CSE 5324: SOFTWARE DEVELOPMENT PROJECT

Feedback: Android Application
Design Document

Team Members	Akshay Mattoo	1000995551
	Adithya Chandrashekar	1000990558
	Tejaswini Naidu Karanam	1001015832
	Harish Baghaepalli Vasu	1000989611
	Vidhyadhar Venkatraman	1000988905
Project Guide	Dr. Christoph Csallner	

Revision History:

Date	Version	Description	Author
01/26/2014	1.0	Initial version of Document	All team members
02/02/2014	1.1	Revised version of document	All team members
02/24/2014	1.2	Iteration 1	All team members
03/24/2014	1.3	Iteration 2	All Team members
04/14/2014	1.4	Iteration 3	All Team members
04/28/2014	1.5	Final Deliverables	All Team members

1. Introduction

Purpose of this document is to provide a description of the Feedback App. The document provides the insight into the structure and design of our Application. Topics covered

- Class design and diagram.
- Database design and schema.
- User Interface.

The document is meant to equip the reader to understand the working of the App.

2. Architecture Design

The architectural design is the design of the entire software system; it gives a high-level overview of the software system, such that the reader can more easily follow the more detailed descriptions in the later sections. It provides information on the decomposition of the system into modules (classes), dependencies between modules, hierarchy and partitioning of the software modules.

There are two major components in Feedback: A client side Android application and server side JAVA application and MySQL database designed in PHP. In the client side application, the app receives information about the customer that includes name, email id, date of birth and phone number. The application also receives the information of admin while signing up for feedback given by the customers. The graphical component in feedback application provides with all the buttons, text boxes and other on screen elements that are required, which allows the user to access any feature in the application.

The server component of the application is comprised of JAVA platform, which manages incoming and outgoing messages and MySQL database, which provides centralized storage for synchronized data. The server application receives serialized data from Android devices and converts it into useful data. This data is then stored in the database and subsequently synchronized into other devices in the same group.

Admin Class

The Admin class is the manager who responds to the feedback given by the customers. The Admin decides what to do depending on the rating given by the customer.
The admin will provide name, email id and phone number to sign up. The Admin can also update the information later.

Admin Update Class

After signing up, the Admin can update the information like email id or phone number.

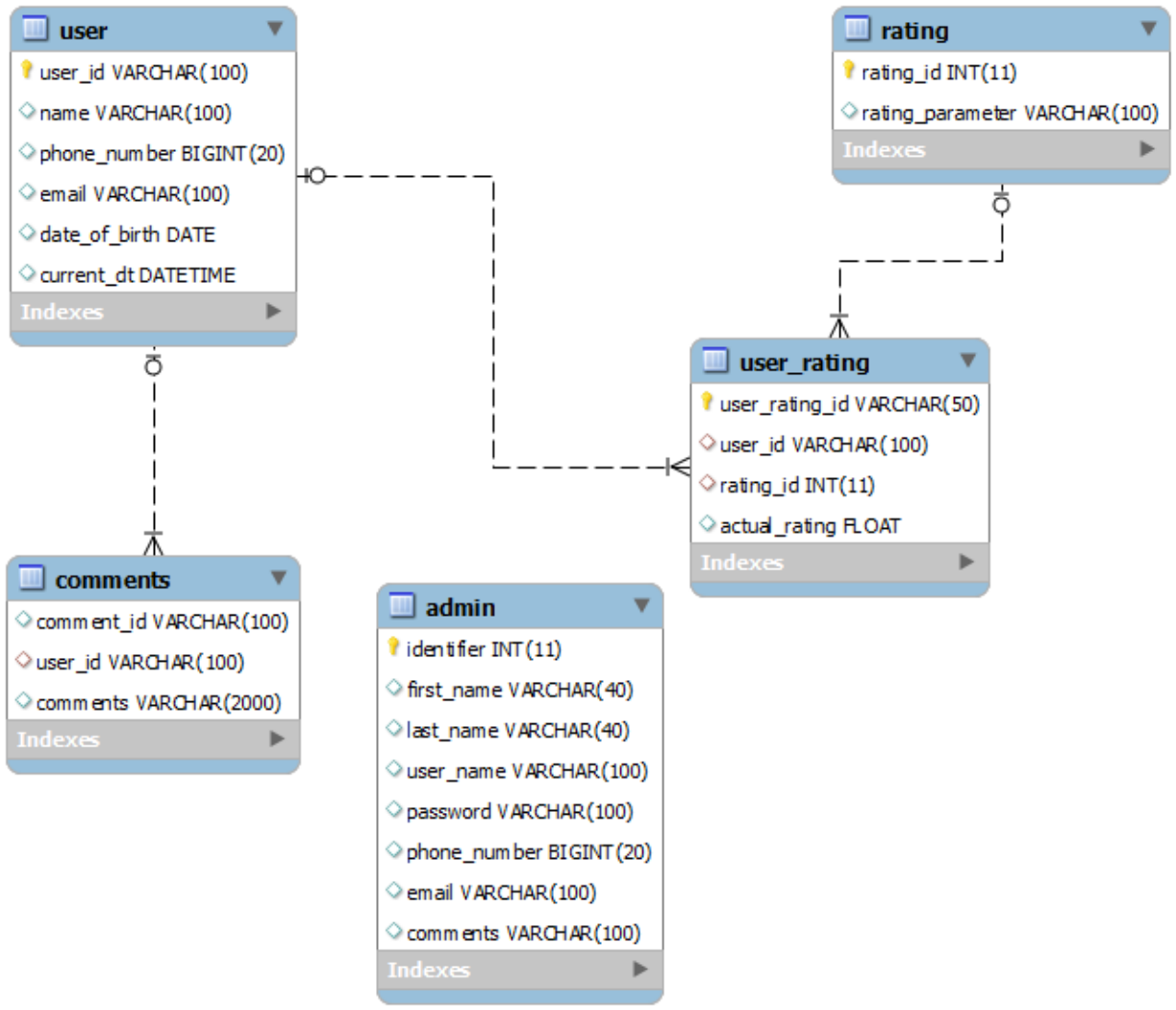
User Class

The user gives his name, email id, phone number and date of birth in order to rate the restaurant. The customer's rate and comment about the restaurant.

VIEW Class

This is the class provided by Android API. This is base for all user interface components. Various graphical elements of Feedback application will interact with this API.

3. Data Base Schema



Data Base Description:

Table Structure for Admin application:

Table Structure for admin:

```
CREATE TABLE `admin` (
  `identifier` int(11) NOT NULL AUTO_INCREMENT,
  `first_name` varchar(40) DEFAULT NULL,
  `last_name` varchar(40) DEFAULT NULL,
  `user_name` varchar(100) DEFAULT NULL,
  `password` varchar(100) DEFAULT NULL,
  `phone_number` bigint(20) DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `comments` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`identifier`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
```

Table Structure for Feedback Application

Table Structure for Rating:

```
CREATE TABLE `rating` (
  `rating_id` int(11) NOT NULL AUTO_INCREMENT,
  `rating_parameter` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`rating_id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
```

Table Structure for Comments:

```
CREATE TABLE `comments` (
  `comment_id` varchar(100) DEFAULT NULL,
  `user_id` varchar(100) DEFAULT NULL,
  `comments` varchar(2000) DEFAULT NULL,
  KEY `user_id` (`user_id`),
  CONSTRAINT `comments_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `user` (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Table Structure for User:

```
CREATE TABLE `user` (
  `user_id` varchar(100) NOT NULL,
  `name` varchar(100) DEFAULT NULL,
  `phone_number` bigint(20) DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `date_of_birth` date DEFAULT NULL,
  `current_dt` datetime ,
  PRIMARY KEY (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Table Structure for User Rating:

```
CREATE TABLE `user_rating` (
  `user_rating_id` varchar(50) NOT NULL,
  `user_id` varchar(100) DEFAULT NULL,
  `rating_id` int(11) DEFAULT NULL,
  `actual_rating` float DEFAULT NULL,
  PRIMARY KEY (`user_rating_id`),
  KEY `rating_id` (`rating_id`),
  KEY `user_id` (`user_id`),
  CONSTRAINT `user_rating_ibfk_1` FOREIGN KEY (`rating_id`) REFERENCES `rating` (`rating_id`),
  CONSTRAINT `user_rating_ibfk_2` FOREIGN KEY (`user_id`) REFERENCES `user` (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Database Structure:

Admin table structure:

admin

Field	Type	Null	Default	Comments	MIME
identifier	int(11)	No			
first_name	varchar(40)	Yes	*NULL*		
last_name	varchar(40)	Yes	*NULL*		
user_name	varchar(100)	Yes	*NULL*		
password	varchar(100)	Yes	*NULL*		
phone_number	bigint(20)	Yes	*NULL*		
email	varchar(100)	Yes	*NULL*		
comments	varchar(4000)	Yes	*NULL*		

Indexes:

Keyname	Type	Cardinality	Field
PRIMARY	PRIMARY	3	identifier

Space usage:

Type	Usage	
Data	340	B
Index	2,048	B
Overhead	104	B
Effective	2,284	B
Total	2,388	B

Row Statistics:

Statements	Value
Format	dynamic
Rows	3
Row length ø	78
Row size ø	796 B
Next Autoindex	19
Creation	Feb 21, 2014 at 06:01 PM
Last update	Feb 28, 2014 at 01:49 PM

User table structure:

user

Table comments: InnoDB free: 48128 kB

Field	Type	Null	Default	Comments	MIME
user_id	varchar(100)	No			
name	varchar(100)	Yes	*NULL*		
phone_number	bigint(20)	Yes	*NULL*		
email	varchar(100)	Yes	*NULL*		
date_of_birth	date	Yes	*NULL*		
current_dt	datetime	No			

Indexes:

Keyname	Type	Cardinality	Field
PRIMARY	PRIMARY	4	user_id

Space usage:

Type	Usage	
Data	16,384	B
Index	0	B
Total	16,384	B

Row Statistics:

Statements	Value
Format	Compact
Rows	4
Row length ø	4,096
Row size ø	4,096 B
Creation	Mar 23, 2014 at 06:31 PM

Rating table structure:

rating

Table comments: InnoDB free: 48128 kB

Field	Type	Null	Default	Comments	MIME
rating_id	int(11)	No			
rating_parameter	varchar(100)	Yes	*NULL*		

Indexes:

Keyname	Type	Cardinality	Field
PRIMARY	PRIMARY	4	rating_id

Space usage:

Type	Usage	
Data	16,384	B
Index	0	B
Total	16,384	B

Row Statistics:

Statements	Value
Format	Compact
Rows	4
Row length ø	4,096
Row size ø	4,096 B
Next Autoindex	5
Creation	Mar 23, 2014 at 03:11 PM

User ratings table structure:

user_rating

Table comments: InnoDB free: 48128 kB; (`rating_id`) REFER `axm5553/rating`(`rating_id`); (`user

Field	Type	Null	Default	Links to	Comments	MIME
user_rating_id	varchar(50)	No				
user_id	varchar(100)	Yes	*NULL*	user -> user_id		
rating_id	int(11)	Yes	*NULL*	rating -> rating_id		
actual_rating	float	Yes	*NULL*			

Indexes:

Keyname	Type	Cardinality	Field
PRIMARY	PRIMARY	12	user_rating_id
rating_id	INDEX	12	rating_id
user_id	INDEX	6	user_id

Space usage:

Type	Usage
Data	16,384 B
Index	32,768 B
Total	49,152 B

Row Statistics:

Statements	Value
Format	Compact
Rows	12
Row length ø	1,365
Row size ø	4,096 B
Creation	Mar 23, 2014 at 03:11 PM

Comments Structure:

comments

Table comments: InnoDB free: 48128 kB; (`user_id`) REFER `axm5553/user`(`user_id`)

Field	Type	Null	Default	Links to	Comments	MIME
comment_id	varchar(100)	Yes	*NULL*			
user_id	varchar(100)	Yes	*NULL*	user -> user_id		
comments	varchar(2000)	Yes	*NULL*			

Indexes:

Keyname	Type	Cardinality	Field
user_id	INDEX	3	user_id

Space usage:

Type	Usage	
Data	16,384	B
Index	16,384	B
Total	32,768	B

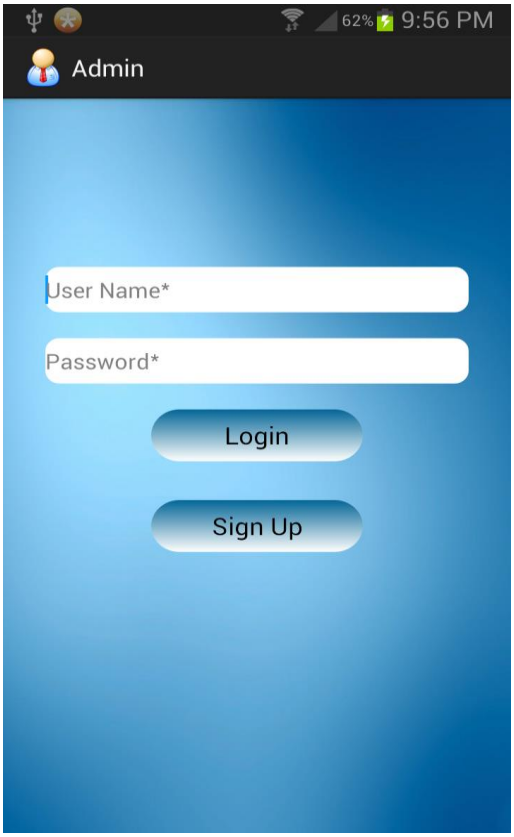
Row Statistics:

Statements	Value
Format	Compact
Rows	3
Row length ø	5,461
Row size ø	10,923 B
Creation	Mar 23, 2014 at 03:11 PM

4. User Interface

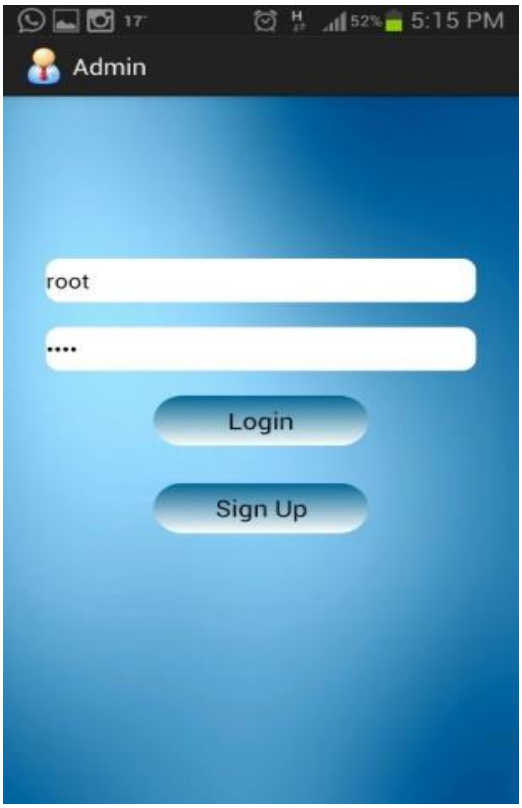
- Provide, in an organized way, the pictures of all the forms in the graphical user interface with a reference to the functional requirement it implements. For each form in the graphical user interface, provide:

Admin Login Screen:



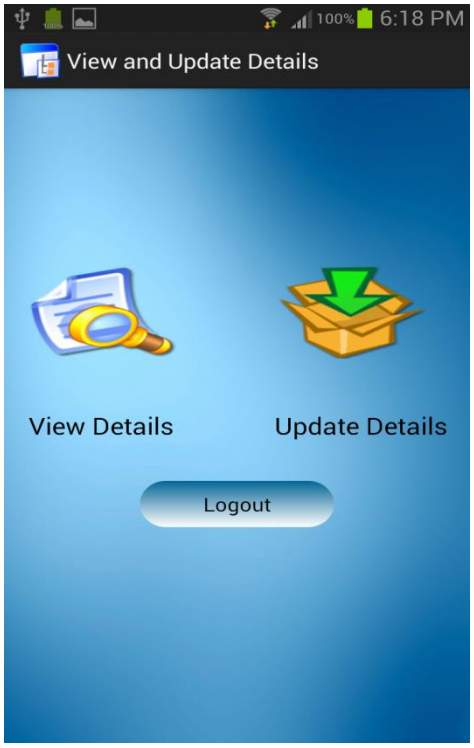
- This screen provides the interface for the admin to login
- The Admin has to provide a valid User Name and Password to Login
- If the admin doesn't have an account, then he can sign up clicking Sign Up button

Admin Logs in:



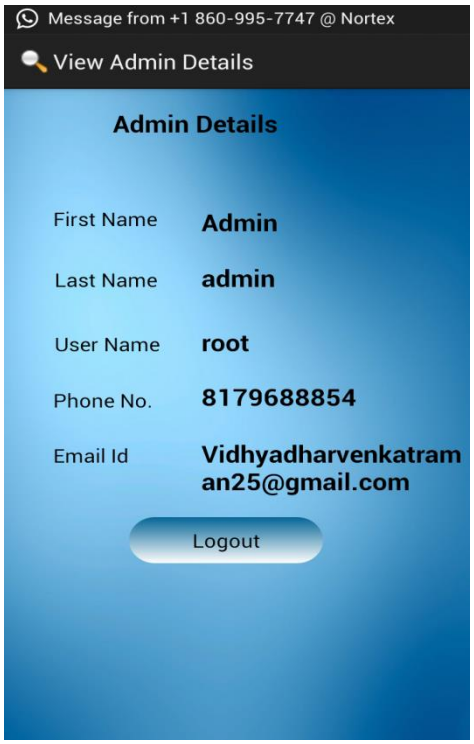
- The Admin provides the User Name as root and a password
- Then he clicks on the Login button

View and Update Details:



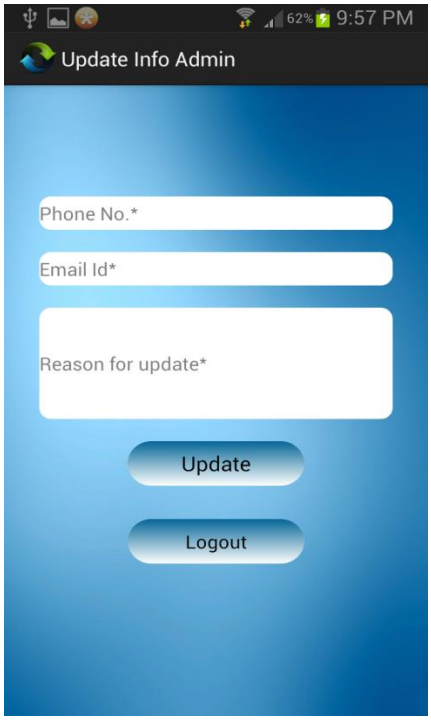
- The Admin can View details by clicking on View Details
- He can Update Details by clicking on Update Details.
- The Admin can also Logout, whenever he wanted.

View Details:



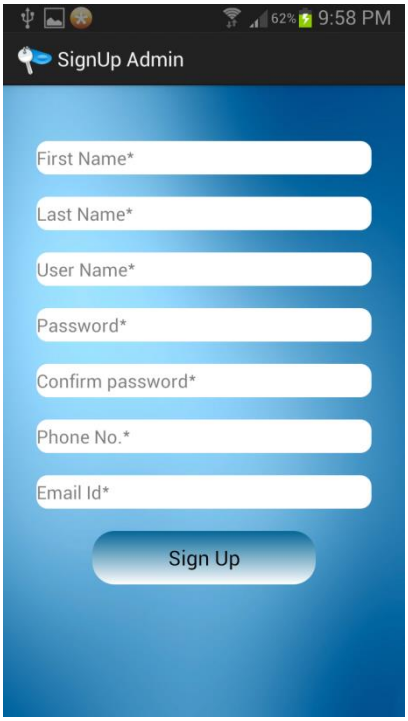
- The page displays the details of the admin
- First Name, Last Name, User Name, Phone No, Email ID can be viewed.
- After the Admin has viewed his details he can Logout by clicking on the Logout button.

Update Details:



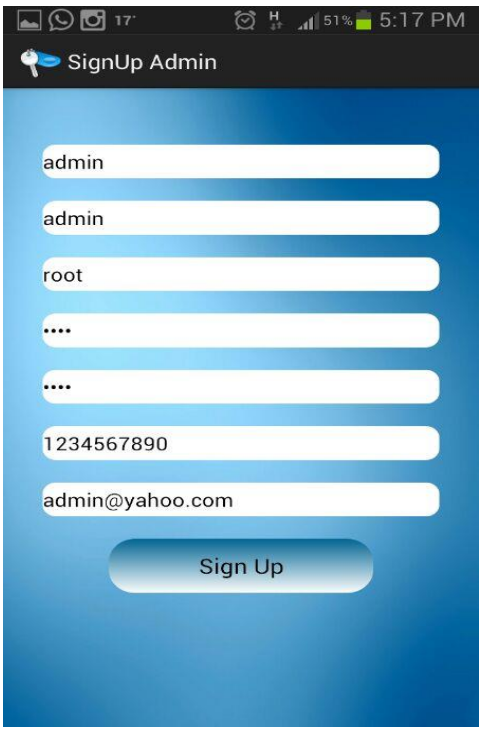
- The Admin can update his details.
- He can provide a new Phone No and Email Id
- He has to provide the reason for update
- After providing the details he can update the information by clicking on update.
- The Admin can Logout by clicking on the logout button.

Signup Details:



- This is the Sign Up page for Admin
- If the Admin doesn't have an account, then he has to provide his First Name, Last Name, User Name, Password of his choice, confirm that password, Phone no and Email Id
- After he has provided the required information, he can Sign up by clicking on Sign up button

Admin Signing Up:

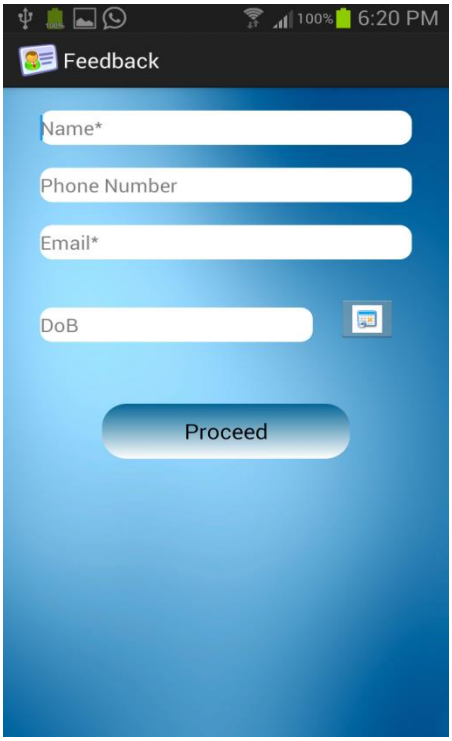


- The Admin provides his First Name as admin, Last Name as admin
- He provides the Username as root
- He provides a password and repeats the password in confirm password
- He provides the Phone no as 1234567890
- He provides the Email ID as admin@yahoo.com
- After providing the details, he can Sign Up by clicking on Sign Up

Splash Screen:

- A splash page is created at the beginning of the user app.
- It also contains a voice host which says “Welcome to Feedback app. Your feedback is valuable”.

User Information:

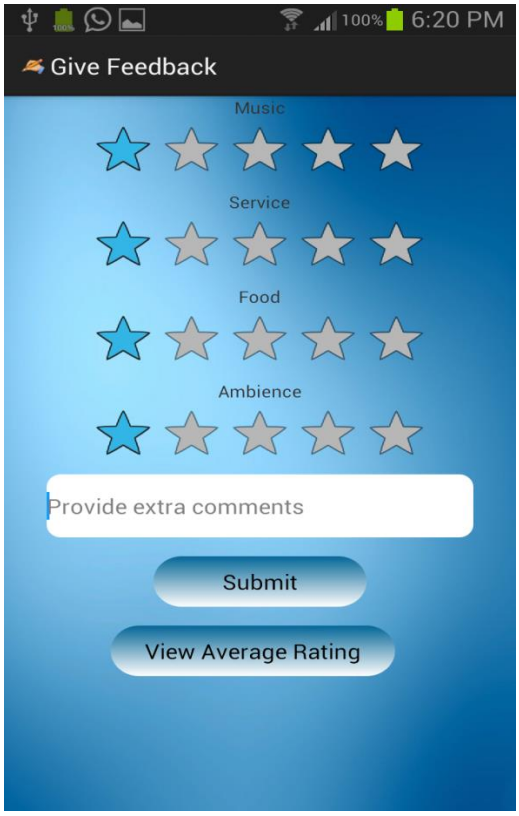


- The User Enters his Name, Phone No, Email Id and DOB and clicks on proceed
- The Name are Email Id are mandatory

User Details:

- The User enters his name as john, Phone no as 6529854123, email ID as john@gmail.com and DOB as 1989-11-19 (yyyy-mm-dd)
- After clicking on proceed it proceeds to the Ratings page

Ratings Page:

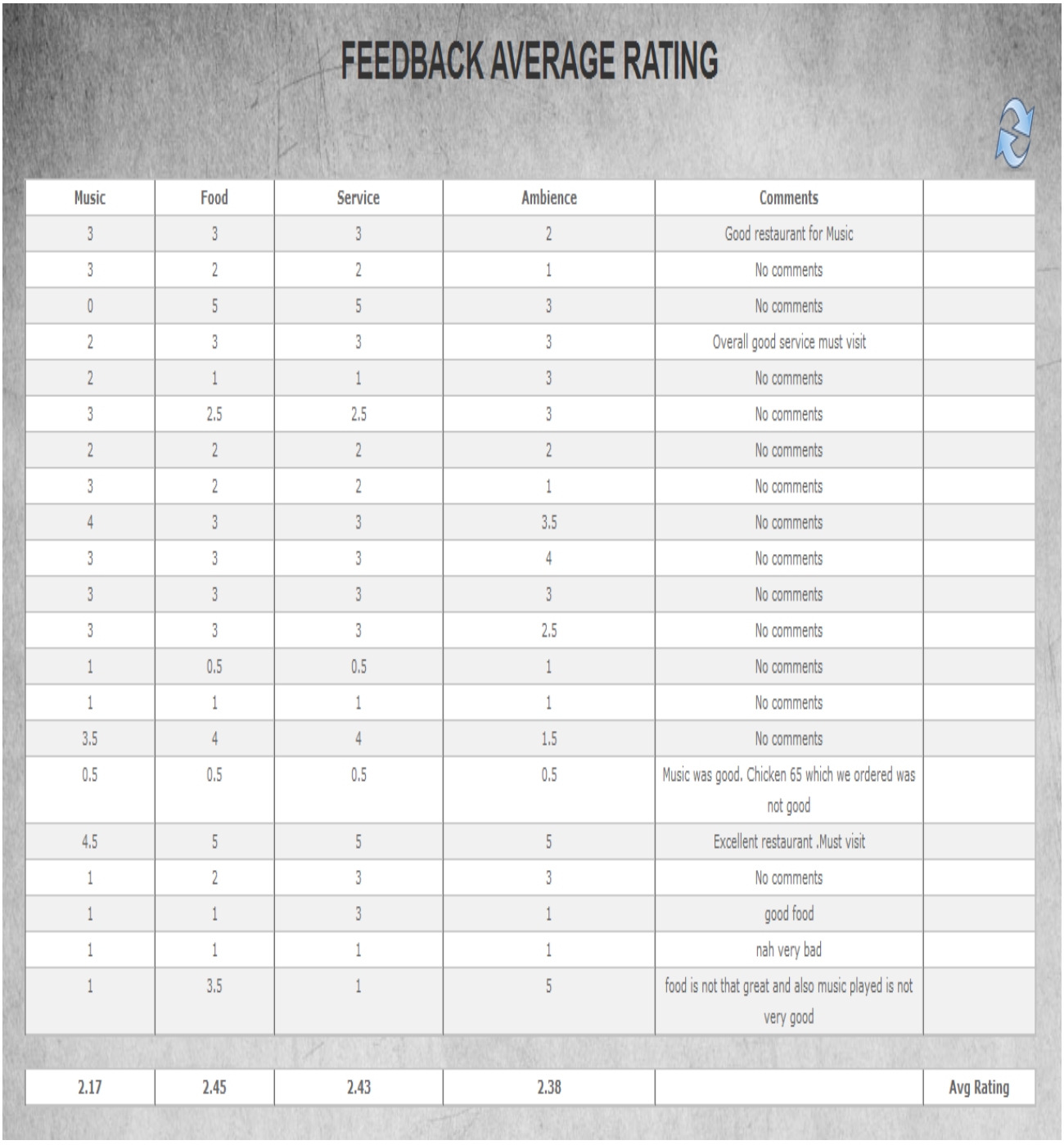


- The User can rate on the above mentioned features
- Giving ratings are mandatory
- The ratings are from 1 to 5 and by default it is set to 1
- A button “View Average Rating” is provided which helps in viewing Average Ratings Page.

Ratings Given:

- The User can give ratings as mentioned above
- If the ratings are below 2, then a message is send to the admin
- Comments section is also provided, however it is optional
- A button “View Average Rating” is provided which by clicking helps the user to view the webpage which includes average ratings and comments.

Feedback Ratings Web page:



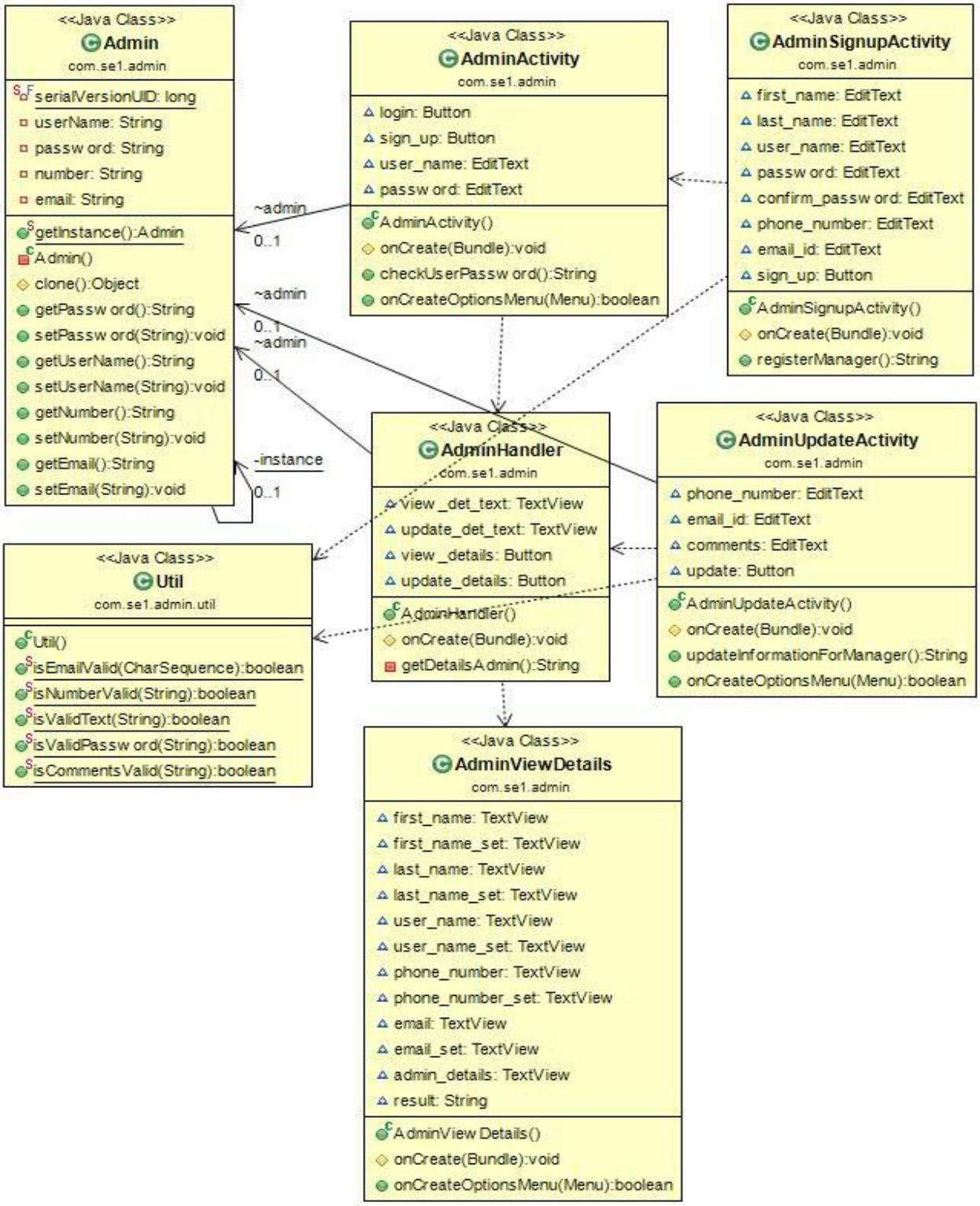
FEEDBACK AVERAGE RATING

Music	Food	Service	Ambience	Comments	
3	3	3	2	Good restaurant for Music	
3	2	2	1	No comments	
0	5	5	3	No comments	
2	3	3	3	Overall good service must visit	
2	1	1	3	No comments	
3	2.5	2.5	3	No comments	
2	2	2	2	No comments	
3	2	2	1	No comments	
4	3	3	3.5	No comments	
3	3	3	4	No comments	
3	3	3	3	No comments	
3	3	3	2.5	No comments	
1	0.5	0.5	1	No comments	
1	1	1	1	No comments	
3.5	4	4	1.5	No comments	
0.5	0.5	0.5	0.5	Music was good. Chicken 65 which we ordered was not good	
4.5	5	5	5	Excellent restaurant .Must visit	
1	2	3	3	No comments	
1	1	3	1	good food	
1	1	1	1	nah very bad	
1	3.5	1	5	food is not that great and also music played is not very good	
2.17	2.45	2.43	2.38		Avg Rating

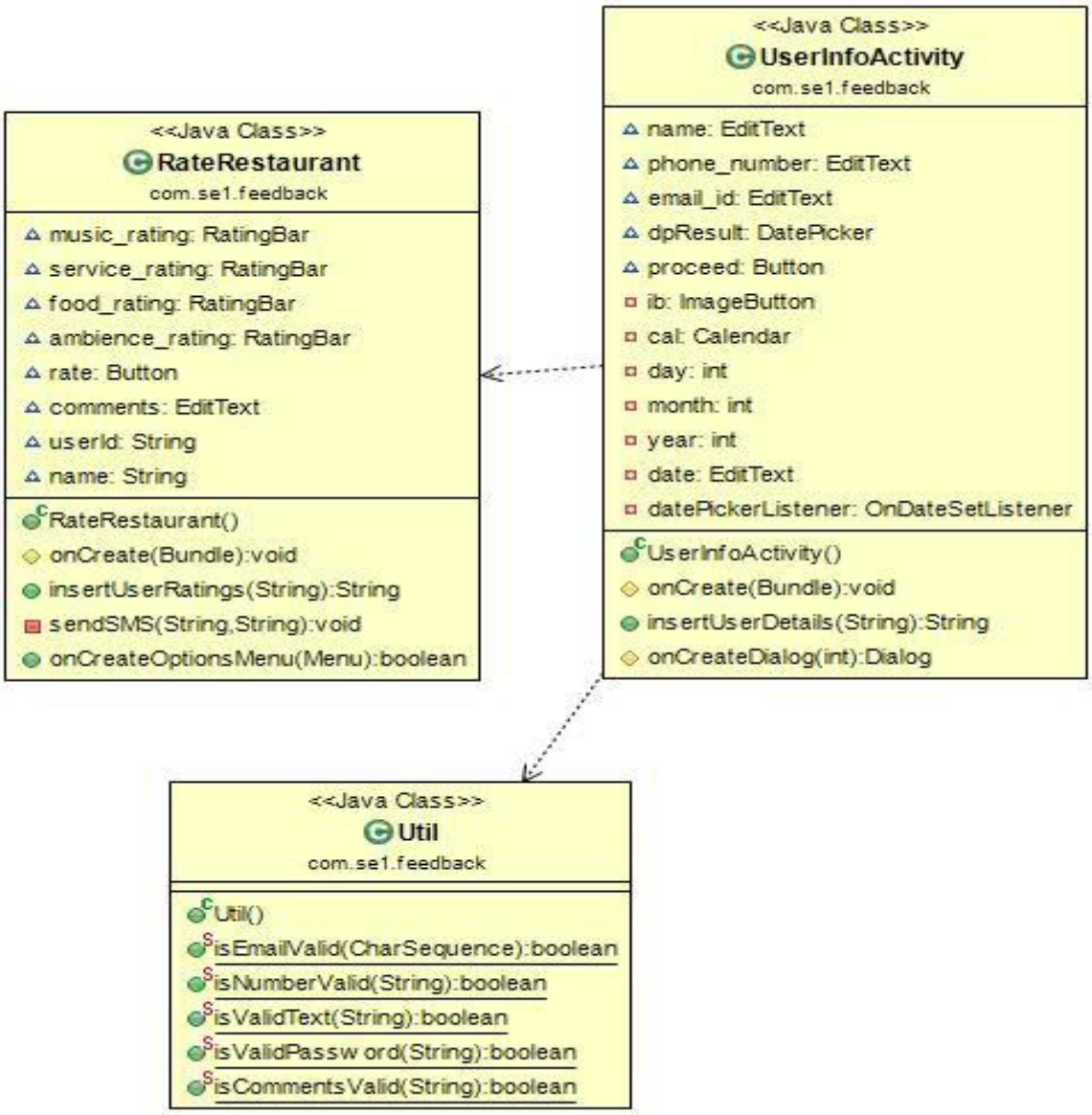
- Open the webpage by clicking the url. http://omega.uta.edu/~axm5553/admin/averagerating.php
- This page shows the average rating of the all rating given till date.

5. Class Diagram:

Admin Class Diagram



Feed Back Class Diagram



Code Snippets:

Android code snippets:

1: The below code snippet is used to connect to a URL from the android.

```
HttpClient httpclient = new DefaultHttpClient();
HttpPost httppost = new HttpPost(
"http://omega.uta.edu/~axm5553/admin/insertManager.php");
List<NameValuePair> pairs = new ArrayList<NameValuePair>();


pairs.add(new BasicNameValuePair("firstname",String.valueOf(first_name.getText())));


httppost.setEntity(new UrlEncodedFormEntity(pairs));
HttpResponse response = httpclient.execute(httppost);
HttpEntity entity = response.getEntity();

InputStream is = entity.getContent();

BufferedReader reader = new BufferedReader(
new InputStreamReader(is, "iso-8859-1"), 8);
StringBuilder sb = new StringBuilder();
String line = null;
while ((line = reader.readLine()) != null) {
sb.append(line + "\n");
}

is.close();
result =sb.toString();
Log.i("INFO", result);
} catch (Exception e) {
Log.e("Error", e.toString());
}
```

2: The below code is used to send a SMS from an android program.

```
 SmsManager sms = SmsManager.getDefault();
 sms.sendTextMessage(phoneNumber, null, message, null, null);
```

3: The below code is used for Singleton Pattern.

```
private static final long serialVersionUID = 1L;
private static Admin instance = new Admin();

//static method to return the instance of the class.
public static Admin getInstance() {
  return instance;
}
//Private constructor to prevent the object to be made from outside class.
   private Admin()
   {

   }
protected Object clone() throws CloneNotSupportedException {
throw new CloneNotSupportedException("Clone Not supported Exception");
}
```

4: Passing objects between Android.

```
Intent intent = new Intent(context, AdminHandler.class);
intent.putExtra("AdminObject", admin);

Admin admin = (Admin) getIntent().getSerializableExtra("AdminObject");
```

PHP Code snippets

1: The below code is used to connect to the database through PHP

```
$con=mysqli_connect("localhost","root","root","Feedback");
if (mysqli_connect_errno())
 {
 echo "Failed to connect to MySQL: " . mysqli_connect_error();
 }


$query="INSERT into table values (value1,value2,……)“;

$result = mysqli_query($con,$query,MYSQLI_USE_RESULT);


mysqli_close($con);
```

2: The below code is for displaying the table with the help of tablecloth.js library .

```
<script type="text/javascript" charset="utf-8">
     $(document).ready(function() {
       $("table").tablecloth({
         theme: "stats",
         striped: true,
         sortable: true,
         condensed: true
       });
     });
   </script>
```

3: The below code is the references that we have to put for the. css and .js files

```
  <link href="assets/css/bootstrap.css" rel="stylesheet">
   <link href="assets/css/bootstrap-responsive.css" rel="stylesheet">
   <link href="assets/css/tablecloth.css" rel="stylesheet">
   <link href="assets/css/prettify.css" rel="stylesheet">
   <script src="assets/js/jquery-1.7.2.min.js"></script>
   <script src="assets/js/bootstrap.js"></script>
   <script src="assets/js/jquery.metadata.js"></script>
   <script src="assets/js/jquery.tablesorter.min.js"></script>
   <script src="assets/js/jquery.tablecloth.js"></script>
```

References:

- http://www.mysql.com/
- http://developer.android.com/
- http://www.websequencediagrams.com/
- http://findicons.com/
- http://stackoverflow.com/
- http://www.php.net/manual/en/
- https://creately.com/
- http://tableclothjs.com/
- http://jquery.com/
- http://getbootstrap.com/
- http://newboston.com/
- https://www.google.com/