

# Project 1 Preliminary Submission

## 1. Project Overview

- 1.1. GitHub repository link (Make sure that the GitHub repository is public or shared with the instructor and TA)

[https://github.com/GarethCarew/CS440\\_GroupProject](https://github.com/GarethCarew/CS440_GroupProject)

- 1.2. Brief description of the system (1 paragraph is enough)

The system we chose to design is a to-do list application. The website currently supports adding, editing, and removing items from a to-do list with the ability to cross off items without deleting them. You can also currently access different lists by editing the url.

- 1.3. Technology stack justification (description of the technologies you used and the rationale behind choosing them)

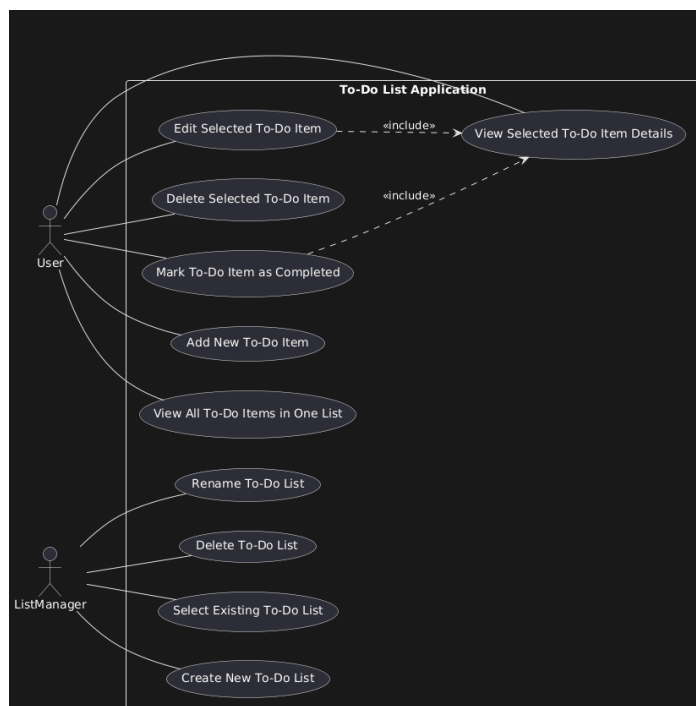
We chose to use Node.js/Express.js as our main framework because it was what we were most familiar with. We also chose to use MongoDB as our database for the same reason.

- 1.4. Architectural decisions and rationale (how you organized your code)

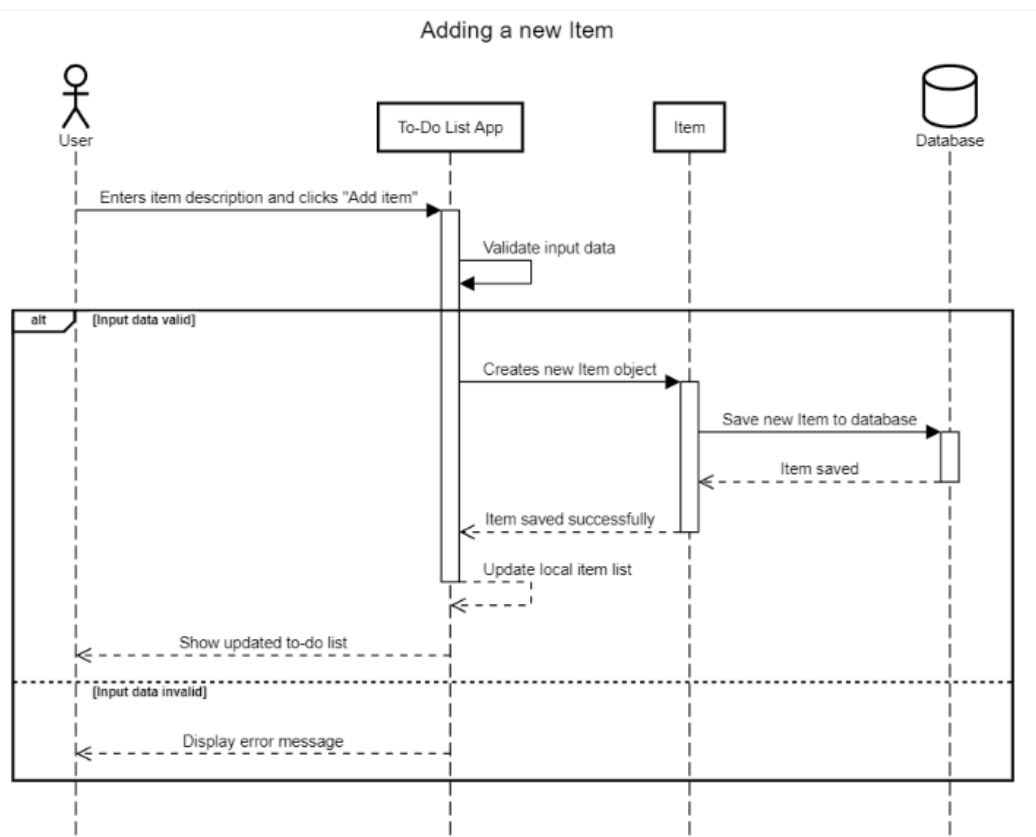
We chose to follow the recommended format for structuring our code with node/express. We have a main index.js file and our index.ejs file in the /views/ folder. We also have the /public/ directory which currently holds the css and a script for the dynamic elements of the ejs file.

## 2. UML Documentation

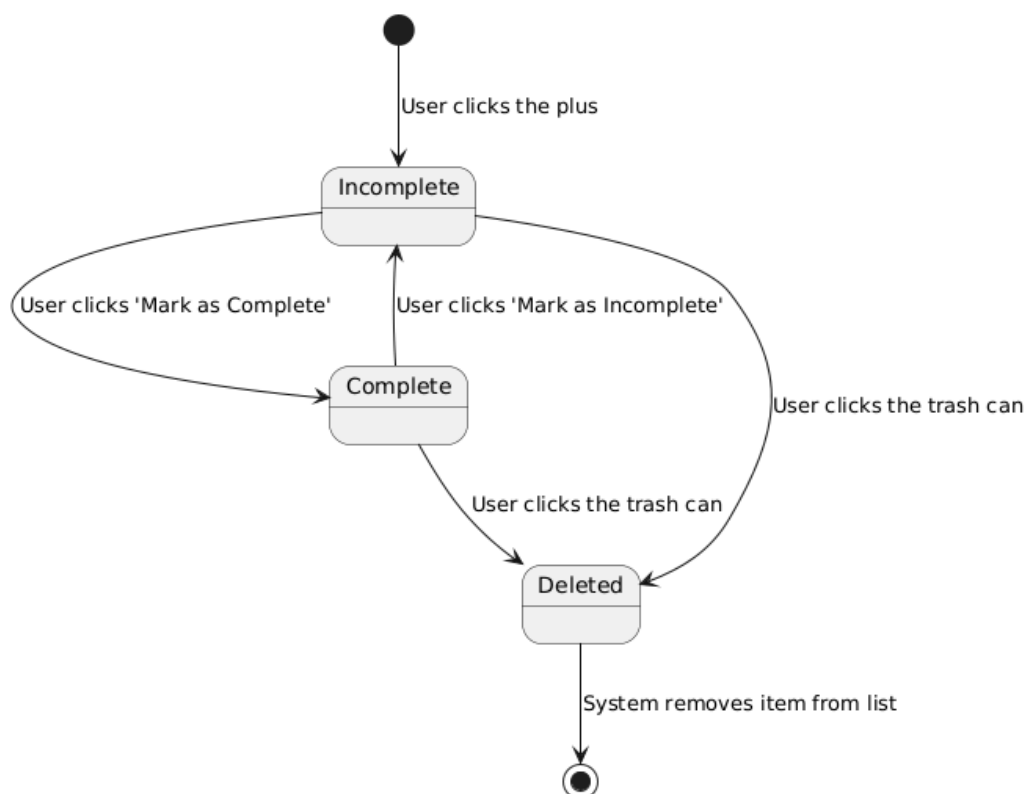
- 2.1. Use case diagram



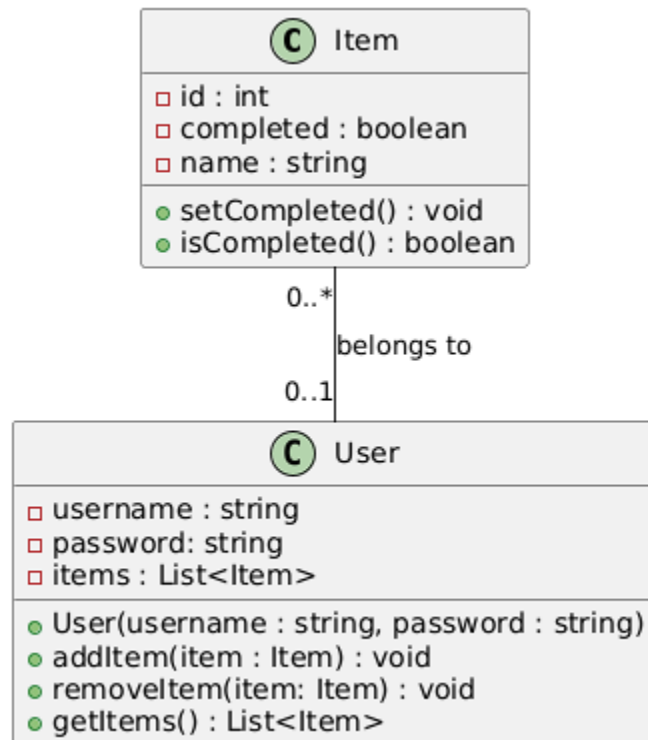
## 2.2. Sequence diagram



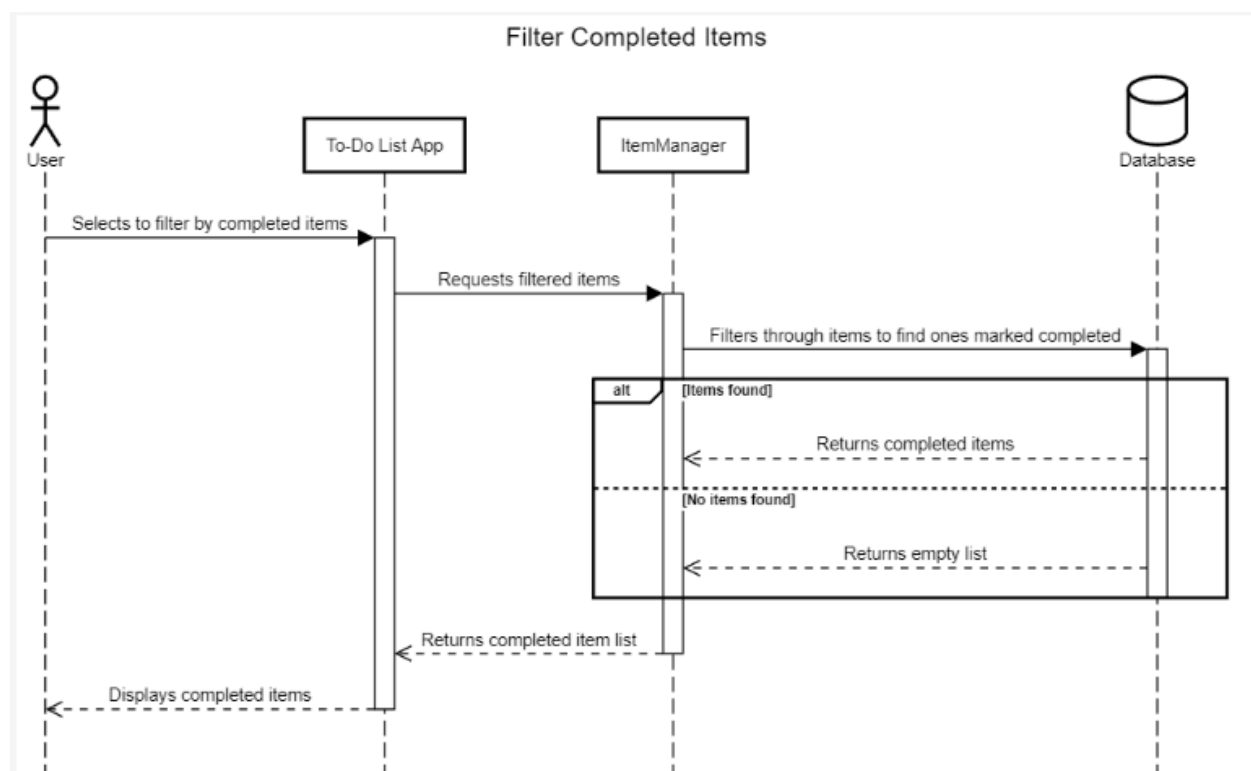
## 2.3. State machine diagram for how the items would work



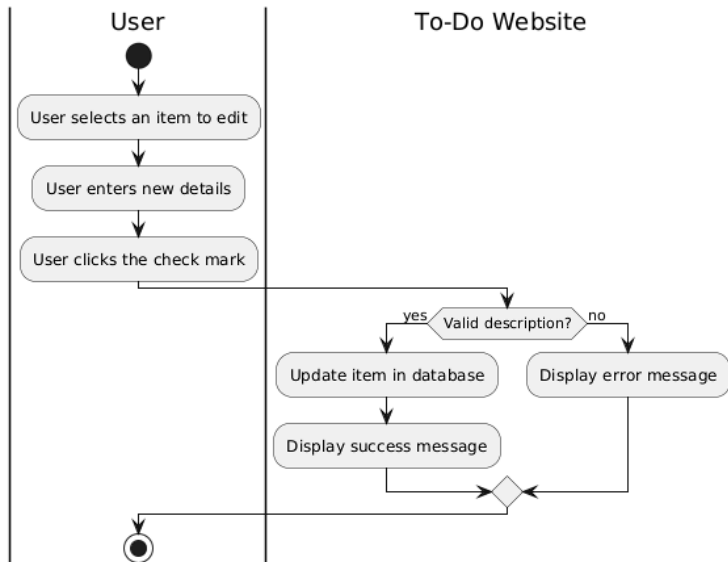
## 2.4. Class diagram



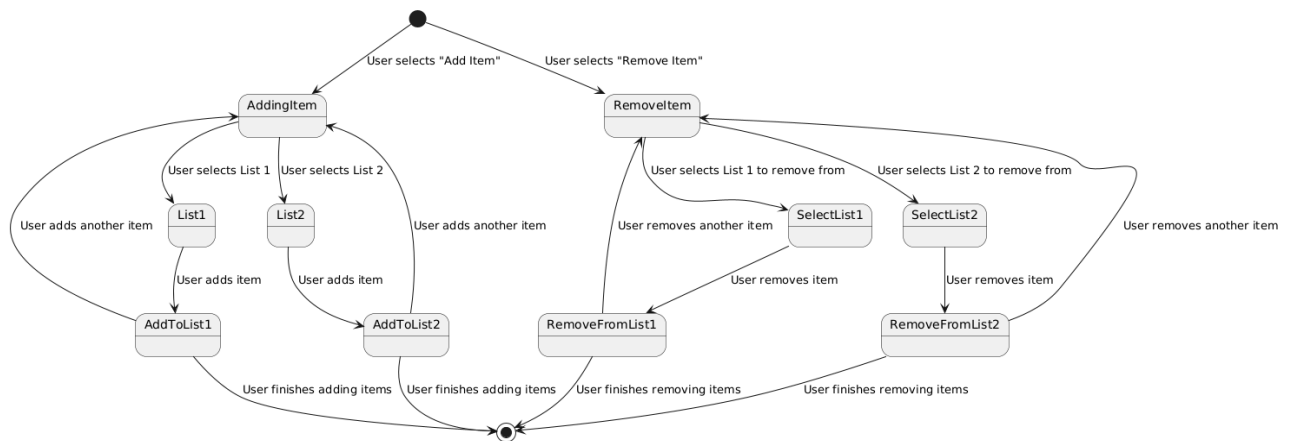
## 2.5. Sequence diagram #2



## 2.6. Activity diagram



## 2.7. State machine diagram for reminders about to-do items



3. Video link

<https://youtu.be/b85P0OpvDyU>

4. Ghat-GPT Comments/Responses

4.1. [Hardcoded MongoDB Connection String](#)

4.1.1. Adapted suggested solution of using .env file.

4.2. [Inconsistent Naming for MongoDB Model](#)

4.2.1. Implemented recommended solution.

4.3. [Wrong Mongoose Query Method](#)

4.3.1. It may not have been a method previously, but it exists now and works as is. No changes.

4.4. [Missing Error Handling](#)

4.4.1. Implemented recommended solution.

4.5. [Inefficient Reload After Database Changes](#)

4.5.1. Changed js to dynamically update the page so a reload on updating tasks is not needed.

4.6. [Global Variables Instead of Function Parameters](#)

4.6.1. I am accessing the value as a global variable since it never changes and is const.

4.7. [Security Issues](#)

4.7.1. Implemented recommended solution.

4.8. [HTML Structure Issues](#)

4.8.1. Combined google fonts link tags.

4.9. [UML Analysis](#)

4.9.1. Use Case Diagram, fixed the use cases to be more descriptive through more specific wording

4.9.2. Sequence Diagram, added handling of multiple outcomes such as the invalid addition of an item into the list

4.9.3. State Machine Diagram, included a more specific listed event to move from state to state

4.9.4. Class Diagram, code calls for item not task, kept unchanged

4.9.5. Sequence Diagram 2, included more cases that could be a path the program takes

4.9.6. Activity Diagram, as the activity diagram is about the editing of an item, it does have similarities to the other diagrams, but does not need the showing of making the item complete, unchanged

4.9.7. State Machine 2, rewrote and changed all the information to completely show what is happening in the system with multiple lists

4.10. [Typos and Formatting Issues](#)

4.10.1. Implemented recommended changes.

- 4.11. [Technology Stack Justification Lacks Explanation](#)
  - 4.11.1. Chat-GPT's complaint is that we were not explanatory enough for why we chose to use what we did. There is not any more reason that we used Node.js/Express.js/MongoDB other than that we were familiar with it. Left as is.
- 4.12. [UML Section Missing Diagram References](#)
  - 4.12.1. Unsure exactly what Chat-GPT is looking for here. Each diagram is placed next to a label for what type of diagram it is.
- 5. Claude Comments/Responses
  - 5.1. [MongoDB connection string contains exposed credentials in plain text \(uri variable in index.js\)](#)
    - 5.1.1. Duplicate.
  - 5.2. [No error handling for database operations in routes \(add, update, delete\)](#)
    - 5.2.1. Duplicated, but mentions other DB operations. Implemented try/catch for those as well.
  - 5.3. [No input validation/sanitization for user inputs](#)
    - 5.3.1. Security is not a large concern for a temporary application for a class project and would take extra time that would be better spent elsewhere.
  - 5.4. [No middleware for error handling](#)
    - 5.4.1.
  - 5.5. [mongoose.Schema definition is overly simplistic and missing required fields](#)
    - 5.5.1. As far as I am aware the only requirement for a schema is to have at least one field. Our schema is 'overly simplistic' because all we need to store is the text for the item, as a UID is automatically generated by MongoDB.
  - 5.6. [All routes are in index.js, violating separation of concerns](#)
    - 5.6.1. As the scope of the project is relatively small and only for one semester, I am not concerned with SoC. If I were planning on continuing development/support I would work on breaking down the index.js file.
  - 5.7. [No route parameter validation](#)
    - 5.7.1. Like above, it is a small, short term project. I was more focused on getting the basics implemented and fixing bugs.
  - 5.8. [InsertOne\(\) is used instead of create\(\) for Mongoose operations](#)
    - 5.8.1. According to the dev of Mongoose, they are the same. No changes.
  - 5.9. [Mongoose model name 'Task' doesn't match schema name 'item'](#)
    - 5.9.1. Duplicate.
  - 5.10. [Inconsistent use of async/await \(some routes missing await\)](#)
    - 5.10.1. async is only used when needed (e.g. A database action requiring an await). No changes.
  - 5.11. [No environment variables for configuration](#)
    - 5.11.1. Duplicate.
  - 5.12. [Missing proper HTTP status codes in responses](#)
    - 5.12.1. Like before, it is a small, short term project. I was more focused on getting the basics implemented and fixing bugs.
  - 5.13. [Static file serving could be configured more securely](#)

- 5.13.1. Like before, it is a small, short term project. I was more focused on getting the basics implemented and fixing bugs.
- 5.14. [Inline JavaScript in index.ejs should be moved to separate file](#)
  - 5.14.1. The only inline JS I have in my ejs file is through ejs, which as far as I am aware, cannot be moved.
- 5.15. [No CSRF protection](#)
  - 5.15.1. Duplicate in relation to security.
- 5.16. [Duplicated ID prefixes in template](#)
  - 5.16.1. Could not find any duplicated ids or id prefixes. No changes.
- 5.17. [Missing proper HTML structure \(no <html> tag\)](#)
  - 5.17.1. Implemented recommended solution.
- 5.18. [Inconsistent indentation in template code](#)
  - 5.18.1. Could not find inconsistent indentation. No changes.
- 5.19. [UML from Claude](#)
  - 5.19.1. Use Case Diagram, Fixed the system to bring the more current features into the diagram
  - 5.19.2. Sequence Diagram, Looking into the views of other sequence diagrams, there were uses of multiple line types, unchanged
  - 5.19.3. State Machine Diagram, Gave more consistent and clear conditions
  - 5.19.4. Class Diagram, There is a completed view of the system with the relationship multiplicity and visibility modifiers showing how it is finished, kept unchanged
  - 5.19.5. Sequence Diagram 2, Similar to the first diagram, the lines were not changed for the specifics of reading the information like a regular sequence diagram
  - 5.19.6. Activity Diagram, The sequence diagrams not validate the changes to ensure it lines up with the activity diagram
  - 5.19.7. State Machine 2, Changed the information to convey the changes to having multiple lists in the system
- 5.20. [Technology stack justification lacks depth](#)
  - 5.20.1. Duplicate.
- 5.21. [Architectural decisions section needs more detail](#)
  - 5.21.1. I believe the level of detail is sufficient for the scale of the project.
- 5.22. [Missing discussion of security considerations](#)
  - 5.22.1. Not of large concern. Detailed response earlier in 5.2.1.
- 5.23. [No mention of testing strategy](#)
  - 5.23.1. Not required for submission.
- 5.24. [Incomplete description of planned features](#)
  - 5.24.1. Not required section. Just meant to be a small description of what we would like to accomplish in the next week.
- 5.25. [Inconsistent heading numbering](#)
  - 5.25.1. Numbering seems to be consistent for the numbers in the original submission. Continued to use format for new submission.
- 5.26. [Missing sections for deployment and testing](#)

- 5.26.1. Testing is duplicate. Deployment not mandatory for submission.
- 5.27. No discussion of limitations or risks
  - 5.27.1. Not required for submission.
- 5.28. Missing timeline for planned features
  - 5.28.1. Second due date is only one week after first submission so no timeline needed. Also not required for submission.
- 5.29. No mention of coding standards or practices
  - 5.29.1. Not required for submission.
- 5.30. Inconsistent capitalization in headings
  - 5.30.1. Adjusted some capitalization
- 5.31. Run-on sentences in project description
  - 5.31.1. Implemented recommended changes.
- 5.32. Missing Oxford commas
  - 5.32.1. Oxford commas are optional.
- 5.33. Inconsistent use of technical terms (todo vs to-do)
  - 5.33.1. Every use of to-do is the same. No changes.
- 5.34. Formatting inconsistencies in bullet points
  - 5.34.1. Bullet points seem to be consistent in the original submission. Continued to use the same format for new submission.
- 6. Project Reflection
  - 6.1. Improvement of diagrams based on AI feedback?
    - 6.1.1.
  - 6.2. Improvement of code based on AI feedback?
    - 6.2.1. Made several changes to the code based on AI feedback but no major refactors or code style changes. 9/10.
  - 6.3. How accurate was the AI?
    - 6.3.1. I'd give it a 9/10 for the code, a 5/10 for the PDF and an X/10 for the UML.
  - 6.4. How much did you learn from the AI?
    - 6.4.1. For the code I learned quite a bit in relation to website/database security. As for the PDF, it had some valid critiques but a lot of the comments it made were not true.
  - 6.5. How much did you learn about the AI from this assignment?
    - 6.5.1. I learned a good bit more about how AI can be used. It can be helpful to review a project for things you may have missed but you still have to be critical of the AI as it is not always 100% correct.