Doormen Design Document

|  |  |
| --- | --- |
| **Mandatory Requirements** | **Optional** |
| See all hunts and their status | Theme |
| Sort the list by date created | Invitation text being set by admin |
| Title | Assign a specific URL for the hunt |
| Create a new hunt | Find players by access code |
| Edit hunts that are ongoing |  |
| Create accounts for users |  |
| Invite users to hunts |  |
| Search existing accounts |  |
| Assign a list of tasks |  |
| Set the order of the list |  |
| Edit existing hunts |  |
| Edit only Pending or Active |  |
| Notify players when a hunt becomes Active |  |
| Can edit any part of hunt except status and creation date |  |
| Create accounts for players using email and phone number |  |
| Send hunt invitation and link by email, access code by text |  |
| Access code unique to person and hunt |  |
| Codes can be activated or disabled or pending invite |  |
| Invite multiple people to the hunt |  |

**Suggestions/Additions to Requirements**

* Have users create a password for their accounts
* Have a way to create Admin accounts
* Verification/application process for Admin privileges
* NFC chips at location points as opposed to QR codes
* Should be targeted towards mobile devices
* Scalable to cloud storage for widespread use

**Potential Risks**

* Data security: accounts should have passwords associated instead of just email and phone number
* Data that is stored should be salted and hashed to protect users
* Accessing location data may incur a fee
* Who is given access/clearance to be an admin?
* Potential for cheating if QR codes are static and can be copied
* Can admins see ALL hunts on the system/server or just hunts they own?

**System Level**

* SQL Database with Hashing to protect data
  + Tables for Hunts, User Info, Destination/Location names and info
  + Codes for locations are unique per location and per hunt
* Geolocation to associate with locations as well
* QR code generator
* Web application for user interface and admin editing
* Web server to handle requests between application and clients
* Entity framework so the application can query the database
* NFC reader integration
* Email/text code automation

|  |  |
| --- | --- |
| Mandatory | Optional |
| Map/ Map markers |  |
| Player Progress tracking | QR scanning |
| In-Game task list/location markers |  |
| Customize Profile |  |
| Join the activity |  |
| Unique code/Authentication |  |
| Website |  |
| Team Up- Shared Code for group |  |
| Accept QR code text |  |
| Record Game Tasks |  |
| Location Tracking |  |
| Track/Sync Team progress |  |

**System Level:**

* Web Interface to view Hunt and enter progress.
* Ability to scan QR code/enter text/Track location to complete task
* Map interface to view tasks on a live map and see which ones are complete
* SQL database to store user profile information, with the ability to change profile information such as username or profile image.
* Entry Validation, if not valid send user error message, if valid send user conformation message.

**Potential Risks**

* Cheating through sharing of codes and QR codes might make actual location the only viable way to enforce this as a lesson.
* Make sure to hide user location from unwanted individuals to ensure user safety

|  |  |
| --- | --- |
| Mandatory | Optional |
| Solution is maintainable | It is not necessary to collect user data in the first iteration. (Manual entering for testing phase) |
| Stored in a proper manner |  |
| Client Server Architecture |  |
| No cookies fully based off a Server |  |
| Works on multiple different kind of devices (Bootstrap) |  |
| Shall always be able to support at least one ongoing game. |  |
|  |  |