

Term project part 1

The purpose of this exercise is to get practice of specifying requirements. In practice, requirements—what users want and need—are uncovered in an iterative process, with the focus of determining what users really want and need.



February 1, 2024

Mahim Choudhury

BOSTON UNIVERSITY

Last Updated 1/26/2024

Table of Contents

[1. Scope and Scenario 1](#_Toc157506006)

[A2 (ChatGPT regarding Scope and Scenario) 1](#_Toc157506007)

[2. Overview/Mission Statement 2](#_Toc157506008)

[3. User Stories 2](#_Toc157506009)

[3.1 First User Story 2](#_Toc157506010)

[3.2 Second User Story 2](#_Toc157506011)

[4. Functional Requirements 2](#_Toc157506012)

[5. Use Cases 2](#_Toc157506013)

[5.1 First Use Case 2](#_Toc157506014)

[5.2 Second Use Case 3](#_Toc157506015)

[6 State Transition Diagram 3](#_Toc157506016)

[6.1 Sub-States 3](#_Toc157506017)

[7 GUI Sketch 3](#_Toc157506018)

[8 Non-Functional Requirements 3](#_Toc157506019)

[Appendix 3](#_Toc157506020)

[References 4](#_Toc157506021)

[Evaluation 4](#_Toc157506022)

[1 Hints 5](#_Toc157506023)

[8.1 Overall Assignment Notes 5](#_Toc157506024)

[8.2 Notes to Selected Grading Criteria 5](#_Toc157506025)

[8.3 Mission Statement/Overview 5](#_Toc157506026)

[8.4 Functional Requirements 5](#_Toc157506027)

[8.5 Use Cases 6](#_Toc157506028)

[8.6 State Transition Diagrams 6](#_Toc157506029)

[8.7 GUI Sketch 7](#_Toc157506030)

[8.8 Non-Functional Requirements 7](#_Toc157506031)

MET CS682 ASSIGNMENT 3

TERM PROJECT PART 1: REQUIREMENTS

Mahim Choudhury

The purpose of this exercise is to get practice of specifying requirements. In practice, requirements—what users want and need—are uncovered in an iterative process, with the focus of determining what users really want and need. You will not necessarily specify requirements for an entire system, but this is your start. Go as far as your imagination takes you

1. USE THIS TEMPLATE: Please respond by using this Word template, including the text boxes, leaving the headings and the gray text unchanged—but exclude the *hints* section (at the end) from your solution.
2. OBSERVE LIMITS: Observe the page and paragraph limitations of the regular text outside the text boxes; however, you may include as many appendices as you wish. All appendices should be referred to in the main text.
3. USE AI GENERATION **but ONLY for question 1**. We would like you to answer this question using a chatbot because we feel that this is the way such writing will be created going forward. We’d like you to practice good use of AI generation, including your own verification, re-prompting, editing, and improvement. You are required to document your chatbot interaction in the text box provided.
4. OUTSIDE RESEARCH: You are also encouraged to do outside research to support your response but use the *References* section as indicated and observe plagiarism rules carefully, including the citation of sources, the use of quotes, and acknowledgement of modifications of external sources.
5. NAME YOUR FILE: Include your last name in the file name of the assignment. (Example: SmithMichael\_CS682Assignment1.docx)
6. EVALUATION: You will be evaluated on your value added to existing and AI-generated material, including your prompts, as described in the evaluation matrix. There is no credit due for unimproved material generated by an AI generator. We will provide feedback comments and will use our best judgement in terms of the evaluation criteria listed.
7. HINTS: Hints are provided at the end of this document.

You are to specify a system called *EmbeddedAI* into a system of your choice which will extend its functionality in some specific way by using generative AI like ChatGPT. (Scenario must be different then assignment 1 and 2). This system must incorporate some element of IoT (internet of things)

* *You* are setting the requirements for *EmbeddedAI*
* Form your own vision and scope. As a starting point consider your own experience and how such a system might help improve this experience.
* It will be helpful to focus on a specific setting – for example, the system could be geared toward helping plan and manage a trip. We suggest linking to your personal experience, industry research, or something that you have business knowledge and interest in to simulate your thinking. Leverage top down and bottom-up analysis outlined in the module to aid your thinking.
* Consider the perspective of more than one actor.

# Scope and Scenario

In no more than a few sentences, outline your scope. Provide a scenario and what your project will focus on.

Your response replaces this.

## A2 (ChatGPT regarding Scope and Scenario)

Show your most relevant prompt to chatGPT and the response which was most relevant to your solution.

# Overview/Mission Statement

One paragraph, about four to six sentences.

Your response replaces this.

# User Stories

Provide two (2) user stories from two different types of actors (system users) in the format provided below. The two user stories should come from two different actor’s point of view. Replace all of the “<…>” parts.

## First User Story

As a <type of user>, I want <some goal> so that <some reason>.

## Second User Story

As a <type of user>, I want <some goal> so that <some reason>.

# Functional Requirements

In about three quarters of single-space page (using 12-point type), specify a total of 12-15 key functional requirements. These requirements define your chosen scope within the whole system. Each requirement should be a 1-3-word title followed by a single sentence.

* 1. Your response replaces this.
  2. Your response replaces this.
  3. …

# Use Cases

In the tables below, specify two important detailed use cases, showing actors, preconditions, actor actions, and system responses. system. Each of these use cases should have approximately 5-6 steps, each step having an actor and/or system component.

## First Use Case

|  |  |  |
| --- | --- | --- |
| **Use case Name** |  | |
| **Actor:** |  | |
| **Description:** |  | |
| **Pre-condition:** |  | |
| **Step #** | **Actor** | **System** |
| **1** |  |  |
| **2** |  |  |
| **3** |  |  |
| **4** |  |  |
| **5** |  |  |
| (more) |  |  |
| **Alternate Courses:** |  | |

## Second Use Case

|  |  |  |
| --- | --- | --- |
| **Use case Name** |  | |
| **Actor:** |  | |
| **Description:** |  | |
| **Pre-condition:** |  | |
| **Step #** | **Actor** | **System** |
| **1** |  |  |
| **2** |  |  |
| **3** |  |  |
| **4** |  |  |
| **5** |  |  |
| (more) |  |  |
| **Alternate Courses:** |  |  |

# 6 State Transition Diagram

Supply a state transition diagram for the system as a whole—or as near to it as possible. Limit this to 5-7 states if possible. Include states other than those corresponding to a GUI. Note that sub-states of one of these states will be added in question 6. You may use Draw.io, Visio, LucidChart, or another design tool of your choice (please check with your facilitator in advance if you are using a different tool).

Your state transition diagram and notes replace this.

# Sub-States

Expand one of the states in question 4 into sub-states consistently with at least one of your detailed use cases. Limit this expansion to approximately 4 sub-states. Show the transitions that affect these sub-states.

Your sub-state transition diagram and notes replace this.

# GUI Sketch

Create a GUI sketch of one important/complex screen of the system. You can use Draw.io, LucidChart, Visio, <https://www.mockflow.com/>, or another design tool of your choice (if you are not simply inserting screenshots to answer this question, please check with your facilitator in advance).

Your GUI sketch and notes replace this.

# Non-Functional Requirements

Specify the two most important non-functional requirements for your system. Each requirement should a single sentence. Explain why you consider these most important. No more then 1/2 of page total for each of the two non-functional requirements.

Your response replaces this.

# Appendix

This section is optional, place any notes here on how you have used the references or any other notes you would like to make.

# References

Show that you used a wide variety of resources by listing them below and clearly indicating in the body above where you used each one (inserting [1], [2] etc.). We suggest using APA format, but other formats are fine as long as your paper clearly distinguishes your work from work of others, being mindful of plagiarism rules.

[1] your first reference replaces this

[2] …

# Evaluation

A picture containing text, screenshot, number, parallel

Description automatically generated

**Please do not include Hints section in your solution.**

# Hints

## Overall Assignment Notes

* 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. In the real world, you would be interviewing people to carry this out but that is not possible in a class setting.
* If you run out of space (after careful editing), you may provide an additional explanation in the appendices section to support your design decisions or to clarify your choices.
* As in the real world, the quality of your work is subjected to judgments, not mathematical exactness. Help your facilitator make favorable judgments by being as clear and specific as possible. “Clarity” is one of the criteria.
* 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.

## Notes to Selected Grading Criteria

**Clarity**: Part of requirements analysis is to understand what your customer wants and needs (and, as we discussed, this is not easy)—so be sure that you understand the problem. Your response will be clearer as a result. Are your requirement statements clear enough so that someone could create a design from them? Are your names (e.g., of states) expressive enough so the reader can understand them? Explain throughout. **Consistency (an aspect of clarity)**: Make sure that your entire solution is consistent. For example, did you introduce some functionality in the use case that you didn't mention in your functional requirement? Then go back to part 3 and make sure to revise it.

**Relevance**: Review the scenario and make sure to stick to it. If you need to expand on them a bit that’s fine; provide a brief introduction with your assumptions, but make sure to stay within the initial framework of the scenario. For example - it's best to describe your approach and/or scope for the diagrams.

## Mission Statement/Overview

**Technical Soundness:**

* A high level overview. Logically, one would write this first, and then the other parts would flow from it. However, our minds do not necessarily operate very logically. To stimulate your thinking about the mission, you may first want to think about a user story, or even just an actor.
* Avoid details—they come later.

**Clarity**

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

## Functional Requirements

**Clarity**

* You do not have to go into the finest details of the requirements, but make sure that you describe the major functionality. Enumerate and describe your functions systematically.
* Look to organize requirements with headers and sub-headers, for example by actor and or functional area.
* If a functional requirement sounds a little bit non-functional, explain your reasoning as to why you decided to keep it functional in a note (as sometimes these are borderline).

**Technical Soundness:**

* Make sure to understand the difference between functional & non-functional requirements.
  + Functional: Reasons for the system and what it’s meant to do.
  + Non-functional: Needed or wanted but not the reasons for the system.

**Thoroughness and Coverage:**

* You may need to come back to this after you complete the assignment to see what you missed, for example, are there key features you outlined in the GUI, state transition or use case which were not covered here?

**Relevance:**

* 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

**Technical Soundness:**

* Complete the use case template we give you in the assignment.
* Additional information is available in the tabular-narrative forms of Figure 4-13 on page 143 of the textbook and in the “Use Cases” section of the Module 3 notes.
* The use case needs to show appropriate sequence (actor/system)
* Clear understanding in difference between constraints and pre-conditions. The use case itself should follow one path as best as possible.
* Avoid branching in use cases, if possible—use only if necessary.

**Thoroughness and Coverage:**

* Complete the use case template we give you in the assignment. This is where we ask you to describe the use case, who uses it, what might be the pre-conditions, and alternate steps.
* Research can be applied by looking at similar systems (i.e. here is what I found and how it relates to my design). This makes your solution real and is something that analysts need to do to understand what technology will be used to implement this.

**Clarity**

* Consider consistency and relevance to the scenario.

## State Transition Diagrams

**Technical Soundness:**

* A good place to start is to review the “State Machine and State Transition Diagrams,” “Components of State Transition Diagram,” and “State Transition Diagram—An example” sections in this week’s lecture notes.
* Consider your use case as a way to start thinking about state transition, ‘what the system does’ are the states of your system, and ‘what the actor does’ could be the events that trigger the transitions, and then look to functional requirements and user stories to add detail.
* Avoid focusing too much on use cases themselves, you want to capture the overall main steps of the system.
* Make sure all transitions are labeled with events. If a state transitions into a state (i.e. search completed) but is not significant to be it’s own state, use it as an event within guard conditions.
* Please see pages 221-227 in the text on Behavioral State Machines – however, please note that this requires Object Oriented approach which we will look at next week. For now, review the mechanics of the state machine as it applies to this assignment.

**Thoroughness and Coverage:**

* Make sure all key functionality is covered (use cases, user stories, what was discussed in the system overview)

**Clarity**

* Make sure to show what the composite state(s) are, in other words note that these contain sub-states.
* Are diagrams clear to read (i.e. no overlapping lines, no non-polished designs)?
* Diagram should be consistent with requirements (i.e. functional, use cases).

## GUI Sketch

**Relevance:**

* Decide what is the most important, potentially complex screen, consider what it should contain and draw a rough mockup sketch.

**Technical Soundness:**

* See the “GUI Mockup Example” section in this week’s module for examples.
* Chapter 10 in the text covers material on human-computer interaction layer design.
* GUI is fairly straight forward; many students have fun with this. As mentioned previously, Visio and Lucidchart have a wireframe template, but you can also check out balsamiq.com, wireframepro.com or mockflow.com

**Clarity**

* Provide “Sticky Notes” to describe functionality which may not be obvious.

## Non-Functional Requirements

**Technical Soundness:**

* Make sure to understand the difference between functional and non-functional requirements.
* This is the "How" the system is implemented (i.e. quality requirements, constraints). Use references to support your choices. Think about what is most important and why; use of references should help here.
* The “Non-Functional Requirements” section of this week’s notes provide examples.
* Please see secondary readings for Week 3 in the textbook for additional examples of non-functional requirements.

**Thoroughness and Coverage:**

* 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.

**Relevance:**

* Review your entire solution after completing it—you will uncover additional considerations. Check for consistency.