A picture containing logo

Description automatically generated

American International University-Bangladesh (AIUB)  
**Department of Computer Science  
Faculty of Science & Technology (FST)**

**Health Guardian –** a health assistant app

A Software Quality and Testing Project Submitted

By

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Semester: Fall\_23\_24** | | | **Section: D** | **Group No: 08** |
| **SN** | **Student Name** | **Student ID** | Individual  Contribution (in %) | Total Marks: 50 |
| Earned Marks: |
| 43 | Asmaul Hossain Akash | 20-44209-3 | 25% |  |
| 21 | Md. Fahad Khan | 20-43328-1 | 25% |  |
| 41 | Naznin Nahar Khanom Mim | 20-43577-1 | 25% |  |
| 44 | Md. Shaiful Islam | 20-42747-1 | 25% |  |

The project will be Evaluated for the following Course Outcomes

|  |  |  |
| --- | --- | --- |
| **EVALUATION CRITERIA** | **Total Marks (50)** | |
|  | |
| Revision History, Test Plan Identifier, Reference Materials, Problem Background, Solutions | [10 Marks] |  |
| Requirements Specification (System feature, Quality Attributes, System Interface, Project Requirements) | [10 Marks] |  |
| Item Not to be tested, Testing approach (Testing levels, tools, meetings), Test cases | [10 Marks] |  |
| Item pass/fail criteria, Test deliverables, Staffing and Training, Responsibilities, Scheduling, Risk | [10 Marks] |  |
| Approval, Format, Submission, and Defense | [10 Marks] |  |

Software Test Plan

for

Health Guardian

Version 2.0 approved.

Prepared by Asmaul Hossain Akash

American International University – Bangladesh (AIUB)

24 December 2023

Table of Contents

[Revision History 3](#_Toc136846938)

[1. TEST PLAN IDENTIFIER: HGV-2.0 4](#_Toc136846939)

[2. REFERENCE MATERIALS 4](#_Toc136846940)

[3. INTRODUCTION 4](#_Toc136846941)

[3.1 Background to the Problem 4](#_Toc136846942)

[3.2 Solution to the Problem 4](#_Toc136846943)

[4. REQUEIREMNT SPECIFICATION 4](#_Toc136846944)

[4.1 System Features 5](#_Toc136846945)

[4.2 System Quality Attributes 8](#_Toc136846946)

[4.3 System Interface 10](#_Toc136846947)

[4.4 Project Requirements 12](#_Toc136846948)

[5. FEATURES NOT TO BE TESTED 12](#_Toc136846949)

[6. TESTING APPROACH 12](#_Toc136846950)

[6.1 Testing Levels 13](#_Toc136846951)

[6.2 Test Tools 13](#_Toc136846952)

[6.3 Meetings 13](#_Toc136846953)

[7. TEST CASES/TEST ITEMS 15](#_Toc136846954)

[8. ITEM PASS/FAIL CRITERIA 23](#_Toc136846955)

[9. TEST DELIVERABLES 23](#_Toc136846956)

[10. STAFFING AND TRAINING NEEDS 24](#_Toc136846957)

[11. RESPONSIBILITIES 24](#_Toc136846958)

[12. TESTING SCHEDULE 26](#_Toc136846959)

[13. PLANNING RISKS AND CONTINGENCIES 26](#_Toc136846960)

[14. APROVALS 27](#_Toc136846961)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Updated by** | **Update Comments** |
| 0.1 | 2023.11.18 | Asmaul Hossain Akash | Introduction, System Features |
| 0.2 | 2023.12.06 | Md. Fahad Khan | System Quality Attributes |
| 0.3 | 2023.12.17 | Naznin Nahar Khanom | Project Requirements, 5, 6.1, 8 |
| 0.4 | 2023.12.18 | Md. Shaiful Islam | Meetings, Test Deliverables, 10 |
| 0.5 | 2023.12.19 | Asmaul Hossain Akash | 11, Testing Schedule |
| 0.6 | 2023.12.22 | Md. Fahad Khan | 6.2, System Interface |
| 0.7 | 2023.12.22 | Asmaul Hossain Akash | Test Cases, 13 |
| 0.8 | 2023.12.24 | Asmaul Hossain Akash | Final Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# TEST PLAN IDENTIFIER: HGV-2.0

# REFERENCE MATERIALS

# <https://www.researchgate.net/figure/Hospital-Management-Top-level-Use-Cases_fig1_316422704>

# <https://www.slideshare.net/kataria55/srs-for-hospital-management-system>

# <https://www.freeprojectz.com/use-case/hospital-management-system-use-case-diagram>

1. <https://www.researchgate.net/publication/283664206_Major_Challenges_with_Mobile_Healthcare_Applications>

# 3. INTRODUCTION

## 3.1 Background to the Problem

In an era characterized by fast-paced lifestyles and evolving healthcare needs, individuals often face challenges in managing their health effectively. Access to timely and personalized healthcare information, monitoring tools, and reliable assistance has become crucial. The increasing prevalence of chronic diseases, coupled with the need for proactive health management, underscores the necessity for a comprehensive solution that empowers users to take charge of their well-being.

Healthcare systems, while advanced, often lack user-centric and integrated approaches to address the broader spectrum of individual health concerns. Traditional methods of health management may not fully leverage the potential of technology to provide personalized, accessible, and real-time support.

## 3.2 Solution to the Problem

*Health Guardian* emerges as an innovative and comprehensive solution to bridge the existing gaps in healthcare management. It is designed to address the root cause of the problem by offering a user-friendly platform that amalgamates health data, personalized guidance, and proactive monitoring.

The proposed solution revolves around providing users with a seamless and intuitive health assistant app that not only centralizes their health information but also leverages advanced algorithms and artificial intelligence to deliver personalized insights. By integrating features like health tracking, medication reminders, and real-time communication with healthcare professionals, Health Guardian aims to empower users to make informed decisions about their well-being.

The feasibility of this solution is rooted in its adaptability to meet diverse business objectives. The software envisions a holistic approach to health management, aligning with the evolving needs of users while ensuring scalability and compatibility with existing healthcare systems.

Existing studies in the problem area highlight a growing demand for comprehensive health management solutions. While some software solutions exist, many fall short in providing an integrated and user-centric experience. Health Guardian positions itself as a cutting-edge solution, drawing on insights from these existing studies to offer a refined and enhanced user experience. By learning from the strengths and weaknesses of predecessors, Health Guardian strives to set a new standard in the realm of health assistant apps.

# REQUEIREMNT SPECIFICATION

## System Features

Health Guardian is envisioned as a comprehensive health assistant app, integrating various features to provide users with a seamless and empowering health management experience. The system's functional requirements are outlined below:

**Functional Requirements**

1. **System Login** 
   1. The software shall provide a secure login mechanism, requiring users to enter their assigned username and password.
   2. In the event of three consecutive incorrect login attempts, the system shall generate a random verification code to be entered by the user before retrying the login process.
   3. If the number of consecutive incorrect login attempts reaches the defined limit of 5, the system shall have an optional function to block the user account login for a specified duration (e.g., one hour).

Priority Level: High  
Precondition: The user possesses a valid user ID and password.

Cross reference: 2

1. **User Registration and Profile Management** 
   1. The system shall provide a secure and user-friendly account creation process, ensuring the confidentiality of user information during registration.
   2. Users must be able to manage their profiles within the system. This includes the ability to input, edit, and update personal information, medical history, and preferences.
   3. Implement security measures during the registration process to verify the authenticity of user-provided information and prevent fraudulent account creation.

Priority Level: High  
Precondition: The user has access to a valid email address or mobile number for account verification.

Cross reference: 1

1. **Admin Login**
   1. Admin will log into the system with admin Username and password.
   2. The app should sync data between the mobile device and the smartwatch in real time.
   3. The app should provide a user-friendly interface on the smartwatch, ensuring that text is readable, and buttons are easily clickable on a smaller screen.

Priority Level: High

Precondition: The user has a compatible smartwatch.

Cross reference: N/A

1. **Notification Support**
   1. The app should send timely notifications to keep users updated about important events, reminders, or health updates. The app should sync data between the mobile device and the smartwatch in real time.
   2. Users should be able to customize the types of notifications they receive to avoid notification overload.
   3. The app should support push notifications, allowing users to receive important updates even when the app is not actively in use.
   4. Notifications should also appear on the smartwatch if one is connected. Priority Level: High Precondition: The user has granted the necessary permissions for the app to send notifications.

Priority Level: High

Precondition: The user has granted the necessary permissions for the app to send notifications.

Cross reference: 1

1. **Health Monitoring** 
   1. The App should be able to track the user’s health data, such as heart rate, steps taken, and sleep patterns, using sensors on the mobile device and/or smartwatch.
   2. The app should provide a daily, weekly, and monthly summary of the user’s health data in an easy-to-understand format.

Priority Level: High

Precondition: The user has granted the necessary permissions for the app to access health data.

Cross reference: 1

1. **Fitness Goals:**
   1. Users should be able to set personal fitness goals in the app, such as a daily step count or a target weight.
   2. The app should track the user’s progress towards their goals and provide motivational notifications when they are close to achieving them.

Priority Level: Medium

Precondition: The user has set at least one fitness goal.

Cross reference: 1,5

1. **Integration with Other Apps:**
   1. The app should be able to integrate with other health and fitness apps, allowing users to import and export data seamlessly.
   2. The app should support integration with social media platforms, enabling users to share their achievements with their friends.

Priority Level: Medium

Precondition: The user has granted the necessary permissions for the app to integrate with other apps.

Cross reference: 1,5,6

1. **Offline Support:**
   1. The app should provide offline support, allowing users to access key features and stored data even without an internet connection.
   2. Any changes made by the user while offline should be synced with the server once the device is back online.

Priority Level: High

Precondition: The user’s device is offline.

Cross reference: 4

1. **Data Security:**
   1. The app should ensure the security and privacy of the user’s data, both while it is stored on the device and when it is transmitted over the internet.
   2. The app should comply with all relevant data protection regulations.

Priority Level: High

Precondition: The user has provided personal data to the app.

Cross reference: 1,2

## System Quality Attributes

There are some software quality attributes as per ISO/ IEC 9126 that are very important to ensure the quality of software.

**QA1 - Functionality:** A valid user can see functionality and do all the tasks after login into the system. Invalid users cannot access the system. After verification the user will be valid.

Priority Level: High

Cross reference: N/A

**QA2 - Security:** In our system, users initiate the login process by registering, receiving a time-sensitive verification code via email or SMS. Upon entering the code, users confirm their identity, with an added encouragement to enable multi-factor authentication for enhanced security. We prioritize secure transmission through robust protocols and provide a reliable account recovery mechanism. Continuous monitoring and regular audits further fortify our system, ensuring resilience against potential security threats.

Priority Level: High

Cross reference: QA4

**QA3 - Reliability:** Our health monitoring app is designed for reliability. Users can trust its consistency and accuracy in delivering timely notifications for water intake and exercise alarms and many more. The system's reliability extends to goal tracking, providing users with a dependable platform for their health objectives.

Priority Level: Medium

Cross reference: QA5

**QA4 - Usability:** The app prioritizes user-friendliness, ensuring ease of registration and system navigation for all users. Our goal is to make health monitoring accessible and understandable to users of varying technological proficiencies, promoting widespread usability.

Priority Level: High

Cross reference: QA4, QA5

**QA5 - Efficiency:** Our system is optimized for efficiency, maintaining a small footprint to ensure compatibility with various devices. This streamlined approach enhances performance, making health monitoring accessible to users without compromising the device's capabilities.

Priority Level: High

Cross reference: QA2, QA5

**QA6 - Maintainability:** We commit to addressing any bugs or issues promptly, prioritizing system maintenance to ensure a seamless user experience. Continuous improvements and swift issue resolution underscore our dedication to maintaining the app's reliability and functionality.

Priority Level: Medium

Cross reference: QA1, QA2, QA5

**QA7 - Portability:** The app's agility facilitates swift host or environment switches, ensuring seamless transitions. Reinstallation is straightforward, contributing to the app's overall portability and adaptability across different platforms.

Priority Level: Low

Cross reference: QA1, QA6

**QA8 - Accessibility:** As a mobile app, our health monitoring system provides users with accessibility from anywhere through internet connectivity. This enables users to monitor their health and receive notifications irrespective of their location.

Priority Level: Medium

Cross reference: QA1, QA5

**QA9 - Installation:**

The installation process is simplified through app stores, guaranteeing easy access for users. The app's straightforward installation from the play store or app store ensures accessibility for a diverse user base.

Priority Level: Medium

Cross reference: QA1, QA2, QA3, QA4, QA7

## System Interface

|  |  |  |
| --- | --- | --- |
| Fig 1. System Login | Fig 2. System Registration | Fig 3. OTP Verification |
| Fig 4. Profile Management | Fig 5. Change Profile Info | Fig 6. Home Page (Health) |
| Fig 7. Exercise Record | Fig 8. Health Report | Fig 9. Interact with Fitness app |
| Fig 10. Set Move Goal | Fig 11. Watch Functionalities | Fig 11. Intaract with watch |

## Project Requirements

* Time: This app-based application may take about 11 months to complete.
* Budget: 15,50,000 BDT
* Size: The final size of this app-based application will not be more than 500-600 MB.
* Java, Flutter, JavaScript, and Ajax will be used to build this app-based application.

if we assume that the project is an organic type and the estimated LOC is 10,000 (or 10 KLOC), we can calculate the effort, development time, average staff, and productivity as follows:

**Effort**: E=a×(KLOC)b=2.4×(10)1.05=25.6 person-months

**Development Time**: D=c×(E)d=2.5×(25.6)0.38=10.6 months

**Average Staff**: Average staff=DE​=10.625.6​=2.4 staff

**Productivity**: Productivity=EKLOC​=25.610​=0.39 KLOC / person-month

# FEATURES NOT TO BE TESTED

**Third-Party Integrations:** The "Health Guardian" app leverages several third-party integrations, such as health data APIs and external services, previously vetted and validated by their respective providers. These integrations have undergone stringent validation processes, including certifications for reliability, security, and compatibility. Given their established reliability and certification status, retesting is deemed unnecessary to avoid redundant assessments of functionalities already confirmed by reputable providers, ensuring stability and security within the app.

**Non-Essential Historical Data:** The exclusion of testing for non-essential historical data within the "Health Guardian" app stems from an assessment of its minimal impact on core functionalities or user interactions. While historical data holds relevance for reference, its absence in direct testing is justified by its non-intrusive nature to the app's primary functions. Prior validations and integrity checks assure the data's correctness without necessitating dedicated testing efforts, ensuring that the focus remains on functionalities crucial to user experience and app performance.

**Minor UI/UX Updates:** Minor updates in the app's user interface and experience, though beneficial, are acknowledged for their limited potential to disrupt core functionalities. These updates aim to enhance user experience without significantly altering the app's fundamental workings. Given their minor scale and historical stability, these updates are presumed to have a negligible impact on the app's functionality. While not subjected to specific testing, their implementation aligns with enhancing user interactions and overall usability without compromising the app's core functionality.

# TESTING APPROACH

## Testing Levels

1. **Unit Testing:** This is the first testing level where individual software components are tested. The developers themselves conduct this testing as they are familiar with the code. However, to ensure thorough testing, it is recommended to involve a separate tester who has both unit testing experience and programming skills. This tester can provide a fresh perspective and catch issues the developers might overlook.
2. **Integration Testing:** This testing phase involves combining individual units and testing them as a group. The tester for this phase should be a full-time professional who is well-versed in the industry, understands the project specifications, and possesses skills in integration testing. Their role is to identify issues with the interaction between different system components.
3. **System Testing:** In this phase, the entire system is tested. The tester should be an industry-savvy full-time professional, experienced in system testing, and familiar with the project specifications. They will check the system’s compliance with specified requirements and evaluate its performance and functionality.
4. **Acceptance Testing:** This is the final phase, where the system is tested for acceptability. The tester should be a full-time professional knowledgeable about the industry, familiar with the project requirements, and experienced in acceptance testing. Their role is to confirm that the system meets the required delivery and end-user use criteria.

## Test Tools

• Performance Testing Tool: JMeter

• API Testing Tool: Postman

• Project Management Tool: Jira

• Development Phase Web Automation Tool: Selenium

## Meetings

**Weekly XP Team Meeting:**

* **Date/Time**: Every Sunday
* **Attendees**: XP team members, including developers, testers, project manager, customer representative
* **Agenda**:
  + Evaluate the status of the user stories and project goals.
  + Discuss any obstacles or problems.
  + Identify potential hazards and ways to mitigate them.
  + Plan for the upcoming week's testing, development, and customer cooperation work.
  + Discuss the customer representative's input and make any required adjustments in the development process.

**Daily Stand-Up Meetings:**

* **Date/Time**: Every weekday morning
* **Attendees**: XP team members, including developers, testers, project manager, customer representative
* **Agenda**:
  + Briefly discuss the tasks each team member completed yesterday, the projects they have planned for today, and any obstacles or difficulties.
  + Determine what needs to be discussed or done next in any situation.
  + Share updates on each other's work and progress with the team.

**Iteration Planning Meeting:**

* **Date/Time**: At the beginning of each iteration (usually 1-2 weeks)
* **Attendees**: XP team members, including developers, testers, project manager, customer representative
* **Agenda**:
  + Examine the user stories and project goals.
  + Calculate how much work each user story will require.
  + Identify potential hazards and ways to mitigate them.
  + Plan for the necessary tasks to complete each user story.
  + Discuss the customer representative's input and make any required adjustments to the development process.

**Iteration Review Meeting:**

* **Date/Time**: At the end of each iteration (usually 1-2 weeks)
* **Attendees**: XP team members, including developers, testers, project manager, customer representative
* **Agenda**:
  + Present the client representative with completed user stories.
  + Ask the client representative for their feedback.
  + Discuss any problems that arose during the iteration and decide how to fix them for the next one.
  + Identify potential ways to improve the development process.

# TEST CASES/TEST ITEMS

Table 1: Test Case for **Login Session**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Asmaul Hossain Akash | | | |
| Test Case ID: TC\_1 | | Test Designed date: 23/12/2023 | | | |
| Test Priority: High | | Test Executed by: | | | |
| Module Name: Login | | Test Execution date: | | | |
| Test Title: Login with valid email and password | |  | | | |
| Description: Check If login works perfectly with valid  email and password. | |  | | | |
| Precondition (If any): User must be registered into the system | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Open app 2. Enter valid email and password. 3. Click login button | Email: akash@gmail.com  Password: 123@$#67 | | Should be able to login |  |  |
| Post Condition: Redirect to Health Dashboard | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_2 | | Test Designed date: 23/12/2023 | | | |
| Test Priority: High | | Test Executed by: | | | |
| Module Name: Signup | | Test Execution date: | | | |
| Test Title: Signup With valid Information | |  | | | |
| Description: Check If signup works perfectly with valid information. | | | | | |
| Precondition (If any): N/A | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Click Signup Option 2. Put valid information’s to all the fields. 3. Click signup button | Fullname: Asmaul Hossain Akash  Dob: 18 Nov 2001  Height: 168cm  Weight: 62 kg  Email: [akasha@gmail.com](mailto:akasha@gmail.com)  Password: 123@#$QWe | | Should be able to register |  |  |
| Post Condition: Redirect to OTP confirmation page. | | | | | |

Table 2: Test Case for **Signup Session**

Table 3: Test Case for **OTP Verification**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_3 | | Test Designed date: 23/12/2023 | | | |
| Test Priority: Medium | | Test Executed by: | | | |
| Module Name: OTP Verify | | Test Execution date: | | | |
| Test Title: Verify email with OTP during signup | |  | | | |
| Description: check if OTP works or not | | | | | |
| Precondition (If any): user must signup with valid information | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Enter the code sent to the email address provided. 2. Click Verify Code. | Verification Code: 1478 | | A popup should appear as email verified successfully. |  |  |
| Post Condition: Redirect to sign in page. | | | | | |

Table 4: Test Case for **Change Profile Information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_4 | | Test Designed date: 23/12/2023 | | | |
| Test Priority: Medium | | Test Executed by: | | | |
| Module Name: Change account information | | Test Execution date: | | | |
| Test Title: Change Personal Information | |  | | | |
| Description: Check if personal information change works or not | | | | | |
| Precondition (If any): user must log in with valid credentials | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Click the account button and go to the personal information page. 2. Click and try to change UserID, Sex, Birthday, Height, Weight 3. Click save option | Sex: Male / Female  Birthday: Jan 01, 2000  Height: 170 cm  Weight: 65 kg | | 1 – UserID cannot be changed.  2 – Other properties should be able to change |  |  |
| Post Condition: Redirect to account page. | | | | | |

Table 5: Test Case for **Update Account Information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_5 | | Test Designed date: 23/12/2023 | | | |
| Test Priority: High | | Test Executed by: | | | |
| Module Name: Update account information | | Test Execution date: | | | |
| Test Title: Change Login Information | |  | | | |
| Description: Check if email, password change works or not | | | | | |
| Precondition (If any): user must log in with valid credentials | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Click the account button and go to the login and security page. 2. Click and try to change email, password, and contact number. 3. Click save option | Email: [newemailakash@gmail.com](mailto:newemailakash@gmail.com)  Password: 123new@Pass  Contact number: 01781770073 | | Should be able to update and redirect to OTP verification page. |  |  |
| Post Condition: Redirect to account page. | | | | | |

Table 6: Test Case for **Logout Session**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_6 | | Test Designed date: 23/12/2023 | | | |
| Test Priority: Low | | Test Executed by: | | | |
| Module Name: Logout | | Test Execution date: | | | |
| Test Title: Logout session | |  | | | |
| Description: Check if logout button works or not | | | | | |
| Precondition (If any): user must log in with valid credentials | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Click the account button. 2. Click logout button. | N/A | | Should be able to logout successfully. |  |  |
| Post Condition: Redirect to login page. | | | | | |

Table 7: Test Case for **Interact with watch.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_7 | | Test Designed date: 23/12/2023 | | | |
| Test Priority: High | | Test Executed by: | | | |
| Module Name: Interact with watch. | | Test Execution date: | | | |
| Test Title: Test the connection of the smart watch. | |  | | | |
| Description: Check if the smart watch can be connected and synced. | | | | | |
| Precondition (If any): user must sign in and smart watch must be put on. | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Click the device button. 2. Click the add device button. 3. Select the desired watch from the device list. | (Exploratory test) | | Should be able to connect with the smart watch and data can be synced. |  |  |
| Post Condition: Connected watch should appear. | | | | | |

Table 8: Test Case for **Activity Record Module**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_8 | | Test Designed date: 23/12/2023 | | | |
| Test Priority: High | | Test Executed by: | | | |
| Module Name: Activity Record | | Test Execution date: | | | |
| Test Title: Test health related activity | |  | | | |
| Description: Check if health related activities works or not | | | | | |
| Precondition (If any): user must sign in and smart watch must be put on. | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Click the health button. 2. Click and test Heart rate, blood pressure, blood oxygen saturation, etc. | Activities data should have generated from the watch synced data individually. | | Should be able to show the generated data. |  |  |
| Post Condition: N/A | | | | | |

Table 9: Test Case for **Heath Report Module**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_9 | | Test Designed date: 23/12/2023 | | | |
| Test Priority: High | | Test Executed by: | | | |
| Module Name: Health Report | | Test Execution date: | | | |
| Test Title: Test health report | |  | | | |
| Description: Check if health report generates or not | | | | | |
| Precondition (If any): user must sign in and smart watch must be put on. | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Click the health button. 2. Click health report button. 3. Select generate new report option. | Reports are synced from the smart watch | | Should be able to generate a report based on synced data. |  |  |
| Post Condition: redirect to a detailed report page | | | | | |

Table 10: Test Case for **Watch Functionality Module**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_10.1 | | Test Designed date: 23/12/2023 | | | |
| Test Priority: High | | Test Executed by: | | | |
| Module Name: Check watch functionality | | Test Execution date: | | | |
| Test Title: Test Incoming call and SMS | |  | | | |
| Description: Check if the watch can remind if there is an incoming call or SMS | | | | | |
| Precondition (If any): the smart watch should be connected to the app. | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Keep app and watch in idle mode. 2. Make a call from another device. 3. Send SMS from another device. | Call from: 01678245321  SMS: Hello, keep testing | | Watch should be able to vibrate and notify user. |  |  |
| Post Condition: N/A | | | | | |

Table 11: Test Case for **App Notification Module**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_10.2 | | Test Designed date: 23/12/2023 | | | |
| Test Priority: High | | Test Executed by: | | | |
| Module Name: Check watch functionality | | Test Execution date: | | | |
| Test Title: Test App Notification | |  | | | |
| Description: Check if the watch can remind if there is an incoming notification from specific apps | | | | | |
| Precondition (If any): the smart watch should be connected to the app. | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Keep app and watch in idle mode. 2. Send a message in WhatsApp from another device. | WhatsApp: hello  Messenger: Keep Testing | | Watch should be able to vibrate and notify user. |  |  |
| Post Condition: N/A | | | | | |

Table 12: Test Case for **Reminder Functionality Module**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_11 | | Test Designed date: 23/12/2023 | | | |
| Test Priority: High | | Test Executed by: | | | |
| Module Name: Reminder Functionality | | Test Execution date: | | | |
| Test Title: Test Drink Water Reminder | |  | | | |
| Description: Check if drink water reminder works or not | | | | | |
| Precondition (If any): app must run in the background. | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Set the reminder to every 10 minutes. 2. Keep the app in idle mode. 3. Turn on the reminder | Reminder time: Each 10 minutes | | App should be able to send reminder notification about drinking water every 10 minutes. |  |  |
| Post Condition: N/A | | | | | |

Table 13: Test Case for **Interact with other Apps Module**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_12 | | Test Designed date: 24/12/2023 | | | |
| Test Priority: High | | Test Executed by: | | | |
| Module Name: Interact with other apps | | Test Execution date: | | | |
| Test Title: Test Fitness app | |  | | | |
| Description: Check if health guardian can interact with fitness app or not. | | | | | |
| Precondition (If any): Apple devices should have updated to IOS 16. | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Tap exercise button. 2. Explore the fitness app. | Tester Expectations | | Should be able to interact with fitness app and generate data from fitness synced data. |  |  |
| Post Condition: N/A  Table 14: Test Case for **Change Daily Move Goal Feature** | | | | | |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_13 | | Test Designed date: 24/12/2023 | | | |
| Test Priority: Low | | Test Executed by: | | | |
| Module Name: Daily Move Goal | | Test Execution date: | | | |
| Test Title: Change Daily Move Goal | |  | | | |
| Description: Check if health guardian can change daily move goal or not. | | | | | |
| Precondition (If any): N/A | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Tap change daily move goal button. 2. Change the value. 3. Tap save | Kilo Kal/Day: 150  Steps Goal: 6000 | | Should be able to change daily move goals. |  |  |
| Post Condition: N/A | | | | | |

Table 15: Test Case for **Offline Support Feature**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Heath Guardian | | Test Designed by: Akash | | | |
| Test Case ID: TC\_14 | | Test Designed date: 24/12/2023 | | | |
| Test Priority: Medium | | Test Executed by: | | | |
| Module Name: Offline Support | | Test Execution date: | | | |
| Test Title: Check the offline availability. | |  | | | |
| Description: Check if health guardian can work in offline or not. | | | | | |
| Precondition (If any): Mobile data / Wi-Fi should have turned off, watch should be connected. | | | | | |
| Test Steps | Test Data | | Expected Results | Actual Results | Status |
| 1. Turn off the internet on the device. 2. Call, send SMS from other apps. 3. Water Drinking Reminder turned on | Call from another device.  SMS: Hello, keep testing.  Water Drink Reminder set to 10 minutes | | Should be able to interact with the health guardian app n offline mode and all the Notification functionalities should work. |  |  |
| Post Condition: N/A | | | | | |

# ITEM PASS/FAIL CRITERIA

The primary aim of this section is to outline the criteria for passing or failing the tests associated with this project. The requirements are as follows:

* **Failure Criteria:** Any system or unit that achieves a score of less than 90% will be deemed to have failed. The component, team, or system must meet the expected standards or requirements.
* **Pass Criteria:** Any component, unit, system, or integrated test item that achieves a score ranging from 90% to 95% will be considered to have passed. This indicates that the thing has met the necessary standards and requirements.

These criteria ensure that all components, units, and systems meet the necessary quality standards, thereby ensuring the project's overall success.

# TEST DELIVERABLES

Test deliverables are essential components of the testing process that serve as comprehensive documentation for stakeholders involved in software development. These documents, tools, and equipment are crucial for facilitating and maintaining practical testing activities throughout a project's lifecycle. The following is a detailed list of essential test deliverables:

* **Test Plan:** Detailed documentation outlining the testing strategy, approach, scope, resources, and schedule.
* **Test Cases:** A comprehensive set of test scenarios and steps to be executed to validate the software's functionality.
* **Test Scripts:** Automated scripts designed to execute test cases, enhancing efficiency and repeatability in testing.
* **Test Data:** Sets of input data are designed to cover various scenarios, ensuring thorough software testing.
* **Traceability Matrix:** Document linking requirements to test cases, ensuring complete test coverage and validation.
* **Defect Reports:** Detailed documentation of identified defects, including their severity, priority, and reproduction steps.
* **Test Summary Report:** Comprehensive overview of the testing process, including test execution results, defect statistics, and overall project quality.
* **Test Environment Setup:** Documentation detailing the configuration and setup of the testing environment, including hardware, software, and network specifications.
* **Test Progress Reports:** Regular updates on the progress of testing activities, highlighting completed tasks, ongoing efforts, and any challenges encountered.
* **Test Completion Report:** A conclusive report summarizing the entire testing process, including achievements, challenges, and recommendations for future improvements.
* **Test Metrics:** Quantitative testing performance measures include metrics such as test coverage, defect density, and test execution efficiency.
* **User Manuals for Testing Tools:** Documentation guiding the usage of testing tools employed during the testing process.
* **Training Materials:** Materials to train testing teams on testing processes, tools, and methodologies.
* **Risk and Mitigation Plan:** Documentation outlining potential risks in the testing process and strategies to mitigate them.
* **Test Sign-Off:** Formal documentation indicating that testing activities have been completed and the software is ready for release.

# STAFFING AND TRAINING NEEDS

Specific staffing and training needs have been identified across various areas to ensure the successful implementation and maintenance of the project. The following outlines the training requirements for different teams involved in the project:

#### Training Needs:

* **EDI Interface:**
  + Target Audience: Developers, Testers, Operations Team
  + Training Focus: Basics of EDI interface, understanding EDI communications procedures.
* **New Screens and Reports:**
  + Target Audience: Sales Administration Employees
  + Training Focus: Familiarization with new screens and reports, including functionalities and usage.
* **Test Planning and Execution:**
  + Target Audience: Testing Team
  + Training Focus: Comprehensive training on test planning strategies and execution methodologies.
* **Defect Management:**
  + Target Audience: Testing Team
  + Training Focus: In-depth understanding of defect management processes, including identification, reporting, and resolution.
* **Test Automation:**
  + Target Audience: Testing Team
  + Training Focus: Proficiency in test automation tools and frameworks, emphasizing efficiency and repeatability.
* **Security:**
  + Target Audience: Not Specified
  + Training Focus: Identify and address security concerns relevant to the project, including best practices and compliance.

#### Staffing Needs:

* **EDI Interface:**
  + Roles: Additional support for developers and testers familiar with EDI integration.
  + Justification: To ensure a smooth integration of the EDI interface, additional expertise is required in the development and testing teams.
* **New Screens and Reports:**
  + Roles: None specified.
  + Justification: Monitor if additional staffing is necessary for the sales administration team to handle new functionalities effectively.
* **Test Planning and Execution:**
  + Roles: Test Leads or Managers.
  + Justification: Additional leadership roles may be required to oversee and manage the testing process effectively.
* **Defect Management:**
  + - Roles: Defect Coordinator or Manager.
  + Justification: Assign a dedicated role to handle the end-to-end defect management process, ensuring timely resolution.
* **Test Automation:**
  + Roles: Automation Engineers.
  + Justification: Employ specialized resources to focus on test automation and maximize efficiency in the testing process.
* **Security:**
  + Roles: Security Analyst or Consultant.
  + Justification: Introduce specialized personnel to assess and address security concerns in the project.

# RESPONSIBILITIES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | TM | PM | Dev. Team | Test Team | Client |
| Acceptance test Documentation & Execution | X | X |  | X | X |
| System/Integration test Documentation & Execution | X |  | X | X |  |
| Unit test documentation & execution | X |  | X | X |  |
| System Design Reviews | X | X | X | X | X |
| Detail Design Reviews | X | X | X | X |  |
| Test Procedures and rules | X | X | X | X |  |
| Screen & Report Prototype reviews |  |  | X | X | X |
| Change control and Regression testing | X | X | X | X | X |

# TESTING SCHEDULE

Time has been allocated within the project plan for the following testing activities. The specific dates and times for each activity are defined in the project plan timeline. Coordination of the personnel required for each task, test team, development team, management and customer will be handled by the project manager in conjunction with the development and test team leaders.

A screenshot of a computer

Description automatically generated

# PLANNING RISKS AND CONTINGENCIES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S/N** | **Risk Description** | **Probability** | **Impact** | **Mitigation Plan** |
| 1 | Unrealistic Deadlines | 60% | Delay project 1 week | Take multiple estimation, prioritize testing tasks and communicate with stakeholders. |
| 2 | Lack of Skilled Testers | 50% | Low-quality testing | Invest in training, hire skilled testers, and cross-train team members to ensure necessary expertise. |
| 3 | Inadequate Requirements | 40% | Incorrect testing, rework | Conduct thorough requirements analysis and involve stakeholders in requirement validation. |
| 4 | Change in Project Requirements | 30% | Project delay, increased costs | Regular communication with stakeholders, flexible project planning, use of agile methodologies. |
| 5 | Technical Debt | 50% | Increased maintenance cost, decreased code quality | Regular code reviews, adherence to coding standards, prioritizing refactoring tasks. |
| 6 | Software/Hardware Failures | 20% | Project delay, data loss | Regular backups, use of reliable hardware, having a disaster recovery plan. |
| 7 | Security Risks | 40% | Data breaches, legal issues | Adherence to security best practices, regular security audits, training staff on security awareness. |
| 8 | Resource Allocation Issues | 30% | Project delay, decreased productivity | Efficient resource management, contingency planning for resource allocation. |

# APROVALS

|  |  |
| --- | --- |
| Project Sponsor |  |
| Development Management |  |
| EDI Project Manager |  |
| RS Test Manager |  |
| RS Development Team Manager |  |
| Reassigned Sales |  |
| Order Entry EDI Team Manager |  |