TEST PLAN FOR
SWE-GROUP-6-SPRING21

ChangeLog

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made
1.0	03/14/21	Team	Initial Document

1 Introduction

Brief introduction of the test strategies, process, workflow and methodologies used for the project

This is Group 6's test plan for their twitter like application written in django. This document will outline the Groups software testing to ensure a functional and secure system.

1.1   Scope

1.1.1 In-Scope

Publishing a Tweet

Liking a Tweet

Unliking a Tweet

Login

Registration

Following a User

Unfollowing a User

Searching for Tweets, Users

1.1.2 Out-of-Scope

UI Elements

1.2 Quality Objective

- Ensure the Application Under Test conforms to functional and non-functional requirements
- Ensure the AUT meets the quality specifications defined by the client
- Bugs/issues are identified and fixed before go live
- Security Issues will not jeopardize user information
- Issues within the software will be kept to a minimum

1.3 Roles and Responsibilities

Name	Net ID	GitHub username	Role
Caden Austin	CBA169	CadenAustin	Leader, Full stack, Documentation
Connor Hicks	ch3083	connorjhicks	Documentation, Back-end Design
Derek Wilson	dlw606	derekwilson99	Front-end Design
Darrion Robinson (FORMER)	dr1374	Darrionr1	Front-end Design
Bryan Alston (FORMER)	bma271	bma1701	Front-end Design

2 Test Methodology

2.1 Overview

We are implementing the agile methodoligy as it merges well with our need to adapt to new requirements and the short ammount of time we have. Also since we have some programmers new to webdesign it's useful to have an iterative test plan to prevent errors accumulating.

2.2 Test Levels

With our testing we are running Unit Tests, Integration Tests, and System Tests. Unit testing is performed when a new component is added and we confirm independently of the other systems that the component tested is functional. When a component passes Unit Testing we then run Integration Tests to check if it interacts well with previously integrated components. Then after the sprint is done we look back at the system as a whole to confirm it works as intended

2.3 Bug Triage

The goal of the triage is to

- To define the type of resolution for each bug
- To prioritize bugs and determine a schedule for all "To Be Fixed Bugs'.

Currently no known bugs exist within the system however we do have a rank choice order for which bugs need to be prioritized over others which is as follows with higher priority being higher:

- Security Compromising Bugs
- System Integrity Compromising Bugs
- Functionality Compromising Bugs
- User Satisfaction Compromising Bugs

As bugs come in their priority is decided based on where they fall on this list.

2.4 Suspension Criteria and Resumption Requirements

Suspension criteria define the criteria to be used to suspend all or part of the testing procedure while Resumption criteria determine when testing can resume after it has been suspended

2.5 Test Completeness

- 100% test coverage
- All Manual & Automated Test cases executed
- All open bugs are fixed or will be fixed in next release

3 Test Deliverables

- Test Plan
- Test Cases
- Bug Reports
- Test Metrics

4 Resource & Environment Needs

4.1 Testing Tools

- Requirements Tracking Tool - GitHub
- Bug Tracking Tool - GitHub
- Automation Tools – Travis CI

4.2 Test Environment

Minimum Hardware Requirments:
- 1 Core CPU
- 512MB Ram
- 4 GB Disk

Software Requirements:
- A Web Browser
- Python Libraries from the Requirements.txt File

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
CI	Continuous Integration