MET CS 682 – Module 3

Assignment 3 – Requirements

*Requirements* are central to IT: in other words, the identification of what's wanted and needed from the intended system. Requirements analysis begins with a (sometimes vague) understanding of wants and needs, and it increases in specificity. This exercise will give you practice and feedback on the requirements analysis process. Please note the page limitations. You may include as many appendices as you wish but these will be read only on an as-needed basis. As always, keep aware of deadlines, the grading criteria (the evaluation matrix in “Syllabus/Grading Criteria for Homework”), and plagiarism rules.

***Scenario:***

You are tasked with creating *WordBlank*, a fully automated assessment system for Massive Online Open Courses ("MOOCs"). The test format consists of paragraphs with missing words. The following is an example from a poetry class:

Two roads \_\_(1)\_\_ in a yellow \_\_(2)\_\_,

And sorry I could not travel both

And be one traveler, long I stood

And looked down one as far as I could

To where it bent in the \_\_(3)\_\_;

Then took the other, as just as \_\_(4)\_\_,

And having perhaps the better \_\_(5)\_\_,

Because it was grassy and wanted wear;

Though as for that the passing there

Had worn them really about the \_\_(6)\_\_,

For each missing word, the test designer designates the correct word, as well as a set of alternatives for partial credit. The system allows the test maker to allocate the test score in accordance with various algorithms. Included are grading, the presentation of grades, and additional feedback for students. Test designers have the option to include a means for obtaining feedback from students within the test.

Some Notes and Hints Before You Begin:

* As usual, the notes are a primary source for explanations and examples; we also encourage you to do outside reading and research to gain additional perspective.
* **It is important that the entire solution is consistent. You will want to iteratively review all of the nine sections to make sure that together they are consistent as a single solution.**
* Feel free to include notes for your facilitator (that are not actually part of your paper). A good way to do this is to insert, for example, “Note 3 to facilitator” and list the notes in the appendix.

***Provide requirements, including supporting sketches, for a WordBlank system:***

Remember that *you* are setting the requirements: you are not being given whole requirements, just the outline above. Your requirements should be consistent with it. But you may provide additional detail and expand the scope. You can also require intelligence from the system such as advice and checklist popups.

1. **Overview:**  Provide a one-paragraph overview which introduces the system.
2. **System Users:** List and describe the different types ofsystem users (actors) and very briefly how they will use the system.

1. **Functional Requirements:** In one single space page or less 12-point type, specify key functional requirements to be provided by the system to the various types of users.

In doing this, you are not required to distinguish between high-level and detailed requirements. These requirements will be used for the construction of use cases – the next part of this homework.

1. **Use Cases:** Specify two use cases in tabular format, showing actors, preconditions, actor actions and system responses.

You may use copies of the following table if you wish, also a suggested example is on page 257 in the textbook. Each of these use cases should have approximately 3-6 steps.

|  |  |
| --- | --- |
| **Use case Name** |   |
| **Actor:**  |   |
| **Description:** |  |
| **Precondition:** |  |
| **Step #** | **Actor** | **System** |
| **1** |   |  |
| **2** |  |   |
| **3** |   |  |
| **4** |  |  |
| **5** |  |  |
| X |  |  |
| **Alternate Courses:** |  |
| **Implementation Constraints:** |  |

1. **State Transition Diagram:** Develop a state-transition diagram for the system at a high level. Limit this to 3-5 states. Note that sub-states of one of these states will be added in question 6.

You can use the Visio or PowerPoint templates below as a start if you wish. If you would like to use another tool, check with your facilitator in advance.





1. **Sub-States:** Expand one of the states in the previous question into at least two sub-states. Show the transitions that affect these sub-states.
2. **GUI Sketches:** Create GUI sketches for two screens for *WordBank*.
3. **Non-Functional Requirements:** Specify what you consider the two most important non-functional requirements. Describe your choices briefly and explain why you have considered these most important

1. **Organizing Requirements:** Requirement can be organized by use case, GUI, or functional requirement to name just three of many options. Describe and contrast two of these three ways in which the requirements for *WordBank* could be organized. Give specific advantages and disadvantages of each for *WordBank*.

Notes/Hints:

**Overview**

* Do not underestimate the time required to write a clear overview that is short enough to be readable, yet long enough to convey what the system does, structured appropriately.
* You may want to come back to this section at the end to make sure it is consistent with the rest of your solution.

**Functional Requirements**

* You do not have to go into the finest requirement but make sure that you describe the major functionality. Enumerate and describe your functions systematically.
* Make sure to understand the difference between functional & non-functional requirements.
	+ Functional: What the system shall do.
* You may want to do some outside research to see relevant examples of how functional requirements are defined for systems. You can include your findings in the Appendices section.

**Use Cases**

* The use case name, primary actor, precondition, and action/system response steps should be shown at minimum.
* Use cases can also be described, in a non-tabular form. See the “Main Functions and Use Cases: An Introduction” and the “Use Cases for Video Game” sections of the Module 3 notes as examples.
* Additional information is available in the in the tabular-narrative forms of Figure 7-12 on p 257 and Figure 7-13 of the textbook.
* The lecture notes and all of Chapter 7 in the textbook provide several of examples of use cases. Many of the use case examples in the book could be decomposed into several smaller use cases and many of the textbook’s steps are themselves use cases. For example, Actor action Step 1 in the use case on page 259 of the textbook (“The club member provides …”) should be considered a reference to a use case. This is a “uses relationship”, described on page 249. Avoid branching in use cases, if possible: use only if necessary.

**GUI Sketches**

* There is no need for elaborate GUI’s here. Decide what information should be contained in each screen and draw rough mockup sketches. Use single rectangles for GUI windows, specify prompts to users, and use rectangles for text fields where users enter data. Indicate system output text to users as a series of x's. See the “GUI Overviews” section on this week’s module for examples.

**State Transition Diagrams**

* If you would like to, you can show both parts 5 and 6 within a single diagram.
* A good place to start is to review Module 3 state diagram example at the end of the “GUI Overviews” for rentals. Note how some of the states which are shown have sub-states.
* Module 4 also has some good examples to consider including section “State-Transition Diagrams” “State Models”, “States and SubStates”, “OnlineShopper”
* You may want to do some outside research to see how state transition diagrams are developed and used. You can include your findings in the Appendices section.
* Make sure to note what your diagram is based on from research as there are some variations within the approaches. For example, if your diagram is based on what you learned from the module notes, make sure to reference the appropriate slide.

**Non-Functional Requirements**

* The “Non-Functional Requirements” section of this week’s notes provides examples.
* Make sure to understand the difference between functional & non-functional requirements.
* Consider constraints vs. non-functional quality requirements.
* You may want to do some outside research to see relevant examples of how non-functional requirements are defined for systems. You can include your findings in the Appendices section.

**Organizing Requirements:**

* A good place to start is to look at the “Methods of Organizing Requirements” chart within the “Introduction of Functional Requirements” section of this week’s notes.
* You will want to think about how you approached this assignment, as well as do some research to help you support your points.

## ****Organizing Your Response****

You will want to organize your response as a paper write-up. The following are suggestions to a clearly organized response.

* Include a title page, and a table of contents.
	+ Although obvious, make sure to include your name in your assignment solution.
* If you want to include some of the elements from the assignment directions and characteristics, organize these in the introduction section, but only include sections which you feel add to the clarity of your response.
* Make sure to use proper headings and sub-headings in organizing your response. For example, this particular assignment has nine parts, with multiple sub-sections.
* For diagrams make sure to paste them into your solution within the appropriate section and that they are legible. If needed, provide the file separately for reference (we only accept Microsoft Office or Visio files, unless approved in advance with your facilitator).
	+ Hand-drawn and scanned diagrams are not accepted, we want you to practice using modern design tools.
* Make sure to use proper referencing in your paper. We suggest is APA format, but other formats are fine as long as it’s clear in distinguishing your work from work of others in your response, be mindful of plagiarism rules.
	+ Make sure to include a reference page, but also show how these references are used within your response.
* Include your last name in the file name of the assignment.
	+ (Example: SmithMichael\_CS682Assignment3.docx)