CS482/495/496 Software Project Proposal: Youth Football League Portal

Oscar Roat, Brenden Bokino, Loren Kim, Reece Watkins

2025-11-10

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. Sibren Isaacman
- Client title: League
- Client email address: snisaacman@loyola.edu
- Client employer: Loyola University Maryland
- How you know the client: CS Professor

2 Project Description

2.1 Overview

Our project is a web application for a youth football league called "Loyal Order of Youth League Athletes". Dr. Isaacman wants an all-in-one site to advertise manage, and broadcast the league. Here, games are scheduled and live streamed, teams are managed, and stats are tracked.

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.]

- **User Roles:**
  - Admin: manage calendar, games, playoff brackets, carousel, stats, live broadcasts, verify parental approval.
  - Coach: invite adults to create youth profiles, manage their own team.
  - Adult: oversee and update their child's youth profile, communicate with youth.
  - Youth: create a personal profile including photo (with privacy restrictions).
  - Guest: comment during live games, rate players/coaches, limited access to youth pages.
- **Calendar:** schedule and display matches with date, time, type (regular, playoff, finals), and participating teams.
- **Team Management:** create, update, and delete teams; include team name, colors, and roster.

- **Game Management:** add, edit, delete games; update and track game statistics; broadcast live games with live stats.
- **Playoff Bracket:** auto-updating bracket to reflect wins/losses during the season.
- **Profiles:**
  - Youth: profile picture and details, linked to adult accounts.
  - Adult: contact info required, can manage youth profiles.
  - Coaches/Users: manage own accounts, invite/manage youth.
- **Communication:** adults can send messages to youth; users can comment on live games and review teams/coaches.
- **Frontpage Features:** carousel of seasonal images to enhance visual appeal.
- **Privacy and Access Control:**
  - Youth cannot chat with other youth.
  - Youth profile pictures visible only to logged-in users.
  - Coaches/adults have additional permissions for managing teams and profiles.

2.3 Areas of CS required

Software Engineering/Testing, Web Development, Data Management, (maybe some) Sports Analytics

2.4 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?]

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 be using an MVC architecture for this project. Since there are four of us working together, the separated design will be the easiest to integrate. For the *Model*, we will need a way to access data for individual users (admin, coaches, adults, youths), teams, and games (to add to the calendar). The *View* will need landing pages for the home screen, team, calendar/games, profile, and livestreams, including comments and game stats.

The main modules involved in this project have to do with common functionalities on the website. Adding games to the calendar is a primary admin function, and there is a lot of info that goes into each game, such as the date, time, location, and teams playing. All of these should be added to the calendar together as a new "game."

Messaging is another key feature, but there are different types of messages and restrictions based on user types. There is messaging between two adults, which should be private for those individuals. Adults can also message youth to inform them of games, but youth cannot message back/each other. Finally, there are live comments during games. Those should be public to everyone watching the live stream.

The score board during live streams is another important piece. Having an efficient way to update the UI during games is important so the score bug is up-to-date. Things that need to be updated on it are what quarter it is, the clock, the down and distance, and each teams score.

Other key features that could be built into modules are getting data from the database(s), user authentication, the front page picture carousel, and profile creation/updatinghjjh

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

We will be using **JavaScript** for building the application and **SQL** for data storage. The frameworks we will use follow from conventional sports websites, such as **IMLeagues**. For the front-end, we will use **React** for UI, especially for the updating scoreboards during broadcasts. For the backend, we will use **Express** for the business logic and **MongoDB** for the database. For testing, we will use **Jest**.

4.4 Coding Standards

Tests will be required for every new commit. Obviously, we want as high coverage as possible, but the baseline for every commit should be **70%**. In person meetings will happen at least once per sprint, preferably earlier so everyone will be on the same page going into the sprint.

4.5 Data

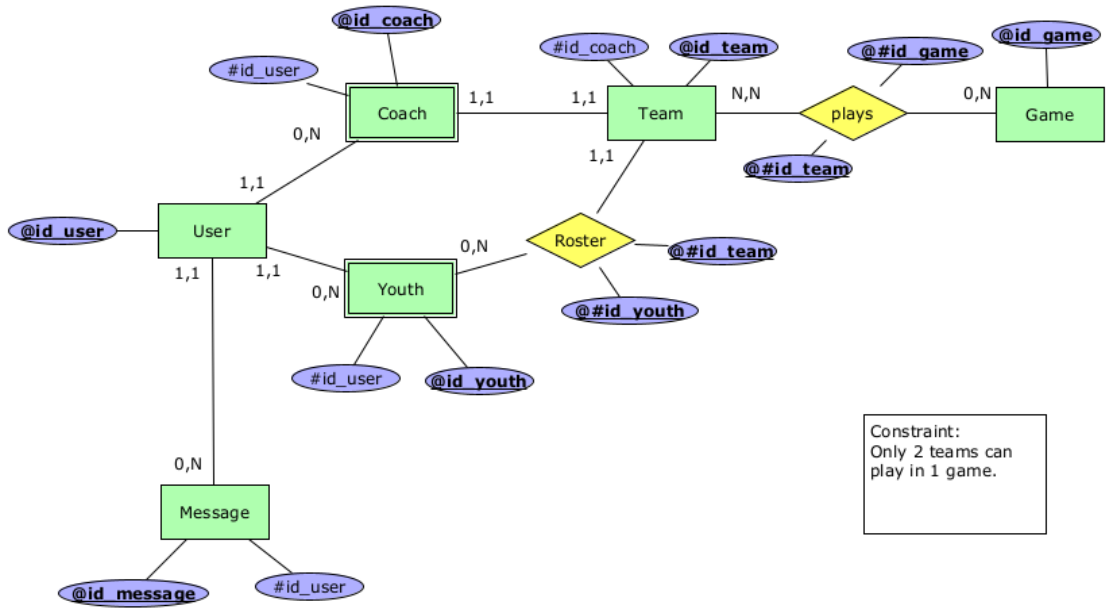
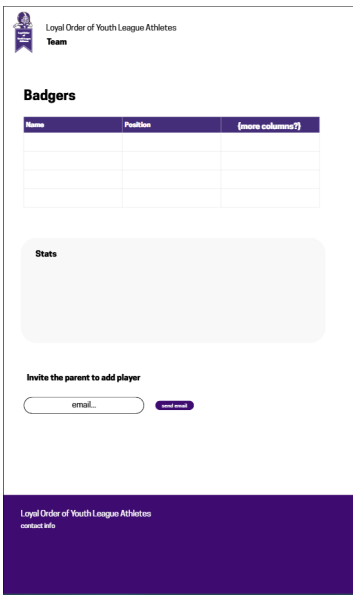


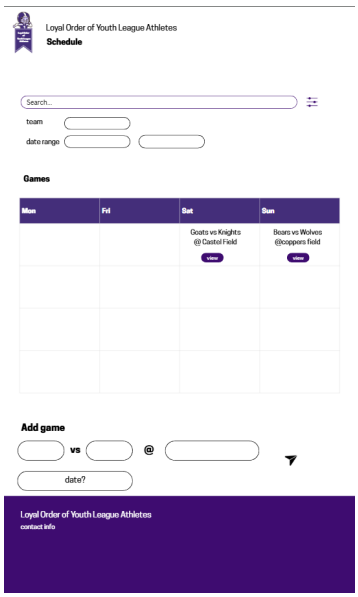
Figure 1: ERD for the data structure.

4.6 UI Mocks

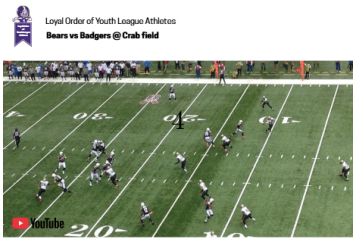
(a) The home page. Information about upcoming games and some ratings.

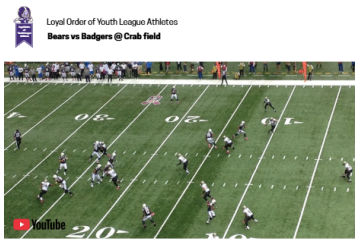


(b) Team page. View team stats and roster, coaches invite from here.



(c) Games page. View games, admins add games. Calendar and bracket views.





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
		Total:	70

Table 3: Iteration Planning for Incremental Deliveries

5.2 Iteration/Sprint 1

5.2.1 Planning

- Loren and Brendan:
  1. (C5) User Account: 8 points
- Oscar:
  1. (Spike) Images to Database: 2 points;
  2. (U15) Login/Logout: 1 point
- Reece:
  1. (C8) Youth Team - Create Team: 1 point
  2. (A0) Calendar - Create Game: 1 point
  3. (Spike) Calendar and event handling: 2 points

Total: 15 points

5.2.2 Work Done

Loren and Brenden:

Oscar: The login/logout functionality was not totally finished. The actions of logging in and logging out worked, but saving the info in a session to remain logged in while browsing the site was not completed. The spike was completed, though.

Reece:

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

Oscar: I definitely underestimated the amount of work/difficulty of work for this project. For this sprint, each task that I had to work on weren't too hard on their own, but having to learn and adjust while writing the code was still difficult. In the coming weeks, when the work will get harder, I need to manage my time better.

5.3 Iteration/Sprint 2

5.3.1 Planning

- Loren:
  1. (U2) Communications: 8 points
  2. (U5) User Account - Finish UI: 1 point
- Brenden:
  1. (C7) Invite Youth: 3 points
  2. (A10) Youth Profile - Adults can view youth profiles: 1 point
  3. Technical Task Webpage UI Design: 1 point
- Oscar:
  1. (U1) Welcome Images: 3 points
  2. (U15) Login/Logout - Finish password encryption and user sessions: 1 point
  3. (A6) Playoff Bracket -¿ UI and interactivity: 1 point
  4. Refactoring: 1 point
- Reece:
  1. (U4) Live Broadcast: 3 points
  2. (C8) Youth Team - Finish UI and tests 1 point
  3. (A0) Calendar - Create game: 1 point,
  4. (A0) Calendar - Create calendar: 2 points
  5. (A0) Calendar - Add game to calendar: 1 point

5.3.2 Work Done

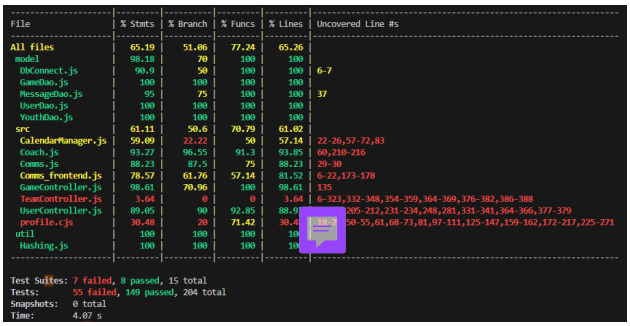
Loren

I partially completed communications up to 8 points. I did the full crud, including UI and testing, and need to finish the reply option for other users to reply to messages. I completed the UI for user account. I also completed the UserDao tests (which I counted as less than a point, but wanted to keep track that it was completed)

Brenden

I completed my technical task and part of A10 Youth Profile. I enabled adults to view youth profiles and ended up spending a lot of time creating youth accounts and profiles themselves. Profiles have dynamic links a DAO was created for a youth account. There isn't much left to do for youth profiles besides individualized information such as a youth players game statistics, so I'd bump the points up to 2 for this iteration. I've also partially completed C7 Invite Youth. The system I ended up going with is one where the coach promotes an adult, the adult creates the account for the youth, and the coach can see all youth not in a team and add them to theirs. It all works except that the button to add doesn't add them currently, the models for teams/coaches need a bit more work before it is ready for that. What is left to do isn't difficult, so I would give myself 2 points for this task, totaling to 4 points.

Oscar



File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	65.19	51.06	77.24	65.26	
model	98.18	70	100	100	
DbConnect.js	90.9	50	100	100	6-7
GameDao.js	100	100	100	100	
MessageDao.js	95	75	100	100	37
UserDao.js	100	100	100	100	
YouthDao.js	100	100	100	100	
src	61.11	50.6	70.79	61.02	
CalendarManager.js	59.09	22.22	50	57.14	22-26,57-72,83
Coach.js	93.27	96.55	91.3	93.85	60,210-216
Comms.js	88.23	87.5	75	88.23	29-30
Comms_frontend.js	78.57	61.76	57.14	81.52	6-22,173-178
GameController.js	98.61	70.96	100	98.61	135
TeamController.js	3.64	0	0	3.64	6-323,332-348,354-359,364-369,376-382,386-388
UserController.js	89.05	90	92.85	88.9	205-212,231-234,248,281,331-341,364-366,377-379
profile.cjs	30.48	20	71.42	30.4	50-55,61,68-73,81,97-111,125-147,159-162,172-217,225-271
util	100	100	100	10	
Hashing.js	100	100	100	10	

Test Suites: 7 failed, 8 passed, 15 total
Tests: 55 failed, 149 passed, 204 total
Snapshots: 0 total
Time: 4.07 s

Figure 3: Sprint 2 Testing Coverage

I completed the Login/Logout functionality that I had leftover from iteration 1, that being password encryption and saving user info in sessions. On the profile page, there is a script that shows that /loggeduser continues to return the user, even after reloading. It doesn't actually show user information yet, but that is more of a semantic issue. I have partially completed the Welcome Images section. The website does link to the photos stored in the db, and the images show up in the image carousel. However, I still need to add admin control over what images are seen in the carousel. Finally for the playoff bracket, I set the UI and added some interactvity. I still need to link the teams in the playoff bracket to the teams in the league, I need to allow only admins to update the bracket, and I need the info to be stored through reloads.

5.3.3 Testing Coverage

Loren
I think this coverage for model and util is good enough. I do think that half of the files in src are good enough, while the other half will need to increase in coverage. I worked on the coverage for MessageDao, UserDao, Coach, Comms. I can increase the coverage in Coach and Comms by writing tests for the remaining 6-7 lines uncovered. Comms-frontend.js is only at 81% because it was more difficult writing the tests that had the html component, document, etc. So I only wrote for some of the functions that were repetitive (ie. Successful delete, delete error code, etc)
Brenden
Coverage on my end was okay. I worked a lot on front-end Javascript and had trouble figuring out the testing for it. I ended up stepping back from those tests and improving coverage of some of the weaker scripts. Coverage for other things like UserController and YouthDao turned out well. I'll hopefully be able to tackle the testing issues this next iteration.
Oscar
The password encryption tests are successful, but there is an issue with the login/logout tests. Even though logging in does work on the website, there is an issue with the test setup that I could not fix after hours of debugging. Furthermore, for the tests that require password hashing, the tests have an issue with calling the password hashing function, even though it is correctly imported in the test file.

5.3.4 Retrospective & Reflection

Loren
The challenges were adding delete and update especially as buttons in the UI. The hardest part was trying to convert the script from the html code into js. I didn't expect it to be difficult, but it was very tricky getting it to work with the posts and document. I did write the test cases so I can try to go back and make the js frontend work with the html. I can avoid this mistake by planning better in the future. I learned how to add buttons in the messages based on ID as well as updating a message both in the database and in the frontend. I started writing everything in the Comms.js and quickly realized that a lot of the logic would also be in the html with the delete/update buttons as well as the view/hide toggling. I learned how to write the tests for posts in Jest as well as start working with error messages. I learned a lot in this iteration and will learn from my mistakes and successes to code better in the next iteration.

Brenden
This iteration was pretty fun. I enjoyed putting together the UI for the profile page and creating the navbar using Bootstrap. What really frustrated me was the testing. I'll be sure to test along the way next time because handling the tests afterward is a huge headache. Another issue I was having was thinking about how I'd handle inviting youth players. I think going to the client for a clearer idea would have been good.
Oscar
This iteration went much better than the previous. That is mostly because of the time I spent working on the project. I spent a long time trying to add admin control to the image carousel, but I wasn't able to figure it out. That will be a primary focus for the next iteration. I am still having trouble with my test writing, both in terms of coverage and effectiveness. I need to write more tests immediately after adding new functionality, so I don't fall behind in the future.

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]