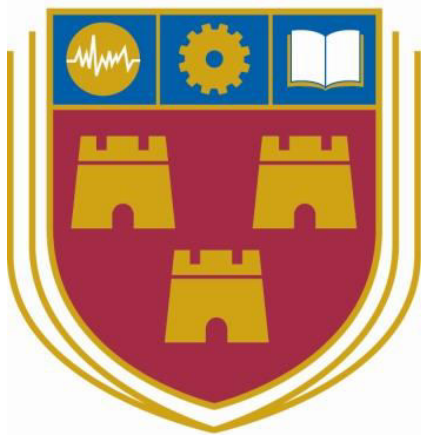


Pro Events

Functional Specification

Institiúid Teicneolaíochta Cheatharlach



INSTITUTE *of*
TECHNOLOGY

CARLOW

At the Heart of South Leinster

Name: Jonathan Finlay

Student Number: C00193379

Course: Bachelor of Science (Honours) Software Development

Tutor: Hisain Elshaafi

Date: 13-11-17

Contents

Introduction	2
Purpose	3
Undergrads:	3
Experienced Workers:.....	3
Companies:	4
Scope.....	5
Use Case.....	6
Brief Use Cases.....	7
Functional Requirement	9
Supplementary Requirements	10

Introduction

In this functional specification, I discuss about the purpose of this app, how it benefits the users, how each section of the app works and what is required. I will be going into detail on the functionality and how to use it successfully.

Pro Events is aimed at anyone who has an interest in their profession or just interested in that area as a hobby. It can be for all groups from Computing, to business to sports. When the user registers, they will fill in a form which will be used for interests and sending out notifications. Other users will be able to view your information and get in contact with you as your email will be able.

The home screen will feature your profile. On this page, the user will be able to navigate around the app. They can access the events page, connections and logout. They can also view their own followers and edit their profile.

Each account is customisable. You will have a profile picture, a description about yourself, your interest which is filled in when they sign up or edit their profile, previously attended events and future attending events, and your connections. Other users can view your profile once you're connected.

Any user can create an event. Events can be used for many reasons, whether it's a normal user who plans on learning something new on a topic and is wondering if anyone else wants to join in, to companies advertising jobs by inviting potential employees over to have them apply. After you attend an event, the app will ask you to rate the event which will let future users know if the events created by this user are a good enough standard. The creator can edit their events and cancel them. Anyone following an event will get a notification just over day before it takes place.

Purpose

The purpose of this document is to describe the functionality of this application. It will let anybody who reads this have an understand of what this app will consist of. Most importantly it will form consensus on what the program will be.

This application is to put help user's find events to their interests or skills. These events can range from a user wanting to have a study session with others on a specific topic or a company advertising a job and inviting potential employees over to have them apply. By viewing these events on the app you can check who else plans on attending the events and send a connection request to them. Once connected you both can view each other's accounts and message each other. Each account is customisable. You will have a profile picture, a description about yourself, your interests, email address for contact and notifications, previously attended events and future attending events, and your connections. Other users can view your profile once you're connected.

This app is aimed at a wide range of a professional audience. It can be used for students that wish to learn more or about the environment they are planning on joining, casual users that use it to find events that suit their hobbies or companies wishing to advertise the projects or a job position.

Undergrads:

It will help these users become more familiar with the field they will be entering. Some graduates when coming fresh out of college don't know much of their field of study other than what they've learned in college. One of the best way to learn your sector is by talking to those who work in the sector or by others who are also seeking more information in that sector.

This app will help users find events of interest and set them on their way. They will be able to read about the event and visit their website, see who else is interested and connect with those who are interested.

You could message others in the same position as you, if you don't want to attend alone or you can message more experienced users asking for information and help.

Experienced Workers:

For those who are already in the workplace, this app can help expand their knowledge. They can easily message the organisers and try get involved and share what they've learn within their area. Just because you already have a job doesn't mean you know it all. You're always learning which is what these events are for, introducing innovative ways to approach certain situations.

Companies:

A companies approach to this app would be seeking undergrads. They would create an event where they invite undergrads over for a day, show them around, introduce them to a working environment and demonstrate the type of work they'd be doing here.

This app isn't only for events, it can be used for scheduling meeting and/or conferences. An employer could organise a conference and add all their co-workers who would be expected to be there. They could set a date and time, and leave in the description what the topics of the conference are.

Scope

- **Android Studio:** This potentially be the main piece of software used throughout the project. It is used for writing code in java. All the development will be revolved around this including UI development, hard code, server and database setup. This will also connect to android phones to test on the device.
- **Apache PredictionIO:** This is a program that sets up a server to hold the data you send to it. It takes in the .JSON file and puts it on the server. All the data (e.g. from the user, events created) will be saved here. From here, it can be pulled and used suggestions and other users to view.
- **Raccoon:** This is a collaborative filtering recommendation engine built on Node.js and Redis. The engine uses the Jaccard coefficient to determine the similarity between users. It uses likes and dislikes to judge what to recommend to the user but in this case, only the like(interests) will be used. Similarly, to PredictionIO, it stores on the data from the user here and can be easily accessed.
- **PhoneGap/Cordova:** This could also potentially be the main piece of software either. It works with HTML, CSS and PHP. Once coded in NotePad++, PhoneGap converts it into a mobile app.
- **NotePad++:** This is where the HTML, CSS and PHP could be coded. It connects PhoneGap to convert it into a mobile app.
- **Nox:** This is an android phone emulator. This can be used to test the app if without and android phone.
- **Appery.io**

Appery.io is a cloud-based platform with visual development tools and integrated backend services. It provides tools to build the app UI. It uses HTML, CSS, JavaScript, jQuery Mobile and Cordova. It has a user-friendly UI and encourages new users as it's easy to use. It's one downfall compared to technologies is, it's pay to use.
- **Blacknight**

Blacknight is a great, easy to use database to store the data received from the users. Redis is an open-source in-memory database[4]. Redis supports different kinds of abstract data structures such as strings, lists, maps, sets, sorted sets and bitmaps. It supports many languages including Java and JavaScript.

Data that would need to be stored is:

 - Accounts and their associated username and passwords
 - Events
 - User interests

Depending on the software and algorithms chosen will depend on what type of database will chosen.

- **Azure**

Azure is a cloud computing service providing by Microsoft for building, testing, deploying and managing applications and services through a global network of Microsoft managed data centres.

The storage services provides REST and SDK APIS for storing and accessing data on the cloud. It's libraries can work with .Net, Java, Node.js, and Python. [4]

Benefits of using this service is:

- Easy set up.
- Ease of use with Visual Studio Xamarin.
- Storages data efficiently.
- Data can be accessed easily.
- Designed well for android devices.

- **Xamarin**

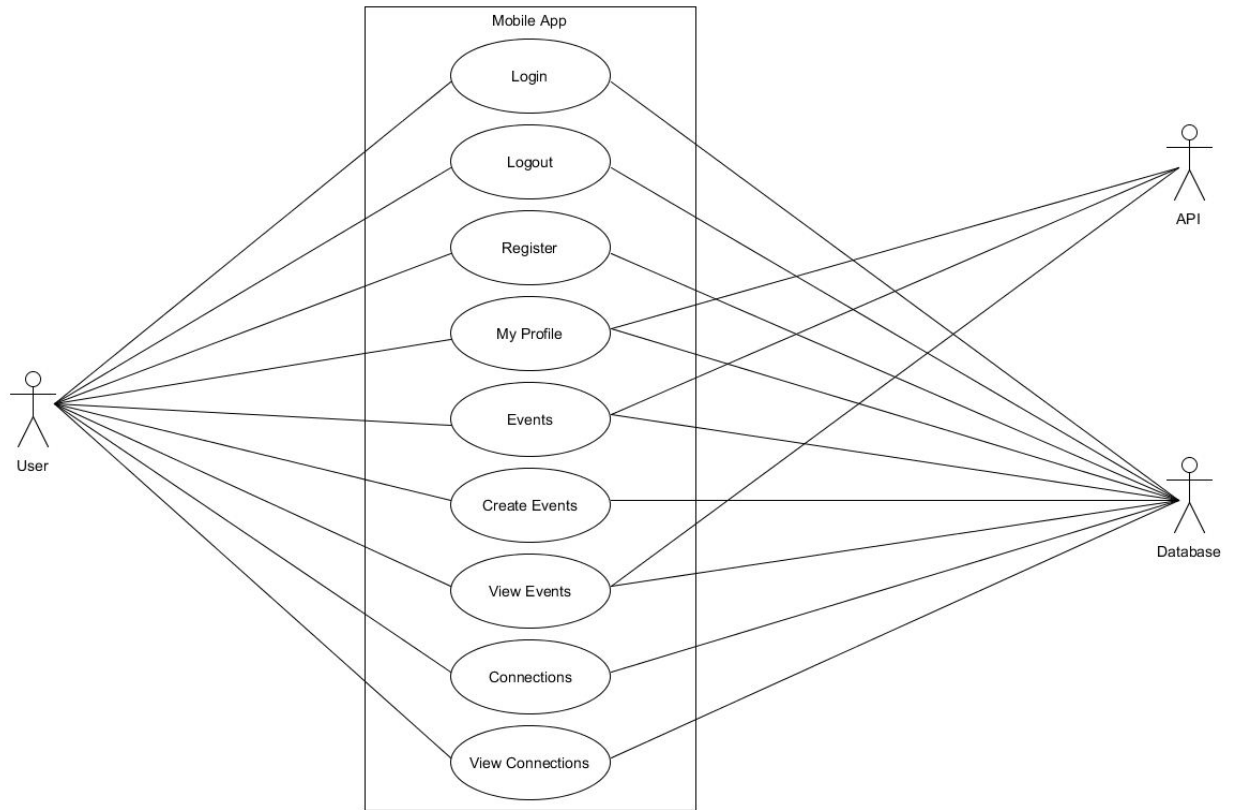
Xamarin is Microsoft own software company by the engineers that created Mono. It is known for cross platform implementations of Common Language Infrastructure(CLI) and .Net.[5]

As it has a C# shared codebase, Xamarin tools can be used to write native Android applications with native user interfaces.

The biggest bonus of using Xamarin is, it has a native android environment. It can take advantage of all the android tools available unlike apps make using PhoneGap/Cordova.

By installing the azure package available to Visual Studio, developers can use this service with some API calls. Developers can use these calls to create, add, update and delete tables.

Use Case



Brief Use Cases

NAME	REGISTRATION
ACTORS	User, App, API, Database
PRECONDITIONS	The app has successfully opened, and the initial login screen successfully loaded.
ACTIVITY	Begins when the user opens the app. The user clicks the register button and a form appears. The user fills in the form with relevant information and clicks the submit button. The account creation request is sent to the API for processing.
CONSEQUENCE	The system validates the registration request.

NAME	LOGIN
ACTORS	User, App, API, Database
PRECONDITIONS	The app has successfully opened, and the initial login screen successfully loaded. The user must have already signed up for an account.
ACTIVITY	Begins when the user starts the app. The system listens for the input of user login credentials (username and password). The credentials are sent to for validation purposes.
CONSEQUENCE	The system validates the credentials entered.

NAME	LOGOUT
ACTORS	User, App, API, Database
PRECONDITIONS	The user has successfully logged into the app.
ACTIVITY	Begins when the user has completed all interaction with the app. The user clicks the logout button and is asked to confirm that this was intended.
CONSEQUENCE	The user's app session is terminated, and the initial login screen appears once again.

NAME	SEARCH
ACTORS	User, App, API, Database
PRECONDITIONS	The user has successfully logged into the app.
ACTIVITY	Begins when the user has writes keywords into the search bar. The user intends to search for either connections or events based on the keywords.
CONSEQUENCE	The system displays all results from the database suited towards the user's request. These results being either connections or events.

MY PROFILE	
NAME	
ACTORS	User, App, API, Database
PRECONDITIONS	The user has successfully logged into the app.
ACTIVITY	Begins when the user selects on their profile picture at the top of the screen.
CONSEQUENCE	The system redirects the user to their profile page.

EVENT	
NAME	
ACTORS	User, App, API, Database
PRECONDITIONS	The user has successfully logged into the app.
ACTIVITY	Begins when the user clicks on the event page. System displays events based on the search. The user can interact with these events by viewing the event page and following the event.
CONSEQUENCE	The system displays all results from the database suited towards the user's request.

CONNECTIONS	
NAME	
ACTORS	User, App, API, Database
PRECONDITIONS	The user has successfully logged into the app.
ACTIVITY	Begins when the user clicks on the suggest connections page. System displays other user's pages with some information where the user can follow them. If following, they can view their page and view what events they are following.
CONSEQUENCE	The system displays all results from the database suited towards the user's request. Showing most suited first and nearby.

CREATE EVENT	
NAME	
ACTORS	User, App, API, Database
PRECONDITIONS	The user has successfully logged into the app.
ACTIVITY	Begins when the user clicks on create event. A form appears, and the guest fills in the relevant information for the event and clicks the submit button. The system then processes the event page.
CONSEQUENCE	The system creates the page and fills in the information from the form onto the page. The event will be added to the database, available to appear in searches under relevant interests and skills.

Functional Requirement

The first function requirement is logging in. The user will register an account if they have not done so already. Once registered, they proceed to the login page. They will enter their username and password and the system checks for their account. If they have an account, they'll proceed to their profile but if not, they will be brought to a page where they can create their account. Once at their profile page, the user can navigate around the app, view their followers and edit their profile from this page. When viewing their followers, they can follow them back and view their profile. Once viewing their profile, they can see their information and what events they are following which will also be mentioned below.

The second functionality is the events system. Here users can search for events related to certain categories. The categories will be limited at first but can widen in the future. Users can view the events details, who else is following it and can follow it there self. Any event the user is following will be added to a list so the user can stay organised. At any time, users can unfollow these events.

Any event a user creates will be added to a list for organisation. Any of these events can be edited or cancelled by the creator.

A third functional requirement is notifying the user of an upcoming event. As the event is upcoming, it would be useful for the user to get a reminder of this. This app aims to send a push notification with over a days' notice to the user. In an app such as this, where it organises events, sending out a reminder is crucial.

A fourth functionality is finding connections. A user will be able to search for other users based on their category. You may follow these users where they will be added to a list for organisation. Once added to this list, they can view their profile and unfollow them. When viewing their profile, you can also see what events they're following where you can also interact with these events.

Supplementary Requirements

Functionality: The app must have access to the internet to interact with the API and database to access the data.

It should be a swift response when accessing this data, so it doesn't leave the user waiting.

Usability: The user doesn't need to be very experienced with the app. The interface will be friendly, appealing and easy to work. The app will work like most apps currently on the market. It'll have a search bar, profile, event pages to view and connections to connect too.

Reliability: The app is an online app as it is connected to a database, which means it requires a database. There should be no problems with the app on most updated Androids. Passwords will be encrypted for security; all other information will be displayed on the users page. If the user enters invalid data, they will be prompted as they enter it informing them of the mistake, this should prevent any problems.

Performance: Newer phones should have a fast response time. This app will be designed to give the fastest and easiest experience to the user. Wi-Fi connection also matters when it comes to response time. Since the app will have to gather data from the database, a solid internet connection will improve the response time. All tasks should be completed in an acceptable time.

Supportability: Foreseen extensions to the app would be adding social medias to the login phase and gather interests and skills from there. Twitter, you could analysis the users tweets and catch keywords that would interest the user, LinkedIn could help the search function and help find job opportunities. Admins would maintain the app. Any bugs are new additions to the app would be dealt with as they come up and an update will be released. Each version will be kept in case a major fault appears and it needs to be reverted. It will only be released on Android market at first but if it's a success, it could be adapted to iOS devices.

Interface: The interface will be appealing, friendly and easy to use for the user. The user will be able to scroll and search easily. All data entered will be compared to what's in the database and the search will output what suits the users' interests. The output will be nice, tidy and easy to read.

Operation: The owners/ developers of the app will maintain the app and update it if any updates are necessary.

Packaging: The user downloads the app from the Google Play store. There should be little updates in the future, if bugs appear or if new features get added, e.g. social medias added the to login.