|  |
| --- |
|  |
| Offers Zone App |
|  |
|  |
| **Tushar Shende**  **Version 1.0**  **16** |
|  |

|  |
| --- |
|  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **By** | **Comment** |
| 1.0 | 2016-02-16 | Tushar Shende | 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.

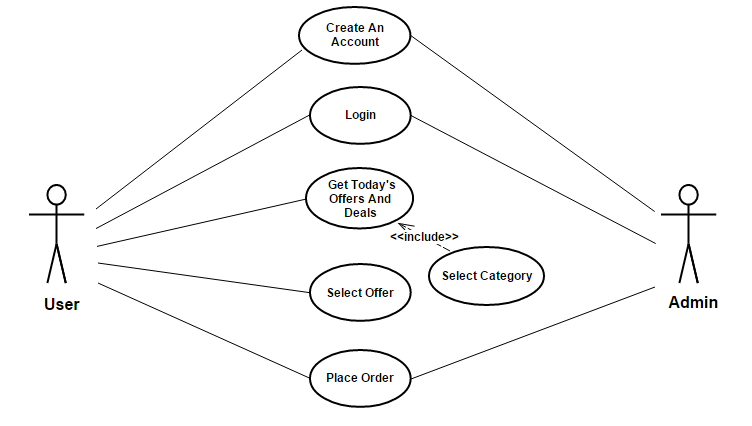
# Conceptual Solution

The main purpose of the "Offers Zone App" is to allow users to search for offers and deals from seller. This app will help user by fundamental procedure of e-commerce application. Using this app user can set their choice while buying product.

This app will help user to track offers on daily/weekly basis. User can also see the offers by selecting categories. User can buy product by redirecting to seller's site.

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**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C:\Users\tushende\Desktop\MockUp\User Authentication.png | **2.1 User Authentication**  When the app starts, it will ask user to login. User can login either by User Id or by email-id and password.  On successful login user will be navigated to Home Screen.  If user is not registered, then user can register by clicking on SIGN UP button.  When SIGN UP button is clicked, the SIGN UP screen is shown (see 2.2 below). | | | |
| C:\Users\tushende\Desktop\MockUp\Registeration.png | **2.2 User Registration**  For registration, User will have to submit information like User Id, Email Id and Mobile No. and Password. | | | |
| C:\Users\tushende\Desktop\MockUp\Daily Offers.png | **2.3 Home Screen (Daily Offer's)**  On Home screen, In Tab Bar User will have following three tabs :   * Daily Offers * Amazon Offers * Settings   For details refer 2.3,2.5, 2.6 below.  On Daily Offers screen, User will get daily offers like Today's Deal or Deal Of The Day from various e-commerce site like Amazon, Flipkart etc.  All offers will be shown to user in tableview like list. Offers have small description like Image, Title, Offer, Price.  User have to click on offer to see product detail. After clicked on offer user will be navigated to offer/product detail screen refer 2.4. | | | |
| C:\Users\tushende\Desktop\MockUp\Daily Offers.png | **2.4 Offer Detail**  This screen shows details about product.  It will include following points :   * Product Image and Title * Manufacturer Name * Actual Price * Offer Price * Product Description in 3-4 lines * Buy Option.   If user click on Buy button , it will redirect user to corresponding E-commerce website like Amazon, Flipkart in WebView. Because user get offers from different website. User can share offer with others on Facebook.  User can purchase the product from their portal. | | | |
| C:\Users\tushende\Desktop\MockUp\Web View Purchase.png | **2.4.1 Purchase Detail**  After clicked on Buy button, User will be redirected to this screen.  User will purchase the product from offer's corresponding E-commerce portal.  Portal will be shown in WebView. | | | |
| C:\Users\tushende\Desktop\MockUp\Amazon Categories.png  C:\Users\tushende\Desktop\MockUp\Amazon Offers.png | **2.5 Amazon (E-Commerce Site) screen**  On Second Tab , User will navigate to Amazon screen where user will get different categories of products.  User have to select one category out of them according to their choice.  Categories will be shown in List i.e. table view.  **2.5.1 Amazon Offers**  After selecting category from Amazon tab. User will be getting various offers of product from selected category.  Offers details are same as Daily offers screen.  It will be having details like Product Image, Title, Offer and Price. | | | |
| C:\Users\tushende\Desktop\MockUp\Amazon Offer Detail.png | **2.5.2 Amazon Offer Detail**  This screen will show the detail of Amazon offer about product/offer selected by user.  This screen includes Product Image and Title, Manufacturer Name, Actual and Offer Price and Product description in 3-4 lines. User can also share offer with others on Facebook.  User have to click on Buy button to purchase product and User will be redirected to Amazon Web portal in Webview which is same as Purchase Detail refer 2.4.1. | | |
| C:\Users\tushende\Desktop\MockUp\Settings.png | | **2.6 Settings**  On Settings Tab , There are Three Options   * Edit User Profile * About Us * Logout   User can edit personal information by clicking on Edit User Profile.  User will get information in POP-UP about Application in About Us.  User can logout of the application by using logout button. | |
| C:\Users\tushende\Desktop\MockUp\User Profile.png | | | **2.7 User Profile**  This screen will help the user to update Profile Detail.  User can update personal details like Name, Email-Id, Mobile No. by clicking on edit button.  User can save changes using Save button. |
|  | | |  |

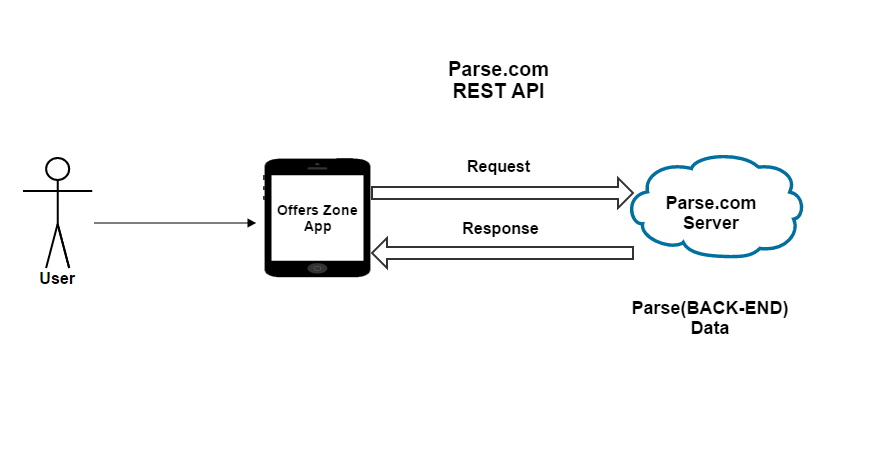
Here are a number of non-functional requirements:

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 apps would need to authenticate(i. e . password) to use the web service
4. The web service should not be directly available on the Internet
5. The web service should have full access to the back-end systems
6. The web service should be able to handle a load of ... request per ...
7. The average response time from the web service should be ... seconds
8. The availability of the web service will be 99.x%
9. ...

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

# 3 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.

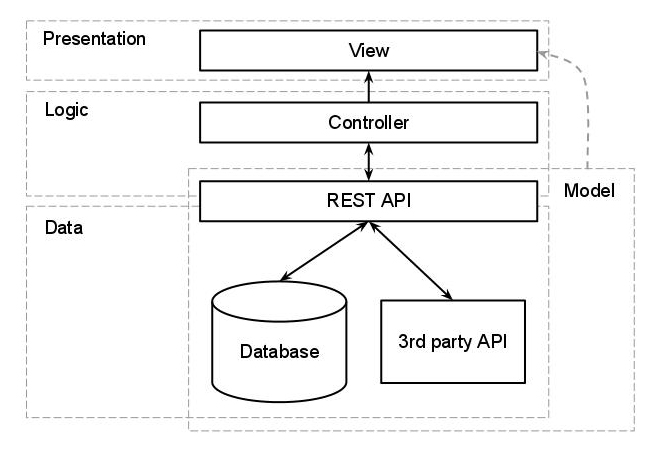
Parse utilizes Firewalls to restrict access to systems from external access to the networks and between systems internally. By default all access from outside is denied and its allowed only for ports and protocols are allowed based on business need.

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.

Parse 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**

Offers Zone 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's, 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**

Offer's Zone app will use "Parse.com". To store and retrieve data app will use Parse REST API. On Parse , First it creates a table according to requirement then parse will provide API(url) for each and every task such login, sign up, store and retrieve the data. Parse provide Access Key and Application id which will be used to verify the app on parse server.

When application request for any service like Login or Sign Up page., It should have add Access key and Application-id as header while requesting for service with POST, GET and more. When application hit the url , it will be returned data in JSON format.

**4.1.4 Model**

The Parse 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 Parse.com to store/retrieve data. The Parse platform provides a complete backend solution for mobile application. To integrate parse in app, we need Application ID and Client Key.

**4.2 API Connection-**

**4.2.1 API calls**

Here is an overview of the API calls:

|  |  |
| --- | --- |
| Name | Signup |
| Documentation link | https://parse.com/docs/rest/guide#users-signing-up |
| Usage | Used to create new user. This will store user specific information like User Id, Email Id, Mobile No. |

|  |  |
| --- | --- |
| Name | Login |
| Documentation link | https://parse.com/docs/rest/guide#users-logging-in |
| Usage | Used to allow user to log in to application. User can log in using user id/email id and password. |

|  |  |
| --- | --- |
| Name | Retrieve Offers |
| Documentation link | https://parse.com/docs/rest/guide#objects-retrieving-objects |
| Usage | Used to retrieve daily offers. Object Id can be used to retrieve Parse Object using parse query. Using get X method, we can get value out of Parse Object. |

|  |  |
| --- | --- |
| Name | Retrieve Offer's Details |
| Documentation link | https://parse.com/docs/rest/guide#objects-retrieving-objects |
| Usage | Used to retrieve Offer detail. Object Id can be used to retrieve Parse Object using parse query. Using get X method, we can get value out of Parse Object. |

**4.3 Techniques**

**4.3.1 OS**

Offers Zone Application will support iOS version 8.1 and Above.

**4.3.2 Device Support**

Offers Zone App support iPhone and iPad.

**4.3.3 Orientation**

Offers Zone application will support portrait mode.