|  |
| --- |
|  |
| Traveller App |
|  |
|  |
| **Adithya Bharadwaj**  **Version 1.0**  **25** |
|  |

|  |
| --- |
|  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **By** | **Comment** |
| 1.0 | 2016-07-25 | Adithya Bharadwaj | First draft |

Contents

1 Introduction 3

2 Conceptual Solution 3

3 Logical Solution 10

4 Physical Solution 11

# Introduction

This document provides a description of the solution on the following two levels:

* Conceptual; answers the question "What?", and include the overview and requirements
* Logical; answers the question "How?", and outline the technical solution
* Physical; answers the question "With what?", and describe the actual products and technologies used as well as solution details (service interface, database structure, etc)

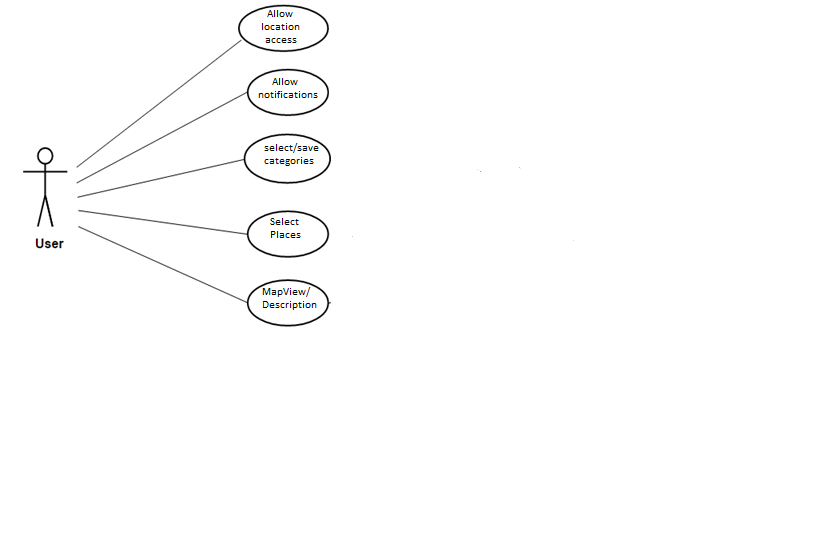
During the development of the system, this document will be continuously updated.

# Conceptual Solution

The main purpose of the "Traveller App" is to allow users to get notified when the user location changes. This app will help user by providing the nearest places to visit based on the user choice.

This app will help user to get everything through app without going outside and it is time saving.

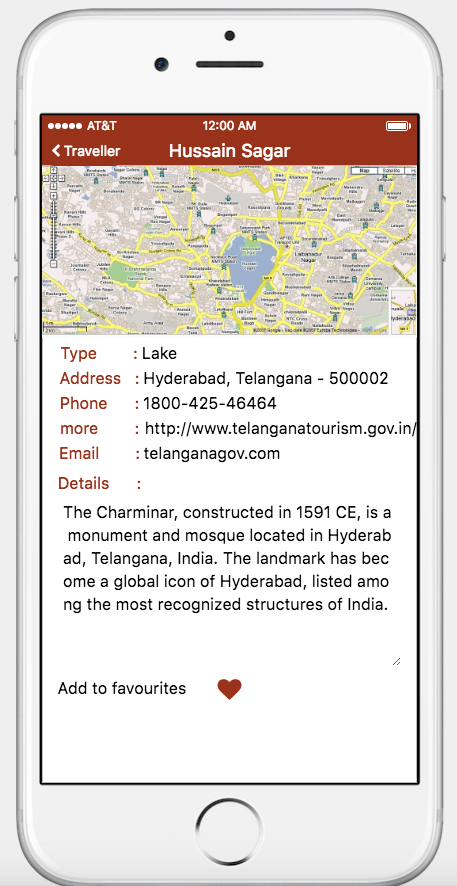
In Figure below, you can see the use case that apply to the app, on the basis of this diagram the functionality of the app will be explained and be supported by mockup designs.



**Here is an overview of the app for iOS**

|  |  |
| --- | --- |
| *Macintosh HD:Users:adithyabharadwaj:Desktop:Screen Shot 2016-07-28 at 4.02.30 PM.png* | **2.1 User Permissions**  **2.1.1 Push Notifications**  When the app starts, it will ask user to allow access to Push Notifications.  On successful access, user is able to get notifications  If user does not allow access for Notifications, he can change it at any point of time by going to the settings App on IOS device. |
| Macintosh HD:Users:adithyabharadwaj:Desktop:Screen Shot 2016-07-28 at 4.02.48 PM.png | **2.1.2 Location Access**  User is prompted to allow access to his location.  On Successful access, the app always access user location and notifies him about the nearest places.  If user does not allow access for Location, he can change it at any point of time by going to the settings App on IOS device. |
| Macintosh HD:Users:adithyabharadwaj:Desktop:Screen Shot 2016-08-08 at 11.49.03 AM.png | **2.2 Home Screen**  This screen shows list of the nearest places.  It will includes:   * Image of place * Name of place * Type of place * Rating for place * Address of place   It displays data in a table view, so that the user can click on any row (image/name) to view the detailed description of the place and the root map. |
| Macintosh HD:Users:adithyabharadwaj:Desktop:Screen Shot 2016-08-08 at 11.50.14 AM.png | **2.3 Search Screen**  In Search Screen, user can select the categories and search for the places based on his current location or manually entering the city/state/zip code/address  The User can change his preferences at any point of time in the app by clicking on the search tab bar item . |
| Macintosh HD:Users:adithyabharadwaj:Desktop:Screen Shot 2016-08-08 at 11.51.24 AM.png | **2.4 Favorites Screen**  This screen allows user to view his/her added favorite places.  It displays data in a table view, so that the user can click on any row (image/name) to view the detailed description of the place and the root map. |

2.5 Details Screen

This screen allows user to view the details of the selected place and the route map to the place from his/her location.

It will include :

* Type of place
* Address of place
* Phone number
* Email (if available)
* More (link)

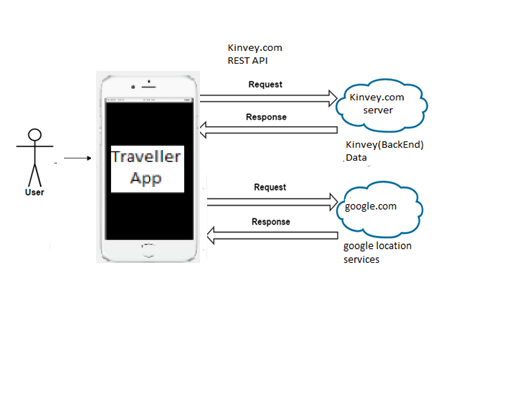
Here are a number of non-functional requirements:

1. The communication between the apps and the web service should be compressed
2. The communication between the apps and the web service should be secure (encrypted)
3. The web service should not be directly available on the Internet
4. The web service should have full access to the back-end systems
5. The web service should be able to handle a load of ... request per ...
6. The average response time from the web service should be ... seconds
7. The availability of the web service will be 99.x%
8. ...

Any other requirements that will affect the technical solution should also be included here.

# Logical Solution

Here is the overview of logical solution.



The significance of the system is the mobile service that provides suitable information or data by connecting to the back-end systems.

Using back-end system, it is able to provide service to number of application on various platforms , and all integration issues will be handled.

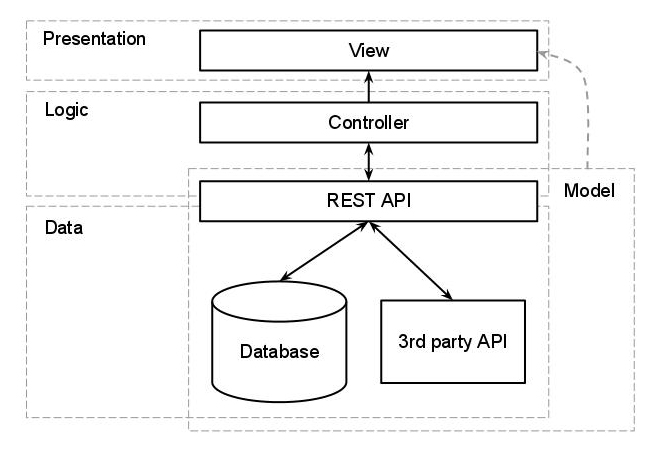
Kinvey utilizes Firewalls to restrict access to systems from external access to the networks and between systems internally. By default all access from outside is denied and its allowed only for ports and protocols are allowed based on business need.

Each system is assigned to a firewall security group based on the system’s function. Security groups restrict access to only the ports and protocols required for a system’s specific function to mitigate risk.

Google infrastructure provides DDoS (Distributed Denial of Service) mitigation techniques including TCP (Transmission Control Protocol) Sync cookies and connection rate limiting in addition to maintaining multiple backbone connections and internal bandwidth capacity that exceeds the Internet carrier supplied bandwidth.

# Physical Solution

Here is an overview of the physical solution:

****

**4.1**

Traveller Application follows the pattern shown in the figure above

**4.1.1 Presentation**

The presentation layer of the app consist of view which includes Storyboard, where the layout will be defined. This is top layer of the application architecture. This layer does not have any knowledge about the Logic and Data.

**4.1.2 Logic**

In iOS, main purpose of the View is to display data from application's model objects. We can easily configure and reuse views. UIKit and AppKit framework provides various classes to customize view and make UI interactive. View can know about changes in model through Controller.

Controller is intermediate between view and model. Views are appeared on the screen by using ViewController's Lifecycle. There are different methods which are called during viewController lifecycle. By using this method view will be appear on screen. This methods are useful to set layout, View can be appear and disappear according to task such as Web service call, refreshing view.

Application will use Lifecycle methods to increase the performance and reduce utilization of web service call i.e. REST API or loading external data. Logic will be written in Controller that will able to get all the data from Model and display it on View.

**4.1.3 Data**

Traveller app will use "Kinvey.com". To store and retrieve data app will use Kinvey REST API. On Kinvey , First it creates a table according to requirement then Kinvey will provide API(url) for each and every task such as Push Notification, store and retrieve the data. Kinvey provide App Secret and App ID which will be used to verify the app on kinvey server.

When application request for any service like Storing or Retrieving Data., It should have add App Secret and App ID while requesting for service. When application hit the url , it will be returned data in JSON format.

**4.1.4 Model**

The Kinvey API sends back data in the form of JSON. To call network API app will use AFNetworking/ NSURLSession class and data will be in JSON format. To convert data into required format app uses NSJSONSerialization. NSJSONSerialization is used to convert JSON data into Foundation Objects i.e. NSString, NSArray, NSDictionary. It is helpful to bind that converted data to UI. Data return by NSJSONSerialization is of NSData type and it is having Apple parser built-in that is why it's faster in terms of operation.

The main purpose of the model is to represent data used in application. Using Model objects application organize data in suitable structure and that will be sent to view to display to the User.

This app will use Kinvey.com to store/retrieve data. The Kinvey platform provides a complete backend solution for mobile application. To integrate Kinvey in app, we need App Secret and App ID.

**4.2 API Connection-**

**4.2.1 API calls**

Here is an overview of the API calls:

|  |  |
| --- | --- |
| Name | Push Notifications |
| Documentation link | http://devcenter.kinvey.com/ios/guides/push |
| Usage | Used to send Push Notifications to inform users about updates, new services available, etc,.. |

|  |  |
| --- | --- |
| Name | Location |
| Documentation link | http://devcenter.kinvey.com/ios/guides/location |
| Usage | Used to allow user to notify about the nearest places using latitude and longitude. |

|  |  |
| --- | --- |
| Name | Data Store |
| Documentation link | http://devcenter.kinvey.com/ios/guides/datastore |
| Usage | Used to Store and Retrieve the information of the nearest places based on the location services. |

**4.3 Techniques**

**4.3.1 OS**

Traveller Application will support iOS version 8.1 and Above.

**4.3.2 Device Support**

Traveller App support iPhone .

**4.3.3 Orientation**

Traveller application will support portrait mode.