CS482/495/496 Software Project Proposal: add your tentative project title here

your name(s) here

2025-10-27

1 Client Information

By sharing this client information and the rest of this document, you are stating that this client has provided this project as something they want (not something you created and asked if they wanted), and that they are interested in having you complete this project for your capstone.

- Client name: Dr. Isaacman
- Client title:
- Client email address:
- Client employer: Loyola University Maryland
- How you know the client:

2 Project Description

2.1 Overview

[Add a few paragraphs describing your project succinctly. What problem are you trying to solve, what is the purpose of your project? Why does your client want this project?]

2.2 Key Features

[At this point you should have a basic understanding of your client's needs. List out the key features of the software system the client wants you to build.]

2.3 Why this Project is Interesting

[Why did you decide this project was interesting enough to you to be a capstone project? What about this project is enticing? Why should anyone care?]

2.4 Areas of CS required

[What subfields of computer science seem most likely to be relevant to your project? A capstone must involve multiple.]

2.5 Potential Concerns and Questions

[Is there any aspect of this project that makes you unsure if it will work, either due to your own interests/background, or that you aren't sure if it fits the requirements? Are there questions you have about this project that you want instructor feedback about?]

2.6 Summary of Efforts to Find a Project

(Not necessary for 482) [Briefly list out when/how you've discussed with this client, and if you've discussed with other clients who either didn't work out or didn't respond. If you considered a different project and it didn't work out, why didn't it work out?]

[Most CS495 projects end here. The sections below are for CS482 and CS496 software projects].

2.7 Comparison to Draft

[For CS496 only, focus on highlighting the major differences between the draft proposal in CS495 and this one here. If there are no major differences, you can remove this subsection.]

3 Requirements

3.1 Non-Functional Requirements

[Non-functional requirements are just as important as functional requirements. Dont forget to specify them.]

ID	NFR Title	Category	Description
NFR1	NFR Example 1	Usability	Description of the NFR (it does not follow a user story template)
NFR2	NFR Example 2	Security	Description of the NFR (it does not follow a user story template)

Table 1: Non-Functional requirements

3.2 Functional Requirements (User Stories)

[In CS482, all functional requirements are written as User Stories. In CS496, some projects may use a different template to write the requirements. The table below is an example of writing the Stories. Adapt accordingly to different templates or if you want to display more info.]

ID	Story Title	Points	Description
S1	Story Example 1	5	As a user, I want to write a user story example, so that people will understand them.
S2	Story Example 2	2	As a user, I want to write a user story example, so that people will understand them.

Table 2: Functional requirements as User Stories.

4 System Design

4.1 Architecture

We are following the MVC (Model View Controller) architecture.

4.2 Diagrams

[CS482, on sprints/iterations 2-3, you need to create and update a diagram (check the assignment for which type of diagram). On CS496, since before sprint/iteration 1 you should have a class diagram and keep it up-to-date.]

4.3 Technology

For our tech stack, we are using JavaScript as our programming language and, by association, Jest for our testing framework. For our database, we'll use MySQL. For libraries, we'll use this list of frameworks/libraries (bcrypt *in case we add more*). For the front end of the application, we'll use insert front end here

4.4 Coding Standards

[Are your team going to follow any coding standards? For example, using a naming convention for Database tables (like only singular lowercase names). Another example, only allowing code with unit tests and above 60% coverage to be committed (good convention since testing is going to be evaluated). If you need inspiration to define your coding standards, the Extreme Programming approach has a set of coding, design, and test rules.] Our team will only use a naming convention for database tables consisting of lowercase and underscores. Our code will test with Jest and require above 90% coverage to be committed.

4.5 Data

[What is the main structure of your data? In SQL-like databases, this would be the planning of the main tables, their attributes, and interactions with other tables (basically an ER diagram). In NoSQL databases, this would be the main collections and general attributes of the JSON you will store in each collection.]

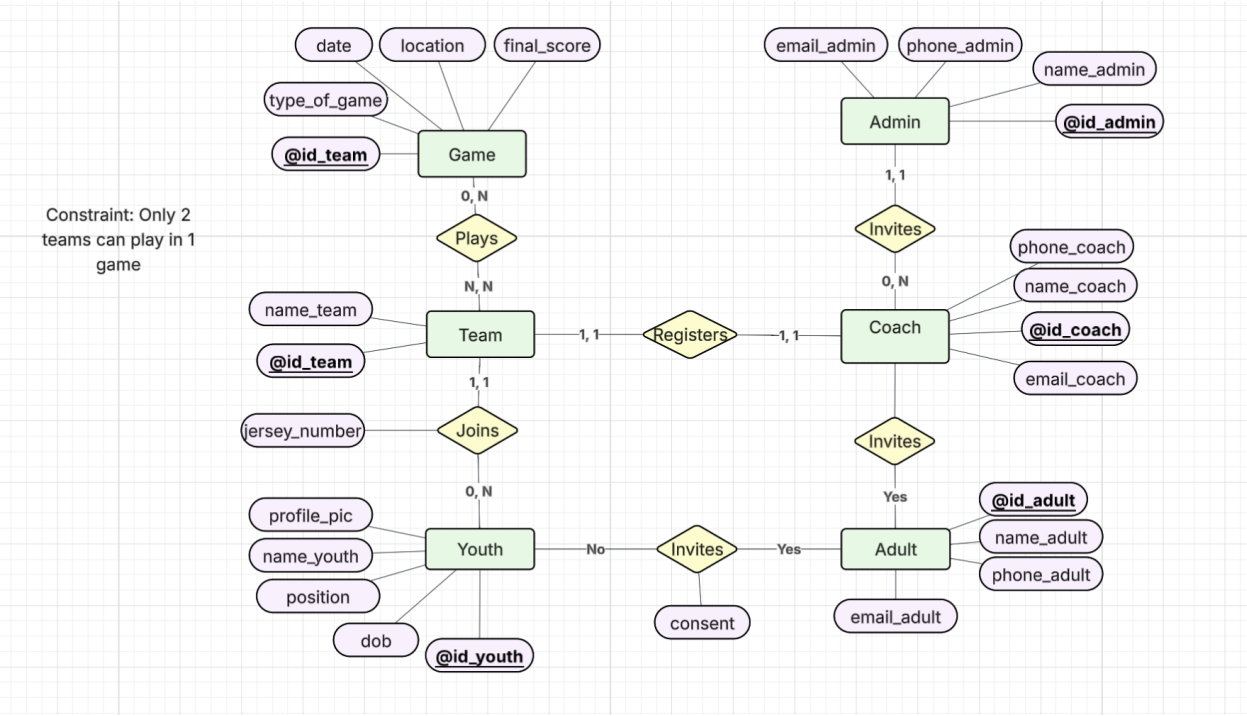


Figure 1: ERD for the data structure.

4.6 UI Mocks

[Define and draw/sketch/code the main UIs your user will interact with in your software. Add your UI mocks here and a short caption about it. Do not forget about the main forms and CRUD UIs.] will be done in canva

5 Iterations

5.1 Iteration Planning

[In CS496, you plan all iterations beforehand. In CS482, you update the planning here at each iteration.]

Iteration	Dates	Stories	Points
1	10/09 - 10/23	Coach Account, etc.	08
2	02/01 - 03/01	S3 Story Title, S4 Story Title, S5 Story Title, S6 Story Title	17
3	03/01 - 04/01	S7 Story Title, S8 Story Title, S9 Story Title, S10 Story Title, S11 Story Title	21
4	04/01 - 05/01	S12 Story Title, S13 Story Title, S14 Story Title, S15 Story Title	19
5	05/01 - 06/01	S16 Story Title, S17 Story Title	06
		Total:	70

Table 3: Iteration Planning for Incremental Deliveries

5.2 Iteration/Sprint 1

5.2.1 Planning

We planned initially 1/2 a CRUD for the coach account, however, Dr. Rocha graciously allowed for students to work over the weekend, which was enough time to complete the full CRUD. This is 8 points. The reason was to get started on one account type early to work out the kinks and figure out the errors early on. The major features include connecting with MongoDB and using UserDao.js. Additionally, it should create a coach account, view the account, update any aspect of the account, and delete the account.

5.2.2 Work Done

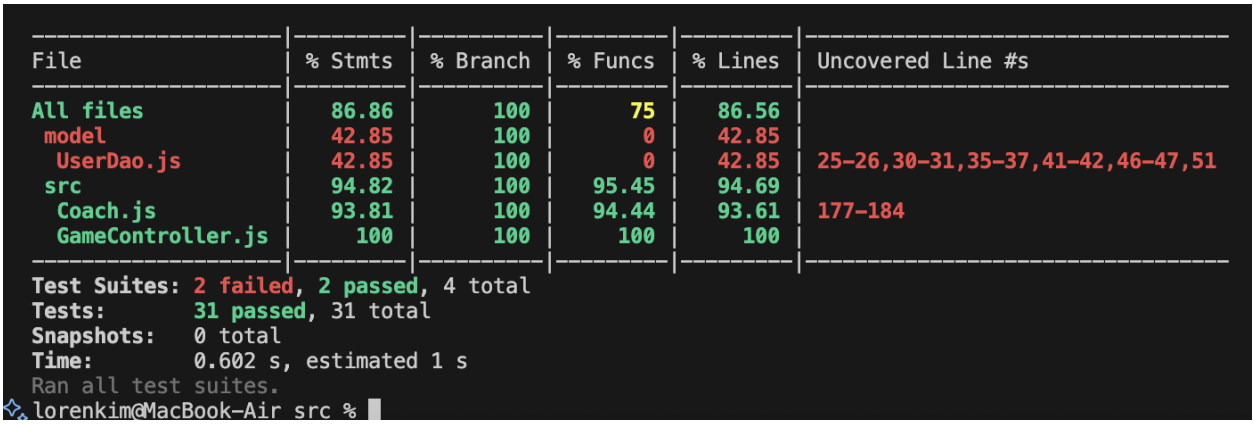
I (Loren) worked on story C5 (coach account). I did not partially complete any others but did take time to figure out DB connection.

5.2.3 Testing Coverage

I think the coverage for Coach.js is good enough since it covers 94 percent of lines and 100 percent of branches. I could increase the coverage by creating a Jest test for run() but ran out of time since it calls menu() and choice() so the code would be longer than the other tests.

5.2.4 Retroespective & Reflection

I should've started way earlier. With the midterm last week, I got distracted and didn't put enough time into the sprint. I had the most issued with mocking in Jest and had to take some time in other files to figure out how it works. I also had to do a similar process with connecting with MongoDB from user input in JS. I think the next sprint will go smoother since I was able to figure out these challenged, but I also found online textbooks and resources (mainly in JavaScript and Jest) that I can reference for the next sprint. Eventually, everything did go to plan, but it took a while to get used to MongoDB and mocking specifically. I learned to open a separate noncommitable file for my own learning and practice. I learned in this iteration how to connect user JS input with MongoDB, along with editing the database values from JS, and finally, mocking user input and console log in Jest.



File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	86.86	100	75	86.56	
model	42.85	100	0	42.85	
UserDao.js	42.85	100	0	42.85	25-26,30-31,35-37,41-42,46-47,51
src	94.82	100	95.45	94.69	
Coach.js	93.81	100	94.44	93.61	177-184
GameController.js	100	100	100	100	

Figure 2: Coverage for sprint 1.

5.3 Iteration/Sprint 2

5.3.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.3.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.3.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.3.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

5.4 Iteration/Sprint 3

5.4.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.4.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.4.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.4.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

5.5 Iteration/Sprint 4

[CS496 has 5 sprints. CS482 only has only 3 sprints (remove Iterations 4 and 5 from this doc if you are writing a doc for 482]

5.5.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.5.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.5.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.5.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

5.6 Iteration/Sprint 5

5.6.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.6.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.6.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.6.4 Retroespective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

6 Final Remarks

6.1 Overall Progress

[Have you completed everything? If so, present evidence on how you brought value to your client, and the overall client satisfaction. Otherwise, estimate how much progress you done and how long it would take to finish this project.]

6.2 Project Reflection

[Your personal reflection on the project. What lessons did you learned. What would you have done differently. How can you do better work in future projects? You may write this as a team or per person (or both)]

Appendix

[Appendix section if needed]