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 behaviour. 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: Fall Detection Model

Unit testing for a fall detection model within an Android app involves breaking down the code into individual units, typically methods or functions, and testing each unit in isolation to ensure that it behaves as expected. For a fall detection feature, this entails testing various aspects of the model's functionality, such as its ability to accurately detect a fall event, trigger notifications to emergency contacts, and handle different scenarios gracefully. The unit tests for the fall detection model would focus on verifying its core functionality,

including the algorithm's accuracy in identifying falls based on sensor data collected by the device. By thoroughly testing each unit of the fall detection model, developers can gain confidence in its reliability and robustness, helping to ensure that it performs effectively in real-world situations. In the unit testing framework for the fall detection model, an important test case to include is the handling of multiple fall detections within a short time frame. Specifically, the model should be designed to treat any subsequent falls detected within 15 minutes of the first as false positives. This parameter is essential to avoid over-alerting emergency contacts or triggering unnecessary response actions, thereby reducing the risk of undue stress and resource deployment.

Test Case	**Input Data**	**Expected Output**	**Actual Output**	**Pass/Fail**
Test 1	Basic Fall Detection	Fall is detected and family is notified.	Fall is detected and family is notified.	Pass
Test 2	False Positive Test Case	A subsequent fall in under 15 minutes is considered false positive.	A subsequent fall in under 15 minutes is considered false positive.	Pass
Test 3	One to Many Emergency Contacts	All the emergency contacts are notified.	All the emergency contacts are notified.	Pass
Test 4	Network Connectivity Issue	Message is sent to a nearby bystander through Wi-Fi.	No message is sent.	Fail

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-1: User Interface Testing

The user interface (UI) module is a critical component of our system, providing users with an intuitive and engaging platform.

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 a medicine's data.	Reminder is set up, image and other data is backed up to cloud.	Reminder is set up, image and other data is backed up to cloud.	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 sends a query to the chatbot	The chatbot responds based on the query and if a particular threshold is met, then notifies the family.	The chatbot responds based on the query and if a particular threshold is met, then notifies the family.	Pass
Test 5	User roams out of the designated area.	Routine Analysis notifies the family and sends location update every minute.	Routine Analysis notifies the family and sends location update every minute.	Pass
Test 6	Family member accesses the elder's records.	All the medical records and data is displayed.	All the medical records and data is displayed.	Pass

Table 7.2 Unit Testing - 2

In these test cases:

- "Input/Action" describes the user action or input being tested.
- "Expected Output" indicates the expected behaviour 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.