|  |  |
| --- | --- |
|  | Test Plan |
|  |  |
|  | The A-Team  Off\_Campus\_Review |

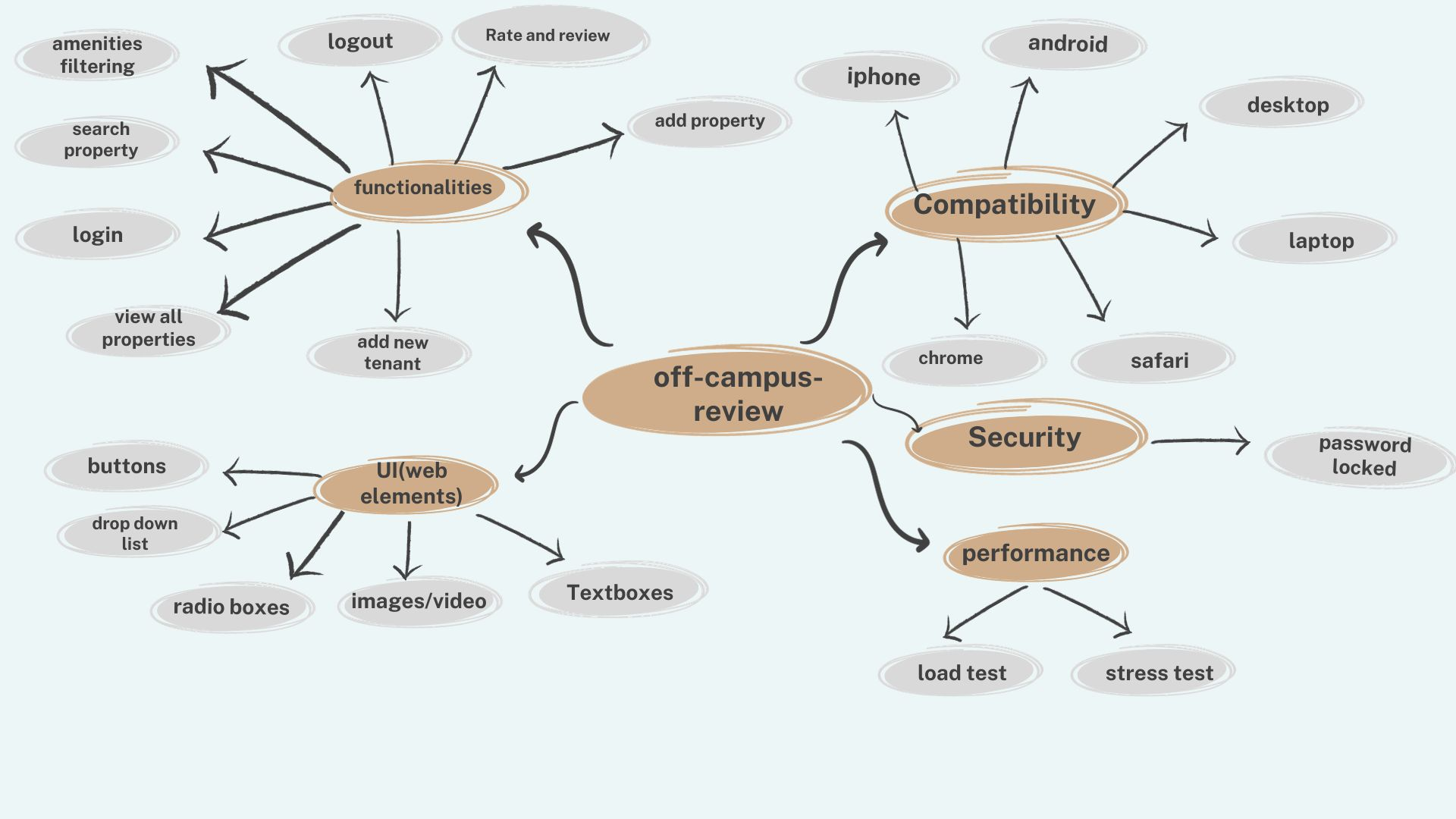
Table of Contents

[Test Plan version 1.0 2](#_Toc146929603)

[1. OVERVIEW 2](#_Toc146929604)

[1.1 Objectives and Tasks 3](#_Toc146929605)

[1.2 Mind Map of how the main tested functions 5](#_Toc146929606)

[ 5](#_Toc146929607)

[2. Testing Approach 5](#_Toc146929608)

[2.1 Unit Testing 5](#_Toc146929609)

[2.2 Integration Testing 5](#_Toc146929610)

[2.3 System Testing 5](#_Toc146929611)

[2.4 Usability Testing 5](#_Toc146929612)

[2.5 Performance Testing 5](#_Toc146929613)

[2.6 Compatibility Testing 5](#_Toc146929614)

[3. Requirements to be Testing 6](#_Toc146929615)

[3.1 TC1: User login functionality 6](#_Toc146929616)

[3.2 TC2: Usability Testing 6](#_Toc146929617)

[3.3 TC3: 10 Website Heuristic testing. 6](#_Toc146929618)

[3.4 TC4: Test the data creation, deletion ,retrieval, reporting and security functionality. 6](#_Toc146929619)

[4. Resource Planning 6](#_Toc146929620)

[4.1 Human Resource 6](#_Toc146929621)

[4.2. System Resource 7](#_Toc146929622)

[5. Test Scheduling and deliverables 7](#_Toc146929623)

[5.1. Project Milestones 7](#_Toc146929624)

[5.2 Testing Deliverables 7](#_Toc146929625)

[6. Risks, Limitations, Assumptions and Contingencies 8](#_Toc146929626)

[6.1 Risks and mitigations 8](#_Toc146929627)

[6.2. Limitations 11](#_Toc146929628)

[6.3 Assumptions 11](#_Toc146929629)

[6.4 Contingencies 11](#_Toc146929630)

[7. Defect Management Plan 11](#_Toc146929631)

[7.1. Identify Defects: 11](#_Toc146929632)

[7.2 Defect Reporting: 11](#_Toc146929633)

[7.3 Defect Triage: 11](#_Toc146929634)

[7.4 Defect Resolution: 12](#_Toc146929635)

[7.5 Defect Verification: 12](#_Toc146929636)

[7.6 Approval and Revision: 12](#_Toc146929637)

[8. Definition of test criteria 12](#_Toc146929638)

[8.1 Acceptance criteria 12](#_Toc146929639)

[8.2 Suspension criteria 12](#_Toc146929640)

[8.3 Exit Criteria 12](#_Toc146929641)

[9. Approvals 13](#_Toc146929642)

# Test Plan version 1.0

Date: 01/09/2023

## OVERVIEW

The purpose of this test plan is to verify that our Off\_Campus\_Review system’s website meets its design and system requirements and/or functionalities. This plan includes assessments of the website's functionalities, usability, and conformity to the specifications documented in the domain documentation. Testing will include both static and dynamic aspects with the main focus being on improving user experience.

* Who will use the system?

The system will be used by individuals who are looking to view rental accommodations around Makhanda, individuals who have stayed/staying in an accommodation around Makhanda and looking to review and rate the accommodations and agents or the landlords, and lastly will be used by rental agents to advertise their accommodations and rent the tenants.

* What is it used for?

The system is mainly used to rate and review rental accommodations, agents and tenants.

### **1.1 Objectives and Tasks**

**Objectives: For a clear roadmap for the testing process and help ensure that all of the testing aspects are well-defined and executed effectively**

* **Software Identification:** Identifying the software under evaluation and associated information.
* **Requirements Determination:** Identify and enumerate the specific requirements earmarked for testing. Ensure that each requirement is thoroughly documented and can be traced back accurately.
* **Test Strategy Formulation:** Establish the strategies and methodologies that will be employed during testing. This includes delineating the testing levels (such as unit, integration, system) and the types of tests (e.g., functional, performance).
* **Resource Allocation**: Allocate the essential resources for testing, encompassing both human resources (such as testers and developers) and system resources (comprising testing tools, hardware, and software).
* **Deliverables Specification**: Identify the expected outcomes that will result from the testing process. This encompasses the development of comprehensive test plans, detailed test cases, test reports, defect reports, and any other pertinent documents.
* **Risk Evaluation:** Evaluate the potential risks and challenges associated with the testing process and formulate strategies for managing and mitigating these risks.
* **Test Schedule Development**: Create timetables and schedules for the various testing phases, setting milestones and deadlines for the different testing activities.
* **Testing Environment Setup:** Define the prerequisites and configuration of the testing environment, ensuring its alignment with the production environment as closely as possible.
* **Defect Management Plan:** Institute a transparent process for reporting, tracking, prioritizing, rectifying, and retesting defects.
* **Communication Strategy:** Establish a communication strategy for reporting progress, issues, and test results to stakeholders, including regular updates on the status of the testing process.
* **Testing Criteria Specification**: Outline the criteria that must be met for the testing process to be considered complete. These criteria should include both qualitative and quantitative measures to ascertain the success of the testing.
* **Acceptance Criteria Definition:** Define the criteria for acceptance in each testing phase, specifying the conditions that must be satisfied for the software to progress to the subsequent testing phase or move into production.
* **Exit Criteria Identification:** Specify the conditions that must be met before the testing can be concluded for each testing phase and for the overall project.

**Tasks:** covers various aspects of testing, from test preparation to execution and reporting, helping ensure a comprehensive and effective testing process.

* **Test Plan Development:** Create a comprehensive test plan detailing the objectives, tasks, resource allocation, and project’s milestones, deliverables, assumptions and limitations.
* **Testing Approach Formulation**: Define the overarching testing strategy, encompassing the levels and approaches to be employed.
* **Test Data Preparation:** Generate or collect the necessary test data sets required to execute test cases and scenarios.
* **Test Environment Configuration:** Configure and maintain the test environment to closely mirror the production environment.
* **Test Case and Scenario Design:** Develop detailed test cases and scenarios that comprehensively cover various aspects of the software.
* **Test Case Evaluation**: Conduct peer reviews and step-by-step analyses of test cases and scenarios to ensure accuracy and completeness.
* **Test Execution:** Methodically execute test cases and scripts to verify the software's functionality.
* **Exploratory Testing:** Perform exploratory testing to identify unforeseen errors and assess usability.
* **Regression Testing**: Re-execute test cases to confirm that changes or fixes do not adversely impact existing functionality.
* **Performance Testing:** Assess software performance under varying load conditions.
* **Compatibility Testing:** Verify software compatibility with diverse platforms and configurations as well as responsiveness.
* **Usability Assessment:** Evaluate usability and collect user feedback regarding software interfaces.
* **Risk Monitoring:** Continuously monitor and evaluate potential risks and challenges throughout the testing process.
* **Defect Management and Tracking:** Establish and maintain a structured process for identifying, categorizing, prioritizing, and resolving defects.
* **Testing Progress Monitoring:** Continuously monitor testing progress, reporting on completed test cases and any deviations from the test plan.
* **Documentation and Reporting:** Prepare detailed test reports summarizing results and providing recommendations for further actions.
* **Test Closure Activities:** Complete necessary documents and conduct a test closure meeting to review the testing process and results.

### Mind Map of how the main tested functions

### 

Figure 1.2

## Testing Approach

### 2.1 Unit Testing

* Unit testing will be utilized to validate the performance of individual components, guaranteeing their compliance with defined requirements.

### 2.2 Integration Testing

* Integration testing will assess the interactions among internal modules within the system.

### 2.3 System Testing

* System testing will concentrate on evaluating the functionality of the entire system and its interfaces with external systems.

### 2.4 Usability Testing

* Testing the user friendliness and responsiveness of the Off\_Campus\_Review system. This will involve real users interacting with the system to gauge its ease of use, efficiency and overall user satisfaction.

### 2.5 Performance Testing

* This is to evaluate our system’s ability to handle accommodation ratings under various load conditions. It includes stress testing to assess its limits.

### 2.6 Compatibility Testing

* Testing if our system functions seamlessly using different devices and websites.

## Requirements to be Testing

### 3.1 TC1: User login functionality

* The system should open a login page.
* Required information for login:
* Username
* Password
* The required information will be sent through email to the user by the admin.
* Users must be able to enter the username and password and must be validated.
* Users must be able to use “remember password” functionality.

### TC2: Usability Testing

#### Tenant’s Usability testing

#### Agent’s Usability testing

#### Prospective User Usability testing

#### Super User Usability testing

### TC3: 10 Website Heuristic testing.

### 3.4 TC4: Test the data creation, deletion, retrieval, reporting and security functionality.

## 4. Resource Planning

### 4.1 Human Resource

|  |  |  |
| --- | --- | --- |
| Role | Resources allocated | Specific Responsibilities |
| Test Lead/Manager | Takunda | * Provides technical direction and test coordinating. * Acquire appropriate resources. * Oversee the whole testing process. * Design the test cases. * Check if the testing process is meeting specified requirements. |
| Tester | Rinae and Takunda | * Usability Testing * Performance Testing * Defect Managing |
| Developers | Denzel, Kudzai, Denzel | * Develop the system and rectify if any defects are found |
| Test Designers | Rinae, Takunda | * Test environment setup * Test cases and test plan design. |

### 4.2. System Resource

|  |  |  |
| --- | --- | --- |
| No. | Resources | Description |
| 1 | Server | is3-dev.ict.ru.ac.za |
| 2 | Network | eduroam Wi-Fi |
| 3 | Devices | android device, an iPhone, Hamilton desktop, a laptop and a projector |

## 5. Test Scheduling and deliverables

### 5.1. Project Milestones

|  |  |
| --- | --- |
| Deliverable | Date |
| Test Plan | 01/09/2023 |
| Test Cases | 10/09/2023 |
| Testing | 21/09/2023 – 29/09/2023 |
| Complete tested user-friendly software | 01/09/2023 |

### 5.2 Testing Deliverables

#### 5.2.1 Deliverables before the testing phase

* Test Plan
* Test Cases

#### Deliverables during the testing phase

* Test Scripts
* Test Data
* Defect logs and execution logs

#### Deliverables after the testing phase

* Test Results/reports
* Defect Report

## 6. Risks, Limitations, Assumptions and Contingencies

### 6.1 Risks and mitigations

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Risk | Definition | Mitigation |
| 1 | **Server Overload** | The risk of overloading the web server during performance testing, which can result in system crashes or downtime. | Implement resource monitoring and auto-scaling for the web server to handle increased load during testing.  Conduct load testing in a controlled environment to identify server capacity limits. |
| 2 | **Incomplete test coverage** | Insufficient test coverage that may lead to untested areas or features, increasing the risk of undiscovered issues | Create a comprehensive test plan that includes all critical features.  Use test case management tools to track coverage and ensure all areas are tested. |
| 3 | **Browser and device compatibility** | The website not displaying or functioning correctly on different browsers and devices. | Utilize cross-browser testing tools and real devices for comprehensive testing.  Maintain a list of supported browsers and devices, and test against those. |
| 4 | **Testing delays** | Delays in testing caused by unexpected issues, resources constrains or incomplete test cases | Create a well-defined test schedule with buffer time for unexpected delays.  Allocate adequate resources and prioritize critical test cases. |
| 5 | **Inadequate Test Documentation** | Poorly documented test cases, plans and procedure that results in misunderstandings and execution errors | Enforce clear documentation standards and provide training for testers.  Implement version control for test documentation to ensure accuracy. |
| 6 | **Data Corruption** | The risk of data corruption during testing, which can impact the integrity. | Backup production data before testing and regularly snapshot test data.  Implement proper data handling procedures and restore mechanisms. |
| 7 | **Network failures** | unexpected network issues, including slow internet connections or disruptions, can affect testing outcomes. | Perform testing in a stable network environment.  Use network emulation tools to simulate various network conditions. |
| 8 | **Performance Bottlenecks** | Risks of performance bottlenecks that slow down website responsiveness or loading times. | Conduct performance testing early in the development cycle.  Optimize code and use performance profiling tools to identify bottlenecks. |
| 9 | **Regression test failures** | Limited testing resources like time that may result in incomplete testing. | Prioritize regression tests based on risk and impact.  Automate regression testing to save time and resources. |
| 10 | **Communication Breakdown** | Risks related to poor communication between the testing team members, leading to misunderstandings and delays. | Establish regular team meetings and use collaboration tools.  Maintain clear channels of communication, and encourage open discussions. |
| 11 | **Scope Creep** | Expanding the scope of testing beyond the defined requirements. | Define and freeze the scope of testing in the test plan.  Follow a strict change control process for any scope changes. |
| 12 | **User Experience Problems** | Issues affecting the website’s user experience, including navigation difficulties, accessibility problems or design flaws. | Conduct usability testing and accessibility testing.  Involve UX experts in the testing process and prioritize user feedback. |
| 13 | **Misinterpreted Requirements** | Risk that there is misinterpretation of requirements that lead to testing the wrong features or aspects of the system. | Verify requirements with stakeholders and maintain a clear traceability matrix.  Use requirement management tools to ensure correct interpretation |
| 14 | **Test Data Loss** | Risk of losing test data due to system failures or mishandling during testing. | Regularly back up test data and implement data recovery procedures.  Ensure test data is handled with care to prevent accidental loss. |
| 15 | **Resource Exhaustion** | Resource exhaustion on the server or testing machines can disrupt the testing process. | Monitor resource usage during testing and establish resource allocation limits.  Conduct resource stress testing to identify potential bottlenecks. |
| 16 | **Scalability Limitations** | Risks related to the system’s inability to handle increased user loads and scalability challenges. | Perform scalability testing to identify and address limitations.  Consider cloud-based scalability solutions for flexibility. |
| 17 | **User input error** | Potential errors caused by incorrect or malicious user input during testing. | Implement input validation and security testing.  Utilize automated tools to simulate malicious input. |
| 18 | **Disaster Recovery Testing** | Risks associated with inadequate disaster recovery testing, which may not adequately prepare for unexpected events**.** | Create comprehensive disaster recovery plans and conduct regular drills.  Document and review the recovery process to ensure readiness. |
| 19 | **Documentation Gaps** | Incomplete or outdated documentation can lead to misunderstanding and testing. | Keep documentation up to date and perform regular reviews.  Use documentation templates to ensure completeness and consistency. |
| 20 | **Inadequate Test Data Generation** | Risks associated with the automated generation of test data that does not accurately reflect real-world scenarios. | Develop realistic test data scenarios that reflect real-world usage.  Use data anonymization techniques to protect sensitive information. |
| 21 | **Conflict Resolution** | Risks related to resolving conflicts and disagreements within the testing team. | Establish a clear decision-making process and escalation path.  Encourage team members to resolve conflicts professionally and constructively. |
| 22 | **Performance Monitoring Failures** | Inadequate performance monitoring tools or practices can result in undiscovered performance issues | Invest in robust performance monitoring tools and establish performance baselines.  Conduct continuous monitoring during and after testing. |
| 23 | **Scope Misalignment** | Misalignment between the scope of testing and project requirements can result in overlooked risks. | Ensure alignment between testing scope and project requirements through regular reviews.  - Involve project stakeholders in scope definition and validation. |
| 24 | **Test Execution Errors** | Potential errors in executing test cases, leading to incorrect test results | Develop and maintain detailed test scripts with clear steps.  Implement peer reviews of test case execution for quality assurance. |

### 6.2. Limitations

* Website can only run with eduroam.
* The data provided for testing is incomplete
* Time limitation for the regression testing
* Lack of knowledge for CSS/HTML/PHP.
* User Feedback reliance.

### 6.3 Assumptions

* All the main functions stated in the mind map on figure 1.2 are functioning properly.

### 6.4 Contingencies

* Regular communication with the developers
* Performance optimization efforts as needed.

### 7. Defect Management Plan

### 7.1. Identify Defects:

* Identify defects in the website related to ratings, reviews, and associated functionalities.
* Categorize defects as usability, functionality, security, or data-related.

### Defect Reporting:

* Report defects through a standardized defect form.
* Attach relevant screenshots and a message describing the nature of the defect and how it will be fixed.

### 7.3 Defect Triage:

* Prioritize defects based on severity and impact.
* Categorize them according to critical, high, medium and low.

### 7.4 Defect Resolution:

* Developers will resolve defects based on the priority
* Implement fixes, document changes, and verify resolutions.

### 7.5 Defect Verification:

* Testers verify defect resolutions and conduct regression testing.
* Implement fixes, document changes, and verify resolutions.

### 7.6 Approval and Revision:

* All team members must sign and approve the defect plan

## 8. Definition of test criteria

### 8.1 Acceptance criteria

These are specific conditions or requirements that must be met before the testing phase is considered successful.

* All functional requirements as outlined in the domain documentation to be met.
* Website must respond quickly; response times and throughput must be achieved.
* User friendly website, this aspect to be known through user feedback.
* The website should be compatible with designated platforms and browsers.

### 8.2 Suspension criteria

Conditions where the testing phase may be temporarily halted.

* When severe defects are discovered that renders further testing meaningless
* Insufficient testing resources
* Environmental issues like network failures and server outages.

### 8.3 Exit Criteria

Conditions to be met for a testing phase or entire testing process to be considered complete.

* 98% of the test cases should pass successfully
* All critical and high priority defects identified during testing should be resolved.
* All the team members must approve and happy with the testing done.
* All test documentations must be completed and reviewed.
* Regression testing should have been performed, and no new defects are identified.

## 9. Approvals

|  |  |  |
| --- | --- | --- |
| Name | Position | Signature |
| Takunda | Test Manager, Tester |  |
| Denzel | Developer |  |
| Kelechi | Project Manager |  |
| Khudzai | Developer |  |
| Rinae | Tester |  |