System Requirements Specification

12.2023

Team Zeta

Jada Bonnett, Ashley Moore, Ellie Stamper, Mason Holland, Luke Balkema

Taylor University

Upland, IN

Table Of Contents

1. General System Information

1.1. Introduction

This System Requirements Specification describes functional and non-functional requirements for a bible quizzing scoreboard and practice system for the FEC churches. This document provides a coherent design that will assist software engineers and system analysts in the creation and implementation of the system.

1.2. System User Roles

1.2.1. Player/Quizzer

In this quizzing community, the main person is the quizzers themselves. This person will vary from 7th-12th grade and belongs to one of the FEC churches that participate in the Quiz Off's. In our system, the player will have multiple different functionalities. The main purpose of this for the player is to practice and study different passages or questions to prepare for their next quiz. Along with practicing questions, coaches will also be able to send out messages and announcements to team members about upcoming practices or quiz-offs.

1.2.2. Team Coach

A coach in this regard is one person for each team, that is "in charge" and puts together practices and organizes groups of 3 or more players or students per team. For the coach, the main purpose of use here is to communicate with and encourage players in their studies. There are a few functions that coaches will have to accomplish this. First off, they will be able to create specific practice quiz questions to allow players to practice with. Having this functionality gives more individual questions to practice based on the team. Along with creating questions, coaches will also be able to create announcements for their team. This can include things like the next practice time or details for the Quiz Offs.

1.2.3. Quiz Masters

For all Quiz Masters, we plan on allowing users listed as "Quiz Masters" to be able to do all three roles, and each will just use the specific part needed for their

role at that time. Below we will go into each specific purpose and describe the possibilities.

1.2.4. Time-Keeper

This Quiz Master's role is to operate the lightbox equipment and timer. Each player's seat is connected to this lightbox. When a player is ready to answer the question, they will jump. The light box will indicate which player has jumped first, and they are then able to answer the question. The Time-Keepers are responsible for monitoring this process and resetting it between questions.

1.2.5. Score-Keeper

This specific Quiz Master's purpose during the quiz-off is to keep track of the score between teams and who exactly scored the points. To take this person's job from paper to screen, we have various functions to help them complete this job. They will enter a score, assign it to a person, subtract points, be able to undo an action, record timeouts, fouls, etc. They also act as an Assisting Judge to the Question-Keeper by relaying answer updates such as point deductions, errors, and fouls, and helping to determine the answer accuracy if requested by the Question-Keeper.

1.2.6. Question-Keeper

This Quiz Master's role is to "run the room." They are "in charge" of the room and will ask questions and determine the accuracy of the answers. The Question-Keeper will have access to all quiz questions and answers. They are responsible for delivering the questions and announcing the outcome.

1.2.7. Churches

Multiple churches are involved in a quiz bowl. Each church will have multiple teams of players that will compete. The churches will be able to supply their teams with a set of study materials such as the text and practice questions. The churches can also be the location of the quiz bowl and will therefore have and participate in a schedule of when and where the quiz bowls will take place. They will have access to add these materials and update the schedule in the system.

1.2.8. Quiz Bowl Leadership

There are six leadership positions. Bruce Rocke is the Quiz Director. He is responsible for creating the pool of questions for the quiz-off, creating the schedule, updating the rule book, and is the main representative that has the “final word” on any debate, confusion, or questions. In addition to the Quiz Director, the Quiz Master Coordinator will have access to review and edit quizzes. The other leadership positions will have access to view and interact with the system in a higher role. They will be able to add to/update the schedules, send out announcements, and add study materials.

1.3. Integrations/Collaborating Systems

These integrations including the Bible App, Quizlet, and Google Calendar allow this system to have added functionality and incorporation. These APIs will be integrated into the system with varying functions.

1.3.1. Bible App via Web API

Bible.com will be used to provide access to the bible, to read and study. There will be functionality added to the bible.com API using local processes.

1.3.2. Quizlet via Web API

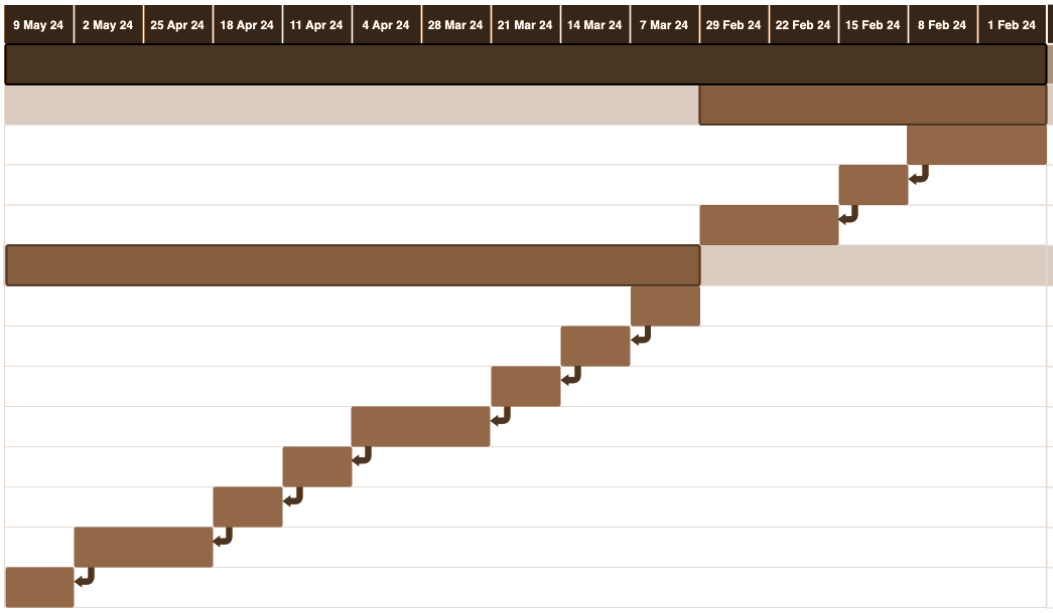
Quizlet will be used to provide access to flashcards that students can use to study for bible bowls.

1.3.3. Google Calendar via Web API

Google Calendar will be used to give coaches the ability to add dates for Bible Bowls and practices, players will be able to see these dates but will not have the ability to edit or add events.

1.4. Project Timeline

This document is the expected timeline for the application design process. This schedule is created within the necessary spring semester time restriction. The spring semester lasts 16 weeks, 15 weeks of classes, and 1 week of finals, with students expected to spend 6-8 hours every week working on the project. The schedule is created with the assumption of a team of 5 students. The team is expected to work 16 weeks, with the project will begin development on February 1, 2024. The project will begin with 5 weeks of pre-development, with the rest of the time devoted to system design, with an expected end date of May 16, 2024.



ETA	Start	Duration	Task Name	
5.16.12	2.01.24	15 weeks	Complete project execution	1
3.07.24	2.01.24	5 weeks	Pre-development	2
2.15.24	2.01.24	2 weeks	Design Visuals	3
2.22.24	2.15.24	1 week	Choose Framework	4
3.07.24	2.22.24	2 weeks	Develop Requirement Tests	5
5.16.12	3.07.24	10 weeks	Development	6
3.14.24	3.07.24	1 week	Developing User Section of the Database	7
3.21.24	3.14.24	1 week	Developing Practice Section of Database	8
3.28.24	3.21.24	1 week	Developing Past Quizoff Section of Database	9
4.11.24	3.28.24	2 week	Check Security and Bug Fix	11
4.18.24	4.11.24	1 week	Create Documentation	12
4.25.24	4.18.24	1 week	Setup on Hosting Server	13
5.09.24	4.25.24	2 week	Requirement Verification	14
5.16.24	5.09.24	1 week	Develop User Manual	15

2. Functional Requirements

2.1. System Response Table

The System Response Table (SRT) lists various events that the system will be able to perform, labeled with a title, ID, source, trigger, event response, major output, and external destination. This aids in understanding the system's functions, which will aid in system construction.

The particular source triggers a response on behalf of the system, which will produce one or more major sources of output.

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Create account	U1	User	Desire to create account	User account created	Account created	User
Log in	U2	User	Desire to log in	User logged in	New session	Server
Log out	U3	User	Desire to log out	User logged out	Notification	User
					Session deleted	Server
Listen to Bible	U4	User	Desire to listen to Bible	Bible audio is played	Bible audio	User
Read Bible	U5	User	Desire to read Bible	Bible page is opened	Bible page	User
Bookmark place in Bible	U6	User	Desire to save place in Bible	Page is marked	Bible page saved	User Device
Highlight verses	U7	User	Desire to highlight verse	Verse is marked and highlighted	Highlighted verse	User Device
View current scoresheet	U8	User	Desire to know current score	Scoresheet is shown	Scoresheet	User
View current leaderboard	U9	User	Desire to see summary of team scores	Leaderboard is shown	Leaderboard	User

View tournament bracket	U10	User	Desire to see team competition schedule	Schedule is shown	Schedule	User
View announcements	U11	User	Desire to view announcements	Announcements are shown	Announcements	User
View calendar	U12	User	Desire to see upcoming events	Calendar is shown	Calendar	User
View past scoresheets	U13	User	Desire to see past scoresheets	Scoresheet is shown	Scoresheet	User
View statistics	U14	User	Desire to see individual and team rankings	Statistics are shown	Team or individual statistics	User
Use practice questions	U15	User	Desire to practice for events	Practice questions displayed as flashcards	Flashcard-styled questions	User
Edit current scoresheet	M1	Quiz Master	Desire to add scores or edit mistakes on current scoresheet	Numbers change in scoresheet	Updated scoresheet	Quiz Master
Display current leaderboard	M2	Quiz Master	Desire to display current leaderboard	Leaderboard projected for audience	Leaderboard display	Quiz Master
Verify role	A1	Admin	Desire to assign role to system user	Role verified	Role added to profile	Database
					Notification	Admin
Create round robin	A2	Admin	Desire to create round robin matches for teams	Teams assigned to round robin matches	Round robin schedule	Database
					Notification	Admin

Add quiz off event to calendar	A3	Admin	Desire to add Quiz Offs to calendar	Quiz Off event added to calendar	New event	Database
					Notification	Admin
Edit quiz off event in calendar	A4	Admin	Desire to edit mistakes in Quiz Off events in calendar	Quiz Off event edited	Updated event	Database
					Notification	Admin
Add scoresheet to database	A5	Admin	Desire to add scoresheet to past scoresheet database	Scoresheet added to database	Scoresheet	Database
Create team	C1	Coach	Desire to create a new team	Team created	Team	Database
Add player to team	C2	Coach	Desire to add player to a team	Player added to team	Team members	Database
Remove player from team	C3	Coach	Desire to remove player from a team	Player removed from team	Team members	Database
Add team event to calendar	C4	Coach	Desire to add team event to calendar	Team event added to calendar	New event	Database
					Notification	Coach
Edit team event in calendar	C5	Coach	Desire to edit mistakes or dates in team events already in calendar	Team even edited	Updated event	Database
					Notification	Coach
View chat	C6	Coach, Quizzer	Desire to view chat messages	Chat displayed	Chat	User
Send message to chat	C7	Coach, Quizzer	Desire to send a message to someone in the chat	New message in chat sent	Message	Database

Send announcement	C8	Coach, Quiz Master, or Admin	Desire to send announcement to all users	New messages in announcements sent	Message	Database
Upload practice questions	C9	Coach or Admin	Desire to add practice questions	New practice questions added	Practice questions	Database
Edit practice questions	C10	Coach or Admin	Desire to edit mistakes in existing practice questions	Existing practice questions edited	Updated practice questions	Database

2.2. Use Cases and Logical Process Models

Use cases expand on the SRT events by describing what will happen as particular users interact with the system. They aim to describe with greater detail how the user acts with the system to create a specific output or reach a particular goal, as well as how the system looks at the beginning and end of an event.

Each use case contains a Data Flow Diagram, which represents the flow of system state from beginning to end of the process, and the Process Specification describes the process verbally as it might be represented in code.

U1 – Create an account

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Create account	U1	User	Desire to create account	User account created	Account created	User

ID: U1

Name: Create Account

Primary Actor: User

Other Actors: N/A

Description: The user will open the app and create a new account

Preconditions: User does not already have account in system

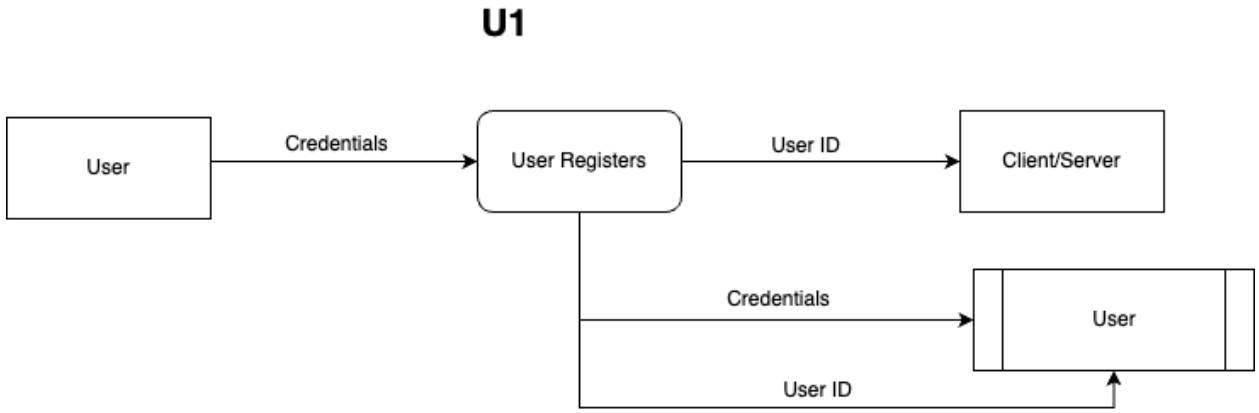
Trigger: User desires to create a new account

Typical Event Flow:

- Actor: clicks on “Create Account” tab
- Actor: completes all required account information fields
- Actor: clicks on “Create Account” button
- System: creates new account for specified user and saves account information

Conclusion: User has created a new account in the system

Postconditions: A new account has been created in the system for the user and the account settings have been saved to the database



Process Specification:

Create Account

Get User Credentials

Create User ID

Credentials and User ID combine into User Account

Using User Account, create User instance(s)

U2 – Log In

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Log in	U2	User	Desire to log in	User logged in	New session	Server

ID: U2

Name: Log In

Primary Actor: User

Other Actors: N/A

Description: A user logs into their account, which creates a new session on the web server.

Preconditions: The user already has an account in the system.

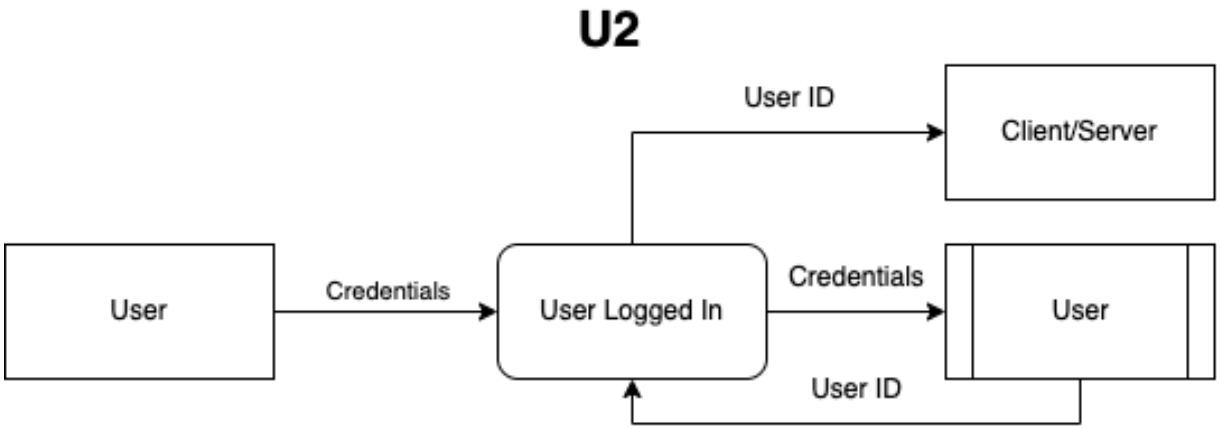
Trigger: The user desires to log into their account.

Typical Event Flow:

- Actor: types in the username and password on the page
- Actor: clicks “log in” button
- System: allows access to the user’s account and permissions and displays the home page
- System: creates a new session tied to the user’s account

Conclusion: The user is logged into their account.

Postconditions: A session has been created on the web server for the user’s account.



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

U3 – Log Out

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Log out	U3	User	Desire to log out	User logged out	Notification	User
					Session deleted	Server

ID: U3

Name: Log out

Primary Actor: User

Other Actors: N/A

Description: A user logs out of their account and ends their session in the web server.

Preconditions: The user has an account in the system and is currently logged into their account.

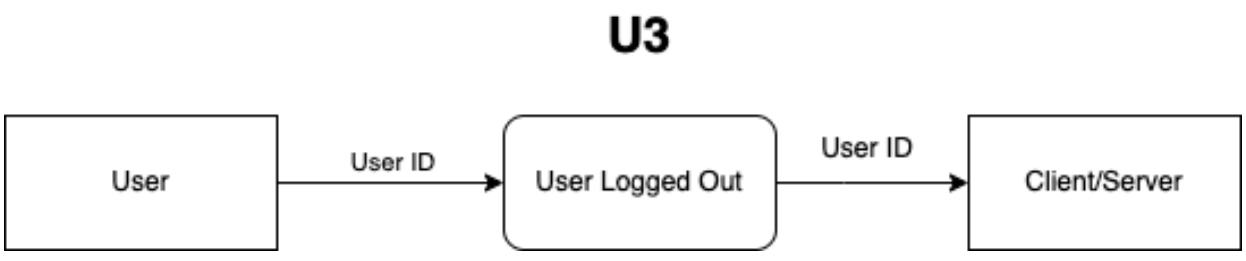
Trigger: The user desires to log out of their account.

Typical Event Flow:

- Actor: The user clicks on their profile
- Actor: The user clicks “log out”
- System: The user receives a notification that they have been successfully logged out and is returned to the log in page.
- System: The current session is ended

Conclusion: The user has logged out of their account.

Postconditions: The user’s session has been ended in the system.



Process Specification:

Log user out of the system

Get User ID

Using User ID, delete the User’s cookie/session on the Client/Server

U4 – Listen to Audio Bible

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Listen to Bible	U4	User	Desire to listen to Bible	Bible audio is played	Bible audio	User

ID: U4

Name: Listen to Bible

Primary Actor: User

Other Actors: N/A

Description: The user logs into the system and listens to the audio Bible

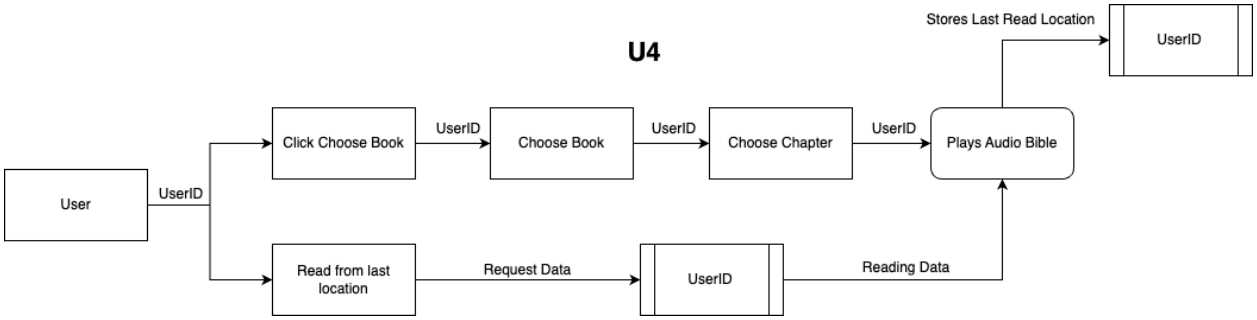
Preconditions: User has logged into the system.

Trigger: User desires to listen to the Bible

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Audio Bible”
- System: displays books and chapters or continuation option
- Actor: selects book and chapter or chooses to continue where they left off
- System: plays audio of the selected text

Conclusion: The user has played the audio of the selected Bible text



Process Specification:

Identify User

Get Credentials
Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Choose Book

Get Book and Chapter information

Using Book and Chapter information, Play Audio

Read from Last Location

Using User ID, request data from last location

Using data from last location, Play Audio

Update Last Read Location

Using User ID, store last read location to User ID

U5 – Read Bible

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Read Bible	U5	User	Desire to read Bible	Bible page is opened	Bible page	User

ID: U5

Name: Read Bible

Primary Actor: User

Other Actors: N/A

Description: The user logs into the system and reads the Bible.

Preconditions: User has logged into the system.

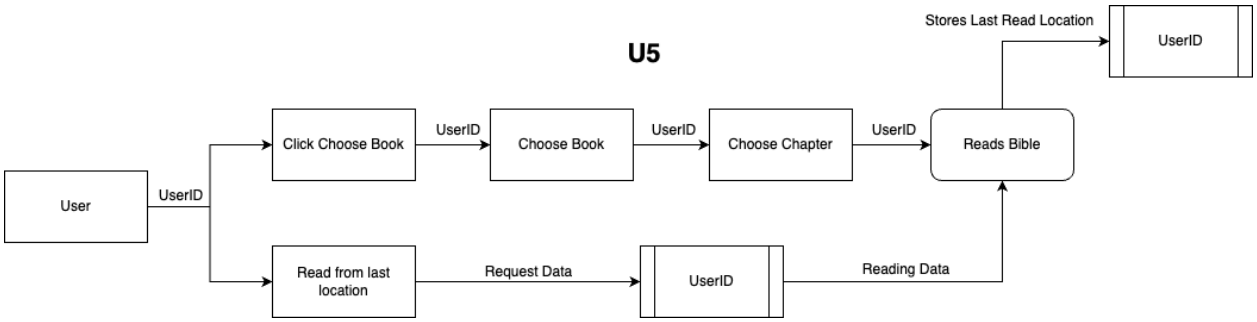
Trigger: User desires to read the Bible.

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Read Bible”
- System: displays books and chapters or continuation option
- Actor: selects book and chapter or chooses to continue where they left off
- System: opens the selected text

Conclusion: The user has opened the Bible page and text is displayed

Postconditions: The system returns to a ready state.



Process Specification:

Identify User

Get Credentials
Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Choose Book

Get Book and Chapter information

Using Book and Chapter information, display Bible text

Read from Last Location

Using User ID, request data from last location

Using data from last location, display Bible text

Update Last Read Location

Using User ID, store last read location to User ID

U6 – Save Place in Bible

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Bookmark place in Bible	U6	User	Desire to save place in Bible	Page is marked	Bible page saved	User Device

ID: U6

Name: Save place in Bible

Primary Actor: User

Other Actors: N/A

Description: The user stops reading or listening and wants to save where they left off for the next time they read or listen to the Bible.

Preconditions: User has logged into system and is currently reading or listening to the Bible.

Trigger: User has a desire to save their progress in the Bible.

Typical Event Flow:

- Actor: types in username and password on the page
- Actor: clicks “log in”
- System: allows access to the user’s account and permissions and displays the home page
- Actor: clicks “Audio Bible” or “Read Bible”
- System: displays books and chapters or continuation option
- Actor: selects book and chapter or to continue where they left off
- System: opens the selected text
- Actor: reads or listens to the Bible
- Actor: clicks “Save”
- System: logs the actor’s current spot in the continuation option and overrides any previous data

- System: sends notification to actor that their place has been saved

Conclusion: The user has saved their place in the Bible, receives a notification

Postconditions: The system adds the current place of the user to the continuation option.

U6

Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Save Place

Get Book and Chapter information

Using Book and Chapter information, create Page Number

Save Page Number to User ID

Using User ID, store Page Number on Client/Server

U7 – Highlight Verses

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Highlight verses	U7	User	Desire to highlight verse	Verse is marked and highlighted	Highlighted verse	User Device

ID: U7

Name: Highlight Verses

Primary Actor: User

Other Actors: N/A

Description: User selects a verse and highlights it.

Preconditions: User has logged into the system.

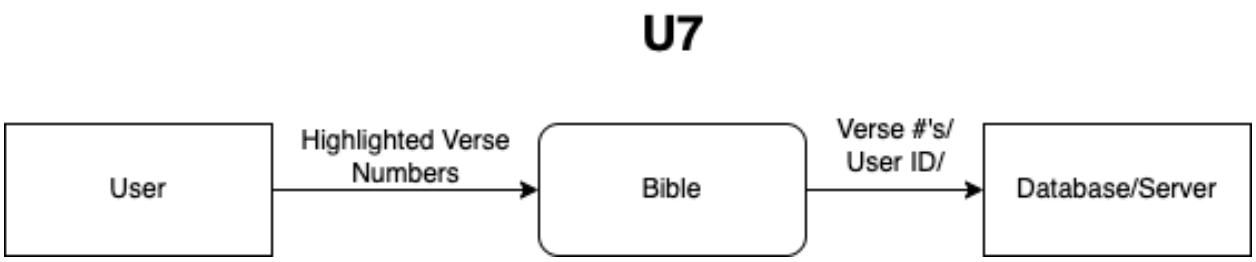
Trigger: User desires to highlight a verse

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Read Bible”
- System: displays books and chapters or continuation option
- Actor: selects book and chapter or chooses to continue where they left off
- System: opens the selected text
- Actor: clicks on verse to select
- Actor: clicks “Highlight” button
- System: displays highlighted verse

Conclusion: User marks a verse and verse is highlighted

Postconditions: selected verses are highlighted and saved to database



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Highlight Verse

Get Verse(s) information

Using Verse(s) information, color background of text

Using User ID and Verse(s) information, store to Database/Server

U8 – View Current Scoresheet

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
View current scoresheet	U8	User	Desire to know current score	Scoresheet is shown	Scoresheet	User

ID: U8

Name: View Current Scoresheet

Primary Actor: User

Other Actors: N/A

Description: User views current scoresheet

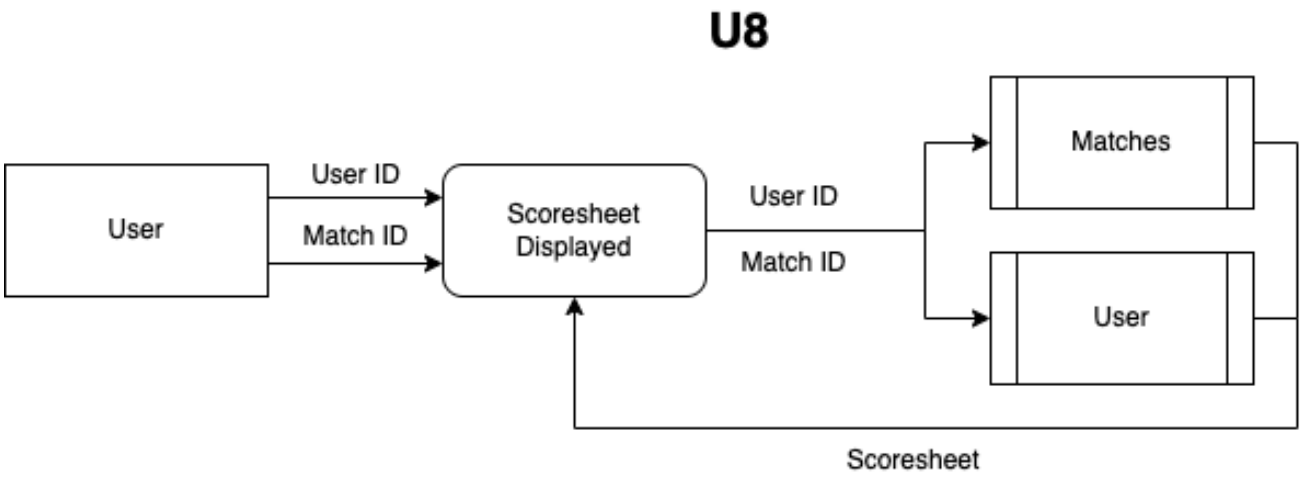
Preconditions: User has logged into the system.

Trigger: User desired to know the current score

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks on “Active Quiz Bowl” tab
- System: displays all current scoresheets
- Actor: selects the scoresheet for the match they wish to view the score of
- System: displays the selected scoresheet

Conclusion: Scoresheet of the selected match is displayed



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

View Current Scoresheet

Using User ID, get Match ID

Using User ID and Match ID, display scoresheet with matching Match ID

Store User ID and Match ID to Matches and User information

U9 – View Current Leaderboard

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
View current leaderboard	U9	User	Desire to see summary of team scores	Leaderboard is shown	Leaderboard	User

ID: U9

Name: View Current Leaderboard

Primary Actor: User

Other Actors: N/A

Description: User views current leaderboard

Preconditions: User has logged into the system.

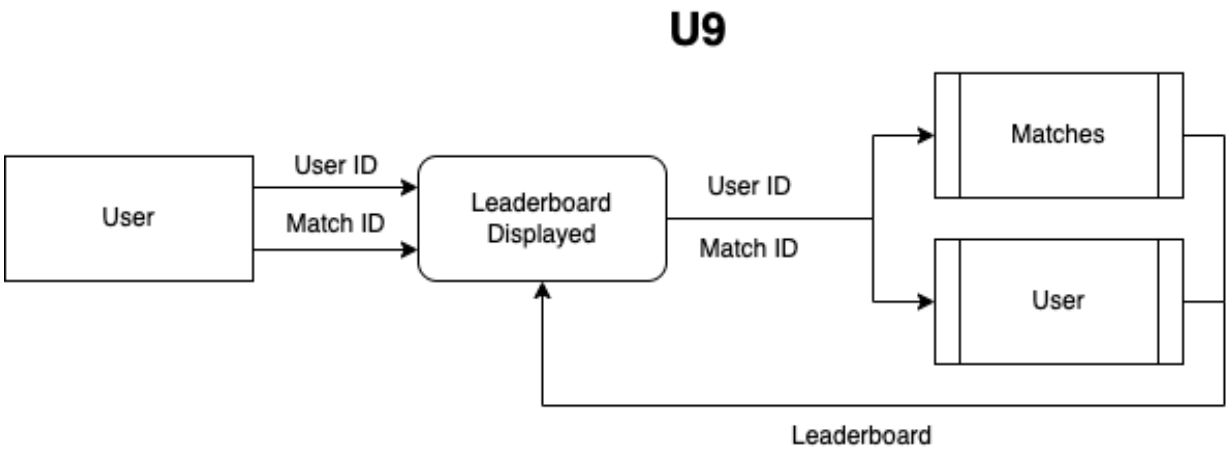
Trigger:

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks on “Active Quiz Bowl” tab
- System: displays all current scoresheets
- Actor: selects the scoresheet for the match they wish to view the score of
- System: displays the selected scoresheet
- Actor: clicks “Leaderboard” button
- System: displays leaderboard and summary of scores

Conclusion: Leaderboard of the selected match is displayed

Postconditions: The system returns to a ready state.



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

View Current Leaderboard

Using User ID, get Match ID

Using User ID and Match ID, display leaderboard with matching Match ID

Store User ID and Match ID to Matches and User information

U10 – View Tournament Bracket

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
View tournament bracket	U10	User	Desire to see team competition schedule	Schedule is shown	Schedule	User

ID: U10

Name: View tournament bracket

Primary Actor: User

Other Actors: N/A

Description: User views current schedule for round robin matches

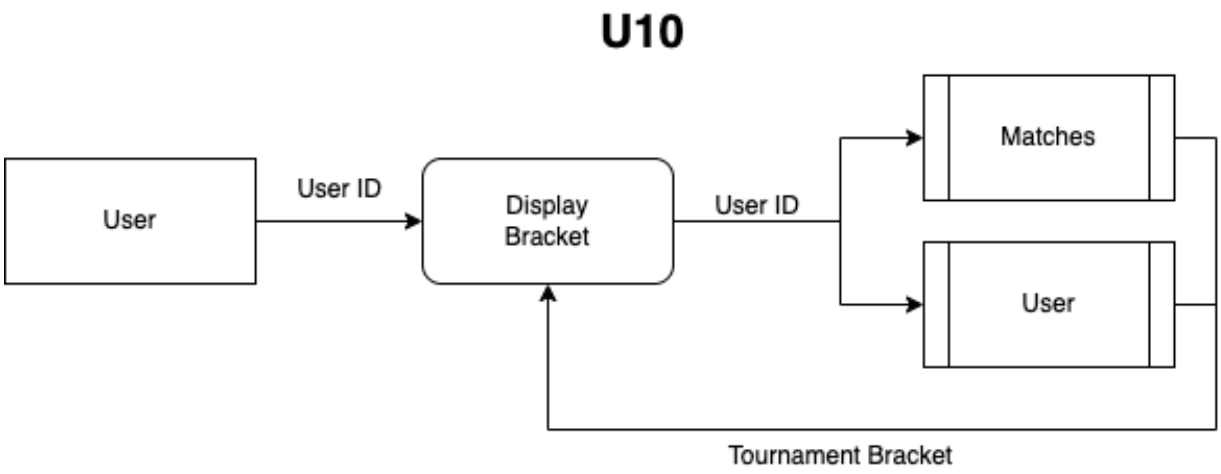
Preconditions: User has logged into the system.

Trigger: User desires to see the team competition schedule

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks on “Active Quiz Bowl” tab
- System: displays Active Quiz Bowl page
- Actor: clicks “Schedule”
- System: displays current schedule of round robin matches

Conclusion: Current Bracket is displayed



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

View Tournament Bracket

Using User ID, display bracket

Store User ID to Matches and User information

U11 – View Announcements

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
View announcements	U11	User	Desire to view announcements	Announcements are shown	Announcements	User

ID: U11

Name: View Announcements

Primary Actor: User

Other Actors: N/A

Description: User views the announcements

Preconditions: User has logged into the system.

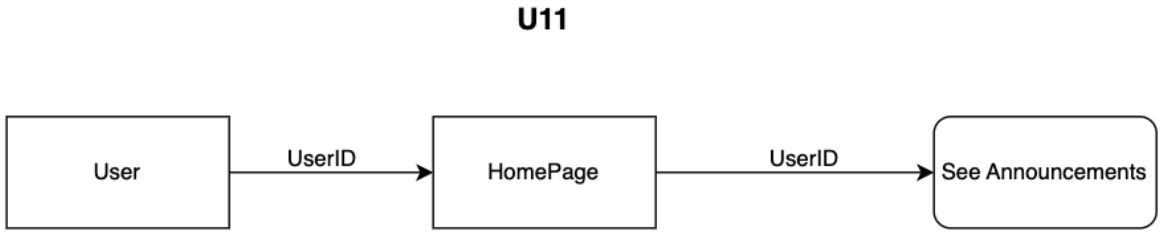
Trigger: User desires to view announcements

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Announcements”
- System: displays most recent announcements with previous announcements above it

Conclusion: Announcements are displayed

Postconditions: The system returns to a ready state



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

View Announcement

Using User ID, display announcements

U12 – View Calendar

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
View calendar	U12	User	Desire to see upcoming events	Calendar is shown	Calendar	User

ID: U12

Name: View Calendar

Primary Actor: User

Other Actors: N/A

Description: User views the calendar

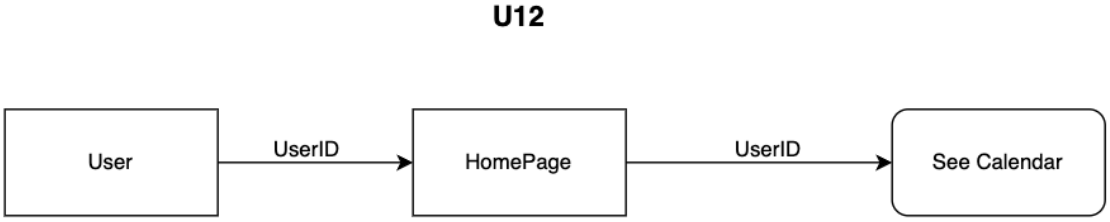
Preconditions: User has logged into the system.

Trigger: User desires to view the upcoming events

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: scrolls down homepage and clicks “Calendar”
- System: expands calendar
- Actor: selects week, month, year they wish to view
- System: displays events of selected time frame on calendar

Conclusion: Events on calendar are displayed



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Get User Team

Get User Team from page

Display Calendar

Get Team ID

Using Team ID, get Team Calendar Info

Using Team Calendar Info, display Calendar

U13 – View Past Scoresheets

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
View past scoresheets	U13	User	Desire to see past scoresheets	Scoresheet is shown	Scoresheet	User

ID: U13

Name: View Past Scoresheets

Primary Actor: User

Other Actors: N/A

Description: User views a past scoresheet

Preconditions: User has logged into the system.

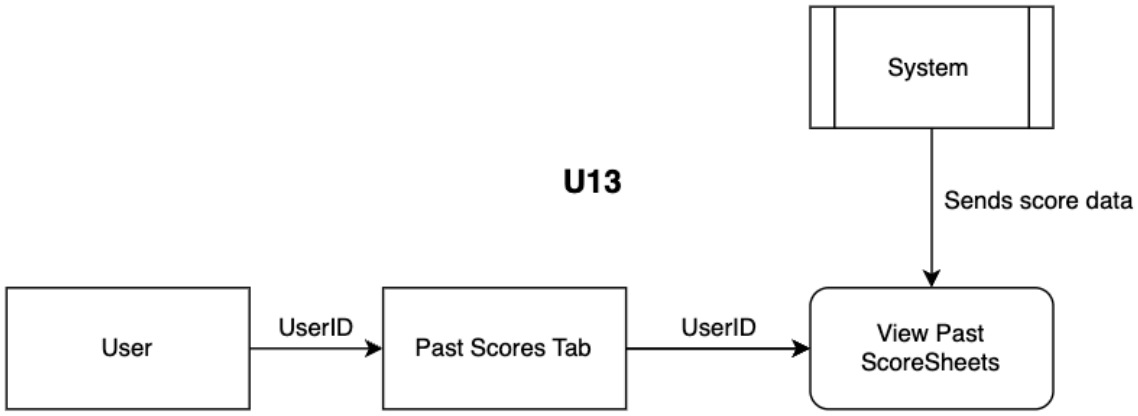
Trigger: User desires to see past scoresheets.

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks on “Past Scores” tab
- System: displays all past scoresheets
- Actor: selects the team for the scoresheets they wish to view and the relevant Quiz Off
- System: displays the selected teams’ scoresheets of the specified Quiz Off
- Actor: clicks the scoresheet to expand view

Conclusion: Past scoresheets are displayed

Postconditions: The system returns to a ready state



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

View Past Scoresheet

Using User ID, get Match ID

Using User ID and Match ID, display scoresheet with matching Match ID

Store User ID and Match ID to Matches and User information

U14 – View Statistics

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
View statistics	U14	User	Desire to see individual and team rankings	Statistics are shown	Team or individual statistics	User

ID: U14

Name: View Statistics

Primary Actor: User

Other Actors: N/A

Description: The user desires to view current rankings of players or teams, or past rankings for individual players or teams

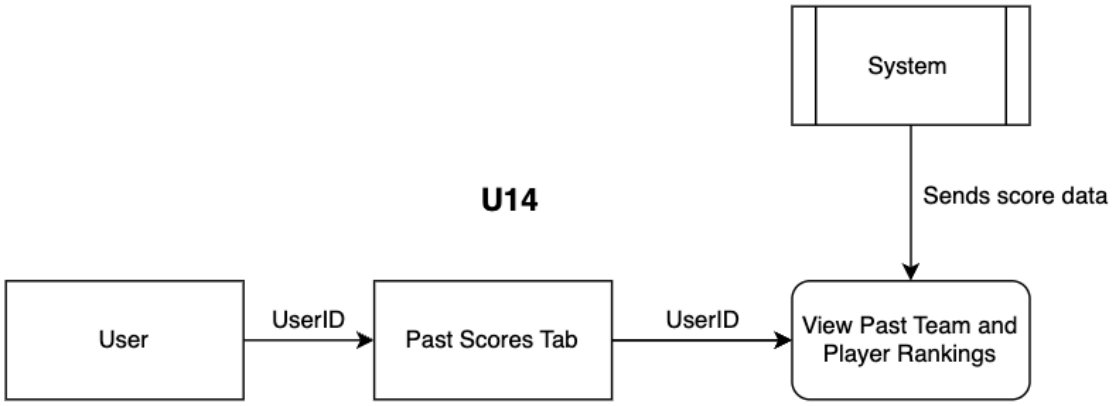
Preconditions: User has logged into the system

Trigger: Desire to view current or past rankings of individual players or teams

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks on “Past Scores” tab
- System: displays option to view past scoresheets or other Quiz Off information
- Actor: clicks “View Past Team and Player Rankings”
- System: displays tables of a summary for the years’ Quiz Off of the top number of scoring individuals and teams, as well as the rankings for each Quiz Off of the year, with the most recent at the top of the page and earlier years available to view as the user scrolls down

Conclusion: Past rankings for individuals and teams are displayed



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

View Statistics

Using User ID, request Score Data from system

Using Score Data and User ID, display Past Team and Player Ranking statistics

U15 – Practice Quiz Questions

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Use practice questions	U15	User	Desire to practice for events	Practice questions displayed as flashcards	Flashcard-styled questions	User

ID: U15

Name: Practice Quiz Questions

Primary Actor: User

Other Actors: N/A

Description: The user practices for an event with flash cards of preloaded questions of a chapter

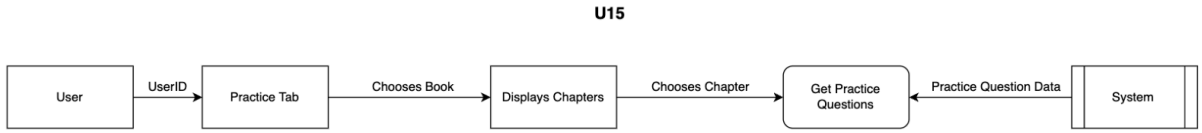
Preconditions: User has logged into the system

Trigger: User desires to practice for events

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Practice” tab
- System: displays practice page
- Actor: selects which book and chapter they wish to practice with
- System: displays flashcards of selected book and chapter
- Actor: clicks flashcard
- System: displays backside of flashcard revealing the answer
- Actor: clicks arrow under flashcard
- System: displays next/previous flashcard

Conclusion: User has practiced questions displayed as flashcards



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Practice Questions

Get Book and Chapter information from user

Using Book and Chapter information, request Practice Question Data from System

Using Practice Question Data, display Practice Questions

M1 – Edit Current Scoresheet

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Edit current scoresheet	M1	Quiz Master	Desire to add scores or edit mistakes on current scoresheet	Numbers change in scoresheet	Updated scoresheet	Quiz Master

ID: M1

Name: Edit Current Scoresheet

Primary Actor: Quiz Master

Other Actors: N/A

Description: The Quiz Master wants to add or adjust scores for a scoresheet during an active match at a Quiz Off event

Preconditions: User has logged into the system and has been granted Quiz Master permissions

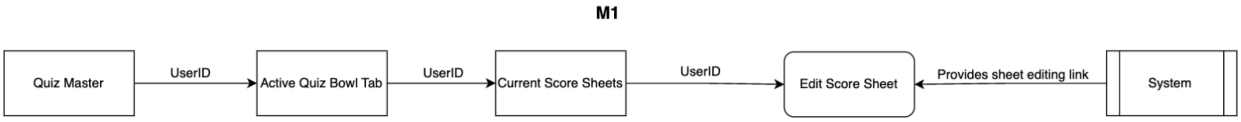
Trigger: Desire to add or edit scores in an active scoresheet

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks on the “Active Quiz Bowl” tab
- System: displays an option to view all current scoresheets, current player and team rankings, or schedule of matches
- Actor: clicks on “All Current Scoresheets”
- System: displays a list of all current scoresheets for competing teams at the Quiz Off
- Actor: clicks on scoresheet for the match they are scoring

- System: displays the current scoresheet and grants capacity for the user to edit it
- Actor: clicks an individual cell to type in scores, or to edit already present scores
- System: automatically saves input and recalculates team and individual totals

Conclusion: Scoresheet is updated with points assigned to teams and individuals



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Edit Current Scoresheet

Using User ID, get Match ID

Using User ID and Match ID, display scoresheet with matching Match ID

Store User ID and Match ID to Matches and User information

M2 – Display Current Leaderboard

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Display current leaderboard	M2	Quiz Master	Desire to display current leaderboard	Leaderboard projected for audience	Leaderboard display	Quiz Master

ID: M2

Name: Display Current Leaderboard

Primary Actor: Quiz Master

Other Actors: N/A

Description: The Quiz Master wants to display the current leaderboard to the audience.

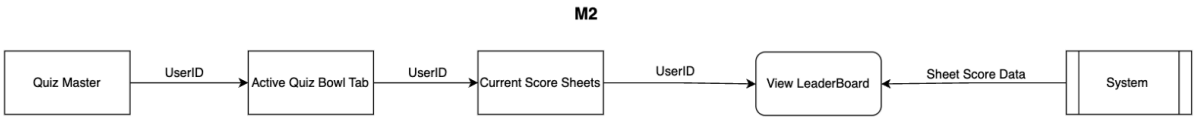
Preconditions: User has logged into the system

Trigger: Quiz Master desires to display current leaderboard

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Active Quiz Bowl” tab
- System: displays all current scoresheets
- Actor: selects the scoresheet for the match they wish to view the score of
- System: displays the selected scoresheet
- Actor: clicks “Leaderboard” button
- System: displays leaderboard and summary of scores
- Actor: clicks “Display” button
- System: display shown on external device

Conclusion: Current leaderboard is displayed to audience



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Display Current Leaderboard

Using User ID, request Score Sheet Data from System

Using Score Sheet Data, display Leaderboard

A1 – Verify Role

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Verify role	A1	Admin	Desire to assign role to system user	Role verified	Role added to profile	Database
					Notification	Admin

ID: A1

Name: Verify Role

Primary Actor: Admin

Other Actors: N/A

Description: An administrator receives a notification about adding a Quiz Master role to a user in the system and grants the appropriate permissions

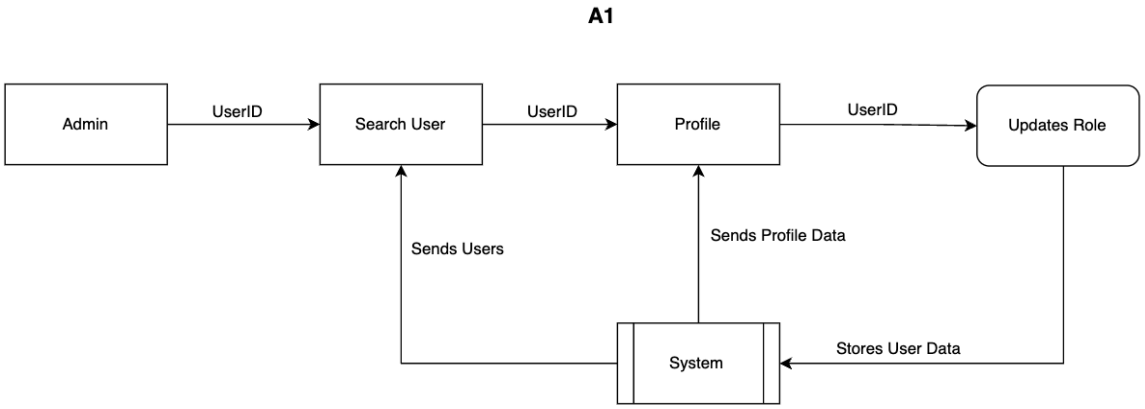
Preconditions: The requesting user has an account in the system.

Trigger: Desire to assign a role to a system user

Typical Event Flow:

- Actor: receives notification about a system user's request to add a Quiz Master Role
- Actor: verifies that the user is appropriate to receive the proper permissions
- Actor: clicks "Verify Role"
- System: adds Quiz Master role to the corresponding system user, and updates the user's permissions

Conclusion: Administrator has verified the role, and the user now has the appropriate role assigned.



Process Specification:

Get Permissions

Get admin's User ID

Using admin's User ID, get all Permissions

Get Role Info

Get Role ID

Using Role ID, get Role Info

Verify Role

If Role:

verify Role Instance

Notify admin

Send notification confirming verification

A2 – Create Round Robin

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Create round robin	A2	Admin	Desire to create round robin matches for teams	Teams assigned to round robin matches	Round robin schedule	Database
					Notification	Admin

ID: A2

Name: Create Round-Robin of Opponents

Primary Actor: Admin

Other Actors: N/A

Description: The admin creates the round-robin lineup schedule.

Preconditions: User has logged into the system

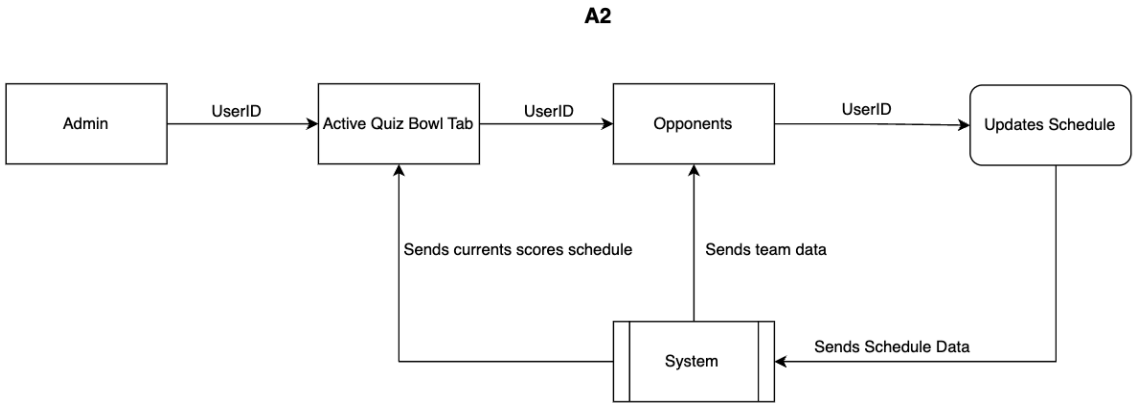
Trigger: Desire to create round robin matches for teams

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Active Quiz Bowl” tab
- System: displays “Active Quiz Bowl” page
- Actor: clicks “Opponents” tab
- System: ask for input of team pairs to play eachother
- Actor: enters pairs and clicks “Save”
- System: saves schedule to database
- Actor: receives notification about updated schedule

Conclusion: Round-robin schedule created, notification received

Postconditions: Current schedule of opponents and round robin lineup is created/updated to database.



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Update Schedule

Using User ID, get Opponents information

Using Opponents information, Update Schedule

Create Round Robin

Using Updated Schedule, send Schedule Data to System

Request Team Data and Current Score Schedules from System

Using Team Data and Current Score Schedules, display Round Robin Schedule

A3 – Add Quiz Off Event to Calendar

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Add quiz off event to calendar	A3	Admin	Desire to add Quiz Offs to calendar	Quiz Off event added to calendar	New event	Database
					Notification	Admin

ID: A3

Name: Add Quiz Off Event to Calendar

Primary Actor: Admin

Other Actors: N/A

Description: An administrator logs into the system and adds an event to the calendar, which allows all users to view it.

Preconditions: User has logged into the system and has administrative permissions

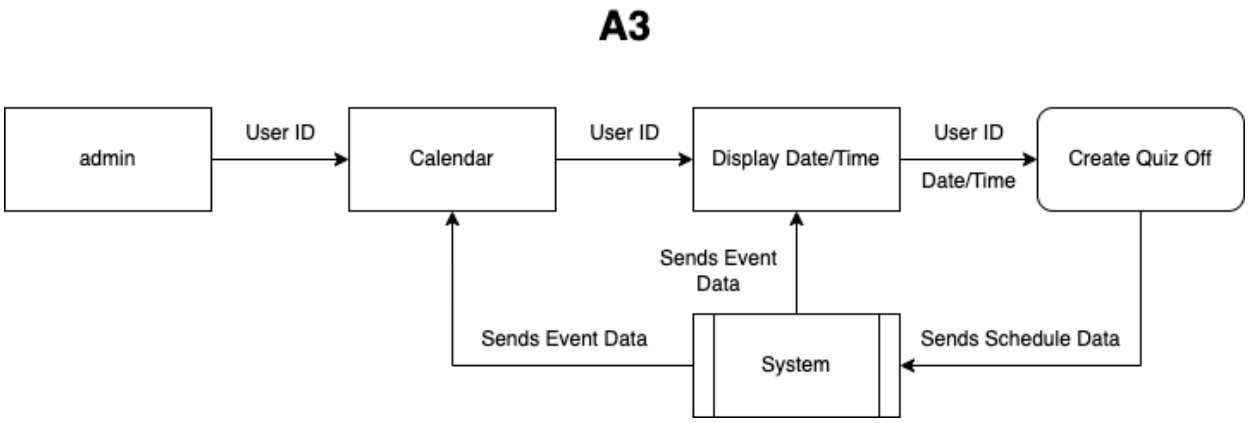
Trigger: desire to add a Quiz Off event to the calendar

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: scrolls down homepage and clicks “Calendar”
- System: expands calendar
- Actor: selects week, month, year they wish to view
- System: displays events of selected time frame on calendar
- Actor: selects day which they wish to add an event to
- System: provides form to input date, time, and name of event
- Actor: inputs required information and clicks “Add Event”

- System: calendar updates with the new event which allows all users to view it, and sends a notification to all users that a new event has been added to the calendar
- System: reloads week, month, and year page with the new event present

Conclusion: New event is added to the calendar



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Add Event

Get Event Information from User

Using Event Information, send Event Information to Database System

From Calendar, Request Event Information

Using Event Information, Display Calendar

A4 – Edit Quiz Off Event in Calendar

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Edit quiz off event in calendar	A4	Admin	Desire to edit mistakes in Quiz Off events in calendar	Quiz Off event edited	Updated event	Database
					Notification	Admin

ID: A4

Name: Edit Quiz Off Event in Calendar

Primary Actor: Admin

Other Actors: N/A

Description: An administrator logs into the system and edits an event in the calendar, allowing all users to view the updated information.

Preconditions: User has logged into the system and has administrative permissions

Trigger: Desire to edit mistakes in Quiz Off event in calendar

Typical Event Flow:

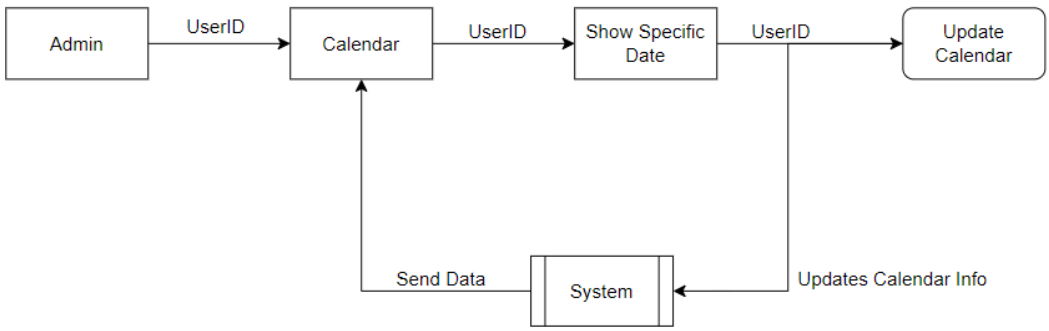
- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: scrolls down homepage and clicks “Calendar”
- System: expands calendar
- Actor: selects week, month, year they wish to view
- System: displays events of selected time frame on calendar
- Actor: selects event which they wish to make edits to
- System: provides form to change date, time, or name of event
- Actor: makes desired edits and clicks “Save”

- System: calendar updates with the changed event which allows all users to view it, and sends a notification to all users that an event has been changed in the calendar
- System: reloads week, month, and year page with the updated event

Conclusion: An already existing event in the calendar is updated to display the correct information.

Postconditions: The system returns to a steady state.

A4



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Edit Event

Get modified Event Information from User

Using Event Information, send Event Information to Database System

From Calendar, Request Event Information

Using Event Information, Display Calendar

A5 – Add Scoresheet to Database

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Add scoresheet to database	A5	Admin	Desire to add scoresheet to past scoresheet database	Scoresheet added to database	Scoresheet	Database

ID: A5

Name: Add Scoresheet to Database

Primary Actor: Admin

Other Actors: N/A

Description: Admin logs in to system and uploads a scoresheet file to save to database.

Preconditions: User has logged into the system

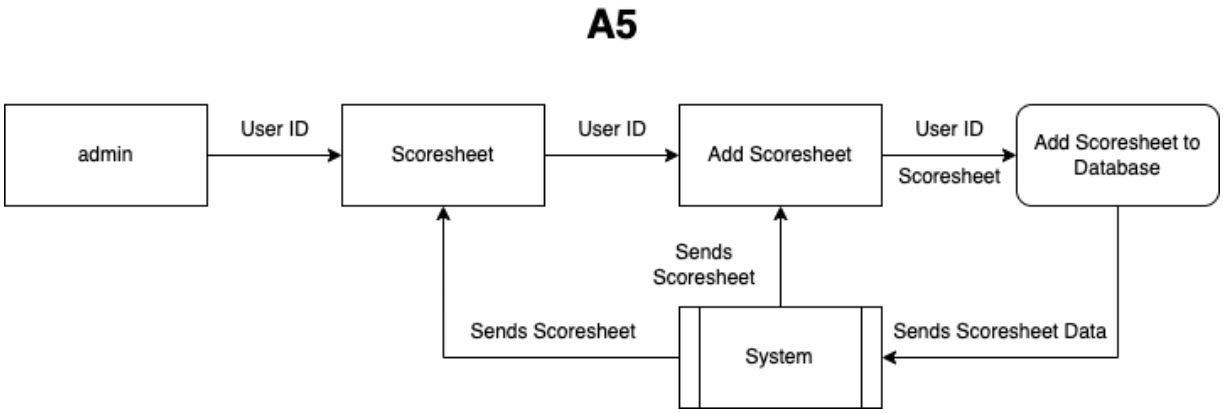
Trigger: Desire to add scoresheet to past scoresheet database

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Scores” tab
- System: displays all past scoresheets
- Actor: clicks “Add Scoresheet” button
- System: prompts to upload file
- Actor: Uploads file and completes other information – team, names, date, etc. and clicks “Save”
- System: updates and displays added scoresheet on page

Conclusion: Scoresheet is added to database

Postconditions: Database is updated with new scoresheets added



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Add Scoresheet

Get Scoresheet ID from User

Using Scoresheet ID, send Scoresheet Data to Database

Request Scoresheet Data from Database

Update Database

C1 – Create Team

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Create team	C1	Coach	Desire to create a new team	Team created	Team	Database

ID: C1

Name: Create Team

Primary Actor: Coach

Other Actors: N/A

Description: Coach will create a new team of players

Preconditions: User has logged into the system

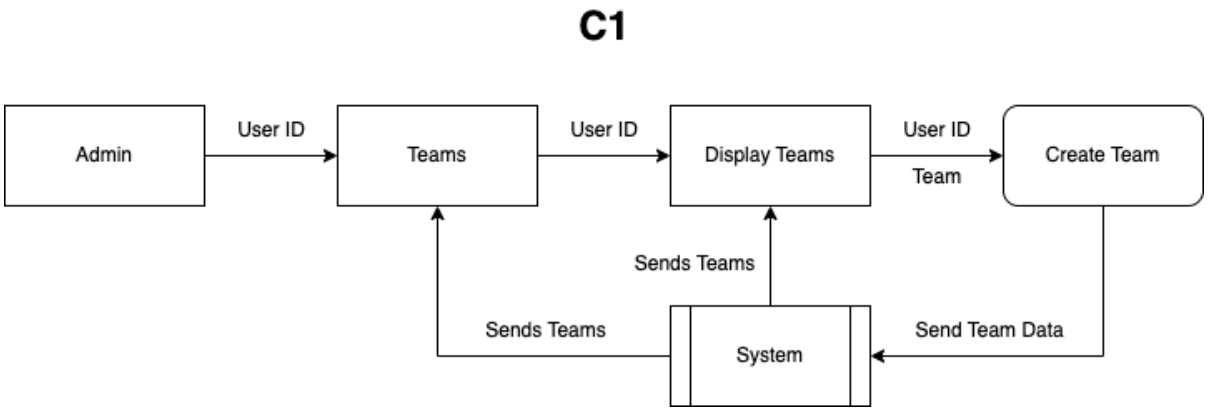
Trigger: Desire to create a new team

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks profile icon
- System: displays list of teams available
- Actor: clicks “Add a Team” button
- Actor: completes required information
- Actor: clicks “Save”
- System: adds and displays new team created on page

Conclusion: New team is created

Postconditions: New team saved and updated to database



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Create Team

Get Team Information from User

Using Team Information, create Team ID

Using Team ID, send Team Information to System

Request Team Information from System and display Teams

C2 – Add Player to Team

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Add player to team	C2	Coach	Desire to add player to a team	Player added to team	Team members	Database

ID: C2

Name: Add Players to Team

Primary Actor: Coach

Other Actors: N/A

Description: Coach adds new players to team

Preconditions: User has logged into the system

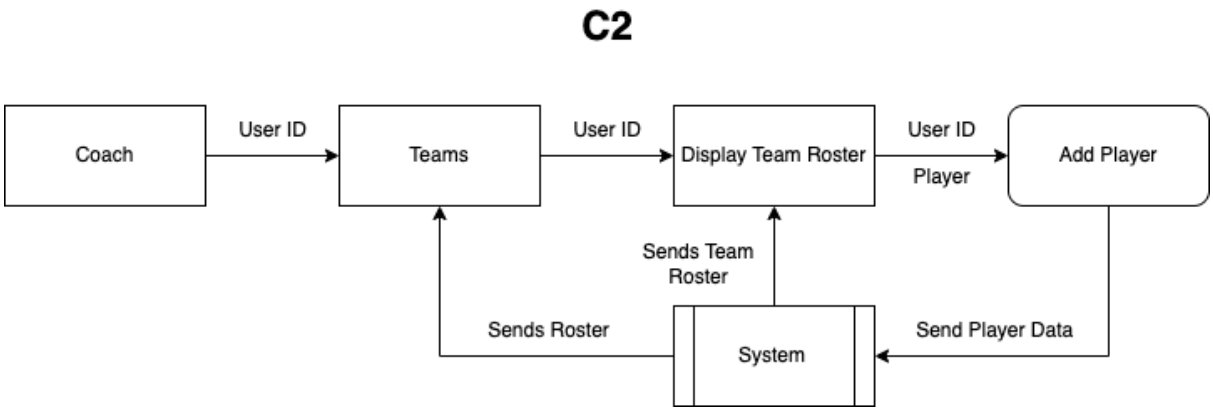
Trigger: Desire to add player to a team

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: click profile icon
- System: displays current teams created by coach
- Actor: clicks team they wish to add player to
- System: displays list of existing players
- Actor: clicks “Add” button next to player name or can search for a player’s name
- System: displays current team of players

Conclusion: New player has been added to a team

Postconditions: New player is added to team and updated in database



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Add Players to Team

Get Team ID from User

Using Team ID, get Player Information

Create Player ID

Using Player ID, match to Team ID

Send matched IDs to System

Request matched IDs from System, send to Roster

Update Roster in System

C3 – Remove Player from Team

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Remove player from team	C3	Coach	Desire to remove player from a team	Player removed from team	Team members	Database

ID: C3

Name: Remove Players from Team

Primary Actor: Coach

Other Actors: N/A

Description: Coach removes a player from a team

Preconditions: User has logged into the system

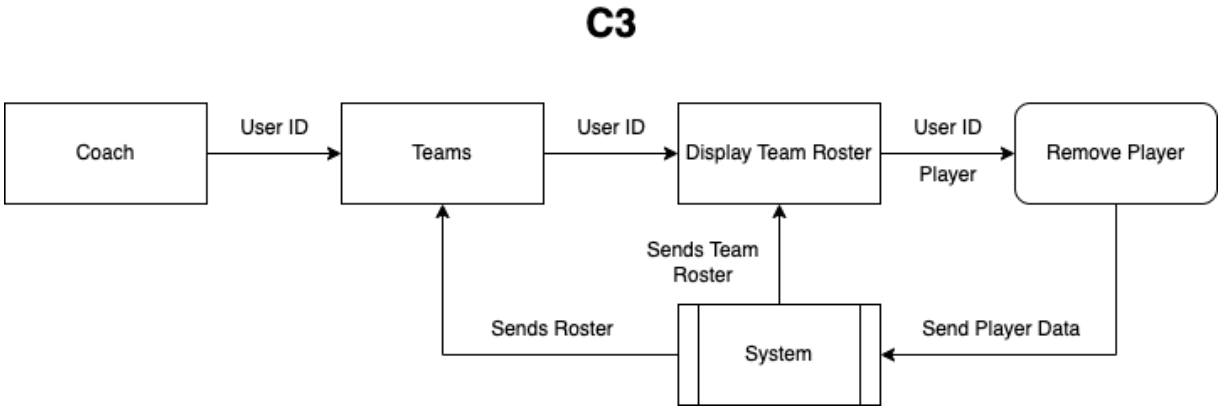
Trigger: Desire to remove player from a team

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: click profile icon
- System: displays current teams created by coach
- Actor: clicks team they wish to remove player from
- System: displays list of players in team
- Actor: clicks “Remove Player from Team” button next to player name or can search for a player’s name
- System: prompts with confirmation removal
- Actor: clicks “Confirm” button or “Deny” button
- System: displays current team of players

Conclusion: Player is removed from team

Postconditions: Player is removed from team and updated in database



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Remove Players from Team

Get Team ID and/or Player ID from User

Remove link between Player ID and Team ID in System

Remove Player ID from Roster

Update Roster in System

C4 – Add Team Event to Calendar

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Add team event to calendar	C4	Coach	Desire to add team event to calendar	Team event added to calendar	New event	Database
					Notification	Coach

ID: C4

Name: Add Team Event to Calendar

Primary Actor: Coach

Other Actors: N/A

Description: Event is added to calendar

Preconditions: User has logged into the system

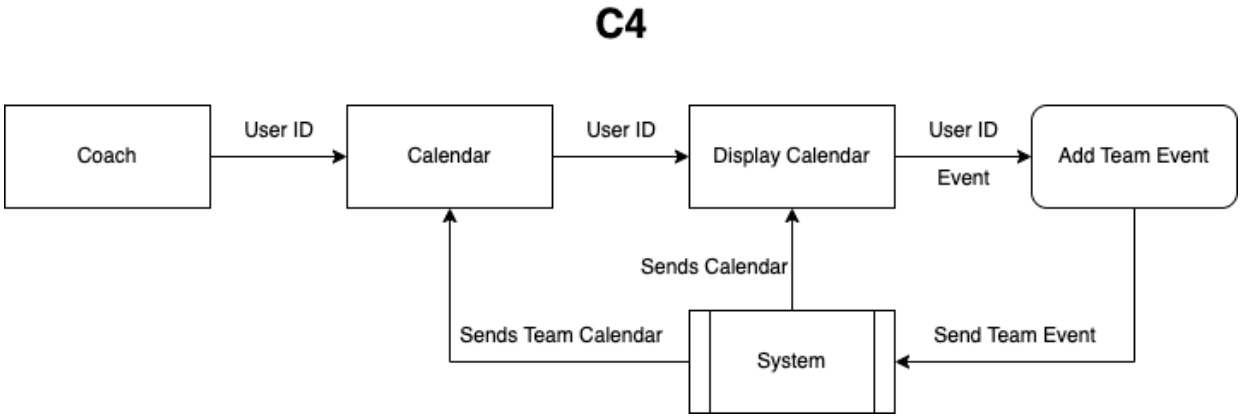
Trigger:

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Calendar” tab
- System: displays current calendar with events
- Actor: clicks “Add Event” button
- System: prompts menu to enter date, time, name of event, team, submit button
- Actor: completes required fields and clicks “Submit”
- Actor: receives notification that events have been successfully added
- System: adds event, updates and displays current calendar

Conclusion: New team event is added to calendar, notification received

Postconditions: New event is added and stored in database



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Add Event

Get Event Information from User

Using Event Information, send Event Information to Database System

From Calendar, Request Event Information

Using Event Information, Display Calendar

C5 – Edit Team Event in Calendar

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Edit team event in calendar	C5	Coach	Desire to edit mistakes or dates in team events already in calendar	Team even edited	Updated event	Database
					Notification	Coach

ID: C5

Name: Edit Team Event in Calendar

Primary Actor: Coach

Other Actors: N/A

Description: The coach logs into the system and edits an existing event in the team calendar.

Preconditions: User has logged into the system

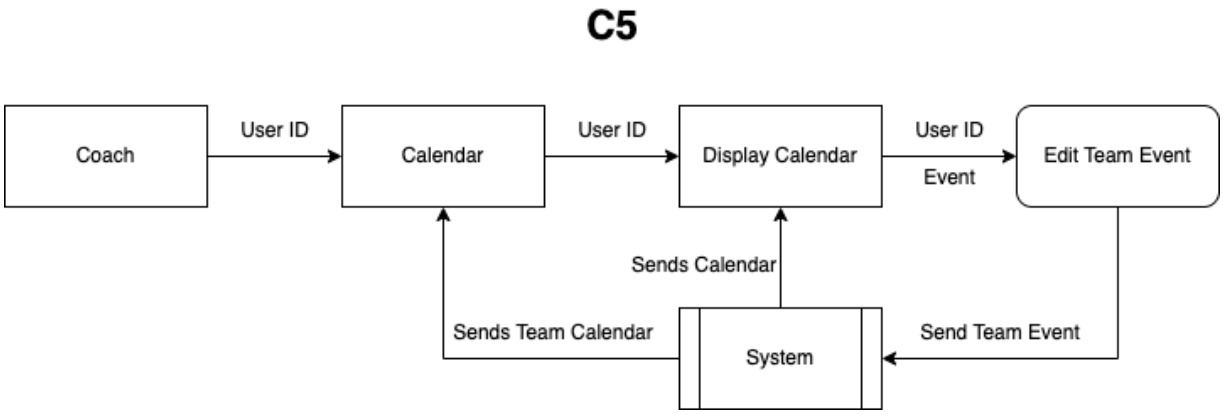
Trigger: Desire to edit mistakes or dates already in the team calendar.

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Calendar” tab
- System: displays current calendar with events
- Actor: clicks on an existing event
- System: prompts menu to edit date, time, name of event, team, submit button
- Actor: edits current information and clicks “Save”
- Actor: receives notification events have been successfully edited and saved

- System: updates event details for all team users to view and displays current calendar

Conclusion: Team event is updated and displays for all users in the team, notifications received



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Edit Event

Get modified Event Information from User

Using Event Information, send Event Information to Database System

From Calendar, Request Event Information

Using Event Information, Display Calendar

C6 – View Chat

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
View chat	C6	Coach, Quizzer	Desire to view chat messages	Chat displayed	Chat	User

ID: C6

Name: View Chat

Primary Actor: Coach

Other Actors: Quizzer

Description: the user logs into the system and views chat messages sent for their team.

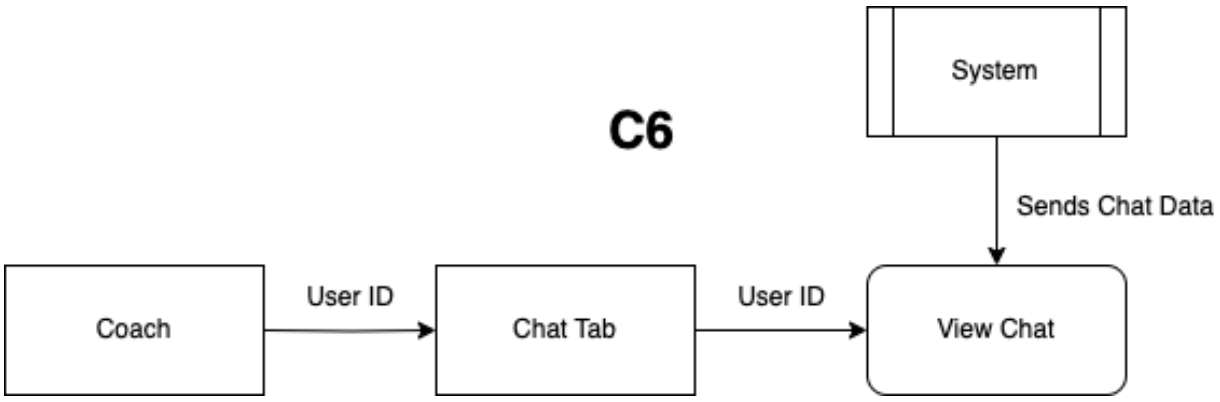
Preconditions: User has logged into the system and is a team member

Trigger: Desire to view chat messages.

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Chat”
- System: displays most recent chat message, as well as the user that sent them, with previous messages above it

Conclusion: Chat messages are displayed.



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

View Chat

Using User ID, display Chat

C7 – Send Message to Chat

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Send message to chat	C7	Coach, Quizzer	Desire to send a message to someone in the chat	New message in chat sent	Message	Database

ID: C7

Name: Send Message in Chat

Primary Actor: Coach

Other Actors: Quizzer

Description: the user logs into the system and sends a message in the chat for other team members to view.

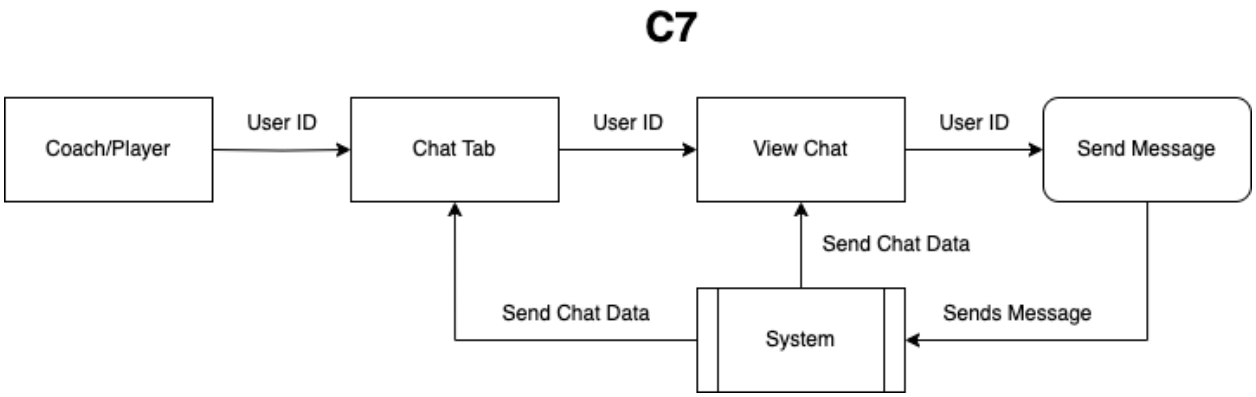
Preconditions: User has logged into the system and is a team member

Trigger: Desire to send a message to someone in the chat

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Chat”
- System: displays most recent message with previous messages above it
- Actor: clicks “Add message”
- System: displays form field which allows text to be typed, or files and images to be uploaded
- Actor: adds text, file, and/or image and clicks “Send message”
- System: returns actor to viewing chat
- System: updates running list of chat with the new information, which allows all users that are a part of the team to view it

Conclusion: New message is sent in the chat and available for team members to view.



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Send Message

Get Message Information from User

Send Message Information to System

Request Message Information from System and display

Notification

Send Message notification to other recipient

C8 – Send Announcement

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Send announcement	C8	Coach, Quiz Master, or Admin	Desire to send announcement to all users	New messages in announcements sent	Message	Database

ID: C8

Name: Send Announcement

Primary Actor: Coach

Other Actors: Admin, Quiz Master

Description: the user logs into the system and sends an announcement to all users

Preconditions: User has logged into the system and has the appropriate permissions

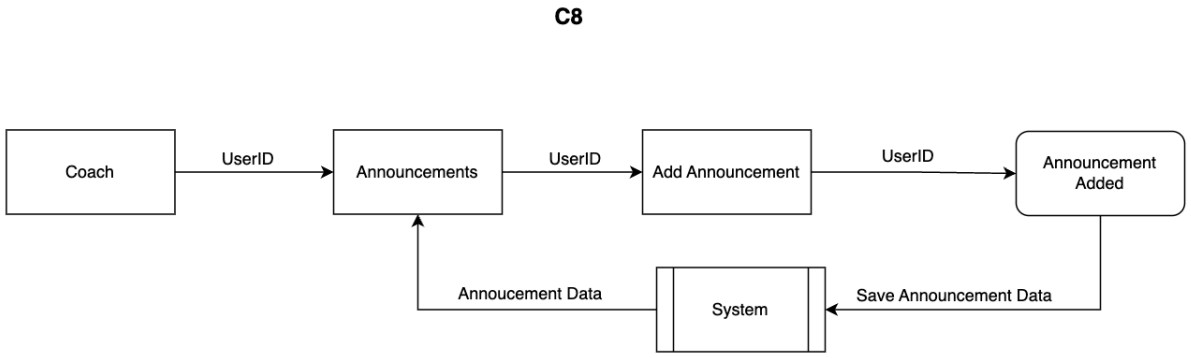
Trigger: Desire to send announcement to all users

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Announcements”
- System: displays most recent announcements with previous announcements above it, as well as an option to add an announcement
- Actor: clicks “Add announcement”
- System: displays form field which allows text to be typed, or files and images to be uploaded
- Actor: adds text, file, and/or image and clicks “Send announcement”
- System: returns actor to viewing announcements

- System: updates running list of announcements with the new information, which allows all users to view it

Conclusion: New message sent in announcements



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Add Announcement

Get Announcement Information from User

Send Announcement Information to System

Request Announcement Information from System and display

Notification

Send confirmation notification to User

Send announcement notification to other Users

C9 – Upload Practice Questions

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Upload practice questions	C9	Coach or Admin	Desire to add practice questions	New practice questions added	Practice questions	Database

ID: C9

Name: Upload Practice Questions

Primary Actor: Admin

Other Actors: Coach

Description: Admin or Coach adds practice questions to resources

Preconditions: User has logged into the system

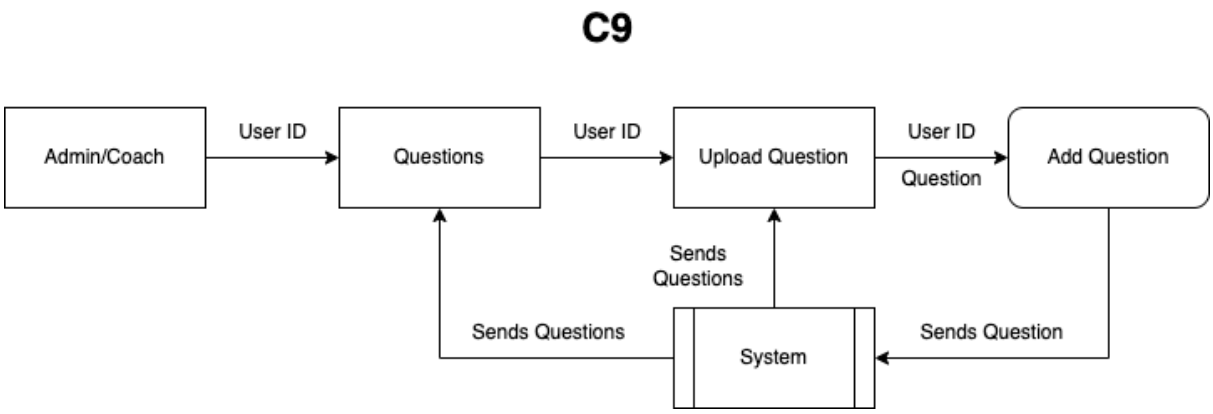
Trigger: Desire to add practice questions

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Practice” tab
- System: displays current resources
- Actor: selects which book and chapter corresponding to the question
- System: displays list of existing questions
- Actor: clicks “Add Question” button
- Actor: completes information fields
- Actor: clicks “Save”
- System: adds and displays new question with list of other questions

Conclusion: New practice questions are added

Postconditions: New practice questions are added and saved to database



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Upload Practice Questions

Get Practice Question Information from User

Using Practice Question Information, send information to Database

Save and update Database

C10 – Edit Practice Questions

Event Title	Event ID	Source	Trigger	Event Response	Major Output	External Destination
Edit practice questions	C10	Coach or Admin	Desire to edit mistakes in existing practice questions	Existing practice questions edited	Updated practice questions	Database

ID: C10

Name: Edit Practice Questions

Primary Actor: Admin

Other Actors: Coach

Description: Admin or Coach edits already existing practice questions

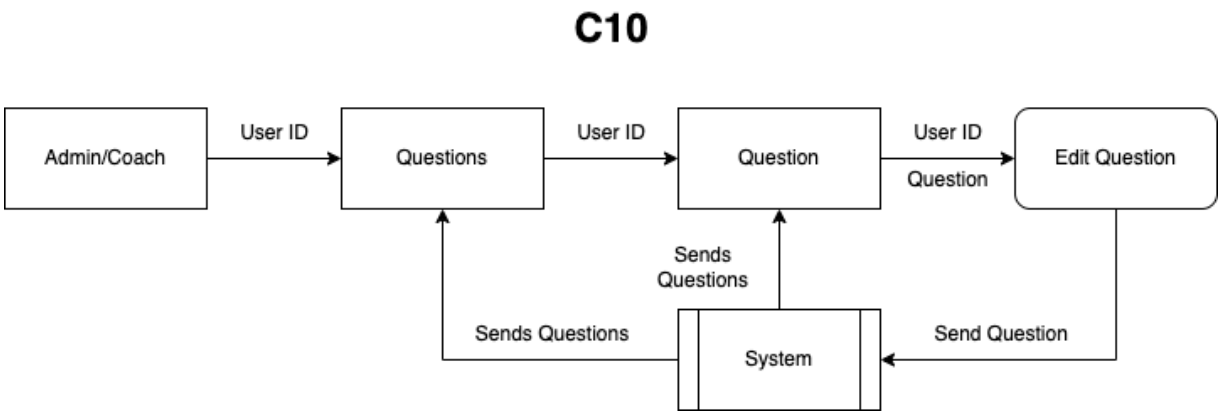
Preconditions: User has logged into the system

Trigger: Desire to edit mistakes in existing practice questions

Typical Event Flow:

- Actor: types in username and password on login page
- Actor: clicks “Login”
- System: allows access to user’s personal account and displays homepage
- Actor: clicks “Practice” tab
- System: displays current resources
- Actor: selects which book and chapter corresponding to the question
- System: displays list of existing questions
- Actor: finds and selects question they wish to edit
- Actor: edits question text and clicks “Save”
- System: saves and displays the updated question with list of other questions

Conclusion: Updated question is saved and available to practice



Process Specification:

Identify User

Get Credentials

Using Credentials, get User ID

Log user into the system

Using User ID, create a cookie/session on the Client/Server

Edit Practice Questions

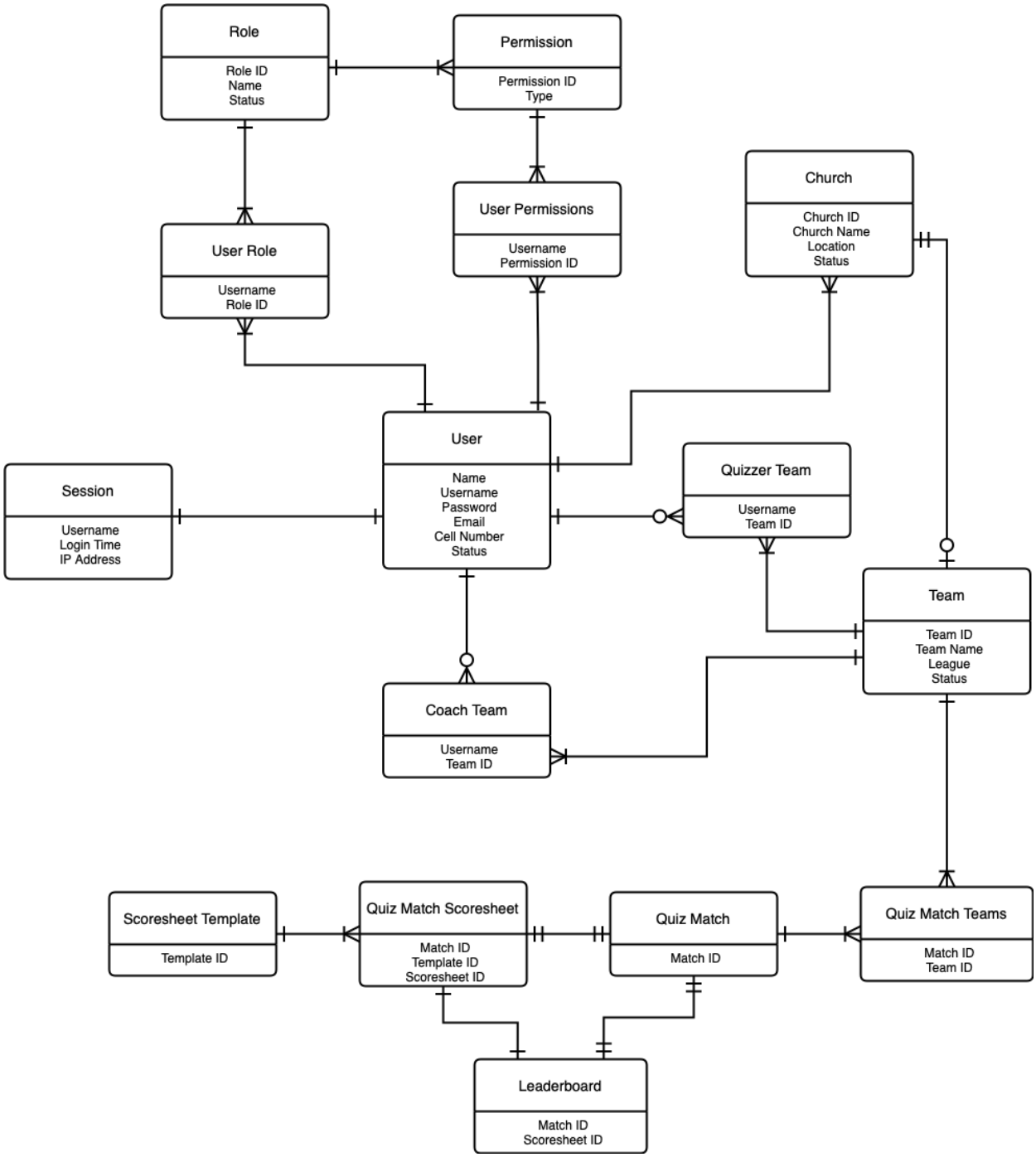
Get Edited Practice Question Information from User

Using Edited Practice Question Information, send new information to Database

Save and update Database

2.3. Entity Relationship Diagram

The Entity Relationship Diagram (ERD) describes the database that supports the system. Each table contains an entity with its associated attributes. The tables are connected by “keys” that are unique to that table and may be connected to zero, one, or many other entities.



2.4. Data Dictionary

The data dictionary defines terms and attributes (as well as connections) seen in the Entity Relationship Diagram or logical data model

User – Someone who wishes to have an account in the system

Relationships:

- Has a one-to-many relationship with User Roles
- Has a one-to-many relationship with User Permissions
- Has a one-to-optional-many with Quizzer Team
- Has a one-to-optional-many with Coach Team
- Has a one-to-one relationship with Session
- Has a one-to-optional-many relationship with Church
- Has a one-to-optional-many relationship with Church

Attributes:

- Name- the name of the account holder
- Username- a unique identifier chosen for the user
- Password- a secure combination of characters only known to the user
- Email- the email address of the user
- Cell Number- the cell phone number of the user
- Status- describes if the user's account is active or not

Role – A method of assigning the same permissions to multiple users

Relationships:

- Has a one-to-many relationship with User Role
- Has a one-to-many relationship with Permissions

Attributes:

- Role ID- the primary key for Role entity
- Name- the name of the role type (admin, coach, etc.)
- Status- describes if the user's account is active or not

User Role – A link between a user and their role through their username

Relationships:

- Has a many-to-one relationship with User
- Has a many-to-one relationship with Role

Attributes:

- Username- a unique identifier chosen for the user
- Role ID- the primary key for Role entity

Permission – The level of access granted to a user through their user role

Relationships:

- Has a many-to-one relationship with Role
- Has a one-to-many relationship with User Permissions

Attributes:

- Permission ID- the primary key for the Permission entity
- Type- a description of the permissions

User Permissions – A link between a user and their permissions through their username

Relationships:

- Has a many-to-one relationship with Permissions
- Has a many-to-one relationship with User

Attributes:

- Username- a unique identifier chosen for the user
- Permission ID- the primary key for the Permission entity

Session – Active time and activity of a user in the system

Relationships:

- Has a one-to-one relationship with User

Attributes:

- Username- a unique identifier chosen for the user
- Login time- the time of login for the current user
- IP address- the IP address of the computer/device that the user is currently logged in on

Team – A list of all teams available for users to join

Relationships:

- Has a one-to-many relationship with Coach Team
- Has a one-to-many relationship with Quizzer Team
- Has an optional-many-to-one relationship with Church
- Has a one-to-many relationship with Quiz Match Teams

Attributes:

- Team ID- the primary key for the Team entity
- Team Name- the name of the team
- League- can be Jr. High or High School

- Status- describes if the user account is active or not

Quizzer Team – A link between a user and their team through their username

Relationships:

- Has a many-to-one relationship with Team
- Has a optional-many-to-one relationship with User

Attributes:

- Username- a unique identifier chosen for the user
- Team ID- the primary key for the Team entity

Coach Team – A link between a coach and their team through their username

Relationships:

- Has a many-to-one relationship with Team
- Has an optional-many-to-one relationship with User

Attributes:

- Username- a unique identifier chosen for the user
- Team ID- the primary key for the Team entity

Church – A list of all participating churches in the quiz bowl

Relationships:

- Has a many-to-one relationship with Team
- Has an optional-many-to-one relationship with User

Attributes:

- Church ID- the primary key for the Church entity
- Church Name- the name of the Church
- Location- the city/address of the church
- Status- describes if the church is active or not

Quiz Match Teams – A list of all teams involved in a given match

Relationships:

- Has a many-to-one relationship with Team
- Has a many-to-one relationship with Quiz Match

Attributes:

- Match ID- the primary key for the Quiz Match entity
- Team ID- the primary key for the Team entity

Quiz Match – A given match in the quiz bowl

Relationships:

- Has a one-to-one relationship with Quiz Match Scoresheet
- Has a one-to-one relationship with Leaderboard
- Has a one-to-many relationship with Quiz Match Teams

Attributes:

- Match ID- the primary key for the Quiz Match entity

Scoresheet Template – A template to create a new Quiz Match Scoresheet

Relationships:

- Has a one-to-one relationship with Quiz Match Scoresheet
- Has a one-to-one relationship with Leaderboard
- Has a one-to-many relationship with Quiz Match Teams

Attributes:

- Template ID- the primary key for the Scoresheet Template entity

Quiz Match Scoresheet – A sheet to track the scores of the Quiz Match

Relationships:

- Has a one-to-one relationship with Quiz Match
- Has a many-to-one relationship with Scoresheet Template
- Has a one-to-one relationship with Leaderboard

Attributes:

- Match ID- the primary key for the Quiz Match entity
- Template ID- the primary key for the Scoresheet Template entity
- Scoresheet ID- the primary key for the Quiz Match Scoresheet entity

Leaderboard – A link between a Quiz Match Scoresheet and a Quiz Match through the Scoresheet ID

Relationships:

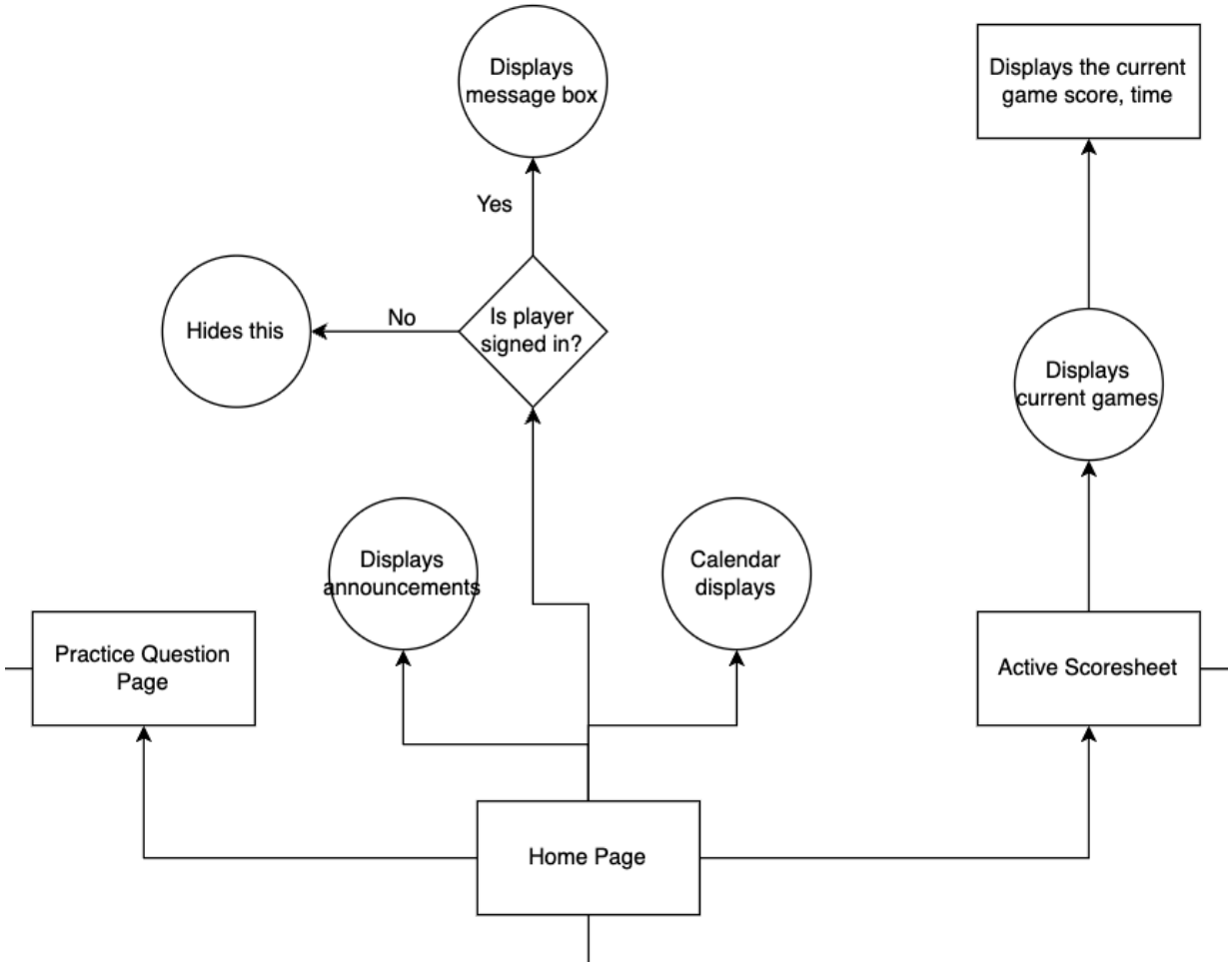
- Has a one-to-one relationship with Quiz Match Scoresheet
- Has a one-to-one relationship with Quiz Match

Attributes:

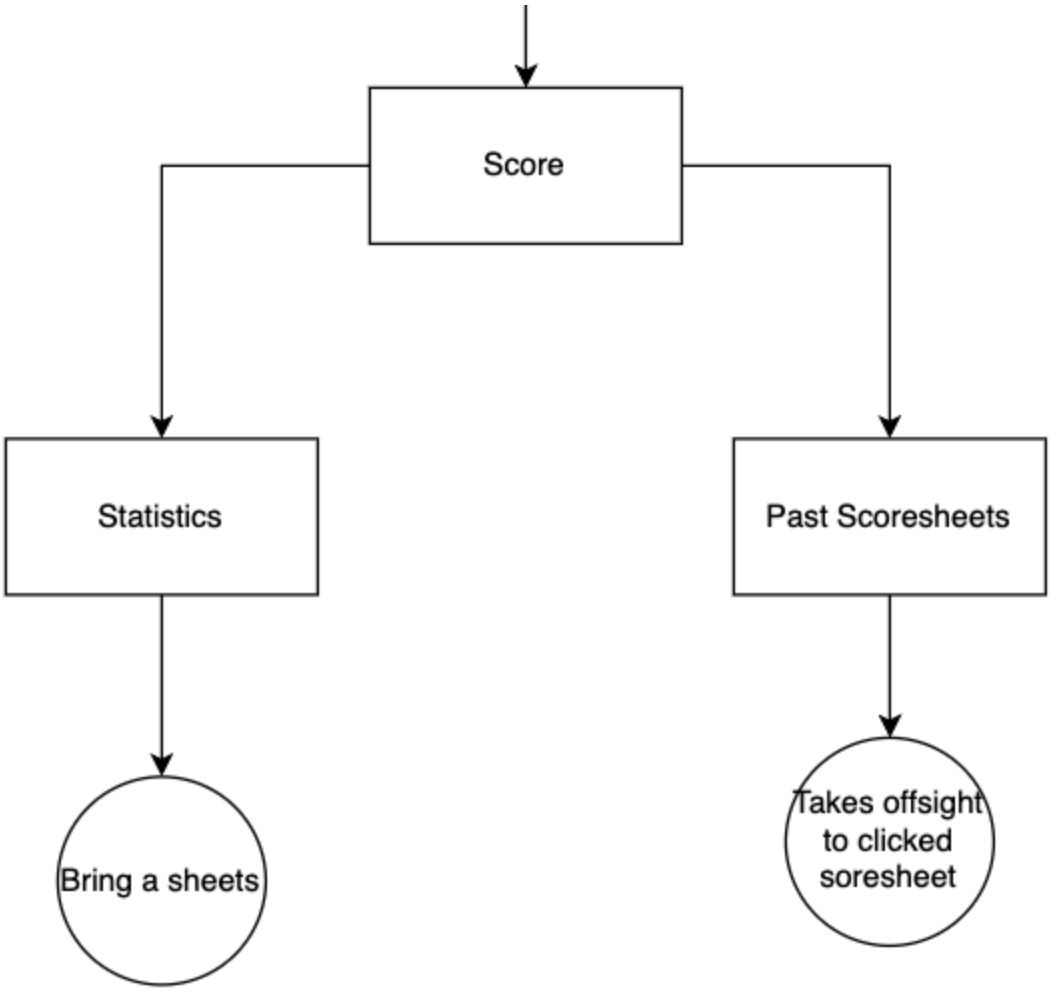
- Match ID- the primary key for the Quiz Match entity
- Scoresheet ID- the primary key for the Quiz Match Scoresheet entity

2.5. User Experience Diagrams

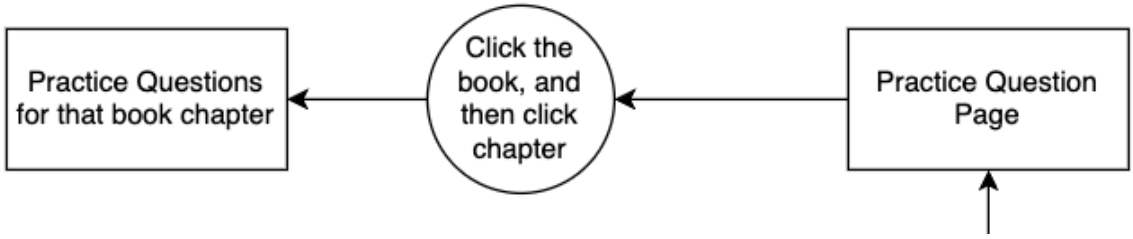
HOME PAGE:



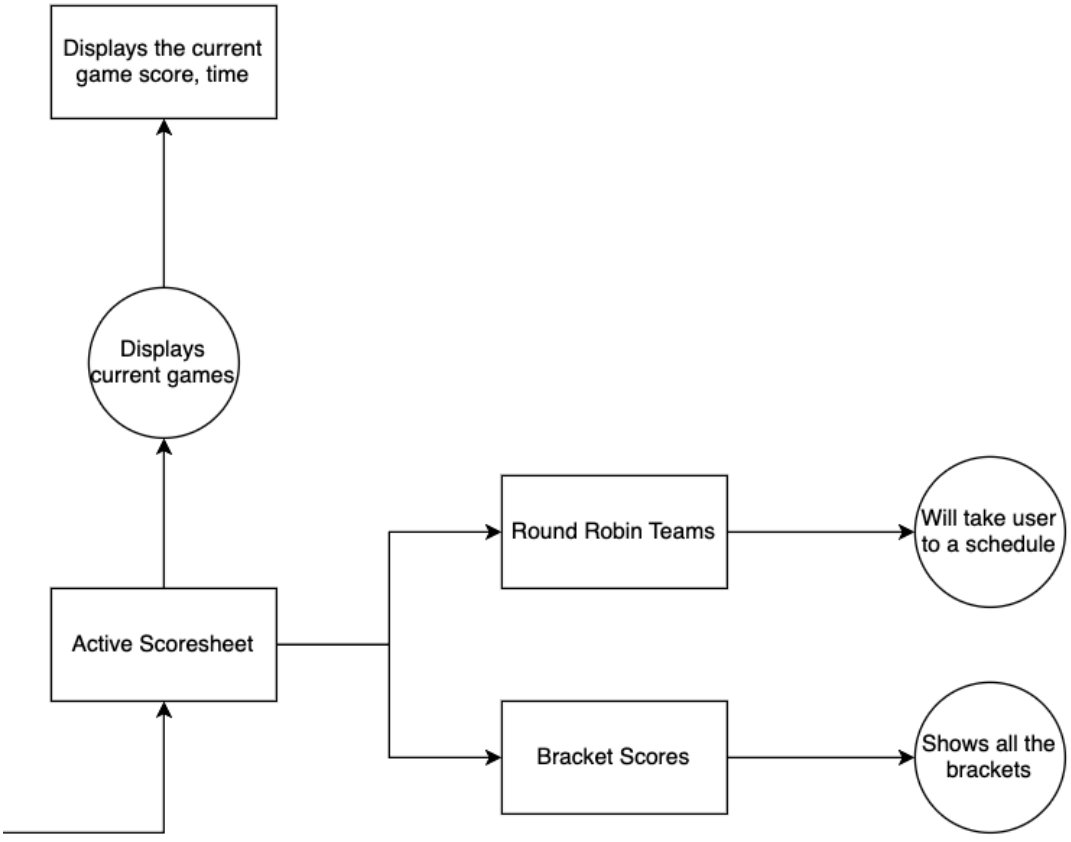
Score Page


Score
Statistics
Past Scoresheets
Bring a sheets
Takes offsight to clicked soresheet

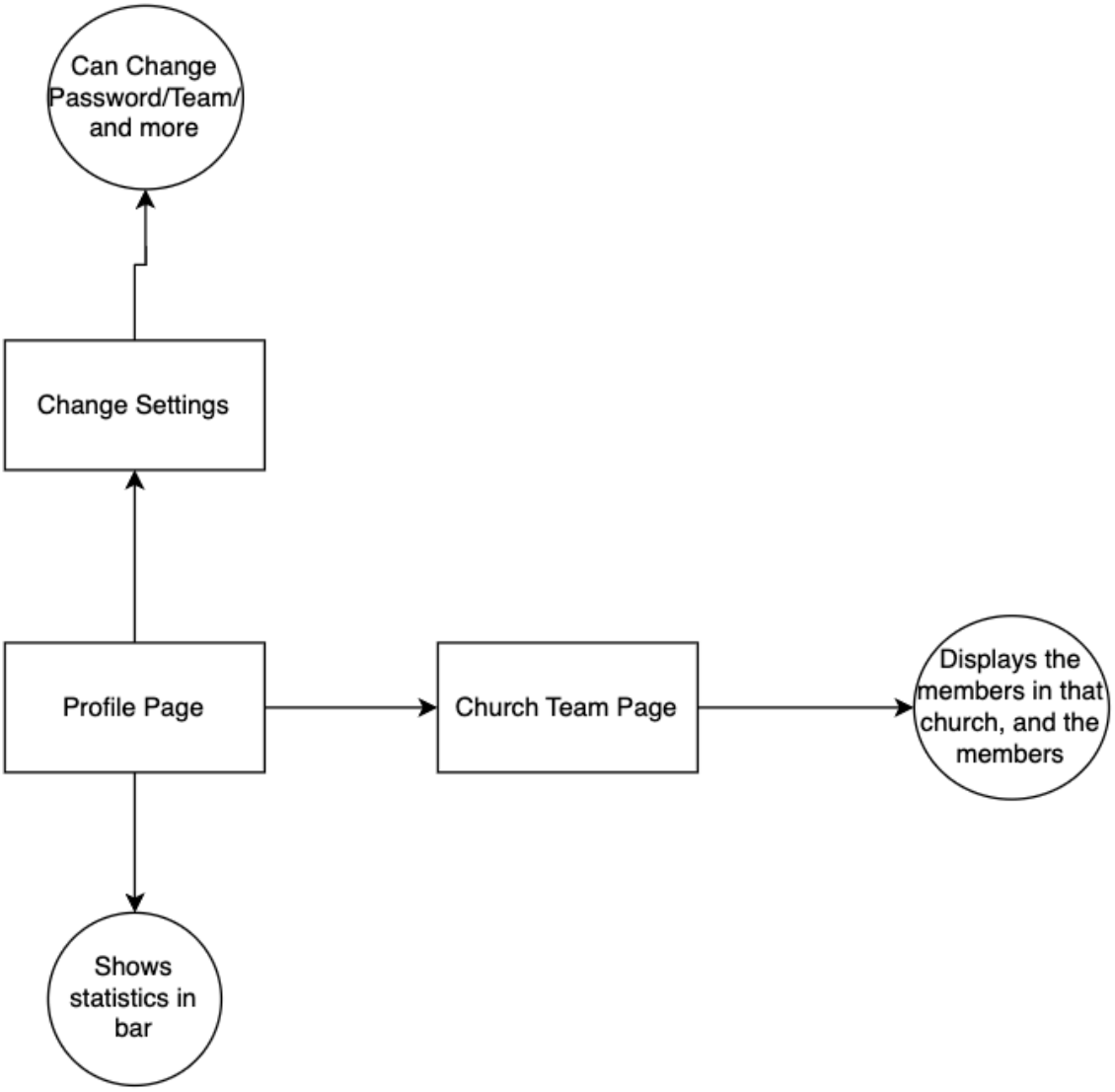
Practice Question Page



Active Scoresheet Page

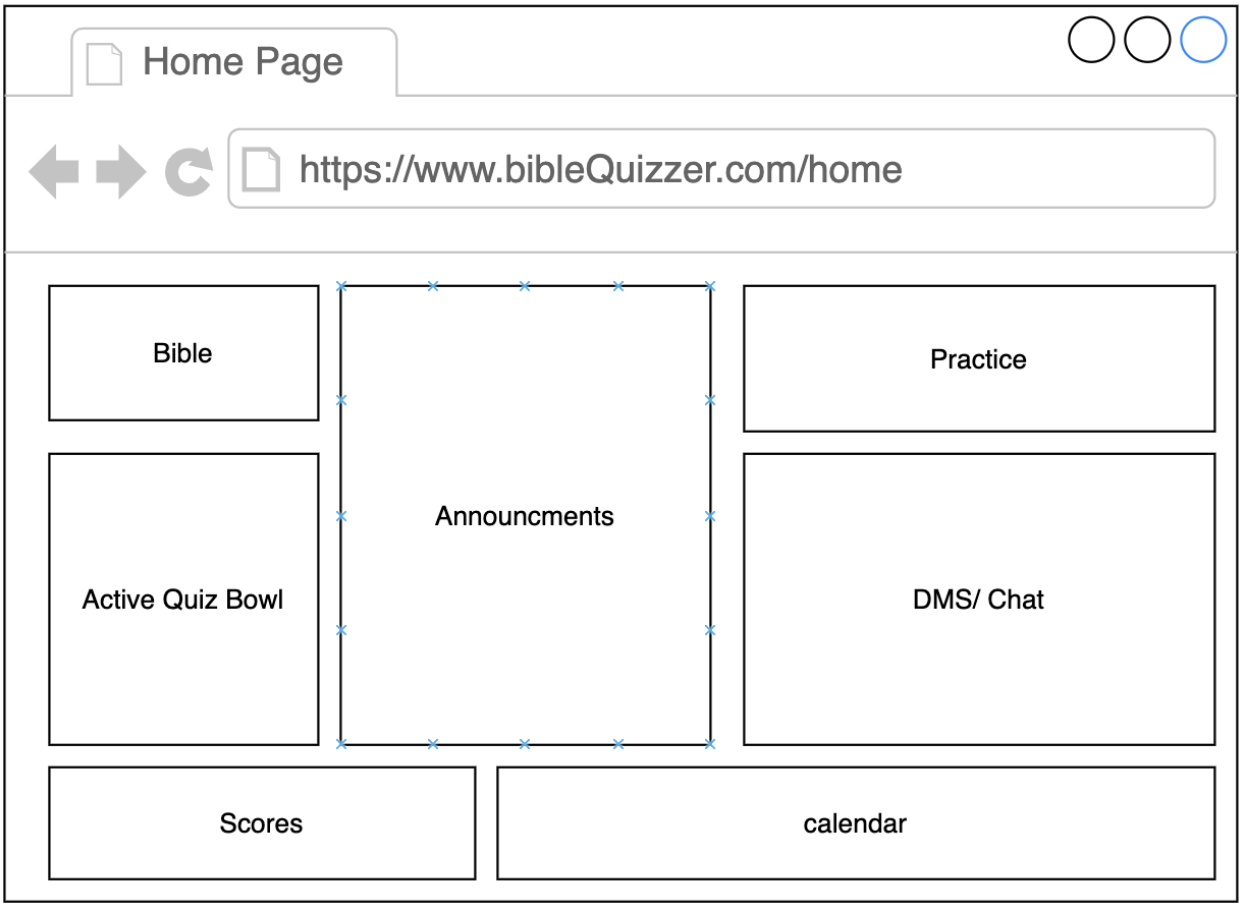


Profile Page

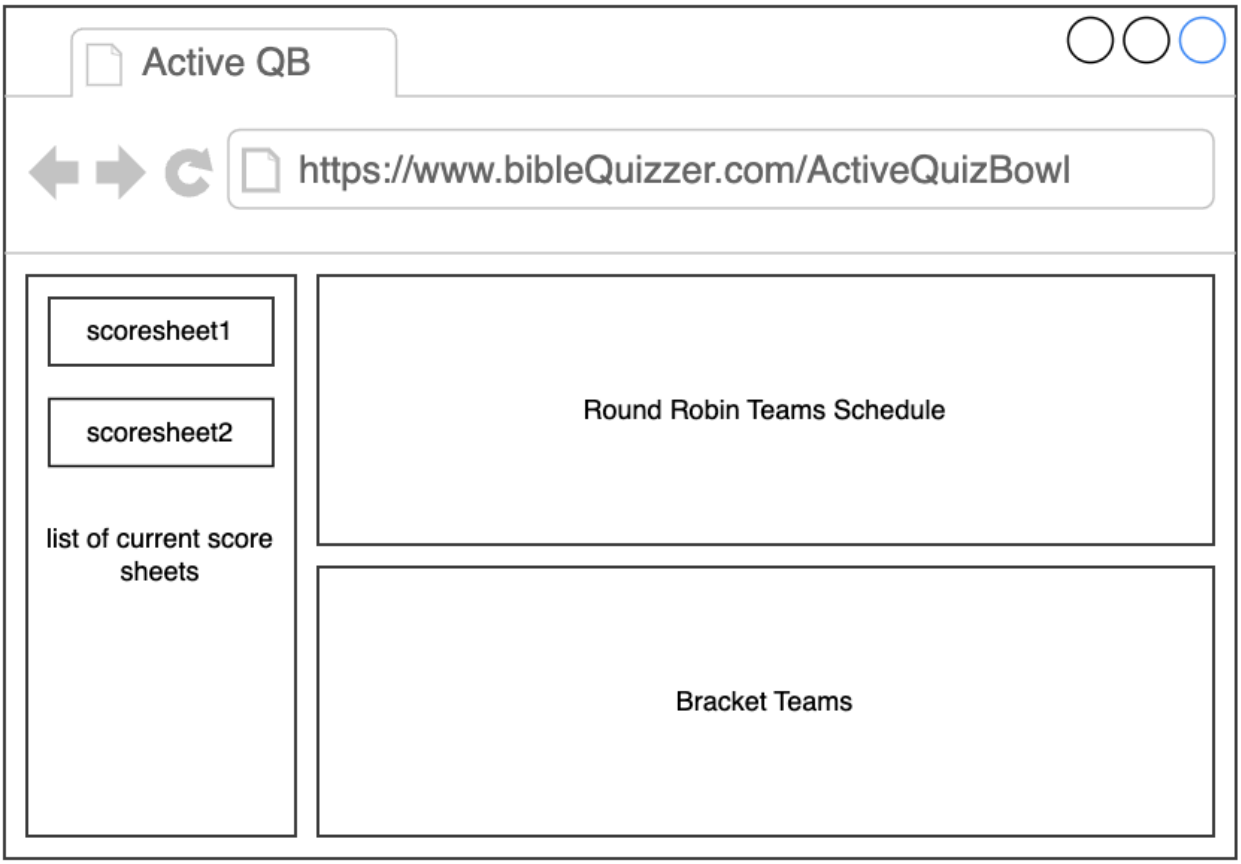


2.6. Wireframes

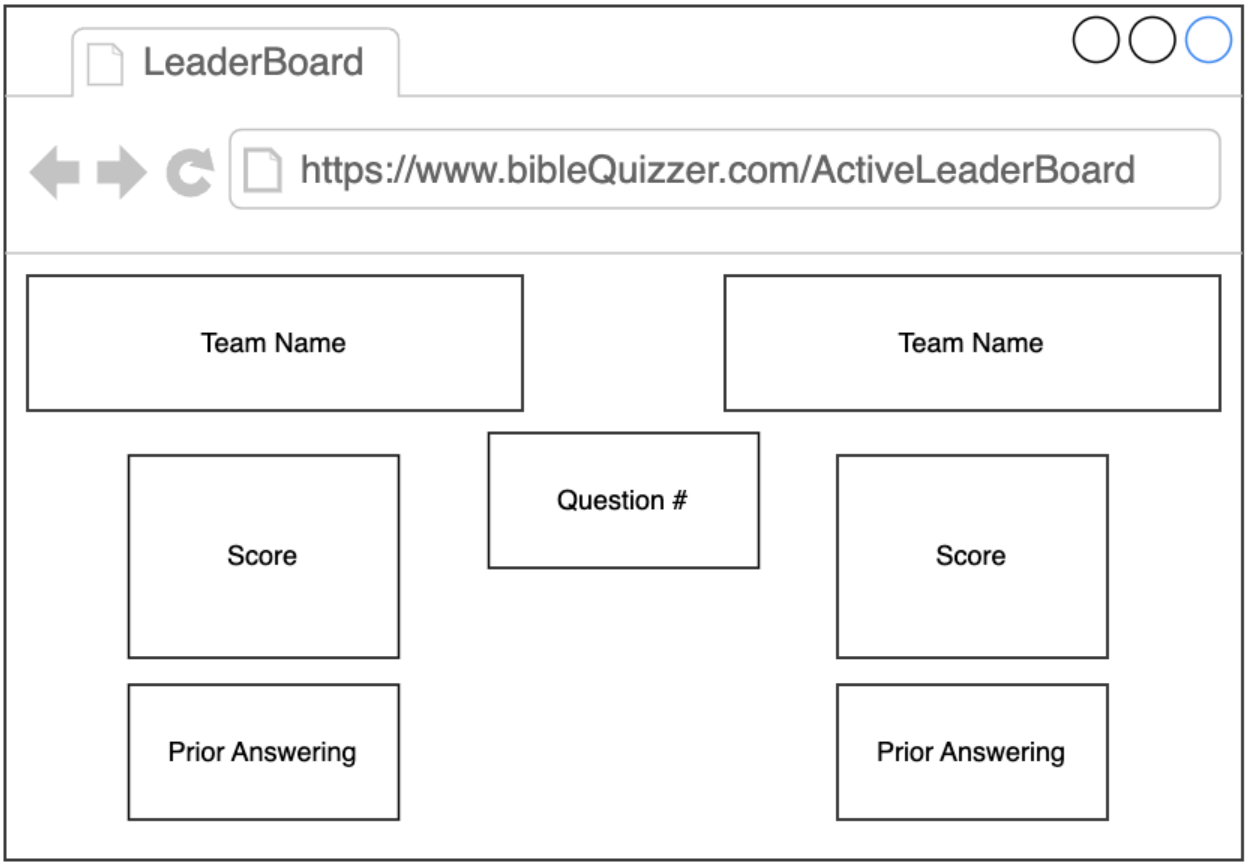
Home Page:



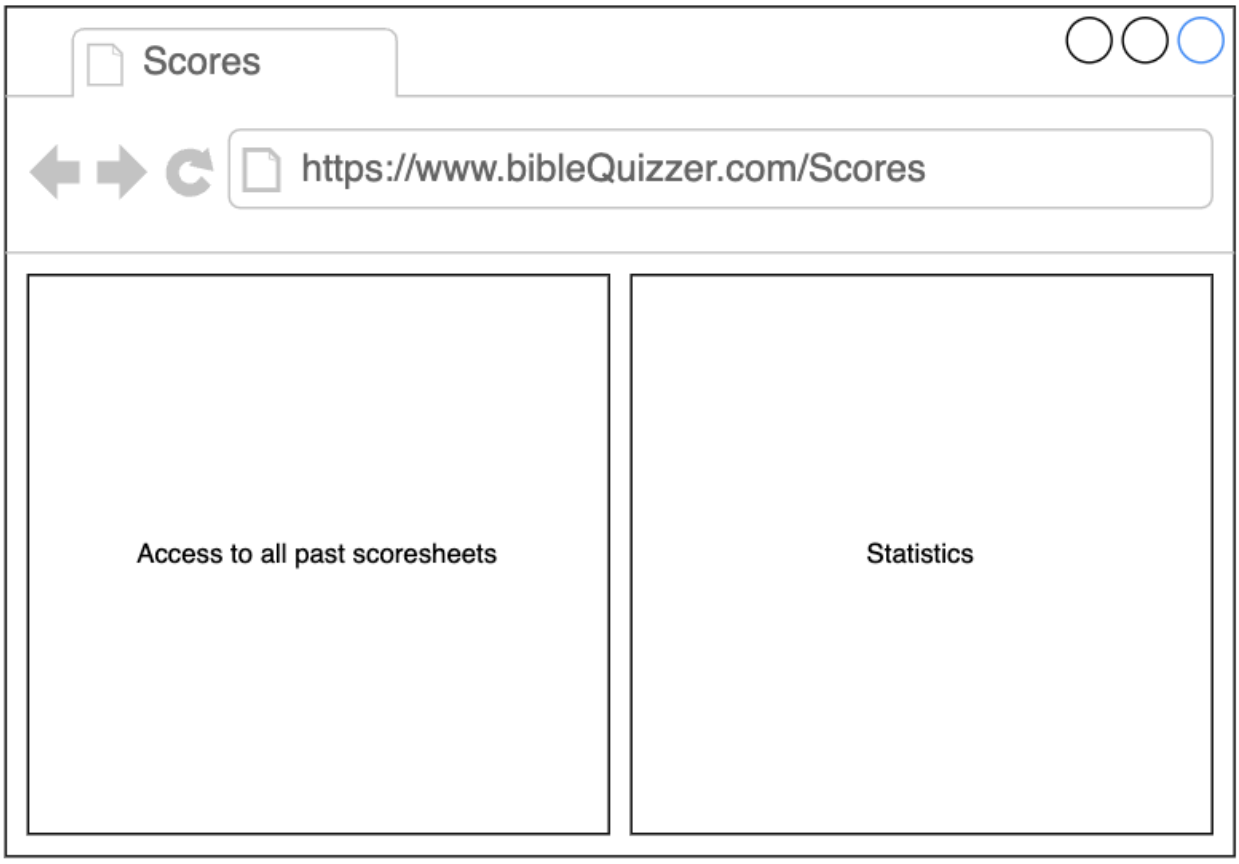
Active Quiz Bowl Page:



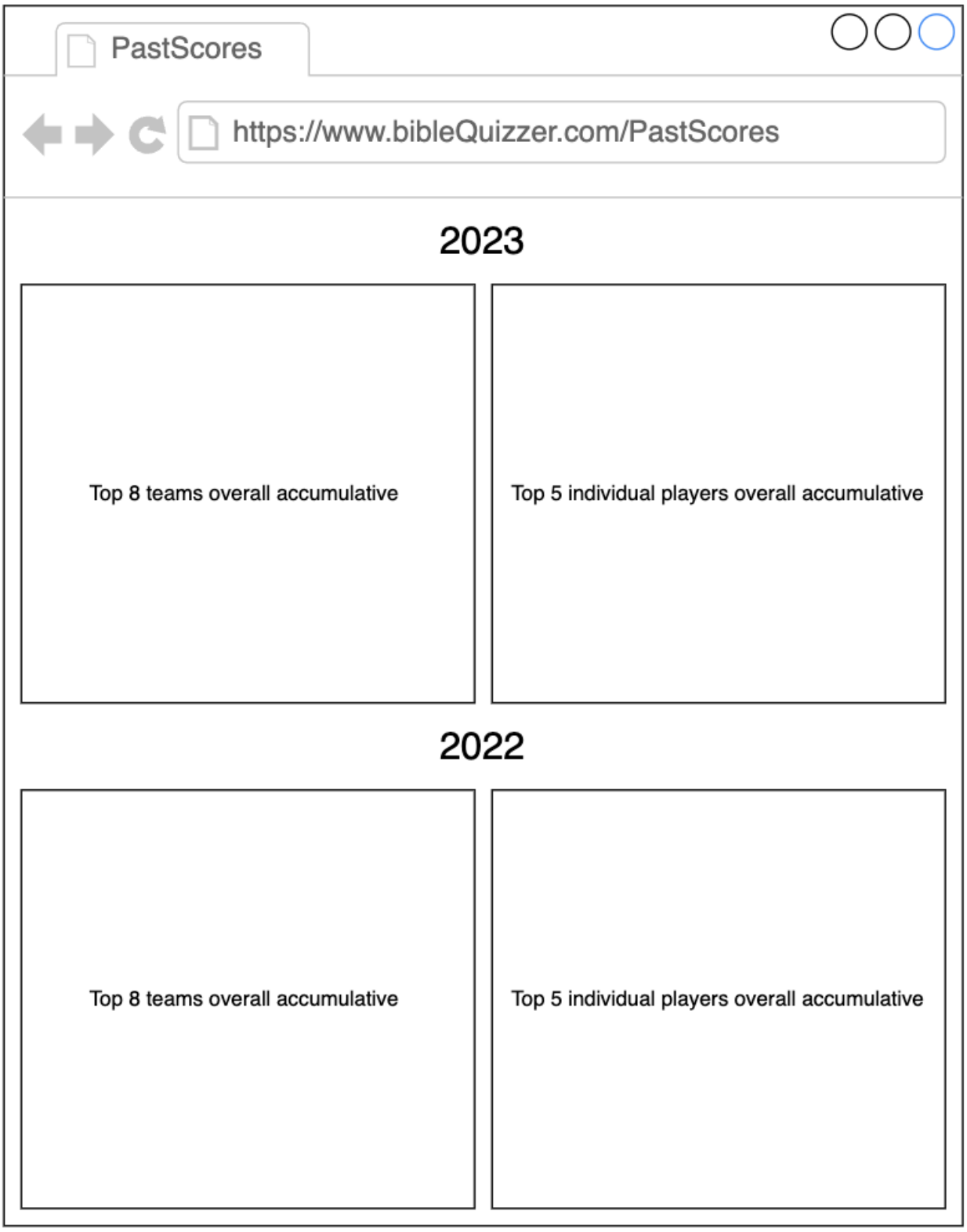
Active Leaderboard Page:



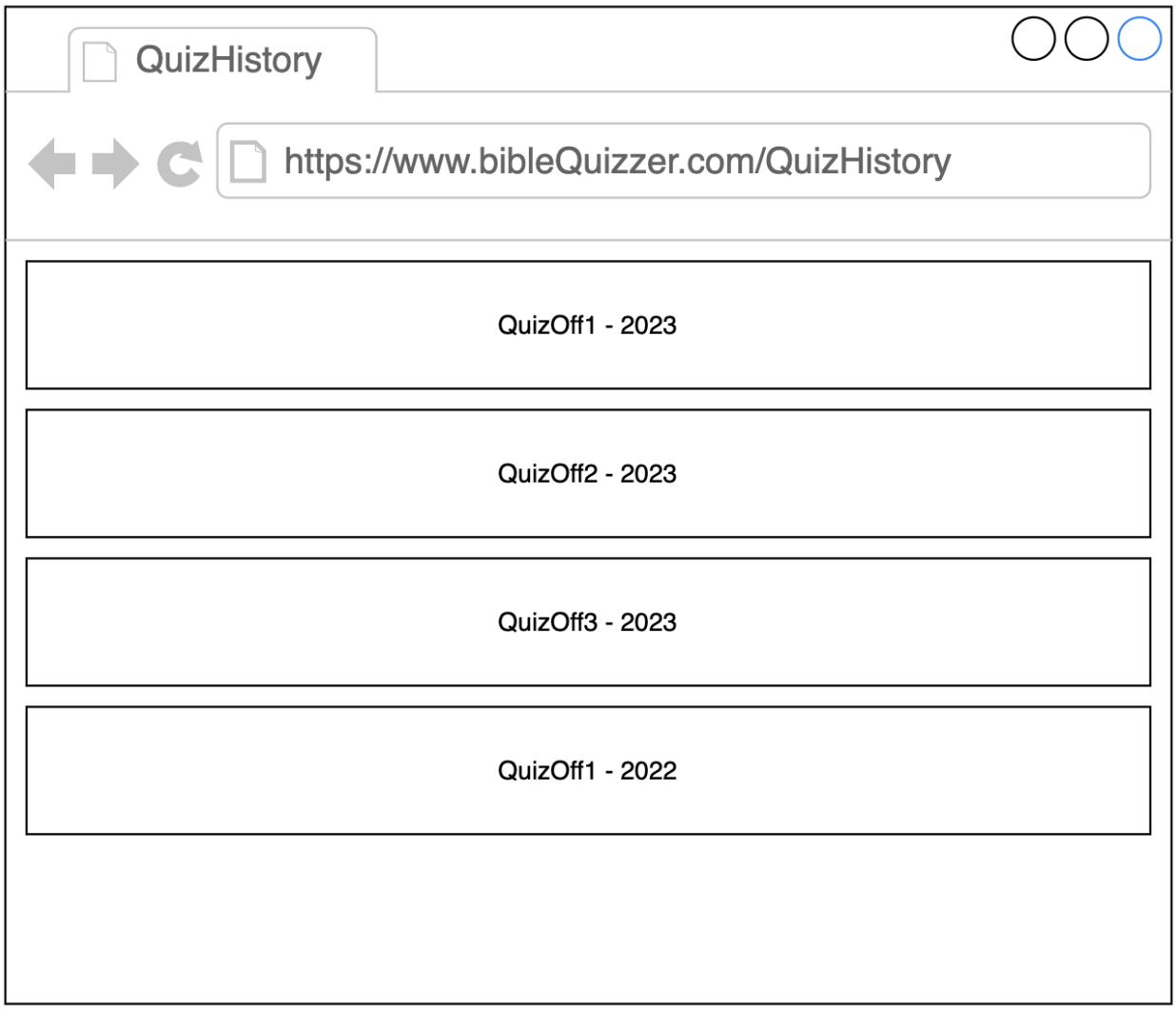
Score Page:


Scores
https://www.bibleQuizzer.com/Scores
Access to all past scoresheets
Statistics

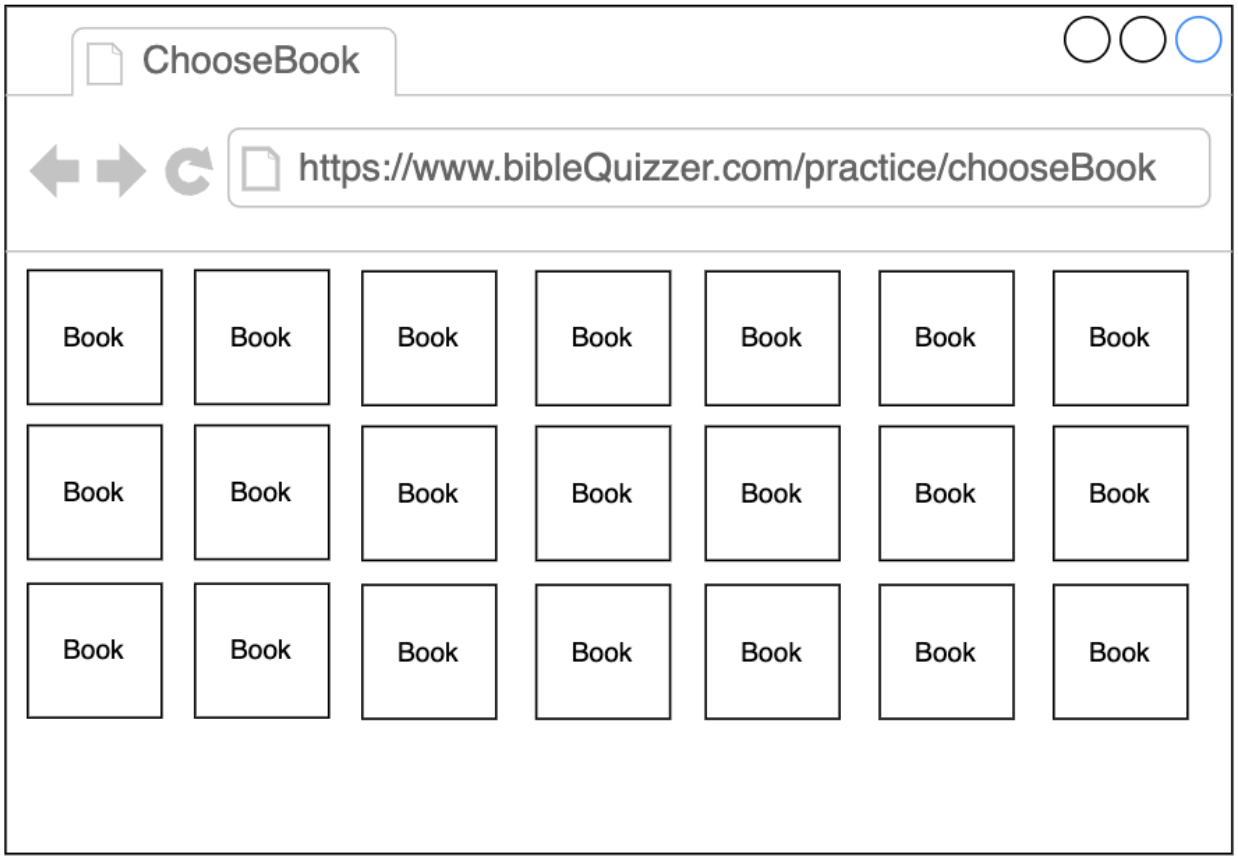
Past Scores Page:



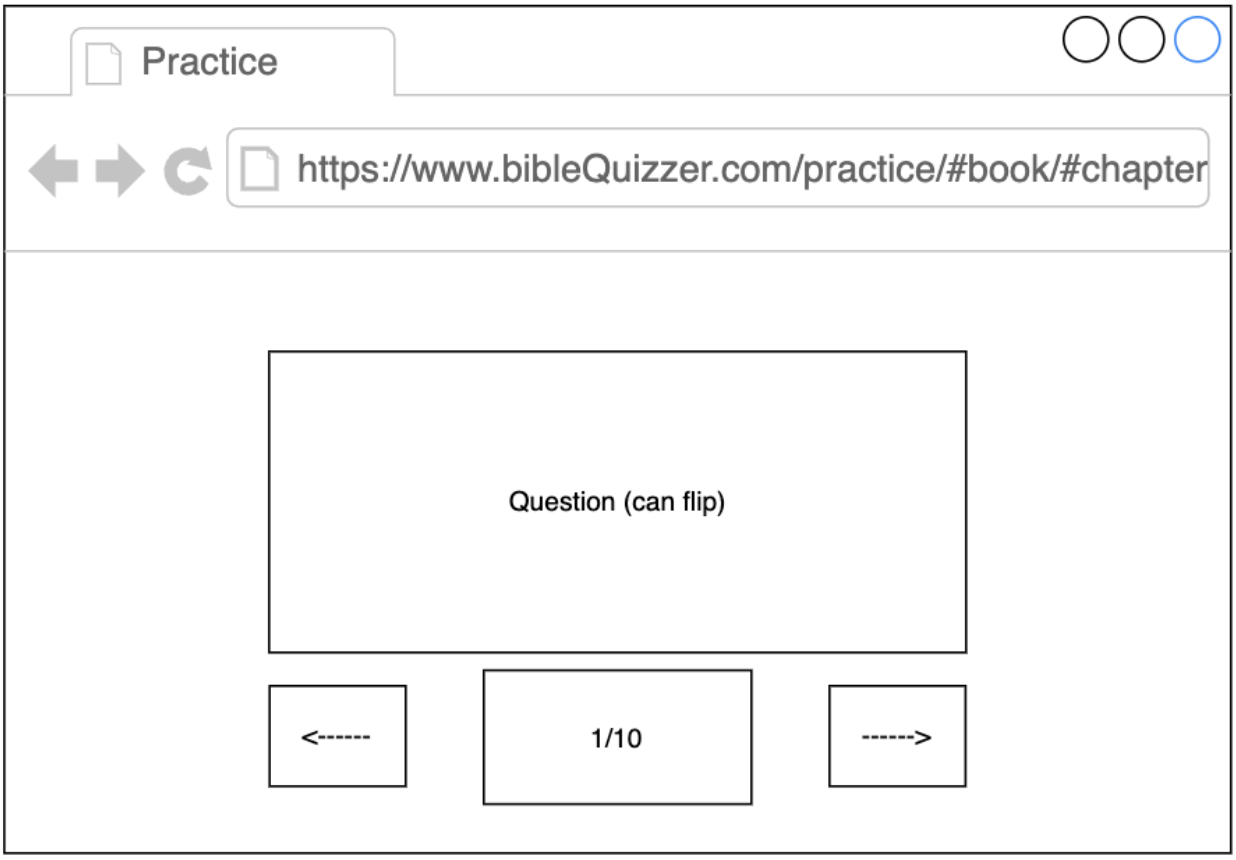
Quiz History Page:



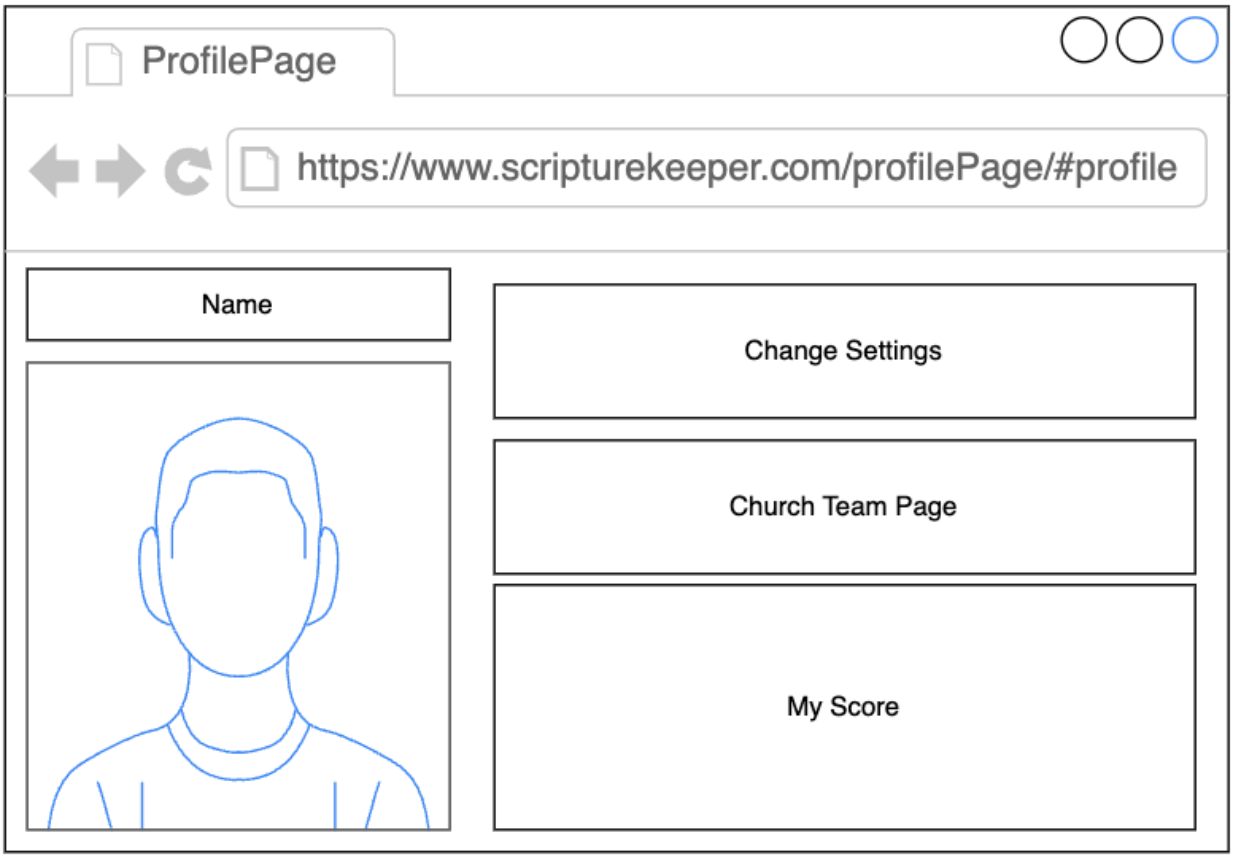
Practice Choose Book Page:



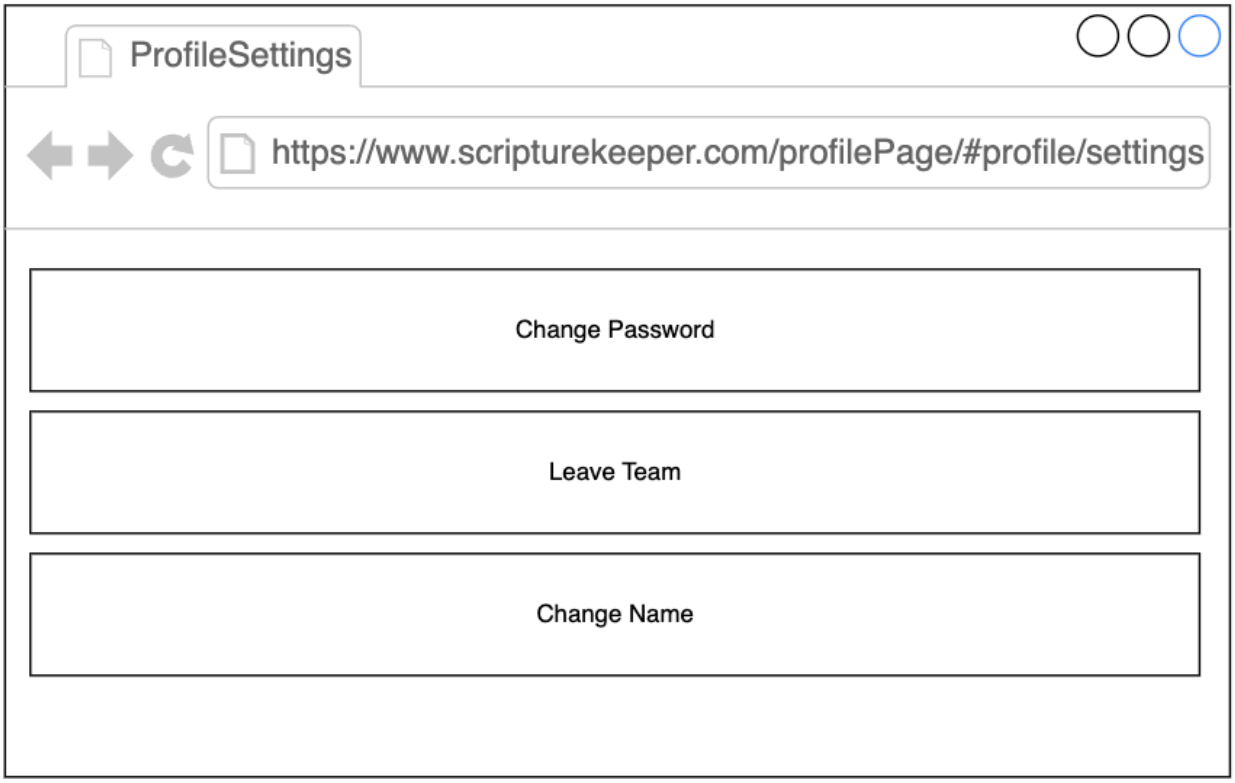
Practice Page:



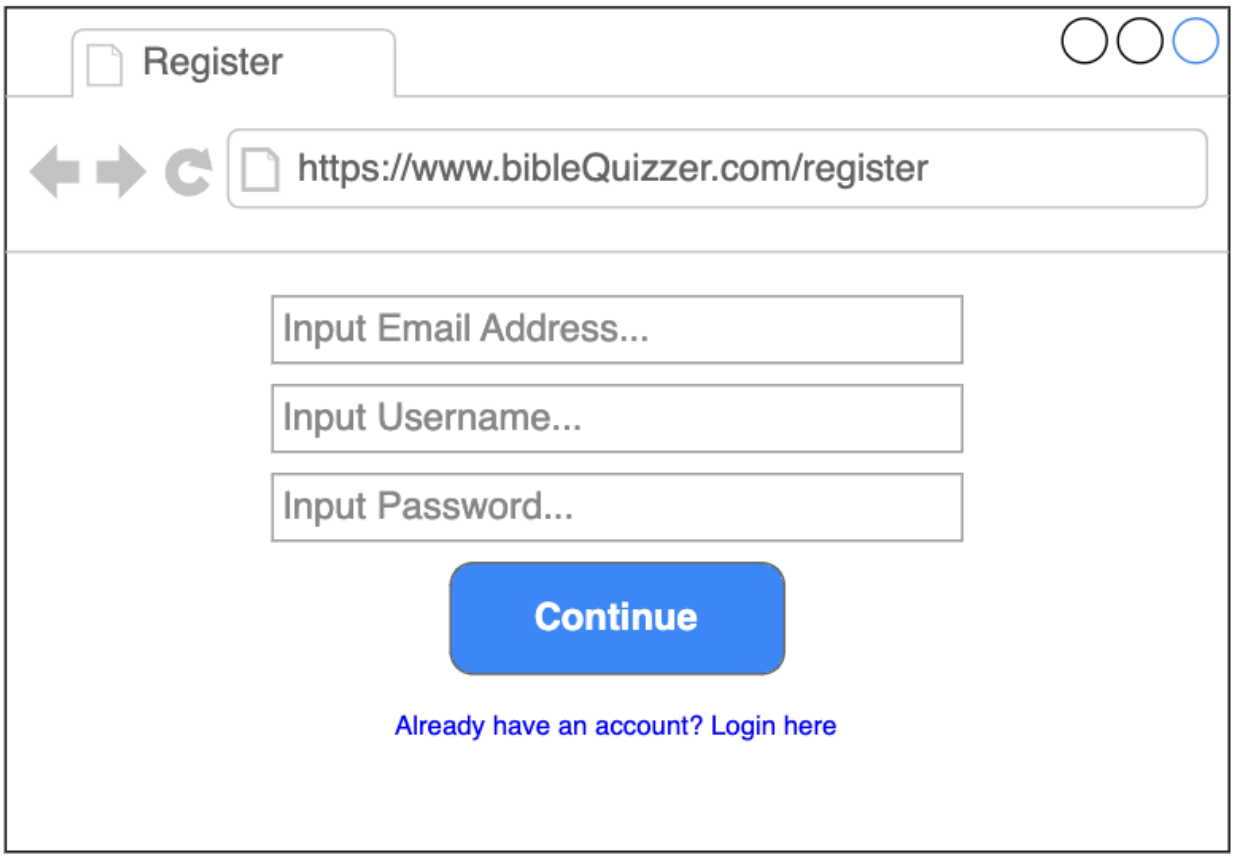
Profile Page:



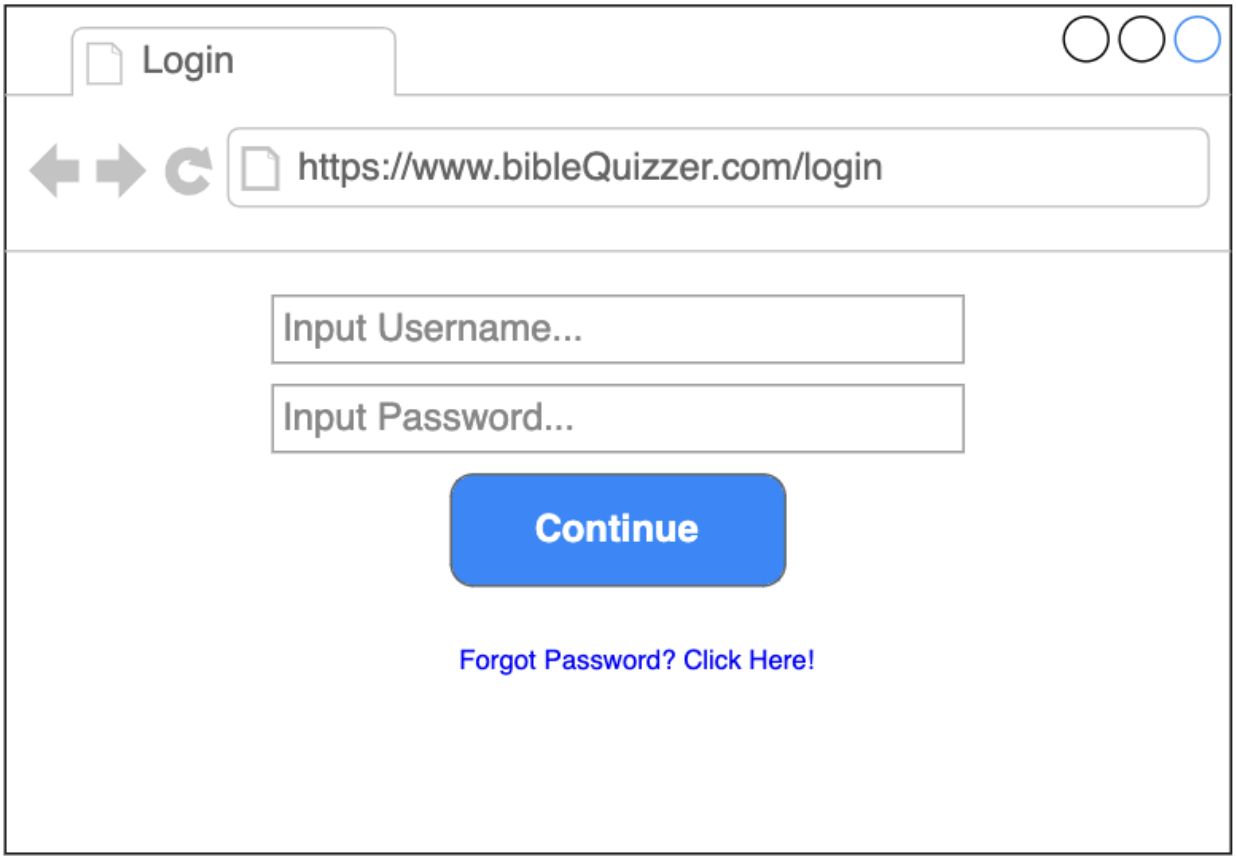
Profile Settings Page:



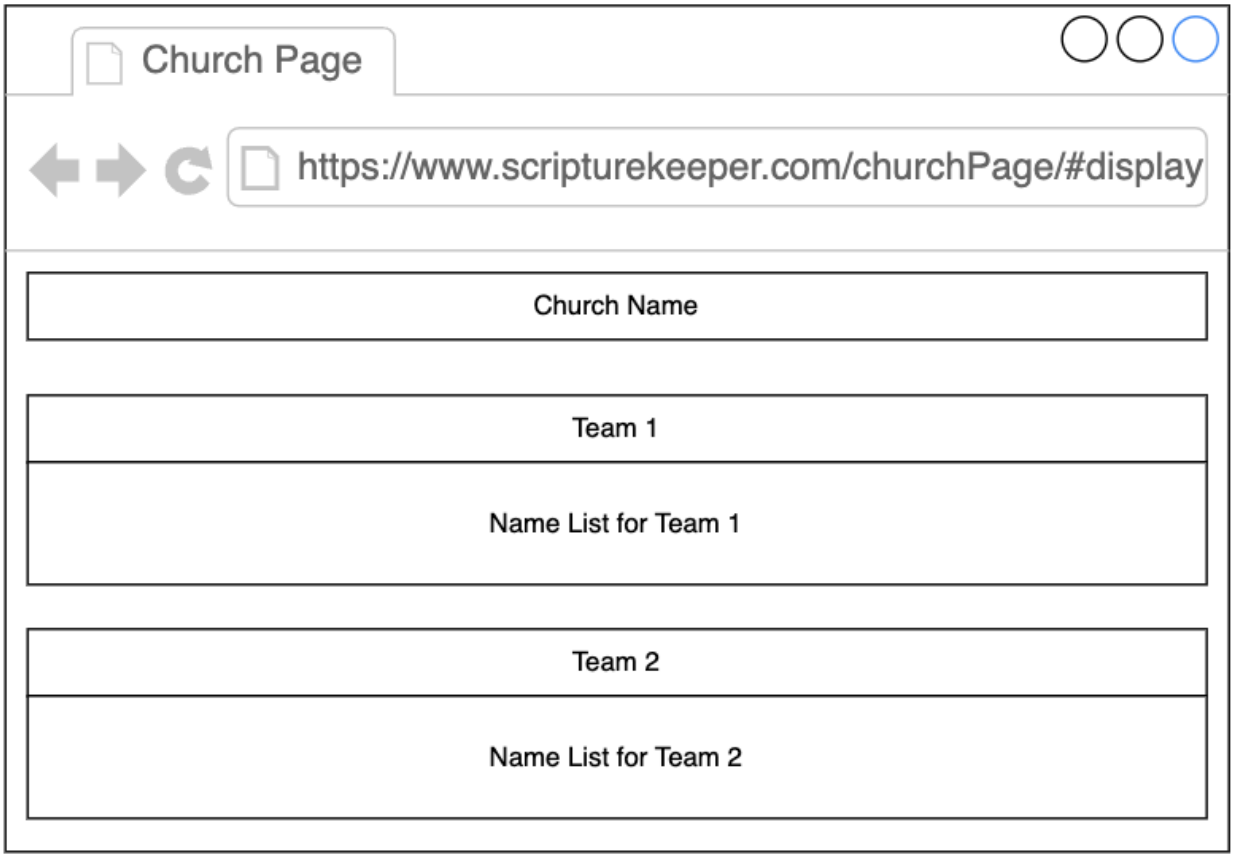
Register Page:



Login Page:


Login
https://www.bibleQuizzer.com/login
Input Username...
Input Password...
Continue
Forgot Password? Click Here!

Church Page:


Church Page
https://www.scripturekeeper.com/churchPage/#display
Church Name
Team 1
Name List for Team 1
Team 2
Name List for Team 2

3. Non-Functional User Requirements

3.1. Usability

This system aims to not only provide a way to meet their needs in prepping and competing in the quiz bowls but to do so smoothly. While it is being used, we want the user to be able to access all they need without straining to figure out where it's located. To meet this goal, we have implemented a couple of different things.

Firstly, during the process of creating this system, we will routinely meet with our client to affirm that the setup/look of the site will bring simple usability for each type of user. Along with that, all of the tabs will be self-explanatory, and it is made to be obvious what you can find on the next page. Overall it will be simple yet effective to reach our audience of players grades 7th-12th as well as coaches and quiz masters of older ages.

3.2. Accessibility

This system will be used in countries across the United States, which is why, for now, this system will be implemented in the English language alone. If in future years this system will be used in more ways and places, we will be able to update this to work across multiple languages. Another possibility further down the line could also be the application of the Bible in Hebrew to learn more of the historical background and original language.

To accommodate assistance with people with hearing or visual impairments, we've brought a couple of things into consideration. The plan for our setup is to have normal-sized writing, always erroring on having larger font than smaller. Since we are creating this system as a website, there are options to add different plug-ins to have the text read aloud. Since our entire system is visual, those with hearing impairments should have no issue using the site.

3.3. Availability

This system will be run via a web application. We chose this method to display this system because of its easily accessible nature. The site will be available 24/7 at any time of day and allows you to log in and access your personal profile

from any device. All functionalities will be available at all times as well. This allows users to practice at any time and coaches/admin to update questions or announcements at any time.

The only case that the site would or could be down is if maintenance is occurring, and even then, the aim is to keep the site going. When maintenance is needed or updates are going to be introduced those will be done overnight when the likelihood of people using the site is much lower so there are not many people to interrupt.

3.4. Documentation

For this project, the important factor is the user and each user type being able to take full advantage of this system that will be created. In order to maximize this, we plan on providing a couple of forms of documentation. As we have included currently, we will provide a final glossary for use by all users in order to understand some of the language we use on the site. For more complex users and actions such as the quiz masters running a quiz bowl, we will provide information on how to begin those as well as a possible video tutorial. As a whole, like we mentioned before, we aim that our site will be simple and intuitive, so no documentation should be needed for the classic uses of our system.

4. Non-Functional System Requirements

4.1. Performance

4.1.1. Responsiveness

The most typical interactions with the system are viewing scoresheets, inputting scores, starting matches, practicing questions, and interacting with announcements.

- Navigating between pages should take less than 4 seconds.
- Creating matches should update the database and return within a few seconds.
- Updating scores will be a continuous operation, taking no longer than a second to save scores to the database.
- Loading practice questions will vary in time because of the size of the data. These operations should show a loading symbol if the process takes longer than 3 seconds.
- The creation and modification of accounts, groups, and permissions will take milliseconds because these types of modifications will not require database migrations.

4.1.2. Concurrent Use

The system must be capable of hosting several users at a time since multiple matches may be taking place and entering data at the same time. It will be able to host a bible bowl event without hindering the system's performance. This system must also be able to handle multiple quizzers practicing questions at the same time. It will be able to host a moderately sized user load (up to a hundred) without hindering the system's performance.

4.2. Capacity

4.2.1. Current Capacity

The system is expected to store accounts for all FEC administrators, coaches, and players involved with Bible quizzing. All Quiz Off data is kept, and historical data not in the current year may be archived and used for historical reporting. At the very least, the system will hold data and accounts for 5 years.

4.2.2. Future Capacity

Since there are new players and Quizoffs every year, the system's database will grow steadily, with roughly the same amounts of new accounts and Quizoff data entered each year. This will require the roughly same amount of archived storage to be added every year. If the number of players, teams, or groups grows too large, drop-down menus may contain too many items to display in a clean fashion. If this eventually becomes the case, it may be necessary to make changes to the interface of the system to better handle a large number of these entities. However, since accounts, teams, and groups can all be deactivated when they are no longer needed, this is not expected to be a problem.

4.3. Longevity

The system will last for 5-10 years. While technology does improve rapidly, FEC is not expected to implement new systems at the same pace. Periodic updates to the system will be provided.

4.4. Security

4.4.1. General Security

These are the different types of roles and their abilities:

- User
    - Create an account
    - Log in
    - Log out
    - Create team
    - Listen to audio Bible

    - Read Bible
    - Bookmark/Save a place in Bible
    - Highlight verses
    - View current scoresheet
    - View current leaderboard
    - View the current schedule of opponents
    - View announcements
    - View Calendar
    - View past scoresheets
    - View statistics
    - Practice quiz questions
- Coach
    - Add players to a team
    - Remove players from a team
    - Send announcement
    - View chat
    - Send a message to player
    - Edit practice questions
- Quiz Master
    - Edit current scoresheet
    - Display current leaderboard
    - Send announcement
- Admin
    - Create round-robin of opponents
    - Send announcement
    - Add quizoff event to calendar
    - Edit quizoff event in calendar
    - Add scoresheets to database
    - Upload practice questions
    - Edit practice questions
- Player
    - View chat
    - Send message to coach

4.4.2. Malicious Access

The only page visible to a user who is not signed in is the login page. This prevents any user without an account from accessing any page where they could easily do harm to the system. System administrators control account creation to prevent any unwanted users having access to the system. Verifying sessions, validating interface inputs, and ensuring database table modifications match the desired formats will prevent string format vulnerabilities, SQL injection, and other undesired inputs in the system. In addition, the user interface will be set up so that users cannot send repeated requests to the server or database without the system navigating them to the source page after data submission.

4.4.3. Privacy and Confidentiality

- Users assigned the coach role edit the practice schedules for their groups, while other coaches only view rosters pertaining to other groups.
- System administrators are the only ones able to assign roles and create, deactivate, and reactivate accounts. They will thereby control users' access to confidential data.