**CS673 Software Engineering** 

**Team 1 - Chit Chat**

**Software Test Document**

| Team Member | Role(s) | Signature | Date |
| --- | --- | --- | --- |
| Deasia Little | QA Leader | *Deasia Little* | September 21, 2025 |
| Masih Vahida | Team Leader | Masih Vahida | September 21, 2025 |
| Robin Roeoesli | Requirement Leader | *Robin Roeoesli* | September 21, 2025 |
| Jordyn Lipsey | Configuration Leader | *Jordyn Lipsey* | September 21, 2025 |
| Ardit Briskaj | Design and Implementation Leader | *Ardit Briksaj* | September 21, 2025 |
| All | Security Leader |  | September 21, 2025 |
|  |  |  |  |
|  |  |  |  |

**Revision history**

| **Version** | **Author** | **Date** | **Change** |
| --- | --- | --- | --- |
| **1** | **Deasia Little** | **September 21, 2025** | **Initial Draft** |
|  |  |  |  |

[Testing Summary](#_sm5odwyvuk3j)

[Manuel Tests Reports](#_pqso2mbjyzx4)

[Automated Testing Reports](#_mtfbusfb0eq3)

[Testing Metrics](#_rijyjeu2ojqa)

[References](#_15tmymhipvdv)

[Glossary](#_8n34lvocupub)

# Testing Summary

In this section, you will summarize what was tested, who is involved in testing, when to test, testing techniques used, and testing result. You may have the following tests

* + Unit Testing
    - Currently not applicable, as unit tests have not been implemented in this sprint release. They are planned for future iterations.
  + Integration testing
    - Basic end-to-end flows exercising the back-end and front-end through browser automation, using Playwright as the testing framework.
  + System Testing
    - App startup, navigation, and core user flows have been validated in this iteration.
  + Acceptance Testing
    - Acceptance criteria defined in the user stories were verified with automated Playwright tests.
  + Regression Testing
    - The CI pipeline is currently being built. The automated tests and QA metrics will be run with each pull request when complete.

# Manual Testing Report

In this section, you will give a detailed description of each manual test case performed and the result. If this is a previous You shall list what are existing tests developed in the previous semester and what are new tests developed currently.

Here is a sample template that can be used for each test case. For system tests or acceptance tests, you may also include some screenshots.

* Test case ID, name
* New or old:
* Test items: (what do you test )
* Test priority (high/medium/low)
* Dependencies (to other test case/requirement if any):
* Preconditions: (if any)
* input data:
* Test steps:
* Postconditions:
* Expected output:
* Actual output:
* Pass or Fail:
* Bug id/link: (this should link to your github issue id)
* Additional notes:

(You can use an additional spreadsheet for this section as well)

# Automated Testing Report

Describe briefly the automated testing you have done, including where the test code resides in your code repository, what test frameworks are used, and the screen shots or generated testing report.

* + Framework: Playwright
  + Config: tests/[playwright.config.ts](http://playwright.config.ts)
  + Tests: tests/e2e
  + CI: The CI pipeline is currently in progress but the Playwright tests will run on PR and uploads reports/ as artifacts.
  + Reports/Artifacts: tests/plawright-report/index.html (Attached with this document)

# Testing Metrics

In this section, you shall report any metrics used for the evaluation, e.g. # of test cases, test coverage, defects rate, etc.

* + Lines of Code (LOC): reports/metrics-summary.json
  + Average Cyclomatic Complexity: reports/plato/plato.json
  + Function Count: reports/metrics-summary.json
  + Automated Tests: reports/playwright.json and tests/playwright-report/index.html
  + Automated Tests Pass Rate: metrics-summary.json
  + Manual Tests: See attached spreadsheet
  + Manual Tests Pass Rate: 100%
  + Known Defects: Github Issues

# References

* + Project Repo: <https://github.com/BUMETCS673/cs673olf25project-cs673olf25_team1>

# Glossary

* + E2E (End-to-end): Tests that run the app in a browser and verify user-visible flows
  + Smoke Test: A small set of test that check critical paths
  + Cyclomatic complexity: A metric indicating the number of independent paths through code. A high number means the code could be harder to test and maintain.
  + Pass Rate: Percentage of tests that passed in a run.