**CHAPTER 5**

**SYSTEM DESIGN**

**Chapter 5**

# SYSTEM DESIGN

## 5.1 System Architecture

System architecture refers to the overall design and organization of a computer system, which includes hardware components, software components, and the communication and interaction between them. It defines the way in which the system's components are connected, how they operate together to achieve the system's objectives, and how they are managed and maintained over time. A well-designed system architecture is critical for ensuring that a computer system is scalable, secure, reliable, and maintainable

A diagram of a system

Description automatically generated

Fig 5.1 System Architecture

The architecture is strategically divided into three main sections – General User Interface, Elderly Interface, and Family Interface – with a focus on addressing the unique needs of

both elderly individuals and their family members. The General User Interface serves as the central hub for all users, providing robust authentication mechanisms for secure access

to the platform. Upon authentication, users are seamlessly categorized into two distinct groups: Elderly and Family.

The Elderly Interface is meticulously designed to cater to the specific needs and preferences of elderly users, offering a suite of tailored features aimed at enhancing their overall well-being and independence. The Space Reality View immerses them in an intuitive and engaging virtual environment, fostering cognitive stimulation and interactive experiences. The World Health Report empowers elderly users with valuable insights into global health trends and relevant information, enabling them to stay informed about their health status and make informed decisions. The Real-Time Health Monitor System with an Emergency Response System ensures continuous monitoring of vital signs and immediate assistance in case of emergencies, providing peace of mind and security. Additionally, the Appointment Booking feature simplifies the process of scheduling essential services such as doctor visits and cleaning, streamlining daily management tasks for elderly users.

Conversely, the Family Interface is specifically tailored to meet the needs of family members, providing comprehensive tools for monitoring and managing the health and well-being of their elderly loved ones. Data Statements offer detailed reports and analytics on various health metrics, allowing family members to track progress and identify any potential concerns. Notifications keep family members informed about important events and updates related to the well-being of their elderly relatives, facilitating proactive intervention when necessary. Real-Time Monitoring provides a live feed of vital health information and activity updates, enabling family members to stay connected and responsive to the needs of their elderly loved ones.

Overall, this architecture embodies a holistic approach to remote elderly care, leveraging technology to bridge the gap between elderly individuals and their families. By combining intuitive interfaces with advanced features, our application aims to facilitate seamless communication, monitoring, and management of health-related services, ultimately enhancing the quality of life for elderly users and providing peace of mind for their families.

## 5.2 Module Decomposition

**NLP Voice Assistant:**

The NLP (Natural Language Processing) Voice Assistant serves as the primary interface for elderly users to interact with the application. It facilitates hands-free navigation and control, enabling users to perform various tasks and access information using voice commands. The voice assistant integrates advanced NLP algorithms to understand and interpret natural language input from users. It supports a wide range of commands related to scheduling appointments, managing medication, checking health metrics, and accessing entertainment features like the Space Reality View. This module provides a user-friendly and accessible interface for elderly users, especially those with limited mobility or dexterity, enhancing the overall user experience by streamlining navigation and reducing reliance on manual input methods.

**Fraudulent SMS Detection:**

This module is designed to safeguard elderly users from potential scams and fraudulent activities conducted via SMS messages. It analyzes incoming messages in real-time to identify suspicious content and alert users about potential threats. Leveraging machine learning algorithms, the system examines message content, sender information, and other metadata to detect patterns indicative of fraudulent behavior. It can recognize common scam tactics such as phishing attempts, lottery scams, and fake prize notifications. The benefits include protecting elderly users from falling victim to scams and financial fraud, thereby preserving their financial security and peace of mind. It helps instill trust and confidence in using mobile communication channels for staying connected with friends, family, and caregivers.

**Mental Health Chatbot:**

The Mental Health Chatbot offers emotional support and basic mental health assistance to elderly users through conversational AI. It serves as a virtual companion capable of

engaging in empathetic conversations and providing guidance on coping strategies and self-care practices. Powered by sophisticated natural language processing models, the chatbot can engage users in meaningful dialogues, assess their emotional state, and offer appropriate responses and interventions. It may provide relaxation techniques, mindfulness exercises, or referrals to professional mental health services if needed. The benefits include addressing the mental and emotional well-being of elderly users by providing them with a supportive and non-judgmental outlet for expressing their feelings and concerns. It helps alleviate feelings of loneliness, anxiety, and depression commonly experienced by elderly individuals living alone or in isolation.

**Fall Detection**

The Fall Detection module utilizes sensors and algorithms to detect falls and alert caregivers or emergency services promptly. It continuously monitors the user's movements and analyzes sensor data to identify sudden changes indicative of a fall. Upon detecting a fall, the system triggers an alert notification, which can be sent to designated caregivers or emergency contacts. This module provides elderly users with added safety and peace of mind, ensuring prompt assistance in case of accidents or medical emergencies. It enhances their independence and allows them to live with confidence knowing that help is readily available when needed.

**Medication, Prescription, and Reports Management:**

This module helps elderly users manage their medication schedules, prescriptions, and medical reports for easy access and monitoring. It provides a user-friendly interface for organizing and tracking medications, including dosage instructions, refill reminders, and medication histories. Additionally, users can upload and store medical reports and prescriptions securely within the app, ensuring they have quick access to important health information whenever needed. The benefits include promoting medication adherence, reducing the risk of medication errors, and facilitating better communication with healthcare providers. This module empowers elderly users to take control of their medication regimen and maintain their overall health more effectively.

**Tracking Vitals:**

The Tracking Vitals module monitors vital signs like heart rate and blood pressure to track the health status of users in real-time. It utilizes wearable sensors or smart devices to collect and analyze vital sign data continuously. The system provides visualizations and alerts for abnormal readings, allowing users and caregivers to monitor changes in health status and take appropriate action if necessary. This module enhances proactive healthcare management by enabling early detection of potential health issues and facilitating timely interventions. It promotes a proactive approach to health monitoring, empowering elderly users to maintain their well-being and quality of life.

**Family Interface with Notification System:**

The Family Interface with Notification System allows family members to stay informed about the elderly’s condition and any alerts generated by the system. It provides a centralized dashboard where family members can view real-time updates on the health status and activity of their elderly relatives. The notification system sends alerts for events such as falls, missed medication doses, or abnormal vital sign readings, ensuring that family members can respond promptly to any emergencies or concerns. This module fosters communication and collaboration among family members, facilitating a coordinated approach to elderly care and support.

**Routine Analysis:**

The Routine Analysis module analyzes daily activity patterns to offer insights and suggestions for healthy habits. It tracks various aspects of the user's routine, including sleep patterns, physical activity levels, and dietary habits, using data from sensors or wearable devices. The system uses machine learning algorithms to identify trends and patterns in the user's behaviour .

## 5.3 Use Case

A diagram of a health care system

Description automatically generated

Fig 5.3 Use Case Diagram

The use case diagram outlines the interactions between users and the system functionalities of your Android app. It features two types of users: the Elderly User and the Family User, who interact with various system functions. The Elderly User can store medication data, view daily activities, detect falls, and view their daily routine. These actions are part of the Medication Management, Mental Health Chatbot, Routine Tracking, and Emergency Response functionalities, respectively. The Family User has the ability to log in and receive updates, which fall under Authentication and Notifications. Both user types interact with the Unusual Activity update, indicating a shared concern for any irregularities detected by the system. Health Data is a central component that is specifically linked to Medication Management, emphasizing the importance of accurate health information in the medication process. This diagram effectively represents the dynamic between users and the system, ensuring that both the elderly and their family members are actively engaged in the health management process.

## 5.4 Algorithm Design

**Algorithm: Fraudulent SMS Detection**

1. Start: Initialize the system to begin SMS analysis.
2. Detect Fraudulent Activity: Analyze incoming SMS for potential fraudulent activity.
3. Keyword Matching:

If the message contains predefined fraudulent keywords, proceed to step 4.

If not, go to step 5.

1. Message Contains Keyword:

Mark the message as fraudulent.

End the algorithm.

1. Unusual Pattern Detection:

Check for unusual patterns in the message.

If no unusual patterns are found, mark the message as safe and end the algorithm.

If unusual patterns are detected, proceed to step 6.

1. Sender Verification:

Verify if the sender is in the user’s contact list.

If the sender is not in contacts and the message contains unexpected characters, mark the message as fraudulent.

Otherwise, mark the message as safe.

This algorithm utilizes a combination of keyword matching, pattern recognition, and sender verification to determine the legitimacy of an SMS message. It ensures that messages are thoroughly checked for common signs of fraud before being classified.

Fig 5.4 Flow Chart

**CHAPTER 6**

**IMPLEMENTATION**

**Chapter - 6**

# IMPLEMENTATION

## 6.1 Implementation Approaches

### 6.1.1 Firebase for Infrastructure Management

Firebase is a comprehensive platform provided by Google that offers a wide range of services for mobile and web application development. It includes features such as authentication, real-time database, cloud storage, and cloud messaging, among others. In your application, Firebase plays a crucial role in enabling secure authentication mechanisms, efficient data management, and seamless communication between users and the application's backend services.

First and foremost, Firebase's authentication service provides a robust framework for user authentication, allowing users to securely log in or sign up to access the application's features. With Firebase Authentication, you can implement various authentication methods, including email/password authentication, social login (such as Google or Facebook), and phone number authentication, ensuring flexibility and convenience for users.

Additionally, Firebase Realtime Database serves as a centralized repository for storing and synchronizing real-time data across devices. This is particularly valuable for your application, as it allows you to store crucial user data such as vital signs, medication schedules, prescriptions, and medical reports in a structured and easily accessible manner. The real-time synchronization feature ensures that any updates made to the database are immediately reflected across all connected devices, providing users with up-to-date information and ensuring consistency across the application.

Moreover, Firebase Cloud Storage offers scalable and secure cloud storage solutions for storing and serving user-generated content, such as images, documents, and multimedia files. By utilizing Firebase Cloud Storage, you can efficiently store and manage images of medication, prescriptions, and reports, as well as any other multimedia assets associated with user data. This helps optimize storage resources, improve performance, and ensure reliable access to user-generated content.

In summary, Firebase provides a comprehensive suite of services that are instrumental in powering the core functionalities of your application. From secure authentication and real-time data synchronization to scalable cloud storage solutions, Firebase offers the tools and infrastructure needed to build a robust and user-friendly application for remote elderly care. Its seamless integration with your application architecture streamlines development efforts and enhances the overall user experience, making Firebase an indispensable component of your application's backend infrastructure.

A screenshot of a computer

Description automatically generated

Fig 6.1.1 Firebase Dashboard

### OpenStreetMap for Routine Analysis

OpenStreetMap (OSM) is an open-source mapping platform that provides detailed and up-to-date geographic data for locations worldwide. It offers a vast repository of map data, including streets, landmarks, points of interest, and geographical features, contributed and maintained by a global community of volunteers. In your application, OpenStreetMap serves as a valuable tool for geolocation services and spatial analysis, particularly for implementing the routine analysis feature that triggers alerts when a user moves beyond a predefined boundary.

One of the key advantages of OpenStreetMap is its extensive coverage and level of detail, which surpasses many commercial mapping services. The data available on OSM is constantly updated and refined by a diverse community of contributors, ensuring that users have access to accurate and reliable geographic information. This high level of detail and accuracy makes OpenStreetMap well-suited for applications that require precise location data, such as your routine analysis feature.

Moreover, OpenStreetMap offers flexibility and customization options that allow developers to tailor map data to their specific needs. This includes the ability to define custom boundaries, overlay additional layers of data, and perform spatial analysis to extract insights from geographic data. In your application, OpenStreetMap enables you to define a circular boundary around a user's home location and monitor their movements in real-time. By leveraging OSM's APIs and libraries, you can efficiently track the user's location and trigger alerts when they move beyond the predefined boundary, providing peace of mind for family members and caregivers.

Additionally, OpenStreetMap's open-source nature and community-driven model make it a cost-effective and accessible solution for developers. Unlike proprietary mapping platforms that may come with licensing fees or usage restrictions, OpenStreetMap provides free access to its map data and APIs, allowing developers to integrate mapping functionality into their applications without incurring additional costs. This democratization of mapping technology ensures that developers of all sizes and budgets can leverage the power of geolocation services to enhance their applications.

In summary, OpenStreetMap offers a robust and flexible mapping platform that is well-suited for implementing geolocation services and spatial analysis in your application. Its extensive coverage, accuracy, and customization options make it a valuable tool for monitoring user movements, triggering alerts, and enhancing the overall functionality and user experience of your remote elderly care application.

A map of a city

Description automatically generated

Fig 6.2.1 OpenStreetMap Snapshot

**CHAPTER 7**

**TESTING**

**Chapter - 7**

# TESTING

## 7.1 Testing Approach

Testing is a systematic and disciplined process of evaluating a software application or system to identify defects, errors, or discrepancies between expected and actual behavior. It involves executing the software with the intention of finding bugs, verifying that it meets specified requirements, and ensuring its overall quality.

The primary goal of testing is to uncover issues in the software and provide feedback to the development team, allowing them to address and fix any identified problems. Testing helps to ensure that the software functions as intended, is reliable, and meets the needs and expectations of its users.

### 7.1.1 Unit Testing

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules. Unit Testing is done during the development (coding phase) of an application by the developers. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. Unit testing finds problems early in the development cycle. This includes both bugs in the programmer's implementation and flaws or missing parts of the specification for the unit

#### UTC-1: User Performance Prediction Model

The final user performance prediction model uses the linear regression model for prediction. In linear regression, the parameters, also known as coefficients or weights, represent the relationship between the input features and the output,the input features consist of various metrics such as time spent, quiz scores, and quiz attempts, while the output represents the predicted performance class label. Each parameter in the linear regression model corresponds to a specific feature, indicating how much that feature contributes to the prediction. For instance, a higher coefficient for the "time spent" feature suggests that more time spent correlates with better performance. When we provide input data to the model, it multiplies each feature value by its corresponding parameter and sums them up, along with an intercept term if present. This calculation forms the basis for predicting the output performance class label. The output of the linear regression model is a continuous value representing the predicted performance class label. However, since you're performing classification with three classes (0, 1, 2). This is typically done by rounding the predicted value to the nearest integer. During unit testing, we evaluate the model's performance by comparing the predicted output with the expected output for a set of known input data. If the predicted output matches the expected output within an acceptable margin of error, the test case passes; otherwise, it fails. The purpose of unit testing is to ensure that the model behaves as expected for various input scenarios. By systematically validating the model against known data, we can identify any discrepancies or errors in its predictions. Ultimately, the goal is to achieve a reliable and accurate linear regression model that can effectively classify users into different performance categories based on their input metrics.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input Data** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| Test 1 | [10, 20, 3] | 1 | 1 | Pass |
| Test 2 | [5, 15, 2] | 0 | 0 | Pass |
| Test 3 | [25, 30, 4] | 2 | 2 | Pass |
| Test 4 | [12, 18, 3] | 1 | 1 | Pass |
| Test 5 | [8, 22, 2] | 0 | 0 | Pass |

Table 7.1 Unit Testing – 1

In this table:

* “Input Data” represents the input features (e.g., time spent, quiz scores, quiz attempts).
* “Expected Output” represents the expected class label predicted by the model.
* “Pass/Fail” indicates whether the predicted output matches the expected output.

#### UTC-2: User Performance Prediction Model

Homomorphic encryption is a powerful technique used to perform computations on encrypted data without decrypting it first. In our user performance prediction model, homomorphic encryption enhances data security by allowing us to process sensitive user metrics while preserving privacy.

We employ a partially homomorphic encryption scheme, specifically the Paillier cryptosystem, which supports addition and multiplication operations on encrypted data. This enables us to train and utilize the linear regression model on encrypted user input features without revealing the raw data to the model or any third party.

In our implementation, the user input features are encrypted before being sent to the model for prediction. The linear regression model operates on the encrypted data, performing the necessary computations to predict the user’s performance class label. The result is then decrypted to obtain the final prediction, ensuring that sensitive user information remains confidential throughout the process.

During unit testing, we evaluate the homomorphic encryption module by comparing the decrypted predicted output with the expected output for a set of known encrypted input data. If the decrypted output matches the expected output within an acceptable margin of error, the test case passes; otherwise, it fails.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input Data** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| Test 1 | [17, 23, 4] | 1 | 1.002 | Pass |
| Test 2 | [6, 14, 3] | 0 | 0.001 | Pass |
| Test 3 | [23, 31, 6] | 2 | 1.998 | Pass |
| Test 4 | [11, 19, 2] | 1 | 1.012 | Pass |
| Test 5 | [8, 22, 2] | 0 | 0.003 | Pass |

Table 7.2 Unit Testing – 2

In these test cases:

* “Input Data “ describes the user input features (e.g., time spent, quiz scores, quiz attempts).
* “Expected Output” indicates the expected behavior or result of the action.
* “Pass/Fail” determines whether the actual output matches the expected output.

#### UTC-2: User Interface Testing

The user interface (UI) module is a critical component of our system, providing users with an intuitive and engaging platform to interact with the federated learning system. Here, users can access course content, view statistics, take tests, and manage their accounts.

Ensuring the UI functions as intended is essential for delivering a seamless user experience.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input Data** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| Test 1 | User submits test by answering the questions | User gets course recommendations  according to the test score | User gets course recommendations  according to the test score | Pass |
| Test 2 | User logs in with valid credentials | User is directed to the home page | User is directed to the home page | Pass |
| Test 3 | User attempts to log in with invalid credentials | Error message: “Invalid username or password” | Error message: “Invalid username or password” | Pass |
| Test 4 | User navigates to the “Courses” section | List of available,  recommended and popular courses are displayed | List of available,  recommended and popular courses are displayed | Pass |
| Test 5 | User takes a test | Test is displayed with questions and options | Test is displayed with questions and options | Pass |
| Test 6 | User accesses statistics page | Personalized statistics are displayed | Personalized statistics are displayed | Pass |

Table 7.3 Unit Testing - 3

In these test cases:

* "Input/Action" describes the user action or input being tested.
* "Expected Output" indicates the expected behavior or result of the action.
* "Actual Output" shows what the system actually does in response to the input. • "Pass/Fail" determines whether the actual output matches the expected output.