Phase III deliverables:

Due date:  May 10, 2023

The quality of the deliverables will be measured by:

* Completeness, (all functionalities that satisfies the requirements)
* Consistency (description and functions should be used in a consistent way)
* Traceability (should be able to trace from requirements to test case and user manual in terms of functionalities)
* Traceability should be maintained among major deliverables, e.g., requirements specification, use cases, ERD, GUI, user manual, and test cases.

1. Improvement:
2. Enhancement from phase I and II deliverables, if necessary
3. Complete **user manual** that covers all functionalities regardless of implementation status
4. Introduction

 Summary of the items and features to be tested

 References to related documents such as project authorization, project plan, QA plan, configuration management plan, relevant policies, relevant standards.

4. Test cases

Each test case must contain the following:

1. Test case ID and Name
2. Related use case number(s), sequence diagram name (s) or any other relevant design document
   * + - 1. Must explain precisely which requirement the test case is covering
3. Description of test case
4. Input data description
5. Pre- and post- condition (before and after testing)
6. Testing procedure
7. Step-by-step description on:
8. How to execute the test case (including what and where to input data)
9. Specify the criteria to be used to determine whether each test item has passed or failed
10. Expected result vs. actual result (should be blank. It will be filled by testing teams)

5. Test data: a set of data items that are necessary to run the target test cases

* + 1. Test data can be in any format: Excel, Word, Notepad, etc.
  1. Traceability analysis (enhancement of traceability table from the phase II)

1. add test cases to the table and modify the table when necessary.
   1. Deployment procedure and corresponding diagram
2. Software and hardware description and their relationships
3. Location and interface descriptions among:
4. Cloud service, if any
5. Web server
6. Database server
7. Your application software – show the locations and the way the components are interacting each other
8. Any other device needed to run your application
9. An example: Diagram

   Description automatically generated

* 1. A table of prioritized use cases: from the highest priority to the lowest

The table must contain:

* Use case ID, Name, priority, implementation status, corresponding Test case ID, Name, and remark

* 1. Each functionality in SRS must be covered by at least one test case

1. However, limit the number of test cases up to 20. If you need more than 20, prioritize and make it equal to or under 20.
2. List of all functionalities specified in the SRS and select 20 test cases by priority for testing

**Final Deliverables**

* A presentation file that illustrates:
  + - Illustration of test cases
    - Test plan on how to run your test cases
    - how to set up the test environment.
* Well-documented document from the item 1 through 8 in a Word file
* Item 4 and 5 (a landscape format table) for testing in an Excel/Word.. file. Testing document contains:
* Title: Project team name and application name
* Table heading: Test case number, test case name, short description, expected result, actual result, comments, pass/fail, tested by
* Each row must have enough space for comments and test result.
* Source code in zip file
* Meeting minutes