ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

Event Planner Using Android Application

[1]G Geetha Devi, [2]SidraTakreem, [3]Rathod Soniya

[1]Assistant Professor, Department of Information Technology, Bhoj Reddy Engineering College for Women, Hyderabad, India

[2,3] Student, Department of Information Technology, Bhoj Reddy Engineering College for Women, Hyderabad, India

Abstract

This paper presents the design and implementation of an android application for managing various events. The software is supported to eliminate and in some cases reduce the hardships faced by any event organizer. It is mainly designed to carry out the operations smoothly and efficiently. Users will have to create accounts with their credentials. This software provides a user-friendly interface which allows the clients to contact service providers effectively. It also provides a note-keeping service which gives reminders time to time to the busy organizer. It makes it easy for the user and the clients to store their past expenses and events. This software makes the event organizer plan his event more strategically and provides better services to the clients.

***Keywords**: Event, android, google play, mobile, layouts, linear layouts, intents.*

1. Introduction

Event planning can be stressful. While you are watching through endless checklists and last- minute mishaps, it can be hard to make sure you are on track. As a professional event planner you should have the Super planner app that has a variety of planning tools. Our event planning software acts as a virtual event manager. From concerts to classes, plan and manage what you need for your event with ease. Event planning apps that will help you and your event team stay organized and save time planning your next event. So this is why we need this Event Management Application.

2. Related Work

The proposed project is a combination of two or more existing systems that are in the market, hence it is a hybrid project and hence is not available in the market.

In other words, the proposed project contains features that does not preexist or are not available in a single mobile application.

3. Proposed Work

The app enables the user (client) to search the desired services required to organize an event. It will contain information about different production houses, catering, entertainment, workforce, sponsors etc. with respect to the price and the location the user desires. It will also facilitate user with messaging service, notes for reminder and schedules. Different event organizers can advertise their event and also seek out for the need if volunteers and helpers. The user, the event organizers, merchants and the service provider will have a separate log–in portal leading them to different interface. The app will store previous user history and recommend the best suitable service to the user by analyzing the need of the user and also the given ratings to the service by other users [3].

Development Method

Rapid application development method aims to produce high quality system in low time cost schedule. (CMS, 2003).

4. Experimental Results

Fig. 1: Splash-screen

This layout consists of a welcome image for the user. The layout is attached to an activity called “SplashActivty” which runs a thread that sleeps for 2 seconds and destroys the activity while moving on to next activity.

XML Code:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayoutxmlns:android="http://schem
as.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"
android:background="@drawable/splash"
tools:context="com.example.eventmanagement.SplashActivity">
</android.support.constraint.ConstraintLayout>
```

As we can see in the above code, “LinearLayout” being the parent layout hosts the “splash” named image as background [4]. The height and width of the parent layout is set to „match_parent" meaning the layout is occupying the entire screen.

JAVA Code:

```
package com.example.eventmanagement;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.CountDownTimer;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class SplashActivityextendsAppCompatActivity {
CountDownTimerCDt;
SharedPreferencessharedPreferences;
SharedPreferences.Editoreditor;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_splash);
sharedPreferences= getSharedPreferences("Event",MODE_PRIVATE);
editor=sharedPreferences.edit();
CDt=new CountDownTimer(2000,1000) {
@Override
public void onTick(long l) {
         }
@Override
public void onFinish() {
intJ=sharedPreferences.getInt("login",0);
if(J==0) {
startActivity(new Intent(SplashActivity.this, MainActivity.class));
                  finish();
               }
else
{
startActivity(new Intent(SplashActivity.this, HomeActivity.class));
                  finish();
               }
            }
         }.start();
     }
}
```

As we can see in the above java code, we linked the XML layout[5] file of splash-screen "activity_splash.xml" to "Splash Activity.java" using

```
setContentView(R.layout.activity_splash);
```

As we need some event to trigger code executing we have onCreate method. This method is run when application is started. Inside of this method we will set the layout for application.

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

In the next part of the code we started a new thread.

```
new Thread() {
@Override
public void run()
         {
super.run();
}
}.start();
```

This thread is used to do the following things:

- Sleep for 2000ms (keep the current running thread on hold)
- Destroying the current (Splash Screen) activity while moving onto next activity using „intents" in android[6]

startActivity(**new** Intent(SplashActivity.**this**,HomeActivity.**class**));
finish();

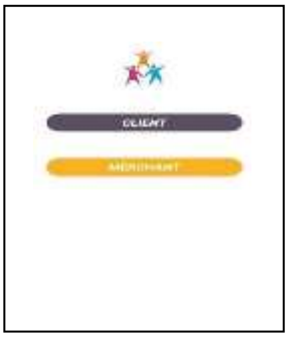


Fig. 2: Client merchant screen

In this above Login page screen we have two edit texts and the three buttons[7].Here the edit texts are the user name and the other is the password boxes to enter the credentials in it to get login and the Login button is here to get login when clicked and moves to the home screen. Register button is used to move to the register screen and the Admin button is here to move to the admin screen when clicked [10].

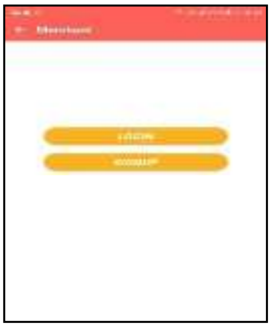


Fig. 3: Merchant screen

In this Register screen we have various columns to enter our personal details. After entering the details when we click on the register button then the details about us will be get registered and the account can be opened for us which can be signed in through the login screen that which appears in text page followed by this.

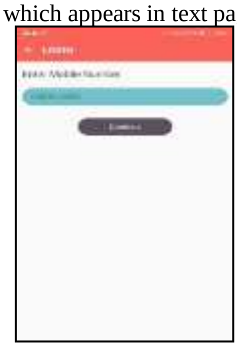


Fig. 4: Merchant login screen

In this above Login page screen we have two edit texts and the Login button[8].Here the edit texts are the user name and the other is the password boxes to enter the credentials in it to get login and the Login button is here to get login when clicked and moves to the home screen[9].

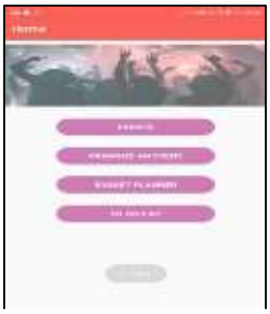
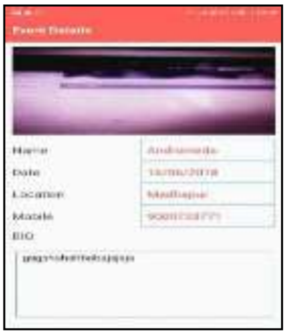


Fig. 5: Home screen

This screen shows the home screen once merchant or user gets login.



Event Details

Name Andromeda
Date 13/06/2018
Location Madhapur
Mobile [illegible]
BIO
[illegible]

Fig. 6: Event details entry screen

The details of events can be added in this page like name, date, location, mobile etc.

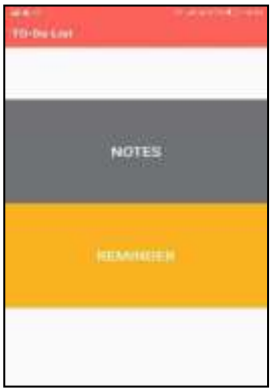


Fig. 7: To-do list screen

Any list of activities to be done or remainders can be added with the help of this page.

5. Conclusion

This paper presents the design and implementation of an android application for managing various events. In this work, the proposed prototype that is developed can perform various activities by saving the data into the available database and also can be used for retrieving the data. In this work, the user will have an option to select any of the events presented.

6. Future Scope

It is possible to add more functionality for the application. Organize private groups of developers and idea makers. Add other social network support. Create one of the web sources for example a web page which can be accessed from other devices with different OS. Registering application with Google service and uploading it into Google Play Store.

References

[1] Android Developers, 2013. http://developer.android.com/sdk/.

[2] Cnet, JRE, 2013. http://download.cnet.com/Java-Runtime-Environment-JRE/30002378_410009607.html#ixzz2mLmJ28Kg.

[3] CMS, 2013. http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-informationTechnology/XLC/Downloads/Selecting DevelopmentApproach.pdf.

[4] Holla S & Katti MM, "Android based mobile application development and its security", *International Journal of Computer Trends and Technology*, Vol.3, No.3,(2012), pp.486-490.

[5] Kathuria A & Gupta A, "Challenges in Android Application Development: A Case Study", *International Journal of Computer Science and Mobile Computing*, Vol.4, No.5, (2015), pp.294-299.

[6] Android Developer site, http://developer.android.com/index.html

[7] Surachat K, Kajkamhaeng S, Damkliang K, Tiprat W & Wacharanimit T, "First Aid Application on Mobile Device", *World Academy of Science, Engineering and Technology, International Science Index*, Vol.7 , No.5, (2013).

[8] Joorabchi ME, Mesbah A & Kruchten P, "Real Challenges in Mobile App Development", *ACM/IEEE International Symposium Empirical Software Engineering and Measurement*, (2013), pp.15-24.

[9] Dehlinger J & Dixon J, "Mobile application software engineering: Challenges and research directions", *Workshop on mobile software engineering*, Vol.2, (2011), pp.29-32.

[10] Jose DV, Lakshmi Priya C, Priyadarshini G & Singh M, "Challenges and Issues inAndroid App Development-An Overview", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol.5, No.1, (2015).