PROJECT DEPLOYMENT :

Project Deployment :

Github Site for Traveler’s Eye project is <https://github.com/umkc1>

The project URL is <http://jsfiddle.net/konerusindhu27/u4g28/104/embedded/result/>

**Project (Pre)Proposal :**

**Introduction:**

Being international students we all understand the intricacies involved in travel. We all feel the stress of keeping track of a lot of information about the destination and also the trip itself, especially when it comes to travelling alone. Traveler’s eye/Guide is aimed to make a traveler’s life easy by providing relevant useful information whenever the user needs it, at a simple button click or a finger touch.

**Project Goals and Objectives:**

Booking flight or train tickets is just the begging, you still has to decide the place of stay and then the proximity of your place of stay to your places of interests. For example if you are on a business trip then you need to select a place of stay which is near to your office, if you are on a holiday then the place of stay must not be far away from all the places you want to visit. Then comes the mode of travel, you need to select weather you want to hire a private vehicle or use the public transport, you will need information like the traffic conditions and the weather conditions to make that choice.

Traveler’s Eye/Guide will be able to provide information regarding convenient places of stay with respect to the user’s area of interest or place of interest. Also a user on a business trip or a personal trip would be able to tell the app about the kind of trip he/she is on by simply selecting the correct option. Based on the information provided by the user the app will be able to assist the user in a better way.

Once the destination has been selected the app will be able to provide the traffic and weather related data so that the user can make an informed decision on the modes of transport or the time of travel. Even if it’s a short distance it’s not a very good idea to try to walk to your destination just before a thunder storm or try to take a cab home during a heavy traffic jam. The app also provides a lot details about the destination like the currency used, language spoken, history of the location, crime rate, which areas are a must visit and which places to avoid.

Also hunger is a very important factor especially while traveling. You might be very hungry but there might not be any restaurants in sight but there could be more than 10 restaurants in the street parallel to you. Again in this case the Traveler’s Eye/Guide comes to the rescue and by just one click on the “Find Food” button the app will automatically locate your position and then based on that search for the nearby restaurants and mark them on a google map for you.

The app will also load the local police station number, hospital number and any such emergency numbers as soon as the user selects the destination. So even in emergency all the user has to do is press once button. Additionally the app also allows some emergency contacts to be defined by the user.

Now comes the fun part of the Traveler’s Eye/Guide experience. The app lets the user to take pictures, listen to music and then it also lets the user to connect with social networking sites like Facebook or twitter.

**Related work:**

There are a lot of different applications available which offer some of the similar features but they are mainly localized to only one specific location or city. Tourist Guide Paris, Yokohama Tourist Guide (Local), Tourist Guide for Greece, Manchester Tourist Guide etc. There are a few other applications which are not localized, but they are mostly based on booking trips. Such applications manly focus on selling tickets rather than providing the user with good usable relevant information in a convenient and hassle-free way.

**Final Proposal:**



**TRAVELER’S EYE**

**Team:** Phoenix

**Project Title:**

Traveler’s Eye

**Introduction:**

Being international students we all understand the intricacies involved in travel. The stress of keeping track of all the information about the destination and the travel, especially when it comes to travelling alone, takes away from the fun one can have while travelling. Traveler’s eye is aimed to make a traveler’s life easy by providing relevant useful information whenever the user needs it, at a simple button click or the touch of a finger.

**Project Goals and Objectives:**

Traveler’s Eye will be able to provide information regarding convenient places of stay with respect to the user’s area or place of interest. Also the user would be able to select if he/she is on a business trip or a leisure trip. Based on the information provided by the user the app will be able to assist the user in a better way. The app will also provide information like the traffic conditions and the weather conditions.

Once the destination has been selected the app will be able to provide the traffic and weather related data so that the user can make an informed decision on the modes of transport or the time of travel. The app also provides a lot of details about the destination like the currency used, language spoken, history of the location, crime rate, which areas are a must visit and which places to avoid.

The overall objective is to provide the user a very safe and comfortable experience while he is travelling.

**Related work:**

There are a lot of different applications available which offer some of the similar features but they are mainly localized to only one specific location or city. Tourist Guide Paris, Yokohama Tourist Guide (Local), Tourist Guide for Greece, Manchester Tourist Guide etc. There are a few other applications which are not localized, but they are mostly based on booking trips.

None of the other applications offer the complete package like the Traveler’s Eye. Almost all the single city based tourist guide applications only concentrate on simply showing the user all the popular places in that particular city, a user is not allowed to choose any other city so if your trip covers multiple cities then you will have to install new app for each new city.

And all the tourist guide applications which are dynamic and let the user select different destinations are more concerned about booking tickets, hotels, luxury trips, cruises etc. Also they are only concerned about leisure trips and the most exotic locations and destinations. Such applications manly focus on selling tickets rather than providing the user with good usable relevant information in a convenient and hassle-free way. A normal day to day traveler or a low budget traveler has no relevant information in these applications.

In Traveler’s Eye we try to be as dynamic and flexible as possible. The user can travel to any number of cities using just one application. Also the Traveler’s eye provides more than just information about the tourist spots, it provides information about weather, traffic conditions, navigation etc. These services provide useful information to all the users no matter what kind of trip they are on. Also the security features in the Traveler’s Eye application are very unique. So far we did not come across any application which provides the automatic location update service.

**Proposed System**

**Requirement Specification**

**Functional requirements:**

1. Emergency services: (priority high)

* This function facilitates user to inform the local emergency services like local police, hospital by just giving a click on the corresponding button, in an emergency case.
* It also handles the task of updating his location to the emergency contacts.
* It also allows user to update/notify user’s location to the contacts defined by the user at regular intervals of time. Thus allowing him to stay in touch with group people as he intends, throughout his journey.
* It has to function in such a way that it helps user in the needful times.

1. Location details: (priority high)

* People on travel either on business trip or leisure trip will attain a situation at some point of time to know their current location. System must be able to retrieve the user’s location and display it for him using geo-location services.
* User will also have the facility of navigation. He can give some destiny and attain the route map to his desired location from his location.

1. Navigation: (priority high)

* Navigation is one of the important service provided to a user.
* It allows user to get directions from place to another place. It serves as a guiding tool that rests in the hands of user. As user enters his destination, system makes use of the google maps and displays direction from the source to the destination.
* If there are multiple paths from source to destination then those paths are to be clearly displayed so that the choice could be made by the user. Eventually making user feel comfortable to move between places.

1. Exploring the city: (priority high)

* As user enters his destiny system provides the facility of exploring more about their place of stay like placed to be visited, its usual temporal conditions, history of the place, interesting facts of the place, kind of currency that is been used, languages spoken, crime rate of the locality etc.
* Individual travelling to the place for the first time may not have the ease of travelling which thereby pulls him away from enjoying his journey. In traveler’s eye system allows him to get the info with a single click.
* System takes his mode of travelling and guide him accordingly throughout his journey.

1. Traffic and weather related information: (priority high)

* Traveler plans his plan and starts his journal accordingly. But traffic and weather of the place keeps varying from time to time.
* It is of good purpose if he knows the situations prevailing in the city as he enters the city. Thus Traffic and Weather are other two prominent issues that are to be considered from the traveler’s perspective.
* After reaching the place, say the person is here to attend some meeting then he will turn keen to know the traffic conditions of the place, so that he could reach the meeting on time.
* Weather allows user to decide the means of transportation to be used. System allows him to know the temperature of the place before he reaches the location.
* System retrieves the weather information using weather API basing on the geo co-ordinates of the user’s current location.

1. Finding food in the proximity: (priority medium)

* System allows user to avail a special feature that is providing the exact location of some source of food say restaurants. By just clicking on the “find food” button user can get the location of the restaurants in the proximity.
* Once user clicks the option system must find the nearest location of the restaurants and display it to the user upon google maps. For this purpose system makes use of the API that locates restaurants using geo co-ordinates of the user.
* User can also get the help of navigating to the selected restaurant.
* Besides providing the locality of the restaurant, system includes a little detail of the kind of food that is available in the restaurant.

1. Social Networking: (priority low)

* System also provides user to update himself user the social networking websites like Facebook, Twitter, Google+ etc.
* For enabling this feature API’s of the networking sites are been used.

**Non-Functional Requirements**:

1. As Emergency issue is been addressed in traveler’s eye, system has to put some threshold on the delay time in notifying the emergency contacts in the database.
2. While intimating the emergency contacts the system must notify with the exact location of the user making use of his current geo-location co-ordinates.
3. Weather of the place must have to be retrieved exactly. System must update the weather information repeatedly, so that he can get himself prepared to face the forthcoming weather conditions.
4. Traffic of the location must also be refreshed at regular intervals of time say every 10 minutes as traffic keeps changing with respect to time.
5. Considering the finding food service, the distance of finding must be set with some value as the bound value say 5-6miles.
6. The number of places has also to be set up with some value may be 10 restaurants in the closest proximity.

**Technological & architectural requirements:**

**Server Operating System:**

Windows operating system.

**Presentation layer:**

HTML5, Java Script, JQuery, CSS

**Web Framework services & languages:**

.net framework, Restful and SOAP web services, C#, IIS server

**Mobile phone requirements:**

Any mobile device that supports jQuery and HTML5 can be used for this project.

**Data base:**

ADO.net

**Business Requirements & Work Flow Analysis:**

In this project, we use Agile Process Management for project development.

**Agile Development process:**

Agile Development process is gaining a great acceptance in software mainstream development community. It can be used for the Development, testing and the management aspects of the project. Agile Process is favorably accepted because it improves the communication among the team members, and can help in achieving the desired goals. Agile Process follows some set of principles which are useful to the project development.

ScrumDo is one of the Agile Software Development model

1. ScrumDo is different from the traditional sequential Approach (Water fall model)
2. It mainly focus on the holistic and flexible development of the product
3. ScrumDo is useful to perform:

* Backlog management
* Iteration Planning
* Story
* Epic Story

**Work Flow Analysis:**

1. Initially a proper planning is made to the project approach.
2. Then the work flow is divided in to Iteration’s.
3. For each Iteration’s stories are created.
4. A brief meeting of approximately 15minutes will be performed daily in order to know about the status of the work done and what we have to do.
5. Continuous testing and Continuous Integration is done in order to rectify the negative outcomes.
6. Assessment of the project progress is done and new requirements will be added by updating the product backlog.
7. Finally, the desired project is delivered.

**Over all System Architecture:**



**Page Layout:**



**System Specification:**

**Primary Services:**

* The important service that the application provides is that it traces the current location of the user and shows the navigation using maps. Using this feature the user can search the nearby restaurants, hotels and make his trip comfortable.
* Once the user uploads his destination he can view the traffic and weather information at the destination. This information helps him to prepare well for his tour. He can choose the right mode of transport for his travel in the destination once he is familiar with these conditions.
* The application provides the emergency contact service. Depending on the destination the application loads the local police station number, hospital number and any emergency number that will be helpful to user at the time of emergency. This is the unique feature of this application.

**EXISTING SERVICES:**

Some of the services in this application uses some predefined API’s.

->Google Maps API:

These are used to give the view of the street or city or any place on the globe. This view can be a map or a satellite image. This api is the heart of the application.

<https://developers.google.com/maps/>

->Yahoo Weather API:

The weather service provided here mainly depends on yahoo weather services .The yahoo weather service provides some API which helps us to get the current weather information of the place we need.

<http://developer.yahoo.com/weather/>

->Google Traffic Layer API :

Google provides traffic layer API .This helps to show the traffic at the requested place dynamically. The application uses this to implement the traffic service.

<https://developers.google.com/maps/documentation/javascript/layers#TrafficLayer>

->Facebook API:

These helps us to use some of the social plugins and Facebook login functionalities into our application.

<https://developers.facebook.com/docs/reference/apis/>

->Twitter API:

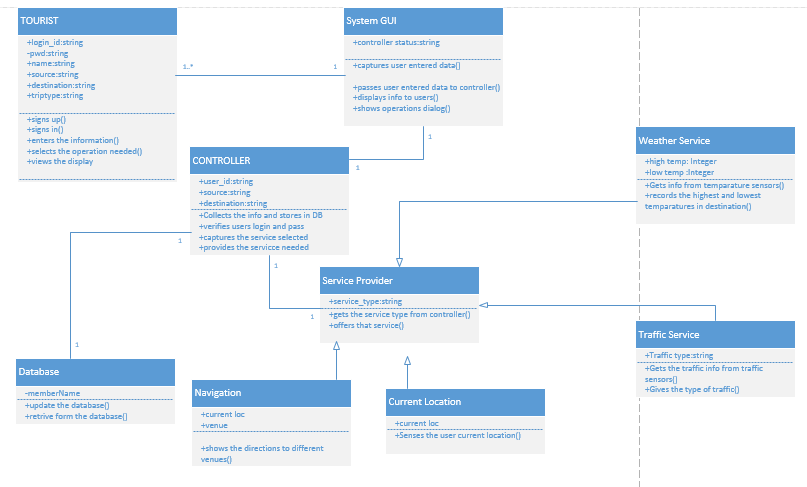
By using these API’s we can login into twitter and can tweet or follow and can provide some publicity to our application.

<https://dev.twitter.com/>

**New Services:**

* Suggests the nearby places of interest to the user according to his tastes and convenience.
* Automatic updating of emergency contacts.
* Automatic location updating to the user specific contacts.
* User can connect with social networking sites like Facebook and twitter.
* The app allows the user to take pictures and also to listen to the music.

**CLASS DIAGRAM FOR OVERALL APPLICATION:**



The class diagram describes the static structure of the application. The class diagram consists of classes, attributes and operations.

**Classes:**

Tourist: This class refers to the user of the application. The user should first sign up with application and them login into it and give his details. And then can use the services of the application.

System GUI: This is the interface between the user and the application. It captures the user’s data and passes it on to the controller .And displays the useful information to user as requested by the controller.

Controller: This is the heart of the whole application. It analyses everything. It updates and verifies database. Contacts the service provider and then brings the service needed into play.

Database: The database is managed here. The storage of information is done here.

Service Provider: This is used to provide services to the application. Each service is a sub class to this super class.

1. Weather service: This gives the weather conditions at the given destination. This gives the highest and lowest temperatures at the given position.
2. Traffic service: This gives the traffic updates at the place required by the user.
3. Current Location: The current location of the user is identified.
4. Navigation: This shows the navigation from to user’s current position to the place he wants to move.

**Relationship-Multiplicity:**

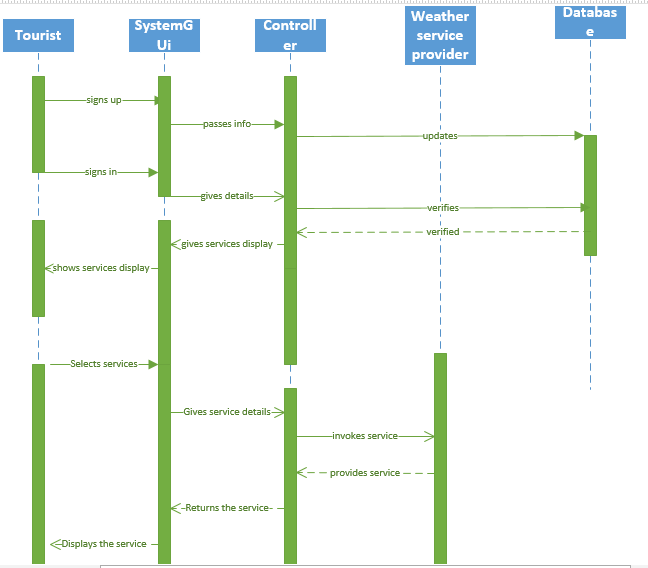
Tourist-System GUI: The System GUI is same and it can be used by multiple user’s .Hence the multiplicity is one or many-one. The relationship is association relationship.

System GUI-Controller: There is a purely one to one relationship between these two. As only single piece of them exists. They have an association relationship.

Controller-Service Provider: These both also have association relationship. And they have one to one multiplicity.

Controller-Database: They do have one to one multiplicity and association relation.

Service provider-weather service, Service provider-traffic service, Service provider-navigation, service provider-current location: These have parent child relationship. The Service provider is the parent class and the individual services are the child classes.

**SEQUENCE DIAGRAM FOR WEATHER SERVICE:**

This is an interaction diagram which shows the interaction of one class with another class with respect to time.

**Steps:**

1. The user first signs up using System GUI.
2. The GUI informs the controller.
3. The controller stores the information in the database.
4. The user now logs in again by taking the help of System GUI.
5. The System GUI informs the controller.
6. The controller verifies the database.
7. The database sends the authentication.
8. The controller asks the System GUI to display the services.
9. The System GUI displays the services to the user.
10. The user selects one of the services.
11. This is informed to the controller with the help of the System GUI.
12. That service is provided by the service provider by contacting its sub classes.
13. The service provider provides the service to the controller.
14. The controller passes it to System GUI.
15. Finally, the required service is displayed to the user.

**SERVICE SPECIFICATION:**

After adding the google maps to the application using google API, The application should detect the current location of the tourist. And then it should be able to show the direction to the place that the tourist requests to move to.

To display the weather services we use yahoo weather API and the application should take the input of the destination and show the current weather over there. .For each destination it should request the yahoo services and bring the required output.

Google provides traffic layer we use this and obtain the traffic information that is just required by the user. We should get the traffic updates just for the required area.

The app will also load the local police station number, hospital number and any such emergency numbers as soon as the user selects the destination. So even in emergency all the user has to do is press one button. Additionally the app also allows some emergency contacts to be defined by the user. The app also has an automatic location update feature which send’s the user’s location information to the user defined emergency contacts at regular intervals of time. The same feature can also automatically send the user’s location to the emergency services in case of an emergency.

The user can connect with social networking sites like Facebook and twitter by just giving his login id and password in our application.

The user can take pictures and post to his friends and he can listen to music of this interest by specifying it while signing up.

Priorities:

|  |  |  |
| --- | --- | --- |
| **SERVICE** | **IMPORTANCE** | **DIFFICULTY** |
| Tracking the current location of user | High | High |
| Navigation | High | Moderate |
| Emergency Contacts | High | Moderate |
| Traffic | High | Moderate |
| Weather | High | Moderate |
| Facebook | Moderate | Low |
| Twitter | Moderate | Low |
| Find Food | Moderate | Moderate |
| Taking pictures and posting | Low | High |
| Music | Low | Moderate |

**First Increment:**

Services Designed in First Increment:

In the first increment we created the user interface.

Here along with user interface two of the services are implemented.

In the first service we obtain the current location of the user and show the restaurants nearby his location.

In the second service we show the direction from source to destination.

Temporary GUI.

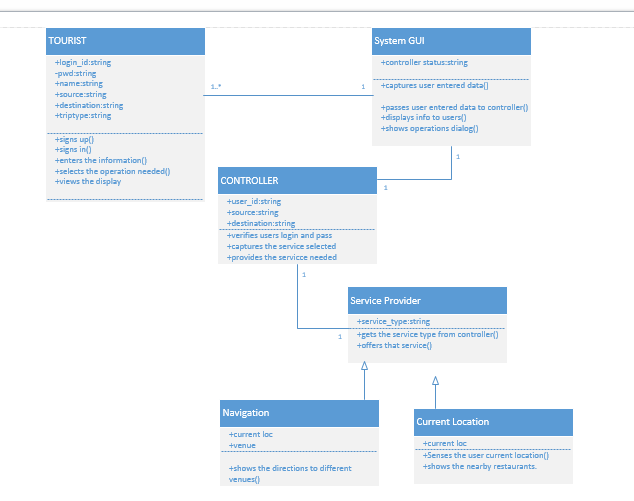
Testing GUI.

API’S:

Food Service uses Google Maps API.

Navigation Service uses Google Maps API.

CLASS DIAGRAM FOR FIRST INCREMENT:



**Classes:**

Tourist: This class refers to the user of the application. The user should first sign up with application and them login into it and give his details. And then can use the services of the application.

SystemGUI: This is the interface between the user and the application. It captures the users data and passes it on to the controller .And displays the useful information to user as requested by the controller.

Controller: This is the heart of the whole application. It analyses everything. It updates and verifies database. Contacts the service provider and the brings the service needed into play.

Service Provider: This is used to provide services to the application. Each service is asub class to this super class.

1)Current Location: The current location of the user is identified and the restaurants nearby are located.

2)Navigation: This shows the navigation from one place to other.

**Relationship-Multiplicity:**

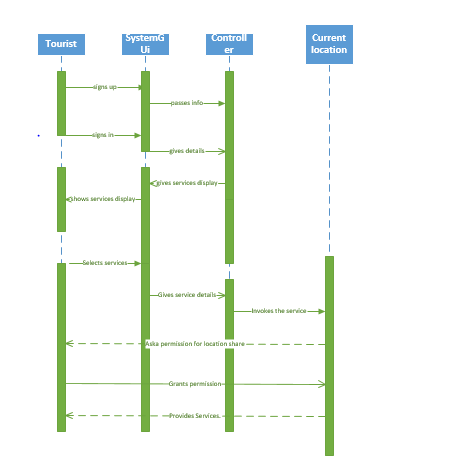
Tourist-SystemGUI: The SystemGUI is same and it can be used by multiple user’s .Hence the multiplicity is one or many-one. The relationship is association relationship.

SystemGUI-Controller: There is a purely one to one relationship between these two. As only single piece of them exists. They have an association relationship.

Controller-Service Provider: These both also have association relationship. And they have one to one multiplicity.

Service provider-navigation, service provider-current location: These have parent child relationship. The Service provider is the parent class and the individual services are the child classes.

**SEQUENCE DIAGRAM FOR Finding Restaurant:**



**Steps:**

The user first signs up using SystemGUI.

The GUI informs the controller.

The controller stores the information in the database.

The user now logs in again by taking the help of SystemGUI.

The SystemGUI informs the controller.

The controller asks the SytemGUI to display the services.

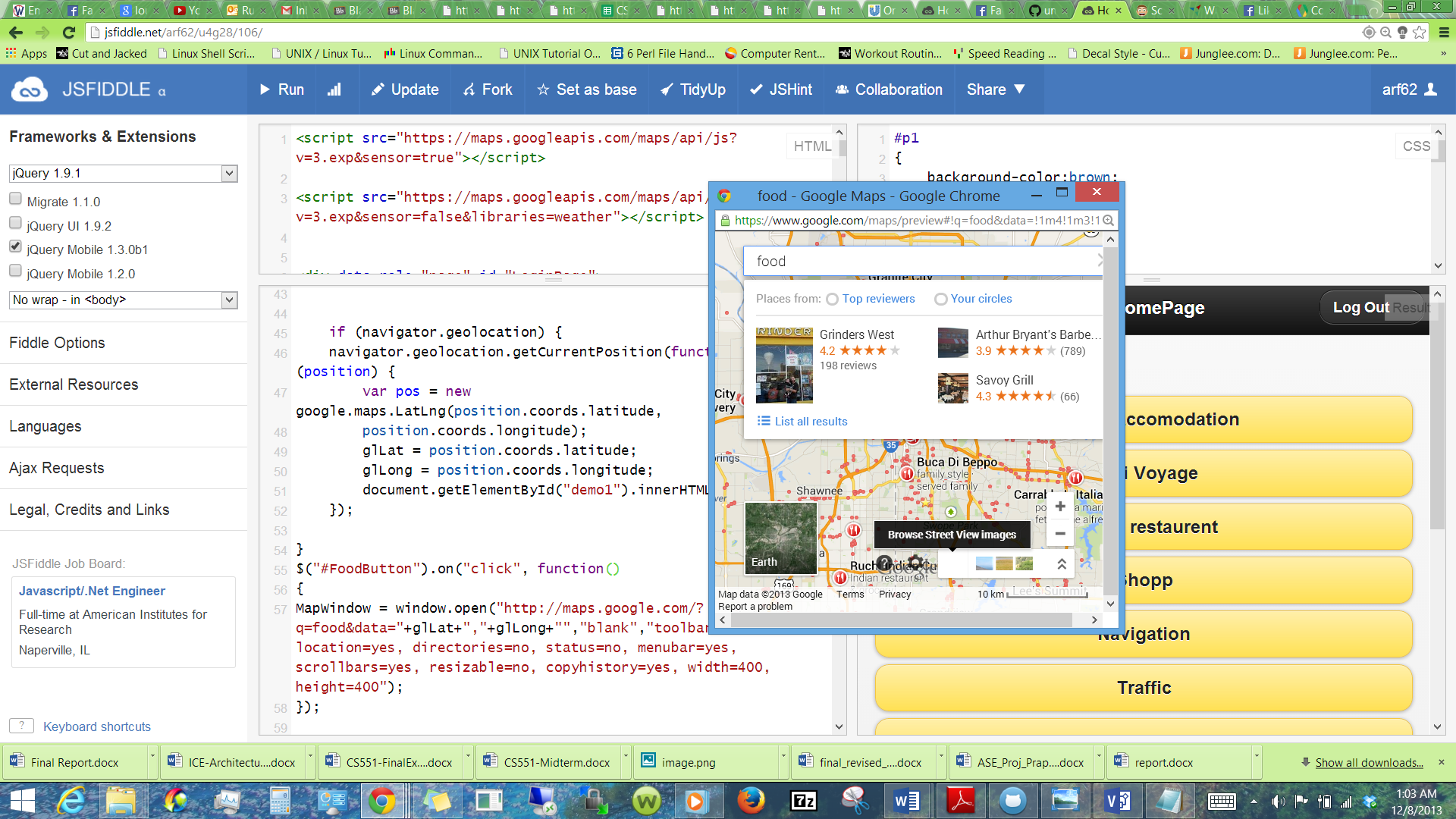
The SystemGUI displays the services to the user.

The user selects one of the services.

This is informed to the controller with the help of the SystemGUI.

That service is provided by the service provider by contacting its sub classes.

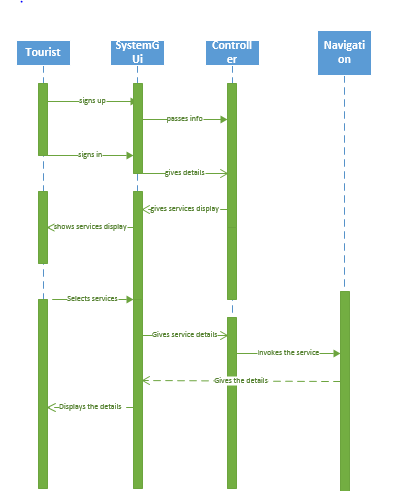
The service provider provides the service to the user using sub class current location.



1) Find Restaurant Using Current Location Of the User :

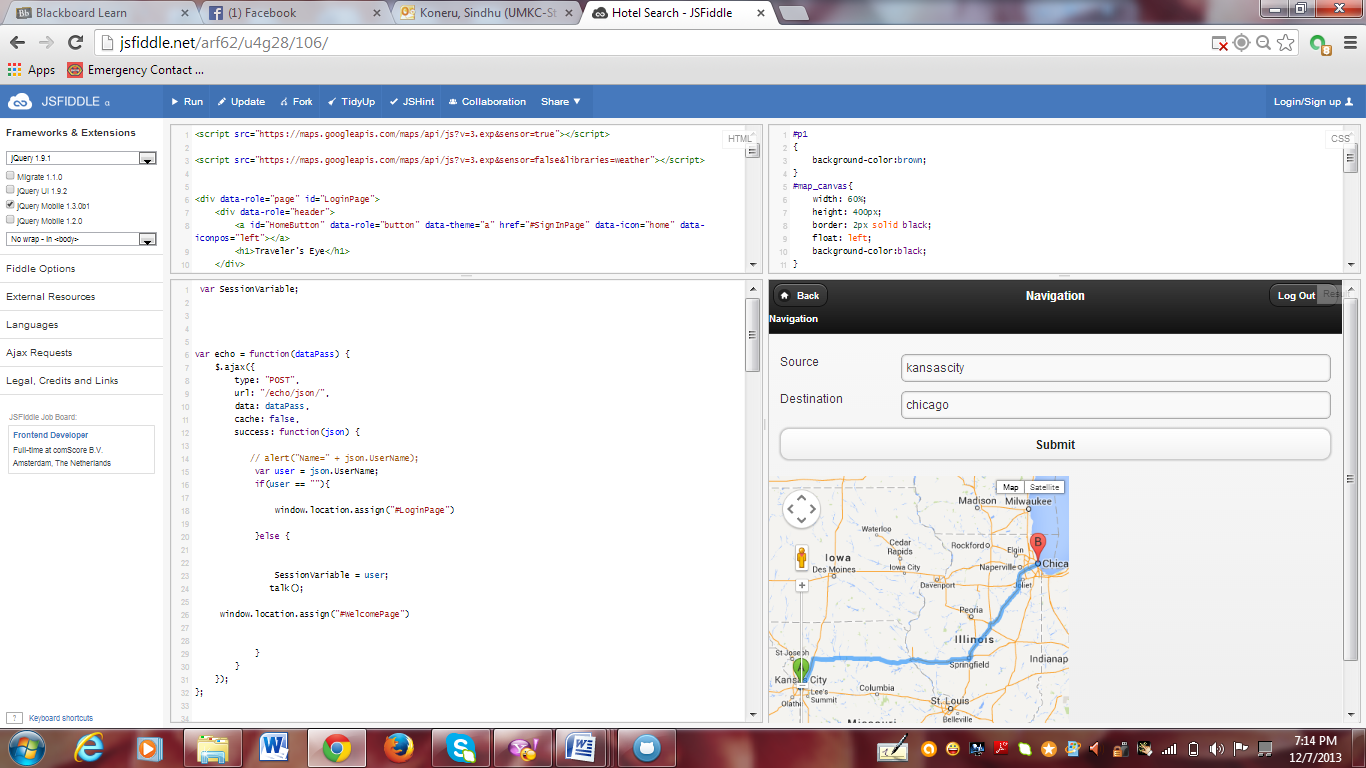
Food is the basic need of a person wherever he is. When the traveler is in the new place it would be difficult for him to know where the restaurants are, even if they are very nearby also. To avoid this situation Traveler’s Eye application provides this service. Just by the click of a button the application takes the current location of the user and displays him all the restaurants nearby him. This display is done using google maps.

**SEQUENCE DIAGRAM FOR NAVIGATION:**

****

2)Navigation :

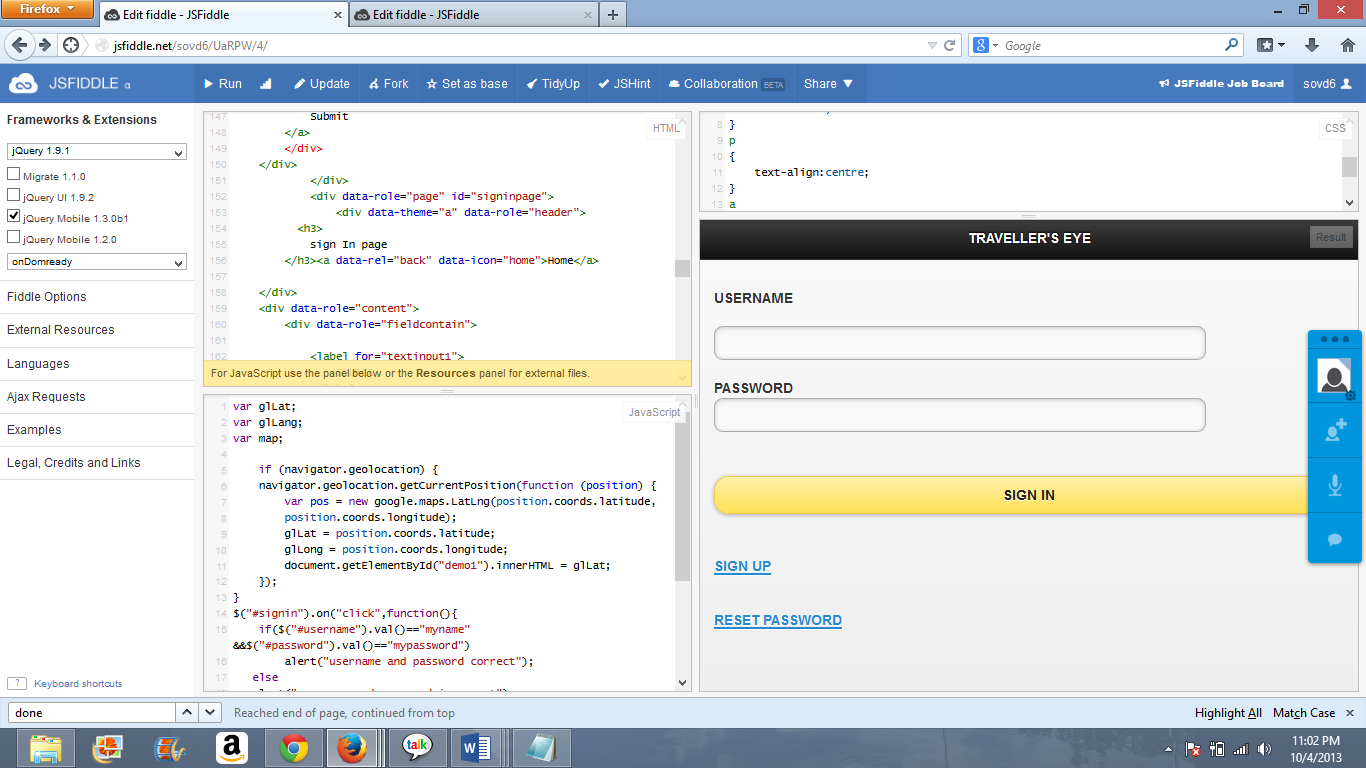
Travelling is all about moving from one place to other. It would always be better for the user to know the route when he is moving. This service enables him to find the route between any two places. And loads the directions dynamically with the help of google maps. It shows the directions all over the world.



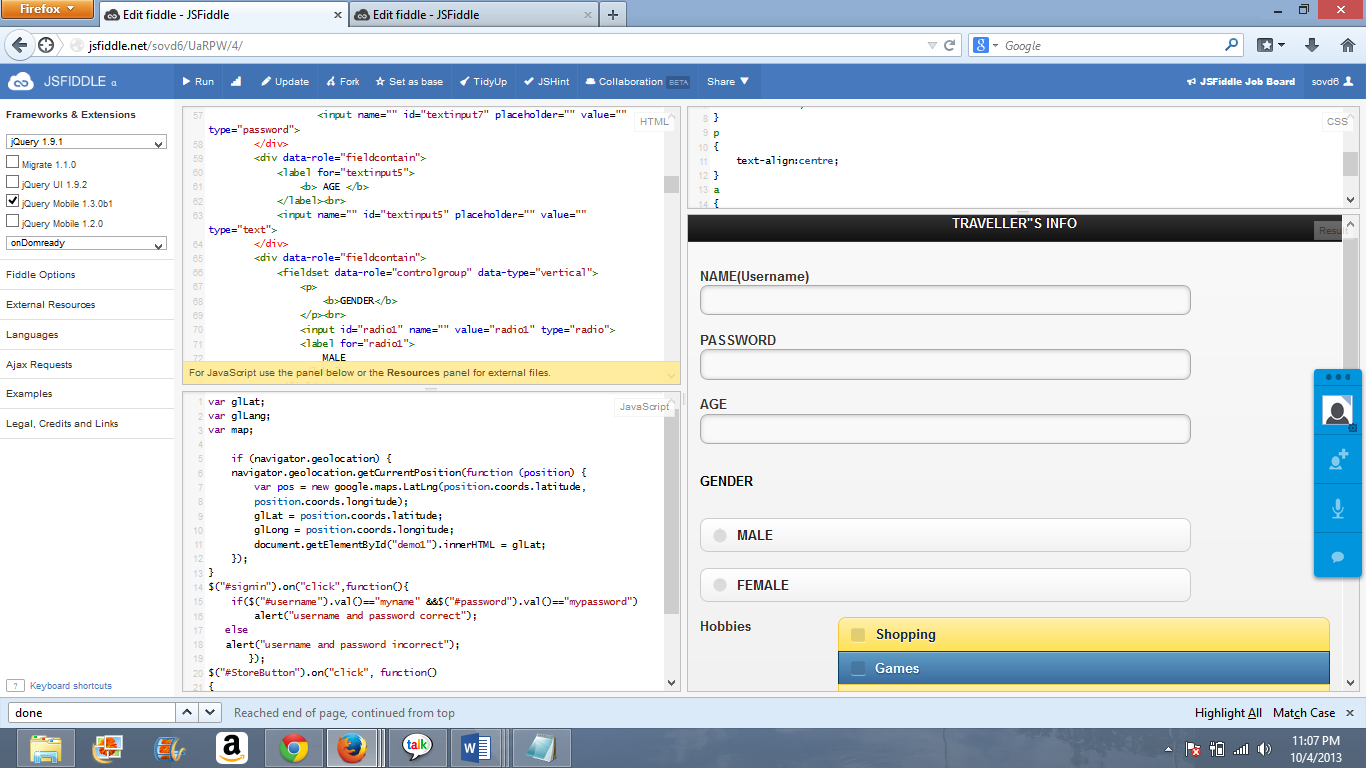
All that the user needs to enter here is the souce and destination. Then he will get the directions displayes in the map automatically.

3)Temporary Mobile GUI :

A temporary mobile GUI with sign in and sign up and reset password functionalities is created to accommodate the above two services is created.



Sign Up



**Testing the GUI (Graphical User Interface):**

When the application is executed on the first click it is displaying the proper screen fit output.

Each and every field in the page is aligned properly i.e; text boxes, radio buttons.

By clicking on each input text field the pointer of the mouse is changing as the Cursor.

When the “Tab” button is clicked, the cursor is shifting its control to the next input text box.

The application is allowing the user to select a single radio button at a single instance.

It is allowing multiple selection of check boxes.

When the user enters the password the password should not be displayed and the password is not displayed as characters.

Github link :

<https://github.com/umkc1/inc01>

ScrunDO :

[**https://www.scrumdo.com/projects/project/project-150/iteration/78367**](https://www.scrumdo.com/projects/project/project-150/iteration/78367)

**Second Increment:**

Services Designed in Second Increment:

1)Weather service.

2)Traffic service.

3)Validating Jquery mobile GUI

4)Part of database.

5)Text to Speech.

API’S and Web Service:

Google traffic layer

Google Weather Layer

cdyne weather service

Validation plugins: <http://ajax.aspnetcdn.com/ajax/jquery.validate/1.11.0/additional-methods.js>

<http://ajax.aspnetcdn.com/ajax/jquery.validate/1.11.0/jquery.validate.min.js>

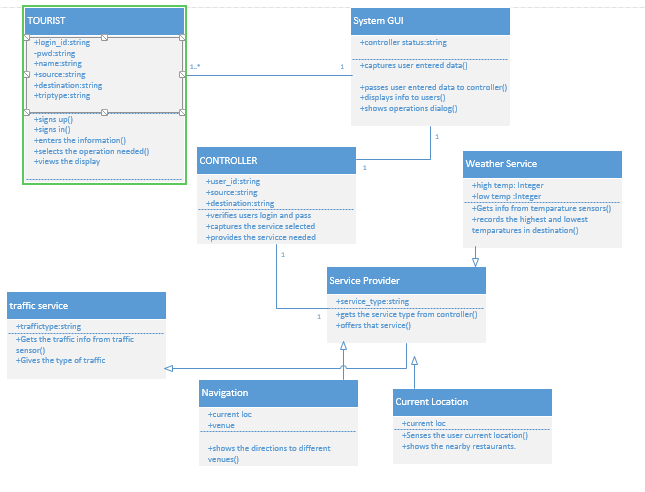
Google Text to Speech

http://weston.ruter.net/2009/12/12/google-tts/

The Unofficial Google Text-To-Speech API

http://techcrunch.com/2009/12/14/the-unofficial-google-text-to-speech-api/

CLASS DIAGRAM:



The class diagram describes the static structure of the application. The class diagram consists of classes, attributes and operations.

**Classes:**

Tourist: This class refers to the user of the application. The user should first sign up with application and them login into it and give his details. And then can use the services of the application.

System GUI: This is the interface between the user and the application. It captures the user’s data and passes it on to the controller .And displays the useful information to user as requested by the controller.

Controller: This is the heart of the whole application. It analyses everything. It updates and verifies database. Contacts the service provider and brings the service needed into play.

Service Provider: This is used to provide services to the application. Each service is a sub class to this super class.

1) Weather service: This gives the weather conditions at the given destination.This gives the highest and lowest temperatures at the given position.

2) Traffic service: This gives the traffic updates at the place required by the user.

3) Current Location: The current location of the user is identified.

4) Navigation: This shows the navigation from to user’s current position to the place he wants to move.

5) Text to speech: Greets the user as soon as his credential are accepted and he is allowed to log in.

6)Database: Design and set up database and tables in side for the application.

**Relationship-Multiplicity:**

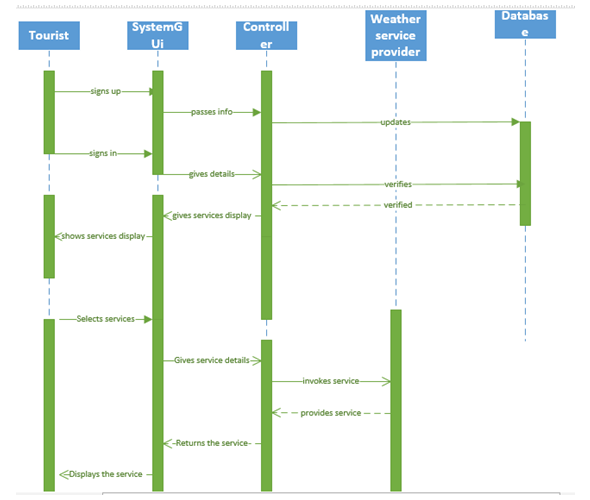
Tourist-SystemGUI: The SystemGUI is same and it can be used by multiple user’s .Hence the multiplicity is one or many-one. The relationship is association relationship.

SystemGUI-Controller: There is a purely one to one relationship between these two. As only single piece of them exists.They have an aasociation relationship.

Controller-Service Provider: These both also have association relaitonsip.And they have one to one multiplicity.

Service provider-weather service ,Service provider-traffic service ,Service provider-navigation, service provider-current location: These have parent child relationship. The Service provider is the parent class and the individual services are the child classes.

**SEQUENCE DIAGRAM FOR WEATHER SERVICE:**

****

This is an interaction diagram which shows the interaction of one class with another class with respect to time.

**Steps:**

The user first signs up using SystemGUI.

The GUI informs the controller.

The controller stores the information in the database.

The user now logs in again by taking the help of SystemGUI.

The SystemGUI informs the controller.

The controller verifies the database.

The database sends the authentication.

The controller asks the SytemGUI to display the services.

The SystemGUI displays the services to the user.

The user selects one of the services.

This is informed to the controller with the help of the SystemGUI.

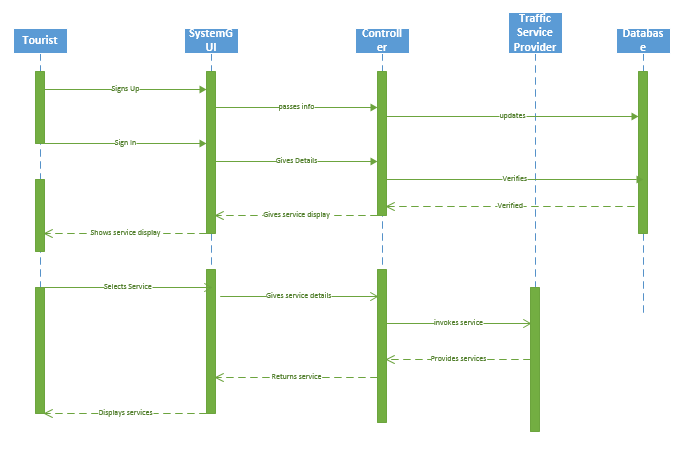
That service is provided by the service provider by contacting its sub classes.

The service provider provides the service to the controller.

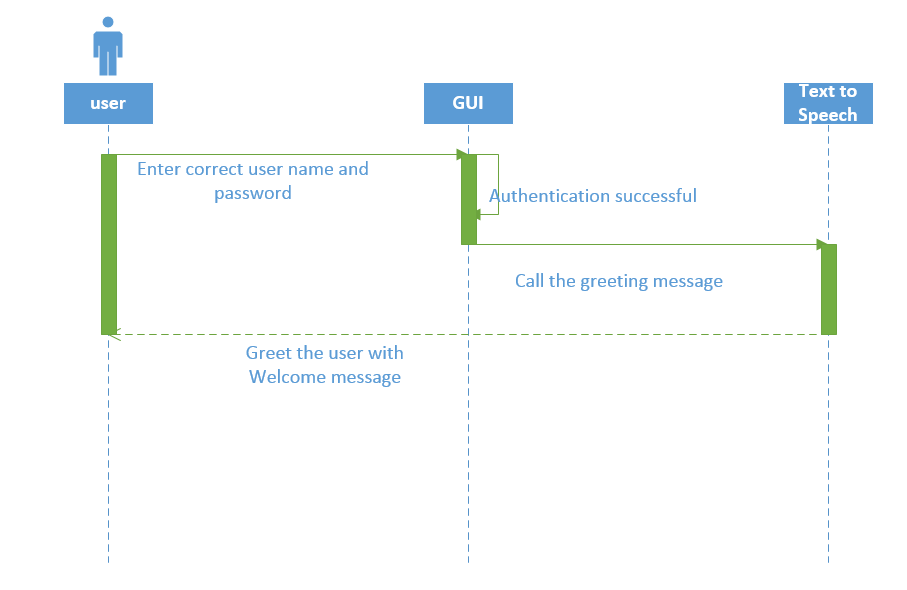
The controller passes it to SystemGUI.

Finnally,the required service is displayed to the user.

**SEQUENCE DIAGRAM FOR TRAFFIC SERVICE:**

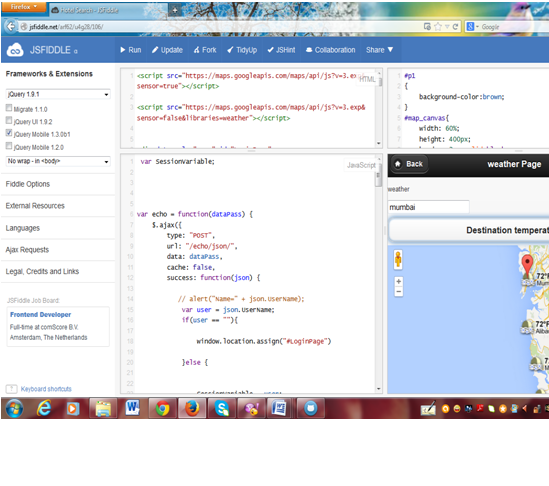
****

**Sequence Diagram for Text to speech:**

****

**WEATHER SERVICE:**

Our application provides with weather service to the user. So that he/she can plan things well. Like using this information he can choose the modes of travel example if there is a thunderstorm it is always better to take a cab. The user can know the weather of not only his place he can know the weather conditions all over the world. If we give the name of the destination we get the temperature.



The traveler’s eye website made use of a weather service which displays temperature by giving the zip code. For that cdyne weather service is used.



**Traffic service:**

This service enables user to have a look on traffic of a region, as traveler’s eye assured him to predict the traffic of destiny before reaching there.

In order to facilitate this traffic layer’s service from google maps are been utilized.

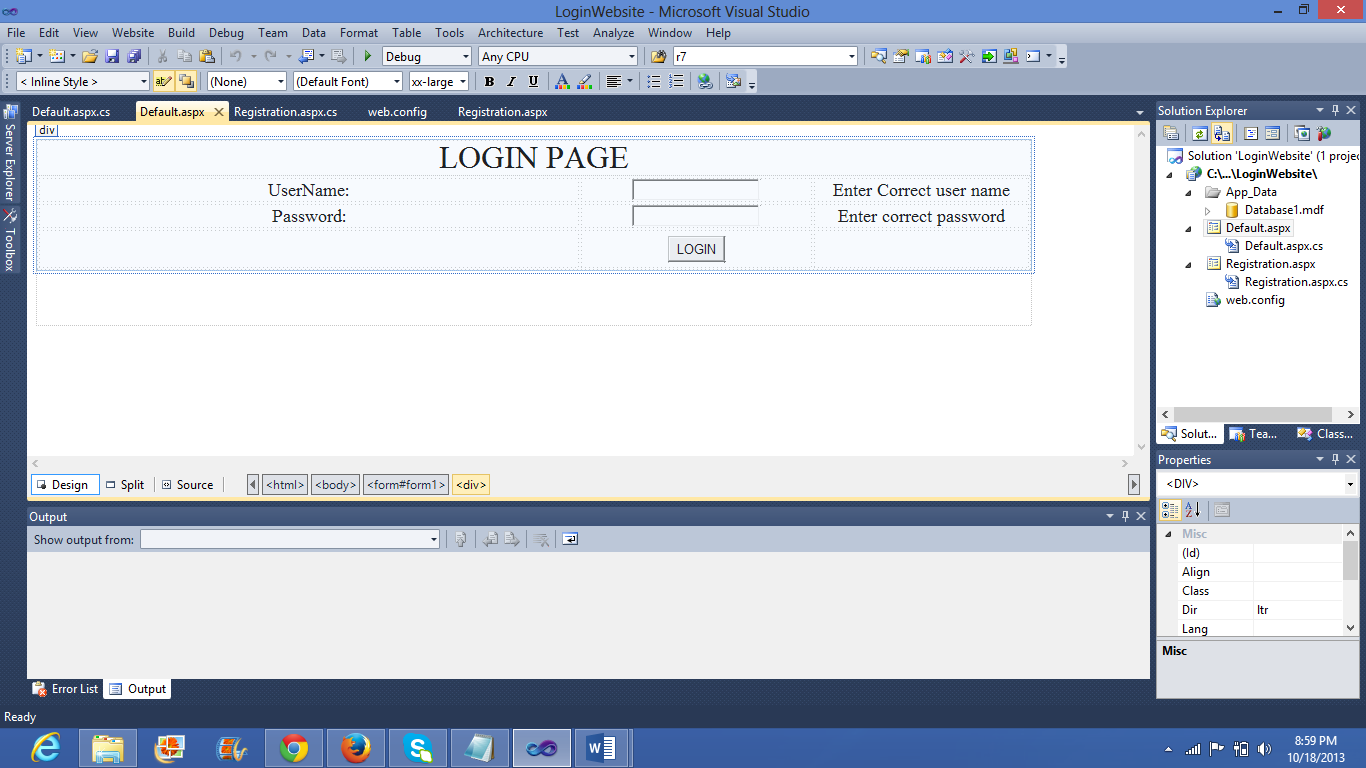
Initial step involving creating an object for google maps which gets rendered on the output screen with traffic intensity plotted over it.

Place on google can be plotted using the geocode or latitude and longitude of the location. But traveler’s takes in the name of location and plots it on the map by retrieving the geocode of the location.

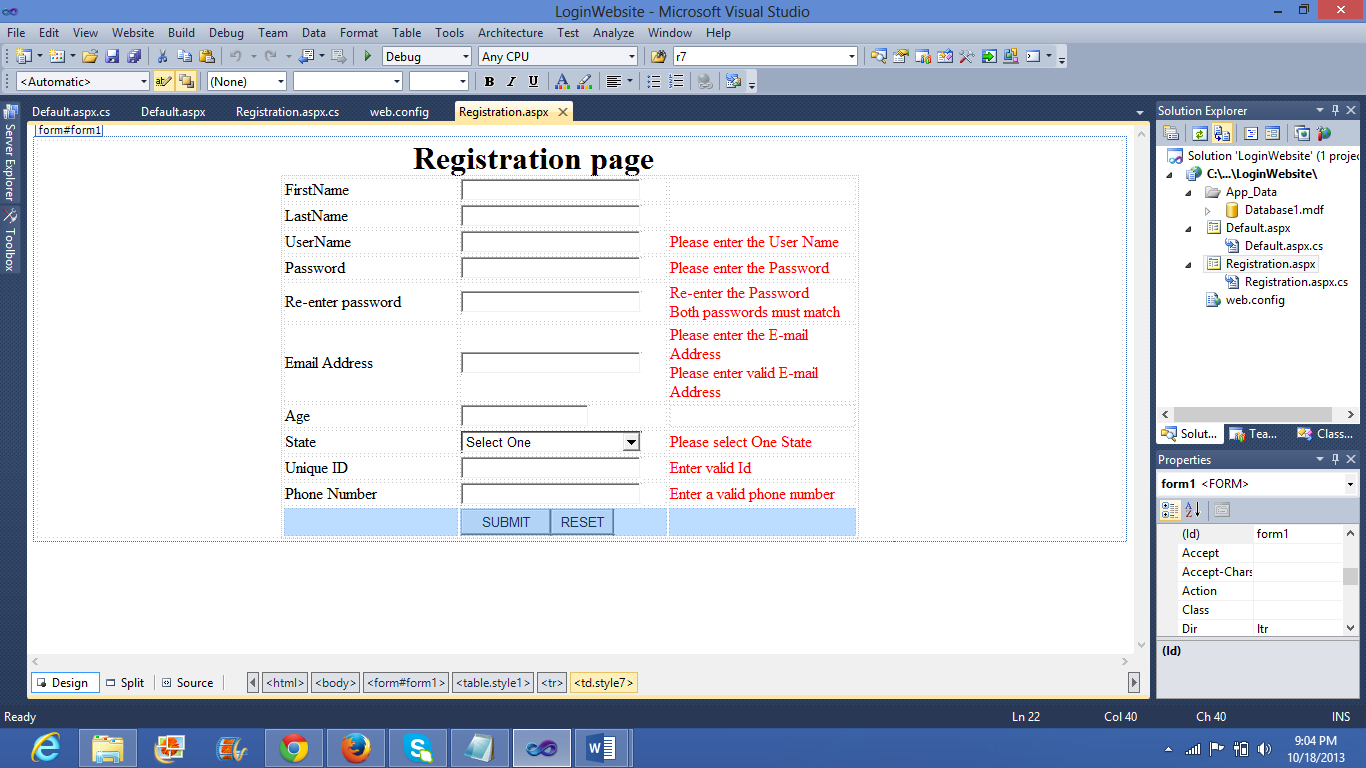
Once geocode of location is found, next step is to create an object for the traffic layer. Now traffic layer is overlaid on the map object with clear distinction of the location with a marker.

**Mobile GUI Validations :**

**WebSite Registration :**

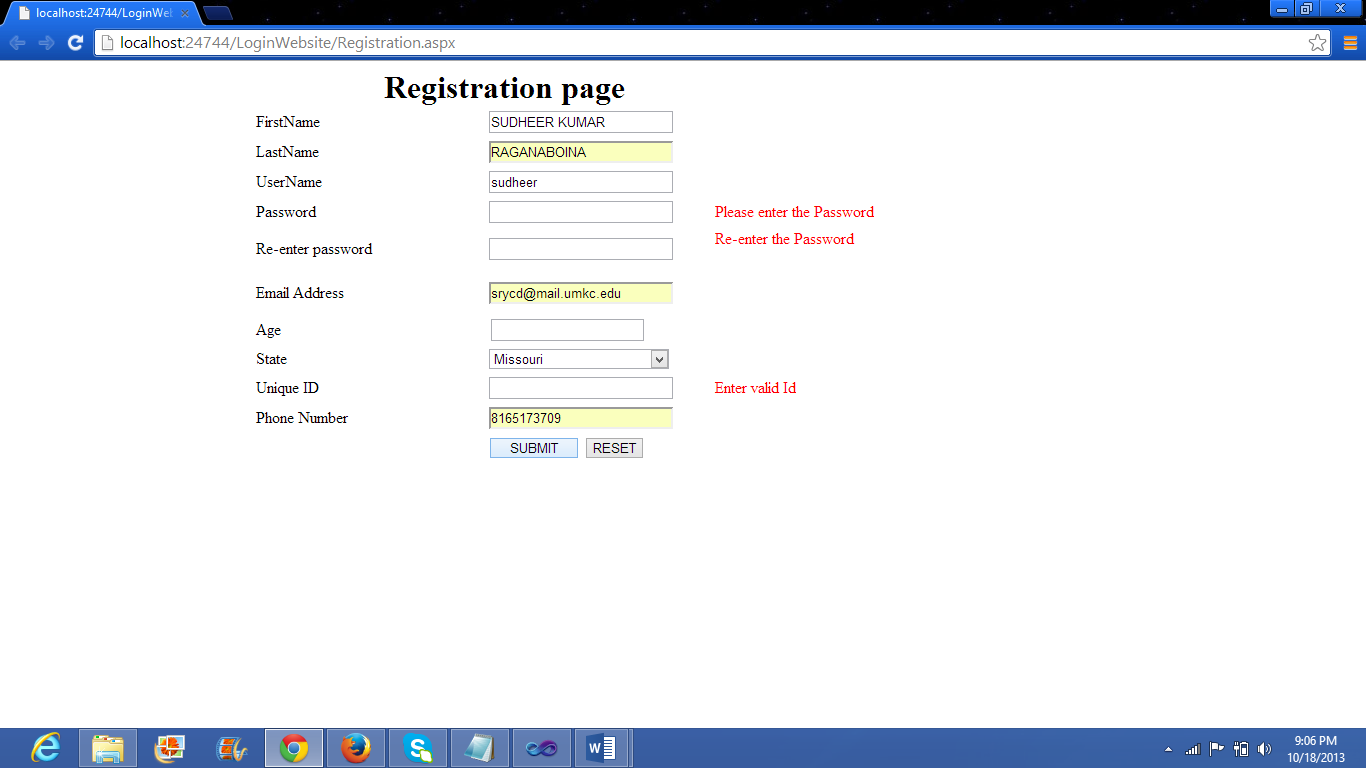


**Creating and validating Registration page:**

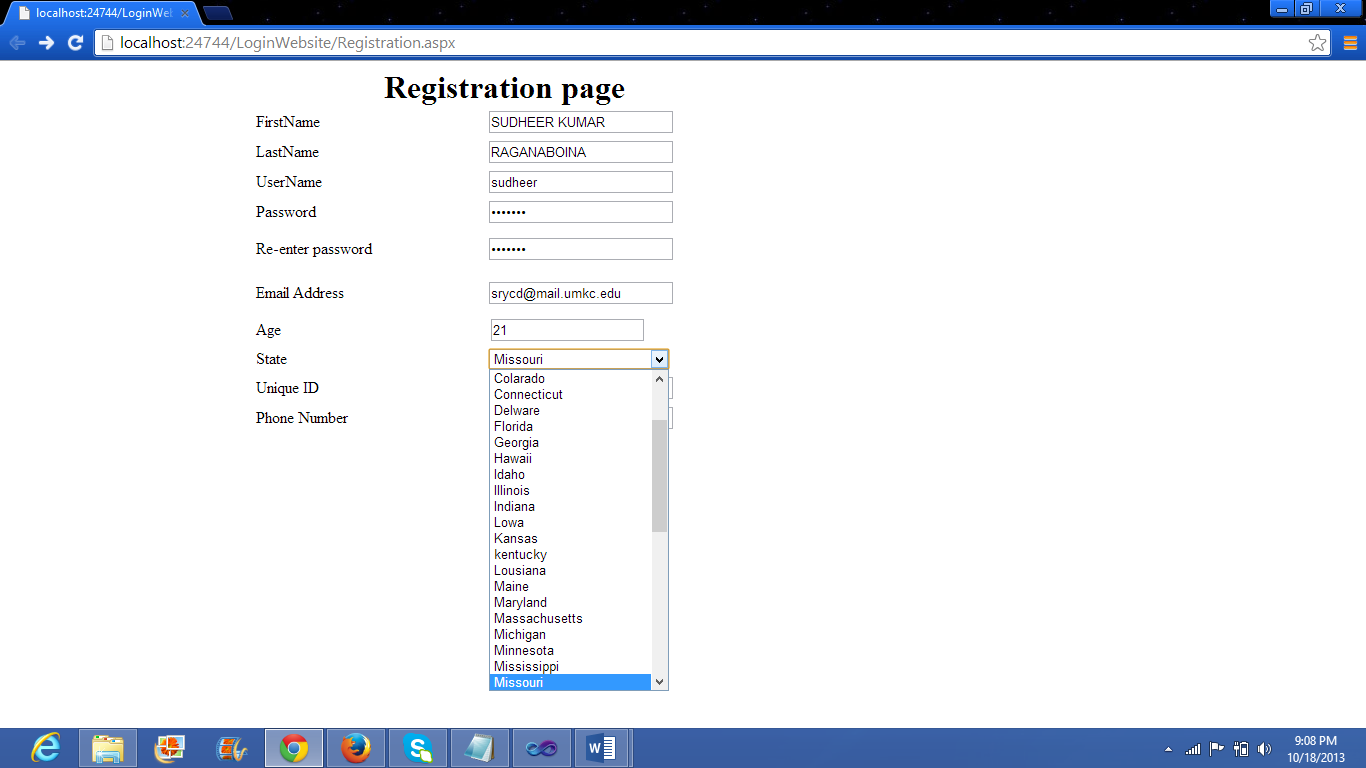


Validating and executing the registration page in the **local host**:

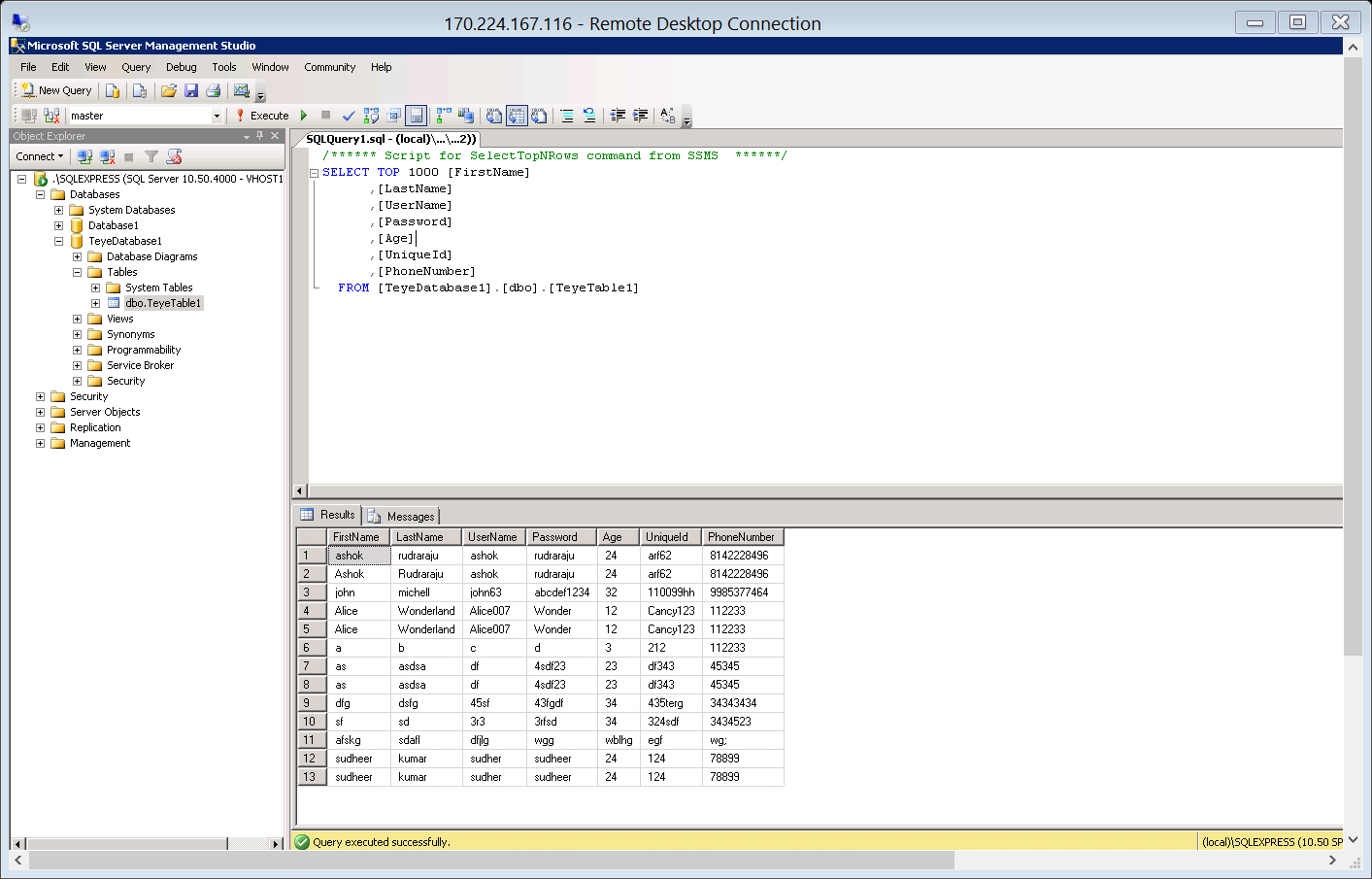
http://localhost:24744/LoginWebsite/Registration.aspx





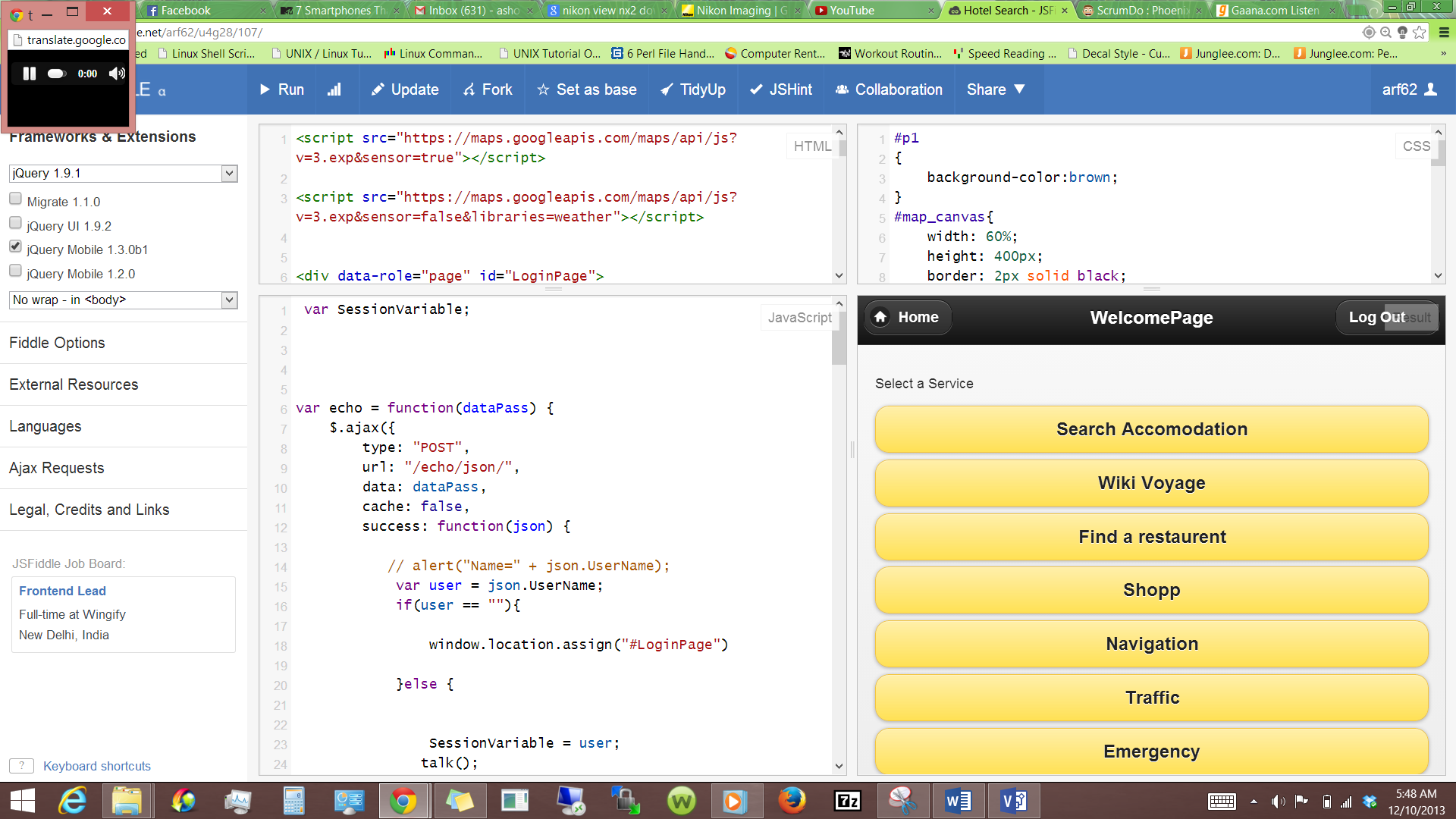


**Database:**



**Text to speech:**

When the user enters the correct user Id and password the authentication is successful. Then the condition is satisfied and the test to speech functionality is invoked. A small browser window is opened up and the google translate function is executed. After the message has been delivered successfully the new popup window closes automatically with the help of a javascript function.



Github Link :

<https://github.com/umkc1/increment2>

SrumDO Link :

[https://www.scrumdo.com/projects/project/project-150/iteration/78368](https://bluprd0113.outlook.com/owa/redir.aspx?C=CtgPW8roXU6HIkxb3nWDDnDsOPcsx9AIt4g1J2ePWemuclILBoPYl8iuW2XWEC-H8y8sHtS0x5Q.&URL=https%3a%2f%2fwww.scrumdo.com%2fprojects%2fproject%2fproject-150%2fiteration%2f78368)

**Third Increment :**

Services Designed in Third Increment:

1)Emergency Contacts

2)Updated Traffic Service

3)Uploading photos into the application

4)Retrieving data from database

5)Stock Updates for business users

6)Wiki Voyager Travel Guide Information

7)Web service for inserting data to the database and retrieving the data

8)Find Accommodations for the users.

API’S Used :

Google Traffic Layer.

Take Photos

[http://www.xarg.org/project/jquery-webcam-plugin/](https://bluprd0113.outlook.com/owa/redir.aspx?C=CtgPW8roXU6HIkxb3nWDDnDsOPcsx9AIt4g1J2ePWemuclILBoPYl8iuW2XWEC-H8y8sHtS0x5Q.&URL=http%3a%2f%2fwww.xarg.org%2fproject%2fjquery-webcam-plugin%2f)

[https://hacks.mozilla.org/2012/04/taking-pictures-with-the-camera-api-part-of-webapi/](https://bluprd0113.outlook.com/owa/redir.aspx?C=CtgPW8roXU6HIkxb3nWDDnDsOPcsx9AIt4g1J2ePWemuclILBoPYl8iuW2XWEC-H8y8sHtS0x5Q.&URL=https%3a%2f%2fhacks.mozilla.org%2f2012%2f04%2ftaking-pictures-with-the-camera-api-part-of-webapi%2f)

[http://stackoverflow.com/questions/10456140/how-take-a-photo-using-jquery-and-show-in-canvas](https://bluprd0113.outlook.com/owa/redir.aspx?C=CtgPW8roXU6HIkxb3nWDDnDsOPcsx9AIt4g1J2ePWemuclILBoPYl8iuW2XWEC-H8y8sHtS0x5Q.&URL=http%3a%2f%2fstackoverflow.com%2fquestions%2f10456140%2fhow-take-a-photo-using-jquery-and-show-in-canvas)

Save Pictures

[http://stackoverflow.com/questions/7951326/save-image-to-users-disk-using-javascript](https://bluprd0113.outlook.com/owa/redir.aspx?C=CtgPW8roXU6HIkxb3nWDDnDsOPcsx9AIt4g1J2ePWemuclILBoPYl8iuW2XWEC-H8y8sHtS0x5Q.&URL=http%3a%2f%2fstackoverflow.com%2fquestions%2f7951326%2fsave-image-to-users-disk-using-javascript)

stocks :

[http://markets.financialcontent.com/stocks](https://bluprd0113.outlook.com/owa/redir.aspx?C=CtgPW8roXU6HIkxb3nWDDnDsOPcsx9AIt4g1J2ePWemuclILBoPYl8iuW2XWEC-H8y8sHtS0x5Q.&URL=http%3a%2f%2fmarkets.financialcontent.com%2fstocks)

wii Voyage :

[http://en.wikivoyage.org/wiki/Main\_Page](https://bluprd0113.outlook.com/owa/redir.aspx?C=CtgPW8roXU6HIkxb3nWDDnDsOPcsx9AIt4g1J2ePWemuclILBoPYl8iuW2XWEC-H8y8sHtS0x5Q.&URL=http%3a%2f%2fen.wikivoyage.org%2fwiki%2fMain_Page)

Google Text to Speech

http://weston.ruter.net/2009/12/12/google-tts/

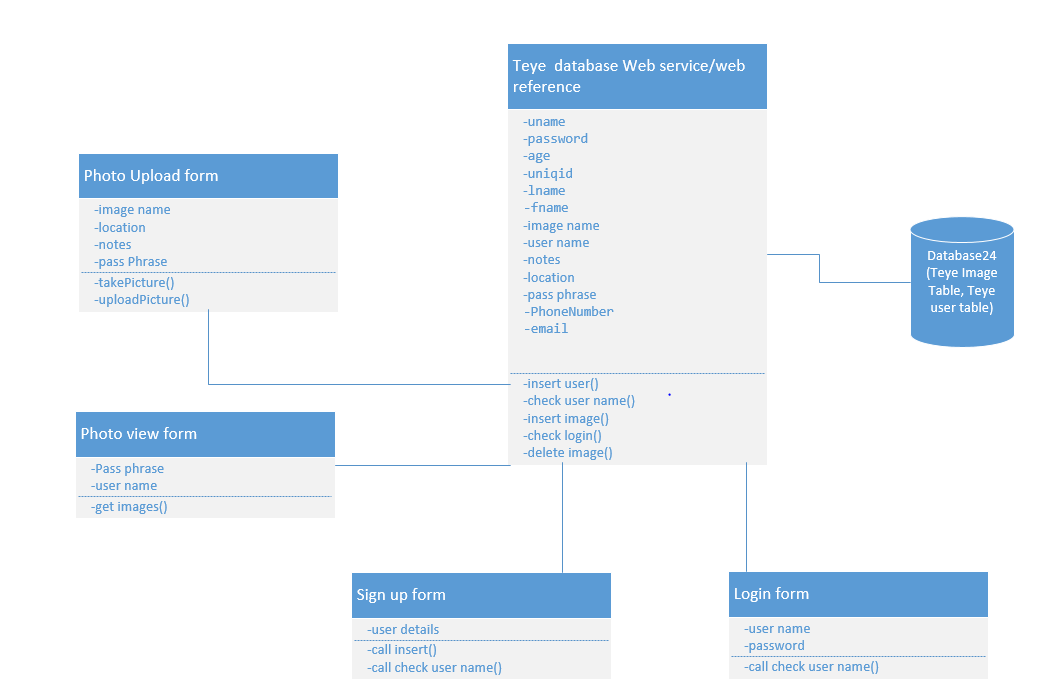
The Unofficial Google Text-To-Speech API

http://techcrunch.com/2009/12/14/the-unofficial-google-text-to-speech-api/

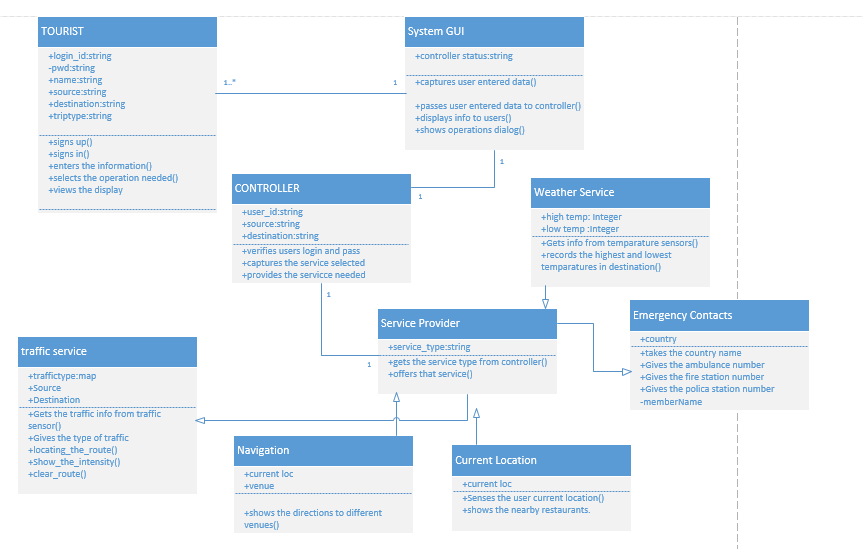
Accommodation:

http://Kayak.com/

**Class Diagram for user details and images insertion and deletion:**



General class diagram:



The class diagram describes the static structure of the application. The class diagram consists of classes, attributes and operations.

**Classes:**

Tourist: This class refers to the user of the application. The user should first sign up with application and them login into it and give his details. And then can use the services of the application.

SystemGUI: This is the interface between the user and the application. It captures the users data and passes it on to the controller .And displays the useful information to user as requested by the controller.

Controller: This is the heart of the whole application. It analyses everything. It updates and verifies database. Contacts the service provider and the brings the service needed into play.

Service Provider: This is used to provide services to the application. Each service is asub class to this super class.

1) weather service: This gives the weather conditions at the given destination.This gives the highest and lowest temperatures at the given position.

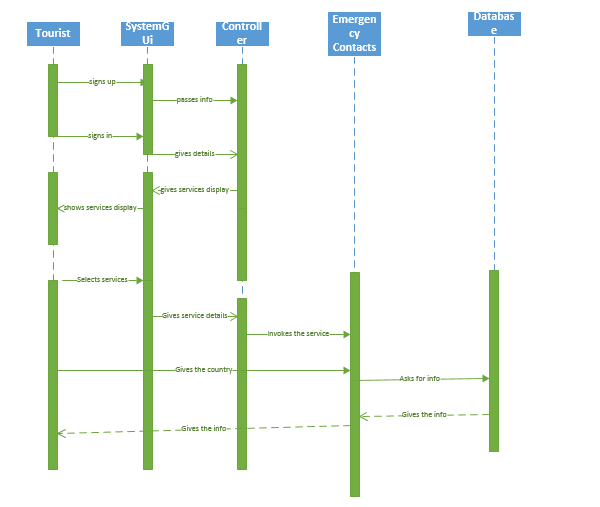
2)Traffic service: This gives the traffic updates at the place required by the user.

3)Current Location: The current location of the user is identified.

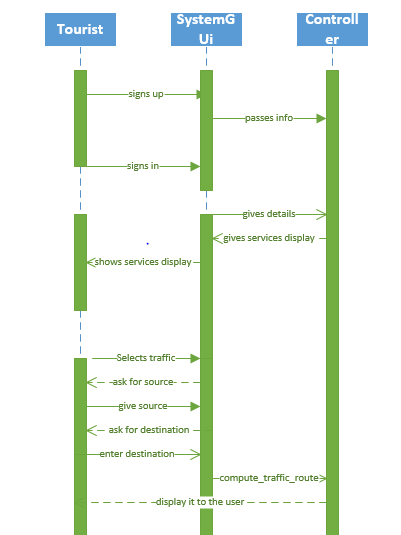
4)Navigation: This shows the navigation from to user’s current position to the place he wants to move.

5)Emergency Contacts: The user can get the emergency contacts like ambulance fire and police man.

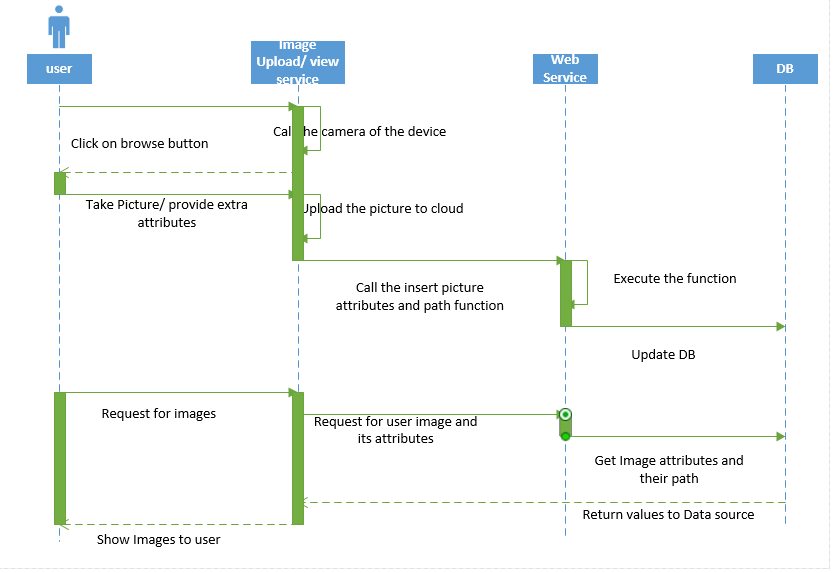
Sequence Diagram for Emergency Contacts:



Sequence diagram for traffic service:



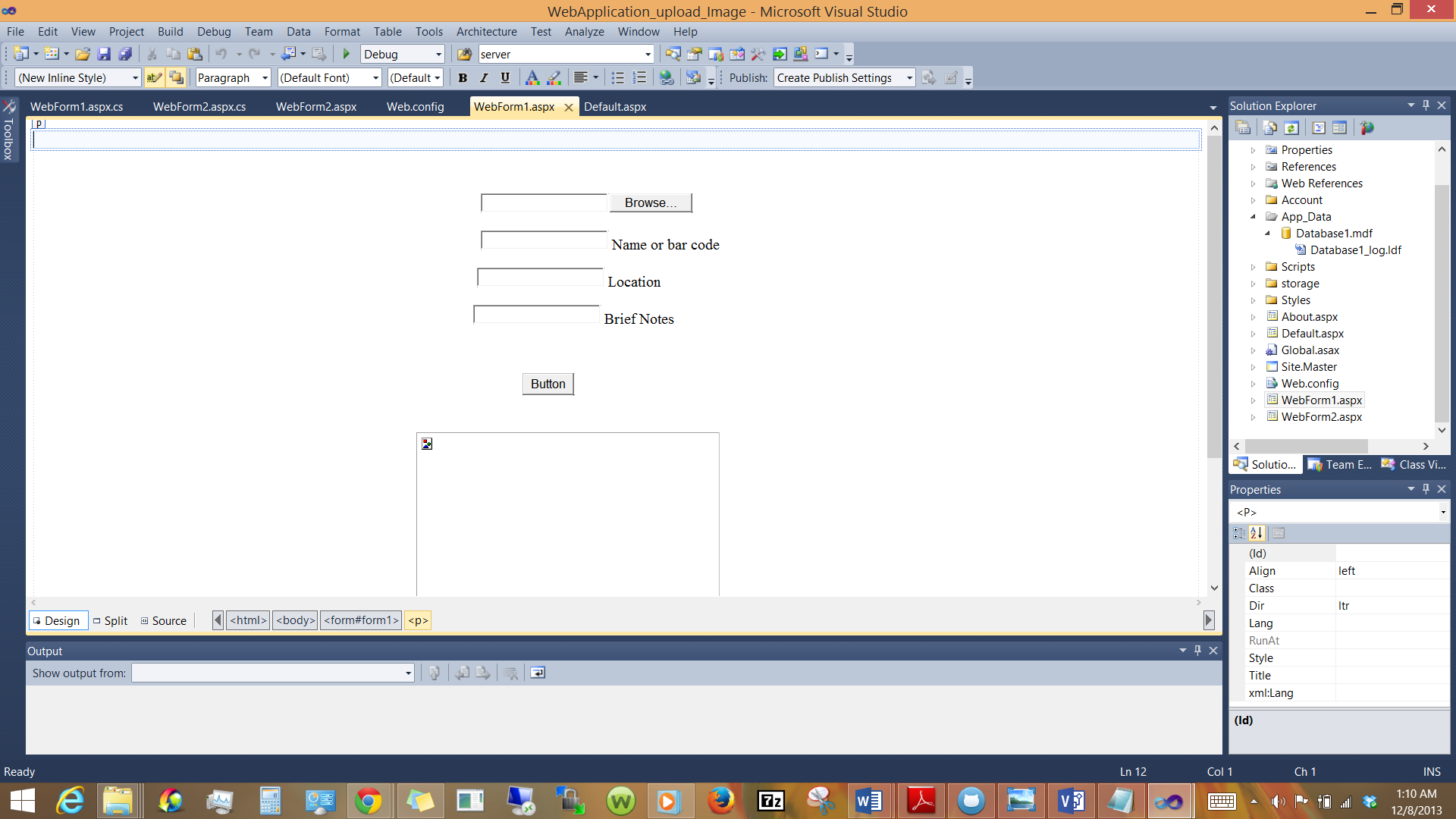
Sequence Diagram for Image upload and view:



Tacking picture and uploading:

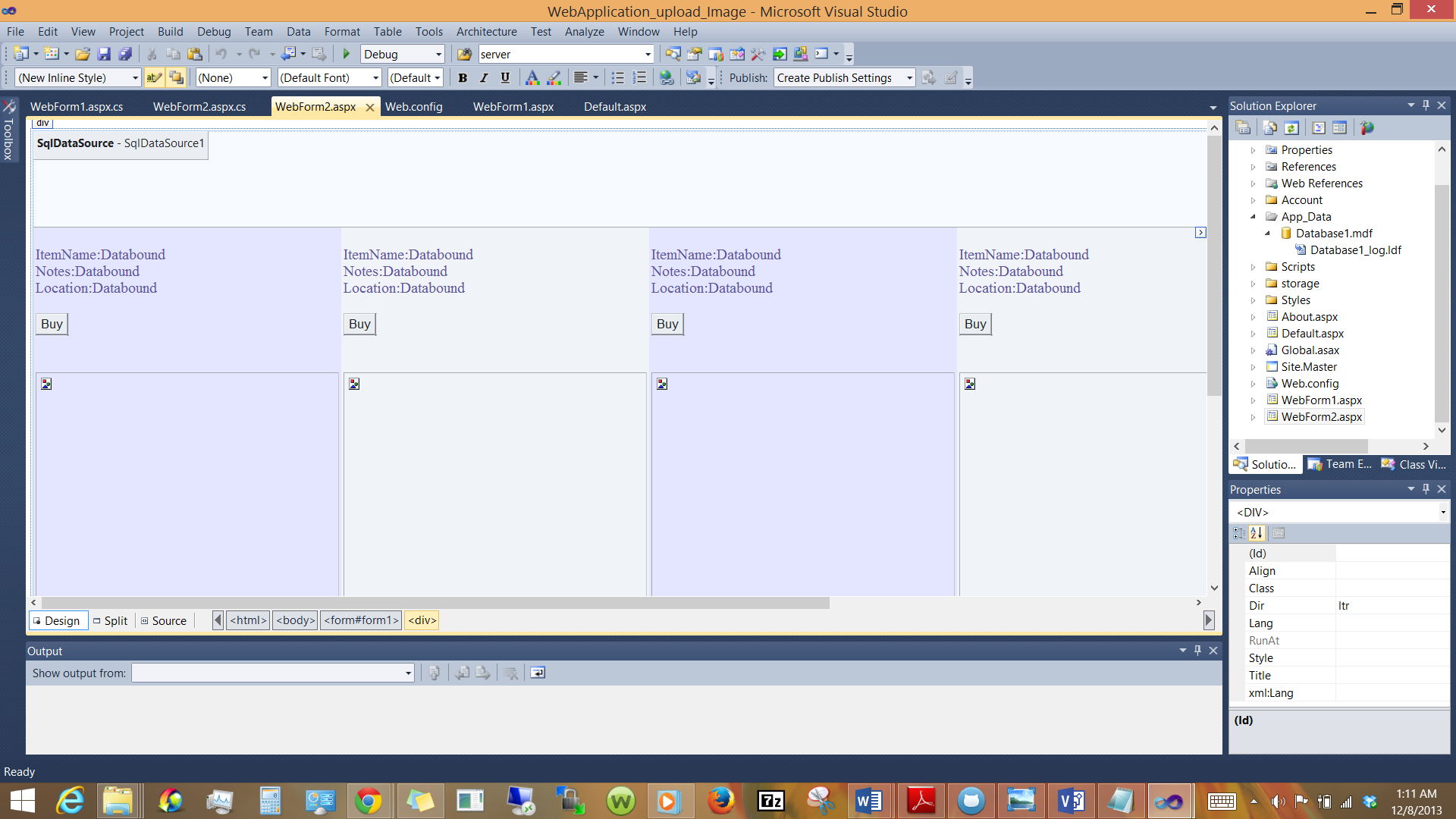
The following form is used to take a picture, as soon as the user clicks on the browse button a java code is invoked which lets the user take a picture from the application and upload it to the cloud. Only the attributes of the picture are stored in a database to save precious memory in the db. The pictures are uploaded to a folder on the cloud called the “Storage” folder.

The attributes of the image and the path of the image are all stored in the db table by calling a web service which has been added as a web reference.



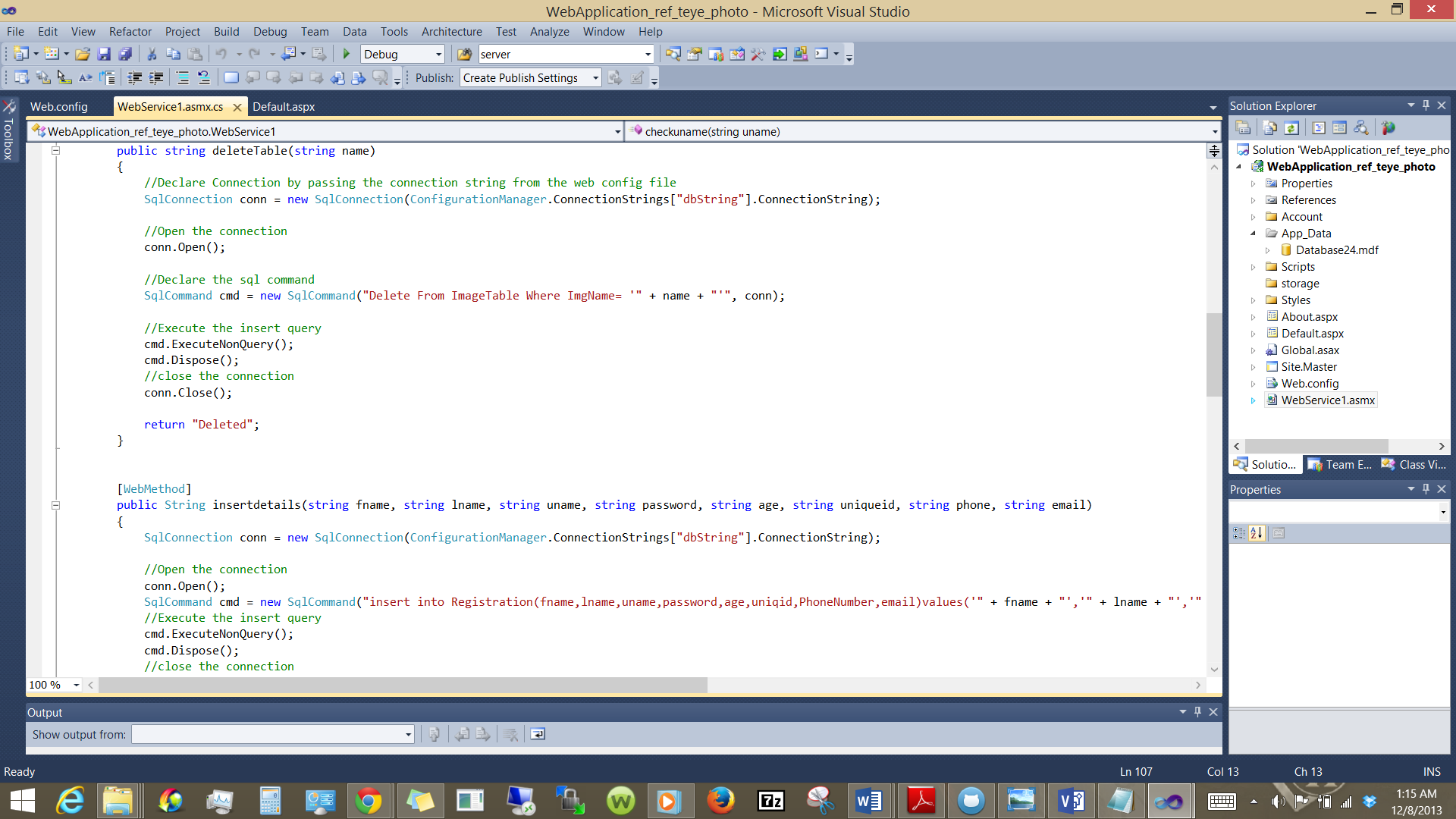
Viewing the uploaded pictures:

When the user logs in the pictures uploaded by the user are displayed by using this template. The data is populated using a SQL data source and data list functionality.



**Common Traveler’s eye web service for data operations on the Traveler’s eye database:**

This is the web service responsible for all the database operations, It has many functions /methods which are called by different funcationalities.

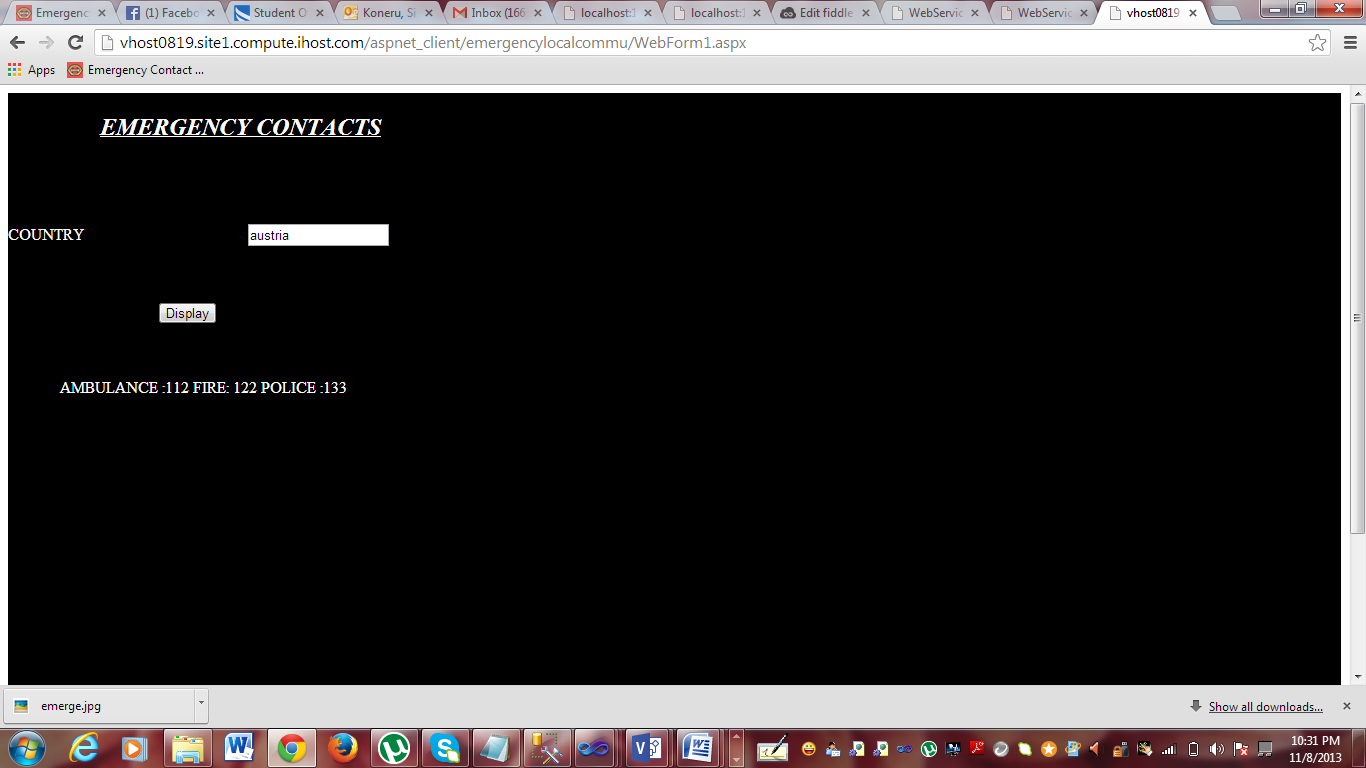


Emergency Contacts:

The emergency contact s like the local police station number, hospital number and any such emergency numbers as soon as the user selects the destination. So even in emergency all the user has to do is press one button. Additionally the app also allows some emergency contacts to be defined by the user.

Initially the code is written in Asp.net and in local machine. And then it is deployed in cloud.

Created a remote web service for emergency contacts. This service can also be used by other applications.



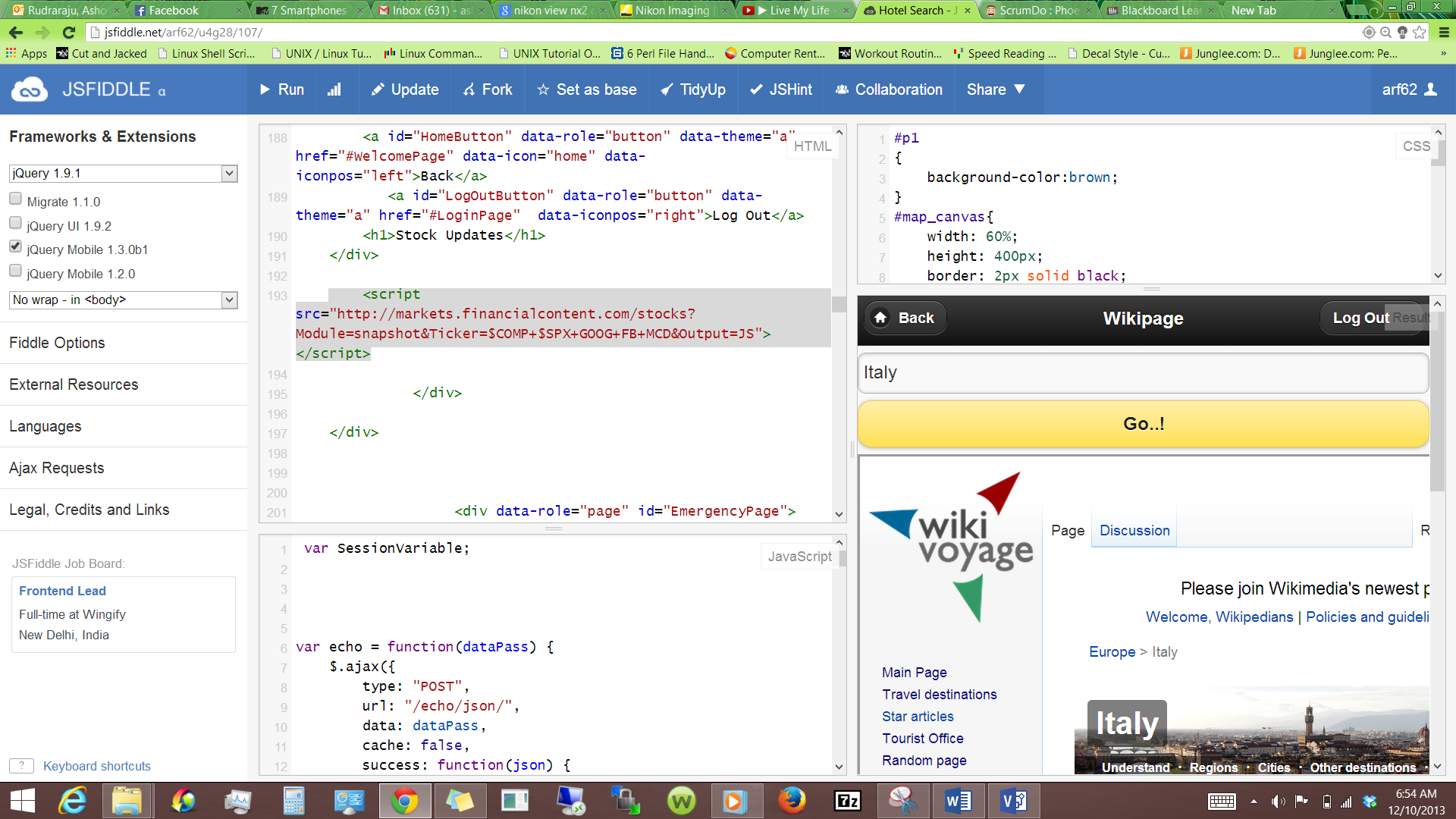
Made the client to communicate with the web service.

**Stock market updates for business users**.

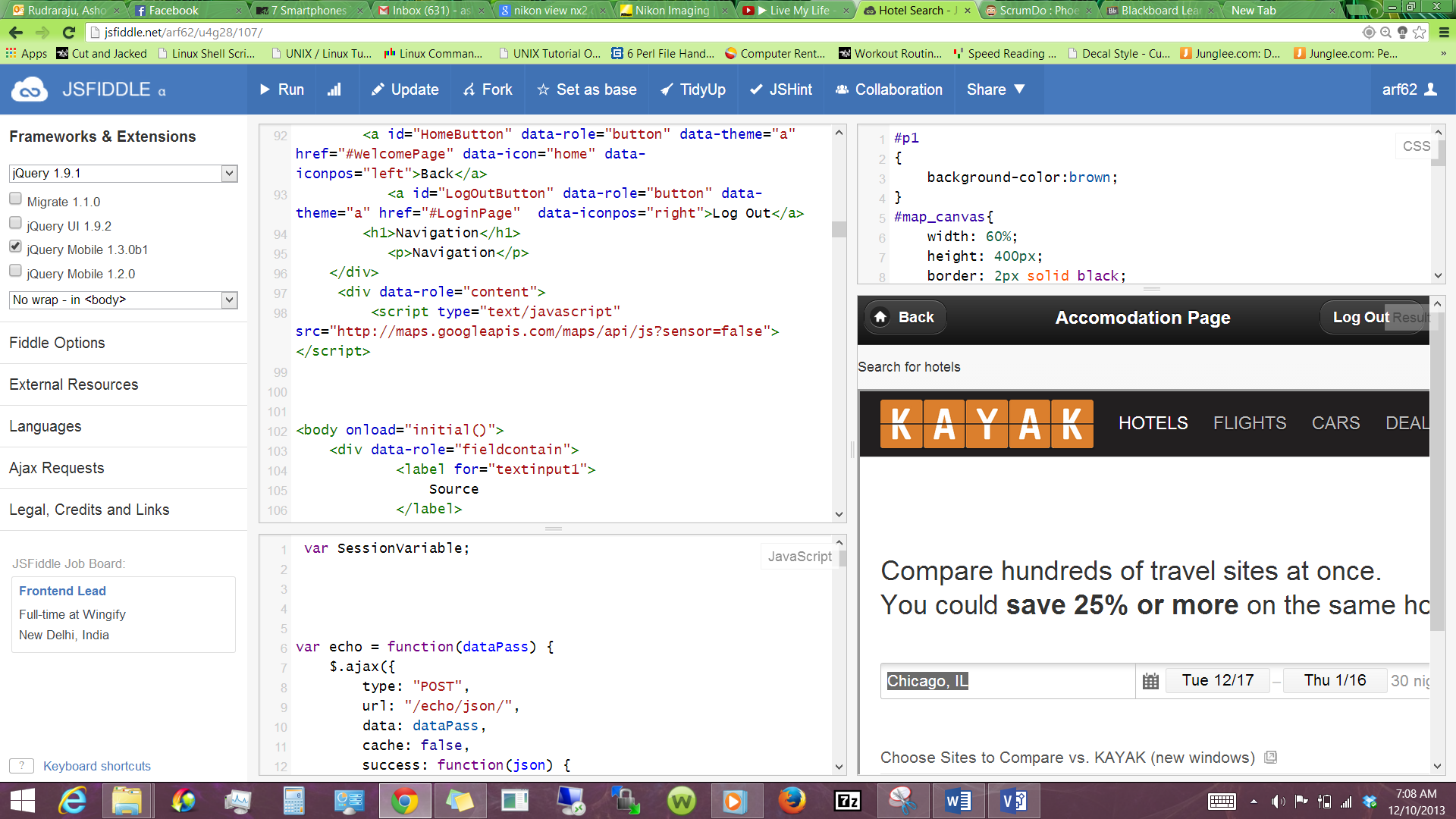
This funcationality is achieved using a widget provided by the “financialcontent.com” As soon as the user clicks on the stock market button, the user is re directed to the below page with the stock updates of prominent companies.



**Wiki Voyager Travel guide information.**

The user provides the name of the state, province or country and then clicks on the “Go..!” button. Then immediately the space below the button displays the wiki voyage page of the provided state, province or country. 

**Find Accommodation:** This functionality is implemented using “Kayak.com”, As soon as the user clicks on the “Find Accommodation” the user is redirected to the kayak.com website and the user can choose his accommodation options.



**FOURTH INCREMENT:**

**Services Designed in Fourth Increment:**

Sending an email to the user using SMTP

Entertaining user by providing a game(Snake Food)

Deployment of redesigned DB and sign up and login functionality

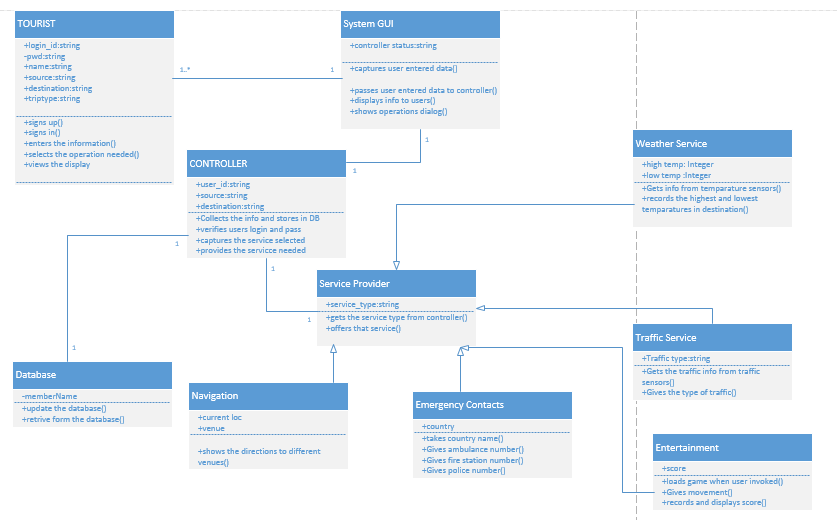
Session Management

Photo and Gallery

Integration with facebook

Traveler’s Eye Website

**CLASS DIAGRAM :**



The class diagram describes the static structure of the application. The class diagram consists of classes, attributes and operations.

**Classes:**

Tourist: This class refers to the user of the application. The user should first sign up with application and them login into it and give his details. And then can use the services of the application.

SystemGUI: This is the interface between the user and the application. It captures the users data and passes it on to the controller .And displays the useful information to user as requested by the controller.

Controller: This is the heart of the whole application. It analyses everything. It updates and verifies database. Contacts the service provider and the brings the service needed into play.

Service Provider: This is used to provide services to the application. Each service is asub class to this super class.

1) Weather Service: This gives the weather conditions at the given destination.This gives the highest and lowest temperatures at the given position.

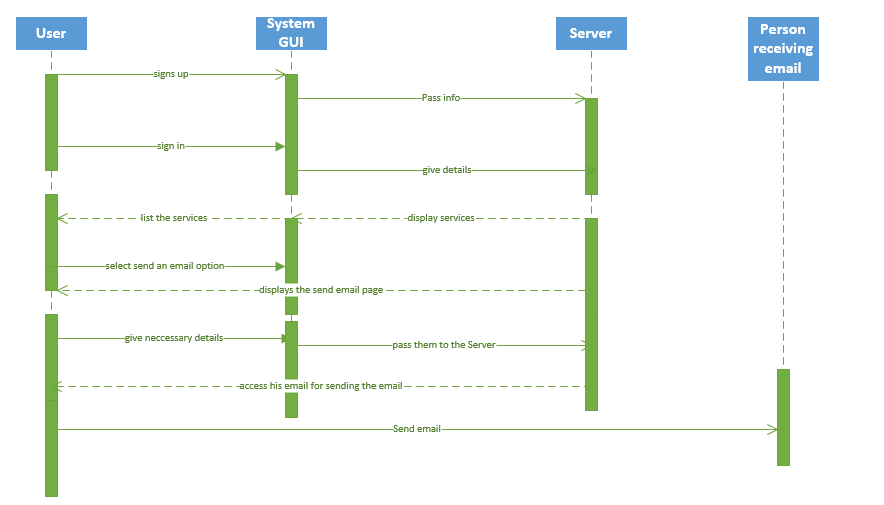
2)Traffic service: This gives the traffic updates at the place required by the user.

3)Entertainment: Provides a game to entertain the user as travelling is boring activity.

4)Navigation: This shows the navigation from to user’s current position to the place he wants to move.

5)Emergency Contacts: The user can get the emergency contacts like ambulance fire and police man.

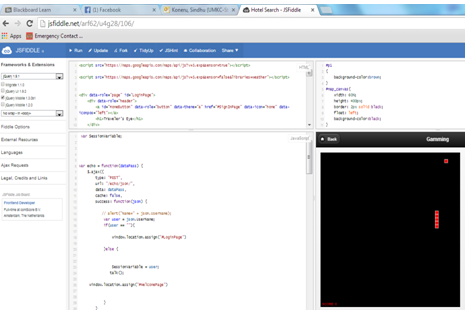
Sequence Diagram For Email System :



**Entertainment (Snake food game) :**

Travelling is always a boring activity. Especially when we travel alone. So our application also provides the user with some entertainment. Using this application he can play games like snake food game.

When the user is bored he just need to click a button and he will get the game and he can play the game. If once this application is downloaded this game can be played even if there is no internet also as we cannot guarantee internet everywhere.



**Integration with facebook:** The application has a facebook page and the users of the application can click on the like or share button as shown below in the screen shot and follow the updates and news about the app.

This functionality is achieved using the Developer.facebook.com scrip functionality.



**Session Management:**

Session Management is done both at the front end and the back end. The front end session management consists of javascript and ajax while the back end session management consists of session variables.

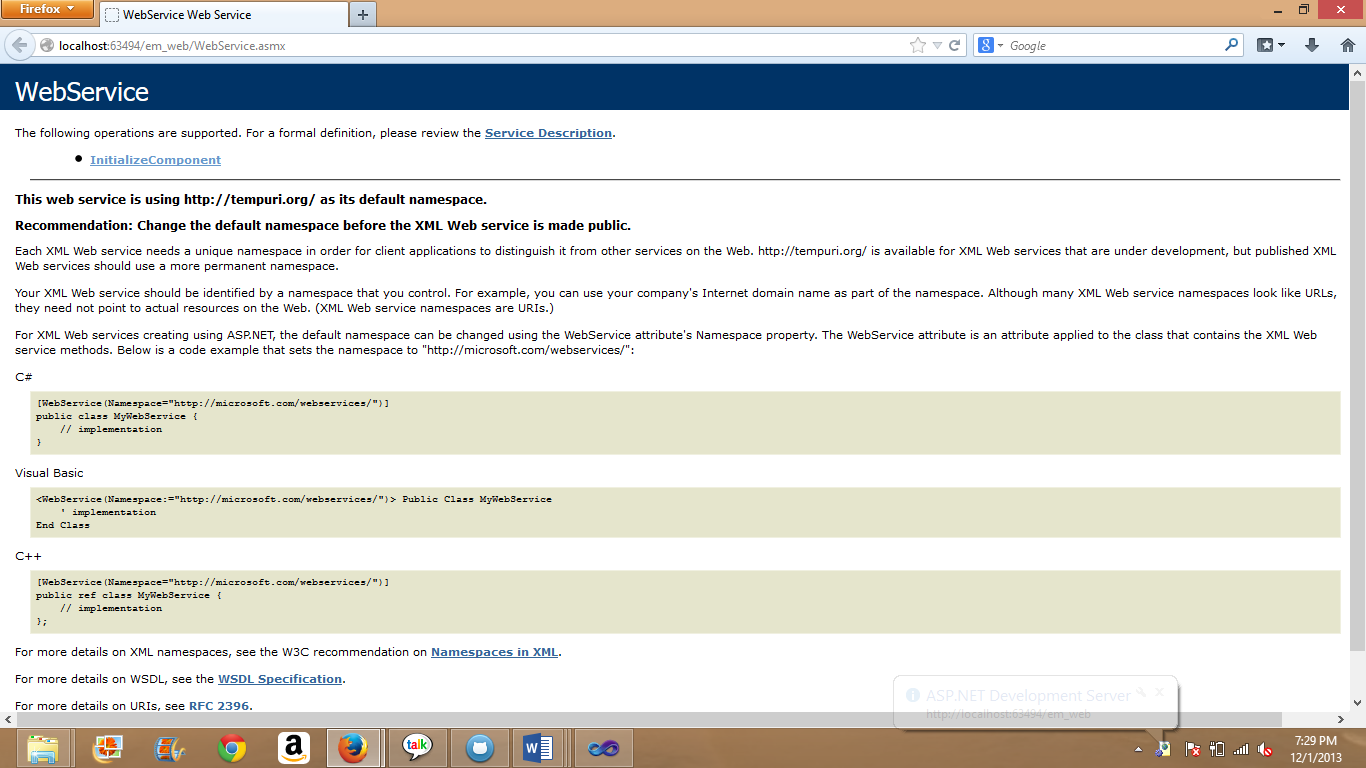
The session is primarily needed for the user galary or the picture viewing functionality. Each user have a different set of images upladed on the web. This is taken care of by using a pass phrase and the user name in a combination.

**Sending an email to the user using SMTP:**

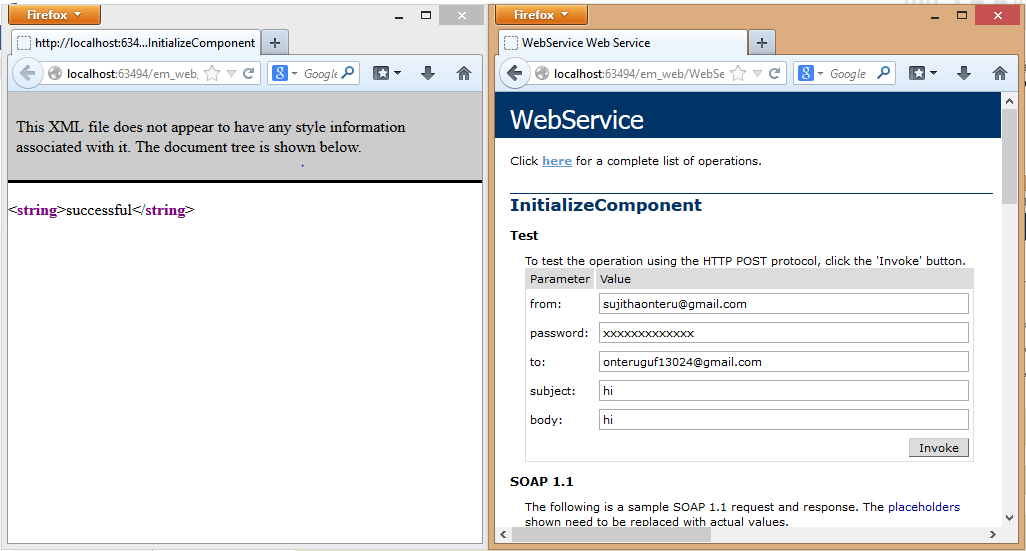
This feature is adverted into the project with an intension of allowing user to leave an email through the app if intended. The moment he signs in he will the flexibility in using the services of the system. While exploring places, if he wants to send his travelling information to his friends via email he can directly use his credentials and compose a message and send it to the user.

For enabling this a web service is been created. EASendMail is been used in creating the web service. EASendMail is an SMTP Component which facilitates us in sending emails through ASP.NET.

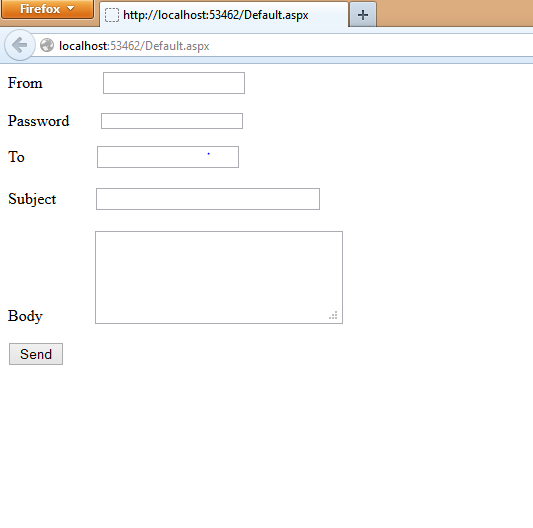
Once webservice is created and debugged on the machine we can see that it opens as below:



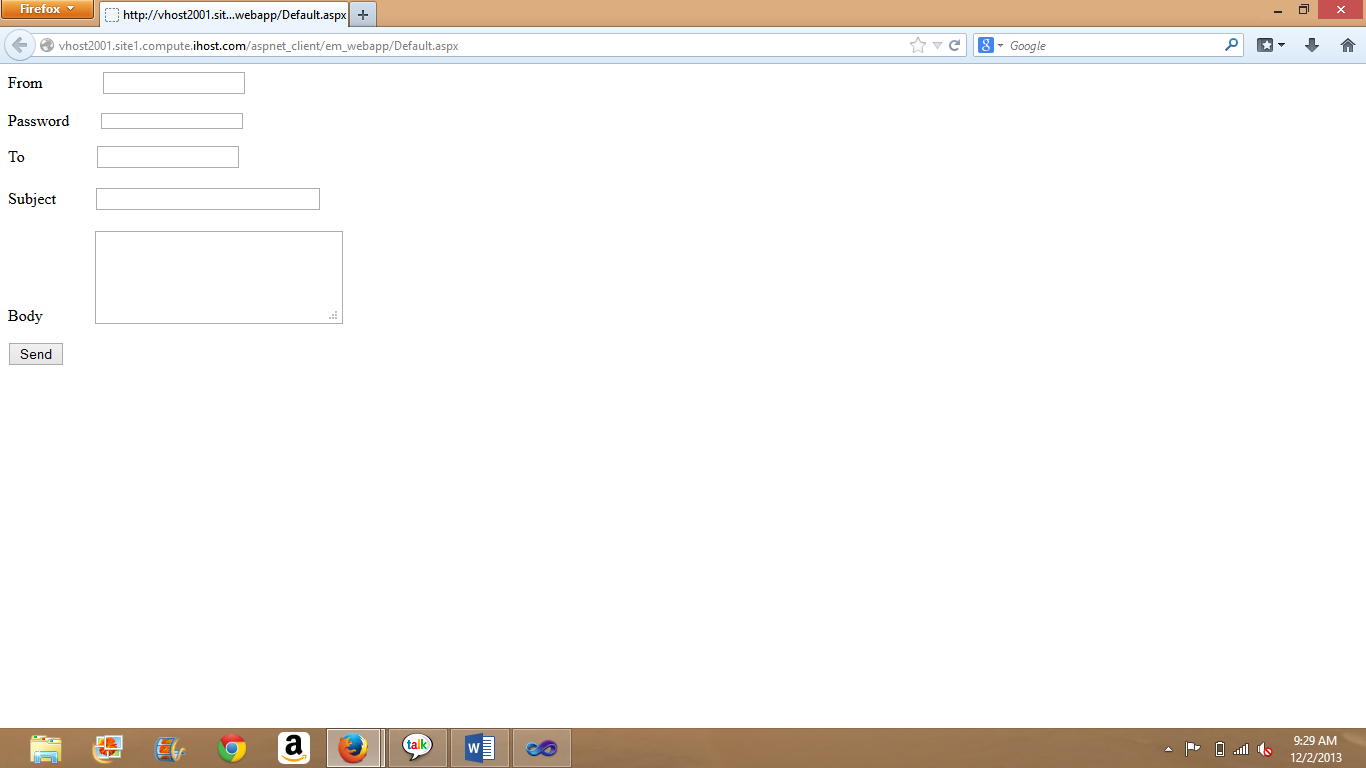
On clicking upon Initialize Component we can notice that a page displays asking him to enter to and from address along with required password and body of the message. It will now display a successful string in an xml format.



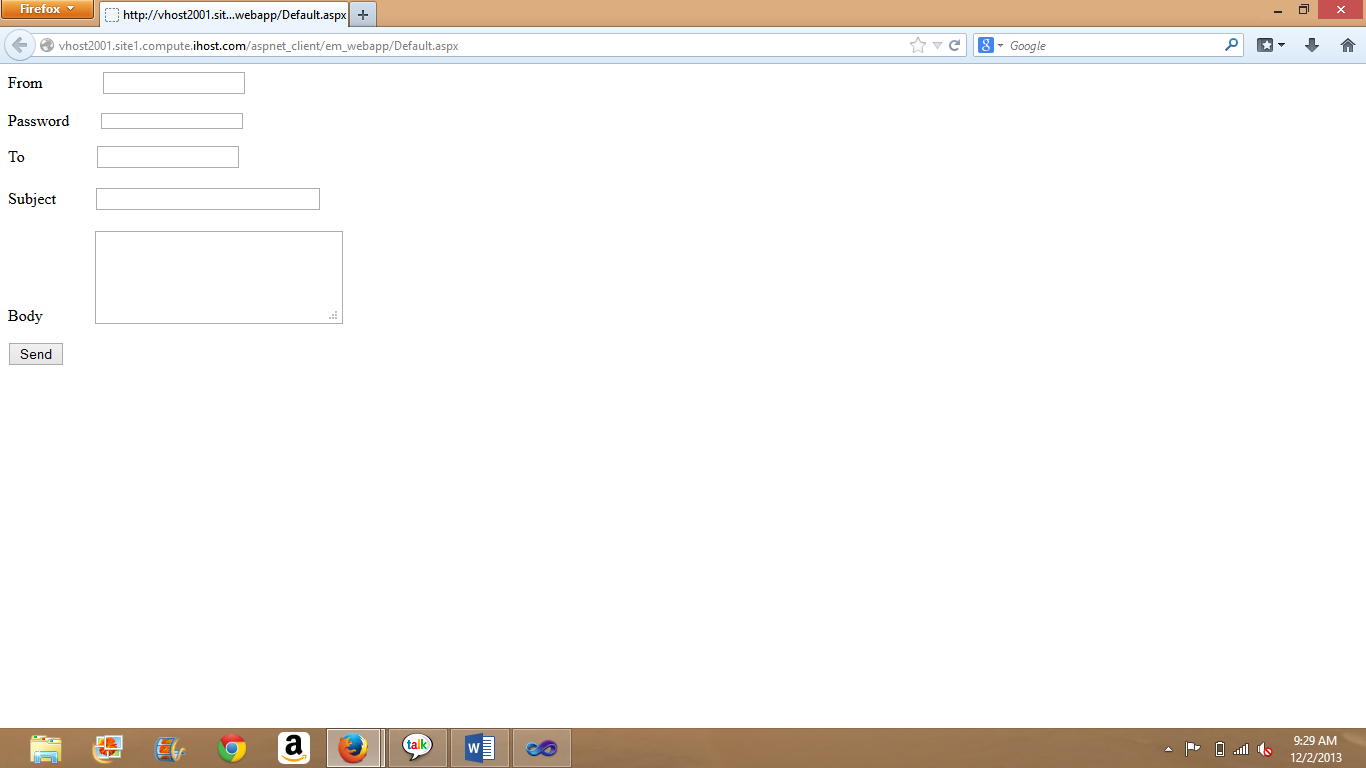
Now the service description of the local service is copied and is added as a web reference to the web application em\_webapp. The needed default design layout is designed for taking the inputs from the user. And Default.aspx will have the necessary invocation code for sending the email. On debugging this on local machine it will look as follows:



As a next step the webservice and web and web application are taken to the cloud for deployment. Once they are successfully deployed on the cloud, the remote is link is copied which is now capable of running on any machine.



Clearly the picture shows that remotely running of the web app is still showing allows us to get the output. It is now included into the mobile interface in jsfiddle.

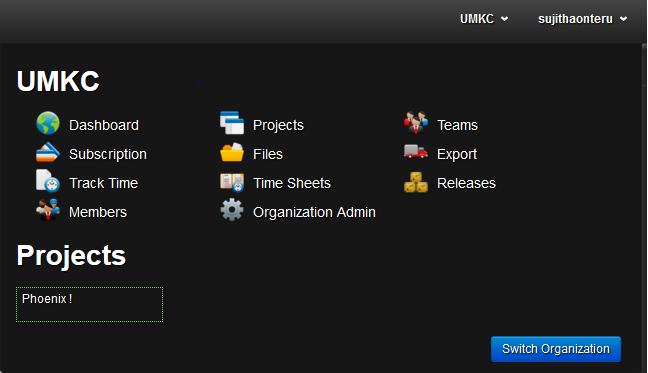


Youtube project video link:

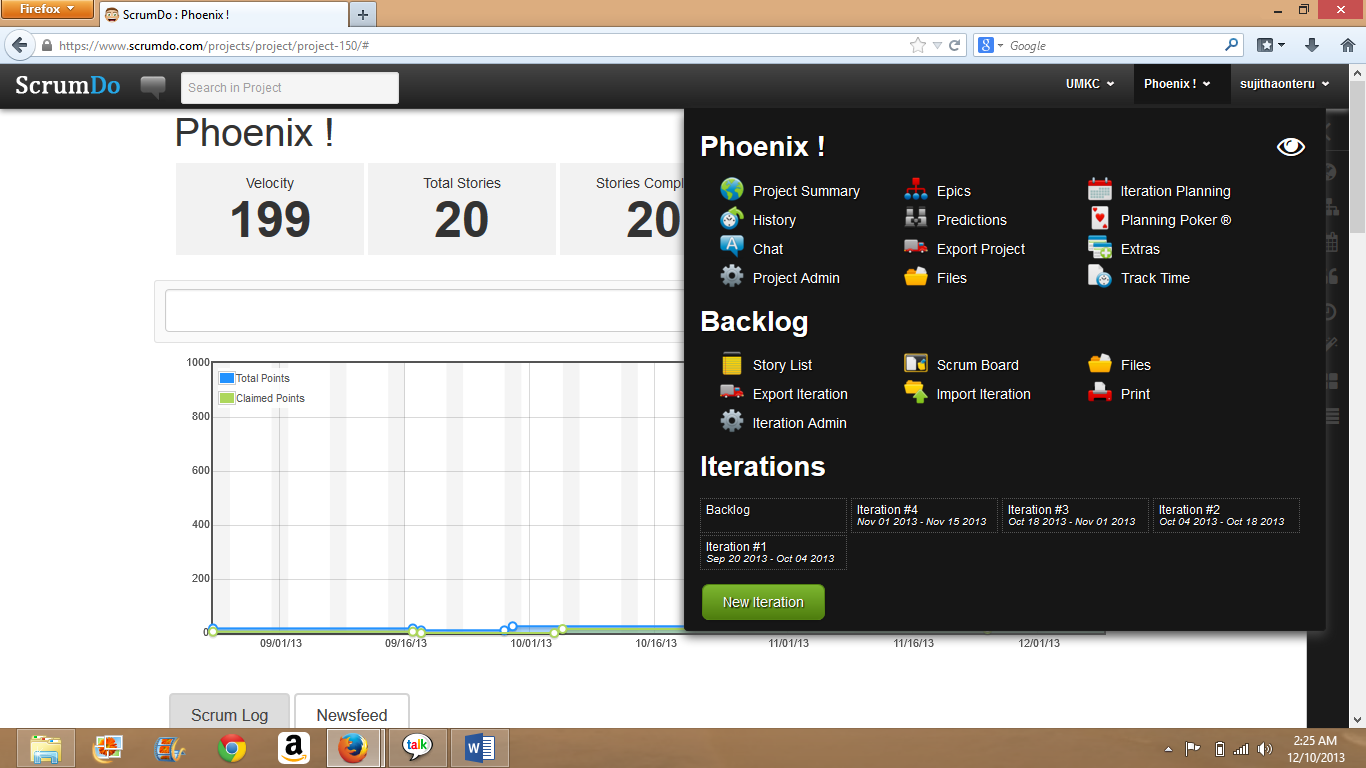
<http://www.youtube.com/watch?v=42kPz2HvCzA>

**Project Management Report:**

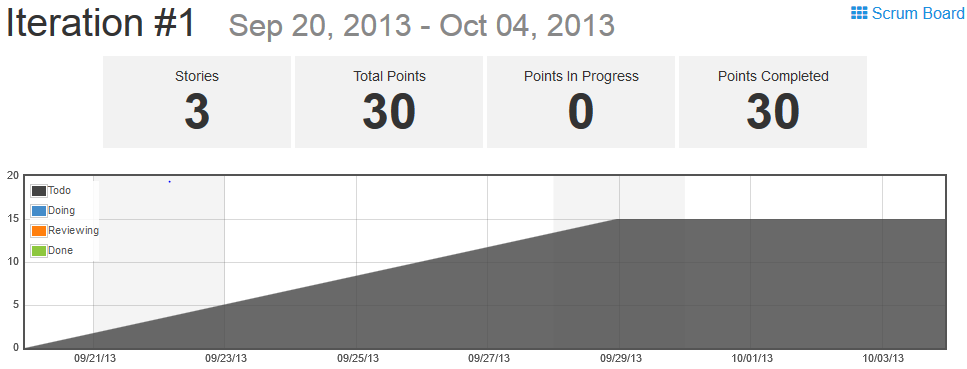
This team project helped each and every member to improve our managing skills apart from technical skills. We adopted the Agile process method namely scrum for this purpose. Scrum is an iterative incremental process of software development commonly used with agile software development. The project is developed within four iterations.



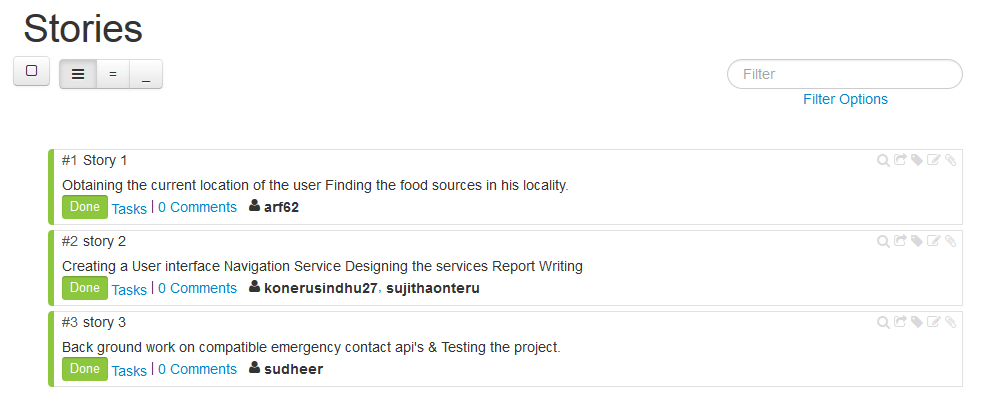
In accordance to follow agile process we used ScrumDO where our project named phoenix is created.



The project is completed in four iterations .We get a finished output for each iteration.



This is how each iteration looks like .We also get burndown chart for each iteration. So that our estimation becomes easy.



Stories are created for each iteration. And those stories are accomplished.

In order to increase productivity and performance of individual members we followed a “Kanban” approach, in this approach, each team member was allowed to pull work from a queue based on is preference. The idea was that if a person is interested in doing something and pulls it form the queue by his own choice then he is more likely to spend more time and produce a good output. The jobs that were not pulled by any team member were assigned to the management (in this case the group leader) on top of the regular job.

**Final project evaluation:** The final output of project satisfies our original requirements by more than 85%. We had to improvise on some of the things that we wanted to implement like the hotel booking service and the difference between a leisure traveler and a business traveler. After spending some time on both of these services we decided that a lot of very convenient hotel booking services were already available and decided to make use of “kayak.com”. Then we stumbled onto the fact that more than 70% of international travel trips are hybrids (they have both leisure and business aspects) thus we decided to make a generic app that would cater to both the services. Also our original design was confined to the United States, but the final output is accurate throughout the world.

From the very beginning we had a very clear structure and idea of how the app should work at the end. We followed our original design process for the entire life time of the project. We incorporated a lot of agile methodologies into our project management and we are very pleased with the result. Instead of going with the do it one and do it right approach we started working on all our major functionalities from the very beginning and we followed up iteration after iteration of a better product. We had regular team meeting which were scheduled to match every team member’s convenience.

If this was a real world project we probably would have followed a slightly different and scalable design and pay more attention to testing.

We did have faced human and technical challenges throughout the project development. But we overcame then in the most professional ways possible, through the guidance of Dr.Lee and Feichen.

Biblography:

<https://play.google.com/store/apps/details?id=com.sekond.touristguide.paris>

<https://play.google.com/store/apps/details?id=yokohama_guide.a2ydesigns>

<https://play.google.com/store/apps/details?id=Quinto.app.GreekTouristGuide>

<https://play.google.com/store/apps/details?id=com.mymobilemanchester>

<https://play.google.com/store/apps/details?id=jp.softbank.mb.fa>

<http://www.slideshare.net/DennisDegryse/agile-development-8481635>

<http://research.microsoft.com/apps/pubs/default.aspx?id=56015>

[https://play.google.com/store/apps/details?id=com.sekond.touristguide.paris](https://bluprd0113.outlook.com/owa/redir.aspx?C=Vgr7hPZzZ02Th3QBARU_jlkYDsVoydAIAlc97V4A8nra1NCj5Z8krH4kM_WjBY7EWKPBdo4xnwQ.&URL=https%3a%2f%2fplay.google.com%2fstore%2fapps%2fdetails%3fid%3dcom.sekond.touristguide.paris)

[https://play.google.com/store/apps/details?id=yokohama\_guide.a2ydesigns](https://bluprd0113.outlook.com/owa/redir.aspx?C=Vgr7hPZzZ02Th3QBARU_jlkYDsVoydAIAlc97V4A8nra1NCj5Z8krH4kM_WjBY7EWKPBdo4xnwQ.&URL=https%3a%2f%2fplay.google.com%2fstore%2fapps%2fdetails%3fid%3dyokohama_guide.a2ydesigns)

[https://play.google.com/store/apps/details?id=Quinto.app.GreekTouristGuide](https://bluprd0113.outlook.com/owa/redir.aspx?C=Vgr7hPZzZ02Th3QBARU_jlkYDsVoydAIAlc97V4A8nra1NCj5Z8krH4kM_WjBY7EWKPBdo4xnwQ.&URL=https%3a%2f%2fplay.google.com%2fstore%2fapps%2fdetails%3fid%3dQuinto.app.GreekTouristGuide)

[https://play.google.com/store/apps/details?id=com.mymobilemanchester](https://bluprd0113.outlook.com/owa/redir.aspx?C=Vgr7hPZzZ02Th3QBARU_jlkYDsVoydAIAlc97V4A8nra1NCj5Z8krH4kM_WjBY7EWKPBdo4xnwQ.&URL=https%3a%2f%2fplay.google.com%2fstore%2fapps%2fdetails%3fid%3dcom.mymobilemanchester)

[https://play.google.com/store/apps/details?id=jp.softbank.mb.fa](https://bluprd0113.outlook.com/owa/redir.aspx?C=Vgr7hPZzZ02Th3QBARU_jlkYDsVoydAIAlc97V4A8nra1NCj5Z8krH4kM_WjBY7EWKPBdo4xnwQ.&URL=https%3a%2f%2fplay.google.com%2fstore%2fapps%2fdetails%3fid%3djp.softbank.mb.fa)

[http://www.slideshare.net/DennisDegryse/agile-development-8481635](https://bluprd0113.outlook.com/owa/redir.aspx?C=Vgr7hPZzZ02Th3QBARU_jlkYDsVoydAIAlc97V4A8nra1NCj5Z8krH4kM_WjBY7EWKPBdo4xnwQ.&URL=http%3a%2f%2fwww.slideshare.net%2fDennisDegryse%2fagile-development-8481635)

[http://research.microsoft.com/apps/pubs/default.aspx?id=56015](https://bluprd0113.outlook.com/owa/redir.aspx?C=Vgr7hPZzZ02Th3QBARU_jlkYDsVoydAIAlc97V4A8nra1NCj5Z8krH4kM_WjBY7EWKPBdo4xnwQ.&URL=http%3a%2f%2fresearch.microsoft.com%2fapps%2fpubs%2fdefault.aspx%3fid%3d56015)

Google Image Search API

https://developers.google.com/image-search/

Yahoo Image Search API

http://www.codeproject.com/Articles/22218/An-API-for-Yahoo-Image-Search

Flickr Image Search API

http://www.flickr.com/services/api/flickr.photos.search.html

A free online Talking Dictionary of English Pronunciation

http://www.howjsay.com/

Speech to Text API

http://mycaption.com/resources/api

Google Text to Speech

http://weston.ruter.net/2009/12/12/google-tts/

The Unofficial Google Text-To-Speech API

http://techcrunch.com/2009/12/14/the-unofficial-google-text-to-speech-api/

Speech Recognition

http://developer.android.com/reference/android/speech/RecognizerIntent.html

73 Voice APIs: Twilio, Skype and Tropo

http://blog.programmableweb.com/2012/04/10/73-voice-apis-twilio-skype-and-tropo/

54 New APIs: Wikipedia, Live Chat, Text-to-Speech and Cloud Collaboration

http://blog.programmableweb.com/2011/03/06/54-new-apis-wikipedia-live-chat-text-to-speech-and-cloud-collaboration/

Google Earth Driving Simulation

http://earth-api-samples.googlecode.com/svn/trunk/demos/drive-simulator/index.html

Google Maps API for Business

https://developers.google.com/maps/documentation/business/

ArcGIS Server REST API

<http://sampleserver1.arcgisonline.com/ArcGIS/SDK/REST/index.html?servicesdirectory.html>

and materials provided in class.