**Thesis Guidelines**

**Font Style**: Times New Roman

Main Heading (Chapter 01: Name) : 16 size

Sub Heading 1 : 14 size

Sub Heading 2: 13 size

Sub Heading 3: 12 size

**Text Size**: 12

**Line Spacing**: 1.5

**Table Caption**

Position: Top

Size: 10

**Figure Caption**

Position: Bottom

Size: 10

**Reference Style** : IEEE Format.

**Paper Margin:**

Top = 1.1” Right = 0.8” Bottom = 0.8” Left = 1”

Footer = 0.2” Gutter = 0.3” Gutter Position = Left

**Pagination**

* The **Title Page** is counted as page i (assumed, do not print number)
* The **Approval Page** is counted as page ii (assumed, do not print number)
* The remaining preliminary pages are numbered with **lower case Roman numerals** (iii, iv, v, vi, etc). After the title page and approval page, begin numbering the next page with iii.
* The **main body** of the text and the reference section are consecutively numbered with Arabic numerals beginning **page “1”** and continuing throughout, including text, illustrative materials, bibliography, and appendices.

**Binding and Submission**

The original and two good quality photocopies are to be submitted to the university. Soft bindings are preferable for initial submission and evaluation by the GEC.

Hard bound copies should be submitted after incorporating the suggested changes and corrections.

**Title**

There are two title pages: Margins for the title pages should be atleast 1.5” for right, left, top and bottom respectively.

* The Binding Title
* Inner Title

The **binding title** should contain the full title of the thesis, the full name and registration number of the candidate, AU monogram, the name of the department, college/institute and university, followed by the year of graduation printed in golden.

The **Inner title** is the first printed page in the thesis (leave one blank page in the beginning).

**Color**

In order to keep the theses well organized and uniform, the covers/bindings of theses should have the following colors:

BS Navy Blue

Text on the binding should be in golden color.

**Spine**

The spine of the bound thesis should bear the students’ name, name of department and year of graduation in golden color.

# Dedications

# Acknowledgement

Table of Contents

[Dedications 2](#_Toc24467411)

[Acknowledgement 2](#_Toc24467412)

[List of Tables 5](#_Toc24467413)

[List of Figures 6](#_Toc24467414)

[Abstract 9](#_Toc24467415)

[Chapter 1: Introduction 2](#_Toc24467416)

[1.1 Background 2](#_Toc24467417)

[1.2 Objectives 3](#_Toc24467418)

[1.3 Scope 3](#_Toc24467419)

[1.4 Problem Statement 3](#_Toc24467420)

[Summary 4](#_Toc24467421)

[Chapter 2: Literature Review 6](#_Toc24467422)

[Chapter 3: Planning and Methodology 8](#_Toc24467423)

[3.1 Project Deliverables 8](#_Toc24467424)

[3.2 Work Break down Structure 8](#_Toc24467425)

[3.4 Process Model 10](#_Toc24467426)

[Summary 10](#_Toc24467427)

[Chapter 4: System Specification 12](#_Toc24467428)

[4.1 Business Requirements 12](#_Toc24467429)

[4.1.1 User Requirements 12](#_Toc24467430)

[4.2 Process Flow 13](#_Toc24467431)

[4.2 Functional Requirements 13](#_Toc24467432)

[4.2.1 FR01: User Login 13](#_Toc24467433)

[4.3 Non-Functional Requirements 14](#_Toc24467434)

[4.3.1 NFR01: Performance 14](#_Toc24467435)

[4.4 Assumptions and Constraints 15](#_Toc24467436)

[4.4.1 Development Languages and Tools 15](#_Toc24467437)

[4.4.2 Operating System 15](#_Toc24467438)

[4.5 Actors 15](#_Toc24467439)

[4.6 Use-Cases 15](#_Toc24467440)

[4.6.1 UC01- Log into the System 15](#_Toc24467441)

[4.7 Use Case Modeling 17](#_Toc24467442)

[4.7.1 Complete System Diagram 17](#_Toc24467443)

[4.8 Traceability Matrix 18](#_Toc24467444)

[4.9 Behavioral Model 18](#_Toc24467445)

[4.9.1.1 Use Case 01- Login 18](#_Toc24467446)

[4.10 Class Diagram 19](#_Toc24467447)

[4.11 Graphical User Interface (GUI) 20](#_Toc24467448)

[4.11.1 GUI 01- Login 20](#_Toc24467449)

[4.11.2 GUI 04- Sign Up 21](#_Toc24467450)

[Summary 22](#_Toc24467451)

[Chapter 5: System Design 24](#_Toc24467452)

[5.1 Design Goals 24](#_Toc24467453)

[5.2 System Architecture Diagram 25](#_Toc24467454)

[5.3 Data Model 26](#_Toc24467455)

[5.3.1 Context Level 26](#_Toc24467456)

[5.3.2 Level 1 DFD 26](#_Toc24467457)

[5.4 ER-Diagram 27](#_Toc24467458)

[5.5 Data Dictionary 27](#_Toc24467459)

[5.5.1 Table Name (e.g User) 27](#_Toc24467460)

[5.5.2 Personal Info 27](#_Toc24467461)

[5.6 Structural Model 28](#_Toc24467462)

[5.6.1 Structural Model of the System 28](#_Toc24467463)

[5.7 Deployment Diagram 29](#_Toc24467464)

[Chapter 6: Coding 32](#_Toc24467465)

[6.1 Android Side Development 32](#_Toc24467466)

[6.2 Web API on Server Side 32](#_Toc24467467)

[Chapter 7: Software Testing 34](#_Toc24467468)

[7.1 Test Case 01: LOGIN 34](#_Toc24467469)

[7.2 Test Case 02: Sign Up 36](#_Toc24467470)

[Chapter 8: Conclusion 39](#_Toc24467471)

[User Guide 40](#_Toc24467472)

[Reference 42](#_Toc24467473)

[Glossary 43](#_Toc24467474)

# List of Tables

[Table 2. 1 Schedule Table 9](#_bookmark15)

[Table 3. 1 Login Traceability Matrix 31](#_bookmark90)

[Table 3. 2 Campaign Traceability Matrix 34](#_bookmark92)

[Table 3. 3 Delete user/Group Traceability Matrix 34](#_bookmark94)

[Table 3. 4 Create Group Traceability Matrix 35](#_bookmark97)

[Table 3. 5 Sign up Traceability Matrix 36](#_bookmark98)

# List of Figures

Figure 2. 1 Work Breakdown Structure 11

Figure 2. 2 Work Plan 11

Figure 2. 3 Process Model 13

Figure 3. 1 Process Flow Diagram 18

Figure 3. 2 Complete System Diagram 27

Figure 3. 3 Login Use Case 28

# Abstract

**CHAPTER 1**

**INTRODUCTION**

# Chapter 1: Introduction

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

## Background

*<The project background should include information about the reasoning why you want to implement this specific project in this specific location in this specific manner. It has to explain the current situation and its problems and the way in which you want to solve these problems. These explanations and assumptions should be backed up by reliable data.>*

## Objectives

*<A project objective describes the desired results of a project, which often includes a tangible item. An objective is specific and measurable, and must meet time, budget, and quality constraints.>*

## Scope

*<A project scope, or project scope statement, is a tool used to describe the major deliverables of a project including the key milestones, high level requirements, assumptions, and constraints. The project scope statement is a useful tool for future decision making when new change requests are considered to modify the project scope. It also defines the boundaries of a given project and clarifies what deliverables are in and out of scope.>.*

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

## Problem Statement

*<Briefly addresses the question: What is the problem that the research will address?>*

## Summary

<*Give a short description of the chapter.>*

**Chapter 2**

**Literature Review**

# Chapter 2: Literature Review

*<The literature review should include the following:*

* *Objective of the literature review*
* *Overview of the subject under consideration.*
* *Clear categorization of sources selected into those in support of your*
* *particular position, those opposed, and those offering completely different arguments.*
* *Discussion of both the distinctiveness of each source and its similarities with the others.>*

**Chapter 3**

**Planning and Methodology**

# Chapter 3: Planning and Methodology

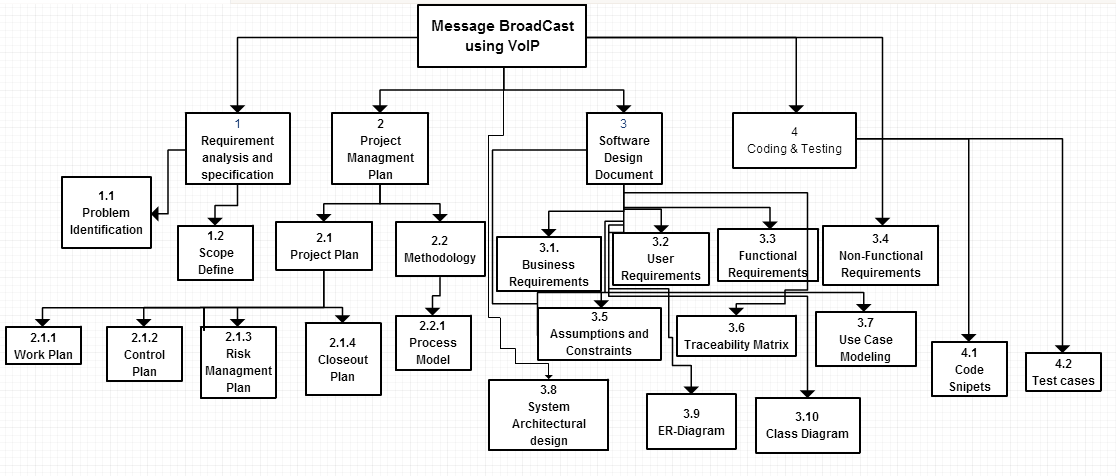
This chapter is related to project scheduling which describe what are deliverable, when they submitted to the supervisor and what methodology adopted to accomplish the task.

## Project Deliverables

The list of project deliverables is:

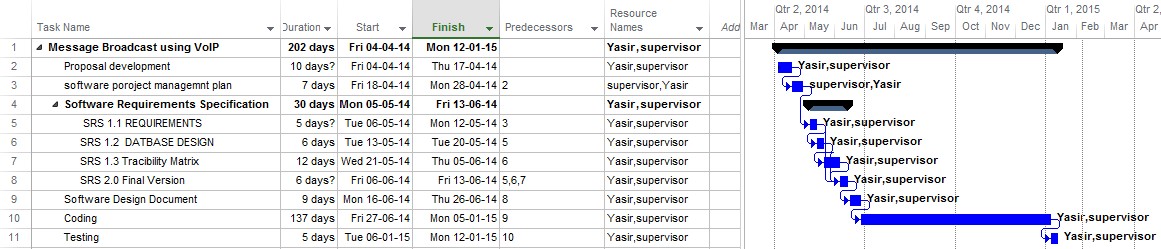
* + 1. Project Management Plan
    2. Software Requirements Specification
    3. Software Design Description
    4. Software Quality Assurance Plan (including the Software Verification and validation Plan and the Test Design Document)
    5. Working System with Relational Database Design
    6. Final Thesis Document

## Work Break down Structure



**Figure 2. 1 Work Breakdown Structure**

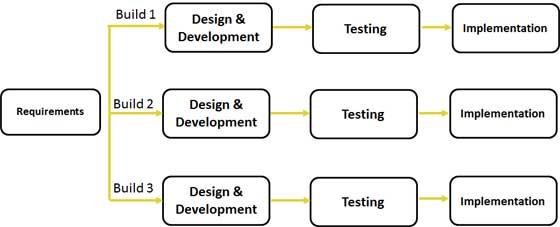
#### Process Gantt chart



**Figure 2. 2 Process Gantt Chart**

## Process Model

The Message Broadcasting on VoIP for AUMC will be implemented and executed using the Iterative Model.



**Figure 2. 3 Process Model**

The purpose of selecting this model is that requirements are complete and clearly define. The task was to develop a system through which admin can broadcast message using web portal and clients can listen it on their android devices no additional functionality was added into the system which cause change in model selection. But in design phase many changes were made.

So the process mode I adopt for developing this project is Iterative model. Because this is the only SDLC model I found suitable for my project development.

## Summary

Project plan for a gives details about the deliverables when they start and what is finishing date and mostly constructs before starting project. It’s a timetable that project team constructs to achieve goals in estimated time. Methodology is an important aspect which describe how we are going to achieve our goals. Iterative model is used in this project due to predefined requirements. Iterative model is fast development process that allow to produce prototypes so that it become easy to identify faults and make refined final product. The condition of using iterative model is, requirements must be clear in advance.

**Chapter 4**

**System Specification**

# Chapter 4: System Specification

Requirement engineering is one of the most important task in the process of software development life cycle. For developing a system, the main requirement is to first understand the system requirements. What system shall do, and what are user expectations. Better the understanding about the product, better will be the quality which is one of the major concern of a product. In this document called Software Requirement Specifications (SRS) I’ll cover all the requirements of my final year project “Message Broadcast using VoIP” which is web based for admin as well as android based application for clients.

By creating SRS it is help full for designers and developers to achieve milestones without any hurdle. The purpose of this document is to gather all the required information’s for this system. This document contains all the business requirements, user requirements, functional and nonfunctional requirements, constraints and all information that helpful in building this.

The scope of this document is to specify the requirements of the project to be delivered. To make things simple and clear this SRS is divided into three portions. This first deliverable will consist of requirements and use cases.

This document will contain:

* + 1. Business Requirements
    2. User Requirements
    3. Functional Requirements
    4. Non-Functional Requirements
    5. Use Cases

## Business Requirements

This project is to reduce the communication barrier between organization and its employees. So this system should be an effective communication channel for organization to keep its employees intact. So high speed internet is basic requirement to make good quality audio message.

### User Requirements

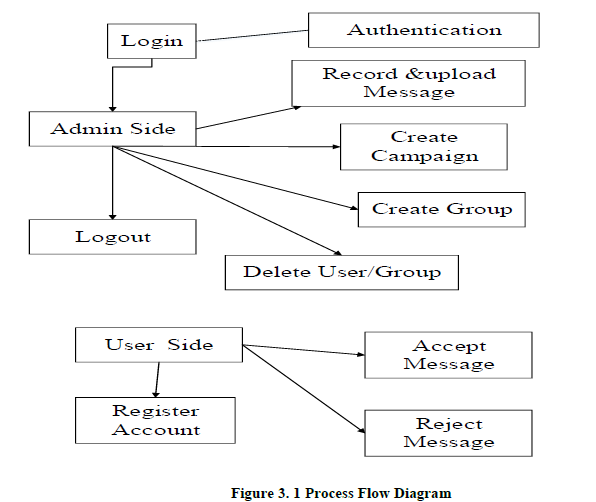
**Admin Side:**

* + Login into the System
  + Username Password Authentication
  + Record message
  + Delete user/Group
  + Sign out

**Client Side:**

Allow user to Sign up into system Via Application

## Process Flow



## 4.2 Functional Requirements

### FR01: User Login

|  |  |
| --- | --- |
| FR01- 01 | System allows admin to login into the system. |
| FR01- 02 | System allows admin to enter admin user name and password. |
| FR01- 03 | System allows admin to reset password. |
| FR01- 04 | System allows admin to remember username and password. |

|  |  |
| --- | --- |
| FR01- 05 | System authenticate username and password when clicking on Login  button. |

## Non-Functional Requirements

### NFR01: Performance

|  |  |
| --- | --- |
| NFR01- 01 | The Average load time of the starting page of the system must be  less than 5 seconds. |
| NFR01- 02 | Average processing time taken by the system to complete a  transaction/task by a user should be less than 10 seconds. |
| NFR01- 03 | System Mean Time to Failure should not be more than 5 minutes  within 24 hours of use. |

## Assumptions and Constraints

### Development Languages and Tools

*Server Side*: PHP

*DBMS*: MySQL database

*Client Side*: Android Application CSipSimple

*Platform*: Eclipse, Drupal

### Operating System

The system built in CentOS. At server side it runs on Air University Server and at client side it accessed from android mobile using CSipSimple.

## Actors

The actor for this systems are:

* + - Administrator
    - End Users

Where admin is one who is performing main task which is broadcasting messages and also delete users form system. And end users are the users who will actually use application on their mobile and their task is just receive or reject the message.

## Use-Cases

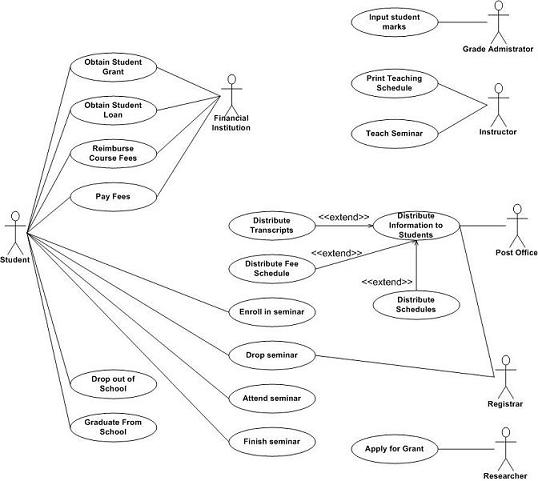
### UC01- Log into the System

|  |  |
| --- | --- |
| **Description:** | This use case defines the User Login process by which user logged into the system according to the privileges assigned. |
| **Actors:** | Administrator |
| **Pre-condition(s):** | Users must have an account for login. |
| **Main Flow:** | Use case starts when actor (admin) access the system through its web address.  The system displays a screen for login.  Admin shall enter Username and Password on login screen  Admin shall press Submit button.  System shall authenticate the actor and grant privileges.  The system takes the actor to home page.  Use case ends with the successful login of the user. |
| **Alternative Flows:** | 5a) System prompts the user for missing username and password.  5b) if login information is not authenticated, system prompts the user for correct and valid login information. |
| **Exceptions:** | If the database server is not responding, system shall display the message:  *“The database server is down. Please check back after a few minutes.”*  If web server is down, system shall display the message:  *“The web server is down. Please check back after a few minutes.”* |
| **Post-condition(s):** | User is logged in to the system and can use the system as the privileges assigned to that user and is redirected to home page. |

*<Mention all the use-cases using same format.>*

## Use Case Modeling

### Complete System Diagram



**Figure 3. 2 Complete System Diagram**

## Traceability Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Functional Requirements | | | | | |
| Use Cases | FR1 | FR2 | FR3 | FR4 | FR5 |
| UC1 | **X** |  |  |  |  |
| UC2 |  | **X** |  |  |  |
| UC3 |  |  | **X** |  |  |
| UC4 |  | **X** |  | **X** |  |
| UC5 |  |  |  | **X** |  |
|  |  |  |  |  |  |

## Behavioral Model

* + 1. **Sequence Diagrams**

### Use Case 01- Login



**Admin**

**1: press()**

**2: login()**

**invalid entry**

**3: fill()**

**4: Press()**

**5: Validation()**

Successful Login

**Login**

**Login**

**Button**

**Login**

**form**

**Login**

**button**

**Figure 4. 1 Login Use Case**

## Class Diagram



**1**

-UserName

-UserID

-email

-Password

**1**

**\***

**1**

**\***

**1**

**1**

**1**

**1**

**1**

**\***

**1**

**\***

**Group**

**\***

-GroupID

-GroupName

+CreateGroup(GroupID,GroupName) : void

+ ViewGroups() : void

**1**

+UploadMessage() : void

+RemoveMessage() : void

+ViewMessage() : void

+createExtension(ExtNumber,UserID) : void

+ViewExtension() : void

-ExtNumber

-userID

-Recording\_NAme\_ID

-RecordingName

-Description

-Path

**Extension**

**Record**

+SignUP(email,Password,UserName) : void

+SignIn(email,Password) : void

+UpdateInfo() : vod

+ViewUser() : void

+DefineRole(RoleID, RoleName) : void

+DeleteRole(RoleName) : void

+ViewRoles() : void

-RoleID

-RoleName

**Role**

**User**

**\***

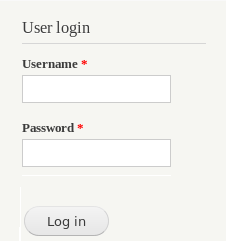
|  |
| --- |
| **Campaign/Broadcast** |
| -CampName  -GroupName  -Schedule |
| +CreateCampaign(CampName,GroupName,Schedule) : void  +ViewCampaign() : void  +ViewSummery() : void |

**Figure 3. 14 Class Diagram**

## Graphical User Interface (GUI)

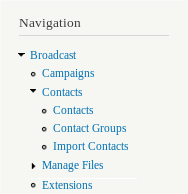
### GUI 01- Login

**Admin Side:**



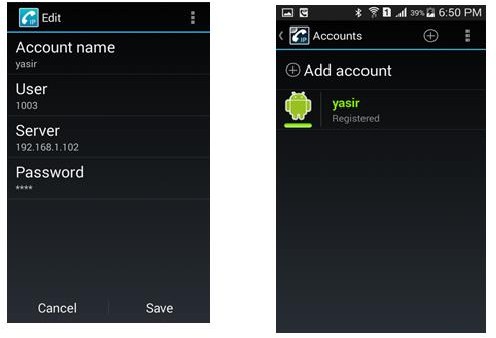
**Menu:**

**Figure 3. 15 Admin Login GUI**



**Figure 3. 16 Admin Menu GUI**

### GUI 04- Sign Up



**Figure 3. 19 Client Side GUI**

## Summary

This chapter is detail description of architectural design, use cases, functional requirements, non-functional requirements, Class diagram, ER-diagram, Traceability Matrix etc. All the specifications are presented in this chapter with full detail.

**Chapter 5**

**System Design**

# Chapter 5: System Design

This chapter consist of design diagram of the system. All use case models, behavior and structural models explained in detail.

## Design Goals

**Performance**

Performance is major concern of this system. The rapid increase in number of users is irrefutable. So the system must be able to perform better in every situation. The performance aspect plays very important role in quality of product. If product is working properly in its peak time then it is assume to be good. So in this project performance, from the beginning is major concern. By better performance I as a developer, assume that admin will be able to upload files and start multiple campaign without system crash. Increase in memory should not effect on system performance.

**Security**

Security is another major concern. As the product will deploy on internet security will be very important requirement. The people with different roles will access the software. The user groups that exist in the organization that will use the system are the administrator and student in this University scenario but will change according to type of organization e.g business organization. Each user group has different access rights. These access rights should be implemented for any security or privacy issues that affect the protection of the data created by the software system.

**Usability**

The most important aspect for any software is its usability. The system must be easy to understand and easy to use. Its interface should be understandable. And user must not be stuck at any stage. The system must tell himself how to use it. Help should be available at each stage and the system should be like, once user go through with it, he/she never forget about its usage even after a big break. So to keep it simple and clear I’ve try my best to make this system easy and fruitful for all type of users.

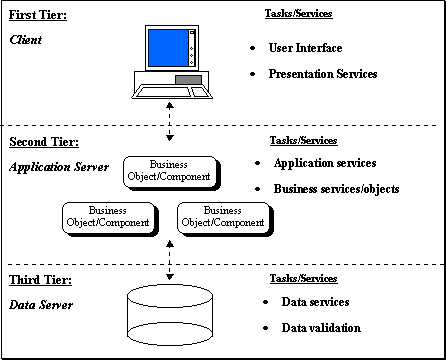
**Availability**

The system should be available to the users all the time at each cost. As this system is web based as well as mobile base. So on both ends it must be available. It should take milliseconds to perform operations like login, signup etc.

**Maintainable**

After the software deployment, if client want to add some more functionality to the software system, the software must be flexible enough to accomplish any change in it.

## System Architecture Diagram

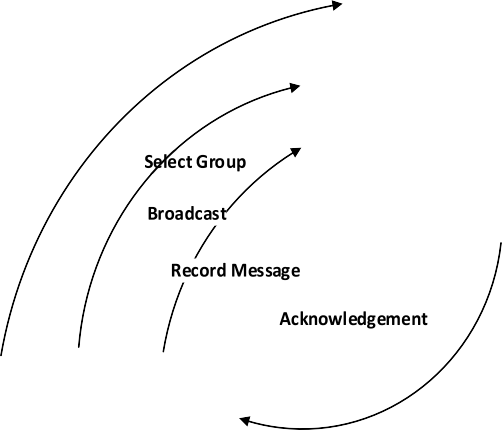
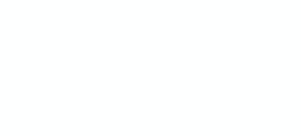
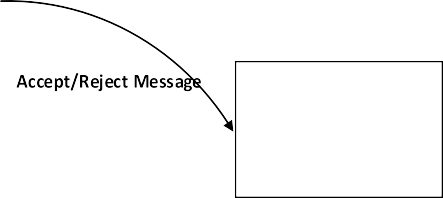
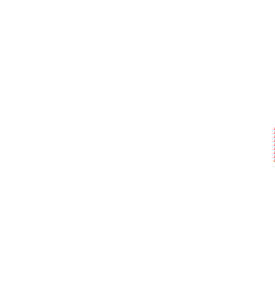


**Figure 3. 10 System Architecture Diagram**

## Data Model

**Data Flow Diagram of Message Broadcasting on VoIP**

### Context Level



Message

Broadcasting

End User

Admin

### Level 1 DFD

**Figure 3. 11 Context Level Diagram**

**Select Group request**



**Upload Req.**



**Delete User Request**

**Send view request**

**Delete**

**User**

**View Users**

**Delete User**

**Upload File**

**Message**

**Record Message**

**Record Message Request**

**Broadcast**

**Send Message**

**Admin**

**Delete Req.**

**Select Group**

**New Group req.**

**Insert Req.**



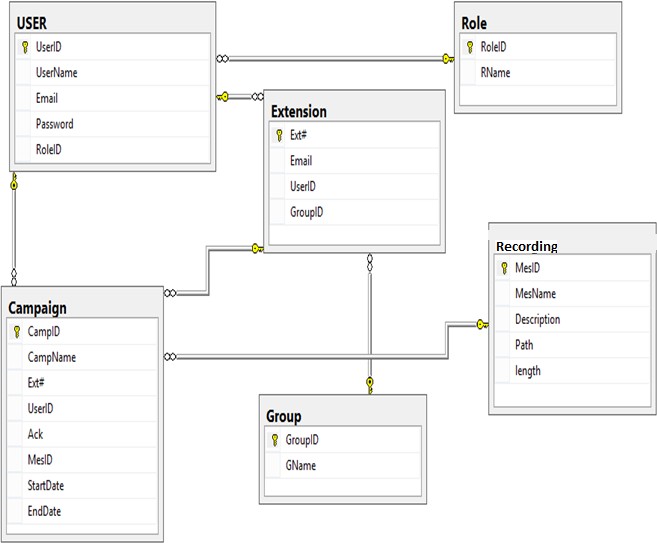
**Add Members**

**Create Group**

**Group**

**Figure 3. 12 Level 1 Data Flow Diagram**

## ER-Diagram



**Figure 3. 13 ER Diagram**

## Data Dictionary

## Table Name (e.g User)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Is Primary Key** | **Is Null** | **Comment** |
| UserID | int | Yes | No | Stores user ID |

## Personal Info

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Is Primary Key** | **Is Null** | **Comment** |
| CNIC | Int(14) | Yes | No | Stores CNIC |

**Table 3. 7 Data Dictionary**

## Structural Model

**\***

### Structural Model of the System



-UserName

-UserID

-email

-Password

**1**

**1**

**\***

**1**

**\***

**1**

**1**

**1**

**1**

**1**

**\***

**1**

**\***

**Group**

**\***

-GroupID

-GroupName

+CreateGroup(GroupID,GroupName) : void

+ ViewGroups() : void

**1**

+UploadMessage() : void

+RemoveMessage() : void

+ViewMessage() : void

+createExtension(ExtNumber,UserID) : void

+ViewExtension() : void

-ExtNumber

-userID

-MessageID

-MessageName

-Description

-Path

**Extension**

**Recording**

+SignUP(email,Password,UserName) : void

+SignIn(email,Password) : void

+UpdateInfo() : vod

+ViewUser() : void

+DefineRole(RoleID, RoleName) : void

+DeleteRole(RoleName) : void

+ViewRoles() : void

-RoleID

-RoleName

**Role**

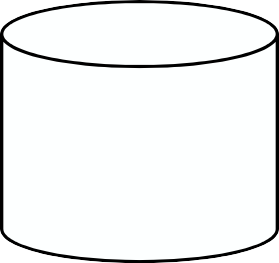
**User**

|  |
| --- |
| **Campaign/Broadcast** |
| -CampName  -GroupName  -Schedule |
| +CreateCampaign(CampName,GroupName,Schedule) : void  +ViewCampaign() : void  +ViewSummery() : void |

**Figure 4. 7 Structural Model**

## Deployment Diagram

**Database Server**



<<Private Network>>

**Application Server**

<<Public Network>>

<<Public Network>>

**Message Broadcast Interface**

**Message Receiving Interface**

**Figure 4. 10 Deployment Diagram**

**Summary**

Design plays vital role in acceptance or rejection of a software and it is the most difficult part of development. At this point all non-functional requirements have to meet. This chapter describe the flow of steps using sequence diagrams, behavioral model and deployment diagram of the system etc.

**Chapter 6**

**Coding**

# Chapter 6: Coding

This chapter contains code of only the major functionalities of the project. Full code is provided in disk along with the thesis.

## Android Side Development

## Web API on Server Side

**Chapter 7**

**Software Testing**

# Chapter 7: Software Testing

To ensure quality of the product black box testing performed on final product to make sure there is no more errors left. This chapter consist of black box testing on product.

## Test Case 01: LOGIN

This test case is implemented on admin side, and will be used whenever admin of the site try to access the page. This test case will check that user will not

* + - Enter invalid username or password.
    - Fields must not be left empty.
    - Proper sequence of email will be tested e.g. [username@somedomain.com](mailto:username@somedomain.com)

**Traceability Matrix Reference**

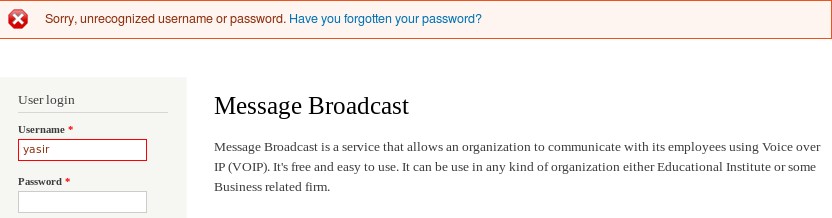
4.9-1- FR01: User Login

**Database Table**

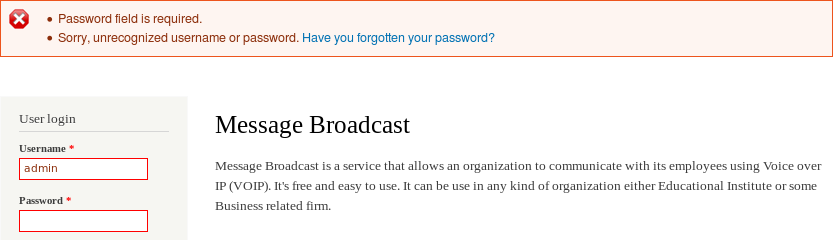


**Screen Shot**:

**1-**

****

**2-**



**Test case Ingredients:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test**  **case ID** | **Pre-Condition** | **Actions** | **Expected Result** | **Test**  **Result** |
| Login- 01 | User must have admin rights.  Must have valid username and password.  Connect to internet and currently on login page is mandatory. | Enter valid Username and Password and  press login button | Redirect to main page. | Pass |
| Login- 02 | Enter Username and no password and press login button | Password field is required.  Sorry, unrecognized username or  password | Pass |
| Login- 03 | Enter Invalid Username and valid password and press  login button | Sorry, unrecognized username or  password. | Pass |
| Login- 04 | Both fields are empty and press login button | Sorry, unrecognized username or  password. | Pass |
| Login- 05 | Invalid username and password entered and press login button | Sorry, unrecognized username or  password. | Pass |

**Table 6. 1 Login Test Case**

## Test Case 02: Sign Up

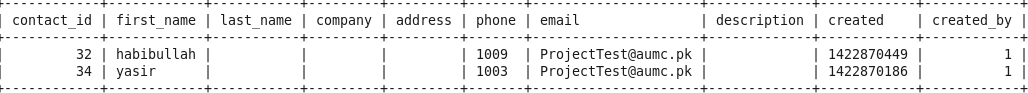
This test case is implemented on user side. User will enter Account name, username. In this use case it will be tested that

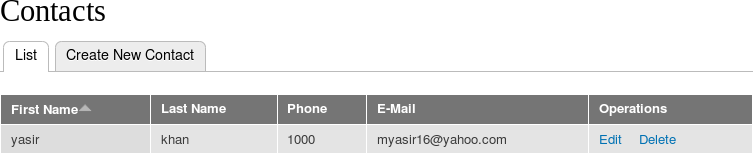
* + 1. User as enter all the required fields. And no mandatory field left empty.
    2. No special character, spaces, or integer values are not allowed in Account name.
    3. Characters are disabled are username, server address, and password field.
    4. Password and server id are fixed by default in code. So there is no need to enter them.

**Traceability Matrix Reference**

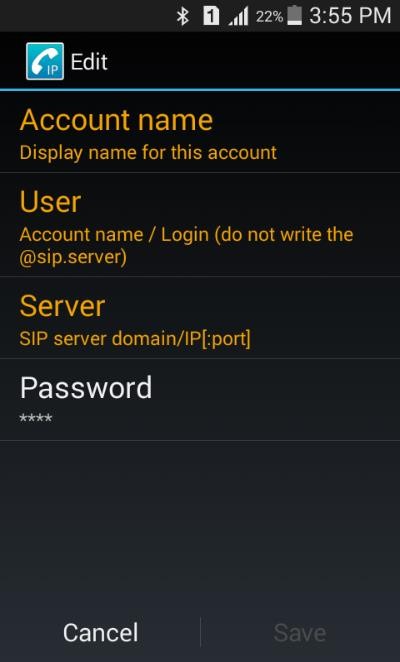
4.9-.5- FR05- Sign up

**Database Table**





**Screen Shot:**



**Test case Ingredients:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case**  **ID** | **Pre-Condition** | **Actions** | **Expected Result** | **Test**  **Result** |
| Signup- 01 | -User have android mobile set.  -Client side  application “CSipSimple” installed on android mobiles.  -Connection to internet is mandatory. | Enter Account name in character form. | Value successfully entered. | Pass |
| Signup-  02 | Enter username in  character format. | Input doesn’t  enter. | Pass |
| Signup-  03 | Enter username in digit  format. | Successfully  entered. | Pass |
| Signup-  04 | Enter password 1234 | Successfully  entered. | Pass |
| Signup-  05 | Click Save button | Button disabled | Pass |
| Signup-  07 | Enter Server IP Address | Successfully  entered. | Pass |
| Signup-  07 | Click Save Button. | Account  Registered. | Pass |

**Table 6. 2 Signup Test Case**

**Chapter 8**

**Conclusion**

# Chapter 8: Conclusion

This thesis document contains the descriptive detail of project Message Broadcasting. Message Broadcast is a VoIP based application that provides one way communication between organization and its employees. Through this project it will be possible to communicate with employees in a totally new way. Currently emails and GSM technology is implemented which is not cost effective and requires time but message broadcast is VoIP based application that provides communication free of cost and user doesn’t need to read the message, he can listen whatever message is sent during traveling or during any busy routine. He just need to connect with internet.

This project only overcome one part that is making a call. But this project can be extended to voice message. Currently, if due to some reason user didn’t receives call then he cannot listen it again. But if a voice message will be send then user will be able to listen it any time and this message will be store in his message box. So in future, if anyone interested in enhancement of the scope of this project it will be a great effort.

# User Guide

**User Manual:**

On client side CSipSimple application is running. To register into the application user need perform following actions:

* + 1. Open application
    2. Click on add account
    3. From pop up screen click on sign up button
    4. Fill form by providing appropriate values.
    5. In account name enter name like Bob. (only characters allow)
    6. In username field enter extension number which is like a contact number but it should be In range between 1000-1019

Currently, application is running on private network so Server IP is 192.168.1.102. IP other than this will cause problem. Future developers can change it through code.

Password is set to 1234. Do not change it.

* + 1. After filling form click on save button.

You will see new screen with account name and registered word. Which indicates that you are connecting with server.

**Admin Manual:**

To start campaign admin must have to start Free switch on server so that campaign can start.

Whenever new user register admin must have to assign him a group otherwise user won’t be able to receive call.

Major task of admin is to start campaign. Default username and password to enter into the system is *“admin”.* To create new campaign admin has to perform following operations.

* + 1. Click on Campaign from menu.
    2. Click on Voice Campaign.
    3. Click on Create new Campaign
    4. Enter Campaign Name
    5. Enter Delay. Default is 0 which means call will generate at once to all contacts.
    6. Select message type “Voice”.
    7. Select message file that want to broadcast.
    8. Select Contact Group to whom message to be sent.
    9. Select user from trunk.
    10. [Optional] change time to start campaign.
    11. Click submit.

To upload record message admin needs to simply click on **manage file  Upload File.** File that is to be upload must be in *.wav* format.

# Reference

1. Regis,”sip\_transport. Error processing 531 bytes packet from UDP”,”CSIPSIMPLE Development”,[online]2014,”[https://groups.google.com/forum/#!forum/csipsimple- dev](https://groups.google.com/forum/%23!forum/csipsimple-dev)” (Accessed: 10-December-2014);
2. “Calling REST API from Android Application”,”Sourcecodemania”, [online] 2014,”<http://sourcecodemania.com/calling-rest-api-from-android-application/>” (Accessed: 15-December-2014).
3. Falak Nawaz,”Freeswitch and Plivo”,” ICTDialer Discussion. View Forum”,[online] 2014, “[http://forum.ictdialer.org/viewforum.php? f=8&sid= 7124078d777 af6fcbe f2aaeae4098edd](http://forum.ictdialer.org/viewforum.php?%20f=8&amp;sid=%207124078d777%20af6fcbe%20f2aaeae4098edd) ” (Accessed: 21-December-2014).
4. “Voice Over IP”,”Voice Over IP”, [online] 2012,”[http://en.wikipedia.org/ wiki/ Voice\_over\_IP](http://en.wikipedia.org/%20wiki/%20Voice_over_IP)” (ACCESSED: 04-January-2015).
5. “Work Break Down Structure”,” Work Break Down Structure”, [Online] 2011,”<http://en.wikipedia.org/wiki/Work_breakdown_structure>”, (Accessed: 12- January-2015).
6. Kashif Ghafoor, “[VoIP Regulation in Pakistan ? A General Perspective](http://www.pta.gov.pk/index.php?option=com_content&amp;view=article&amp;id=668%3Avoip-regulation-in-pakistan--a-general-perspective&amp;catid=150%3Apublications-articles-views-expressed-in-articles-are-of-independent-authors)”, “[VoIP Regulation in Pakistan ? A General Perspective](http://www.pta.gov.pk/index.php?option=com_content&amp;view=article&amp;id=668%3Avoip-regulation-in-pakistan--a-general-perspective&amp;catid=150%3Apublications-articles-views-expressed-in-articles-are-of-independent-authors)”, [Online] 2005, “[http://www.pta.gov.pk/index.php?option=com\_content&view=article&id=668:voip](http://www.pta.gov.pk/index.php?option=com_content&amp;view=article&amp;id=668%3Avoip-regulation-in-pakistan--a-general-perspective&amp;catid=150%3Apublications-articles-views-expressed-in-articles-are-of-independent-authors)

[-regulation-in-pakistan--a-general-perspective&catid=150:publications-articles-](http://www.pta.gov.pk/index.php?option=com_content&amp;view=article&amp;id=668%3Avoip-regulation-in-pakistan--a-general-perspective&amp;catid=150%3Apublications-articles-views-expressed-in-articles-are-of-independent-authors) [views-expressed-in-articles-are-of-independent-authors](http://www.pta.gov.pk/index.php?option=com_content&amp;view=article&amp;id=668%3Avoip-regulation-in-pakistan--a-general-perspective&amp;catid=150%3Apublications-articles-views-expressed-in-articles-are-of-independent-authors)”, (Accessed: 03-February- 2015).

1. Macnews, “VoIP applications”, “MacWorld”, [Online] 2009, “<http://www.macworld.com/article/1051658/voip.html>”, (Accessed: 03-February- 2015).
2. Nadeem Unuth, “Software based VOIP services and Applications”, “Soft Phones”, [Online] 2014, “<http://voip.about.com/od/voipsoftware/a/SoftphoneList.htm>”, (Accessed: 03-February-2015).
3. “SDLC Overview”, “SDLC Quick Guide”, [Online] 2010, “<http://www.tutorialspoint.com/sdlc/sdlc_quick_guide.htm>”, (Accessed: 12-January- 2015).

# Glossary

**Circuit Switching**: the path/channel from source to destination reserves at initial level and release when communication ends.

**ER-Diagram**: Entity Relationship Diagram

**GSM:** Global System for Mobile Communications.

**NGN**: Next Generation Networks

**Packet Switching:** a method for sending data whereby the data is divided in **packets**. Each **packet** is given a header containing information of the destination. Each **packet** is forwarded through the network to the destination using this information.

**SDLC:** Software development life cycle.

**VOIP:** A methodology and group of technologies for the delivery of voice communications and multimedia sessions over Internet Protocol (IP) networks, such as the Internet.