**Test Deliverables**

As part of the test deliverables, our team will convey our findings in the form of a test plan that includes the following components:

* Test plan document
* Test design specifications
* Test cases
* Test procedure steps
* Test Data
  + Test logs (Bugs/Errors & Execution)
  + Test summary report
  + Test revision history for each bug or error

\*Software components (Modules & Algorithms) are not included in test deliverables\*

**Test Tasks**

|  |  |
| --- | --- |
| Skill Level | Description |
| 0 | Consumer Level (No software experience, may need help navigating UI) |
| 1 | Intermediate (User knows how to navigate and input data) |
| 2 | Advanced (User is aware of what is going on behind the scenes, but is not familiar with the implemented modules) |
| 3 | Developer / Tester (User knows how to implement, design and write code pertaining to specific system) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Test Deliverable | Tasks | Inter-Task Dependencies | Skill Level Required |
| 01 | Test Plan Document | 1. Identify specifications for test components 2. Develop test case for components 3. Test cases 4. Develop test logs 5. Generate test summary reports 6. Record test revisions | A ->  B -> A  C -> B  D -> A, B, C  E -> A, B, C, D  F -> A, B, C, D, E | 2-3 |
| 02 | Test Design Specifications (What needs to be tested) | 1. Features to be tested 2. Analyze modules specifications 3. Create specifications list for each module tested (Include unique identifier, version date, number, author & contact information) 4. Create test identifications (Description of test or any test relationships) 5. Describe pass/fail criteria | A ->  B -> A  C -> A, B  D -> A, B, C  E -> A, B, C, D | 3 |
| 03 | Test Cases (What is tested) | 1. Create unique identifier for individual test cases 2. Test modules and features (Derived from TDS) 3. Record input 4. Record output 5. Describe environmental needs (Anything missing from TDS) 6. Record special procedures 7. List inter-case dependencies | A -> ID#02  B -> ID#02, A  C -> B  D -> B, C  E -> ID#2, B, C, D  F -> B, C, D  G -> ID#02, A | 3 |
| 04 | Test Procedure Steps | 1. Create a unique identifier for test procedures 2. Describe objective of test procedure 3. Describe special requirements for test 4. List procedure steps taken to test components | A ->  B ->  C -> B  D -> ID#03 | 3 |
| 05 | Test Data | 1. Create unique identifier for each test 2. Test modules based on test case procedures 3. Create summary based on acquired test data 4. Generate revision history for each failed test performed | A ->  B -> ID#03  C -> ID#03, B  D -> ID#03, B, C | 2-3 |

**Environmental Needs**

Special requirements for this specific test plan include:

* Communications: Network access for web based application
* Test data (Ex: Student information)
* Test cases for testing modules

Test Data will be provided in the form of a test case word document template.

Special requirements are the developers of this project will be carrying out the testing procedures and test cases.

Multi-parts are divided up between the developer team and will be swapping roles to test each other’s modules. Cross testing will also be fair game to each of the team members. Please refer to the WA1.1 in the engineer’s repository for specifics on team member module testing.

**Staffing & Training Needs**

Training for faculty will be provided with a simple README.txt file or if instructions are unclear, team members will assist faculty in operating the system.

No training is required for the developers who will be testing individual modules.

**Responsibilities**

|  |  |  |
| --- | --- | --- |
| Team Member | Task | Description |
| Patrick | Test Plan Components | Test Plan Components (11-16) |
| Patrick | Test Cases | UI Test Cases   * Features to be tested / not tested * Ensuring all elements are present for testing * Setting risks * Training for UI |
| Patrick | Testing | Will test scheduling module |
| Geoffrey | Test Plan Components | Test Plan Components (6-10) |
| Geoffrey | Test Cases | Advisement Module Cases   * Features to be tested / not tested * Ensuring all elements are present for testing * Setting risks * Training for advisement modules * Module breakdown / explanation |
| Geoffrey | Testing | Will test UI modules |
| Alan | Test Plan Components | Test Plan Components (1-5 & 17-18) |
| Alan | Test Cases | Scheduling Module Cases   * Features to be tested / not tested * Ensuring all elements are present for testing * Setting risks * Module breakdown / explanation |
| Alan | Testing | Will test advisement module |
| Patrick, Geoffrey, Alan | Decision Making | Critical go/no go situations will be discussed as a team and action will be implemented as a team |
| Patrick, Geoffrey, Alan | Plan Strategy | This level of plan will be discussed as a team and action will be implemented based off discussions |
| Patrick, Geoffrey, Alan | Scheduling | Conflicts in scheduling will be reviewed as a team. If one team member is absent, remaining team members will discuss matters |
| Patrick, Geoffrey, Alan | Dependencies | Team will discuss dependencies of modules (Ex: UI, Advisement & scheduling modules) Dependencies may effect scheduling |
| Patrick, Geoffrey, Alan | Environmental Needs | Team will determine if any environmental needs are necessary or required for testing |

**Schedule**

\*Please refer to SCH2.1 Microsoft Project Plan for estimates\*

**Handling Scheduling Slippage**

If a single developer notices scheduling slippage or has concerns about slippage, the entire developing team will be notified ASAP through any means of communication. (Phone, Hangouts, Texts, etc.) Team will refer to SCH2.1 Project Plan for any change in scheduling or any slippage. If major slippage occurs, team will decide in what features will be present or removed from the application in order to avoid major slippage. Decisions will be discussed as a whole. Test dates will be recorded along with related development activity dates from the SCH2.1 Project Plan.