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

Fabrizio Guzzo, Dylan Morales, Will Troisi, Zoe Willis

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: Prof. Ebert
- Client title: Professor
- Client email address: eebert@loyola.edu
- Client employer: University Of Loyola Maryland
- How you know the client: Professor at Loyola

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

~~[Which type of software architecture are you team following? Layered architecture, MVC, other? What are the main modules for your software?]~~

Our team has chosen to utilize the MVC structured architecture to allow for obvious focus point on our development. Using MVC we can split up our work into 3 main categories: Model, View, and Controller. Using this style of structure will promote scalability, mendability, and maintenance as we move further down the road. It will allow for developers to handle different parts of the project, avoiding the conflict of waiting on another group member to finish before starting your work. When it comes to tools we will use for each part of MVC that will be explained in our Technology section. The module will be the main branch that handles our data that will contain important information for the game consistently and securely. The controller is the link between the model and the view. It will take in requests data from the user and retrieve data from the model to display on the view. The view is the display of the structure and it allows for the user to see the information our software is displaying. Many benefits will be created from using this structure that will allow for efficiency, debugging, organization, modular development, team collaboration, and maintainable software.

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 the front-end, we will use React to handle with the user interface. In terms of our back-end we will use JavaScript to handle the interaction of the front-end with data. For API usage, we are going to utilize NodeJS to gather the necessary information to display on our website.

4.4 Coding Standards

-We are going to limit the use of global variables. -Header Format of each different Moduel: Name Date Author Synopse Variables -We are going to utilize CammelCase for Variables and Functions -Final Code Shouldn't be pushed unless tested beforehand or explicitly given noticed of an issue -Comments must be made in order to help entire team understand code

4.5 Data

Logical Model of the database entities and their relationships:

Adult (id_Adult; Name; Dob; Email; Phone; Permissions;);
Player/Child (@id_player; #id_Adult; #id_Team; Dob;);
Coach (@id_Coach, #id_Adult; #id_Team;);
Team (@id_Team; #id_Coach; #id_Manager; Playerlist;
);
Guest (@id_Guest; Permissions;);
Admin (@id_Admin; #id_Adult;);
Manager (@id_Manager; #id_Team; #id_Adult;);
Parent (@id_Parent; #id_Adult;);

4.6 UI Mocks

UI Mocks.pdf

UI Mocks.pdf

5 Iterations

5.1 Iteration Planning

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

5.2 Iteration/Sprint 1

5.2.1 Planning

Will: P0 Establishing Database 1pt. U1 Rate my manager 1pt. Manager invites coach 1pt. Manager Invites player 1pt. Zoe: U11 Landing page 2pts. G0 Guest Page2pts Fabri: U3 Login/Guest 1pt, U2 Create Scoreboard 1pt, U3 standings 1 pt, u6Color scheme 1pt Dylan: u10 Create team 1pt, ADM5 Admin Page 2 pts, ADM1 Season Lockings 1 pt

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
		Total:	70

Table 3: Iteration Planning for Incremental Deliveries

Altered Stories Will: Dylan: Create a Team, View Team (4 pts) Fabri: Created a game, Update Game(Game Status and Score), Save Game(CRUD =4pts). Explored the use of react, while not implemented in terms of the Game method, We still have a basis template to then implement with our html pages. ~~[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.]~~

The stories we completed on this iteration were

Dylan - create a Team and view Teams

Will-

Zoe-

Fabri- create Game Functionality

Partially completed stories

Dylan - Using React

Will -

Zoe -

Fabri - React front end Implementation

In detail

Dylan- Created a website that creates a team using the field's name, manager, and logo. The website then uses the controller and DAO for team to store the information in the database. The website can also display all the teams that are in the database by using the controller and dao.

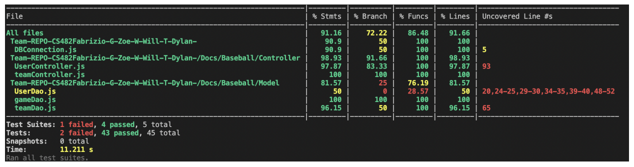
Will -

Zoe -

Fabri - Implemented full game functionality including creating new games, updating existing game information(status and score), and saving completed games covering all core CRUD operations(4 pts). Additionally, we explored integrating React for potential frontend expansion. While React was not directly implemented for the Game module at this stage, we successfully established a foundational template that can be connected to our existing HTML pages in future iterations. This groundwork ensures an easier transition to a more dynamic, component-based user interface later on.

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.]~~



```
----------------------------------------------------------------------|---------|----------|---------|---------|-----------------------------------
File                                                                  | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------------------------------------------------------------------|---------|----------|---------|---------|-----------------------------------
All files                                                             |   91.16 |    72.22 |   86.48 |   91.66 |
 Team-REPO-CS482Fabrizio-G-Zoe-W-Will-T-Dylan-                        |    90.9 |       50 |     100 |     100 |
  DBConnection.js                                                     |    90.9 |       50 |     100 |     100 | 5
 Team-REPO-CS482Fabrizio-G-Zoe-W-Will-T-Dylan-/Docs/Baseball/Controller |   98.93 |    91.66 |     100 |   98.93 |
  UserController.js                                                   |   97.87 |    83.33 |     100 |   97.87 | 93
  teamController.js                                                   |     100 |      100 |     100 |     100 |
 Team-REPO-CS482Fabrizio-G-Zoe-W-Will-T-Dylan-/Docs/Baseball/Model    |   81.57 |       25 |   76.19 |   81.57 |
  UserDao.js                                                          |      50 |        0 |   28.57 |      50 | 20,24-25,29-30,34-35,39-40,48-52
  gameDao.js                                                          |     100 |      100 |     100 |     100 |
  teamDao.js                                                          |   96.15 |       50 |     100 |   96.15 | 65
----------------------------------------------------------------------|---------|----------|---------|---------|-----------------------------------
Test Suites: 1 failed, 4 passed, 5 total
Tests:       2 failed, 43 passed, 45 total
Snapshots:   0 total
Time:        11.211 s
Ran all test suites.
```

Figure 1: Coverage report

The overall coverage is pretty good at around 91, which means most of the code is being tested. The controllers and DAOs have high coverage, but UserDao.js is still low since some of its database and error-handling parts aren't tested yet. The branch coverage is also lower because we didn't include admin or error checks in this version. To improve, we'd add more tests for different CRUD cases and edge scenarios. For now, this level of coverage is solid and shows that most of the main functionality works as expected.

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]