American International University-Bangladesh (AIUB)
Department of Computer Science
Faculty of Science &Technology (FST)
Summer 22 23

Section: C
Software Quality Assurance and Testing

Online Doctor Consultancy Software (**TeleHealthNow**)

A Report submitted
By

SN	Student Name	Student ID
1	S.M Moinul Islam	20-44227-3
2	Sadia Hossain Bushra	20-44238-3
3	I S M Salim Sadman	20-44260-3
4	Shihabun Sakiba Jannat	20-41992-1

Under the supervision of
Md. Moman UL Haque Khan
LECTURER, Faculty , Computer Science, American International University-Bangladesh

Checked By Industry Personnel

Name: Ahanaf zahin ragib

Designation: Tester: mygp app

Company: Grameenphone

Sign:

Date: 17-08-2023

Software Test Plan

for

TeleHealthNow

Version 1.0 approved

Prepared by

I S M Salim Sadman

S.M Moinul Islam

Sadia Hossain Bushra

Shihabun Sakiba Jannat

American International University-Bangladesh

Date:17-08-2023

Table of Contents

Revision History

Revision	Date	Updated by	Update Comments
0.1	09-08-2023	Sadia Hossain Bushra	First Draft
0.2	09-08-2023	S.M Moinul Islam	Second Draft
0.3	09-08-2023	I S M Salim Sadman	Third Draft
0.4	09-08-2023	Shihabun Sakiba Jannat	Fourth Draft
0.5	09-08-2023	S.M Moinul Islam	Fifth Draft
0.6	09-08-2023	Sadia Hossain Bushra	Sixth Draft
0.7	09-08-2023	Shihabun Sakiba Jannat	Seventh Draft
0.8	09-08-2023	I S M Salim Sadman	Eighth Draft
0.9	09-08-2023	Sadia Hossain Bushra	Final Draft

1. **TEST PLAN IDENTIFIER:**TP-TeleHealthNow01.3

2. REFERENCES

- Software Quality And Testing Course PowerPoint Slides
- Software Quality Engineering: Testing, Quality Assurance and Quantifiable Improvement – Jeff Tian

3. INTRODUCTION

Background to the Problem

In recent years, the healthcare industry has been faced with a number of challenges, including increasing healthcare costs, shortages of healthcare providers, and limited access to healthcare services in certain geographic regions.

In response to these challenges, telemedicine has emerged as a viable solution to provide patients with access to medical care regardless of their location. According to Chiron Health (n.d.), their guide provides medical providers with information on building a successful telemedicine business.Telemedicine involves the use of technology to remotely connect patients with healthcare providers for diagnosis, treatment, and medical advice. One aspect of telemedicine is online doctor consultancy, which allows patients to consult with licensed physicians and healthcare professionals over the internet. Despite the many benefits of telemedicine and online doctor consultancy, there are still some barriers to adoption, including concerns around data privacy and security, limited insurance coverage, and a lack of awareness and education about telemedicine among patients and healthcare providers. Doxy.me (2021) published a comprehensive guide on the state of telemedicine in 2021.

Therefore, the problem that TeleHealthNow aims to address is to provide a secure, reliable, and user-friendly platform that connects patients with licensed physicians and healthcare providers to receive medical advice, diagnosis, and treatment remotely, while addressing the barriers to adoption of telemedicine.

Solution to the Problem

a) Data privacy and security concerns can be addressed by implementing robust security measures such as encryption of data, access controls, and regular security audits to ensure the confidentiality, integrity, and availability of patient information.

b) Limited insurance coverage can be addressed by partnering with insurance companies to cover telemedicine services, or by offering affordable pricing plans for patients who do not have insurance coverage.

c) Lack of awareness and education about telemedicine can be addressed by creating educational content for patients and healthcare providers to understand the benefits and limitations of telemedicine, and by conducting awareness campaigns to promote telemedicine services.

The solutions are feasible and can be implemented by leveraging existing technology and partnerships with healthcare providers and insurance companies. The American Medical Association (2019) published a practical guide on incorporating telemedicine into medical practice.

TeleHealthNow is an online mobile based doctor consultancy platform that connects patients with licensed physicians and healthcare providers to receive medical advice, diagnosis, and treatment remotely. The platform should be user-friendly, secure, and reliable, with features such as video consultations, secure messaging, online prescriptions, and electronic medical records. The purpose of TeleHealthNow is to provide patients with convenient access to medical care from the comfort of their homes, while addressing the barriers to adoption of telemedicine identified in the problem statement. The goals and objectives of the software are to improve patient outcomes, increase patient satisfaction, reduce healthcare costs, and increase the adoption of telemedicine services.

TeleHealthNow can improve access to medical care, reduce healthcare costs, and increase patient satisfaction. There are many existing software solutions available to solve the problem, including Teladoc, Medico, Amwell, and Doctor on Demand. ScienceSoft (n.d.) provides insights on how to develop a successful telemedicine platform. These platforms offer similar features such as video consultations, secure messaging, online prescriptions, and electronic medical records. However, the proposed software aims to address the specific barriers to adoption of telemedicine identified in the problem statement, such as data privacy and security concerns, limited insurance coverage, and lack of awareness and education about telemedicine. Rock Health (2019) conducted a study on digital health consumer adoption in 2019.

4. REQUEIREMNT SPECIFICATION

4.1 System Features

4.1.1 Functional Requirements

• **User Registration**

- Priority: High

- Pre-condition: The user does not have an existing account on the platform.

- The platform should allow users to create an account, including personal information and medical history.

• **User Profile Management**

- Priority: High

- Pre-condition: The user must be registered and logged into their account.

- The platform should allow users to access their account and manage them.

• **Scheduling Appointments**

- Priority: High

- Pre-condition: The user has an active profile and the healthcare provider is available for booking.

-The platform should enable users to schedule appointments with licensed physicians and healthcare providers at their convenience.

• **Video Conferencing**

- Priority: Low

- Pre-condition: Both the patient and healthcare provider have stable internet connections and compatible devices.

- The platform should provide secure and reliable video conferencing capabilities for virtual consultations between patients and healthcare providers.

• **Medical Records Management**

- Priority: High

- Pre-condition: The healthcare provider has been granted access permissions by the patient.

- The platform should allow healthcare providers to access and manage patients' medical records securely.

• **Prescription Delivery**

- Priority: High

- Pre-condition: The healthcare provider has diagnosed or consulted the patient and determined a prescription is necessary.

- The platform should provide a secure method for healthcare providers to send prescriptions electronically to patients' pharmacies of choice.

• **Payment Processing**

- Priority: High

- Pre-condition: The user has received services or consultations from a healthcare provider.

- The platform should enable users to make payments for services rendered securely.

4.2 System Quality Attributes

4.2.1 Non-Functional Requirements

• **Security**

- Priority: Critical

- Pre-condition: The user has registered on the platform and is actively engaging with its features.

- The platform should implement robust security measures to protect patients' data and ensure privacy

• **Reliability**

- Priority: Critical

- Pre-condition: The platform is live and operational.

- The platform should be available 24/7 and provide a reliable video conferencing experience with minimal downtime.

• **Usability**

- Priority: High

- Pre-condition: The platform is designed following modern UI/UX guidelines.

- The platform should be user-friendly and easy to navigate for both patients and healthcare providers.

• **Scalability**

- Priority: High

- Pre-condition: The platform infrastructure is set up with growth projections in mind.

- The platform should be able to handle increasing user demand and provide a seamless user experience.

• **Compatibility**

- Priority: High

- Pre-condition: The platform's development considers a variety of devices and operating systems in its design.

- The platform should be compatible with multiple devices and operating systems.

• **Accessibility**

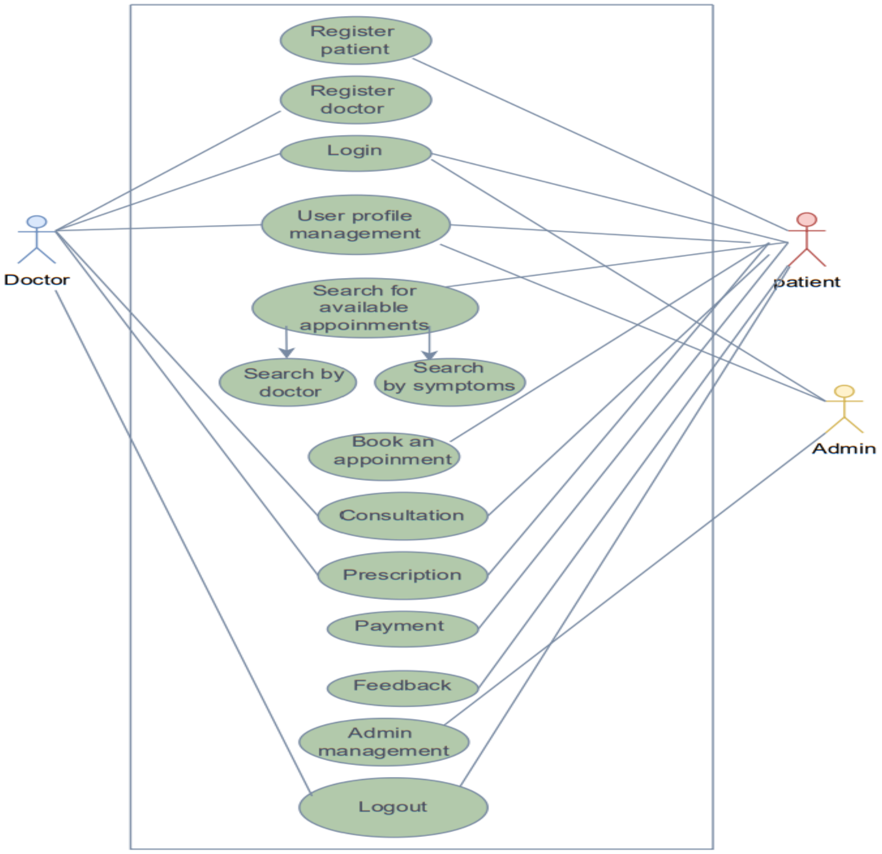
- Priority: High

- Pre-condition: The platform's design and functionalities consider the needs of differently-abled users.

- The platform should be accessible to users with disabilities, such as those who use assistive technologies.

4.3 System Interface (UML Diagram)

4.3.1 Use-Case Diagram:



Actors:

1. Patient - the user who seeks medical advice or treatment from a licensed physician or healthcare provider through the online doctor consultancy software.

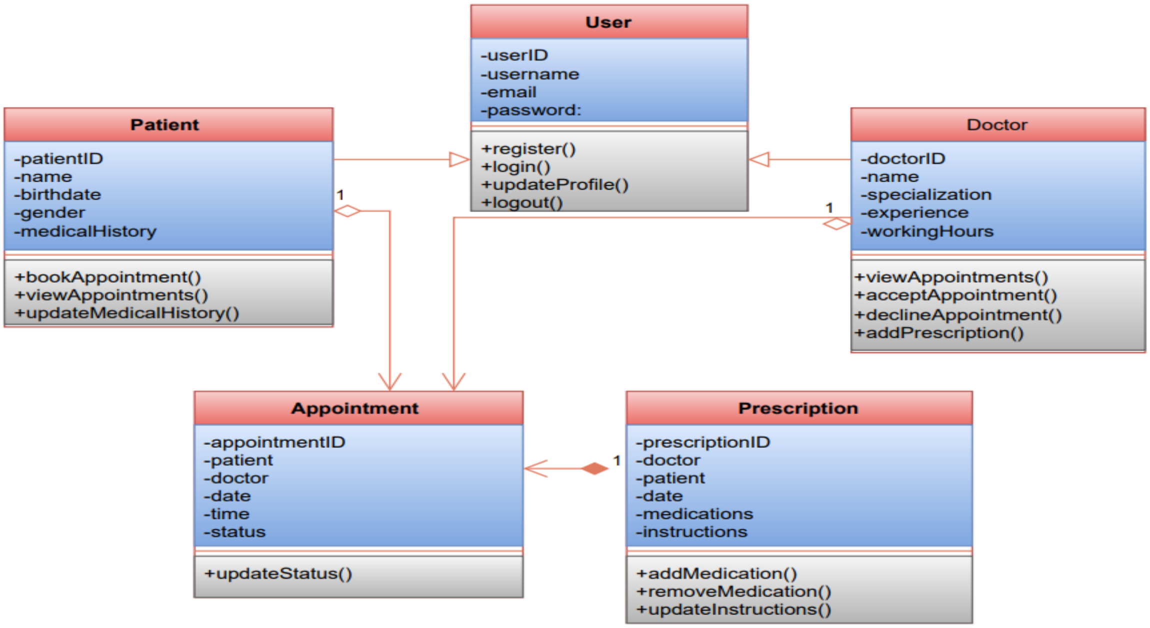
2. Doctor - the licensed physician or healthcare provider who provides medical advice, diagnosis, and treatment to patients through the online doctor consultancy software.

3. Admin - the user who manages the system and has access to all functionalities of the online doctor consultancy software.

Use cases:

1. Patient Registration - the patient creates a profile on the online doctor consultancy software to access medical services.
2. Doctor Registration - the doctor creates a profile on the online doctor consultancy software to provide medical services.
3. Login - the user logs into the online doctor consultancy software with their registered credentials.
4. Search - the patient searches for a doctor or a specific medical service on the platform.
5. Book Appointment - the patient books an appointment with a specific doctor for medical consultation.
6. Consultation - the doctor provides medical consultation and treatment to the patient through the online doctor consultancy software.
7. Prescription - the doctor provides a prescription for the patient's medical treatment.
8. Payment - the patient pays for the medical consultation and treatment provided by the doctor.
9. Feedback - the patient provides feedback on their experience with the online doctor consultancy software and the medical consultation provided by the doctor.
10. Admin Management - the admin manages the system, including user accounts, doctor profiles, and system functionalities.

4.3.2 Class Diagram:



1. **User Class:** The User class denotes the generic attributes and behaviors common among all system users. Attributes: userID, username, email, password. Methods: registration(), login(), updateUserProfile().

2. **Patient Class:** The Patient class symbolizes individuals seeking medical advice through the platform. It inherits attributes from the User class. Attributes: patientID, name, birthdate, gender, medicalHistory. Methods: bookAppointment(), viewAppointment(), updateMedicalHistory().

3. **Appointment Class:** This class captures the nuances of appointments scheduled between patients and doctors. Attributes: appointmentID, date, time, patient, doctor, status. Methods: updateStatus(), rescheduleAppointment().

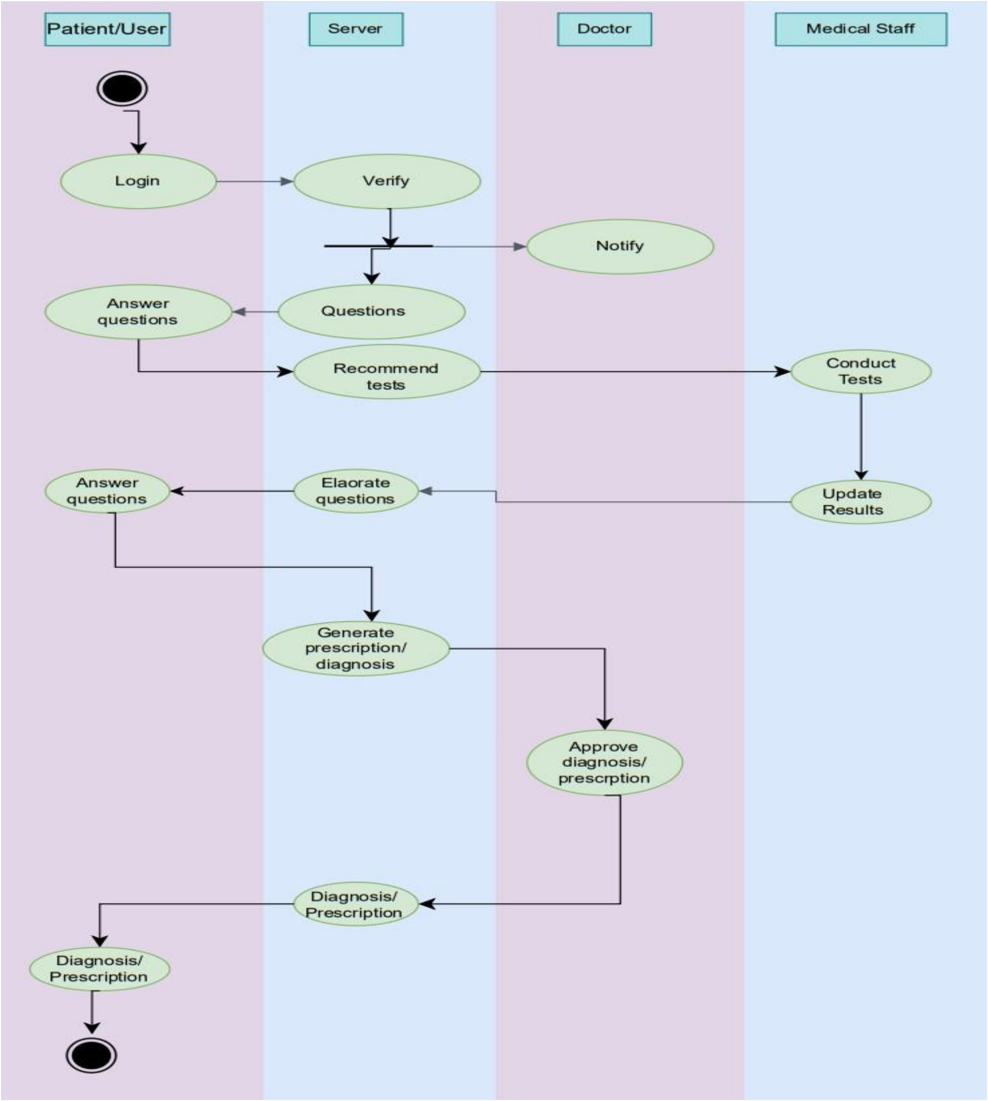
4. **Prescription Class:** This class embodies the electronic prescriptions provided by doctors to patients. Attributes: prescriptionID, patient, doctor, date, medications, instructions. Methods: addMedication(), generatePrescriptionPDF().

5. **Doctor Class:** The Doctor class stands for medical professionals offering consultation via the platform. It takes attributes from the User class. Attributes: doctorID, specialization, licenseNumber. Methods: viewAppointments(), generatePrescription().

Relationships:

- **User-Patient:** The User class relates generally to the Patient class, indicating that users can be patients.
- **User-Doctor:** The User class relates generally to the Doctor class, suggesting that users can also be doctors.
- **Patient-Appointment:** The Patient class associates with the Appointment class through a booking relationship, denoting the appointments patients make.
- **Doctor-Appointment:** The Doctor class associates with the Appointment class, illustrating the appointments involving doctors.
- **Doctor-Prescription:** The Doctor class connects with the Prescription class, showing the prescriptions doctors create.
- **Patient-Prescription:** The Patient class ties up with the Prescription class, indicating the prescriptions that patients receive.

4.3.3 Activity Diagram:



There are four main actors in the system:

1. Patient/User
2. Server
3. Doctor
4. Medical Staff

The process unfolds as follows:

1. The patient/user logs into the server.
2. The server verifies the patient/user's login credentials.
3. Once the verification is successful, the server notifies the doctor of the patient's presence.
4. Subsequently, the server presents a set of questions to the patient/user.
5. The patient/user responds to these questions.
6. Based on the answers, the server suggests specific medical tests and forwards this information to the medical staff.
7. The medical staff conducts the recommended tests. After completion, they update the test results in the system.
8. The server then prompts additional questions to the patient/user, who provides the necessary answers.
9. Considering all the gathered data, the server drafts a preliminary prescription/diagnosis for the patient.
10. This draft is then sent to the doctor for review.
11. The doctor examines the provided prescription and either approves or modifies it.
12. Once approved, the finalized prescription/diagnosis is stored in the server.
13. Finally, the server presents the patient/user with the approved prescription/diagnosis.

4.4 Project Requirements

Project Constraints:

Time Constraints:

- Project duration: The TeleHealthNow platform needs to be launched in a duration of 12 months.
- Milestones:
    1. Requirement Analysis and Design: 3 months
    2. Development Phase: 6 months
    3. Testing and Deployment: 3 months

Budget Constraints:

- Estimated total budget: BDT 10,000,000 Breakdown:
    1. Personnel (Developers, QA, Project Managers, and others): BDT 5,500,000
    2. Infrastructure (servers, software licenses, etc.): BDT 1,500,000
    3. Marketing and Awareness campaigns: BDT 1,500,000
    4. Miscellaneous and contingencies: BDT 1,500,000

Resources Constraints:

- Human Resources: A team of 2 project managers, 4 developers, 2 QA professionals, and 1 UI/UX designer.
- Technical Resources: Cloud servers, development tools licenses, and QA tools.

Environment Constraints:

- Development Environment: Agile methodology with 2-week sprints.
- Deployment Environment: Cloud-based hosting with end-to-end encryption for data security.

Budget Estimation:

Total Budget: BDT 10,000,000

- Personnel:
  - Project Managers: BDT 1,000,000 (2 PMs)
  - Developers: BDT 3,000,000 (4 Developers)
  - QA professionals: BDT 800,000 (2 QAs)
  - UI/UX designer: BDT 700,000 (1 Designer)
- Infrastructure: BDT 1,500,000
- Marketing and Awareness campaigns: BDT 1,500,000
- Miscellaneous and contingencies: BDT 1,500,000

Time Estimation:

Total Development Time: 12 months

Breakdown:

- Requirement Analysis and Design: 3 months
  - Requirement Gathering: 1 month
  - Design: 2 months
- Development Phase: 6 months
  - Frontend Development: 3 months
  - Backend Development: 3 months
- Testing and Deployment: 3 months
  - Testing: 2 months
  - Deployment: 1 month

5. FEATURES NOT TO BE TESTED

The following is a list of the areas that will not be specifically addressed. All testing in these areas will be indirect as a result of other testing efforts.

1. **Third-party Integrations:**
   - Payment gateways, video conferencing APIs, and any other third-party tools that have been integrated will not be tested directly, as they are assumed to have their own testing mechanisms and are externally validated.
2. **Backend Infrastructure:**
   - The underlying cloud infrastructure and physical hardware will not be tested as they are provided by reputed vendors and are assumed to meet the required performance and reliability standards.
3. **Device-specific Compatibility:**
   - While we aim for broad device compatibility, we will not test the application on every available device or operating system version. We'll focus on the most common devices and OS versions.
4. **Network Performance:**
   - While we will ensure that our application performs efficiently, the actual network performance based on different ISPs or geographical locations will not be explicitly tested.
5. **External Databases:**
   - If our platform uses third-party databases (like for drug information, insurance details, etc.), these won't be tested as they are assumed to be accurate and maintained by their respective organizations.
6. **Legacy Data:**
   - While we will ensure that our platform can integrate with older systems if needed, the quality and accuracy of legacy data from previous systems will not be our testing responsibility.
7. **Default Settings of Tools:**
   - Many tools or components come with default configurations and settings which are assumed to be optimal for most use cases. We won't be testing these default configurations unless there's a specific need to customize them.
8. **Documentation:**
   - Though documentation is crucial for user understanding and regulatory compliance, the content of the documentation itself (like user manuals) won't be part of the software testing process.

6. TESTING APPROACH

6.1 Test Level

1. Unit Testing:

Unit testing involves validating individual components of the software application to ensure they work as intended. These components can be methods, functions, classes, or modules. The main goal is to verify that each unit of the software performs as designed.

Based on Requirement Specification:

a. User Registration:

- Test for successful registration with valid details.
- Test for unsuccessful registration with missing or invalid details (like invalid email format, short password).
- Ensure the medical history provided by users during registration is accurately stored.

b. User Profile Management:

- Test for successful login.
- Test for unsuccessful login (wrong password, unregistered email).
- Test for successful profile updates.

c. Scheduling Appointments:

- Ensure users can schedule an appointment with the desired healthcare provider.
- Test for clashes in appointment scheduling.

d. Video Conferencing:

- Verify video connection initialization.
- Check for the termination of video sessions.

e. Medical Records Management:

- Validate the addition and retrieval of patient's medical records.
- Ensure records' confidentiality and integrity.

f. Prescription Delivery:

- Validate that prescriptions are generated with accurate patient and doctor details.
- Test for the electronic sending of the prescription to the desired pharmacy.

g. Payment Processing:

- Validate successful transaction processes.
- Test for transaction failures due to various reasons like insufficient funds or network issues.

Tools: Tools like JUnit (for Java applications), NUnit (for .NET applications), and xUnit can be employed for unit testing.

Entry Criteria for Unit Testing:

1. **Requirement Specification Ready:** Clear and comprehensive requirement specification available for each component.
2. **Code Ready for Testing:** Code implementation complete and reviewed for glaring issues.
3. **Test Environment Set Up:** Dedicated testing environment with infrastructure, databases, and integrations ready.
4. **Unit Testing Framework Configured:** Appropriate unit testing framework set up.
5. **Test Cases Designed:** Detailed test cases with steps, outcomes, and criteria prepared.

Exit Criteria for Unit Testing:

1. **Test Cases Executed:** All test cases executed, with automation where possible.
2. **High Passing Rate:** Majority of test cases pass (ideally close to 100%).
3. **Defects Documented:** Identified defects documented, categorized, and prioritized.
4. **Code Quality Checked:** Codebase reviewed, errors fixed, coding standards ensured.
5. **Documentation Updated:** Detailed test case documentation maintained.
6. **Approval Obtained:** Test results and code quality improvements approved by stakeholders.

2. Security Testing:

Definition: Security testing involves identifying vulnerabilities in the software application and ensuring that its data and resources are protected from potential breaches.

Based on Requirement Specification:

a. Data Privacy and Security:

- Conduct penetration testing to identify any vulnerabilities in the system.
- Test the encryption techniques used to protect patient data during transit and at rest.
- Validate role-based access control for different user types (patient, doctor, admin).

b. User Registration and Login:

- Test for SQL injection vulnerabilities in the registration and login forms.
- Test for brute force attack vulnerabilities on login.

c. Video Conferencing:

- Ensure that video sessions are encrypted.
- Verify unauthorized users cannot access ongoing video consultations.

d. Payment Processing:

- Validate SSL/TLS encryption for transactions.
- Ensure secure storage of payment details with encryption.

e. Medical Records Management:

- Ensure unauthorized access to medical records is prevented.
- Check for vulnerabilities that might allow tampering with medical records.

Tools: Tools like OWASP ZAP, Burp Suite, and Nessus can be helpful for security testing.

3. **Integration Testing:**
    - Ensure different components of the application interact and work seamlessly together.
    - Tools: JUnit, TestNG.

4. **Performance Testing:**
   - Check the software's responsiveness, stability, and scalability under workload.
   - Tools: JMeter, LoadRunner.
5. **Usability Testing:**
   - Ensure the application is user-friendly and intuitive.
   - Methods include user surveys and feedback.
6. **Accessibility Testing:**
   - Ensure the application is usable by people with disabilities.
   - Tools: WAVE, AXE.
7. **Compatibility Testing:**
   - Check the software's performance across different devices, operating systems, and browsers.
   - Tools: BrowserStack, CrossBrowserTesting.
8. **Regression Testing:**
   - Ensure new features or updates haven't negatively impacted existing functionalities.
   - Tools: Selenium, TestComplete.

By adopting a comprehensive testing strategy based on the outlined requirement specifications, we can ensure the developed software is of high quality, secure, and provides a seamless user experience.

Entry Criteria for Security Testing:

1. **Requirement Specification Defined:** Detailed security requirements available.
2. **Functional Testing Done:** Basic functionality of the application tested and stable.
3. **Test Environment Prepared:** Dedicated testing environment set up, mirroring production.
4. **Security Tools Configured:** Relevant security testing tools (e.g., OWASP ZAP, Burp Suite, Nessus) configured.
5. **Test Data Prepared:** Test data, including attack vectors, ready for simulation.

6. **Access Credentials Available:** User role credentials (patient, doctor, admin) provided for testing.

Exit Criteria for Security Testing:

1. **Vulnerabilities Documented:** Detected vulnerabilities documented and categorized.
2. **Encryption Techniques Verified:** Validation of data encryption during transit and at rest.
3. **Access Control Tested:** Role-based access control mechanisms verified.
4. **Video Conferencing Secured:** Video sessions confirmed encrypted and unauthorized access blocked.
5. **Payment Processing Checked:** SSL/TLS encryption and secure payment data storage validated.
6. **Medical Records Secured:** Access controls and data integrity mechanisms tested.
7. **Defects Addressed:** Vulnerabilities resolved by development team.
8. **Documentation Updated:** Detailed documentation of security testing maintained.
9. **Approval Obtained:** Security testing outcomes approved by stakeholders before proceeding.

6.2 Test Tools

1. **Configuration Management**:
    - **Git/GitHub**: Will be used as the source version control tool, ensuring that multiple developers can work simultaneously without conflicts.
    - **Jenkins**: An open-source automation server that will help in automating parts of the development process.
2. **Prototyping**:
    - **Figma**: To design and prototype user interfaces for the platform. The initial layout, user flows, and content will be shown to stakeholders for feedback prior to actual development.
3. **Functional Testing**:
    - **Selenium**: An open-source tool that helps in automating web browsers. It will be used to automate the functional testing of the platform’s web interface.
    - **Appium**: For mobile app testing, automating native, mobile web, and hybrid applications on iOS and Android.
4. **Performance Testing**:

- **JMeter**: An open-source software designed to load test functional behavior and measure performance. It will test the system's behavior under expected loads.

5. **Security Testing**:
   - **OWASP ZAP (Zed Attack Proxy)**: An open-source security tool actively maintained by hundreds of international volunteers. It will be used to identify vulnerabilities in the platform.
6. **API Testing**:
   - **Postman**: To ensure that APIs developed for the platform function as expected.
7. **Database Testing**:
   - **SQLMap**: An open-source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws.
8. **Usability Testing**:
   - **Hotjar**: Will be used to collect user feedback and actions to understand how users interact with the platform and improve usability.

6.3 Meetings

Week	**Meeting Type**	**Agenda**	**Participants**
1	Test Team Meeting	Evaluate progress to date, identify error trends and potential issues.	Test Team Members
2	Test Team Leader, Development & Project Manager Meeting	Discuss testing progress, issues faced, and coordination between testing and development.	Test Team Leader, Development Team Lead, Project Manager
3	Test Team Meeting	Evaluate progress to date, identify error trends and potential issues.	Test Team Members
4	Test Team Leader, Development & Project Manager Meeting	Discuss testing progress, issues faced, and coordination between testing and development.	Test Team Leader, Development Team Lead, Project Manager
...	...	...	...
N	Emergency Meeting (as required)	Address specific emergency situations or critical issues identified during testing.	Relevant participants depending on the nature of the emergency (e.g., specific test team members, developers, etc.)

7. TEST CASES/TEST ITEMS

<table>
<tr><td colspan="3">Project Name: TeleHealthNow</td><td colspan="2">Test Designed by: Sadman</td></tr>
<tr><td colspan="3">Test Case ID: UR_01</td><td colspan="2">Test Designed date:12/08/23</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Sadman</td></tr>
<tr><td colspan="3">Module Name: Login Session</td><td colspan="2">Test Execution date: 14/08/23</td></tr>
<tr><td colspan="3">Test Title: verify login with valid username and password</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: Test app login page</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition (If any): User must have valid username and password</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Open app
2. Tap on ‘registration /create account’</td><td>Username: samsam221
Password: ie43@ljMp</td><td>User should be able to create his account with the valid credentials.</td><td>User account is created successfully.,</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: User is validated with database and successfully login to account. The account session details are saved in the database.</td></tr>
</table>

<table>
<tr><td colspan="3">Project Name: TeleHealthNow</td><td colspan="2">Test Designed by: Sadman</td></tr>
<tr><td colspan="3">Project Name: TeleHealthNow</td><td colspan="2">Test Designed by: Shakiba</td></tr>
<tr><td colspan="3">Test Case ID: UR_02</td><td colspan="2">Test Designed date:13/08/23</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): Medium</td><td colspan="2">Test Executed by: Sadia</td></tr>
<tr><td colspan="3">Module Name: Login Session</td><td colspan="2">Test Execution date: 14/08/23</td></tr>
<tr><td colspan="3">Test Title: catch invalid credentials while trying to login</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: Test login without valid credentials</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition (If any): App must be installed on device</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Open app
2. Tap on ‘registration /create account’</td><td>Username: sam
Password: 1234</td><td>An error message should come up " invalid information".</td><td>Error message appears</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: User is given another chance to login with correct credentials.</td></tr>
</table>

Project Name: TeleHealthNow	Test Designed by: Sadman
Project Name: TeleHealthNow	Test Designed by: Moinul
Test Case ID: SA_04	Test Designed date:17/08/23
Test Priority (Low, Medium, High): High	Test Executed by: Sadman
Module Name: Scheduling Appointments	Test Execution date: 18/08/23
Test Title: Test Appointment Scheduling Outside Physician's Availability	
Description: To check if patients will have conflicting schedules with doctors	

Precondition (If any): app is installed. Patient has account created. User is logged in.

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.go to ‘make appointments’ page. 2.input necessary information, select a schedule, select a doctor who has no available schedule. 3.click on ‘submit’	Doctor: Shamsu Availability: Unavailable Time: 24/07/23	a message should appear "no available schedule for this doctor"	a message appears "no available schedule for this doctor"	Pass

Post Condition: User’s available doctors page is refreshed with doctors who are available as it is updated in the server/database.

Project Name: TeleHealthNow	Test Designed by: Sadia
Project Name: TeleHealthNow	Test Designed by: Sadia
Test Case ID: VC_02	Test Designed date:27/08/23
Test Priority (Low, Medium, High): Low	Test Executed by: Sadman
Module Name: Video Conferencing	Test Execution date: 28/08/23
Test Title: Test Successful video conference initiation.	
Description: To check if patients can consult with doctor during their given time	
Precondition (If any): app is installed. Patient has account created. User is logged in and went to doctor's appointment in his/her slot	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.go to 'my appointments' page. 2.click on 'scheduled appointments' 3.click on 'start video'	Doctor: Kamrul Availability: Available Time: 28/08/23	A successful video call session should be established between the patient and the doctor.	Call failed	Fail

Post Condition: Patient and doctor are on video call

<table>
<tr><td colspan="3">Project Name: TeleHealthNow</td><td colspan="2">Test Designed by: Sadman</td></tr>
<tr><td colspan="3">Test Case ID: MR_01</td><td colspan="2">Test Designed date:2/09/23</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Jannat</td></tr>
<tr><td colspan="3">Module Name: Medical Records</td><td colspan="2">Test Execution date: 2/09/23</td></tr>
<tr><td colspan="3">Test Title: Test to upload medical records</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: To check if patients can store their medical records in our database</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition (If any): app is installed. Patient has account created. User is logged in</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. navigate to profile page.
2. tap on "edit your profile".
3. tap on "upload medical reports"
4.select necessary files from device.
5. click on ‘upload’</td><td>Files.jpg</td><td>User should be able to upload medical reports and other necessary documents within the given size</td><td>User is able to upload medical reports and other documents.</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: User’s profile is edited, and our database has the updated record</td></tr>
</table>

<table>
<tr><td colspan="3">Project Name: TeleHealthNow</td><td colspan="2">Test Designed by: Sadman</td></tr>
<tr><td colspan="3">Test Case ID: PD_03</td><td colspan="2">Test Designed date:4/09/23</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Sadman</td></tr>
<tr><td colspan="3">Module Name: Prescription Delivery</td><td colspan="2">Test Execution date: 8/09/23</td></tr>
<tr><td colspan="3">Test Title: Test send prescriptions electronically to patient</td><td colspan="2"></td></tr>
<tr><td colspan="3">Description: To check if the system can properly deliver right prescriptions to its desired targets</td><td colspan="2"></td></tr>
<tr><td colspan="5">Precondition (If any): both patient and the doctor had made a video conference for consultancy. A prescription is generated by the doctor according to the patient’s requirements.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. tap on "send prescription via email" button.
2. select prescription.
3. tap on "upload medical reports"
4.select necessary files from device.
5. click on ‘upload’</td><td>Prescription.pdf</td><td>Email is sent to patient</td><td>Email is sent to patient</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: Patient has his record which is also stored in server.</td></tr>
</table>

8. ITEM PASS/FAIL CRITERIA

1. Test Item for Login session, UR_01
Pass: User can login with valid credentials.
Fail: User cannot login with valid credentials.

2. Test Item for Login session, UR_02
Pass: User cannot login without valid credentials.
Fail: User can login without valid credentials.

3. Test Item for Scheduling Appointments, SA_04
Pass: User cannot book appointments during unavailable hours.
Fail: User can book appointments during unavailable hours.

4. Test Item for Video Conferencing (Consultancy), VC_02
Pass: User can contact their appointed doctor during consultation time via video call.
Fail: User cannot contact their appointed doctor during consultation time via video call.

5. Test Item for Medical Records Management, MR_01
Pass: User can upload files to our server.
Fail: Users are unable to upload their documents.

6.Test Item for Prescription Delivery, PD_03
Pass: Prescription via Email is sent to patient
Fail: No email is sent to patient despite clicking on the ‘send prescription; button.

Out of the test cases, only one failed due to flawed logic so initially. Success rate was 5/6 hence, 83%. After conducting further tests, success rate was 95% and 5% failure rate was seen.

9. TEST DELIVERABLES

1. **Test Strategy Document**
2. **Test Plan**
3. **Test Cases and Test Scripts**
4. **Test Data**
5. **Traceability Matrix**
6. **Defect Reports and Log**
7. **Test Execution Report**
8. **Test Environment Details**
9. **Performance Test Results**
10. **Security Test Results**
11. **Usability Test Feedback**

10. STAFFING AND TRAINING NEEDS

1. **Staffing Needs:**

- **Test Manager**: To oversee the testing strategy, lead the test team, and ensure alignment with project goals. Should have experience in healthcare software testing and project management.
- **Test Engineers**: Individuals responsible for writing and executing test cases. Should have a mix of junior and senior testers, with some possessing domain knowledge in healthcare.
- **Performance Testers**: Professionals specializing in performance, load, and stress testing. Their role is crucial given the real-time nature of the TeleHealthNow platform.
- **Security Testers**: Experts who specialize in identifying vulnerabilities and ensuring the software's safety, especially given the sensitivity of medical data.

- **Automation Engineers**: Professionals skilled in automating repetitive test cases to improve efficiency and ensure consistency.
- **Usability Testers**: Individuals who focus on the user interface and overall user experience.

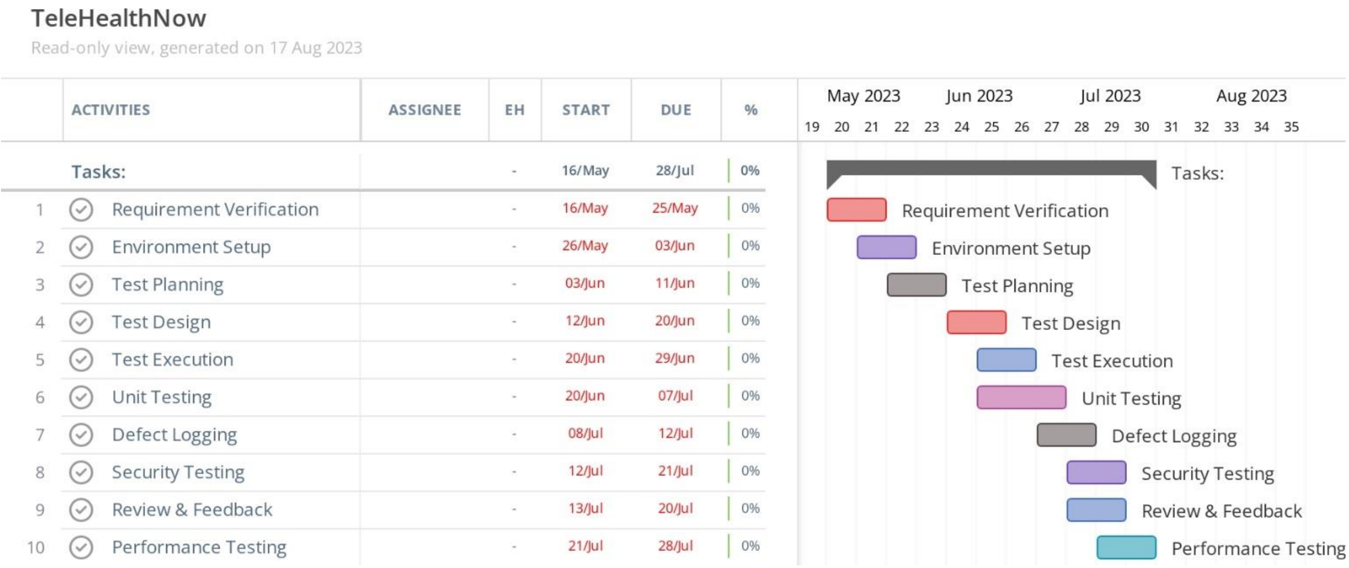
2. **Training Needs:**

- **Domain Training**: Given the specialized nature of the healthcare domain, training might be required for testers to understand medical terminologies, processes, and regulations.
- **Tool Training**: If the project uses specific testing tools or platforms that the team isn't familiar with, training sessions would be needed. This includes tools for test management, defect tracking, and automation.
- **Security and Compliance**: Given the importance of patient data, training in data security, encryption standards, and healthcare regulations (like HIPAA in the U.S.) would be vital.
- **Process Training**: If the project follows specific methodologies (e.g., Agile, Scrum) that some team members aren't acquainted with, introductory sessions or refresher courses might be necessary.
- **Product Orientation**: A comprehensive understanding of the TeleHealthNow platform's features, objectives, and user personas would be crucial for effective testing. Orientation sessions can bridge any knowledge gaps.
- **Continuous Learning**: Given the rapid evolution of both healthcare and IT, regular workshops on emerging trends, technologies, and best practices will keep the team updated and competitive.

11. RESPONSIBILITIES

Responsibilities	TM (Test Manager)	PM (Project Manager)	Dev Team	Test Team	Client
Define testing objectives	X	X			
Define project scope & deliverables		X	X		
Develop the software/app			X		
Design and write test cases	X			X	
Execute test cases				X	
Report defects and issues	X			X	
Prioritize and manage defect fixes	X	X	X		
Review and approve test deliverables	X	X			X
Oversee test automation	X			X	
Monitor test progress	X	X		X	
Provide resources & tools for testing		X			
Ensure compliance and security standards	X	X	X	X	
Accept or reject final product after testing					X
Communication with stakeholders	X	X	X	X	X
Training and orientation for the test team	X				
Continuous process and quality improvement	X	X	X	X	
Feedback and improvement suggestions					X

12. TESTING SCHEDULE



TeleHealthNow

Read-only view, generated on 17 Aug 2023

		ACTIVITIES	ASSIGNEE	EH	START	DUE	%
		Tasks:		-	16/May	28/Jul	0%
1	✓	Requirement Verification		-	16/May	25/May	0%
2	✓	Environment Setup		-	26/May	03/Jun	0%
3	✓	Test Planning		-	03/Jun	11/Jun	0%
4	✓	Test Design		-	12/Jun	20/Jun	0%
5	✓	Test Execution		-	20/Jun	29/Jun	0%
6	✓	Unit Testing		-	20/Jun	07/Jul	0%
7	✓	Defect Logging		-	08/Jul	12/Jul	0%
8	✓	Security Testing		-	12/Jul	21/Jul	0%
9	✓	Review & Feedback		-	13/Jul	20/Jul	0%
10	✓	Performance Testing		-	21/Jul	28/Jul	0%

13. PLANNING RISKS AND CONTINGENCIES

1. Risk: Delays in Development Schedule

- **Contingency**: Allocate buffer time in the schedule for unforeseen delays. If the project does fall behind, consider reallocating resources or prioritizing critical features for the initial release.

2. Risk: Budget Overruns

- **Contingency**: Maintain a contingency fund in the budget. Regularly review project expenditures and adjust as necessary. Prioritize essential features if the budget becomes constrained.

3. Risk: Lack of Knowledge or Expertise in Telemedicine Domain

- **Contingency**: Conduct training sessions or bring in domain experts. Consider partnering with consultants or firms that specialize in telemedicine applications.

4. Risk: Security Vulnerabilities

- **Contingency**: Prioritize regular security audits. Invest in top-tier security tools and practices from the outset. Respond promptly to any identified vulnerabilities.

5. Risk: Inadequate Test Coverage Leading to Undetected Bugs

- **Contingency**: Implement automated testing to increase coverage. Conduct regular code and test reviews. Consider beta testing with a select group of users before full release.

6. Risk: Regulatory Compliance Issues

- **Contingency**: Consult with experts familiar with healthcare regulations in the intended market. Ensure that the application adheres to regional data protection and healthcare laws.

7. Risk: Low Adoption Rates Post-Launch

- **Contingency**: Develop a robust marketing and outreach plan. Consider offering introductory promotions or partnerships with healthcare providers to increase initial adoption.

8. Risk: Technology Stack or Platform Limitations

- **Contingency**: Conduct thorough research and feasibility testing before finalizing technology decisions. Ensure flexibility to pivot or adapt as needed.

9. Risk: Key Personnel Turnover

- **Contingency**: Cross-train team members to ensure multiple individuals are familiar with key project components. Maintain comprehensive documentation to ease the onboarding of new team members.

10. Risk: Data Loss or Corruption

- **Contingency**: Implement robust backup and recovery systems. Conduct regular drills to ensure data can be restored quickly in case of any failures.

11. Risk: Miscommunication or Misalignment Among Teams

- **Contingency**: Facilitate regular communication through meetings, status updates, and collaborative tools. Ensure clear documentation of requirements and decisions.

14. APROVALS

Project Sponsor	S.M Moinul Islam
Development Management	Sadia Hossain Bushra
EDI Project Manager	S.M Moinul Islam
RS Test Manager	I S M Salim Sadman
RS Development Team Manager	Shihabun Sakiba Jannat
Reassigned Sales	Sadia Hossain Bushra
Order Entry EDI Team Manager	I S M Salim Sadman