Software Quality Assurance

**ReserveIt**

|  |  |
| --- | --- |
| Muhammad Fakhar Choudary | CS161025 |
| Rajnesh Bajaj | CS161075 |
| Haris Ejaz | CS161012 |
| Abdul Samad | CS161047 |

Submitted To

**Ubaid Aftab Chawla**

# Method & Quality Planning Document

# Management Responsibility

**Responsibility of the Company**

The responsibility of the company is to ensure the quality of the software that would meet the user requirements with correct output.

**Quality Achievement Steps**

We will take six Quality achievement steps and the following are as follows:

* We will define the quality characteristics of the product
* We will measure each quality characteristics
* We will set the quality standard for each quality characteristics
* We will control quality against those standards
* We will find and correct causes of poor quality
* We will make continuous to make improvements

**Quality Policy**

We will provide service that would meet the expectation of our customers. We will make active and coactive actions to improve our standards and quality and this would enable us to do the job right at the first time.

# Responsibility & Authority

**Responsibility of the Client**

* Client should have a complete business knowledge to have efficient communication with the project manager
* Client should cooperate with the project manager to maintain or increase the quality of a software

**Responsibility of the Team Members**

* Team members should take appropriate actions to ensure the quality of the software
* Team members should make proper test cases for each deliverables
* Team members should fulfill all the quality achievements steps as described above

# Quality System

We will ensure the quality of the system by making correct test cases and use case and checking all the deliverables that whether they are in best in quality or not. This system will get reviewed by the concerned authority and we will update and improve the system accordingly to their feedback.

**Steps taken for CMMI level 3**

We have taken the following steps for the CMMI level 3 and the following are as follows:

* Organization Process focus
* Organization Process Definition
* Training Program
* Integrated Software Management
* Software Product Engineering
* Intergroup Coordination
* Peer Review

**Quality Planning & Standards**

We will check the quality of a software in five different views of quality and we will use CMMI level 3 as a base standard and the following quality planning are as follows:

* Validation
* Verification
* Integration Testing
* Unit Testing
* Peer View and etc.

**Quality Milestones**

|  |  |  |
| --- | --- | --- |
| **Milestones** | **Description** | **Due Date** |
| Login / Sign Up as a customer | Login and Sign Up as a customer | 10/10/19 |
| Login/ Sign Up as restaurant owner | Login and Sign Up as a restaurant owner | 11/11/19 |
| Add a restaurant | Restaurant owner can add their restaurant in the system with number of maximum reserved seats | 12/12/19 |
| Make Reservation | Customer can make a reservation of registered restaurant | 13/12/19 |
| See all the Reservation | Customer can see all the reservation | 14/12/19 |
| Cancel the Reservation | Customer can cancel the reservations | 15/12/19 |
| Logout | Customer or owner can logout their account | 18/12/19 |

# **Document Control**

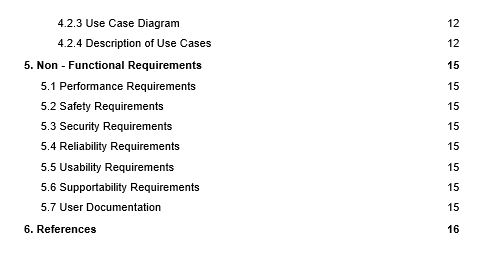
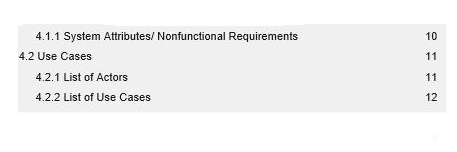
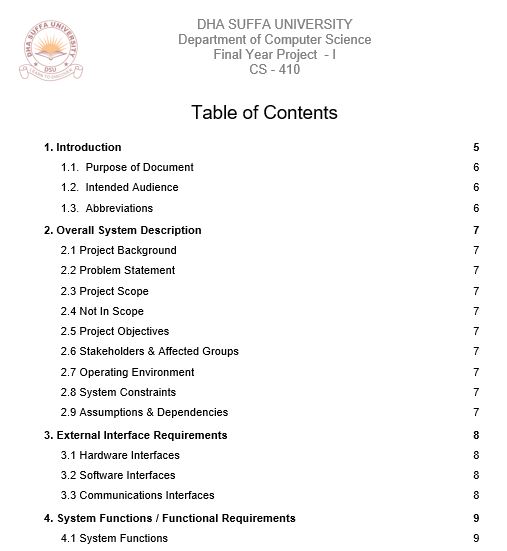
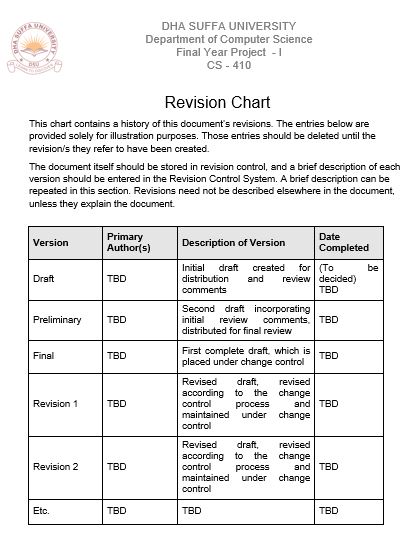
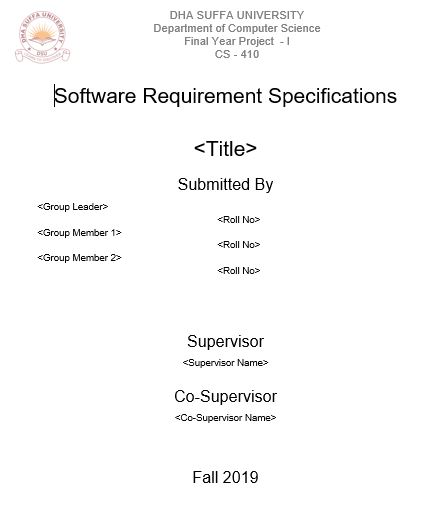
**Document Standards**

We will use ISO 9001:2015 as our document standards and some the protocol are as follows:

* Approve documents for prior of issue
* Review, update as necessary and re-approve the documents
* Identify the changes and current document revision standards
* Make non relevant documents available at point of use
* Prevent obsolete documents from unintended use

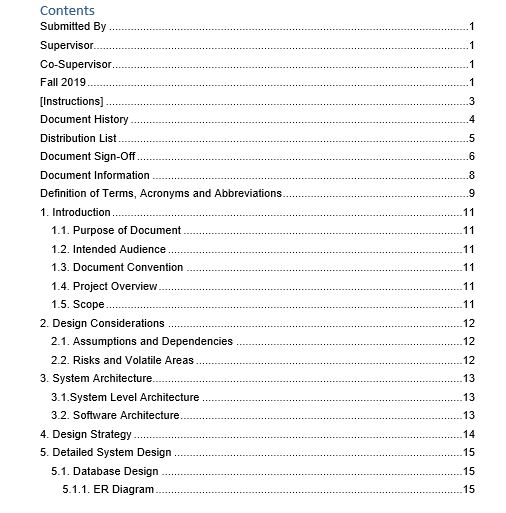
For that project we will use SRS & SDS to define goals and designs and the following pictures will describe the template of SRS & SDS.

**SRS:**



**SDS:**

****



**Identification of Each Document**

This document will consists of all the testing’s and methodologies that used in this project to ensure the best quality software that would fulfil all the requirements.

**Approved procedures**

All the procedures will be approved by the supervisor of this course and supervisor will ensures the proper procedures and documentation.

**Document changes**

There is no document changed or data changed in the file and the following table will describe the document version.

|  |  |  |
| --- | --- | --- |
| **Document** | **Version** | **Date** |
| Testing | 1.0 | 28/12/19 |

**Data Control Document**

The following table will describe the data control procedure for this file.

|  |  |  |
| --- | --- | --- |
| **Document** | **Reviewed** | **Date** |
| Testing | Y | 29/12/19 |

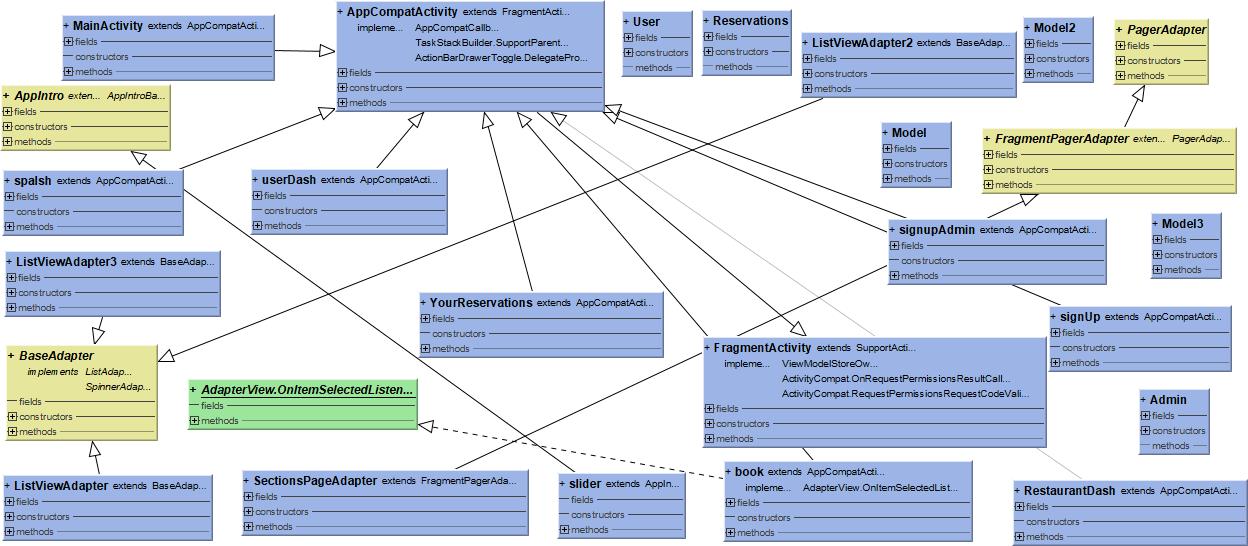
# Process Identification & Traceability Procedures

**Process Control**

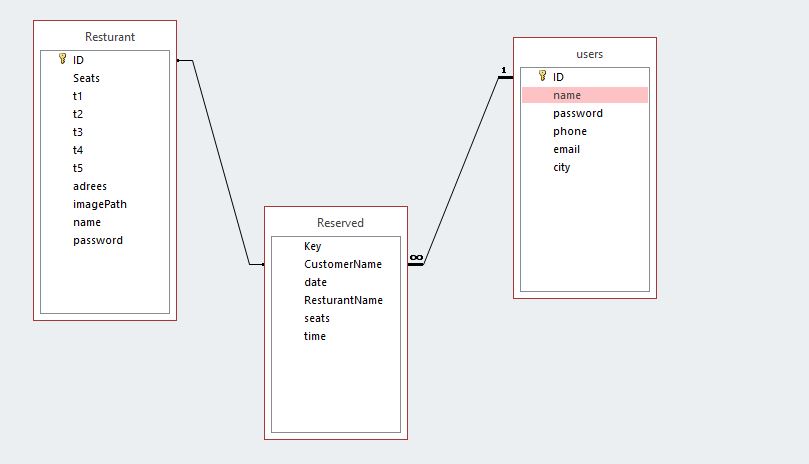
The **Process Model** methodology that is used in this project is **waterfall** and all the phases of waterfall has been completed, although this document will ensure the quality by using different testing techniques and methodologies. The following phases of waterfall are as follows:

* Requirement Definition
* Analysis
* Design
* Implementation
* Testing
* Maintenance

**Class Diagram/ UML diagram**



**ER Diagram:**



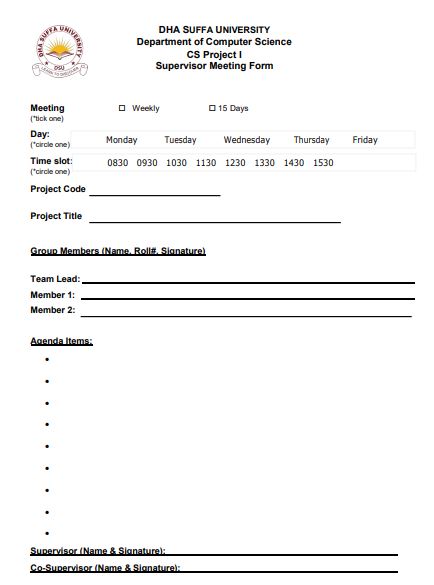
# Inspection & Testing

**Plans for Testing**

The following plans of testing are as follows:

* We will use different test techniques to ensure the quality of the software
* Our supervisor will ensure all the techniques that was used in testing
* There will be regular meetings to ensure that whether the procedures are correctly executed or not.

The following template for the meetings sheets are shown below:



# Training need

* The seminar will be held to ensure that an employee will do the right task at first time.
* The training would encourages employees and this training would help us to reduce the cost of failure.

# Handling, Storage, Packaging & Delivery

This software will distribute online it does not required any packaging and delivery logistics however this software will be stored in Google Store so that user can download this project.

# Testing Strategy & Acceptance Test Specification Documents

**Introduction**

**Document Purpose & Overview**

The purpose of this document is to conduct testing and validate/ verify all the deliverables and this would ensure that the system has a good quality.

**Project Overview**

This project will reserve a seat of your favorite restaurant with your desire time and you can cancel your reservation and restaurant owners can add their place in the system and the customers can book a seats.

**Objectives of Testing**

The following objective of testing are as follows:

* To confine the standards of CMMI level 3
* To ensure the quality of the software
* To learn and familiarize our self to the testing phase
* Perform quality assurance

**Scope & Techniques of Testing**

The scope of the this project is the reserve the seats in the desire hotels and the following testing techniques are as follows

* Unit testing
* Boundary Value analysis
* Equivalence Class Partitioning

# **Test Procedure**

**Testing steps**

We will follow the following Testing steps

* Requirement analysis
* Design process
* Development
* Testing process and debugging
* Maintenance

**Starting & Ending Criteria**

The following Stating Criteria are as follows:

* Test Plan
* Testable code with working database

The following Ending Criteria are as follows:

* Test Logs
* Complete executions of test cases

**Responsibility of People**

|  |  |
| --- | --- |
| Name | Responsibility |
| Muhammad Fakhar Choudary | Testing |
| Rajnesh Kumar Bajaj | Testing |
| Haris Ejaz | Testing |
| Abdul Samad | Testing |

**Testing Environment**

Android Operating system will be the major testing environment for this project.

# Validation Test Report

|  |  |  |
| --- | --- | --- |
| No | Product Requirement | Pass/Fail |
|  | **General** |  |
| RQ1 | ReserveIt shall have Login form, Sign Up, Sign Up as restaurant form, All restaurant list, User Reservation list, Reservation form | Pass |
|  | **Login Form** |  |
| RQ2 | When the Reserve Application is launched the login screen shall be displayed | Pass |
| RQ3 | The user can able to login, go to Sign up or sign in as restaurant | Pass |
| RQ4 | A click button shall be displayed to enable the user to go to the next screen | Pass |
|  | **Sign Up** |  |
| RQ5 | The user can able to sign up their account with correct information | Pass |
| RQ6 | The Sign Up Form shall have a Sign Up button | Pass |
| RQ7 | When user click Sign Up button, the Sign Up form shall be registered | Pass |
| RQ8 | The Sign Up Form shall request User Identification | Pass |
|  | **Sign Up as Restaurant** |  |
| RQ9 | The user can able to sign up their account with correct information | Pass |
| RQ10 | The Sign Up Form shall have a Sign Up button | Pass |
| RQ11 | When user click Sign Up button, the Sign Up form shall be registered | Pass |
| RQ12 | The Sign Up Form shall request User Identification | Pass |
|  | **Show all restaurants list** |  |
| RQ13 | The system will show all the available restaurant list. | Pass |
| RQ14 | The user can search their restaurant | Pass |
|  | **Show User reservation list** |  |
| RQ15 | The system will show all the user reservation | Pass |
| RQ16 | The system will enable user to cancel a reservation | Pass |
|  | **Reservation Form** |  |
| RQ17 | The system will able to reserve the reserve a seat | Pass |
| RQ18 | The system shall contain time and number of seats | Pass |
| RQ19 | The system shall has Reserve Seat Button | Pass |
|  | **Database** |  |
| RQ20 | The database shall contain a table | Pass |
| RQ21 | The database shall have several fields such as time, user, reservation and etc. | Pass |

**Boundary Value Analysis**

**Seats field:** User can reserve the seats.

|  |  |  |
| --- | --- | --- |
| Invalid | Valid | Invalid |
| <0 | **>0** | **>30** |

**Password:** Password is used for authenticated user.

|  |  |  |
| --- | --- | --- |
| Invalid | Valid | Invalid |
| <0 | >=8 | **>**25 |

**Class Partition Analysis**

**Seat field:** User can reserve the seats.

|  |  |  |
| --- | --- | --- |
| EC1 | <30 | 1,2,3,4,6,22,30,28 |
| EC2 | >30 | 69,696969,69,70,81 |

**Password:** Password is used for authenticated user.

|  |  |  |
| --- | --- | --- |
| EC1 | >=8 | 8,9,10 |
| EC2 | >25 | 26,27,29 |

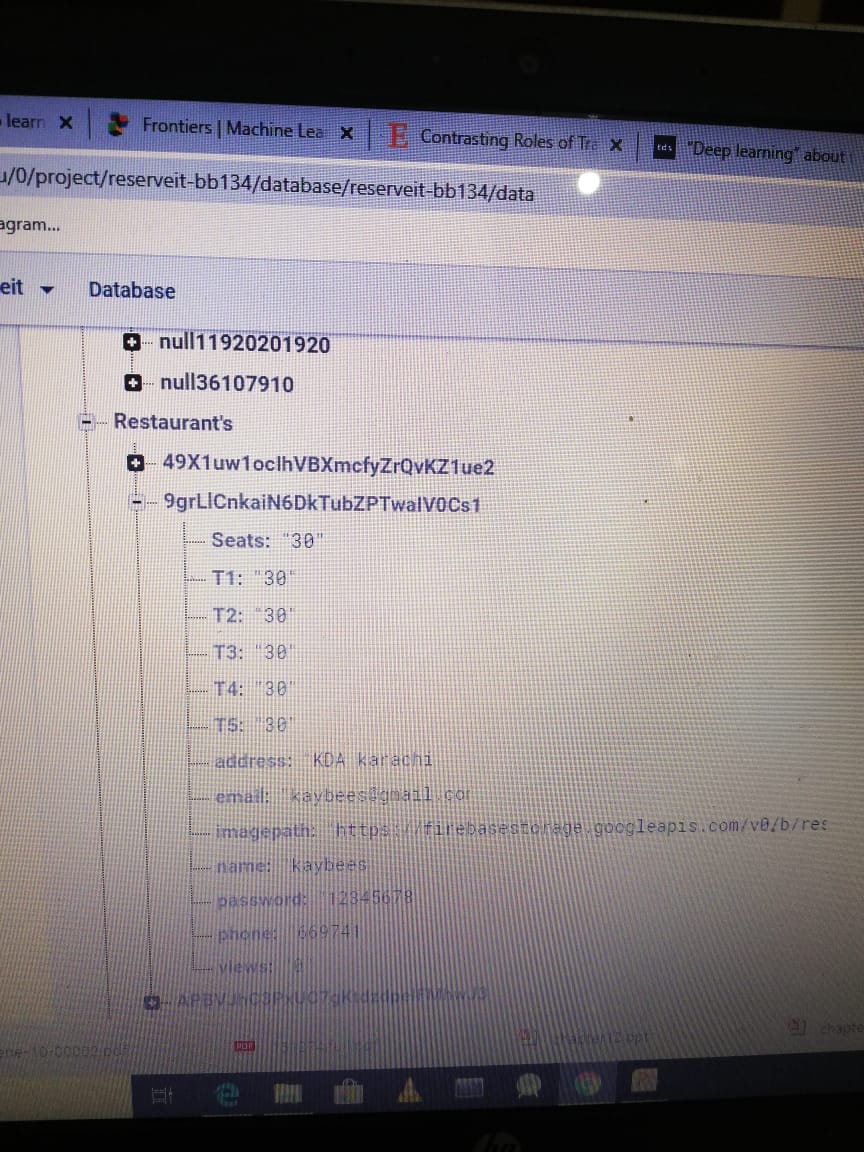
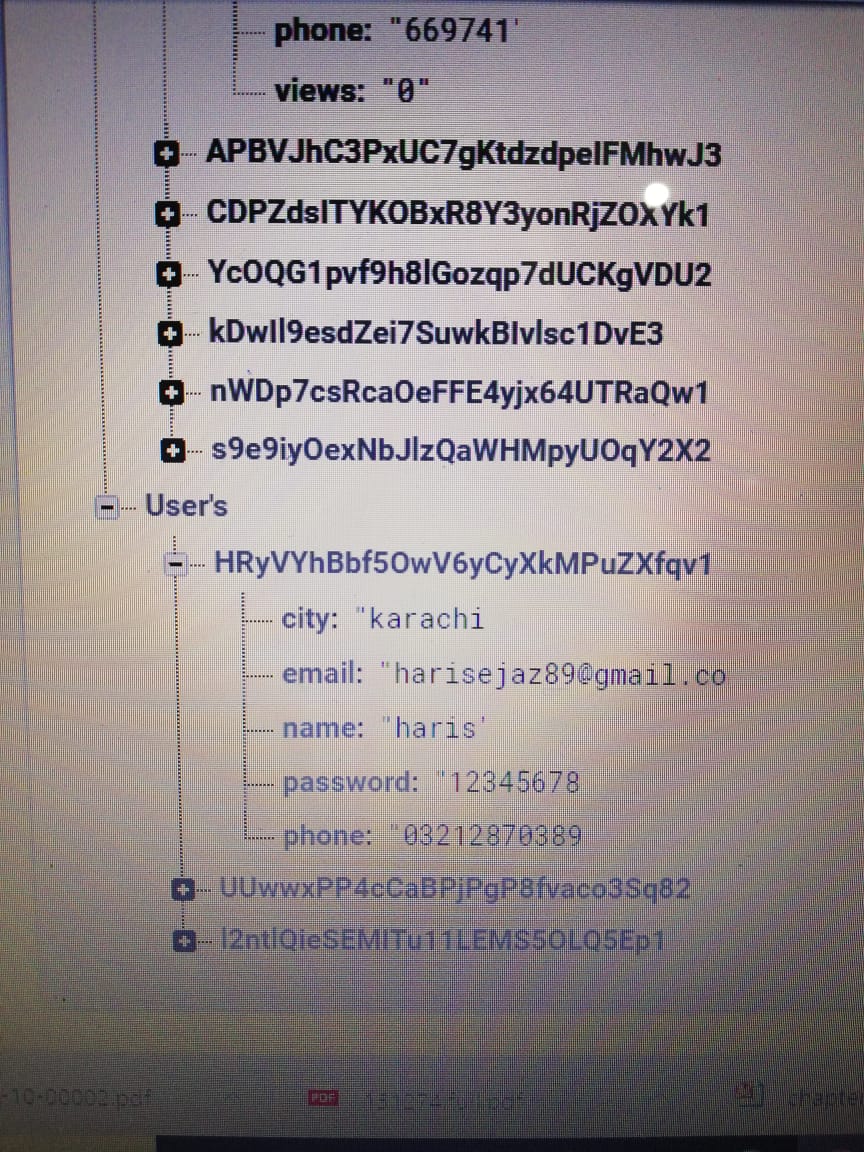
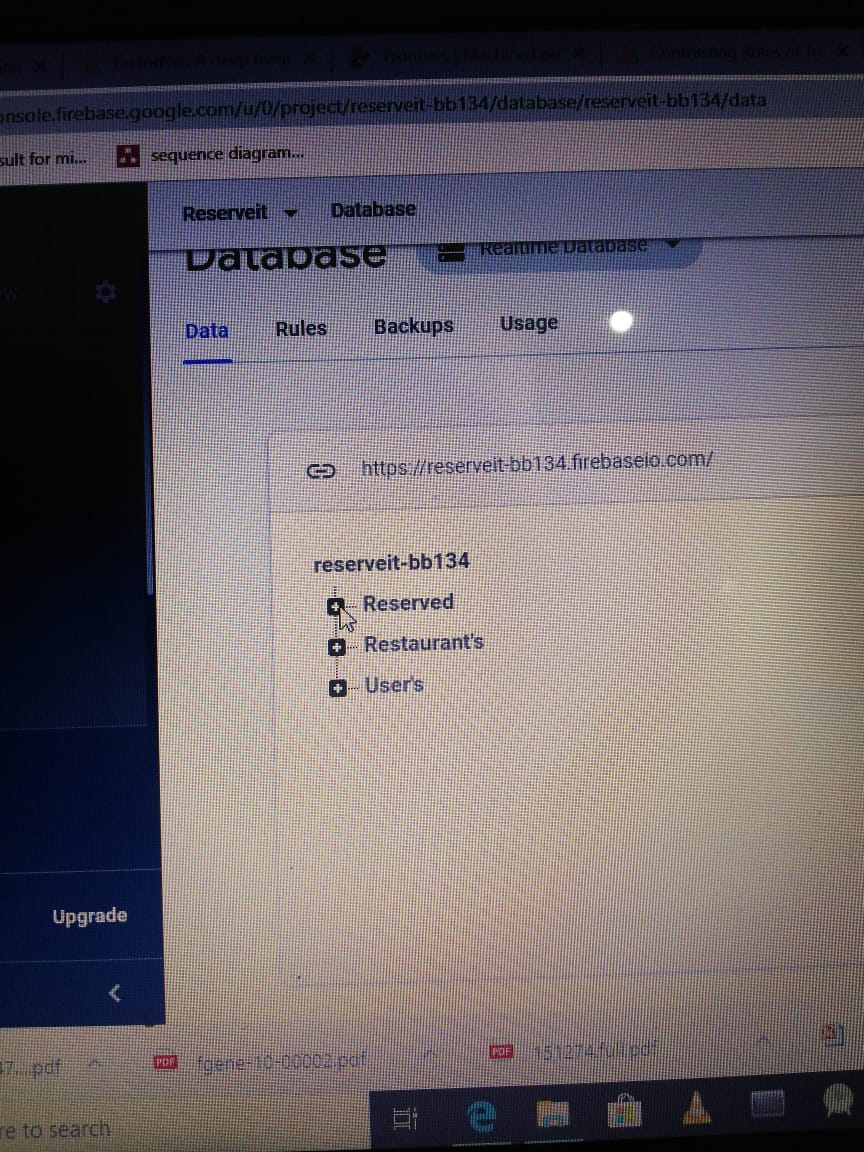
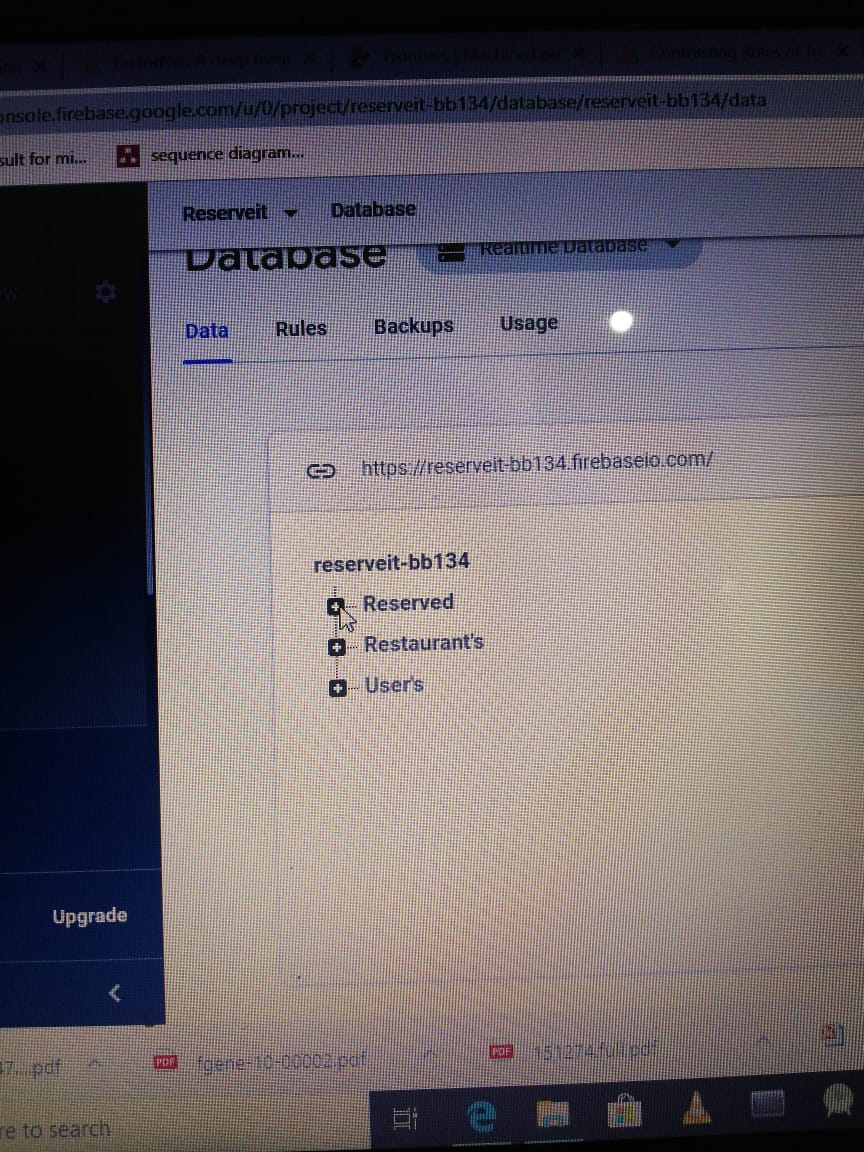
**Acceptance Criteria**

The following user story that fulfill the acceptance criteria are as follows:

* The user will open this application for the first time & and he will sign up and then he will login and then he will select his favorite restaurant and then he will reserve his desire seats and time and he can reserve seats in different restaurant and if he wish he can also cancel the reservation of that particular restaurant
* The user will open this application for the second or several time & and he will sign up and then he will login and then he will select his favorite restaurant and then he will reserve his desire seats and time and he can reserve seats in different restaurant and if he wish he can also cancel the reservation of that particular restaurant
* The user will open this application for the second or several time & and he will sign up and then he will login and then he will select his favorite restaurant and then he will reserve his desire seats and time and close that app
* The restaurant owner can sign up his restaurants with limited seats with location and restaurant names and the user can reserve a seats

**Form Testing, Query Testing & Integrity Testing**

All the form testing has been conducted with the test cases given below and 99% of the test cases were passed. Query Testing & Integrity Testing has been done during the test cases below and also we verified the tables of the database and the pictures are show below:



# Test Cases

|  |  |
| --- | --- |
| **Test case** | **Login screen\_001** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

The user will put all the correct attributes.

|  |  |
| --- | --- |
| **Test case** | **Login screen\_002** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

The user will put invalid password

|  |  |
| --- | --- |
| **Test case** | **Login screen\_003** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

The user will put empty fields

|  |  |
| --- | --- |
| **Test case** | **Login screen\_004** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

The user will put wrong email

|  |  |
| --- | --- |
| **Test case** | **SignUp\_001** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

The user will but valid inputs

|  |  |
| --- | --- |
| **Test case** | **SignUp\_002** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

The user will put invalid inputs

|  |  |
| --- | --- |
| **Test case** | **Rest\_SignUp\_001** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

The user will put valid restaurant sign up

|  |  |
| --- | --- |
| **Test case** | **Rest\_SignUp\_002** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

The user will put invalid inputs.

|  |  |
| --- | --- |
| **Test case** | **Rest\_List \_001** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

App will show the registered restaurant to user

|  |  |
| --- | --- |
| **Test case** | **Res\_seat\_001** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

Users booked seat correctly (put correct information)

|  |  |
| --- | --- |
| **Test case** | **Rest\_List\_002** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

User booked seat with invalid information(test case: seat number is less than reserved seat)

|  |  |
| --- | --- |
| **Test case** | **Rest\_List\_003** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

User has put empty values in form

|  |  |
| --- | --- |
| **Test case** | **Show\_rev\_001** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

The list of restaurant and theirs reserved seats

|  |  |
| --- | --- |
| **Test case** | **Show\_rev\_002** |
| **Test date** | **1/1/20** |
| **Status** | **Tested** |
| **Review Date** | **2/1/20** |
| **Result** | **Pass** |

The user can delete/cancel theirs reservation(delete button)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Steps* | *Verification/Activity* | *Expected result* | *Test 1* | *Test 2* | *Test 3* |
| 1. | User can login to his/her account | * Verifies that ReserveIt application activates * Verify that Login screen is displayed |  |  |  |
| 2. | User continues to the next screen by pressing on the ‘Sign In’,’Sign up’,‘Signup as restaurant ’button | * Verify that Login screen closes * Verify that Entry form screen opens |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 3. | Repeat Step 1 | Same as 1 |  |  |  |
| 4. | User can reserve seat by continues as log in  If user was authenticate | * Verify that List Screen of restaurant closes * Verify that Entry form screen opens |    |    |    |
| 5. | User can fill the form of register as restaurant | * Verify That Login Screen is closed * Verify that entry form open |  |  |  |
| 6. | Same as step 5 for signup but as user | * Same as step 5 but as user |  |  |  |

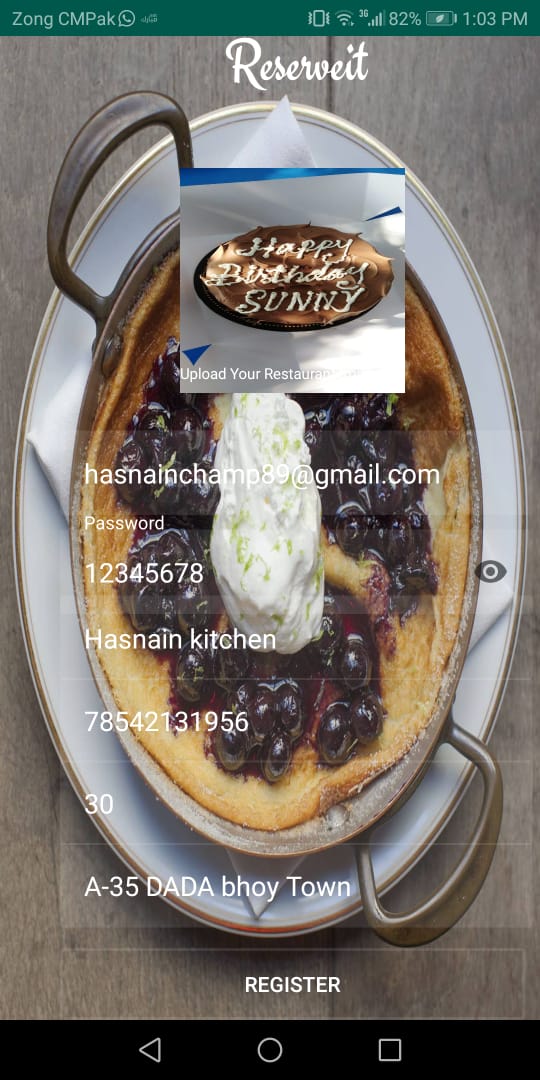
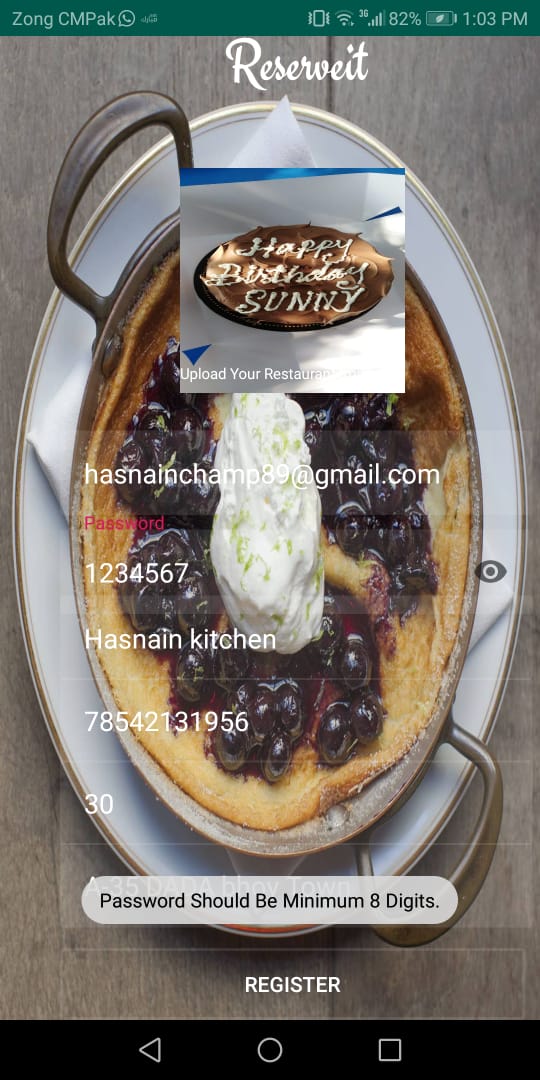
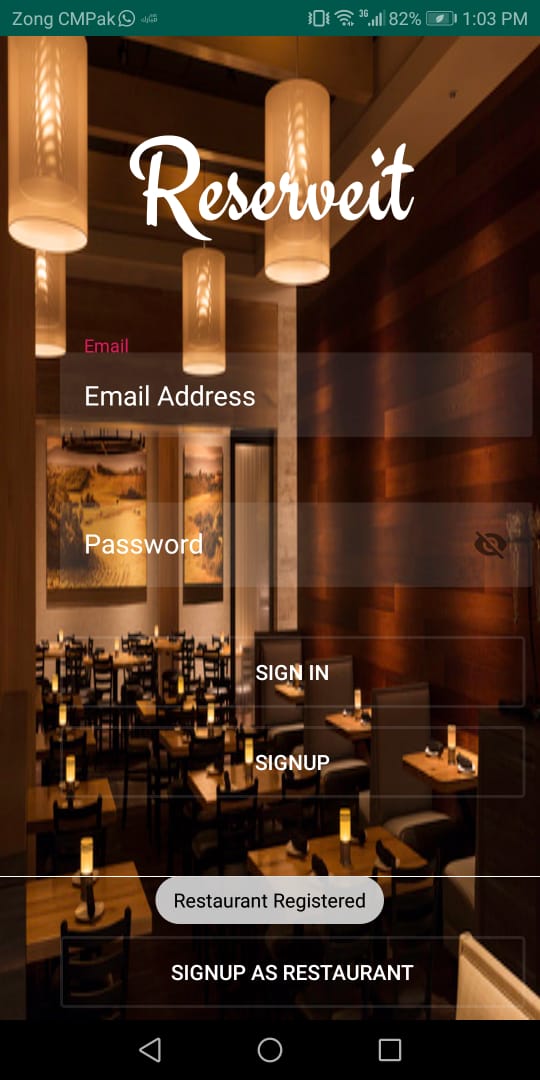
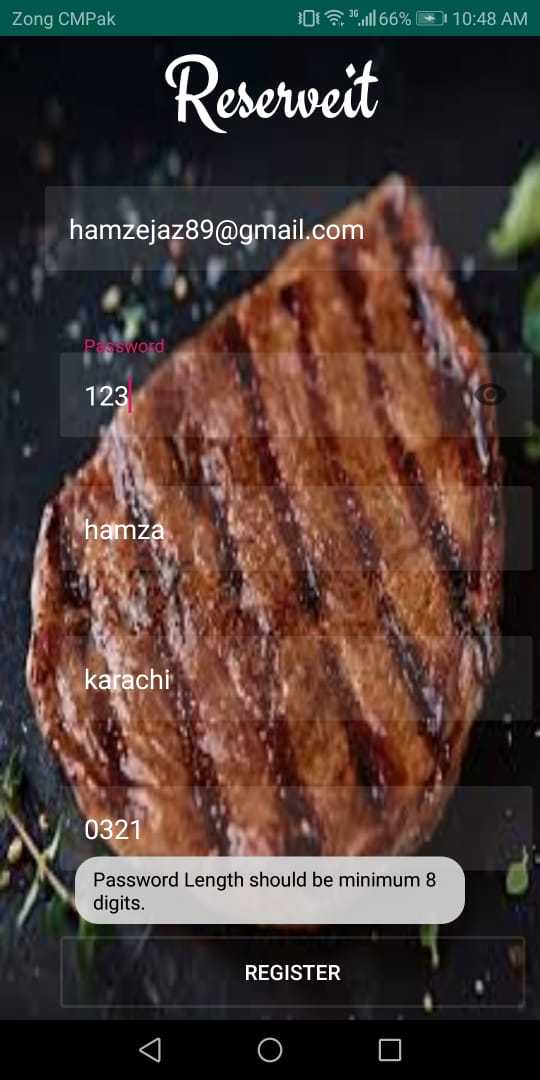
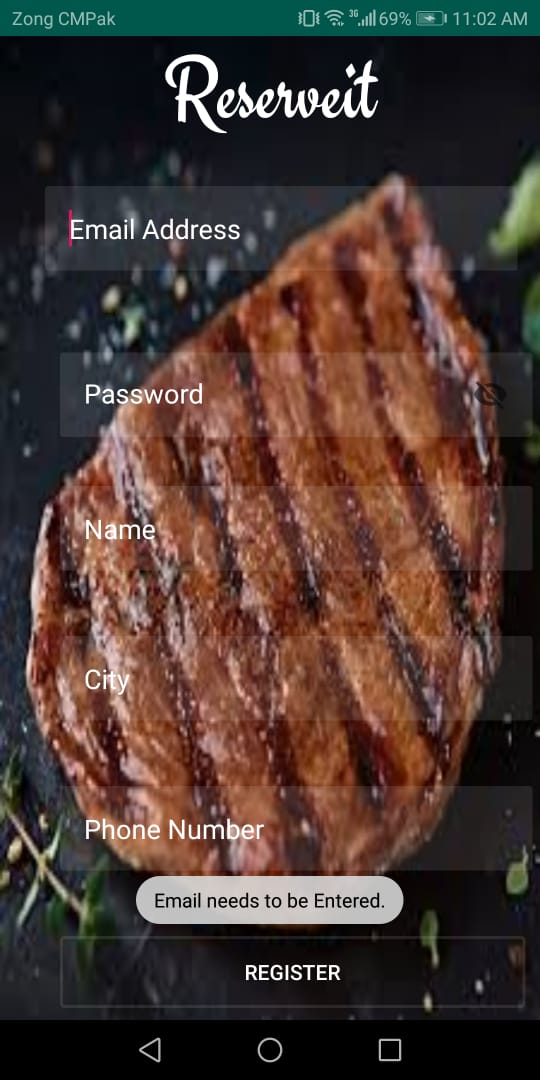
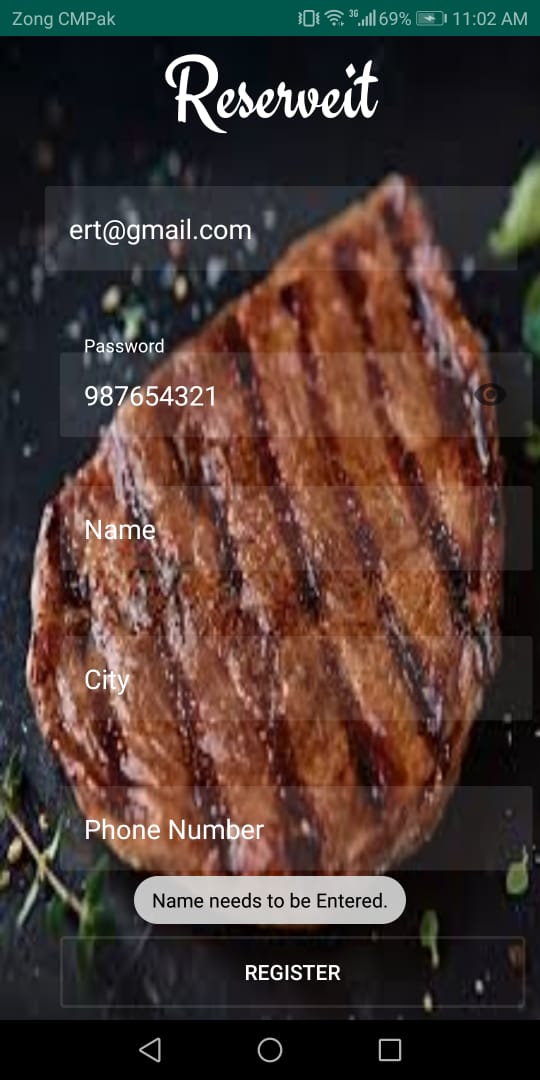
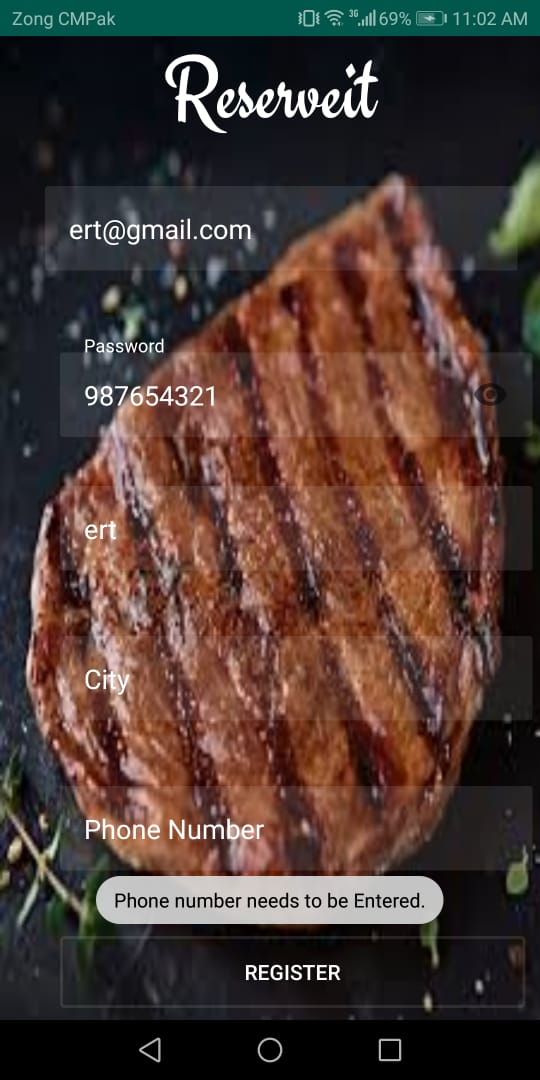
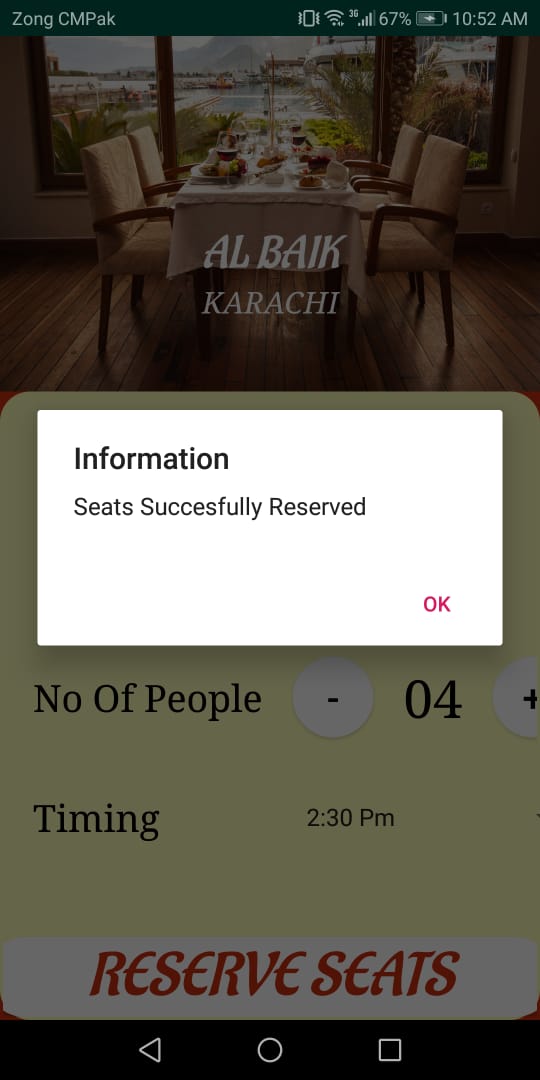
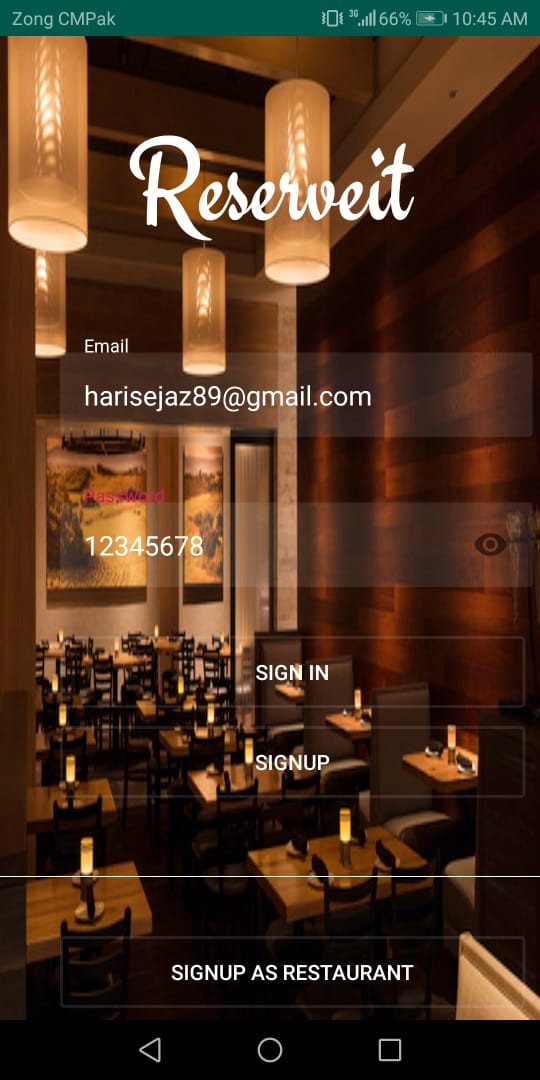
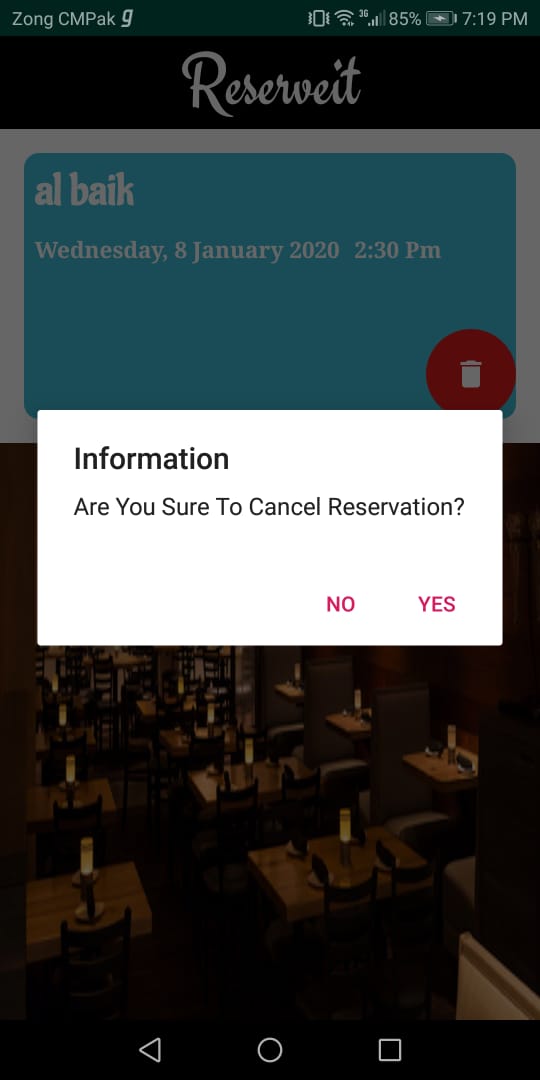
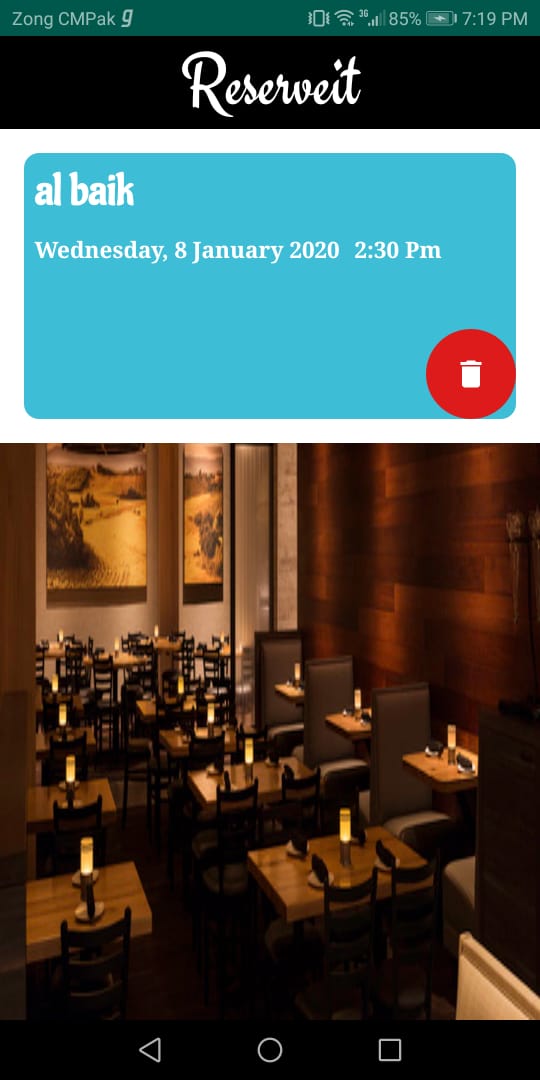
# **Post Condition:**

Login screen opens when the Reseveit application is activated

The login form opens when the user continues tapping to ‘Sign In’,’Sign up’,‘Signup as restaurant ’button

**Issues:**

None.



# User Manual

# Introduction to the System Document

**Purpose of this Document**

The purpose of this user manual is to familiarize the user about the reserveit system.

**Indented readers**

Readers can follow the instruction of user manual to install and use that software.

**System Responsibility**

The system is responsible is to reserve the seat in the selected restaurant and time.

# Installation Instruction

**System Overview**

The system will reserve the seat in your favorite restaurants, and you can cancel your reservation at any time.

**System Requirements**

Quad Core 1.2 GHZ Snap Dragon 225

Android Jelly Bean Operating system

1 GB of RAM, 8GB ROM

**Installation Guide**

1. Open google play store
2. Search ReserveIt
3. Press download
4. The software is downloaded in your system

**Convention Used**

* Font: Times New Roman
* Size: 11
* Heading: 16

**System Navigation**

The following table will illustrate the flow of the system with description & usage.

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow** | **Deliverables** | **Description** | **Usage** |
| 1 | Login/ Logout | It would login and logout the session | Without login the system cannot be used |
| 2 | Select the Restaurant | There will be a list View that would show all the restaurants which were entered in the database | Customer can select a restaurant to make a reservation |
| 3 | Reserve a seat | There will be a timings provided to the customers. | The customer can reserve a seat of their desired time. |

**System Reports & Maintenance**

The system itself does not generate system reports for fixing of error and the user has to manually take the screenshot and email to [cs161xxx@dsu.edu.pk](mailto:cs161xxx@dsu.edu.pk) for error fixing and maintenance and this maintenance will be entertained after two months of deployment of the System.