Project Report

On

Currency converter Application

Submitted by

P.Suseela
P.Gowri Varalakshmi
G.Reshma

Under the guidance of

P.Siva Lakshmi

Department of Computer Science and Engineering

Rajiv Gandhi University of Knowledge and Technologies(RGUKT),

RK.Valley.Kadapa, Andhra Pradesh

Rajiv Gandhi University of Knowledge Technologies

RK.Valley, Kadapa(Dist) , Andhra Pradesh, 516330

CERTIFICATE

This is to certify that the project work titled “CURRENCY CONVERTER APPLICATION “ is a bonafied project work submitted by P.Suseela, P. Gowri Varalakshmi , G.Reshma , in the department of COMPUTER SCIENCE AND ENGINEERING in partial fulfiilment of requirements fpr the award of degree of Bachelor of Technology in Computer Science and Engineering for the year 2021-2022 carried out the work under the supervision.

GUIDE

HEAD OF THE DEPARTMENT

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I am extremely grateful to our respected Director,Prof. K. SANDHYA RANI for fostering an excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head of the Department Mr.P.HARINADH for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college MRS.P.SIVA LAKSHMI for his guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

INDEX

Abstract

Currency is a primary medium of exchange within the times, having way back replaced bartering as a way of trading goods and services. Due to Globalization, there is a tremendous growth of businesses nationally as well as internationally.

So, with different nations, they have their different currency values. So, to deal with these fluctuating currency rates, having a mobile application with the latest exchange rates also with a converter which can be accessed anytime anywhere is very helpful. CurrencyX deals with the development of an android-based application for the conversion of different currencies along with getting the latest exchange rates.

The main aim of this proposedapplication is to provide all the features in a single place, accessible anytime anywhere. The application consists of the following modules, namely:(i) Converter (ii)Chart (iii)News (iv)Payment Gateway. The data is fetched through API and the Chart is displayed using the MPAndroidChart library. This application can be used by international tourists, forex traders, and analysts.

INTRODUCTION

Nowadays, money plays an important factor in various aspects from sociological to economical parts of life. The value of this money varies from country to country depending on the exchange rate value it holds in the global market and also, it is determined by the demand for it, a bit like the worth of products and services. Many people don't concentrate to exchange rates because rarely do they have to. The typical person's lifestyle is conducted in their domestic currency. Exchange rates only inherit a focus for infrequent transactions, like foreign travel, import payments, or overseas remittances.

Hence to monitor and analyse the exchange rates of various currencies, many currency exchange systems came into existence. These exchange systems were early available for the companies or various institutions. But as there were advancements in technology, most of the tasks are being on the mobile itself, so currency converter applications with much more features were made the use by people across the globe. Mobile Application made it simplier accessible anytime and anywhere.

The application comprises the following major modules:

Converter module- The converter module is important as it serves as the homepage of the application. It consists of a list of currencies with their country flag and currency symbol. Euro is the base currency with the other three currencies like the Indian rupee, American dollar, and Australian dollar when the application is launched for the very first time. As the user edits the amount of the base currency in the input, simultaneously the amount value of the other

currencies gets updated. The list has 32 international currencies on which the user can add, remove and clear the list of currencies up to his wish.

Chart module- Chart modules display a line graph of the currencies rates against time which ranges from present-day to a maximum of five years. It can also be customized to view a range of weeks, months, quarters, or years. The currency selected from the list and base currency chart is displayed by default but can be changed by adding another currency from the list. The chart displayed can be inverted too. The chart responds to two-finger pinch and expands gestures, also will scroll when expanded. All these features in the chart are implemented using the MPAndroidChart library- which is an open-source Android chart view/graph view library using which you can draw a line, bar, pie, radar, bubble,candlestick charts.

PURPOSE

The main purpose of currency converter Application is to provide for the qucik conversion of any currency into any other currency. Currency conversion is of practical use to tourists who travel to abroad to businesses. Who do business over seas or are involved in imports and exports.

SCOPE

Different countries use different currency and their is daily variations in these currencies relative to one another. Those who transfer money from one country to another must be updated with the latest currency exchange rates in the market. Such application can be used by any user , but it is mainly useful for business, shares and finance related areas.

Advantages:

- A currency conversion is a service that gathers data from multiple sources to give a single
  place where you can find all information you need.
- This way ,you won't have to spend hours searching for the data througout the internet.
- It is easy to use-- just sign up for an account and you can start using its services
-

Disadvantages:

- Some APIs don’t provide enough information about their services or pricing options.
- And also don’t provide accurate results when converting currencies due to fluctuations in exchange rates.

Requirement Specification

Hardware Configuration:

Client Side:

Ram	512 MB
Hard disk	10GB
Processor	1.0 GHz

Server Side:

Ram	1GB
Hard disk	20GB
Processor	2.0GHz

Software Requirements:

Front end	xml
Server side language	Java
Operating system	windows
Software	Andriod app

Android Studio :

It is the official Integrated Development Environment (IDE) for android application development. Android Studio provides more features that enhance our productivity while building Android apps.

Android Studio was announced on 16th May 2013 at the Google I/O conference as an official IDE for Android app development. It started its early access preview from version 0.1 in May 2013. The first stable built version was released in December 2014, starts from version 1.0.

Since 7th May 2019, Kotlin is Google's preferred language for Android application development. Besides this, other programming languages are supported by Android Studio.

Features of Android Studio

•It has a flexible Gradle-based build system.

•It has a fast and feature-rich emulator for app testing.

•Android Studio has a consolidated environment where we can develop for all Android devices.

•Apply changes to the resource code of our running app without restarting the app.

•Android Studio provides extensive testing tools and frameworks.

•It supports C++ and NDK.

•It provides build-in supports for Google Cloud Platform. It makes it easy to integrate Google Cloud Messaging and App Engine.

JAVA :

The primary objective of java programming

language creation was to make it portable, simple and secure programming language. Apart from this, there are also some excellent features which play an important role in the popularity of this language. The features of Java are also known as Java buzzwords.

A list of the most important features of the Java language is given below:

1.Secure

2.Robust

3.Simple

4.Portable

5.Platform Independent

XML:

(Extensible Markup Language) is **a markup language similar to HTML, but without predefined tags to use**. Instead, you define your own tags designed specifically for your needs. This is a powerful way to store data in a format that can be stored, searched, and shared.

It is **a simple text-based format for representing structured information: documents, data, configuration, books, transactions, invoices, and much more**. It was derived from an older standard format called SGML (ISO 8879), in order to be more suitable for Web use.

Analysis and Design

Analysis:

As technology is growing rapidly we are also moving to a technical world where everything we want to be online .So with the help of this project we are bringing the use of technology in the field of mobile application. By using this mobile application we can easily convert our currencies in couple of minutes.

Design Introduction:

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software. The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

UML Diagrams:

Actor:

A coherent set of roles that users of use cases play when interacting with the use cases.an

observable result of value of an actor.

Use case: A description of sequence of actions, including variants, that a system performs yields an observable result of value of an actor. actor diagram is drawned in a eclipse shape

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

USECASE DIAGRAMS:

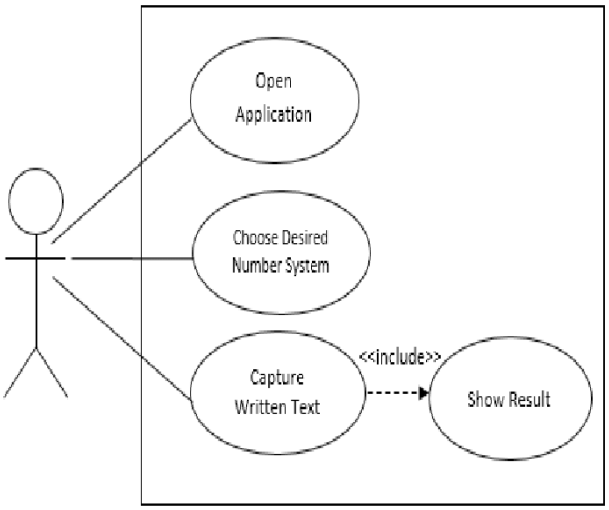
Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor. Use case diagram can be useful for getting an overall view of the system and clarifying that can do and more importantly what they can't do. Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

- The purpose is to show the interactions between the use case and actor.
- To represent the system requirements from user's perspective.
- An actor could be the end-user of the system or an external system.

USECASE DIAGRAM: A Use case is a description of set of sequence of actions. Graphically

it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor – Sender, Secondary Actor Receiver.

UML diagram



ER Diagram:

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

 It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.

 It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.

 In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

ER Notation:

There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there area number of notations used; among the more common are Bachman, crow's foot, and IDEFIX. All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. Thenotation used in this document is from Martin. The symbols used for the basic ER constructs are:

Entities are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.

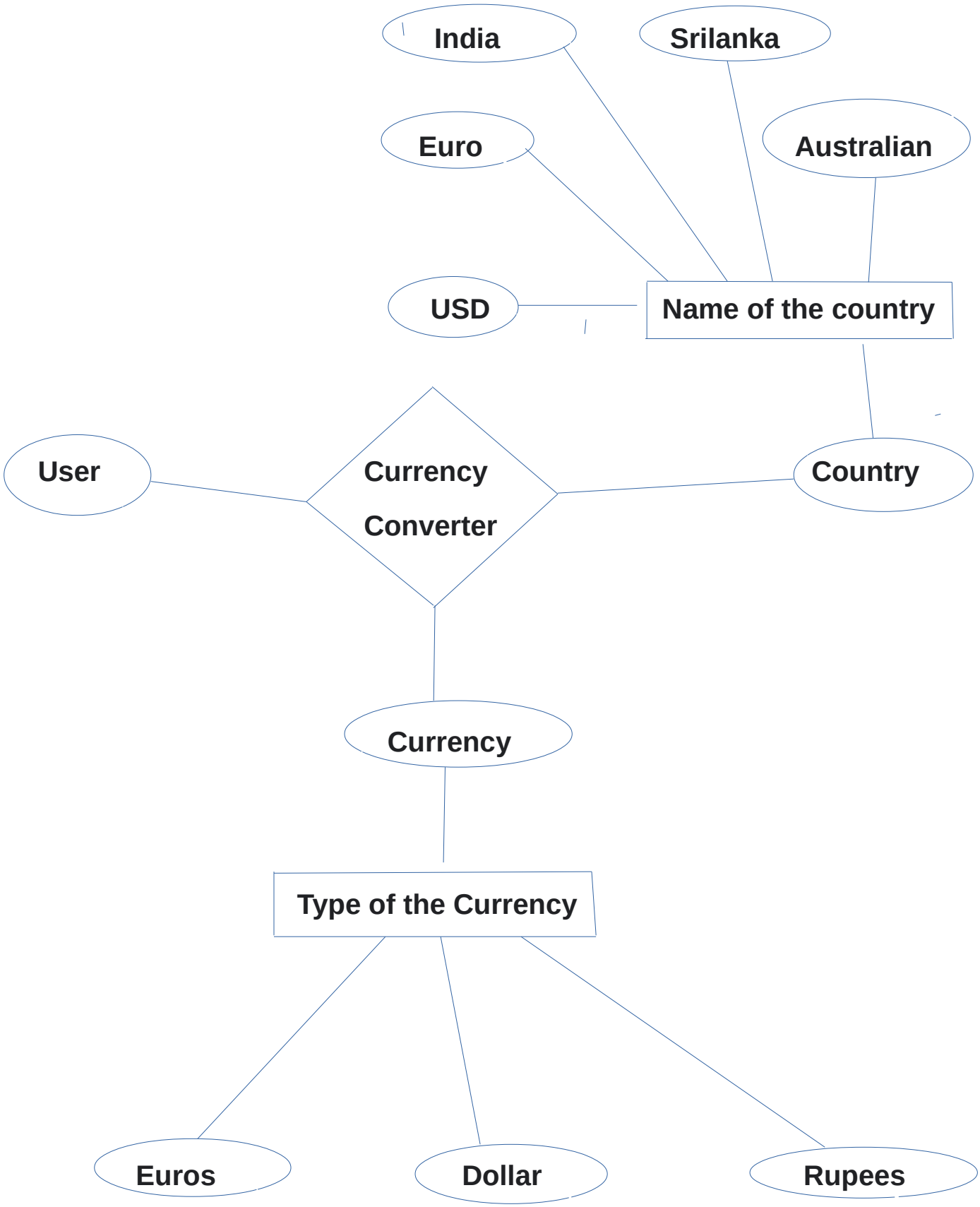
Relationships are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs

Attributes, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.

Cardinality of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.

Existence is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.

ER Diagram:


India
Srilanka
Euro
Australian
USD
Name of the country
User
Currency
Converter
Country
Currency
Type of the Currency
Euros
Dollar
Rupees

Code:

activity_main.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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="40sp"
        android:text="Currency_Converter"/>

    </LinearLayout>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="30sp"
                android:text="Enter the amount"
                android:textColor="@color/design_default_color_primary"
                android:textSize="20sp" />

            <EditText
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:ems="10"
                android:layout_marginTop="20sp"
                android:id="@+id/txtamount"/>

        </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="20sp"
```

```
            android:layout_marginTop="30sp"
            android:text="From"
            android:textColor="@color/design_default_color_primary"/>




        <Spinner
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:layout_marginTop="37sp"
            android:id="@+id/spfrom"/>

    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="20sp"
            android:layout_marginTop="30sp"
            android:text="To"
            android:textColor="@color/design_default_color_primary"/>

        <Spinner
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:layout_marginTop="37sp"
            android:id="@+id/spto"/>

    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="30sp"
            android:text="Convert"
            android:id="@+id/btn1"/>

    </LinearLayout>

    <TextView
        android:id="@+id/tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20sp"
        android:layout_marginTop="30sp"
        android:textColor="@color/design_default_color_primary"
        android:textSize="30sp"
        android:textStyle="bold" />
```

```
</LinearLayout>
```

MainActivity.java

```
package com.example.fourth;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {

    Spinner sp1, sp2;
    EditText ed1;
    Button b1;
    TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ed1 = findViewById(R.id.txtamount);
        sp1 = findViewById(R.id.spfrom);
        sp2 = findViewById(R.id.spto);
        b1 = findViewById(R.id.btn1);
        textView = findViewById(R.id.tv);

        String[] from = {"Australian Dollar", "Euro", "Indian Rupees", "Srilankan Rupees",
"USD"};
        ArrayAdapter ad = new ArrayAdapter<String>(this,
androidx.appcompat.R.layout.support_simple_spinner_dropdown_item,from);

        sp1.setAdapter(ad);

        String[] to = {"Australian Dollar", "Euro", "Indian Rupees", "Srilankan Rupees", "USD"};
        ArrayAdapter ad1 = new ArrayAdapter<String>(this,
androidx.appcompat.R.layout.support_simple_spinner_dropdown_item,to);

        sp2.setAdapter(ad1);

        b1.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {

                Double tot;
                Double amount = Double.parseDouble(ed1.getText().toString());

                if(sp1.getSelectedItem().toString() == "Australian Dollar" &&
sp2.getSelectedItem().toString() == "Euro")
                {
                    tot = amount * 0.68;
                    textView.setText(tot.toString());
                }
```

```
                    else if(sp1.getSelectedItem().toString() == "Australian Dollar" &&
sp2.getSelectedItem().toString() == "Indian Rupees")
                    {
                          tot = amount * 53.53;
                          textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "Australian Dollar" &&
sp2.getSelectedItem().toString() == "Srilankan Rupees")
                    {
                          tot = amount * 239.07;
                          textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "Australian Dollar" &&
sp2.getSelectedItem().toString() == "USD")
                    {
                          tot = amount * 0.67;
                          textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "Australian Dollar" &&
sp2.getSelectedItem().toString() == "Australian Dollar")
                    {
                          tot = amount * 1;
                          textView.setText(tot.toString());
              }

                    else if(sp1.getSelectedItem().toString() == "Euro" &&
sp2.getSelectedItem().toString() == "Australian Dollar")
                    {
                          tot = amount * 1.49;
                          textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "Euro" &&
sp2.getSelectedItem().toString() == "Indian Rupees")
                    {
                          tot = amount * 79.78;
                          textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "Euro" &&
sp2.getSelectedItem().toString() == "Srilankan Rupees")
                    {
                          tot = amount * 356.33;
                          textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "Euro" &&
sp2.getSelectedItem().toString() == "USD")
                    {
                          tot = amount * 1.00;
                          textView.setText(tot.toString());
                    }



                    else if(sp1.getSelectedItem().toString() == "Euro" &&
sp2.getSelectedItem().toString() == "Euro")
```

```
                    {
                          tot = amount * 1;
                          textView.setText(tot.toString());
                     }

                    else if(sp1.getSelectedItem().toString() == "Indian Rupees" &&
sp2.getSelectedItem().toString() == "Australian Dollar")
                    {
                          tot = amount * 0.019;
                          textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "Indian Rupees" &&
sp2.getSelectedItem().toString() == "Euro")
                    {
                          tot = amount * 0.013;
                          textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "Indian Rupees" &&
sp2.getSelectedItem().toString() == "Srilankan Rupees")
                    {
                          tot = amount * 4.47;
                          textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "Indian Rupees" &&
sp2.getSelectedItem().toString() == "USD")
                    {
                          tot = amount * 0.013;
                          textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "Indian Rupees" &&
sp2.getSelectedItem().toString() == "Indian Rupees")
                    {
                          tot = amount * 1;
                          textView.setText(tot.toString());
          }

                    else if(sp1.getSelectedItem().toString() == "Srilankan Rupees" &&
sp2.getSelectedItem().toString() == "Australian Dollar")
                    {
                          tot = amount * 0.0042;
                          textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "Srilankan Rupees" &&
sp2.getSelectedItem().toString() == "Euro")
                    {
                          tot = amount * 0.0028;
                          textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "Srilankan Rupees" &&
sp2.getSelectedItem().toString() == "Indian Rupees")
                    {
                          tot = amount * 0.22;
                          textView.setText(tot.toString());
                    }
```

```
                    else if(sp1.getSelectedItem().toString() == "Srilankan Rupees" &&
sp2.getSelectedItem().toString() == "USD")
                    {
                        tot = amount * 0.0028;
                        textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "Srilankan Rupees" &&
sp2.getSelectedItem().toString() == "Srilankan Rupees")
                    {
                        tot = amount * 1;
                        textView.setText(tot.toString());
            }

                    else if(sp1.getSelectedItem().toString() == "USD" &&
sp2.getSelectedItem().toString() == "Australian Dollar")
                    {
                        tot = amount * 1.48;
                        textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "USD" &&
sp2.getSelectedItem().toString() == "Euro")
                    {
                        tot = amount * 1.00;
                        textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "USD" &&
sp2.getSelectedItem().toString() == "Indian Rupees")
                    {
                        tot = amount * 70.0;
                        textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "USD" &&
sp2.getSelectedItem().toString() == "Srilankan Rupees")
                    {
                        tot = amount * 180.0;
                        textView.setText(tot.toString());
                    }

                    else if(sp1.getSelectedItem().toString() == "USD" &&
sp2.getSelectedItem().toString() == "USD")
                    {
                        tot = amount * 1;
                        textView.setText(tot.toString());
            }

                }
            });
        }
}
```

Implementation and System Testing

After all phase have been perfectly done, the system will be implemented to the server and the system can be used.

System Testing:

The goal of the system testing process was to determine all faults in our project .The programwas subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1. Unit testing

2 .Integration testing

Unit Testing

Unit testing is commenced when a unit has been created and effectively reviewed .In order to test a single module we need to provide a complete environment i.e. besides the section we would require The procedures belonging to other units that the unit under test calls Non local data structures that module accesses .A procedure to call the functions of the unit under test with appropriate parameters

1. Test for the admin module

Testing admin login form-This form is used for log in of administrator of the system. In this form we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask the details.

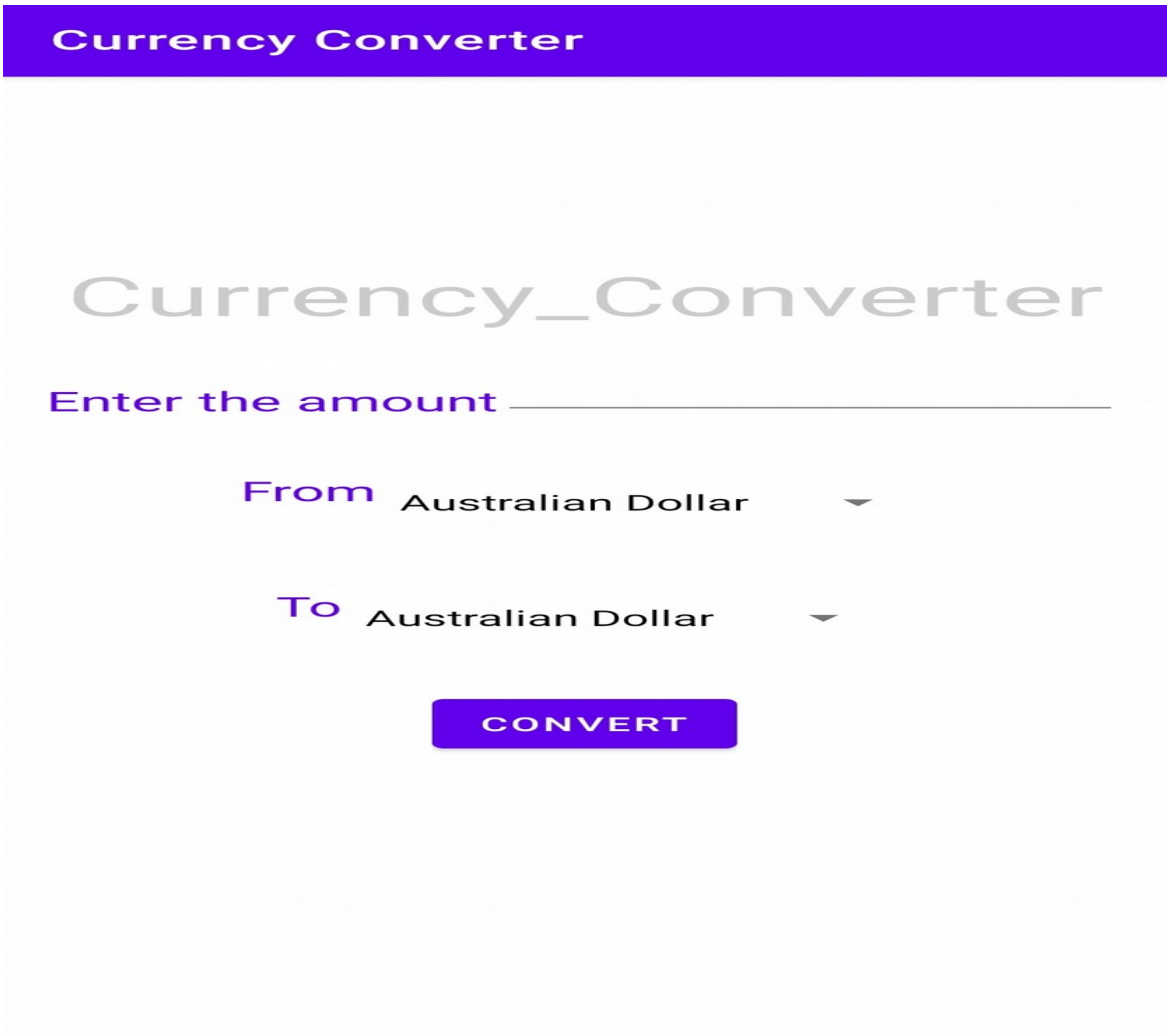
Report Generation: admin can generate report from the main database.

Integration Testing

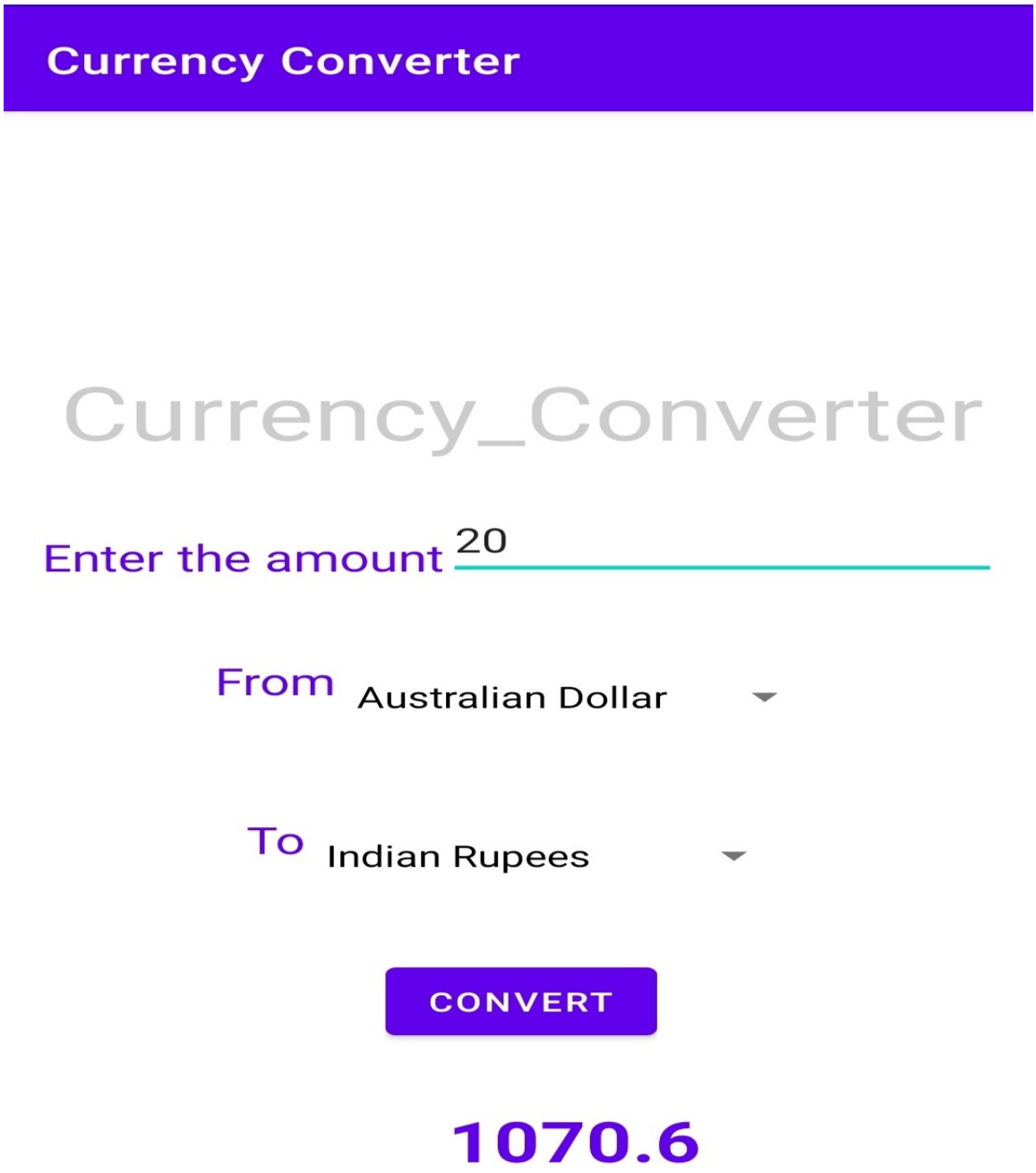
In the Integration testing we test various combination of the project module by providing the input. The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes the other module.

Evaluation

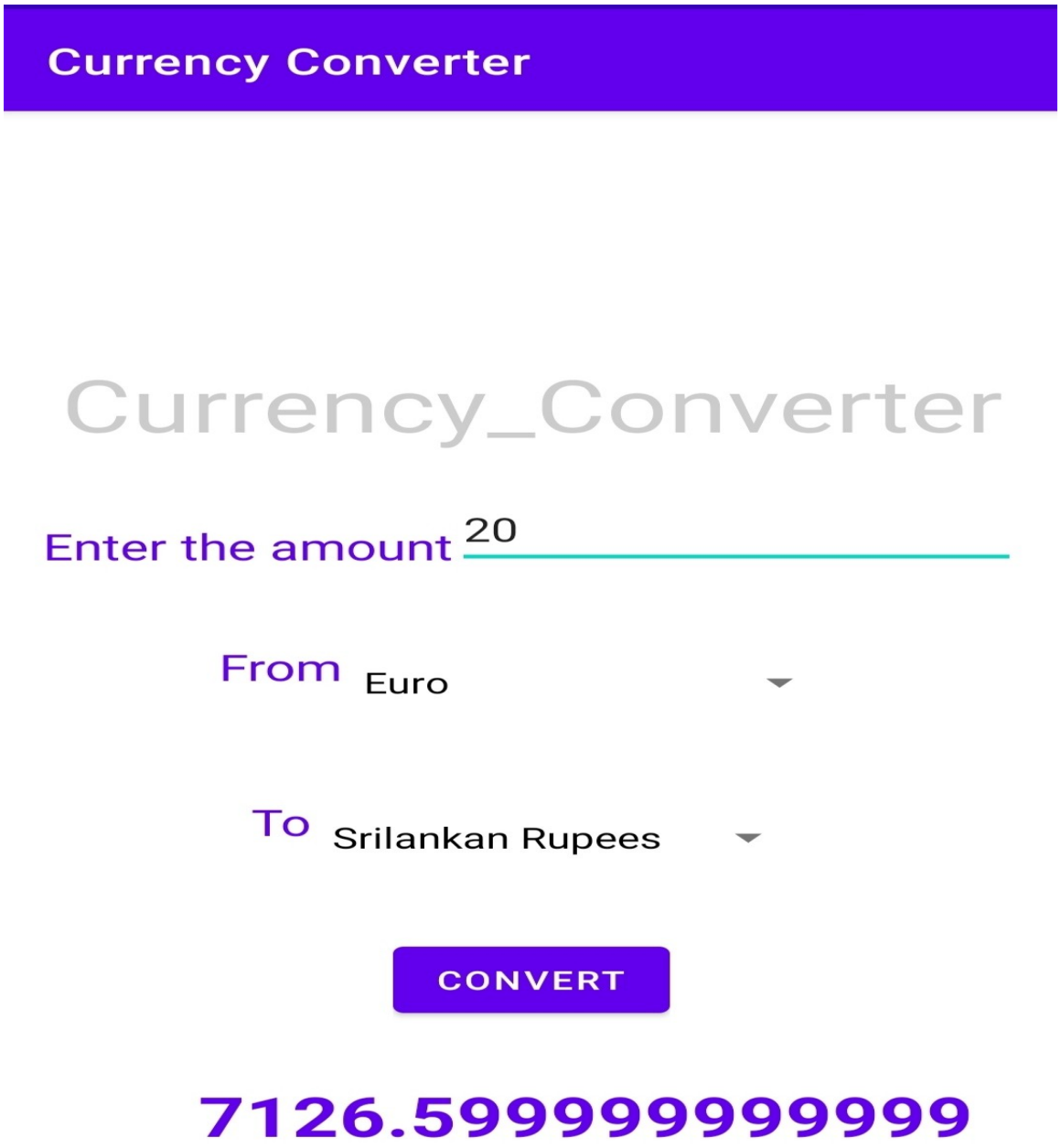
Home Page


Currency Converter
Currency_Converter
Enter the amount
From
Australian Dollar
To
Australian Dollar
CONVERT

Australia to Indian Rupees:



Euro to Srilankan



Indian to USD(US Dollars)

Conclusion:

CurrencyX app will help to fetch, analyze and learn all the information needed regarding foreign currencies under one roof. As compared to many other possible solutions, our proposal is much more feasible and easier to implement. As the system is dynamic in nature, it is easy for future modifications. Apart from just mere converting values, this application provides alerts daily and a payment gateway for making transactions in international markets. CurrencyX App can be used by everyone which includes forex traders, international tourists, analysts with convenience for social and business purposes. Development was done making use of available tools, techniques, while making the system, an eye has been kept on making it as user-friendly. The application has been tested with live data and has provided a successful result. Hence the software has proved to work efficiently.

References:

- Get **Currencies** API response in your terminal by running command `curl http://localhost:8080/currencies`. Provides a list of currencies for which Mastercard publishes rates every day.
- Get **Conversion Rates Status** API response in your terminal by running command `curl http://localhost:8080/status-rates`. Determines whether Mastercard conversion rates are available (issued) for a given date.
- Get **Conversion Rate Details** API response in your terminal by running command `curl http://localhost:8080/transaction-settlement-rate-details`. Provides the conversion rate and calculated settlement amount based on a given date, transaction amount, currency and settlement currency.