A REPORT

ON

ANDROID APPLICATION THAT DISPLAYS SERVICES' DETAILS

BY

Name(s) of the Student(s)	**ID. No.(s)**
Lakhshya Bansal	2012A7PS080G
Parasa Srinivas	2012A7PS173H

AT

INDIAN BANK, CHENNAI
CORPORATE OFFICE, ROYAPETTAH

A Practice School – I station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
JULY, 2014

A REPORT

ON

ANDROID APPLICATION THAT DISPLAYS SERVICES' DETAILS

BY

Name(s) of the Student(s)	**ID. No.(s)**	**Discipline**
Lakhshya Bansal	2012A7PS080G	B.E.(Hons.) Computer Science
Parasa Srinivas	2012A7PS173H	B.E.(Hons.) Computer Science

Prepared in partial fulfillment of the
Practice School – I, Course No.
BITS C221/BITS C231/BITS C241/BITS GC221/BITS GC231

AT

INDIAN BANK, CHENNAI
CORPORATE OFFICE, ROYAPETTAH

A Practice School – I station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
JULY, 2014

Acknowledgements

No piece of work is possible without the generous help of a number of people. We'd like to take this opportunity to thank these persons for their time and effort. A sincere and heart felt thank you to BITS Pilani and the Practice School Division of BITS Pilani for giving us the opportunity to work as interns at Indian Bank. Our gratitude to Indian Bank for training us and giving us a good project to work on. Special thanks to Prof. Sundarasan Raman, our faculty in-charge at Indian Bank, for assisting us throughout our Practice School, Mr. Gopal Sukumaran, the Assistant General Manager of Marketing Department, Corporate Office, for giving us a very good project that we get to learn from, Mr. S. M. Senthil Kumar, the Senior Manager of Marketing Department, Corporate Office, for being kind and considerate and helping out in every stage of the development, and Mr. Rohit Arvind, Manager of Marketing Department, Corporate Office, for helping us and guiding us throughout the course.

Though we have mentioned a few names above, the list of friends and mentors is never complete. A heartfelt thank you to everyone who has helped us directly or indirectly in the preparation of this report and thereby the project.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN)

Practice School Division

Station : Indian Bank **Center :** Corporate Office, Royapettah

Duration : 8 weeks **Date of Start :** May 23rd, 2014

Date of Submission : July 17th, 2014

Title of the Project : Android Application that displays Services' Details

ID No./Name(s)/ Discipline(s) of the student(s):	2012A7PS080G Lakhshya Bansal B.E.(Hons.) Computer Science	2012A7PS173H Parasa Srinivas B.E.(Hons.) Computer Science

**Name(s) and
designation(s)
of the expert(s):**
	Mr. Senthil Kumar
	Senior Manager
	Marketing Division
		Mr. Rohit Arvind
		Manager
		Marketing Division

Name(s) of the PS Faculty:	Prof. Sundarasan Raman	

Keywords: android, marketing, python, mobile, application, indian bank, bank

Project Area(s): Android and Python

Abstract: This is a report for the project done at Indian Bank for Practice School-I of BITS Pilani. The project involved developing an Android based mobile application that displays the details of the various products and services offered by Indian Bank. The application is to be used by Marketing officers when they are on the field promoting the bank's products and services.

Signature(s) of Student(s)
Date:

Signature of PS Faculty
Date:

Table of Contents

Introduction

Students of the BITS Pilani University after successful completion of the second year are sent for internships called Practice School by the institute itself. By virtue of the above policy, both of us, Parasa Srinivas and Lakhshya Bansal students of BITS Pilani, Hyderabad Campus and BITS Pilani, K. K. Birla Goa Campus respectively have been given the opportunity to work at Indian Bank, Corporate Office, Royapettah, Chennai – 600 014, for a period spanning from May 23rd, 2014 till July 17th, 2014. As a part of the internship students were introduced to the various terminologies and technologies used in a bank in India, more specifically about the Marketing Division to which they have been allotted. The students were also given a project which involves building an Android based mobile application for the employees of the Marketing Division.

This is the final project report for the Practice School I. It has been written alongside the completion of the course with a great deal having been learnt and a great lot more to learn. On May 31st, 2014, the interns of Indian Bank had an orientation session that gave them an insight to the working of the bank. A lot was understood about the functioning of various departments. Students had a look at the different problems that such a large organization has to tackle. Also learnt were concepts of basic banking and general office behavior.

This report contains details of the project that had been allotted and the final output that has been developed after 8 weeks of effort and guidance. The report briefs about the application developed and the process of development. It also has specifications of the various technologies and tools used in the process.

1. About the Project

Despite the initial thoughts of the unavailability of computer science related projects, the authorities of the Marketing department were kind enough to reconsider and create a requirement for computer science interns. After a session of collective brain storming, 2 different projects were proposed and freedom was given to choose either project. The first project involved the creation of a software that would help connect the marketing department and the call center facility of Indian Bank so that the leads generated from the call center could be automatically segregated into different departments thus reducing the work of the Marketing division. This project would require frequent visits to the call center as a lot of work was to be done from there itself.

The second project given by the Marketing division of the bank was to develop an smartphone based static mobile application that shall display the details of the various products and solutions that the bank offers, in offline mode. The application post development would be sent to higher authorities and on approval be released as an official application of the bank that marketing employees shall use for reference when they are out on the field promoting the company's products.

Being short term interns, there were many privacy restriction imposed in accordance to which students were not to be allowed any network access. Students were also not allowed any access to internal files and algorithms. Considering the privacy limitations of the bank and the greater utility of the mobile application, the second project was taken up after consent with the instructor, Prof. Sundarasan Raman and the Senior Manager, Mr. Senthil Kumar.

2. Approach Towards the Project

The aim of the project was to develop a mobile application containing static, offline details of the various products and services that the bank offers. The application is intended to be used by marketing officials for reference when they are out in the field promoting the bank's services. The content that the application shall display needs to be easily updatable and amendable.

The mobile application has been developed for Android as it is the most popular operating system used in smartphones today. Windows and iOS are the other operating systems presently being used but Android being free open source software (FOSS) and extremely easy to use was the more practical choice for development. The percentage of marketing officials having Android smartphones surpassed the percentage of all other operating systems by a good number, further supporting the choice of Android as the development platform.

Since the content being displayed on the application needs to be easily amendable by the employees, hence along with the mobile application, the project involves creation of a compiling software that allows even non-Android expert employees to update the application in case there are any changes made to the software of the content. This software shall run on a PC and generate a APK file that is used to install the application on Android devices. The APK file shall be generated with the help of Android Studio.

Having decided on the broad necessities of the project, the next step involved deciding on the tools and softwares required for the development of both the softwares. Android applications can be developed using Python or using the Android SDK (software development kit). Using Python supports only development of basic Android applications and deploying such applications is not straightforward. Installation of Python developed Android applications requires the presence of Python installed on the Android phones which is not a common pre-installed package. Android SDK on the other hand allows programming more complicated and stable Android programs which are easy to deploy and install. For the PC based Android application updating software, Python was chosen. Python is environment independent i.e. the same code can run on different hardware and software environments. Also the GUI development library of Python is easy to use and also flexible. Being a scripting language, changes to the code can also be made easily. Considering all the above, Android SDK was chosen for the Android application and Python was chosen for the PC based updating software.

3. Requirements for the Project

Based on the selected development platforms i.e. the Android SDK for the Android application and Python for the PC based updating software, a few development tools were required in form of both hardware and software. The requirements are as follows:

Tools required for the development of the Android application:

- Android SDK (Software Development Kit)
- Android Studio IDE (Integrated Development Environment)
- Eclipse IDE (Integrated Development Environment)
- JDK (Java Development Kit)
- JRE (Java Runtime Environment)
- Content from https://www.indianbank.in
- A few Android smartphones for testing

Tools required for the development of the PC based updating software:

- Python package
- IDLE IDE (Integrated Development Environment)
- py2exe/cx_Freeze
- Portable version of the Android SDK (Software Development Kit)
- Gradle package
- Zip package

Latest versions of all softwares were downloaded. Android smartphones that were available with certain employees and interns. These were used for testing purposes.

4. About Android

Android is a mobile operating system (OS) based on the Linux kernel that is currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrists (Android Wear). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touchscreen input, it also has been used in game consoles, digital cameras, and other electronics.

As of 2011, Android has the largest installed base of any mobile OS and as of 2013, its devices also sell more than Windows, iOS, and Mac OS devices combined. As of July 2013 the Google Play store has had over 1 million Android apps published, and over 50 billion apps downloaded. A developer survey conducted in April–May 2013 found that 71% of mobile developers develop for Android. At Google I/O 2014, the company revealed that there were over 1 billion active monthly Android users (that have been active for 30 days), up from 538 million in June 2013.

Android's source code is released by Google under open source licenses, although most Android devices ultimately ship with a combination of open source and proprietary software. Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance— a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices.

Android is popular with technology companies which require a ready-made, low-cost and customizable operating system for high-tech devices.[23] Android's open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which add new features for advanced users[24] or bring Android to devices which were officially released running other operating systems. The operating system's success has made it a target for patent litigation as part of the so-called "smartphone wars" between technology companies.

4.1. Android Development Basics

Android application mainly consists of components, manifest and resources. Each and every of Android fundamental aspect is being explain here clearly.

There are 4 main components that will help us to build the application. They are

1. Activities
2. Services
3. Content Providers
4. Broad cast receivers

Activities:

An Activity represents a single screen with user interface. When the user sees the screen, what s/he sees is an activity. Activities are used in all sorts of applications, for example an email application. The list of emails that you see on the screen is an activity. Whenever you click on the particular email the content that gets displayed is another activity. Hence an activity is a primary component of an application. There can be no user interface for an application without an activity. In general an application contains multiple activities.

Services:

Services perform long running operations in the background. These do not contain user interface. Services are initiated by a particular application and it goes on running even if the application that created is closed. An example for the use of services is a music playing application. Even if music application had been closed the song goes on playing. Other example would be GPS. Whenever you click on your navigation application it asks you permission to turn on your GPS. When you turn on it, if you leave the application after your work GPS will still be on.

Content providers:

Content providers are mainly used to store and retrieve data and make it accessible to all applications. Content providers are the only way to provide data to share across applications. Data is exposed as simple tables as a data base as seen in data base model. Android systems contain many content providers within it like media, contacts.

Broadcast receivers:

This is the component that responds to system wide broadcast announcements. Broadcast receivers can be explained using the example that whenever the battery is low, your Android operating system will send out a broadcast which any application listening to that can respond to. Every application can create its own broadcast receiver that different applications can respond to. Just like the services broad cast receivers does not contain user interface but whenever a broadcast receiver is created then a new notification can be shown on screen that will alert the user.

Manifest file:

Any Android application project contains a manifest file in its root directory that is named same across any project with name "AndroidManifest.XML". The main purpose of this file is to provide information about the application to the Android system.

The manifest file describes the components that we have used in the application. It also contains the permissions that are required to access to different features of the Android system. Whenever an Android application is downloaded from the Android market before installation starts users are asked to provide certain permissions to the application which are specified in the manifest. Manifest also declares the minimum Android SDK version required to run the application. Importance of this is that the application will only be visible to devices whose SDK version is same or below that provided min SDK.

Resources:

Resources are generally the images, XML, audio files that are used in the development of an Android program. Resources of a particular application are contained in the resource directory named res in the Android project. Images will be contained in drawable sub directories of res directory.

4.2. User Interface Basics:

The UI (User Interface) of an Android application is designed using a combination of XML files and Java code.

Layout:

As already explained activity is the only component that contains a user interface. The design for the user interface of an activity can be done in the layout XML file or in the Java code. Layout XML file basically contains the components of the user interface and the way the user interface components are arranged. Declaring user components can also be done with Java code during runtime but it is not recommended because of two reasons:

1. The position in which the component will be attached cannot be known which will create a mess while displaying.
2. If the user interface is created in the XML interface and action on the components is written in the Java code then this helps tremendously while debugging the code.

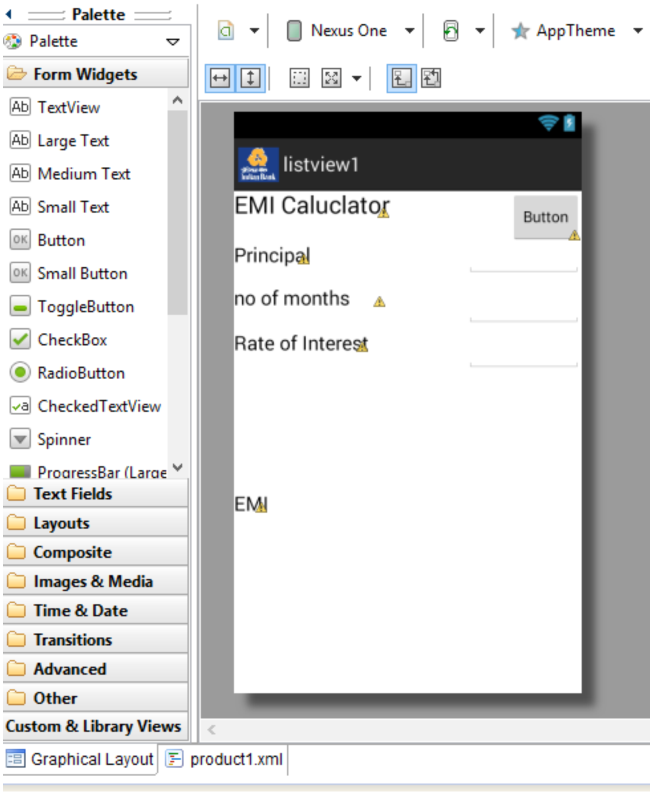


Fig. 4.1. XML designing interface

Figure 4.1 shows the drag and drop tool interface used to design a XML design file. Depending on the way one can arrange the components in the layout, layouts are broadly classified into many types some of which are relative layout, linear layout, table layout and list layout.

Java Code:

It’s already known that user interface is primarily constructed with XML layout graphical interface and Java code is used to deal with the interaction between the Android components whenever user performs an action on any component of the layout.

The Java code is written in the src directory of the Android project. The src sub directory is created by default and it contains a class which extends an activity class or activity subclass.

5. The Android Marketing Application

The Android Marketing Application is the base of the project. The application is meant to display details and specifications of the different products and services that are offered by Indian Bank. The application shall be deployed on the Android smartphones of Marketing employees which they can refer to for accurate and updated data on the services that they are promoting.

For the easy of use and understanding, 2 different interfaces were developed for the application, both of which are available as different applications for installation. Either or both of the applications can be installed on the phones of the Marketing officials based on which interface they are more comfortable with. The core content of both the interfaces is the same and changes or updates need to be made at only one location for it to reflect in both the interfaces. In addition to the details of the various products and services, the application also contains 2 calculators for the benefit of the Marketing officers. An explanation of both the interfaces and calculators developed follows.

5.1. Split Screen Interface

The Split Screen Interface essentially has a ListView that displays the list of the different products that are available in the bank. The ListView as the name suggests is a list of products displayed on the user screen when a particular activity is launched. The primary use of the ListView is that the user can select a particular product from the ListView then it will show the information corresponding to the product.

The interaction between the name of the product and the information to display when clicking on that particular product name can be done in two ways. One way is that whenever the user clicks on particular product, the information can be displayed in other activity and the other way is to display information of the product also within the same activity of product names with the use of fragments. Hence in large screen devices, the list displaying the products stays on the screen allocating a part of the screen to display the details of the product selected. Hence the name Split Screen Interface. Figure 5.1 illustrated the Split Screen in action. In smaller devices, the ListView is completely replaced by a new screen that displays the details of the product selected.

The list of the products to display and their details are stored in the form of an ArrayList. An ArrayList is an inbuilt class offered by Java that allows programers to store a collection of data in a serially organized manner. To link this ArrayList to the display ListView an Adapter is created. An Adapter object is created using the inbuilt array adapter class which takes constructor parameters as context, ListView layout resource id, resource id of string array. This creates the ListView.

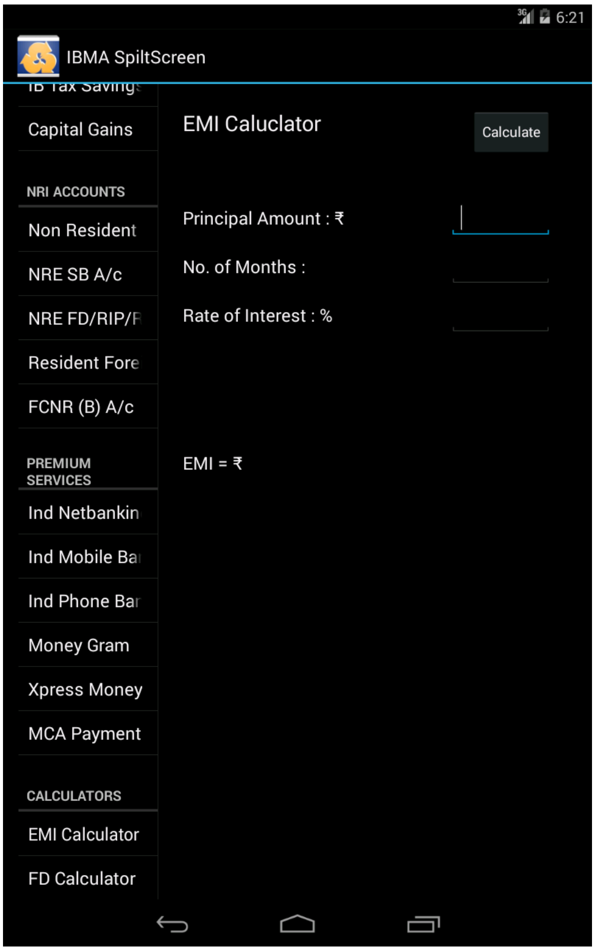


Fig. 5.1. Split Screen Interface

The important task after this is binding the products of the ListView with information of particular products. The Java code of the ListView should be done in a class which extends ListFragment and this fragment is to be added to the container of the MainActivity class layout. The MainActivity is a class that extends Activity class and contains mainactivity.xml as its parent layout. ‘onArticleSelected()’ method of interface which was created in the list fragment activity is implemented in MainActivity class. Whenever user clicks on particular product onListItemClick() method of ListFragment subclass is called which then calls interface method onArticleSelected() which is in MainActivity which has position as its parameter so the code to replace the fragment should be written inside onArticleSelected() method this depending on the position selected.

5.2. Navigation Drawer Interface

The basic working of the Navigation Drawer interface is similar to that of the ListView interface. All the behind the scene implementation is the same. The storing of the details of the various products and services is done in a similar manner using ArrayLists. The linking of the list with the display is also achieved using an Adapter that has been built on the blueprints of the inbuilt ArrayAdapter.

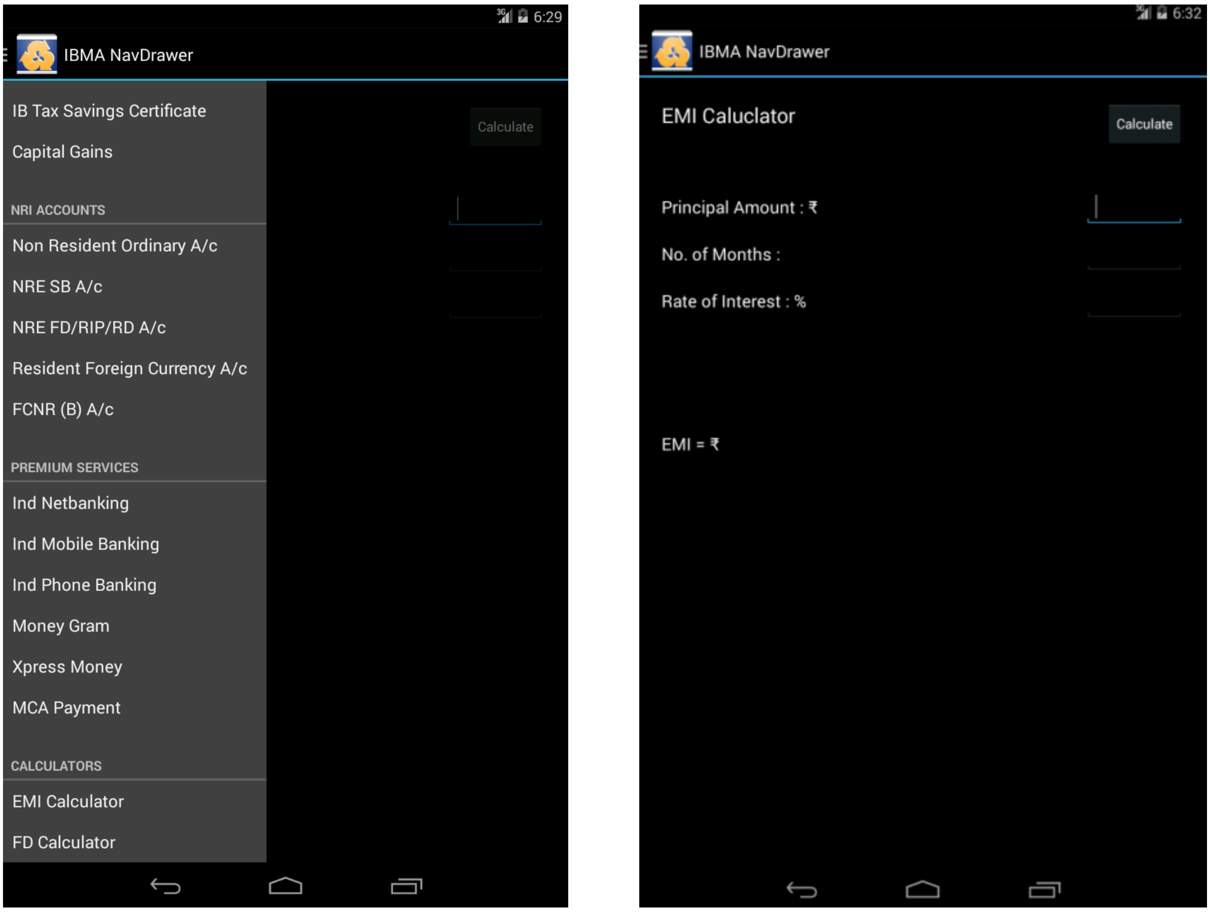


Fig. 5.2. Navigation Drawer Interface

The basic difference lies in the appearance of the application. While the Split Screen Interface has a static ListView that disappears as in the case of small screen devices, the Navigation Drawer Interface is more dynamic. The interface has a Navigation Drawer that can be pulled from the left side of the screen by sliding a finger from the left edge towards the right. On selecting a product/service/calculator the Navigation Drawer slides back towards the left side of the screen. Figure 5.2 depicts the Navigation Drawer Interface with the Navigation Drawer visible and then invisible.

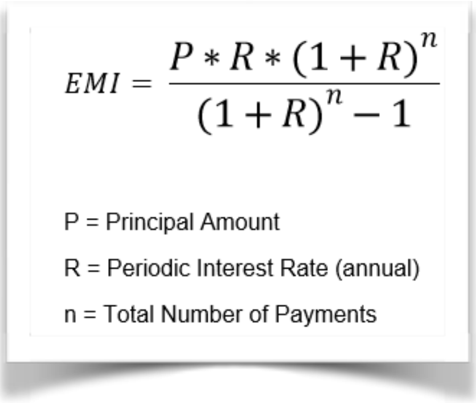
5.3. Calculators

Two calculator have been done as the part of the project. The two calculators are visible in the list of products itself under the section heading ‘Calculators’. The fragments which contain GUIs for the individual calculators has been made depending on the formulas used in the calculations.

The 2 calculators are:

1. EMI Calculator
2. FD Calculator

5.3.1. EMI Formula :



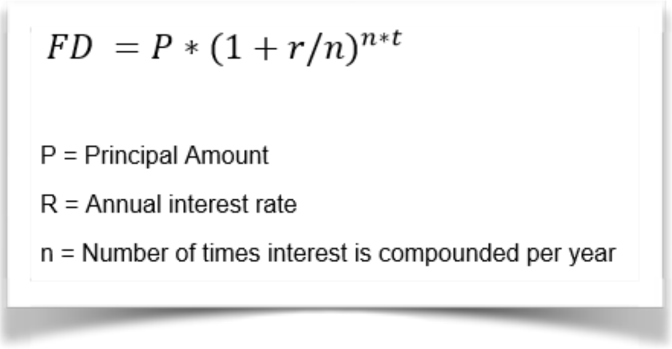
$$EMI = \frac{P * R * (1+R)^{n}}{(1+R)^{n} - 1}$$

P = Principal Amount

R = Periodic Interest Rate (annual)

n = Total Number of Payments

5.3.2. FD Formula :



$$FD \ = P * (1 + r/n)^{n*t}$$

P = Principal Amount

R = Annual interest rate

n = Number of times interest is compounded per year

5.3.3. Constructing UI for formulae :

Each variable name is written in TextViews and EditText boxes are given beside each of them to fill in the respective variable values. The name of the product is given at the top of the layout in order for the users to know the fragment they are in.

There is a 'Calculate' button which when clicked gives the answer in a TextView. An important point is the position of the 'Calculate' button as the soft-keyboard should not cover the button. Hence the button is kept at the top of the layout.

The size of the edit text box is changed for different variables using ems attribute of TextView tag in xml layout. Padding is another important aspect which is used to give margins to the layout components so that the layout looks good. Padding is an attribute of layout which helped in giving margins to whole layout completely. 'InputType' is another important attribute that helped to automatically pop up a soft-keyboard that uses particular variables as input. In the project the input type is phone because the variable input type are only numbers. Relative layout is the layout type that was used in this project because it helps to arrange the UI components more flexibly. Color of the output had been made red and blue so that user can get good focus on answer. This is done by using color attribute of the textview. The color attribute value for red is and the color attribute value for .

5.3.4. Java Code for the Calculator:

Basic math calculations like (product, sum, difference) is done directly using (*, +, -, /) respectively without using any Math class methods. But this calculator requires power operation to be done. So in order to do that the Math class has been used in the program. Math class contain a static method named pow() which calculates power for the parameters. This pow() method is called in the EMI and FD calculator fragment class just by using Math.pow() as pow() is a static method. Math.pow() gives the resultant power which can be used later in the program.

Toast messages are used to display error messages that may arise due to invalid inputs or calculation errors. Exception handling has been implemented that uses 'try-catch' blocks to cover the possible erroneous code. When an error occurs, a Toast message is displayed. Toast messages are small popups that appear towards the lower end of the screen and disappear automatically in a short duration of time.

6. About Python

Python is a widely used general-purpose, high-level programming language. The language was conceived in the late 1980s and its implementation was started in December 1989 by Guido van Rossum at CWI in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C and Java. The language provides constructs intended to enable clear programs on both a small and large scale.

Python is a platform independent programming language that allows code written on one platform to be executed on any other platform with the use of an interpreter. Python programs run faster than Java in most cases and are not much worse than C equivalent programs. Memory comparisons run on similar lines.

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Like other dynamic languages, Python is often used as a scripting language, but is also used in a wide range of non-scripting contexts. Using third-party tools, such as Py2exe, cx_Freeze or Pyinstaller, Python code can be packaged into standalone executable programs. Python interpreters are available for many operating systems.

7. PC based Updating Software

One of the most important aspects of any application are updates. The Android application which has been described in detail earlier is meant to display details of the various services offered by the bank. These details are ever changing and the application needs to be updated whenever any changes in the service parameters are made. The Indian Bank website https://www/indianbank.in is the most reliable and updated source of all of the banks product and service related policies. Had the Android application been an online application achieving auto-updates would be fairly simple. Being an offline application, the application cannot be auto-updated or synchronized automatically with the website.

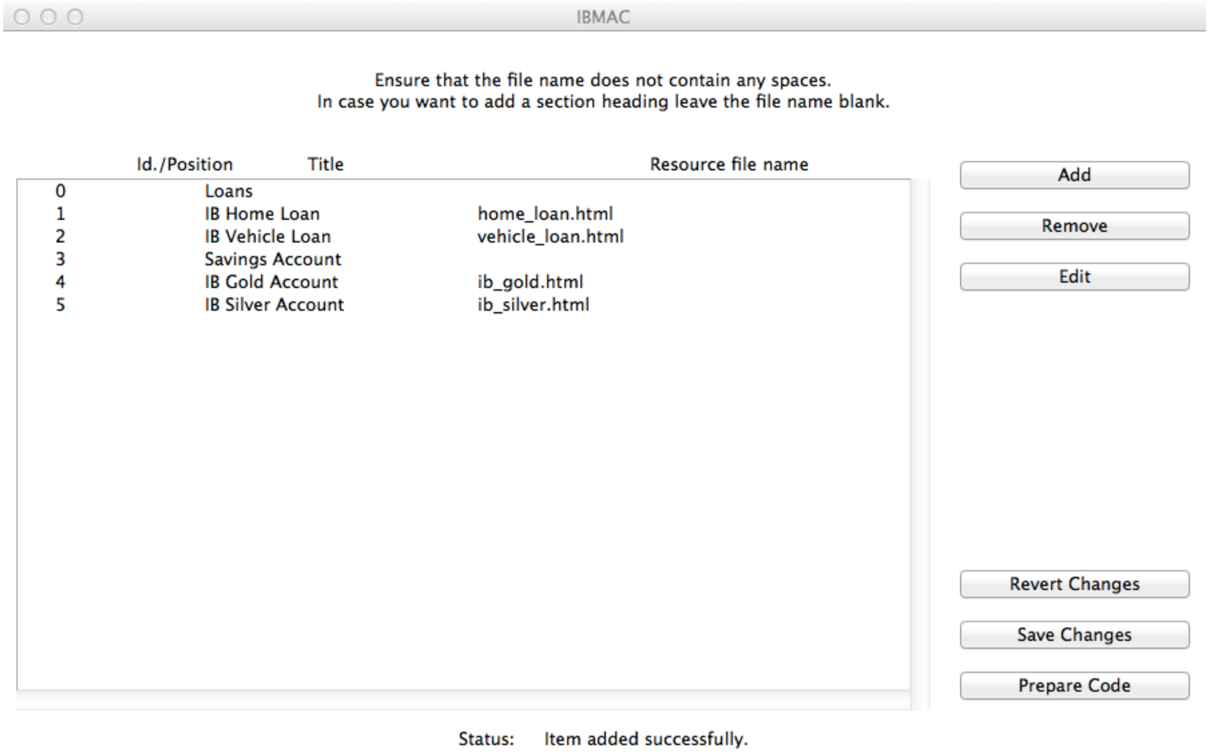
Every time there is a change in the bank's policies, the changes have to be changed in the source code of the Android application, recompiled and redistributed among the Marketing Division employees. The process requires some amount of technical expertise and is not expected to be known by a general employee. It is also not possible to include a new technical member in the marketing team just for the maintenance and update of the application. Hence another software was made that easily allows updates to be made even by non-technical members of the marketing division.

Python being easy to code on coupled with platform independence was chosen as the ideal language for development. The GUI (graphical user interface) development library tinter (TK Interface) also simplified the process of developing a basic UI (user interface).

7.1. Basic Steps of the Software

1. Display an interface that allows user to edit/add/remove products and services.
2. Delete any previously edited versions of the code present in the same directory.
3. Extract the Android application source code which has been stored in a password protected zip file.
4. Copy the resource HTML files into the appropriate folder of the Android code structure.
5. Edit the .Java source code file in order to include the user set products/ services.

7.2. The Interface



Ensure that the file name does not contain any spaces.
In case you want to add a section heading leave the file name blank.

Id./Position	Title	Resource file name
0	Loans	
1	IB Home Loan	home_loan.html
2	IB Vehicle Loan	vehicle_loan.html
3	Savings Account	
4	IB Gold Account	ib_gold.html
5	IB Silver Account	ib_silver.html

Status: Item added successfully.

Fig. 7.1. The Interface

A sample of the interface is shown in Fig. 7.1. The interface was developed using Python's Tkinter library. The name Tkinter comes from Tk interface. Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit and is Python's de facto standard GUI, and is included with the standard Windows and Mac OS X install of Python.

Basic widgets were used in the development of the interface. A widget is a basic GUI tool that is used to design the GUI and assign it some properties. Widgets like labels, buttons, frames, textfields, messages etc. were used. To improve the visual appearance of the application, various tools like padding, margins and aligning were used.

The interface includes options to add/remove/edit products and services that are to be included in the Android application. To add a new product, the user has to press 'Add', input the position number, title and name of the file that contains the details of the product. To include a product in between, the position number can be given as a decimal, e.g.. 1.5. This will add the product between 1 and 2 and reorder the entire table. To add a section heading for a group of products, click 'Add', enter the position number and title and leave the filename blank.

Any changes that are made by the user can be saved. The details are stored in a CSV file for easy access. The changes made during a session can also be reverted to the last saved configuration available.

7.3. Using the Software

After adding/editing/removing the different products to display in the Android application as mentioned above, the user can save the changes made and click on ‘Prepare Code’. On clicking the button, the software shall ask the user to save the changes that have been made. On confirmation, the changes shall be saved and the steps mentioned under Section 7.1 shall be executed. By the end of the process, the user shall have a version of both the templates, ready for compilation by the Android Studio IDE.

The user has to open the project source code generated by the Python based updating software using Android Studio and build the application to generate a APK file that can be distributed to the Marketing employee who has an Android based smartphone. In addition to the products and services entered by the user in the Python PC Software, 2 calculators shall also be added to the application by default under the heading ‘Calculator’. The ‘Calculator’ section shall be the last section in the list and contains a EMI calculator and a FD calculator.

The Python code has been written in a platform independent manner so that it can be run on Windows, Linux as well as Mac OS X platforms without any changes in behavior. The IDE, Android Studio is also easily available for free for all the above mentioned platforms.

Conclusions & Recommendations

Having already spent 8 weeks at Indian Bank, certainly a lot has been learnt. A fruitful exposure to the corporate world and the industry into which students would most probably enter after graduation has been well received, experienced and understood. The project that was allotted has involved a great deal of research, learning and work. The project has seen a great deal of development with great scope of improvement and improvisation. Sticking to timelines and reporting aptly to the seniors in-charge for smooth workflow in the company has been one of the various lessons learnt here. Also the experience of an office environment gave a feel of what the future holds and how to better prepare to be equipped for what is to come. The orientation session at IMAGE helped understand in greater depth about the functioning, present status and history of Indian Bank.

As a part of the Marketing Division, a lot was observed and learned about the processes involved in publicizing and promoting the various products and schemes that are available. Although a lot of improvement has already taken place in the process of marketing, a lot of automation by the use of computer software is still possible.

The project has been a learning and fruit filling experience. The application has been successfully developed and tested and can now be deployed to the Marketing officials. A documentation and the source code has also been provided to that the bank can easily edit and manipulate the application. Apart from the project, preparing the report also taught many aspects which were not known earlier. A heartfelt thank you to everyone associated with the project and the experience at Indian Bank.

References

- http://www.indianbank.in/home.php
- http://www.indianbank.in/cpps.php
- http://developer.android.com/index.html
- http://en.wikipedia.org/wiki/Android_(operating_system)
- http://www.vogella.com/tutorials/Android/article.html
- http://en.wikipedia.org/wiki/Indian_Bank
- http://en.wikipedia.org/wiki/Self-help_group_(finance)
- http://en.wikipedia.org/wiki/Python_(programming_language)
- https://www.python.org/
- Orientation session at IMAGE
- Employs at Indian Bank

Glossary

Activity: An activity is a single, focused thing that the user can do.

Android: Android is a mobile operating system (OS) based on the Linux kernel that is currently developed by Google.

Android Studio: Android Studio is an integrated development environment (IDE) for the Android platform.

Broadcast Receivers: A *broadcast receiver* is a component that responds to system-wide broadcast announcements.

Content Providers: Content providers manage access to a structured set of data. They encapsulate the data, and provide mechanisms for defining data security.

EMI: An Equated Monthly Installment (EMI) is defined by Investopedia as "A fixed payment amount made by a borrower to a lender at a specified date each calendar month.

FD: A fixed deposit (FD) is a financial instrument provided by banks which provides investors with a higher rate of interest than a regular savings account.

FOSS: Free and open source software.

IDE: An integrated development environment (IDE) or interactive development environment is a software application that provides comprehensive facilities to computer programmers for software development.

Python: Python is a widely used general-purpose, high-level programming language.

Service: A Service is an application component representing either an application's desire to perform a longer-running operation while not interacting with the user or to supply functionality for other applications to use.

Widget: A graphical control element or widget is an element of interaction in a graphical user interface (GUI), such as a button or a scroll bar.