

School of Electrical, Electronic and Communications Engineering

**Software Engineering  
Project Deliverable 1**

|  |  |
| --- | --- |
| Team Name | Team ROM |
| Student Names | Mark Purcell; Owen Binchy; Ronan Scally |
| Student Numbers | 10374707; 10374551; 09333541 |
| Lecturer | Mark Scanlon |
| Date Submitted | 17/10/2014 |

# Introduction

Teamer, the online sports team management application, wishes to add functionality to its existing app. The Teamer phone application currently allows users to create a team, make an event for that team, and notify users of the event using SMS text messaging. Teamer wishes to expand its business, and make their app a more sociable experience.

The functionality that Teamer are requesting to add concerns real-time events, and location based data.

Teamer wishes to add a feature to its app that help its users to find games that match to their interests. Users will be able to find nearby games, organised by people who they haven’t previously met. Users will create a profile which will describe their sports interests and availability to play. Location data will be sent to the server, and notifications will be pushed to the user’s phone if an event matches their interests and is nearby.

The functionality of this application is described in more detail in the next sections of this document.

## Glossary

Server: When referring to server in this document we mean the application running on a remote machine which processes requests from the phone application and interacts with the database.

# User Requirements **Sign in**

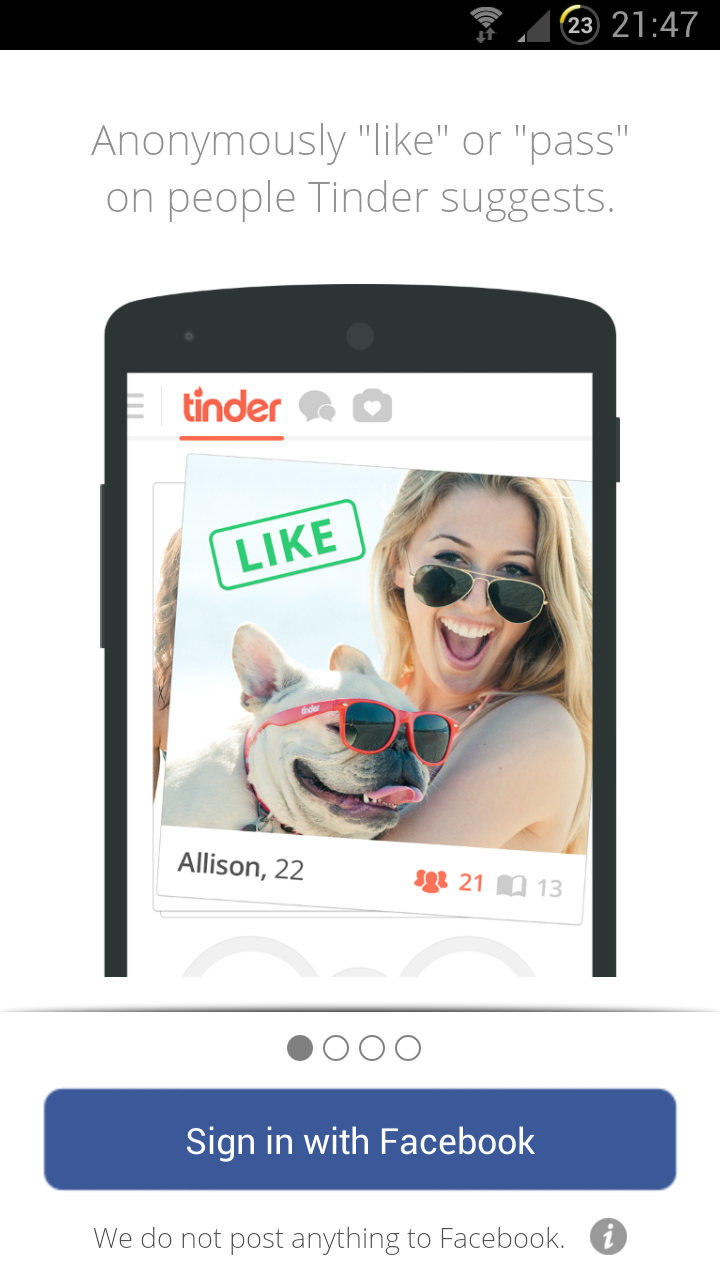
The user shall be able to sign into this app using Facebook login. Certain profile details will be retrieved from Facebook where possible, such as profile picture.

Figure 1: Facebook sign in, from Tinder app

## Create own profile

The user must create their profile on first use of this app. Profile details shall include name, profile picture, date of birth, sports interested in, and phone number (optional). Where possible, these details will be gathered from Facebook.

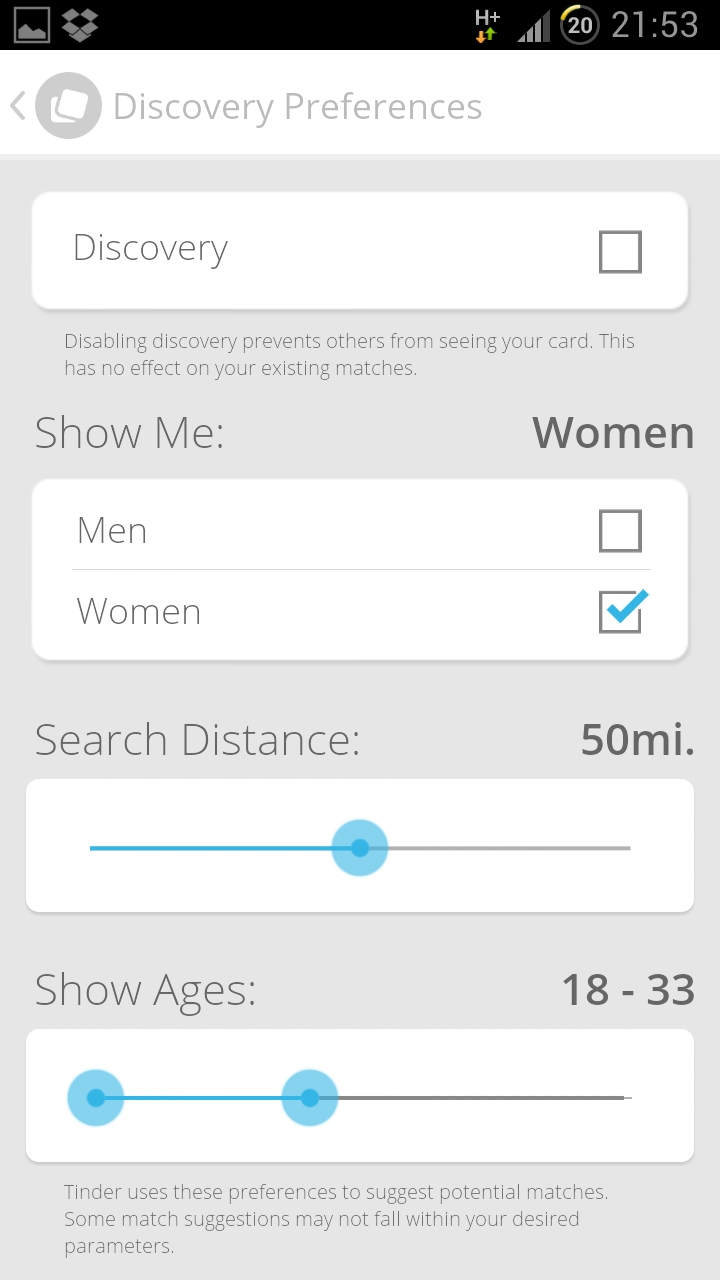


Figure 2: Profile settings, from Tinder app. Some settings will be relevant to this app, such as search distance

Find Nearby Games  
The user shall be able to log on to the app and search for games. The search can be filtered, subject to the user’s criteria. The results shall be displayed as a list, sorted by relevant criteria, such as location and time. This sorting criteria shall be modifiable by the user. The user can then click on these results and view details of the event/page.

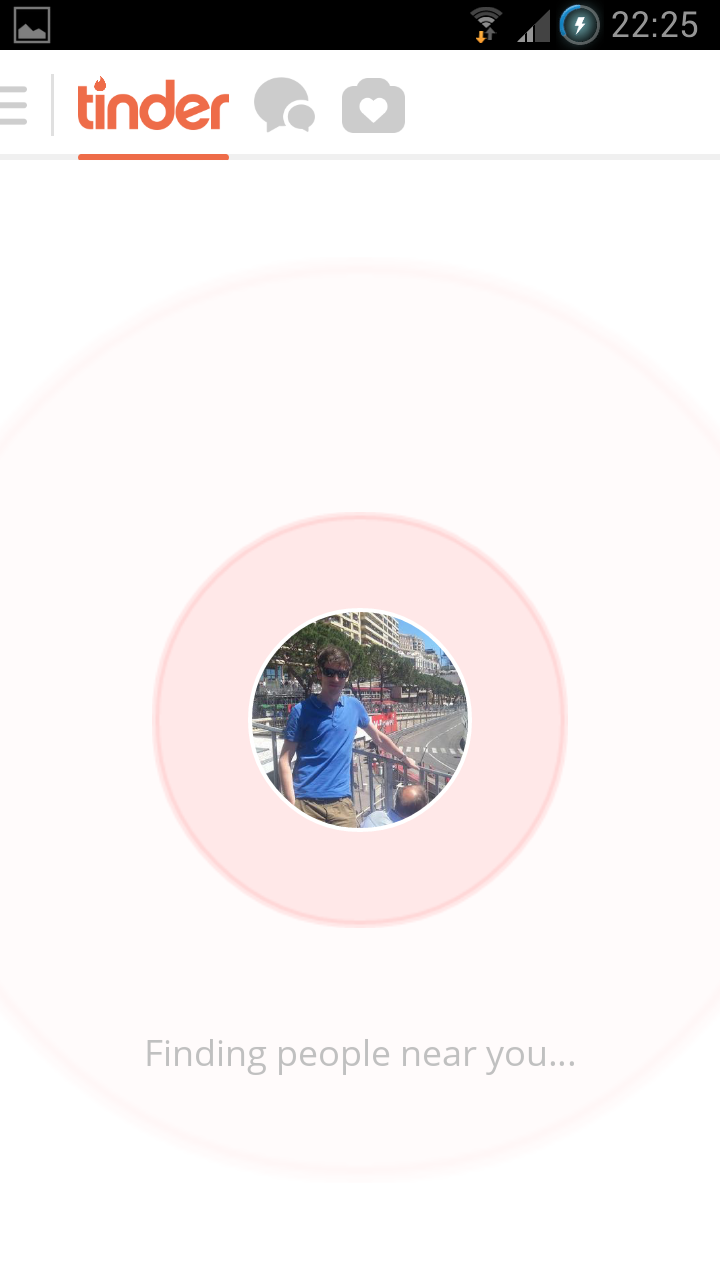
****

Figure 3: Idle screen, taken from Tinder app. The idle screen on this app should be similar

## Create an Event

The user shall be able to create an event. Upon creating an event, the user will input event details, such as time, place, sport, and number of players required. The user can also set constraints on who can join the event subject to parameters including age and skill level. After entering all event details, the user can either submit the event, or cancel.

Once submitted any user can view the event and the creator should be able to invite other people, or invite an entire group. On completion of event creation, the host shall be automatically added to attendees as ‘host’.

The app shall determine the number of spaces in the game remaining, based on the number of people attending the event. The host can also add a guest to enter a person who is not registered user of the app.

## Manage an event

The user shall be able to view all of the events that they are hosting. They shall be able to click into the event and edit details of it. They shall be able to view attendees and their respective profiles. The user shall also be able to kick attendees. The user can also invite more players or groups at this point. The user can also delete the event; if this option is selected, the attendees should be notified.

## Event invitations

The user shall be able to view all events that they are invited to. Once the user has responded to an event with either ‘attending’ or ‘not attending’, the invitation disappears.

## Events attending

The user shall be able to view all of the events they are registered to as ‘attending’. The user shall have the ability to click on a link to the location of the event: this shall open the Google Maps app. The user shall be able to comment on the event. The user can also view attendees of the event.

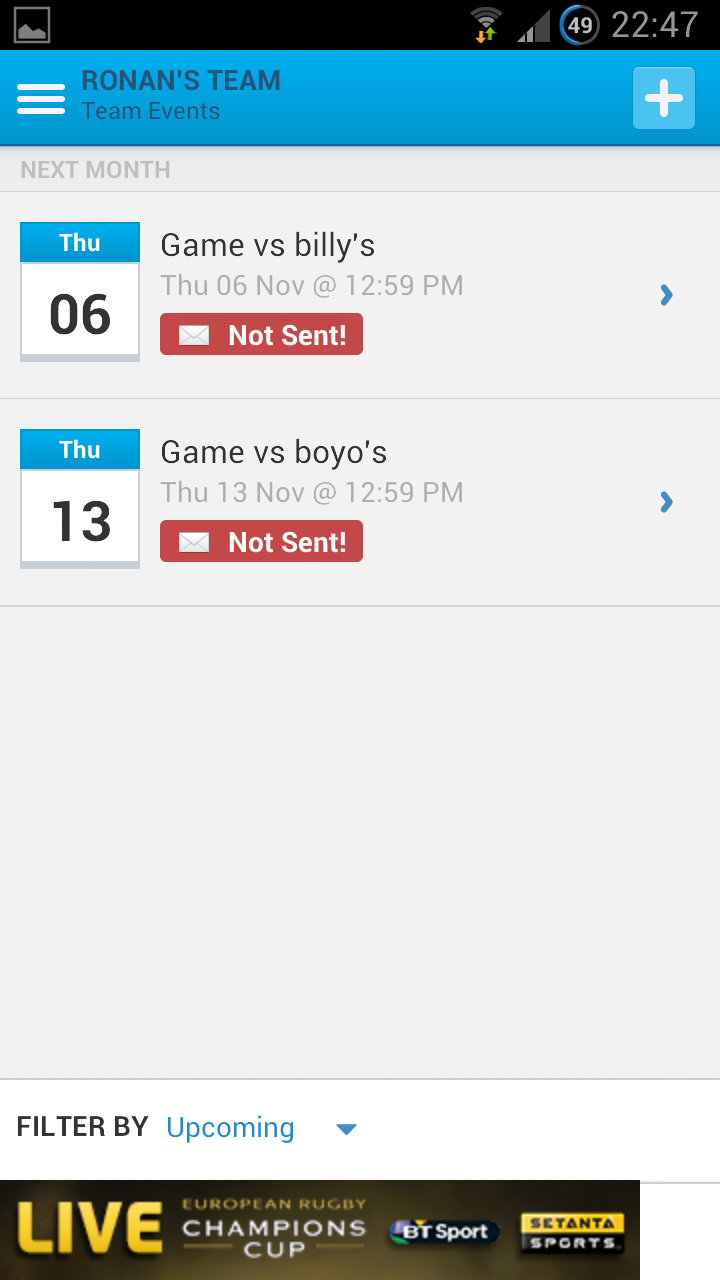


Figure 4: Events being attended by the user, taken from Teamer app

## Create a group

The user should be able to create a group. When creating group the user can add a description. The user can add other users to the group. After group creation, the user becomes administrator of the group. Groups can be batch-invited to events, from users who are members of that group.

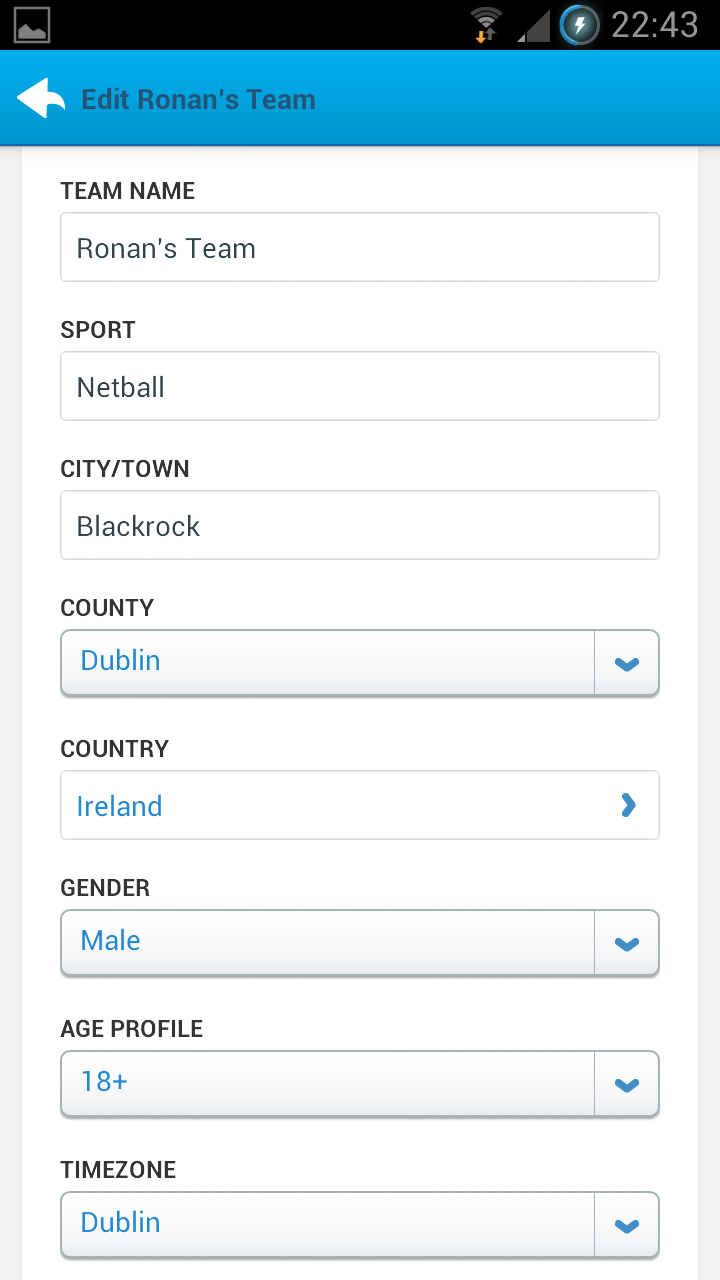


Figure 5: Group creation page, taken from Teamer app

## View groups

The user shall be able to view all of the groups that they are a member of. The user can click into the group and view group details. On entrance to the group home page, a comments page shall be displayed, where the user can leave a comment and view other user’s comments. The user shall be able to view a list of group members. The user also has the option to leave the group.

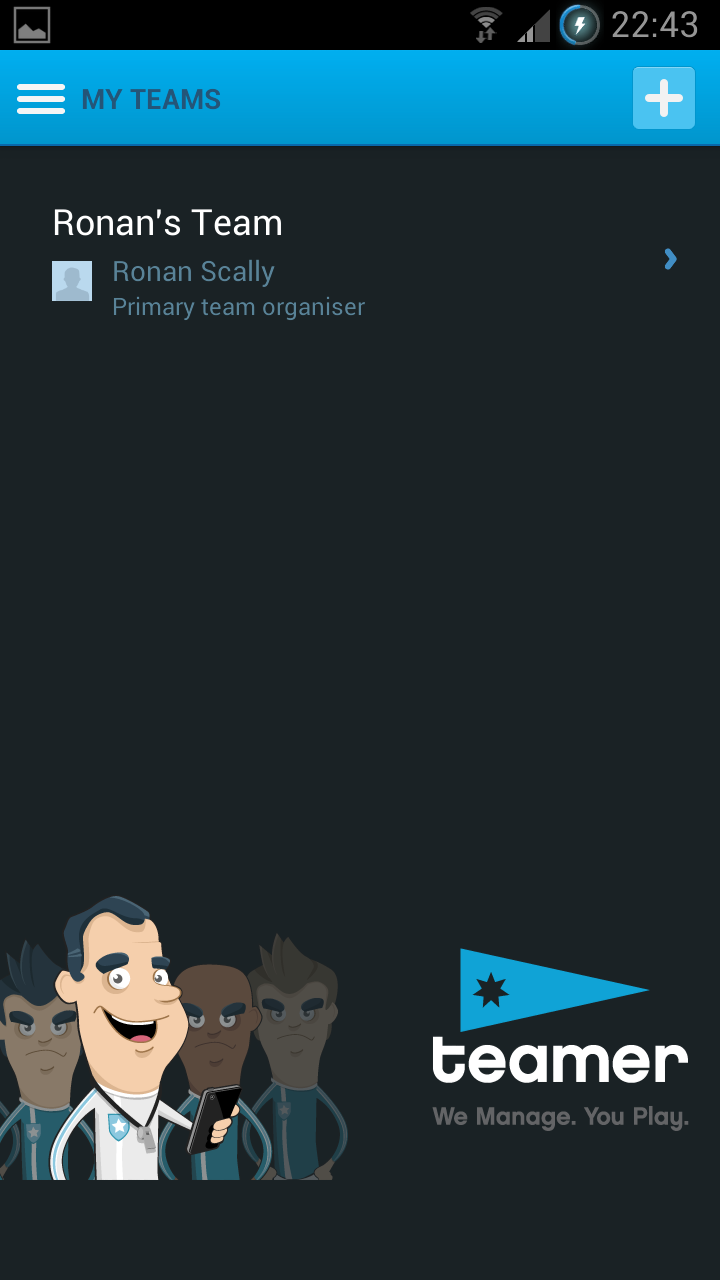


Figure 6: View groups page, taken from Teamer app

## Manage groups

The user shall be able to edit the groups that they administrate. They may remove users from the group, invite other users, or delete the group entirely.

*Optional: give other group members administrator privileges*

## Push notifications

The user shall get a notification under several circumstances:

* Invited to an event: user is taken to the event page
* Upcoming events that they are attending, according to their preferences
* New events occurring nearby, subject to constraints set in the users preferences
* Added to a group: user is taken to the group page

# System Architecture

## Server application

The server application runs on a UNIX machine. The role of the server is to communicate between the app and the database.

## Database

The database is where all of the data will be stored. It runs on the same machine as the server application.

## Phone application

The phone application will run on the Android operating system. This is where the user interfaces with the product. The phone is responsible for data entry from the users including profile, groups and event creation. The app is also responsible for displaying data to the user. Data is retrieved from the database, using the server app as an intermediary. The app will also collect GPS data from the phone to send to the server.

## Data Transfer

When the user creates an event, the app will display a form for the user to fill out. This will contain drop-down menus, check-boxes and text boxes which will contain details about the event. When the user completes the form the app will run a check to ensure the form is valid, i.e. all required entries have been completed. The app will then send this data to the ‘server’.

The server will validate this request to prevent such things as duplicate events or security issues. The server will format this data for the database and submit a request to create an entry in the relevant table for it. The server will then respond to the app, informing the app whether or not it was successful.

## Data Retrieval

When the user wishes to view certain data (i.e. a group, search events), the phone application sends a request to the server application, specifying which data it wishes to view. The server application will then validate the request. The server may need additional information, such as the user’s preferences, before requesting the required data from the database. If this is the case, the server will request this data from the database first. Once all the necessary information is present, the server application will make the request to the database. Upon retrieving the information from the database, the server application will check this data, sorting and filtering it if necessary (in the case of lots of results), and will then send the data to the phone application. The phone application will then display the data to the user.

## GPS

The phone application periodically sends the users location to the server. The server will store this in the database.

When finding nearby games, the server will calculate the user’s distance from the event based off the GPS data.

## Notifications

The server application will periodically poll the database, checking if there are any events that match user’s preferences. When a match is found, the server application will send a notification to the phone application. The server will record that this notification has been sent, and will not send another notification for the same match.

The server application will also be polling the database for events that are about to start. When an event is 1 hour from starting, the server application will send a notification to the phone application of all of the attendees of the event, subject to their preferences.

When an events details are changed (start time, location), the server application will send a notification to the phone application of all the attendees of this event, notifying the users of this change. The server application will also send a notification to the phone application of a user should it detect that a user has been added to a group or invited to an event.

## Login

When the user wishes to use the app they must first login. The phone app will contact the server application and check if the phone’s IP address is associated with any logged-in user. If not, the user must sign in using Facebook. Upon requesting their Facebook login details, the phone app will check these details with Facebook. If Facebook confirms their login to the phone application, the phone will then request the user’s profile from the server application. The server application will retrieve their profile from the database and return the data to the phone. If a profile doesn’t exist, the server application will request the phone to prompt the user to create a new profile.

Once logged-in the server application will remember the IP address associated with that user and will keep the user logged in. This way the server application can contact the phone application and send notifications without just responding to the phone’s requests. If the user logs out of their account, the phone will tell the server application to delete the user’s IP address from the database.

# Milestones

## Week 1: 23/10/14

Facebook login and user registration

After week 1, people should be able to sign into the app through Facebook. If the user does not have a profile, they will be prompted to enter personal details. If the user has used the app before they will be logged in automatically, and brought to the idle screen.

## Week 2: 30/10/14

Event creation

This is a significant milestone as it is a pivotal function of our app. At this stage the user should be able to create an event that other users of the app can see. Creating an event which other people can see and join is the main object of our app and achieving this is a priority which will lay a great foundation for the remainder of the project.

## Week 3: 06/11/14

Friends

This milestone is concerned with getting more people onto the app so that it can grow and expand. The number of users on the app is important so that the number of events also grows. As we are using Facebook for the login mechanism, it would be convenient for users to be able to invite their Facebook friends to join them on the app.

## Week 4: 13/11/14

Notifications

Notifications are a useful way to remind app users of upcoming events. As our app is quite dynamic and events can change on short notice, people need to be updated with the newest information. This implementation will require work on the server and database side, as well as Android OS notification on the phone application.

## Week 5: 20/11/14

Groups

The last implementation we are aiming to introduce to the app is groups. The idea behind this is that there are number of players who play regularly each week. This will let them organise personal events and communicate through the app. Work on this part of the app will mainly include the user interface and database. The remaining person will be solidifying the server as this is the final week edits will be made to the software.

After this day, no more functionality will be added to this project, and a week of testing will commence before delivering the final app on November 27th.

## Work Split

Front end: Owen Binchy Server: Mark Purcell Database: Ronan Scally

The project has been split into 3 sections: front end, server and database, with each person tackling a particular section as shown. As outlined in the milestones there are weeks where one of the three main areas will not have a huge workload. In these weeks, the person with the least workload will either assist another member in his part of the project, or alternatively continue working on his end of the project if he is able to continue.