

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

**Bachelor House Ranting**

A Software Quality and Testing Project Submitted

By

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Semester: Fall\_23\_24** | | | **Section:** | **Group No:** |
| **SN** | **Student Name** | **Student ID** | Individual  Contribution (in %) | Total Marks: 50 |
| Earned Marks: |
|  | Nusrat Tabassum | 20-43094-1 | 25% |  |
|  | Shohara Afrin Bika | 20-43109-1 | 25% |  |
|  | Safwan Bin Rahman | 20-43103-1 | 25% |  |
|  | MD. Irfan Alam | 20-43663-2 | 25% |  |

The project will be Evaluated for the following Course Outcomes

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

Software Test Plan

for

Bachelor House Ranting

Version 1.0 approved

Prepared by Nusrat Tabassum, Shohara Afrin Bika, Safwan Bin Rahman and Irfan Alam

American International University-Bangladesh

12th  Octobor 2023

Table of Contents

[Revision History 3](#_Toc136846938)

[1. TEST PLAN IDENTIFIER: AT-TP01.3 4](#_Toc136846939)

[2. REFERENCE MATERIALS 4](#_Toc136846940)

[3. INTRODUCTION 4](#_Toc136846941)

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

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

[4. REQUEIREMNT SPECIFICATION 4](#_Toc136846944)

[4.1 System Features 4](#_Toc136846945)

[4.2 System Quality Attributes](#_Toc136846946) 6

[4.3 System Interface](#_Toc136846947) 7

[4.4 Project Requirements](#_Toc136846948) 10

[5. FEATURES NOT TO BE TESTED 11](#_Toc136846949)

[6. TESTING APPROACH 11](#_Toc136846950)

[6.1 Testing Levels 11](#_Toc136846951)

[6.2 Test Tools 12](#_Toc136846952)

[6.3 Meetings 13](#_Toc136846953)

[7. TEST CASES/TEST ITEMS 13](#_Toc136846954)

[8. ITEM PASS/FAIL CRITERIA 19](#_Toc136846955)

[9. TEST DELIVERABLES 19](#_Toc136846956)

[10. STAFFING AND TRAINING NEEDS 19](#_Toc136846957)

[11. RESPONSIBILITIES 20](#_Toc136846958)

[12. TESTING SCHEDULE 20](#_Toc136846959)

[13. PLANNING RISKS AND CONTINGENCIES 21](#_Toc136846960)

[14. APROVALS 22](#_Toc136846961)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Updated by** | **Update Comments** |
| 0.1 | 2023.10.04 | Safwan Bin Rahman | First Draft |
| 0.2 | 2023.10.05 | Shohara Afrin Bika | Quality Attributes Updated |
| 0.3 | 2023.10.07 | Nusrat Tabassum | Formatting |
| 0.4 | 2023.10.09 | Md Irfan Alam | Cost Analysis |

# TEST PLAN IDENTIFIER: BH-TP01.3

# REFERENCE MATERIALS

* Software Requirement Specification (SRS) Document
* Bachelor Property Tolet <https://bikroy.com/en/ads/dhaka/room-rentals>
* Room For Rent in Dhaka<https://www.thetolet.com/bd/property/category/bachelor>

# INTRODUCTION

## 3.1 Background to the Problem

Lots of people who live alone and don't have families (bachelors) often struggle to find a good place to live. On the other side, people who have extra rooms to rent out also face challenges in connecting with suitable bachelor tenants. Existing rental websites usually focus on families or individuals, making it hard for bachelors to find the right place.

The main problem is that there isn't a specific platform that understands what bachelors need when it comes to finding a place to live. The current way of doing things is kind of messy for both bachelors looking for rooms and homeowners wanting to rent to them.

## 3.2 Solution to the Problem

The "Batchelor Home Rent" project is important because it aims to fix this issue. By creating a platform that understands and meets the unique needs of bachelors and homeowners, we can make the process of finding and renting a place smoother. This project not only helps individuals directly involved but also improves the overall system of finding housing for bachelors in cities. It's about making things easier for everyone involved and making sure there's a fair and effective way for bachelors to find suitable places to live.

# REQUEIREMNT SPECIFICATION

## System Features

1. **User Login**

**Functional Requirements**

1.1 The software shall allow users to login with their given email and password.

1.2 If the email and/or password has been inserted wrong for more than three times, the random verification code will be generated by the system to retry login.

1.3 If the number of login attempt exceed its limit (5 times), the system shall block the user account login for one hour [optional function]

Priority Level: High

Precondition: user must be registered and provide valid login credentials

Cross reference: N/A

1. **User Register**

**Functional Requirements**

2.1 The software shall allow new users to register new accounts.

2.2 The software shall allow the user to login with their social accounts and email accounts.

2.3 If the user wants to register, the system shall want email, name, phone, address, date of birth.

2.4 If the user doesn't fill up any section, the system shall notify the user.

2.5 If the user inputs an already registered email, the system shall notify the user to login.

Priority Level: High

Precondition: User have valid email address.

Cross-reference: N/A

1. **Account Recovery**

**Functional Requirements**

3.1 The software shall allow the users to recover their account.

3.2 The software shall send a recovery code in email to recover.

3.3 If the user inputs correct recovery code, then the system will provide the user to reset the password.

3.4 If the user inputs wrong code, then the user shall resend the code to his email or phone.

3.5 If the number of login attempts exceeds its limit (3 times), the system shall block the user account recovery for one hour [optional function].

Priority Level: High

Precondition: User has registered email address.

Cross-reference: N/A

1. **Finding Room**

**Functional Requirements**

3.1 Users can select his/her preferable room through this software

3.2 At first user give the location and then the software will find the available house according to this address

3.3 If user like any flat as his preference he can select it then the owner of the flat get request from the user

3.4 After getting the request if the owner wants to rent house to this user, he will accept his/her request then the user can shift that flat.

Priority Level: High

Precondition: User has logged in.

Cross-reference: N/A

1. **Add available Room/House**

**Functional Requirements**

* 1. Any validate user can add available room or flat for rent
  2. If any user need renter, at first, he should login the software and give his flat location and his flat description

Priority Level: Medium

Precondition: User has logged in.

Cross-reference: N/A

## System Quality Attributes

**QA1– Usability:** The attribute usability means the ease with which each user can use the system to accomplish certain tasks. The system is designed in such a way that users will easily understand how to use the system. The features are simply designed and developed so that users can easily understand them by seeing them.

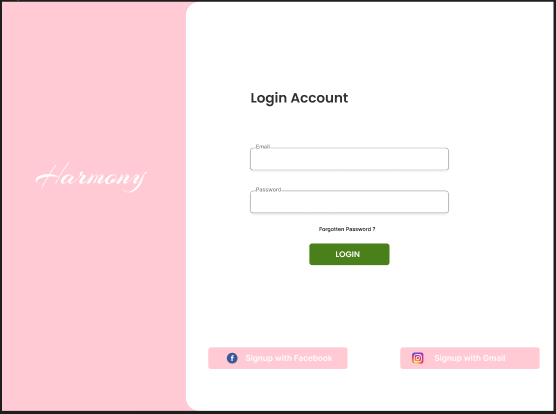
**QA2–** **Reliability:**These attributes are defined as how a system is expected to perform its intended functions with required precession. The system will be implemented in such a way that users get their intended function like when they want to view rental posts, they can see them. No error gets generated. It ensures that the user gets the correct output. Besides, the system also doesn’t take much time to respond to users’ actions.

**QA3–Flexibility:** The system is flexible enough to modify. It is adaptable to other functionalities and easy to add code to the system and upgradation for new features.

**QA4–****Security:** This attribute enables the system to control unauthorized persons to access the system. Users who have a valid email address or phone number and passwords can only sign into the system. This attribute is important since security denotes the ability of the system to protect the data from unauthorized persons.

**QA5–** **Maintainability**: It relates to how easily a software system can be updated, improved, and changed over time without introducing bugs or having undesirable side effects.

## System Interface



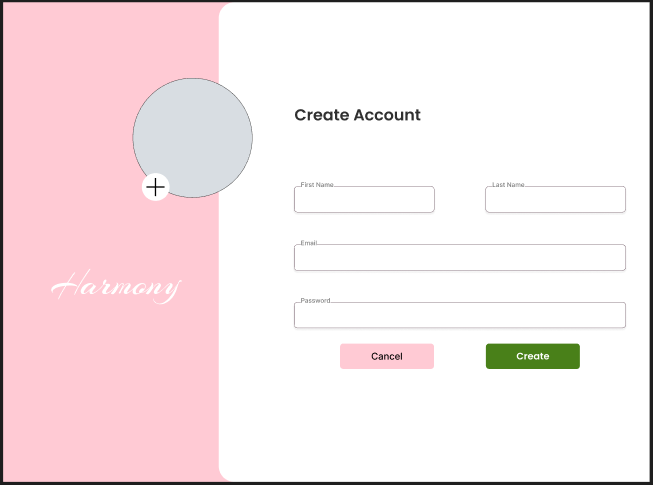
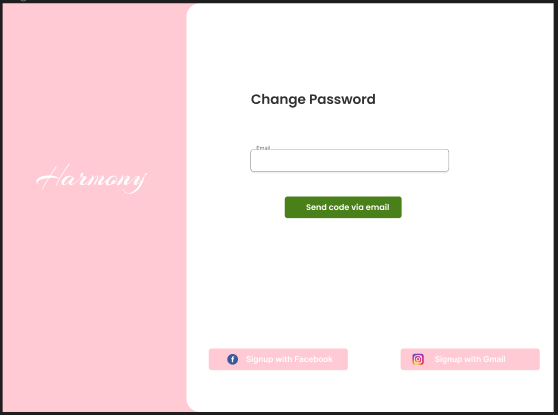
 Figure 1: Login Page

Figure 2: Registration Page

 Figure 3: Change Password Page(1)

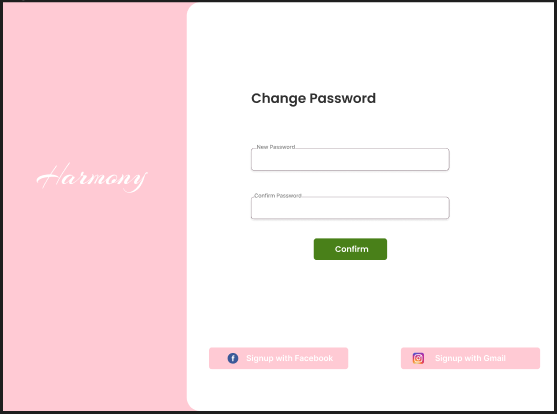


Figure 4: Change Password Page(2)

## Figure 5: Rental House Page

## 

## Figure 6: Add House/Room Page

## Project Requirements

In project management, our main goal is to finish the project on time, within the budget, and with good quality. However, there are many challenges like time, cost, scope, resources, and environment that we need to handle well for the project to succeed.

* Time: We need to finish the project on time. This means setting deadlines and closely watching the progress to avoid delays. For example, if we need 900 hours of work and work 15 hours a day, the project can be done in about 60 days or 2 months.
* Budget: We have to work within a set budget. This involves smartly using resources, keeping an eye on costs, and sticking to the budget.
* Resources: Managing and using resources like people, tools, and technical stuff efficiently is crucial for a smooth project. This helps to avoid any delays or problems during the project.
* Scope: We need to clearly define what the project will include, and everyone involved should agree on it. If there are any changes or extra things that weren't planned, we need to think carefully about them to avoid causing issues with time or resources.
* Security: It's important to make sure the system we are building is secure. This means protecting user data and making sure only the right people can access the system. Using things like encryption and regular security checks helps keep the platform safe.

COCOMO Based Effort Estimation:

Here we using COCOMO semi detected project equation

Assuming LOC=50000

KLOC= 50

Effort =a1\*(KLOC)^a2=3.0\*(50) ^1.12=239 PM

Development Time =b1\*(Effort)^b2=2.5\*(239) ^0.35=16 Months

Cost and Profit analysis:

Project development time = 16 Months

Number of programmers and QA Eng. will work = 64 week

Working days= 5 Day

Working hour per day= 8 Hours

Working hour in 1 week= (5\*8) = 40 Hours

Charge for each programmer and tester per hour = 100 TK

Charge for each programmer and QA Eng

Per week = (100\*40) = 4,000 TK

For seven week = (4000\*64) = 256000TK

Charge for 2 programmers and a QA Eng. for 64 week = (256000\*3) = 768000 TK

Other cost(Project manager, Other employees, Office rent, Electricity and other bills)=150000 TK

Total Estimated cost = 768000+150000) = 918000TK

Total estimated budget [Considering 30% profit] = (918000+(0.3\*918000)) = 3672000 TK

# FEATURES NOT TO BE TESTED

# Testing all the features in our project is really important. This helps make sure that each part of the software works well and is reliable and easy for users. It ensures that the final product is of high quality and user-friendly.

# TESTING APPROACH

## Testing Levels

**Unit Testing:**

Unit testing is the first step in our software development. It involves checking each small part (unit) of the program to ensure it functions correctly. This is crucial for identifying and fixing issues early in the development process. Developers, under the supervision of the development team lead, will isolate and test individual code components. Testing tools from official websites will be utilized to facilitate this process.

**Integration Testing:**

After unit testing, our dedicated testing teams will conduct integration testing, a critical phase in our software testing process. This involves verifying the independent functionality of each software module and ensuring the performance, functionality, and reliability of integrated modules. Various integration testing methodologies such as Big Bang, Incremental, Top-down, Bottom-up, and Sandwich can be chosen based on the development process. Thorough performance validation during integration testing reduces bugs and simplifies debugging. Automated testing tools will execute test cases, identifying and reporting defects for retesting.

**System Testing:**

Following integration testing, system testing is vital to confirm the proper functionality of all software modules as a complete system. Dedicated testing teams will perform system testing as black-box testing, considering requirements and real-world usage. A well-defined test strategy and the identification of critical modules for prioritized testing will precede system testing. End-to-end tests will ensure component interaction and interaction with external applications. Effective system testing significantly simplifies future software maintenance. Lower-priority bugs can be deferred for acceptance testing, while system testing should prioritize examining all software quality attributes to ensure optimal product quality.

**Acceptance Testing:**

Acceptance testing is a pivotal phase following system testing. It involves end-users determining the software's readiness for market release. This test employs an alpha or beta version of the product, requiring users to possess comprehensive product and domain knowledge. Any issues discovered during this phase demand immediate attention and resolution. Acceptance test outcomes validate the combined efforts of testing and development teams, reflecting overall software quality. Audience control during testing and ensuring ample data availability are essential to maintain Acceptance Test integrity.

## Test Tools

**1.Documentation:**

Confluence: Confluence is a collaboration tool for creating and sharing project documentation. It helps in maintaining a centralized and organized repository of project-related information.

**2.Testing Automation Tools:**

Selenium: Selenium is a powerful tool for automating web browsers. It is widely used for testing web applications by simulating user interactions. Selenium supports multiple programming languages, making it versatile for the development team.

JUnit/TestNG: These are testing frameworks for Java that facilitate the execution of test cases and the organization of test suites. They are crucial for implementing automated unit tests and ensuring code reliability.

**3.Version Control:**

Git: Git is a distributed version control system. It tracks changes in the source code, enabling collaboration among developers. Git allows for the management of different versions of the codebase.

GitHub: GitHub is a platform for hosting Git repositories. It provides a centralized location for code storage, version control, and collaboration among developers.

**4.Security Testing:**

OWASP ZAP: ZAP is an open-source security testing tool that helps identify vulnerabilities in web applications. It assists in ensuring the platform's security.

**5. Database Testing:**

DB Unit: DB Unit is a framework for database testing. It uses the database state as test fixtures, enabling the testing of database interactions.

## Meetings

In our project, the testing team will have regular meetings to make sure things are going well. Weekly meetings will check progress and find any problems. The test team leader will also meet with the development team and project manager every two weeks. This helps everyone stay updated and solve problems together. These meetings will happen on different weeks to cover all topics. Extra meetings can happen if there are urgent issues. This way, we make sure everyone knows what's happening and can work together to make the project successful. Our meeting plan shows that we want to communicate well, find issues early, and work together to solve problems, all of which make the project successful.

# TEST CASES/TEST ITEMS

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Bachelor House Ranting System | | | Test Designed by: Nusrat Tabassum | | |
| Test Case ID: B\_1 | | | Test Designed date:15.06.2022 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: | | |
| Module Name: Login Session | | | Test Execution date: | | |
| Test Title: verify login with valid email and password | | |  | | |
| Description: Test website login page | | |  | | |
| Precondition (If any): User must have valid username and password | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| * Go to the website * Enter email * Enter password * Click submit | Username: taba  Password: 111 | User should login into the application | | As expected, | Pass |
| Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database. | | | | | |

Table 2: Test Case for Registration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Bachelor House Ranting System | | | Test Designed by: Shohara Afrin Bika | | |
| Test Case ID: B\_2 | | | Test Designed date: 11.11.2022 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Register | | | Test Execution date: | | |
| Test Title: allow user to register with valid email and check all sections are filled. | | | | | |
| Description: Test website register section | | | | | |
| Precondition (If any): The user has a full name, valid email, password | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) | |
| * Go to the website * Enter sign up button * Fill all info box. * Click submit | Username:  taba123  Password: 2134 | User cannot be register as he/she use invalid email. | | As expected, | Pass | |
| Post Condition: User’s successful register should be stored in database. | | | | | |

Table 3: Test Case for Account Recovery

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Bachelor House Ranting System | | | Test Designed by: Safwan Bin Rahman | | |
| Test Case ID: B\_3 | | | Test Designed date: 11.11.2022 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Account Recovery | | | Test Execution date: | | |
| Test Title: allow user to recovery his account if he forgets his password | | | | | |
| Description: Test account recovery option. | | | | | |
| Precondition (If any): User must remember his email | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| * Go to the website * Enter forgot password button * Fill all info box. * Click submit. | (i)Username: taba  Password: 1111 | System will send code to recovery the account | | As expected, | Pass |
| Post Condition: Recovery info will be stored in database. | | | | | |

Table 4: Test Case for Finding House

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Bachelor House Ranting System | | | Test Designed by: Safwan Bin Rahman | | |
| Test Case ID: B\_4 | | | Test Designed date: 11.11.2022 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Finding House | | | Test Execution date: | | |
| Test Title: Show available flat according to the given location and also highlighted info about the place | | | | | |
| Description: Test find house option. | | | | | |
| Precondition (If any): User must have login. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| * Go to the website * login * Click on find places * Give the location * If like any flat send a request to rent this . | Username: taba123  Password: 111  Login successful  Location: Showing result around 60 feet | System will show available Flat and their location. | | As expected, | Pass |
| Post Condition: Searching info will be stored in database. | | | | | |

Table 5: Test Case for Add Available room/house

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Bachelor House Ranting System | | | Test Designed by: Safwan Bin Rahman | | |
| Test Case ID: B\_5 | | | Test Designed date: 11.11.2022 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Add room/ house | | | Test Execution date: | | |
| Test Title: Add new apartment with user name, user email, user cell no, location, Apartment Building No., rent | | | | | |
| Description: Test the website Add new apartment page | | | | | |
| Precondition (If any): User must have login. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| Go to Add new apartment page  Enter the Name  Enter the Email  Enter the Cell No  Enter the Location  Enter the Apartment Building No  Enter the price of rent of the apartment or room | Name: Nusrat Tabassum  Email: taba@gmail.com  Cell No: 01879617559  Location: Badda  Apartment Building No: 12A  Rent: 10000 | The user should add a new apartment/room to the system. | | As expected, | Pass |
| Post Condition: The apartment/room is validated with the database and successfully add a new apartment/room. The apartment details are saved in the database. | | | | | |

Table 3: Test Case for Usability

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Bachelor House Ranting System | | | Test Designed by:Irfan Alam | | |
| Test Case ID: B\_6 | | | Test Designed date: 11.12.2022 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Usability | | | Test Execution date: | | |
| Test Title: Usability of Website | | | | | |
| Description: Test websites usability | | | | | |
| Precondition (If any): N/A | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| Open website and  check the user friendliness of the website | System will performance its best as customer services. | System should be user friendly for all users. | | As expected, | Pass |
| Post Condition: System will become user friendly | | | | | |

Table 7: Test Case for Reliability

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Bachelor House Ranting System. | | | Test Designed by:Irfan Alam | | |
| Test Case ID: B\_7 | | | Test Designed date: 11.12.2022 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Reliability | | | Test Execution date: | | |
| Test Title: Reliability of Website | | | | | |
| Description: Test websites Reliability | | | | | |
| Precondition (If any): N/A | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| Open website and  check the system reliability | System will perform its intended functions with required precession | System should be user gets the correct output. | | As expected, | Pass |
| Post Condition: System will become reliable | | | | | |

Table 8: Test Case for Flexibility

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Bachelor House Ranting System. | | | Test Designed by:Irfan Alam | | |
| Test Case ID: B\_8 | | | Test Designed date: 11.2.2023 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Flexibility | | | Test Execution date: | | |
| Test Title: Flexibility of Website | | | | | |
| Description: Test websites Flexibility | | | | | |
| Precondition (If any): N/A | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| Open website and  check the system flexibility | System will add code easily and upgrade new features | System should be more flexible | | As expected, | Pass |
| Post Condition: System will become flexible | | | | | |

Table 9: Test Case for Security

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Bachelor House Ranting System. | | | Test Designed by:Irfan Alam | | |
| Test Case ID: B\_9 | | | Test Designed date: 11.2.2023 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Security | | | Test Execution date: | | |
| Test Title: Security of Website | | | | | |
| Description: Test websites Security | | | | | |
| Precondition (If any): N/A | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| Check the security of the system | Make the system more secure | System should be more secured | | As expected, | Pass |
| Post Condition: System will become secured | | | | | |

Table 10: Test Case for Maintainability

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Bachelor House Ranting System. | | | Test Designed by:Irfan Alam | | |
| Test Case ID: B\_10 | | | Test Designed date: 11.2.2023 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: Maintainability | | | Test Execution date: | | |
| Test Title: Maintainability of Website | | | | | |
| Description: Test websites: Maintainability | | | | | |
| Precondition (If any): N/A | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| Check the Maintainability of the website | Make the system more maintainable | System should be updated, improved, and changed easily over time without introducing bugs or having undesirable side effects | | As expected, | Pass |
| Post Condition: System will become maintainable | | | | | |

# ITEM PASS/FAIL CRITERIA

In our project assessment, we use a simple PASS/FAIL system to measure its performance. Units or systems that score between 94% and 98% are considered a PASS, showing successful implementation at a high standard of functionality and quality.

On the other hand, any unit or system scoring below 80% is labeled as a FAIL, pointing out areas that need immediate attention and improvement. Following these clear criteria ensures the delivery of a strong and dependable system that meets user expectations and industry standards.

By using this method, we show our dedication to creating a reliable and user-friendly project. Our thorough evaluation ensures that the final product aligns with our quality goals, providing a positive experience for our users.

# TEST DELIVERABLES

During the testing process, the following documents and materials will be delivered:

1. **Test Plan:** Serves as a roadmap for the testing process, guiding the team on how testing activities will be conducted.
2. **Test Cases:** Guides testers through the steps to be executed during testing, ensuring comprehensive coverage of functionalities.
3. **Test Scripts:** Automated scripts for repetitive testing tasks.
4. **Defect Reports:** Documentation of identified issues, including their severity and steps to reproduce.
5. **Test Summary Report:** Comprehensive overview of testing activities, including results and recommendations.
6. **User Manuals:** Guides for end-users on how to use the software.

# STAFFING AND TRAINING NEEDS

For the project's distribution and stages, it's suggested to have at least one dedicated inspector. This person will spend some time at the beginning of the project, and then be fully involved about six months later. If there's no dedicated tester available, the project or test manager will do the job. It's important for them to be well-prepared on relevant topics. The project's personnel planning is detailed, with specific research tasks for most team members, as explained in the responsibilities section.

Developers and testers need to be good at JavaScript, Php, and MySQL. Automation testers should know relevant tools well. Basic EDI interface operations training is needed for developers and testers. The operational staff will need thorough training on the EDI communications process before the project is finally accepted.

Sales administration personnel must be trained on new screens and reports so they can adapt well to the changes and new functions in the project.

# RESPONSIBILITIES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name of Responsibility | Team Manager | Project  Manager | Dev. Team | Test Team | Client |
| Acceptance test Documentation & Execution | Yes | Yes | No | Yes | Yes |
| System/Integration test Documentation & Execution | Yes | No | Yes | Yes | No |
| Unit test documentation & execution | Yes |  | Yes | Yes | No |
| System Design Reviews | Yes | Yes | Yes | Yes | Yes |
| Detail Design Reviews | Yes | Yes | Yes | Yes | No |
| Test Procedures and rules | Yes | Yes | Yes | Yes | No |
| Screen & Report Prototype reviews | No | No | Yes | Yes | Yes |
| Change control and Regression testing | Yes | Yes | Yes | Yes | Yes |

# TESTING SCHEDULE

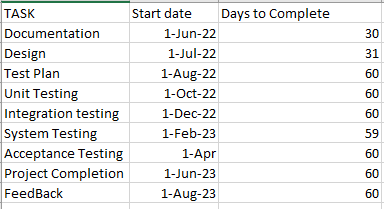


Figure 7: Data

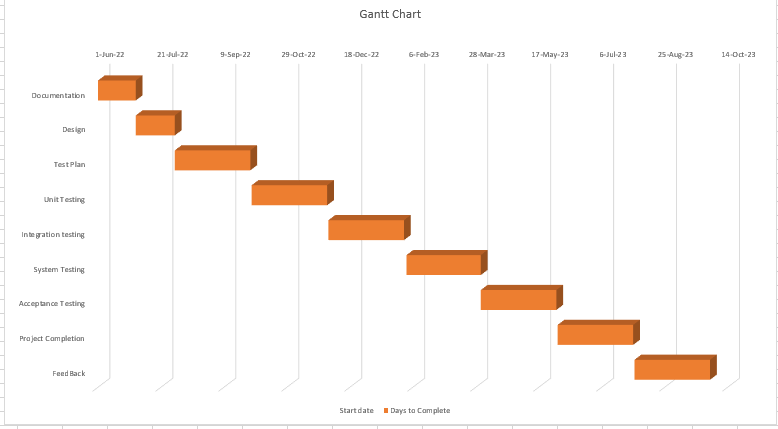


Figure 8 : Gantt Chart

# PLANNING RISKS AND CONTINGENCIES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S/N** | **Risk Description** | **Probability** | **Impact** | **Mitigation Plan** |
| 1 | Unrealistic time estimate | 40% | Delay project 2 weeks | Take multiple estimation |
| 2 | Account security | 20% | Moderate | Allow users to enter 5 times to  enter incorrect password. |
| 3 | Online Payment Security | 30% | Significant | Allow users to enter 3 times to  enter incorrect pin. |
| 4 | Unable to acquire required hardware for  testing | 5% | Low | Start testing after making sure all the required hardware are  available. |
| 5 | Exceeding budget | 60% | High | Take some extra money from  client for safety. |
| 6 | Resource Constraints | 50% | HIGH | Regularly monitor resource usage and allocate additional resources if needed. Prioritize and optimize test cases based on available resources. |
| 7 | Communication Breakdown | 30% | Moderate | Establish clear communication channels and protocols. Schedule regular project meetings to discuss progress, challenges, and updates. Use collaboration tools to facilitate communication among team members. |

# APROVALS

|  |  |
| --- | --- |
| Project Sponsor | Approved |
| Development Management | Approved |
| EDI Project Manager | Approved |
| RS Test Manager | Approved |
| RS Development Team Manager | Approved |