2/14/2025

(Student C00270632) Qadeer Hussain Supervisor: Paul Barry

Software DEVELOPMENT PROJECT YEAR 4

Design Document

Elderly Care Management System(Care Net)

A logo with black text

Description automatically generated

**Table of Contents**

[**Introduction** 3](#_Toc190448913)

[**System Architecture** 4](#_Toc190448914)

[**Technologies & Tools** 5](#_Toc190448915)

[**System Sequence Diagrams** 6](#_Toc190448916)

[**Key Server** 6](#_Toc190448917)

[**Patient Profile** 6](#_Toc190448918)

[**Create Patient Profile** 6](#_Toc190448919)

[**Search Patient Profile** 7](#_Toc190448920)

[**Edit Patient Profile** 8](#_Toc190448921)

[**Delete Patient Profile** 9](#_Toc190448922)

[**Medical Dashboard** 10](#_Toc190448923)

[**Search Patient Medical information on Dashboard** 10](#_Toc190448924)

[**Add Patient Medical information on Dashboard** 11](#_Toc190448925)

[**Edit Patient Medical information on Dashboard** 12](#_Toc190448926)

[**Delete Patient Medical information on Dashboard** 13](#_Toc190448927)

[**Patient Medical Log Dashboard** 14](#_Toc190448928)

[**Prototype GUI Screens** 15](#_Toc190448929)

[**Main Menu** 15](#_Toc190448930)

[**Patient Profile Prototype** 15](#_Toc190448931)

[**Database Design** 16](#_Toc190448932)

[**Patient Profile Table** 16](#_Toc190448933)

[**Detailed Use Cases** 17](#_Toc190448934)

[**Patient Profile Use Case** 17](#_Toc190448935)

[**References** 19](#_Toc190448936)

**Figure Table**

[Figure 1: System Architecture 4](#_Toc190448937)

[Figure 2: Key Server Token and Key Generation System Sequence Diagram 6](#_Toc190448938)

[Figure 3: Create Patient Profile System Sequence Diagram 6](#_Toc190448939)

[Figure 4: Search Patient Profile System Sequence Diagram 7](#_Toc190448940)

[Figure 5: Edit Patient Profile Sequence Diagram 8](#_Toc190448941)

[Figure 6: Delete Patient Profile System Sequence Diagram 9](#_Toc190448942)

[Figure 7: Search Patient Medical info on Dashboard System Sequence Diagram 10](#_Toc190448943)

[Figure 8: Add Patient Medical info on Dashboard System Sequence Diagram 11](#_Toc190448944)

[Figure 9: Edit Patient Medical info on Dashboard System Sequence Diagram 12](#_Toc190448945)

[Figure 10:Delete Patient Medical info on Dashboard System Sequence Diagram 13](#_Toc190448946)

[Figure 11: Patient Medical Log Dashboard 14](#_Toc190448947)

[Figure 12: Main Menu 15](#_Toc190448948)

[Figure 13: Patient Profile 15](#_Toc190448949)

[Figure 14: Patient Profile Table 16](#_Toc190448950)

[Figure 15: Example Patient Data 16](#_Toc190448951)

[Figure 16: Patient Profile ER diagram 16](#_Toc190448952)

# **Introduction**

The purpose of this design document is to outline key features of the Elderly Care Management System(ECMS). The document provides information relating to the design of the application.

* The System Architecture Diagram that outlines the overall structure of the application
* System Sequence Diagrams will be included in the document to illustrate the functionality of the features.
* GUI Prototype Screens.
* The Database Design to illustrate how the data is organised and managed .
* Detailed Use Case will be provided to demonstrate the internal functions of the features.

# **System Architecture**

A diagram of a computer server

AI-generated content may be incorrect.

Figure 1: System Architecture

This system architecture diagram for the Elderly Care Management System illustrates the overall structure of the project. This diagram contains the database that stores all data which is safe guarded by encryption and decryption. The main interface accessible by both Admin and Carer.

# **Technologies & Tools**

This project uses multiple “technologies and tools”. This section of the document outlines the technologies and tools used: Jupyter Notebook, Cryptography Library, Maria DB, Python and Django.

**Jupyter Notebook**

Jupyter Notebook is utilized for prototyping and data exploration of the early phases of the project. This is particularly useful for testing database queries, experimenting with data encryption methods, and visualizing data.

**Cryptography Library**

The cryptography library is crucial to ensure data security within the application. This library was used to encrypt data and decrypt data providing robust protection for sensitive information handled by the system.

**Secrets Library**

**MariaDB**

MariaDB was selected as the database management system to store and manage data throughout the development of this project. This was chosen as it has been reliable in previous projects.

**Python**

Python will be the core language used in this project, helping with backend tasks, handling data, and connecting different parts of this system. It has many libraries that make it easy to work.

**Django**

Django is a high level python web framework. It will be used throughout the project, helping to enhance security and streamlining the development of the application.

**FastAPI**

# **System Sequence Diagrams**

## **Key Server**

A diagram of a key server

AI-generated content may be incorrect.

Figure 2: Key Server Token and Key Generation System Sequence Diagram

## **Patient Profile**

### **Create Patient Profile**

A diagram of a diagram

AI-generated content may be incorrect.

Figure 3: Create Patient Profile System Sequence Diagram

In the Elderly Care Management System their is a key feature which is the Patient Profile. This keeps a secure record of the patient personal and medical information.

### **Search Patient Profile**

A diagram of a software company

AI-generated content may be incorrect.

Figure 4: Search Patient Profile System Sequence Diagram

### **Edit Patient Profile**

A diagram of a computer process

AI-generated content may be incorrect.

Figure 5: Edit Patient Profile Sequence Diagram

### **Delete Patient Profile**

A diagram of a diagram

AI-generated content may be incorrect.

Figure 6: Delete Patient Profile System Sequence Diagram

## **Medical Dashboard**

### **Search Patient Medical information on Dashboard**

A diagram of a software company

AI-generated content may be incorrect.

Figure 7: Search Patient Medical info on Dashboard System Sequence Diagram

### **Add Patient Medical information on Dashboard**

A diagram of a diagram

AI-generated content may be incorrect.

Figure 8: Add Patient Medical info on Dashboard System Sequence Diagram

### **Edit Patient Medical information on Dashboard**

A diagram of a software company

AI-generated content may be incorrect.

Figure 9: Edit Patient Medical info on Dashboard System Sequence Diagram

### **Delete Patient Medical information on Dashboard**

A diagram of a process flow

AI-generated content may be incorrect.

Figure 10:Delete Patient Medical info on Dashboard System Sequence Diagram

### **Patient Medical Log Dashboard**

A diagram of a computer

AI-generated content may be incorrect.

Figure 11: Patient Medical Log Dashboard

## **Roster**

A diagram of a schedule

AI-generated content may be incorrect.

Figure 12: Roster System Sequence Diagram

# **Prototype GUI Screens**

## **Main Menu**

A blue background with orange labels

Description automatically generated

Figure 13: Main Menu

## **Patient Profile Prototype**

A screenshot of a contact form

Description automatically generated

Figure 14: Patient Profile

# **Database Design**

## **Patient Profile Table**

A screenshot of a computer screen

Description automatically generated

Figure 15: Patient Profile Table

**Example Patient Data**

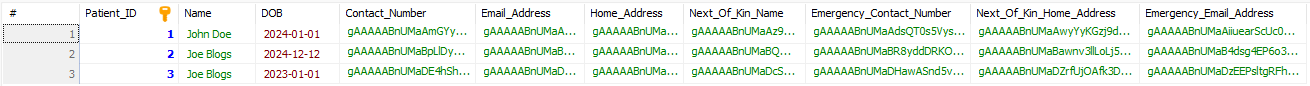
****

Figure 16: Example Patient Data

**Patient Profile ER diagram**

A screen shot of a computer

Description automatically generated

Figure 17: Patient Profile ER diagram

# **Detailed Use Cases**

## **Patient Profile Use Case**

|  |
| --- |
| **Use Case Name:** Create Patient |
| **Actors:** Admin |
| **Pre-Conditions:** The user must be logged into the ECMS system. |
| **Main Success Scenario:**   1. The user logged into the system. 2. The user selects patient profile. 3. The user selects Add new patient. 4. Input the patient information. |
| **Post-Conditions:** The Patient Profile is created |
| **Alternative Scenario:** Incorrect input or invalid data prompt user to correct the data |

|  |
| --- |
| **Use Case Name:** Encrypt |
| **Actors:** Admin, System |
| **Pre-Conditions:** Patient data is valid and is ready for encryption. |
| **Main Success Scenario:**   1. The System gets information from the user. 2. The System encrypts the sensitive data fields. 3. The encrypted patient data is ready to be stored |
| **Post-Conditions:** The data is encrypted. |
| **Alternative Scenario:** Patient data fails to encrypt it notify the user. |

|  |
| --- |
| **Use Case Name:** Store |
| **Actors:** System |
| **Pre-Conditions:** Database connection is established |
| **Main Success Scenario:**   1. The encrypted patient data sent to database. 2. The Data is stored to relevant table and column. 3. The system logs the storage. |
| **Post-Conditions:** The patient data is successfully stored in the database. |
| **Alternative Scenario:** Patient data fails to be stored in the database this notifies thee user. |

|  |
| --- |
| **Use Case Name:** Search/Find |
| **Actors:** Admin, Carer |
| **Pre-Conditions:** The database contains patient data to search. |
| **Main Success Scenario:**   1. The user is logged into the system 2. The user selects patient profile. 3. Search function. 4. System retrieves relevant record. |
| **Post-Conditions:** Successfully find patient profile. |
| **Alternative Scenario:** If no Patient profile is found, the system informs the user that no results are available. |

|  |
| --- |
| **Use Case Name:** Retrieve/Display |
| **Actors:** Admin, Carer, System |
| **Pre-Conditions:** The data exists in the database and is accessible for viewing, |
| **Main Success Scenario:**   1. The user selects patient record. 2. System decrypts the patient data. 3. The system then displays the patients data. |
| **Post-Conditions:** The patient’s data is displayed to the Admin or Carer. |
| **Alternative Scenario:** If patient is unable to display and/or decrypt patient data notify the user. |

# **References**

Django Software Foundation, 2024. *Django.* [Online]   
Available at: https://www.djangoproject.com/  
[Accessed December 2024].

FastAPI Tiangolo, 2025. *FastAPI.* [Online]   
Available at: https://fastapi.tiangolo.com/  
[Accessed 2025].

Jupyter, 2024. *Jupyter.* [Online]   
Available at: https://jupyter.org/  
[Accessed December 2024].

MariaDB Foundation , 2024. *MariaDB Server.* [Online]   
Available at: https://mariadb.org/  
[Accessed December 2024].

PYPI, 2024. *Cryptography.* [Online]   
Available at: https://pypi.org/project/cryptography/  
[Accessed December 2024].

Python Software Foundation, 2024. *Python.* [Online]   
Available at: https://www.python.org/  
[Accessed December 2024].