**Phase II Setup**

**Team Aquarius**

**DELIVERABLE 3**: Stage 1 system:

**i) *Deadline***: Stage 1 system **must** be submitted by the deadline (see project schedule). Late submission will not be accepted.

**ii) *Executable*:** Preliminary version (Demo 1) of a substantial part of the fully enhanced system. This includes the improvements made in DELIVERABLE 2.

**iii) *Requirements Implemented*:** Describe the *enhanced* requirements that have been implemented in the Stage 1 demo system. Please refer to Phase II Goals in the customer’s presentation slides (Mandatory Requirements 2-9 and Stretch Requirements 1-5). Please indicate which implemented requirements are *mandatory*, and which are *stretch* (optional).

At its minimum, a requirement should have the following:

- It’s ID

- A description of what the requirement is.

- It’s origin (where did this requirement come from (give specific reference or name of author, etc.)?)

- Supporting example – as needed: screenshot, diagram, etc. to explain this requirement.

As a starter, please see the following, but there is tons on the web on requirements:

<https://hubtechinsider.wordpress.com/2011/07/28/how-do-you-write-software-requirements-what-are-software-requirements-what-is-a-software-requirement/>

**iv) *Test cases*:**

 \_Create test cases, test the Stage 1 system, and log the test results.

 \_If there are test cases that have failed but not resolved by the time of submission of Stage 1 then these should be identified explicitly.

 \_Draw a matrix showing requirements of the Stage 1 system X test cases involved in its verification.

**v) *System design as at Stage 1***(see Fowler):

a) Use case diagram

b) Original Class diagram

c) Enhanced Class diagram

**NOTE**: Each diagram must be accompanied by an *explanation*:

**For the Use case diagram**: If this is different for the enhanced system from that of the original system, then explain the diagram for the original system and also explain the changes needed for the enhanced system. Give rationale for your decisions.

 \_**For the Class diagram**: Similarly, explain the enhancements you needed to make to the original design. Include in your explanation: (1) *what* the diagram conveys and (2) the *rationale* for the design or enhancements (e.g., components that needed to be changed and why; the choices made for the classes; modularity; the correspondence between the use case and customer’s requirements; etc.).

**iv) *Development plans*:**

You are free to use any appropriate project management tool to show development plans.

a) Timeline showing:

1) Achievements to date (Stage 1) with dates. Example achievements (with appropriate dates):

o Phase I system documentation understood.

o Phase I system’s operational aspects understood.

o Improvement to Phase I system identified and logged.

o Improvements to Phase I system completed.

o Test cases created for the Phase I system and system tested.

o Demo 1 enhancement requirements determined.

o Demo 1 enhancement requirements tasks allocated to specific group members.

o Development infrastructure set up.

o Design accomplished in iterations: v1, v2, etc.

o Features implemented (coded and tested): feature1, feature2, etc.

o Etc.

2) Concrete Plan for achieving Stage 2 (Demo 2) requirements.

o Identify major tasks and plot them on the timeline.

3) Abstract Plan from Stage 2 to the Final submission.

b) Agent-Task view (Excel spreadsheet), showing:

1) Task allocation to date (Stage 1): Who has done what

o Tasks as rows; Agent-names as columns

 \_Task description should be self-explanatory (NOT cryptic): e.g., feature description.

2) Planned task allocation from Stage 1 to Stage 2.

NOTE: C++ programming requirement for all, described earlier, must be explicitly shown in the spreadsheet. This can spread over Stage 1 and Stage 2.

**Task Delegation:**

Ore – Qt Testing/Assessments

??? – UML

Fahim – Development Plans/Checking over assessments

**Mandatory Requirements:**

1. Submit 1 report, "Assessment of the Phase 1 Project"

(as specified by the Instructing Team.) - **ORE**

Examples:

* 1. Assessment of the User Interface
  2. Assessment of the Coding Structure & Efficiency
  3. Assessment of the Documentation (ApplePicker)
  4. Proposal to improve (specific, measurable)

2. Add two new graph types to the reporting options

Extension of exiting graphs at sub-categories

Add 2 New Types: Line, plus one of your choice

Graph / Chart entire department, division or list

1. The ability to save the session state
2. Expand Phase I Project by adding the ability to sort by the member’s division. (ApplePicker Mp4)
3. Expand Phase I Project by adding the ability to sort by a user selected list. (ApplePicker Mp4)
4. Fix all errors so that the finished product is error free. **(Wait until Deliverable 4)**
5. Deliver a “User Proof” install file for final project containing
   1. Executable
   2. Documentation
   3. Supporting Files
6. Verify the code deck for operation on Windows 7 Alternatively Windows 10 could be used, if you wish to use Win10 please specify this in your report
7. Navigation of Erroneous Entries

When the CSV file is loaded for the first time, the error dialog box should report total number of errors and "Find Next/Prev" button which jumps on to the next/prev error. As the errors are fixed, the error count should go down. After all the errors are fixed, "Find Next/Prev" should be disabled.

**Stretch Goals**

* 1. Editable text fields (ApplePicker Mp4)  
     To change any fields before importing them
  2. Change / Modify Sort Order (ApplePicker mp4)
  3. Supply a training video demonstrating installation and use of your program. (ApplePicker mp4)
  4. Implement Advanced Search

Open a new dialog box and user should be provided options to choose from, depending on the type of excel

* 1. Verify the code deck for operation on an iOS platform

(in addition to Win 7 or Win 10)