**Clinical Management System**

**Taha Mohamed Alzain(2669055)**

**Muhammad Nadeem 266704**

**Fadi Atia Dasus(266265)**

**Oskars Arajs(266534)**

**Alexandru Vieru(267013)**

**Supervisors**

Ole Ilsgaard Hougaard

Martin Zmija

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

**2nd Semester**

**Table of content**

[1 Introduction 3](#_Toc515440812)

[2 User Stories and Requirements 4](#_Toc515440813)

[2.1 User Stories 4](#_Toc515440814)

[2.2 Functional Requirements 5](#_Toc515440815)

[2.3 Non-Functional Requirements 6](#_Toc515440816)

[2.4 Scenarios 6](#_Toc515440817)

[3 Analysis 8](#_Toc515440818)

[3.1 Use case diagram 8](#_Toc515440819)

[3.2 Use case description 9](#_Toc515440820)

[3.3 Activity diagram 10](#_Toc515440821)

[4 Design 12](#_Toc515440822)

[5 Implementation 13](#_Toc515440823)

[6 Testing 13](#_Toc515440824)

[7 Conclusion 14](#_Toc515440825)

Appendices

# Introduction

Denmark’s healthcare system has been regarded as excellent in terms of world standards. It is easy to look how complex system they are using for management. Denmark’s health care IT system is rated as one of the most efficient in the world. Other countries which are seeking to rapidly increase health adoption can learn from the Danish experience in healthcare (Kirkegard.P., 2013).

Currently, our countries are still storing information on paper or cards. Nurses are still writing patient information manually and index it to organize in racks. These are creating troublesome, expensive, inefficient and also not secured.

In Recent Decades, use of Information Technology is more in practice, so doctors consider these changes in health care center.

Proposed system (Clinical Management System) is a computerized system that save the patient records. It will reduce the burden of Doctor and Nurses. The system integrates staff and patient records, appointments, prescription, medical cases and medicine.

Project task is to make a System that should be implemented in Client/ Server programming. For this instance, RMI (Remote Method Invocation) is used. All the information should be saved on database through the server, and these information’s could read and write by client (in this project doctor and nurses).

Project is started with the user stories and requirements i.e. what are important functionalities that a system would have. Once the requirement is completed, analysis part will start, where use cases, use case descriptions, and conceptual diagram will discuss. This lead to design part where Class Diagram, Sequence Diagram and …. In the implementation phase, discussion should be on our best part of coding and on Client / Server. In final, system testing must be performed, and J-Unit test is used to analyze all the functionality.

# User Stories and Requirements

In this chapter the purpose will be to establish the requirements based on the user stories from the customer. This is done so that a better understanding of the customer’s needs can be made. All the requirements in this chapter are based on the business case from the clinic.

To categorize the requirements into grades of importance, the SMART model will be used since it provides all the necessary tools to accomplish this.

Requirements can be made and separated into two groups: functional and non-functional requirements.

## User Stories

In this sub chapter the user stories from the customer will be presented since the requirements for this report will be made based on them.

1. As a manager I want to be able to add an employee to the system based on their specialty so that they can handle their tasks.
2. As a manager I want to search for employees so that I can keep track of them.
3. As a manager I want to be able to remove employees.
4. As a manager I want to be able to edit existing employees.
5. As a secretary I want to store patient personal information so that I can book a medical appointment for them.
6. As a secretary I want to be able to search for patient information so that I  can obtain relevant information about a certain patient.
7. As a secretary I want to be able to manage an appointment so that I can organize patient meetings.
8. As a secretary I want to be able to view patient’s medical records so that their condition can be monitored.
9. As a secretary I want to view patients medicine so that I can send request for renew.
10. As a doctor I want to add medicine to the system so that I can add medicine to the prescription of the patient.
11. As a doctor I want to edit Medicine if name or quantity is changed so that patient will get right medicine in right dose.
12. As a doctor I want to remove medicine from the system so that medicine should be deleted from the system.
13. As a doctor I want to add  a patient’s medical condition so that I can form a medical case.
14. As a doctor I want to prescribe medicine that can be stored in the patient’s personal record so that I can handle the data
15. As a doctor I want to renew medicine prescription that can be stored in the patient’s personal record so that I can satisfy his/her requests.
16. As a doctor I want to approve medicine prescription request that comes from the secretary so that patients can get what they requested.

## Functional Requirements

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

1. The system must allow the manager to be able to add new employees based on category.
2. The system must allow the manager to be able to remove employees.
3. The system must allow the manager to be able to edit existing employees.
4. The system must allow the manager to be able to search for certain employees.
5. The system must allow the secretary to be able to add new patient’s information.
6. The system must allow the secretary to be able to remove patient’s information.
7. The system must allow the secretary to be able to edit existing patient’s information.
8. The system must allow the secretary to be able to search for patient’s information.
9. The system must allow the secretary to be able to add an appointment.
10. The system must allow the secretary to be able to remove an appointment.
11. The system must allow the secretary to be able to edit an appointment.
12. The system must allow the secretary to be able to get a list of appointments.
13. The system must allow the secretary to be able to send a request in order to get the medicine renewed by the doctor.
14. The system must allow the doctor to be able to add a patient’s medical case.
15. The system must allow the doctor to be able to add a medical prescription to the patient’s medical record.
16. The system must allow the doctor to be able to renew medicine prescription for the patient.
17. The system must allow the doctor to be able to approve the requested medicine that comes from the secretary.
18. The system must allow the doctor to be able to add new medicine information.
19. The system must allow the doctor to be able to remove medicine.
20. The system must allow the doctor to be able to edit existing medicine.
21. The system must allow the doctor to be able to search for medicine.

## Non-Functional Requirements

1. The system must be developed in java.
2. The usability of the system must be tested by end users.
3. The system must store information in a database.
4. The system must handle multiple requests at the same time.
5. The system must be up 24/7.

## Scenarios

In this sub chapter the scenarios for the current software will be presented. Only one scenario will be shown as an example while the rest can be found in appendix….

1. **Add employee scenario**
2. Manager selects employee type.
3. Manager provides name and relevant information for an employee.
4. System validates employee data by confirming the saving of the entered information.

**Classes;**

* **System**
* **Manager**
* **Employee**
* **Type**

Methods:

* Select
* Validate
* Save
* Provide
* Confirming

# Analysis

In this chapter, the focus will be on analyzing the user requirements stated above and modeling the software based on them using several methods and techniques.

To get a better understanding on how the user will interact with the new system, a use case diagram has been created.

## 3.1 Use case diagram

***Figure 1 Use case diagram***

A close up of a map

Description generated with high confidence

For this case, there are 3 actors involved in the process and each of them have their own individual use cases as shown in ***Figure 1***. These use cases are the actions that each actor needs to perform as part of their daily activities.

Furthermore, use case descriptions have been made for each use case of the actors which participate in this flow. In this chapter only one, use case description will be shown as an example while the rest can be seen in Appendix …………...

## 3.2 Use case description

***Figure 2 Add patient use case description***

A screenshot of a cell phone

Description generated with very high confidence

In this example, the actor performs an action using the system of adding a patient. This figure shows what steps the secretary takes for the system to successfully complete the requested action and what steps does the system make.

For more use case description examples, see Appendix……….

In the following figure, an example of an activity diagram will be made. The diagram is derived from one of the use case descriptions and graphically displays the steps taken by the actor to perform a certain action.

## 3.3 Activity diagram

***Figure 3 Manager add employee Activity diagram***

A picture containing screenshot

Description generated with high confidence

The manager tries to add a new employee using the new system and is required to take certain steps before his request is completed as shown in ***Figure 3.***

For more examples of activity diagrams, see Appendix………

The next step in this chapter is to present a conceptual diagram for the new system. 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 sown in ***Figure 4*** bellow.

***Figure 4 Conceptual diagram***

# Design

# Implementation

# Testing

# Conclusion

**References**

# 

**Appendices**