APS

Google Android Mobile Application

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| 01/Jun/14 | 2.0 | Implementation Progress Documents | Mahbubur Rahman |

Table of Contents

[1 Introduction 4](#_Toc393201479)

[1.1 Purpose 4](#_Toc393201480)

[1.2 Scope 4](#_Toc393201481)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc393201482)

[1.4 References 4](#_Toc393201483)

[1.5 Overview 4](#_Toc393201484)

[2 System Overview 5](#_Toc393201485)

[3 Design Considerations 6](#_Toc393201486)

[3.1 Assumptions and Dependencies 6](#_Toc393201487)

[3.2 Constraints 6](#_Toc393201488)

[4 System Architecture 7](#_Toc393201489)

[5 Design Details 8](#_Toc393201490)

[5.1 Web Application 8](#_Toc393201491)

[5.1.1 Static Structure 8](#_Toc393201492)

[5.1.2 Behavior 9](#_Toc393201493)

[5.1.3 User Interface 11](#_Toc393201494)

[5.1.4 Data Structure 11](#_Toc393201495)

[5.1.5 Dependencies 11](#_Toc393201496)

[5.1.6 Components 11](#_Toc393201497)

[5.2 Mobile Application 15](#_Toc393201498)

[5.2.1 Reminder alarm design 15](#_Toc393201499)

[5.2.2 Static Structure 15](#_Toc393201500)

[5.2.3 Behavior 16](#_Toc393201501)

[5.2.4 User Interface 17](#_Toc393201502)

[5.2.5 Dependencies 19](#_Toc393201503)

[5.2.6 Components 19](#_Toc393201504)

# Introduction

This is the System Design Document for the Google Android Mobile Application project.

## Purpose

The Software Design Document captures the design constraints and assumptions as well as the detailed design of the subsystems and components of the application.

## Scope

This Design Document pertains to the Quick Fix project, with a specific focus on the features and capabilities being implemented in version 1.0.

## Definitions, Acronyms, and Abbreviations

See Glossary Document.

## References

Software Requirements Specification.

## Overview

This Design Document is divided into five major sections. Section 1 is an Introduction that provides information about the document itself. Section 2 is an overview of the application and its primary functionality. Section 3 identifies the assumptions and constraints followed during the design of the software. Section 4 documents the overall system architecture. Finally, section 5 provides the detailed design information for each subsystem and component in the current delivery.

# System Overview

Android is the first complete, open and free mobile platform. It is developed and supported by Google and this project uses a Google Android Mobile SDK 1.0 for testing an application. The software development kit contains the emulator and advanced debugging tools to run and test the applications.

The aim of our project is to develop a household application on the android which can be used in the real world. The project has two types of interfaces: Web and Mobile phone. The Web interface is used by the service seeker who has household problems and they use the web interface to post their problems through their respective accounts. The service providers must belong to a company or they themselves can be a company for getting registered to the web application. The mobile application is supposed to be installed on the mobile phone and this assumed to be with every service provider.

This project focuses on developing a web application that is developed using C# and ASP.NET where the service seeker and service providers can have their profiles to know about each other. This web application increases the security by allowing the service seeker to know the person who is going to fix the problem posted by him. Both services provider and seeker will have an option to choose the person according to the ratings and reviews that are posted previously.

The main part of the project resides on the mobile phone which has application that is developed using java and XML. The service provider uses the application for tracking the service seeker within a given Zip-code that is shown on a Google Map. For displaying the set of service seekers and their problems this project uses a Google Map API and XML for displaying the content on the map. The service provider is allowed to see the problems posted and he is allowed to click on the information for more benefits such as calling a service seeker, route to the destination, route from the source, add to favorites and view profile.

# Design Considerations

This section contains all of the assumptions, dependencies, and constraints that were considered during the design of each subsystem and component.

## Assumptions and Dependencies

The Mobile Application is dependent on the Web Application because the mobile application cannot work without the data from the web application and vice versa.

The default settings on the mobile application will reflect on the Google maps.

## Constraints

Huge data shall not be stored on the mobile phone

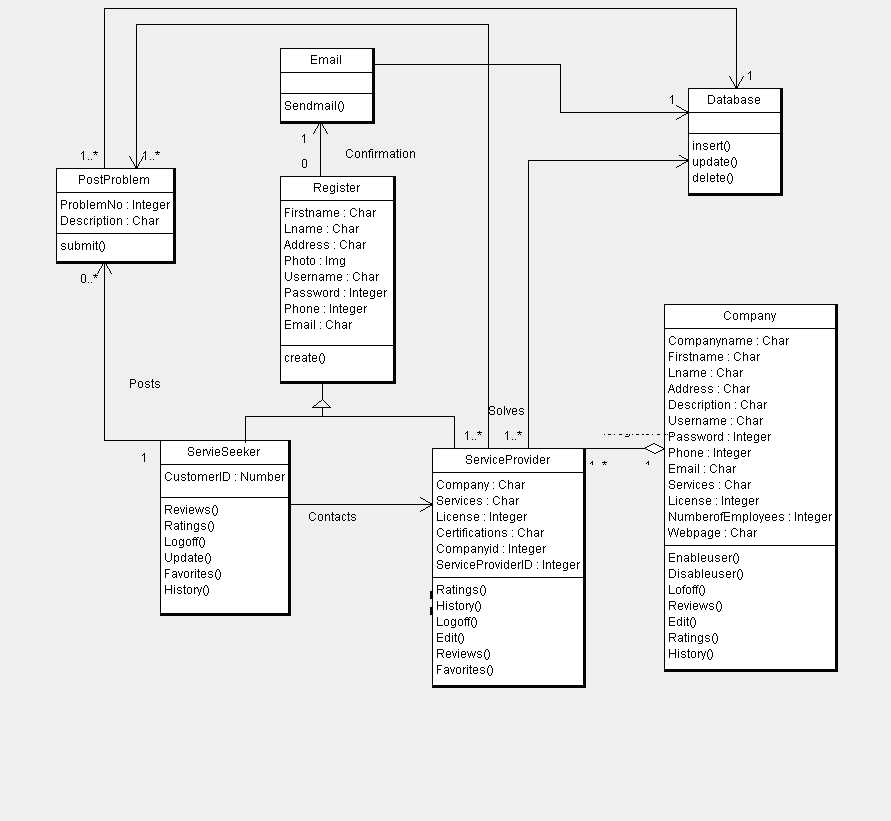
User of the mobile should have internet access

# System Architecture

# Design Details

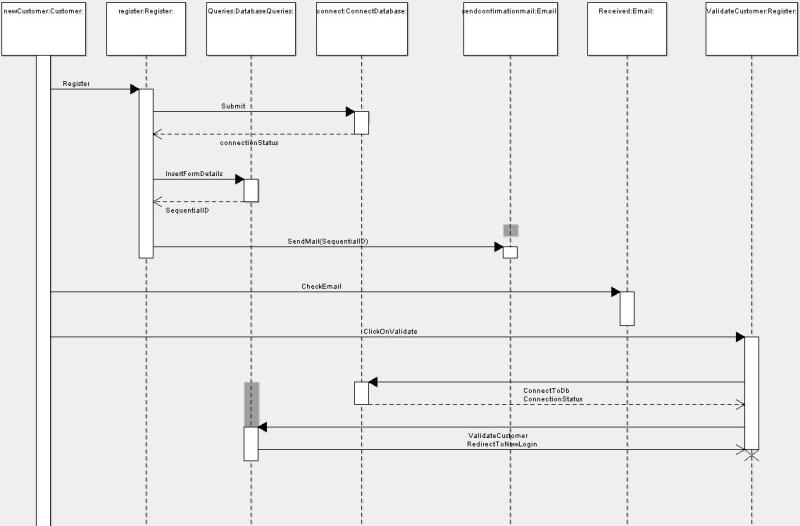
## Web Application

### Static Structure

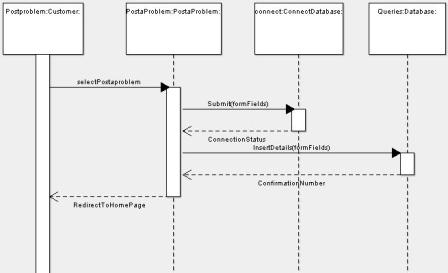


### Behavior

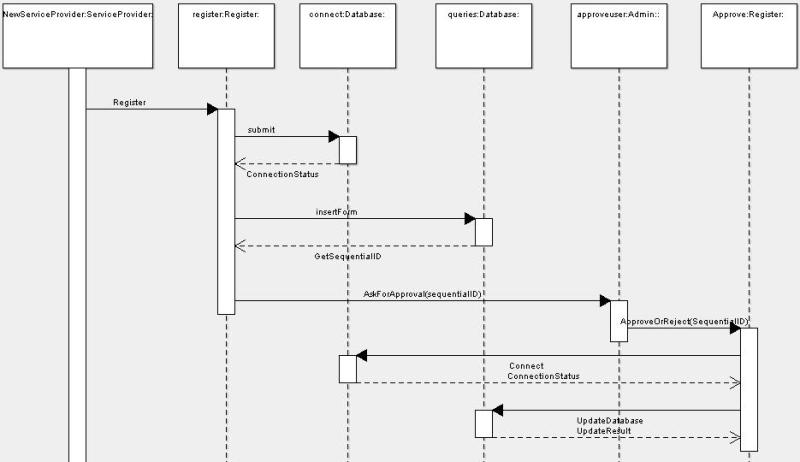
1. Customer Registration



1. Customer Posting a problem



1. Service Provider registration



### User Interface



### Data Structure

None.

### Dependencies

None.

### Components

#### Registration

This class take care of the registration process and saves all the user’s information.

##### Attributes

None.

##### Methods

* public create()

This class displays all the registration fields so that a first time user can register under his/her corresponding section.

##### Exceptions

Exceptions can be thrown under following conditions

1. When the data access fails.

#### ServiceSeeker

##### Methods

* public Update(object sender, EventArgs e)

With the help of this method, user can update or modify his personal information.

* public Ratings()

This method helps a customer to rate a particular service provider

* public Reviews()

This method provides the customer, a chance to give feedback regarding a particular service provider

* public Logoff()

This method ends the session of the user. It automatically redirects the user to corresponding login page.

* public Favorites()

This method is used for adding the particular Service Provider to a specific customer’s favorites list.

* public History()

This method is responsible for storing all the previously posted problems by that specific Customer.

##### Exceptions

Exceptions can be thrown under following conditions

1. When the data access fails.

#### Service Provider

##### Methods

* public Update()

With the help of this method, service provider can update or modify his profile.

* public Ratings()

This method helps a service provider to rate a particular customer

* public History()

This method is responsible for storing all the previously solved problems by that specific service provider.

* public Reviews()

This method provides the Service Provider, a chance to give feedback regarding a particular Customer.

* public Favorites()

This method is used for adding the particular Customer to a specific Service Provider’s favorites list.

* public Logoff()

This method ends the session of the user. It automatically redirects the user to corresponding login page.

##### Exceptions

Exceptions can be thrown under following conditions

1. When the data access fails.

#### Company

This class is responsible for managing all the service providers registered under a particular company.

##### Methods

* public EnableUser()

This method is responsible for approving the newly registered service providers after evaluating his/her credentials.

* public DisableUser()

This method takes care of revoking all the previleges of a particular user, if anything goes wrong.

* public Edit()

With the help of this method, admin can update or modify his company profile.

* public Reviews()

This method provides the service provider, a chance to give feedback regarding a particular user of the company

* public History()

This method is responsible for storing all the previously solved problems by all the service providers.

* public Ratings()

This method helps the admin to rate a particular service provider.

* public Logoff()

This method ends the session of the user. It automatically redirects the user to corresponding login page.

##### Exceptions

Exceptions can be thrown under following conditions

1. When the data access fails.

#### PostProblem

This class takes care of the problems being posted by the customers.

##### Methods

* public Submit()

This method saves all the problems posted by the customers in the database.

##### Exceptions

Exceptions can be thrown under following conditions

1. When the data access fails.

## Mobile Application

### Reminder alarm design

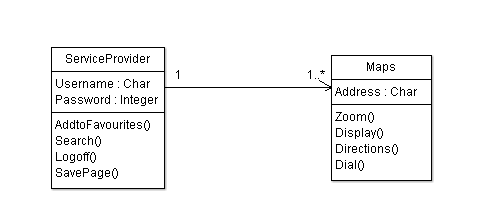
The alarm for the user to input certain data are integrated together to send a single alarm.

The default settings on the mobile application will reflect on the Google maps.

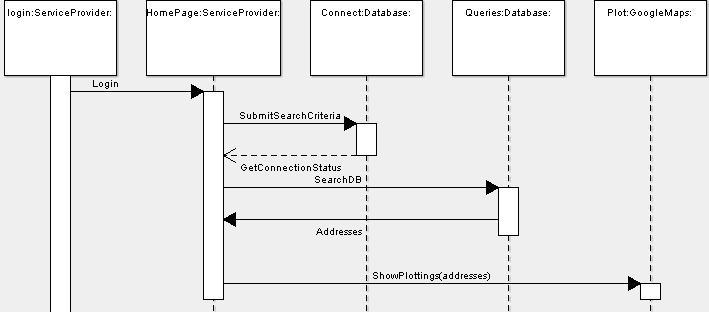
### Peak flow

Peak flow is measured using the peak flow device and twice a day. So, for single day diary information, two types of peak flow will be included. We called these as **AM PEAK FLOW** and **PM PEAK FLOW**.

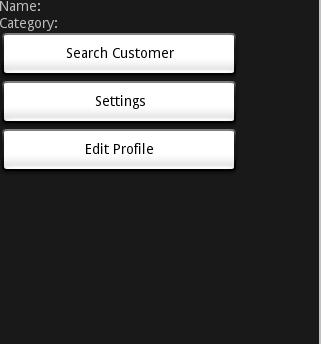
### Static Structure

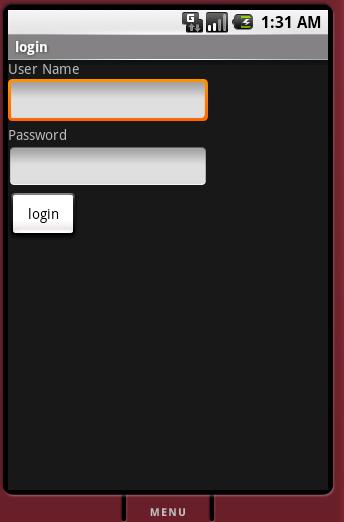


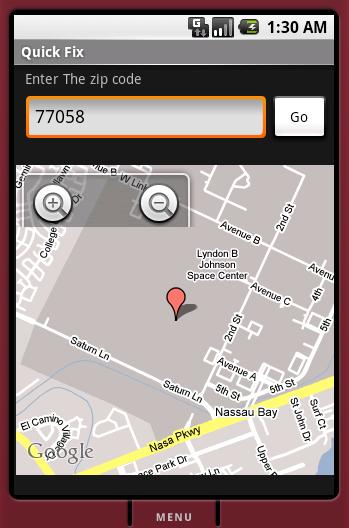
### Behavior



### User Interface







### Dependencies

None.

### Components

#### Maps

Used for plotting all the address in a given Zip code.

##### Attributes

None.

##### Methods

* public Zoom(layoutparams.WRAP\_CONTENT)

This method is used for wrapping the content according to the page size. The user can zoom into the specific address or zoom out of it. This method is called by the User Interface class when the user clicks the zoom button

* public Dial()

This method is used by the service provider to contact the customer.

* public Directions()

This method is used to navigate between addresses. Once this method is called, it returns with a list of directions to the destination address

##### Exceptions

Exceptions can be thrown under following conditions

1. When the data access fails.

#### ServiceProvider

This class is used for Validating and forwarding the user to the home page. Once the used logs in, he can search for the service seekers within a specific range with the help of various settings.

##### Attributes

##### Methods

* public AddtoFavourites()

This method is used by the service provider to add his/her favorite users to a quick list. Once this method is called, it returns with a list of customers for a specific service provider.

* public Search(int zipcode, int radius)

This is a vital method used to search all the customers in a given zip code and radius. The return value of this method is a map plotted with bunch of customer addresses.

* public SavePage()

This method is used to save the last search made by the service provider.

##### Exceptions

Exceptions can be thrown under following conditions

1. When the data access fails.