

TEST PLAN FOR RENTR

ChangeLog

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made
1	26/02/2021	Nathan Carriere	Initial Draft

1	INTRODUCTION.....	2
1.1	SCOPE.....	2
1.1.1	<i>In-Scope</i>	2
1.1.2	<i>Out-of-Scope</i>	2
1.2	QUALITY OBJECTIVE	2
1.3	ROLES AND RESPONSIBILITIES.....	2
2	TEST METHODOLOGY	3
2.1	OVERVIEW.....	3
2.2	TEST LEVELS	3
2.5	TEST COMPLETENESS	3
3	TEST DELIVERABLES	3
4	RESOURCE & ENVIRONMENT NEEDS	4
4.1	TESTING TOOLS	4
4.2	TEST ENVIRONMENT.....	4
5	TERMS/ACRONYMS	4

1 Introduction

The goal of Rentr is to be a platform where landlords and tenants can post and view apartment listings. This document describes the testing process for Rentr. Rentr is using an agile methodology with Travis-CI.

1.1 Scope

1.1.1 In-Scope

There are currently 2 features that are being tested and 1 feature whose testing is under development. The first is user account management, which includes account creation, login/logout, and having listings associated with an account. The second is adding apartment listings, which includes posting a listing, specifying details about a listing, and adding pictures to a listing. A third feature, modifying listings, is under development. This feature includes being able to edit rental listings, being able to mark a listing as rented out, and the ability to delete a listing.

1.1.2 Out-of-Scope

There are 2 features that are not being tested and 1 feature under development. The feature under development is the same feature mentioned in the above section. The remaining two features not under test are Search and our Q&A Section features. Search involves the ability to search for listings based on price, city, number of bedrooms, number of bathrooms, and potentially other metrics. The Q&A Section feature includes the ability for a tenant user to post a Q&A question on a listing, and for any user to post an answer to that question.

1.2 Quality Objective

Rentr has two primary quality objectives when it comes to testings. First, testing should ensure the Application Under Test (AUT) conforms to functional and non-functional requirements. Second, testing should ensure that issues are addressed prior to code entering the production environment.

1.3 Roles and Responsibilities

In general, there are two main roles that team members have taken. These roles are UI Developer and Backend Developer. UI Developers are responsible for client-side software development in the front end and deal with the user interface. Backend Developers are responsible for server-

side and database development. Other group roles, such as QA or documentation, are generally dispersed between group members.

Name	GitHub username	Role
Hualong Qiao (Mike)	Mik3Qiao	UI Developer
Ronnie D’Souza	RonnieDsouza	Backend Developer
Nathan Carriere	NathanCarriere	UI Developer
Azizul Hakim	AzizulHakim96	Backend Developer
Justin Choi	Jcsnorlax97	Backend Developer

2 Test Methodology

2.1 Overview

We have opted for an Agile testing methodology due to the small size of our team and our team’s user of Agile for software development. Our team is developing software and tests in incremental, rapid cycles. Every release of the product will undergo thorough testing.

2.2 Test Levels

Currently, our project has unit testing, integration, and manual system testing. We are checking our UI stuff mostly manually, as development proceeds. Testing will be expanded as sprints continue. We are guiding our system of tests using the Agile Methodology.

2.3 Test Completeness

We will consider the test complete if we have 100% test coverage, all tests have executed and passed, and if all open bugs are fixed or will be fixed in the next release.

3 Test Deliverables

During the testing lifecycle we will deliver the:

- Test Plan
- Test Cases

4 Resource & Environment Needs

4.1 Testing Tools

- Travis-CI
- Postman
- Jest

4.2 Test Environment

Hardware:

- A computer with a stable internet connection.
- A computer capable of running a basic local development.

Following **software's** are required in addition to client-specific software.

- Windows 8 and above
- Travis-CI

5 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
API	Application Program Interface
AUT	Application Under Test
JWT	JSON Web Token
JSON	JavaScript Object Notation
CI	Continuous Integration