

# CS482/495/496 Software Project Proposal: Baltimore's Basketball Tournament

your name(s) here

2025-10-09

## 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:
- Client title:
- Client email address:
- Client employer:
- 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 will use an MVC architecture. As we understand them, the "view" is the actual interface of the website, the "model" is the backend or the data that the website runs on, and the "controller" is what connects them, handling input from the UI that needs to be put in the model and translating data from the model to a displayable format in the UI.

The main modules will be for:

- UI design (color, theme, page layout)
- Handling input from the UI (button, form, upload)
- Accessing and modifying database data (image upload/display, account info)

## 4.2 Diagrams

Not needed for Assignment 2

## 4.3 Technology

- TypeScript - Programming language
- React - Frontend Library
- Material UI - Frontend Library
- Node.js - Runtime Engine
- Express.js - Backend Framework
- MongoDB - NoSQL Database
- Jest - Testing Framework

## 4.4 Coding Standards

- Contact the customer (Dr. Ho) if there are any uncertainties with requirements
- We are using standard React naming conventions: PascalCase for components, classes, and interfaces; camelCase for variables and functions; names starting with "use" for custom hooks; and UPPER\_CASE for constants. All names should show a clear purpose (just name variables what they do).
- Commit small changes so we can go back if something breaks
- Only code with 60% branch coverage should be committed
- Add tests when bugs are found
- Pair program as much as possible (for learning and bug prevention)
- At least one other team member must accept a merge request
- We will update any documentation necessary as we change the codebase

## 4.5 Data

The application will utilize MongoDB, which aligns with the MERN stack. The core data structure is organized into several key sections. The users section stores all accounts, differentiating between roles (Admin, Manager, Guardian, Teen) and managing relationships. The tournaments section contains details for each competition, including rules, sponsor logos, and themes. Teams are managed in the teams section, linking them to their respective tournament, manager, and list of player members. Game schedules, scores, and live stream links are stored in the games section, while individual player performances are tracked per game in the playerStats section. Team-based communication is facilitated through the chat section.

The core information collected includes a unique username, email address, password, first and last name, and date of birth. Users will also have the option of uploading a profile picture. The system will assign each user a specific role: administrator, Manager, Guardian, or Teen. To establish critical relationships, the data model includes links between accounts; Guardian accounts will store a reference to the Teen accounts they represent, and Teen accounts will store a reference to their Guardian. Furthermore, user data will track team affiliations, with Managers storing a list of the teams they manage. This forms the key data that will drive our approach. How we form relationships and how we provide information to the users.

## 4.6 UI Mocks



Figure 1: Home Page

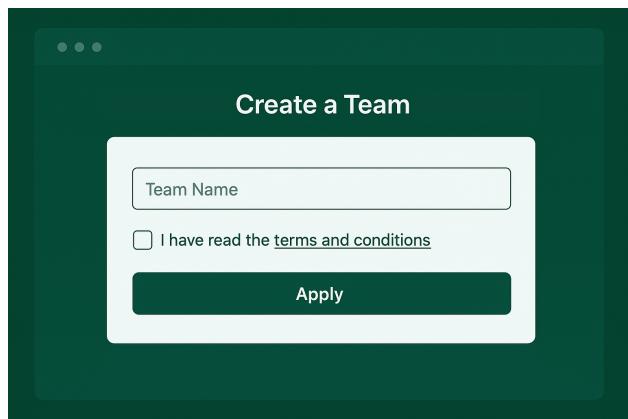


Figure 2: Team Creation

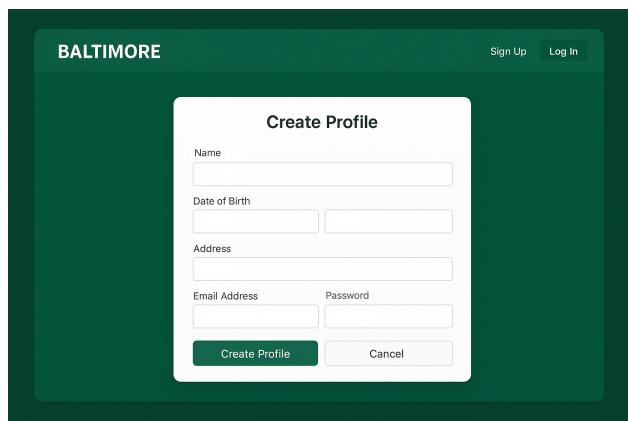


Figure 3: Profile Creation

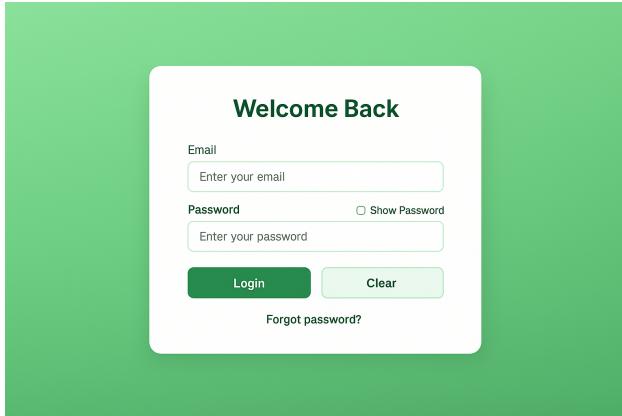


Figure 4: Login Page

## User Settings

John Doe  
john.doe@example.com

Full Name

Email Address

Password

**Save Changes**

Figure 5: User Settings



Figure 6: Team Schedule



Figure 7: Team Chat Room

## 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	01/01 - 02/01	S1 Story Example, S2 Story Example 2	07
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
<b>Total:</b>			<b>70</b>

Table 3: Iteration Planning for Incremental Deliveries

### 5.2 Iteration/Sprint 1

#### 5.2.1 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.]

#### 5.2.2 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.]

#### 5.2.3 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.]

#### **5.2.4 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).]

### **5.3 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]