Phase 3: Development

Team Leader: Preston Engstrom
Working Group: Tyler Kennedy Collins
Yucen Jin
Jaclyn Binch
Andrew Rooney
Jeff Yang
Alex Lawrence

Abstract

This document describes the development phase of Team Project Meta-Management. It contains not only detailed design documents but the complete testing strategy for interactions with the software.

1 Detailed Design

The detailed design documents that the team used to produce the software can be found directly following this phase. These detailed design documents are in the form of user stories which explain necessary interaction between subsystems as well as illustrate what functionality the software is going to have. In terms of the functionality section of the user stories, they essentially are modern day CRC cards. These user stories will be constantly referenced throughout the rest of the development of the software as they have become the acceptance criteria for the software.

For detailed front-end design drawings and references, please see the design phase's user interface section, as they remain the same.

2 Documentation - See Appendix-A

An argument can be made that the most important part of software is the documentation. The use of this documentation is not only relevant to customers, but also programmers when returning to legacy code when needing to fix broken test cases. As such, the team has included the documentation in an easily updatable and maintainable appendix. The appendix in particular is appendix A.

3 Coding

The development phase of any software engineering project must include large amounts of code. As such, the code which makes up the project is contained in Appendix B of the project-binder. Any changes to the code will be reflected in this section.

The remainder of this section contains the version number of all libraries and tools the team has used to produce the software. Upon any conflicts with the software libraries, revert to these safe versions.

- php 7.0
- Laravel 5.2.29
- Image Intervention 2.0.0
- mySQL 14.14
- phpMyAdmin 4.6.0
- HTML 5
- CSS 3

4 Module Testing

Module testing in the software is accomplished by running the database seeder (see documentation) and verifying the results against known data. This routine should not be completed by the programmer of the module as they are biased. Instead this task is to be performed by a typical testing programmer.

If a new system was introduced, it should first be tested in a development environment such that any crashes or issues do not affect the overall stability of the system. Second, the database seeder should be run to insert the default testing values into the database. These values are known and can be found in the database seeder documentation. All steps following are of course dependent upon the particular module. The implication is that all functionality of the module is verified against known data. If the module passes the test, it is safe to proceed to integration testing.

5 Integration Testing

Integration testing is to be completed via the Big Bang model. This approach is accomplished by grouping modules together and validating their interaction. This method was chosen due to its ability to save time during the development process. For this method to be effective, all interaction both positive and negative must be documented. This is to ensure consistent behavior between modules and other parts of the software.

Completing an iteration of integration testing is outlined as follows:

- Document behavior of system before any changes have been made.
- Apply new version of module or entirely new module to the grouping.
- Document new behavior against intended behavior or interaction.
- If there exists any unintended side effects or bugs in interaction, develop a new build and go to step 3. Otherwise proceed to step 5.
- Document new behavior in its current state as it now must be the fully intended behavior.

This process is to be executed always after a module has passed its personal module testing phase.

6 System, Site, and Acceptance Testing

Testing of the software will be accomplished via the chart in the following section. The testing chart is used by starting at the "Start" node and traversing breadth first through the graph. The reason we have chosen this medium for testing is so that it is possible to be understood by everyone in the development team as well as non-native English speakers. When making changes directly to features it is intended for testers to follow the proceeding module test section. After completion of a module test, it follows that integration testing must be done.

All labels in sub procedures have accompanying user stories which explain what exactly is to be tested in detail. This further eliminates any ambiguity in how the testing is to be done, and in what order.

This testing roadmap has the effect of completing the system, acceptance, and site testing requirements of the software. If the software is able to pass through the roadmap without failing at any point it must thus be accepted.

7 Preparation for Customer Training

When the customer takes on the responsibility of the software they will need to be extremely familiar with the testing framework. This is due to how many of the features interact with one another, and how one change can affect a multitude of pages in the system. Every time there is a significant iteration of the software in the future, the customer must be ready to fully test entire sections of the software.

The customer must also be able to understand the documentation that is contained as part of the software. Without it they would be lost as to how the basic functionality of the software is managed. If unable to do this, it could never be extended.

Thus, a basic list of customer training needs has been compiled to highlight the areas of interest:

- Ensure that the testing framework is easily understood and referable.
- Make sure that the documentation is something that all team members, including the possibility of new members, are able to use.
- Provide a method for installing the software.
- Ship the software with default test cases for customers to use at first during acceptance testing of their own.
- Illustrate how to add new features as best as possible. Simple ways, as well as complex ways.
- Provide access to the database tools for system administrators.
- Show how to use the software as a simple user too!

As this is only the preparation step of customer training, this list is subject to change at any time by the test leader of the team.