# TEST PLAN FOR

# DEV DIARIES

*Changelog*

| Version | Change Date | By | Description |
|---|---|---|---|
| 1.0 | Oct. 16, 2023 | Jakob McKenna | Initial Draft |
| | | | |
| | | | |

# 1 Introduction

Dev Diaries is a web application with the goal of making project management for developers easier than ever. This document outlines the testing plan for Dev Diaries, what kind of testing is in scope, and what developers roles are in this project. It also covers the test methodology and resource/environment needs.

## 1.1 Scope

We plan to have five main features for our application. Currently three are under development and are projected to be started and on the way to be completed by the end of iteration 2. The other two are planned to be completed by the end of iteration 3. All features will be fully functional and complete before the end of iteration 4.

User authentication, GitHub API updates, and Bookmarks are all projected to be started by the end of iteration 2. and so, they are in scope for iteration 2. We plan to finish User authentication and GitHub API updates and have test coverage for them by the end of iteration 2. Also, we plan

to develop more tests for these features during iteration 3. By the end of iteration 2 we plan to have both unit tests and manual system tests completed for these features.

User communication and time management tools are projected to be completed by the end of iteration 3, and so they are in scope for iteration 3. The features from iteration 2 are also in scope for iteration 3, as we plan to develop more rigorous tests. By the end of this iteration, we plan to have more detailed test coverage.

By iteration 4, all five main features will be in scope for testing, and we hope to have unit testing, integration testing, system testing, regression testing, and manual system testing completed.

## 1.2 Roles and Responsibilities

| Name | GitHub username | Role |
|---|---|---|
| Jakob McKenna | JakobMcKenna | Testing |
| Tehillah Kangamba | TehillahK | Backend development |
| Susie Kihaile | lovelyturtles | GitHub integrations |
| Isabella Anderson-Gregoire | Isabelle-ag | Frontend development |

# 2 Test Methodology

## 2.1 Test Levels

Tests planned for current features:

1. User authentification
   - Unit tests: Testing individual functions such as createUserPasswordData()
   - Integration tests: Testing the seams between user input, backend functions, and the database
   - Acceptance testing: Making sure a user can create and account, and successfully log in manually
2. GitHub API updates
   - Unit tests: Testing individual functions such as getBranchCommit()
   - Integration tests: Testing seams between the API and our application

- Acceptance testing: Manually test that GitHub API updates are accurate, and that users are getting the information they want

3. Bookmarks
   - Unit tests: Test individual functions at the unit level
   - Integration tests: Test seams between our database where the bookmark is stored all the way to the frontend where they are displayed
   - Acceptance testing: Test that a user can see and successfully click on a bookmark

Tests planned for future features:

4. Communication tools
   - Unit tests
   - Integration tests
   - Acceptance testing
5. Time management tools
   - Unit tests
   - Integration tests
   - Acceptance tests

All of the core features will also be tested using regression testing, where are unit and acceptance tests are activiated any time new code is deployed. This will be done automatically with GitHub actions.

We will also do load testing to ensure our original goal of being able to respond to 50 users with a total of 200 requests per minute concurrently.

## 2.2 Test Completeness

- We plan to have 100% backend code coverage
- We plan to have atleast 10 unit tests for each feature
- We plan to have atleast 10 integration tests that cover the system
- We plan to set up regression testing to automatically run when new code is deployed
- We plan to have acceptance tests for each feature correlated to our user stories

# 3 Resource & Environment Needs

## 3.1 Testing Tools

- We will be using GitHub to setup regression testing with GitHub actions.
- We will use the jest testing tool for automatic testing.
- We will use a web browser for manual testing.

- We will use GitHub to track bugs.

## 3.2 Test Environment

Hardware requirements:

- A computer that can connect to wifi and run simple websites
- A computer that can handle running tests in jest

Software requirements:

- An updated web browser that can run our website
- jest installed

# 4 Terms/Acronyms

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

| TERM/ACRONYM | DEFINITION |
|---|---|
| API | Application Program Interface |
| GitHub actions | CI/CD platform used for automation |
| CI | Continous Integration |
| CD | Countinous Deployment |
| jest | A testing framework for JavaScript |