SE4041 - Mobile Application Design And Development **Assignment 3** **Semester 2, 2020**

Group ID: 10 **Project Topic: Life Insurance Calculator**

Student ID	Student Name	Android Project						Report				Viva(15)	Total (100)
		Wireframes(8)	Project Design and Consistency(8)	Integrated application using a repository(3)	Correct calculations for a working app(8)	using SQLite db(20)	Unit Test cases written in Android Project(10)	Explaining connectivity of interfaces(3)	Explaining UI design principles applied to the project (10)	Additional features(5)	mobile test cases(10)		
IT17001908	M.W.S.B Bandara												
IT16521544	B.D.K Samaraweera												

Application project link : - https://drive.google.com/drive/folders/1fWQSWWSmkwQUbk82sBVyGQuCuI07n_MX?usp=sharing

Wireframe design link : - https://app.moqups.com//SJIvrmn9Bt/edit/page/ad64222d5#

Table of Contents

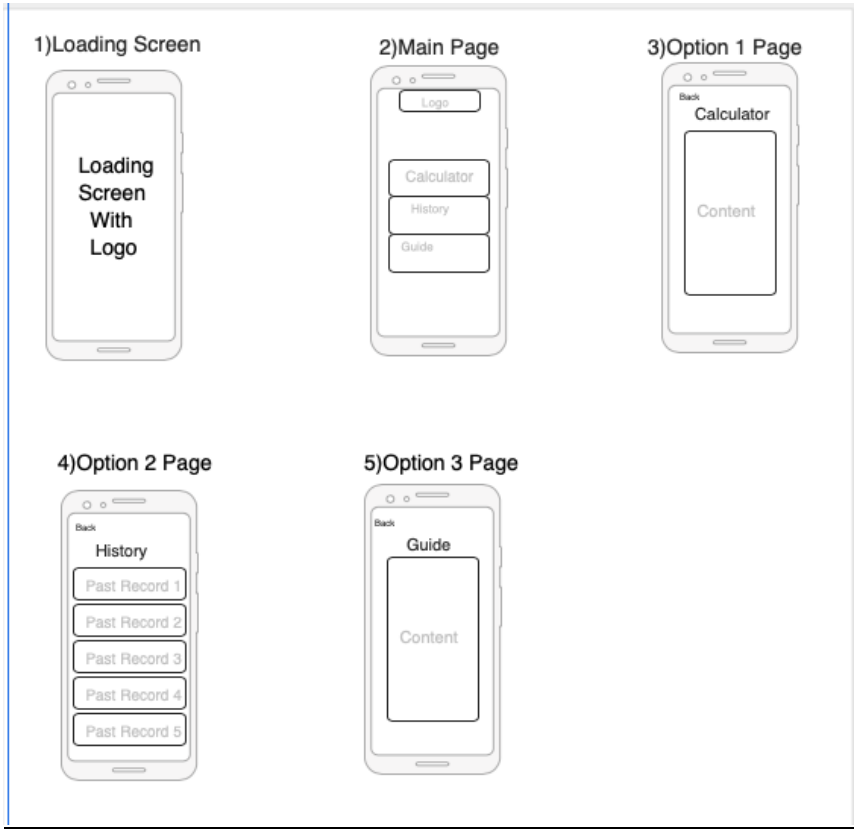
01. Application Introduction, Actual prototype and user interfaces

This Application is named as ‘MY LIC’. The meaning of LIC is ‘Life Insurance Corporation’. We have assigned this name to give a familiar feeling to the user. The specialty of this application is anyone can use without creating an account. So, it will be very convenient for any user who are willing to calculate their life insurance plan.

01.1 Logo

- In this we have decided to use red and orange mixed color combination.
- Because red color is the symbol of energetic and orange color is used to represent confidence, friendliness.
- We have added this logo in png mode. So, it can apply anywhere we want.
- We have added this logo on the middle of the splash screen to emphasize to the user by the first sight.

01.2 Prototype



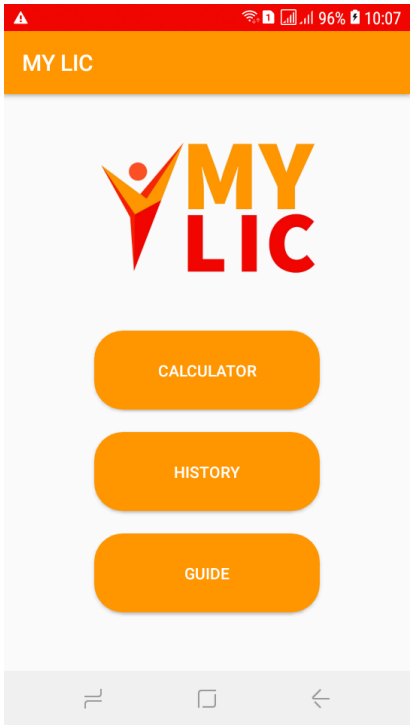
When the app opens, it is starting with a splash screen (Image - 1). Then it will go to the menu (Image - 2). In the menu, it has 3 options. Calculator, History and Guide options.

- Calculator – When we input data to the calculator menu, it gives the monthly installment you have to pay when you are applying for a life insurance and, yearly how much you want to pay for the life insurance.
- History – In this option, it shows past records of you have entered. We can retrieve the data.
- Guide – In this option it included small tips on how you should choose a life insurance, what are the types of life insurances likewise.

01.3 Splash Screen

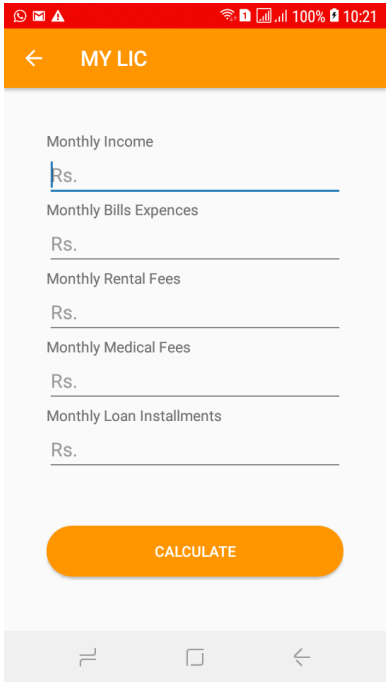
- In this splash screen we have used 3 colors. Red, Orange and Blue. As discussed in the logo section red and orange is used to emphasize energetic and confidence, friendliness.
- This is a life insurance application. So, we have to build a trust between the company and the user. So, Blue color is used to enhance that feeling to the user.
- When a huge incident is happened most of them checks twitter trending. It is because twitter was able to build the trust among people. In the twitter application, they use blue color and white color. It is because of a method of keep engaging the user with the trust.

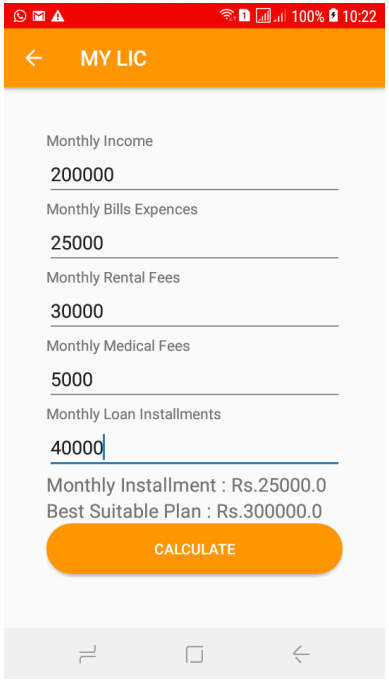
01.4 Main Menu



- We have decided to avoid the complexity. So, we used a white background and only the option buttons and the logo.
- If we added many colors and buttons it gives the complexity feeling to the user.
- In the app we haven't included create account option at the beginning. Because if it implemented, most of the users will give up using the application at the beginning.

01.5 Calculator option



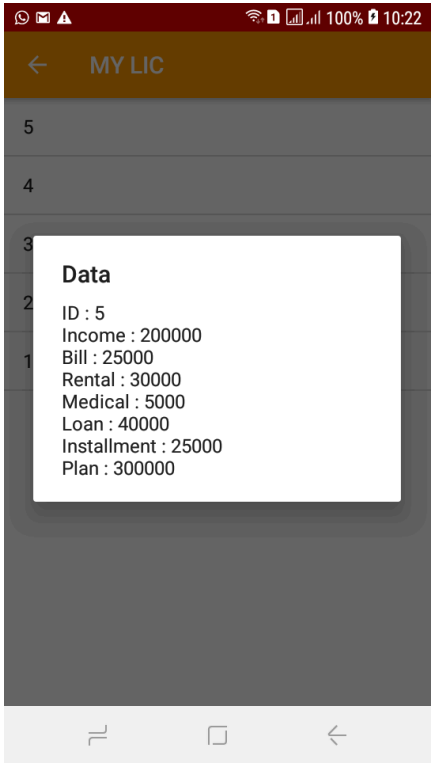


In calculator option basically it is added 5 fields. After adding data to those fields, it will calculate the best plan to the user. With that user will be able to decide, whether this is suitable for me or not. The specialty in this calculator is it will calculate the monthly installment as well. Calculation is as follows.

- Profit = Monthly Income – (Monthly bill expenses + Rental Fees + Medical Fees + Monthly loan installments)
- Monthly installment = Profit X 25%
- Best suitable plan = Monthly Installment X 12

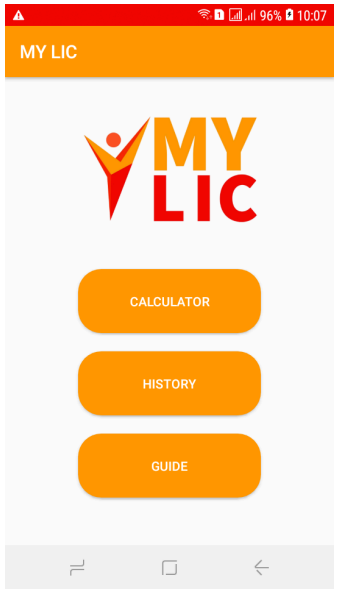
As a group, we have found out information regarding the calculations how real-world application works. They are using 25% as the standard amount from the profit.

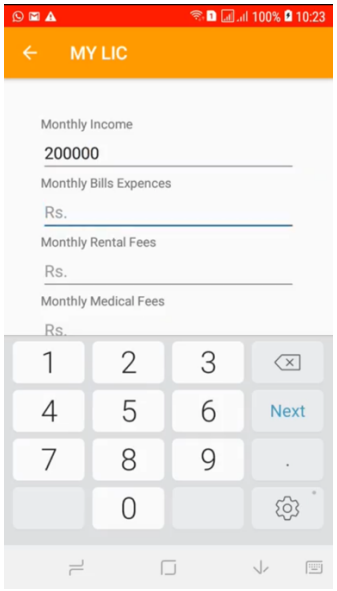
01.6 History Option

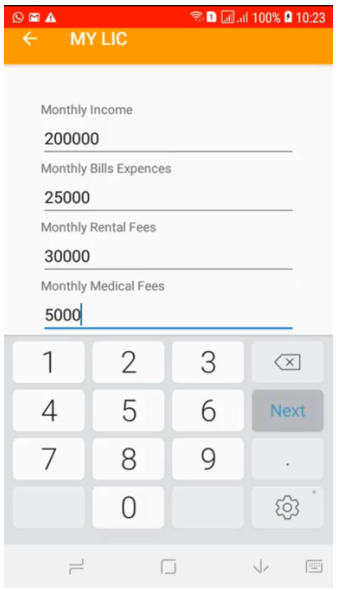


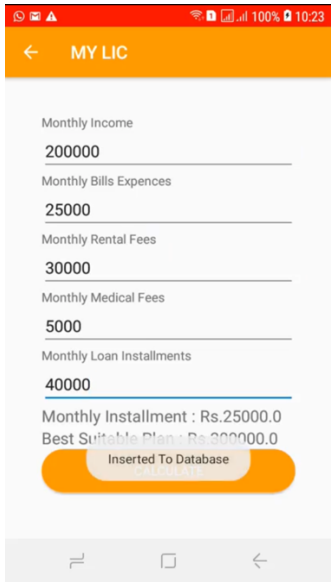
- This is the option where past data is stored. There are more than 5 records are storing. But only 5 is showing in the app.
- In the first image it is shown. After tapping any number, it will pop up the result like in image 2.
- In here also we gave the prioritize to avoid the complexity.
- White color in the background gives the calm feeling to the user.

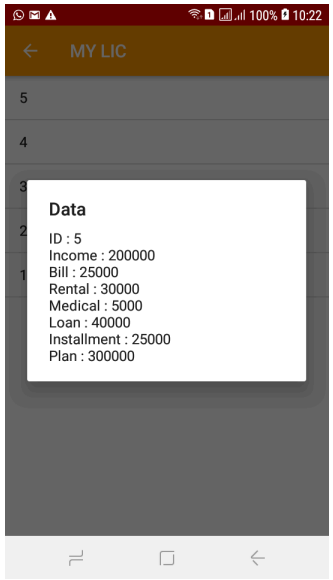
02. Snapshots of running application





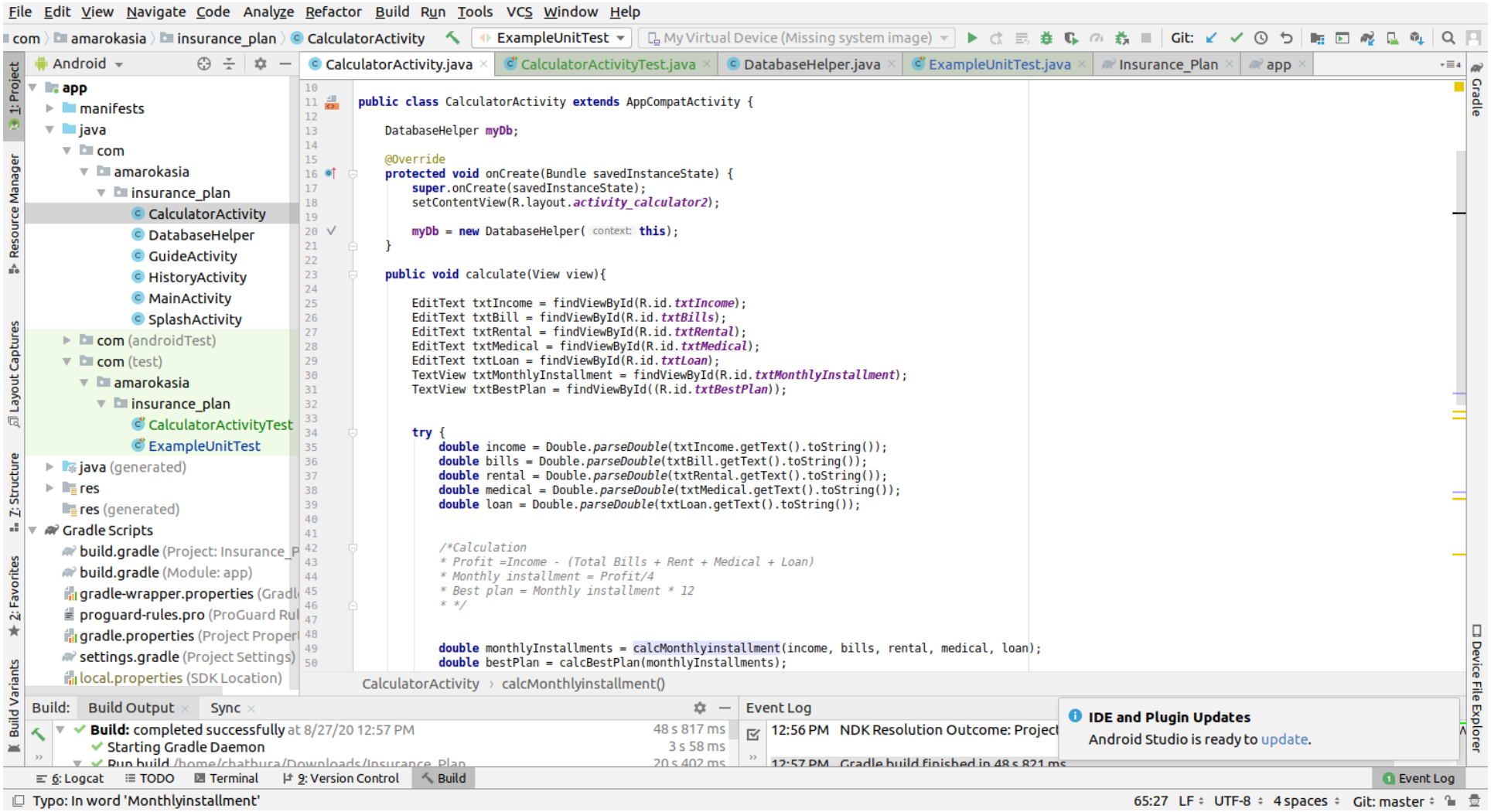


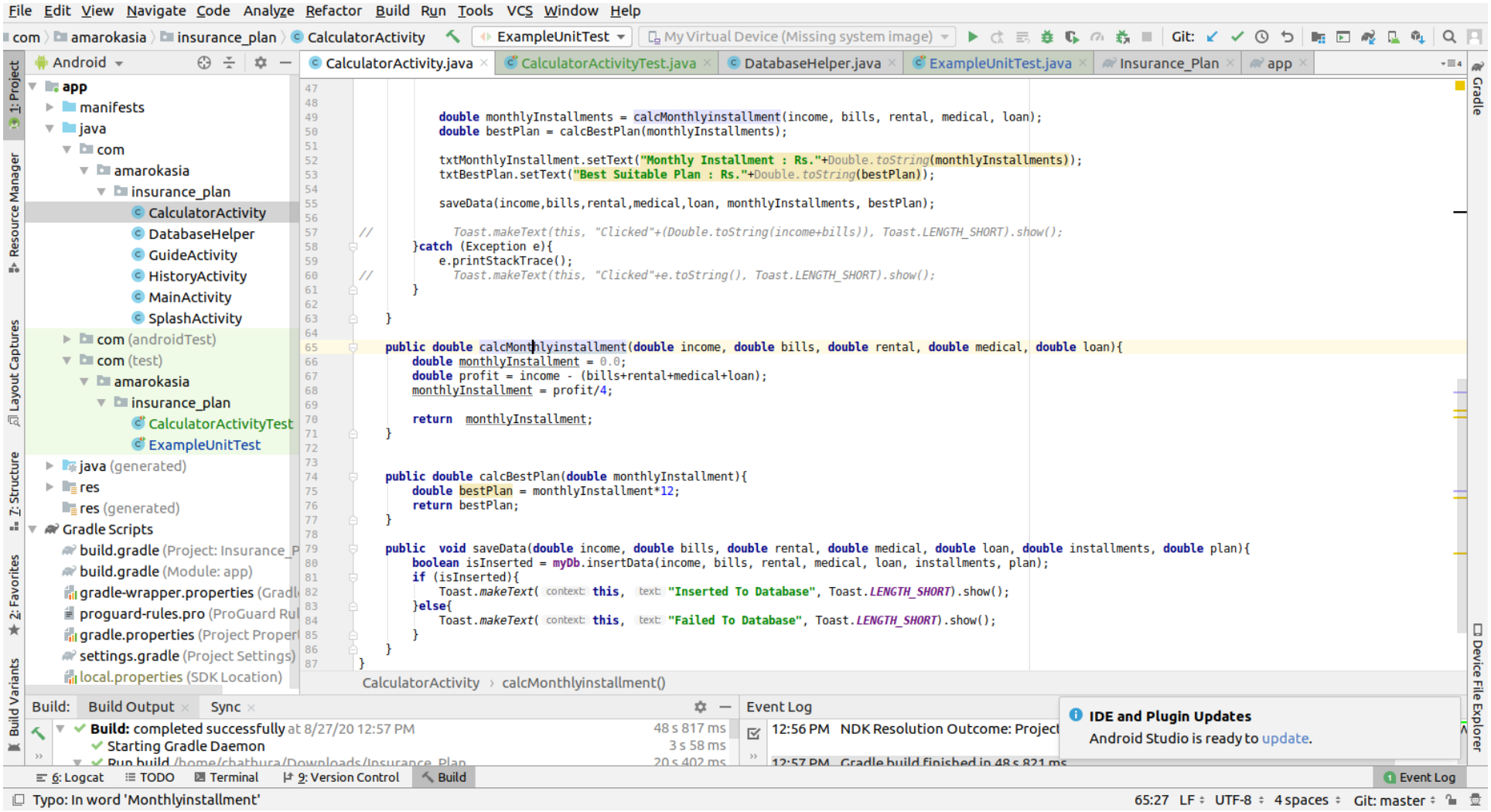




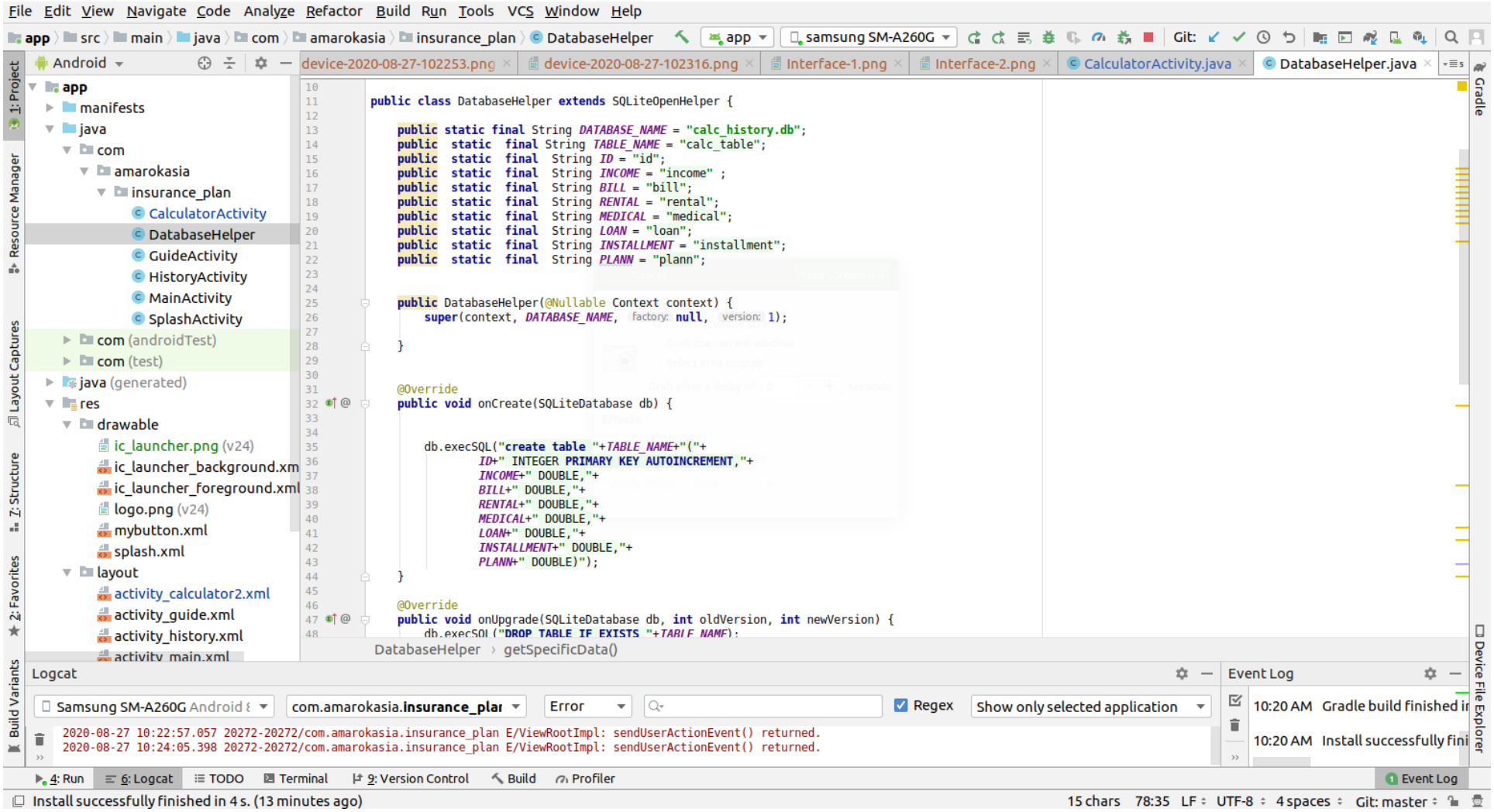
03. Code Screenshots

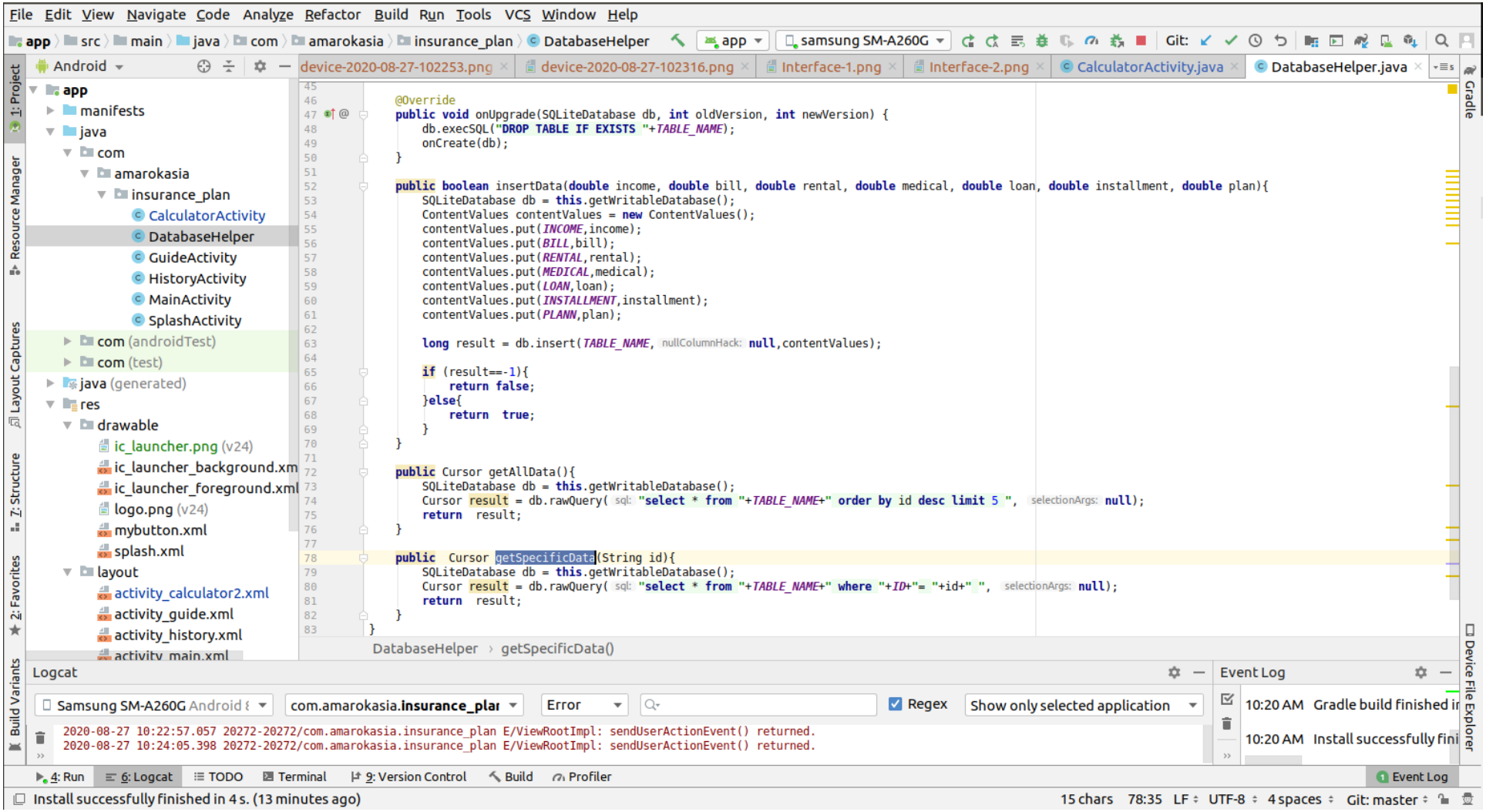
03.1 Calculation




File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
com amarokasia insurance_plan CalculatorActivity
ExampleUnitTest
My Virtual Device (Missing system image)
Git:
Android
CalculatorActivity.java
CalculatorActivityTest.java
DatabaseHelper.java
ExampleUnitTest.java
Insurance_Plan
app
app
manifests
java
com
amarokasia
insurance_plan
CalculatorActivity
DatabaseHelper
GuideActivity
HistoryActivity
MainActivity
SplashActivity
com (androidTest)
com (test)
amarokasia
insurance_plan
CalculatorActivityTest
ExampleUnitTest
java (generated)
res
res (generated)
Gradle Scripts
build.gradle (Project: Insurance_P
build.gradle (Module: app)
gradle-wrapper.properties (Gradl
proguard-rules.pro (ProGuard Rul
gradle.properties (Project Proper
settings.gradle (Project Settings)
local.properties (SDK Location)
double monthlyInstallments = calcMonthlyinstallment(income, bills, rental, medical, loan);
double bestPlan = calcBestPlan(monthlyInstallments);
txtMonthlyInstallment.setText("Monthly Installment : Rs."+Double.toString(monthlyInstallments));
txtBestPlan.setText("Best Suitable Plan : Rs."+Double.toString(bestPlan));
saveData(income,bills,rental,medical,loan, monthlyInstallments, bestPlan);
// Toast.makeText(this, "Clicked"+(Double.toString(income+bills)), Toast.LENGTH_SHORT).show();
}catch (Exception e){
e.printStackTrace();
// Toast.makeText(this, "Clicked"+e.toString(), Toast.LENGTH_SHORT).show();
}
}
public double calcMonthlyinstallment(double income, double bills, double rental, double medical, double loan){
double monthlyInstallment = 0.0;
double profit = income - (bills+rental+medical+loan);
monthlyInstallment = profit/4;
return monthlyInstallment;
}
public double calcBestPlan(double monthlyInstallment){
double bestPlan = monthlyInstallment*12;
return bestPlan;
}
public void saveData(double income, double bills, double rental, double medical, double loan, double installments, double plan){
boolean isInserted = myDb.insertData(income, bills, rental, medical, loan, installments, plan);
if (isInserted){
Toast.makeText(context: this, text: "Inserted To Database", Toast.LENGTH_SHORT).show();
}else{
Toast.makeText(context: this, text: "Failed To Database", Toast.LENGTH_SHORT).show();
}
}
}
CalculatorActivity › calcMonthlyinstallment()
Build:
Build Output
Sync
Build: completed successfully at 8/27/20 12:57 PM
48 s 817 ms
Starting Gradle Daemon
3 s 58 ms
Event Log
12:56 PM NDK Resolution Outcome: Project
IDE and Plugin Updates
Android Studio is ready to update.
6: Logcat
TODO
Terminal
9: Version Control
Build
Event Log
Typo: In word 'Monthlyinstallment'
65:27 LF UTF-8 4 spaces Git: master

03.2 Database




File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app src main java com amarokasia insurance_plan DatabaseHelper
app
samsung SM-A260G
Git:
Android
app
manifests
java
com
amarokasia
insurance_plan
CalculatorActivity
DatabaseHelper
GuideActivity
HistoryActivity
MainActivity
SplashActivity
com (androidTest)
com (test)
java (generated)
res
drawable
ic_launcher.png (v24)
ic_launcher_background.xml
ic_launcher_foreground.xml
logo.png (v24)
mybutton.xml
splash.xml
layout
activity_calculator2.xml
activity_guide.xml
activity_history.xml
activity_main.xml
device-2020-08-27-102253.png
device-2020-08-27-102316.png
Interface-1.png
Interface-2.png
CalculatorActivity.java
DatabaseHelper.java
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
db.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME);
onCreate(db);
}
public boolean insertData(double income, double bill, double rental, double medical, double loan, double installment, double plan){
SQLiteDatabase db = this.getWritableDatabase();
ContentValues contentValues = new ContentValues();
contentValues.put(INCOME,income);
contentValues.put(BILL,bill);
contentValues.put(RENTAL,rental);
contentValues.put(MEDICAL,medical);
contentValues.put(LOAN,loan);
contentValues.put(INSTALLMENT,installment);
contentValues.put(PLANN,plan);
long result = db.insert(TABLE_NAME, nullColumnHack: null,contentValues);
if (result==-1){
return false;
}else{
return true;
}
}
public Cursor getAllData(){
SQLiteDatabase db = this.getWritableDatabase();
Cursor result = db.rawQuery(sql: "select * from "+TABLE_NAME+" order by id desc limit 5 ", selectionArgs: null);
return result;
}
public Cursor getSpecificData(String id){
SQLiteDatabase db = this.getWritableDatabase();
Cursor result = db.rawQuery(sql: "select * from "+TABLE_NAME+" where "+ID+"= "+id+" ", selectionArgs: null);
return result;
}
}
DatabaseHelper › getSpecificData()
Logcat
Samsung SM-A260G Android
com.amarokasia.insurance_plan
Error
Regex
Show only selected application
2020-08-27 10:22:57.057 20272-20272/com.amarokasia.insurance_plan E/ViewRootImpl: sendUserActionEvent() returned.
2020-08-27 10:24:05.398 20272-20272/com.amarokasia.insurance_plan E/ViewRootImpl: sendUserActionEvent() returned.
Event Log
10:20 AM Gradle build finished in
10:20 AM Install successfully fini
4: Run
6: Logcat
TODO
Terminal
9: Version Control
Build
Profiler
Event Log
Install successfully finished in 4 s. (13 minutes ago)
15 chars 78:35 LF UTF-8 4 spaces Git: master

04. Additional Feature

- Extra function is this Guide function.
- This will give some information to the user. If a user has no experience regarding the Life insurances, this will be very helpful.
- To emphasize the content, we decided to use black color font.

05. Mobile test cases with results

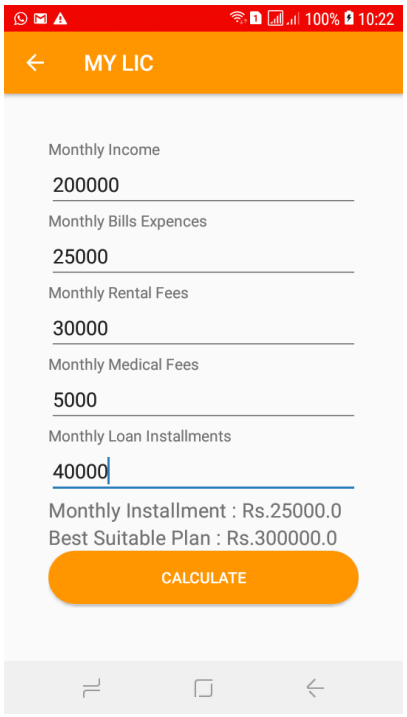


Image 1 – Adding Data

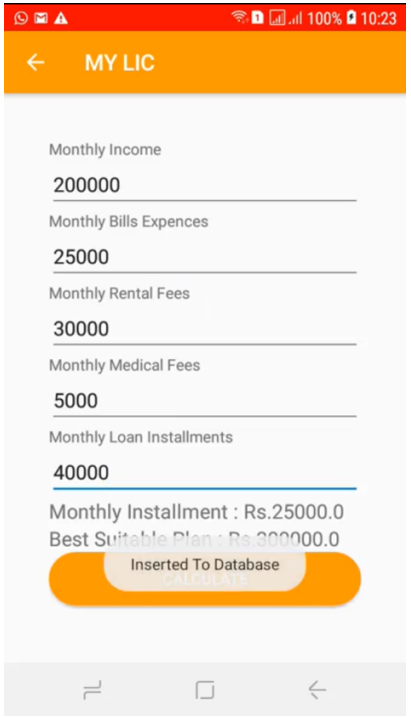


Image 2 – Store calculation

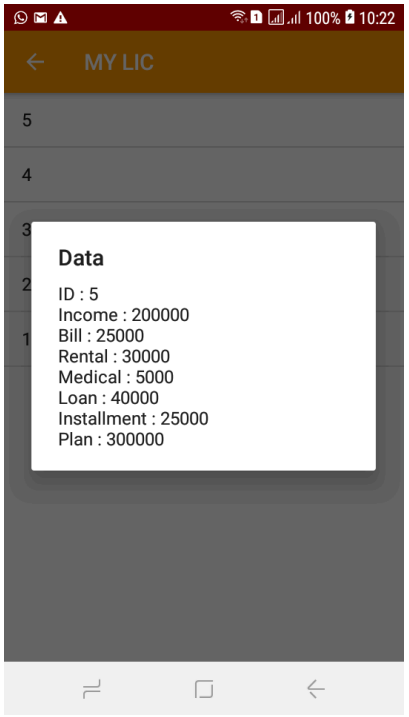
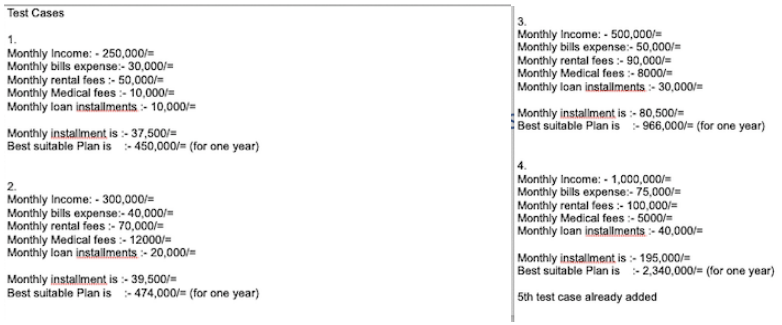


Image 3 – Result

Following data sets are currently stored in the database.



Test Cases

1.
Monthly Income: - 250,000/=
Monthly bills expense:- 30,000/=
Monthly rental fees :- 50,000/=
Monthly Medical fees :- 10,000/=
Monthly loan installments :- 10,000/=

Monthly installment is :- 37,500/=
Best suitable Plan is :- 450,000/= (for one year)

2.
Monthly Income: - 300,000/=
Monthly bills expense:- 40,000/=
Monthly rental fees :- 70,000/=
Monthly Medical fees :- 12000/=
Monthly loan installments :- 20,000/=

Monthly installment is :- 39,500/=
Best suitable Plan is :- 474,000/= (for one year)

3.
Monthly Income: - 500,000/=
Monthly bills expense:- 50,000/=
Monthly rental fees :- 90,000/=
Monthly Medical fees :- 8000/=
Monthly loan installments :- 30,000/=

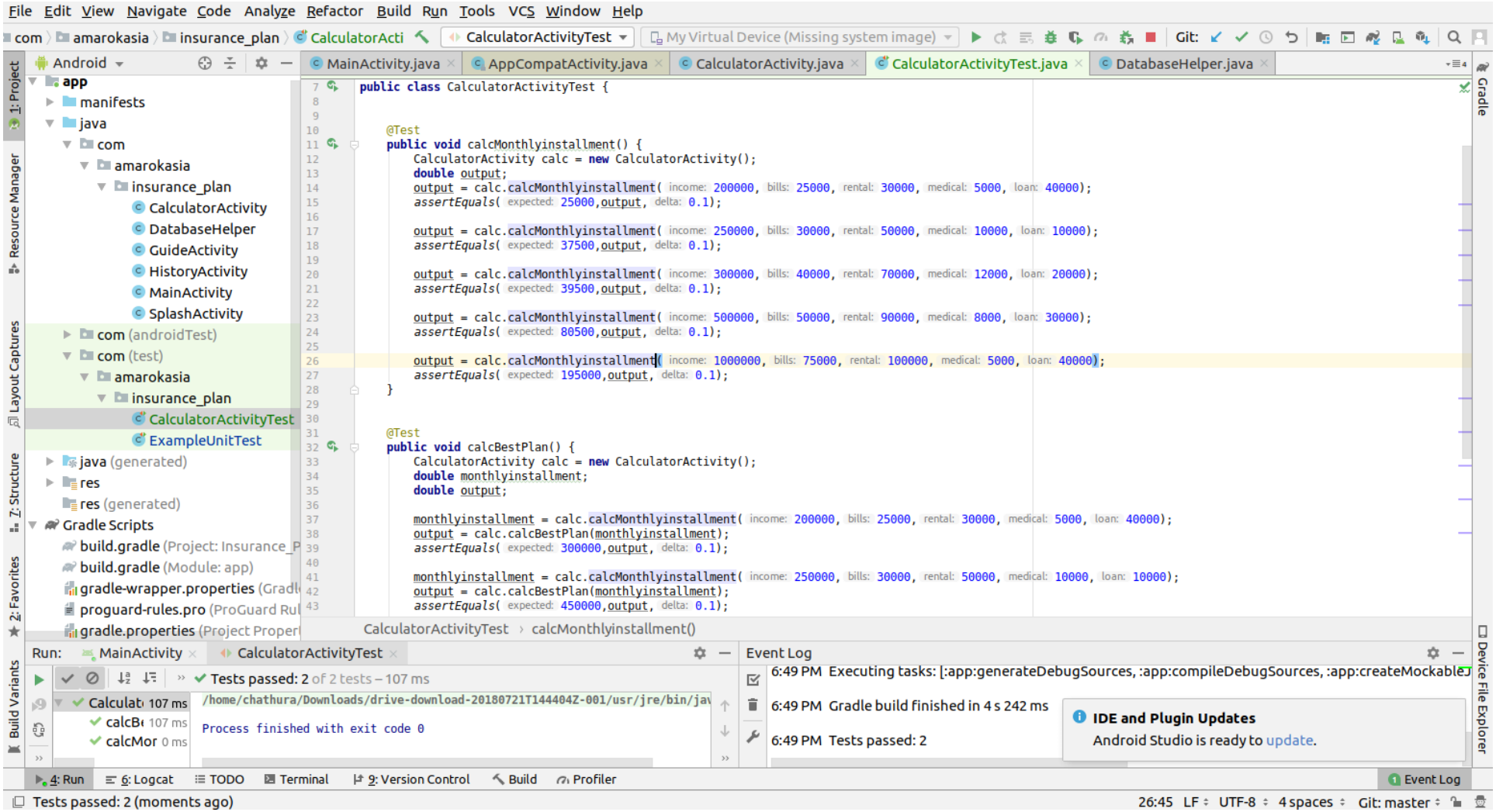
Monthly installment is :- 80,500/=
Best suitable Plan is :- 966,000/= (for one year)

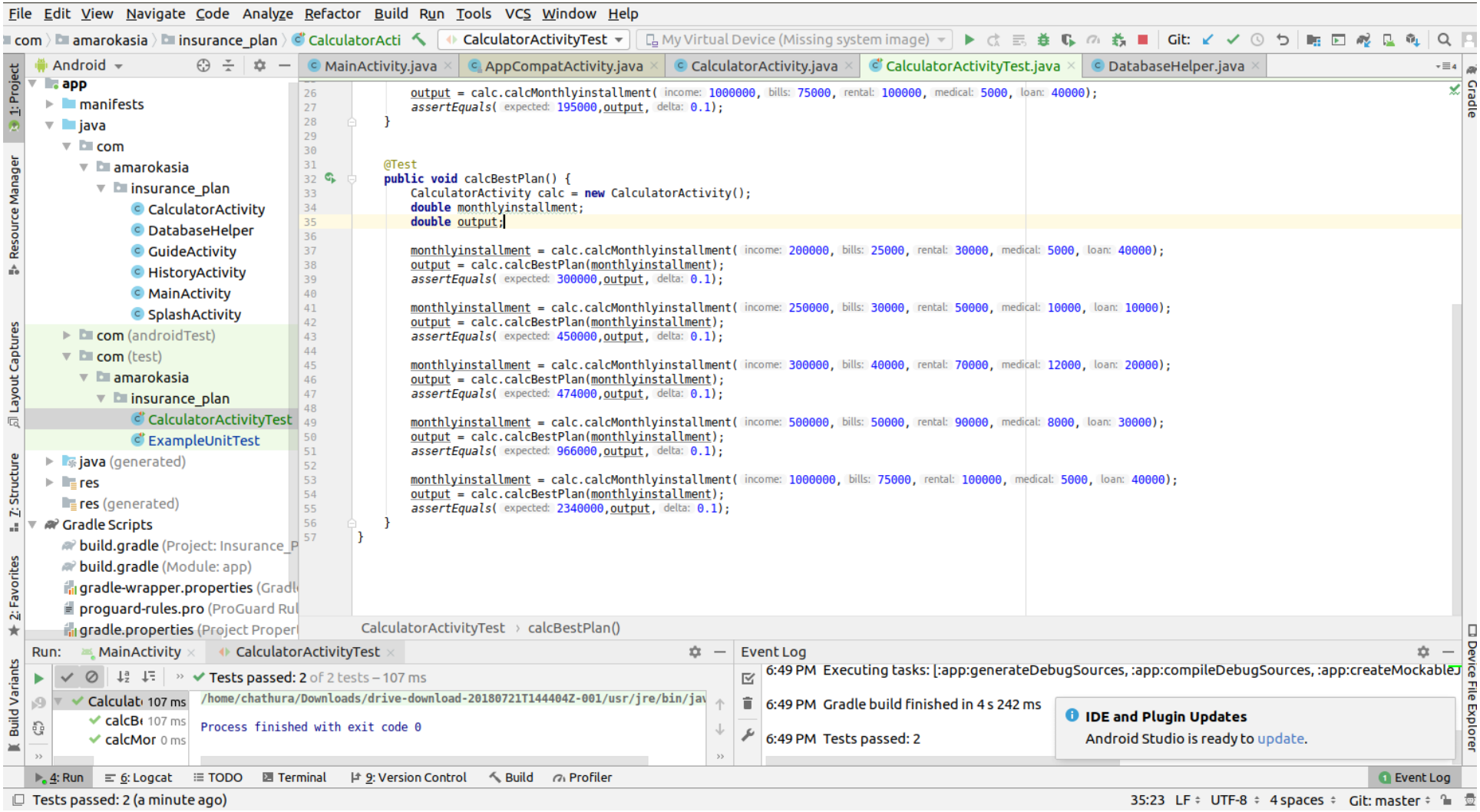
4.
Monthly Income: - 1,000,000/=
Monthly bills expense:- 75,000/=
Monthly rental fees :- 100,000/=
Monthly Medical fees :- 5000/=
Monthly loan installments :- 40,000/=

Monthly installment is :- 195,000/=
Best suitable Plan is :- 2,340,000/= (for one year)

5th test case already added

06. Unit test cases with results





-End of the document-