**Fall 2023 SYS 390**

**System Requirements Specification**

**Team ZEST\_ID**

1. **General**
   1. Introduction
   2. Collaboration
   3. User Roles
   4. Schedule
   5. Glossary
2. **Functional Requirements**
   1. System Response Table
   2. Logical Data Model
   3. Data Dictionary
   4. Processes
      1. Data Description
      2. Use Cases
      3. Logical Process Model
   5. Wireframes
3. **Non-Functional Requirements**
   1. Usability
      1. Look and Feel
      2. Ease of Use
   2. Accessibility
      1. Cultural or Language Groups
      2. Users with Disabilities
   3. Availability
      1. Reliability
      2. Downtime
   4. Documentation
   5. Training
4. **Non-Functional System Requirements**
   1. Performance
      1. Responsiveness
      2. Concurrent Use
   2. Capacity
      1. Current Capacity
      2. Future Growth
   3. Longevity
   4. Security

**1 - General**

1.1 - Introduction

This document, the System Requirements Specification, outlines the system requirements for a FEC Bible Quizzing System to aid Grace Morton Church and the Fellowship of Evangelical Churches in carrying out bible quizzes (called “**quiz-offs**") efficiently and effectively. Bolded words may be referenced in the glossary **(1.5)** for more information.

There are many known process errors during **quiz-offs** that are affecting the fairness and enjoyability of the competition:

Firstly, the entire scoring process is prone to human error because of paper scorecards and a lack of consistency in scoring a **quiz team’s** performance. In a single **quiz off**, there could be hundreds of scorecards with potentially thousands of numbers written down. The human error for tallying points using a complicated scoring system is too high to maintain.

Secondly, the current system for bible quizzing logistics is slow. Each **quiz team** needs to be scored, **seeded**, and assigned a schedule. Since nothing about the process is currently automated, individual **quizzes** are prone to running overtime and are susceptible to large amounts of human error and inefficiency.

Lastly, critical information about the information system is maintained by select individuals. In an ideal system, our hope is that it would be able to store necessary information and business logic inside of itself. This includes effectively scoring **quiz teams**, **seeding brackets**, and allocating **rooms** for **quizzes** without having to rely on a few specialists who have a deep understanding of the entire event structure. This document outlines our understanding of the system.

1.2 - Collaboration

The FEC Bible Quizzing System will regularly interact with the hardware system in the quizzing lightbox (used to keep time and track the first **quizzer** to respond to a question), as there is one **Quizmaster** dedicated to overseeing the lightbox’s operations and ensuring that no answer runs overtime. If an overtime error occurs, this will be recorded in the quizzing system as a **foul** (**1.5**).

Another system the quizzing system will interact with is Google Sheets or Microsoft Excel, as that is a common way to create data spreadsheets that can be shared and edited collectively. While the quizzing system will internally host data from Quiz-Offs, the system must be able to export its data in various common file formats, such as .csv, .txt, and .pdf, for viewing on Windows, iOS, and Android devices. This allows information recorded and stored within the system to easily be transmitted between users and devices external to the system.

The system must be compatible with email accounts, as it will be expected to integrate existing emails into its secure login system. Notifications from the system will also be forwarded to associated user emails in order to facilitate easy information sharing.

The FEC Bible Quizzing System must also interact with various other organizations and their integrated systems to host Bible quizzing tournaments. While the quizzing system is meant to be highly portable and self-contained, it is possible that the system functions may need to be slightly altered to suit the environment that the host organization provides.

1.3 - User Roles

There are three major roles that will be part of this system: **Coaches**, **Quizmasters**, and **Administrators** (see glossary in **1.5**). The system allows for the same human to have multiple roles at once, but for practical purposes should be limited. For example, it is possible for a system user to be both an **administrator** and a **quizmaster** during a **quiz-off**, but the responsibilities for these roles differ greatly, and the system will give each role a different level of access. Therefore, it is not recommended to assign multiple significant roles to one user. **Administrators** should try to give users distinct roles at each **quiz-off** to ensure the smoothest event flow possible.

**Administrators** oversee the most critical operations inside of the FEC Bible Quizzing System. They are given the ability to perform the following operations inside of the system:

* Register participating teams for a **quiz-off**
* Create questions to be used during **quizzes**
* Create brackets from the **seeding phase** quiz results
* Determine which questions should be asked for each **quiz** in the **quiz-off**
* View the **bracket** in the **bracket phase**.
* Transition system from **seeding phase** to **bracket phase**
* Transition system from **bracket phase** to **award phase**

**Quizmasters** enforce the rules during individual **quizzes**. In the system, they can:

* Enter a score, **foul**, **timeout**, error, correct, or incorrect answer from a quizzer.
* Denote whether a question is a bonus question
* View a response time for a quizzer
* Ask a question to a quizzer
* Verify a correct answer
* View all quiz questions assigned for that **quiz** by the **administrator**
* View how many points a team has
* View **room** schedules
* Schedule a **room** for the **quiz**
* Confirm **quiz** results

Every **quiz team** has a **coach**. Inside of the system, coaches can:

* Register their **quiz team** for a **quiz-off**
* Draft an official lineup for **quizzers** competing in the next **quiz-off**
* View the live scoring records that the **quizmasters** are entering during a **quiz-off**

1.4 - Schedule

The FEC Bible Quizzing system is intended to be built within the limited timeframe of Taylor University’s Spring 2024 semester. This means that the development process must take place across 16 weeks in total: 15 weeks of regular classes, plus finals week. Considering that a class worth 4 credit hours translates to approximately 8 hours of homework each week, students may expect to spend an average of 8 hours per week working on this project. This means a single student will average at approximately 128 total work hours by the end of the 16th week. However, considering that development will be handled by students in groups of approximately 5 or more, it is reasonable to assume that development will progress at an accelerated rate.

Development of the project will begin on the first day of the spring semester, January 30. The semester will officially conclude the week of May 13, during which we will present our final product to the client. An initial framework of the new Bible Quizzing System should be completed by the end of March, with the final product being deliverable by May 13. This allows additional leeway between the beginning of finals week and the final presentation to the client, occurring during this course’s exam period.

1.5 - Glossary

**Administrator** – Person or persons who are in charge of organizing **quiz-off** logistics, but who are not **quizmasters** themselves.

**Award phase** – The final phase of the **quiz-off,** in which awards are given out to the winning **quiz teams**.

**Bonus Question** - Extra question in a **quiz**.

**Bracket phase** – After the **seeding phase**, the top eight **quiz teams** compete in a bracket-style competition to determine the winner of the **quiz-off**. There are always 8 teams in the bracket.

**Captain** – Position granted to a designated **quizzer**. Can only have one **captain** per team. The only quizzer on a quiz team who may contest answers.

**Church** – An organization that produces one or more **quiz teams**. Can call timeouts but may not answer for the team during a quiz off.

**Co-captain** – If the **captain** is absent or **quizzes out**, may become the new captain.

**Coach** – The leader of a **quiz team.**

**Foul** - Called by a **quizmaster**. Influences points negatively.

**Lineup** – A list of all participating **quizzers** and substitutes, submitted to the scorekeeper before each **quiz**.

**Quiz** – In a **quiz-off**, there will be many quizzes. Quizzes usually last around 20 minutes. Judged by three **quizmaster**s– the timekeeper, the scorekeeper, and the reader.

**Quiz-off** – An event that happens approximately three times per year. Several **teams** from many **churches** get together to compete against one another.

**Quiz out** – When a **quizzer** answers 5 correct questions, they quiz out and no longer compete for the rest of the quiz. May be replaced by a **substitute**.

**Quiz team** – A group consisting of at least three **quizzers** and one coach (with no upper limit on the number of students).

**Quizmaster** – The people who read the questions and enforce the rules during a **quiz**. There are three types – the timekeeper, the scorekeeper, and the reader.

**Quizzer** – An individual who competes on a **quiz team**.

**Room** – Q**uiz-offs** are often held in churches, schools, or college campuses that contain many different rooms. Each **quiz** is assigned a room that will host that **quiz**.

**Seeding phase** – The phase of the **quiz-off** in which **quiz teams** compete in four **quizzes** to determine who will make it into the **bracket phase.**

**Substitute** – Each **quiz team** may have any number of extra quizzers to replace players that “quiz out” during a **quiz**.

**Timeout** – A **coach** or a **captain** may call a **timeout.** Each team gets three 45 second **timeouts** in which normal **quiz** activity ceases.

**2 - Functional Requirements**

2.1 - System Response Table

The System Response Table (SRT) is designed to showcase all major actions that users may perform within the system. Table items are organized by keys, which correspond to a specific user type (i.e. IDs beginning with “A” are associated with “Admin” users).

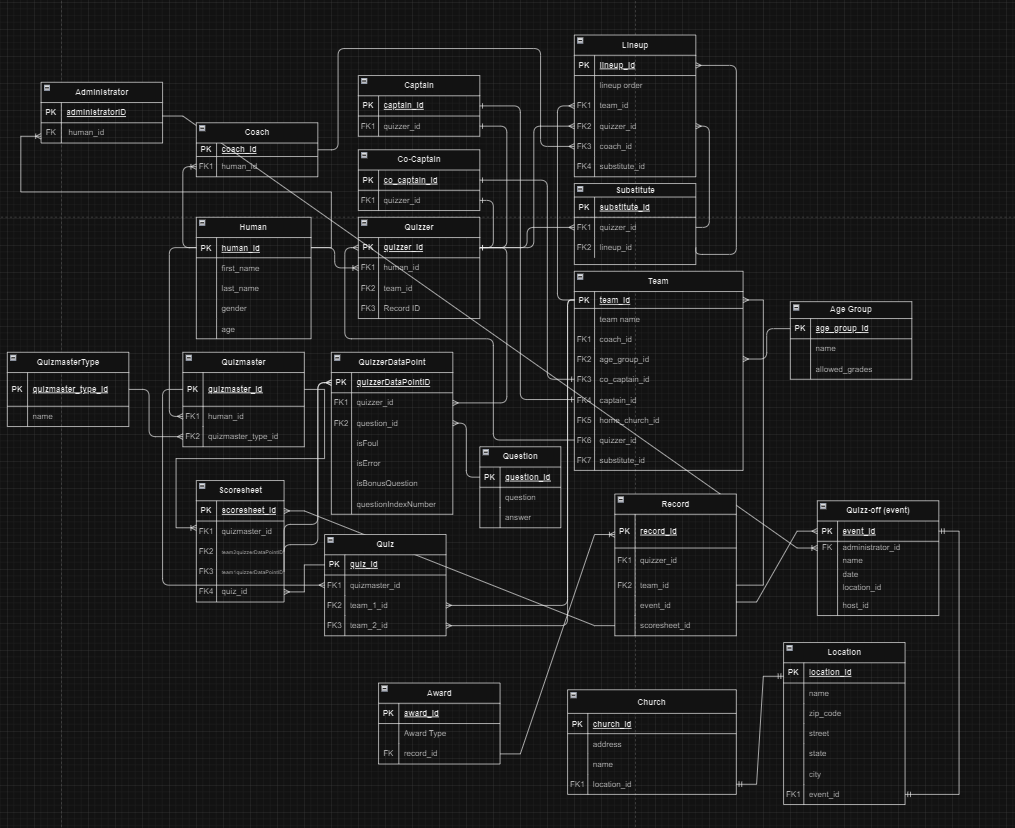
Users are described as the “Source” of a system response, and actions they perform in the system are the “Trigger.” The system then provides a “Response” in the form of “Major Outputs” at the indicated “Destination.”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Source | Trigger | Response | Major Outputs | Destinations |
| A1 | Admin | Desire to register participating teams | Team registration information compiled | Registered teams recorded in database | Quizzing bracket database |
| A2 | Admin | Desire to create quiz questions | Questions created | Questions | Questions database |
| A3 | Admin | Desire to create quiz bracket | Quiz bracket generated | Quiz bracket information displayed | Quizzing app schedules |
| A4 | Admin | Desire to select questions for different rounds | Quiz questions assigned | Questions sent to Quizmasters | Questions database |
| A5 | Admin | Desire to view quiz bracket | Quiz bracket generated | Quiz bracket information displayed | Quizzing app schedules |
| A6 | Admin | Desire to start the seeding phase of a quiz-off | Active quiz-off state begun, seeding phase begun | Application pages updated | Quizzing app statistics |
| A7 | Admin | Desire to transition from seeding phase to bracket phase | Seeding phase ended, bracket phase begun | Application pages updated | Quizzing app statistics |
| A8 | Admin | Desire to transition from bracket phase to award phase | Bracket phase ended, award phase begun | Application pages updated | Quizzing app statistics |
| A9 | Admin | Desire to give awards | Awards distributed, active Quiz-off state ended | Application pages updated | Quizzing app statistics |
| A10 | Admin | Desire to create room schedule | Room schedule in system | Room scheduling confirmation page | Quizzing App |
| A11 | Admin | Desire to edit a bracket | Bracket edited | Bracket edits applied in database | Quizzing app, database |
| A12 | Admin | Desire to edit a room schedule | Room schedule edited | Room schedule edits applied in database | Quizzing app, database |
| A13 | Admin | Desire to delete accounts | Accounts deleted | Accounts and related information removed from database | Quizzing app, database |
| C1 | Coach | Desire to register team for a tournament | Admin contacted, team entered into system | Team registered for Quiz Off | Quizzing app, Database |
| C2 | Coach | Desire to make a substitute on a team | Student lineup changed | Scoresheet tracking adjusted | Quizzing app |
| C3 | Coach | Desire to record a score | Fill personal score sheet | Personal score sheet filled | Quizmaster quizzing sheet |
| C4 | Coach | Desire to train team | Upload practice quizzes | Practice quizzes | Quizzing app - practice quizzes |
| C5 | Coach | Desire to view score records | Score records displayed | Score record information page displayed | Quizzing app |
| C6 | Coach | Desire to submit a student line up | Student lineup created | Quiz data updated | Quizzing app |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Source | Trigger | Response | Major Outputs | Destinations |
| C7 | Coach | Desire to create a team | Team created | Team is created and students are added to the team | Quizzing app, Database |
| C8 | Coach | Desire to have accurate scores | Scoresheet able to be checked against quizmaster's | Scoresheet added to database | Quizzing app, Database |
| Q1 | Quizmaster | Desire to enter a score | Data entry field created | Score recorded | Quizzing App |
| Q2 | Quizmaster | Desire to ask a question | Question displayed | Question | Projected Screen |
| Q3 | Quizmaster | Desire to verify correct answer | Student answer checked | Points or penalty awarded | Quizzing App, Database |
| Q4 | Quizmaster | Desire to calculate final quiz score | Earned points calculated | Team score recorded | Quizzing App, Database |
| Q5 | Quizmaster | Desire to view room schedule | Room availability viewed | Current room schedule page | Quizzing App |
| Q6 | Quizmaster | Desire to enter a foul | Data entry field created | Foul recorded | Quizzing App, Database |
| Q7 | Quizmaster | Desire to upload quiz results | Quiz results saved in database | Quiz-off database updated | Database |
| Q8 | Quizmaster | Desire to view quiz questions | Quiz questions displayed | Questions | Quizzing App |
| S1 | Student | Desire to register for a team | Student assigned to team | Student added to team roster | Quizzing app, Database |
| S2 | Student | Desire to view score records | Score records displayed | Score record information page displayed | Quizzing app |
| S3 | Student | Desire to take a practice quiz | Practice quiz generated | Practice quiz displayed | Quizzing app |
| S4 | Student | Desire to view their quizzing statistics | Statistics generated | Statistics page displayed | Quizzing app |
| U1 | User | Desire to create an account | Account created | Account added to database | Quizzing App, Database |
| U2 | User | Desire to log in | User logged in | User can see their information | Quizzing App |
| U3 | User | Desire to add account information | User information added | Database and user profile updated | Quizzing App, Database |
| U4 | User | Desire to log out | User logged out | User information no longer shown | Quizzing App |

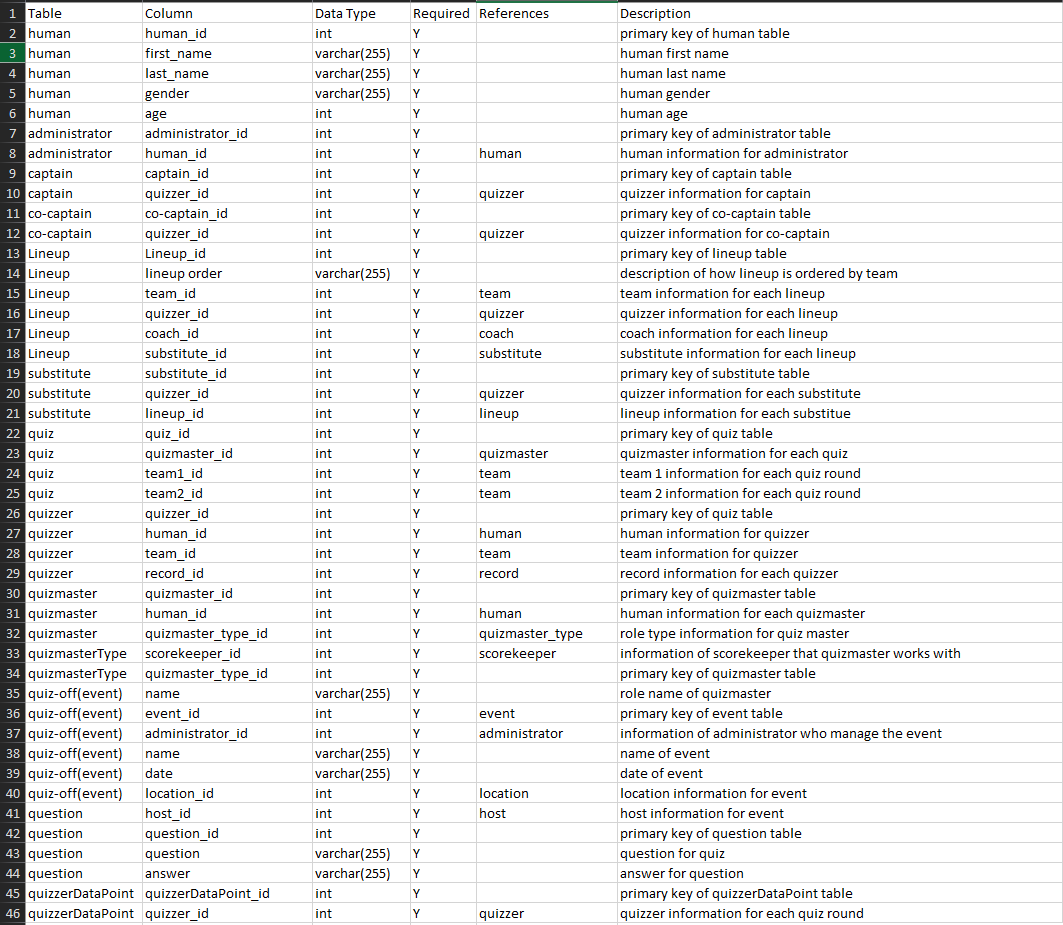
2.2 - Logical Data Model

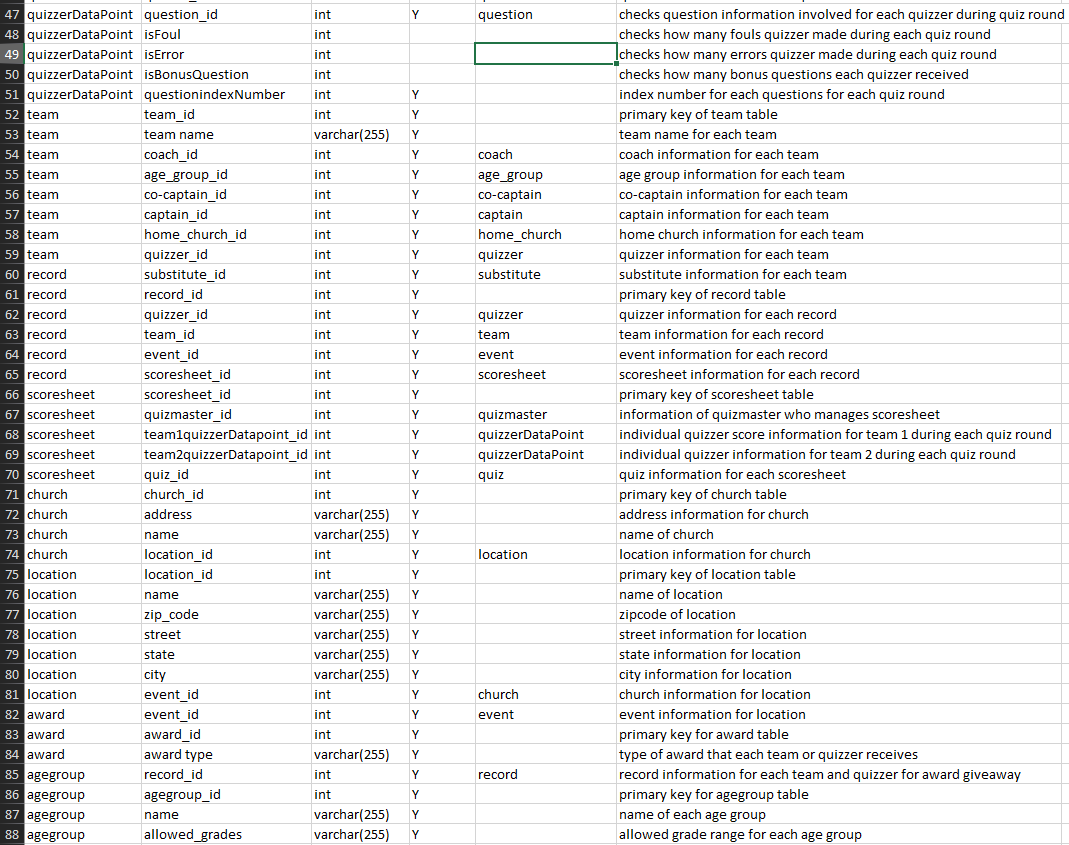
A logical data model is used to graphically represent the relationships between people, objects, places, concepts, or events in a system. It is used in database modeling to determine what data must be stored and how to design the database structure.



2.3 - Data Dictionary

Data Dictionaries are used to provide detailed information about the attributes of each entities, such as their data types, whether required or not, referencing entity, and text descriptions. A data dictionary provides a concise guide to understanding and using the data within the database.





2.4 - Processes

2.4.1 - Data Description

The data description specifies the sources of data used in the data flow diagrams (**section 2.4.2**). Each main bullet point is a function outlined in the data flow diagrams, while the subpoints describe relevant entities from the Entity Relationship Diagram (**section 2.2**) and the data they supply to the data flow.

Bolded and underlined items are entities, while bolded items are table keys. Non-bolded items are attributes native to their parent entity.

* Register Team
* **From administrator**: **administrator\_id**, **human\_id**
* **From team**: **team\_id**, team name, **coach\_id**, **age\_group\_id**, **co-captain\_id**, **captain\_id**, **home\_church\_id**, **quizzer\_id**
* From **quiz-off(event)**: **event\_id**, **administrator\_id**, name, date, **location\_id**
* Create Question
* **From administrator: administrator\_id, human\_id**
* **From question**: **host\_id**, **question\_id**, **question**, **answer**
* Create Quiz Bracket
* **From administrator: administrator\_id, human\_id**
* **From quiz**: **quiz\_id**, **quizmaster\_id**, **team1\_id**, **team2\_id**
* Choose Questions
* **From administrator: administrator\_id, human\_id**
* **From question**: **question\_id**, **question**, **answer**
* View Bracket
* **From administrator: administrator\_id, human\_id**
* **From quiz**: **quiz\_id**, **quizmaster\_id**, **team1\_id**, **team2\_id**
* Start Seeding
* **From administrator: administrator\_id, human\_id**
* **From scoresheet: scoresheet\_id, quizmaster\_id, team1quizzerDataPoint\_id**, **team2quizzerDataPoint\_id**, **quiz\_id**
* Start Bracket
* **From administrator: administrator\_id, human\_id**
* **From scoresheet**: **scoresheet\_id**
* **From quiz**: **quiz\_id**
* Start Awards
* **From administrator: administrator\_id, human\_id**
* **From award: event\_id, award\_id, award type**
* **From quiz-off(event): event\_id, administrator\_id, name, date,** **location\_id**
* Give Awards
* **From administrator: administrator\_id, human\_id**
* **From award: event\_id, award\_id, award type**
* **From team: team\_id, team name**
* **From quizzer**: **quizzer\_id**, **human\_id**
* Create Schedule
* **From administrator: administrator\_id, human\_id**
* **From quiz-off(event): event\_id, name, date, location\_id**
* **From location**: **location\_id**, **name**, **zip\_code**, **street**, **state**, **city**
* Edit Quiz Bracket
* **From administrator: administrator\_id, human\_id**
* **From quiz**: **quiz\_id**
* Edit Schedule
* **From administrator: administrator\_id, human\_id**
* **From quiz-off(event)**: **event\_id**, **name**, **date**, **location\_id**
* Delete Accounts
* **From administrator**: **administrator\_id**, **human\_id**
* Create **Account:**
* **From** **Human**: **human\_id**, **first\_name**, **last\_name**, **gender**, **age**
* User **Login:**
* **From** **Human**: **human\_id**
* Edit **Account:**
* **From** **Human**: **human\_id**, **first\_name**, **last\_name**, **gender**, **age**
* User **Logout:**
* **From** **Human**: **human\_id**
* Team **Registration:**
* **From Human: human\_id, first\_name, last\_name**
* **From Team: team\_id**
* **From** **Coach**: **coach\_id**
* View **Records**
* **From Human: human\_id, first\_name, last\_name**
* **Fro**m **quizzerDataPoint**:**quizzerDataPoint\_id**
* T**ake Practice Quiz:**
* **From Human: human\_id, first\_name, last\_name .**
* **Fr**om **Quiz**: **quiz\_id**
* View **Quiz Statistics:**
* **From Human: human\_id, first\_name, last\_name**
* **F**rom **quizzerDataPoint**: **quizzerDataPoint\_id**
* Enter **Score**
* **From quizmaster: quizmaster\_id, human\_id**
* **From scoresheet: scoresheet\_id**
* **From quizzerDataPoint**: **quizzerDataPoint\_id**, **quizzer\_id**
* **Ask question**
* **From quizmaster: quizmaster\_id, human\_id**
* **From question**: **question\_id**, **question**, **answer**
* Answer **Reveal**
* **From quizmaster: quizmaster\_id, human\_id**
* **From questions**: **question\_id**, **answer**
* Score **Calculation**
* **From Quizmaster: quizmaster\_id, human\_id**
* **From Scoresheet: scoresheet\_id**
* **From QuizzerDataPoint: quizzerDataPoint\_id, quizzer\_id, question\_id**, **isFoul**, **isError**, **isBonusQuestion**, **questionindexNumber**
* View **Room schedule**
* **From Quizmaster: quizmaster\_id, human\_id**
* **From Location**: **location\_id**, **name**, **zip\_code**, **street**, **state**, **city**
* Foul **Entry**
* **From Quizmaster: quizmaster\_id, human\_id**
* **From Scoresheet: scoresheet\_id**
* **From QuizzerDataPoint**: **quizzerDataPoint\_id**, **quizzer\_id**, **isFoul**
* Upload **Quiz Results**
* **From Quizmaster: quizmaster\_id, human\_id**
* **From Scoresheet: scoresheet\_id**
* **From Quiz: quiz\_id**
* **From Quizzer**: **quizzer\_id**
* View **Questions**
* **From Quizmaster**: **quizmaster\_id**, **human\_id**
* **From Questions**: **question\_id**, **question**, **answer**
* **Team Registration / Team Acceptance**
* **From Coach: coach\_id, human\_id**
* **From Team: team\_id, team name, coach\_id, age\_group\_id, co-captain\_id, captain\_id, home\_church\_id, quizzer\_id**
* **From Admin**: **administrator\_id**, **human\_id**
* Sub **Student**
* **From Coach: coach\_id, human\_id**
* **From Substitute: substitute\_id, quizzer\_id**
* **From Lineup: Lineup\_id, lineup order, team\_id, quizzer\_id, coach\_id**, **substitute\_id**
  + **Scoresheet**: **scoresheet\_id**
* Record **Score**
* **From Coach: Coach\_id, human\_id**
* **From Scoresheet: scoresheet\_id**
* **From QuizzerDataPoint**: **quizzerDataPoint\_id**, **quizzer\_id**, **question\_id**
* **Create Practice Quiz**
* **From Coach: coach\_id, human\_id**
* **From Quiz: quiz\_id, quizmaster\_id**
* **From Question: question\_id, answer**
* **From Quiz-off(event): If the practice quiz is associated with** a specific event, then **event\_id**, **administrator\_id**, **date**, **location\_id** might also be relevant
* View **Records**
* **From Coach: coach\_id, human\_id**
* **From Team: team\_id, coach\_id**
* **From Record: record\_id, quizzer\_id, team\_id, event\_id, scoresheet\_id**
* **From Scoresheet**: **scoresheet\_id**, **quizmaster\_id**, **team1quizzerDataPoint\_id**, **team2quizzerDataPoint\_id**, **quiz\_id**
* Submit **Lineup**
* **From Coach: coach\_id, human\_id**
* **From Lineup: Lineup\_id, lineup order, team\_id, quizzer\_id, coach\_id, substitute\_id**
* **From Team:** **team\_id**, **team name**, **coach\_id**
* **Team Creation**
* **From Coach: coach\_id, human\_id**
* **From Team: team\_id, team name, coach\_id, age\_group\_id, co-captain\_id**, **captain\_id**, **home\_church\_id**, **quizzer\_id**
* Submit **Scoresheet**
* **From Coach: coach\_id, human\_id**
* **From Scoresheet: scoresheet\_id, quizmaster\_id, team1quizzerDataPoint\_id**, **team2quizzerDataPoint\_id**, **quiz\_id**

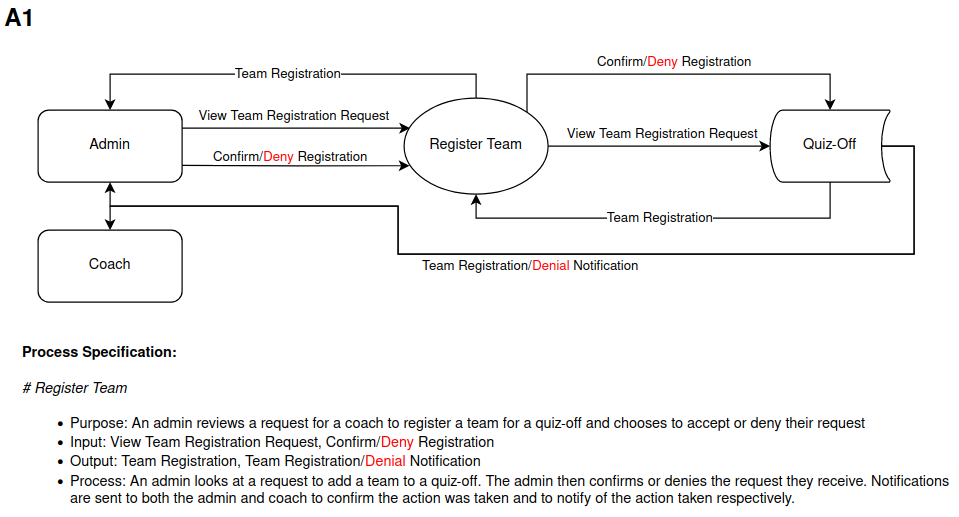
2.4.2 - Use Cases & 2.4.3 - Logical Process Model

This section contains use cases and logical process models for our system. The use cases are the tables, and the logical process models consist of data flow diagrams (charts) and process specifications (bullet points), which are located below their associated use cases.

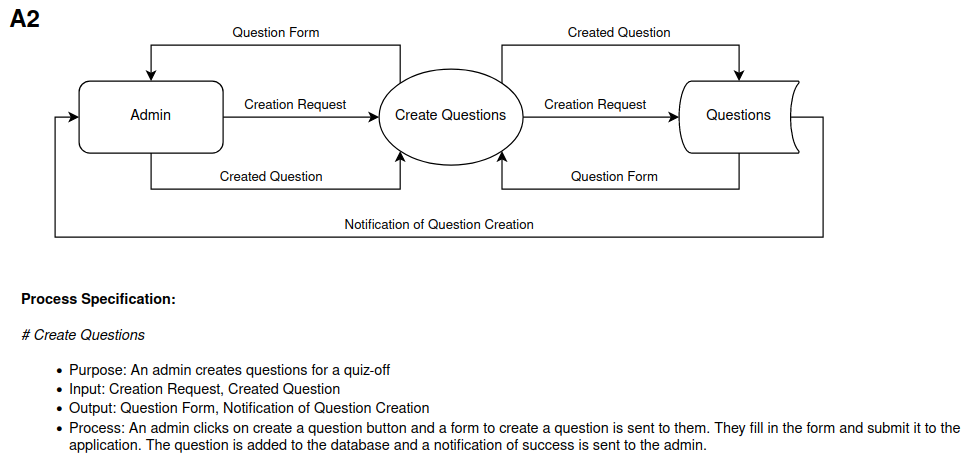
The use cases describe the general processes that the FEC Bible Quizzing System will contain to fulfill the client’s requirements. The logical process models complement the use cases by highlighting the main functions and the data that the system requires to execute them. The models show involved entities, where the data moves, and more functional specifics than the use cases.

When reading the logical process models, the red lines and labels refer to alternative event flows, while the black lines refer to typical event flows in the use cases.

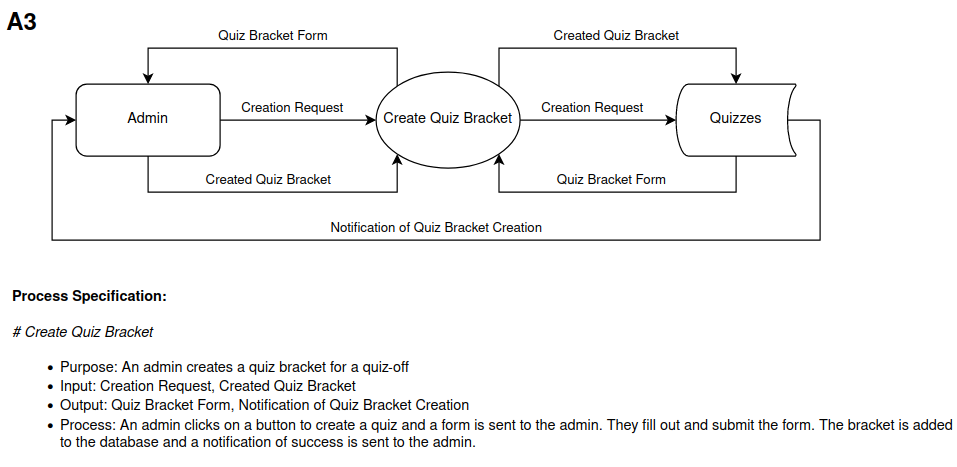
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | A1 |  |  | |
| **Name** | Register Teams | | | |
| **Primary Actor** | Admin | | | |
| **Other Actors** |  | | | |
| **Description** | Admins are in charge of organizing quiz-offs. An admin will log into their account and see pending quiz-off team requests. They then can confirm or deny the request with a comment section for denials. Confirmed teams are registered for the quiz-off, and denied applicants are notified that their team was not registered. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Admin clicks on a request to review. | |  | |
| 2) Admin confirms request. | |  | |
|  | | 3) System sends confirmation to the coach who registered the team. | |
|  | | 4) System sends confirmation that the team registration was accepted. | |
|  | | 5) System adds the confirmed team to a database holding registered quiz-off teams. | |
| **Alternative Event Flow** | 1) Admin clicks on a request to review. | |  | |
| 2) Admin denies request. | |  | |
|  | | 3) System sends denial to the Coach who registered the team. | |
|  | | 4) System sends confirmation that the registration denial went through. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to register teams for a quiz-off. | Preconditions | Admin has requests for team registrations, is logged in and is navigated to the team registration request page. |
| **After Event** | Conclusion | Teams are registered for the quiz-off. | Postconditions | Registered teams are added to a database for the quiz-off and denied teams are notified of their registration denial. |



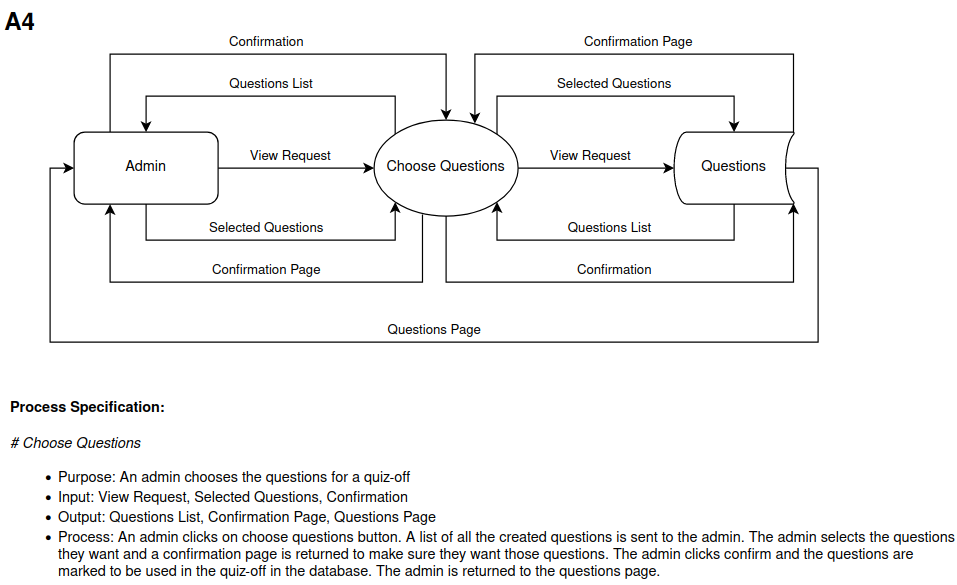
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | A2 |  |  | |
| **Name** | Create Quiz Questions | | | |
| **Primary Actor** | Admin | | | |
| **Other Actors** |  | | | |
| **Description** | Admin creates quiz-off questions to add to a pool of questions for a quiz-off. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Admin clicks on the create question button. | |  | |
|  | | 2) Question creation form is displayed. | |
| 3) Admin fills in information required to create the question. | |  | |
| 4) Admin submits question creation form. | |  | |
|  | | 5) Question added to the database and updated on the questions page. | |
|  | | 6) Notification sent to admin of question creation being a success. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to create quiz questions. | Preconditions | Admin is logged in and navigates to the questions page. |
| **After Event** | Conclusion | Quiz questions are created. | Postconditions | Questions are created and stored in a database for quiz-off questions. |



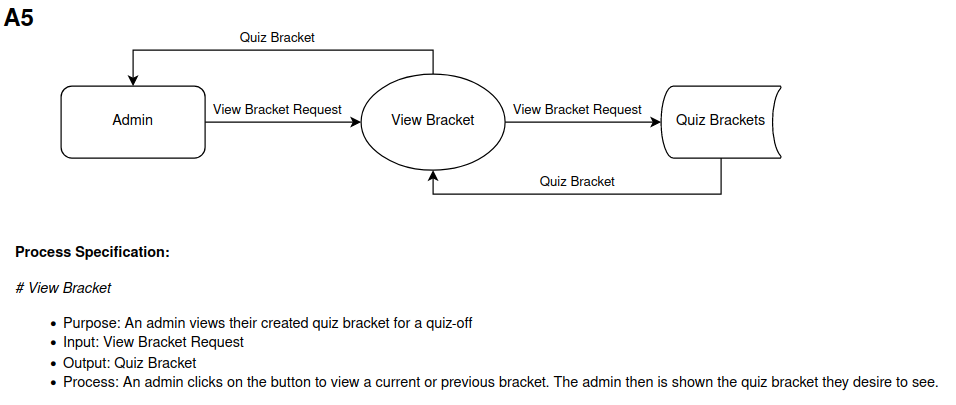
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | A3 |  |  | |
| **Name** | Create Quiz Bracket | | | |
| **Primary Actor** | Admin | | | |
| **Other Actors** |  | | | |
| **Description** | Admin navigates to the quiz scheduling page and creates a quiz bracket which is then stored in the database and updated on the quiz scheduling page. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Admin clicks create quiz bracket button. | |  | |
|  | | 2) Bracket generation form is displayed. | |
| 3) Admin fills in information for the bracket. | |  | |
| 4) Admin submits the form. | |  | |
|  | | 5) Bracket information is stored in the database. | |
|  | | 6) Created bracket is updated on the quiz scheduling page. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to create a quiz bracket for a quiz-off. | Preconditions | Admin is logged in and has navigated to the quiz scheduling page. |
| **After Event** | Conclusion | Quiz bracket has been created. | Postconditions | Quiz bracket is created and stored in the quiz-off schedule page. |



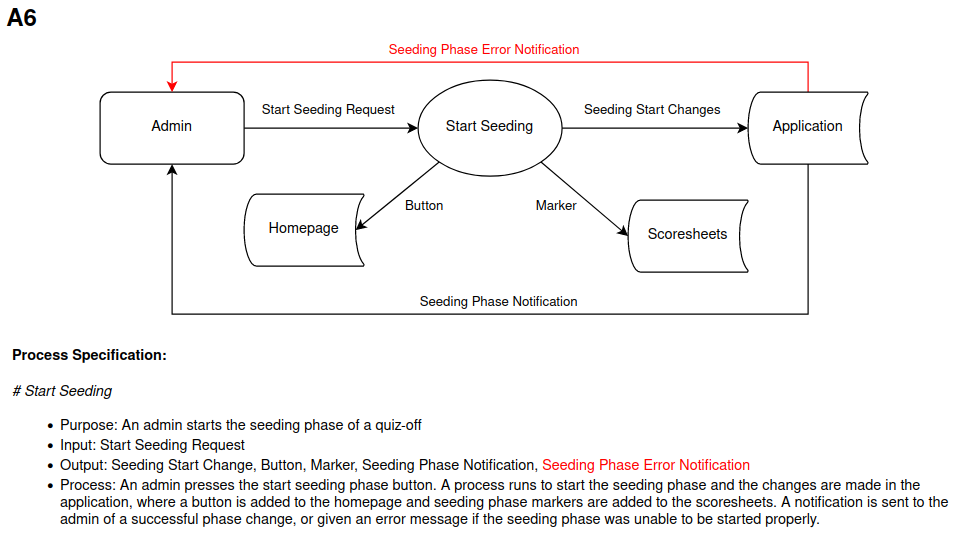
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | A4 |  |  | |
| **Name** | Select Quiz-off Questions | | | |
| **Primary Actor** | Admin | | | |
| **Other Actors** |  | | | |
| **Description** | Admin selects the questions for the current quiz-off so the quizmasters can look at and read the questions while hosting the quizzes. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Admin clicks on the button to organize questions. | |  | |
|  | | 2) Page displays all questions from the database for a quiz-off. | |
| 3) Admin clicks on questions they would like in the quiz-off rotation. | |  | |
| 4) Admin clicks submit. | |  | |
|  | | 5) Page displays confirmation with a list of all chosen questions. | |
| 5) Admin clicks confirm. | |  | |
|  | | 6) Chosen questions sent to the database marked as current quiz-off questions and are accessible to Admins and Quizmasters on the questions page. | |
|  | | 7) Page is routed back to the questions page. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to select questions for quiz-off rounds. | Preconditions | Admin is logged in and navigates to the questions page. |
| **After Event** | Conclusion | Questions for quiz-off rounds are selected and accessible to quizmasters. | Postconditions | Quiz-off questions are sent to quizmasters and stored in a database. |



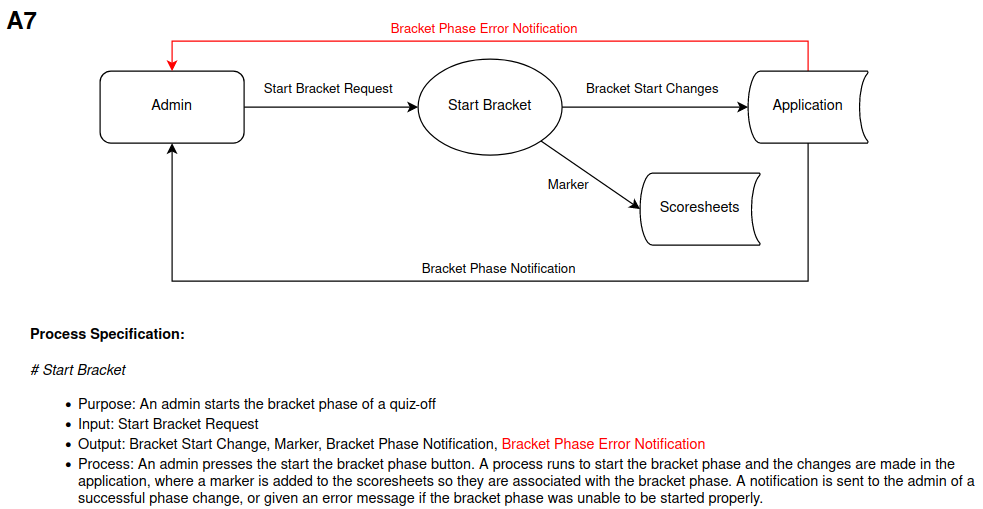
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | A5 |  |  | |
| **Name** | View Quiz Bracket | | | |
| **Primary Actor** | Admin | | | |
| **Other Actors** |  | | | |
| **Description** | Admin views a quiz bracket that has already been created in the system. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Admin clicks view current quiz bracket button. | |  | |
|  | | 2) Page displays the current quiz bracket. | |
| **Alternative Event Flow** | 1) Admin clicks view past quiz brackets button. | |  | |
|  | | 2) Page displays a list of past brackets. | |
| 3) Admin clicks on desired bracket. | |  | |
|  | | 3) Page displays desired bracket. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to view the quiz bracket. | Preconditions | Quiz bracket has been created, and Admin is logged in and viewing the scheduling page. |
| **After Event** | Conclusion | Quiz bracket has been viewed. | Postconditions | Quiz bracket is shown on the screen to the Admin. |



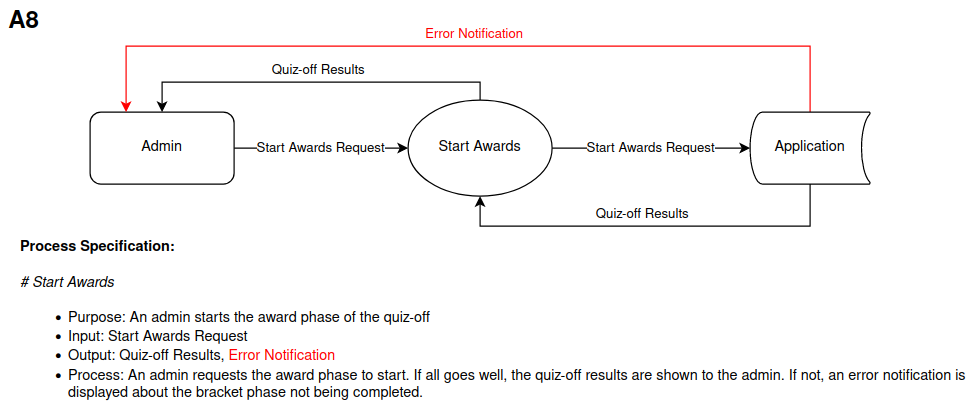
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | A6 |  |  | |
| **Name** | Quiz-off Seeding | | | |
| **Primary Actor** | Admin | | | |
| **Other Actors** |  | | | |
| **Description** | Admin starts the seeding phase of a quiz-off, which tells the system to associate any generated and submitted scoresheets with the seeding phase. Adds a button to determine top 8 performing teams. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Admin presses start seeding phase button. | |  | |
|  | | 2) Application moves to seeding phase, adding a marker to associate all scoresheets with this phase. | |
|  | | 3) Button is added to the home page for Admin to determine the top 8 performing teams. | |
|  | | 4) Notification sent to Admin that the seeding phase has started. | |
| **Alternative Event Flow** | 1) Admin presses start seeding phase button. | |  | |
|  | | 2) Notification is sent to Admin that there are either no questions, or the seeding bracket is not complete. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to start the seeding phase of a quiz-off. | Preconditions | There are questions for the quiz-off, seeding brackets are created, and the Admin is logged in and is navigated to the quiz-off page. |
| **After Event** | Conclusion | Quiz-off is in the seeding phase. | Postconditions | Application is in the seeding phase of a quiz-off. |



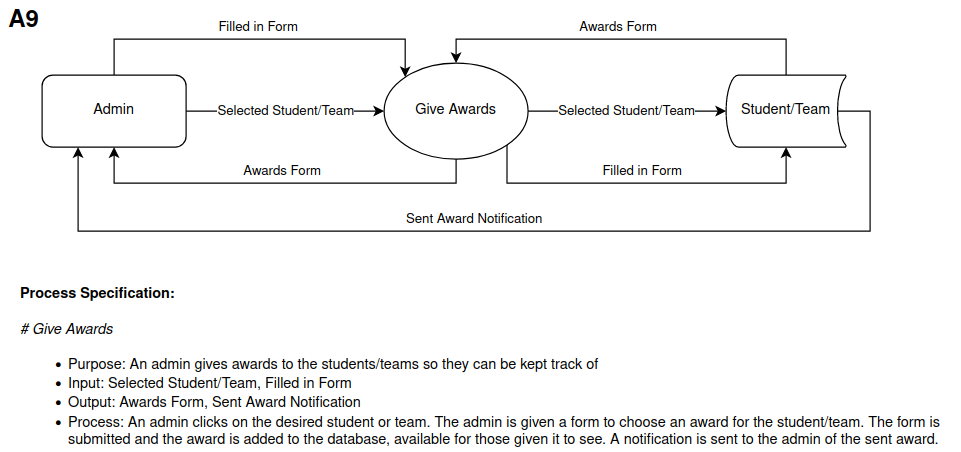
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | A7 |  |  | |
| **Name** | Quiz-off Seeding to Bracket Phase | | | |
| **Primary Actor** | Admin | | | |
| **Other Actors** |  | | | |
| **Description** | Admin starts the bracket phase of a quiz-off, which tells the system to associate any generated and submitted scoresheets with the bracket phase. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Admin presses the start bracket phase button. | |  | |
|  | | 2) Application moves to the bracket phase, adding a marker to associate all scoresheets with this phase. | |
|  | | 3) Notification sent to Admin that the seeding phase has started. | |
| **Alternative Event Flow** | 1) Admin presses the start bracket phase button. | |  | |
|  | | 2) Notification sent to Admin that the seeding phase has not been completed. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to transition quiz-off state from seeding phase to bracket phase. | Preconditions | Application is in the seeding phase of a quiz-off. |
| **After Event** | Conclusion | Quiz-off is in the bracket phase. | Postconditions | Application is in the bracket phase of a quiz-off. |



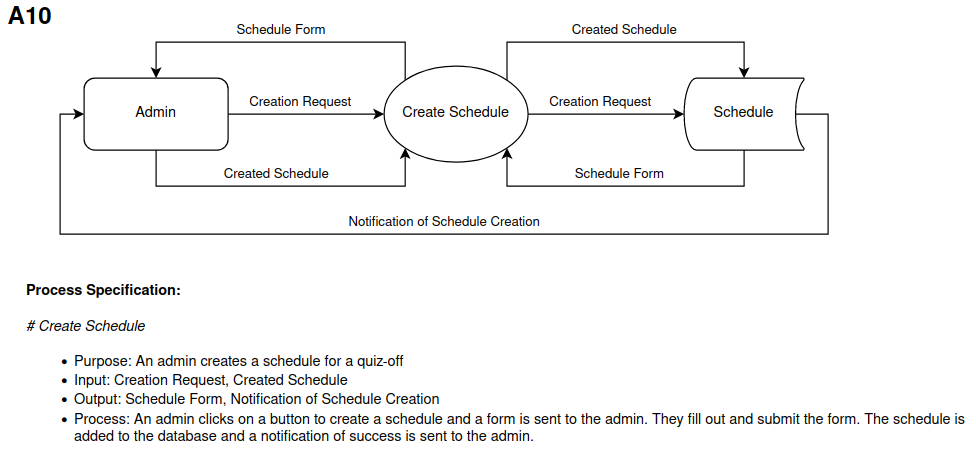
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | A8 |  |  | |
| **Name** | Quiz-Off Bracket to Award | | | |
| **Primary Actor** | Admin | | | |
| **Other Actors** |  | | | |
| **Description** | Admin wants to know the top three teams, and the top students in certain statistics, such as point amount. System displays the statistics for the Admin to view. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Admin presses start award phase button. | |  | |
|  | | 2) Application displays the top three teams and general statistics for the top students for the Admin to view. | |
| **Alternative Event Flow** | 1) Admin presses start award phase button. | |  | |
|  | | 2) Notification sent to Admin that the bracket phase has not been completed. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to transition quiz-off state from the bracket phase to the award phase. | Preconditions | Application is in the bracket phase. |
| **After Event** | Conclusion | Quiz-off is in the award phase. | Postconditions | Application is in the award phase. |



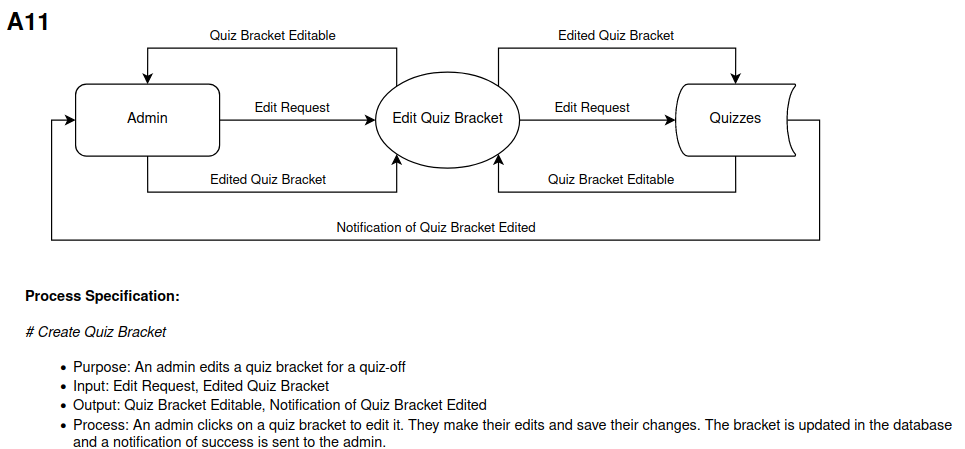
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | A9 |  |  | |
| **Name** | Quiz-off Awards | | | |
| **Primary Actor** | Admin | | | |
| **Other Actors** |  | | | |
| **Description** | Admin gives awards out to teams and students. The application notifies the students and holds records of their awards. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Admin chooses who gains the award and clicks on the appropriate student or team. | |  | |
|  | | 2) Page displays the award form. | |
| 3) Admin selects which award is going to the student or team. | |  | |
|  | | 4) Database is updated on what student or team has the award and makes it available to them. | |
|  | | 5) Notification sent to Admin that award was given. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to dispense awards. | Preconditions | Application is in the award phase. |
| **After Event** | Conclusion | Awards are dispensed. | Postconditions | Awards are sent to relevant teams/students. Application returns to non-quiz-off state. |



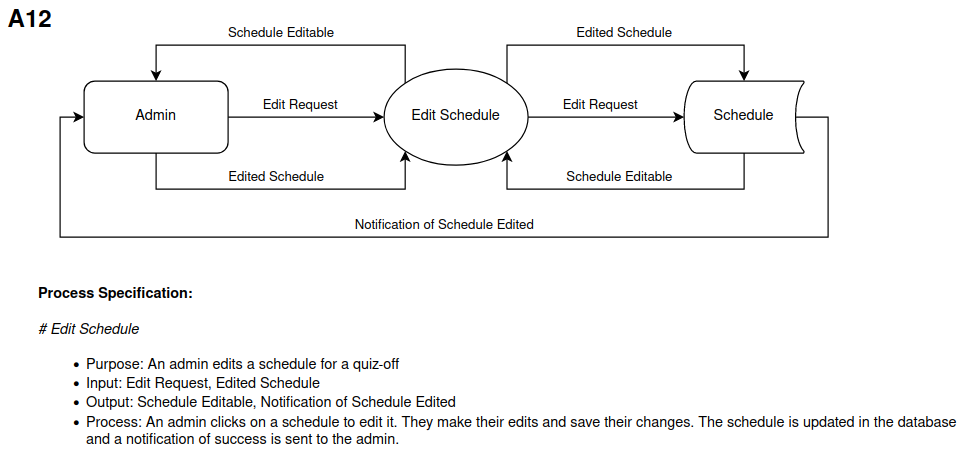
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | A10 |  |  | |
| **Name** | Create Room Schedule | | | |
| **Primary Actor** | Admin | | | |
| **Other Actors** |  | | | |
| **Description** | Admin wants to create a room schedule for a quiz-off. The room schedule is able to be created and accessible to coaches, quizmasters and admins. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Admin clicks on the create room schedule button. | |  | |
|  | | 2) Page displays the room schedule form. | |
| 3) Admin fills out the room schedule form with desired schedule and submits. | |  | |
|  | | 4) Room schedule added to database and is accessible to view on the schedule page for Coaches, Quizmasters and Admins. | |
|  | | 5) Notification sent to Admin that schedule was created successfully. | |
|  | | 6) Page displays newly created room schedule. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to create a room schedule for a quiz-off. | Preconditions | All participating teams are registered and in the system database. Admin is logged in and has navigated to the scheduling page. |
| **After Event** | Conclusion | Quiz-off room schedule created. | Postconditions | Room schedule created, stored and displayed. |



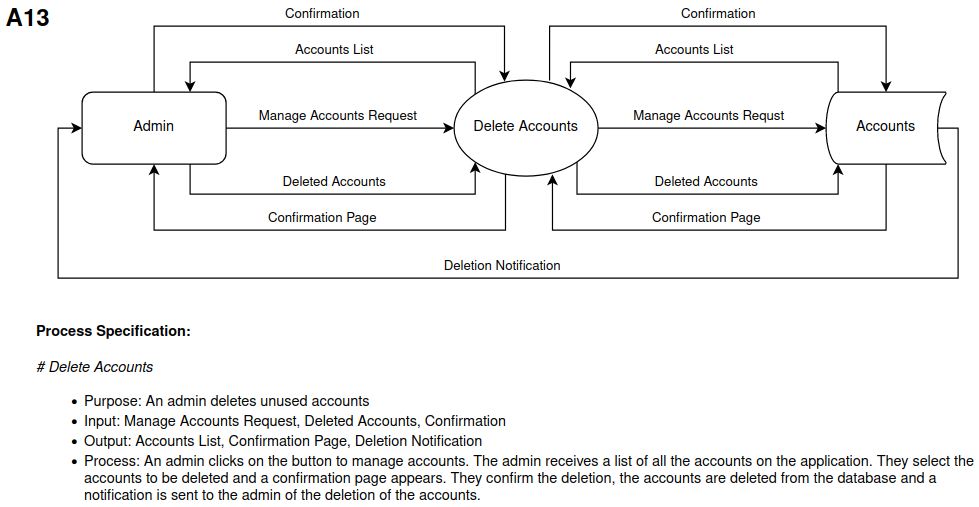
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | A11 |  |  | |
| **Name** | Edit Bracket | | | |
| **Primary Actor** | Admin | | | |
| **Other Actors** |  | | | |
| **Description** | Admin has already created a bracket but wants to edit it. Admin edits the bracket and the results are shown in the application. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Admin clicks on the edit bracket button. | |  | |
|  | | 2) Bracket is displayed for editing. | |
| 3) Admin makes edits and submits the bracket. | |  | |
|  | | 4) Bracket edits are made in the database. | |
|  | | 5) Newly edited bracket is displayed on the page with a notification of success. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to edit the quiz bracket. | Preconditions | Quiz bracket has been created and Admin is logged in and navigated to the scheduling page. |
| **After Event** | Conclusion | Quiz bracket has been edited. | Postconditions | Quiz bracket changes shown on screen and in the database. |



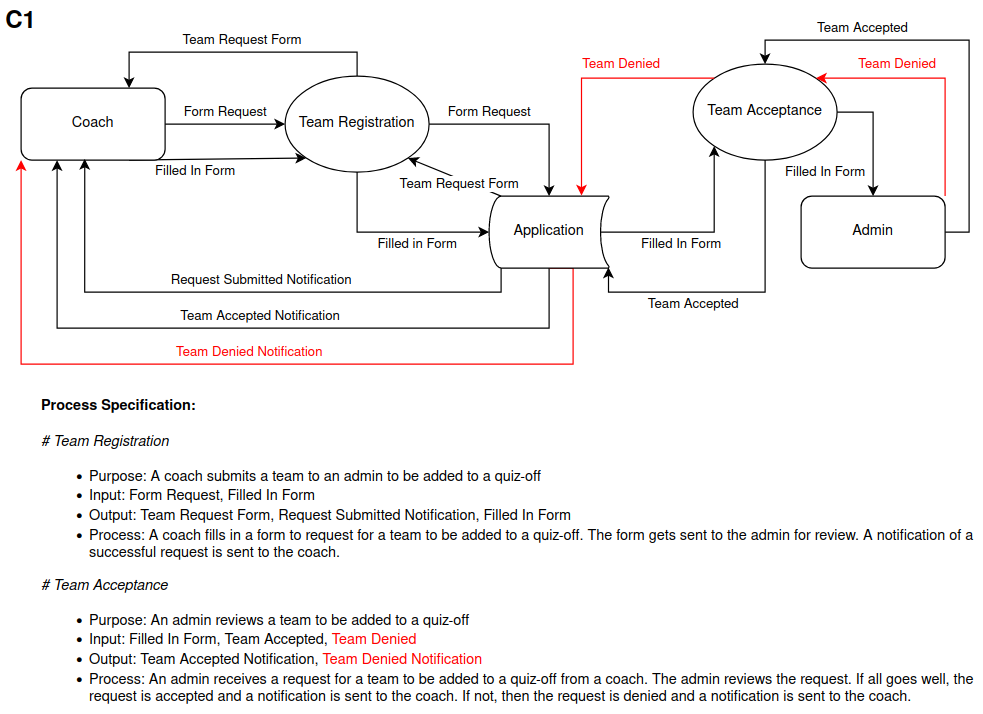
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | A12 |  |  | |
| **Name** | Edit Room Schedule | | | |
| **Primary Actor** | Admin | | | |
| **Other Actors** |  | | | |
| **Description** | Admin edits an existing room schedule and the changes are shown on the application. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Admin clicks on the edit room schedule button. | |  | |
|  | | 2) Room schedule is displayed for editing. | |
| 3) Admin makes edits and submits them. | |  | |
|  | | 4) Room schedule changes made in the database and edited the schedule is able to be viewed on the schedule page for Coaches, Quizmasters and Admins. | |
|  | | 5) Notification sent to Admin that schedule was edited successfully. | |
|  | | 6) Page displays the newly changed room schedule. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to edit a room schedule for a quiz-off. | Preconditions | Room schedule exists in the database. Admin is logged in and has navigated to the scheduling page. |
| **After Event** | Conclusion | Quiz-off room schedule edited. | Postconditions | Room schedule edited, stored and displayed. |



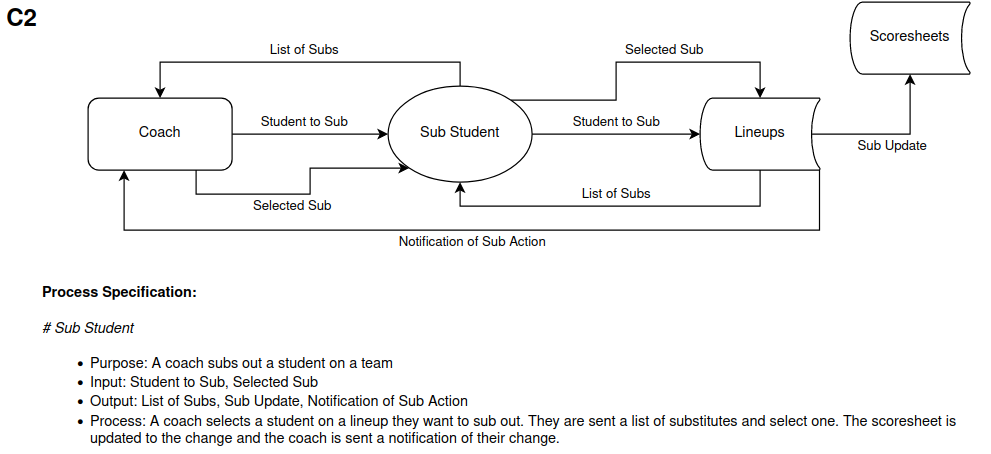
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | A13 |  |  | |
| **Name** | Account Management | | | |
| **Primary Actor** | Admin | | | |
| **Other Actors** |  | | | |
| **Description** | Admin is able to delete accounts that are no longer being used, emptying space in the database. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Admin clicks on the account management button. | |  | |
|  | | 2) Page displays a list of all the accounts. | |
| 3) Admin selects account(s) to be deleted and presses delete. | |  | |
|  | | 4) Confirmation page is displayed. | |
| 5) Admin confirms deletion | |  | |
|  | | 6) Accounts and related information are removed from the database. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to delete accounts. | Preconditions | Admin is logged in and other accounts exist. |
| **After Event** | Conclusion | Accounts are deleted. | Postconditions | Accounts and related information deleted from database. |



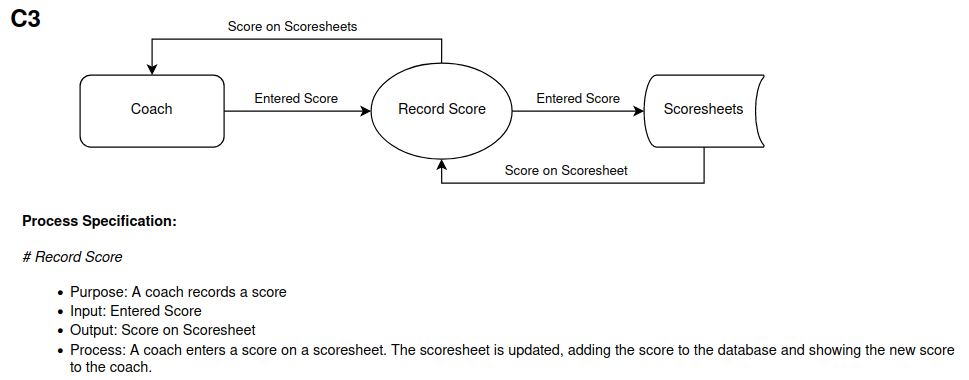
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | C1 |  |  | |
| **Name** | Team Registration | | | |
| **Primary Actor** | Coach | | | |
| **Other Actors** | Admin | | | |
| **Description** | Coach wants to submit a team to join a quiz-off. The coach submits a team and receives a notification of acceptance status. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Coach clicks on the team registration button. | |  | |
|  | | 2) Team registration request form is displayed. | |
| 3) Coach fills in the team registration form and submits. | |  | |
|  | | 4) Team registration information is added to the database and a request is sent to Admin. | |
|  | | 5) Notification of request sent successfully to Coach. | |
| 6) Admin reviews registration and confirms. | |  | |
|  | | 7) Notification sent to Coach for successful team registration. | |
| **Alternative Event Flow** | 1) Coach clicks on the team registration button. | |  | |
|  | | 2) Team registration request form is displayed. | |
| 3) Coach fills in the team registration form and submits. | |  | |
|  | | 4) Team registration information added to the database and a request is sent to Admin. | |
|  | | 5) Notification of request sent successfully to Coach. | |
| 6) Admin reviews registration and denies. | |  | |
|  | | 7) Notification sent to Coach of denied team registration. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to register a team for the quiz-off. | Preconditions | Students on the team have accounts on the application and Coach clicks on the team registration button. |
| **After Event** | Conclusion | Team registered for the quiz-off | Postconditions | Registered team is added to the database. |



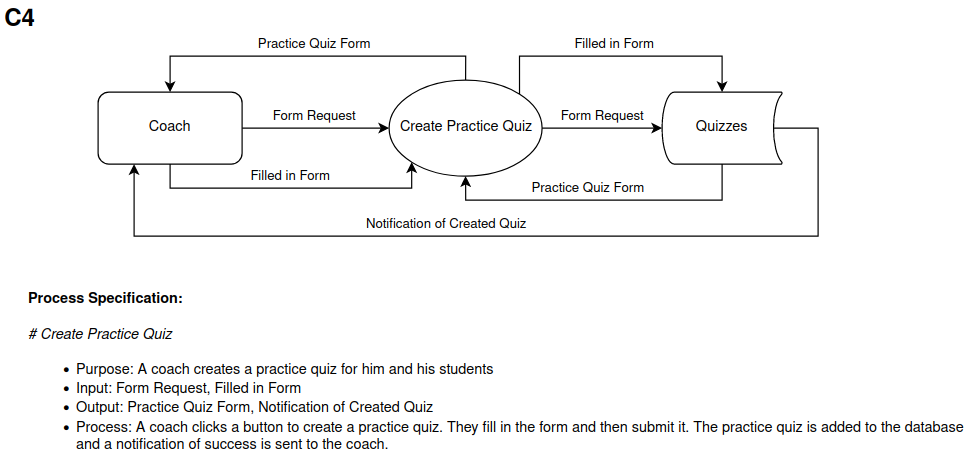
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | C2 |  |  | |
| **Name** | Team Substitutes | | | |
| **Primary Actor** | Coach | | | |
| **Other Actors** |  | | | |
| **Description** | Coach wants to substitute a team member and accesses the team lineup to swap in a substitute. The system updates the scoresheets and database. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Coach views the team lineup to make a substitute on. | |  | |
| 2) Coach clicks on the student to sub out. | |  | |
|  | | 3) Page displays possible substitutes for the student. | |
| 3) Coach selects a substitute. | |  | |
|  | | 4) Confirmation of action is sent to Coach. | |
|  | | 5) Database and scoresheets are updated about the substitute accordingly. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to make a substitute on the team. | Preconditions | Team lineup page is open. |
| **After Event** | Conclusion | A substitute on the team is made. | Postconditions | Team lineup is changed and all scoresheets being tracked are adjusted. |



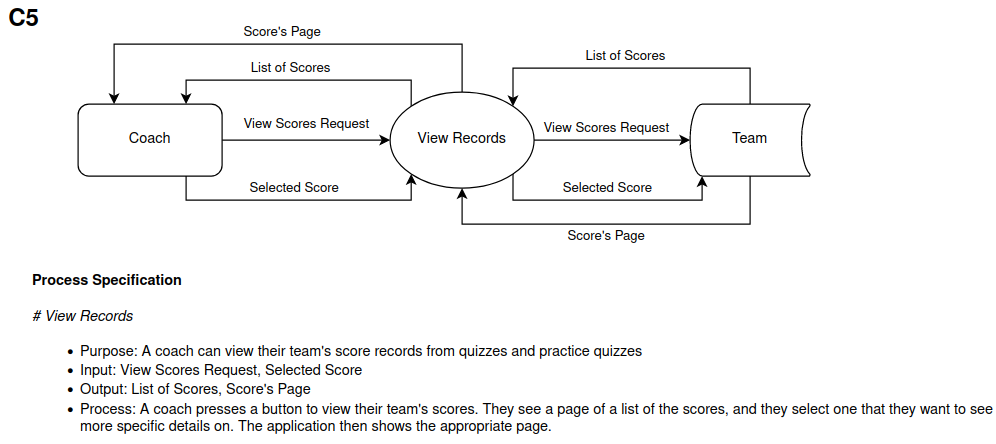
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | C3 |  |  | |
| **Name** | Record Scores | | | |
| **Primary Actor** | Coach | | | |
| **Other Actors** |  | | | |
| **Description** | Coach records score during quiz-off to maintain score accountability and accuracy. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Coach clicks on the cell on the scoresheet corresponding to the student answering and question number. | |  | |
| 2) Coach selects the answer type. | |  | |
|  | | 3) Score updated on the scoresheet and stored in the database. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to record a score. | Preconditions | Quiz-off or practice quiz is started. Scoresheet is open. |
| **After Event** | Conclusion | Score is recorded. | Postconditions | Score is added to the database and shown on the scoresheet. |



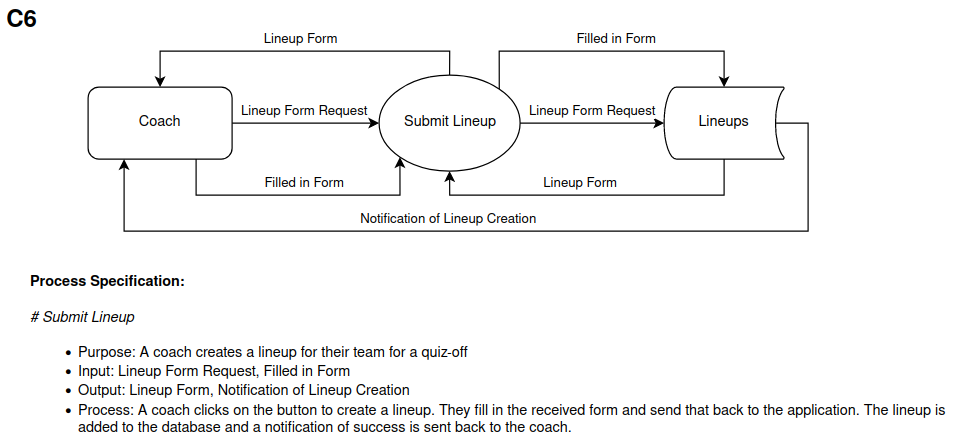
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | C4 |  |  | |
| **Name** | Practice Quizzes | | | |
| **Primary Actor** | Coach | | | |
| **Other Actors** |  | | | |
| **Description** | Coach creates a practice quiz to use for his team on the application. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Coach clicks on the create practice quiz button. | |  | |
|  | | 2) Practice quiz form is displayed. | |
| 2) Coach fills in and submits the practice quiz form. | |  | |
|  | | 3) Practice quiz is added to the database and available to the Coach on the practice quiz page. | |
|  | | 4) Notification of success sent to Coach. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to train a team. | Preconditions | Coach is logged in and viewing the practice quiz page. |
| **After Event** | Conclusion | Practice quiz is created. | Postconditions | Practice quiz added to the database and is accessible to the Coach. |



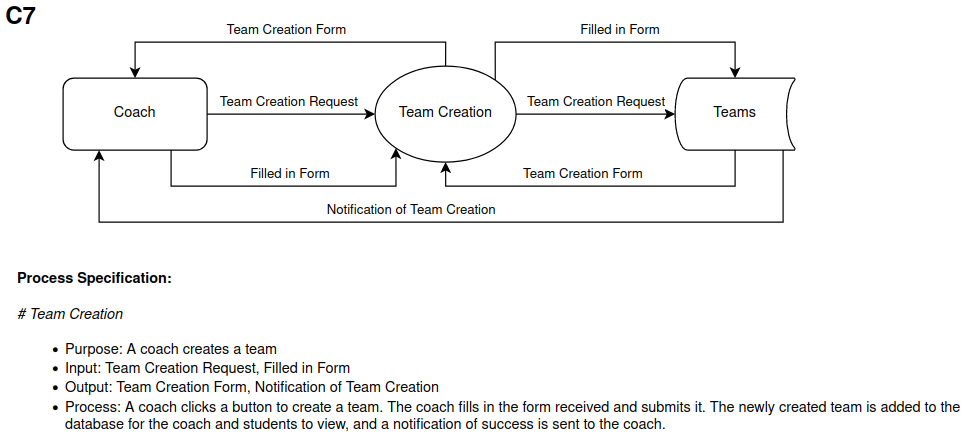
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | C5 |  |  | |
| **Name** | View Score Records | | | |
| **Primary Actor** | Coach | | | |
| **Other Actors** |  | | | |
| **Description** | Coach views previous or active score records of his team(s), which also shows the general statistics of his students. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Coach clicks on the view records button. | |  | |
|  | | 2) Page displays a list of records available to view. | |
| 3) Coach selects the desired record. | |  | |
|  | | 4) Page displays the desired record and statistics with it. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to view score records. | Preconditions | At least 1 record that the Coach can access is stored in the database. Coach is logged in. |
| **After Event** | Conclusion | Score records are viewed. | Postconditions | Score record is displayed on the screen. |



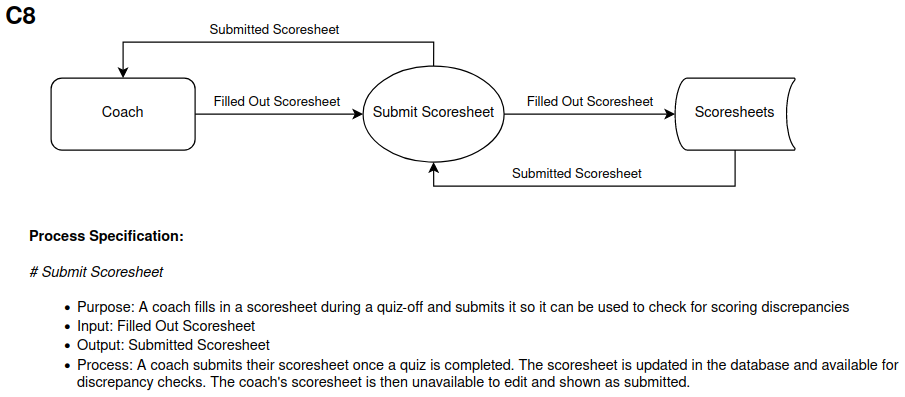
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | C6 |  |  | |
| **Name** | Submit Student Lineup | | | |
| **Primary Actor** | Coach | | | |
| **Other Actors** |  | | | |
| **Description** | Coach creates and submits a team lineup for a quiz-off. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Coach clicks the button to create a lineup. | |  | |
|  | | 2) Page displays lineup form. | |
| 3) Coach fills in and submits the lineup form. | |  | |
|  | | 4) Lineup is added to the database and is accessible to the team, Coach and Quizmaster. | |
|  | | 5) Notification sent to Coach that lineup creation is successful. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to submit a student lineup. | Preconditions | All students for the lineup exist in the database. Coach is viewing the team he wants to create a lineup for. |
| **After Event** | Conclusion | Student lineup submitted. | Postconditions | Quiz data relating to the lineup updated. Notification sent to the Coach that the lineup was submitted successfully. |



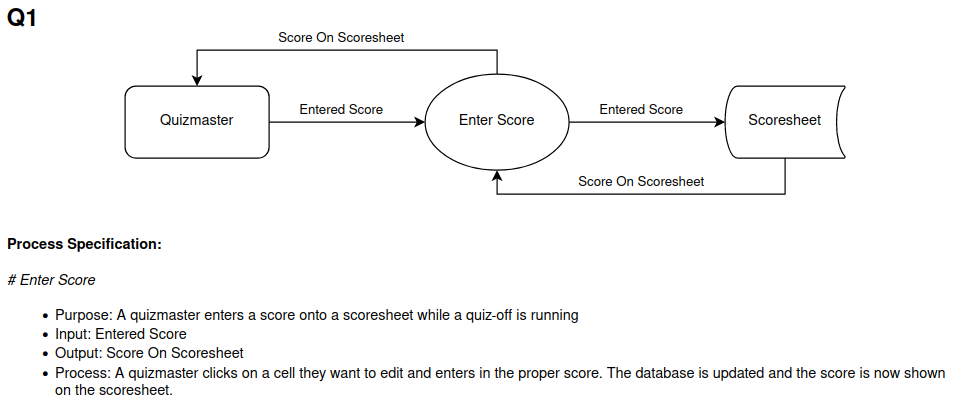
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | C7 |  |  | |
| **Name** | Team Creation | | | |
| **Primary Actor** | Coach | | | |
| **Other Actors** |  | | | |
| **Description** | Coach creates a team from scratch with input from the team requests students give. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Coach clicks on the create team button. | |  | |
|  | | 2) Page displays the team creation form and any student requests for team creation. | |
| 3) Coach fills out and submits the form to create teams, using autofill data from student requests if desired. | |  | |
|  | | 4) Team created and added to the database, letting students and the Coach access the team information and records. | |
|  | | 5) Notification sent to Coach on successful team creation. | |
|  | | 6) Page displays newly created team page. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to create a team. | Preconditions | All students to be part of the team are in the database. Coach is logged in. |
| **After Event** | Conclusion | Team is created. | Postconditions | Team is created and added to the database. Students and Coach have access to team records. |



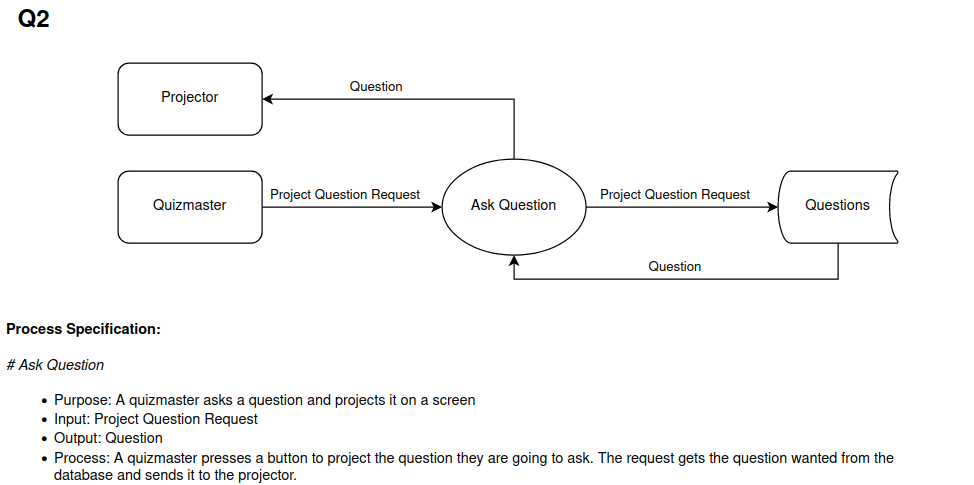
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | C8 |  |  | |
| **Name** | Scoresheet Discrepancy Check | | | |
| **Primary Actor** | Coach | | | |
| **Other Actors** |  | | | |
| **Description** | Coach submits scoresheet so that the quizmaster may use them to check for scoring discrepancies and fix them. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Coach clicks on the submit button on the scoresheet. | |  | |
|  | | 2) Scoresheet is added to the database. | |
|  | | 3) Scoresheet is available to be used in the discrepancy check when Quizmaster starts calculating the final quiz score. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to have correct scores. | Preconditions | Coach is working on a finished scoresheet. |
| **After Event** | Conclusion | Scores are able to be checked. | Postconditions | Scoresheet is submitted to the database and sent to the Quizmaster. |



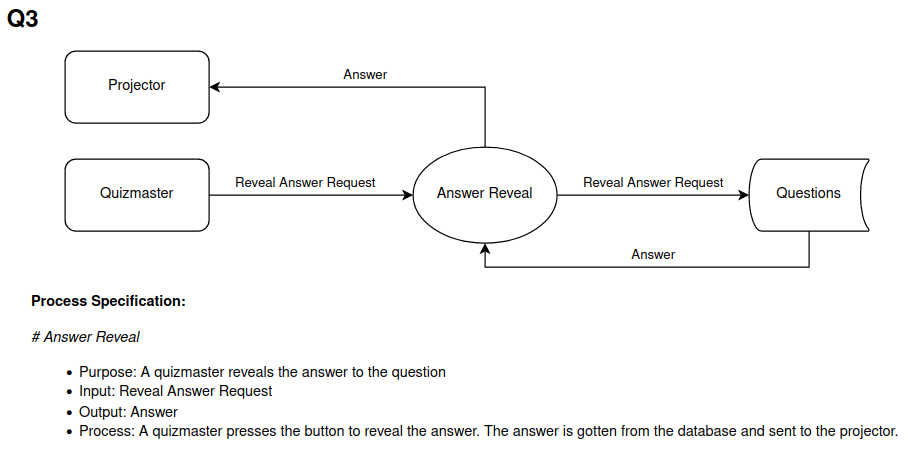
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | Q1 |  |  | |
| **Name** | Score Entry | | | |
| **Primary Actor** | Quizmaster | | | |
| **Other Actors** |  | | | |
| **Description** | Quizmaster enters a score into the application while a quiz-off is going on. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Quizmaster clicks on the cell on the scoresheet corresponding to student answering and question number. | |  | |
| 2) Quizmaster selects the answer type. | |  | |
|  | | 3) Score updated on the scoresheet and stored in the database. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to enter a score. | Preconditions | Quiz-off seeding or bracket phase is ongoing and a scoresheet is open. |
| **After Event** | Conclusion | Score is recorded. | Postconditions | Score recorded in the database and displayed on the scoresheet. |



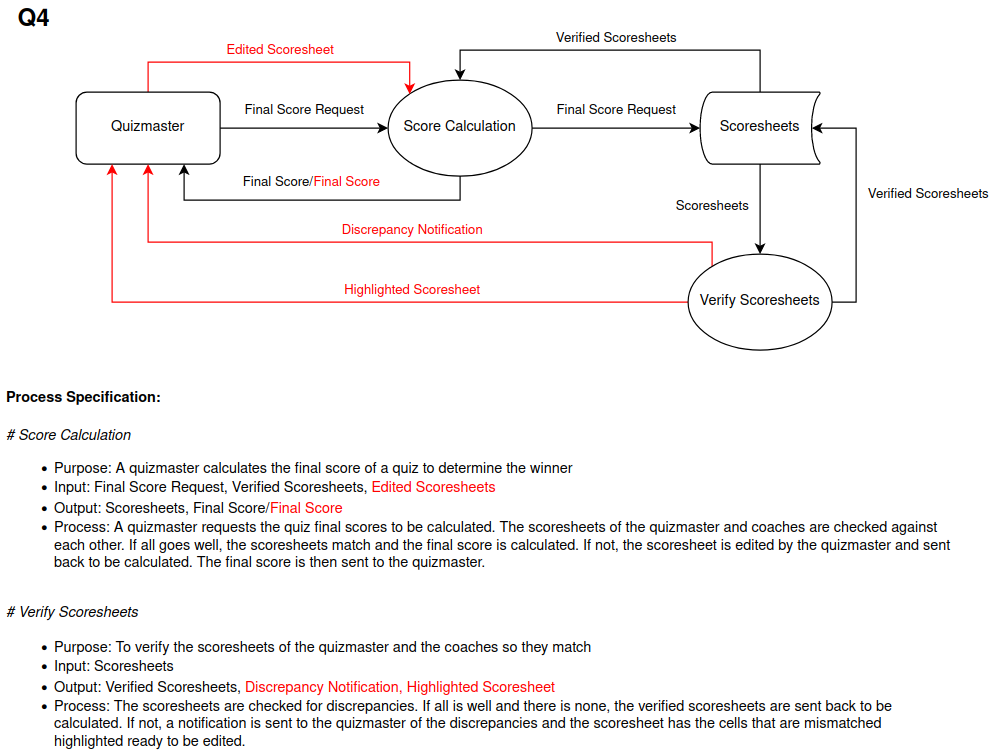
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | Q2 |  |  | |
| **Name** | Ask a Question | | | |
| **Primary Actor** | Quizmaster | | | |
| **Other Actors** |  | | | |
| **Description** | Quizmaster asks a question and has the same question displayed on the screen. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Quizmaster clicks the button to project a question. | |  | |
|  | | 2) Chosen question is projected on the screen. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to ask a question. | Preconditions | Quiz-off seeding or bracket phase is started. Questions page is clicked on. |
| **After Event** | Conclusion | Question asked. | Postconditions | Question is displayed on a screen. |



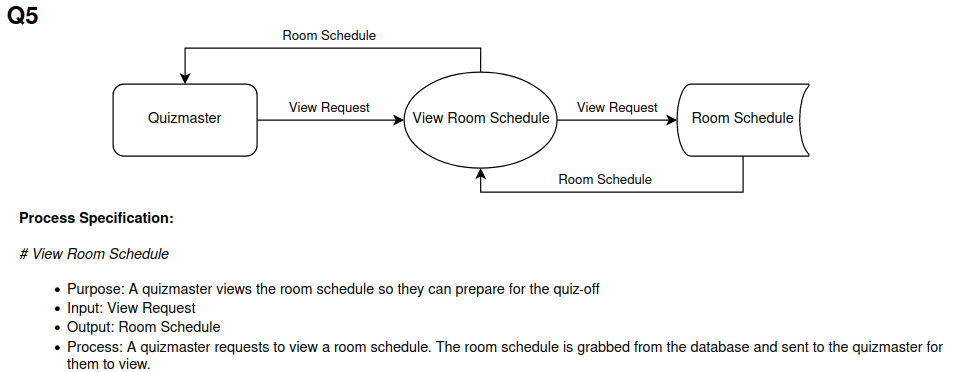
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | Q3 |  |  | |
| **Name** | Verify Answer | | | |
| **Primary Actor** | Quizmaster | | | |
| **Other Actors** | Student | | | |
| **Description** | Quizmaster verifies the answer of a Student and displays a possible correct answer on the projected screen. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Student answers a question. | |  | |
| 2) Quizmaster reveals the answer by pressing reveal answer button. | |  | |
|  | | 3) Projected screen shows the answer to the question. | |
| **Alternative Event Flow** | 1) Student answers a question. | |  | |
| 2) Student’s question is contested. | |  | |
| 3) Contestation of Student’s answer is concluded. | |  | |
| 4) Quizmaster reveals the answer by pressing the reveal answer button. | |  | |
|  | | 5) Projected screen shows the answer to the question. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to verify a Student’s answer. | Preconditions | Question is asked and is displayed on a screen. |
| **After Event** | Conclusion | Student’s answer is verified and points are awarded. | Postconditions | Correct answer is displayed on screen. Points are recorded and stored in the database and displayed on the scoresheet. |



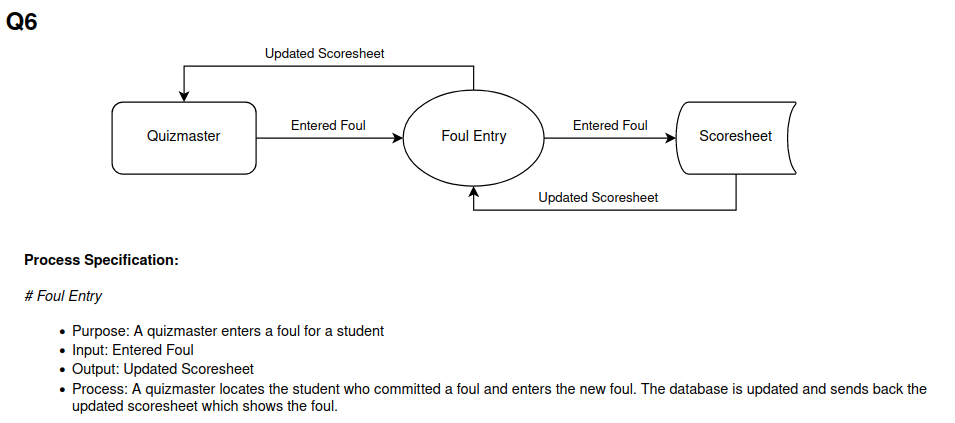
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | Q4 |  |  | |
| **Name** | Calculate Final Score | | | |
| **Primary Actor** | Quizmaster | | | |
| **Other Actors** | Coach | | | |
| **Description** | Quizmaster has the application calculate the final score for the quiz, which also checks for scoring discrepancies. If there are discrepancies, quizmasters edit the highlighted cells and discuss with coaches. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Quizmaster clicks calculate final score button. | |  | |
|  | | 2) Application checks that Coaches’ scoresheets match quizmaster’s scoresheet. | |
|  | | 3) Application calculates the final score and displays it on screen. | |
| **Alternative Event Flow** | 1) Quizmaster clicks calculate final score button. | |  | |
|  | | 2) Application checks that Coaches’ scoresheets match quizmaster’s scoresheet. | |
|  | | 3) Quizmaster notified of discrepancies and the scoresheet has highlighted cells for mismatched scores. | |
| 4) Quizmaster and Coaches discuss discrepancies. | |  | |
| 5) Quizmaster edits cells and clicks on calculate final score button. | |  | |
|  | | 6) Application calculates the final score and displays it on screen. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to calculate final quiz score. | Preconditions | Quiz-off seeding or bracket phase quiz ending. Scoresheets filled out for both teams. |
| **After Event** | Conclusion | Final quiz scores for both teams calculated and seen. | Postconditions | Final quiz score for both teams calculated and displayed. Scores and scoresheets stored in the database. |



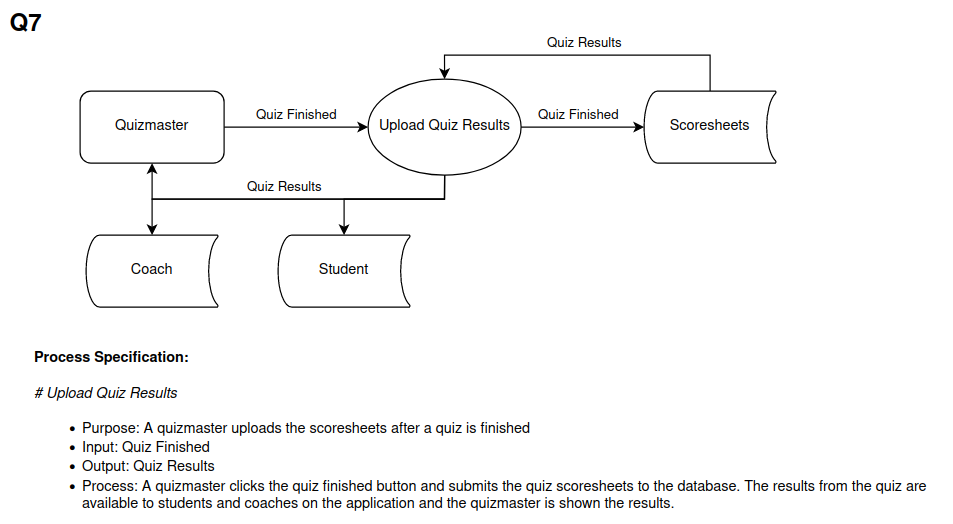
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | Q5 |  |  | |
| **Name** | View Room Schedule | | | |
| **Primary Actor** | Quizmaster | | | |
| **Other Actors** |  | | | |
| **Description** | Quizmaster views the room schedule that the admin has created. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Quizmaster clicks on the view room schedule button. | |  | |
|  | | 2) Room schedule is displayed on the page. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to view room schedule. | Preconditions | Room schedule has been created and is accessible to Quizmasters. Scheduling page is displayed to Quizmaster. |
| **After Event** | Conclusion | Room schedule viewed. | Postconditions | Room schedule displayed on screen. |



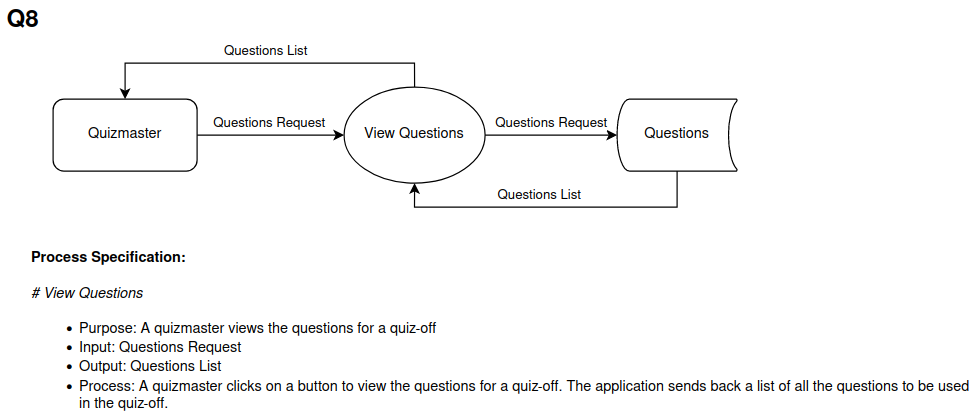
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | Q6 |  |  | |
| **Name** | Enter a Foul | | | |
| **Primary Actor** | Quizmaster | | | |
| **Other Actors** |  | | | |
| **Description** | Quizmaster has noted that a foul occurred during a quiz-off and inputs the foul into the quiz scoresheet. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Quizmaster locates the person who got a foul on the scoresheet. | |  | |
| 2) Quizmaster navigates to the fouls row and increases the value by one by clicking a plus symbol next to the foul count. | |  | |
|  | | 3) Foul is added to database and effects on score is calculated. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to enter a foul. | Preconditions | Quiz-off quiz is active and scoresheet is available. |
| **After Event** | Conclusion | Foul entered into the scoresheet. | Postconditions | Scoresheet has updated foul displayed and is recorded in the database. |



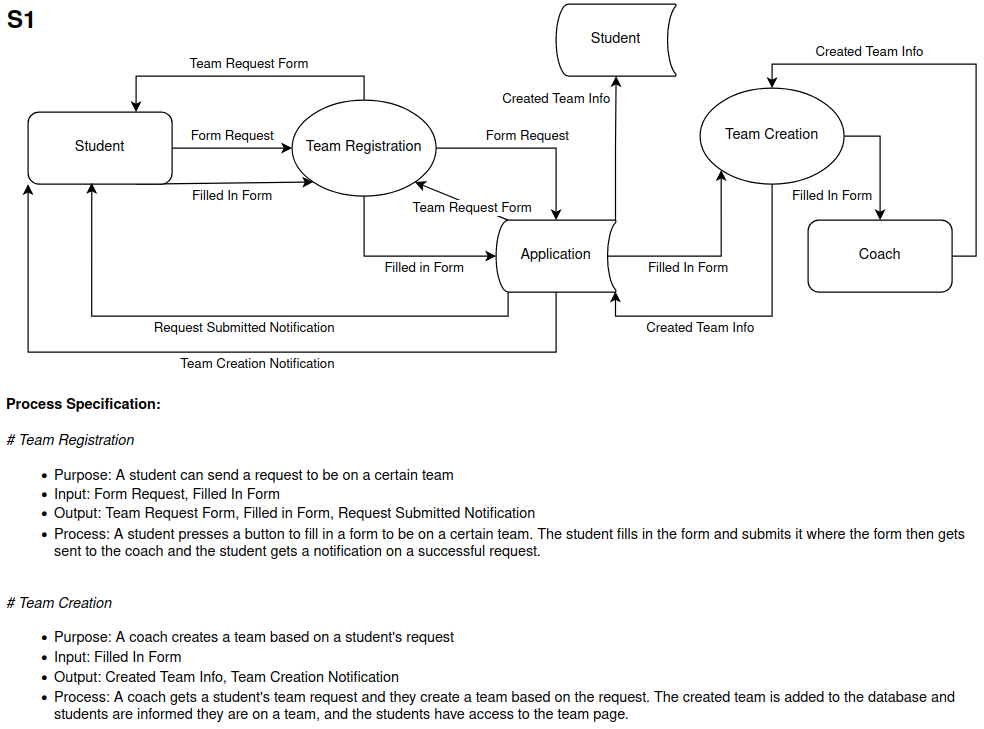
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | Q7 |  |  | |
| **Name** | Upload Quiz Results | | | |
| **Primary Actor** | Quizmaster | | | |
| **Other Actors** |  | | | |
| **Description** | Quizmaster has finished giving a quiz and uploads the final scoresheet to the database. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Quizmaster clicks quiz finished button. | |  | |
|  | | 2) Scoresheet data is added to the database. | |
|  | | 3) Scoresheet data is available to Students and Coaches. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to upload quiz results. | Preconditions | Finished quiz scoresheet from a quiz-off. |
| **After Event** | Conclusion | Quiz results uploaded and saved in the database for students and coaches to view. | Postconditions | Scoresheet recorded in database, databases containing student and team records are updated and are made accessible to students and Coaches. |



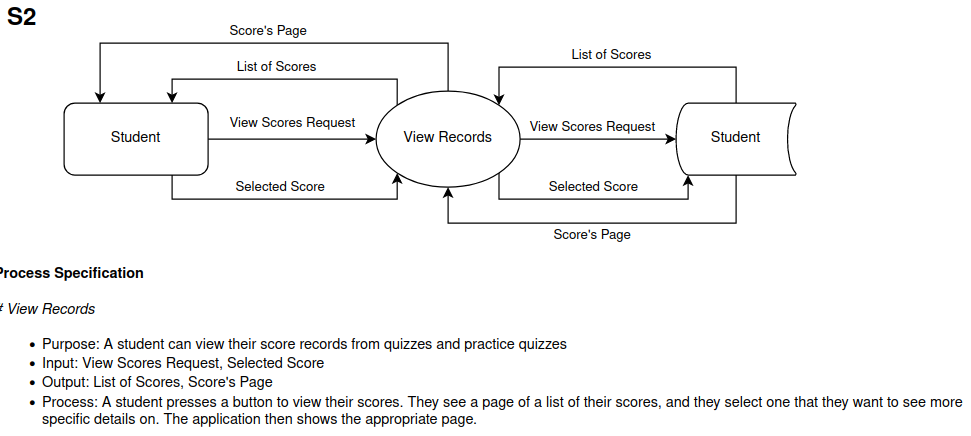
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | Q8 |  |  | |
| **Name** | View Quiz Questions | | | |
| **Primary Actor** | Quizmaster | | | |
| **Other Actors** |  | | | |
| **Description** | Quizmaster looks at the quiz questions which an Admin has chosen and sent out for the quiz-off. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Quizmaster clicks on the button to view quiz-off questions. | |  | |
|  | | 2) Page displays a list of quiz-off questions the Admin created and chose. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to view quiz questions. | Preconditions | Admin must have uploaded the quiz questions for the quiz-off for Quizmasters to view. Quizmasters are logged in. |
| **After Event** | Conclusion | Quiz questions viewed. | Postconditions | Quiz questions displayed on screen. |



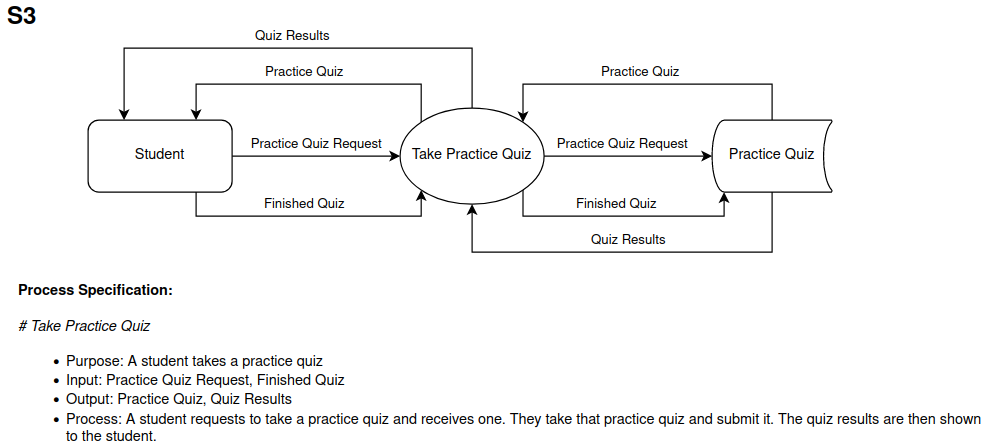
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | S1 |  |  | |
| **Name** | Team Registration Request | | | |
| **Primary Actor** | Student | | | |
| **Other Actors** | Coach | | | |
| **Description** | Student sends a request to be put on a team to their coach. Coach adds them to a team. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Student clicks on the team request button. | |  | |
|  | | 2) Page displays form for a team request. | |
| 3) Student fills in and submits the request form. | |  | |
|  | | 4) Notification is sent to student that the request was successful. | |
| 5) Coach creates a team that the student is on. | |  | |
|  | | 7) Database is updated and the student is on a team. The student’s page displays their team and gives access to its records. | |
|  | | 8) Notification sent to Student that they are on a team | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to register for a team. | Preconditions | Student has an account and is logged in. |
| **After Event** | Conclusion | Students are assigned to a team. | Postconditions | Student is on a team. Information is stored in the database and students have access to team records. |



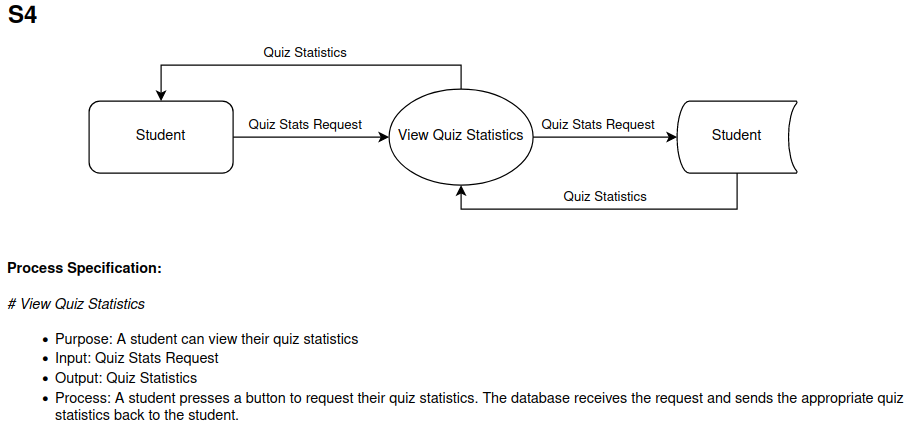
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | S2 |  |  | |
| **Name** | View Score Records | | | |
| **Primary Actor** | Student | | | |
| **Other Actors** |  | | | |
| **Description** | Student views their score records on the application | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Student clicks on the button to view their score records. | |  | |
|  | | 2) Page displays a list of score records. | |
| 3) Student selects which score record to view. | |  | |
|  | | 4) Page displays the desired score record. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to view score records. | Preconditions | Student has at least one score record in the database. Student is logged in. |
| **After Event** | Conclusion | Desired score record viewed. | Postconditions | Score record is displayed on the screen. |



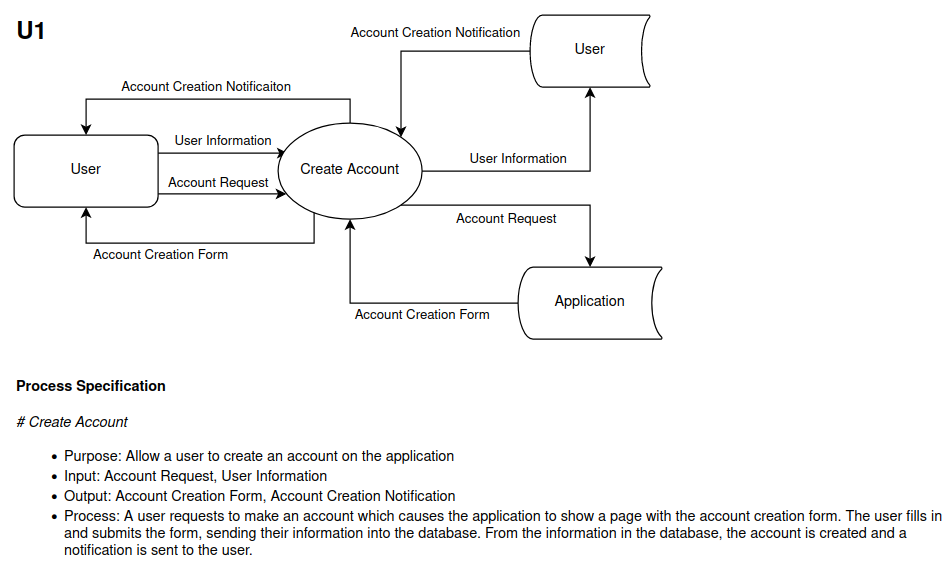
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | S3 |  |  | |
| **Name** | Take Practice Quiz | | | |
| **Primary Actor** | Student | | | |
| **Other Actors** |  | | | |
| **Description** | Student uses the application to take a practice quiz created by their coach, which tells them their statistics afterwards. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Student clicks on the practice quiz button. | |  | |
|  | | 2) Page displays a list of practice quizzes. | |
| 3) Student selects a practice quiz. | |  | |
|  | | 4) Practice quiz is displayed. | |
| 5) Student fills out and submits the practice quiz. | |  | |
|  | | 5) Practice quiz results are stored in the database. | |
|  | | 6) Page displays quiz results. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to take a practice quiz. | Preconditions | Coach has created a practice quiz the student can access. Student is logged in. |
| **After Event** | Conclusion | Practice quiz taken and feedback received. | Postconditions | Practice quiz is finished and results and statistics are shown to the student. |



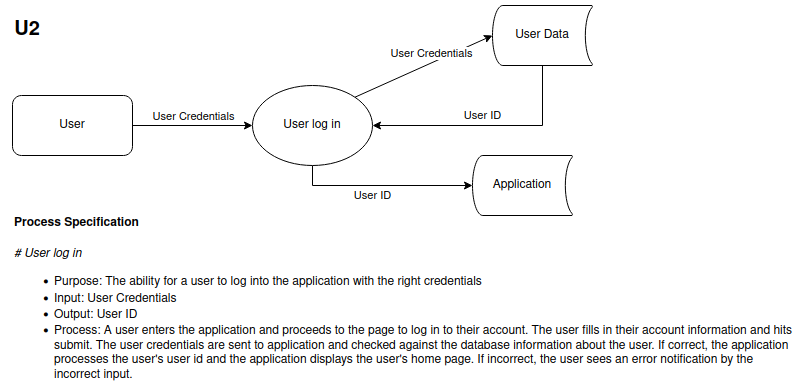
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | S4 |  |  | |
| **Name** | View Quizzing Statistics | | | |
| **Primary Actor** | Student | | | |
| **Other Actors** |  | | | |
| **Description** | Student views their overall statistics for the quizzes they have taken. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) Student presses the button to view quizzing statistics. | |  | |
|  | | 2) Quiz statistics generated and displayed on the page. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to view quiz statistics. | Preconditions | Student has an account and has taken at least one or more quizzes. Student is logged in. |
| **After Event** | Conclusion | Quiz statistics page is viewed. | Postconditions | Student quiz statistics displayed on screen. |



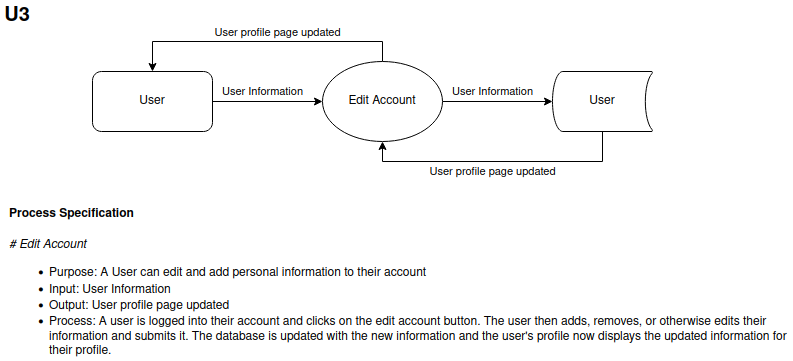
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | U1 |  |  | |
| **Name** | Create Account | | | |
| **Primary Actor** | User | | | |
| **Other Actors** |  | | | |
| **Description** | A user creates an account so they can use the application associated with the quiz-offs. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) User clicks create account button. | |  | |
|  | | 2) Page displays account creation form. | |
| 3) User fills in and submits the account creation form. | |  | |
|  | | 4) Database updated with the new account. | |
| **Alternative Event Flow** | 1) User clicks create account button. | |  | |
|  | | 2) Page displays account creation form. | |
| 3) User fills in and submits the account creation form. | |  | |
|  | | 4) Page displays errors and notifies them that account creation was not successful. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to create an account. | Preconditions | User navigates to the account creation page. |
| **After Event** | Conclusion | Student account created. | Postconditions | User account is added to the database. |



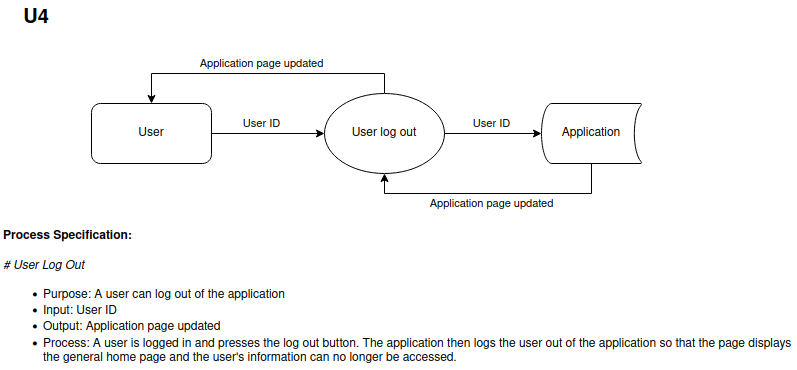
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | U2 |  |  | |
| **Name** | Log in | | | |
| **Primary Actor** | User | | | |
| **Other Actors** |  | | | |
| **Description** | A user can log in to their account. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) User clicks on the log in button. | |  | |
|  | | 2) Page displays form for logging in. | |
| 3) User fills in and submits the form with log in information. | |  | |
|  | | 4) Page displays the user’s home page. | |
| **Alternative Event Flow** | 1) User clicks on the log in button. | |  | |
|  | | 2) Page displays form for logging in. | |
| 3) User fills in and submits the form with log in information. | |  | |
|  | | 4) Page displays errors and notifies the user that the login was not successful. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to log in. | Preconditions | User account exists and has navigated to the login page. |
| **After Event** | Conclusion | User is logged in. | Postconditions | User is logged in and sees their information, |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | U3 |  |  | |
| **Name** | Input Information | | | |
| **Primary Actor** | User | | | |
| **Other Actors** |  | | | |
| **Description** | User adds personal information to their account | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) User clicks on the edit profile button. | |  | |
|  | | 2) Editable user profile page displayed | |
| 3) User adds and submits the information they desire. | |  | |
|  | | 4) User information added to the database and shown on their profile. | |
| **Alternative Event Flow** | 1) User clicks on the edit profile button. | |  | |
|  | | 2) Editable user profile page displayed | |
| 3) User adds and submits the information they desire. | |  | |
|  | | 4) Page displays errors and unmet conditions. Data is not submitted. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to input information. | Preconditions | User is logged in and is looking at their profile. |
| **After Event** | Conclusion | Information is added into the application. | Postconditions | User information added to the database. |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | U4 |  |  | |
| **Name** | Log out | | | |
| **Primary Actor** | User | | | |
| **Other Actors** |  | | | |
| **Description** | A User can log out of their account. | | | |
|  | **Actor Action** | | **System Response** | |
| **Typical Event Flow** | 1) User presses the log out button. | |  | |
|  | | 2) User information no longer accessible, page displays the main page for account creation and logging in. | |
|  | **Business Focus** | | **System Focus** | |
| **Before Event** | Trigger | Desire to log out. | Preconditions | User is logged in. |
| **After Event** | Conclusion | User logged out. | Postconditions | User information is no longer accessible. |



2.5 - Wireframes

Wireframes showcase functionality and raw user experience outside of any graphics or color that may be used to enhance a design. Because the FEC Bible Quizzing System is intended to be built as a web application, these wireframes correspond to the pages users might see while viewing the site on a desktop device. Mobile compatibility is planned but not implemented in these wireframes.

**General: Homepage – Logged Out**

**A screenshot of a computer

Description automatically generated**

The standard view for any user who is not logged into the web application.

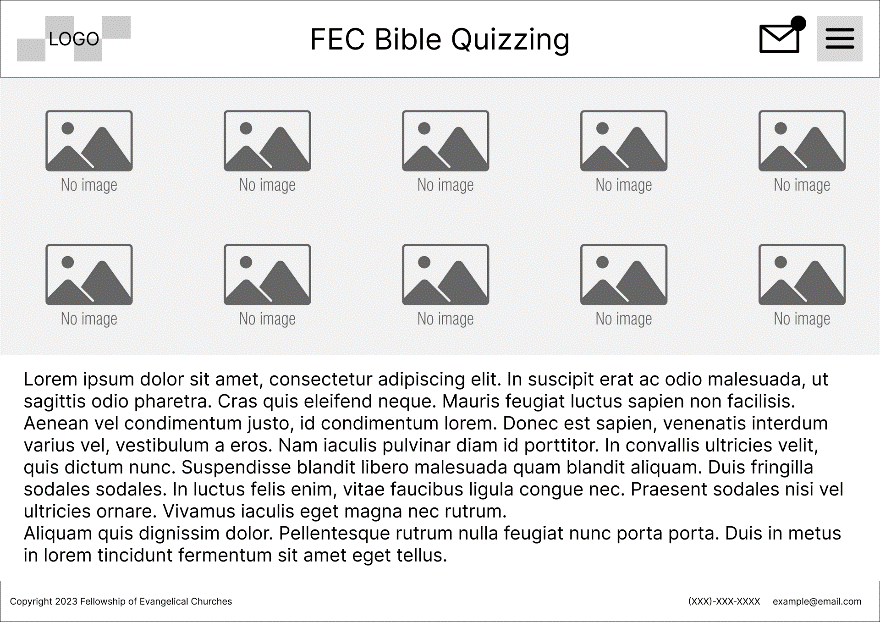
**General: Login Page**

A screenshot of a computer

Description automatically generated

Upon clicking the “Log In” button, users will be directed to log in using their credentials.

**General: Homepage – Logged In**



The homepage when logged in, featuring the Inbox and Menu buttons.

**Admin: Homepage – Menu Open**

A screenshot of a quiz

Description automatically generated

The homepage when logged in, featuring the menu for site navigation. Menu options will vary depending on user type.

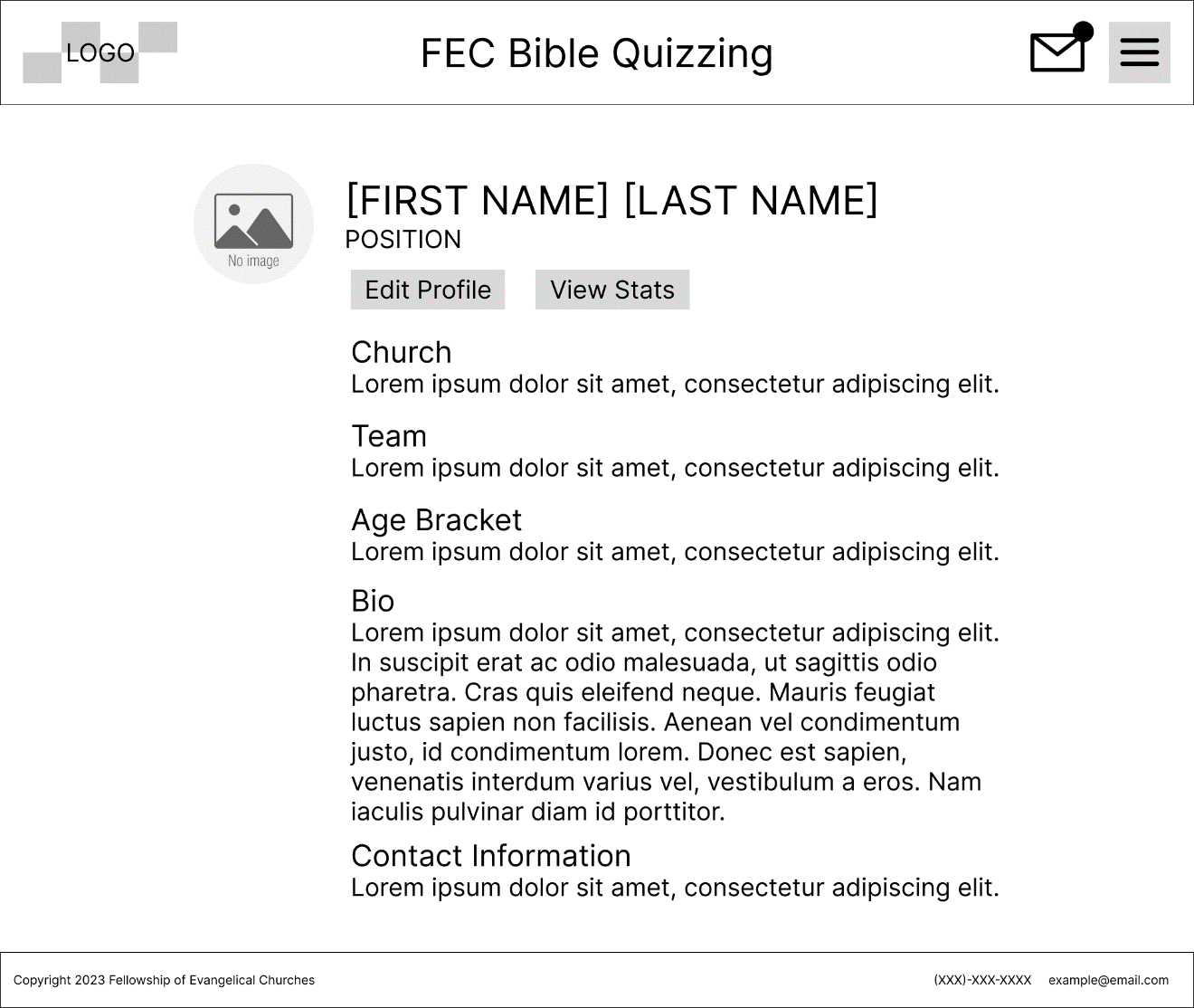
**Admin: Homepage – Inbox Open**

A screenshot of a bible quizzing

Description automatically generated

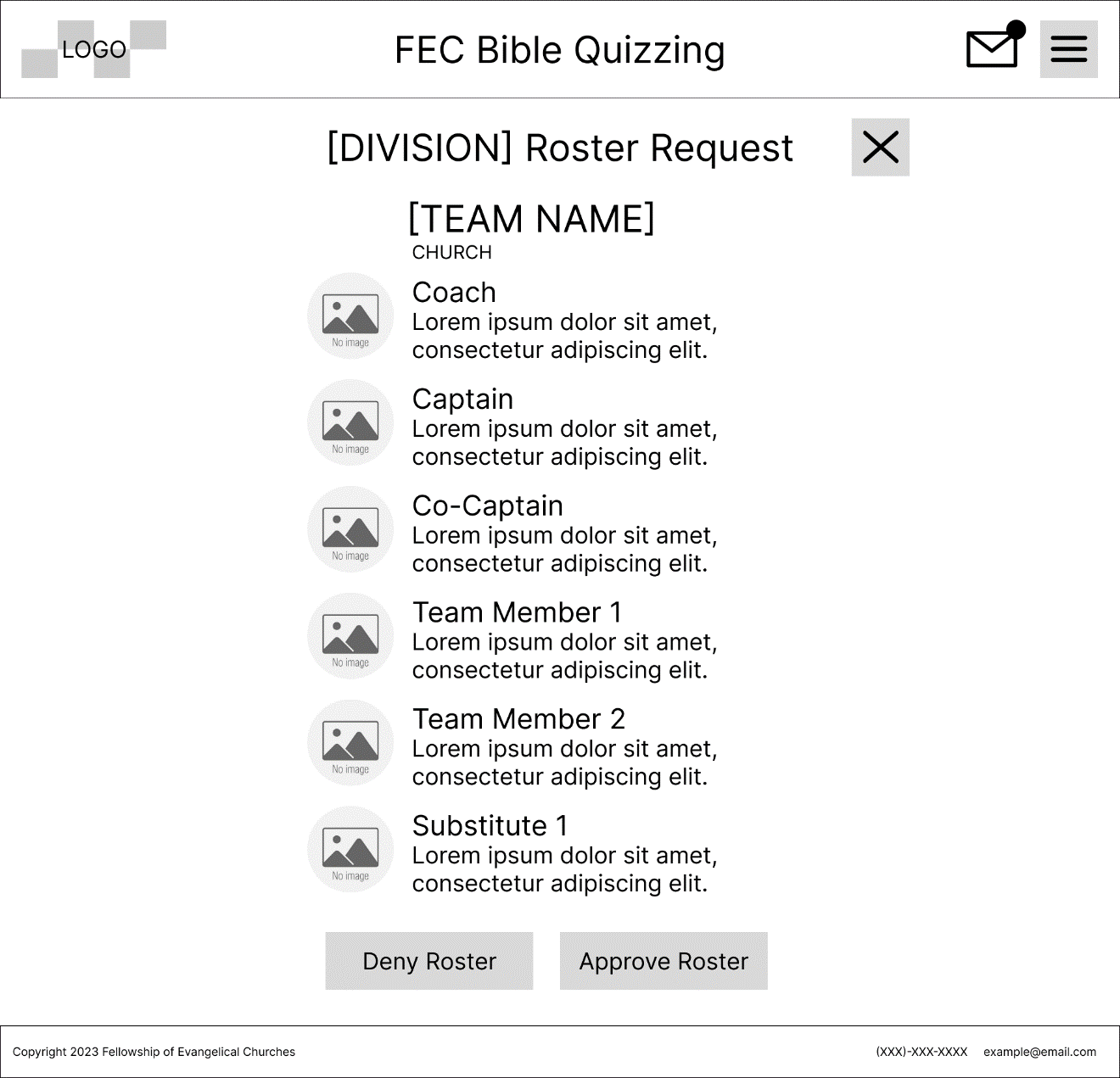
The homepage when logged in, featuring a quick view of new messages in a user’s inbox. The full Inbox page will contain messages, roster request forms, and other communications. Inbox contents will vary depending on user type.

**General: User Page**



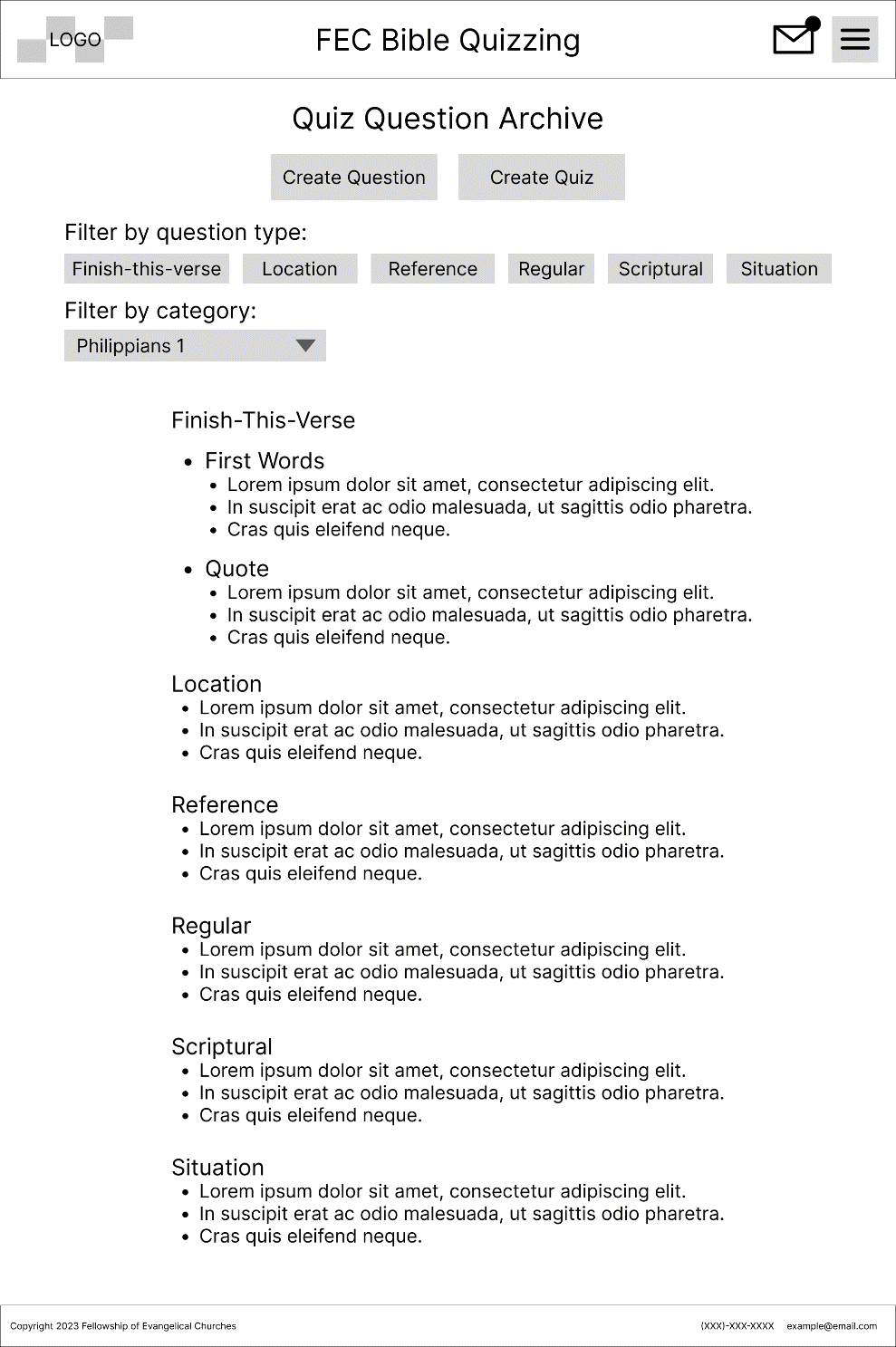
A user’s view of their own profile page. Other users will not be able to view edit profile buttons. Profile page fields will change depending on user type.

**Admin: View Roster Request Form**



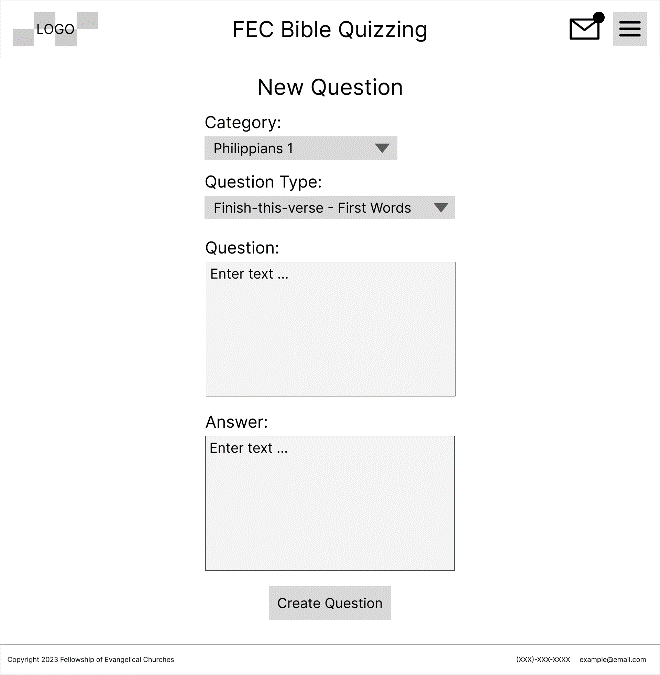
An admin’s view of a roster request submitted by a coach, accessed from the inbox. The user pages of each team member are linked in the roster, and admins are able to approve or deny the roster on this page (additional field for optional feedback will be made visible upon rejection).

**Admin: Quiz Question Archive**



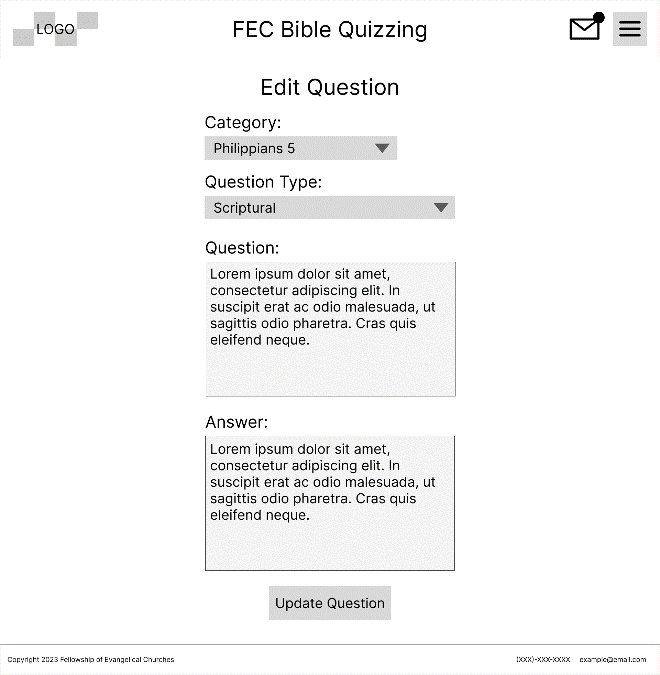
A full list of the questions to be used in **Quiz-offs**. Allows Admins to create new questions or new quizzes using preexisting questions directly. Contains filter options to allow for easy access to specific question types and content.

**Admin: Create Question**



Admins are able to create a question for a specific content category and question type. Question contents and accepted answer(s) have fields for manual entry.

**Admin: Edit Question**



Admins are able to edit all attributes of a previously created question.

**3 - Non-Functional Requirements**

3.1 - Usability

3.1.1 - Look and Feel

Throughout the development process, we would like to keep in mind three key principles for our look and feel: simplicity, elegance, and minimalism. It is important that our application is not overly complex or unnecessarily flashy while still containing necessary functionality. These three principles will help foster our goals of eliminating human error, increasing efficiency, and decentralizing the system knowledge base. User experience research will be conducted with different designs to ensure that the final system’s look and feel is minimal while remaining fully usable by system users.

To eliminate human error, a simple design would make the most sense for the client’s use case. Complicated, flashy designs without proper user experience research tend to produce beautiful designs with terrible functionality. We would like to avoid this common pitfall.

3.1.2 - Ease of Use

Our solution would require internet access to use, which is a reasonable requirement for this system considering the venues that most often host **quiz-offs**, churches and colleges. The system should be easy and intuitive to navigate through, log in and out of, and enter data. Buttons should be clearly labelled, input fields should have intuitive placeholder texts, and visual system components should be laid out in a way that makes sense to someone who has never used the system before. General users should not be required to have full knowledge of the rulebook or the quizzing system to be capable of using the FEC Bible Quizzing web application.

3.2 - Accessibility

3.2.1 - Cultural or Language Groups

The initial development of this application should use English as the primary language. This decision is based on the FEC’s lack of desire to grow the program beyond existing membership. Should the FEC decide to expand the Bible Quizzing program into regions of the world that do not speak English as a native language or decide that there is a significant need for another language to be incorporated, the necessary changes will be included in a later round of development.

3.2.2 - Users with Disabilities

This software will be created for mobile phones as well as for desktop computers. The following procedures should be followed to ensure that the software is built to maximally accommodate disabled system users.

* **Keyboard navigation**: The system should be easily navigable by keyboard using keyboard shortcuts.
* **Screen reader compatibility**: All images should have alternative texts when added to the web application, and any non-text content must have a textural explanation coded in so that screen readers will be able to parse the content and read it out loud to the visually impaired.
* **Contrast and color choices**: Color combinations that will be used should have sufficient contrast. When color is used to designate an action (such as a button being red or green), there should be at least one other visual indicator other than color, such as shape or accompanying text. This accommodates colorblind or visually impaired users.
* **Text Size**: The text should be a consistent size for headers and body paragraph text. The text should be large enough that everyone should be able to read it, erring on the side of larger text for maximum accessibility.
* **Mobile Compatibility:** Buttons should be sufficiently large so that all human users are able to tap them accurately on a mobile screen.
* **Error handling**: All displayed error messages should clearly communicate to users what the problem is, and how it may be solved.

3.3 - Availability

3.3.1 - Reliability and Availability

The system should operate under industry standard uptimes. This means that the system should be available for use 99.9% of the time, taking 0.01% of the time for dependency maintenance or system outages.

3.3.2 - Downtime

The system must be created in such a way that a user does not have the ability to crash the system by entering faulty data.

Very few things will make the system go down. Things that would make the system go down include:

* **Power outages**: Because the devices accessing and maintaining the system must have battery charge or continuous power to operate, it is possible that a power outage or similar could prevent users from accessing the system.
* **Internet outage**: Because the system requires constant access to the internet to work, faulty internet connections could hinder system operations and decrease efficiency.
* **Dependency outage**: It is possible that one of the systems integrated with the revised FEC Bible Quizzing System (database, user authentication service, etc.) could experience an outage. In this event, certain functions of the Bible Quizzing application would break down. This is rare and would only happen 0.01% of the total uptime. This means that every year, the system would not work for one hour on average.
* **System Failure**: It is possible that the development team could introduce bugs into the system that are not obvious until the system is live. In this case, the system would need to be reanalyzed, identified, and fixed before normal operations could continue.

3.4 - Documentation

While under development, the system components will be documented so that future developers can build on the work that will already have been done.

3.5 - Training

The system should be able to be used by an Administrator, Quizzer, Quizmaster, or Coach in conjunction with the FEC Bible Quizzing Rulebook, without any additional training. If a user has read the rulebook and understands their role at a quiz-off, they should be able to understand and use the system without training. Documentation will be available in case there are any specific functionality questions.

**4 - Non-Functional System Requirements**

4.1 - Performance

Performance refers to the speed at which technical operations are executed and resolved within the web application. These technical operations refer to:

* Data additions
* Data deletions
* Data updates
* Data deletions
* Loading a new page of the website

Each of these operations should take an absolute maximum of 2 seconds to complete. This is generally standard for a web application of this magnitude. Because there will not be much image manipulation, we can expect much faster page loads than on a website that has many images being displayed (i.e. YouTube or Google Images).

The system should be able to handle fluctuations in capacity without noticeable negative influence on system performance.

4.1.1 - Responsiveness

Responsiveness in a web application means that no matter what screen the user is viewing the content on, it will still show up in an easily readable and understandable format.

The web application should be fully responsive, and all users should be able to access all features, information, and systems.

Mobile and computer screens should be compatible with any existing smartphone, tablet, or computer resolution sizes. The screens should be responsive to user requests (i.e. if a user resizes a window, the page layout should be able to adjust and scale to the new window size).

4.1.2 - Concurrent Use

The system must be able to host multiple users concurrently. This is because, at any given point, many quizzes will be happening at the same time and many users will be accessing the system for various purposes. Concurrent operations should include updating database tables, creating database table entries, and reading database information.

4.2 - Capacity

4.2.1 - Current Capacity

Given the current number of churches and participants that participate in quiz-offs, the system will be able to functionally accommodate up to 5,000 total unique user accounts. All data will be maintained in online storage in the cloud.

Because the client currently has no plans for significant growth, 5,000 total user accounts will be the temporary maximum capacity for the FEC Bible Quizzing system.

The system should be able to handle fluctuations in capacity without noticeable negative influence on system performance.

4.2.2 - Future Growth

Significant future growth is not planned upon by the customer organization, so capacity increases will not be planned for the short-term or medium-term. It is worth noting that, should the system be required to scale in terms of user account maximum capacity, it would not be difficult to increase the maximum capacity of the system.

4.3 - Longevity

The system should be able to be fully independent and fully functional in the long term (5-10 years). Technology changes at a rapid pace, and special attention must be taken to maintain to the services and dependencies that are included in the system to optimize longevity.

4.4 - Security

Industry standard security practices should be used in the Bible Quizzing application.

* **Encryption** - All data transmitted throughout the application and the server should be encrypted using HTTPS.
* **Authentication –** Users should be able to validate their identity before logging in using an industry-standard practice (i.e. two-factor authentication). Passwords will be required to prevent malicious access to the system.
* **Access Control** - Users should be able to securely log in to their accounts. Each type of user (**quizzer**, **coach**, **quizmaster**, **administrator**) should be able to only view data that is pertinent to their user (see **Sections** **2.4.2** & **2.4.3**).
* **Input Validation** – Because this application involves significant amounts of data entry, proper care must be taken to ensure that no malicious inputs can cause the system to crash.
* **Error Handling –** All system errors should be appropriately handled to make sure that the system is secure 100% of the time.
* **Data Backups** – All system data should be regularly backed up to prevent unexpected or malicious data loss.