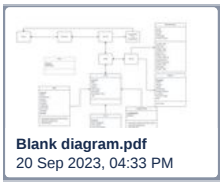
09/20/2023 In-Class Design Exercise

The A-Team

Team Members

- Daniel Tucker
- Christian Crawford
- Patrick Vergason
- Grant Scutt
- Carson Crysdale
- Benjamin Merritt
- Anthony Vandergriff

UML Diagram



Blank diagram.pdf
20 Sep 2023, 04:33 PM

Design Requirements

User Roles:

- **System Administrator** (Admin):
  - View all hunts and their respective status.
  - Create a hunt
  - Edit a hunt
- Unique username and password for access.

Trivia Types:

Requirements Analysis

Administrative:

- **Mandatory**
  - See all hunts and their status
  - Create a hunt with a start date, end date, title, theme, invitation creation, and assign tasks
  - Set task order

- Multiple Choice trivia.
- Single Word answer Trivia.
- Combination trivia formats with QR codes.

Task Presentation Formats:

- Fixed list.
- Random order (per access code).
- Incremental (tasks revealed upon completion of the previous one).

Task Features:

- Each task should have:
- A display label for participants.
- An answer (free text).
- Ability to scan and associate a QR Code with one or multiple tasks.
- The QR Code's associated text should be stored with the task.
- Answer matching to consider string matching, ignoring case and whitespace.

Admin Functionalities:

- Ability to create, edit, or delete tasks.
- Define how tasks are presented (choice between fixed list, random, or incremental).
- Define the start and end time for the hunt:
- Users trying to access outside this timeframe will receive an informative message.
- Edit the introduction text displayed at the beginning of the task list.
- Generation and management of a list of access codes for the hunt.
- Associate each access code with either an email or a phone number.
- Initiate notifications for selected access codes. This can be through an email and/or text message.

Hunt Features:

- There is only one hunt, and it's always active.
- The task list presentation is consistent for all participants.

Miscellaneous:

- A comprehensive error-handling system.

    - Assign URL for hunt
    - Edit existing hunt
    - Create an account using an email address and phone number
        - A player access code is unique to the individual hunt
    - Access codes can be active, disabled, or pending
- **Optional**
    - Able to invite one or more people to a specific hunt
    - Can only edit to pending or active hunts
    - Can edit any part of the hunt except status and creation date
    - If the admin changes the status to active, all players associated with the hunt will be notified by a text message
    - Able to quickly create multiple accounts from a list (no manual entry)

Player (User):

- Mandatory
    - Able to join a hunt
    - Have a unique access code for them as well as the hunt
    - The player enters the hunt access code on the URL page
    - Ease of recording completions
    - For camera-enabled players, a QR code is used for the hunt
    - For non-camera-enabled players, text the QR code to enable them to play
    - Can view the status of hunt
    - Leaderboard view access
- **Optional**
    - A player is able to team up with other players to form a group
        - They use the same access code, and the sessions must be refreshed
    - For location-enabled players, compare the player's location to the task location and allow for “I am here” to start the task.

- A secure database to store tasks, QR Code associations, access codes, and other relevant information.
- Integration capability with email and SMS services to send notifications.
- Responsive design to ensure accessibility from various devices.

Non-functional Requirements:

1. **Security**:

- Secure storage of admin username and password.
- Encryption of sensitive data, especially user emails and phone numbers.

1. **Performance**:

- The system should respond swiftly to user inputs and actions.
- Efficient database queries to ensure smooth functioning during high traffic.

1. **Scalability**:

- Even though there's only one hunt, the system should be designed to scale in the future if more hunts are added or if user traffic increases.

1. **Usability**:

- The admin interface should be user-friendly and intuitive.
- Participants should have a clear and simple user experience.

1. **Reliability**:

- Ensure that the system has minimal downtime and can recover quickly from potential failures.

1. **Maintainability**:

- Code should be well-documented and modular, allowing for easy future enhancements or fixes.

This list covers the functional and non-functional requirements for your application as described. Further details can be elaborated based on the technical platform you choose and any additional specifications or constraints.