

# Message Box

By

**Bilal Ahmed Mughal**

**2008-NUST-BIT-007**

**Abdullah Chohan**

**2008-NUST-BIT-003**

Submitted to

**Advisor:**

**Muhammad Qaisar Chaudhary**

**Co-Advisor:**

**Dr. Abdul Ghafoor Abbasi**



**Project documentation submitted in partial fulfillment of the requirements  
for the degree of**

**Bachelors in Information Technology (BIT)**

**NUST School of Electrical Engineering & Computer Science (SEECS)**

**National University of Sciences and Technology (NUST)**

**Islamabad, Pakistan**

# Project Report

---

## CERTIFICATE

It is certified that the contents and form of report entitled “Message Box” submitted by *Bilal Ahmed Mughal (2008-NUST-BIT-007)* and *Abdullah Chohan (2008-NUST-BIT-003)* have been found satisfactory for the requirement of the degree.

Advisor: \_\_\_\_\_

(Mr. Muhammad Qaisar Chaudhary)

Co-Advisor: \_\_\_\_\_

(Dr. Abdul Ghafoor Abbasi)

**DEDICATION**

To Allah the Almighty

&

To Our Family, Faculty and Friends  
Who have always been there for us

## **ACKNOWLEDGEMENTS**

First of all we are thankful to Almighty Allah for giving us the strength and ability to complete this project and we are extremely thankful to our advisor and co-advisor, Mr. Muhammad Qaisar Chaudhary, and Mr. Dr. Abdul Ghafoor Abbasi respectively for helping us throughout the course in accomplishing our final year project. Their guidance, support and motivation encouraged us in achieving the objectives of the project. It is an honor and pleasure for us to work on this project under their supervision.

Secondly, we acknowledge our parents; their love, prayers and trust in us helped us in completing this task, We will like to thank our friends Muhammad Usman Mansha, Samran Zulfiqar, Zumair Ali Zafar, Osama Misbah, Asim Mushtaq, Sarosh Bilal, Fahad Yasin and last but not the least Muneeb Ahmad for their help and motivation to accomplish this task successfully.

# Project Report

---

## Abstract

Around 3 billion mobile applications based on Google android and iOS platform are available in the market. In Aug 2012, Microsoft launched highly anticipated Windows 8 to make their place in smartphone market. .

Smartphones are more common among young people and they prefer messaging over calling using build in as well as third party applications. Such applications have basic features like scheduling of messages, group messaging. Those applications lack in extended features like categorization of messages, sync/backup data from server and messages searching, also do not have all features together.

The Message Box brings the exceptional functions together in a signal application. The application allows user to send free messages over internet while providing special functionality like scheduling of messages with respect to time, categorization of messages in folders with respect to any level in hierarchy, group messaging, contact grouping, synchronization/backup with server and message searching within inbox based on date, sender/receiver and folder. The key factors in making Message Box a solid utility are HCI aspects, managing server resources to provide better infrastructure for message delivery and providing best user experience.

# Project Report

---

## Table of Contents

	Abstract	5
1	INTRODUCTION	8
1.1.	Domain Knowledge:	8
1.2.	Problem Statement:	9
1.3.	Goals and Objectives	9
1.4.	Requirement Analysis	10
1.4.1.	Functional Requirements	10
1.4.2.	Non-Functional Requirements:	10
1.5.	Solution:	10
1.6.	Core Modules:	11
1.6.1.	Grouping Module:	11
1.6.2.	Categorization Module:	11
1.6.3.	Synchronization Module:	11
1.6.4.	Intelligent Searching Module:	11
1.6.5.	Scheduling Module:	12
1.7.	Mobile Application Development:	12
1.7.1.	Messaging Applications:	12
2	LITERATURE REVIEW	13
2.1.	SMS Applications Comparison	13
2.1.1.	WhatsApp	13
2.1.2.	Viber	14
2.1.3.	Go SMS pro	15
2.1.4.	Group SMS Plus 2.0	16
2.1.5.	Group me	17
2.1.6.	SMS Scheduler:	18
2.1.7.	Buddy Groups	19
2.1.8.	Zune:	20
2.2.	Comparison:	21
2.3.	End Product Description:	22
3	FUNCTIONALITY AND DESIGN	23
3.1.	Architecture	23

# Project Report

---

3.1.1.	MVC Architecture	23
	Model:	23
	View:	23
	Controller:	23
3.1.2.	Client Side Architecture:	24
3.1.3.	Server Architecture (as a whole)	25
3.1.4.	Entity Relation Ship Diagram:	28
3.2.	Modules:	29
3.2.1.	Schedule:	29
3.2.2.	Grouping:	29
3.2.3.	Folder Making:	30
3.2.4.	Instant Messaging	30
3.2.5.	Synchronization and backup.	31
3.3.	Deliverables:	31
3.4.	Schedule of Deliverables (Tentative):	32
4	PROJECT IMPLEMENTATION	33
4.1.	Contact Grouping Module:	33
4.2.	Scheduling Module:	35
4.3.	Inbox & Conversations Module:	37
4.4.	Contacts Adding Module:	39
4.5.	Foldering Module:	41
4.6.	Instant Messaging	44
4.7.	Searching Module:	46
4.8.	Use Case Diagram:	47
5	RECOMMENDATIONS AND CONCLUSIONS	48
6	REFERENCES	48

---

# Project Report

---

## LIST OF FIGURES

Figure 1 WhatsApp on Windows 8.....	14
Figure 2 Viber on Windows Phone.....	15
Figure 3 Go SMS pro.....	16
Figure 4 Group SMS Plus 2.0.....	17
Figure 5 Group me.....	18
Figure 6 SMS Scheduler.....	19
Figure 7 Buddy Groups.....	20
Figure 8 Zune Windows Based.....	21
Figure 9: MVC.....	25
Figure 10: Client Side Architecture.....	26
Figure 11: Server Architecture.....	27
Figure 12 Windows Azure Architecture.....	28
Figure 13: Entity Relationship Diagram.....	29
Figure 14 Entity Relationship Diagram.....	29
Figure 15 Contact Grouping Module.....	34
Figure 16: Activity Diagram Grouping.....	35
Figure 17 Scheduling Module.....	36
Figure 18 Scheduling Module.....	37
Figure 19: Activity Diagram Scheduling.....	37
Figure 20 Inbox & Conversations Module.....	38
Figure 21 Inbox & Conversations Module.....	39
Figure 22: Activity Diagram Conversation.....	40
Figure 23 Contacts Adding Module.....	41
Figure 24: Activity Diagram Contacts.....	41
Figure 25 Foldering Module.....	42
Figure 26 Foldering Module.....	43
Figure 27 Foldering Module.....	43
Figure 28: Activity Diagram Folders.....	44
Figure 29 Searching Module.....	47
Figure 30: Activity Diagram Searching.....	48
Figure 31: Use Case Diagram.....	48



## Chapter 1

### 1. INTRODUCTION

#### 1.1. Domain Knowledge:

Today smart mobile and application user are growing rapidly, and Gartner forecasts that by year 2014, there are more than 70 billion mobile applications downloads from app stores every year<sup>[1]</sup>. With the ever increasing use of mobile smart-phones the application development is moving from consumer-only phenomena to an enterprise focus.

In order successfully satisfy the user and provide solutions to the problem of the users, IT solution companies develops applications for the smart phones and different platform for different phones such as:

- iOS Application Development
- Android Application Development
- BlackBerry Application Development
- Windows Phone Application Development

There are enormous types of applications developed by companies and developers due to outsourcing of the logic as well as independent development of such applications, free lancing is such platform that provide development from free developer and companies also take services from these developers, we can say today that monopoly of application development from past year is now transferred to the free developers also to larger extent.

So the types of applications are follows:

- Location-Based Services Providing apps
- Mobile Search and Browsing
- Mobile Health Monitoring
- Money Transfer and Payment
- Mobile Music App
- Instant Messaging App
- Advertising

# Project Report

---

Making any application proper planning and strategically analysis is required for the successful working application. Importance of messaging application is obvious in this era where users have no time to talk face-to-face and it's expensive to call, so people prefer sending messages which in our case is free of costs unlike normal SMS which is sent over cellular network.

## **1.2. Problem Statement:**

There are many Messaging applications for smart phones including built-in as well as third party applications but Messaging applications having extended features are very rare and "expensive" and also lack in user experience those rare extended features are grouping of the messages, filtering the messages, efficient scheduling and searching, categorization and synchronization (backup).

## **1.3. Goals and Objectives**

To develop an application that:

- Have user interactive interface encouraging HCI standards and rules.
- Is scalable for grouping having more members than usual.
- Has bottomless business value.
- Has efficient memory management
- Has bug less business logic.
- Can provide full scale features availability.
- Can target SMS and instant messaging together.

## 1.4. Requirement Analysis

### 1.4.1. Functional Requirements

Following are the project's functional requirements:

- User shall be able to group messages based on contact information.
- User shall be able to schedule messages with respect to time
- User must be able to search intelligently with in message directory
- User must be able to synchronize messages to server
- User shall be able to understand the interface of the application easily
- User must hope for a crash proof and bug-less end product.
- User must hope for the end product capable of above features.
- User shall hope for 24/7 available, scalable and reliable product.

### 1.4.2. Non-Functional Requirements:

Following are the project's non-functional requirements:

- Application should be made in C-sharp programming language native to Windows.
- Every request's response time should not be more than 3-4 seconds.
- Application must be scalable to handle large group off-course.
- Application must be scalable to provide efficient searching using algorithms.
- Interaction between user and application should be made easy
- The code and architecture must be generic in nature and can accept changes to cater future needs and demands.

## 1.5. Solution:

The main aim of this project is to provide cost effective messaging application having extensive and rare features that are not available is existing applications resembling our application, such as

- Grouping and categorization management of Messages
- Efficient scheduling of Messages
- Searching intelligently within the inbox
- Filtering and blocking the Messaging
- Synchronizing the Messages directory with the Server

## **1.6. Core Modules:**

### **1.6.1. Grouping Module:**

In this module grouping of messaging in groups are made possible, User can make groups of contacts that allows user to send message to multiple contacts by just sending once. Also this module enables make relevant groups to communicate with the added members of the group efficiently and easily.

### **1.6.2. Categorization Module:**

This module enables user to categorize and make folder of the conversations and messages. It allows creation of folders on multiple hierarchical levels to manage messages by users.

### **1.6.3. Synchronization Module:**

Synchronization module provides feature just like backup the mobile phone with server for backup, and whenever the user delete the app or loss the phone, the whole message data in this application can be re-synced from the sever.

### **1.6.4. Intelligent Searching Module:**

Searching module provides services of the searching within the message inbox with respect to any keyword, folder, message's sender and contact number.

## **1.6.5. Scheduling Module:**

This module allows user to schedule messages for any specified time period, as a user can be busy in other activities so this module enables user to create a message and input the time for the message to be send to the recipient right away, like a person is sleeping and birthday message is to be send at 12am, scheduling message at that time makes sure that the message is sent to the intended recipient.

## **1.7. Mobile Application Development:**

Mobile Application Development is a very broad and growing field. New applications are coming in market day in day out and targeting every group alike may it be young, old or males and females. The applications, which are being developed and are famous, have many categories but famous applications in management category allows its users to manage their contacts, messages, music, photos and utilities of the mobile phone. We have covered the SMS and messages applications for this project and provide management aspects also.

### **1.7.1. Messaging Applications:**

There are huge number of applications developed that deal in management of messages allowing messages to be scheduled for specific time and be managed in groups. Some messaging applications also allow us to backup of messages, photos, music, and contacts with some cloud or with personal computer. Some famous and renowned Messaging applications are as follows:

- WhatsApp
- Viber
- Go SMS pro
- Group SMS & Scheduler
- Group SMS Plus 2.0
- CALL+SMS Blocker
- SMS texting from Computer
- Group me
- SMS Scheduler
- Buddy Groups
- SMS Filter

### 2. LITERATURE REVIEW

Apple, Android and Windows Phone are currently the main competitors in the smartphone market; all of the cited competitors have different type of technology and development languages that run on different technological device like iPhone, Android phone and Windows Phone etc. In today's market, the smartphone industry is crawling with developers developing applications that run on these devices, so comparison of application of our domain as follows:

#### 2.1. SMS Applications Comparison

##### 2.1.1. WhatsApp

An application that is available on all smartphones. It allows instant messaging over internet. It is available free on all marketplaces. Also allows sending messages and media content over internet and maintains user's profile also [\[2\]](#).

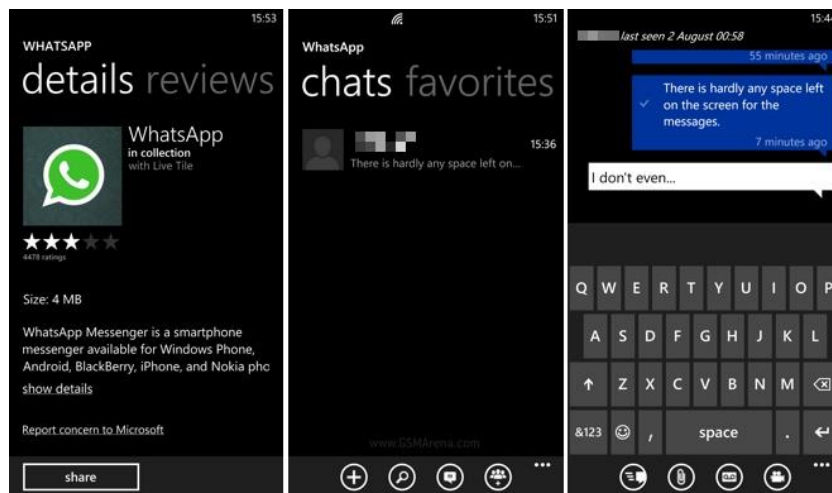


Figure 1 WhatsApp on Windows 8

# Project Report

---

## Positive Aspects:

- Simple User Interface
- Ability to send messages on internet
- Allows sending media files also.
- Cloud based backup

## Negative Aspects:

- Lack in graphics.
- Available on iPhone, Android, Symbian, Blackberry, Windows platform.
- Don't allow scheduling of messages.
- Crashing and sometimes doesn't respond to push notifications when running in background.

### 2.1.2. Viber

Viber [\[3\]](#) is a free application available on multiple platforms like Windows, Mac, and iOS, Android, Windows Phone, Blackberry, Nokia and Bada OS allowing free calls and messaging.

In Comparison to WhatsApp, Viber allows online free calling as well, but still Viber has many bungs in it, it crashes a lot and user have to open it to get notifications and to receive calls.



Figure 2 Viber on Windows Phone

# Project Report

## Positive Aspects:

- Allows free calls and messaging.
- Allows group creation.
- Popular application so large number of people using it.
- Free application

## Negative Aspects:

- Unstable.
- Availability in multiple platforms.
- Call feature have some quality problems
- Problems with push notifications.
- Doesn't allow scheduling of messages.

### 2.1.3. Go SMS pro

It is an android based free application placed on Google play store, provides convenient SMS/MMS experience, customization and understandable user interface [\[4\]](#).



Figure 3 Go SMS pro



# Project Report

---

## Positive Aspects:

- Simple and user friendly graphical user interface
- Ability to make groups.
- Allow management of SMS messages in folders.
- Capable of Scheduling and blocking.
- Themes, Facebook chat integration availability
- Cloud backup
- Private box.

## Negative Aspects:

- Only available on Android.
- Hangs sometimes
- Conversations deletion is an issue.
- Works only with SMS, not online free messages.

### 2.1.4. Group SMS Plus 2.0

Group SMS [\[5\]](#) plus Version 2.0 is paid application capable of grouping the SMS messages and allow user to send messages to multiple contacts in just one go.

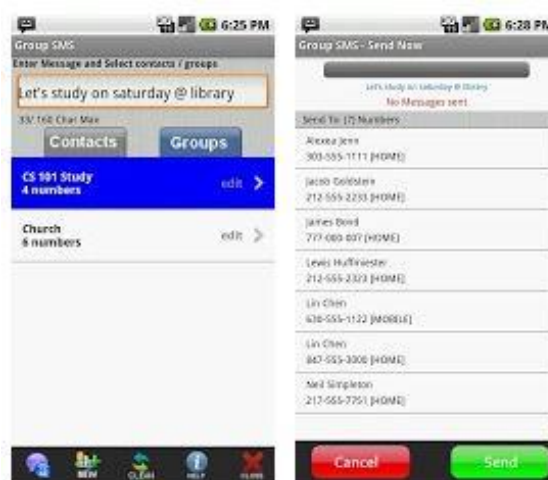


Figure 4 Group SMS Plus 2.0

# Project Report

## Positive Aspects:

- Can create and send SMS to group.
- Handle SMS sending to multiple contacts

## Negative Aspects:

- Available only on Android
- Pain Application
- Provides groups for SMS sending only.
- Crashes when browsing in the group created.
- User interface is confusing.
- Limited scope of development.

### 2.1.5. Group me

Very Successful and free application capable of messages group management, chatting and email id integration, it is easy to reach everyone, anytime, anywhere [\[6\]](#).



Figure 5 Group me

# Project Report

---

## Positive Aspects:

- For windows, android, iPhone and blackberry
- Create groups
- Group chat rooms.
- Facebook and email integration
- Allows sharing photos and location

## Negative Aspects:

- Response time sometime exceeds 10mins.
- Application crashes sometimes when sending sms multiple times.
- Limited scope, cater only one feature.

### 2.1.6. SMS Scheduler:

Windows Based free SMS Scheduler [\[7\]](#) application developed by Muhammad Naveed capable of saving and scheduling a SMS messages for future use and can be sent to any number of recipients anytime.

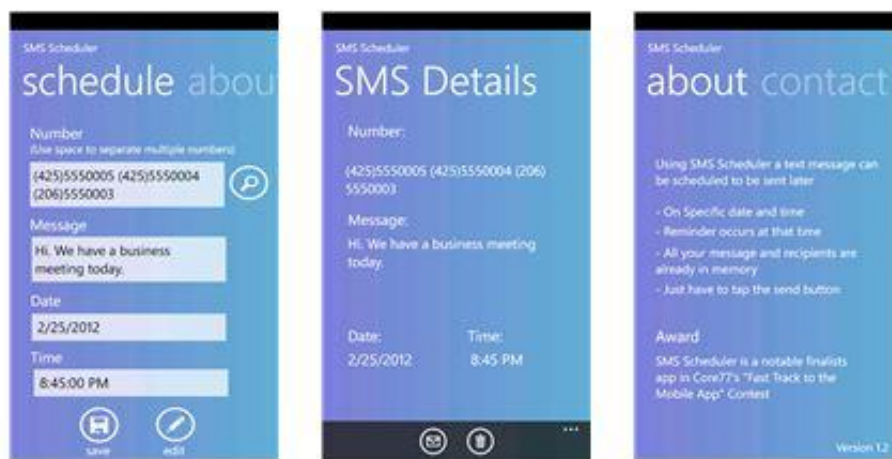


Figure 6 SMS Scheduler

# Project Report

---

## Positive Aspects:

- Automatic sending of SMS.
- Good User Experience
- Interactive and Easy
- View saved message (SMS)
- Date and time picker helps in input
- Multiple Recipients support
- Add Recipients from your phone memory
- Reminder occurs at scheduled time

## Negative Aspects:

- Limited Scope
- Only available on windows phone
- App crashes when choosing multiple contacts
- So many permissions required

### 2.1.7. Buddy Groups

Windows based Paid Application helps user to organize contacts and friends into groups. User can create groups for family members, friends and contacts. Application enables user to find buddies or members faster and user don't have to search in the mobile when sending SMS or calling to a contacts [\[10\]](#).

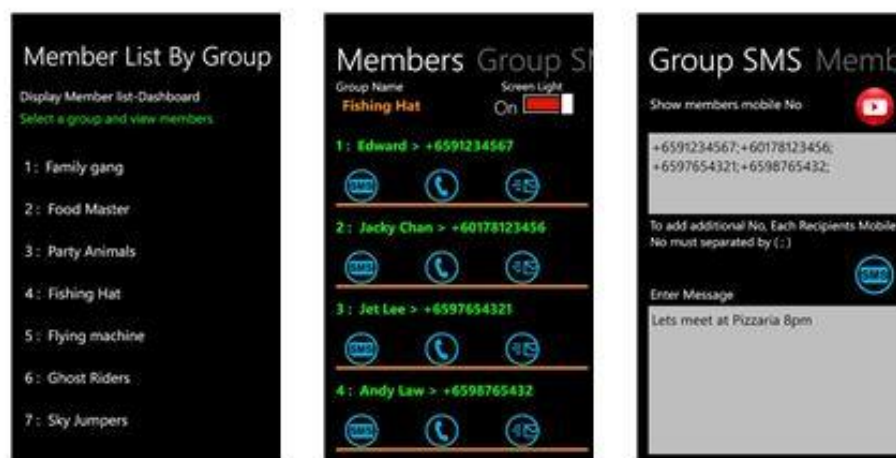


Figure 7 Buddy Groups

# Project Report

## Positive Aspects:

- Windows Based Application
- Organize contacts into groups by creating groups
- Send Group SMS with option to delete or add additional mobile no
- Create groups, add members, updating members information.
- Can store additional info of members.

## Negative Aspects:

- Less HCI Technique are followed.
- Color combination confuses the user.
- Hence confuses the user towards Interface.

### 2.1.8. Zune:

Windows based downloadable synchronization software able of making backup of photos, music, contacts and all other contents except messages [\[11\]](#).

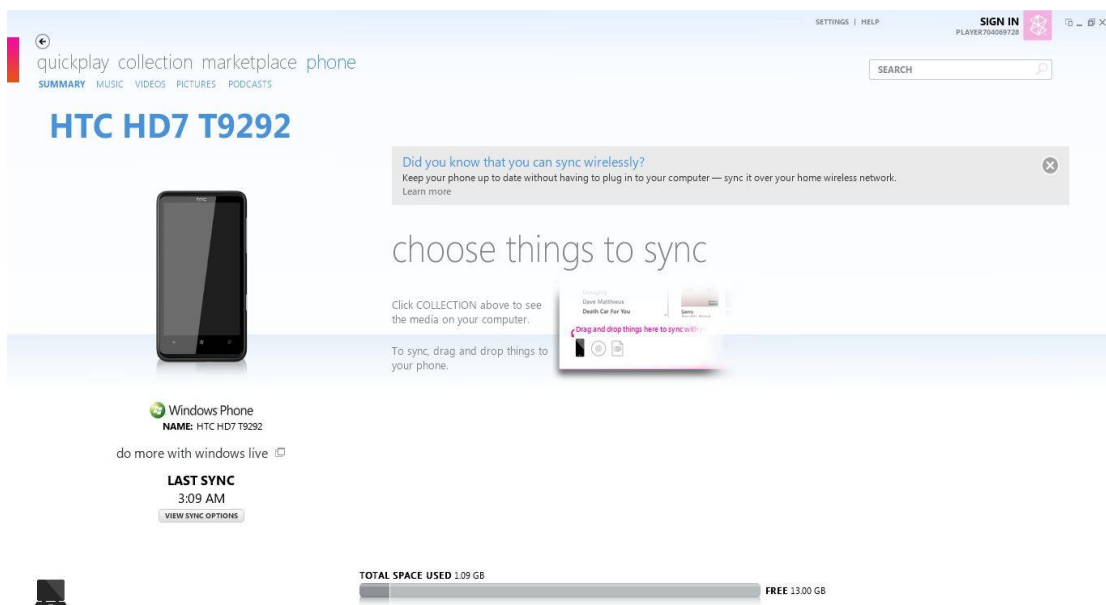


Figure 8 Zune Windows Based

# Project Report

---

## Positive Aspects:

- Easy to use.
- Automatically sync all contents and take backup
- A media organizer which helps organizing media content of Smartphone running windows.

## Negative Aspects:

- Doesn't allow taking backup manually
- Doesn't allow taking backup of Messages and contacts.

## 2.2. Comparison:

	Free Messages	Good User experience	Multiple Platforms	Backup/sync	Groups	Schedule	Folders	Search
WhatsApp	✓	✓	✓	✓	✓			
Viber	✓	✓	✓	✓	✓			
GoSMSPro		✓		✓	✓		✓	✓
Group SMS					✓			
Group me	✓	✓	✓	✓	✓			✓
SMS Scheduler	✓					✓		
Buddy Group		✓			✓			
Zune				✓				✓
Message Box	✓	✓		✓	✓	✓	✓	✓

Checking all the above mentioned applications, only one conclusion can be concluded, and that is, not application have all these features completely.

## **2.3. End Product Description:**

The end product comprises of our state of the art application which delivers grouping of contacts allowing user to group contacts and get messages from group in one place. Apart from that it allows user to schedule messages and specify time to send the specific message to any group or contact. The application also allows user to synchronize messages and other content of mobile with personal computer and other devices. The application also provides intelligent searching on content and contacts bases. The user can also organize messages in different folders as per his/her requirement. The final product combines beautiful GUI with powerful functionality, giving the user the ultimate messaging experience.

### 3. FUNCTIONALITY AND DESIGN

#### 3.1. Architecture

Message Box is based upon complete architecture called as MVC Architecture.

##### 3.1.1. MVC Architecture

Model View Control architecture is considered the standard now in modern application construction era, this architecture consists of three independent components i.e. Model, View, Controller. The beauty of the architecture is that all the components residing in the application are independent but works together by sharing information and interacting with each other to get all the necessary objectives and goals of the project. The main idea behind the MVC is to separate the presentation layer, business logic layer, and data model from each other, it also provides the loose bounded coupling between the components. Components details are as follows:

- **Model:**

Model is considered as essential design element in MVC as known as Data Model or Database Layer, it is placed at the bottom of the system, all the behavior of data is managed by the model, and it can also be placed on server besides the web services, in our application all the components are designed according to standards of Model.

- **View:**

This is called as presentation layer and front end of the applications; it's the components where the user interacts with the system. It is the interactive component of the system where user gives input and receives the output after necessary manipulations, the View its self can't be placed independently rather it need models and controller too.

- **Controller:**

Also known as Model-view, It is middle layer which contains all the business logic behind the system, this layer is linked with the view and models, the main objective behind the controller I to fetch the data from the model and present that data in presentable form on view and also synchronized the User Interface, it is responsible for interaction between model and view.



# Project Report

---

In addition to MVC, the Message Box application is the mixture of the MVC and Web services architecture, the web service and server architecture described in next coming section

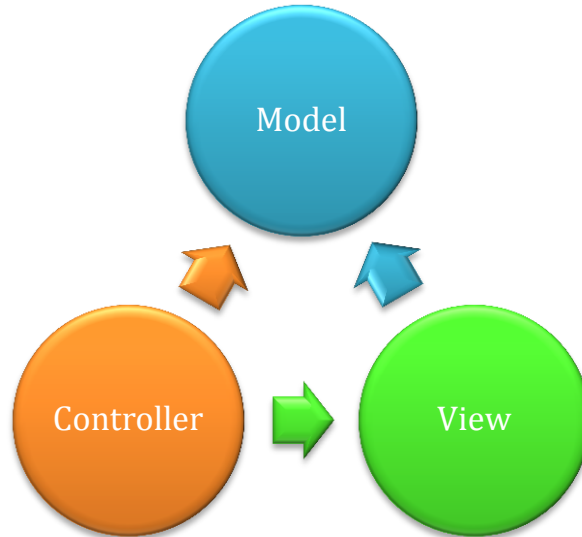
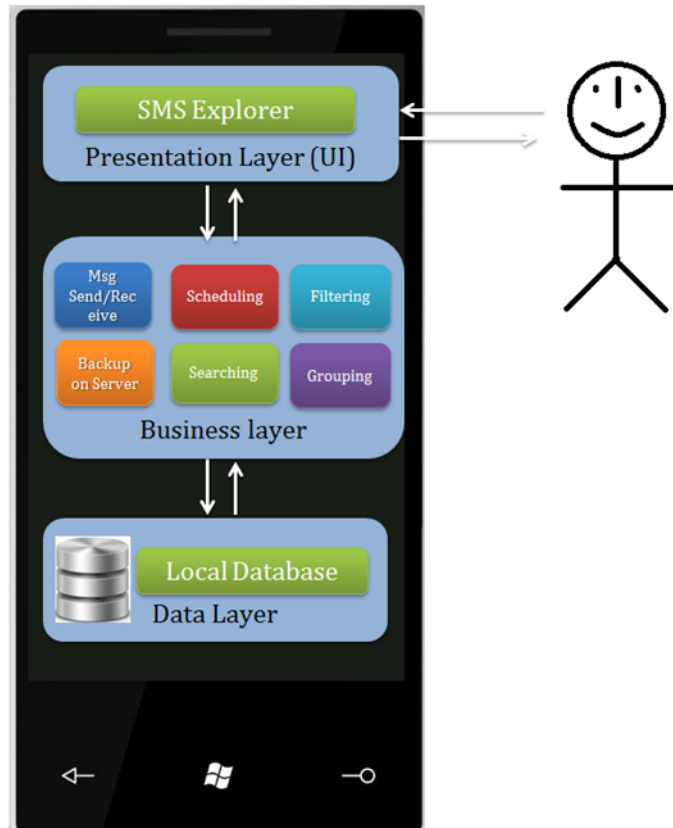


Figure 9: MVC

### 3.1.2. Client Side Architecture:

Client side architecture contains database as local database for offline transactions in the database, to fetch the respective message from the database, as database is consist of table names as group, contact, folder, message, and bridge tables to facilitate in the transaction of query. All the business logic resides in the business layer called as models, views used to show data retrieved from database, controllers to maintain smooth flow of application. Messages are first saves on the application database and then the web service that is hosted on the Windows Azure is invoked via web service and the data is send to the server. The web service architecture is discussed in the next section



**Figure 10: Client Side Architecture**

### **3.1.3. Server Architecture (as a whole)**

The Windows Azure provides a very rich framework for hosting web services and web sites. The application server module is hosted on Windows Azure cloud. The cloud allows user to connect to data server and access information. The message sending occurs, when the user send the message from smart phone, he sends the data to Windows Azure cloud via web service and goes into database on windows Azure server. This database figures out where the message is headed and it then checks the members list on server to fetch information about destination client. The message then is directed to the destination client and received on the destination's smart phone. This all process occurs when the client is already registered on our server.

# Project Report

If client is using the application first time, he sends his credentials to server where after authentication the server saves the credentials in members list. After that all the contacts of the person are checked against members list to check which of the contacts of the new members are already registered on the server. It forms the contact list for member and after that he is able to send and receive messages.

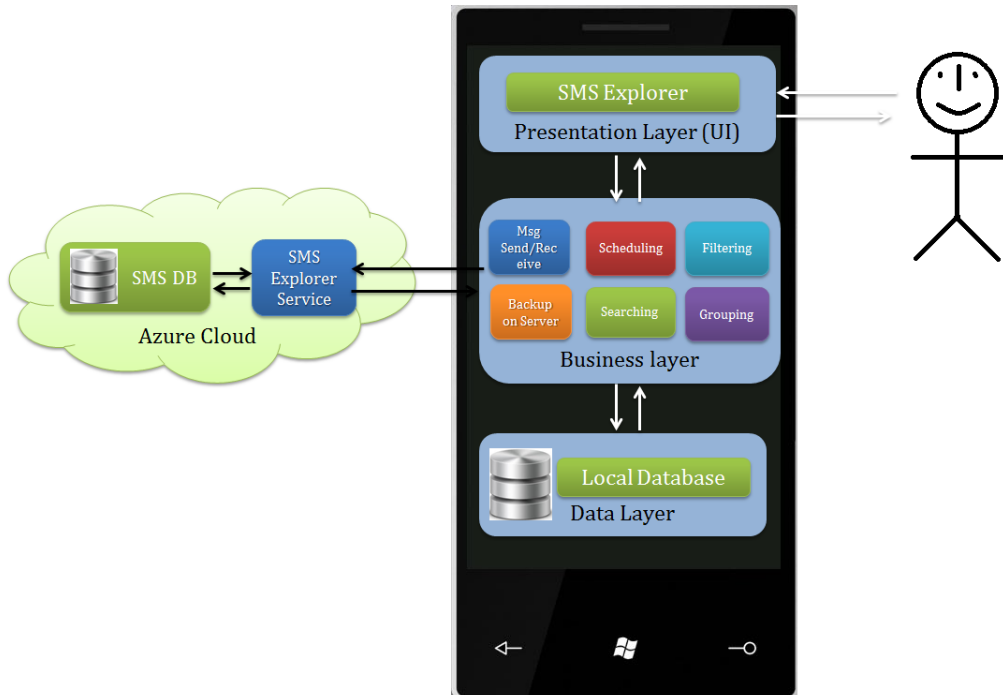


Figure 11: Server Architecture

# Project Report

The following Snap shows how Windows Azure cloud service works as a whole, The Facebook Integration is provided for login and authentication using Windows Azure's own Facebook's authentication module, which then contacts Facebook by using Facebook's API and gets a unique ID assigned when user gives correct initials of Facebook Login and password. Once logged in, the user remains logged in until he logged himself.

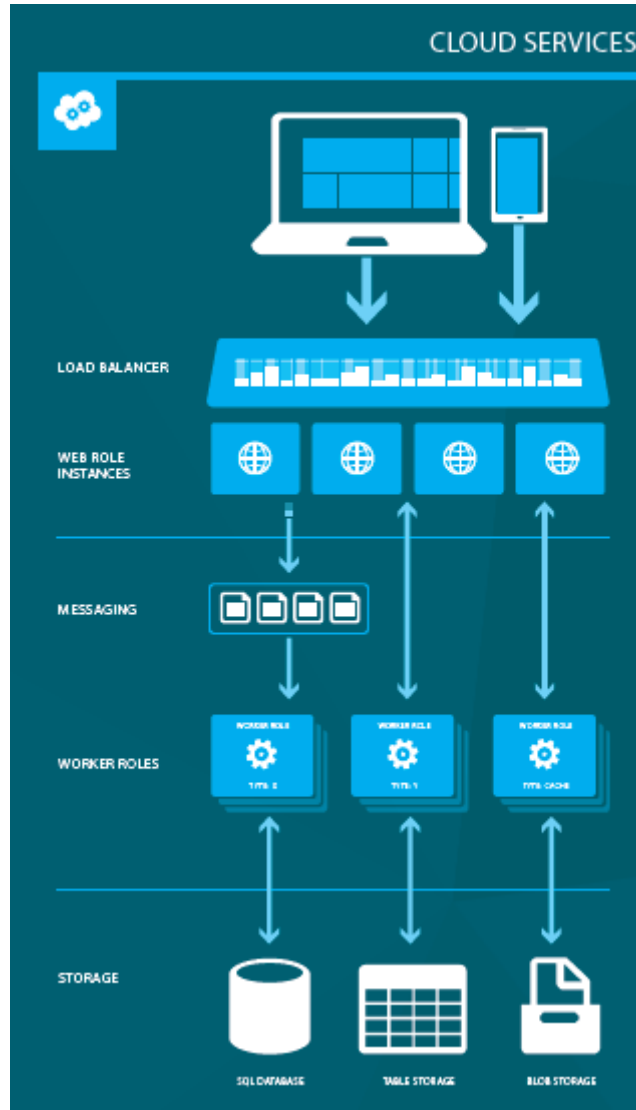


Figure 12 Windows Azure Architecture

# Project Report

## 3.1.4. Entity Relation Ship Diagram:

Given are the entity relationship diagrams of client and server side:

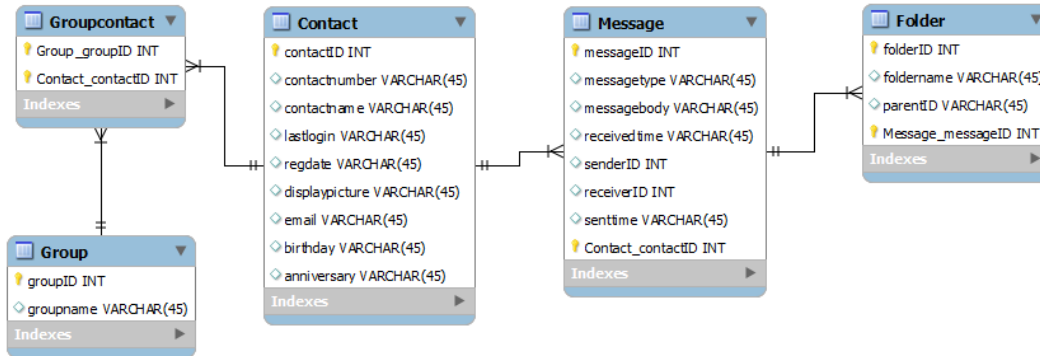


Figure 13: Entity Relationship Diagram

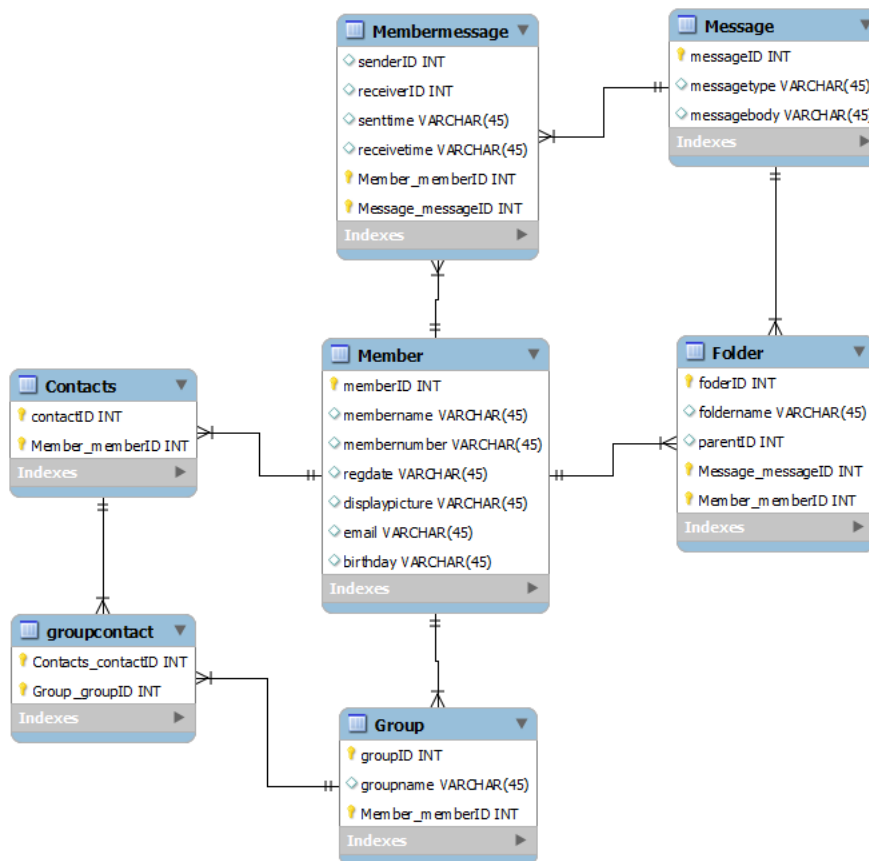


Figure 14 Entity Relationship Diagram

## 3.2. Modules:

### 3.2.1. Schedule:

Main purpose of this modules is to maintain a list of messages to be forwarded at specified time and date, for that reason Microsoft provides the scheduling API for scheduling the desired functionality with respect to time here in the module we have basic form type interface right now to enter information for message to be send. These messages are assigned the respective GUID by the system automatically. User selects the contacts using the Contact chooser from the window phone and initialized the text boxes with the information then select the time and date via date and time picker then schedules the respective message to be send on specified time defined by the user.

These message are listed in the list and when system time become equal to the specified time described by the user then window phone reminder invokes and tells the user to send the respective message as time become equal to the system time just like an alarm. The Windows phone has introduced security principle for which no application can send messages, without user interaction, This module allows user to schedule SMS messages for any specified time period, as a user can be busy in other activities so this module enables user to create a message and input the time for the message to be send to the recipient right away.

This modules works with the SMS and online messages, in case of online messages user interaction is not required and the message is send as soon as user's specified time becomes equal to the system time, in case of SMS, due to restrictions, the SMS cannot be sent without user's interaction, therefore this scheduler acts like alarm.

### 3.2.2. Grouping:

The main functionality of this modules is send the message in group just like send an email to numerous receivers, for that reason a local database is used having tables regarding storing the contacts and groups information's at this time we have three tables i.e. Contacts, Groups, and their relationship table between them. Contacts table have the information regarding all the contacts and group table have contacts group names and the respective contacts in the groups.

The relationship is based on the ID's assigned to the contacts and groups, grouping of SMS messages are made possible, User can make groups of contacts that allows user to send message to multiple contacts by just sending once. Also this module enables making relevant groups to communicate with the added members of the group efficiently and easily.

# Project Report

---

The grouping module contain three views, contact, group and bridge where the groups are made and contact are listed against any specific group to send SMS to contacts residing in the groups.

### **3.2.3. Folder Making:**

In the folder module the messages are arranged in folders. The beauty of this feature is that it allows multiple hierarchies of folders to allow organizing messages. This is achieved by messages having folder ID and parent ID saved in database. Folder ID represents the folder they belong to and parent ID represents the ID of the immediate parent folder of the folder in the question.

Here the parent id provides the necessary functionality for making the user define folder where we can go to any level of hierarchy. For instance let's have a look at how the records are saved. As for folder we have folder ID, folder Name, parent ID and message ID, initially the first record is empty parent ID as it is the main folder, Now when user wants to make new folder under the parent he/she simply adds the folder and specifies the parent ID, which is then use as folder ID of its parent folder.

For data retrieval we simply make a LINQ query, joining the table's message and folder table and in the where clause it has the parent id and the result are displayed in the list box.

### **3.2.4. Instant Messaging**

Instant messaging services are the future of new age messaging, with free and fast messaging over internet. The young generation is moving fast towards applications like WhatsApp and Viber. Such applications allow the user to send messages via internet in an efficient way. In windows mobile there are not many such applications and the applications which exist are not much user friendly. The instant messaging involves server and client architecture and use web services to connect to server and allow the user to send messages. The message after composition is sent to server where it is sorted out and forwarded to the destination client. For this to work, both the sender and receiver need to have some messaging application installed on their smartphones.

# Project Report

---

## **3.2.5. Synchronization and backup.**

The synchronization and backup is a major part of the application. The application is running server and client architecture, which allows us to synchronize messages and contacts on server and client. Everything being saved in server's database allows us to easily have a backup and later synchronize the client's and server's databases that allows us to be robust and minimize the information loss.

## **3.3. Deliverables:**

Our project is divided into multiple milestone, each milestone with partially completed deliverables. At the end of each milestone some of the modules would be completed and deliverables would be delivered accordingly.

The main milestones for our project are as follows:

### **Milestone one**

- Abstract
- User Stories
- Class Diagram
- ER Diagram

### **Milestone two**

- Block Diagram
- Flow Chart
- GUI Mock-ups
- Grouping Module

### **Milestone three**

- GUI Design
- Scheduling module

### **Milestone four**

- GUI Design (Revised)
- Categorization Module



## **Milestone five**

- GUI Design (revised)
- Searching Module

## **Milestone six**

- GUI Design (revised)
- Instant Messaging

## **Milestone seven**

- GUI Design (Finalized)
- Integration
- Testing

### **3.4. Schedule of Deliverables (Tentative):**

- Milestone one – (5 December, 2012)
- Milestone two – (2 January, 2013)
- Milestone three – (30 January, 2013)
- Milestone four – (20 February, 2013)
- Milestone five – (13 March, 2013)
- Milestone six – (3 April, 2013)
- Milestone seven – (1 May, 2013)

### 4. PROJECT IMPLEMENTATION

#### 4.1. Contact Grouping Module:

The main functionality of this modules is send the message in group just like send an email to numerous receivers, for that reason a local database is used having tables regarding storing the contacts and groups information's at this time we have three tables i.e. Contacts, Groups, and their relationship table between them. Contacts table have the information regarding all the contacts and group table have contacts group names and the respective contacts in the groups.

The relationship is be based on the ID's assigned to the contacts and groups, grouping of sms messages are possible, User can make groups of contacts that allows user to send message to multiple contacts by just sending once. Also this module enables to make relevant groups to communicate with the added members of the group efficiently and easily.

The grouping module contain three views, contact, group and bridge where the groups are made and contact are listed against any specific group to send message to contacts residing in the groups. The snapshots for the views are as follows:

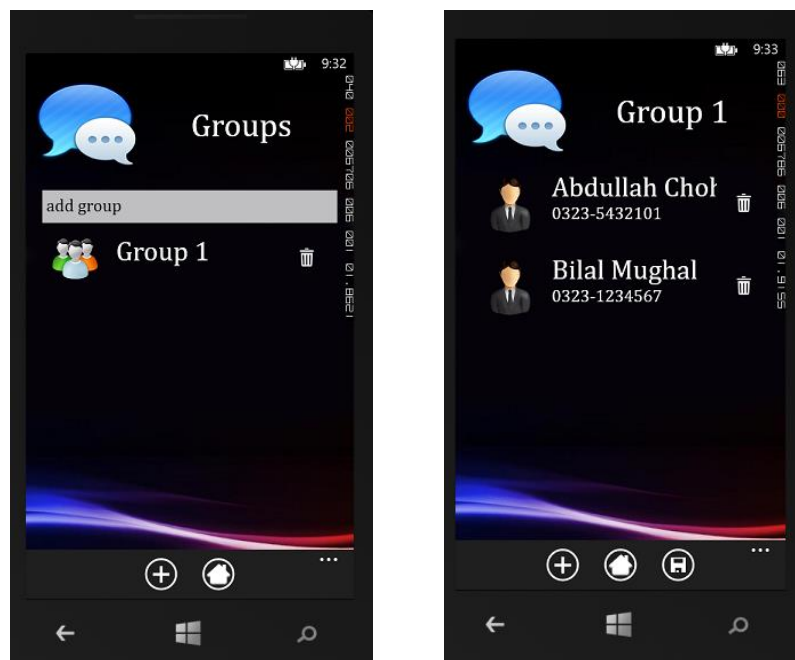


Figure 15 Contact Grouping Module

# Project Report

The main views of grouping module have, UI elements as list boxes, text boxes, buttons, the lists of groups and contacts that lies in specific group are placed in the observable collection, where the lists is gathered via LINQ to SQL services and queries

The gathering of data from database is as follows:

```
var toDoItemsInDB = from GroupsTable groups in toDoDB.Groups
                    join GroupContact groupcontact in toDoDB.GroupContacts on groups.GroupID equals groupcontact.GroupID
                    join ContactsTable acontact in toDoDB.Contacts on groupcontact.ContactID equals acontact.ContactID
                    where groupcontact.GroupID == (Int32.Parse(parameterGID))
                    select acontact;

ContactsList = new ObservableCollection<ContactsTable>(toDoItemsInDB);

App.contactsList = ContactsList;
NotifyPropertyChanged("ContactsList");

// Call the base method.
base.OnNavigatedTo(e);
App.contactsListBoxcount = contactshowListBox.Items.Count;
```

These two main views holds the data for groups and contacts as shown in figure 15 and furthermore these groups are used to send messages and SMS to groups, the contacts are gathered from particular groups and used in receiver field in scheduling the SMS etc.

The activity diagram for the above module is as follows:

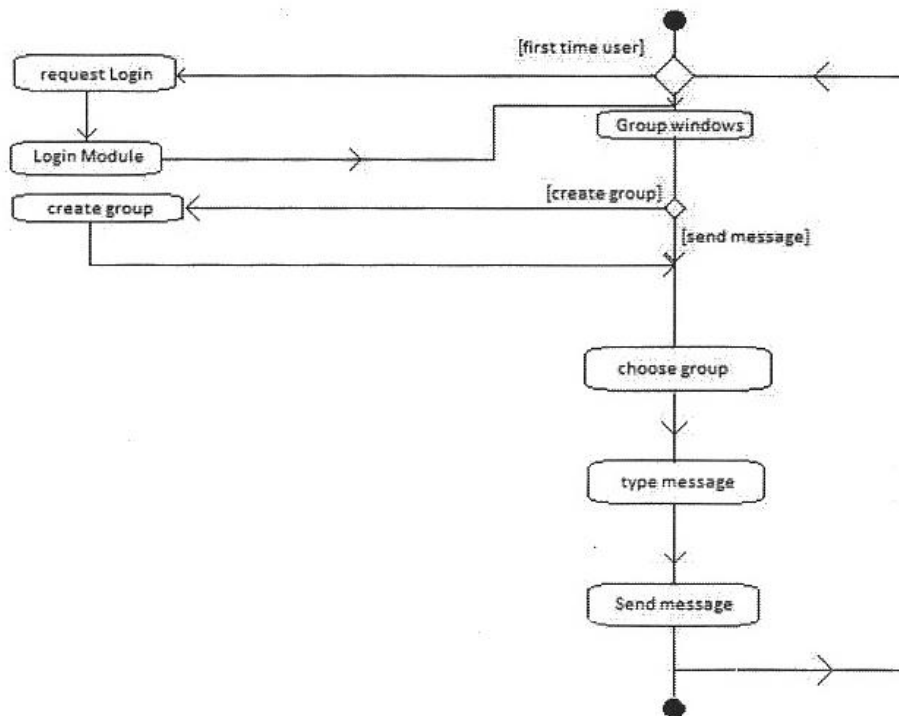


Figure 16: Activity Diagram Grouping

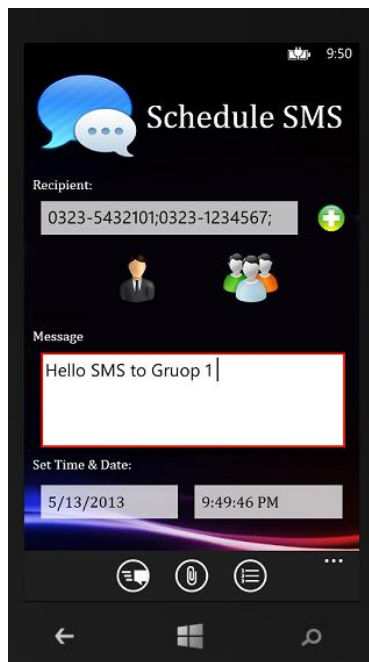
# Project Report

---

## 4.2. Scheduling Module:

Purpose of this modules is to maintain a list of messages to be forwarded at specified time and date, for that reason Microsoft provides the scheduling API for scheduling the desired functionality with respect to time here in the module we have basic form type interface right now to enter information for message to be send. These messages are assigned the respective GUID by the system automatically. User selects the contacts using the Contact chooser from the window phone and initialized the text boxes with the information then select the time and date via date and time picker then schedules the respective message to be send on specified time defined by the user.

The Scheduling View is as follows:



**Figure 17 Scheduling Module**

Here in the above snap it is quite clear that contacts are gathered from the group and used as receivers, the one who receives SMS when user press send button

These message are listed in the list and when system time become equal to the specified time described by the user then window phone reminder invokes and tells the user to send the respective message as time become equal to the system time just like an alarm. The snap shot of the scheduling list views is as follows:

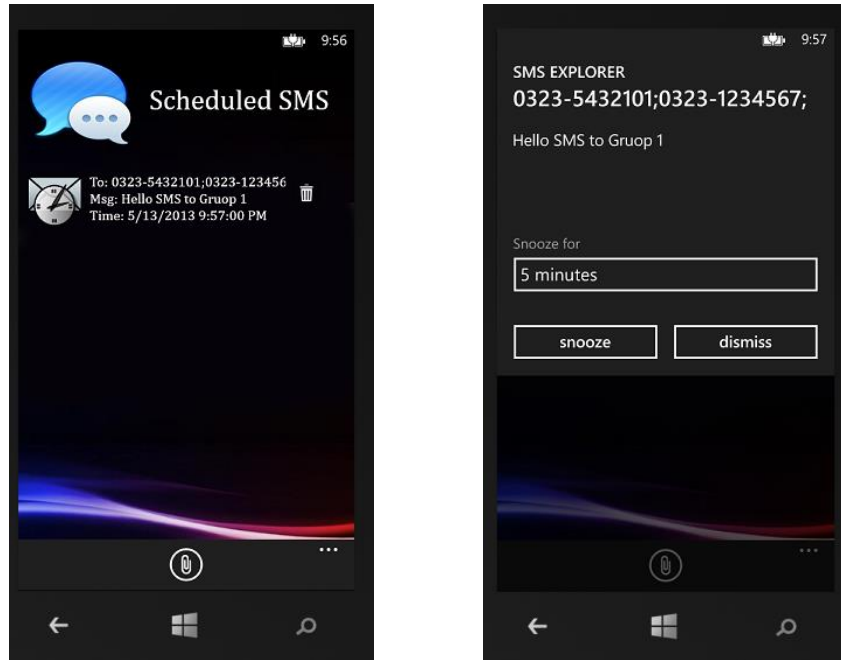


Figure 18 Scheduling Module

When a specific SMS is scheduled, three parameters are passed to the scheduling page via navigation properties, the parameters are receivers, message body and sending time.

This module allows user to schedule SMS messages for any specified time period, as a user can be busy in other activities so this module enables user to create a message and input the time for the message to be send to the recipient right away.

The Activity Diagram is as follows:

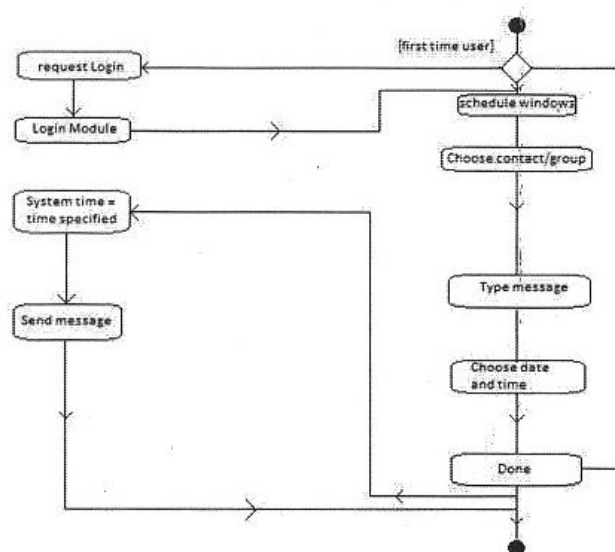


Figure 19: Activity Diagram Scheduling

## 4.3. Inbox & Conversations Module:

We have created a inbox view just like the native application of windows phone to facilitate the instant messaging service feature of our product, the main idea behind the inbox module is to witness the sending and receiving of messages round the clock, for this module the local database has played a big role in fetching the required data against the respective receiver and results are stored in the list boxed and the messages are made quite pleasant able view so that user not gets bored of the application and use the app frequently. The views are as follows:



Figure 20 Inbox & Conversations Module

Results are fetched from the database using the contact id and messages id to make a good looking UI and to show only the required messages that were send to the respective receiver. Furthermore there is another view to select the messages and we can enter these messages in the folder, this view discussed in the folder module, in the inbox module we can forward or delete the respective message using the context menu, also we can delete the entire conversation against the specific receiver, so the view are as follows:

# Project Report

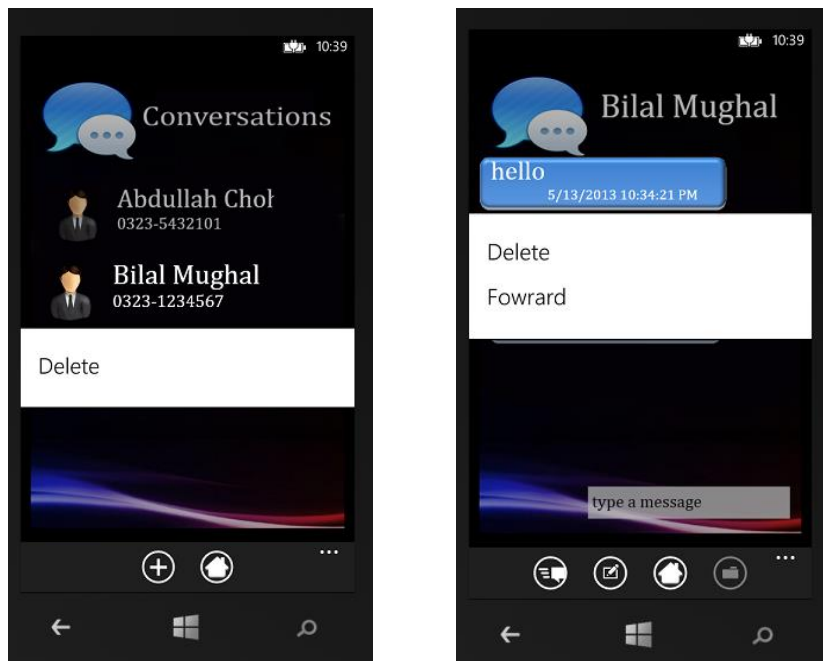


Figure 21 Inbox & Conversations Module

Code to delete the particular conversation/specific message from the inbox is as follows; here we have used the context menu the UI toll from UI toolkit. Table object is casted from data context to be used with the menu item of context menu.

```
ContactTableMsg contact = (sender as MenuItem).DataContext as ContactTableMsg;
var remove = (from ContactTableMsg onecontact in toDoDB.contactmsg
              where onecontact.ContactID == contact.ContactID
              select onecontact).FirstOrDefault();

if (remove != null)
{
    ContactsListMsg.Remove(remove);
    toDoDB.contactmsg.DeleteOnSubmit(remove);
}
toDoDB.SubmitChanges();

// Put the focus back to the main page.
this.Focus();
```

Code to forward the specific message to the desired receiver is as follows, this is achieved by using the same context menu the UI toolkit tool. Table object is casted from data context to be used with the menu item of context menu. Then values are fetched from the item of the list box and then the values are passed to the “compose new view” and used to send message to that receiver whose ID and number are received via navigation service.

# Project Report

```
private void ForwardMessage_click(object sender, RoutedEventArgs e)
{
    MessageTable message = (sender as MenuItem).DataContext as MessageTable;
    string parameter1Value = message.MessageID.ToString();
    string parameter2Value = message.MessageBody;

    NavigationService.Navigate(
        new Uri(string.Format("/ViewsDB/CreateNewSms.xaml?myparameter1={0}&myparameter2={1}",
            parameter1Value, parameter2Value),
            UriKind.Relative));
}
}
```

The activity diagram explaining the working of conversation windows is given below:

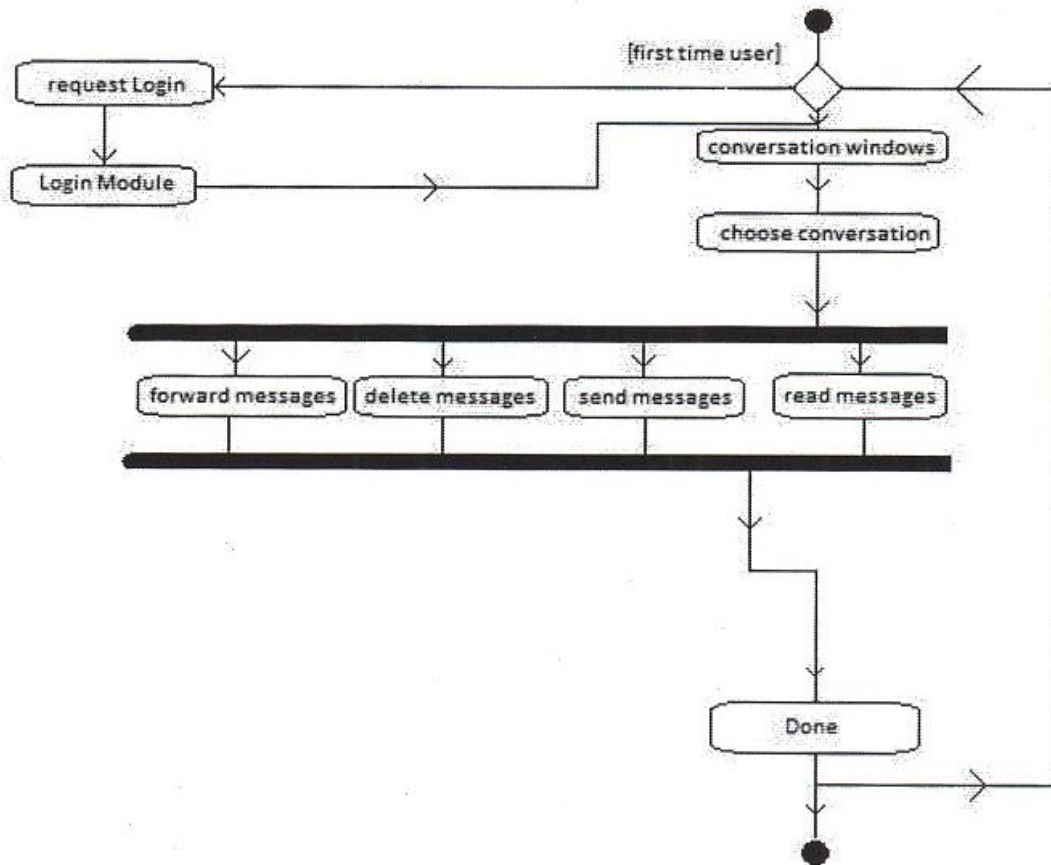


Figure 22: Activity Diagram Conversation

## 4.4. Contacts Adding Module:

This module has a massive impact on the project as this module gets the contact's number and name from the native contacts of the phone and then these are used in the database to keep record of all the messages that are conveyed to the receivers. Contact adding and deleting are followed by some checks and user confirmation is required to facilitate the user.



# Project Report

Contacts adding module has the following view:

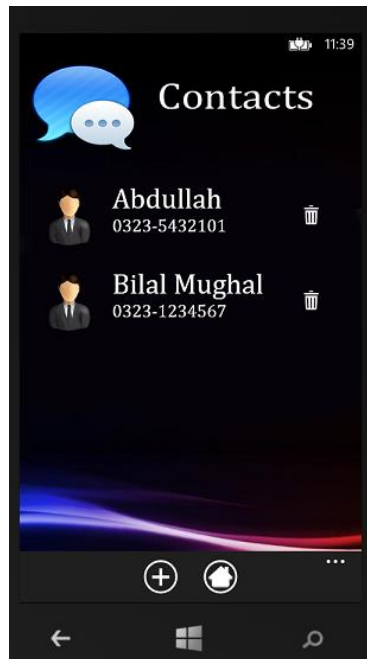


Figure 23 Contacts Adding Module

To show the contacts, Observable collection is used, which works with the contactsListMsg to show and add contacts via our application. It allows the user to gather contact from phone, detection by user confirmation and contacts retrieval.

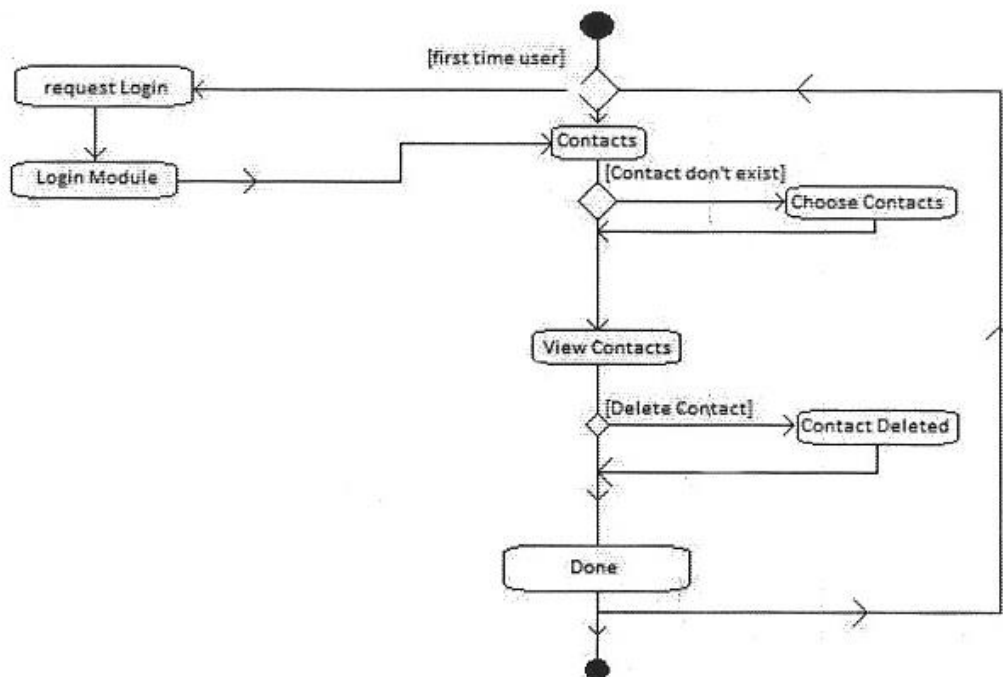


Figure 24: Activity Diagram Contacts

# Project Report

---

## 4.5. Foldering Module:

In the Foldering module all the messages are stored in the database table called as folder and having some attribute i.e. folder ID, folder name, parent ID, Message ID from Message table, the message are stored with the help of message id and folder id with some little essence of parent id and then for retrieval all the messages are gathered against a specified parent id with join with message and folder, with respect to any specified contact.

Whole database is developed and implemented to help this module where the entire tables are with interaction with each other. Folder views are as follows; three views, one to make folder, second to view messages inside the folder. And third to select the messages from conversations and add that to the respective folder, let's have a look

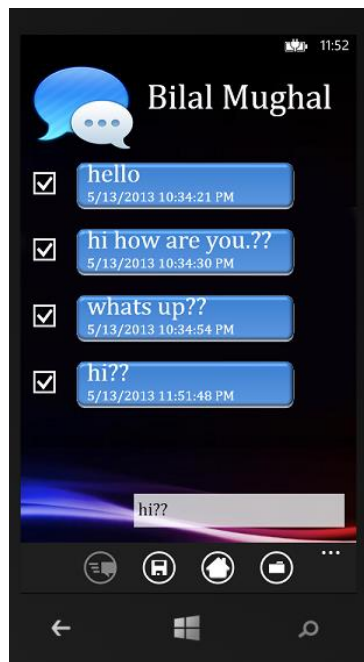


Figure 25 Foldering Module

Here the parent id provides the necessary functionality for making the user define folder where we can go to any level in folder making, for instance let's have a look at records entrance as for folder we have folderID, folderName, parentID and messageID,



**Figure 26 Foldering Module**

Initially the first record is empty parent as it is the main folder, now when user wants to make new folder under the parent he/she simply adds the folder it of parent in the parent id and then so on going in the hierarchy. To view the messages in the folder lets have look at the following snapshot:



**Figure 27 Foldering Module**

# Project Report

For data retrieval we simply make a LINQ query, joining the table's message and folder table and in the where clause it has the parent id and the result displays in the list box. Let's have look at the code behind the view:

The codes snippets for the folder modules are as follows: these include, adding the folder, deleting, adding messages to the folder, and then fetching messages from the particular folder, so:

```
public void AddMessage(string fname, int fid, int pid)
{
    // Create a new to-do item based on the text box.
    FolderTableMsg afolder = new FolderTableMsg
    {
        FolderID = fid,
        FolderName = fname,
        MessageID = 1,
        ParentID = pid
    };
};
```

The following diagram shows the activity diagram of Foldering module:

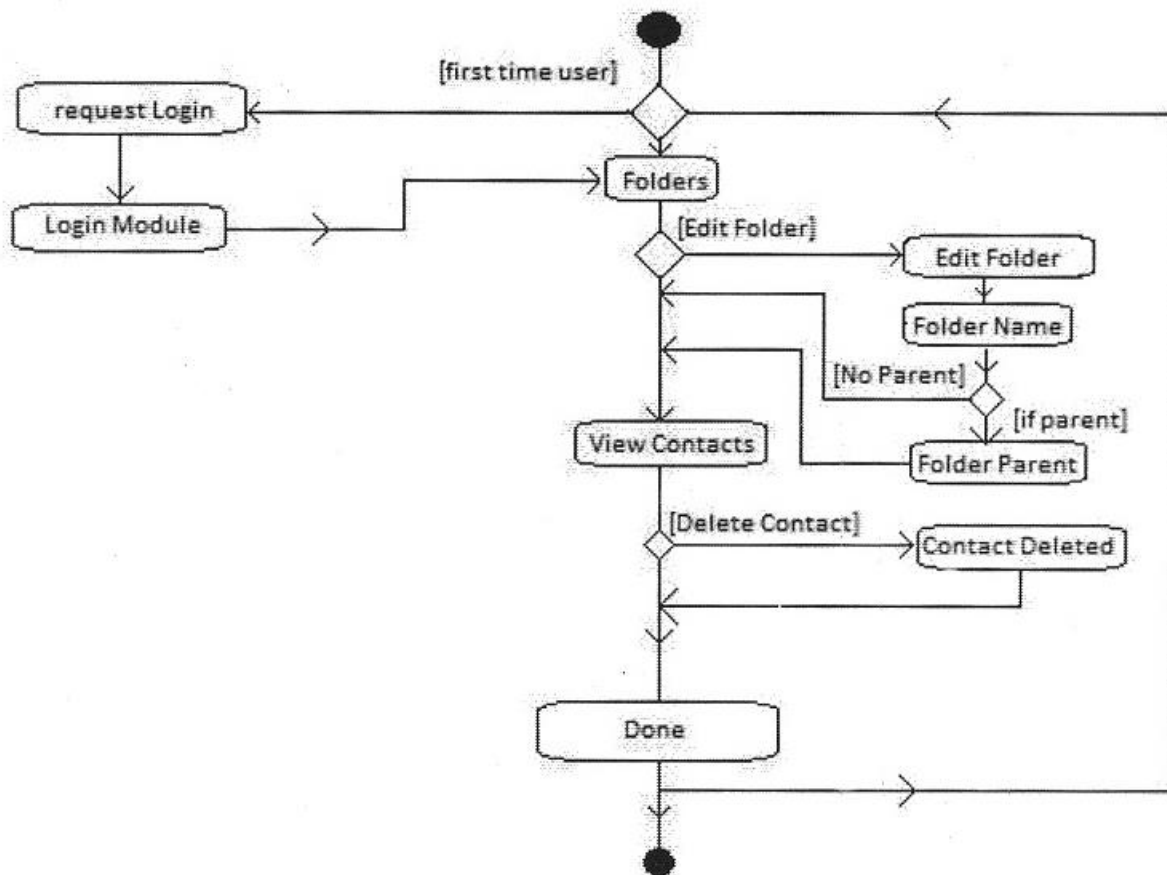


Figure 28: Activity Diagram Folders

# Project Report

---

## 4.6. Instant Messaging

Instant messaging services are the future of new age messaging, with free and fast messaging over internet. The young generation is moving fast towards applications like WhatsApp and Viber. Such applications allow the user to send messages via internet in an efficient way. Due to the evolution of service providers and GSM into 3G and 4G, the smartphones can access fast internet readily anywhere. In windows mobile there are not many applications allowing you to send instant messages and the applications which exist are not much user friendly and lack stability. The instant messaging involves server and client architecture and use web services to connect to server and allow the user to send messages. The message after composition is sent to server where it is sorted out and forwarded to the destination client.

The Windows Azure cloud environment is used to manage all the work load of the messaging over the internet. The cloud manages all the data which is not yet received on cell. The data which is received on cell is saved locally on cell. The system is designed to take on the entire load yet keep all operations robust and efficient.

The code below shows the code which manages all the data in our database allowing the successful sending and receiving of messages.

```
class Message
{
    private int messageId { get; set; }
    private String messageBody { get; set; }
    private String messageTime { get; set; }

    public Message()
    {
    }

    public Message(int messageId, String messageBody, String messageTime)
    {
        this.messageId = messageId;
        this.messageBody = messageBody;
        this.messageTime = messageTime;
    }
}
```

# Project Report

---

```
class User
{
    private int userId { get; set; }
    private String userName { get; set; }
    private String contactNumber { get; set; }
    private String joinDate { get; set; }
    private String lastLogin { get; set; }
    private List<ReceivedMessage> receivedMessages { get; set; }
    private List<Contact> contacts { get; set; }

    public User()
    {
    }

    public User(int userId, String userName, String contactNumber, String joinDate, String lastLogin,
        List<ReceivedMessage> receivedMessages, List<Contact> contacts)
    {
        this.userId = userId;
        this.userName = userName;
        this.contactNumber = contactNumber;
        this.joinDate = joinDate;
        this.lastLogin = lastLogin;
        this.receivedMessages = receivedMessages;
        this.contacts = contacts;
    }
}
```

```
class Contact
{
    private int contactId { get; set; }
    private String contactName { get; set; }
    private String contactNumber { get; set; }
    private String joinDate { get; set; }
    private String lastLogin { get; set; }
    private List<Message> messages { get; set; }

    public Contact(int contactId, String contactName, String contactNumber, String joinDate, String lastLogin,
        List<Message> messages)
    {
        this.contactId = contactId;
        this.contactName = contactName;
        this.contactNumber = contactNumber;
        this.joinDate = joinDate;
        this.lastLogin = lastLogin;
        this.messages = messages;
    }
}
```

```
class ReceivedMessage : Message
{
    private Contact sender { get; set; }

    public ReceivedMessage()
    {
    }

    public ReceivedMessage(int messageId, String messageBody, String messageTime, Contact sender)
        : base(messageId, messageBody, messageTime)
    {
        this.sender = sender;
    }
}
```

# Project Report

## 4.7. Searching Module:

To facilitate the user an searching mechanism is introduced in the application, this searching is faced on messages that send to the receivers on the other side, local database is used to accomplished this module, in the module some of the UI components of the windows SDK are used i.e. List box, Observable collection, autocomplete box and radio button, user can search the messages on the basis of contact number, name, message body and off course by folder name.

The basic view of the searching modules is as follows:



Figure 29 Searching Module

The necessary code behind the searching modules is as follows:

```
private ObservableCollection<MessageTable> messageListMsg;
public ObservableCollection<MessageTable> MessageListMsg
{
    get
    {
        return messageListMsg;
    }
    set
    {
        if (messageListMsg != value)
        {
            messageListMsg = value;
            NotifyPropertyChanged("MessageListMsg");
        }
    }
}
```

# Project Report

The following diagram shows the activity diagram of searching module:

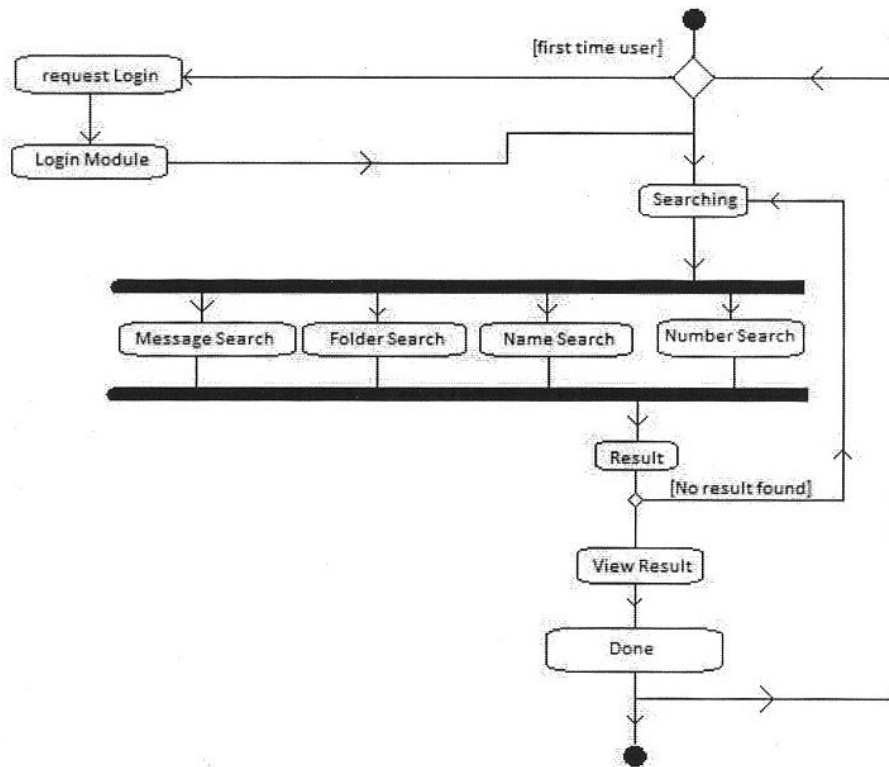


Figure 30: Activity Diagram Searching

## 4.8. Use Case Diagram:

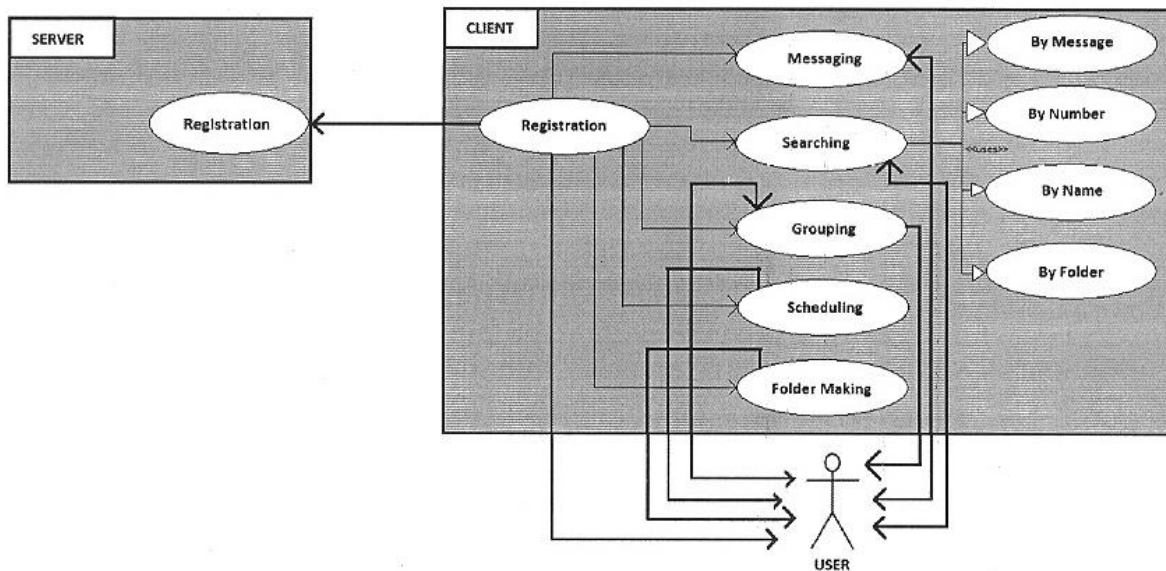


Figure 31: Use Case Diagram



## 5. RECOMMENDATIONS AND CONCLUSIONS

The project comes with a complete integrated product, all the proof of concepts are full filled regarding contact grouping , Scheduling the messages, Searching the messages, Categorization of messages, finally this product uploads to windows market and ready to be downloaded and allow user to enjoy the unique set of features for SMS. User is able to manage contents in better way and allow user to forwarding. Complete GUI design in implemented and can also be improvised in future.1

In recommendations can we can add voice calling feature in the system where one user can call other over the network, Medias can be shared also as pictures, sound files and also files transferring can be added, User interface can be improvised to facilitate the user. Other functionalities like customize the inbox, draft feature can be implemented also.

## 6. REFERENCES

- [1]. [Top 10 Strategic Technology Trends for 2013](#), [online], (Accessed: October 2012)
- [2]. [Go SMS pro](#), [online], (Accessed: October 2012)
- [3]. [WhatsApp](#), [online], (Accessed: October 2012)
- [4]. [Viber](#), [online], (Accessed: October 2012)
- [5]. [Group SMS](#), [online], (Accessed: October 2012)
- [6]. [Group me](#), [online], (Accessed: October 2012)
- [7]. [SMS Scheduler](#), [online], (Accessed: October 2012)