CS691 – Computer Science, Spring 2023

Pace University



SYSTEM TEST PLAN

StayMatch

Project Manager: Prathima Seethalam Radhakrishna

Authors: Shubham Mishra, Akshath Shanbhag, Rahul Mendes, Kritika Agarwal, Ajay Peduri, Dhruv Chamaria, Daulet Kapezov

Date: 04/29/2023

**Table of Contents**

[INTRODUCTION 3](#_heading=h.30j0zll)

[TESTING SCOPE 3](#_heading=h.3znysh7)

[TESTING OBJECTIVES 3](#_heading=h.tyjcwt)

[Features to be Tested 3](#_heading=h.1t3h5sf)

[Features not to be Tested 4](#_heading=h.2s8eyo1)

[TEST PROCESS DEFINITION 4](#_heading=h.17dp8vu)

[Test Process Phases and Tasks 4](#_heading=h.26in1rg)

[Deliverables 5](#_heading=h.35nkun2)

[APPROACH TO SYSTEM TESTING 5](#_heading=h.1ksv4uv)

[Approach to Functional Testing 5](#_heading=h.2jxsxqh)

[ENTRY/EXIT CRITERIA 6](#_heading=h.3j2qqm3)

[Entry Criteria 6](#_heading=h.1y810tw)

[Exit Criteria 6](#_heading=h.4i7ojhp)

[ENVIRONMENTAL NEEDS 6](#_heading=h.3whwml4)

[ROLES AND RESPONSIBILITIES 7](#_heading=h.2bn6wsx)

[TEST CYCLES AND SCHEDULE 8](#_heading=h.3as4poj)

[RISKS AND CONTINGENCIES 8](#_heading=h.49x2ik5)

# INTRODUCTION

This document describes the System Test Plan that provides a common understanding for the renters and the landlords and the StayMatch Project Stakeholders a clear idea on the project scope, objectives, and approach to performing the system testing. Also, the document explains the features to be tested, testing entry/exit criteria, resources and responsibilities, and testing schedule.

# TESTING SCOPE

The testing scope of the document includes two perspectives namely, the functional scope and the technical scope.

The functional scope consists of the following module of the “StayMatch” application: User Account Management module.

The technical scope consists of the following architectural components:

* Web browser
* Application server
* Database server
* UI/UX experience

# TESTING OBJECTIVES

The primary focus of this System Test Plan is functional testing with the goal of evaluating system implementation stability. Non-functional testing necessitates the use of specialized tooling to monitor performance characteristics, which is not available on this project.

The basis for developing functional tests and evaluating the system functionality includes the following sources:

* Business Requirements Document (BRD)
* User Stories (functional requirements)
* Requirements Composition Table (supplementary requirements)

## Features to be Tested

Below, you will find a compilation of essential functionalities categorized by their respective application modules, which are scheduled for testing. We have chosen to prioritize the features within the Account Management module for implementation this semester.

Account Management:

| **Features from RCT** | **Explanation** |
| --- | --- |
| 01.01 Register Renter | To test whether a renter can create an account |
| 01.02 Modify Renter | To test whether a renter can change information related to the account |
| 01.04 Register Landlord | To test whether a landlord can create an account |
| 01.05 Modify Landlord | To test whether a landlord can change information related to the account |
| 01.07 Show Profile | To test whether both renter and landlord can view their profile information |

Additionally, apart from the core features included in the testing scope, the functional testing process will encompass crosscutting concerns that are relevant to the specific context of each individual core feature. Please refer to the RCT for further details.

## Features not to be Tested

As mentioned above, system performance will not be tested for the lack of required tools. Also, usability and security will not be tested as well.

# TEST PROCESS DEFINITION

## Test Process Phases and Tasks

The test process consists of five phases, which include test planning, design, preparation, execution, and reporting. Each phase has a few tasks as defined below:

* Test Planning
  + Define scope and objectives of testing
  + Define roles and responsibilities
  + Define testing approach
* Test Preparation
  + Setup a test environment
  + Provision test data
  + Install the software in the test environment
* Test Execution
  + Execute all test cases
  + Find and report software defects
  + Evaluate the system stability
  + Validate all target features
* Test Reporting
  + Summarize and report the test execution results
  + Report defect metrics
  + Evaluate the test exit criteria
  + Create a test completion report, submit for stakeholder approval
  + Obtain stakeholder signoff on system testing

## Deliverables

On this project, the test process deliverables include:

* System Test Plan document
* Software Defects
* Test Execution Logs
* Test Completion Report

# APPROACH TO SYSTEM TESTING

## Approach to Functional Testing

The overall approach to functional testing will be based on the Black-box method:

* Test cases will be designed using some formal black-box techniques such as boundary-value analysis, equivalent-class partitioning, and decision tables, where applicable.
* Test execution will be conducted manually, from the user perspective and based on formal test case specifications.
* The test execution results will be captured and reported in test execution logs.
* Test Completion report will be generated based on the test execution results.

# ENTRY/EXIT CRITERIA

This section defines both Entry and Exit Criteria for test execution and is intended to establish a common understanding about the conditions when the test execution can start and when it can stop.

## Entry Criteria

The test Entry Criteria include the following items:

* The web application build is produced and deployed to the test environment.
* The system test plan is produced and approved
* The test environment is ready for testing.
* Test designs and test preparation specifications are completed.

## Exit Criteria

The test Exit Criteria include the following items:

* All requirements, in scope of testing, are covered by test cases.
* All test cases have been executed.
* Zero defects of critical and high severity remain open.
* Open defects of medium and low severity have known work-arounds.
* Test Completion report will be produced and published.

# ENVIRONMENTAL NEEDS

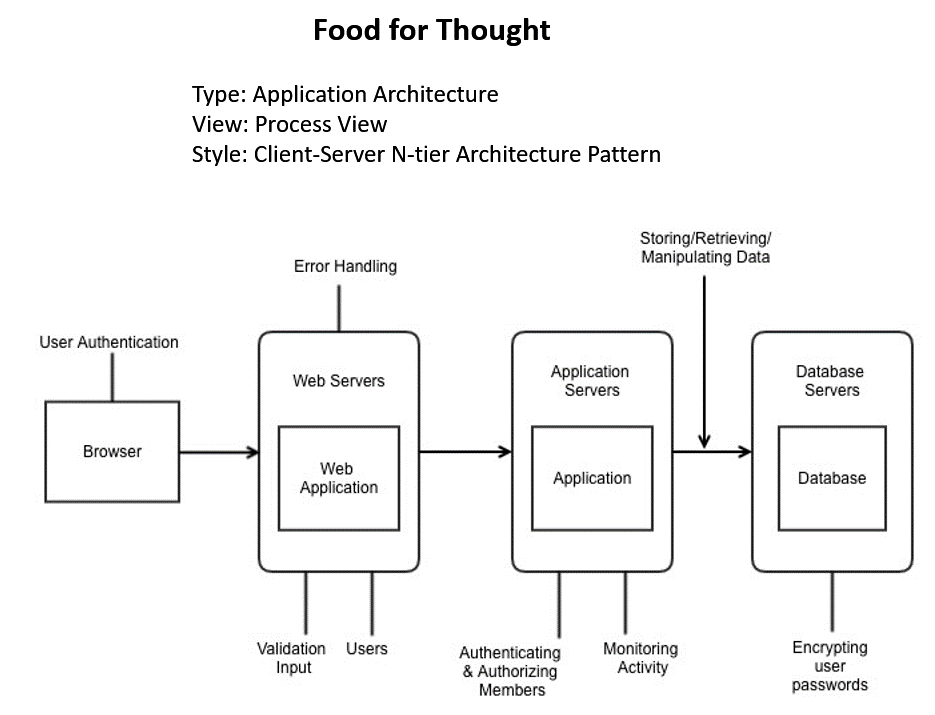
The Test Environment should be available to start test execution. It includes a laptop with running the latest version of node.js and respective packages through npm for running React web server and database connected with MongoDB, and internet browsers (Chrome, Firefox, Internet Explorer and Safari) to access the application which have node.js running for the backend. The architecture of the test environment is shown below.

StayMatch

Type: Application Architecture

View: Process View

Style: Client-Server N-tier Architecture Pattern



# ROLES AND RESPONSIBILITIES

The project team has seven members that are assigned various project roles including Project Manager, Product Owner, Lead Business Analyst, Lead Developer, DBA, Lead QA Analyst. Their responsibilities are defined in the table below.

| **Project Role** | **Role Responsibilities** |
| --- | --- |
| Project Manager | Reviewing and approving the System Test Plan, test design specifications.Managing the test environment preparation. Tracking the testing schedule and results. |
| Lead QA Analyst | Designing a test plan, establishing a test repository, developing test case specifications, executing testing and reporting defects. |
| Product Owner | Contributing to the test plan and test case specifications. Reviewing test results. |
| Lead Business Analyst | Contributing to the test plan and test case specifications. Reviewing test results. |
| Lead Developer | Establishing and maintaining the test environment, assisting a Lead QA Analyst throughout the testing process. |
| DBA | Assisting the Lead Developer in establishing and maintaining the test environment. |

# TEST CYCLES AND SCHEDULE

The system test execution will be conducted as three test cycles that are aligned with three application modules as follows:

Cycle 1. User Experience I

* This cycle concentrates on testing the first part (Register and authentication of an user account) of the User Experience Module

Cycle 2. User Experience II

* This cycle concentrates on testing the second part (adding and viewing properties) of the User Experience Module.

Cycle 3. Payment and User Login

* This cycle concentrates on testing the Payment and User Login Module.

See the schedule of the test execution cycles in the project plan.

# RISKS AND CONTINGENCIES

This section highlights a few potential risks and contingencies that maybe happened during the system testing.

* Risk: The system may not be able to handle a large volume of users and transactions. Contingency: Perform load testing to determine the system's capacity and ensure scalability.
* Risk: There may be compatibility issues with different operating systems and browsers. Contingency: Conduct extensive compatibility testing on various operating systems, browsers, and devices.
* Risk: The system may have security vulnerabilities that could compromise user data. Contingency: Perform thorough security testing to identify and address any vulnerabilities, and implement robust security measures.
* Risk: The system may not meet performance requirements, leading to slow response times and user frustration. Contingency: Conduct performance testing to identify and address any bottlenecks, and optimize the system's performance.
* Risk: There may be issues with system integration, leading to errors and data inconsistencies. Contingency: Conduct integration testing to ensure that different components of the system work seamlessly together and that data is correctly transferred between them.