

# **Contents**

## **1 Introduction**

1. Purpose
2. Scope
3. Overview
4. Definitions

## **2 General Description**

1. Product perspective
2. Product Functions
3. User Characteristics

## **3 Specific Requirements**

1. Functional Requirements
  1. Requirements of Website
  2. Requirements of Administrator
- 3.2 Hardware Interfaces
  1. User Interfaces
  2. Software Interfaces
2. Performance Requirements
3. Other Requirements
  1. Data Base

## **3 Test Case**

1. Login Page
2. Registration Page
3. Inbox Functionality(Receive Email)
4. Compose Mail Functionality

## **4 Case Diagram**

1. Diagram

## **5 DFD (Data Flow Diagram)**

1. Diagram

# Chapter-1

## Introduction

### 1. Purpose

This Document describes the software requirements of Gmail website. Gmail is used for sending and receiving e-mail message worldwide.

### 2. Scope

The main Function for making Gmail is to help users emailing, chatting, and Archiving your mail and conversations very easy and convenient.

### 3. Overview

In this document all requirements for making Gmail website are mentioned along with these all definitions related to Gmail website are written in section 1. All functional and non-functional requirements for Gmail are specified in section 2.

### 4. Definitions

- Account- Account term basically refers to an identity on Gmail website. A user can easily make an account on Gmail website by just giving some credentials like name, address, E-mail id , phone number.
- Inbox- Inbox by Gmail is an email service developed by Google. Inbox is the name given to the folder where your newly-delivered email message appear.
- User- User is a person who access website for emailing, chatting, and archiving your mail and conversation. User can also be named as Customer.
- Starred- Gmail's star system allows you to mark your most important emails so you can easily find them later. By default, starred message are labeled with a yellow star, but you can add other color and types of stars. Stars display to the left of the sender's name in your inbox.
- Sent mail- Sent mail is Gmail's own system label where .Sent-mail contain all of your mail that has gone out via Google's Gmail servers.
- Draft- Draft represent an unsent messages with the DRAFT system label applied. This is perfect to avoid accidental data loss.
- Unwanted- It helps the user to block all the unwanted messages that the user do not want to receive in future.

# Chapter-2

## General Description

### 1. Product Perspective

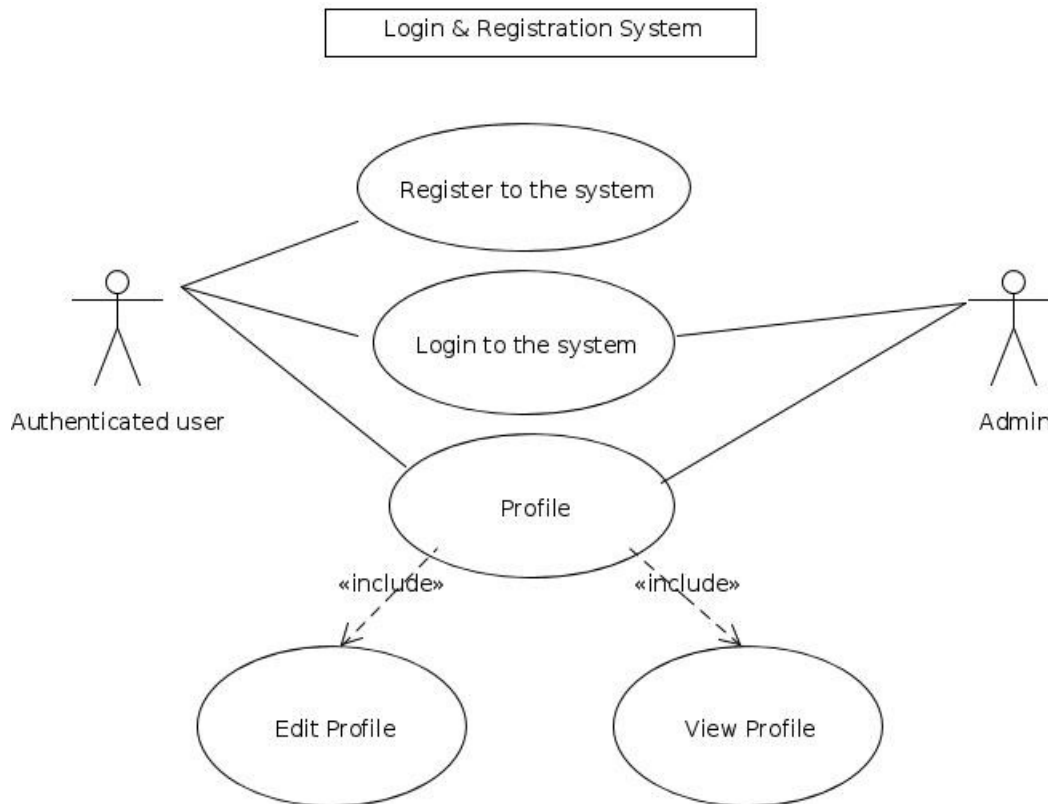
The Gmail website is Operated by Computer/ Software owned by the administrator of Gmail. Website can be accessed by anyone at any of the system, but only in user mode.

### 2. Product functions

The software should support a computerized network over internet. Gmail provides a various way of sending mails. User have the facility to send mail even in group also. User also have the facility to attach the document as per there need. Users can also save the message as a draft. They also have the visibility of send mails.

Chatting privilege is also there for the user interactions among themselves.

Blocking facility helps the user to block unwanted messages.



## 2.3 User Characteristics

These are several users of our website-

- Customer- Website should be user friendly and easy to use. All the categories must be well organized so that our user must not face any type of difficulties.
- Administrator- For administrator purpose it should be well organized website. All the data which he want to organize can be done easily.

# **Chapter-3**

## **Specific Requirements**

### **1. Functional Requirements**

Functional requirements are further divided into two parts-

1. For Customer
2. For Administrator

#### **1. Requirements for Customer**

##### **General Requirements**

##### **Functional Requirement 1**

- Description-  
After entering URL user will directed to Homepage.
- Input-  
URL using keyboard.
- Output-  
Homepage Shown to user.

##### **Functional Requirement 2**

- Description-  
If server is busy or Maintenance period.
- Input-  
URL using keyboard.
- Processing-  
Check Network traffic on web.
- Output-  
Display an error message.

##### **Functional Requirement 3**

- Description-  
Ask user to login or signup in his/her account.
- Input-  
E-mail Id, Name, Phone number and password.
- Processing-  
Store all the information In database and send a confirmation code on Contact number.
- Output-

Display a message that Signed in Successfully and redirect user to homepage.

#### Functional Requirement 4

- Description-  
Creating message.
- Input-  
Compose message.
- Processing-  
System opens prompt window to write a message.
- Output-  
Display the command message send successfully.

#### Functional Requirement 5

- Description-  
View all received messages.
- Input-  
Click on Inbox.
- Output-  
All Categories of messages are displayed to user.

#### Functional Requirement 6

- Description-  
Important messages to view later
- Input-  
Click on Starred.
- Processing-  
System will check in database for all the starred message.
- Output-  
All the starred messages are displayed to user.

#### Functional Requirement 7

- Description-  
History of send mails.
- Input-  
Click on Sent Mail
- Processing-  
System check in database about all the send messages.
- Output-  
Displays list of all send messages.

### Functional Requirement 8

- Description-  
List of messages that were failed to send.
- Input-  
Click on Drafts Button.
- Processing-  
System checks in database for all the message that were failed to send.
- Output-  
Display list of unsent messages.

### Functional Requirement 9

- Description-  
Transaction page.
- Input-  
Bank name, card number, one time password.
- Output-  
Display Order placed with unique order id.

### Functional Requirement 10

- Description-  
Send message with warning.
- Input-  
Click on send button.
- Processing-  
System checks whether the e-mail id is valid or not
- Output-  
Display a message if the e-mail id is not valid.

### Functional Requirement 11

- Description-  
Acknowledgment of send mail.
- Input-  
After clicking on send button.
- Processing-  
System checks in database, the accept delivery report.
- Output-  
Display receipt notification to the user.

### Functional Requirement 12

- Description-  
Recovery of deleted messages.
- Input-  
Click on Trash.
- Process-  
Database checks the record of deleted messages.
- Output-  
Display “successfully recovered” message.

## **2. Requirements for Administrator**

### **Functional Requirement 1**

- Description-  
Adding Canned response.
- Inputs-  
Insert, saves, delete.
- Process-  
Save canned on gmail.
- Output-  
Saved “Canned “Successfully”.

### **Functional Requirement 2**

- Description-  
Default “Reply All”.
- Input-  
Default reply behavior.
- Process-  
Default behavior added to database.
- Output-  
Two behavior added successfully.

### **Functional Requirement 3**

- Description-  
Video and audio chat.
- Input-  
Chat options.
- Process-



Modify and updating chat option .

- Output-  
Display “Chat option updated” message.

#### Functional Requirement 4

- Description-  
Total search power.
- Input-  
Check power consumption.
- Process-  
Save all details in database.
- Output-  
Display Button “Power”.

#### Functional Requirement 5

- Description-  
Change Id name.
- Input-  
Click to change username.
- Process-  
Search username in database.
- Output-  
Display “change username”.

## 2. Hardware Interfaces

### 1. User Interface

It must fulfill user demands and their aspects. It should be easy to use and it must fulfill user demands and their aspects. It should be easy to use and it must contain useful information's not useless information.

### 2. Software Interfaces

This software provide interface to-

1. Gmail.com
2. Gmail application

## 3. Performance Requirements

### Performance Requirement 1

- Description

If user forgot to logout the session then it should be automatically logged out.

### Performance Requirement 2

- Description

There should be pop setting that would automatically allow to check for new message after every X amount of minutes.

## 4. Other requirements

### 3.4.1 Database

Website must able to store every kind of data means every format of data.

Website should be frequently connectable to database or it must be database sensitive.

# **Chapter-4**

## **TEST CASES**

### **1. Test Case 1:**

Test case Name: Login Page

- Verify that user is not able to login with invalid username and password
- Verify that the login screen is having option to enter username and password with submit button and option of forgot password
- Verify that user is able to login with valid username and password
- Verify that validation message gets displayed in case user leaves username or password field as blank
- Verify that validation message is displayed in case user exceeds the character limit of the user name and password fields
- Verify that there is reset button to clear the field's text
- Verify if there is checkbox with label "remember password" in the login page
- Verify that the password is in encrypted form when entered
- Verify that there is limit on the total number of unsuccessful attempts
- For security point of view, in case of in correct credentials user is displayed the message like "incorrect username or password" instead of exact message pointing at the field that is incorrect. As message like "incorrect username" will aid hacker in brute forcing the fields one by one

- Verify the timeout of the login session
- Verify if the password can be copy-pasted or not
- Verify that once logged in, clicking back button doesn't logout user

## **2. Test Case 2:**

### **Test Case Name: Registration Page**

- Verify that all the specified fields are present on the registration page
- Verify that the required/mandatory fields are marked with \* against the field
- Verify that for better user interface drop downs, radio buttons and check boxes etc fields are displayed wherever possible instead of just text boxes
- Verify the page has both submit and cancel/reset buttons at the end
- Verify that clicking submit button after entering all the required fields, submits the data to the server
- Verify that clicking cancel/reset button after entering all the required fields, cancels the submit request and resets all the fields
- Verify that whenever possible validation should take place at client side
- Verify that not filling the mandatory fields and clicking submit button will lead to validation error
- Verify that not filling the optional fields and clicking submit button will still send data to server without any validation error
- Check the upper limit of the text boxes

- Check validation on date and email fields (only valid dates and valid email Ids should be allowed)
- Check validation on numeric fields by entering alphabets and special characters
- Verify that leading and trailing spaces are trimmed
- Verify that entering blank spaces on mandatory fields lead to validation error

### **3. Test Case 3:**

#### **Test Case Name: Inbox Functionality(Receive Email)**

- Verify that a newly received email is displayed as highlighted in the Inbox section.
- Verify that a newly received email has correctly displayed sender email Id or name, mail subject and mail body(trimmed to single line).
- Verify that on clicking the newly received email, user is navigated to email content.
- Verify that the email contents are correctly displayed with the desired source formatting.
- Verify that any attachments are attached to the email and is downloadable.
- Verify that the attachments are scanned for viruses before download.
- Verify that all the emails marked as read are not highlighted.
- Verify that all the emails read as well as unread have a mail read time appended at the end on the email list displayed in the inbox section.

- Verify that count of unread emails is displayed alongside 'Inbox' text in left sidebar of G Mail.
- Verify that unread email count increases by one on receiving a new email.
- Verify that unread email count decreases by one on reading an email (marking email as read).
- Verify that email recipients in cc are visible to all user.
- Verify that email recipients in bcc are not visible to user.
- Verify that all received emails get piled up in the 'Inbox' section and gets deleted in cyclic fashion based on the size availability.
- Verify that email can be received from non-mail email Ids like – yahoo, hot mail etc.

#### **4. Test Case 4:**

##### **Test Case Name: Compose Mail Functionality**

- Verify that on clicking 'Compose' button, a frame to compose a mail gets displayed.
- Verify that user can enter email Ids in 'To', 'cc' and 'bcc' sections and also user will get suggestions while typing the emails based on the existing email Ids in user's email list.
- Verify that user can enter multiple comma separated email Ids in 'To', 'cc' and 'bcc' sections.
- Verify that user can type Subject line in the 'Subject' text box.

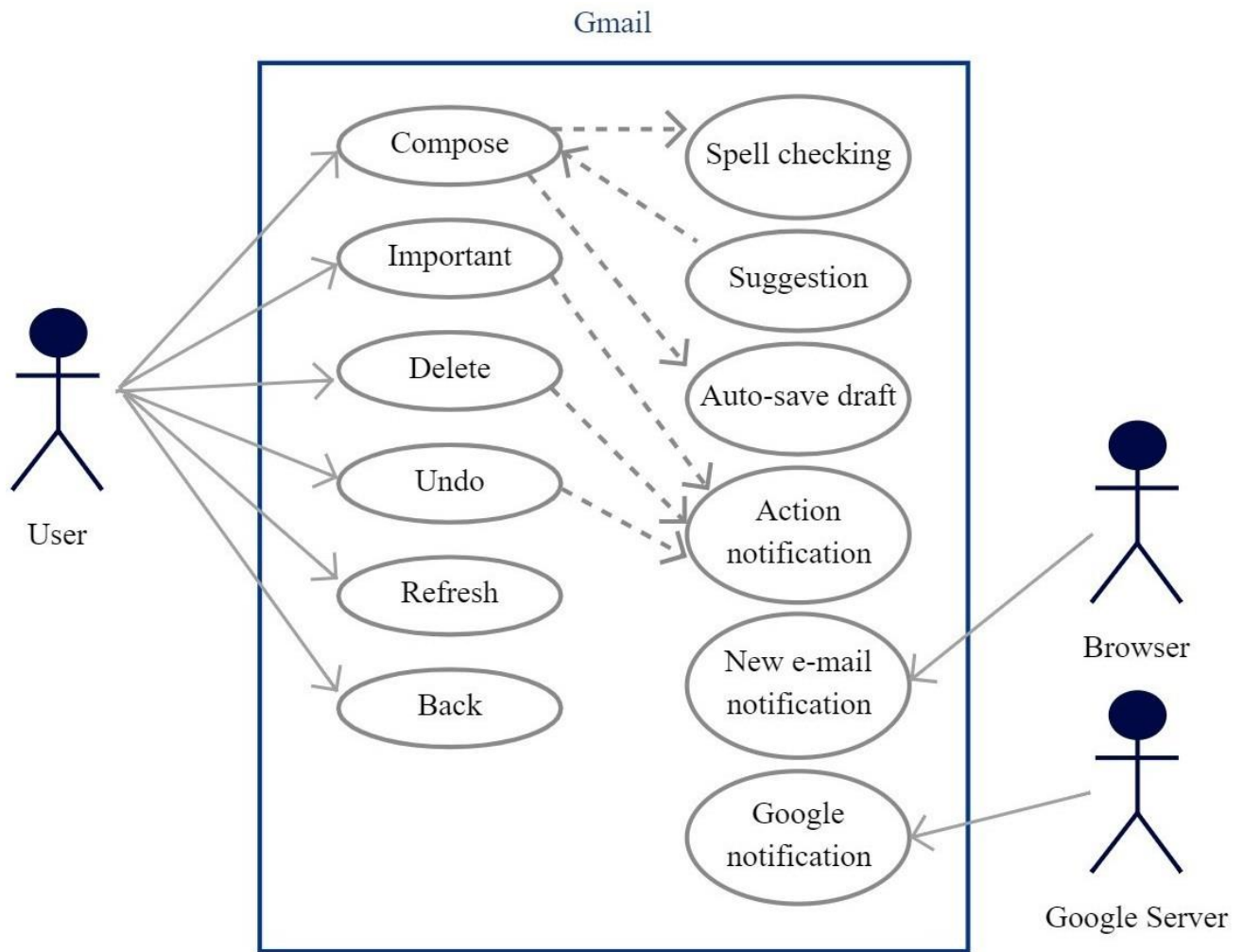
- Verify that user can type the email in email-body section.
- Verify that user can format mail using editor-options provided like choosing font-family, font-size, bold-italic-underline etc.
- Verify that user can attach file as an attachment to the email.
- Verify that user can add images in the email and select the size for the same.
- Verify that after entering email Ids in either of the 'To', 'cc' and 'bcc' sections, entering Subject line and mail body and clicking 'Send' button, mail gets delivered to intended receivers.
- Verify that sent mails can be found in 'Sent Mail' sections of the sender.
- Verify that mail can be sent to non-mail email Ids also.
- Verify that all sent emails get piled up in the 'Sent Mail' section and gets deleted in cyclic fashion based on the size availability.
- Verify that the emails composed but not sent remain in the draft section.
- Verify the maximum number of email recipients that can be entered in 'To', 'cc' and 'bcc' sections.
- Verify the maximum length of text that can be entered in the 'Subject' text box.
- Verify the content limit of text/images that can be entered and successfully delivered as mail body.
- Verify the maximum size and number of attachment that can be attached with an email.
- Verify that only the allowed specifications of the attachment can be attached with an email/

- Verify that if email is sent without Subject, a pop-up is generated warning user about no subject line. Also, verify that on accepting the pop-up message, user is able to send the email.



# Chapter-5

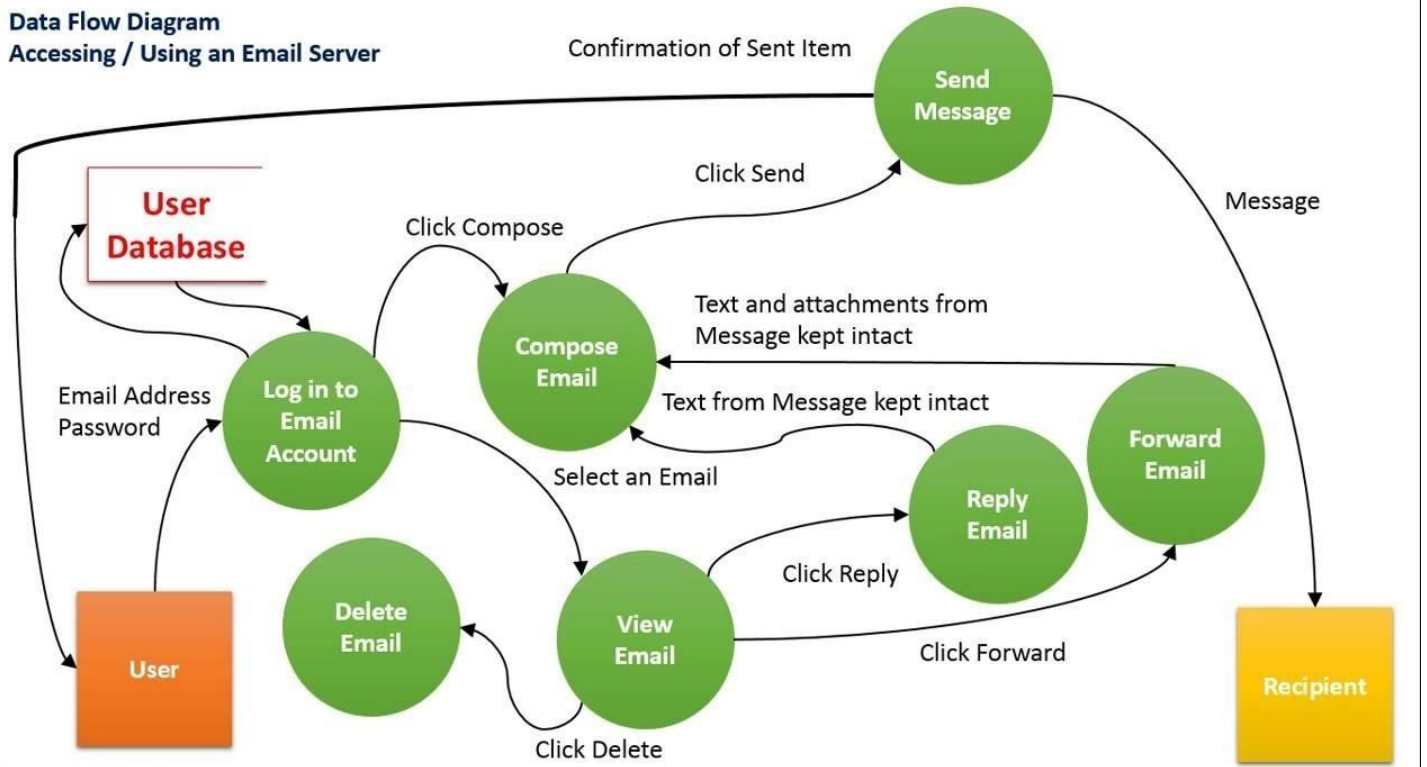
## Case Diagram



5.1(Diagram)

# Chapter-6

## DFD ( Data Flow Diagram)



6.1 (Diagram)