**Online Matrimony**

A Software Quality And Testing Project Submitted

By

|  |  |  |
| --- | --- | --- |
| **Sunjida Afrin** | **Section: B** | **ID: 2014751149** |

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] |  |

Table of Contents

[1. TEST PLAN IDENTIFIER: AT-TP01.3 4](#_Toc126659481)

[2. REFERENCES 4](#_Toc126659482)-9

[3. INTRODUCTION](#_Toc126659483) 9-10

[3.1 Background to the Problem 9](#_Toc126659484)

[3.2 Solution to the Problem](#_Toc126659485) 10

[4. REQUEIREMNT SPECIFICATION](#_Toc126659486) 11-21

[4.1 System Features 11-1](#_Toc126659487)4

[4.2 System Quality Attributes 14-1](#_Toc126659488)5

[4.3 System Interface 16-2](#_Toc126659489)0

[4.4 Project Requirements 21](#_Toc126659490)

[5. FEATURES NOT TO BE TESTED 2](#_Toc126659491)1

[6. TESTING APPROACH 22-2](#_Toc126659492)5

[6.1 Testing Levels 22-2](#_Toc126659493)4

[6.2 Test Tools 24](#_Toc126659494)

[6.3 Meetings 24-2](#_Toc126659495)5

[7. TEST CASES/TEST ITEMS 26-3](#_Toc126659496)6

[8. ITEM PASS/FAIL CRITERIA 36](#_Toc126659497)-39

[9. TEST DELIVERABLES 39](#_Toc126659498)

[10. STAFFING AND TRAINING NEEDS 40-4](#_Toc126659499)1

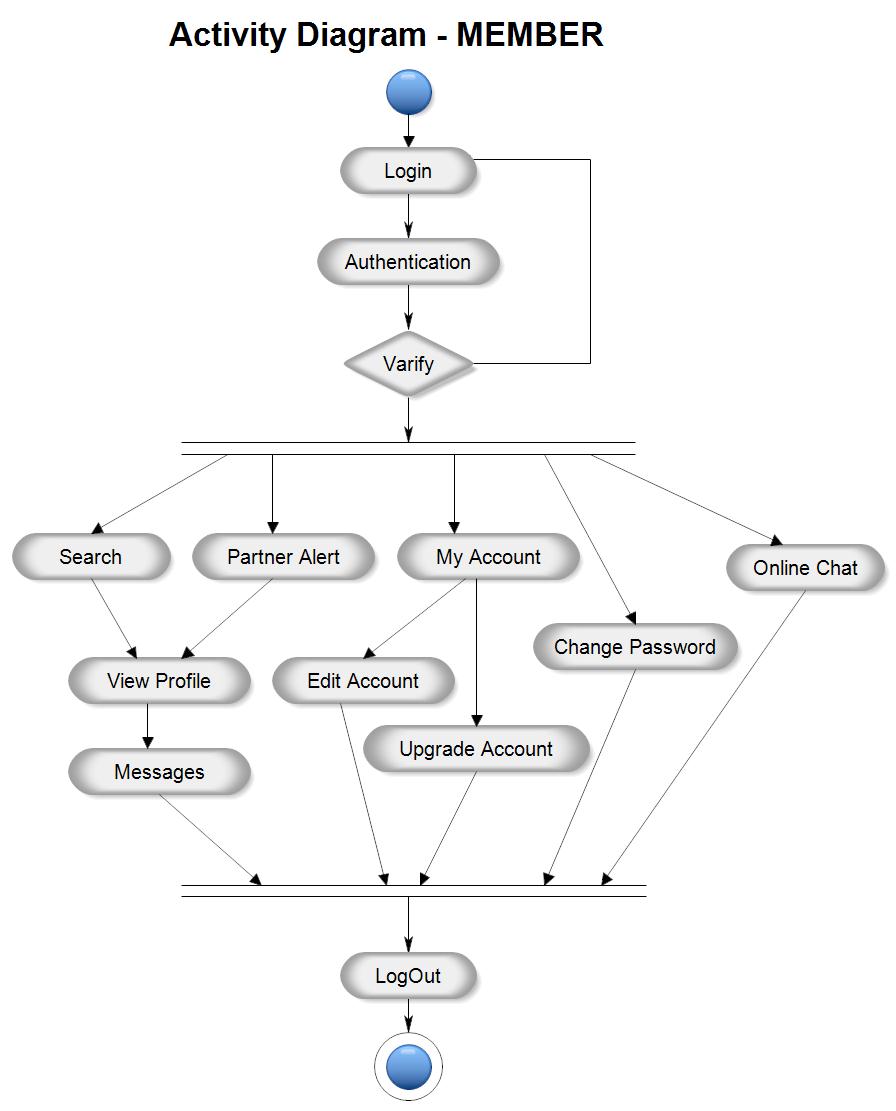
[11. RESPONSIBILITIES 4](#_Toc126659500)2

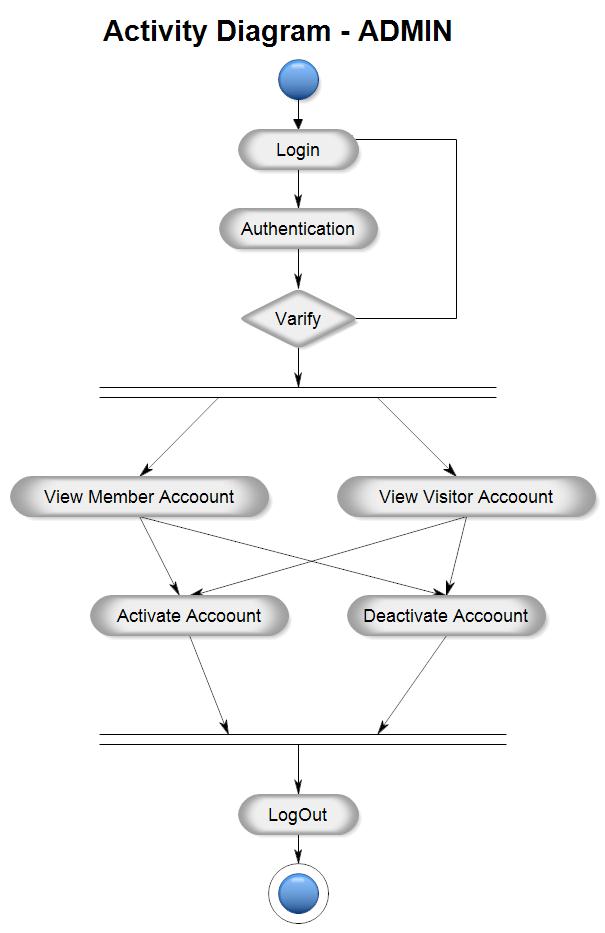
[12. TESTING SCHEDULE 4](#_Toc126659501)2

[13. PLANNING RISKS AND CONTINGENCIES 43](#_Toc126659502)

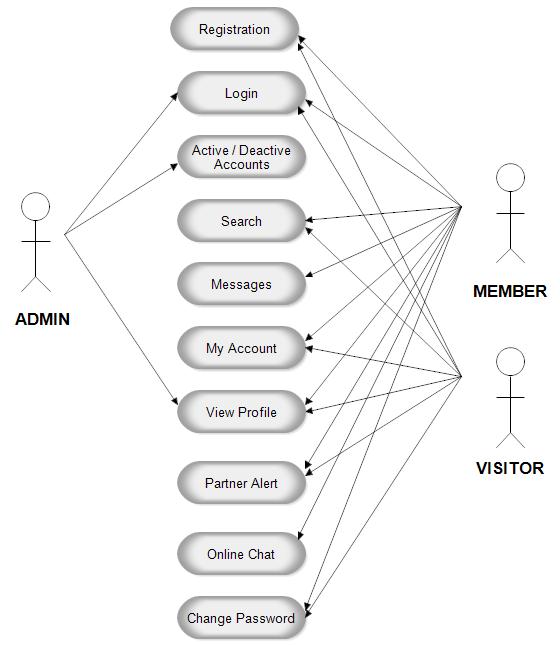
# TEST PLAN IDENTIFIER: OM-101

# REFERENCES

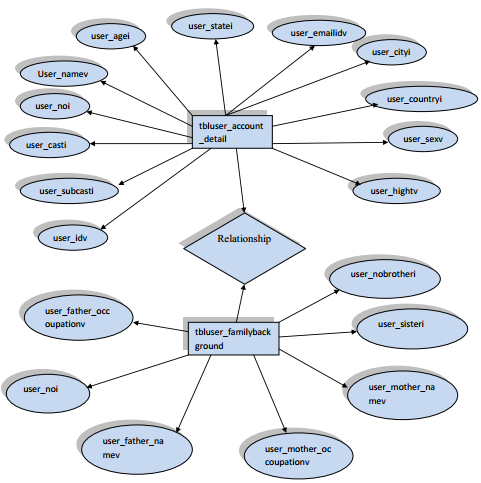
**SRS Diagram:**



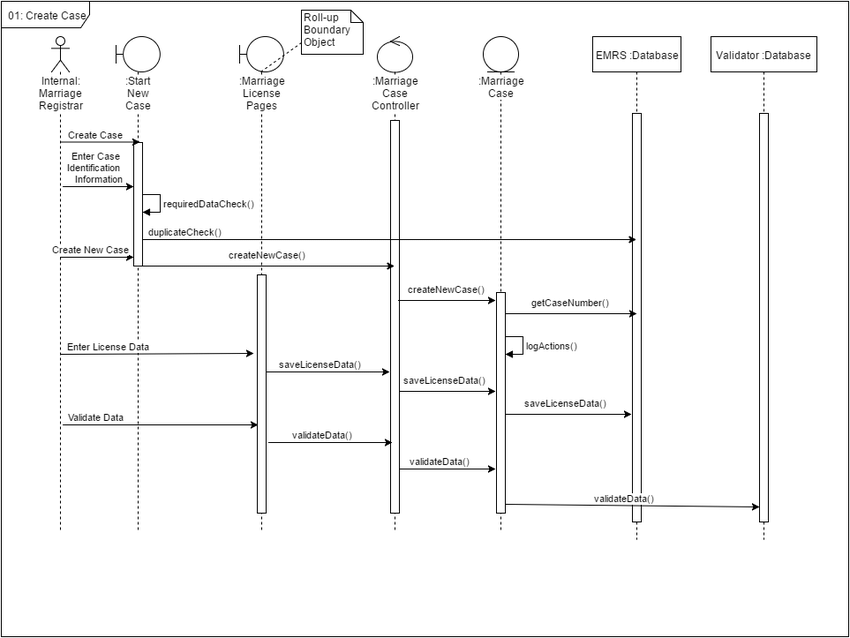
**Use Case Diagram:**



**ER Diagram:**



**Sequence Diagram:**



**References:**

1. **SRS Diagrams:**

**[1] Meeraacadamy. (n.d.). Activity Diagram for Member. Meeraacademy.** **<https://meeraacademy.com/activity-diagram-for-matrimonial-website-project/>**

**[2] Meeraacadamy. (n.d.). Activity Diagram for Admin. Meeraacademy. <https://meeraacademy.com/activity-diagram-for-matrimonial-website-project/>**

1. **Use Case Diagram:**

**[1] Meeraacadamy. (n.d.). Use Case Diagram for Matrimonial Website. Meeraacademy.**

**<https://meeraacademy.com/use-case-diagram-for-matrimonial-website-project/>**

1. **ER Diagram:**

**[1] 1000projects. (n.d.). ER Diagram for Matrimony Website. 1000projects.** **<https://1000projects.org/satvara-matrimony-php-project.html>**

1. **Sequence Diagram:**

**[1] Research Gate. (n.d.). Sequence Diagram for Matrimony Website. Research Gate.**

**<https://www.researchgate.net/figure/Sequence-diagram-for-Use-Case-1_fig5_313870078>**

# INTRODUCTION

## 3.1 Background to the Problem

* The technical scope involved in developing and implementing this matrimonial project can let an individual find their potential matches for marriage according to their priorities set. This project allows us to get the thought of online marriage and make customers put their trust in online-based platforms for marriage. This application allows browsing profiles of those who have registered themselves on this site as well as those who have not registered yet, they can have an overview of the site. This allows individuals to give their information such as Name, Gender, Religion, Caste, Country, Marital status, Current salary, Occupation etc. This application also allows uploading photos of the individual registering through verification. The person looking for marriage can register and search for a profile that matches their requirements. They can also search by their own preferences like gender, age, religion, caste, marital status, height etc. and allow individuals to view the verified information which today is at the highest priority. When an individual selects a profile which matches his or her requirements, it will also show the opposite person’s interests and they can communicate through the application. Thus, this application allows individuals to let others view its profile and allows it to view other’s profiles depending on the priority set.
* The People have their individual choices and own personal reasons for their tastes but especially in our country many people who are not in a marital relationship get bullied for various reasons. In Bangladesh getting married is one of the difficult jobs one can face when they want to get married. They meet in person to person, talk with each other and take time to decide if they are a perfect match for each other or not. It causes time problem when setting for a meeting as in Bangladesh most of the person is doing jobs or businesses. So, online platform is a great way to communicate, and one thing is that they don’t have to meet in person to talk unless they liked each other profile or they don’t have to ask for the bio data as it will be in the system when a person register in the website. So, this project will help many individuals to find their own matches considering their situations.

Considering all of these issues, it is clear that this is a matter of concern for all of us.

## 3.2 Solution to the Problem

* The main objective of matrimonial sites is to provide grooms and brides with excellent matchmaking experience by exploring the opportunities to find out their partners by their own preferences. Keeping our objective in mind we want to create a reputed online platform that touches the souls of people all over the world. Also, as this concept is not so familiar in our country, we want people to trust this site and get accustomed. Since the internet is a pivot for modern business, our project which is based on the internet paves a path for modernization in trade.
* As There are few matrimonial sites in our country because it is not widely used here. But the thing is conventional way of marriage, more specifically finding partners is actually tough in considering different situations. Sometimes people get harassed, the matchmaker could not fulfill preferences of both sides and some unwanted situations can happen. Since our world is getting modernized and almost everything is now online based so we thought of making a trustable matrimonial site which people can rely on. Now as there is already some application based on this problem our project is going to be in category B. So here the question arises why we want to develop this kind of site again. The answer is a huge part of our population don’t trust online based platforms and 90% of them will say it is not safe. So, our target is to make an online matrimonial platform which is strongly secured, requires verification of the users, safe and user friendly.

There will be mainly two types of users. The admin part and the customers. Customers who are having issues finding their partners or they don’t have enough time to follow the conventional way they can easily register and find by their own match as well as can communicate. They can see the CV and other verified information along with the photos of the other, saving their time as well. Admins along with the group members will maintain the data of the users, activity of the users and if they find any suspicious things going on or any kind of data mismatch or false information, they can take actions against the user.

* At present, there are a couple of software solutions available to support the proposed concept of online matrimony in our country. However, the project can initiate effective measures by following the logistical operations conducted by companies like Bangladeshi Matrimony and BD Matrimony.

The proposed project’s objective is to reach out to people of different levels and encourage them to join the initiative. The initiative hopes to do this in order to address the significant problems and shortcomings that have prevented the concept's use for a long time. Through this approach, the project hopes to engage the support of the wider community.

# REQUEIREMNT SPECIFICATION

## System Features

1. **Visitor Page**

**Functional Requirements**

* 1. The software should allow the users to select “Matrimony profile for” option.
  2. The users should be able to enter the name in the specific space.
  3. The users should be able to enter the mobile number in the specific space.
  4. The user should be able to click the “Register Free” button. This will take the user directly to the registration page.
  5. The software shall allow the users to click on the “Login” button. This will take the users directly to the login page.

Priority Level: High

Precondition: user should give valid name and mobile number.

Cross-references: 2, 3

1. **Software Login**

**Functional Requirements**

* 1. The login page should have two options, login and sign up. It should allow the users to login with their given username and password.
  2. The username and password should be verified with the database records.
  3. If the login is successful, the homepage of the user account will be displayed.
  4. If the inserted username and password is wrong, the random verification code will be generated and sent it to the user email address by the system to retry login.
  5. If the number of login attempt exceed its limit which is 3 times, the system will show “Forget Password?”
  6. If anyone clicks in the forget password option, a page will show, and the user must write his or her mail address. A verification code will send to user’s mail.
  7. When User put the verification code in the required place, it will allow user to change the password. Then the user will auto log in and the home page will come.
  8. If the user is new, they will click for sign up option and it will take the user to register page.

Priority Level: High

Precondition: user should have existing valid name and password.

Cross-references: 3, 4

# Registration Page

**Functional Requirements**

1. The user should be able to do registration smoothly by filling in the bellow information.

* Select gender
* Choose country code
* Mobile number
* Email address

1. Now the user should be able to click on the submit button. An email should be sent to the user’s mail address with a verification code. Now the user should:

* Submit the code (which has been sent to the email)
* The system should verify the code and successfully create the user’s profile.

1. After the verification the user now should be able to fill the “Basic information form”, where they have to fill up:
   * + Name
     + Date of birth
     + Height
     + Weight
     + Religion
     + Marital status
     + Interests
     + NID number
     + Picture of NID and birth certificate
     + Photo
     + Education and carrier

* Higher education
* Occupation
* ID of working Institution
* Income Status
  + - Location
    - Details
* Country
* City
* Area
* Permanent address
* Family Details
* Fathers name
* Mothers name
* Siblings

1. The users should be able to view their complete profile and there they should be able to write a:

* Short biography

Priority Level: High  
Precondition: user should have valid NID/Birth Certificate/ID of valid working institution and password.  
Cross-references: 4

# 4 Home page

**Functional Requirements**

1. The user should be able to see the home page where there is a menu bar containing:

* Home (user shall see their own profile details here)
* Profile view (users shall start seeing others profile from here)
* Show biodata
* Search
* History (users shall see their search history from here)
* Report a problem
* Help center
* Safety center
* Logout

1. In the home page users should be able to see their profile picture and name.

Priority Level: High  
Precondition: Users should see their profile, others profiles as well as sould be able to do report if needed.  
Cross-references: 5

# 5 Search Option

**Functional Requirements**

1. When a user wants to search profiles, they should be able to search by names.
2. Also, they should be able to search by their requirements. For that they need to choose several options like:

* Search by age
* Search by height
* Search by country
* Search by skin color
* Search by occupation

1. Based on users searching requirements system shall show a list of their match.
2. Users shall visit those profiles and see their basic information.

Priority Level: High  
Precondition: User should fill up the search criteria by their own preferences.  
Cross-references: 4.1, 6

# Communication

**Functional Requirements**

1. If the user finds their match they shall communicate through this application. There will be three types of communication system:

* Call
* Message
* Video call

Priority Level: High  
Precondition: User should communicate with the profiles they are interested into.  
Cross-references**: N/A**

# 7 Logout

**Functional Requirements**

* 1. Users can logout anytime they want.

Priority Level: High  
Precondition: User account required.  
Cross-references: 4

# Admin Module

**Functional Requirements**

* 1. The main functions of admin are:
* Login
* Verify members
* Receive reports from all the members.
* User management
* Logout

Priority Level: High  
Precondition: Administration access, Internet, the software itself.  
Cross-references: N/A

## System Quality Attributes

**QA1: Usability:** A trained user shall be able to complete the whole account creation procedure within 5 – 7 minutes.

Priority Level: High  
Precondition: Stable and fast internet  
Cross-references: N/A

**QA2: Usability:** The software should warn the user about the user policy and agreement if they are not willing to resister their profiles via the app.

Priority Level: High  
Precondition: Mobile phone, Internet, the software itself.  
Cross-references: N/A

**QA3: Reliability:** The software should be able to connect the users through proper verification by sending verifying codes.

Priority Level: High  
Precondition: Stable and fast internet. Must provide verified credentials.  
Cross-references: N/A

**QA4: Reliability:** The users should be verified properly and have to give all the required information carefully because it will be public as every user of this software shall see each other’s details.

Priority Level: High  
Precondition: Mobile phone, Internet, the software itself.  
Cross-references: QA3

**QA5: Security:** Unauthorized login should be notified to the user.

Priority Level: High  
 Precondition: Mobile phone, Internet, the software itself.  
 Cross-references: QA3

**QA6: Flexibility****:** The user should be able to do reports if in any situation it is needed. The procedure and the feedback should be given as soon as possible.

Priority Level: High  
Precondition: Mobile phone, Internet, the software itself.  
Cross-references: N/A

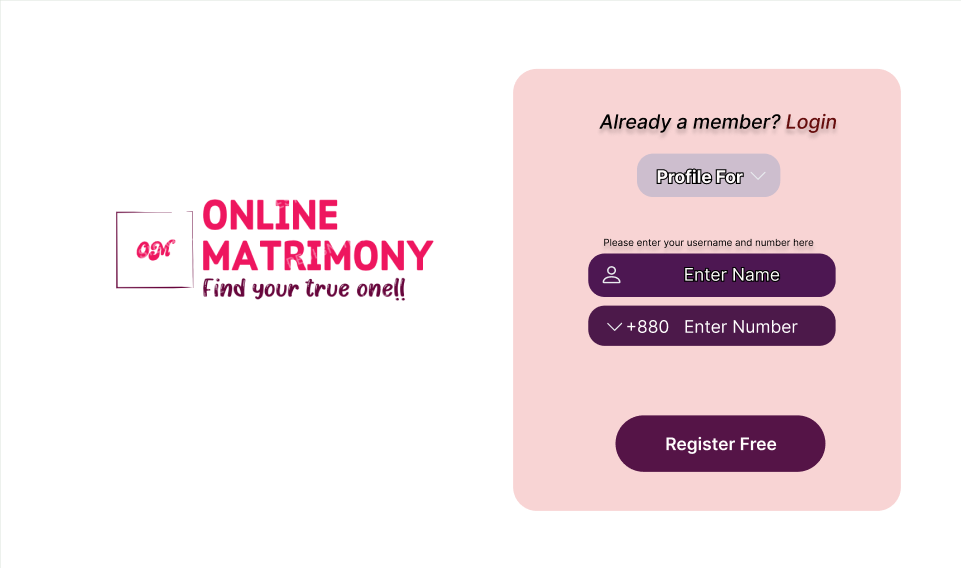
**QA7: Modularity:** The software should be separated into modules or components that can be independently tested. This simplifies the debugging process by making it easier to pinpoint the cause of issues.

Priority Level: Medium  
Precondition: Low coupling, High Cohesion architecture.  
Cross-references: N/A

**QA8: Portability:** The system should be available across two major operating systems, Android and iOS, so that it may reach out to people with diverse preferences for greater accessibility.

Priority Level: High  
Precondition: Reliable developer team, testing team.  
Cross-references: QA

## System Interface



**Figure 4.3.1: Visitors Page**

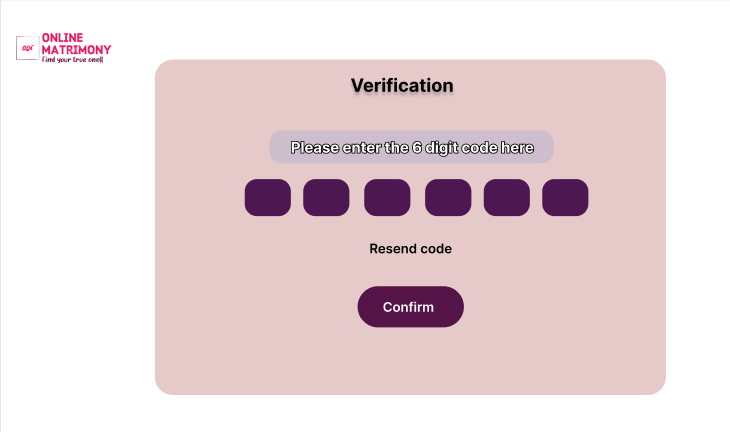
A screenshot of a login screen

Description automatically generated**Figure 4.3.2: Login Page**

**A screenshot of a registration form

Description automatically generated**

**Figure 4.3.3: Registration Page**

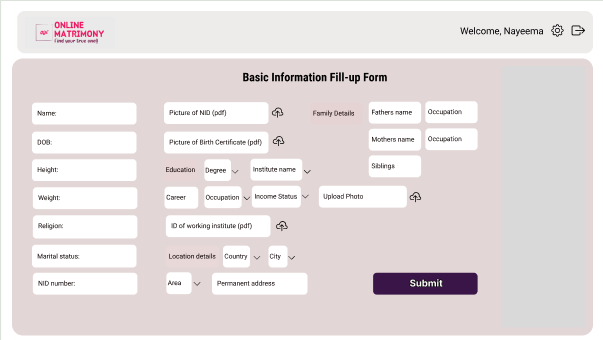
****

**Figure 4.3.4: User verification**

**A screenshot of a registration form

Description automatically generated**

**Figure 4.3.5: Registration completed by verifying the user through email.**

****

**Figure 4.3.6: basic information fill-up form to complete the registration process.**

**A screenshot of a computer

Description automatically generated**

**Figure 4.3.7: User profile page**

**A screenshot of a phone

Description automatically generated**

**Figure 4.3.8: Home page**

**A screenshot of a search box

Description automatically generated**

**Figure 4.3.9: Search by criteria page**

**A screenshot of a computer

Description automatically generated**

**Figure 4.3.10: Showing results based on users search preferences.**

## Project Requirements

Software project type: Organic

So, Coefficient = 2.4

P = 1.05

T = 0.38

SLOC = 5000

PM = Coefficient <Effort estimation> \* (SLOC/1000) ̂ P

= 2.4 \* (5000/1000)1.05

= 13

DM = 2.05 \* (PM) ̂T

= 2.05 \* (13)0.38

= 6.62 months

ST = PM/DM

= 13/6.62

= 1.96

≈ 2 people

Let’s assume the average salary of each employee in our company is Tk 20,000.

Estimated Project Cost = PM \* Average Cost Per Person-Month

= 13 \* 20,000

= 260,000 tk

# 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. For example:

* + - 1. Minor UI/UX updates: Small updates to the user interface or user experience can be considered for exclusion from testing, as they are unlikely to affect the application's functionality.
      2. Third-party integrations: If the application integrates with third-party services such as payment and so on, these features may be ignored if they have already been tested and validated by their respective suppliers.

# TESTING APPROACH

## Testing Levels

* **UNIT TESTING:** During unit testing, we will run tests on individual software modules to look for flaws. The developer will conduct this testing, which will be approved by the development team leader. Before unit testing is accepted and passed on to the test person, the programmer must present proof of unit testing (test case list, sample output, data printouts, defect information) to the team leader. The goal is to ensure that each unit of code works as it should.
* **INTEGRATION TESTING:** Following unit testing, we will perform integration testing. Modules are used for testing in unit testing, and these modules are merged and tested in integration testing. The software is built with a number of software modules that are coded by several coders or programmers. The purpose of integration testing is to ensure that all modules communicate correctly. We will use the Sandwich strategy in our software to integrate, and once each module is completed, we will merge and test them again.

**Scope:** The scope of this integration testing plan includes the following features:

• User account creation

• User login and signup

• Search user by specific requirements

• See user profile

• See availability of user

• Take a meeting through calls

• Attach documents or photos for verification of the user

• User updates his valid profile

• developer checks their reports

**Test Cases:** The following are the test cases that will be performed during the integration testing phase:

• TC-1: Verify that users can create an account with a valid email address and password.

• TC-2: Verify that users cannot create an account with an invalid email address or password.

• TC-3: Verify that users cannot create more than one account with the same email address.

• TC-4: Verify that users can login with their name and password.

• TC-5: Verify that users cannot login with an invalid name or password.

• TC-6: Verify that users can search for specific requirements .

• TC-7: Verify that users can see a list of the match by their search criteria.

• TC-8: Verify that users can see the match profiles, including their all information.

• TC-9: Verify that users can communicate with the matched person.

• TC-10: Verify that the software development member can view the documents or photos and use them to help with the consultation.

• TC-11: Verify that users can do report (for valid reason only).

• TC-12: Verify that the development members are notified when reports are submitted.

**Entry criteria:**

• All unit testing has been completed and all defects have been fixed.

• The system architecture has been finalized and all dependencies have been identified.

• The integration test plan has been developed and approved.

• The test environment has been set up and is ready for testing.

• The data for testing has been created.

**Exit criteria:**

• All integration tests have been executed and passed.

• All defects found during integration testing have been fixed.

• The system is ready for system testing

**Expected Results:** The expected results for each test case will be the same as the requirement.

* **SYSTEM TESTING:** After completing integration testing, we proceed with system testing, where we will thoroughly test the fully integrated system to ensure it meets all the specified requirements. This testing will be done by using the "Black Box Testing" technique, as it focuses on the system's overall functionality without examining the internal code or structure. Our goal is to verify that the system functions as a complete, integrated unit and meets all the requirements.
* **Acceptance Testing:** Then we will do Acceptance testing which is a testing technique performed to determine whether the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it has met the required criteria for delivery to end users. Acceptance testing (UAT) is often overlooked because it is seen as a formality, rushed, not well-defined, and not user centric. Despite these challenges, UAT is an important part of the software development process and should be done thoroughly to ensure that the product meets the requirements of the users and is ready to be released. To make UAT effective, it should start early in the development process, involve the users, use a variety of testing methods, document the results, and follow up on defects.
* **REGRESSION TESTING:** for regression testing, the test manager will be responsible for ensuring that all test cases that were previously executed without issue will continue to execute correctly after changes have been made to the system. The test manager will conduct this type of testing after each major development phase has been completed, and it will be conducted by executing previously executed test cases.
* **BLACK-BOX TESTING:** The testing team will do black box testing with no prior knowledge of the system's internal workings. The testing team will test the system from the standpoint of the user, ensuring that it meets all of the needs given in the user's requirements document. The testing team will have access to the user's requirement document and will write test cases based on it.

The test case will be run on the system, and any faults discovered will be documented and sent to the development team for rectification. The development team will then remedy the issues, and the testing team will double-check that the defects have been addressed before retesting the system.

The test manager will oversee the black box testing process, ensuring that the testing team follows the testing plan and that all problems are correctly recorded and reported. The test manager will also be in charge of ensuring that the system is ready for end-user acceptance testing.

## Test Tools

The list of test tool which will be used in the project are as follows,

1. **Unit Testing:**
   * PHPUnit: a unit testing framework for PHP programming language.
2. **Integration Testing:**
   * Selenium: a web application testing tool that supports multiple browsers
   * SoapUI: a web services testing tool that supports SOAP and REST protocols.
   * Apache JMeter: a load testing tool that supports HTTP, FTP, JDBC and more.
3. **System Testing:**
   * HP Quality Center: a comprehensive testing management tool that includes test planning, execution, and reporting features.
4. **Regression Testing:**
   * Jenkins: a continuous integration tool that can run automated regression tests.
5. **Black Box Testing:**
   * Selenium: a web application testing tool that supports multiple browsers
   * Apache JMeter: a load testing tool that supports HTTP, FTP, JDBC and more.
   * SoapUI: a web services testing tool that supports SOAP and REST protocols.
6. **Acceptance Testing:**
   * Cucumber: a testing tool that supports behavior-driven development (BDD)
7. **Excel:**

* Excel is a great tool for creating timeline charts and graphs.

## Meetings

Here's a scheduled format for the XP (Extreme Programming) model:

* **Stand-Up Meetings:**

Date/Time: Every weekday morning

Attendees: XP team members, including developers, testers, project manager, customer representative

Agenda:

1. Briefly discuss what each team member worked on yesterday, what they plan to work on today, and any blockers or challenges.
2. Identify any issues that require further discussion or action and determine next steps.
3. Keep the team informed about each other's work and progress.

* **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:

1. Demonstrate completed user stories to the customer representative.
2. Receive feedback from the customer representative.
3. Discuss any issues that arose during the iteration and determine how to address them in the next iteration.

* **Coding Standards and Practices Meetings:**

Teams will often hold meetings or discussions to establish and maintain coding standards, development practices, and guidelines. These will ensure consistency in code quality, style, and approach across the team.

# TEST CASES/TEST ITEMS

**Table – 1:** **TC**: Verifying that users can create an account with a valid email address and password.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Online Matrimony | | | Test Designed by: Abid Hasan | | |
| Test Case ID: OM\_1 | | | Test Designed date: 2023.08.20 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Verify that users can create an account with a valid email address and password. | | | Test Execution date: | | |
| Test Title: User validation | | | | | |
| Description: The goal of this test case is to validate and justify whether the user is real or not. As, the system is an online matrimony site, so it is highly recommended to justify this step. | | | | | |
| Precondition (If any): Complete Account Registration, User profile Photo, valid NID. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| * Open app * Login. * Users are asked to give their email. * Users are asked to give their password. * Display success or failure message after confirming. | Email: [nayeema@gmail.com](mailto:nayeema@gmail.com)  Password: 17n09z\_@#$ | Users should be able give their email address.  Users should be able to write the password.  After the user has given their valid email and password, they should be able to register successfully. | |  |  |
| Post Condition: If the user’s given email is verified the user should be able to do the complete registration. Otherwise, it will also inform about the failure and will suggest the users to re-verify the given email to go to the next step of registration. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Online Matrimony | | | Test Designed by: Abid Hasan | | |
| Test Case ID: OM\_2 | | | Test Designed date: 2023.08.20 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Verify that the users cannot create more than one account with the same email address. | | | Test Execution date: | | |
| Test Title: Complete checking after of an account registration. | | | | | |
| Description: The goal of this test case is to validate and justify the whole account registration processes so that after the registration the user should not be able to create another account with the same email. | | | | | |
| Precondition (If any): Valid email, mobile number, password. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open the system. 2. Tap on register free button. 3. Fill up all the required fields (gender, country code, email, mobile number) 4. Tap confirm button. 5. Show if the registration is successful or not. | Email: [nayeema@gmail.com](mailto:nayeema@gmail.com)  Mobile No.: 01316610\*\*\*  Country code: +880  Gender: Female | Should be able to identify whether the email is already in the database record or not.  Should inform user about if the email already has an account.  A message should pop up about register is not successful if the email address has been used already. | |  |  |
| Post Condition: User is validated with database and successfully check the following details. The account session details are logged in the database. | | | | | |

**Table – 2:** **TC:** Verifying that users cannot create more than one account with the same email .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Online Matrimony | | | Test Designed by: Nayeema Hoque | | |
| Test Case ID: OM\_3 | | | Test Designed date: 2023.08.22 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Verify that users can login with their name and password. | | | Test Execution date: | | |
| Test Title: Login verification | | | | | |
| Description: The goal of this test case is to check whether the users can login to the system with their registered name and email smoothly and successfully. | | | | | |
| Precondition (If any): Valid user account. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Open the system. 2. Click on the sign in button 3. Enter name 4. Enter password 5. Click on the submit button 6. Login Successful. | Name: Nayeema Hoque  Password: 17n09z\_@#$ | 1. The system should be able to match the given name and password with the database record.  2. The user should be able to login successfully if the given name and password is valid.  3. If the login is not successful, users should be able to attempt login 2 more times. | |  |  |
| Post Condition: User should be able to go to the home page and see the dashboard if login is successful. If the login is not successful, the users will be able to change their password by clicking on forget password. | | | | | |

**Table –3:** **TC:** Verify that users can login with their name and password.

**Table – 4:** **TC:** Verify that users can see a list of the match by their search criteria.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Secured Bus Transportation System for Women | | | Test Designed by: Nayeema Hoque | | |
| Test Case ID: OM\_4 | | | Test Designed date: 2023.08.22 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Verify that the users can see a list of the match by their search criteria. | | | Test Execution date: | | |
| Test Title: Checking search and search by criteria module. | | | | | |
| Description: The goal of this test case is to check whether the search by name and the search by criteria is showing preferable list or not. | | | | | |
| Precondition (If any): Valid user account, internet. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. From the home page click on the search button. 2. Search by name. 3. Search by criteria by choosing several options like age, height, occupation etc. 4. Show a list. | Search by any name or search by choosing the given criteria. | 1. The system should be able to provide a list based on the search criteria.  2. The user should be able to view the profiles. | |  |  |
| Post Condition: After searching the system will provide a list of the users preference and the users will be able to see the profiles and details of other users. | | | | | |

**Table – 5:** **TC:** Verify that users can see the match profiles, including their all information.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Online Matrimony | | | Test Designed by: Zerin Hasan Sahosh | | |
| Test Case ID: OM\_5 | | | Test Designed date: 2023.08.24 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Verify that the users can see the matched profiles, including all their information. | | | Test Execution date: | | |
| Test Title: Checking the systems is able to view others profile details. | | | | | |
| Description: The goal of this test case is to check whether the user gets see theirs matched profiles details or not. | | | | | |
| Precondition (If any): Valid user account, internet, stable phone status. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Search by criteria by choosing several options like age, height, occupation etc.  2. Show a list.  3. click on the profiles to see their all information. | Search by any name or search by choosing the given criteria. | The app/system should be able to allow the users to see their matched profiles details. | |  |  |
| Post Condition: The users will be able to communicate with their matched profiles via message, call and video call option. | | | | | |

**Table – 6:** **TC:** Verify that users can communicate with the matched person.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Online Matrimony | | | Test Designed by: Zerin Hasan Sahosh | | |
| Test Case ID: OM\_6 | | | Test Designed date: 2023.08.24 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Verify that the users can communicate with the matched person. | | | Test Execution date: | | |
| Test Title: Checking the system is able to make communication between two users. | | | | | |
| Description: The goal of this test case is to observe whether the app is working for the communication between the users. | | | | | |
| Precondition (If any): Valid user account, internet, stable phone status. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Click one of the matched profiles. 2. Click on the message option. | Send a text to another user. | * + 1. The system should send the message to the other user.     2. The users should be able to receive the massages and reply as well. | |  |  |
| Post Condition: The user can skip the profiles if not interested. | | | | | |

**Table – 7: TC:** Checking scalability.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Online Matrimony | | | Test Designed by: Nayeema Hoque | | |
| Test Case ID: OM\_7 | | | Test Designed date: 2023.08.25 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: | | |
| Module Name: Checking whether the app could handle large amount of data/group of users as well as checking whether the database reactive integrated by the app. | | | Test Execution date: | | |
| Test Title: Checking App Scalability. | | | | | |
| Description: The goal of this test case is to check whether the database is reactive and could handle large amount of data and user operating at the same time. | | | | | |
| Precondition (If any): Valid user account, internet. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Test tools can be used to accomplish the test. 2. User feedbacks can be collected. | Tester/Developer Expectation | The app/system should handle multiple request at the same time. The database should response accordingly without any failure. | |  |  |
| Post Condition: The app can tested based on further criteria. | | | | | |

**Table – 8:** **TC:** Checking Usability.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Online Matrimony | | | Test Designed by: Abid Hasan | | |
| Test Case ID: OM\_8 | | | Test Designed date: 2023.08.25 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Checking whether every module interaction is user friendly, user convenient and flawless. | | | Test Execution date: | | |
| Test Title: Checking App Usability. | | | | | |
| Description: The goal of this test case is to observe whether the app is user friendly. | | | | | |
| Precondition (If any): Various Test Tools. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Checking if the buttons are placed where the user may focus easily. 2. Checking where a basic form takes more than 5 minutes. 3. Checking whether the app features are placed on user’s eye focus. | Tester/Developer Expectation | The app/system’s navigation system should fit the general mass’s muscle memory.  The forms should consume less user time. | |  |  |
| Post Condition: N/A | | | | | |

**Table – 9:** **TC:**  Verify that users can do report (for valid reason only).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Online Matrimony | | | Test Designed by: Zerin Hasan Sahosh | | |
| Test Case ID: OM\_9 | | | Test Designed date: 2023.08.25 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Checking whether the users can do report about something. | | | Test Execution date: | | |
| Test Title: Checking on the report function. | | | | | |
| Description: The goal of this test case is to observe whether the users can successfully report a problem if needed. | | | | | |
| Precondition (If any): N/A | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Click on the report button from the home page menu, 2. Write a short description of the problem. 3. Send it. | Write the short description of the problem in the report option. | The system should be able to send the report to the admin. | |  |  |
| Post Condition: The admin will take necessary actions against the report. | | | | | |

**Table – 10:** **TC:** Checking robustness.

|  |  |
| --- | --- |
| Actual Results | Status (Pass/Fail) |
|  |  |

…..

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Secured Bus Transportation System for Women | | | Test Designed by: Zerin Hasan Sahosh | | |
| Test Case ID: EUPH\_10 | | | Test Designed date: 2023.08.26 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: | | |
| Module Name: Checking whether the user gets recovered from any inconveniences if the internet connection is lost/the phone gets switched off/the app crashes. | | | Test Execution date: | | |
| Test Title: Checking App Robustness. | | | | | |
| Description: The goal of this test case is to check whether the user gets to retrieve lost data from any sudden interruption. | | | | | |
| Precondition (If any): Valid user account, internet. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| * + - 1. Turn off internet connection/switch off phone.       2. Turn the internet connection back on/ turn on the phone.       3. Check progress. | Tester/Developer Expectation | The app/system should be able to automatically update the ongoing tasks that were previously interrupted. | |  |  |
| Post Condition: The app must operate as usual after the progress or ongoing tasks are retrieved. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Secured Bus Transportation System for Women | | | Test Designed by: Zerin Hasan Sahosh | | |
| Test Case ID: EUPH\_10 | | | Test Designed date: 2023.08.26 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: | | |
| Module Name: Checking whether the user gets recovered from any inconveniences if the internet connection is lost/the phone gets switched off/the app crashes. | | | Test Execution date: | | |
| Test Title: Checking App Robustness. | | | | | |
| Description: The goal of this test case is to check whether the user gets to retrieve lost data from any sudden interruption. | | | | | |
| Precondition (If any): Valid user account, internet. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| * + - 1. Turn off internet connection/switch off phone.       2. Turn the internet connection back on/ turn on the phone.       3. Check progress. | Tester/Developer Expectation | The app/system should be able to automatically update the ongoing tasks that were previously interrupted. | |  |  |
| Post Condition: The app must operate as usual after the progress or ongoing tasks are retrieved. | | | | | |

**Table – 11:** **TC:** Verify that the admins are notified when reports are submitted.

# ITEM PASS/FAIL CRITERIA

1. **User Registration and Login:**

**Developers:** Develop the registration and login functionality.

**Testers:** Test the registration and login process for errors, security vulnerabilities, and user experience.

**Project Manager:** Ensure the registration and login process meets the project requirements and timelines.

**Client:** Provide feedback on the registration and login process.

**Pass:** User can register and log in smoothly, securely, and without errors.

**Fail:** User encounters issues with registration and login, such as errors, bugs, or security concerns.

**Next Steps:** Developers fix any issues identified by testers, and the process is retested.

1. **Basic Information Form:**

**Developers:** Develop basic information forms and update functionality.

**Testers:** Test the basic information form and update process for errors, security vulnerabilities, and user experience.

**Project Manager:** Ensure the basic information form process meets the project requirements and timelines.

**Client:** Provide feedback on the profile creation and update process.

**Pass:** User can fill-up and update their profile smoothly, without errors.

**Fail:** User encounters issues with the basic information form fill-up process, such as bugs, errors, or security concerns.

**Next Steps:** Developers fix any issues identified by testers, and the process is retested.

1. **Search by Criteria Option:**

**Developers:** Develop the search by criteria option functionality.

**Testers:** Test the search by criteria option process for errors, accuracy of information, and user experience.

**Project Manager:** Ensure the search by criteria option process meets the project requirements and timelines.

**Client:** Provide feedback on the search by criteria option process.

**Pass:** User can search by their own preferences.

**Fail:** User encounters issues with the searching process, such as incorrect information, errors, or bugs.

**Next Steps:** Developers fix any issues identified by testers, and the process is retested.

1. **Communication:**

**Developers:** Develop the communication system functionality, including call, message and video call.

**Testers:** Test the communication system for accuracy, reliability, and security.

**Project Manager:** Ensure the communication system meets the project requirements and timelines.

**Client:** Provide feedback on the communication system.

**Pass:** communication system is efficient, and the driver rating system is accurate and reliable.

**Fail:** Issues with the communication system, such as errors, or security concerns.

**Next Steps:** Developers fix any issues identified by testers, and the process is retested.

1. **Safety Features:**

**Developers:** Develop the safety features, including reporting with any issues.

**Testers:** Test the safety features for functionality, accuracy, and user experience.

**Project Manager:** Ensure the safety features meet the project requirements and timelines.

**Client:** Provide feedback on the safety features.

**Pass:** Safety features are functional and provide a sense of security to the users.

**Fail:** Issues with the safety features, such as bugs, errors, or security concerns.

**Next Steps:** Developers fix any issues identified by testers, and the process is retested.

1. **App Performance:**

**Developers:** Develop app performance, including speed, stability, and resource usage.

**Testers:** Test the app performance for speed, stability, and resource usage.

**Project Manager:** Ensure the app performance meets the project requirements and timelines.

**Client:** Provide feedback on the app performance.

**Pass:** App performance is fast, stable, and uses resources efficiently.

**Fail:** Issues with the app performance, such as slow loading times, crashes, or high resource usage.

**Next Steps:** Developers fix any issues identified by testers, and the process is retested.

1. **Scalability:**

**Developers:** Develop the app scalability, including the ability to handle large amounts of users and data.

**Testers:** Test the app scalability for performance, stability, and resource usage under high load.

**Project Manager:** Ensure the app scalability meets the project requirements and timelines.

**Client:** Provide feedback on the app scalability.

**Pass:** App can handle high loads without performance degradation, crashes, or other issues.

**Fail:** Issues with app scalability, such as crashes, slow response times, or high resource usage under high loads.

**Next Steps:** Developers fix any issues identified by testers, and the process is retested.

# TEST DELIVERABLES

* **Acceptance Test Plan:** The Acceptance Test Plan outlines the criteria for determining whether the Online matrimony is acceptable to the client. The project manager created this plan, which specifies the acceptance criteria, test scenarios, and expected results. It also contains the duties and responsibilities of each member of the acceptance testing team, as well as the testing schedule and timetable. The acceptance test plan guarantees that the software meets project specifications and provides the desired user experience.
* **System/Integration Test Plan:** The System/Integration Test Plan guarantees that the Online Matrimony system is completely integrated with the underlying systems, which include the database, servers, and other applications. The testing methodology, test cases, and expected outcomes are all outlined in the plan. The testing encompasses the system's functional and non-functional characteristics, such as performance, security, and dependability. The project manager creates the System/Integration Test Plan in collaboration with developers and testers to assure the app's stability and dependability.
* **Unit Test Plans/Turnover Documentation:** The testing procedure for specific app components or modules is outlined in Unit Test Plans and Turnover Documentation. These blueprints are created by developers to guarantee that each module works as planned and connects seamlessly with the rest of the system. The turnover manual offers instructions for installing, configuring, and testing each module. These documents are required to ensure that the system is fully functional and simple to maintain.
* **Screen Prototypes:** Screen prototypes are visual representations of the user interface of an app. Designers create these prototypes, which are used to evaluate the design and ensure that the user interface is intuitive and simple to use. Screen prototypes are required to ensure that the app's user experience matches project criteria and is appealing to end users.
* **Test Logs and Turnover Reports:** Turnover Reports and Test Logs chronicle the testing process, including test cases, results, and any issues discovered during testing. These documents are used to track testing progress and ensure that all requirements are met. Test logs and Turnover Reports are required to ensure that the app has been thoroughly tested and is ready for delivery to the client.

# STAFFING AND TRAINING NEEDS

It is preferred that at least one full-time tester be allocated to the project for the system/integration and acceptability testing phases. This will necessitate the appointment of a person part-time at the start of the project to engage in reviews and so on, and they will be allocated full-time roughly six months into the project. If a dedicated tester is not available, the project manager/test manager will take on this responsibility. In order to give complete and proper testing, the following training areas must be addressed.

* **Unit Testing:** Developers will be responsible for unit testing their own code. However, it is advised that this phase of testing be assigned to a different tester to ensure complete testing of each individual unit. The tester should be familiar with programming languages and have prior expertise with unit testing. The tester should be available to engage in reviews and testing part-time at the start of the project and full-time during the unit testing phase.
* **System Testing:** It is advised to allocate at least one full-time tester to the project for system testing. The tester should be knowledgeable about the sector, have experience with system testing, and be conversant with the project objectives. During the system testing phase, the tester should collaborate closely with the development team and be accessible for testing.
* **Integration Testing:** A minimum of one full-time tester should be allocated to the project in order to conduct integration testing. The tester ought to have previous integration testing expertise, be acquainted with the project requirements, and be knowledgeable about the sector. During the integration testing phase, the tester needs to collaborate closely with the development team and be accessible for testing.
* **Acceptance Testing:** At least one full-time tester should be allocated to the project in order to do acceptance testing. The tester should be conversant with the project requirements, have experience with acceptance testing, and have industry knowledge. The tester ought to collaborate closely with the development team and be accessible for testing throughout the acceptance testing stage.
* **Regression Testing:** The same tester who is allocated to system and integrated testing can also be assigned to regression testing. The tester should be knowledgeable about the project requirements and have experience with regression testing.
* **Black Box Testing:** It is recommended to allocate at least one full-time tester to the project for black box testing. The tester should be knowledgeable about the sector, have experience with black box testing, and be conversant with the project objectives. During the black box testing phase, the tester should collaborate closely with the development team and be accessible for testing.

**Training Needs:**

* **EDI Interface:** The fundamental functions of the EDI interface will need to be taught to the developers and tester(s). This will guarantee that the testing team can completely evaluate the system's EDI capabilities. Before the project is officially accepted, the operations employees will also need to complete training in the EDI communications procedure.
* **Test Planning and Execution:** It will be necessary to train the testing crew in test planning and execution. This will guarantee that the testing team can organize and carry out tests successfully and efficiently, and that all system components are completely examined.
* **Defect Management:** Defect management training will be required for the testing team. This will make sure that errors are correctly recorded, reported, and quickly fixed.
* **Test Automation:** The testing staff will require test automation training. As a result, the team will be able to automate tests and boost testing efficiency. The overall testing effort and expense can be decreased with the use of test automation.
* **Security Testing:** The testing group will require security testing training. This will guarantee the safety of the system and the protection of user information. The testing team should be knowledgeable with industry-accepted security testing methods and guidelines.
* **Performance Testing:** Performance testing training will be required for the testing team. This will guarantee that the system can manage predicted traffic and usage levels. The performance testing team should be familiar with tools and procedures used by the industry.
* **User Acceptance Testing:** The testing group will require user acceptability testing training. This will guarantee that the system is user-friendly and meets end-user needs. To make sure that the system satisfies the end users' needs, the testing team should collaborate closely with them throughout user acceptability testing.

In general, it's crucial to make sure the testing crew has the proper training to carry out all the testing tasks. This will guarantee that the system has undergone extensive testing and fulfills the needs of the final users. To guarantee that the testing team is knowledgeable about the most recent testing methodologies and tools, training should be continuing throughout the project.

# RESPONSIBILITIES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria/Protocols** | **TM** | **PM** | **DEV Team** | **Test Team** | **Client** |
| Test Procedures and Rules | X | X |  |  |  |
| System Design Reviews. | X | X |  |  | X |
| Detail Design Reviews. | X | X |  |  |  |
| Unit Test Documentation & Execution | X |  | X | X |  |
| Integration Test Documentation & Execution | X |  | X | X |  |
| System Test Documentation & Execution | X | X | X | X |  |
| Change Control and Regression Testing | X |  | X | X |  |
| Black Box Documentation & Execution | X |  | X | X | X |
| Screen and Report Prototype Testing | X | X | X | X | X |
| Acceptance Test Documentation & Execution | X | X |  | X | X |

# TESTING SCHEDULE

**Figure 12.1: Testing Schedule**

# PLANNING RISKS AND CONTINGENCIES

* **Delays in Development:** There is a chance that the project's development phase will be delayed because of unforeseen technical problems, adjustments to the requirements, or other circumstances. The project manager should provide extra time in the schedule to account for unforeseen delays in order to reduce this risk. The project manager should also make sure that the development team has enough personnel and the tools it needs to finish the project on schedule.
* **Integration Issues:** The process of integrating the various system parts can be difficult and drawn out. There is a chance that problems will emerge during the integration phase and cause the project to be delayed. Before moving on to the following phase, the project manager should make sure that the integration plan has been thoroughly tested and that all components have been correctly integrated in order to reduce this risk
* **User Acceptance Issues:** There is a chance that the system won't be used as intended by the end users. The project manager should include the end users early in the development process and make sure that their feedback is considered during the design and testing phases in order to reduce this risk.
* **Security Issues:** There is a chance that the system will have security lapses or flaws that could jeopardize user data or the system's general security. The development team should adhere to industry-standard security procedures, and the testing team should do extensive security testing to reduce this risk.
* **Resource Constraints:** There is a chance that resource limitations, such as a lack of funding, staff, or technical resources, could have an impact on the project's quality or timing. The project manager should distribute resources wisely and rank project tasks according to importance to reduce this risk.
* **Dependence on vendors:** The possibility of vendor dependence on external elements like libraries, frameworks, or APIs exists. The project manager should make sure that the third-party components are dependable, well-documented, and have a support plan in case of errors in order to reduce this risk.

**Appropriate contingencies should be implemented to address these risks, including:**

* Schedules should provide extra time to account for unforeseen delays.
* At every level of the development process, extensive testing and quality checks are performed.
* including end users in the creation process and considering their suggestions during the testing and design phases.
* confirming that the development team does extensive security testing and adheres to industry-standard security procedures.
* properly allocating resources and giving project activities the priority, they deserve.

**The End**