
Technical Application Architecture

Year 4 Project 2014

X00084900, Christopher Newell

1. Use Cases:

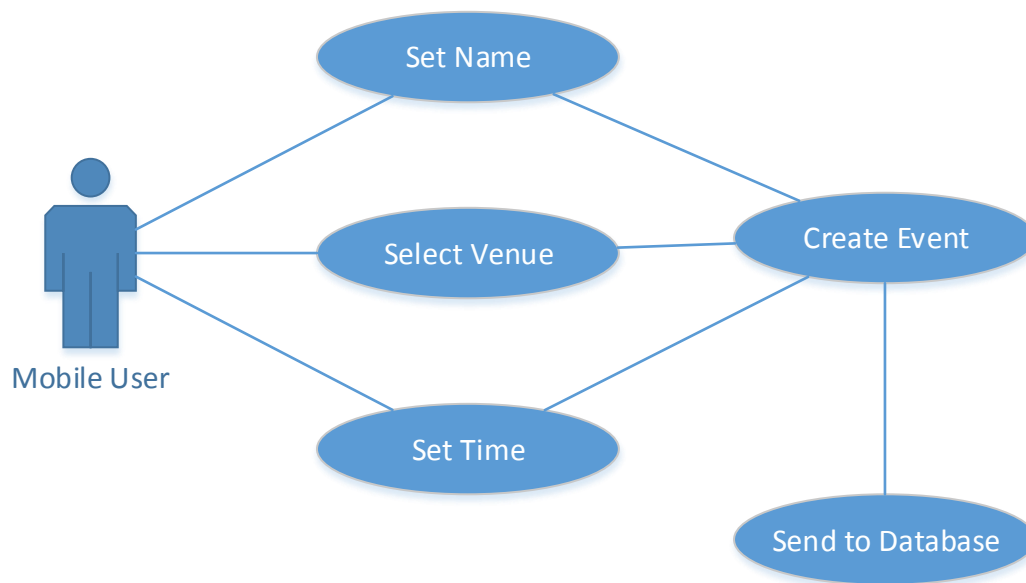
1.1 Title: User creating an event.

1.2 Primary Actor: Mobile User

1.3 Scope: Mobile User uses mobile application to create a new event.

1.4 Level:

1.5 Story: When the user wishes to create an event in order to store information about that event such as: venue, time, date, list of attendees. The user can add data relevant to that specific event and then create it by sending it to the database to be stored and available for retrieval.



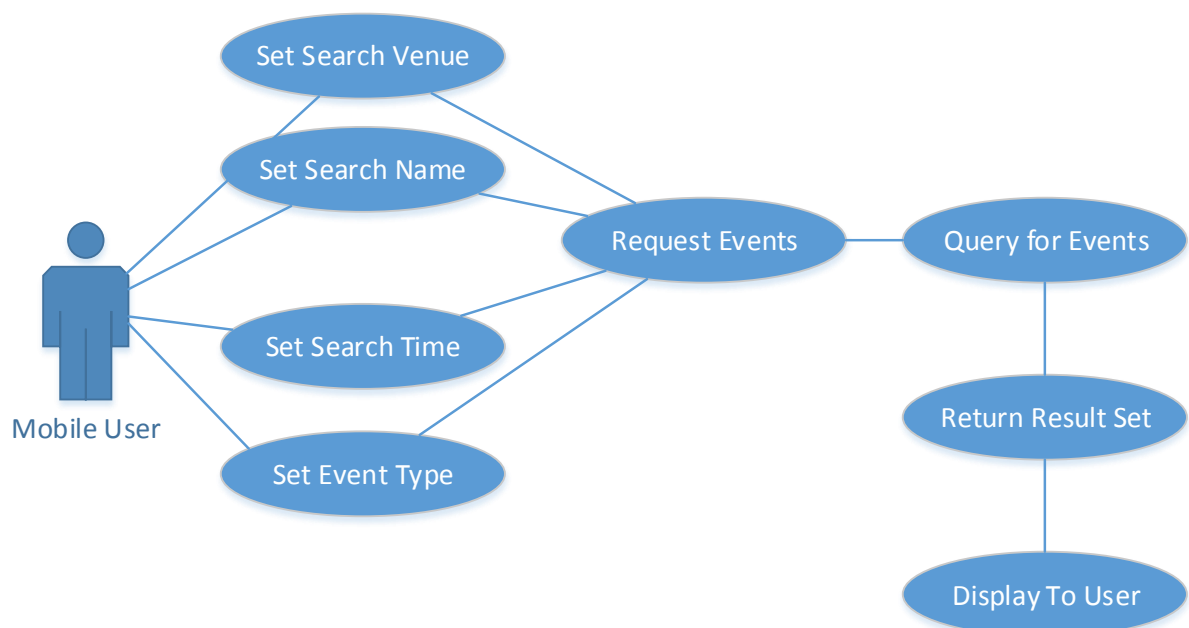
1.1 Title: User retrieving list of events.

1.2 Primary Actor: Mobile User

1.3 Scope: Mobile User uses mobile application to retrieve a list of events.

1.4 Level:

1.5 Story: When the user wishes to retrieve a list of events or a specific event, they set some search criteria which is then taken by the application and sent through the internet to the web API to be sent as a query to the database. the returned result set is then sent back to the user through the internet and displayed on screen, in the event a query fails due to the event not present in the database, the user will be notified.



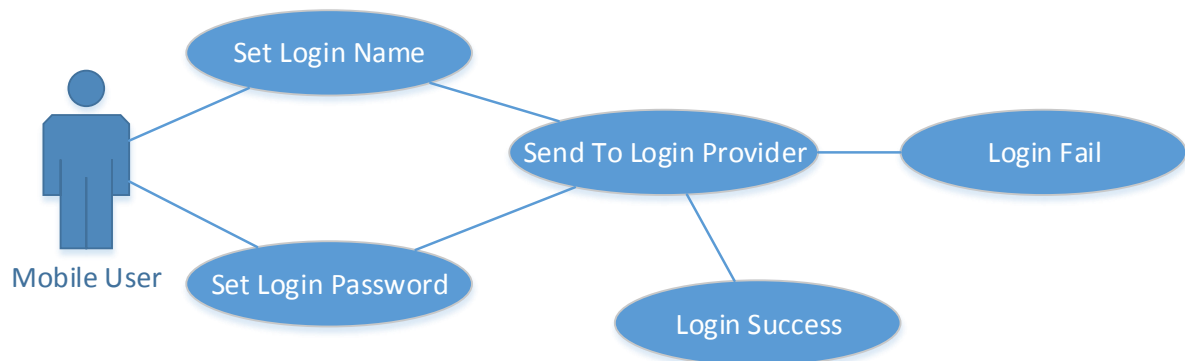
1.1 Title: User logging into application.

1.2 Primary Actor: Mobile User

1.3 Scope: Mobile User uses mobile application to log into application using social media login provider

1.4 Level:

1.5 Story: When the user wishes to log into the application to see what events they have created/are attending, they can log in through some social media login form as a form of security. In the event of success, the user logs in. In the event of failure, the user does not log in and is prompted again.



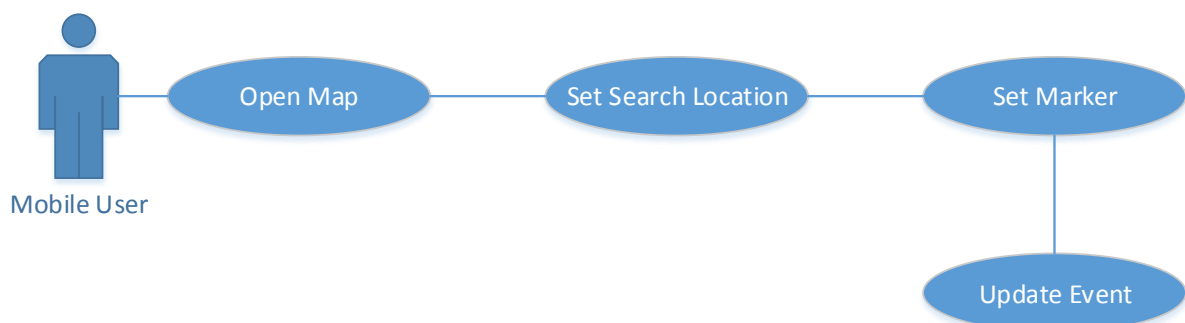
1.1 Title: User selecting a venue on map

1.2 Primary Actor: Mobile User

1.3 Scope: Mobile User uses mobile application to select a venue on map

1.4 Level:

1.5 Story: When the user wishes to select a venue on map in order to specify exactly where the event is taking place and have it displayed as a marker on the map, the user opens the map on the mobile device and places a marker, then the positional data regarding that marker is taken and stored as an attribute of the event they have created.



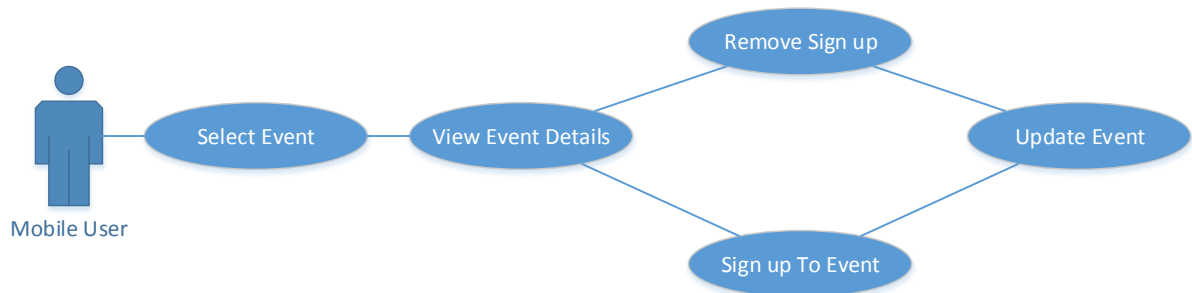
1.1 Title: User signing up to event.

1.2 Primary Actor: Mobile User

1.3 Scope: Mobile User uses mobile application to sign up to event.

1.4 Level:

1.5 Story: When the user wishes to sign up to event, The user selects an event and is shown the specifics of that event. The user can then decide to either sign up or if they already have signed up, delete their sign up



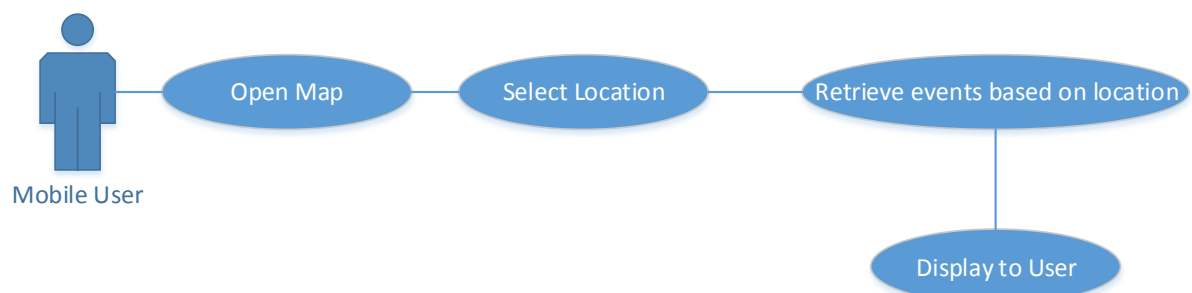
1.1 Title: User searching location on map

1.2 Primary Actor: Mobile User

1.3 Scope: Mobile User uses mobile application to search location on map and view events taking place at that location

1.4 Level:

1.5 Story: When the user wishes to search location on map and view events taking place at that location, The user enters as search criteria the location they wish to view events for. then the application will return a list of events for that location.



2. Technical Architecture:

2.1 Software Components

- Android
- Visual Studio
- Android Studio
- MySQL database
- Windows Azure
- ASP .NET MVC
- Entity Framework
- LINQ
- JSON

Android:

The Android platform is an open source mobile device operating system that has become the most popular mobile device operating system in the world. Android is the chosen platform to build the mobile application for our project because it features a robust collection of native libraries which provide most of the essential functionality for our app such as; handling http requests and responses, providing a customizable Graphical User Interface as well as handling touch screen events such as tapping, sliding or pinching of GUI elements.

On top of this, there exists a vast library of externally developed, open source libraries for specific functionality such as displaying a map to a user and allowing interaction to take place such as marking locations and finding the distance between two points.

Visual Studio:

Microsoft Visual studio is a development tool which supports creation of projects built using ASP .NET MVC as well as integration with the Windows Azure web hosting platform. Using this tool, The creation of a web API to interact with our database will be much simpler to develop and will also be fully supportive of the technologies that we will be taking advantage of.

Android Studio:(TODO)

MySQL Database:

MySQL is an open source relational database management system and is natively integrated as a component of the Windows Azure platform. This technology will be how the project handles data persistence and will have an interface in the form of a RESTful ASP .NET MVC web application. This interface will be how our mobile application will interact with the database.

Windows Azure:

Windows Azure is a web hosting service which comes bundled with a number of useful components to make the building of our web API easier. It will also provide us with hosting of a MySQL database which will allow us to access data over the internet using function calls to our RESTful web API built using C# and ASP .NET MVC. As an added benefit, there exists functionality to automatically publish the web API directly to Windows Azure through Visual Studio.

ASP .NET MVC:

In order to accept and respond to requests sent from the Mobile Application to the web server's database, a RESTful web API is required to handle this traffic. The web API will be built using the web application framework ASP.NET MVC, this framework is used to build web applications using the MVC design. The MVC programming model is the newer version of the previous Web Forms approach to developing web applications and is integrated and supported by Visual Studio.

Entity Framework:

Entity Framework is an open-source ORM (Object/Relational Mapping) framework which removes the requirement to write a ton of data access code which is used for programs to interact with databases (Create Read Update Delete operations). Entity Framework will automate much of the database code we will be required to write otherwise.

LINQ:

LINQ is a component of the Microsoft .NET Framework. It will provide a general purpose querying mechanism to our web API for specific requests sent to the database that will need to return result sets of data to the mobile application. LINQ is integrated into .NET languages such as C# so we will be able to write the necessary functions in C# to retrieve these result sets of data using LINQ.

JSON:

JSON is a lightweight standard for interchanging data across the internet. JSON is a language independent format which makes it excellent for sending data to multiple different mobile platforms such as Windows Phone and iOS devices. JSON uses attribute-value pairs to organise data sent and because of this, it is a much more lightweight and cost-effective tool to transmit data in comparison to another data exchange format such as XML which makes use of tags and bodies to send data.

2.2 Platform Libraries

- Google Maps API for Android V2
- Java (Android SDK)
- C# (.NET MVC)

Google Maps API for Android:

Google Maps Android API V2 is a library which allows developers of Android applications to embed maps into an activity.

These maps offer a variety of features such as 3D maps, indoor maps, satellite maps of large stretches of land, user customization of a map by placing markers to indicate places and points of interest.

This library also works with a large variety of related APIs such as the Places API which access a database of over 100 million points of interest, the Location APIs which can allow a mobile application to be location-aware (i.e. the mobile application will be able to pinpoint the location of the device it's running on in the world).

Java(Android SDK):

The Android SDK is a software development kit that allows developers to create applications for the Android platform. This platform includes tools such as; Sample projects to explain pieces of functionality, development tools to create and run Android projects, libraries of core functionality to build Android applications for a mobile device, a device emulator for testing a mobile Android application on a wide range of devices with different system specifications and hardware configurations.

Android applications are written in the Java programming language which means that Java will be the language chosen to build the mobile application for the project.

C#(.NET MVC):

As outlined above in the Software Components section of this Technical Application Architecture document, The web service or application which will service requests to and from the MySQL database will be built using ASP .NET MVC and written in the .NET language C#. This will allow the project to take full advantage of technologies and support featured in the .NET framework for building MVC applications and hosting them through a web server.

2.3 Distribution and Deployment

- Cloud Database with RESTful web API
- Log in through social media

Cloud Database with RESTful web API

This project will make use of a MySQL database hosted on a Microsoft Azure Web Server. This database will be how the project maintains data persistence. The database will be required to store, update, delete and retrieve data for our mobile application.

Mobile Devices have limited system resources, especially when internet requests are involved. Because of this limitation, opening up and maintaining a continuous session with the database is a bad idea due to a number of reasons:

- maintaining sessions from a large, distributed collection of clients over the internet will be a very large load for the database to manage.
- Mobile devices are likely to simply time out of the session and shut it down if the request/response/authentication takes any longer than a couple of seconds
- The number of mobile platforms will be very limited, each requiring a connecting class for each system's programming language to set up the session to the database
- Security becomes an issue when you have multiple different platforms each using its own language to request data from the database

Because of these reasons, we require a different approach to service requests sent over the internet to our database and to send results back to our mobile application.

A RESTful implementation of a web API will act as the mediator between the database and the mobile application. This API will make use of simple Http requests to different URLs located in our web API. These URLs will be tied to functions which then act on the data in the database and return the data back to our application.

This approach does away with having to maintain a session to the database with the mobile application and also opens up the application to be ported to other platforms such as iOS or Windows Phone as there is no platform-specific code involved. The only thing the device will need to access the database is the ability to make Http requests which if they have a browser installed, is a likely scenario.

Logging in through social media websites

Managing User Authentication is a critically important area of mobile development. If it is handled poorly or oversights are allowed to persist, the user's data or device itself could be at risk of compromise or attack.

Rather than risk a security flaw in building a user authentication system from scratch, a social media service will provide the log in operation which will then provide the mobile application with a means of user authentication for this project.

there are plenty of different authentication providers such as Twitter, Facebook and Google+. each has support for adding a log in button to the application which will then ask the user to provide their account details for each of those social media platforms.

Managing user authentication this way will save a lot of development time that would otherwise be spent on building an authentication service which may end up becoming vulnerable and insecure. Including the ability to log in with a social media account and then using the access token that creates to authenticate users is a much simpler solution.

2.4 Risks

- Windows Azure being unavailable
- Google Maps being unavailable
- Level of experience with technologies involved with the project

Windows Azure Unavailability:

The Windows Azure web hosting platform is this project's only means of servicing requests made through the internet from the project's mobile application to the database. In fact, the database itself will be contained entirely on their web servers and will be inaccessible in the event of the Windows Azure service becoming unavailable for whatever reason.

This also means that once the subscription to Windows Azure runs out, both the web service, the MySQL database and the data pertaining to the mobile device's users will be inaccessible until the subscription is renewed.

Google Maps Unavailability:

The only library the project uses to display maps and provide map customisation functionality to the user of the mobile application will come from the Google Maps API for Android V2 library.

If, for whatever reason this service becomes unavailable then the mobile application will be unable to display maps to the users, provide map customization features such as marker placement and location searching, any other APIs which are used to provide additional functionality such as the Places API or Locations API will also be affected by this.

Inexperience with technologies involved with the project:

This project makes use of a variety of technologies which are used to create web based applications. Each of these technologies will require some degree of revision or practice to use effectively for the project. In the event that a piece of functionality does not behave as intended or there is uncertainty on how to achieve some functionality, this scenario could cause the project to run into some extra difficulties which require more work to resolve.

3. Prototype

3.1 Prototype Deliverable for week 8

Use Cases to be delivered:

1. User Creating an Event
2. User retrieving list of events
3. User signing up to event

3.1 Testing Strategy

The Testing Strategy for this iteration will be focused on building the web API and Android application and setting up some of the basic functionality that the project will require.

In the Android application's case, the usage of emulators included in the Android SDK to test different versions of Android will be very useful to ensure that the application can run on as wide a variety of devices and versions of Android as possible.

For the web application, the use of Unit Tests for the different query functions the API will possess and dummy data to simulate a database connection will initially allow us to test that communication between the mobile device and the web API through the internet using Http requests is working and that the correct data is being sent and retrieved through the internet.

This will also allow us to test how we will transmit the data using JSON and how each end of the project will handle parsing this data and representing it correctly.

3.2 Prototype Deliverable for week 11

Use Cases to be delivered:

1. User logging into app
2. User selecting venue on map
3. User searching location on map

3.2 Testing Strategy

The testing strategy for this prototype will be focused on enabling Location based data using the Google maps API and how that data will be taken from the user input on the map and saved as an attribute of an event.

The web API will continue to make use of Unit Tests to ensure that no new code introduced to the API will have an adverse effect on how the older functions operate.

For the Android application, emulators for different devices and versions of Android will be used to continue testing changes and updates to the project