

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

**RentPro (Rental Management System)**

A Software Requirement Engineering Project Submitted

By

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Semester: Summer\_22\_23** | | | **Section: B** | **Group No:5** |
| **SN** | **Student Name** | **Student ID** | Individual  Contribution (in %) | Total Marks: 50 |
| Earned Marks: |
| 31 | Pranta Das | 20-43888-2 | 40% |  |
| 32 | Shidhartha Chakrabarty Turzo | 20-43891-2 | 40% |  |
| 09 | Muhammad Abdullah | 20-42322-1 | 20% |  |
|  |  |  |  |  |

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

RentPro

Version 1.0 approved

Prepared By: Pranta Das, Shidhartha Chakrabarty Turzo,Muhammad Abdullah

American International University – Bangladesh (AIUB)

25 August, 2023

Table of Contents

[Revision History 3](#_Toc144065503)

[1. TEST PLAN IDENTIFIER: 4A-2CK 4](#_Toc144065504)

[2. REFERENCE MATERIALS 4](#_Toc144065505)

[3. INTRODUCTION 4](#_Toc144065506)

[3.1 Background to the Problem 4](#_Toc144065507)

[3.2 Solution to the Problem 5](#_Toc144065509)

[4. REQUEIREMNT SPECIFICATION 6](#_Toc144065510)

[4.1 System Features 6](#_Toc144065511)

[4.2 System Quality Attribute 8](#_Toc144065512)

[4.3 System Interface: 9](#_Toc144065513)

[4.4 Project Requirements 12](#_Toc144065514)

[5 FEATURES NOT TO BE TESTED 13](#_Toc144065515)

[6 TESTING APPROACH 13](#_Toc144065516)

[6.1 Testing Levels 13](#_Toc144065517)

[6.2 Test Tools 16](#_Toc144065518)

[6.3 Meetings 16](#_Toc144065519)

[7. TEST CASES/TEST ITEMS 18](#_Toc144065520)

[Test case for Registration: 20](#_Toc144065521)

[Test case for Owner Post: 21](#_Toc144065522)

[Test case for Profile update: 22](#_Toc144065523)

[Test case for Update Owner Post: 23](#_Toc144065524)

[Test case for Apartment Search: 24](#_Toc144065525)

[Test Case for Message Session: 25](#_Toc144065526)

[8.ITEM PASS/FAIL CRITERIA 27](#_Toc144065527)

[9.TEST DELIVERABLES 27](#_Toc144065528)

[10.STAFFING AND TRAINING NEEDS 27](#_Toc144065529)

[11.RESPONSIBILITIES 29](#_Toc144065530)

[12.TESTING SCHEDULE 29](#_Toc144065531)

[13.PLANNING RISKS AND CONTINGENCIES 30](#_Toc144065532)

[14.APROVALS 30](#_Toc144065533)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Updated by** | **Update Comments** |
| 0.1 | 20023/05/01 | Pranta Das | First Draft |
| 0.2 | 20023/05/16 | Shidhartha Chakrabarty | Second Draft |
| 0.3 | 20023/05/29 | Shidhartha Chakrabarty | Third Draft |
| 0.4 | 20023/06/10 | Pranta Das | Fourth Draft |
| 0.5 | 20023/06/25 | Pranta Das | Fifth Draft |
| 0.6 | 20023/07/04 | Pranta Das | Sixth Draft |
| 0.7 | 20023/07/15 | Shidhartha Chakrabarty | Seventh Draft |
| 0.8 | 20023/07/27 | Pranta Das | Eighth Draft |
| 0.9 | 20023/08/06 | Muhammad Abdullah | Ninth Draft |
| 1.0 | 20023/08/16 | Shidhartha Chakrabarty | Tenth Draft |

# TEST PLAN IDENTIFIER: 4A-2CK

# REFERENCE MATERIALS

* <https://t4tutorials.com/test-cases-generation-of-real-estate-management-system-project/>
* <http://www.tutorialspoint.com>
* <https://www.softwaretestinghelp.com>
* <https://blog.qasource.com>

# INTRODUCTION

## Background to the Problem

## RentPro is an application that helps users find rental properties suitable for bachelors, families, and hostels. It allows owners to list their properties and users to search, negotiate, and select their desired houses. The application provides real-time location tracking, online transactions through bkash app, and various other benefits.

In the traditional rental landscape, managing rentals has often been a cumbersome and fragmented process. Multiple systems, spreadsheets, and manual paperwork have led to issues such as double bookings, missed rental opportunities, inaccurate billing, and limited visibility into rental performance. This fragmented approach not only wastes time and resources but also hampers overall customer satisfaction.

* Lack of Centralization: Absence of a unified system results in disjointed processes and inaccurate data.
* Manual Workflows: Relying on manual data entry leads to errors and inefficiencies.
* Limited Insights: Inadequate systems hinder data-driven decisions and trend analysis.
* Customer Impact: Poor coordination negatively affects customer experience and business reputation.

Efficient rental management improves operations, enhances customer experiences, offers a competitive edge, and enables data-driven decisions. Rentpro addresses these concerns, offering streamlined solutions for effective rental management.

## Solution to the Problem

* Centralized Rental Platform: Develop a comprehensive software platform like Rentpro that centralizes all rental-related activities. This solution consolidates inventory, schedules, billing, and reporting in one system, reducing errors and improving coordination.
* Automation and Integration: Implement automated processes that seamlessly integrate with existing systems, reducing manual data entry and ensuring accurate data transmission between departments.
* Real-time Insights: Incorporate data analytics tools to provide real-time insights into rental performance, enabling data-driven decisions and trend analysis.
* Customer Self-Service Portal: Create a customer portal where clients can view available rentals, make bookings, track orders, and manage payments, enhancing customer convenience.

The proposed solutions are appropriate because they directly address the root causes of the problem. They eliminate manual processes, centralize information, offer real-time insights, and improve customer interactions. By automating tasks and providing a unified platform, these solutions streamline operations, reduce errors, and enhance customer satisfaction.

The feasibility of the solutions largely depends on the technical capabilities of the development team and the resources available. Modern technology stacks and cloud-based solutions make building such systems feasible. Integration with existing software may require careful planning and cooperation between different departments.

Rentpro is an all-in-one rental management software designed to simplify and streamline the rental process. Its purpose is to provide businesses with a comprehensive platform to manage inventory, scheduling, billing, and reporting in one place. The software's benefits include increased operational efficiency, reduced errors, enhanced customer experiences, and data-driven decision-making. The objectives are to centralize rental processes, automate manual tasks, and improve overall rental management.

Existing software solutions in the rental management space include:

* Rentman: A rental management software that offers tools for managing inventory, scheduling, and customer interactions.
* InTempo: Provides rental and inventory management solutions for equipment rental businesses.
* Booqable: A cloud-based rental software designed to streamline the rental process, from booking to return.

These solutions aim to address similar challenges by centralizing rental operations and providing automation features, much like Rentpro. However, Rentpro distinguishes itself by offering a comprehensive solution covering multiple industries and providing advanced analytics capabilities.

# REQUEIREMNT SPECIFICATION

## System Features

1. **Owner Login Session**   
   **Functional Requirements**
   1. The software shall allow the owner to login with their given username and password.
   2. If the username and/or password has been inserted wrong more than three times, the random verification code will be generated by the system to retry login.

Priority Level: High  
Precondition: Owner must have valid username and password.

Cross reference: 3.3, 4, 5, 6, 8, 9

1. **Test case for Tenant:**

**Functional Requirements**

* 1. The software shall allow the tenant to login with their given username and password.
  2. If the username and/or password has been inserted wrong more than three times, the random verification code will be generated by the system to retry login.

Priority Level: High  
Precondition: Tenant must have valid username and password.

Cross reference: 3.3, 5, 7, 8, 9

1. **Registration**

**Functional Requirements**

* 1. The software shall allow the user to put email, phone number, National id, Gender, Select user category (owner or tenant), username, password, date of birth.
  2. If the registration is successful, the login page will be displayed.
  3. If the registration isn’t successful, the system will provide a message on where changes are a must.
  4. If user select owner, the account session details are stored in the owner database. If User select tenant, the account session details are stored in the tenant database.

Priority Level: High  
Precondition: Enter the name, e-mail, phone number, and all the necessary details.

Cross reference: 1,2

1. **Create Post Session**

**Functional Requirements**

* 1. The software shall allow the owner to put details of apartment, attach picture and video, mobile number, location, set price and criteria.
  2. Fill all the criteria with valid credential.

Priority Level: Medium  
Precondition: User must be Owner. Users must have a valid username and password.

Cross reference: 1

1. **Profile Update Session**

**Functional Requirements**

* 1. The software shall allow the user to put updated email, updated phone number, updated national id, updated gender, updated select user category (owner or tenant), updated username, updated password, updated date of birth.
  2. If the user is owner then login in owner section. If user is tenant then login tenant section

Priority Level: Medium  
Precondition: User must have valid username and password

Cross reference: 1,2

1. **Update Feed Post Session**

**Functional Requirements**

* 1. The software shall allow the owner to put details of updated apartment, attach updated picture and video, updated mobile number, updated location, updated set price and criteria.
  2. Fill all the criteria with valid credential.

Priority Level: Medium  
Precondition: User must be Owner. Users must have a valid username and password.

Cross reference: 1

1. **Apartment Search by place**

**Functional Requirements**

* 1. The software shall allow the tenant to select, buy or rent an apartment.
  2. The software shall allow the tenant to enter location, property type, price range
  3. Fill all the criteria with valid credentials and tenant will see all recommended apartment within expected location and the price range.

Priority Level: High  
Precondition: User must be Tenant. Users must have a valid username and password.

Cross reference: 2

1. **Message Session**

**Functional Requirements**

* 1. Tenant can select apartment if it is available and within price range then he/she chat through the webpage with owner.
  2. Once an appointment is scheduled, both owner and tenant will receive notifications and reminders, ensuring timely and efficient communication.

Priority Level: Low  
Precondition: User must have valid username and password.

Cross reference: 1, 2

1. **Payment Session**

**Functional Requirements**

* 1. To facilitate seamless transactions between users and lawyers, the platform will integrate a secure online payment system.
  2. Users can make secure online payments for legal services directly through the platform.

Priority Level: Low  
 Precondition: User must have valid username and password.

Cross reference: 1, 2

## 4.2 System Quality Attribute

**QA1 - Usability:** A user should be able to search, book, and manage rentals with minimal training within 5 minutes.

**Priority level:** High

**Precondition:** N/A

**Cross-reference:** QA2

**QA2 - Portability**: The software must be capable of running in any platform or operating system. Such as Windows, Linux, Android, Apple, and so on.

**Priority level:** High

**Precondition:** N/A

**Cross-reference:** QA2

**QA3 - Performance:** A user can easily within few seconds access the upcoming interfaces right after providing the information.

**Priority level:** High

**Precondition:** N/A

**Cross-reference:** QA1, QA4

**QA4 - Reusability**: The software’s function should be created in such way that those function can be used in any other software.

**Priority level:** Medium

**Precondition:** N/A

**Cross-reference:** QA1, QA4

**QA5 – Availability:** The system should have a 99% uptime rate, minimizing downtime for maintenance.

**Priority level:** High

**Precondition:** N/A

**Cross-reference:** N/A

**QA4. Integrity:** Only admin and editor have access to view user’s information.

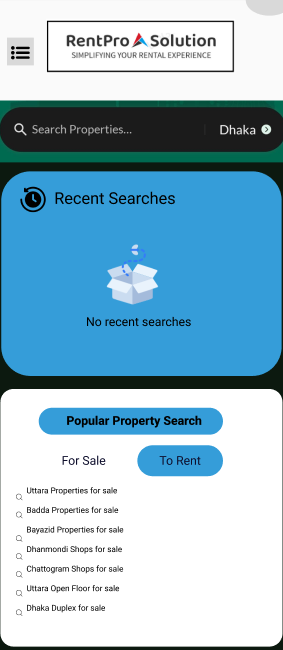
**Priority level:** High

**Precondition:** User must register him/herself.

**Cross-reference:** QA2

## System Interface:

Owner feature:

 A screenshot of a phone

Description automatically generated

A screenshot of a phone login

Description automatically generatedA screenshot of a blue box

Description automatically generated

Screens screenshot of a web page

Description automatically generated

User feature

Screens screenshot of a black and green screen

Description automatically generated

A screenshot of a phone

Description automatically generated

Payment Feature:

A screenshot of a phone

Description automatically generated

## Project Requirements

* The project must be completed within 6 months.
* The budget for the project is $200,000.
* The development team consists of 2 front-end developers, 2 back-end developers, and 1 UI/UX designer.
* The system should be compatible with major web browsers (Chrome, Firefox, Safari).
* The system should be hosted on a cloud server with scalable infrastructure.

# FEATURES NOT TO BE TESTED

The following areas will not undergo specific testing:

* Cross-platform portability encompassing Android and iOS.
* The functionality for logging out across multiple devices.
* Device identification and verification for potential password-free logins.

# TESTING APPROACH

## Testing Levels

* **UNIT Testing:** During our unit testing, we conducted tests on individual software module and check for any errors. This testing will be conducted by the developer and will be approved by the development team leader. The goal is to verify that each unit of code functions as intended.
* **Integration Testing:** After unit testing, we do the integration testing. Unit testing uses modules for testing purposes, and these modules are combined and tested in integration testing. The Software is developed with several software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules. We used Bottom-Up technique in our software.

**Purpose:** The purpose of this integration testing plan is to verify that the different components of the apartment hiring website are integrated properly and that they work together as expected.

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

* User account creation
* User login and signup
* Search apartments by type
* See apartment status/profile.
* See availability of apartments.
* Live messaging between owner and renter.
* Attach documents or photos to help the consultation.
* Owner updates his valid profile for apartment.
* The owner rejects an appointment with valid reason.
* Owner set their own apartment rents.
* Owner set their payment method.
* The owner checks their earnings reports.
* Payment withdrawals for Owner
* User payment

**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 email address and password.
* TC-5: Verify that users cannot login with an invalid email address or password.
* TC-6: Verify that users can sign up with a valid email address and password.
* TC-7: Verify that users can search for apartment by type.
* TC-8: Verify that users can see a list of apartments that match their search criteria.
* TC-9: Verify that users can see the apartment profiles, including their location, quality, rent amount and availability.
* TC-10: Verify that users can see the apartment status, such as whether available or not.
* TC-11: Verify that users can see the owner’s profile, including their past tenants review and availability.
* TC-12: Verify that users can see if the owner is available for online live messaging.
* TC-13: Verify that users can text the owner to book an appointment to check apartment for live inspections and queries.
* TC-14: Verify that the owner receives the text and can communicate with the users.
* TC-15: Verify that users can attach documents or photos to check apartment as per their requirements.
* TC-16: Verify that the owner can view the documents or photos and use them to help with the tenants.
* TC-17: Verify that users receive reminders for their appointments.
* TC-18: Verify that the reminders include the time, date, and location of the appointment.
* TC-19: Verify that users can cancel their appointments.

**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.
* 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 requirement.

* **System Testing**: After completing integration testing, we proceed with system testing, where we thoroughly tested the fully integrated system to ensure it meets all the specified requirements. This testing has been done 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.

**User Acceptance Testing (UAT) Plan**

**Purpose:** The purpose of this user acceptance testing (UAT) plan is to verify that the owner is renting system meets the requirements of the users and that it is ready for deployment.

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

* + User account creation
  + User login and signup
  + Search apartment by type
  + See apartments status/profile.
  + See availability of apartments.
  + Live messaging between owner and renter.
  + Attach documents or photos to help the consultation.
  + Owner updates his valid profile for apartment.
  + owner rejects an appointment with valid reason.
  + set their own apartment rents.
  + set their payment method.
  + The owner checks their earnings reports.
  + Payment withdrawals for Owner
  + User payment

**Testers:** The testers for this UAT plan will be a group of users who have been selected to represent the different types of users who will be using the system.

**Test Data:** The test data for this UAT plan will be provided by the project team.

**Procedures:** The following procedures will be followed during the UAT phase:

* + The testers will be given a training session on how to use the system.
  + The testers will then test the system using the test cases that have been developed.
  + The testers will report any defects that they find to the project team.
  + The project team will fix the defects and then the testers will retest the system.
  + This process will continue until all the defects have been fixed and the testers are satisfied with the system.

**Acceptance Criteria:** The system will be considered to have passed the UAT phase if it meets the following criteria:

* + The system must meet the requirements of the users.
  + The system must be free of defects.
  + The system must be easy to use and navigate.
  + The system must be reliable and available 24/7.

**Entry criteria:**

* All integration testing has been completed and all defects have been fixed.
* The system is ready for use by users.
* The system documentation is complete and up to date.
* The test environment is ready for testing.
* The data for testing has been created.

**Exit criteria:**

* All system tests have been executed and passed.
* All defects found during system testing have been fixed.
* The system is ready for user acceptance testing.
* **Acceptance Testing:** Then we did 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.

## Test Tools

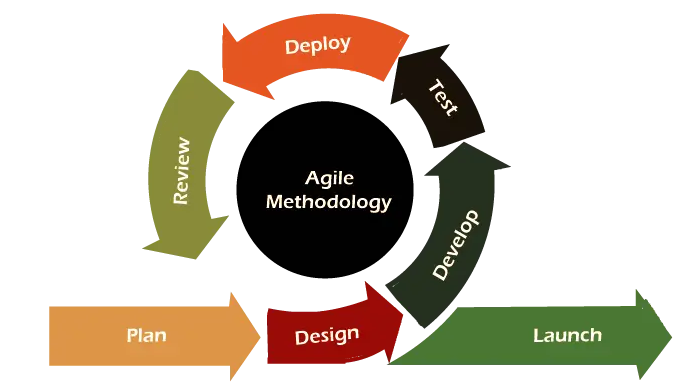
* **Selenium Web driver:** Selenium Web driver is an open-source collection of APIs which is used for testing web applications. The Selenium Web driver tool is used for automating web application testing to verify that it works as expected or not. It mainly supports browsers like Firefox, Chrome, Safari, and Internet Explorer. It also permits you to execute cross-browser testing.
* **NUnit**: This is a unit testing framework for .NET applications. It allows you to write tests in C#, VB.NET, or other .NET languages.
* **Excel:** Excel is a great tool for creating timeline charts.
* **Integrated Development Environment (IDE):** Visual Studio Code
* **Version Control:** Git, GitHub
* **Database Management:** MySQL Postgre SQL (pgAdmin), Workbench

## Meetings

The testing team will hold a weekly meeting to assess the status of the project, pinpoint emerging issues, and detect potential problems at an early stage. They will also hold a biweekly meeting with the development team and project manager to discuss the overall progress of the project and to ensure that the testing team and development team are working together effectively. In urgent cases, extra meetings may be arranged.

Scrum model plan:

1. Sprint Planning Meeting: This is a meeting that occurs at the beginning of each sprint, which is a time-boxed period (usually 2-4 weeks) during which development work takes place. In the context of RentPro, the team would gather to discuss and plan the upcoming sprint's goals and tasks. The Product Owner would present the prioritized user stories from the product backlog, and the development team would estimate the effort required for each one. Together, they would select the user stories to be included in the sprint based on capacity and priority.
2. Daily Stand-up (Daily Scrum): This is a brief daily meeting where the development team members provide updates on their progress, discuss any challenges they're facing, and coordinate their work. In the context of RentPro, team members might share what they accomplished since the last daily stand-up, what they plan to work on next, and if they're encountering any obstacles. The focus is on quick communication and removing any impediments to progress.
3. Sprint Review Meeting: At the end of each sprint, the team holds a sprint review meeting to showcase the completed work to stakeholders, including the Product Owner and potentially other members of the organization. In the context of RentPro, the team would present the features, improvements, or fixes that were completed during the sprint. This allows stakeholders to provide feedback and make any necessary adjustments to the product backlog.
4. Sprint Retrospective Meeting: Also held at the end of each sprint, the retrospective meeting is an opportunity for the team to reflect on their processes and identify ways to improve. In the context of RentPro, the team would discuss what went well during the sprint, what didn't go well, and what improvements can be made in terms of processes, communication, and collaboration. The aim is continuous improvement.
5. Backlog Refinement (Grooming) Meetings: These meetings are held throughout the project to refine and clarify items in the product backlog. In the context of RentPro, the team and the Product Owner would review and prioritize user stories, break them down into smaller tasks, and ensure that the backlog items are well-defined and ready for inclusion in future sprints.



# TEST CASES/TEST ITEMS

**Test case for Login:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Rent Pro | | | Test Designed by: Pranta Das | | |
| Test Case ID: R\_P 1 | | | Test Designed date: 12/08/2023 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: **Owner Login Session** | | | Test Execution date: | | |
| Test Title: verify login with valid username and password | | |  | | |
| Description: Test website login page | | |  | | |
| Precondition (If any): Owner must have valid username and password | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Go to the website. 2. Enter username. 3. Enter password. 4. Click submit | Username: Pranta123  Password: O1234 | User should login into the application | |  |  |
| Post Condition: User is validated with owner database and successfully login to account. The account session details are logged in the owner database. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: Rent Pro | | | Test Designed by: Pranta Das | | |
| Test Case ID: R\_P 2 | | | Test Designed date: 12/08/2023 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: **Tenant Login Session** | | | Test Execution date: | | |
| Test Title: verify login with valid username and password | | |  | | |
| Description: Test website login page | | |  | | |
| Precondition (If any): Tenant must have valid username and password | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Go to the website. 2. Enter username. 3. Enter password. 4. Click submit | Username: Sidartha123  Password: T1234 | User should login into the application | |  |  |
| Post Condition: User is validated with tenant database and successfully login to account. The account session details are logged in the tenant database. | | | | | |

**Test case for Tenant:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: RentPro | | | Test Designed by: Shidhartha Chakrabarty Turzo | | |
| Test Case ID: R\_P 3 | | | Test Designed date: 12/08/2023 | | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: | | |
| Module Name: **Go to** **Registration** | | | Test Execution date: | | |
| Test Title: verify login with valid username and password | | |  | | |
| Description: Test website registration page | | |  | | |
| Precondition (If any): Enter the name, e-mail, phone number, and all the necessary details. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| * 1. Go to the website’s registration page.   2. Click Signup Button   3. Enter e-mail.   4. Enter phone number.   5. Enter National id.   6. Select Gender   7. Select Owner or Tenant   8. Enter username.   9. Enter password.   10. Confirm Password   11. Enter Date of Birth   12. Click submit | * Name: Pranta Das * Email: [daspranta7@gmail.com](mailto:daspranta7@gmail.com) * Phone: 01863618432 * NID:87689936279 * Gender: Male * User: Owner * Username: Pranta123 * Password: O1234 * Confirm password: O1234 * Dob: 16-07-2000 | User should register himself/herself into the application. | |  |  |
| Post Condition: User (Owner) is validated with database and successfully registered an account. The account session details are stored in the owner database. If User select tenant, the account session details are stored in the tenant database | | | | | |

# Test case for Registration:

# Test case for Owner Post:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: RentPro | | | Test Designed by: Pranta Das | | |
| Test Case ID: R\_P 4 | | | Test Designed date: 12/08/2023 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: | | |
| Module Name: **Create Post Session** | | | Test Execution date: | | |
| Test Title: Create post | | |  | | |
| Description: Test website create post page | | |  | | |
| Precondition (If any): User must be Owner. Users must have a valid username and password. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Go to the website. 2. Click Create Post. 3. Enter Details of Apartment. 4. Attach picture and video. 5. Set Mobile 6. Set location. 7. Set Rent price and Criteria. 8. Click submit | Details: 1300 sqft Apartment, 3 bedrooms with Attached bathroom, drawing room, dining room, kitchen.  Add: picture and video of the apartment.  Mobile: 01864321278  Location: Bashundhara R/A, Block C Road 11, Dhaka.  Price: 32,0000  Criteria: Only for family | Owner should create a post into the application | |  |  |
| Post Condition: User is validated with database and successfully create a post. | | | | | |

# Test case for Profile update:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: RentPro | | | Test Designed by: Pranta Das | | |
| Test Case ID: R\_P 5 | | | Test Designed date: 12/08/2023 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: | | |
| Module Name: **Profile Update Session** | | | Test Execution date: | | |
| Test Title: Update user profile with valid username, email, password, and other information | | |  | | |
| Description: Test website Profile Update page | | |  | | |
| Precondition (If any): User must have valid username and password | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Go to the website. 2. If the user is owner then login in owner section. If user is tenant then login tenant section 3. Click Edit Profile Button 4. Enter updated e-mail. 5. Enter updated phone number. 6. Enter updated National id. 7. Select Gender 8. Select Owner or Tenant 9. Enter updated username. 10. Enter updated password. 11. Confirm updated password 12. Enter updated Date of Birth 13. Click submit | * Name: Pranta Das * Email: [daspranta7@gmail.com](mailto:daspranta7@gmail.com) * Phone: 01863618432 * NID:87689936279 * Gender: Male * User: Owner * Password: O1234 * Confirm password: O1234 * Dob: 16-07-2000 | User should update his/her profile of the application | |  |  |
| Post Condition: User is validated with database and successfully update its profile. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: RentPro | | | Test Designed by: Pranta Das | | |
| Test Case ID: R\_P 6 | | | Test Designed date: 12/08/2028 | | |
| Test Priority (Low, Medium, High): Medium | | | Test Executed by: | | |
| Module Name: **Update Feed Post Session** | | | Test Execution date: | | |
| Test Title: Update post with title and details | | |  | | |
| Description: Test website update post page | | |  | | |
| Precondition (If any): User must be Owner. Users must have a valid username and password. | | | | | |
| Test Steps | Test Data | Expected Results | | Actual Results | Status (Pass/Fail) |
| 1. Go to the website. 2. Click Create Post. 3. Click Update 4. Enter updated Details of Apartment. 5. Attach updated picture and video. 6. Set updated mobile num 7. Set updated location. 8. Set updated Rent price and Criteria. 9. Click submit | Details: 1300 sqft Apartment, 3 bedrooms with Attached bathroom, drawing room, dining room, kitchen.  Add: picture and video of the apartment.  Mobile: 01864321278  Location: Bashundhara R/A, Block C Road 11, Dhaka.  Price: 32,0000  Criteria: Only for family | User should update a post into the application | |  |  |
| Post Condition: User is validated with database and successfully update a post. | | | | | |

# Test case for Update Owner Post:

# Test case for Apartment Search:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Name: RentPro | | | | Test Designed by: Shidhartha Chakrabarty Turzo |
| Test Case ID: R\_P 7 | | | | Test Designed date: 12/08/23 |
| Test Priority (Low, Medium, High): High | | | | Test Executed by: |
| Module Name: **Apartment Search by place** | | | | Test Execution date: |
| Test Title: verify that the search button is working or not | | | |  |
| Description: Test website search apartment | | | |  |
| Precondition (If any): User must be Tenant. Users must have a valid username and password. | | | | |
| Test Steps | Test Data | Expected Results | Actual  Results | Status  (Pass/Fail) |
| 1. Go to the website. 2. Select if user is tenant select buy or rent apartment. 3. Enter Location. 4. Select property type 5. Select price range 6. Click search now button. | I want to: Rent  Location: Bashundhara R/A  Property type: Apartment  Price Range: 30,000 | Tenants can see all recommended apartments within expected location and the price range. |  |  |
| Post Condition: User can search their desired apartment. | | | | |

# Test Case for Message Session:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: RentPro | | | Test Designed by: Shidhartha Chakrabarty Turzo | | |
| Test Case ID: R\_P 8 | | | Test Designed date: 12/08/23 | | |
| Test Priority (Low, Medium, High): Low | | | Test Executed by: | | |
| Module Name: **Message Session** | | | Test Execution date: | | |
| Test Title: press and confirm a time slot for Message Session | | | | | |
| Description: Test Message Session button | | | | | |
| Precondition (If any): User must have valid username and password | | | | | |
| Test Steps | Test Data | Expected Results | | Actual  Results | Status  (Pass/Fail) |
| 1. Go to the website 2. Search for desired Apartment 3. Click profile 4. Click live Chat button | Username: turzo123  Password: O1234 | User should able to click on Message Session button | |  |  |
| Post Condition: User is validated with database and successfully able to create a Message Session. This information is saved into the database. | | | | | |

**Test case for Payment:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Name: RentPro | | | Test Designed by: Pranta Das | | |
| Test Case ID: R\_P 9 | | | Test Designed date: 12/08/23 | | |
| Test Priority (Low, Medium, High): Low | | | Test Executed by: | | |
| Module Name: **Payment Session** | | | Test Execution date: | | |
| Test Title: now press pay button and confirm a payment | | | | | |
| Description: now Test pay button | | | | | |
| Precondition (If any): User must have valid username and password | | | | | |
| Test Steps | Test Data | Expected Results | | Actual  Results | Status  (Pass/Fail) |
| 1. Go to the website. 2. Search for desired Apartment 3. Click profile 4. Click consult button. 5. Click pay now button | Username: Pranta123  Password: O1234 | User should able to pay the required fee. | |  |  |
| Post Condition: User is validated with database and successfully create a free Messaging Session. This information is saved into database. | | | | | |

# 8.ITEM PASS/FAIL CRITERIA

If the program will reach around 80-85%, we will think this project is satisfied and as passed project.

# 9.TEST DELIVERABLES

• **Acceptance Test Plan**: The plan for acceptance testing encapsulates the acceptance criteria, test cases, goals, scope, methodology, allocated resources, and timeline. This comprehensive document outlines how the acceptance testing phase will unfold.

• **System/Integration Test Plan**: Within this plan, you'll find the criteria for system and integration testing, meticulously outlined test cases, objectives, the range of testing, chosen approach, resource allocation, and the projected timeline. It serves as a roadmap for executing these vital testing phases.

• **Screen Prototypes**: The prototypes of screens present a visual representation of the testing approach's layout and design. These blueprints give insight into the intended user experience and user interface elements.

• **Transmittal Reports for Test Items**: These are the documents provided by developers when handing over their completed work. These reports provide details about the completed items and their readiness for testing.

• **Test Logs**: These records encapsulate the outcomes of various tests conducted. They serve as a record of the test executions and the results obtained.

• **Incident Reports**: When unexpected outcomes or issues arise during testing, incident reports document these occurrences. These reports detail the discrepancies encountered and the context surrounding them.

• **Investigation Report Logs**: These reports summarize the incidents detailed in the incident reports. They offer a condensed overview of the issues, their origins, and any corrective actions taken.

• **Test Summary Report**: At the conclusion of testing, a test summary report is generated. This comprehensive report provides an overview of the entire testing process, encompassing the conducted tests, outcomes, issues encountered, resolutions, and a holistic evaluation of the software's performance.

In essence, these documents and reports collectively form the foundation for effective testing and ensure that the software undergoes a thorough and well-documented evaluation.

# 10.STAFFING AND TRAINING NEEDS

For Prototyping total of 80 hours is needed

For Development total of 860 hours is needed

For Revision total of 60 hours is needed

For testing and debugging total of 200 hours is needed

Total Working hours: 1210 hours

Daily working hours is 8 Hours per day with this total day’s need: 1200 hours / 8 Hours per day = 151.25 Hours or 20 weeks or 5 months in total

With three developers, three software testers with some custom build pc and some with smartphones it should take around 151.25 Hours.

For the evaluation, the individual will be granted a certain amount of time at the project's outset, followed by a period of approximately six months, during which they will be required to commit full-time. In case an alternate tester is unavailable, the project/test manager will assume responsibility. In order to encompass a comprehensive and relevant study, it is important to consider the following preparation-related subjects. The personnel for this project have already been selected well in advance. A significant portion of the team will be engaged in specific research duties, which are expounded upon extensively in the responsibilities section.

• The developers and testers will receive training in Java, C++, Dart, Flutter, and MySQL.

• The automation tester should acquire both the necessary knowledge and proficiency in operating the relevant tools.

COCOMO (CONSTRUCTIVE COST MODEL)

As our project type is Organic,

So, Effort = PM= Coefficient<Effort Factor>\*(SLOC/1000) ^P

= 2.4\*(7000/1000) ^1.05

= 18.52 working hours.

Development time = DM = 2.50\*(PM)^T

= 2.5\*(18.52) ^ 0.38

= 7.58 weeks days

Required number of people = ST = PM/DM

= 18.52/ 7.58

= 2.44

= 3 persons

# 11.RESPONSIBILITIES

A checklist with check marks

Description automatically generated

# 12.TESTING SCHEDULE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test/Week | 1 | 2 | 3 | 4 |
| Documentation |  |  |  |  |
| Design |  |  |  |  |
| Test Plan |  |  |  |  |
| Unit Testing |  |  |  |  |
| Integration Testing |  |  |  |  |
| System Testing |  |  |  |  |
| Acceptance Testing |  |  |  |  |
| Project Completion |  |  |  |  |
| Feedback |  |  |  |  |

# 13.PLANNING RISKS AND CONTINGENCIES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S/N** | **Risk Description** | **Probability** | **Impact** | **Mitigation Plan** |
| 1 | Unrealistic time estimate | 40% | Delay project 2 weeks | Take multiple estimation |
|  |  |  |  |  |
|  |  |  |  |  |

# 14.APROVALS

|  |  |
| --- | --- |
| Project Sponsor – Rofik sponsor |  |
| Development Management – Priyom manager |  |
| EDI Project Manager – Promit Project |  |
| RS Test Manager - Sabrina Tester |  |
| RS Development Manager - Samma Tester |  |
| Reassigned Sales - Jasmin Sales |  |
| Order Entry EDI Team Manager – Shourov Order |  |