Playing with Murder Mobile Application

Nate Hoffman, Jack David, Colin Maresh, Matthew Cargille, Kevin Lin, Xuetong Du (Team 1)

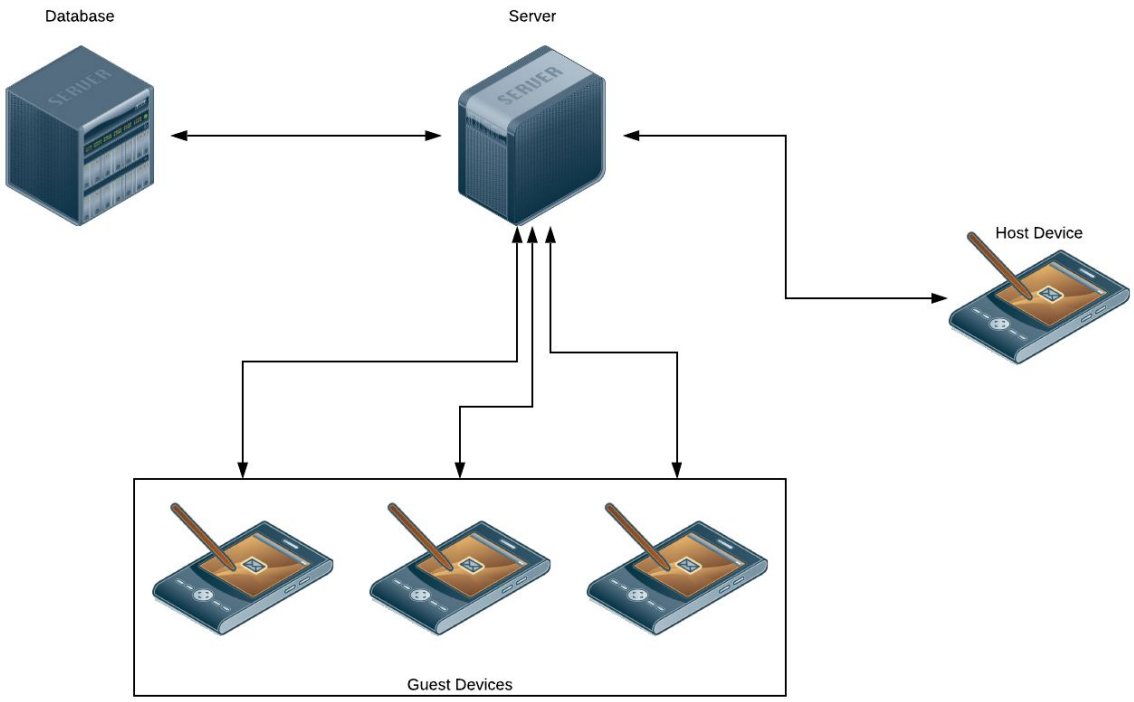
2019-02-18, version 1.0

Document Revision History

Rev. 1.0 <2019-03-01>: initial version

System Architecture

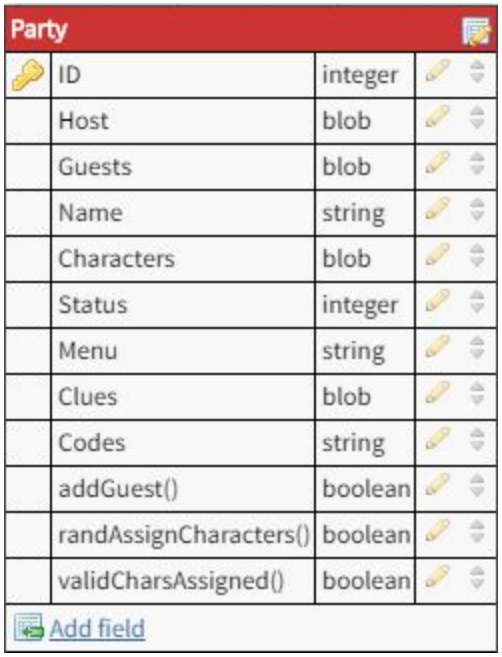
Our primary system architecture is a client-server application structure, with an event-based communication model. We will have a server-side application responsible for database management and event propagation, and a mobile application which acts as the main point of user interaction. For each party, different functionality will be available on the mobile application depending on whether you are a guest of or the host of the party.



The mobile application will utilize an MVC architecture to organize its components. The canonical model will be our Firebase database. Firebase is a cloud database service managed by Google, with excellent android integration capabilities. The controller will be device-side code responsible for loading the data and presenting to the view. The view will be standard Android XML display activities.

Design Details

This section is meant to show how we plan on organizing our database by tables.



Party	
ID	integer
Host	blob
Guests	blob
Name	string
Characters	blob
Status	integer
Menu	string
Clues	blob
Codes	string
addGuest()	boolean
randAssignCharacters()	boolean
validCharsAssigned()	boolean
Add field	

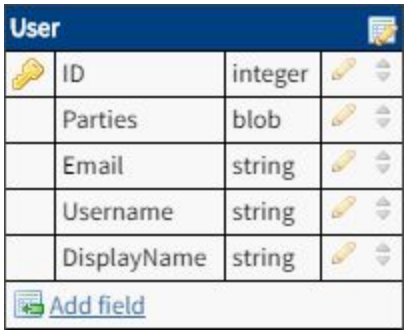
Party Class:

The Party class is going to be the primary object referenced throughout our application. This will contain a lot of the information necessary for running the game including references to guests and characters in the game.

1. ID
    a. The ID is the primary key of the Party object. This will also auto-increment so no Party object has the same ID as another.
2. Host
    a. The Host is a reference to a User object. This will allow the User to have more privileges than the Guests.
3. Guests
    a. The Guests will be an array of Guest Objects that references each Guest invited to the Party.

4. Name
    a. The Name will be the Name of the Party pack purchased by the Host.
5. Characters
    a. The Characters will be an array of Character Objects that references all of the characters available to the story.
6. Status
    a. The Status will be a reference to what stage the party is in.
        i. If Status is 0, the Host is still assigning characters and the Guests cannot view anything
        ii. If Status is 1, the Guests are allowed to view their Character's 1st Act info.
        iii. If Status is 2, the Guests are allowed to view their Character's 2nd Act info.
        iv. If Status is 3, the Guests are allowed to view their Character's 3rd Act info.
        v. If Status is 4, the Party is over and Users and Guests can view it as a previous party attended.
7. Menu
    a. The Menu will be an array of Strings where the Host stores the dishes that they will be preparing for their party. This will just store the title of the dish.
8. Clues
    a. The Clues will be an array of Clue Objects that the Host is able to manipulate if the Clue was found.
9. Codes
    a. The Codes will be an array of Strings where the Codes sent to the Guests are stored. These Codes are what the Guest uses to register their account.
10. addGuest()
    a. The addGuest() function will add a Guest to the Guests array.
    b. Returns True if the Guest was added successfully
    c. Returns False if there was an error adding the Guest
11. randAssignCharacters()
    a. The randAssignCharacters() function will randomly assign Characters to the Guests. This will ensure all of the required Characters are assigned to Guests.
    b. Returns True if the Characters were able to be assigned properly.
    c. Returns False if there was an error in the assigning process.
12. validCharsAssigned()

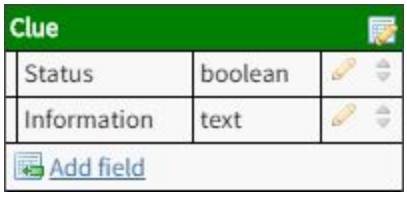
a. The validCharsAssigned() function makes sure that all of the valid Characters in the Characters array are assigned.
b. Returns True if all required Characters assigned.
c. Returns False if not all required Characters assigned.



User Class:

The User class is what is generated when a User creates an account. They can log into their account and view the parties they are a part of and have been a part of in the past.

1. ID
    a. The ID is the primary key of the User object. This will also auto-increment so no User object has the same ID as another.
2. Parties
    a. The Parties will be an array of Party objects where the User can view current and previous parties.
3. Email
    a. The Email will be a String that stores the User's email address given when registering.
4. Username
    a. The Username will be a unique identifier the User creates to be known by.
5. DisplayName
    a. The DisplayName is the actual name of the user that other Guests and Hosts will see.



Clue Class:

The Clue class is what stores the information about clues that can be found during the party. They will be primarily referenced by Party objects.

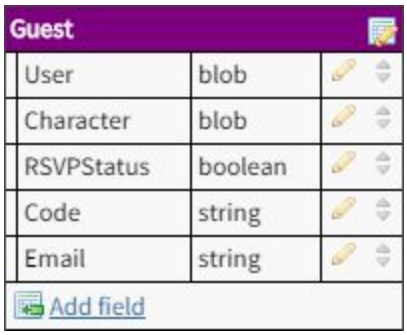
1. Status
    a. The Status will be a boolean that is evaluated to true if the clue has been found and false if it is still hidden.
2. Information
    a. The Information contains the text that is released to the Guests when the Clue is found.

Character Class:

The Character class is what stores the information about each Character sheet that gets assigned to Guests. The information in the Character class will be distributed depending on the Status field in the Party object. Each Guest will be able to view one Character that was assigned to them by the Host.

1. FirstActInfo
    a. The FirstActInfo contains all of the information that is available to the Guest when their Character is assigned to them. They will be able to view this information before the Party begins.
2. SecondActInfo
    a. The SecondActInfo contains all of the information that is available to the Guest when the 2nd Act begins. This event is triggered by the Host which reveals this information.

3. ThirdActInfo
    a. The ThirdActInfo contains all of the information that is available to the Guest when the 3rd Act begins. This event is triggered by the Host which reveals this information.
4. Required
    a. The Required field is a boolean that states whether or not the Character is required for the story. This is True if the Character is required and False if the Character is not required.



Guest Class:

The Guest class is what stores the information about each Guest attending a party. The Guest object is primarily referenced by the Party object.

1. User
    a. The User is a User object that contains the User represented by this Guest object.
2. Character
    a. The Character field is a Character object that represents the Character assigned to the Guest.
3. RSVPStatus
    a. The RSVPStatus is a boolean that returns True if the Guest has RSVP'd that they will attend the party and False if they will not be coming.
4. Code
    a. The Code is the value sent to the Guest through the invitation that the Guest must input into the system or they cannot access the Party.
5. Email
    a. The Email is the email address of the Guest input when they register.

View Details

Login [0]

Here, the user will enter their credentials to get access to the rest of the app. For users with an existing PWM account, they may enter their PWM credentials. For guests, they may enter the code that was distributed to them with their invitation.

Content	Component	Description
Email Address	Text Field	Users with existing PWM accounts will enter their PWM email here.
Password	Text Field	Users with an existing PWM account enter their password here.
Log in	Button	Submits this user's PWM login information for verification If this user has attended ● One party: to [1.1] for an ongoing party, [6] for a completed party ● Multiple parties: to [1.2]
Guest code	Text Field	Guest enters their guest code here
Check in	Button	Submits this user's guest code for verification If the party is ● Closed: to [7.1] ● Open: to [7.2]

Party List [1.1]

Users with existing PWM accounts will be able to view all past and current parties in which they have been a guest or host. This list is not displayed if the current user has only been a guest or host for one party in the lifetime of the account.

Content	Component	Description
Party list	List	Each list item will contain three items – The name of the game used for the party, the date the party took place, and whether the user was the host or guest for this party. Selecting any of the list items will take the user to the party page corresponding to the item's game and role. If this party is finished: to [6] Else User's role is ● Host: to [1.2] ● Guest of a closed party: to [7.1] ● Guest of an open party: to [7.2]

Host Page [1.2]

This is the primary page for hosts to access information about this game, as well as control the guestlist and flow of events in the party.

Content	Component	Description
Guidebook	Button	Takes the host to the guidebook page for this game. To [2]
Guestlist	Button	Takes the host to the guestlist. To [3]
Clue	Button	Takes the host to a page where a clue can be marked as found by one of the guests. To [5]
Proceed	Button	A contextual button that will serve four purposes over the course of the party: If act one has not been entered: ● Display text 'proceed to act one.' When the host selects this option, game information will be released to the current guestlist Act one has been entered: ● Display text 'proceed to act two.' When the host selects this option, new game information will be released to all guests Act two has been entered: ● Display text 'begin voting.' Selecting this option opens the voting option to all guests Voting has been opened: ● Display text 'close voting.' Selecting this disables voting, and moves the host to the recap page. To [6]

Guidebook [2]

This page will actually be a set of pages that allows the host to access parts of the party guidebook. Page headers and subheaders will be nested items of lists, allowing the host to easily navigate the 30+ pages of text to get to the appropriate location.

Content	Component	Description
Guidebook	List	A nested list structure containing the contents of the pdf guidebook. Selecting a list item will take the user to one of two places: If the item is a section header: ● A page with a list of subsection headers If the item is a section subheader: ● A page with the appropriate text, or another list of subheaders

Guestlist [3]

A list of this party's guests. Guestlist is determined by the host's invitations as sent out from either this application, or the PWM website.

Content	Component	Description
Guestlist	List	A list of guest names. Each item will contain the guest's name as well as their current character assignment. Selecting a guest item takes the host to the guest's page. To [4]

Guest Profile [4]

This page gives the host information and control over the current guest instance.

Content	Component	Description
Character info	Button	Brings the guest to the information sheet for the character currently assigned to this guest.
Assign new character	Button	Displays a popup list of all available characters in the game
Remove guest	Button	Display popup alert warning user about action they are currently taking
Character list	List (popup)	A list of the game's characters. When a character is selected, it will be assigned to the currently selected guest and updated on the guest page.
Remove guest warning	Text (popup)	This field will have two buttons with the options 'proceed' and 'cancel.' If proceed is selected, the guest is removed from the game's guestlist. If cancel is selected, no action is taken.

Host Clue [5]

Here, the host will have the option to set clues as being found by the guests. If a clue is found, new information will be viewable in the guests' apps. The elements listed here are repeated for as many clues as are in the game.

Content	Component	Description
Clue information	Text	A text field describing the nature of the clue
Clue found	Switch	If on, information about this clue will be enabled for viewing in the guests' apps. If off, this information will be disabled.

Party Recap [6]

After the party has concluded, the events will be summarized on this page. The summary will contain a brief synopsis of the story's ending, the results of the guest voting, and the highest rated pictures taken during the event.

Content	Component	Description
Synopsis	Text	A brief, prewritten segment detailing the events of the story.
Vote results	Text	A listing of all the voting categories and their winners
Gallery	Images	A selection of the highest rated pictures taken during the event

Guest Page - Closed [7.1]

A page with a notification that the host is still getting the character assignments organized. The guest will also be presented with the option to RSVP to the event.

Content	Component	Description
RSVP – Yes	Button	When this button is selected, this guest will be counted as attending the party. Display the 'cancel' button
RSVP – No	Button	When selected, this guest will be counted as not attending. Display the 'Can attend' button
RSVP – Cancel	Button	This guest is cancelling a previously confirmed attendance. Display 'can attend' button
RSVP – Can attend	Button	This guest is confirming their attendance after initially cancelling. Display 'cancel' button

Guest Page – Live [7.2]

This is the primary page for hosts to interact with and access information about this game.

Content	Component	Description
Character sheet	Button	Direct the guest to their character sheet. To [8]
Clue	Button	Open up QR code scanner for guest to scan clue. To [9]
Gallery	Button	Take user to image gallery for this party. To [10]
Vote	Button	Take user to ballots for this party. This option is closed until the host allows voting from their device. To [11]

Character sheet [8]

Here, the guest will be able to reference all information about their character. Information will be presented in a similar manner to the Guidebook [2].

Content	Component	Description
Character sheet	List	A nested list structure containing the contents of the pdf character sheet. Selecting a list item will take the user to one of two places: If the item is a section header: ● A page with a list of subsection headers If the item is a section subheader: ● A page with the appropriate text, or another list of subheaders

Clue Found [9]

After the guest successfully scans a clue, they are brought to this page to see a synopsis of the clue's contents.

Content	Component	Description
Synopsis	Text	A brief description of the clue's contents
Back	Button	Brings the user back to the game's main page. To [7.2]

Gallery [10]

Here, the guest will be able to see all pictures uploaded by guests of this party.

Content	Component	Description
Gallery images	Image	A picture uploaded by a guest of the party. Swiping left or right on this picture will allow the user to view other pictures in the gallery.
Like	Button	A guest can say that they 'like' a picture. This will be taken into account when generating the party recap.
Upload photo	Button	Takes the user to their camera, where they can take a picture to be uploaded to the gallery

Voting [11]

On this page, guests will be able to place votes for members of the party according to category, e.g. 'Best dressed,' 'Best actor.' Guests will also vote for who they think the murderer is here.

Content	Component	Description
Category description	Text	Description of what this category is
Ballot	List (popup)	This list contains all guests of the party. When a name is selected, it is set as the current vote for this category.
Submit	Button	When this button is selected, the vote is finalized and added to the tally for each category. To [7.2]

Implementation Plan

Task	Create repository
Description	Create a repository on Github for the project
Difficulty/Time units	1
Iteration	1
Assigned to	Jack
Priority	1

Task	Create PWM Account
Description	Create an account with the existing authentication system
Difficulty/Time units	2
Iteration	1
Assigned to	Matthew
Priority	4

Task	Log in as host
Description	Connect to authentication system. Authenticate user when log in.
Difficulty/Time units	2
Iteration	1
Assigned to	Colin
Priority	2

Task	Access party as guest
Description	Verify access codes from database to grant access as anonymous user.
Difficulty/Time units	1.5
Iteration	1
Assigned to	Kevin
Priority	2

Task	Access party as host
Description	After authenticated, host will select a certain party, and query the database for party information.
Difficulty/Time units	1.5
Iteration	1
Assigned to	Colin
Priority	1

Task	Host assign characters to guests
Description	Add a character class reference to each guest class.
Difficulty/Time units	3.5
Iteration	1
Assigned to	Jack
Priority	1

Task	Invite guests to party
Description	Import contacts information from phone using Android API or manually enter contacts to send invitation to guests.
Difficulty/Time units	4
Iteration	1
Assigned to	Matthew
Priority	2

Task	Host opens content for guests
Description	Changes party status to Act1 and make content available to guests.
Difficulty/Time units	2
Iteration	1
Assigned to	Kevin
Priority	1

Task	Remove guest from party
Description	Remove guest from Party class and rearrange character list.
Difficulty/Time units	3
Iteration	2
Assigned to	Selina
Priority	4

Task	Change guest character
Description	A panel to allow hosts to edit each assigned character.
Difficulty/Time units	3
Iteration	2
Assigned to	Nate
Priority	4

Task	Guest RSVP
Description	After guest clicks link and confirmed RSVP status, system sends notification to host.
Difficulty/Time units	3
Iteration	1
Assigned to	Colin
Priority	1

Task	Register 'murder' event
Description	Send push notifications to guest phones. Change dead character after murder. Change Party status to Act2.
Difficulty/Time units	2
Iteration	1
Assigned to	Jack
Priority	1

Task	Register clue found as host
Description	Send push notifications to guest phones.
Difficulty/Time units	1.5
Iteration	1
Assigned to	Jack
Priority	1

Task	Register clue found as guest (QR code)
Description	Access camera function on phone and import QRcode library. After QRcode is read, send push notifications to guest phones.
Difficulty/Time units	3
Iteration	2
Assigned to	Selina
Priority	5

Task	Open voting for party
Description	Host starts voting process and bring every guests to voting page. Send push notifications to guest phones.
Difficulty/Time units	2
Iteration	1
Assigned to	Colin
Priority	1

Task	Vote as guest
Description	Submit vote to database.
Difficulty/Time units	2
Iteration	1
Assigned to	Matthew
Priority	1

Task	Close voting for party
Description	Host closes voting process and system unveil voting results.
Difficulty/Time units	1.5
Iteration	1
Assigned to	Nate
Priority	1

Task	Generate party recap
Description	After the party, system creates a video or a picture of memory recap.
Difficulty/Time units	4.5
Iteration	2
Assigned to	Kevin
Priority	3

Task	Enter menu items as host
Description	Hosts add notes to party class.
Difficulty/Time units	2
Iteration	2
Assigned to	Selina
Priority	4

Task	Upload picture as guest
Description	Upload images to Firebase cloud storage. Connect to social media using respective APIs.
Difficulty/Time units	3.5
Iteration	2
Assigned to	Nate
Priority	3

Task	UI Design - Login
Description	Design and implement elements on the Login page.
Difficulty/Time units	2
Iteration	1
Assigned to	Matthew
Priority	1

Task	UI Design - Party List
Description	Design and implement elements on the Party List.
Difficulty/Time units	1
Iteration	1
Assigned to	Kevin
Priority	1

Task	UI Design - Host Page
Description	Design and implement elements on the Host Page.
Difficulty/Time units	2
Iteration	1
Assigned to	Selina
Priority	1

Task	UI Design - Guidebook

Description	Design and implement elements on the Guidebook.
Difficulty/Time units	1
Iteration	1
Assigned to	Nate
Priority	4

Task	UI Design - Guestlist
Description	Design and implement elements on the Guestlist.
Difficulty/Time units	1
Iteration	1
Assigned to	Colin
Priority	1

Task	UI Design - Guest Profile
Description	Design and implement elements on the Guest Profile.
Difficulty/Time units	2
Iteration	1
Assigned to	Jack
Priority	1

Task	UI Design - Host Clue

Description	Design and implement elements on the Host Clue.
Difficulty/Time units	1
Iteration	1
Assigned to	Selina
Priority	4

Task	UI Design - Party Recap
Description	Design and implement elements on the Party Recap.
Difficulty/Time units	1
Iteration	1
Assigned to	Matthew
Priority	4

Task	UI Design - Guest Page(Closed)
Description	Design and implement elements on the Guest Page.
Difficulty/Time units	1
Iteration	1
Assigned to	Jack
Priority	1

Task	UI Design - Guest Page(Live)

Description	Design and implement elements on the Guest Page.
Difficulty/Time units	2
Iteration	1
Assigned to	Kevin
Priority	1

Task	UI Design - Character Sheet
Description	Design and implement elements on the Character Sheet.
Difficulty/Time units	1
Iteration	1
Assigned to	Colin
Priority	4

Task	UI Design - Clue Found
Description	Design and implement elements on the Clue Found.
Difficulty/Time units	1
Iteration	1
Assigned to	Matthew
Priority	4

Task	UI Design - Gallery

Description	Design and implement elements on the Gallery.
Difficulty/Time units	1
Iteration	1
Assigned to	Colin
Priority	4

Task	UI Design - Voting
Description	Design and implement elements on the Voting.
Difficulty/Time units	2
Iteration	1
Assigned to	Nate
Priority	1

Testing Plan

Unit Testing:

As we are building our application, we will be creating unit tests for small portions of code, and testing as we go. For our background logic (model) and controller, we will use JUnit, which is the most common unit testing framework for the Java language. To test the visual output of our program, we will run it with an emulator on Android Studio, making sure that it looks the way we want it to on different screen sizes.

Integration Testing:

As we integrate different parts of the program together, we will be testing along the way to ensure that our classes, activities, and database can all work together seamlessly. Android Studio provides a useful debugging tool, which will be helpful, especially for

local logic on a single device. We will also need to test that we can successfully store and access data from our database. We can use a local database for testing as we continue to work with our customer on a solution to integrate into their existing database architecture. Finally, we will need to ensure that different devices running the app can communicate with each other correctly, namely that the host can send invitations to guests, the guests can RSVP, the host can update the party status, the guests can inform the host and other guests about clues found, and the guests can vote and receive party information at the end. The emulators in Android Studio can access the internet through wifi, so we can use them to test these features.

System testing:

As our project is approaching fully integrated, we plan to use the Black box testing in case there is any incorrect or missing functions and behavioral errors. We will be focusing on the usability testing, regression testing and functional testing.

- Functional testing
  Scheduled Time: Iterations 2 and 3
  The Functional Testing aims to verify each functions are operated under our project specification. This is the part which mainly involved in the black box test without concerning about the source code. For this part, we will provide different types of inputs properly and comparing the output with the expected results. It basically involves in every part of our project from User Interface to Database. The tools we plan to use for this testing is JUnit since it is mainly for java applications.

- Regression testing:
  Scheduled Time: Concurrently during Iterations 1-3
  To confirm what we have modified does not unintentionally affected the system performance, we will run the regression testing every time when an issue has been marked “fixed” or a repository commit has been pushed to the database. This testing will be maintained continuously and strictly throughout the whole development to avoid any unnoticed errors as well as ensure the program is well-tested at the end of the semester. Moreover, such testing plan will be simple enough for every group member to easily understand and execute.
  In order to achieve the ultimate result while maintaining the high efficiency, we will create a “Representative Sample” of test which fits in most of the situation. In our case would be inviting a random number of guests and confirming only a part

of them will show up for instance to test our early staged program. We could also use the tools in the Unity Test Tools alternatively to see the breaks in the code so that no more testing scripts are required to be written.

- Usability testing:
  Scheduled Time: Iterations 2 and 3
  This part of our system testing is mainly about the User Experience testing, which focuses on the user's ease to use the App as well as how well in handling each controls during different game periods. The targets of this test is to make sure our App is useful, accessible and desirable as well as justify its efficiency and effectiveness. We will improve our design based on the feedback from the PWM company.