THE NATIONAL COLLEGE

AUTONOMOUS

BASAVANAGUDI, BANGALORE – 560 004

॥ ಶ್ರದ್ಧಾಹಿ ಪರಮಾ ಗತಿಃ ॥

ANDROID PROJECT WORK

"ONLINE CATERING MANAGEMENT SYSTEM"

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF POSTGRADUATE DEGREE

MASTER OF SCIENCE IN COMPUTER SCIENCE

BY

ARCHANA. H. K (19NCBPGCS03)
ARPITHA. G. M (19NCBPGCS04)

UNDER THE GUIDANCE OF

Prof. RASHMI M. M.Sc.

DEPARTMENT OF COMPUTER SCIENCE
THE NATIONAL COLLEGE
AUTONOMOUS
BASAVANAGUDI, BANGALORE-04
YEAR 2020

Acknowledgement

The constant guidance and encouragement received from Prof. Rashmi. M **Lecturer Department of Computer Science** has been of great help in carrying out the project work and is acknowledgment with reverential thanks.

We would like to express a deep sense of gratitude and thanks profusely to Prof. Ravi Hegde **HOD Department of Computer Science, The National College** without their wise counsel and able guidance, it would have been impossible to complete the report in this manner.

Finally, I am indebted to all who have contributed in this report work and its success.

ARCHANA. H. K
ARPITHA. G. M

INDEX

INTRODUCTION

1. INTRODUCTION

1.1 Introduction to Project:

Online Catering Management System is the mobile application which is used as a medium between the user and catering service. It will help user to choose the best rated catering service from which the user is able to book particular catering service for those who search top catering services online.

The online catering management system is a bridge between the user and the catering services to get the particular type of service booked from any part of the world.

Caterers are growing, as they are able to manage all the parties by themselves and handle all the risks. This mobile application will help caterer to manage catering services for particular events. By using this application, the customer should signup first and then login by entering their email and password. After login is successful, the customer gets the event options to choose like wedding, birthday, house warming, baby shower, etc.

Suppose if the customer chooses birthday event, they will get the list of best top catering services with ratings and location. Ones the customer selects any catering service they will be navigated to the event details form where they are supposed to fill the event details which includes venue, date, time, etc. After submitting the details, they will be able to choose the buffet along with decoration theme.

The catering service will be booked for the particular date and time. The customer can do UPI payment or cash on delivery. Next, they can give anonymous feedback and checkout. Also, the admin can add new catering services by inserting new data.

1.2 Objective:

The proposed system's objectives are to overcome all the limitations and drawbacks of the existing system. The **Online Catering Management System** is user-friendly android application. The main objective of the application is its simplicity of design and ease of implementation that shows and helps the customer to request catering service online and get service by nearby selected catering service.

The main objectives of the proposed system can be enumerated as follows:

- Gives customer to book catering service by selecting nearby catering service made available.
- Provides caterer a customer to earn money.
- This application provides the communication bridge between the Customer and the catering service in the form of booking.

1.3 Scope:

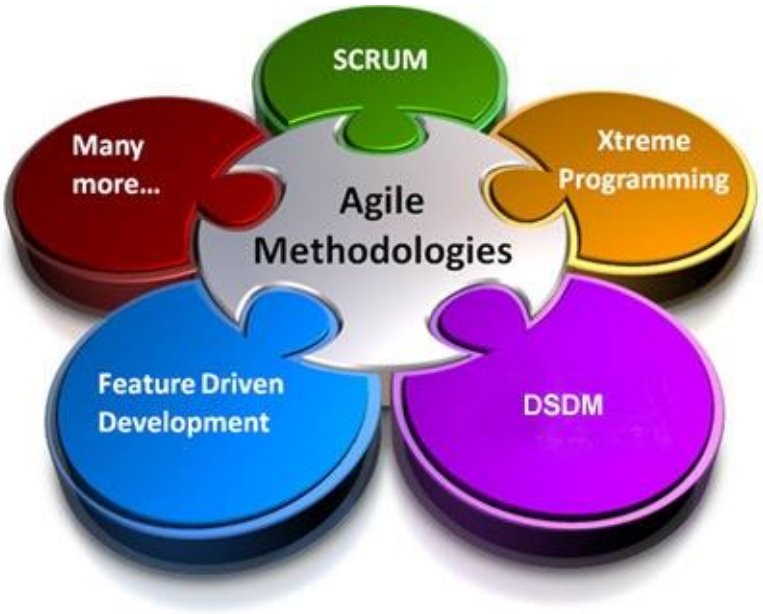
Booking catering services online by the customer, responding customer online by the catering service is well designed in this application and it makes satisfaction to the customer by providing service on the particular event and the caterer will provide the service for a customer by going to the mentioned venue on particular date and time.

1.4 Methodology:

Agile Methodology:

What is Agile?

The Agile movement proposes alternatives to traditional project management. Agile approaches are typically used in software development to help businesses respond to unpredictability.



What is Scrum?

Scrum is an agile framework for developing, delivering, and sustaining complex products, with an initial emphasis on software development, although it has been used in other fields including research, sales, marketing and advanced technologies. It is designed for teams of ten or fewer members, who break their work into goals that can be completed within timeboxed iterations, called sprints, no longer than one month and most commonly two weeks. The Scrum Team track progress in 15-minute time-boxed daily meetings, called daily scrums. At the end of the sprint, the team holds sprint review, to demonstrate the work done, and sprint retrospective to continuously improve.

A typical scrum team has between five and nine people, but Scrum projects can easily scale into the hundreds. However, Scrum can easily be used by one-person teams and often is. This team does not include any of the traditional software engineering roles such as programmer, designer, tester or architect. Everyone on the project works together to complete the set of work they have collectively committed to complete within a sprint. Scrum teams develop a deep form of camaraderie and a feeling that "we're all in this together."

Product owner: The product owner is the project's key stakeholder and represents users, customers and others in the process. The product owner is often someone from product management or marketing, a key stakeholder or a key user.

Scrum Master: The Scrum Master is responsible for making sure the team is as productive as possible. The Scrum Master does this by helping the team use the Scrum process, by removing impediments to progress, by protecting the team from outside, and so on.

Product backlog: The product backlog is a prioritized features list containing every desired feature or change to the product. Note: The term "backlog" can get confusing because it's used for two different things. To clarify, the product backlog is a list of desired features for the product. The sprint backlog is a list of tasks to be completed in a sprint.

Sprint planning meeting: At the start of each sprint, a sprint planning meeting is held, during which the product owner presents the top items on the product backlog to the team. The Scrum team selects the work they can complete during the coming sprint. That work is then moved from the product backlog to a sprint backlog, which is the list of tasks needed to complete the product backlog items the team has committed to complete in the sprint.

Daily Scrum: Each day during the sprint, a brief meeting called the daily scrum is conducted. This meeting helps set the context for each day's work and helps the team stay on track. All team members are required to attend the daily scrum.

Sprint review meeting: At the end of each sprint, the team demonstrates the completed functionality at a sprint review meeting, during which, the team shows what they accomplished during the sprint. Typically, this takes the form of a demonstration of the new features, but in an informal way; for example, PowerPoint slides are not allowed. The meeting must not become a task in itself nor a distraction from the process.

Sprint retrospective: Also at the end of each sprint, the team conducts a sprint retrospective, which is a meeting during which the team (including its ScrumMaster and product owner) reflect on how well Scrum is working for them and what changes they may wish to make for it to work even better.

Each of the Scrum terms has its own page within the Scrum section, so be sure to check out all the pages in the navigation.

Where Did Agile Come From?

You hear a lot about "agile innovation" these days. Teams using agile methods get things done faster than teams using traditional processes. They keep customers happier. They enjoy their work more. Agile has indisputably transformed software development, and many experts believe it is now poised to expand far beyond IT.

Ironically, that's where it began outside of IT.

Some trace agile methodologies all the way back to Francis Bacon's articulation of the scientific method in 1620. A more reasonable starting point might be the 1930s, when the physicist and statistician Walter Shewhart of Bell Labs began applying Plan-Do-Study-Act (PDSA) cycles to the improvement of products and processes. Shewhart taught this iterative and incremental-development methodology to his mentee, W. Edwards Deming, who used it extensively in Japan in the years following World War II. Toyota hired Deming to train hundreds of the company's managers, eventually capitalizing on his expertise to develop the famous Toyota Production System — the primary source of today's "lean" thinking. Iterative and incremental development methods were also a major contributor to the successful creation of the X-15 hypersonic jet in the 1950s.

In 1986, one of us (Takeuchi) and coauthor Ikujiro Nonaka published an article in Harvard Business Review called "The New Product Development Game." Studying manufacturers that were releasing successful innovations far faster than competitors, the authors identified a team-oriented approach that changed the design and development process for products such as copiers at Fuji-Xerox, automobile engines at Honda, and cameras at Canon. Rather than following conventional "relay race" methods of product development — in which one group of functional specialists hands off its completed phase to the next functional stage — these companies were using what Takeuchi and Nonaka called a "rugby" approach, "where a team tries to go the whole distance as a unit, passing the ball back and forth."

In 1993, another of us (Sutherland) faced what seemed like an impossible task: Easel Corporation, a software company, needed to develop a new product to replace its legacy offerings in less than six months. Sutherland already had a strong background in methodologies such as rapid application development, object-oriented design, PDSA cycles, and skunkworks. He hoped to create a skunkworks-like culture in the middle of corporate headquarters, blending the benefits of both organizational separation and integration. So, he began by learning everything he could about maximizing organizational productivity. Reading hundreds of papers and interviewing leading product-management experts, he found himself intrigued by several provocative ideas.

One came from a Bell Labs article on the Borland Quattro Pro team, suggesting that short daily team meetings increased group productivity dramatically. But the capstone concept for Sutherland was the discovery of Takeuchi's and Nonaka's rugby approach, even though it focused on manufacturing rather than software. Borrowing many of the HBR article's key ideas and filling in specific operational practices, Sutherland created a new way of developing software; honouring the rugby imagery, he dubbed his approach "scrum." Scrum methods enabled him to finish his seemingly impossible project on time, under budget, and with fewer bugs than any previous release. He then collaborated with long-time colleague Ken Schwaber to codify the approach, and in 1995 the pair presented scrum to the public for the first time.

Of course, Sutherland and Schwaber weren't alone in their search for innovative methods. The Information Age was exploding. Disruptive technologies were terrorizing slow-footed competitors. Start-ups and incumbents alike sought better ways to adapt to the unfamiliar and turbulent environment. Software was becoming an integral part of nearly every business function, and many creative software developers were working hard on better methods of programming to increase adaptability.

In 2001, 17 developers who called themselves "organizational anarchists" met in Snowbird, Utah, to share their ideas. Sutherland and other proponents of scrum were among them. But the group included advocates of several competitive approaches, including extreme programming (XP); crystal; adaptive software development (ASD); feature-driven development (FDD); and the dynamic-systems-development method (DSDM). All these approaches were often known as "lightweight" frameworks because they used fewer, simpler rules to allow faster adaptation to rapidly changing environments. Not many of the attendees found the "lightweight" terminology flattering.

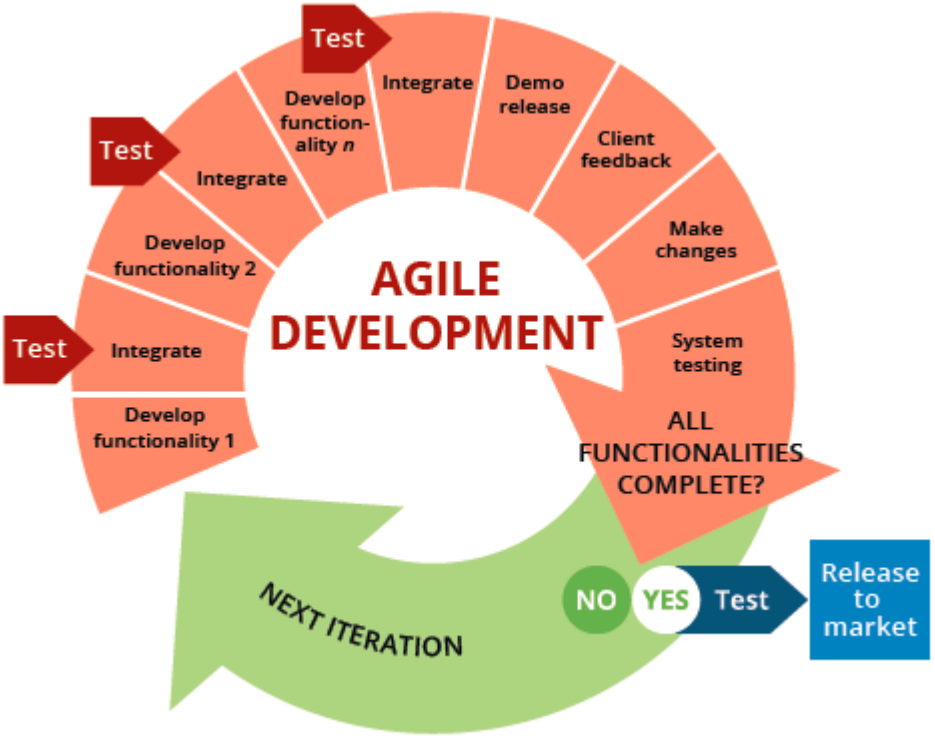
Why Agile?

Agile iterative and incremental nature allows companies to focus on the main things and not forget the product or services they are looking to launch – while allowing development teams to adapt their approach as they go.

Typically, two-to-four-week sprints contrast sharply with timescales of traditional project management that stretched into months and even years. This often resulted as the end product being obsolete before it was ready.

There are many benefits to using Agile. Here are **10 reasons why PM teams use this flexible project management process**:

- Agile looks evolutionary. It gives teams an opportunity to learn with each new iteration or draft.
- Agile allows teams to deliver a prototype and improve it with every cycle.
- Agile supports regular and collaborative troubleshooting.
- Agile helps teams and individuals effectively prioritize features and work in general.
- Teams can make quick-course corrections based on stakeholder feedback.
- Team members may prototype a solution or process for the next project's version.
- Teams get rapid feedback from each version or iteration.
- Agile empowers team members to work creatively and effectively.
- Stakeholders and clients can provide feedback as the project evolves.
- This flexible process increases team's productivity.



LITERATURE SURVEY

2.LITERATURE SURVEY

2.1 Existing System:

The Existing system was a Manual system. The Customer itself should go to nearby hotels or restaurants to order food for particular number of members which would take longer time and it consumes more energy and money. For each item customer should go to different shops in case of decorations. If customer choose to cook by himself, it takes longer time to cook where ingredients can be left off. He alone cannot serve food for number of members who attend the party.

Problems of Existing System:

- It consumes more time and money.
- It takes customer longer to order.
- Expensive.
- Complicated to handle.
- Customer should go to nearby shops by his own vehicle.

2.2 Proposed System:

The web server (SQLite Database) is a database server which makes the application real, and can access anywhere in the world. This mobile application help customers to order and book catering services for different events for particular date and time online.

Features and Benefits:

- Less time consuming.
- Configurable and Extensible Application UI Design.
- Providing Security.
- Basic computer knowledge required.

The proposed application can be used even by the native users and it requires minimum educational level, experience and technical expertise in computer field but it will be good if the user has the good knowledge of how to operate an application.

2.3 Feasibility Study:

A feasibility study is a short, focused study, which aims to answer a number of questions:

- Does the system contribute to the overall objectives of the organizations?
- Can the system be implemented using current technology and within given cost and schedule constraints?
- Can the system be integrated with systems which are already in place?

2.3.1 Technical Feasibility:

- Is the project feasibility within the limits of current technology?

- Does the technology exist at all?
- It is available within given resource constraints(schedule).

2.3.2 Operational Feasibility:

Define the urgency of the problem and acceptability of any solution; if the system is developed, will it be used? Includes people-oriented and social issues: internal issues, such as man power problem, labour objection, manager resistance, organizational conflicts and policies; also, external issues including social acceptability, legal aspect sand government regulations.

2.3.3 Financial Feasibility:

- Is the project possible, given resource constraints?
- Are the benefits that will accrue from the new system worth the cost?
- What are the savings that will result from the system, including tangible and intangible ones?
- What are the development and operational costs?

2.4 Tools And Technologies Used

2.4.1 Java

Java is a very popular programing language developed by Sun Microsystems (now owned by Oracle).

Developed long after C and C++, Java incorporates many of the powerful features of those powerful languages while addressing some of their drawbacks. Still, programming languages are only as powerful as their libraries. These libraries exist to help developers build applications.

Some of the Java's important core features are:

It is easy to learn and understand.

It is designed to be platform-independent and secure, using virtual machines.

Its object-oriented.

Android relies heavily on these Java fundamentals. The Android SDK includes many standard java libraries (data structure libraries, math libraries, graphics libraries, networking libraries and everything else you could want) as well as special Android libraries that will help you develop awesome Android applications.

Platform Independence Importance with many programing languages, you need to use a compiler to reduce your code down into machine language that the device can understand. While this is well and good, different devices use different machine languages. This means that you might need to compile your applications for each different device or machine language in other words, your code is not very portable. This is not the case with Java. The Java compliers convert your code from human readable Java source files to something called byte code in the Java world. These are interpreted by a Java Virtual Machine, which operates much like a physical CPU might operate on machine code, to actually execute the complied code. Although it might seem like this is inefficient, much effort has been put into making this process very fast and efficient. These efforts have paid off in that Java performance in generally second only to C/C++ in common language performance comparisons.

Android applications run in a special virtual machine called the Dalvik VM. While the details of this VM are unimportant to the average developer, it can be helpful to think of the Dalvik VM as a bubble in which your Android application runs, allowing you to not have to worry about whether the device is a Motorola Droid, an HTC Evo, or the latest toaster running Android. You don't care so long as the device is Dalvik VM friendly and that is the device manufacturers job to implement, not yours.

Why is Java Secure?

Let us take this bubble idea a bit further. Because Java applications run within the bubble that is a virtual machine, they are isolated from the underlying device hardware. Therefore, a virtual machine can encapsulate, contain, and manage code execution in a safe manner compared to languages that operate in machine code directly. The Android platform takes things a step further. Each Android application runs on the (Linux- based) operating system using a different user account and in its own instance of the Dalvik VM. Android applications are closely monitored by the operating system and shut down if they don't play nice (e.g. use too much processing power, become unresponsive, waste resources, etc.).

Therefore, it is important to develop applications that are stable and responsive. Applications can communicate with one another using well-defined protocol.

2.4.2 PhP

Although most web developers have to a range of scripting languages, such as CGI, ASP, JPS and Perl, most tend to favor PHP. There are several reasons why this programming language is in the forefront of website development.

1. Easy and Simple to Learn

PHP is considered one of the easiest scripting languages. Compared to other web languages, PHP doesn’t require a manual or intensive studying. PHP syntax is logical and well-organized. Even command functions are easy to understand, as they tell the developer what

function they perform. As a result, web developers find it very easy to create and optimize the application.

2. Extremely Flexible

PHP is highly flexible whether it is during an ongoing project or after completing the project. Flexibility in a scripting language is very crucial, as functionality can change anytime during the course of a project. The best part about PHP is the ability to make changes even after starting the project and this saves valuable time.

A developer does not have to write fresh codes or command functions, as changes to the existing codes and functions can be done and used.

3. Easy Integration and Compatibility

PHP is compatible with a large majority of operating systems. It can easily run on different platforms, including UNIX, Solaris and Linux. As it can be integrated without effort with other technologies, such as Java, existing software does not require re-development. This saves time and money.

4. Efficient Performance

Depending on how the web developer codes, PHP has the potential to turn in an efficient language. It is scalable when used for writing codes and can also be used for creating a large number of applications. It is the programming language of choice when a website has several webpages.

5. Cost-Efficient

PHP is an open-source web language, hence is completely free. There is no expense involved in purchasing expensive licenses or software. It can work efficiently with different databases, such as MySQL, Apache, and PostgreSQL. The cost of developing a website using PHP is minimal.

6. Gives Web Developer More Control

Compared to other programming languages, PHP allows the website developer to have more control. Other programming languages are bogged down by long, complicated scripts, but this isn't true for PHP. A few simple lines of code are sufficient. Furthermore, PHP allows tags, and hence, website developers can add and/or mix HTML tags, making the content extremely dynamic.

Developers don't have to worry about placing codes in the right place when using PHP, as it is written between tags. Hence, functions and codes do not have to be written in any specific order, as long as they are within the tags.

PHP has a very helpful, active, and widespread PHP community. Also, this scripting language offers a lot of resources, such as commands, functions, and codes, which can easily be rewritten and used without incurring any cost. The ease of use, easy integration, cost efficiency, and easy access makes PHP one of the most popular server-side programming languages

2.4.2 Android Development Tools

Android SDK:

The Android Software Development Kit (Android SDK) contains the necessary tools to create, compile and package Android applications. Most of these tools are command line based. The primary way to develop Android applications is based on the Java programming language.

Android debug bridge (ADB):

The Android Software Development Kit (Android SDK) contains the necessary tools to

create, compile and package Android applications. Most of these tools are command line based. The primary way to develop Android applications is based on the Java programming language.

Android debug bridge (ADB):

The Android SDK contains the Android debug bridge (ADB), which is a tool that allows you to connect to a virtual or real Android device, for the purpose of managing the device or debugging your application.

Android Developer Tools and Android Studio:

Google provides two integrated development environments (IDEs) to develop new applications. The Android Developer Tools (ADT) are based on the Eclipse IDE. ADT is a set of components (plugins), which extend the Eclipse IDE with Android development capabilities. Google also supports an IDE called Android Studio for creating Android applications. This IDE is based on the Intellij IDE. Both IDEs contain all required functionality to create, compile, debug and deploy Android applications.

They also allow the developer to create and start virtual Android devices for testing. Both tools provide specialized editors for Android specific files. Most of Android's configuration files are based on XML. In this case these editors allow you to switch between the XML representation of the file and a structured user interface for entering the data. Dalvik Virtual Machine.

The Android system uses a special virtual machine, i.e., the Dalvik Virtual Machine (Dalvik) to run Java based applications. Dalvik uses a custom bytecode format which is different from java bytecode.

Android RunTime (ART):

Therefore, you cannot run Java class files on Android directly; they need to be converted into the Dalvik bytecode format.

With Android 4.4, Google introduced the Android RunTime (ART) as optional runtime for Android 4.4. It is expected that versions after 4.4 will use ART as default runtime. ART uses Ahead OfTime compilation. During the deployment process of an application on an Android device. The application code is translated into machine code. This results in approx. 30% larger compile code, but allows faster execution from the beginning of the application.

2.4.3 Security and Permission Concept in Android

Security Concept in Android:

The Android system installs every Android application with a unique user and group ID. Each application file is private to this generated user, e.g., other applications cannot access these files. In addition, each Android application is started in its own process. Therefore, by

means of the underlying Linux kernel, every Android application is isolated from other running applications. If data should be shared, the application must do this explicitly via an Android component which handles the sharing of the data, e.g., via a service or a content provider.

Permission concept in Android:

Android contains a permission system and predefines permissions for certain tasks. Every application can request required permissions and also define new permissions. For example, an application may declare that it requires access to the Internet.

Permissions have different levels. Some permissions are automatically granted by the Android system, some are automatically rejected. In most cases the requested permissions are present to the user before installing the application. The user needs to decide if these permissions shall be given to the application.

If the user denies a required permission, the related application cannot be installed. The check of the permission is only performed during installation, permissions cannot be denied or granted after the installation.

An Android application declares the required permissions in its AndroidManifest.xml configuration file. It can also define additional permissions which it can use to restrict access to certain components.

2.5 Hardware and software Requirements

2.5.1 Software Interface

Client on Internet	:	Android Operating System.
Data Base Server	:	SQLite Database.
Development End	:	Java, Android XML File, Android Manifest File, Php File.
Front End	:	Java, Android XML, Android Manifest File.
Back End	:	SQLite.

2.5.2 Hardware Interface

Operating System	:	Android Operating System.
Processor	:	Dual Core 1.2GHz(Any).
RAM	:	1GB
Memory	:	4GB
Others	:	High Speed Internet

SYSTEM REQUIREMENT SPECIFICATION

3.SYSTEM REQUIREMENT SPECIFICATION

3.1 Functional Requirements

The system must provide the following functionalities

- Admin can login and add new catering services.
- User can register and login.
- User can book events from particular catering service for particular date and time.
- User can do payment from upi or cash on delivery.
- Storing the bookings of customer in database.

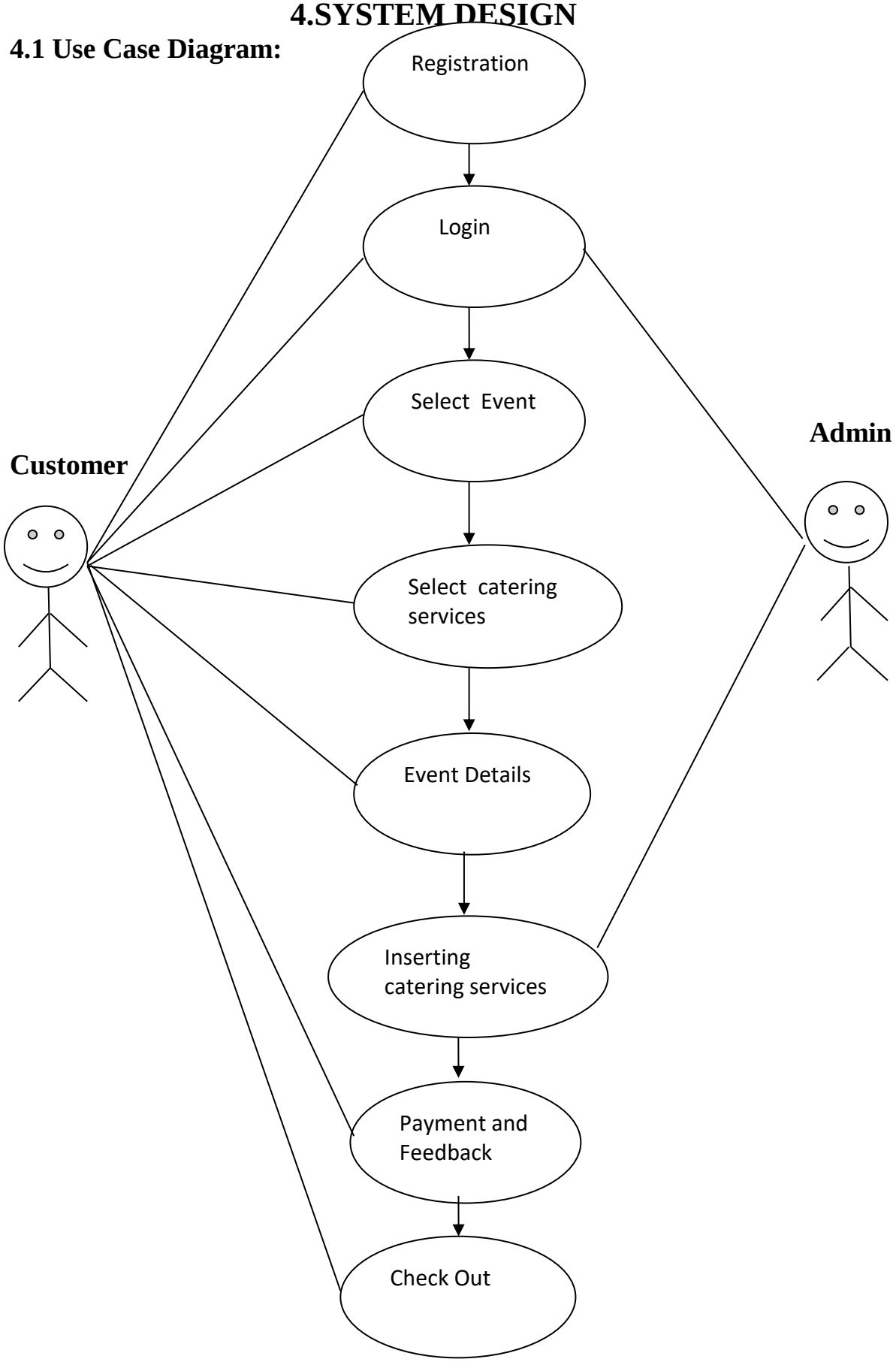
3.2Non-Functional Requirements

- **Performance**: This is essentially how fast your app works. A performance requirement for the way finding app could be that it plots a route in less than 20 seconds.
- **Responsiveness**: This requirement ensures that your app is ready to respond to a user's input or an external event no matter what it's doing currently.
- **Scalability**: Scalability is how well your app deals with increasing use, or size of data.
- **Usability**: This relates to how easily people can use your app. A measure of usability could be the time it takes for end users to become familiar with your app's functions, without training or help.
- **Reliability**: This is the percentage of time that your app works correctly to deliver the desired results, despite potential failures in its environment.
- **Security**: Say that your app saves all the previous routes it calculated and lets you reuse a saved route rather than recalculate it.
- **Modifiability**: This requirement governs how easily your app may be changed. For the way finding app, this requirement might state that the app can use more than one map without needing to be rebuilt.
- **Maintainability**: This relates to the ease at which your app finds bugs and fixes them.

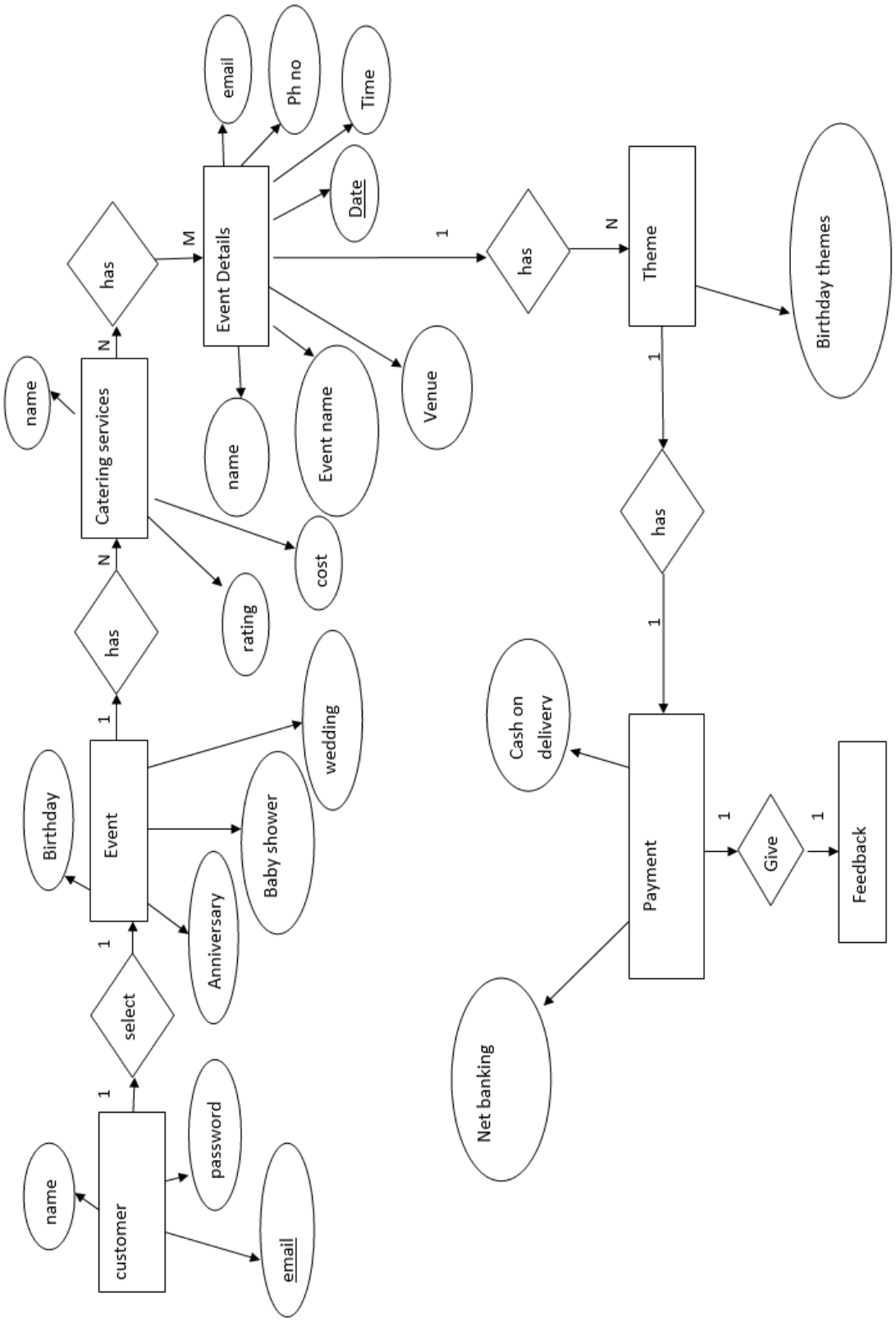
SYSTEM DESIGN

4.SYSTEM DESIGN

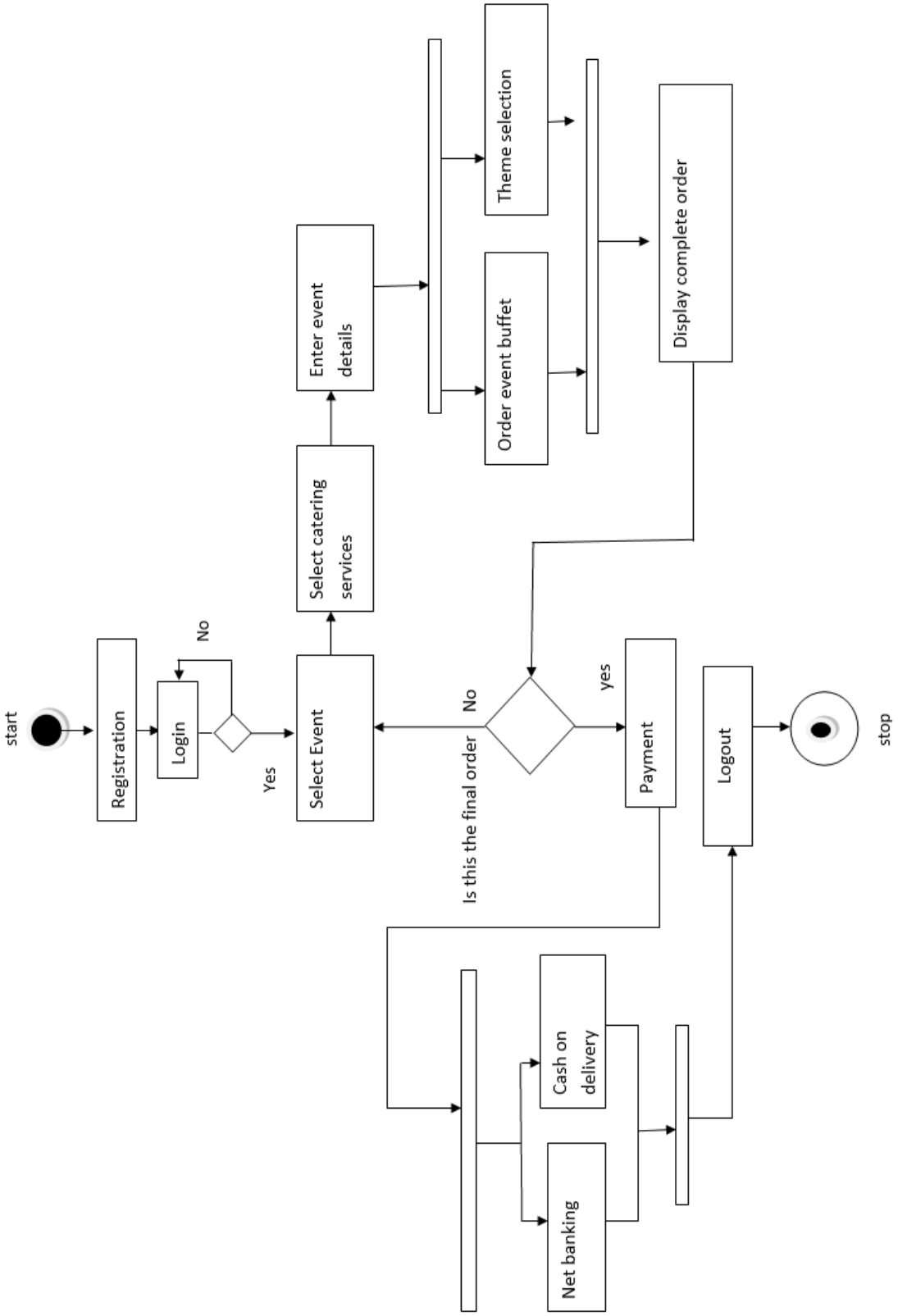
4.1 Use Case Diagram:



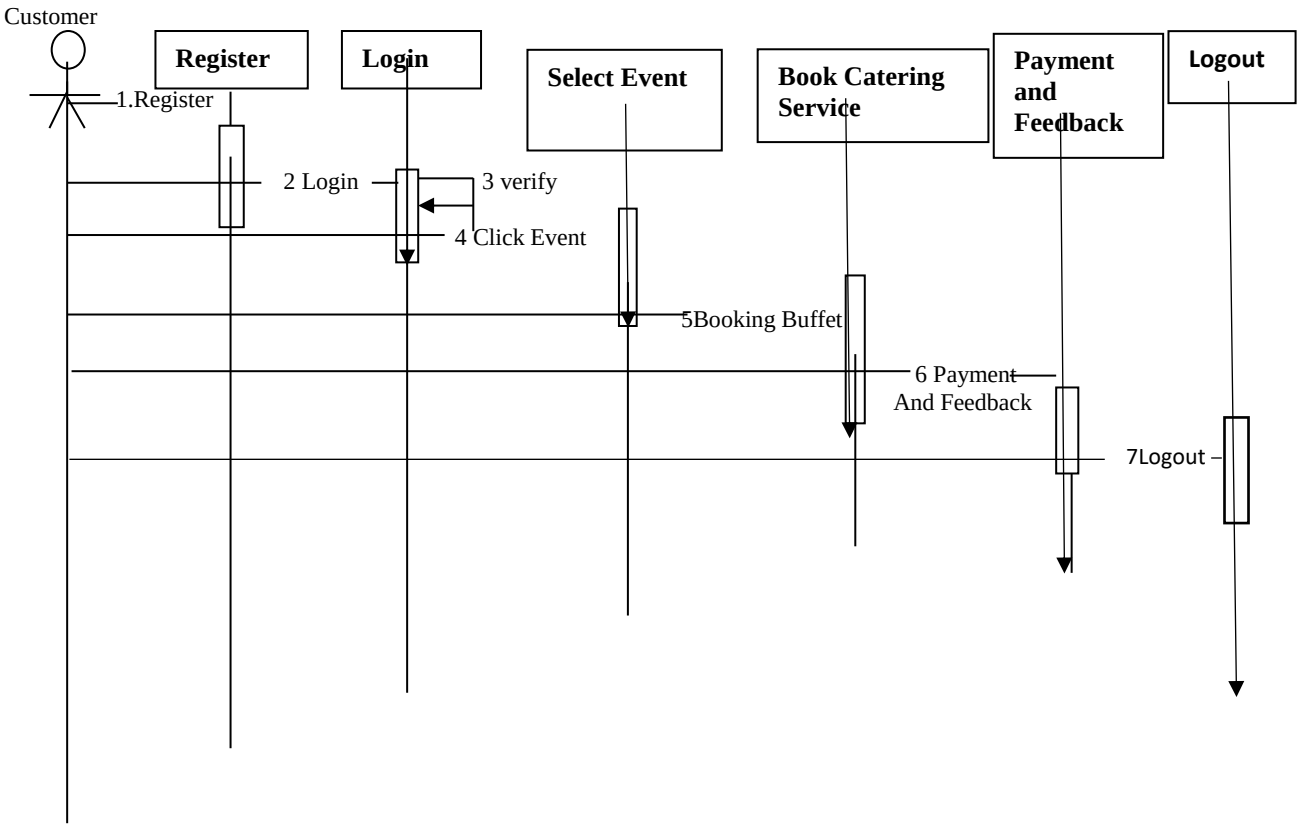
4.2 E-R Diagram

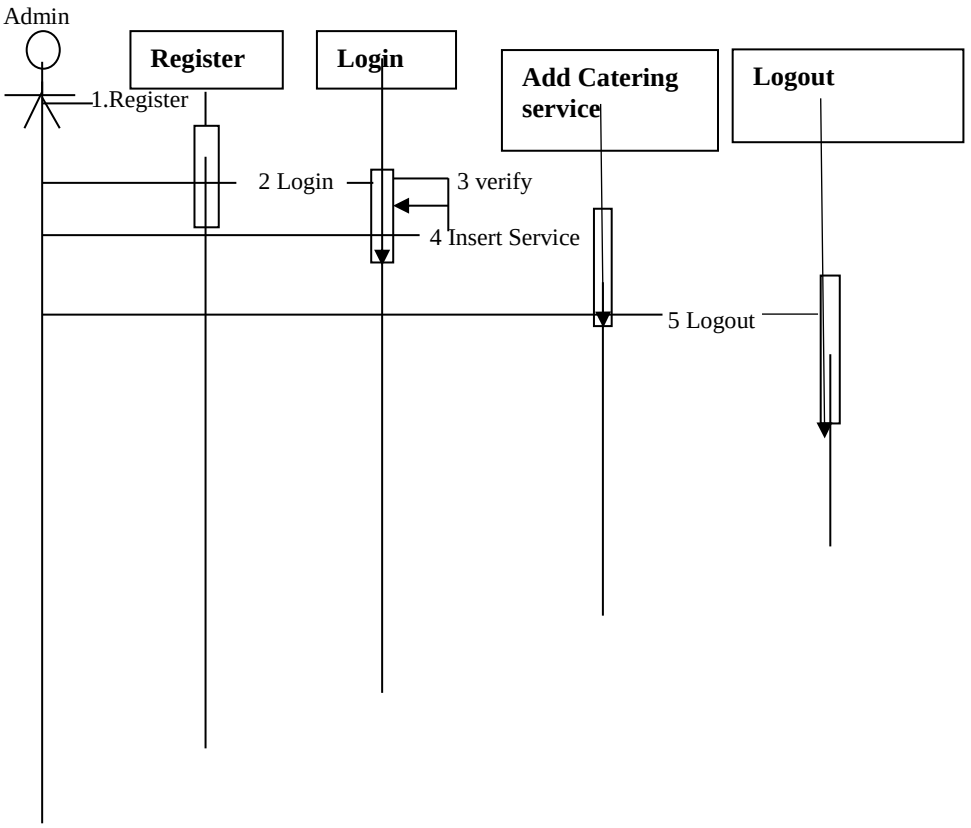


4.3 Activity Diagram



4.4 Sequence Diagram




Admin
Register
Login
Add Catering service
Logout
1.Register
2 Login
3 verify
4 Insert Service
5 Logout

IMPLEMENTATION

5. IMPLEMENTATION

5.1 Source Code

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="@drawable/logo1">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imageView4"
            android:layout_width="wrap_content"
            android:layout_height="150dp"
            android:layout_marginLeft="65dp"
            android:layout_marginTop="300dp"
            app:srcCompat="@drawable/img" />

        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="115dp"
            android:layout_marginTop="10dp"
            android:fontFamily="cursive"
            android:text="FlavorTime Catering"
            android:textColor="#FCFCFC"
            android:textColorHighlight="#00FAF8F8"
            android:textSize="25dp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="100dp"
            android:fontFamily="casual"
            android:text="Our Taste Beyond Your Expectations"
            android:textColor="#FDFCFC"
            android:textColorHighlight="#00FCFAFA"
            android:textSize="13dp"
            android:textStyle="bold" />

    </LinearLayout>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.cardview;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    private static int SPLASH_TIME_OUT = 1000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportActionBar().hide();
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run(){
                Intent homeIntent= new Intent(MainActivity.this, login.class);
                startActivity(homeIntent);
                finish();
            }
        },SPLASH_TIME_OUT);

    }
}
```

activity_card.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.viewpager.widget.ViewPager
        android:foregroundGravity="center"
        android:id="@+id/viewPager"
        android:layout_centerInParent="true"
```

```
        android:overScrollMode="never"
        android:clipToPadding="false"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </androidx.viewpager.widget.ViewPager>

    <Button
        android:id="@+id/btnOrder"
        android:text="Book Now!"
        android:textColor="#fff"
        android:background="@drawable/round"
        android:paddingLeft="30dp"
        android:paddingRight="30dp"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="60dp"
        android:layout_centerHorizontal="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

Item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_vertical">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardCornerRadius="20dp"
        android:layout_margin="8dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="300dp">

            <ImageView
                android:id="@+id/image"
                android:src="@drawable/anniversary"
                android:scaleType="centerCrop"
                android:layout_width="match_parent"
                android:layout_height="200dp" />

            <TextView
                android:id="@+id/title"
                android:textColor="#262626"
```

```
            android:layout_below="@id/image"
            android:layout_marginTop="10dp"
            android:layout_marginLeft="16dp"
            android:text="Birthday"
            android:textSize="16sp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <TextView
            android:id="@+id/desc"
            android:layout_below="@id/title"
            android:layout_marginTop="3dp"
            android:layout_marginLeft="16dp"
            android:layout_marginRight="16dp"
            android:maxLines="3"
            android:drawablePadding="10dp"
            android:ellipsize="end"
            android:text="Description"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

    </RelativeLayout>


</androidx.cardview.widget.CardView>

</LinearLayout>
```

Cardview.java

```
package com.example.cardview;

import android.animation.ArgbEvaluator;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.viewpager.widget.ViewPager;

import java.util.ArrayList;
import java.util.List;

public class cardview extends AppCompatActivity {

    ViewPager viewPager;
    Adapter adapter;
    List<Model> models;
    Integer[] colors = null;
    ArgbEvaluator argbEvaluator = new ArgbEvaluator();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_card);
        getSupportActionBar().hide();

        models = new ArrayList<>();
        models.add(new Model(R.drawable.anniversary, "Anniversary", "Anniversary is a
ceremony to appreciate the ability to still be together after so many years has passed."));
        models.add(new Model(R.drawable.wedding, "Wedding", "Wedding is a ceremony and
its associated rituals by which two people vow to spend their lives together in marriage."));
        models.add(new Model(R.drawable.baby, "Baby Shower", "Baby shower is a gift-giving
party, where an expectant or new mother is showered or given gifts for the baby."));
        models.add(new Model(R.drawable.birthday, "Birthday", "Birthday is a time when a
person acknowledges the anniversary of his or her birth."));

        adapter = new Adapter(models, this);

        viewPager = findViewById(R.id.viewPager);
        viewPager.setAdapter(adapter);
        viewPager.setPadding(130, 0, 130, 0);

        Integer[] colors_temp = {
                getResources().getColor(R.color.color1),
                getResources().getColor(R.color.color2),
                getResources().getColor(R.color.color3),
                getResources().getColor(R.color.color4)
        };

        colors = colors_temp;

        viewPager.setOnPageChangeListener(new ViewPager.OnPageChangeListener() {
            @Override
            public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels)
{

                if (position < (adapter.getCount() -1) && position < (colors.length - 1)) {
                    viewPager.setBackgroundColor(

                            (Integer) argbEvaluator.evaluate(
                                    positionOffset,
                                    colors[position],
                                    colors[position + 1]
                            )
                    );
                }

                else {
                    viewPager.setBackgroundColor(colors[colors.length - 1]);
                }
            }

            @Override
            public void onPageSelected(int position) {
```

```
        }

        @Override
        public void onPageScrollStateChanged(int state) {

        }
    });

  }
}
```

Adapter.java

```
package com.example.cardview;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.viewpager.widget.PagerAdapter;

import java.util.List;

public class Adapter extends PagerAdapter {

    private List<Model> models;
    private LayoutInflater layoutInflater;
    private Context context;

    public Adapter(List<Model> models, Context context) {
        this.models = models;
        this.context = context;
    }

    @Override
    public int getCount() {
        return models.size();
    }

    @Override
    public boolean isViewFromObject(@NonNull View view, @NonNull Object object) {
        return view.equals(object);
    }

    @NonNull
```

```
    @Override
    public Object instantiateItem(@NonNull ViewGroup container, final int position) {
        layoutInflater = LayoutInflater.from(context);
        View view = layoutInflater.inflate(R.layout.item, container, false);

        ImageView imageView;
        TextView title, desc;

        imageView = view.findViewById(R.id.image);
        title = view.findViewById(R.id.title);
        desc = view.findViewById(R.id.desc);

        imageView.setImageResource(models.get(position).getImage());
        title.setText(models.get(position).getTitle());
        desc.setText(models.get(position).getDesc());

        view.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(context, top.class);
                intent.putExtra("param", models.get(position).getTitle());
                context.startActivity(intent);
                // finish();
            }
        });

        container.addView(view, 0);
        return view;
    }

    @Override
    public void destroyItem(@NonNull ViewGroup container, int position, @NonNull Object
object) {
        container.removeView((View)object);
    }
}
```

Model.java

```
package com.example.cardview;

public class Model {

    private int image;
    private String title;
    private String desc;

    public Model(int image, String title, String desc) {
        this.image = image;
        this.title = title;
        this.desc = desc;
```

```
    }

    public int getImage() {
        return image;
    }

    public void setImage(int image) {
        this.image = image;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getDesc() {
        return desc;
    }

    public void setDesc(String desc) {
        this.desc = desc;
    }
}
```

login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#E53935"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="65dp"
        android:layout_marginTop="150dp"
        android:fontFamily="cursive"
        android:text="FlavorTime Catering"
        android:textColor="#FCFCFC"
```

```
    android:textColorHighlight="#00FAF8F8"
    android:textSize="40dp"
    android:textStyle="bold" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="50dp"
    android:text="Username"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
    android:id="@+id/etUserName"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp">

    <requestFocus />
</EditText>

<TextView
    android:id="@+id/textView5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="30dp"
    android:text="Password"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
    android:id="@+id/etPassword"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:inputType="textPassword"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<Button
```

```
        android:id="@+id/button1"
        android:layout_width="390dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="60dp"
        android:background="#CC100C"
        android:onClick="OnLogin"
        android:text="LOGIN"
        android:textColor="#FCFCFC"
        android:textColorHighlight="#00FAF8F8"
        android:textSize="20dp" />

    <Button
        android:id="@+id/button"
        style="?android:attr/borderlessButtonStyle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:background="#E53935"
        android:text="No account yet? Create one"
        android:textAllCaps="false"
        android:textColor="#FCFCFC"
        android:textColorHighlight="#00FAF8F8"
        android:textSize="20dp" />

</LinearLayout>
```

Login.java

```
package com.example.cardview;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import androidx.appcompat.app.AppCompatActivity;

public class login extends AppCompatActivity {

    EditText UsernameEt, PasswordEt;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.login);
        getSupportActionBar().hide();
        UsernameEt = (EditText)findViewById(R.id.etUserName);
        PasswordEt = (EditText)findViewById(R.id.etPassword);
```

```
        Button button=(Button)findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v){
            Intent i = new Intent(login.this, register.class);
            startActivity(i);
        }});}

    public void OnLogin(View view) {
        String username = UsernameEt.getText().toString();
        String password = PasswordEt.getText().toString();
        String type = "login";
        BackgroundWorker backgroundWorker = new BackgroundWorker(this);
        backgroundWorker.execute(type, username, password);
    }

}
```

BackgroundWorker.java

```
package com.example.cardview;

import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;


public class BackgroundWorker extends AsyncTask<String,Void,String> {
    Context context;
    AlertDialog alertDialog;
    BackgroundWorker (Context ctx) {
        context = ctx;
    }
    @Override
    protected String doInBackground(String... params) {
        String type = params[0];
        String login_url = "http://192.168.43.59/login.php";
        if(type.equals("login")) {
            try {
```

```
            String user_name = params[1];
            String password = params[2];
            URL url = new URL(login_url);
            HttpURLConnection httpURLConnection =
(HttpURLConnection)url.openConnection();
            httpURLConnection.setRequestMethod("POST");
            httpURLConnection.setDoOutput(true);
            httpURLConnection.setDoInput(true);
            OutputStream outputStream = httpURLConnection.getOutputStream();
            BufferedWriter bufferedWriter = new BufferedWriter(new
OutputStreamWriter(outputStream, "UTF-8"));
            String post_data = URLEncoder.encode("user_name","UTF-
8")+"="+URLEncoder.encode(user_name,"UTF-8")+"&"
                    +URLEncoder.encode("password","UTF-
8")+"="+URLEncoder.encode(password,"UTF-8");
            bufferedWriter.write(post_data);
            bufferedWriter.flush();
            bufferedWriter.close();
            outputStream.close();
            InputStream inputStream = httpURLConnection.getInputStream();
            BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(inputStream,"iso-8859-1"));
            String result="";
            String line="";
            while((line = bufferedReader.readLine())!= null) {
                result += line;
            }
            bufferedReader.close();
            inputStream.close();
            httpURLConnection.disconnect();
            return result;
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    return null;
}


@Override
protected void onPreExecute() {
    alertDialog = new AlertDialog.Builder(context).create();
    alertDialog.setTitle("Login Status");
}

@Override
protected void onPostExecute(String result) {
    if(result.equalsIgnoreCase("login success !!!!! Welcome user"))
    {
```

```
            alertDialog.setMessage(result);
            alertDialog.show();
            context.startActivity(new Intent(context,cardview.class));

        }
        else
        {
            alertDialog.setMessage("Sorry Login Failed!");
            alertDialog.show();
            context.startActivity(new Intent(context,login.class));


        }
    }

    @Override
    protected void onProgressUpdate(Void... values) {
        super.onProgressUpdate(values);
    }
}
```

activity_register.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#E53935"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="65dp"
        android:layout_marginTop="95dp"
        android:fontFamily="cursive"
        android:text="FlavorTime Catering"
        android:textColor="#FCFCFC"
        android:textColorHighlight="#00FAF8F8"
        android:textSize="40dp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/textView6"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="30dp"
```

```
    android:text="Email"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
    android:id="@+id/email"
    android:inputType="textEmailAddress"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp">
    <requestFocus/>
</EditText>

<TextView
    android:id="@+id/textView7"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="30dp"
    android:text="Username"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
    android:id="@+id/UserName"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp"
    android:inputType="textPersonName"/>

<TextView
    android:id="@+id/textView5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="30dp"
    android:text="Password"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
```

```
        android:id="@+id/Password"
        android:layout_width="390dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:ems="10"
        android:inputType="textPassword"
        android:textColor="#FCFCFC"
        android:textColorHighlight="#00FAF8F8"
        android:textSize="20dp" />

    <Button
        android:id="@+id/reg_button"
        android:layout_width="390dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="60dp"
        android:background="#CC100C"
        android:text="CREATE ACCOUNT"
        android:textColor="#FCFCFC"
        android:textColorHighlight="#00FAF8F8"
        android:textSize="20dp"
        android:onClick="Register1"/>

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:background="#E53935"
        android:text="Already a member? Login"
        android:textColor="#FCFCFC"
        android:textColorHighlight="#00FAF8F8"
        android:textSize="20dp"
        android:textAllCaps="false"
        android:onClick="OnLogin"
        style="?android:attr/borderlessButtonStyle"/>

</LinearLayout>
```

Register.java

```
package com.example.cardview;

import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
```

```
import java.util.concurrent.ExecutionException;

public class register extends AppCompatActivity {
    int flag=0;
    EditText username, password,email;

    String str_register_number, str_username, str_password, str_email;
    public static String id, register_id, login_username, department;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        getSupportActionBar().hide();
        username=(EditText)findViewById(R.id.UserName);
        password=(EditText)findViewById(R.id.Password);
        email=(EditText)findViewById(R.id.email);
        Button reg_button=(Button)findViewById(R.id.reg_button);
    }

    public void Register1(View view)
    {
        str_username = username.getText().toString();
        str_password = password.getText().toString();
        str_email = email.getText().toString();


        if(TextUtils.isEmpty(str_username)){
            flag=1;
            Toast.makeText(register.this, "Enter Username", Toast.LENGTH_LONG).show();
            //     Snackbar snackbar1 = (Snackbar) Snackbar.make(register.this,"Enter
Username",Snackbar.LENGTH_SHORT).show();


        }
        if(TextUtils.isEmpty(str_password)){
            flag=1;
            Toast.makeText(register.this, "Enter Password", Toast.LENGTH_LONG).show();

        }


        if(TextUtils.isEmpty(str_email)){
            flag=1;
            Toast.makeText(register.this, "Enter Email", Toast.LENGTH_LONG).show();

        }
        if(flag==0) {
            String type = "login";

            register_student backgroungWorker = new register_student(this);
```

```
            try {
                String r = backgroungWorker.execute(type, str_username, str_password,
str_email).get();
            } catch (InterruptedException e) {
                e.printStackTrace();
            } catch (ExecutionException e) {
                e.printStackTrace();
            }
        }
        else
        {

        }

    }


}
```

register_student.java

```
package com.example.cardview;

import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;


public class register_student extends AsyncTask<String, Void, String>
{
    Context context;
    AlertDialog alertDialog;
    register_student(Context ctx) {
        context = ctx;
    }
    @Override
    protected String doInBackground(String... params) {
        String type = params[0];
```

```
        String login_url = "http://192.168.43.59/register.php";
        if(type.equals("login")) {
            try {
                String username = params[1];
                String password = params[2];
                String email = params[3];
                URL url = new URL(login_url);
                HttpURLConnection httpURLConnection =
(HttpURLConnection)url.openConnection();
                httpURLConnection.setRequestMethod("POST");
                httpURLConnection.setDoOutput(true);
                httpURLConnection.setDoInput(true);
                OutputStream outputStream = httpURLConnection.getOutputStream();
                BufferedWriter bufferedWriter = new BufferedWriter(new
OutputStreamWriter(outputStream, "UTF-8"));
                String post_data = URLEncoder.encode("username","UTF-
8")+"="+URLEncoder.encode(username,"UTF-8")+ "&"
                        + URLEncoder.encode("password","UTF-
8")+"="+URLEncoder.encode(password,"UTF-8")+"&"
                        + URLEncoder.encode("email","UTF-
8")+"="+URLEncoder.encode(email,"UTF-8");
                bufferedWriter.write(post_data);
                bufferedWriter.flush();
                bufferedWriter.close();
                outputStream.close();
                InputStream inputStream = httpURLConnection.getInputStream();
                BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(inputStream,"iso-8859-1"));
                String result="";
                String line="";
                while((line = bufferedReader.readLine())!= null) {
                    result += line;
                }
                bufferedReader.close();
                inputStream.close();
                httpURLConnection.disconnect();
                return result;
            } catch (MalformedURLException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        return null;
    }

    @Override
    protected void onPreExecute() {
        alertDialog = new AlertDialog.Builder(context).create();
        alertDialog.setTitle("Register Status");
    }
```

```
    @Override
    protected void onPostExecute(String result) {
        if(result.equalsIgnoreCase("Registration Successful"))
        {

            alertDialog.setMessage(result);
            alertDialog.show();
            context.startActivity(new Intent(context,login.class));

        }
        else
        {
            alertDialog.setMessage("Email Already Exist!");
            alertDialog.show();
            context.startActivity(new Intent(context,register.class));


        }
    }

    @Override
    protected void onProgressUpdate(Void... values) {
        super.onProgressUpdate(values);
    }
}
```

top.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".top"
    android:orientation="vertical">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?attr/colorPrimary"
        android:minHeight="?attr/actionBarSize"
        android:theme="?attr/actionBarTheme"
        app:title="FlavorTime Catering" />

    <TextView
        android:id="@+id/textview"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
```

```
        android:layout_marginLeft="15dp"
        android:layout_marginTop="15dp"
        android:fontFamily="@font/aclonica"
        android:text="Top Catering Services"
        android:textColor="#E53935"
        android:textColorHighlight="#E53935"
        android:textSize="25dp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/textView9"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="15dp"
        android:fontFamily="serif-monospace"
        android:text="Search like never before"
        android:textColor="#E53935"
        android:textColorHighlight="#E53935"
        android:textSize="20dp"
        android:textStyle="bold"/>


    <androidx.appcompat.widget.SearchView
        android_id="@+id/sv"
        android_layout_height="50dp"
        android_layout_width="match_parent"
        app_defaultQueryHint="Search.."
        android:layout_width="400dp"
        android:layout_height="wrap_content">

    </androidx.appcompat.widget.SearchView>


    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recylcerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="10dp"
        tools:layout_editor_absoluteX="745dp"
        tools:layout_editor_absoluteY="-51dp" />


</LinearLayout>
```

Listview_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
```

```
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="8dp">

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="120dp"
            android:layout_height="90dp"
            android:padding="4dp" />

        <TextView
            android:id="@+id/textViewTitle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="5dp"
            android:layout_toRightOf="@id/imageView"
            android:text="Apple MacBook Air Core i5 5th Gen - (8 GB/128 GB SSD/Mac OS
Sierra)"
            android:textAppearance="@style/Base.TextAppearance.AppCompat.Small"
            android:textColor="#000000"
            android:textStyle="bold"/>

        <TextView
            android:id="@+id/textViewShortDesc"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/textViewTitle"
            android:layout_marginLeft="5dp"
            android:layout_marginTop="5dp"
            android:layout_toRightOf="@id/imageView"
            android:text="13.3 Inch, 256 GB"
            android:textAppearance="@style/Base.TextAppearance.AppCompat.Small" />

        <TextView
            android:id="@+id/textViewRating"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/textViewShortDesc"
            android:layout_marginLeft="5dp"
            android:layout_marginTop="5dp"
            android:layout_toRightOf="@id/imageView"
            android:background="@color/colorPrimary"
            android:paddingLeft="15dp"
            android:paddingRight="15dp"
            android:text="4.7"
            android:textAppearance="@style/Base.TextAppearance.AppCompat.Small.Inverse"
            android:textStyle="bold" />

    </RelativeLayout>
```

```
</LinearLayout>
```

Top.java

```
package com.example.cardview;

import android.os.Bundle;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

//import android.support.v7.widget.LinearLayoutManager;
//import android.support.v7.widget.RecyclerView;

public class top extends AppCompatActivity {

    //this is the JSON Data URL
    //make sure you are using the correct ip else it will not work
    private static final String URL_PRODUCTS = "http://192.168.43.59/MyApi/Api.php";

    //a list to store all the products
    List<Product> productList;

    //the recyclerview
    RecyclerView recyclerView;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.top);
        getSupportActionBar().hide();

        //getting the recyclerview from xml
```

```
        recyclerView = findViewById(R.id.recylcerView);
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        //initializing the productlist
        productList = new ArrayList<>();

        //this method will fetch and parse json
        //to display it in recyclerview
        loadProducts();
        Button b = (Button) findViewById(R.id.search);
    }

    private void loadProducts() {

        /*
         * Creating a String Request
         * The request type is GET defined by first parameter
         * The URL is defined in the second parameter
         * Then we have a Response Listener and a Error Listener
         * In response listener we will get the JSON response as a String
         * */
        StringRequest stringRequest = new StringRequest(Request.Method.GET,
URL_PRODUCTS,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        try {
                            //converting the string to json array object
                            JSONArray array = new JSONArray(response);

                            //traversing through all the object
                            for (int i = 0; i < array.length(); i++) {

                                //getting product object from json array
                                JSONObject product = array.getJSONObject(i);

                                //adding the product to product list
                                productList.add(new Product(
                                        product.getInt("id"),
                                        product.getString("title"),
                                        product.getString("shortdesc"),
                                        product.getDouble("rating"),
                                        product.getString("image")
                                ));
                            }

                            //creating adapter object and setting it to recyclerview
                            ProductAdapter adapter = new ProductAdapter(top.this, productList);
                            recyclerView.setAdapter(adapter);
                        } catch (JSONException e) {
                            e.printStackTrace();
```

```
                }
            }
        },
        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {

            }
        });

    //adding our stringrequest to queue
    Volley.newRequestQueue(this).add(stringRequest);
  }
}
```

ProductAdapter.java

```
package com.example.cardview;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;

import java.util.List;

//import android.support.v7.widget.RecyclerView;

/**
 * Created by Belal on 10/18/2017.
 */

public class ProductAdapter extends
RecyclerView.Adapter<ProductAdapter.ProductViewHolder> {


  private Context mCtx;
  private List<Product> productList;

  public ProductAdapter(Context mCtx, List<Product> productList) {
    this.mCtx = mCtx;
    this.productList = productList;
  }
```

```
    @Override
    public ProductViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(mCtx);
        View view = inflater.inflate(R.layout.listview_layout, null);
        return new ProductViewHolder(view);
    }

    @Override
    public void onBindViewHolder(ProductViewHolder holder, int position) {
        Product product = productList.get(position);

        //loading the image
        Glide.with(mCtx)
                .load(product.getImage())
                .into(holder.imageView);

        holder.textViewTitle.setText(product.getTitle());
        holder.textViewShortDesc.setText(product.getShortdesc());
        holder.textViewRating.setText(String.valueOf(product.getRating()));
    }

    @Override
    public int getItemCount() {
        return productList.size();
    }

    class ProductViewHolder extends RecyclerView.ViewHolder {

        TextView textViewTitle, textViewShortDesc, textViewRating;
        ImageView imageView;

        public ProductViewHolder(View itemView) {
            super(itemView);

            textViewTitle = itemView.findViewById(R.id.textViewTitle);
            textViewShortDesc = itemView.findViewById(R.id.textViewShortDesc);
            textViewRating = itemView.findViewById(R.id.textViewRating);
            imageView = itemView.findViewById(R.id.imageView);
            itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Intent intent = new Intent(mCtx, wedding.class);
                    mCtx.startActivity(intent);
                }
            });

        }
    }
}
```

Product.java

```
package com.example.cardview;

public class Product {
    private int id;
    private String title;
    private String shortdesc;
    private double rating;
    private String image;

    public Product (int id, String title, String shortdesc, double rating, String image) {
        this.id = id;
        this.title = title;
        this.shortdesc = shortdesc;
        this.rating = rating;
        this.image = image;
    }

    public int getId() {
        return id;
    }

    public String getTitle() {
        return title;
    }

    public String getShortdesc() {
        return shortdesc;
    }

    public double getRating() {
        return rating;
    }

    public String getImage() {
        return image;
    }
}
```

eventdetails.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="#E53935">


    <ScrollView
        android:layout_width="match_parent"
```

```
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

<androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?attr/colorPrimary"
    android:minHeight="?attr/actionBarSize"
    android:theme="?attr/actionBarTheme"
    app:title="New Event" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="65dp"
    android:layout_marginTop="4dp"
    android:fontFamily="cursive"
    android:text="Enter Event Details"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="40dp"
    android:textStyle="bold" />

<TextView
    android:id="@+id/textView7"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="30dp"
    android:text="Name"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
    android:id="@+id/Name"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp">
    <requestFocus/>
</EditText>
```

```
<TextView
    android:id="@+id/textView6"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="30dp"
    android:text="Email"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
    android:id="@+id/Email"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:inputType="textEmailAddress"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />



<TextView
    android:id="@+id/textView8"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="30dp"
    android:text="Event Name"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
    android:id="@+id/eventname"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp"/>

<TextView
    android:id="@+id/textView9"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
```

```
    android:layout_marginTop="30dp"
    android:text="Date"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />


<EditText
    android:id="@+id/date"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="date"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp"
    android:layout_marginLeft="20dp"/>

<TextView
    android:id="@+id/textView10"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="30dp"
    android:text="Time"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
    android:id="@+id/time"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="time"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp"
    android:layout_marginLeft="20dp"/>

<TextView
    android:id="@+id/textView11"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="30dp"
    android:text="Location"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
```

```
        android:id="@+id/location"
        android:layout_width="390dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPostalAddress"
        android:textColor="#FCFCFC"
        android:textColorHighlight="#00FAF8F8"
        android:textSize="20dp"
        android:layout_marginLeft="20dp"/>

    <TextView
        android:id="@+id/textView12"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="30dp"
        android:text="Phone"
        android:textColor="#FCFCFC"
        android:textColorHighlight="#00FAF8F8"
        android:textSize="20dp" />

    <EditText
        android:id="@+id/phone"
        android:layout_width="390dp"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="phone"
        android:textColor="#FCFCFC"
        android:textColorHighlight="#00FAF8F8"
        android:textSize="20dp"
        android:layout_marginLeft="20dp"/>

    <Button
        android:id="@+id/submit"
        android:layout_width="390dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="60dp"
        android:background="#CC100C"
        android:onClick="Register"
        android:text="SUBMIT"
        android:textColor="#FCFCFC"
        android:textColorHighlight="#00FAF8F8"
        android:textSize="20dp" />
        </LinearLayout>
    </ScrollView>


</LinearLayout>
```

Event.java

```
package com.example.cardview;

import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.util.concurrent.ExecutionException;

public class wedding extends AppCompatActivity {
    int flag=0;
    EditText Name, Email, eventname, date,time, location, phone;

    String str_Name, str_Email, str_eventname, str_date, str_time, str_location, str_phone;
    public static String iid, register_id, login_username, department;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.wedding);
        getSupportActionBar().hide();
        Name =(EditText)findViewById(R.id.Name);
        Email=(EditText)findViewById(R.id.Email);
        eventname=(EditText)findViewById(R.id.eventname);
        date=(EditText)findViewById(R.id.date);
        time=(EditText)findViewById(R.id.time);
        location=(EditText)findViewById(R.id.location);
        phone=(EditText)findViewById(R.id.phone);
        Button submit=(Button)findViewById(R.id.submit);
    }

    public void Register(View view)
    {
        str_Name = Name.getText().toString();
        str_Email = Email.getText().toString();
        str_eventname = eventname.getText().toString();
        str_date = date.getText().toString();
        str_time = time.getText().toString();
        str_location = location.getText().toString();
        str_phone = phone.getText().toString();


        if(TextUtils.isEmpty(str_Name)){

            flag=1;
            Toast.makeText(wedding.this, "Enter Your Name", Toast.LENGTH_LONG).show();
```

```
        }


        if(TextUtils.isEmpty(str_Email)){
            flag=1;
            Toast.makeText(wedding.this, "Enter Email", Toast.LENGTH_LONG).show();
            //     Snackbar snackbar1 = (Snackbar) Snackbar.make(register.this,"Enter
Username",Snackbar.LENGTH_SHORT).show();


        }
        if(TextUtils.isEmpty(str_eventname)){
            flag=1;
            Toast.makeText(wedding.this, "Enter Event Name", Toast.LENGTH_LONG).show();

        }


        if(TextUtils.isEmpty(str_date)){
            flag=1;
            Toast.makeText(wedding.this, "Enter Date", Toast.LENGTH_LONG).show();

        }

        if(TextUtils.isEmpty(str_time)){
            flag=1;
            Toast.makeText(wedding.this, "Enter Time", Toast.LENGTH_LONG).show();

        }

        if(TextUtils.isEmpty(str_location)){
            flag=1;
            Toast.makeText(wedding.this, "Enter Location", Toast.LENGTH_LONG).show();

        }

        if(TextUtils.isEmpty(str_phone)){
            flag=1;
            Toast.makeText(wedding.this, "Enter Phone Number",
Toast.LENGTH_LONG).show();

        }

        if(flag==0) {
            String type = "login";

            register_wedding backgroungWorker = new register_wedding(this);
            try {
                String r = backgroungWorker.execute(type, str_Name, str_Email, str_eventname,
str_date, str_time, str_location, str_phone).get();
            } catch (InterruptedException e) {
                e.printStackTrace();
```

```
            } catch (ExecutionException e) {
                e.printStackTrace();
            }
        }
        else
        {

        }

    }


}
```

event_register.java

```
package com.example.cardview;

import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;


public class register_wedding extends AsyncTask<String, Void, String>
{
    Context context;
    AlertDialog alertDialog;
    register_wedding(Context ctx) {
        context = ctx;
    }
    @Override
    protected String doInBackground(String... params) {
        String type = params[0];
        String login_url = "http://192.168.43.59/wedding.php";
        if(type.equals("login")) {
            try {
                String Name = params[1];
                String Email = params[2];
```

```
            String eventname = params[3];
            String date = params[4];
            String time = params[5];
            String location = params[6];
            String phone = params[7];

            URL url = new URL(login_url);
            HttpURLConnection httpURLConnection =
(HttpURLConnection)url.openConnection();
            httpURLConnection.setRequestMethod("POST");
            httpURLConnection.setDoOutput(true);
            httpURLConnection.setDoInput(true);
            OutputStream outputStream = httpURLConnection.getOutputStream();
            BufferedWriter bufferedWriter = new BufferedWriter(new
OutputStreamWriter(outputStream, "UTF-8"));
            String post_data = URLEncoder.encode("Name","UTF-
8")+"="+URLEncoder.encode(Name,"UTF-8")+ "&"
                    + URLEncoder.encode("Email","UTF-
8")+"="+URLEncoder.encode(Email,"UTF-8")+ "&"
                    + URLEncoder.encode("eventname","UTF-
8")+"="+URLEncoder.encode(eventname,"UTF-8")+"&"
                    + URLEncoder.encode("date","UTF-
8")+"="+URLEncoder.encode(date,"UTF-8") +"&"
                    + URLEncoder.encode("time","UTF-
8")+"="+URLEncoder.encode(time,"UTF-8") +"&"
                    + URLEncoder.encode("location","UTF-
8")+"="+URLEncoder.encode(location,"UTF-8") +"&"
                    + URLEncoder.encode("phone","UTF-
8")+"="+URLEncoder.encode(phone,"UTF-8");
            bufferedWriter.write(post_data);
            bufferedWriter.flush();
            bufferedWriter.close();
            outputStream.close();
            InputStream inputStream = httpURLConnection.getInputStream();
            BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(inputStream,"iso-8859-1"));
            String result="";
            String line="";
            while((line = bufferedReader.readLine())!= null) {
                result += line;
            }
            bufferedReader.close();
            inputStream.close();
            httpURLConnection.disconnect();
            return result;
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    return null;
```

```
    }

    @Override
    protected void onPreExecute() {
        alertDialog = new AlertDialog.Builder(context).create();
        alertDialog.setTitle("Event Status");
    }

    @Override
    protected void onPostExecute(String result) {
        if(result.equalsIgnoreCase("Event Successfully Submitted"))
        {

            alertDialog.setMessage(result);
            alertDialog.show();
            context.startActivity(new Intent(context,birthday.class));

        }
        else
        {
            alertDialog.setMessage("Enter Valid Details!");
            alertDialog.show();
            context.startActivity(new Intent(context,wedding.class));

        }
    }

    @Override
    protected void onProgressUpdate(Void... values) {
        super.onProgressUpdate(values);
    }
}
```

birthday.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".top"
    android:orientation="vertical">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?attr/colorPrimary"
        android:minHeight="?attr/actionBarSize"
        android:theme="?attr/actionBarTheme"
```

```
    app:title="FlavorTime Catering" />

<TextView
    android:id="@+id/textview"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="15dp"
    android:fontFamily="@font/aclonica"
    android:text="Birthday Decoration Themes"
    android:textColor="#E53935"
    android:textColorHighlight="#E53935"
    android:textSize="23dp"
    android:textStyle="bold" />

<TextView
    android:id="@+id/textView9"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="15dp"
    android:fontFamily="serif-monospace"
    android:text="Search like never before"
    android:textColor="#E53935"
    android:textColorHighlight="#E53935"
    android:textSize="20dp"
    android:textStyle="bold"/>


<EditText
    android:id="@+id/shortdesc"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:hint="Search Your Favourite Theme"
    android:inputType="textPersonName" />

<Button
    android:id="@+id/search"
    android:layout_width="wrap_content"
    android:layout_height="30dp"
    android:layout_gravity="center"
    android:layout_marginLeft="150dp"
    android:layout_marginTop="-40dp"
    android:background="#CC100C"
    android:gravity="center"
    android:text="Search "
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8" />

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/recylcerView"
```

```
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="20dp"
        tools:layout_editor_absoluteX="745dp"
        tools:layout_editor_absoluteY="-51dp" />


</LinearLayout>
```

Listviewbirthday.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="8dp">

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="350dp"
            android:layout_height="250dp"
            android:layout_marginLeft="20dp"
            android:padding="4dp" />

        <TextView
            android:id="@+id/textViewTitle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="false"
            android:layout_marginLeft="-325dp"
            android:layout_marginTop="260dp"
            android:layout_toRightOf="@id/imageView"
            android:text="Apple MacBook Air Core i5 5th Gen - (8 GB/128 GB SSD/Mac OS
Sierra)"
            android:textAppearance="@style/Base.TextAppearance.AppCompat.Small"
            android:textColor="#000000"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/textViewShortDesc"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/textViewTitle"
            android:layout_marginLeft="-325dp"
            android:layout_marginTop="5dp"
            android:layout_toRightOf="@id/imageView"
```

```
            android:text="13.3 Inch, 256 GB"
            android:textAppearance="@style/Base.TextAppearance.AppCompat.Small" />

        <TextView
            android:id="@+id/textViewRating"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/textViewShortDesc"
            android:layout_marginLeft="-325dp"
            android:layout_marginTop="5dp"
            android:layout_toRightOf="@id/imageView"
            android:background="@color/colorPrimary"
            android:paddingLeft="15dp"
            android:paddingRight="15dp"
            android:text="$4.7"
            android:textAppearance="@style/Base.TextAppearance.AppCompat.Small.Inverse"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/textViewBuy"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/textViewShortDesc"
            android:layout_marginLeft="-65dp"
            android:layout_marginTop="5dp"
            android:layout_toRightOf="@id/imageView"
            android:background="@color/colorPrimary"
            android:paddingLeft="15dp"
            android:paddingRight="15dp"
            android:text="BUY"
            android:textAppearance="@style/Base.TextAppearance.AppCompat.Small.Inverse"
            android:textStyle="bold" />

    </RelativeLayout>

</LinearLayout>
```

Birthday.java

```
package com.example.cardview;


import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
```

```
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.List;

//import android.support.v7.widget.LinearLayoutManager;
//import android.support.v7.widget.RecyclerView;

public class birthday extends AppCompatActivity {

    //this is the JSON Data URL
    //make sure you are using the correct ip else it will not work
    private static final String URL_PRODUCTS = "http://192.168.43.59/feed.php";

    //a list to store all the products
    List<BProduct> productList;

    //the recyclerview
    RecyclerView recyclerView;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.birthday);
        getSupportActionBar().hide();



        //getting the recyclerview from xml
        recyclerView = findViewById(R.id.recylcerView);
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        //initializing the productlist
        productList = new ArrayList<>();

        //this method will fetch and parse json
        //to display it in recyclerview
        loadProducts();
        Button b = (Button) findViewById(R.id.search);
    }

    private void loadProducts() {
```

```
        /*
         * Creating a String Request
         * The request type is GET defined by first parameter
         * The URL is defined in the second parameter
         * Then we have a Response Listener and a Error Listener
         * In response listener we will get the JSON response as a String
         * */
        StringRequest stringRequest = new StringRequest(Request.Method.GET,
URL_PRODUCTS,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        try {
                            //converting the string to json array object
                            JSONArray array = new JSONArray(response);

                            //traversing through all the object
                            for (int i = 0; i < array.length(); i++) {

                                //getting product object from json array
                                JSONObject product = array.getJSONObject(i);

                                //adding the product to product list
                                productList.add(new BProduct(
                                        product.getInt("id"),
                                        product.getString("theme"),
                                        product.getString("des"),
                                        product.getString("amt"),
                                        product.getString("image")
                                ));
                            }

                            //creating adapter object and setting it to recyclerview
                            BirthdayAdapter adapter = new BirthdayAdapter(birthday.this,
productList);
                            recyclerView.setAdapter(adapter);
                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {

                    }
                });

        //adding our stringrequest to queue
        Volley.newRequestQueue(this).add(stringRequest);
```

```
    }
}
```

birthdayadapter.java

```
package com.example.cardview;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;

import java.util.List;

//import android.support.v7.widget.RecyclerView;

/**
 * Created by Belal on 10/18/2017.
 */

public class BirthdayAdapter extends
RecyclerView.Adapter<BirthdayAdapter.ProductViewHolder> {


    private Context mCtx;
    private List<BProduct> productList;

    public BirthdayAdapter(Context mCtx, List<BProduct> productList) {
        this.mCtx = mCtx;
        this.productList = productList;
    }

    @Override
    public ProductViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(mCtx);
        View view = inflater.inflate(R.layout.listview_birthday, null);
        return new ProductViewHolder(view);
    }

    @Override
    public void onBindViewHolder(ProductViewHolder holder, int position) {
        BProduct product = productList.get(position);

        //loading the image
```

```
        Glide.with(mCtx)
                .load(product.getImage())
                .into(holder.imageView);

        holder.textViewTitle.setText(product.getTitle());
        holder.textViewShortDesc.setText(product.getShortdesc());
        holder.textViewRating.setText(String.valueOf(product.getRating()));
    }

    @Override
    public int getItemCount() {
        return productList.size();
    }

    class ProductViewHolder extends RecyclerView.ViewHolder {

        TextView textViewTitle, textViewShortDesc, textViewRating;
        ImageView imageView;

        public ProductViewHolder(View itemView) {
            super(itemView);

            textViewTitle = itemView.findViewById(R.id.textViewTitle);
            textViewShortDesc = itemView.findViewById(R.id.textViewShortDesc);
            textViewRating = itemView.findViewById(R.id.textViewRating);
            imageView = itemView.findViewById(R.id.imageView);
            itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Intent intent = new Intent(mCtx, payment.class);
                    intent.putExtra("textViewText", textViewRating.getText().toString());
                    mCtx.startActivity(intent);
                }
            });


        }
    }
}
```

Bproduct.java

```
package com.example.cardview;

public class BProduct {
    private int id;
    private String theme;
    private String des;
    private String amt;
    private String image;
```

```
    public BProduct (int id, String theme, String des, String amt, String image) {
        this.id = id;
        this.theme = theme;
        this.des = des;
        this.amt = amt;
        this.image = image;
    }

    public int getId() {
        return id;
    }

    public String getTitle() {
        return theme;
    }

    public String getShortdesc() {
        return des;
    }

    public String getRating() {
        return amt;
    }

    public String getImage() {
        return image;
    }
}
```

payment.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".payment">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?attr/colorPrimary"
        android:minHeight="?attr/actionBarSize"
        android:theme="?attr/actionBarTheme"
        app:title="Payment" />

    <TextView
        android:id="@+id/textViewRating"
        android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="70dp"
    android:layout_marginRight="20dp"
    android:text=""
    android:textColor="#0F0101"
    android:textSize="20dp"
    android:textStyle="bold" />


<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="UPI ID"
    android:visibility="gone"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="20dp"
    android:id="@+id/upi_id"
    android:layout_below="@+id/textViewRating"
    android:text="flavor time catering@upi"/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Name"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="20dp"
    android:id="@+id/name"
    android:layout_below="@+id/upi_id"
    android:text=""/>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Note"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="20dp"
    android:id="@+id/note"
    android:layout_below="@+id/name"
    android:text=""/>

<Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:background="@color/colorPrimary"
    android:textColor="#fff"
    android:id="@+id/send"
    android:layout_below="@+id/note"
    android:layout_marginTop="20dp"
```

```
        android:layout_centerHorizontal="true"
        android:text="send by upi"/>


    <ImageButton
        android:id="@+id/simpleImageButtonYouTube"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/simpleImageButtonHome"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="440dp"
        android:padding="10dp"
        android:src="@drawable/cash" />


</RelativeLayout>
```

Payment.java

```
package com.example.cardview;

import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;

public class payment extends AppCompatActivity {

    EditText noteEt, nameEt, upiIdEt;
    Button send;
    TextView textViewRating;
    public static String amount;

    final int UPI_PAYMENT = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.payment);
        getSupportActionBar().hide();
        ImageButton simpleImageButtonYouTube =
    (ImageButton)findViewById(R.id.simpleImageButtonYouTube);
        Bundle extras = getIntent().getExtras();
        if (extras != null) {
            TextView textView = (TextView) findViewById(R.id.textViewRating);
            textView.setText(extras.getString("textViewText"));
        }

        initializeViews();

        send.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //Getting the values from the EditTexts
                    amount = textViewRating.getText().toString();
                String note = noteEt.getText().toString();
                String name = nameEt.getText().toString();
                String upiId = upiIdEt.getText().toString();
                payUsingUpi(amount, upiId, name, note);

            }
        });
        simpleImageButtonYouTube.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v){
                Intent i = new Intent(payment.this, feedback.class);
                startActivity(i);
            }
        });
    }

    void initializeViews() {
        send = findViewById(R.id.send);
        textViewRating = findViewById(R.id.textViewRating);
        noteEt = findViewById(R.id.note);
        nameEt = findViewById(R.id.name);
        upiIdEt = findViewById(R.id.upi_id);
    }

    void payUsingUpi(String amount, String upiId, String name, String note) {

        Uri uri = Uri.parse("upi://pay").buildUpon()
                .appendQueryParameter("pa", upiId)
                .appendQueryParameter("pn", name)
                .appendQueryParameter("tn", note)
                .appendQueryParameter("am", amount)
                .appendQueryParameter("cu", "INR")
                .build();


        Intent upiPayIntent = new Intent(Intent.ACTION_VIEW);
```

```
        upiPayIntent.setData(uri);


        // will always show a dialog to user to choose an app
        Intent chooser = Intent.createChooser(upiPayIntent, "Pay with");

        // check if intent resolves
        if(null != chooser.resolveActivity(getPackageManager())) {
            startActivityForResult(chooser, UPI_PAYMENT);
        } else {
            Toast.makeText(payment.this,"No UPI app found, please install one to
continue",Toast.LENGTH_SHORT).show();
        }

    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        switch (requestCode) {
            case UPI_PAYMENT:
                if ((RESULT_OK == resultCode) || (resultCode == 11)) {
                    if (data != null) {
                        String trxt = data.getStringExtra("response");
                        Log.d("UPI", "onActivityResult: " + trxt);
                        ArrayList<String> dataList = new ArrayList<>();
                        dataList.add(trxt);
                        upiPaymentDataOperation(dataList);
                    } else {
                        Log.d("UPI", "onActivityResult: " + "Return data is null");
                        ArrayList<String> dataList = new ArrayList<>();
                        dataList.add("nothing");
                        upiPaymentDataOperation(dataList);
                    }
                } else {
                    Log.d("UPI", "onActivityResult: " + "Return data is null"); //when user simply
back without payment
                    ArrayList<String> dataList = new ArrayList<>();
                    dataList.add("nothing");
                    upiPaymentDataOperation(dataList);
                }
                break;
        }
    }

    private void upiPaymentDataOperation(ArrayList<String> data) {
        if (isConnectionAvailable(payment.this)) {
            String str = data.get(0);
            Log.d("UPIPAY", "upiPaymentDataOperation: "+str);
            String paymentCancel = "";
            if(str == null) str = "discard";
```

```
            String status = "";
            String approvalRefNo = "";
            String response[] = str.split("&");
            for (int i = 0; i < response.length; i++) {
                String equalStr[] = response[i].split("=");
                if(equalStr.length >= 2) {
                    if (equalStr[0].toLowerCase().equals("Status".toLowerCase())) {
                        status = equalStr[1].toLowerCase();
                    }
                    else if (equalStr[0].toLowerCase().equals("ApprovalRefNo".toLowerCase()) ||
equalStr[0].toLowerCase().equals("txnRef".toLowerCase())) {
                        approvalRefNo = equalStr[1];
                    }
                }
                else {
                    paymentCancel = "Payment cancelled by user.";
                }
            }

            if (status.equals("success")) {
                //Code to handle successful transaction here.
                Toast.makeText(payment.this, "Transaction successful.",
Toast.LENGTH_SHORT).show();
                Log.d("UPI", "responseStr: "+approvalRefNo);
            }
            else if("Payment cancelled by user.".equals(paymentCancel)) {
                Toast.makeText(payment.this, "Payment cancelled by user.",
Toast.LENGTH_SHORT).show();
            }
            else {
                Toast.makeText(payment.this, "Transaction failed.Please try again",
Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(payment.this, "Internet connection is not available. Please check and
try again", Toast.LENGTH_SHORT).show();
        }
    }

    public static boolean isConnectionAvailable(Context context) {
        ConnectivityManager connectivityManager = (ConnectivityManager)
context.getSystemService(Context.CONNECTIVITY_SERVICE);
        if (connectivityManager != null) {
            NetworkInfo netInfo = connectivityManager.getActiveNetworkInfo();
            if (netInfo != null && netInfo.isConnected()
                    && netInfo.isConnectedOrConnecting()
                    && netInfo.isAvailable()) {
                return true;
            }
        }
        return false;
```

```
    }
}
```

feedback.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".feedback">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?attr/colorPrimary"
        android:minHeight="?attr/actionBarSize"
        android:theme="?attr/actionBarTheme"
        app:title="FlavorTime Catering" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="250dp"
        android:ems="30"
        android:fontFamily="@font/amarante"
        android:inputType="textPersonName"
        android:text="                    Rate Us!"
        android:textColor="#DA0A0A"
        android:textStyle="bold"/>

    <com.hsalf.smilerating.SmileRating
        android:id="@+id/smile_rating"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true" />

    <Button
        android:id="@+id/button"
        android:layout_width="390dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="550dp"
        android:background="#CC100C"
        android:text="LOGOUT"
        android:textColor="#FCFCFC"
        android:textColorHighlight="#00FAF8F8"
        android:textSize="20dp" />
```

```
</RelativeLayout>
```

Feedback.java

```
package com.example.cardview;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.hsalf.smilerating.BaseRating;
import com.hsalf.smilerating.SmileRating;


public class feedback extends AppCompatActivity {
    SmileRating smileRating;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.feedback);
        getSupportActionBar().hide();


        smileRating = (SmileRating) findViewById(R.id.smile_rating);
        smileRating.setOnSmileySelectionListener(new
SmileRating.OnSmileySelectionListener() {
            @Override
            public void onSmileySelected(@BaseRating.Smiley int smiley, boolean reselected) {

                switch (smiley) {
                    case SmileRating.BAD:
                        Toast.makeText(feedback.this, "BAD", Toast.LENGTH_SHORT).show();
                        break;
                    case SmileRating.GOOD:
                        Toast.makeText(feedback.this, "GOOD", Toast.LENGTH_SHORT).show();
                        break;
                    case SmileRating.GREAT:
                        Toast.makeText(feedback.this, "GREAT",
                                Toast.LENGTH_SHORT).show();
                        break;
                    case SmileRating.OKAY:
                        Toast.makeText(feedback.this, "OKAY", Toast.LENGTH_SHORT).show
                                ();
                        break;
                    case SmileRating.TERRIBLE:
```

```
                Toast.makeText(feedback.this, "TERRIBLE",
                        Toast.LENGTH_SHORT).show();
                break;
            }
        }
    });

    Button button=(Button)findViewById(R.id.button);
    button.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v){
            Intent i = new Intent(feedback.this, login.class);
            startActivity(i);
        }});}
}
```

activity_add.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#E53935"
    android:orientation="vertical">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="160dp"
        android:layout_marginTop="95dp"
        android:fontFamily="cursive"
        android:text="ADD"
        android:textColor="#FCFCFC"
        android:textColorHighlight="#00FAF8F8"
        android:textSize="40dp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/textView6"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
    android:layout_marginLeft="20dp"
    android:layout_marginTop="30dp"
    android:text="Title"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
    android:id="@+id/title"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp">
    <requestFocus/>
</EditText>

<TextView
    android:id="@+id/textView7"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="30dp"
    android:text="Location"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
    android:id="@+id/shortdesc"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="30dp"
    android:text="Rating"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
```

```
    android:id="@+id/rating"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<TextView
    android:id="@+id/textView66"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="30dp"
    android:text="Image URL"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp" />

<EditText
    android:id="@+id/image"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:ems="10"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp"
    android:textIsSelectable="true"/>

<Button
    android:id="@+id/reg_button"
    android:layout_width="390dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="60dp"
    android:background="#CC100C"
    android:text="ADD"
    android:textColor="#FCFCFC"
    android:textColorHighlight="#00FAF8F8"
    android:textSize="20dp"
    android:onClick="Register"/>

<Button
    android:id="@+id/button2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:background="#E53935"
    android:text="UPDATE"
    android:textColor="#FCFCFC"
```

```
        android:textColorHighlight="#00FAF8F8"
        android:textSize="20dp"
        android:textAllCaps="false"
        style="?android:attr/borderlessButtonStyle"/>

        </LinearLayout>
    </ScrollView>

</LinearLayout>
```

Add.java

```
package com.example.update;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.util.concurrent.ExecutionException;

public class register extends AppCompatActivity {
    int flag=0;
    EditText title,shortdesc,rating,image;

    String str_title, str_shortdesc, str_rating, str_image;
    public static String id, register_id, login_username, department;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update);
        getSupportActionBar().hide();
        title=(EditText)findViewById(R.id.title);
        shortdesc=(EditText)findViewById(R.id.shortdesc);
        rating=(EditText)findViewById(R.id.rating);
        image=(EditText)findViewById(R.id.image);
        Button reg_button=(Button)findViewById(R.id.reg_button);
        Button button2=(Button)findViewById(R.id.button2);
        button2.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v){
                Intent i = new Intent(register.this, update.class);
                startActivity(i);
            }});
    }
```

```
    public void Register(View view)
    {
        str_title = title.getText().toString();
        str_shortdesc = shortdesc.getText().toString();
        str_rating = rating.getText().toString();
        str_image = image.getText().toString();


        if(TextUtils.isEmpty(str_title)){
            flag=1;
            Toast.makeText(register.this, "Enter Title", Toast.LENGTH_LONG).show();
            //     Snackbar snackbar1 = (Snackbar) Snackbar.make(register.this,"Enter
Username",Snackbar.LENGTH_SHORT).show();



        }
        if(TextUtils.isEmpty(str_shortdesc)){
            flag=1;
            Toast.makeText(register.this, "Enter Location", Toast.LENGTH_LONG).show();

        }


        if(TextUtils.isEmpty(str_rating)){
            flag=1;
            Toast.makeText(register.this, "Enter Rating", Toast.LENGTH_LONG).show();

        }

        if(TextUtils.isEmpty(str_image)){
            flag=1;
            Toast.makeText(register.this, "Enter Image URL", Toast.LENGTH_LONG).show();

        }
        if(flag==0) {
            String type = "login";

            register_student backgroungWorker = new register_student(this);
            try {
                String r = backgroungWorker.execute(type, str_title, str_shortdesc, str_rating,
str_image).get();
            } catch (InterruptedException e) {
                e.printStackTrace();
            } catch (ExecutionException e) {
                e.printStackTrace();
            }
        }
        else
        {

        }
```

```
    }


}
```

registeradd.java

```
package com.example.update;

import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;


public class register_student extends AsyncTask<String, Void, String>
{
    Context context;
    AlertDialog alertDialog;
    register_student(Context ctx) {
        context = ctx;
    }
    @Override
    protected String doInBackground(String... params) {
        String type = params[0];
        String login_url = "http://192.168.43.59/add.php";
        if(type.equals("login")) {
            try {
                String title = params[1];
                String shortdesc = params[2];
                String rating = params[3];
                String image = params[4];
                URL url = new URL(login_url);
                HttpURLConnection httpURLConnection =
(HttpURLConnection)url.openConnection();
                httpURLConnection.setRequestMethod("POST");
                httpURLConnection.setDoOutput(true);
                httpURLConnection.setDoInput(true);
                OutputStream outputStream = httpURLConnection.getOutputStream();
```

```
            BufferedWriter bufferedWriter = new BufferedWriter(new
OutputStreamWriter(outputStream, "UTF-8"));
            String post_data = URLEncoder.encode("title","UTF-
8")+"="+URLEncoder.encode(title,"UTF-8")+ "&"
                    + URLEncoder.encode("shortdesc","UTF-
8")+"="+URLEncoder.encode(shortdesc,"UTF-8")+"&"
                    + URLEncoder.encode("rating","UTF-
8")+"="+URLEncoder.encode(rating,"UTF-8")+"&"
                    + URLEncoder.encode("image","UTF-
8")+"="+URLEncoder.encode(image,"UTF-8");
            bufferedWriter.write(post_data);
            bufferedWriter.flush();
            bufferedWriter.close();
            outputStream.close();
            InputStream inputStream = httpURLConnection.getInputStream();
            BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(inputStream,"iso-8859-1"));
            String result="";
            String line="";
            while((line = bufferedReader.readLine())!= null) {
                result += line;
            }
            bufferedReader.close();
            inputStream.close();
            httpURLConnection.disconnect();
            return result;
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    return null;
}

@Override
protected void onPreExecute() {
    alertDialog = new AlertDialog.Builder(context).create();
    alertDialog.setTitle("Insert Status");
}

@Override
protected void onPostExecute(String result) {
    if(result.equalsIgnoreCase("Added Successfully"))
    {

        alertDialog.setMessage(result);
        alertDialog.show();
        context.startActivity(new Intent(context,register.class));

    }
    else
```

```
        {
            alertDialog.setMessage("Enter Valid Details");
            alertDialog.show();
            context.startActivity(new Intent(context,register.class));


        }
    }

    @Override
    protected void onProgressUpdate(Void... values) {
        super.onProgressUpdate(values);
    }
}
```

login.php

```
<?php

require "conn.php";

$user_name = $_POST["user_name"];

$user_pass = $_POST["password"];

$mysql_qry = "select * from user where username like '$user_name' and password like
'$user_pass';";

$result = mysqli_query($conn ,$mysql_qry);

if(mysqli_num_rows($result) > 0) {

echo "login success !!!!! Welcome user";

}

else {

echo "login not success";

}

?>
```

Register.php

```
<?php

require "conn.php";

$username=$_POST["username"];
```

```
$password=$_POST["password"];

$email=$_POST["email"];

$mysql_qry="insert into
user(username,password,email)values('$username','$password','$email')";

if($conn->query($mysql_qry)==TRUE)

{

echo "Registration successful";

}

else

{

echo "email already exist";

}

$conn->close();

?>
```

Service.php

```
<?php


        //database constants

        define('DB_HOST', 'localhost');

        define('DB_USER', 'root');

        define('DB_PASS', '');

        define('DB_NAME', 'login');


        //connecting to database and getting the connection object

        $conn = new mysqli(DB_HOST, DB_USER, DB_PASS, DB_NAME);


        //Checking if any error occured while connecting
```

```
	if (mysqli_connect_errno()) {
		echo "Failed to connect to MySQL: " . mysqli_connect_error();
		die();
	}

	//creating a query
	$stmt = $conn->prepare("SELECT id, title, shortdesc, rating, image FROM service;");

	//executing the query
	$stmt->execute();

	//binding results to the query
	$stmt->bind_result($id, $title, $shortdesc, $rating, $image);

	$products = array();

	//traversing through all the result
	while($stmt->fetch()){
		$temp = array();
		$temp['id'] = $id;
		$temp['title'] = $title;
		$temp['shortdesc'] = $shortdesc;
		$temp['rating'] = $rating;
		$temp['image'] = $image;
		array_push($products, $temp);
	}

	//displaying the result in json format
```

```
        echo json_encode($products);
```

event.php

```
<?php
require "conn.php";
$Name=$_POST["Name"];
$Email=$_POST["Email"];
$eventname=$_POST["eventname"];
$date=$_POST["date"];
$time=$_POST["time"];
$location=$_POST["location"];
$phone=$_POST["phone"];
$mysql_qry="insert into wedding(Name, Email, eventname, date, time, location,
phone)values('$Name','$Email','$eventname','$date','$time','$location','$phone')";
if($conn->query($mysql_qry)==TRUE)
{
echo "Event Successfully Submitted";
}
else
{
echo "Event Id Already Exist";
}
$conn->close();
?>
```

Birthday.php

```
<?php

        //database constants
```

```
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_PASS', '');
define('DB_NAME', 'login');

//connecting to database and getting the connection object
$conn = new mysqli(DB_HOST, DB_USER, DB_PASS, DB_NAME);

//Checking if any error occured while connecting
if (mysqli_connect_errno()) {
        echo "Failed to connect to MySQL: " . mysqli_connect_error();
        die();
}

//creating a query
$stmt = $conn->prepare("SELECT id, theme, des, amt, image FROM feed;");

//executing the query
$stmt->execute();

//binding results to the query
$stmt->bind_result($id, $theme, $des, $amt, $image);

$products = array();

//traversing through all the result
while($stmt->fetch()){
        $temp = array();
```

```
                $temp['id'] = $id;
                $temp['theme'] = $theme;
                $temp['des'] = $des;
                $temp['amt'] = $amt;
                $temp['image'] = $image;
                array_push($products, $temp);
        }

        //displaying the result in json format
        echo json_encode($products);
```

add.php

```
<?php
require "conn.php";
$title=$_POST["title"];
$shortdesc=$_POST["shortdesc"];
$rating=$_POST["rating"];
$image=$_POST["image"];
$mysql_qry="insert into
service(title,shortdesc,rating,image)values('$title','$shortdesc','$rating','$image')";
if($conn->query($mysql_qry)==TRUE)
{
echo "Added Successfully";
}
else
{
echo "Enter Valid Details";
}
$conn->close();
```

?>

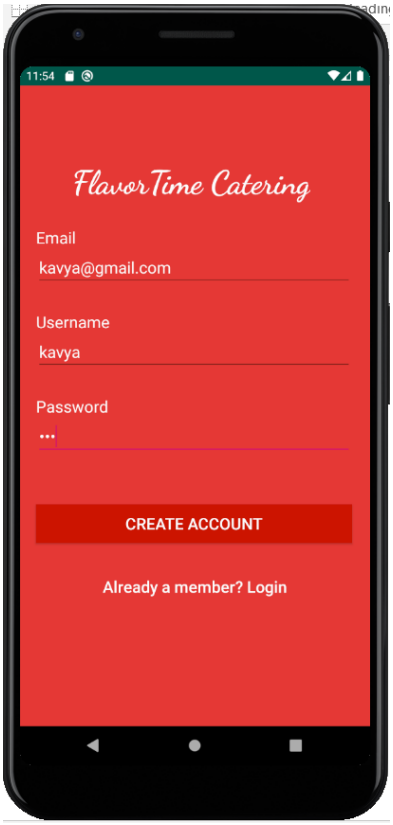
AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.cardview">
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:usesCleartextTraffic="true"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>b
        </activity>
        <activity android:name=".cardview">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:name=".top">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:name=".birthday">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:name=".login">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.DEFAULT" />
```

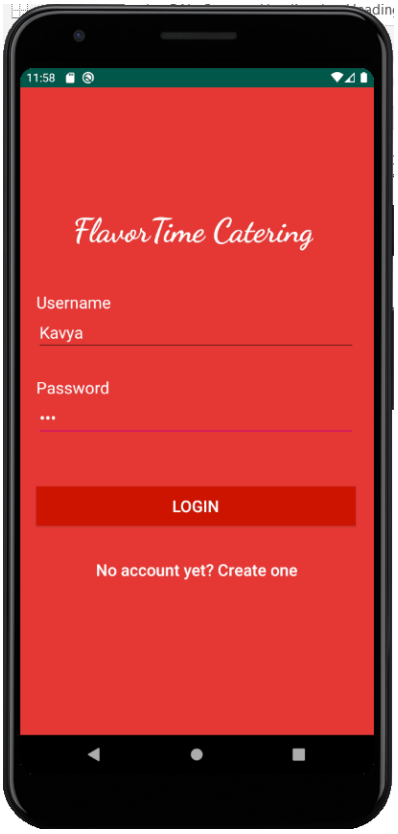
```
            </intent-filter>
        </activity>
        <activity android:name=".register">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:name=".wedding">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:name=".method">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:name=".payment">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity android:name=".feedback">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>


        <meta-data
            android:name="preloaded_fonts"
            android:resource="@array/preloaded_fonts" />
    </application>

</manifest>
```

5.2 Screenshots

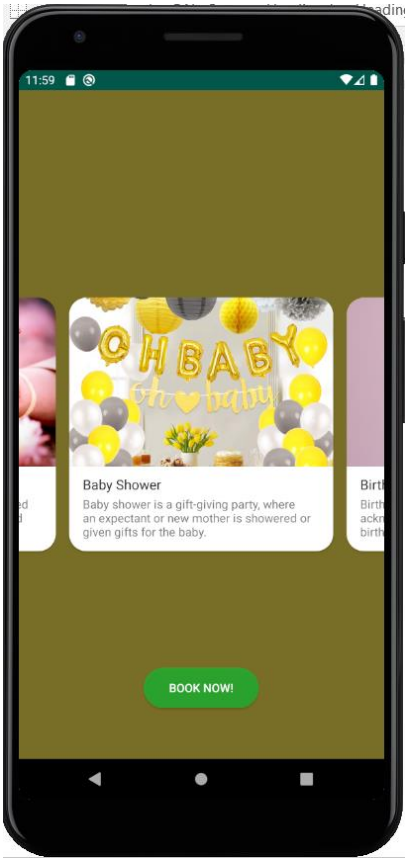
Splash Screen



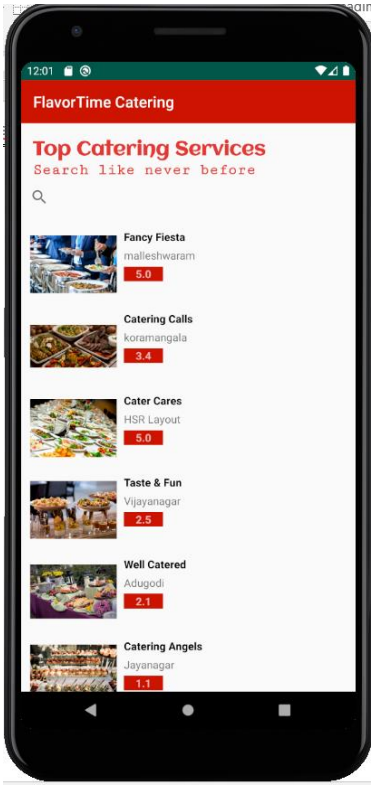
User Registration



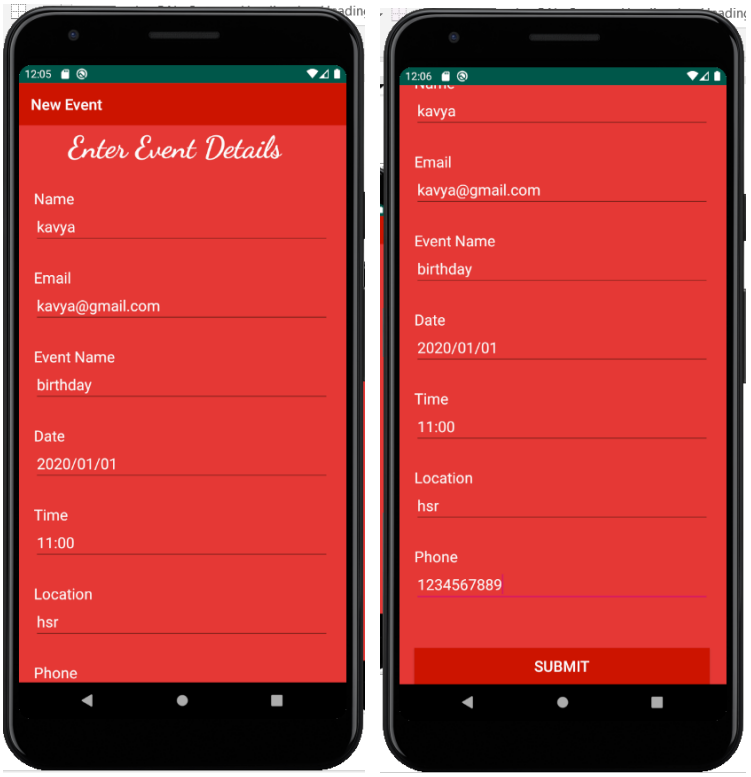
User Login



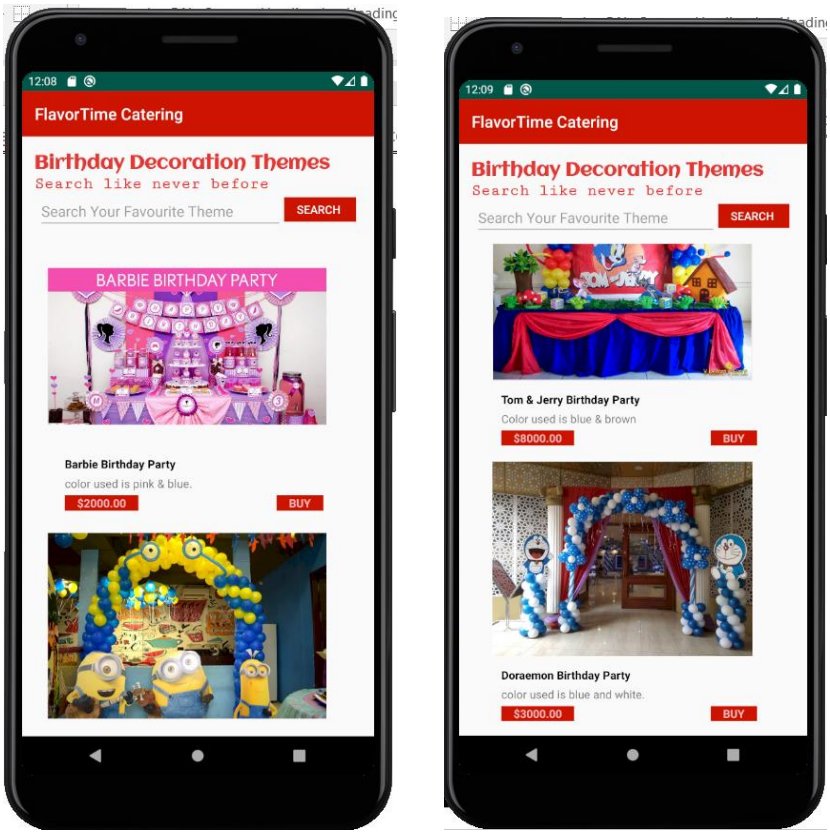
Choose Event



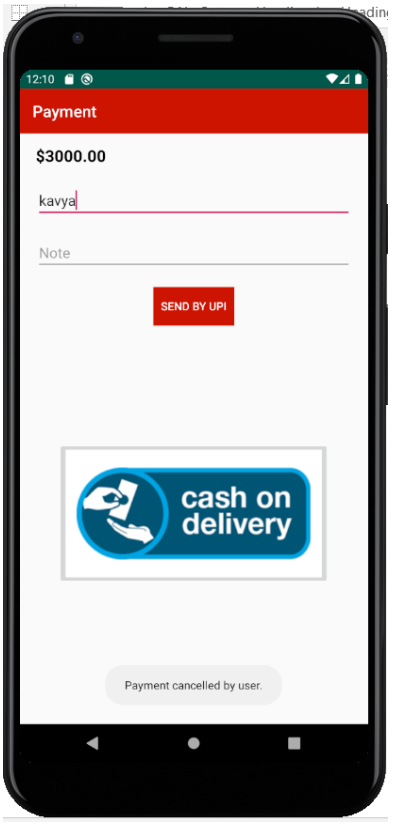
Catering Services



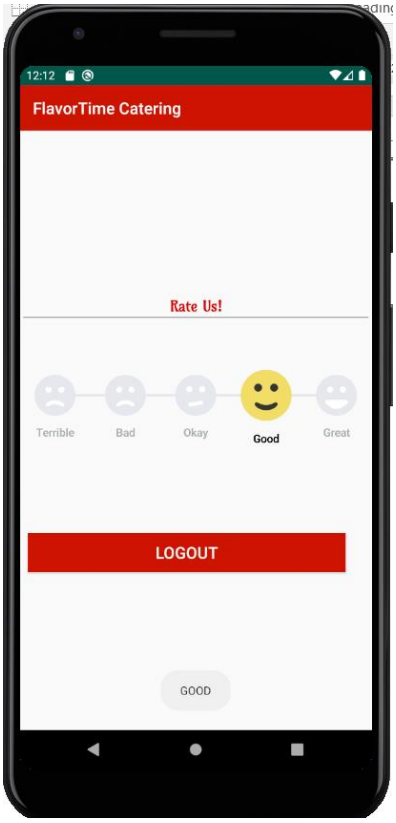
Event Details Form



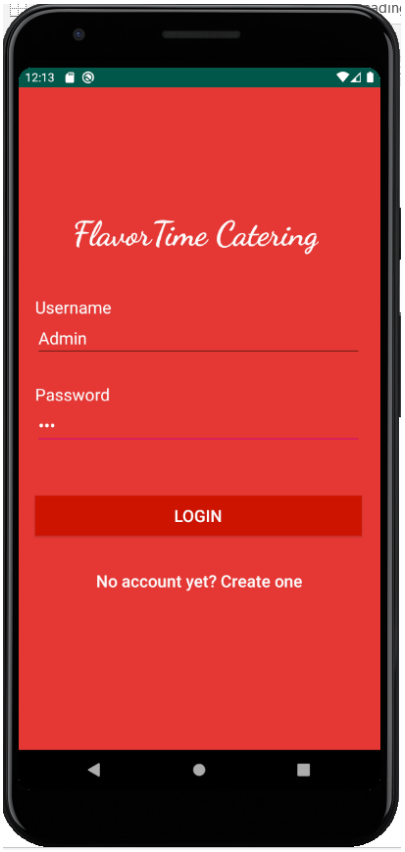
Birthday Theme & Buffet

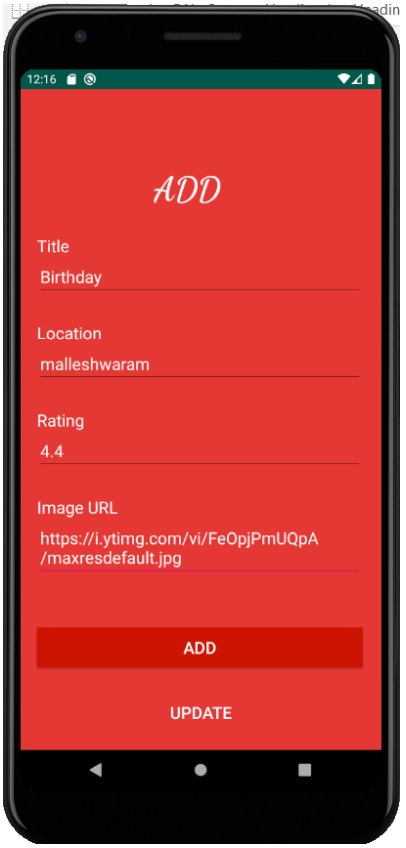


Payment Options



Feedback





Admin login & adding values

+ Options

	register_number	username	password	email
Edit Copy Delete	5	manju	789	manju@gmail.com
Edit Copy Delete	1	Arpitha	123	arpi@gmail.com
Edit Copy Delete	3	Archana	456	acchu@gmail.com
Edit Copy Delete	4	arp	1234	arphi@gmail.com
Edit Copy Delete	2	ar	789	ar@gmail.com
Edit Copy Delete	6	ARR	123	555a@gmail.com
Edit Copy Delete	7	Akarsh	456	aka@gmail.com
Edit Copy Delete	1237	sha	123	sha@gmail.com
Edit Copy Delete	1238	shh	123	shh@gmail.com
Edit Copy Delete	1239	sneha	123	sneha@gmail.com
Edit Copy Delete	1240	kavana	123	kavana@gmail.com
Edit Copy Delete	1241	kavya	123	kavya@gmail.com
Edit Copy Delete	1242	Divya	123	divya@gmail.com

Login and Registration Database

+ Options

	id	title	shortdesc	rating	image
Edit Copy Delete	1	Fancy Fiesta	malleshwaram	5.0	https://miro.medium.com/max/500/1*PKPkgjb1vRwfp4uf...
Edit Copy Delete	2	Catering Calls	koramangala	3.4	https://conferences.ucla.edu/wp-content/uploads/20...
Edit Copy Delete	3	Cater Cares	HSR Layout	5	https://www.theelegantweddings.com/wp-content/uplo...
Edit Copy Delete	6	Taste & Fun	Vijayanagar	2.5	https://cdn.fash.com/assets/images/bg/thumbs/weddi...
Edit Copy Delete	4	Well Catered	Adugodi	2.1	https://dinebydesign.net/wp-content/uploads/2019/0...
Edit Copy Delete	7	Catering Angels	Jayanagar	1.1	https://www.theprimavera.com/images/catering/off-p...
Edit Copy Delete	5	Delicate Dishes	Okalipuram	3.7	https://www.law.berkeley.edu/wp-content/uploads/20...
Edit Copy Delete	8	A Touch of Class	Nelamangala	5	https://ihouse.berkeley.edu/sites/default/files/st...
Edit Copy Delete	18	Fancy Food	Jayanagar	4.4	https://cdn.fash.com/assets/images/bg/thumbs/weddi...
Edit Copy Delete	20	Food Fiest	hsr	3.3	https://www.kenresearch.com/blog/wp-content/upload...

Catering Services Database

+ Options

	id	Name	Email	eventname	date	time	location	phone
Edit Copy Delete	1	Arpitha	arpi@gmail.com	wedding	2020-01-01	11:00:00.000000	#12,loc	1234567895
Edit Copy Delete	2	Arpitha	arpi@gmail.com	wedding	2020-01-01	11:00:00.000000	#12,loc	1234567895
Edit Copy Delete	3	usha	ushh@gmail.com	wedding	2020-01-01	11:00:00.000000	#11,bnglr	1234567895
Edit Copy Delete	4	Manjula	manju@gmail.com	wedding	2020-01-01	11:00:00.000000	#11, kor	1234567895
Edit Copy Delete	5	Manjula	manju@gmail.com	wedding	2020-01-01	11:00:00.000000	#11, kor	1234567895
Edit Copy Delete	6	Arp	arp@gmail.com	weddding	2020-01-01	11:00:00.000000	4, k	123
Edit Copy Delete	7	aka	aka@gmail.com	wedding	2020-01-01	11:00:00.000000	4,k	1234567895
Edit Copy Delete	17	Arp	ar@gmail.com	wedding	2020-01-01	11:00:00.000000	14, kormangla	1234567895
Edit Copy Delete	21	shh	shh@gmail.com	Birthday	2020-01-01	11:00:00.000000	14, kor	1234567890
Edit Copy Delete	22	Sneha	sneha@gmail.com	Birthday	2020-01-01	11:00:00.000000	#14, HSR	1234567890
Edit Copy Delete	23	Arpotha	ar@gmail.com	birthday	2020-01-01	11:00:00.000000	kormangla	1234567896
Edit Copy Delete	24	Arp	arpbirthday	birthday	2020-01-01	11:00:00.000000	kor	1234567895
Edit Copy Delete	25	arpitha	arp@gmail.com	birthday	2020-01-01	11:00:00.000000	hsr	1234567890
Edit Copy Delete	26	arp	arp@gmail.com	birthday	2020-01-01	11:00:00.000000	hsr	1234567890
Edit Copy Delete	27	kavana	kavana@gmail.com	birthday	2020-01-01	11:00:00.000000	hsr	1234567890
Edit Copy Delete	28	kavana	kavana@gmail.com	birthday	2020-01-01	11:00:00.000000	hsr	1234567890
Edit Copy Delete	29	kavya	kavya@gmail.com	birthday	2020-01-01	11:00:00.000000	hsr	1234567890
Edit Copy Delete	30	Divya	divya@gmail.com	birthday	2020-01-01	11:00:00.000000	hsr	1234567895
Edit Copy Delete	31	Divya	divya@gmail.com	birthday	2020-01-01	11:00:00.000000	hsr	1234567895
Edit Copy Delete	32	Divya	divya@gmail.com	birthday	2020-01-01	11:00:00.000000	hsr	1234567895
Edit Copy Delete	33	kavya	kavya@gmail.com	birthday	2020-01-01	11:00:00.000000	hsr	1234567889

Event Details Database

	id	theme	des	amt	image
Edit Copy Delete	1	Barbie Birthday Party	color used is pink & blue.	$2000.00	https://i.ytimg.com/vi/FeOpjPmUQpA/maxresdefault.j...
Edit Copy Delete	2	Minion Birthday Party	color used is yellow &dark blue.	$5000.00	https://deowgxgt4vwfe.cloudfront.net/uploads/15605...
Edit Copy Delete	3	Tom & Jerry Birthday Party	Color used is blue & brown	$8000.00	https://www.vdecorsandevent.com/images/THEMED%20BI...
Edit Copy Delete	4	Doraemon Birthday Party	color used is blue and white.	$3000.00	https://chinchincelebration.com/wp-content/uploads...

Birthday Event Buffet and Theme Database

SOFWARE TESTING

6.SOFTWARE TESTING

Various Testing Activities are as follows:

1. **Blackbox testing**: Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

2. **Whitebox testing**: This testing is based on knowledge of the internal logic of an applications code. Also known as Glass Box testing. Internal software and code working should be known for this type of testing.

3. **Unit testing**: Testing's of individual software components or modules. Typically done by programmers, not by testers, as it requires detailed knowledge of the internal program design and code may require developing test driver modules or test harnesses.

4. **Integration testing**: Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network.

5. **Functional testing**: This type of testing ignores the internal parts and focus on the output as per requirement or not.

6. **System testing**: Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

7. **End-End-testing**: Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

8. **Acceptance testing**: Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer to this testing to determine whether to accept the application.

9. **Usability testing**: User-friendliness check. Application flow is tested, Can new user understand application easily, Proper help documented whenever user struck at any point. Basically, system navigation is checked in this testing.

CONCLUSION AND FUTURE ENHANCEMENT

7. CONCLUSION AND FUTURE ENHANCEMENT

Conclusion:

By using this application, Customer should register first and login next by entering their Email Id and password. After login is successful, the user gets the event options to choose like Wedding, Birthday, House warming and Baby shower. Suppose if the user selects birthday event, he will get list of best top catering services with ratings.

Once the user selects any catering services, he will be navigated to the event details where he is supposed to fill the event details which includes date, time, event name, venue, phone number etc.

After submitting the details he will be able to choose the birthday decoration theme, then it will navigate to payment page, where the user have options to select online payment and cash on delivery, if he choose online payment he will get number of online payment app, after payment customer can give feedback and can logout.

Future Enhancements:

There is always a room for improvement in any software package, however good and efficient it may be. The important thing is that the application should be flexible enough for further modifications. Considering this important factor, the application is designed in such a way that the provisions are given for further enhancement.

REFERENCES

8.REFERENCES

BOOKS:

- **Android Beginning**
- **Android Professional**

https://www.javaTpoint.com
https://www.tutorialspoint.com
www.androidhive.info
https://stackoverflow.com
https://www.android.developers.com