Use Cases and Logical Architecture

* XID: X00119025
* Name: Martin Casey
* Project Title: Community Application

**Provide at least 6 Use-cases describing the functionality of the proposed system**

## Section 1: For Each Use Case:

|  |  |
| --- | --- |
| Title (goal) | What is the name of this function? |
| Primary Actor | Who is the user? |
| Story | Describe in detail what happens here |

Below is the current work available for this draft of the UCLA.

**Use Case 1: Access the Cloud**

|  |  |
| --- | --- |
| Use Case Element | The user of the application accesses the cloud hosting the application. |
| Use Case Number | 1 |
| Application | This function relates to the cloud server. |
| Use Case Name | Access Cloud |
| Use Case Description | The intended user of the application is about to access the application, while the application begins running, the user must connect to the cloud running the application from their mobile device. |
| Primary Actor | The user who is trying to access the application. |
| Precondition | 1. The user must have access to an internet service. 2. The user must be logged in to the application. |
| Trigger | The user logging in to the application. |
| Basic Flow | 1. The user logs into the application. 2. The application sends the users details via HTTP to the cloud server hosting the application. 3. The cloud server processes and sends off the user information to the server logs table. 4. The server logs table stores the user data. 5. The cloud server sends the access permissions to the user’s device via HTTP. 6. The user can now access the services provided by the application.   The basic flow should be the events of the use case when everything is perfect; there are no errors, no exceptions. This is the "happy day scenario". The exceptions will be handled in the "Alternate Flows" section. |
| Alternate Flows | 1. The user enters incorrect details and is unable to log into the application. 2. The cloud server is unavailable due to technical issues, therefore now services can be provided to the user. |

Use Case: Access Cloud

User Log

Cloud Server

User

Send Services

Store Data

HTTP Data

**Use Case 2: Search for Activity**

|  |  |
| --- | --- |
| Use Case Element | The user of the application is searching for an activity in the community for them to partake in. |
| Use Case Number | 2 |
| Application | The function relates to the Community application. |
| Use Case Name | Search for Activity |
| Use Case Description | The user of the application is deciding to look into activities that are hosted in their local community to find one they are interested in. |
| Primary Actor | The user who is trying to search for an activity in the community to take part in. |
| Precondition | 1. The user must have access to an internet service. 2. The user must be logged in to the application. |
| Trigger | The user accessing the search bar in the Community application. |
| Basic Flow | 1. The user inputs areas of interest into the search bar of the application. 2. The application sends a HTTP request to the cloud server to return area locations that host the activity the user is searching for. 3. The cloud server accesses the stored log of organisations registered on the application. 4. The log of stored information is queried using the request sent to the could from the application. 5. The log returns a collection of all organisations that host the activity the user is interested in. 6. The cloud server sends the collection to the application via HTTP. 7. The application loads the results of the collection of organisations and the activities they host to the user in a results list. |
| Alternate Flows | 1. The user enters their areas of interest incorrectly into the search bar. The application analyses the text the user entered and calculates what it believed the user tried to input. It then sends this calculated text to the cloud server, and after it retrieves some results, that application asks the user “Is this what you meant?” while displaying the search results of what the application thinks the user meant to input. 2. The activity the user was interested in does not exists, or could not be found among the organisations registered on the application. The cloud server returns this information to the application via HTTP, where the application displays a message to the user stating that it could not find the activity the user is looking for. |

Use Case: Search for Activity

Cloud Server

User

Send Request

Display Results

Input text to search bar

Send Collection

Return Collection

Query Log

Query Request

Applications

Organisation Log

**Use Case 3: Book Ticket for Event**

|  |  |
| --- | --- |
| Use Case Element | The user of the application is booking a for an event |
| Use Case Number | 3 |
| Application | The function relates to the Community application. |
| Use Case Name | Book Ticket for Event |
| Use Case Description | The user of the application is looking to book themselves a ticket for an event/activity they are interested to part take in. |
| Primary Actor | The user who is trying to book a ticket for an event/activity. |
| Precondition | 1. The user must have searched for an activity using the applications search bar. See use case 2. |
| Trigger | The user clicking on the “book ticket” button. |
| Basic Flow | 1. The user clicks on the book ticket button after finding an activity they want to part-take in. 2. The user selects the time and day they wish to part-take in the event/activity. 3. The application sends a book event request via HTTP to the cloud server. 4. The cloud server retrieves the request. 5. The cloud server accesses the stored log organisations registered on the application. 6. The organisation log is queried to find a free slot for a booking on the activity the user wishes to partake in. 7. A free slot is found and is then taken up by the user’s request. 8. The log informs the server the booking was made. 9. The cloud server sends a confirmation HTTP message to the application. 10. The application displays a message to the user stating that they have been booked for the activity. |
| Alternate Flows | 1. No free slot is found, the organisation log informs the cloud server that a booking is not possible. The cloud server sends a message to the application that the booking the user wants is not possible. The application displays a message to the screen stating to the user that they will not be able to book this event. |

Use Case: Book Ticket for Event

Display Booking

Applications

Book Event

User

Send Confirmation

Return Booking

Query Request

Query Booking

Organisation Log

Cloud Server

Book Request

Select Time & Day

**Use Case 4: Register Organisation**

|  |  |
| --- | --- |
| Use Case Element | The business user wants to register their organisation onto the application. |
| Use Case Number | 4 |
| Application | The function relates to the cloud server |
| Use Case Name | Register Organisation |
| Use Case Description | A business user is made aware of the application and wishes to register their organisation on the application, so that users of the application can be informed of events the organisation hosts in the community. |
| Primary Actor | The business user trying to register their organisation on the application. |
| Precondition | 1. The business user must have access to the internet. 2. The business user must have a user account on the application. |
| Trigger | The business user goes to the options menu and clicks “register organisation”. |
| Basic Flow | 1. The business user goes to the options menu and clicks register organisation. 2. The business user fills out a form on the application, detailing information about their organisation, such as name, location, and activities hosted. 3. The application sends the information from the form to the cloud server. 4. The cloud server sends the information of the form to the organisation log, to be processed. 5. The organisation log processes the information and creates a new organisation account where it stores the information. 6. The organisation log informs the cloud server that a new organisation account has been created. 7. The cloud server sends a confirmation message to the application via HTTP, informing the application that a new organisation account had been made. 8. The application informs the business user that their organisation is now registered on the application. |
| Alternate Flows | 1. The organisation could not be logged if the cloud service runs into a scaling issue, in where it becomes unable to increase the amount of data it stores due to a limit placed on the service by the owner of the cloud account for financial reasons. |

Use Case: Register Organisation

Display Registration

Applications

Register Organisation

Business User

Send Confirmation

Organisation Log

Cloud Server

Send Form

Fill out Form

Return Account Information

Create new Organisation Account

Register Request

**Use Case 5: Examine Crime Areas**

|  |  |
| --- | --- |
| Use Case Element | The user wants to examine what areas in the local community have had recent crime activity. |
| Use Case Number | 5 |
| Application | The function relates to the GPS functionality of the application. |
| Use Case Name | Examine Crime Areas |
| Use Case Description | A user wants to examine the local area around them and see which areas have been prone to crime activities in recent time. |
| Primary Actor | The user trying to use the application to find out what areas in the community are currently prone to crime. |
| Precondition | 1. The business user must have access to the internet. 2. The area the user is operating in has access to satellite and cell tower transmissions. |
| Trigger | The user clicks on the check criminal activity button in the application. |
| Basic Flow | 1. The user clicks the check crime in local area button. 2. The satellites that record GPS information transmits their current location information to the cell towers in the local area. 3. The application reads the location transmissions emitted by the cell towers in the area, to understand the user’s current location and the area around them. 4. The application sends the users area location to the cloud server. 5. The cloud server sends a check crime request to the crime log. 6. The crime log quires its records for crime activity in the area the user is in, based on recent activity only. 7. The crime log returns the crime activity list to the cloud server. 8. The cloud server sends the crime activity list to the application. 9. The application loads up its map and highlights elements of the map using the list of crime activity that it received from the cloud server. 10. The user clicks on one of the highlighted areas on the map to be presented with text, detailing what crime took place recently. |
| Alternate Flows | 1. There are now satellites in range or available due to technical information. The application at the current moment in time can’t receive any location based information. 2. The crime log could not find an any recent crime activity in the user’s local area. The crime log returns a no crime activity message to the cloud server. The cloud server sends a no crime in area message to the application. The application informs the user that there is no reporting of any crime activity in their local area. |

Use Case: Examine Crime Areas

User

Cloud Server

Organisation Log

Applications

Check Crime in Area

Show Crimes on Map

Check Crime Request

Query Criminal Activity

Return Criminal Activity

Send Crime List

Send User Location

Satellite

Cell-Tower

Send GPS Location

Transmit Location

Click on Crime

**Use Case 6: Access Map of Local Area**

|  |  |
| --- | --- |
| Use Case Element | The user wants to access the map functionality on the application to display a map of the user’s local area. |
| Use Case Number | 6 |
| Application | The function relates to the GPS functionality of the application. |
| Use Case Name | Access Map of Local Area. |
| Use Case Description | A user wants to examine a map of the local area that they are currently located in while using the application. |
| Primary Actor | The user trying to use the application to display a map of the local area they are currently located in. |
| Precondition | 1. The business user must have access to the internet. 2. The area the user is operating in has access to satellite and cell tower transmissions. |
| Trigger | The user clicks on the map button in the application. |
| Basic Flow | 1. The user clicks the map button on in the application. 2. The satellites that record GPS information transmits their current location information to the cell towers in the local area. 3. The application reads the location transmissions emitted by the cell towers in the area, to understand the user’s current location and the area around them. 4. The application sends the users location information to the cloud server. 5. The cloud server sends an activity request to the organisation log. 6. The organisation log quires the records for organisations in the local area. 7. The organisation log returns a list of activities to the cloud server. 8. The cloud server sends a collection of activities in the local area to the application. 9. The application generates a map of the local area based on the location the user is currently located in, highlighting any activities that are near the user’s current location. 10. The application displays the map of the local area to the user. |
| Alternate Flows | 1. There are now satellites in range or available due to technical information. The application at the current moment in time can’t receive any location based information. Therefore, application cannot display a map of the current location. 2. The organisation log could not find any activates present in the local area. The organisation log sends a no activities found message to the cloud server. The cloud server sends a no activates in the area message in the area. The application generates a map of the local area to the user, but now activates are highlighted on the map as noon can be found. |

Use Case: Access Map of Local Area

Show Map

Generate Map

Applications

User

Check Crime in Area

Organisation Log

Cloud Server

Send User Location

Send Activities Collection

Send GPS Location

Cell-Tower

Return Activities

Check Activity Request

Transmit Location

Satellite

Query Activities

Section 3

Here we will talk about the details of each component in the logical architecture diagram.

* User:
  + The user of the application. They will access and interact with the application using their mobile phone.
* Mobile Phone:
  + The mobile phone will act as the access point of the application for the user. The user will need to use the download app of the application to login and experience the features of the application.
  + The mobile phone will run on android operating system, and must have location tracking enabled in the phones settings for it to receive GPS transmissions so that the application can work.
* Satellite:
  + The satellite hovers around the earth with other satellites in the outer reaches of space as part of the Global Positioning Network. From here it transmits GPS signals down to the earth to be received by GPS enabled device, such as a mobile phone.
* GPS Signal:
  + Signal transmitted by satellite to GPS enabled devices. The signals are used by these devices to pin-point the latitude and longitude co-ordinates of a GPS enabled device, or the co-ordinates of the areas around the device.
* HTTPS:
  + The Hyper Text Transfer Protocol Secure that is sent to the cloud from the application on the mobile device via internet.
  + The HTTPS contains the login details of the mobile user on the application, and is designed to ensure that while the details are transferred from phone to cloud, they are not under threat by any malicious hackers.
* Azure Cloud:
  + An online based storage platform service, that is supported by a network of servers across multiple regions.
  + The cloud hosts the application and its features, features which are only accessible after a user has logged in to the application.
* Azure Storage Encryption Service:
  + An encryption service provided by Azure that protects data when its being sent to storage in the database in the Azure cloud.
  + The service allows for the safe transfer of data for while an application exists within the Azure cloud.
* Cloud Server:
  + The compartment in the cloud that acts as the access point for data to be transferred back and father between the cloud services and the application, and the application and the mobile device accessing the application.
* Request File:
  + A file that server sends to the database when it needs it to perform an action.
  + The request file is sent in a JSON format across the network.
* Cloud Database:
  + A relational database service that is accessible in the cloud.
  + The database stores information in an SQL format about details of application that are needed to be stored. Such as account information of users. Data related to activates, such as location and organisation hosting the activity.
* SQL Query
  + An SQL command that is used on tables in the database.
  + This is used every time a value needs to be stored or retrieved from the database, such as in instances of returning the location of an activity, or creating a new record for a new account that is being registered on the application.
* Storage Table:
  + Used to store data values in the database.
  + A database will consist of many variations of these with specific purposes.
* Notification Request:
  + A JSON formatted file sent from the cloud server to the Azure notifications hub to request that a new notification be created for the user of the application.
* Azure Notifications Hub:
  + A service provides by the Azure cloud that creates notifications for users of applications.
  + The notifications are created in an abstract format, so that they can be sent off to the users respected mobile service notifications provider.
* Abstract Notification:
  + A HTTP formatted piece of code that contains specific information about the interests of the user of the application.
  + This code contains all the recommended consist of a list of all the recommended services that the user may be interested in.
* Platform Notification Provider:
  + A separate cloud service provided by the users selected service provider. Which also provided the users phone with the operating system it uses.
  + This service receives the abstract notification sent to it by the Azure notifications hub and modifies it so that it works with the users related mobile operating system.
* Notification:
  + A modified version of the abstract notification sent out by the Azure notifications hub.
  + Once it reached the users mobile phone, it will display a notification pop up indicating to the user what services the application recommends to the user, and possibly what location the user has entered if they are on the move with the application.
* Community Application
  + This is the area in the cloud where the Community Application and its features are accessed and by and provided to the user.
  + The application runs in an MVC asp.net format. The functionality of the separate parts of the MVC framework are as follows:
  + Model:
    - Houses the various integration features of the application.
    - Acts as the section in the application which communicates with the database that is linked to the application.
    - The model is programmed using a combination of C# code, with SQL integration and some light HTML services also used.
  + View:
    - Acts of the interface between the user and the application.
    - Is responsible for display all Graphical user interface of the application to the user.
    - Built with a combination of HTML and CSS, with integration to the Model or Controller to send requests to the various methods in the application so that the application can provide the requested services.
  + Controller:
    - The controller stores most of methods that will be used by the application.
    - The controller is built only using C#, as all its functionality only relates to the backend of the application.
  + Google Maps JavaScript API:
    - The API that allows for the application to understand both the latitude and longitude of GPS transmissions, as well as use those same transmissions to generate a map of the area the use is in.
    - The API is constructed of JavaScript protocols which could be modified by the developer of the application, in order to meet the applications exact requirements.
* Azure Application Service:
  + The Azure application service acts an application engine to manage, develop and run the application, but it also allows for the ability to deploy an application to the Azure cloud.
  + Is the primary area in the cloud where the application is hosted, and is where coded messages that affect the functions of the application are sent to, or in other cases received from.