VISVESVARAYA TECHNOLOGICALUNIVERSITY

BELAGAVI, KARNATAKA, INDIA

A MINI-PROJECT REPORT

ON

”CAR EMI CALCULATOR”

Submitted in partial fulfillment of the requirement for the VI semester BE in
Information Science and Engineering
Mobile application development mini-project-18CSM6P68

Submitted by

ROHITH GANAPATHI BHAT	*[1SG19IS086]*
S.KAUSHAL	*[1SG19IS087]*
SARVAJIT ASHOK BHAT	*[1SG19IS091]*
SHASHANK.K.G	*[1SG19IS093]*

Under the guidance of

Prof. SUDRASHANAN D
DESIGNATION
Dept. of I.S.E., S.C.E.

SAPTHAGIRI
College of Engineering
Creating Tomorrow

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
SAPTHAGIRI COLLEGE OF ENGINEERING
Bengaluru-57

2021-22

SAPTHAGIRI COLLEGE OF ENGINEERING

14/5, Chikkasandra, Hesaraghatta Main Road, Bengaluru-560057

[*AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI*]

Department of Information Science & Engineering

CERTIFICATE

Certified that the project work entitled **"Car EMI calculator"** carried out by **ROHITH GANAPATI BHAT (1SG19IS086),S.KAUSHAL(1SG19IS087), SARVAJITH ASHOK BHAT(1SG19IS091),SHASHANK.K.G(1SG19IS093)** bonafide studentsof 6th semester, department of **Information Science & Engineering** carried out at our college **Sapthagiri College of Engineering**, Bengaluru in partial fulfillment for the 6th Semester BE, MOBILE APPLICATION DEVELOPMENT Mini-Project-18CSMP68 by **Visvesvaraya Technological University,** Belagavi during the year 2020-21. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

.

Signature of the Guide

Prof.SUDARSAN D

Lab incharge

Signature of the HOD

Dr H R Ranganatha

HOD, I.S.E., S.C.E.

External Viva

Name of the Examiners

1. ……………………………………

2. ……………………………………

Signature of the Examiners with date

1………………………..

2…………………………

ABSTRACT

This is an integrated application for calculating EMI for buying a car.However not all are able to do so, those who have large families are more likely to pay in monthly installments. For example for car loan payment. Therefore our group has developed an application that will facilitate their affairs that cannot afford for the full installment for the car itself. We developed an application called “Car Loan Calculator” to reduce their problem.

The purpose of this application is to help people who want to purchase a car by using monthly installments. With this application also, people will find out if they can afford their dream car using this user friendly car loan calculator. by using this car loan calculator, user just need to insert total amount of car, down payment, interest rate and loan period then the application will calculate to the user how much user need to pay monthly within a certain period of time.

ACKNOWLEDGEMENT

Any achievement doesn't depend solely on the individual efforts but on the guidance, encouragement and co-operation of intellectuals, elders and friends. A number of personalities have helped us. We would like to take this opportunity to thank them all.

We would like to express our heart-felt gratitude to **Dr. H Ramakrishna,** Principal, Sapthagiri College of Engineering, Bengaluru, for his help and inspiration during the tenure of the course.

It is great privilege to extend our deep sense of gratitude to **Dr. H R Ranganatha,** Head of the Department, Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru, who patronized throughout our career, for his constant support and encouragement and for the facilities provided to carry out this work successfully.

We wish to express our sincere thanks to our guide **Prof SUDARSANAN D,** DESIGNATION, Department of Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru for helping us throughout and guiding us from time to time. We also extend our sense of gratitude and sincere thanks to non-teaching staff members of Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru for their views and encouraging ideas.

Finally, we also thank our family and friends for their co-operation and motivation.

ROHITH GANAPATI BHAT	(1SG19IS086)
S KAUSHAL	(1SG19IS087)
SARVAJIT ASHOK BHAT	(1SG19IS091)
SHASHANK K G	(1SG19IS093)

TABLE OF CONTENTS

Chapter 1: INTRODUCTION 1

Chapter 2: SYSTEM ANALYSIS 5

2.1 Literature Survey 5

2.2 Existing System 5

2.3 Proposed System 6

2.3.1 Scope of the Project 6

2.3.2 Aim of the Project 7

Chapter 3: REQUIREMENT SPECIFICATIONS 8

3.1 System Requirements 8

3.1.1 Hardware Configuration 8

3.1.2 Software Configuration 8

3.2 Development Environment 8

Chapter 4: SYSTEM DESIGN 11

4.1 Project Flow Chart 11

4.2 Design using XML 12

Chapter 5: SYSTEM IMPLEMENTION 22

Chapter 6: SNAPSHOTS 57

CONCLUSION AND FUTURE ENHANCEMENT 64

REFERENCES 65

LIST OF FIGURES

Figure No.	Figure name	Page No.
3.1	STRUCTURE LAYOUT OF ANDROID STUDIO	09
4.1	FLOW CHART OF THE PROJECT	11

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014.The first stable build was released in December 2014, starting from version 1.0.

On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Javais still supported, as is C++.

A specific feature of the Android Studio is an absence of the possibility to switch autosave featureoff. The following features are provided in the current stable version:

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to previewlayouts on multiple screen configurations
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.

- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows 7 or later. As of March 2015, the SDK is not available on Android itself, but software development is possible by using specialized Androidapplications.

Until around the end of 2014, the officially-supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. As of 2015, Android Studio made by Google and powered by IntelliJ, is the official IDE; however, developers are free to use others, but Google made it clear that ADT was officially deprecated since the end of 2015 to focus on Android Studio as the official Android IDE. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache ANT are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

1.2 INTRODUCTION TO XML

Extensible Markup Language (**XML**) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The World Wide Web Consortium's XML 1.0 Specification of 1998 and several other related specifications—all of themfree open standards—define XML.The design goals of XML emphasize simplicity, generality, and usability across the Internet.It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary datastructures such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while programmers havedeveloped many application programming interfaces (APIs) to aid the processing of XML data.

The design goals of XML include, "It shall be easy to write programs which process XML documents." Despite this, the XML specification contains almost no information about how programmers might go about doing such processing. The XML Info set specification provides a vocabulary to refer to the constructs within an XML document, but does not provide any guidance on how to access this information. A variety of APIs for accessing XML have been developed and used, and some have beenstandardized.

Existing APIs for XML processing tend to fall into these categories:

- Stream-oriented APIs accessible from a programming language, for example SAX and Tax.
- Tree-traversal APIs accessible from a programming language, for example DOM.
- XML data binding, which provides an automated translation between an XML documentand programming-language objects.
- Declarative transformation languages such as XSLT and XQuery.
- Syntax extensions to general-purpose programming languages, for example LINQ and Scala.

1.3 INTRODUCTION TO JAVA

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typicallycompiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java wasone of the most popular programming languages in use according to GitHub, particularly for client- server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired byOracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own Hotspot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open source software and used by most developers and is the default JVM for almost all Linux distributions.

CHAPTER 2

SYSTEM ANALYSIS

2.1 LITERATURE SURVEY

If you are not sure what interest rate you are being charged for your car loan, try our **EMI Rate Calculator.**

Disclaimer: The calculation performed by EMI Calculator is based on the information you entered and is for illustrative purposes only. This calculation reflects amounts in Indian Rupee rounded to the nearest whole figure. Estimated monthly payments DO NOT include any processing or other possible fees which may depend on the financial institution / banks.All loan figures are based upon non-commercial usage and are subject to credit approval from an independent lending source. Actual down payment and resulting monthly payments may vary depending upon type and use of vehicle, regional lender requirements, and the strength of your credit. Check with your dealer for exact monthly payment.

2.2 EXISTING SYSTEM

The Equated Monthly Instalment (or EMI) consists of the principal portion of the loan amount and the interest. Therefore, EMI = principal amount + interest paid on the Car Loan. The EMI, usually, remains fixed for the entire tenure of your loan, and it is to be repaid over the tenure of the loan on a monthly basis.

Mathematically, EMI is calculated as under:

P x R x (1+R)^N / [(1+R)^N-1]

P = Principal amount of the loan

R = Rate of interest

N = Number of monthly installments.

Car Loan EMI Calculator, we offer you the provision to identify the exact principal amount, with the interest rate and equated monthly installments payable over a specific tenure before you intend to apply for your Car Loan.

2.3 PROPOSED SYSTEM

Simply key in the amount, rate and tenure for which the car loan is sought, the Car Loan EMI Calculator will automatically reflect the approximate EMI's applicable for the loan tenure Car Loan EMI calculator to know the indicative EMI for the chosen interest rate and tenure tool that provides you with the answer in a split second and enables you to understand how much will be your EMI outgo. All you got to do is use the slider to enter the loan amount, car loan interest rate, and the tenure of your loan

2.3.1 SCOPE OF THE PROJECT

EMI Calculator is an android based application, and enables the user to check their incometax rates and EMI rates. It will assist to assess, how much, we should pay as our income tax or EMI. The app is user friendly, and the user shall find it extremely easy to calculate the income tax that the user has to pay based of their income and government rules and also to calculate the EMI rates on their loan. The application is user friendly, interactive and provides easy to use options for the aforementioned financial computations

2.3.2 AIM OF THE PROJECT

The project has been implemented in modules so as to make it user friendly i.e. if a code viewer is finding trouble with a particular function it can go to a particular module and make corresponding changes in it instead of searching in the program Also it is more advantageous for future modifications as when modifications are to be made instead of searching the whole program module search can be useful as it decreases the time consumption.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 SYSTEM SPECIFICATION

3.1.1 HARDWARE SPECIFICATION

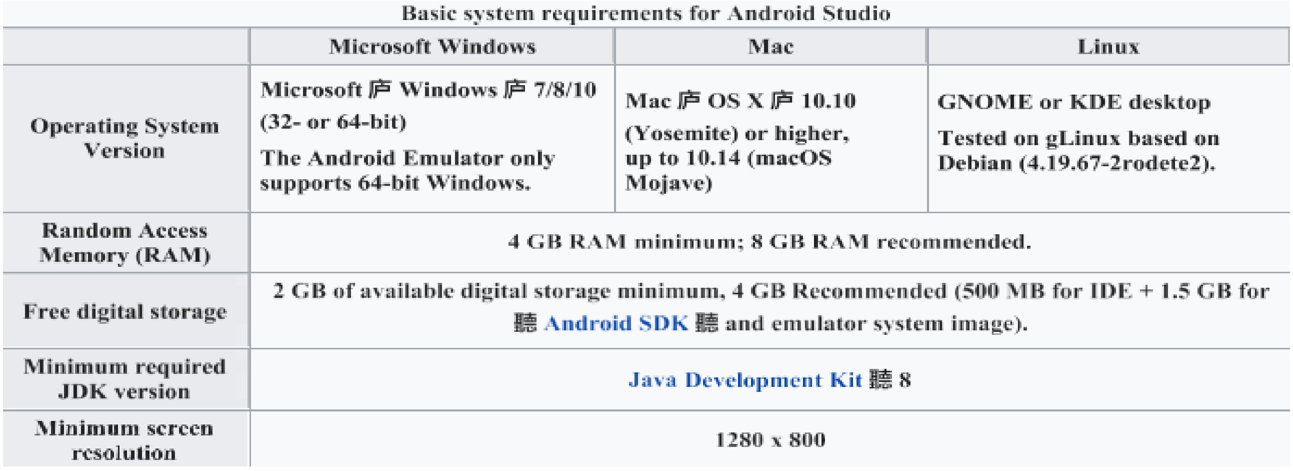
- Processor : Any Processor of Speed 1.60GHz
- RAM : 6.00 GB or More
- Hard Disk : 20 GB or More
- Input Device : Keyboard
- Output Device : Connected Android Mobile / Android Emulator-Pixel 3a API 27

3.1.2 SOFTWARE SPECIFICATION

Operating System	: 64 Bit Operating System, X64-Based
Processor Tools	: Android Studio, JAVA JDK
Programming Language	: Java

3.2 DEVELOPMENT ENVIRONMENT

- ANDROID STUDIO



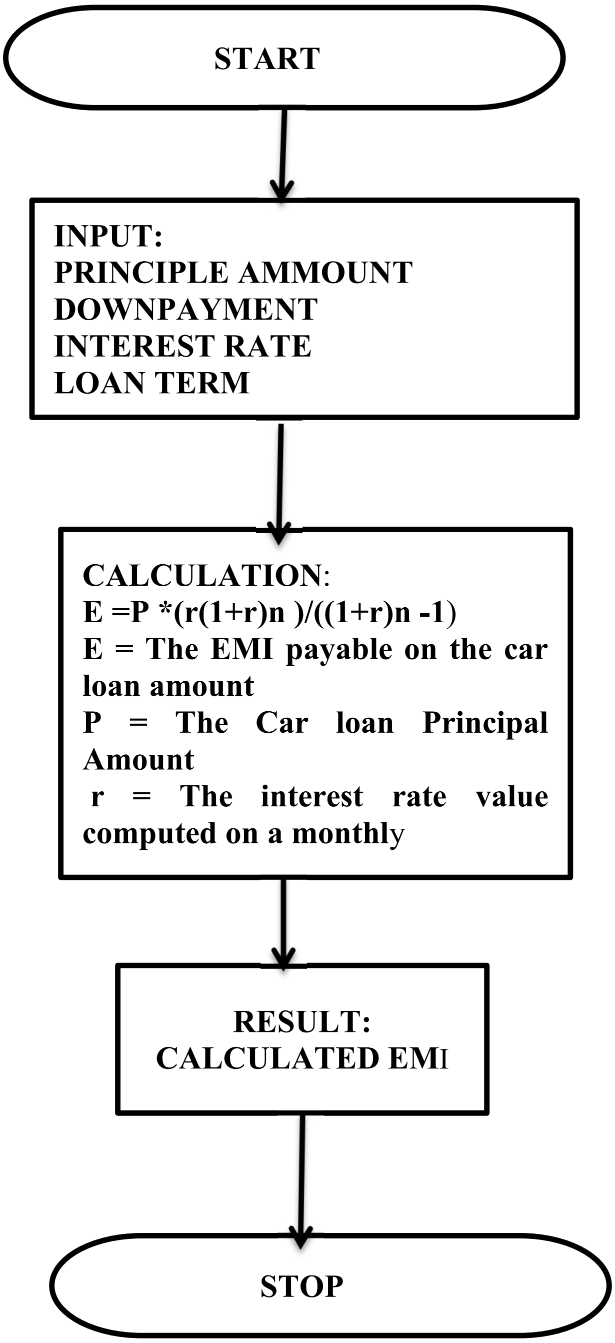
Basic system requirements for Android Studio

	Microsoft Windows	Mac	Linux
Operating System Version	Microsoft 庐 Windows 庐 7/8/10 (32- or 64-bit) The Android Emulator only supports 64-bit Windows.	Mac 庐 OS X 庐 10.10 (Yosemite) or higher, up to 10.14 (macOS Mojave)	GNOME or KDE desktop Tested on gLinux based on Debian (4.19.67-2rodete2).
Random Access Memory (RAM)	4 GB RAM minimum; 8 GB RAM recommended.		
Free digital storage	2 GB of available digital storage minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for 聽 Android SDK 聽 and emulator system image).		
Minimum required JDK version	Java Development Kit 聽 8		
Minimum screen resolution	1280 x 800		

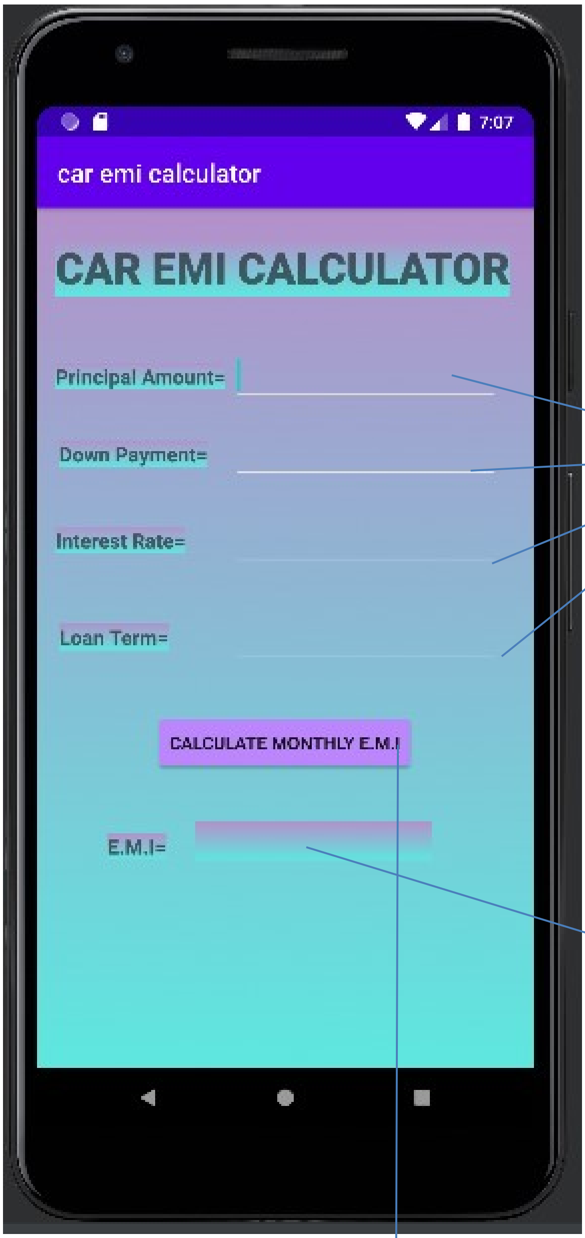
Chapter 4

SYSTEM DESIGN

4.1 Project Flow Chart


START
INPUT:
PRINCIPLE AMMOUNT
DOWNPAYMENT
INTEREST RATE
LOAN TERM
CALCULATION:
E =P *(r(1+r)n)/((1+r)n -1)
E = The EMI payable on the car loan amount
P = The Car loan Principal Amount
r = The interest rate value computed on a monthly
RESULT:
CALCULATED EMI
STOP

4.2 Design using XML



Go to palette -> Common ->
Select Text View -> Drag and Drop

- Change attribute property
- Apply constrain to all the fields

Go to palette -> Text ->
Select Plain Text -> drag and drop

- Change attribute property
- Apply constrain to all the fields

Go to palette -> Button ->
Select Button -> drag and drop

- Change attribute property
- Apply constrain to all the fields

Chapter 5

SYSTEM IMPLEMENTION

ACTIVITYMAIN.JAVA

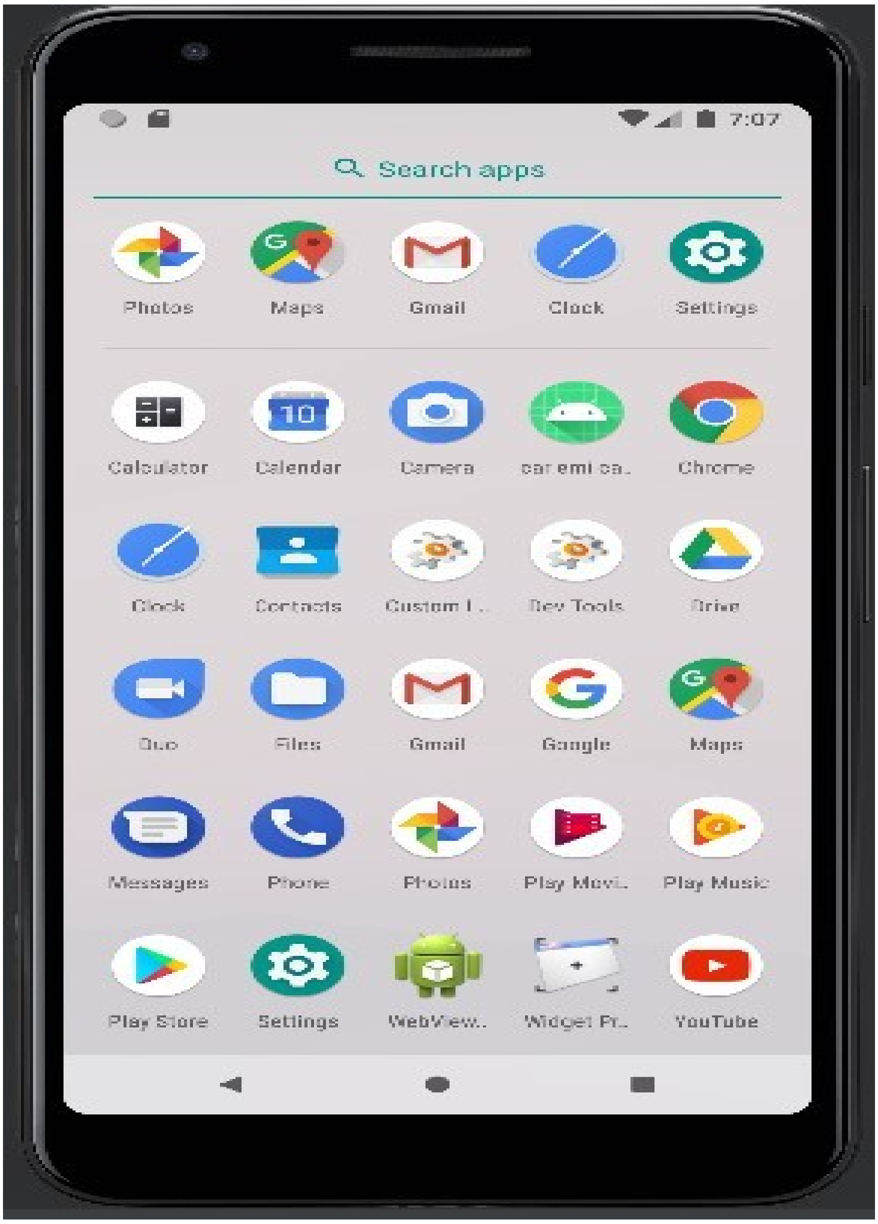
```
package com.example.caremicalculator;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import     android.widget.Button;

import   android.widget.EditText;

import android.widget.TextView;


public     class     MainActivity     extends     AppCompatActivity     implements

   View.OnClickListener{Button b1;

   EditText t1,t2,t3,t4;

   TextView        tt1;

   @Override

   protected  void  onCreate(Bundle  savedInstanceState)

      {                  super.onCreate(savedInstanceState);

      setContentView(R.layout.activity_main);

      b1=(Button)findViewById(R.id.b1);

      b1.setOnClickListener(this);

      t1=(EditText)findViewById(R.id.pa);
```

```
        t2=(EditText)findViewById(R.id.dp);

        t3=(EditText)findViewById(R.id.ir);

        t4=(EditText)findViewById(R.id.lt);

        tt1=(TextView)findViewById(R.id.emi);

    }


    public    void    onClick(View

        v){double p,r,n,d,emi;

        p=Double.parseDouble(t1.getText().toString());

        d=Double.parseDouble(t2.getText().toString());

        p=p-d;

        r=Double.parseDouble(t3.getText().toString());

        r=r/(12*100);

        n=Double.parseDouble(t4.getText().toString());

        emi=p*(r*Math.pow((1+r),n))/(Math.pow((1+r),n)-1);

        emi=emi/12;

        // double emi = principleAmount * (interestRate * Math.pow((1 + interestRate),
loanTerm)) / (Math.pow((1 + interestRate), loanTerm) - 1);

        tt1.setText(String.valueOf(emi));

    }

}
```

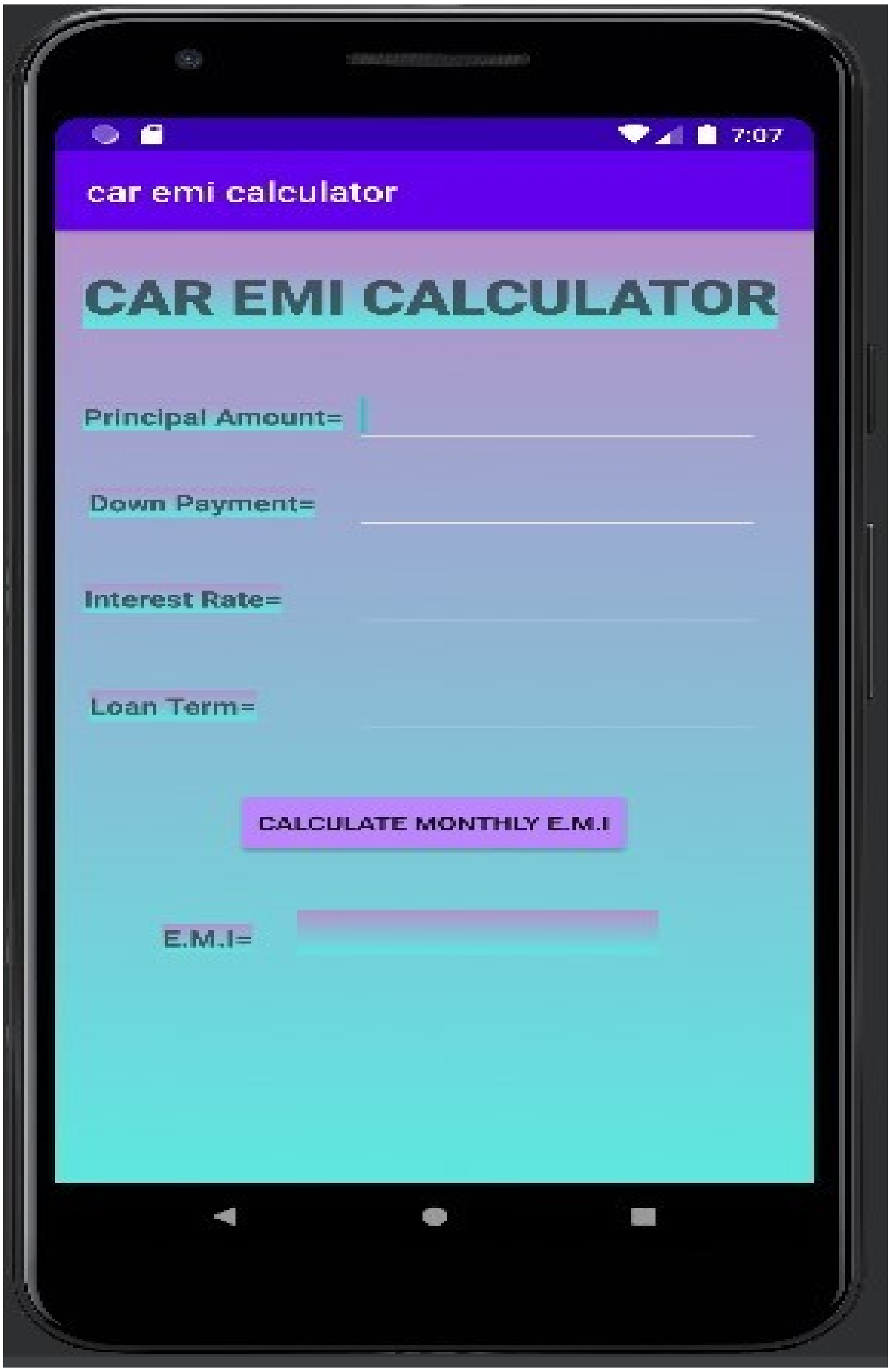
Chapter 6

SNAPSHOTS

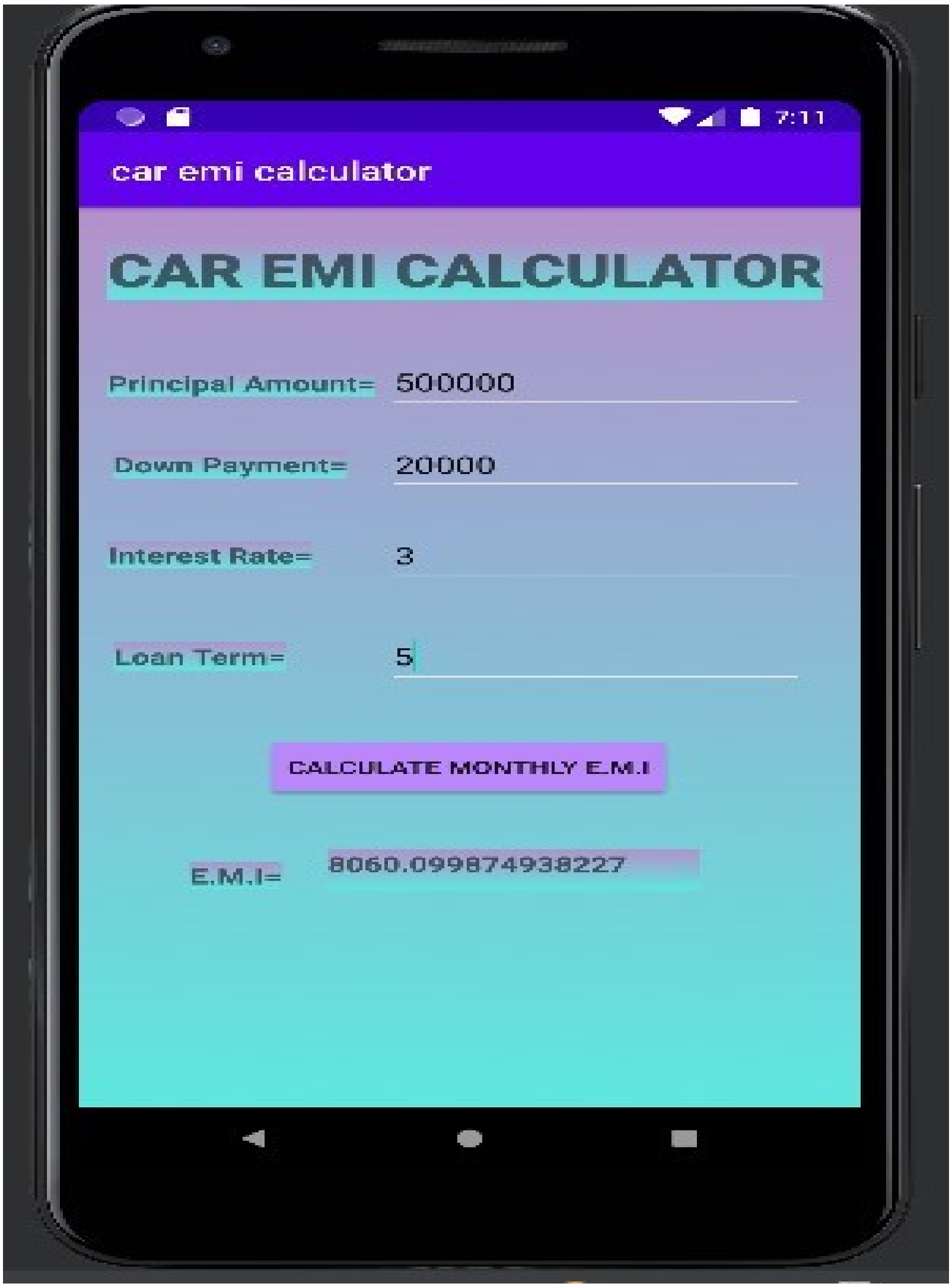
6.1 ANDROID OPENING



6.2 MAIN INTERFACE



6.3 CAR EMI CACLUTION INTERFACE



Principal Amount= 500000

Down Payment= 20000

Interest Rate= 3

Loan Term= 5

E.M.I= 8060.099874938227

CONCLUSION AND FUTURE ENHANCEMENT

This application makes it very easy for the user to calculate the EMI This can be done with the help of an EMI calculator. Calculating your EMI's will help you prepare to make the EMI payments on time.The other benefits of the EMI calculator include:Accessibility: You can use the EMI calculator anytime, anywhere, and make your budget calculations.Financial Planning: EMI calculations must be done even before you start applying for your loan. An EMI calculator can help you with this.

The **"CAR EMI CALCULATOR"** was successfully designed and is tested for accuracy and quality. During this project we have accomplished all the objectives and this project meets the needs of the organization. The developed will be used in searching, retrieving and generating information for the concerned requests

REFERENCES

- Introduction to Android: http://developer.android.com/guide/index.html.
- Android API: http://developer.android.com/reference/packages.html
- Java 6 API: http://docs.oracle.com/javase/6/docs/api/
- Android Fundamentals:
- http://developer.android.com/guide/components/fundamentals.html
- Android User Interfaces: http://developer.android.com/guide/topics/ui/index.html
- Layout: http://developer.android.com/guide/topics/ui/declaring-layout.html