**Software Engineering Project Checklist**(Not highlighted = Done; Highlighted in green = Need to do but have questions  
first; Highlighted in yellow = Need to do)

**Cover Page**

(This will be for your report only. You do not need create a cover page for your GitHub repository)

**Individual Contributions Breakdown**

A. This will be placed in the Appendix (You are not completing an appendix for your repository)

**Introduction**

* Capture reader’s interest
* Build case through logic
* Topic sentence/thesis statement
* Scope/ limitations of the project
* Identify anyone and everyone who has interest in this system (users, managers,  
  sponsors, etc.). Stakeholders should be humans or human organizations.
  + Actors and Goals
    - Identify the roles of people or devices that will directly interact with the system,  
      their types (initiating vs. participating) and the goals of the initiating actors
* **Writing Assignment 1: Individual Learning Plan**

The focus of the second assignment is on devising a personal learning plan. This assignment is to be completed by each student.

***Writing Assignment 1: Individual Learning Plan***

Note: This is an individual, not a group, assignment. The focus of this assignment is laying out an individual learning plan for each member of your team. This is to help ensure you are being proactive in learning new techniques and technologies needed for your project. This learning plan includes:

* + What are the specific technical skills that are needed for the project that you need to

learn to succeed on the project?

* + For each of these skills, what resources will you use to learn what is needed for the

project? Note: you do not need to be an expert, but you need to know enough to be

effective as part of your team.

* + For each of these skills, what is your timeline for learning this skill? This may have a

single deadline, or multiple deadlines, depending on what resources you are using and

when this information is needed.

Project Template for Plan:

\*\*\*\* This an object import from MS Project or something similar. Do NOT use a simple Table.

* **Writing Assignment 2: Getting Started**

The focus of the first assignment is on the materials needed to get started. This includes the team charter and name; an initial collection of personas, scenarios, and features; configuration management rules; rules around the implementation, including the technology stack, onboarding information, and linting information; and an overview of how the system will be tested. Your limitations:

***Writing Assignment 2: Getting Started***

The focus of this assignment is on whatever is needed to get started. This includes:

* + Team name
  + Team charter/ Team Lead selected
  + 2 initial scenarios
  + 3 initial features, mined from the scenarios (more is fine)
  + User stories which are mined from the scenarios and relate to the features (at least one for each feature)
  + Tasks for the user stories (at least one per user story, but it should be more)
  + Configuration management rules: how will GitHub be used for the Team project? What are the rules around commits and branches? What is expected of commit messages, and how will this be enforced (if it is)? What type of workflow will be used (feature branches? GitFlow? No branches?)
  + Code rules:
    - * what is the technology stack? How can new team members get set up (onboarding)?
      * What linters and analysis tools will be used?
      * Are there other ideas for tools that need to be explored? What technologies do some team members need to learn?
      * How has this been factored into the project plan?
  + Testing rules: at a high level, how will you test your product? You need to automate at

least part of this this semester, so how can you automate? How does testing interact

with commits (always test before commit? all tests pass? all tests pass before merge?)

What tools will be used?

* + Licensing- you must identify proper Creative Commons License and POST it to your GitHub, you must refer to the license holder.
  + Requirements
    - You will need both System Requirements  
      1. Functional (Minimum 5)  
      2. Non-Functional (Minimum 5)
  + Customer Statement of Requirements (CSR)
    - not written from the developer’s perspective, describing the features of the planned  
      system (Minimum 5)
  + Make Sure you use the following template:
    - "As a \_\_\_ (who) I want to \_\_\_ (action) So That \_\_\_ (goal)
  + Describe the problem that your customer is facing and his or her suggestions about  
    how a software system could help
  + Functional Requirements Specification
    - Stakeholders
* **Writing Assignment 3: Getting Started Review**

The focus of the third assignment is a review of writing assignment 1 and 2. Each team reviews the

material submitted by one other team (This means switch with another team in the class). Each team will provide feedback- at least 2 (pieces of feedback) from each member. To share, decide on the best course of action and approach. However, there MUST be 5 professional documented pieces of feedback from each team member to the other team. Hint: MS Teams/Word with Track Changes would be the ideal way of doing this

* **Writing Assignment 4: Architecture and Design**

The focus of the fourth assignment is on architecture and design. This includes an overview of

the architecture; architectural styles; logical components; important classes, files, or functions; a

description of the database; and documentation of non-obvious design decisions.

* You will need to identify 1 total scenario of your system to diagram
* You will need diagram
  + Use case
    - Show how your system requirements (identify at least 1 scenario) map to your use cases. Calculate the  
      priority weights of your use cases. The use cases with the highest priority  
      should be elaborated and planned for the first demo.
  + Sequence Diagram
  + Infrastructure Diagram
  + Process flow/ activity
    - Your event flows must show step-by-step every action that the initiating  
      actor (“user”) can take while running the given use case.
  + Planning poker results (screen capture/ export results)
  + Your Estimations from planning poker and the Azure Scrum Board
  + For a given use case, show step-by-step how the user enters information and how  
    the results appear on the screen. This should be 1 aspect of the system that is functional. If your system is not functional, do your best to diagram the activity.
* **Writing Assignment 5: Architecture and Design Review**

The focus of the fifth assignment is a review of writing assignment 4. Each team reviews the material submitted by one other team. Include any comments.

**Writing Assignment 6: Testing**

The focus of the sixth assignment is on testing the software being developed. Since an overview

of testing was included in Assignment 1, this focuses on specific steps to be performed for

manual and for automated testing. For this, you should identify a specific scenario. You will not need to build the test, ONLY provide steps.

**Writing Assignment 7: Getting Started Revision**

The focus of the seventh assignment is on responding to the review feedback received from

Assignment 3. Students respond to each item of feedback, indicating how they have incorporated

the given feedback. Students also have the option to not use this feedback, but they need to

explain why they are choosing not to (per each item, not overall).

**Writing Assignment 8: Architecture and Design Revision**

The focus of the eighth assignment is on respond to the review feedback received from

Assignment 5. Students respond to each item of feedback, indicating how they have incorporated

the given feedback. Students also have the option to not use this feedback, but they need to

explain why they are choosing not to (per each item, not overall).

**Writing Assignment 9: Wrapping Up**

The focus of the ninth assignment is on providing end-of-project wrap-up material. This includes information about high-quality commits and tests, as well as the selected license for the product. All content must be moved into GitHub and organized as a final report. Map all images, apply figures where necessary, and include all images supporting docs, etc in your GitHub repository