**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](#_Toc515269466)

[2 User Stories and Requirements 4](#_Toc515269467)

[3 Analysis 5](#_Toc515269468)

[4 Design 6](#_Toc515269469)

[5 Implementation 7](#_Toc515269470)

[6 Testing 7](#_Toc515269471)

[7 Conclusion 8](#_Toc515269472)

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

## User Stories

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 patients 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

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

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

# Design

# Implementation

# Testing

# Conclusion

**References**

# 

**Appendices**