

Epoch Testing Plan

Sprint 2

February 27, 2024

Change Log

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made
0.0.01	Feb. 02	Aymanhki	Initial testing implementation
0.0.02	Feb. 02	Aymanhki	Accounts creation and login front-end
0.0.02	Feb. 02	Aymanhki	Accounts creation and login back-end
0.0.02	Feb. 03	Aymanhki	Accounts creation and login regression test
0.0.02	Feb. 04	Aymanhki	Accounts creation and login front-end
0.0.03	Feb. 06	Aymanhki	CI/CD pipeline
0.0.03	Feb. 07	cedricb12	Added Follow/unfollow feature
0.0.03	Feb. 07	cedricb12	Unit test for follows/unfollows
0.0.03	Feb. 08	cedricb12	Integration tests for follows/unfollows
0.0.05	Feb. 19	Alvinavinante,	Added profile page
0.0.05	Feb. 19	Vishal-singh-heer	Added navigation bar
0.0.05	Feb. 19	castmallards	Added posts
0.0.05	Feb. 19	cedricb12	Added search page

Introduction

Scope

Current Functional Features Being Tested:

- Creating/Deleting Account Feature (Unit, integration, acceptance)
- Getting user account info (Unit, integration, acceptance)

- Following/Unfollowing Feature (backend unit and integration tests)
- Webserver and API router (Integration, load)

Current Back End Coverage: 72%

Functional Features Not Being Tested:

- Posts and Feed
- Favorites
- Hashtags

Roles and Responsibilities

Name	Net ID	GitHub username	Role
Ayman Agamy	agamyahk	Aymanhki	Developer
Obaid Ali	alio	castmallards	Developer
Vishal Singh Heer	heervs	Vishal-singh-heer	Developer
Alvin Avinante	avinanta	alvinavinante	Developer
Cedric Berard	berardc5	cedricb12	Developer

Test Methodology

Test Levels

Unit Tests

- For the backend code the API endpoints and objects for each core feature we will have unit tests with mocked database interfaces that will allow us to test the logic of these scripts quickly.

Integration Tests

- The files that cannot easily have all their interactions mocked like the router, webserver and database, will be covered by integration tests by sending mock requests through the server and verifying functionality.

Acceptance Tests

- Each core feature will have an acceptance test testing the interaction between the client side and server side of the web application.

Regressions Tests

- Regression testing will execute whenever a commit is pushed to the main branch. It will run all unit and integration tests and our continuous integration pipeline will push the new branch live if these tests pass.

Load Tests

- We currently run a load test on the webserver script as part of the CI pipeline tests. In the future we will make sure at least two requests associated with every core feature are included in the test load

Test Completeness

Testing will be complete when we have the mandatory 100% back-end code coverage, decent front-end coverage, load testing for the server and database, and acceptance tests for each core feature.

Resource & Environment Needs

Testing Tools

Required tools for Unit and Integration tests;

- Pytest
- Pytest-cov

Required tools for front end tests;

- Npm/React-Scripts/Jest

Required tools for acceptance tests;

- Selenium
- Chromedriver

Test Environment

It mentions the minimum **hardware** requirements that will be used to test the Application.

Example, following **software's** are required in addition to client-specific software.

- Windows 8 and above
- Docker
- Github actions
- Stable internet connection
- Browser
- Computer capable of running a basic local development

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