**Tournament Management Software for UA Badminton Club**

*Junfeng Xu, Enfa George, Rupal Jain, Urvika Gola*

# **Table of Contents**

[**Table of Contents 2**](#_4d53py4w4xa6)

[**1. Introduction 3**](#_b2d83gxefejn)

[Highlights 3](#_ry7m0lrww8qb)

[Changes 4](#_v3giom65wcjo)

[**2. Customer Need 5**](#_yu83ypnpdjyb)

[Customer’s desired overall experience. 5](#_dj0illm1mfep)

[Outcome from our sprint review meeting with the customer: 5](#_drwl5y98nicq)

[**3. Project Goals 6**](#_84w65quwucb)

[Customer Problem Update 6](#_z1q4l4l7co5)

[Benefit Provided 6](#_6ijbadrshnzc)

[**4. System Description 7**](#_5w7m1f389heh)

[System Challenges: 8](#_g6a77on2ipy8)

[Interactions within and external to the System: 9](#_n1duo3604pqx)

[Module Overviews 9](#_ak8ufxzbd5az)

[Database structure (ER diagram): 10](#_dwya4v37euhq)

[**5. Current Status 13**](#_wdo4upste6ps)

[Goals for this Iteration: 13](#_6jj0ry53znue)

[Screenshots of What’s Working: 13](#_xmd23kwqko2b)

[Challenges We are facing: 18](#_9v4mvf3gwjcs)

[Block diagram for the current state of the system: 19](#_u7o1g6aud4ib)

[System overview block diagram: 19](#_vxcn1dwkwm4b)

[Modular block diagram: 20](#_z4sy2loxmi68)

[Registration module: 20](#_j6ymqkwfwue6)

[User Management Module: 21](#_gl6uu3eth8ek)

[Seed & Draw Module: 22](#_5zw3ekw4co0r)

[Match Management Module: 22](#_xs1qbcxoihl1)

[Test Performed: 23](#_kp012xu9iyzz)

[Total number of lines of code we collectively written: 23](#_kiib72jatr5d)

[**6. Project Management 24**](#_q2e2mwni1uv8)

[Change Log 24](#_ersopnc7k1sk)

[Goals for the next iteration: 24](#_41mgx6qvepk3)

[Plan for the rest of the semester 25](#_vm0zsdymkqgb)

[Project Management Tools 25](#_dgnvn5w9d62r)

[Jira to manage the sprint. 25](#_ydeh9vfqwx5u)

[Confluence to manage daily updates. 26](#_ilo96vcrfvl4)

[Github for version control and managing releases 27](#_9ow2r69mn48f)

[**7. Team 28**](#_bxibr6eltyrk)

[Team roles for this iteration? 28](#_ybwo2sds82zh)

[Team Member Contribution: 28](#_w23139a9a219)

[Rough estimates of the each team member contributions to the project (in percentage) 29](#_fmq0uqh9ii2d)

[**8. Reflection 30**](#_ooh3obj0nfxf)

[What went well? 30](#_p71mvzxdv6s2)

[What didn’t go well? 30](#_c68nc1rdnx69)

[Goals that were not met and their issues 31](#_nahjy2nlg8mc)

[Plans to overcome the issues 31](#_uggzhsahoek6)

[Plans to do differently in the next iteration 31](#_izbze8sogjgv)

# **1. Introduction**

## Highlights

The plan that our team made to start this iteration is to complete the following items collectively as a team:

* Acceleration Phase:
  + Each team member will set up the project development environment and test run a dummy program.
* Drive Phase:
  + Initialization of the project on the project management tools and utilization of the tools throughout the iteration
* Design Phase:
  + Design and creation for the database.
  + Design for the front-end user interfaces and back-end controller for our project.
* Implementation Phase:
  + Implementation for the user sign-in and sign-up function.
  + Implementation for the player dashboard and admin dashboard front-end user interfaces.
* Sprint Review:
  + Meet with the customer at the end of this iteration and collect feedback.

Team accomplishment:

* Each team member has set up the project environment on their machine and successfully ran the test program.
* Project has been set up on all the project management tools including Jira, atlassian, and Github. We have been using it for the whole iteration.
* Database design has completed. The database creation is not finished and has been added back to the backlog and will be worked on next iteration.
* Design for the front-end user interfaces and back-end controller for our project are completed.
* The user interface and back-end controller portion of the implementation for the user sign-in and sign-up functions are done but the connection to the database portion is not done. That has been added back to the backlog and will be worked on next iteration..
* The implementation for the player dashboard and admin dashboard front-end user interfaces are completed but could use some polish. We have added that as a task for the next iteration.
* We met with the customer at the end of the iteration and collected feedback. The feedback items have been added to the backlog.

## Changes

We have made a few changes since the proposal, including a front-end user interface design change after team review, the customer desired overall experience change since our first meeting with them, and database design changes.

Here is a table for the date, motivation, description, and implications of each change:

| Date | Description | Motivation | Implication |
| --- | --- | --- | --- |
| 03/02/2023 | UI Design Change | To make the admin dashboard more user-friendly and easier to update. | Updated UI design to have an admin dashboard in a single screen instead of multiple screens. This will make it easy for the Admin to make updates. |
| 03/07/2023 | Database redesign | To fix errors with primitive data types in the initial design. | Reworked structure of database and created a new ER diagram. |
| 03/12/2023 | Customer desired overall experience change | To satisfy the customer’s updated requirements that we have learned from the customer’s feedback. | Given our current design for the project, it is still able to address customer’s new surfaced problems. |

# **2. Customer Need**

## Customer’s desired overall experience.

The customer’s desired overall experience was to have new tournament management software with intuitive interfaces and low or no cost to use. We learned that from our initial meeting with the customer. However, in the latest meeting with the customer, their desired overall experience has changed. They used to run software that they could use to organize tournaments. That software is really hard to use and has a high annual license fee. They came up with the original desired overall experience during our initial meeting with them. Now they have lost access to that software for an undisclosed reason. This year, they are organizing the tournament manually. Includes manually creating the events tables from the player registration list, manually making draw brackets, manually managing matches and progressing event brackets. That is a massive amount of manual effort. As a result, their desired overall experience is changed to have a tournament management software that can help them eliminate all the manual efforts that mentioned above and automate everything. The original desired overall experience from the customer is still valid and will be taken into consideration for our product design but the emphasis can be changed to satisfy their latest desired overall experience.

## Outcome from our sprint review meeting with the customer:

We met with our customer representative, the UA badminton Club president, Julie Fan. We reviewed our high/low priority stories. She agrees with our prioritization for the stories. We gave her an update on our user interface design and received some feedback listed below. We also shared our overall progress of the project and she understands the time constraint we had and is happy with our progress.

Feedback received on user interface design:

1. When updating matches on the admin dashboard, it would be nice to sync the match information to the bracket automatically.
2. For the Events page and the Draws page on the public view website, it would be nice that all the events can be displayed as a collapsible list.
3. It would be nice to have the flexibility to change among the three U19/U17/U15 for the U events.
4. They could use the round robin tournament format.

The customer agreed that we should add the feedback items as low priority tasks in our product backlog.

# **3. Project Goals**

## Customer Problem Update

The original problem we would be addressing was their current tournament management software is very complex and hard to use and has a high cost of annual license. In the sprint review meeting with the customer, we learned that the customer's current experience on hosting a tournament has changed. Now they have lost access to their current software so that they have to organize and run their tournaments all manually. Their problem has shifted to needing a tournament management software that can automate their manual effort. Our current design of the project will still address their new problem and satisfy their requirement.

## Benefit Provided

The project will provide tournament organizers (our customer) with the ability to automate the tournament event tables creation, to automate the tournament event drawing making process, and to automate the match management on the tournament date. Those benefits will directly solve customer’s current problems.

# **4. System Description**

The system is designed using the MVC architecture i.e Model - View - Controller architecture.

**View:** The user interface and presentation are handled in the view. The technology chosen is html/css. The view also handles form validations.

**Controller:** Handles all major logic of the application. The technology chosen is flask framework in python

**Model :** Handles the various objects in the system and their storage and retrieval from the database. The technology chosen is the libraries associated with the Flask framework in python.

**Database :** The only external element of the system, it stores all relevant data needed for the application to perform well. The database interacts with the Model only.

User

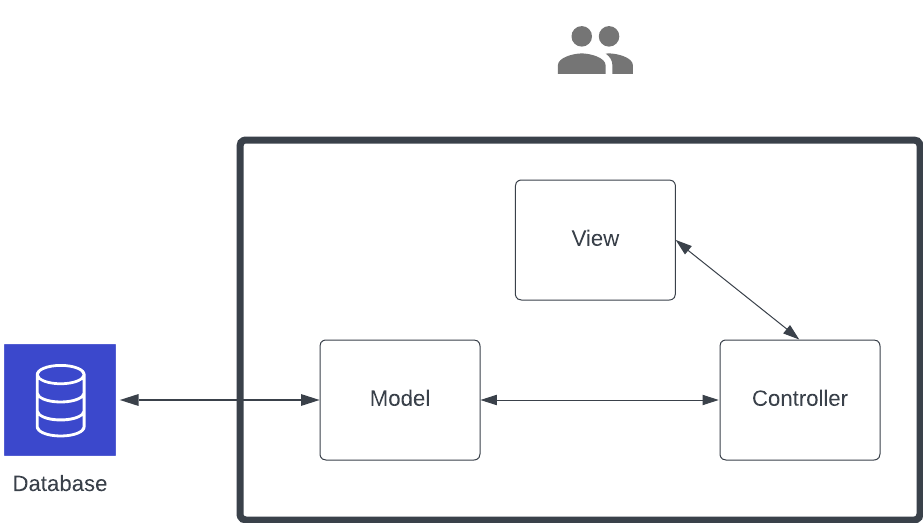


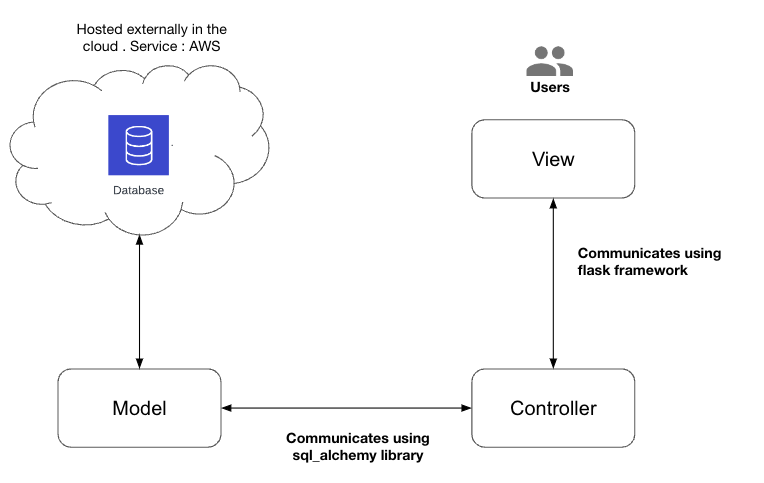
Figure: Context diagram for the application. The thick black border marks the boundary of the system. The system interacts externally only with the database.

## System Challenges:

1. **Complexity of the model and controller:**  
     
   The system has to handle the complex logic of creating matches within the rules and requirements of badminton, show in live the match results, automatically move players and create draws in the ABCD format. There are many rules and requirements to handle. Managing such complexity might pose challenges to data consistency and scalability if care is not taken in every step of the way.
2. **Database Management and Security:**  
     
   The database collects Personal Identifiable Information from players, hence contains sensitive info. The database as well as the system need to be secure and comply with local and federal regulations.
3. **Interaction with the cloud service:**  
     
   Since the database is hosted in AWS, this external interaction might cause latency. System must be designed in such a way that there are minimum db calls, otherwise the system would be too slow for the user and ruin the desired experience of the users.
4. **Integration, Testing and Maintenance:**  
     
   Since the view, controller and model are separate components and built separately, routine integration testing needs to be done frequently to avoid big roadblocks round the road. The system must be tested and maintained frequently since its internal logic determines how the badminton match progresses and who plays when. Bugs in the system stemming from incorrect logic implementation can cost the users the game.

## 

## Interactions within and external to the System:



## 

## Module Overviews

The system is aimed at streamlining the tournament process, from registration to the final match results, and everything in between. The following is an overview and detailed description of the system components and features. Modular block diagrams can be found in Section 5.

1. **Registration:** The system provides an online platform for players to register for tournaments. Players can enter their personal information, such as name, age, gender, and contact details. The system also allows players to register as a team or as individuals, and it tracks player information and updates it as necessary.
2. **User Management:** The system includes a user management module that allows tournament organizers to create and manage user accounts for players, including assigning points (seeding).
3. **Seed & Draw:** The system generates tournament schedules based on the number of players and teams, and it assigns players to matches based on their proficiency levels. The system also allows tournament organizers to manually adjust the schedule if necessary. The system can be configured to schedule matches at specific times and dates.
4. **Match Management:** The system gives the admin the ability to update matches live. It tracks the progress of each match and displays the results on a live scoreboard. Players and spectators can access the scoreboard from their mobile devices or computers to see the current score, game and set counts, and other relevant information

## Database structure (ER diagram):

The ER diagram has 8 major entities: User, Player, Permission, Login, Match, Result, Tournament, and Event. These entities are connected by relationships that describe how they are related to each other.

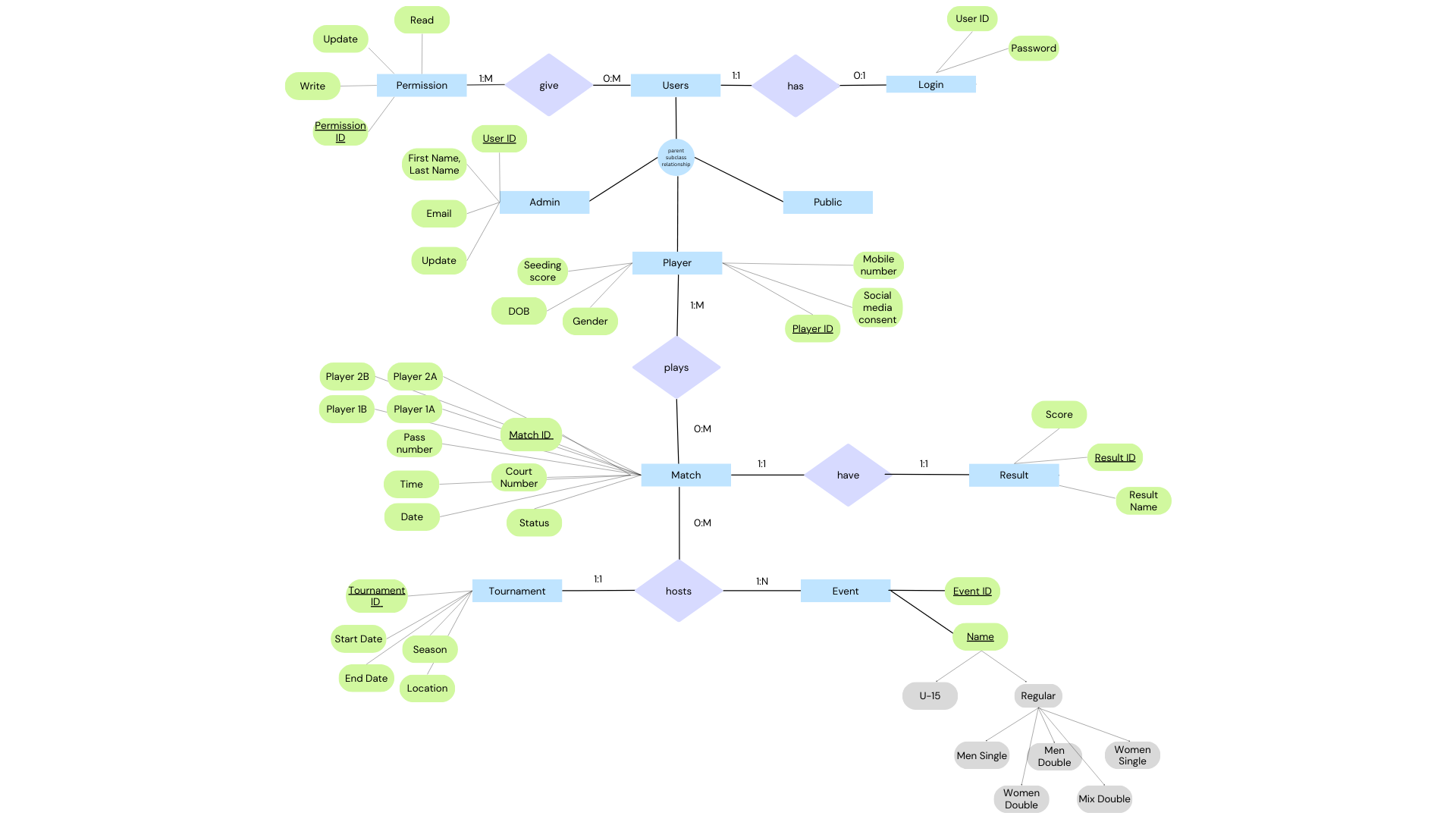
**User**

There are three types of users: admin, player, and public users. Each user type has different privileges and permissions.

**The Admin** user has full access to the system and can perform tasks such as creating tournaments, scheduling matches, assigning courts, and managing player and court information.

**The Player** user can participate in the tournaments, view their match schedules and scores, and update their personal information. The Player entity represents the badminton players who participate in the tournament.

**The Public** user can view information about the tournaments such as the schedule, results, and rankings, but cannot perform any actions on the system. The public users also do not have any login information, they can view information without signing up and signing in.



**Login-user relationship**:

Each login record is associated with one player and one admin. Public users do not have any login information stored.

**The Match entity:**

It represents the matches that are played during the tournament. Each match has a unique Match ID, and is associated with one or two players, and attributes such as a date and time, a court where the match will take place, and a score.The court represents the courts where the matches are played. The match status signifies whether the match is upcoming, in progress or completed.

**The Tournament entity:**

It represents the overall tournament that the matches belong to. Each tournament has a unique Tournament ID, that is associated with one match. Additional attributes for the Tournament entity may include the tournament name, start date, end date, season, and location.

**The Event entity:**

It represents specific events that take place within a tournament.

**The Result entity:**

It represents the outcome corresponding to a match. Each Result record would be associated with a specific Match within a Tournament, and would have attributes that describe the winner side, and final score of the match.

# 

# **5. Current Status**

## Goals for this Iteration:

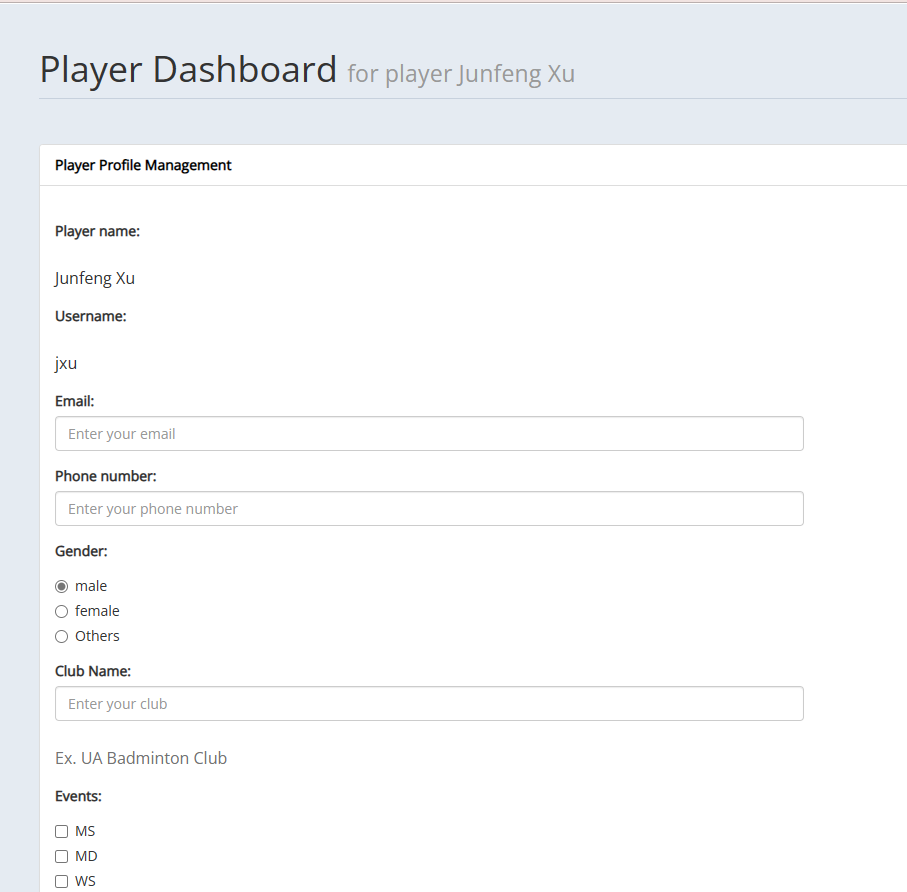
Refer to the Introduction section.

## Screenshots of What’s Working:

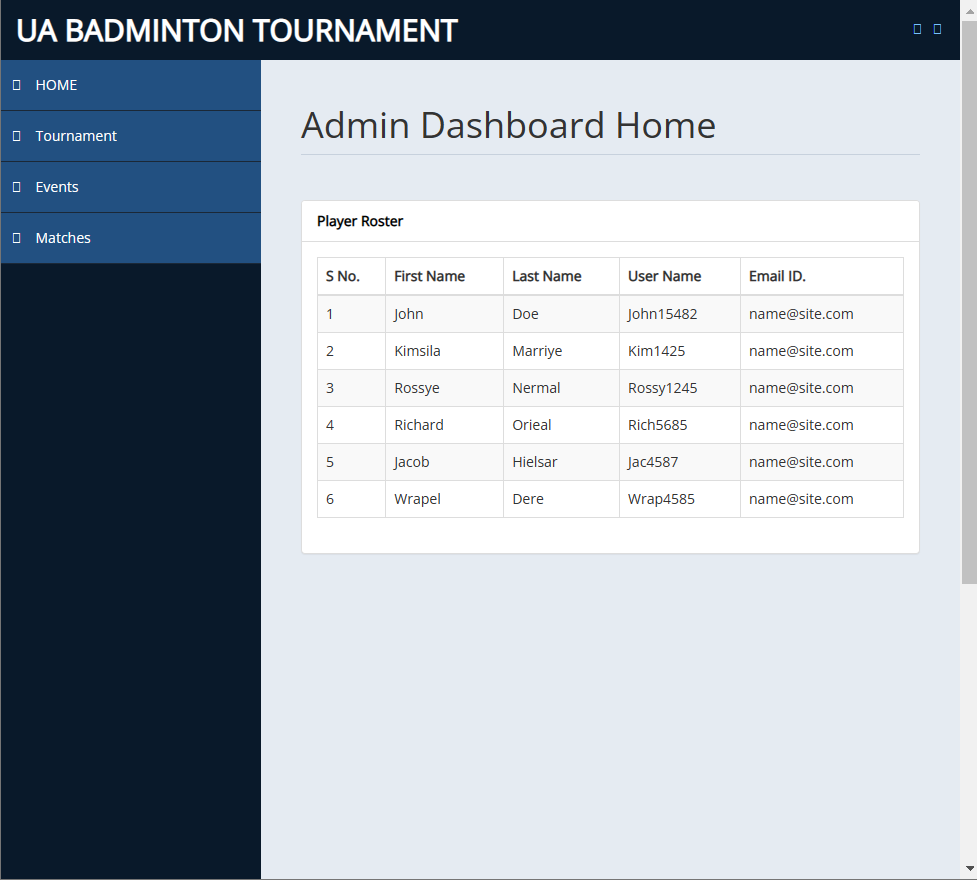
1. Sign in and Sign up page for Player and Admin:



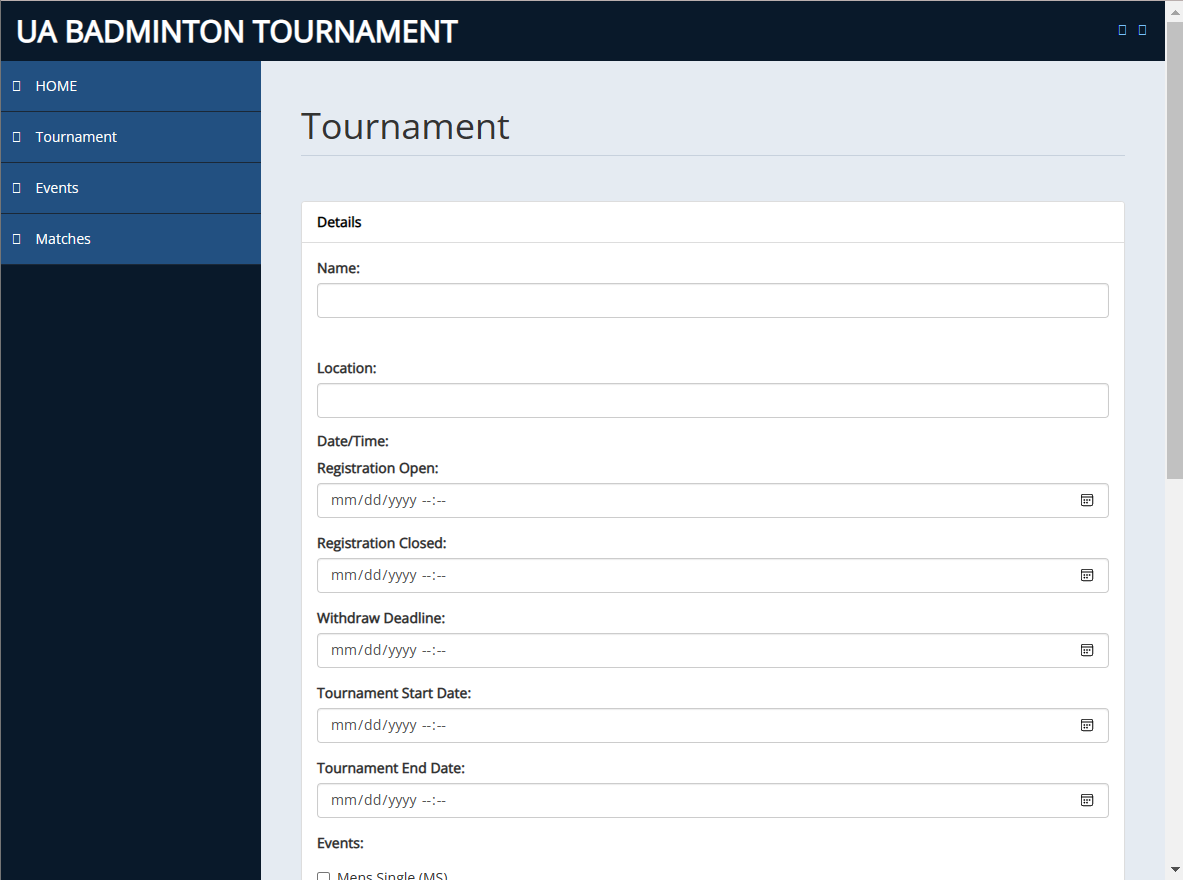
1. Player Dashboard: After sign up, the player has to register for the events, he/she needs to participate in. They can change their information by signing in again before the deadline of registration.



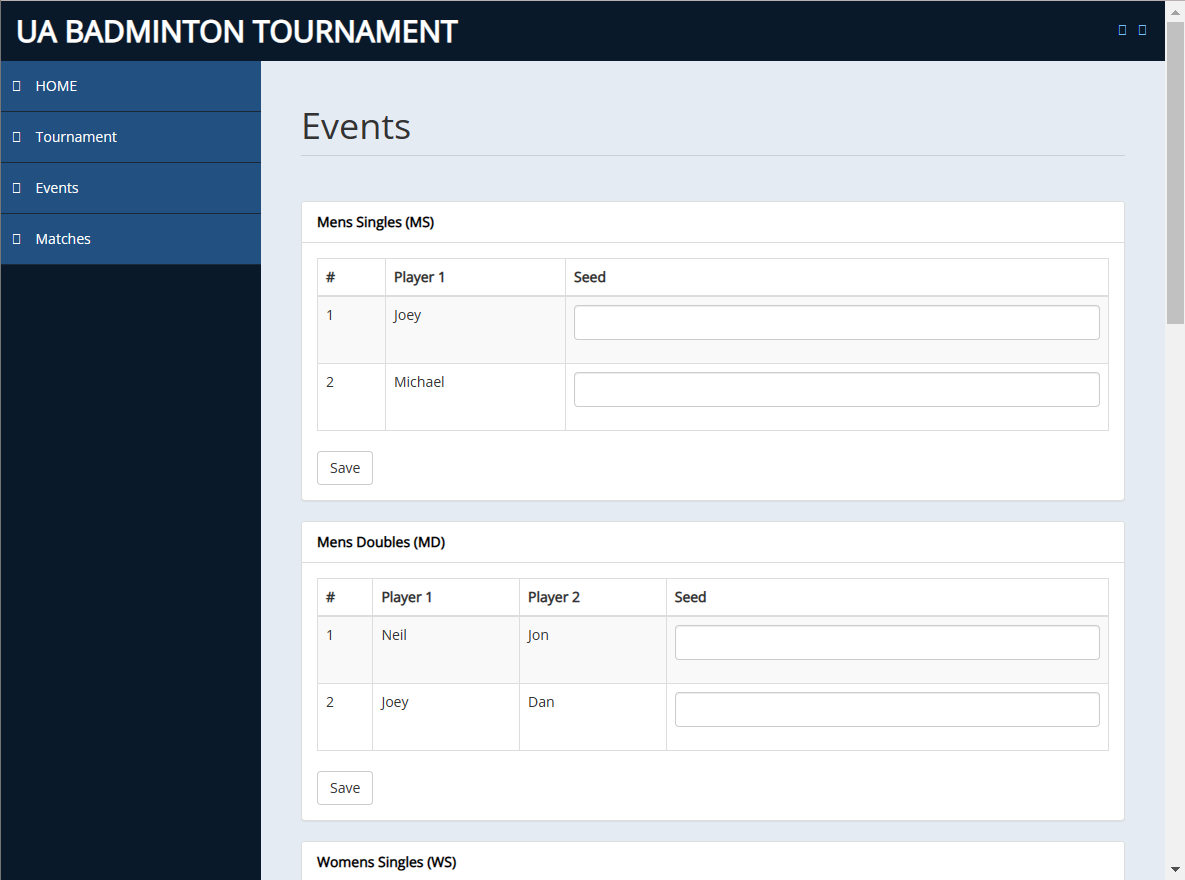
1. Admin Dashboard
   1. Player Roaster: This will display the details of all the players participating in the tournament. There are 3 tabs on the left bar and a profile button in the top right corner of the application.



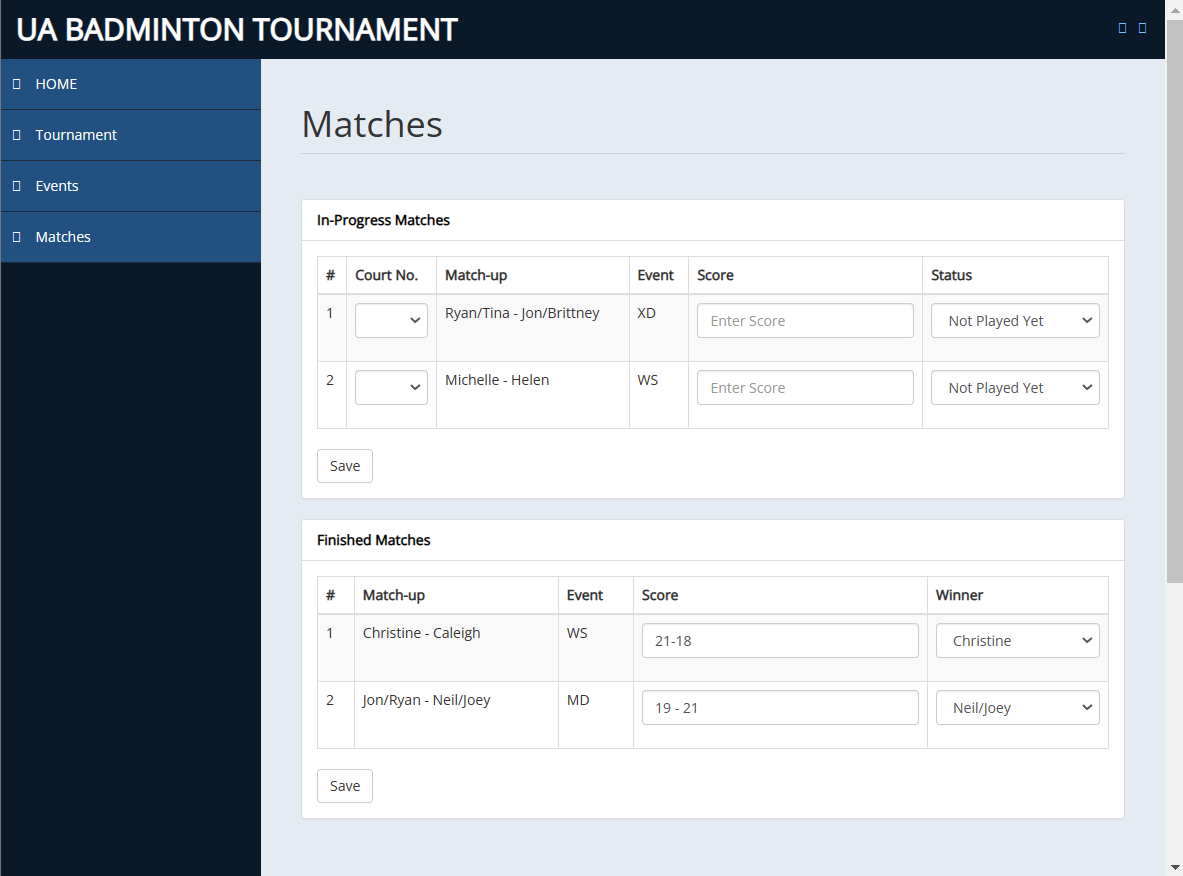
* 1. Tournament Creation: Admin will be able to enter the tournament details and all the announcements and edit them at any time. The admin will click on Create button, at the end of page, and the public page will get populated with the details



* 1. Events: All the participating players for each event will be listed.



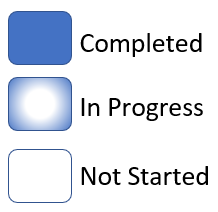
* 1. Matches: Real time in-progress and finished matches track record.



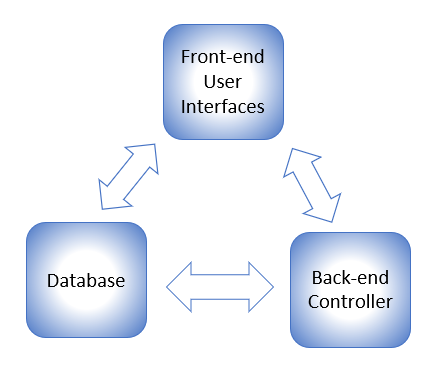
## Challenges We are facing:

1. One of the main challenges we faced was designing and implementing the database using SQLAlchemy and Flask framework for the badminton tournament. There are a significant number of tables or entities required such as Events, Tournaments, Matches, that form a ternary relationship, in addition to Admin, Player and Result. This required a deep understanding of database design principles and technical skills using these frameworks (SQLAlchemy and Flask). Since database design was new to everyone in the team, we had less expertise in this area.
2. Hosting the database on a centralized server was also a significant challenge, as it required selecting an appropriate hosting platform and ensuring that all team members had access to the database. We wanted to avoid using a local server because it requires everyone to setup the database on their machine locally, this is a repeated effort.
3. Since all of us are graduate students with other jobs, it is often difficult to find a time that works for everyone. Sometimes a preset meeting needs to be rescheduled because one or more of us had a change in schedules and are unavailable.
4. As a team, we faced a challenge with developing the frontend web application since it was our first time building a complete frontend from scratch. This required us to spend additional time researching and experimenting with different frameworks and tools to identify the best approach. We had to ensure that the frontend was user-friendly, responsive, and visually appealing while maintaining compatibility with the backend functionalities.

## Block diagram for the current state of the system:

(Block color code:)

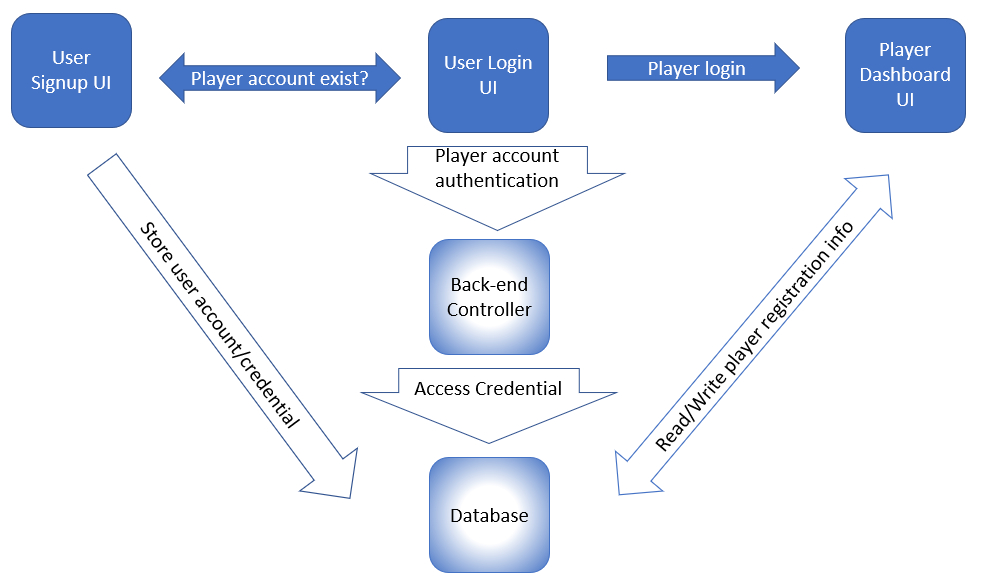
### System overview block diagram:



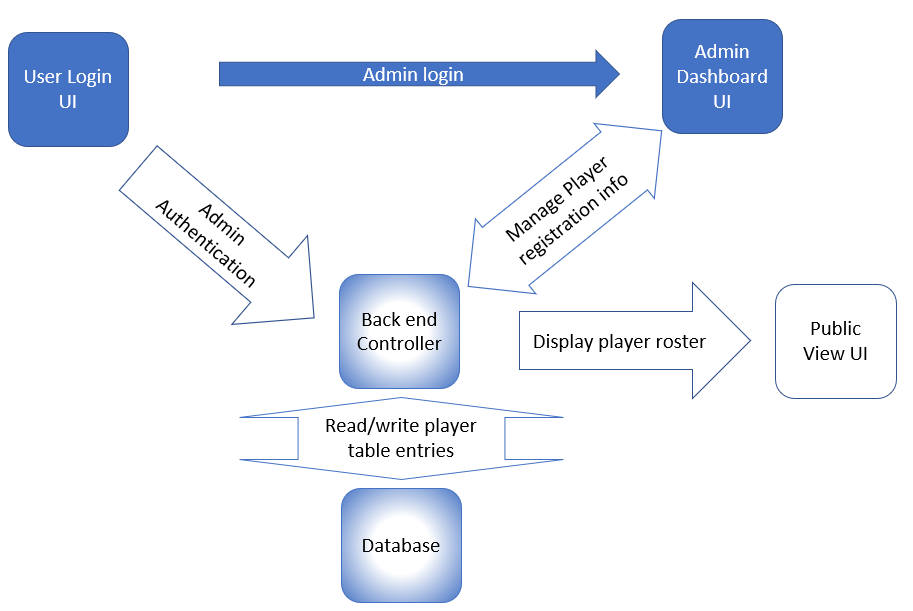
### Modular block diagram:

Note: System modules are Registration, User Management, Seed & Draw, and Match Management.

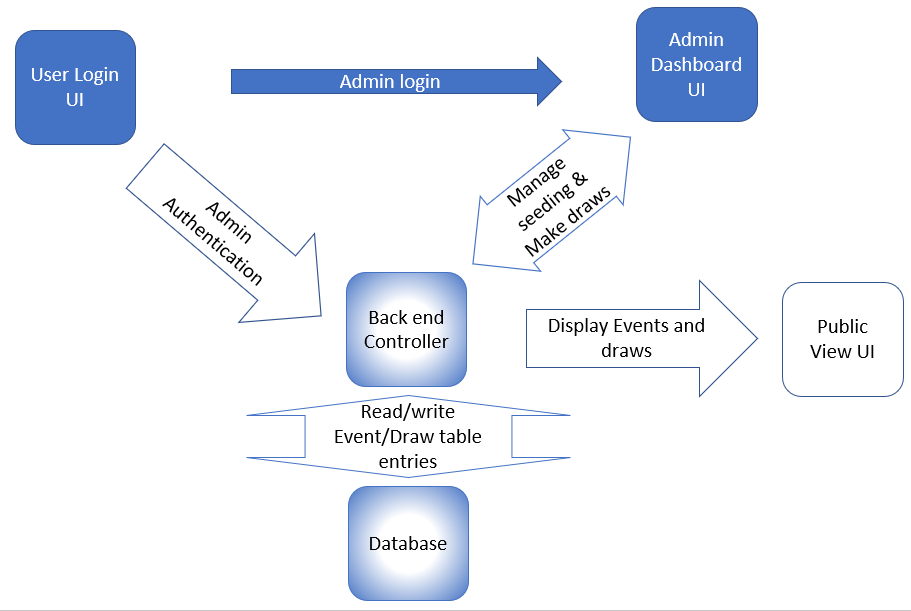
#### Registration module:



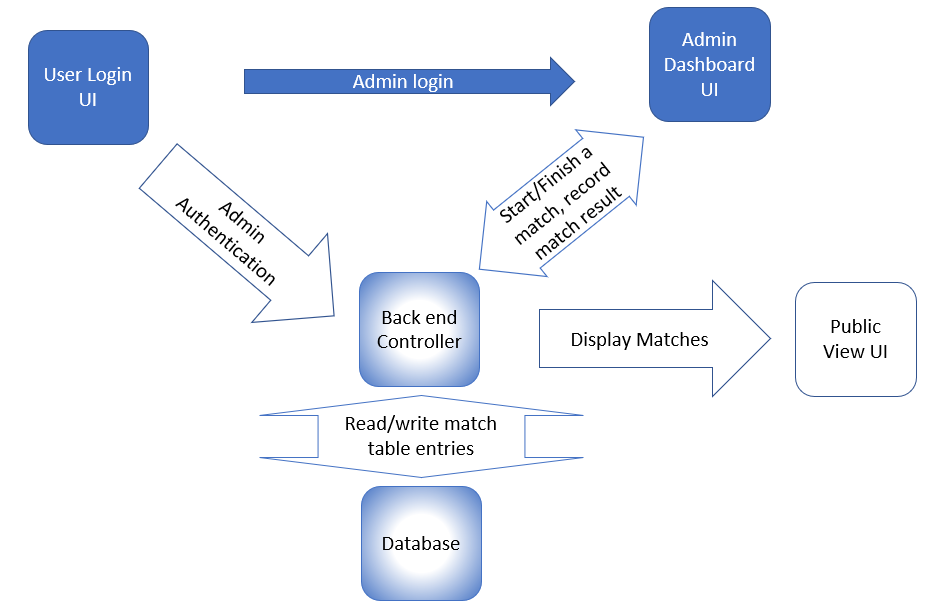
#### User Management Module:



#### Seed & Draw Module:



#### Match Management Module:



## Test Performed:

We have performed unit testing on all the front end user interfaces we have implemented so far which consist of HTML and CSS code. Scaffolding was used in order to invoke the front end code since the backend support is not completed yet.

## Total number of lines of code we collectively written:

~ 1689 loc

# 

# **6. Project Management**

Continue to maintain the Change Log. Add any new changes to the project, tracking the date, motivation, description, and implications of each change.

## Change Log

| Date | Description | Motivation | Implementation |
| --- | --- | --- | --- |
| 03/02/2023 | UI Design Change | To make the admin dashboard more user-friendly and easier to update. | Updated UI design to have an admin dashboard in a single screen instead of multiple screens. This will make it easy for the Admin to make updates. |
| 03/07/2023 | Database redesign | To fix errors with primitive data types in initial design. | Reworked the structure of the database as previously we found some design error, and created a new ER diagram. |
| 03/12/2023 | Customer desired over all experience change | To satisfy the customer’s updated requirements that we have learned from the customer’s feedback. | Given our current design for the project, it is still able to address customer’s new surfaced problems. |

## Goals for the next iteration:

1. The primary goals for the next iteration are to complete the remaining backend functionality and integrate it with the frontend UI, primarily the login/signup feature.
2. Creating the database tables.
3. Creating a player dashboard that can collect player profile information.

We also aim to conduct extensive testing to identify and resolve any potential bugs or issues, as well as improve the user experience through user testing and feedback.

## Plan for the rest of the semester

For the rest of the semester, our plan is to focus on completing the remaining features and functionality of the application. This includes:

1. Finalizing the player dashboard, ensuring that it is fully functional and can collect and display all necessary player profile information.
2. We will also work on developing the admin dashboard, with a specific focus on implementing tournament creation functionality, implementing make draw functionality and ensuring that tournaments can be started and managed seamlessly.

Another key focus area will be on match management, with the goal of implementing a feature that can display all first round matches in a clear and user-friendly format.

Additionally, towards the end we will brainstorm on optimizing the application's performance and scalability to ensure it can handle large volumes of data and traffic. Overall, our goal for the next iteration is to deliver a robust, user-friendly, and fully functional application that meets a major portion of customer’s requirements and exceeds their expectations.

## **Project Management Tools**

### Jira to manage the sprint.

By utilizing Jira to manage our sprints, we were able to track our progress, track our time making sure we are not underestimating and overestimating timelines and make adjustments as needed to ensure that we stayed on track towards achieving our sprint goals.  
  
During sprint planning, we focused on three critical areas:

**Sprint goals:** To get the team to think about the essence of the sprint and what value we will create for our customers when the sprint concludes.

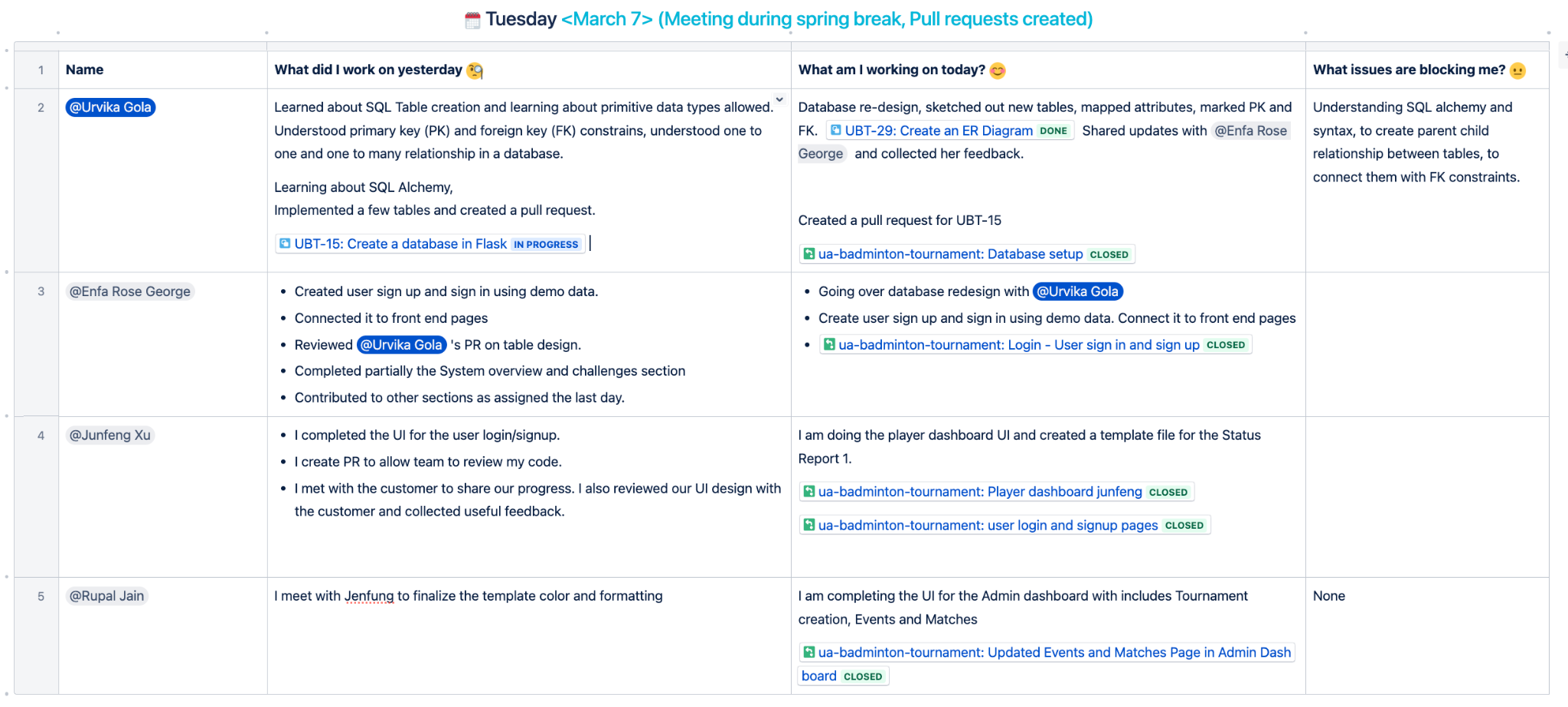
**Sprint tasks:** After this, we identified the specific tasks and user stories to be picked from the product backlog that will help fulfill the sprint goal.

**Sprint execution:** We planned how the tasks would be accomplished by dividing into subtasks. and who would be responsible for each.

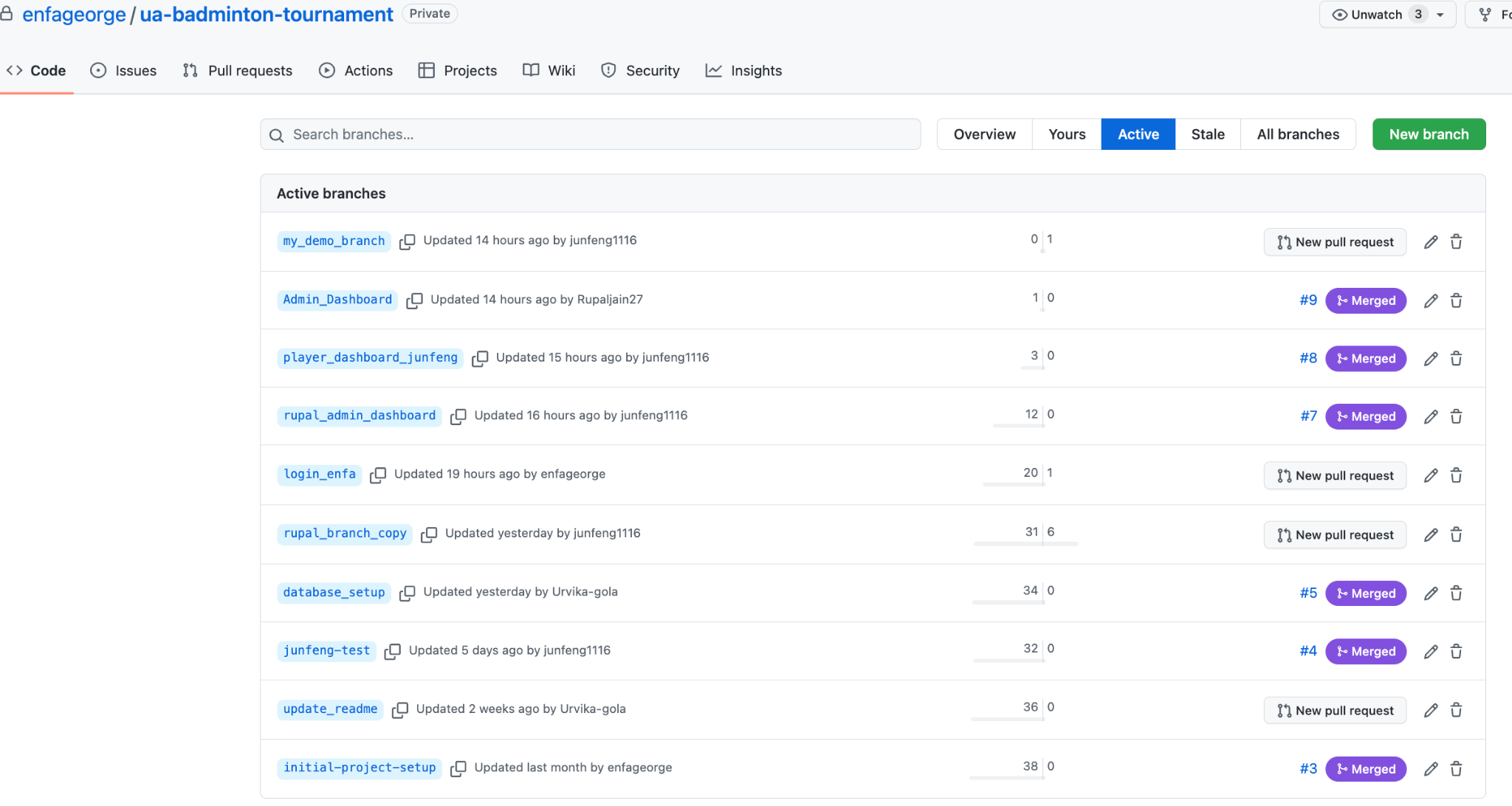
### 

### Confluence to manage daily updates.

Helped us track progress and identify roadblocks. We also integrated Github and Jira to Confluence to view everything under one umbrella.



### Github for version control and managing releases



# 

# **7. Team**

## Team roles for this iteration?

During this iteration, our team had a well-defined set of roles and responsibilities. Junfeng Xu served as the Product Owner and contributed to the designing and development of the Player Dashboard UI. Urvika Gola acted as the Scrum Master, ensuring that the team stayed on track and met customer goals. In addition to managing the Jira board, Urvika also played a key role in designing the database from scratch and also ensured optimal performance of the system. She designed the ER diagram, which gives the overall picture of the system. Rupal Jain was one of the developer and contributed to the design of the system's UI. She was responsible for completing the development of the Admin Dashboard UI. Enfa Rose served as our Software Architect, and she was instrumental in designing the basic structure of the system. Additionally, Enfa and Urvika together designed the database with necessary fixes to ensure optimal performance.

## Team Member Contribution:

1. Junfeng Xu contributed on the overall high level design based on the customer need as a product owner, communicating with the customer and collecting feedback, the user interface design and implementation, and peer reviews on code and design.
2. Urvika contributed to database design along with Enfa during one of the brainstorming sessions. Urvika reworked on this database design and created an ER diagram that represented the new entities, relationships, and constraints of the database. This was to represent the logical representation of the database. Urvika learned about SQLAlchemy in the Flask framework and implemented a part of the total tables required for the project. Being the scrum master, Urvika worked on the project management front with the team, and performed code reviews for backend related pull requests.
3. Rupal played a key role in designing the front end of the complete system. She began by brainstorming the overall flow of the application, starting from tournament creation to player signup, registering for events, making draws, and concluding the tournament. Rupal also contributed to finalizing the UI design, including the color scheme, tables, tabs, and other elements. She took the lead in developing the Admin Dashboard, which included sub-pages such as tournament creation, displaying events, and matches under each event, as well as tracking in-progress and finished matches.
4. Enfa set up the basic project in github with instructions so that everyone can start off the project. She extensively brainstormed with Urvika about database design. Specifically, what data structures are needed for each of the customer requirements. How can we design it such a way that there are minimal db calls and logic is handled well. Enfa and Urvika together built the first iteration of database design. Once database design was fixed (it needed one more iteration - handled by Urvika), Enfa implemented the functions needed in backend for the signin and signup functionalities for users and admin.

## Rough estimates of the each team member contributions to the project (in percentage)

Based on our team's estimates, it appears that each team member made significant contributions to the application’s success. Rough estimates of the percentage contributions indicate that all team members contributed equally, with each member accounting for 25% of the project's overall progress. This demonstrates the collaborative and cooperative nature of our team, and the commitment of each member to ensuring the project's success. By sharing ideas, supporting one another, and working together, we were able to accomplish our goals and deliver a high-quality outcome of the sprint.

# 

# **8. Reflection**

## What went well?

Frontend Development: We were successful in achieving their frontend development goals, having designed the complete frontend of the system. One of the initial hurdles was figuring out how to separate the Admin, Player, and public views while maintaining the integrity of the information. We came up with a solution that involved having separate dashboards for players and admins, and a separate website for public view. As a result of this decision, changes had to be made to the database design as well. Despite these changes, we were able to surpass the initial frontend development goals, which was a significant success. We used HTML, CSS, Font Awesome, and Bootstrap to make the application responsive and user-friendly.

Database creation: We were able to learn and implement new techniques for database creation, including the use of SQL Alchemy and the Flask framework. This allowed us to create a robust and efficient database system that could handle the complex requirements of our tournament management software.

Project management: Our project management process was highly effective, thanks to tools like the Scrum dashboard and Atlassian Confluence. We used these tools to stay organized and ensure that everyone on the team was up-to-date on our progress. For example, we held daily stand-up meetings where each team member could share their updates and any issues they were facing, and we used the Scrum dashboard to track our progress and identify areas where we needed to focus our efforts.

Project setup: We took great care in preparing the initial setup of our project, using GitHub to ensure seamless version control and collaboration. This allowed us to easily manage changes and updates to our codebase, and made it easy for everyone on the team to work together efficiently.

## What didn’t go well?

Frontend redesign: Although we were able to deliver a comprehensive frontend that exceeded our initial plan, we received feedback from the team that the design could be improved. In response, we held additional meetings to gather feedback from all team members and develop a more cohesive design that meet everyone's needs.

Database redesign: Despite careful planning and preparation, we discovered a design flaw in the initial database structure that required a complete redesign. It required us to spend additional time and resources on reworking the database. However, we were able to learn from this experience and put in place a more robust system.

## Goals that were not met and their issues

For the database creation goal that did not go well, we underestimated the scope of work required for the database creation, as well as the level of technical expertise required to implement it effectively.

## Plans to overcome the issues

1. Better communication: We recognize that effective communication is critical to project success. We will encourage all team members to share their ideas and feedback openly and collaborate more closely to ensure that everyone's perspectives are taken into account.
2. Better estimation of project tasks: We acknowledge that accurate project estimation is key to delivering projects on time and on budget. To improve our estimation processes, we will conduct more thorough research and analysis before setting project goals and breaking down tasks. We will also work to identify potential risks and roadblocks early on in the project and develop contingency plans to address them.

## Plans to do differently in the next iteration

1. Communicate asynchronous more often: During the next iteration, we plan to improve our communication by scheduling regular asynchronous check-ins. This will allow team members to share their progress, ask questions, and offer feedback on their own schedules. Asynchronous communication can be a useful tool to ensure that everyone stays up-to-date on the project without interrupting their workday.
2. Check in with others who have worked on similar tasks: To improve our project estimation, we plan to reach out to team members who have worked on similar tasks in the past. This will help us get a better understanding of the potential challenges and time required to complete each task. By leveraging the knowledge and experience of our team members, we can develop more accurate estimates and ensure that we are on track to deliver the project on time.