American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Skin Disease Identifier Application

A Software Quality Testing Project Submitted
By

Semester: Spring_22_23		**Section: I**	**Group Number: 5**	
SN	Student Name	Student ID	Contribution (CO1+CO2)	Individual Marks
02	AKIB TAHMID	19-39309-1		
08	MD AMIR HAMZA	19-41219-2		
09	TAHIRA MIZAN TANHA	19-41282-3		
32	SADIA ISLAM SHAFINA	20-43539-1		
34	NAYEEM ABDUL QAIYUM	20-43581-1		

Under the supervision of
Dr. M. Mahmudul Hasan
Assistant Professor

Software Test Plan

for

< Skin Disease Identifier Application>

Version 1.0 approved

Prepared by <Group-5>

<AIUB>

<25.03.23>

Table of Contents

Revision History......3
1. TEST PLAN IDENTIFIER: SD-NS-1.3 4
2. REFERENCES......4
3. INTRODUCTION......4
3.1 Background to the Problem...... 4
3.2 Solution to the Problem...... 4
4. REQUEIREMNT SPECIFICATION......5
4.1 System Features 5
4.2 System Quality Attributes...... 9
4.3 System Interface...... 11
4.4 Project Requirements 12
5. FEATURES NOT TO BE TESTED......12
6. TESTING APPROACH......12
6.1 Testing Levels...... 12
6.2 Test Tools...... 13
6.3 Meetings...... 14
7. TEST CASES/TEST ITEMS......14
8. ITEM PASS/FAIL CRITERIA......23
9. TEST DELIVERABLES......23
10. STAFFING AND TRAINING NEEDS......23
11. RESPONSIBILITIES......23
12. TESTING SCHEDULE......25
13. PLANNING RISKS AND CONTINGENCIES......25
14. APROVALS......26

Revision History

Revision	Date	Updated by	Update Comments
0.1	2023.03.25	Nayeem Abdul Qaiyum	First Draft
0.2	2023.03.26	Akib Tahmid	Second Draft
0.3	2023.04.15	Akib Tahmid	Third Draft
0.4	2023.04.21	TAHIRA MIZAN TANHA	Fourth Draft
0.5	2023.04.24	Nayeem Abdul Qaiyum	Fifth Draft
0.6	2023.04.26	Sadia Islam Shafina	Sixth Draft
0.7	2023.04.26	Nayeem Abdul Qaiyum	Seventh Draft
0.8	2023.04.28	MD AMIR HAMZA	Eighth Draft

1. TEST PLAN IDENTIFIER: SD-NS-1.3

2. REFERENCES

- Software Quality and Testing Course Materials
- Software Testing and Quality Assurance – Theory and Practice - Kshirasagar Naik & Priyadarshi Tripathy
- Skin Disease Website
  - (www.tibot.ai)
  - https://www.niams.nih.gov/health-topics/skin-diseases
  - https://my.clevelandclinic.org/health/diseases/21573-skin-diseases

3. INTRODUCTION

3.1 (Background to the Problem)

Skin conditions negatively affect people's life and health. Skin disease is a significant issue in our nation. But the majority of people don't give it any thought. And there's a reason why people don't visit doctors when they have skin issues. They believe it to be a minor thing in their day-to-day lives. Also, the villagers are unaware of the symptoms of skin diseases and the proper skin care regimen. The most common skin conditions in our nation are psoriasis, herpes, acne, eczema, and sunburn. Yet, most individuals are not fully informed about these conditions. Thus, this skin condition worsens every day. If skin disease is not treated when it first appears, it can also be a serious issue.

3.2 (Solution to the Problem)

As the world everyday moves closer to the peak of technological development, our country is doing its best to keep up with the rest of the world by digitizing its conventional processes, and people are more interested in finding online-based answers to their problems than in finding offline ones. Here, we want to create a platform where they(user) can have the opportunity to find someone suitable whom they can reach out to anytime with any problem. A user will have the option to call a doctor to ask questions regarding his/her specific skin problem and our doctor will get back to him/her with the solution. If anyone does not like to go with the available option, they may also interact with a doctor regarding their problem conducting the live consulting session. Having a 24/7-staffed helpdesk or chat service can also help increase engagement and drive user satisfaction with this application.

4. REQUEIREMNT SPECIFICATION

❖ System Features

4.1.1 Home

Functional Requirements:

- On home page all link can be check.
- To access a link the software shall allow users to log in first.
- By clicking on link can go in other page

Priority Level: Medium

Precondition: N/A

Cross Reference: Uploading Image.

4.1.2 Uploading Image

Functional Requirements:

- Users will be able to add pictures of the affected area to the website.
- The application must support PNG, JPG, and JPEG image formats.

Priority Level: Medium

Pre-Condition: User has access to an internet-connected device.

Cross Reference: Home, Skin Disease Identification.

4.1.3 Skin Disease Identification

Functional Requirements:

- The program must analyze the uploaded image to determine the skin condition.
- The program must evaluate the image using machine learning methods and deliver an accurate diagnostic.

Priority Level: High

Pre-Condition: The user has to upload a appropriate picture of the affected area.

Cross Reference: Uploading Image, User Login Session

4.1.4 User login Session

Functional Requirements:

- The website shall allow users to login with their given phone number, email and password.
- If the phone number, email and/or password has been inserted wrong the user can reset the password with.
- From the help of his email, a random verification code will be generated by the system to retry login.

Priority Level: High.

Precondition: User has valid user id and password.

Cross Reference: Home, Uploading Image, Skin Disease Identification, User Login Session, User Login Validation Session, User Registration Session, Edit Profile, Change Password, Forgot Password.

4.1.5 User Login Validation Session

Functional Requirements:

- Enter valid Phone Number and Email.
- Enter valid Password.
- If the phone number, email and password is wrong wont login the site.

Priority Level: High.

Precondition: User has valid user id and password.

Cross Reference: User Login Session, User Login Validation Session, Edit Profile, Change Password, Forgot Password.

4.1.6 User Registration Session

Functional Requirements:

- User has to click on the “Sign Up” option
- If user is new member, he can connect using Facebook/Email.
- After clicking on create an account, user can request to enter your phone number and slide to the send and wait to receive the verification code
- Then after checking phone or mail and enter the verification code
- User has to enter password
- Then after entering information registration will be successful

Priority Level: High.

Precondition: User must have valid phone number.

Cross Reference: User Login Session, User Login Validation Session.

4.1.7 Treatment Recommendation

Functional Requirements:

- Depending on the diagnosis, the application should show users the best ways to treat the problem.
- The application will tell users about each recommended treatment option, including how well it works and what side effects it might cause.

Priority Level: High

Precondition: The application has to figured out what was wrong with the user's skin.

Cross Reference: Skin Disease Identification.

4.1.8 Feedback Submission

Functional Requirements:

- The application shall allow users to submit feedback regarding the accuracy of the diagnosis and the efficacy of the suggested treatment.
- Users will be able to rate their application experience.

- Users can write additional description

Priority Level: Medium

Precondition: User must have received a skin disease diagnosis.

Cross Reference: Skin Disease identification, Treatment Recommendation.

4.1.9 Forgot password

Functional Requirements:

- User has to go to forgot password
- Enter email address or phone number
- After verifying everything user can change new Password

Priority Level: High

Cross Reference: User Login Session, User Login Validation Session

4.1.10 Edit profile

Functional Requirements:

- After logging in user can edit profile
- User can Manage account by clicking to my profile
- Then user can update profile and save changes

Priority Level: High.

Precondition: User must be logged in.

Cross Reference: User Login Session, User Login Validation Session.

4.1.11 Change password

Functional Requirements:

- User can change password by logging into account
- User has to go to manage account and click change password
- By entering new password user can change but for that user has to provide old one.

Priority Level: High.

Precondition: User must be logged in.

Cross Reference: User Login Session, User Login Validation Session.

4.1.12 Language Support

Functional Requirements:

- The application must support multiple languages, which includes Bangla and English.
- The user can change the language through the settings menu.

Priority Level: Medium

Precondition: N/A

Cross Reference: N/A

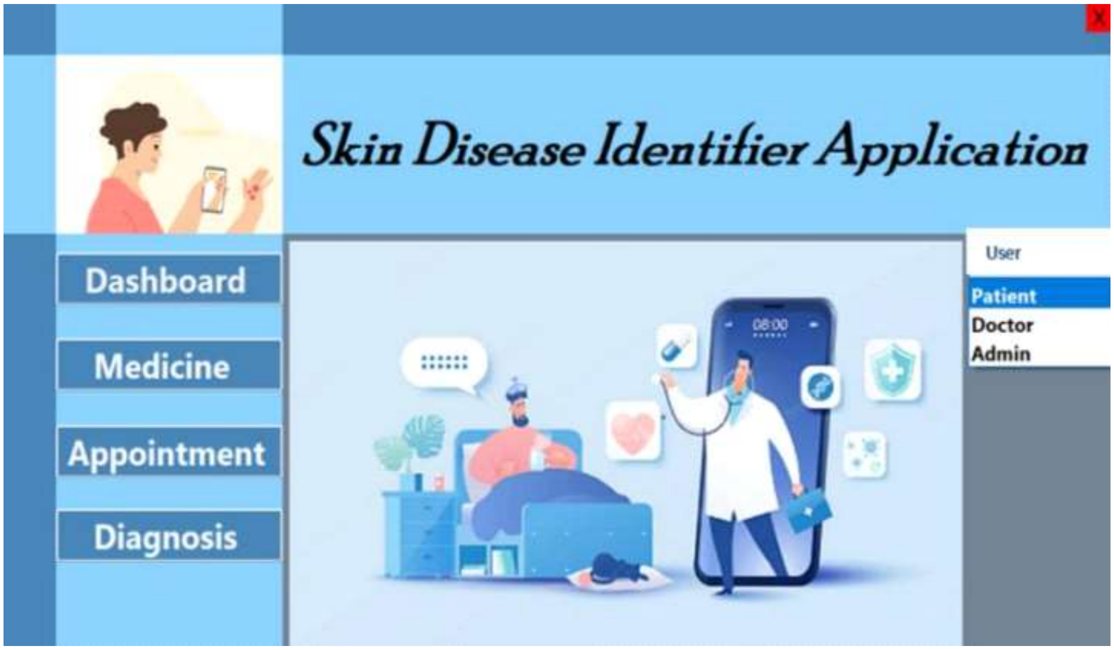
❖ System Quality Attributes

- **Reliability:**
  - ➢ Mean Time Between Failures: 300 hours (The system should run for an average of 300 hours between failures)
  - ➢ Failure rate: 5% (No more than 5% of tests should fail due to software errors)
- **Performance:**
  - ➢ Response time: 5 seconds (System should respond to user requests within 5 seconds)
  - ➢ Throughput: 50 diagnoses per hour (System should be able to handle up to 50 diagnoses per hour)
- **Scalability:**
  - ➢ Concurrent user capacity: 100 users (System should be able to handle up to 100 concurrent users)
  - ➢ Vertical scaling factor: 2 (System's capacity should double when additional resources are added)

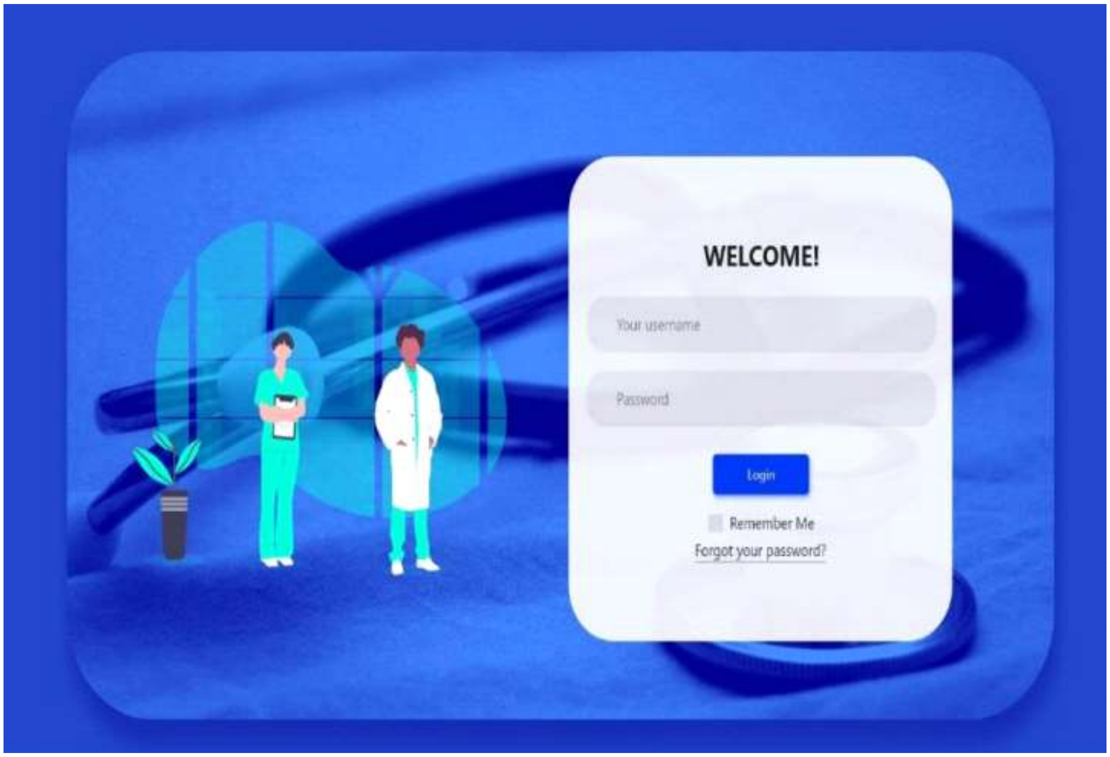
- **Security:**
  - Vulnerability count: 0 (Must not have any security vulnerabilities in the system)
  - Encryption strength: 256-bit (System should use 256-bit encryption to protect user information)
- **Maintainability:**
  - Code complexity: 50 (Codebase should have a complexity score of no more than 50)
  - Code coverage: 70% (At least 70% of the code should be covered by automated tests)
- **Usability:**
  - User satisfaction: 90% (At least 90% of users should rate the application as user-friendly and easy to use)
  - Task completion time: 2 minutes (Users should be able to complete common tasks within 2 minutes)
- **Availability:**
  - Downtime: 1 hour per month (System should be unavailable for no more than 1 hour per month)
  - Mean Time to Repair: 30 minutes (System should recover from any outages within 30 minutes)
- **Compatibility:**
  - Browser compatibility: Chrome, Firefox, Safari, and Edge (Application should be compatible with these browsers)
  - Operating system compatibility: Windows, MacOS, and Linux (The application should be compatible with these operating systems)
- **Testability:**
  - Test coverage: 75% (At least 75% of the requirements should be covered by automated tests)

- **System Interface**

System Feature:



User Login Page:



❖ Project Requirements

In this project we will use Agile Software Development Methodology. Agile development is ideal for the Skin Disease Identifier app since it allows for flexibility and adaptability. Agile allows the development team to quickly adapt to changing application requirements. Agile technique also allows stakeholders and end-users to provide constant feedback, ensuring the final product satisfies their demands.

- Duration: 6 months (Estimated). To track progress and finish on time, break this timetable into smaller tasks or milestones.
- Budget: 5 Lac BDT (Estimated)
- Team: At least 5 members. All members should have proper technical knowledge.
- Dedicated IDE (Integrated Development Environment) for development.
- Dedicated QA (Quality Assurance) team for testing.
- A space for collaboration.
- Proper Project Management to track project progress, activities, manage risks and communicate with stakeholders.
- Proper Documentation should be maintained throughout the project's lifecycle.

5. FEATURES NOT TO BE TESTED

The feature which are not to be tested are as follows:

- The compatibility of hardware from third parties.
- Validating user input or handling errors that go beyond basic needs.
- Integration with other healthcare systems or electronic medical records.
- Types of users.
- Maintenance of servers.
- Maintenance of user records.

6. TESTING APPROAC

❖ Testing Levels

In this project, we will implement a test plan for the Skin Disease Identifier Application. To continue testing software in relation to the project, we need to go through major stages of testing. Several test levels need to be addressed.

- **Unit Testing:** Unit testing is the procedure of verifying the functionality of individual software units or components. Unit testing for the Skin Disease Identifier application could entail testing individual software functions or modules. Unit testing can be implemented using frameworks and tools for automated testing such as JUnit, NUnit, and TestNG. Unit tests should be created as early as possible in the development process in order to detect bugs as soon as possible.

- **Integration Testing:** Integration testing is the process of evaluating how the various software components interact. Integration testing for the Skin Disease Identifier application could entail evaluating the interaction between the user interface, the algorithm that analyzes skin images, and the database that stores patient information. Integration testing can be conducted manually or with automated testing tools like Selenium or Appium.

- **System Testing:** System testing is the process of evaluating the software as a whole to ensure that it conforms to all requirements and specifications. System testing could involve evaluating the Skin Disease Identifier application's functionality, performance, security, and usability. Depending on the test cases, system testing can be performed manually or using automated testing instruments.

- **User Acceptance Testing:** User Acceptance Testing is the process of testing software from the perspective of the user to ensure that it meets their requirements and expectations. It could include testing the Skin Disease Identifier application's usability, simplicity of use, and accuracy in identifying skin diseases. A representative group of end-users who are not part of the development team should conduct user acceptance testing.

❖ Test Tools

For a proper bug free Skin Disease Identifier Application, we need to test the system and look for the bugs in the system. We need to test the system based on functional and non-functional requirements. There is always a need of manual testing besides automated testing. Automated testing is effective for proper bug tracking which is quite difficult for manual testing. For that reason, in our SDIA (Skin Disease Identifier Application) testing we will be using these testing tools:

- **Selenium (UI Testing):** Selenium is a free tool that can be used to test web apps by automating web browsers. By automating the browser actions, it can be used to test the user interface of the Skin Disease Identifier Application.

- **JMeter (Load Testing):** JMeter is a testing tool that can be used to test the application under load and measure how well it works. This tool can be used to simulate multiple users trying to access the Skin Disease Identifier Application at the same time so that the response time, throughput, and other performance metrics of the application can be measured.

- **Postman (API Testing):** Postman is a testing tool for APIs that can be used to test the RESTful APIs of the Skin Disease Identifier Application. It can be used to send requests to APIs and check the answers.
- **Apache Test Tool (Integration Testing):** The Apache Test Tool is a command-line program that can be used to test HTTP servers and applications. It can be used to test the Skin Disease Identifier Application's back-end functionality by sending HTTP requests and checking the responses.
- **Watir (User Acceptance Testing):** Watir is a toolkit for automating browser-based tests as part of the UAT process. Watir uses Ruby programming language to write test scripts and it can be easily integrated with continuous integration tools. It also works with different browsers like Chrome, Firefox, Edge, and Safari, and it has headless testing modes to make testing go faster.

❖ Meetings

Every two weeks, there should be a meeting so that the project can be done faster and on time. In project evaluation, a testing team will get together to look at the project's progress and keep it up to date, checking for bug trends and problems as soon as possible. Also, the leader of the test team meets every week with the development team and the project manager.

7. TEST CASES/TEST ITEMS

Test Case: 1

Project Name: Skin disease identifier application	Test Designed by: Group - 5
Test Case ID: SD_1	Test Designed date: 23.04.23
Test Priority (Low, Medium, High): Medium	Test Executed by: Sadia
Module Name: Home Page Skin Disease Identification	Test Execution date: 24.04.23
Test Title: Home Page	
Description: Access links, log in, and upload image for skin disease diagnosis.	
Precondition (If any): The user must have a valid account and upload a skin image file.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Verify that all links on the home page are functional. 2. Clicking on a link can take users to another page.	User have a valid account and link access.	All links on the home page are functional and accessible. Logging in is required to access features that require authentication.	As expected,	Pass
Post Condition: The user receives a diagnosis or recommendation for an uploaded image.				

Test Case: 2

Project Name: Skin disease identifier application	Test Designed by: Group - 5
Test Case ID: SD_2	Test Designed date: 23.04.23
Test Priority (Low, Medium, High): High	Test Executed by: Sadia
Module Name: Verify the functionality of uploading images for skin disease identification.	Test Execution date: 24.04.23
Test Title: Uploading image	
Description: User can upload image for skin disease identification.	
Precondition (If any): User has to access to internet-connected device to identify skin disease.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
Click on the "Upload Image" link to go to the image upload page. 1. Verify the file is in a supported format and not too large.	The application must support multiple image formats.	The software accepts the uploaded image file, processes it, and displays a diagnosis or recommendation based on the image.	As expected,	Pass
Post Condition: The user receives a diagnosis or recommendation for the uploaded image.				

Test Case: 3

<table>
<tr><td colspan="2">Project Name: Skin disease identifier application</td><td colspan="3">Test Designed by: Group - 5</td></tr>
<tr><td colspan="2">Test Case ID: SD_3</td><td colspan="3">Test Designed date: 23.04.23</td></tr>
<tr><td colspan="2">Test Priority (Low, Medium, High): High</td><td colspan="3">Test Executed by: Sadia</td></tr>
<tr><td colspan="2">Module Name: Check the skin disease identification function's accuracy.</td><td colspan="3">Test Execution date: 24.04.23</td></tr>
<tr><td colspan="2">Test Title: skin disease identification.</td><td colspan="3"></td></tr>
<tr><td colspan="2">Description: This test case verifies that the skin disease identification function can accurately analyze an uploaded image of the affected area and provide an accurate diagnostic using machine learning methods.</td><td colspan="3"></td></tr>
<tr><td colspan="5">Precondition (If any): The user must log in to the skin disease identification website and has to upload an appropriate picture of the affected area.</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. Upload an appropriate picture of the affected area using the "Upload Image" function.
2. Verify that the image has been successfully uploaded and processed.
3. Verify that the software provides accurate diagnosis of skin condition.</td><td>A skin image file in PNG, JPG, or JPEG format</td><td>The software provides an exact diagnosis of the skin condition after accurately analyzing the submitted photograph.</td><td>As expected,</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: The user receives an accurate diagnostic for the uploaded image.</td></tr>
</table>

Test Case: 4

Project Name: Skin disease identifier application	Test Designed by: Group - 5
Test Case ID: SD_4	Test Designed date: 20.04.23
Test Priority (Low, Medium, High): High	Test Executed by: Nayeem
Module Name: Login Session	Test Execution date: 20.04.23

Test Title: verify login with valid number or email and password				
Description: Test Application login system				
Precondition (If any): User must have valid number or email and password				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the login page 2. Enter number or email 3. Enter password 4. Click submit	Number: 01********* Password: 333421	User login successfully using valid number or Email and password	As expected,	Pass
Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database.				

Test Case: 5

Project Name: Skin disease identifier application			Test Designed by: Group - 5	
Test Case ID: SD_5			Test Designed date: 23.04.23	
Test Priority (Low, Medium, High): High			Test Executed by: Sadia	
Module Name: The User Login Validation Session for skin disease identification.			Test Execution date: 24.04.23	
Test Title: User login validation session.				
Description: Verify that a user can log in successfully with a valid phone number, email, and password.				
Precondition (If any): User has valid user id and password.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the login page of the website. 2. Enter a valid phone number and email in the corresponding fields. 3. Enter a valid password in the password field. 4. Click on the "Login" button.	Phone number: 1234567890 Email: test@test.com Password: test@123	The system should allow the user to log in and redirect them to the dashboard page.	As expected,	Pass
Post Condition: User is logged in and redirected.				

Test Case: 6

Project Name: Skin disease identifier application	Test Designed by: Group - 5
Test Case ID: SD_6	Test Designed date: 23.04.23
Test Priority (Low, Medium, High): High	Test Executed by: Nayeem
Module Name: User registration session for skin disease identification.	Test Execution date: 20.04.23
Test Title: User registration session.	
Description: Verify that a new user can register successfully with a valid phone number, email, and password.	
Precondition (If any): User must have a valid phone number	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Click on the "Sign Up" option. 2. Choose to connect using either Facebook or Email. 3. Enter a valid phone number and click on "Send". 4. Wait to receive the verification code and enter it and enter a password and confirm it. 5. Click on "Create an Account".	Phone number: 1234567890 Verification code: 1234 Password: test@123	The system should register the new user and redirect them to the dashboard page.	As expected,	Pass
Post Condition: User is registered and redirected.				

Test Case: 7

Project Name: Skin disease identifier application	Test Designed by: Group - 5
Test Case ID: SD_7	Test Designed date: 27.04.23
Test Priority (Low, Medium, High): High	Test Executed by: Sadia
Module Name: Treatment recommendation for skin disease identification.	Test Execution date: 28.04.23

Test Title: Treatment recommendation.				
Description: Verify that the application shows the best ways to treat a skin disease based on diagnosis.				
Precondition (If any): The application has identified the skin disease				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Select the identified skin disease. 2. View the recommended treatment options for the skin disease.	Skin disease diagnosis, treatment options	The application displays accurate information about the recommended treatment options for the identified skin disease.	As expected,	Pass
Post Condition: The application shows the recommended treatment options for the identified skin disease				

Test Case: 8

Project Name: Skin disease identifier application			Test Designed by: Group - 5	
Test Case ID: SD_8			Test Designed date: 23.04.23	
Test Priority (Low, Medium, High): Medium			Test Executed by: Sadia	
Module Name: Feedback submission for skin disease identification.			Test Execution date: 24.04.23	
Test Title: Feedback submission.				
Description: Feedback submission feature is tested to ensure accuracy, efficacy, and experience.				
Precondition (If any): 1. User must have received a skin disease diagnosis. 2. User must have access to the feedback submission feature in the application.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in to Skin Disease Identification application. 2. Provide feedback on diagnosis and treatment effectiveness. 3. use feedback to rate the application	The diagnosis and treatment was accurate and effective.	Feedback submission feature is accessible and Feedback is submitted successfully.	As expected,	Pass

experience and gain more insight.				
Post Condition: Feedback is submitted successfully.				

Test Case: 9

Project Name: Skin disease identifier application	Test Designed by: Group - 5
Test Case ID: SD_9	Test Designed date: 23.04.23
Test Priority (Low, Medium, High): High.	Test Executed by: Nayeem
Module Name: Verify the functionality of the forgot password feature in the Skin Disease Identification application.	Test Execution date: 20.04.23
Test Title: Forgot Password;	
Description: Test case checks if forgot password feature is functioning properly.	
Precondition (If any): User must have an account in the Skin Disease Identification application.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Launch Skin Disease Identification application and click "Forgot Password" option. 2. Reset password by entering email or phone number. 3. Confirm the new password and log in to the application.	new password must be Valid password and confirmation	The Skin Disease Identification application launch successfully, the user redirected to the Forgot Password page, and the user able to enter a new password and log in.	As expected,	Pass
Post Condition: User is able to reset their password successfully.				

Test Case: 10

Project Name: Skin disease identifier application	Test Designed by: Group - 5

Test Case ID: SD_10			Test Designed date: 26.04.23	
Test Priority (Low, Medium, High): High			Test Executed by: Sadia	
Module Name: Verify the functionality of the Edit Profile feature in the Skin Disease Identification application.			Test Execution date: 28.04.23	
Test Title: Edit profile.				
Description: User change given profile information.				
Precondition (If any): User must be logged in to the Skin Disease Identification application.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to home page 2. Go to login 3. Click on the "My Profile" option. 4. Click on the "Edit Profile" button. 5. Click on the "Save Changes" button	Valid login credentials. Updated profile information.	Save all changing information and show in profile	As expected,	Pass
Post Condition: User profile is updated successfully.				

Test Case: 11

Project Name: Skin disease identifier application			Test Designed by: Group - 5	
Test Case ID: SD_11			Test Designed date: 20.04.23	
Test Priority (Low, Medium, High): High			Test Executed by: Nayeem	
Module Name: Verify the functionality of changing password in the Skin Disease Identification application.			Test Execution date: 20.04.23	
Test Title: Change password.				
Description:. User can change password in skin disease identification application.				
Precondition (If any): User must be logged in to the Skin Disease Identification application.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)

1. Login the application. 2. Click 'Manage Account' option. 3. Click "Change Password" button 4. Enter the old password and new password in the respective fields. 5. Click "Save Change" button.	Valid login credentials Old password and new password for the user account	Save all updated information and display it on your profile.	As expected,	Pass
Post Condition: User is able to change their password successfully.				

Test Case: 12

Project Name: Skin disease identifier application	Test Designed by: Group - 5
Test Case ID: SD_12	Test Designed date: 23.04.23
Test Priority (Low, Medium, High): Medium	Test Executed by: Sadia
Module Name: Language support test case for Skin Disease Identification	Test Execution date: 24.04.23
Test Title: Language support.	
Description: Language support ensures system can detect and diagnose skin diseases in both languages.	
Precondition (If any): N/A	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the Settings menu and language option. 2. Select any option (Bangla, English).	At a time select one option.	. Application can accurately detect and diagnose skin disease in both languages.	As expected,	Pass
Post Condition: The application supports multiple languages and allows users to change the language.				

8. ITEM PASS/FAIL CRITERIA

Here, the test step can be considered successful if the test phase was finished successfully and the expected outcome is 95% functional with no significant faults. A test step must be considered failed and the requirement not verified if it could not be finished, could not be completed but the expected outcome did not occur, or if the expected result could not be validated from the record. All of the paths need to be thoroughly examined for the software to pass testing.

9. TEST DELIVERABLES

- ✓ Unit Test Plan
- ✓ Test logs and turnover reports
- ✓ Acceptance Test Plan
- ✓ System Integration Test Plan
- ✓ Incident reports and summaries
- ✓ Report mock-ups

10. STAFFING AND TRAINING NEEDS

The importance of skilled personnel and teams in the construction and delivery of a project cannot be overstated. A team of skilled professionals is required to perform this job properly. Because the project can be finished on budget and on schedule with skilled workers and staff; otherwise, a lack of skilled workers can make it difficult to complete the project on time. As a result, it should be thoroughly tested to ensure that the personnel and workers are qualified for the project. To develop and ensure that various skill-based training sessions can be scheduled. Project manager, senior test engineer (test lead), junior test engineer, testing manager, database analyst and other professionals make up the team. If a senior test engineer is unavailable for a period of time, the project manager can take over or replace the junior engineer with someone who has the necessary abilities.

11. RESPONSIBILITIES

Project Manager	1.The Test Controller should monitor the progress of the test on a frequent basis.

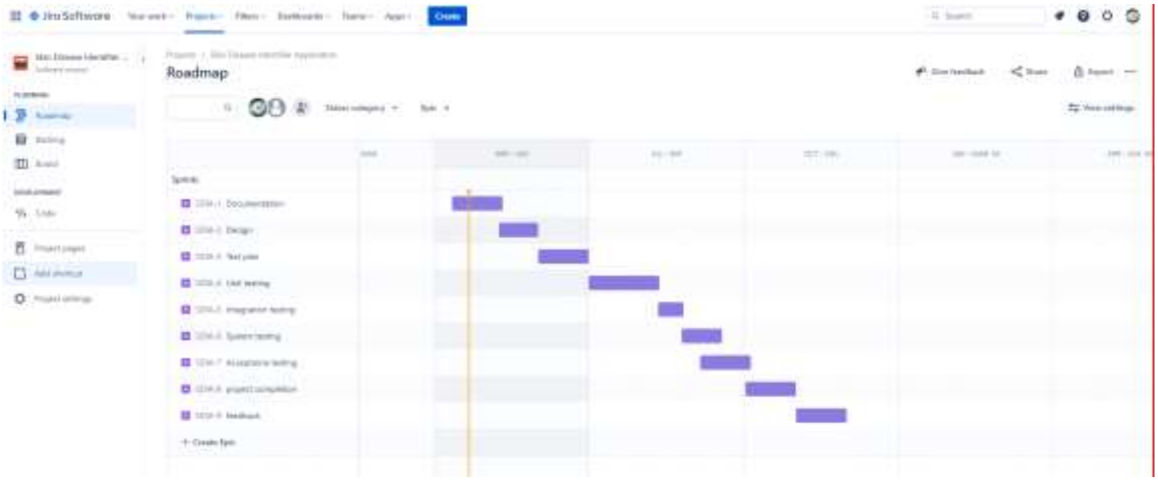
	2.Manage risks that are related to the project or that are not under the control of the Test Team.
	3.Review the methodology, plans, and timetable for the test.
Development Team	1.Conduct unit testing.
	2.Fix errors and make sure the system is usable.
	3.Assist the test team in ensuring quality control
Test Planner	1.Create detailed and high-level test conditions.
	2.Achieve the desired outcomes.
	3.At regular status reporting meetings, update on progress.
	4.Ensure that any outages or difficulties with the test systems are reported and followed up on as soon as possible.

Tester	1.Create a test data set and run it.
	2.Examine the conditions and make a note of the results
	3.Resolve backup-related queries.

End User	1.Assists with acceptability testing.
	2.Make sure the system is easy to use and performs well.

12. 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. The persons required for each process are detailed in the project timeline and plan as well. 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.



13. PLANNING RISKS AND CONTINGENCIES

The success of a software development project can be harmed by a number of risks. As a result, we must be prepared in every manner to ensure that the project is not affected by any risks. Due to schedule associated risk, time related risk is formed. If the schedule is not completed correctly, there may be issues with project timing, and the project may take longer to prepare. The project will not be completed on time, and the end-user will be dissatisfied. Costing-related risk can be caused by incorrect cost estimation. The majority of the costing risk stems from a scheduling issue. The longer it takes to prepare a project, the higher the cost will be. When a project's quality falls short of stakeholder expectations, there's a good chance it won't succeed. Technical debt can be caused by low-quality code. Other developers may find it difficult to examine or alter the code since it is tough to read. It could have been hurried and delivered without proper testing, resulting in a large number of bugs that could have been avoided. Because you rely heavily on their comments, it's critical to have an open, quick, and informative conversation with your stakeholders. If a client or stakeholder with whom you are working does not engage as much as they should with you or your team, your workflow and efficiency will suffer significantly.

14. APROVALS

Project Manager	NAYEEM ABDUL QAIYUM
Developer	AKIB TAHMID
Test Planner	SADIA ISLAM SHAFINA
Test Lead	AKIB TAHMID MD AMIR HAMZA
Tester	NAYEEM ABDUL QAIYUM SADIA ISLAM SHAFINA
End User	TAHIRA MIZAN TANHA MD AMIR HAMZA