**Software Implementation and Testing Document**

**For**

**Group 22**

Version 1.0

**Authors**:

Olivia J.

Nicolas C.

Felipe L.

# Programming Languages (5 points)

Python, backend development, want to learn how to make web applications with Python.

HTML/CSS, frontend development, simplest frontend development to pair with Python.

# Platforms, APIs, Databases, and other technologies used (5 points)

Django – the backend framework, role-based access, authentication, building, and running our web application

Bootstrap – using with html for our frontend development

# Execution-based Functional Testing (10 points)

1. User Posts

* Test the ability to upload photos and verify that captions can be added.
* Remove a photo and confirm that the system handles the deletion correctly.

2. Account Management

* Test account creation by entering valid information, ensuring users can create an account successfully.
* Test the profile editing page by changing the profile name and verify that change upholds.
* Test the login and logout functionality by trying various correct/incorrect usernames and passwords, to ensure authentication.

3. Interactions with Other Posts

* Test liking a post by verifying that the like counter updates.
* Test commenting functionality to ensure users can post comments on others’ posts.
* Follow another user and verify that user is now following them.

4. Admin Features

* Test admin features by logging in with an admin account and verifying that posts can be removed and user accounts can be suspended or deleted.

5. Bucket List Generation and Management

* Verify that a randomized list of 25 items is provided upon account creation.
* Test the reroll feature to ensure users can reroll bucket list items up to two times.
* Ensure that the completion status of items is accurately tracked and updated.

6. Search and Navigation

* Test the search feature by searching for user profiles and verifying their profile is displayed.
* Test the home page and profile navigation buttons to ensure they work as intended.

7. Security and Authentication

* Test the login attempt lockout to ensure the system locks the account after 3 failed attempts within 30 minutes.

# Execution-based Non-Functional Testing (10 points)

1. Security

* Test brute force mechanisms by attempting to login with incorrect passwords multiple times to ensure that the system enforces account lockouts.

2. Performance

* Simulate a load of 100 concurrent users to ensure the application handles multiple suers without significant performance degradation.

3. Reliability and Usability

* Verify that the system behaves as expected on the first click of any button.
* Perform code reviews to check that the code is well-documented and maintainable.

4. Data Integrity

* Test data storage and consistency by ensuring that user progress, posts, and interactions are saved correctly and remain consistent after server restarts.

# Non-Execution-based Testing (10 points)

1. Code Reviews

* Conduct code reviews and PRs to ensure coding standards, maintainability, and seamless integration.

2. Inspections

* Review key components of system, such as user authentication, database interactions, etc. to ensure there are no issues.

3. Walkthroughs

* Perform walkthroughs with the development team and the stakeholders, presenting parts of the code or the design to gather feedback on potential issues or areas for improvement.