For more examples of activity diagrams, see* ***Appendix 4.***

## 3.5 Conceptual diagram

Conceptual Diagram demonstrates how all of the classes in their packages interact with each other. This is important since it provides a graphical representation on how the system will be created and how each component will interact with another to make it functional.

This is represented as a conceptual diagram as shown in ***Figure 3.4*** bellow.

***Figure 3.4 Conceptual diagram***

## 3.6 Database

***Figure 3.5 Sql Query***

***Figure 3.6 ER Diagram for Clinical Management System***

# 4 Design

## 4.1 Sequence Diagram

To get a better understanding on how the system behaves while certain commands are being given a sequence diagram is created. He sequences diagram will provide a visual representation on what steps the system will take and which methods will be executed to fulfil a certain task.

The following figure will present an example of a sequence diagram from the report.

Add an appointment diagram has been selected for this example.

***Figure 4.1 sequence diagram***

## 4.2 Class Diagram

## 4.3 GUI

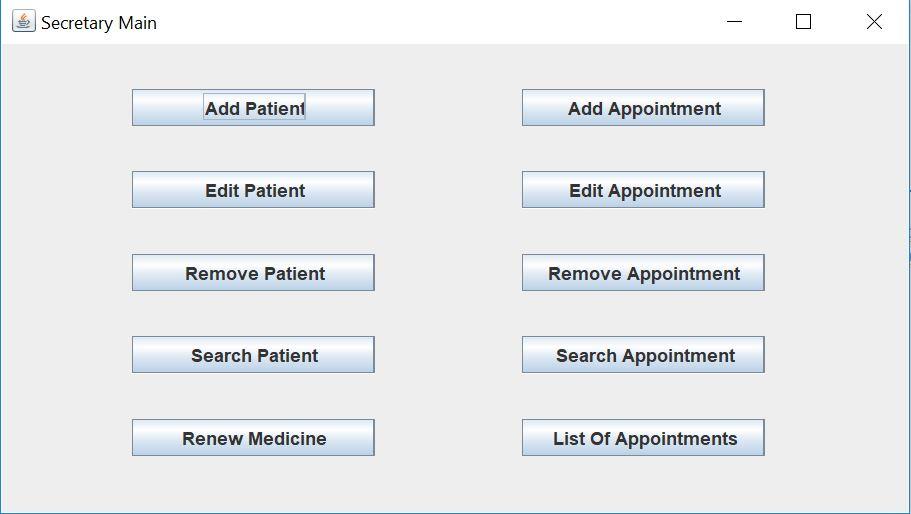
In this subchapter the discussion will be about GUI. This is important since it will be the main tool that the customer will use to perform his daily activities.

The method chosen to create this is swing as shown in the example below.

***Figure 4.6 Swing***

In this document, only one example will be shown regarding the GUI, for the rest see ***Appendix 3 - User Guide.***

***Figure 4.7 Movie System main GUI***

******

.

## 

## 4.4 Client/ Server (RMI)

Program must be implement for Client / Server system. RMI is chosen for client and server system because client/server system can pass data back and forth.

# 5 Implementation

***Figure 5.1***

# 6 Testing

When the program is finished it is time to test it.

The testing of the program consists of all methods being tested to assure that each and every method is working as it should and no exceptions are being thrown. To achieve the desired result Junit testing was used. Unit testing has been chosen since it offers the possibility to test methods one by one.

# 7 Conclusion

# 8 References

Guidelines Development and Summary, IT-SDJ2X-S18 Sesion Material,[Last accessed 05/04/2018] via link: <https://studienet.via.dk/Class/IT-SDJ2X-S18/Session%20Material/SDJ2-S18-19GuidelinesDeploymentAndSummary.pdf>

Ken Schwaber, Jeff Sutherland, 2011; The Scrum Guide, [Last accessed 10/04/2018 ] via link: <https://studienet.via.dk/Class/IT-SWE1X-S18/Session%20Material/Scrum_Guide.pdf>

Project report, 2017 (Appendix 3) VIA Engineering Guidelines [Last accessed 10/04/2018] via link: <https://studienet.via.dk/projects/Engineering__project_methodology/General/Guidelines/2017%20Project%20Report%20(Appendix%203)%20-%20VIA%20Engineering%20Guidelines.pdf>

Project description, 2017 (Appendix 1) VIA Engineering Guidelines [Last accessed 27/02/2018] via link:<https://studienet.via.dk/projects/Engineering__project_methodology/General/Guidelines/2017%20Project%20Description%20(Appendix%201)%20-%20VIA%20Engineering%20Guidelines.pdf>

Process report,2017 (Appendix 2) VIA Engineering Guidelines, [Last accessed 05/04/2018] via link:<https://studienet.via.dk/projects/Engineering__project_methodology/General/Guidelines/2017%20Process%20Report%20(Appendix%202)%20-%20VIA%20Engineering%20Guidelines.pdf>

Rohit Joshi, 2015; Java Design Patterns, [Last accessed 14/03/2018] via link; <http://enos.itcollege.ee/~jpoial/java/naited/Java-Design-Patterns.pdf>

Thomas Connolly, Carolyn Begg, 2015; Database Systems A Practical Approach to Design,Implementation and Management, [Last accessed 18/05/2018] via link;

<http://people.stfx.ca/x2011/x2011asx/5th%20Year/Database/Database%20Management%20Textbook.pdf>

**Videos:**

Programming Knowledge Learning, 2014,July,Java Eclipse GUI Tutorial 1 Creating First GUI Project in Eclipse, [Video file]. Retrieved from <https://www.youtube.com/watch?v=r8Qiz9Bn1Ag>

9 Appendices

9.1 Appendix 1 Project Description

9.2 Appendix 2 Use case description

9.3 Appendix 3 USER GUIDE

9.4 Appendix 4 Diagrams

9.5 Appendix 5 Scenarios

9.6 Appendix 6 Testing