SFWRTECH 3RQ3: Software Requirements and Specifications

Project

Instructor: Sean Watson

Final Deliverable Due: 02/12/2020

Course Policy: Read all the instructions below carefully before you start working on the assignment.

- Remember to include your name and student number on all submissions, as well as the name and student numbers of anyone you worked with.
- Deliverables are due before the end of the day on the due date and will be submitted via Avenue.
- Deliverables submitted later than the noted due date time will receive a grade of 0.
- All sources of material must be cited in accordance with McMaster's Academic Integrity policy.

Project Definition

This project will take you through a course of requirements gathering, specification and test design to ensure that your(hypothetical) successors are ready for success when they begin to implement the product. All deliverables combined are worth a total of 25% of your final grade.

The project is made up of four deliverables:

1. Written project proposal (10%)
2. Written requirements documentation (30%)
3. UML diagrams to supplement written documentation (30%)
4. Python test suite that would ensure any implementation meets the requirements (30%)

The project may be done individually or in groups of up to 4 students. Groups must be defined at the time of submission of the first deliverable and any changes afterward *will not* be permitted.

This document will be updated with more details as further deliverables approach.

Deliverable 1 – Project Proposal

(10% of Project Grade)

Carefully examine the *Dinner is Served* example website. You have two options for your project:

1. Expand *Dinner is Served* to include at least **2** new substantial features. A substantial feature means a major piece of brand new functionality that the current system does not offer, not a minor improvement like a facelift to the menu or updating the selection method for customizing menu items.
2. Build a new application of the same approximate complexity as *Dinner is Served*
    - Ensure it is not too *high* complexity, i.e. a full definition of an autonomous vehicle is not possible in one term
    - Ensure it is not too *low* complexity, i.e. a light switch does not require a full term to define
    - Potential examples of appropriate projects could include: a flight arrival display board, a bank ATM or a Netflix style streaming service.

Submit:

- Your idea, along with approximately two paragraphs explaining how you will tackle the problem as well as why you were attracted to it(why you think it is important and interesting) and why you believe it is both achievable and sufficiently complex.

Deliverable 2 – Requirements Documentation (30% of Project Grade)

For this deliverable, you are requirement to begin building an SRS document in the format given in the Week 5 lecture. The document will not yet be complete, but the sections expected for this deliverable are:

- Introduction
- Description
- System Features
- Interface Requirements
- Quality Attributes

These sections will not be 100% complete at this point in time, but should represent a substantial portion of the project. The next deliverable will iterate upon the work you have done here in order to improve your document.

In addition to the sections listed above, there should be two appendices that show your work for how you decided in the requirement scope. The first of these should include the background research you did in order to decide on your scope and the second appendix will contain a set of scenarios for how you managed to define your requirements. The second appendix should contain at least one of each type of scenario: positive normal, positive abnormal and negative.

Submit:

- Your SRS including all the sections detailed above, with both appendices
- Aim for approx. 20-25 total requirements between functional and non-functional sections
- Ensure that at least one scenario is given for each action you expect a user to take

Check the sample document posted on Avenue to see what will be expected(Appendix A in this document, Stakeholder Analysis, is not required).

Deliverable 3 – Diagrams (30% of Project Grade)

For this deliverable, you are to add use case and class diagrams to your SRS that you developed in deliverable 2. In creating these diagrams, you are expected to uncover several more requirements related to your features that will need to be defined, or clarified. Remember, in an Agile process we always make time for refactoring, so this will be a perfect time to return to your requirements and update where necessary.

You should place these diagrams in the following locations:

- Use case diagram(s) should go in the *Description* section, under a new heading for 'Use Case Diagrams'
- Class diagram(s) should be placed in a new *Data Requirements* section, which should also include headings for an 'Overview' that should explain what the data means and why it needs to be tracked (referring to the diagram(s) where appropriate) and one for 'Class Diagrams' which will contain the diagrams themselves.

Deliverable 4 – Tests (30% of Project Grade)

You should write a full Python test suite to test any potential implementation against your requirements. Each requirement should have at least one related test and each test should be related to at least one requirement. Each test should be commented with the requirement(s) that it is responsible for testing. Some requirements may not be possible write automated tests for, these requirements should be enumerated in a new SRS appendix, Appendix C in which the requirements are mentioned and it is explained why testing is not feasible. Make any further updates your SRS that you think are needed to improve it at this point. You will need to submit:

- A Python .py file containing all your tests or a link to a github page where your code is checked in
- Your updated SRS

Sample tests will be given during the week 11 lecture.