**CS673 Software Engineering** 

**Team 1 - Spring Alert**

**Software Test Document**

| Team Member | Role(s) | Signature | Date |
| --- | --- | --- | --- |
| Jim Penner | Team Leader | *Jim Penner* |  |
| Shenxiao (Sherry) Lee | Requirement Leader | *Sherry Lee* |  |
| Chaozheng Zhu | Configuration Leader | *Chaozheng zhu* |  |
| Hunter Washburne | QA Leader |  |  |
| Jordany Arnaud | Design & Implementation & Security leader | *Jordany Arnaud* |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Revision history**

| **Version** | **Author** | **Date** | **Change** |
| --- | --- | --- | --- |
| **1** | **Jim Penner** | **05/22/24** |  |
| **2** | **Jim Penner** | **06/09/24** | **Update summary and added S/sheet** |

[Testing Summary](#_heading=h.gjdgxs)

[Manuel Tests Reports](#_heading=h.30j0zll)

[Automated Testing Reports](#_heading=h.1fob9te)

[Testing Metrics](#_heading=h.3znysh7)

[References](#_heading=h.2et92p0)

[Glossary](#_heading=h.tyjcwt)

# Testing Summary

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

* + Unit Testing - JUnit is being used as we are working in Java. ***Jordany has built out user stories ie: registration requests/invalid user account etc built out in project test folder. Also in the UserManagement Service Test a @beforeeach test is run. This triggers the annotated method to be executed before each test method. This establishes any common state or resources needed for individual test cases. This properly initialises before each test case runs. UserInvalid test run as well.***
  + Integration testing - ***Each microservice to be tested in isolation first. This may mean that we address integration testing at Iteration3. We need to identify all common scenarios where the microservices interact. In our case an example is User Registration triggering a notification. Focus on integrating different layers of the app.***
  + System Testing - Postman was discussed and I have used Jira before
  + Acceptance Testing - This is part of Iteration2
  + Regression Testing

# 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. ***This has been started including building out the spreadsheet with all the various testing examples.***

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.

# Testing Metrics

In this section, you shall report any metrics used for the evaluation, e.g. # of test cases, test coverage, defects rate, etc. - JP to expand on this before Iteration2 submission. Some of the testing metrics that we will cover are:

* Response time (latency),
* Error rate, and try to identify between User and Notification which has higher error rates.
* Throughput: requests per second.
* More to come.

# References

* [A Complete Guide to Unit Testing - Software Automation Testing Consultancy Service (shiftasia.com)](https://shiftasia.com/column/a-complete-guide-to-unit-testing/)
* [Automation Testing in Agile Software Development - Software Automation Testing Consultancy Service (shiftasia.com)](https://shiftasia.com/column/automation-testing-in-agile-software-development/)
* [Automation Testing in Agile Software Development - Software Automation Testing Consultancy Service (shiftasia.com)](https://shiftasia.com/column/automation-testing-in-agile-software-development/#toc0)
* <https://sourcemaking.com/refactoring>
* <https://softwaretestingmaterial.com>
* API Integration testing: <https://www.youtube.com/watch?v=QI-d5Gc6R5g>
* <https://www.jetbrains.com/help/idea/create-tests.html>
* JUnit5 user guide: [junit.org/junit5/docs/current/user-guide](http://junit.org/junit5/docs/current/user-guide)
* [engineering.atspotify.com/2018/01/testing-of-microservices](http://engineering.atspotify.com/2018/01/testing-of-microservices)

# Glossary