XII. Testing Strategy

Our testing approach is designed to ensure each component in the Qualtrics based survey and the self-monitoring mobile app functions efficiently, consistently, and accurately. We will confirm that each feature of the system aligns with the appropriate specifications and operates reliably. This will be done using a methodical and iterative process and guide us through every step of the testing lifecycle.

1. Identify the Requirements to be Tested: Using the Software Requirements Specification, we will create each test case to make sure it satisfies both functional and non-functional requirements. This way, we will be able to prioritize the essential features like PDF generation, automated email delivery, and clustering algorithms.

2. Determine Test Cases for Each Module: We will create particular test cases for each module to verify its functionality. These test cases will be linked to the relevant requirements.

3. Review Test Data and Test Cases: Before test cases and data are executed, we will review them to make sure they adequately cover the anticipated operating scenarios and edge cases. This will help in identifying any gaps or inconsistencies early in the testing process.

4. Define Expected Results: For every testing input, it must have a well-defined expected output. For example, the email the survey participant inputs must be the email the PDF will be sent to.

5. Document Test Configuration and Data: How the test was set up, test data, and expected outputs will all be documented in a test plan. This will allow future potential teams to reproduce and replicate it as well as use it as a reference.

6. Performing the Tests: Following the test plan, each test case will be executed.

7. Document Tests from Performance: For tests that failed, we will write a report that outlines the problem, potential causes, and the series of events leading up to the failure.

8. Perform Integration and System Testing: Integration testing will allow us to see if the system components or wiring interact properly. This is to ensure the entire application functions as a unified entity.

9. Continuous Integration (CI): Since we are running this project in an iterative nature, we will use CI. This will allow us to automate testing for new code changes. We will also be able to maintain code quality and identify issues early.

10. Test Documentation and Reports: A detailed report will be prepared upon completion of the testing phases. This will include test results, setups, and any problems discovered.