



CHRIST
UNIVERSITY

B A N G A L O R E , I N D I A

Declared as Deemed to be University under Section 3 of UGC Act 1956

@CU

by

Akshay Sadarangani (1115905)
Lorraine Amanda Hoover (1115931)

Under the guidance of
Ms. Rupali Wagh
&
Ms. Smitha Vinod

A project report submitted in partial fulfillment of the
requirements for the award of degree of Bachelor of
Computer Applications of Christ University

March - 2014



CHRIST
UNIVERSITY

B A N G A L O R E , I N D I A

Declared as Deemed to be University under Section 3 of UGC Act 1956

CERTIFICATE

This is to certify that the report titled @CU is a bona fide record of work done by Akshay Sadarangani (1115905) and Lorraine Amanda Hoover (1115931) of Christ University, Bangalore, in partial fulfillment of the requirements of VI Semester BCA during the year 2014.

Head of the Department

Project Guide

Valued-by:

1.	Name	: Akshay Sadarangani
	Register Number	: 1115905
	Examination Centre	: Christ University
2.	Date of Exam	:

ACKNOWLEDGEMENTS

@CU would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We would like to thank our parents, for their care, encouragement and being our moral support at all stages of the project. We would like to thank our Head of the Department, Coordinator and faculty staff of the Department of Computer Science, Christ University. We are thankful to Ms. Rupali Wagh and Ms. Smitha Vinod and for their guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

Sincere thanks to the FHWS faculty for their kind co-operation and encouragement which helped us in the completion of this project.

We would like to express our special gratitude and thanks to the industry people for giving us their time and attention. Our thanks and appreciation also goes to all our colleagues from FHWS and Christ University who were part of the India Gateway Programme (IGP) in developing the project and our friends who have willingly helped us out with their abilities.

Last but not the least, a special and sincere thanks to the Christ University Lab administrators and lab staff for allowing us to use the systems in complete freedom and supplying us with all the software needed and their technical support.

This project would not have been possible without every contribution, support and guidance of everyone mentioned.

ABSTRACT

Foreign students coming to Christ University, India face problems in getting access to information and services. Language barrier adds to the lack of proper guidance.

Currently the best source of information available to them is via the Internet viz. Google and

Google Maps which has various assorted data with its limitations of information not pertaining to the students.

The proposed system aims to solve the various difficulties faced by the international students by incorporating all necessary and relevant information on a mobile application which is built on the Android platform. The application is meant to aid students in travelling and exploring in-campus areas as well as areas around the campus.

The features of the proposed system include news and updates pertaining to the university, events which are directly synchronized with the built-in calendar and aided by push notifications, multimedia gallery for photos and videos of the university.

Apart from the in-campus features, the application will also have features such as transportation finder to help find the easiest and best possible route to and from the campus, shopping areas, hotels and eateries near the campus, places of interest in the city and emergency contact services. All this in a language suitable for the students i.e. English or German.

LIST OF FIGURES

Figure Number	Figure Name	Page Number
3.1	Block Diagram	10
4.1	System Flowchart	16
4.2	Map Flowchart	17
4.3	Notes Flowchart	18
4.4	Currency Flowchart	19
4.5	Translator Flowchart	20
4.6	Table Design	23
4.7	ER Diagram	26
4.8	Login Screen	27
4.9	Home Screen	27
4.10	Dashboard Screen	28
4.11	Maps Screen	28
4.12	Translator Screen	29
4.13	Currency Rate Screen	29
4.14	Contacts Screen	30
4.15	Notes Screen	30
6.1	Note Creation	49
6.2	Note Rename	49
6.3	Note Deletion	50
6.4	Currency- Blank Field	50
6.5	Translator- No Internet	51

LIST OF TABLES

Table Number	Table Name	Page Number
1.1	Project Plan	2
3.1	Functional Requirements	12
4.1	Remote- Login Table	21
4.2	Remote- GCM_Details Table	21
4.3	Remote- Events Table	21
4.4	Remote- Notes Table	21
4.5	Local- Login Table	22
4.6	Local- Events Table	22
4.7	Local- Notes Table	22
4.8	Data Dictionary	24
6.1	Test Cases	47

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	III
ABSTRACT.....	IV
LIST OF FIGURES	V
LIST OF TABLES	VI
1. INTRODUCTION	1
1.1 OVERVIEW OF THE SYSTEM.....	1
1.2 PROJECT PLAN	2
2. SYSTEM ANALYSIS	6
2.1 EXISTING SYSTEM.....	6
2.2 PROPOSED SYSTEM.....	6
2.3 LITERATURE REVIEW	8
2.4 SOFTWARE TOOLS USED	8
3. SYSTEM REQUIREMENTS	10
3.1 SYSTEM MODEL.....	10
3.2 FUNCTIONAL REQUIREMENTS	12
3.3 HARDWARE REQUIREMENTS.....	14
3.4 SOFTWARE REQUIREMENTS	14
3.5 ADDITIONAL REQUIREMENTS	14
4. DESIGN SPECIFICATION	15
4.1 ARCHITECTURAL DESIGN.....	15
4.2 FLOWCHARTS.....	16
4.3 DATABASE DESIGN.....	21
4.4 USER INTERFACE DESIGN	27
5. IMPLEMENTATION.....	31
5.1 SOURCE CODE	31
6. TESTING.....	45
6.1 TEST PLAN.....	45
6.2 TEST STRATEGIES	46
6.3 TEST CASES.....	47
6.4 TEST REPORT.....	49
7. CONCLUSION.....	52

7.1	ADVANTAGES.....	52
7.2	LIMITATIONS	52
7.3	FUTURE ENHANCEMENTS.....	53
	REFERENCES	54

1. INTRODUCTION

1.1 OVERVIEW OF THE SYSTEM

International students coming to Christ University, India from different parts of the world face problems in getting access to information and services. Language barrier adds to the lack of proper guidance. Currently the best source of information available to them is via the Internet viz. Google and Google Maps which has various assorted data with its limitations of information not pertaining to the students.

The proposed system aims to solve the various anomalies faced by the international students by incorporating all necessary and relevant information on a mobile application which is built on the Android platform. The application is meant to aid students in travelling and exploring in-campus areas as well as areas around the campus.

The features of the proposed system include a virtual tour of the entire campus, news and updates pertaining to the university, events which are directly synchronized with the built-in calendar, multimedia gallery for photos and videos of the university, library services for searching availability of books, food finder, people directory and a universal search. Apart from the in-campus features, the application will also have features such as transportation finder to help find the easiest and best possible route to and from the campus, shopping areas, hotels and eateries near the campus, places of interest in the city and emergency contact services. All this in a language suitable for the students i.e. English, German, French, etc.

1.2PROJECT PLAN

Table 1.1: Project Plan

Student Name: Akshay Sadarangani, Lorraine Hoover	Register Number: 1115905 and 1115931
Title: @CU	
Department: Computer Science	Guides: Rupali Wagh and Smitha Vinod

Date	Phase	Start Time	End Time	Regular Hours	Overtime Hours	Total Hours
5/11/13	Title Discussion	02:00pm	04:00pm	2	0	2
6/11/13	Introductory Phase	09:00am	01:00pm	4	0	4
7/11/13	Introductory Phase	11:00am	04:00pm	4	0	4
8/11/13	Introductory Phase	11:00am	04:00pm	4	0	4
11/11/13	Synopsis Submission	09:00am	11:00am	2	1	3
12/11/13	Requirement Analysis	02:00pm	04:00pm	2	0	2
13/11/13	Requirement Analysis	09:00am	01:00pm	4	0	4
14/11/13	Requirement Analysis	11:00am	04:00pm	5	1	6
15/11/13	Synopsis Presentation	11:00am	04:00pm	4	1	5
18/11/13	Requirement Analysis	09:00am	11:00am	2	0	2
19/11/13	Requirement Analysis	02:00pm	04:00pm	2	0	2

20/11/13	Requirement Analysis	09:00am	01:00pm	4	0	4
21/11/13	Requirement Analysis	11:00am	04:00pm	4	0	4
22/11/13	Requirement Analysis	11:00am	04:00pm	4	1	5
25/11/13	System Design	09:00am	11:00am	2	0	2
26/11/13	System Design	02:00pm	04:00pm	2	2	4
27/11/13	System Design	09:00am	01:00pm	4	0	4
28/11/13	System Design	11:00am	04:00pm	4	0	4
29/11/13	System Design	11:00am	04:00pm	4	1	5
2/12/13	System Design	09:00am	11:00am	2	1	3
3/12/13	System Design	02:00pm	04:00pm	2	0	2
4/12/13	System Design	09:00am	01:00pm	4	0	4
5/12/13	System Design	11:00am	04:00pm	4	0	4
6/12/13	System Design	11:00am	04:00pm	4	0	4
9/12/13	System Design	09:00am	11:00am	2	0	2
10/12/13	System Design	02:00pm	04:00pm	4	0	4
12/12/13	System Design	09:00am	01:00pm	4	0	4

13/12/13	System Design	11:00am	04:00pm	4	1	5
17/12/13	System Development	06:00pm	08:00pm	0	2	2
19/12/13	System Development	05:00pm	08:00pm	0	3	3
26/12/13	System Development	10:00am	12:00pm	0	2	2
27/12/13	System Development	11:00am	04:00pm	0	4	4
28/12/13	System Development	04:00pm	05:00pm	0	1	0
02/01/14	System Development	11:00am	04:00pm	4	0	4
03/01/14	System Development	11:00am	04:00pm	4	0	4
06/01/14	System Development	09:00am	11:00am	2	2	4
07/01/14	System Development	02:00pm	04:00pm	2	2	4
08/01/14	System Development	09:00am	01:00pm	4	1	5
09/01/14	System Development	11:00am	04:00pm	4	0	4
10/01/14	System Development	11:00am	04:00pm	4	0	4
20/01/14	System Development	09:00am	11:00am	2	1	3
21/01/14	System Development	02:00pm	04:00pm	2	1	3
22/01/14	System Development	09:00am	01:00pm	4	1	5

23/01/14	System Development	11:00am	04:00pm	4	2	6
24/01/14	System Development	11:00am	04:00pm	4	2	6
25/01/14	System Development	1:00pm	05:00pm	0	4	4
27/01/14	System Development	09:00am	01:00pm	2	2	4
28/01/14	Testing	02:00pm	05:00pm	2	1	3
29/01/14	Testing	09:00am	03:00pm	4	2	6
30/01/14	Testing	11:00am	05:00pm	4	1	5
31/01/14	Testing	11:00am	04:00pm	4	0	4
3/02/14	Testing	09:00am	11:00am	2	0	2
4/02/14	Testing	02:00pm	05:00pm	2	1	3
5/02/14	Testing	09:00am	01:00pm	4	0	4
6/02/14	Testing	11:00am	05:00pm	4	2	6
7/02/14	Testing	11:00am	04:00pm	4	1	5
10/02/14	Testing	09:00am	11:00am	2	0	2
11/02/14	Testing	02:00pm	04:00pm	2	0	2
12/02/14	Testing	09:00am	02:00pm	4	1	5
13/02/14	Testing	11:00am	04:00pm	4	0	4
14/02/14	Testing	11:00am	05:00pm	4	1	5
17/02/14	Testing	09:00am	11:00am	2	0	2
19/02/14	Testing	03:00pm	06:00pm	1	2	3
20/02/14	Testing	11:00am	04:00pm	4	0	4
21/02/14	Testing	11:00am	04:00pm	4	0	4
24/02/14	Testing	09:00am	11:00am	2	0	2
25/02/14	Testing	04:00pm	08:00pm	0	4	4
27/02/14	Testing	10:00am	08:00pm	0	8	8
28/02/14	Testing	08:00am	10:00am	0	2	2
Total				193	66	259

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

There is no existing system to provide the required services and information in an android application. The existing facilities provide information which is distributed among different locations and services. For navigation, users need to use map services such as Google Maps, MapMyIndia, etc. and for finding information related to the campus they need to either visit the Christ University website or physically visit the campus and find out.

LIMITATIONS OF THE EXISTING SYSTEM

- As there is no such application available in the current market, there is not much required information available to the user.
- Users need to search for different components in different places and there is no centralized application for the desired information and services.
- Users face the difficulty of language and navigation as well.
- No campus map
- No virtual tour of the campus
- No dedicated routing service, etc.

2.2 PROPOSED SYSTEM

The proposed system aims to solve the problems of international students who visit Christ University, Bangalore for the India Gateway Programme (IGP) or otherwise. This system encapsulates a variety of information sources, services and utilities which are required by the students and specifically documented by the German students of Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt. With the help of this application, users will be able to navigate their way around the campus and outside campus easily and all their required information will be at their fingertips. With @CU, users will be constantly updated with the latest happenings in the campus and will be able to connect with other students easily.

This application is being developed on one of the most widely available and portable operating system existing in the market, that is, the Android OS. The android mobile application will help the users in quick access of information and will help them in keeping track of their routine tasks. With features such as automatic sync of data, the users can easily shift between devices and still retain their data just by logging in from their existing login credentials.

BENEFITS OF THE PROPOSED SYSTEM

- A one of a kind all-in-one mobile application with an easy to use user interface (UI)
- Multiple-language interface- English, German, French, etc.
- Integrated components with optimized code to keep the application light-weight and avoid memory constraints
- Beginner friendliness for new users who are not used to the application structure
- Interactive and attractive graphical user interface (GUI)
- Centralize database to store user data remotely
- Local database to make data available offline
- Use of fragments for re-use of component
- Flexible coding standards for future enhancements
- Help overlays to direct beginners
- Basic information to help students learn about the provisions of the campus
- Event updates to keep students updated with the current happenings in the university
- Routing with step-by-step directions and high accuracy (GLONASS support) to keep the students in control of their movements
- Social connect with other students for quick connection
- Currency convertor to help the students get an estimate of their expenses
- Language translator with voice playback of translations to help the students communicate in local languages
- Push notifications of events gives immediate event news to users and eradicates the need to constantly open the application and refresh feeds
- Daily schedule of events with calendar integration and reminder support

- Emergency contacts with direct-dial support to have a one-touch dialling interface

@CU is a one of a kind all-in-one application running on the Android OS which helps students in reducing their work load by providing a variety of top-class services. It is the ultimate utility application which will help all international students coming to Christ University to stay updated and informed about their surroundings.

2.3 LITERATURE REVIEW

WEBSITES

- <http://www.stackoverflow.com>
- <http://developer.android.com>
- <http://www.androidhive.com>
- <http://www.androidexample.com>

BOOKS

- Professional Android 4 Application Development by Retro Meier

2.4 SOFTWARE TOOLS USED

FRONT END

Android Development Tools: ADT (Android Developer Tools) is a plugin for Eclipse that provides a suite of tools that are integrated with the Eclipse IDE. It offers access to many features that help develop Android applications quickly. ADT provides GUI access to many of the command line SDK tools as well as a UI design tool for rapid prototyping, designing, and building of the application's user interface. Because ADT is a plugin for Eclipse, it encompasses the functionality of a well-established IDE, along with Android-specific features that are bundled with ADT.

Android SDK: The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU (Quick EMUlator), documentation, sample code, and tutorials.

Android NDK: Libraries written in C and other languages can be compiled to ARM, MIPS or x86 native code and installed using the Android Native Development Kit. Native classes can be called from Java code running under the Dalvik VM using the `System.loadLibrary` call, which is part of the standard Android Java classes.

Eclipse IDE: In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications in Java.

Adobe Photoshop CC: Adobe Photoshop is a graphics editing program developed and published by Adobe Systems. Adobe Photoshop CC is the latest addition to the Adobe Creative Cloud Suite. This project makes use of Adobe Photoshop CC for its various graphical layouts used such as icons and background gradients.

BACK END

MySQL: MySQL is the world's second most widely used open-source relational database management system (RDBMS). This project makes use of MySQL as a centralized remote database.

SQLite: SQLite is a relational database management system (RDBMS) contained in a C programming library. In contrast to other database management systems, SQLite is not a separate process that is accessed from the client application, but an integral part of it. Unlike client–server database management systems, the SQLite engine has no standalone processes with which the application program communicates. Instead, the SQLite library is linked in and thus becomes an integral part of the application program. The application program uses SQLite's functionality through simple function calls, which reduce latency in database access.

PHP: PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Its main advantages are that it is a very simple and easy-to-use yet very powerful scripting language and it is open source.

3. SYSTEM REQUIREMENTS

3.1 SYSTEM MODEL

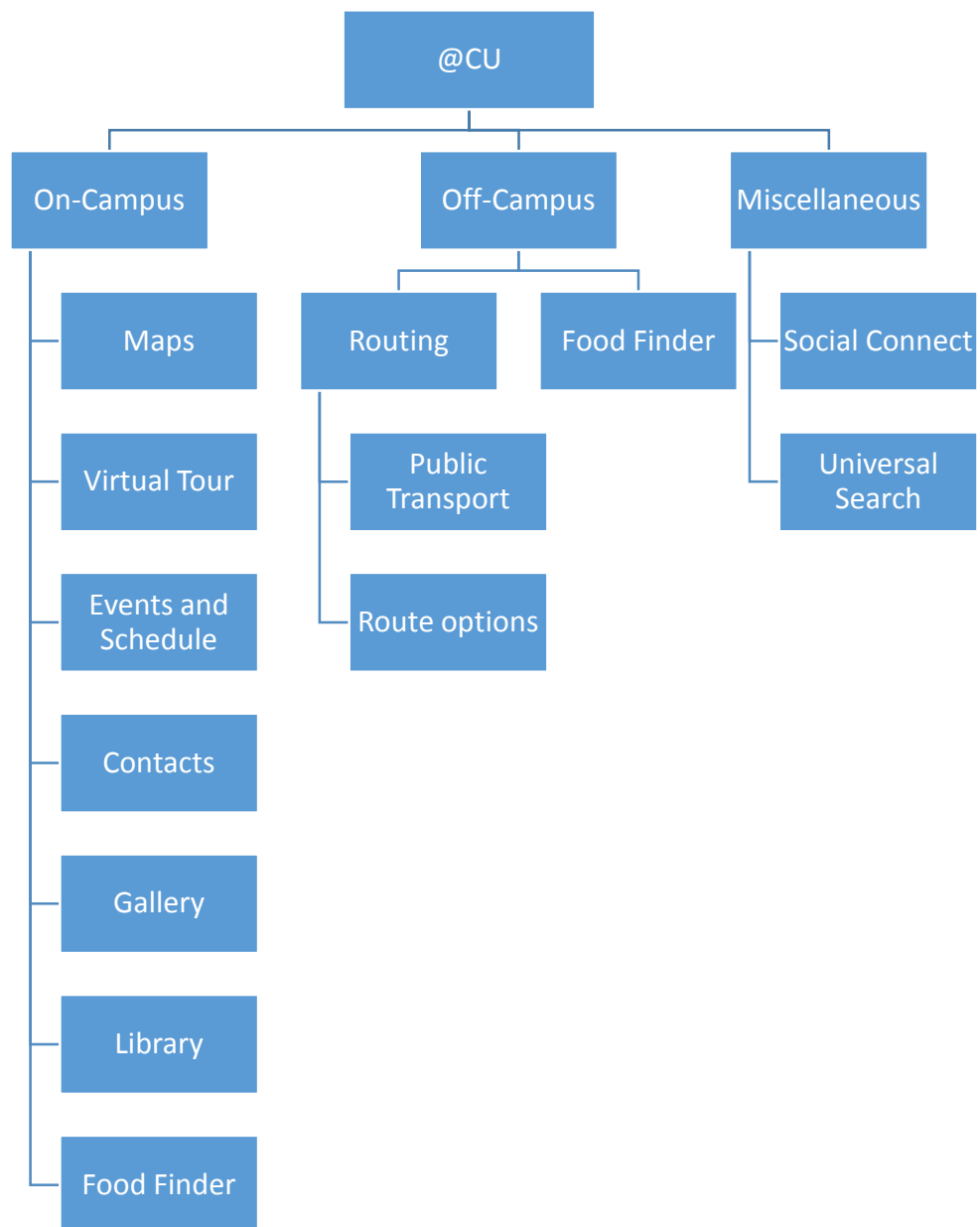


Fig 3.1: Block Diagram Model

SYSTEM MODULES

Maps: To cover the campus and area around for easy navigation using GPS or network signals

Virtual Tour: Virtual tour of the entire campus to help the students locate and navigate each part of the campus virtually

Events: The latest events in and around campus synced to the mobile calendar

Schedule: The daily schedule of students fetched from the centralized database

Contacts: Local contact information of campus with one-click direct dial numbers

Gallery: Campus liveTV, photo gallery and videos available on the university website

Library: Search for book availability using book name/author information/publisher and check for due date of issued books and location of books

Food Finder: Search for food in and around campus with filters like cuisine and food item

Routing: Get directions to and from your location to anywhere in the city. Public transit options are available for easy travelling.

Social Connect: Connect with other students from social network links and friends in the same or different courses

Universal Search: Enter a search key to search the entire application resources and get search results from various search areas

3.2 FUNCTIONAL REQUIREMENTS

Table 3.1: Functional Requirements

Modules	Entries	Function	Requirement
Registration	Input: A username and a valid Email ID Output: User entry in database and access to application	Get the user's Email ID for future communication and register device into database	A valid Email ID for activation of account
Push Notification	Input: Event addition in database Output: Client gets a notification about new events	Notifies users about new and upcoming events in campus without having to open the application	Event change in database as calculated by a Cron Job
Events	Input: New event Output: New calendar entry with reminder option	Syncs calendar entries with latest events	There must be an event to sync
One-click call	Input: User clicks on a phone number Output: Direct dial of the number	Call a contact by tapping a phone number	Phone number must have proper extensions
Routing	Input: Pick a point to travel to or from Output: Shortest route with directions including public transit information	Get the route for a point in the map to travel to from campus or to reach campus from it	Point on map must be selected

Map	Input: User touch Output: Places near user's location highlighted on the map	Show current location with nearby places	GPS or location through network must be enabled
Virtual Tour	Input: User's choice of in-campus location Output: 3D tour of the campus	3D tour of the campus with each places	User selection
Social Connect	Input: User's login credentials Output: Social media links and user's friends	Social connect	Successful authentication of users
Gallery	liveTV, photos and videos	Media gallery	Media files in the repository
Library	Input: Book name/author name/publisher/category Output: Book availability, location and status	Library book finder	Book must be available
Food Finder	Input: Cuisine, item, location Output: Location	Find food in and around campus	Food item availability in the perimeter
Universal Search	Input: Search key Output: Search results	Search for anything as part of the app	Non-null search key
Language Translator	Input: Native language Output: Translated result with voice support	Translate between various languages	Input text must be provided for translation
Currency Convertor	Input: Input and output currency format Output: Converted currency amount	Convert currency between various formats with live conversion rates	Both input and output currency formats need to be specified

3.3 HARDWARE REQUIREMENTS

CPU: 600 MHz ARMv7 and above

RAM: 256 MB and above

GPU: Any recent GPU

Storage: 5 MB and above

GPS with GLONASS (recommended)

3.4 SOFTWARE REQUIREMENTS

Android 2.2 (Froyo) and above

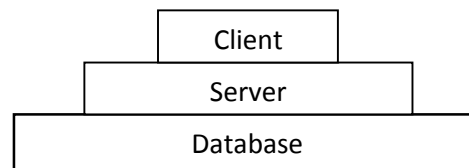
3.5 ADDITIONAL REQUIREMENTS

Working data pack for online content

4. DESIGN SPECIFICATION

4.1 ARCHITECTURAL DESIGN

@CU implements a 3-tier architecture to support all the functionalities of the application.



The database tier forms a base for most of the modules of the application where data storage is necessary so it could be retrieved in a later stage or be used for any other analytical purposes. The database used for this application is MySQL and SQLite as these are the most compatible databases which can run fast and easily on a server and locally respectively.

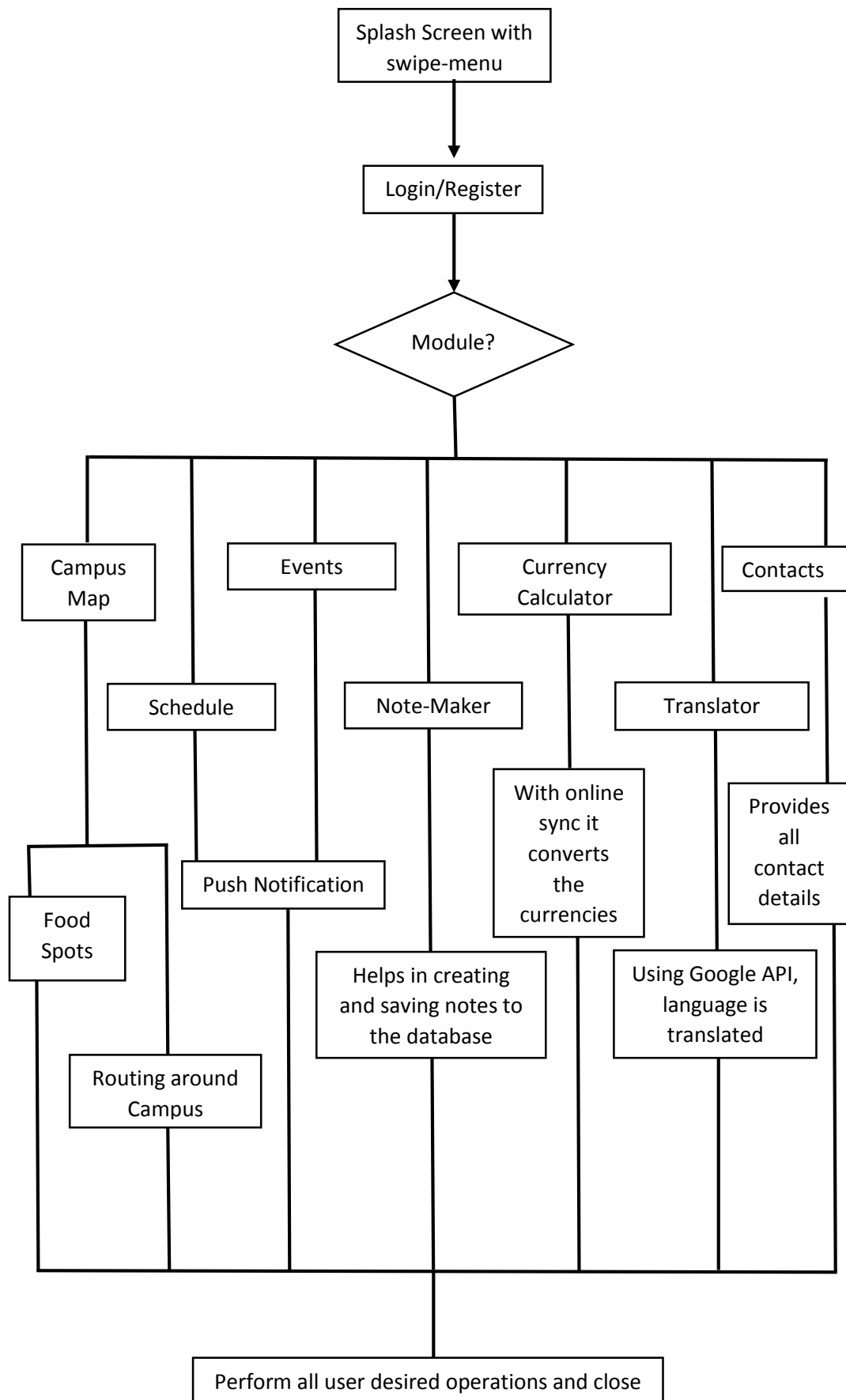
The server tier which provides the connectivity for the client to the database through PHP script. It also includes the Java code of the application which goes about the manipulation of complete data that are relevant to the application. Java initiates the connectivity to the database. In general a server helps in performing the logical tasks of the application where necessary or just completes tasks as per the user's inputs.

The client tier or the final tier is from where the users get to access the application. The main component of this tier is the user interface which contains each module of the application in a simplistic view to make it easy for the user to utilize the application. Since this application is being created keeping in mind the clients' needs, therefore the layouts are also decided for the user interface based on their requirements.

SUBSYSTEMS

- **Currency Calculator:** Since currency rates keep changing every now and then, a separate system is needed to keep up with these changes. So that at any moment when the calculator is put to use it helps in calculating according to the present rates. The calculator is kept in sync with an online site to calculate according to the current rates.
- **Translator:** When changes are made in the content of data/language in the User Interface layout, when the user decides to translate the content another subsystem is required to perform the translation. This translation is done using the Google Translator API.

4.2 FLOWCHARTS



MODULE ARCHITECTURE

MODULE: MAPS

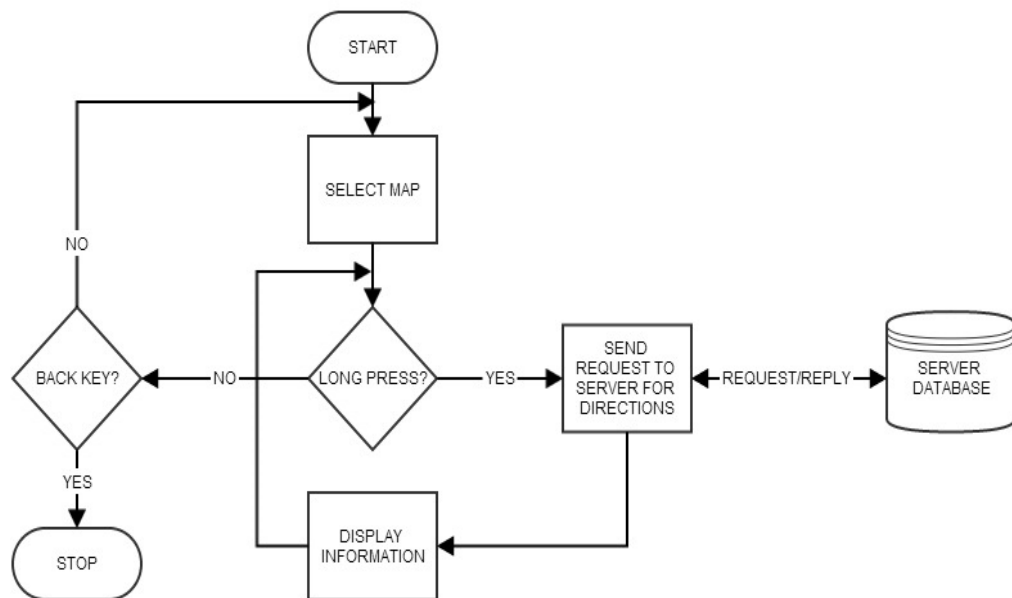


Fig 4.2: Map Module Flowchart

MODULE: NOTES

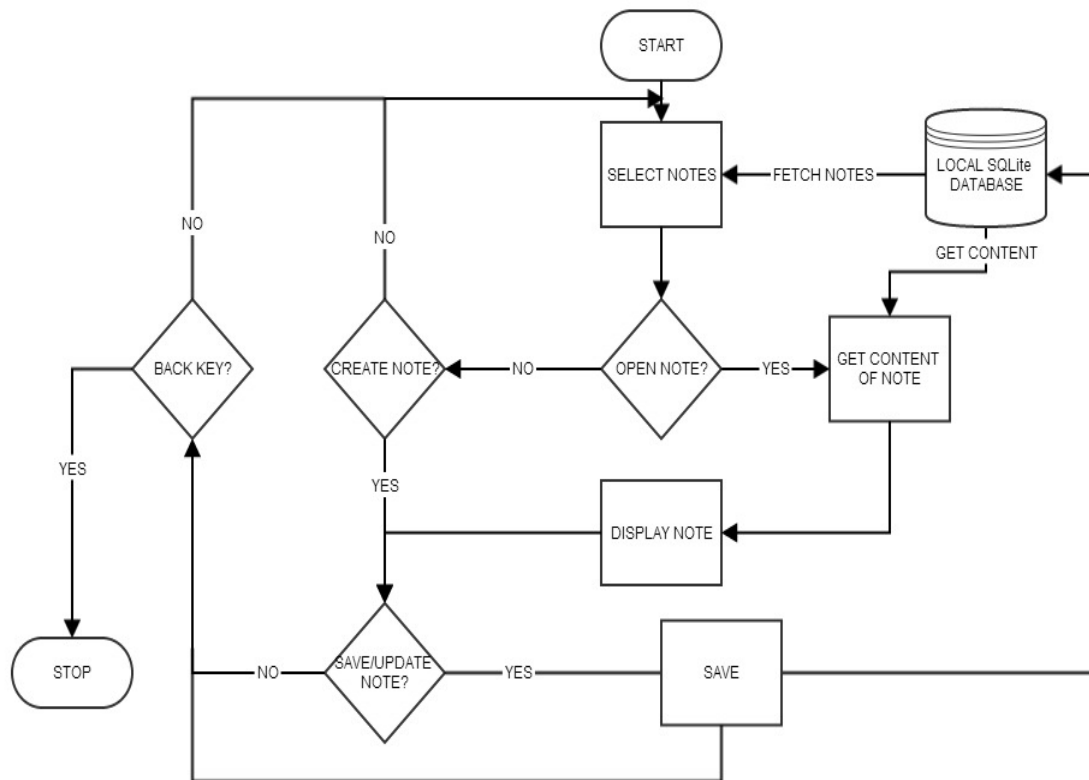
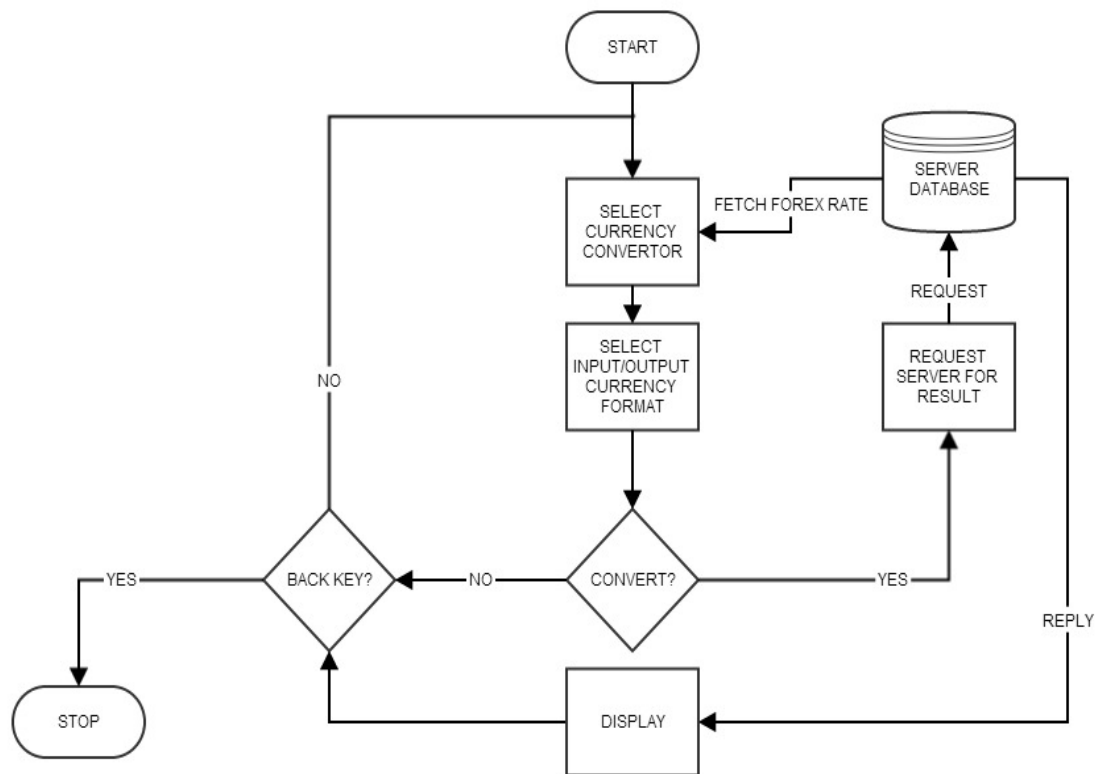
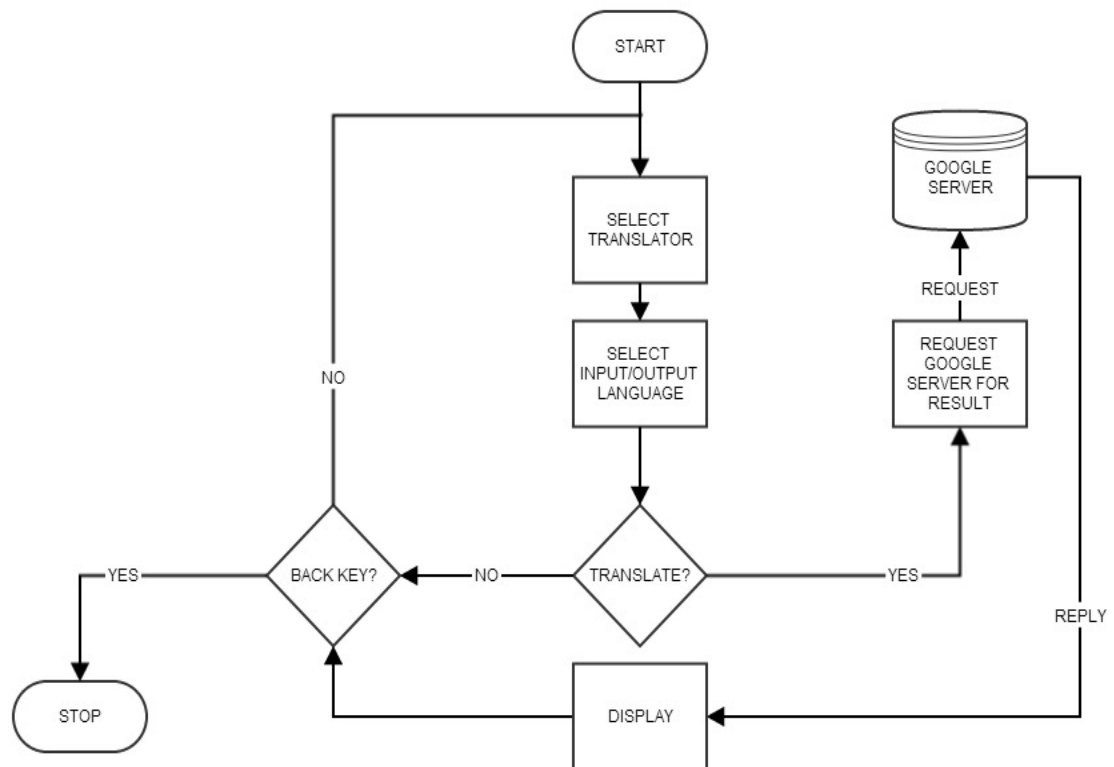


Fig 4.3: Notes Module Flowchart

MODULE: CURRENCY CONVERTOR**Fig 4.4: Currency Converter Module Flowchart**

MODULE: TRANSLATOR**Fig 4.5: Translator Module Flowchart**

4.3 DATABASE DESIGN

4.3.1 TABLE DESIGN

The table design gives a tabular representation of all the tables that are required by @CU with each and every table related to its respective module. Each table has a primary key marked in bold and is underlined whereas some tables also have a foreign key which have been represented with a star next to it. @CU makes use of a centralized database as well as a local database for online and offline content respectively.

REMOTE DATABASE

Login Table

Table 4.1: Login Table

<u>UID</u>	Username	Password	Email	Status
-------------------	----------	----------	-------	--------

GCM_Details Table

Table 4.2: GCM_Details Table

<u>ID</u>	GCM_Reg_ID	Name*	Email*
------------------	------------	-------	--------

Events Table

Table 4.3: Events Table

<u>ID</u>	Event	Created_At
------------------	-------	------------

Notes Table

Table 4.4: Notes Table

<u>ID</u>	UID*	Username*	Note_Name	Content
------------------	------	-----------	-----------	---------

LOCAL DATABASE

Login Table

Table 4.5: Login Table

<u>UID</u>	Username	Email	Status
------------	----------	-------	--------

Events

Table 4.6: Events Table

<u>ID</u>	Event	Created_At
-----------	-------	------------

Notes Table

Table 4.7: Notes Table

<u>ID</u>	UID*	Username*	Note_Name	Content
-----------	------	-----------	-----------	---------

4.3.2 TABLE DESIGN

The diagram below show the inter connectivity of all the tables to each other with arrows pointing the direction of flow.

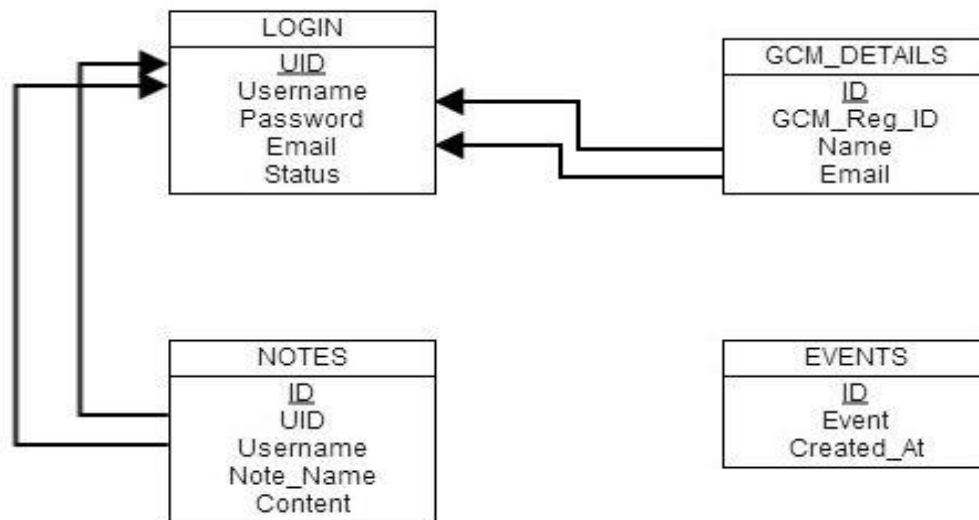


Fig 4.6: Table Design

4.3.2 DATA DICTIONARY

Table 4.8: Data Dictionary

Serial No.	Table Name	Field	Type	Size	Default	Description
1.	Login	UID	Varchar(2)	21	-	IMEI number which is unique for each phone.
		Username	Varchar(2)	30	-	The name used for logging in by the user.
		Password	Varchar(2)	15	-	A method of encrypting the user's access.
		Email	Varchar(2)	45	-	User's email ID is used for activation purpose.
		Status	Varchar(2)	15	Pending	Keeps tabs on whether the user's account has been activated.
2.	GCM_Details	ID	Integer	4	A_I	Serial number that auto increments based on

						every new user.
		GCM_Reg_ID	Varchar(2)	50	-	Google Cloud Messaging Identification Number.
		Username	Varchar(2)	30	Current User	To refer to the username of the current phone.
		Email	Varchar(2)	45	Current User	To refer the mail ID used by the current phone
3.	Events	ID	Integer	3	0	Unique identification of each event.
		Event	Varchar(2)	25	-	Name of the event.
		Created_at	Date	-	Current Date	Date of when the event is going to take place.
4.	Notes	ID	Integer	3	A_I	Serial number which is unique for each note and increments based on every new note added.

		UID	Varchar(2)	21	Current User	Based on the UID the personal note-maker of the user is loaded.
		Username	Varchar(2)	30	Current User	Saves the notes created in the local server under the present username.
		Note_Name	Varchar(2)	20	-	Name used to identify the note
		Content	Varchar(2)	250	-	Contains the note body.

4.3.3 ER DIAGRAM

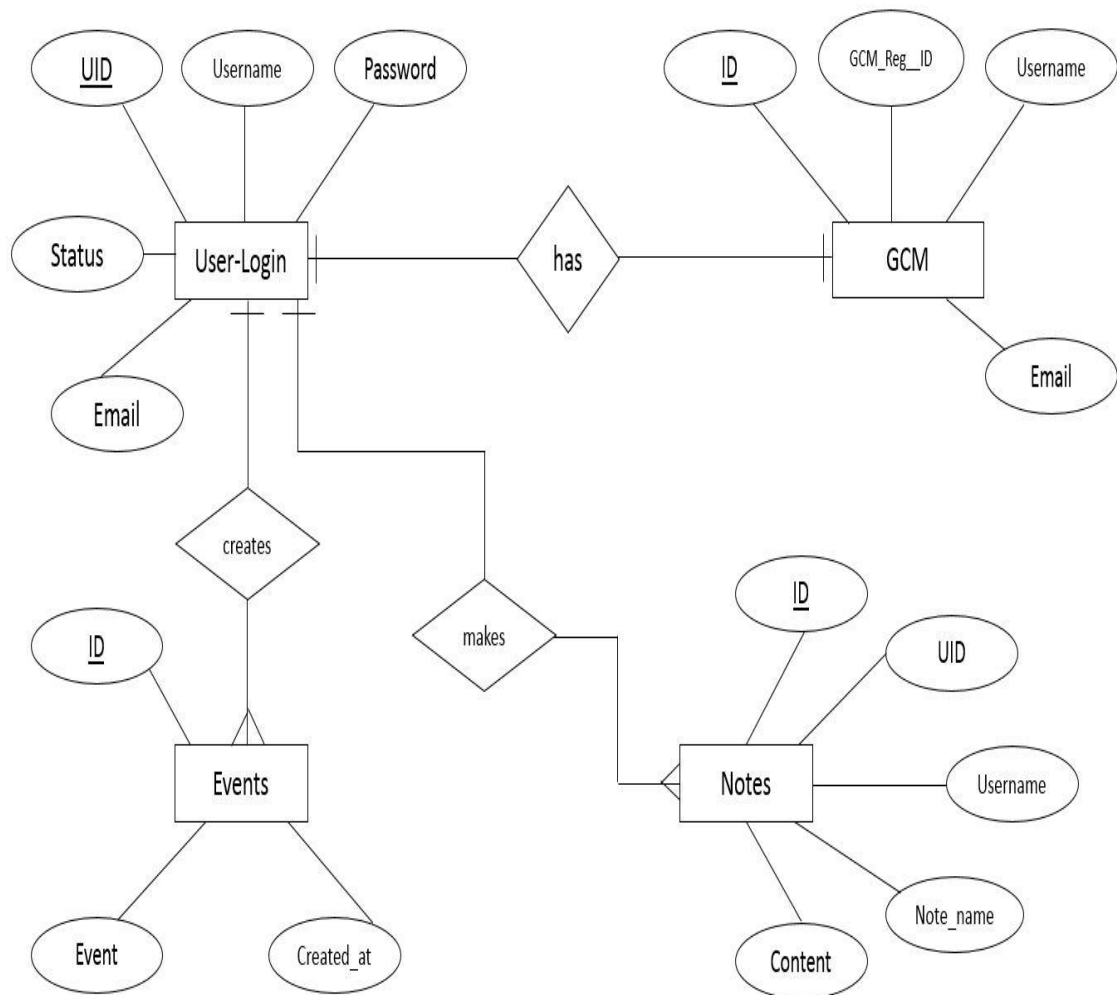


Fig 4.7: Entity Relationship Diagram

4.4 USER INTERFACE

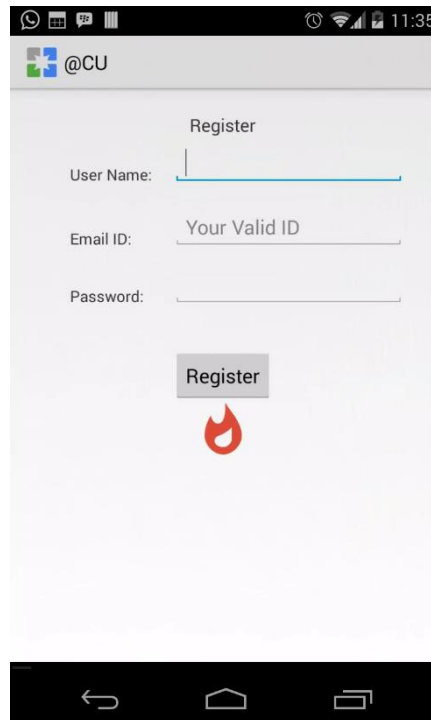


Fig 4.8: Login Screen

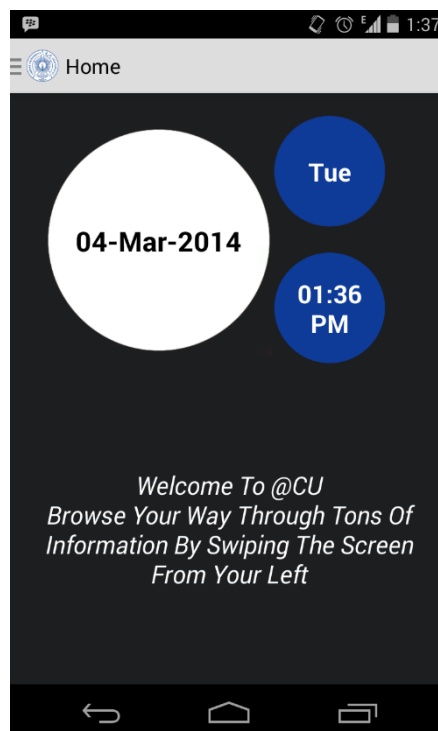


Fig 4.9: Home Screen

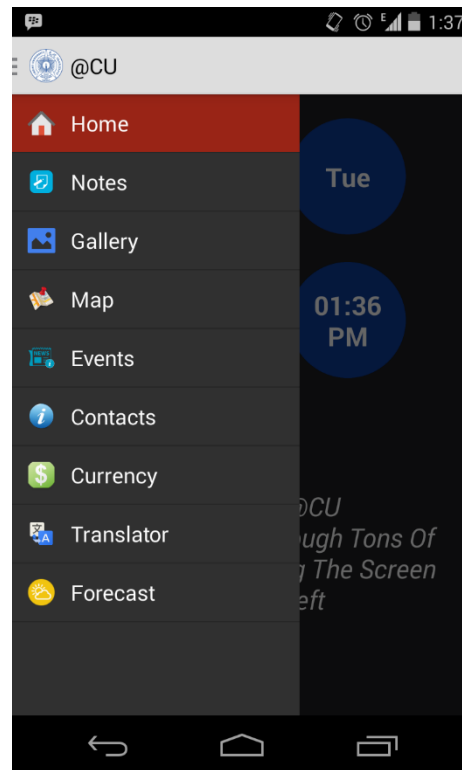


Fig 4.10: Dashboard

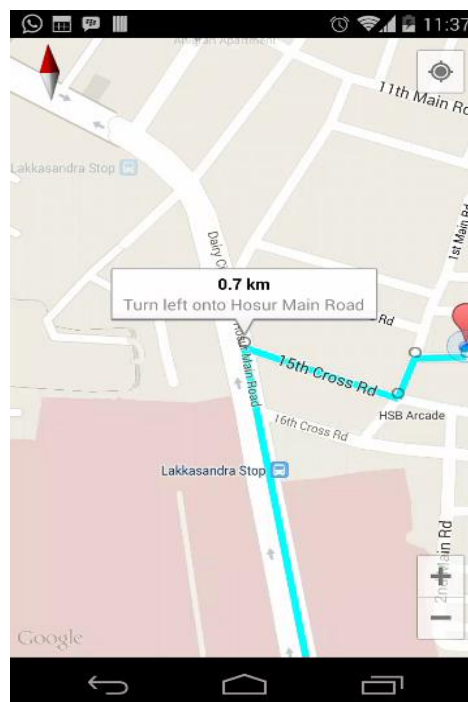
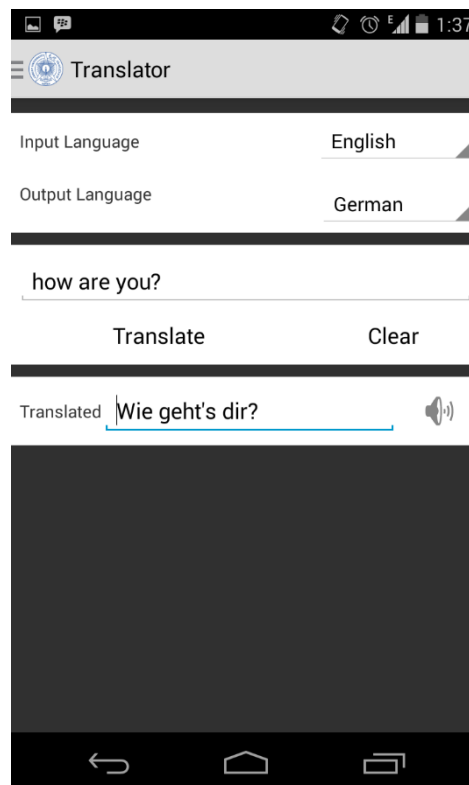
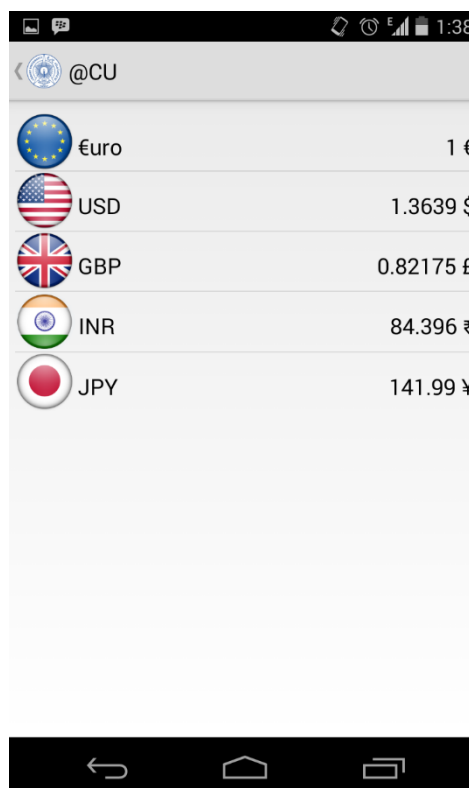


Fig 4.11: Maps

**Fig 4.12: Translator****Fig 4.13: Currency Calculator**

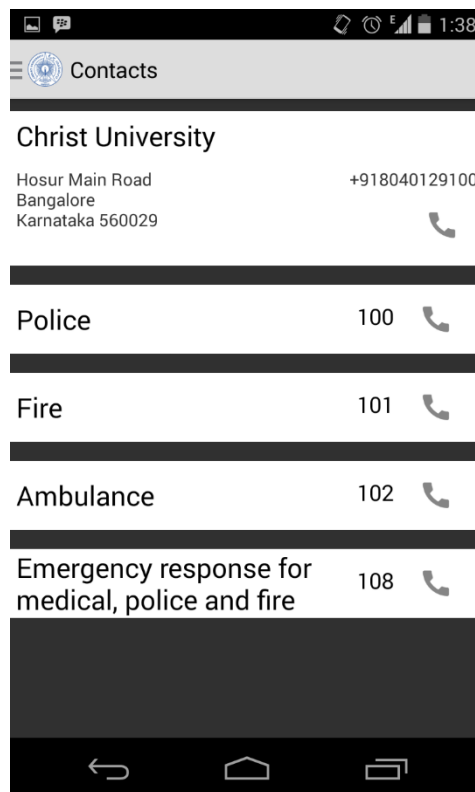


Fig 4.14: Emergency Contacts

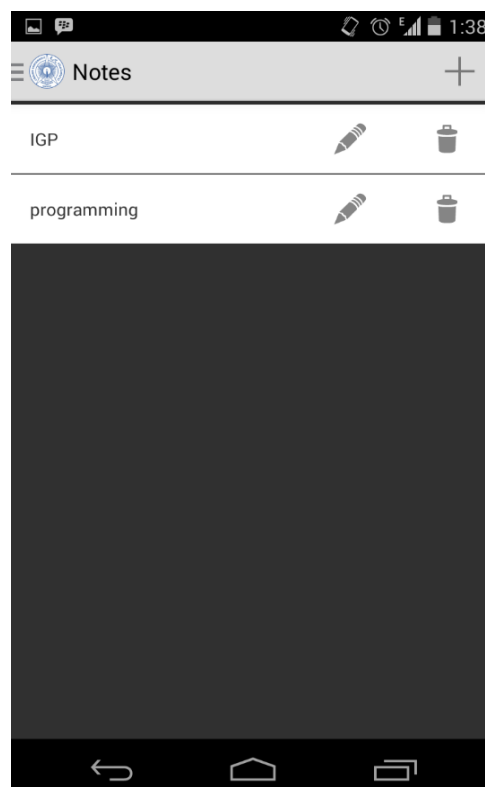


Fig 4.15: Notes

5. IMPLEMENTATION

LAYOUTS (XML)

Register.xml

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout_width="match_parent"

    android:layout_height="match_parent" >

    <TextView

        android:id="@+id/day"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_alignParentTop="true"

        android:layout_marginRight="34dp"

        android:layout_marginTop="68dp"

        android:layout_toLeftOf="@+id/txtLabel"

        android:text="User Name:" />

    <TextView

        android:id="@+id/txtLabel"

        android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"

android:layout_alignParentTop="true"

android:layout_centerHorizontal="true"

android:layout_marginTop="22dp"

android:text="Register"

android:textSize="16sp" />
```

<TextView

```
android:id="@+id/textView2"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_alignLeft="@+id/textView3"

android:layout_below="@+id/textView3"

android:layout_marginTop="33dp"

android:text="Password:" />
```

<TextView

```
android:id="@+id/textView3"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_alignLeft="@+id/day"

android:layout_below="@+id/editText1"

android:layout_marginTop="39dp"

android:text="Email ID:" />
```

<Button

```
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:onClick="register"
    android:text="Register" />
```

<ImageView

```
    android:id="@+id/imageView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/button1"
    android:layout_centerHorizontal="true"
    android:src="@drawable/ic_whats_hot" />
```

<EditText

```
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/day"
    android:layout_alignLeft="@+id/editText3"
```

```
android:ems="10"

android:singleLine="true"/>
```

```
<EditText
```

```
    android:id="@+id/editText3"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignBottom="@+id/textView3"

    android:layout_alignLeft="@+id/button1"

    android:ems="10"

    android:hint="Your Valid ID"

    android:singleLine="true">
```

```
    <requestFocus />
```

```
</EditText>
```

```
<EditText
```

```
    android:id="@+id/editText2"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_alignBottom="@+id/textView2"

    android:layout_alignLeft="@+id/editText3"

    android:layout_alignRight="@+id/editText3"

    android:ems="10"
```

```
    android:singleLine="true"
```

```
    android:inputType="textPassword" />
```

```
<requestFocus
```

```
    android:layout_alignTop="@+id/editText1"
```

```
    android:layout_centerHorizontal="true" />
```

```
</RelativeLayout>
```

SCRIPTING (PHP)

Register_New.php

```
<?php
```

```
// array for JSON response
```

```
$response = array();
```

```
// check for required fields
```

```
if (isset($_POST['userid']) && isset($_POST['email']) && isset($_POST['pass']) &&
isset($_POST['uid']) && isset($_POST['stat'])) {
```

```
    $uid = $_POST['uid'];
```

```
    $userid = $_POST['userid'];
```

```
    $pass = $_POST['pass'];
```

```
    $pass = md5($pass);
```

```
    $email = $_POST['email'];
```

```
    $stat= $_POST['stat'];
```

```
// include db connect class
```

```
require_once __DIR__ . '/db_connect.php';
```

```
// connecting to db
```

```
$db = new DB_CONNECT();
```

```
// mysql inserting a new row
```

```
$result = mysql_query("INSERT INTO Login(uid,email,userid, pass,stat)
VALUES('$uid','$email','$userid','$pass','$stat')");

// check if row inserted or not

if ($result) {

    // successfully inserted into database

    $response["success"] = 1;

    $response["message"] = "User successfully registered.";

    // echoing JSON response

    echo json_encode($response);

//Email

$mailfrom="mail@igp.com";

$mailbcc="akshay.sadarangani@gmail.com";           //To be removed
in release version

$message = "<b>Dear $userid,</b><p>You have successfully registered for
@CU.</p>\r\n<p>Your Login Credentials
are:</p>\r\n<p><b>UserID:</b>$userid</p>\r\n<p><b>Password:</b>$pass</p>\r\n
<p><a href='http://aki.bugs3.com/Aki/activation.php?uid=$uid&eid=$email'>Click
here to confirm your account.</a></p>\r\n\r\n\r\nDo Not Reply To This Mail";

mail($email, "Welcome To @CU", $message, "From: $mailfrom\r\nContent-type:
text/html\r\n");

} else {

    // failed to insert row

    $response["success"] = 0;
```

```
$response["message"] = "Oops! An error occurred.";

    // echoing JSON response
    echo json_encode($response);
}

} else {

    // required field is missing
    $response["success"] = 0;
    $response["message"] = "Required field(s) is missing";

    // echoing JSON response
    echo json_encode($response);
}

?>
```


PROGRAMMING (JAVA)

Register.java

```
package com.akisoft.slidingmenu;
```

```
//Import Statements
```

```
public class Register extends Activity {           //Class Declaration
```

```
    private ProgressDialog pDialog;
```

```
    JSONParser jsonParser = new JSONParser();      //JSONParser Object
```

```
    EditText text1;                                //XML Element Objects
```

```
    EditText text2;
```

```
    EditText text3;
```

```
    // url to create new product
```

```
    private static String url_register_new =  
    "http://igp.com/Aki/register_new.php";
```

```
    // JSON Node names
```

```
    private static final String TAG_SUCCESS = "success";
```

```
    public Register(){ }
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

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

```
text1=(EditText) findViewById(R.id.editText1);

text2=(EditText) findViewById(R.id.editText2);

text3=(EditText) findViewById(R.id.editText3);

Button reg=(Button) findViewById(R.id.button1);

reg.setOnClickListener(new View.OnClickListener() {           //Click
Event

    @Override

    public void onClick(View view) {

        InputMethodManager inputManager = (InputMethodManager)

        getSystemService(Context.INPUT_METHOD_SERVICE);

        inputManager.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(),

        InputMethodManager.HIDE_NOT_ALWAYS);           //Hide
Keyboard

        if(!text1.getText().toString().matches("") &&
text2.getText().toString().matches("") && text3.getText().toString().indexOf("@")>0
&&text3.getText().toString().indexOf(".")>0)

        {

            boolean conn= new registerNew().isConnected();

            if(conn==true)

                new registerNew().execute();

            else

                {
```

```
        Toast.makeText(getApplicationContext(), "No Internet
Connection!", Toast.LENGTH_SHORT).show();

    }

}

else

    Toast.makeText(getApplicationContext(), "Please Fill In All The Details
And Check Your Email ID", Toast.LENGTH_SHORT).show();

}

});

}
```

```
class registerNew extends AsyncTask<String, String, String> {    //Background
Task
```

```
    @Override
```

```
    protected String doInBackground(String... args) {
```

```
        String userid=text1.getText().toString();
```

```
        String pass=text2.getText().toString();
```

```
        String email=text3.getText().toString();
```

```
        TelephonyManager tm;

tm=(TelephonyManager)getSystemService(Context.TELEPHONY_SERVICE);

        final String uid;

        uid = tm.getDeviceId();

        //if (userid.trim().length() > 0 && pass.trim().length()>0) {

                List<NameValuePair> params = new
ArrayList<NameValuePair>();

                params.add(new BasicNameValuePair("uid",uid));

                params.add(new BasicNameValuePair("email",email));

                params.add(new BasicNameValuePair("userid",

userid));

                params.add(new BasicNameValuePair("pass", pass));

                params.add(new
BasicNameValuePair("stat","pending"));

                // getting JSON Object

                try {

                        JSONObject json =

jsonParser.makeHttpRequest(url_register_new,

                                "POST", params);

                                int success = json.getInt(TAG_SUCCESS);

                                if (success == 1) {

                                        // successfully created product
```

```
//Entry in local DB

DatabaseHandler db = new
DatabaseHandler(getApplicationContext());

// inserting new user into local database
db.insertUser(uid,userid,pass,"pending");

//Main Screen
Intent i = new Intent(Register.this,
Login.class);

startActivity(i);

finish();

}

} catch (JSONException e) {

    Toast.makeText(getApplicationContext(),
"Failed. . .", Toast.LENGTH_SHORT).show();

    e.printStackTrace();

}

return null;

}

public boolean isConnected(){

    ConnectivityManager connectivity = (ConnectivityManager)
getApplicationContext().getSystemService(Context.CONNECTIVITY_SERVICE);

    if (connectivity != null)

    {
```

```
NetworkInfo[] info = connectivity.getAllNetworkInfo();

if (info != null)

    for (int i = 0; i < info.length; i++)

        if (info[i].getState() == NetworkInfo.State.CONNECTED)

            {

                return true;

            }

    }

return false;
}

protected void onPreExecute() {                                //Progress Bar

    super.onPreExecute();

    pDialog = new ProgressDialog(Register.this);

    pDialog.setMessage("Registering..");

    pDialog.setIndeterminate(false);

    pDialog.setCancelable(true);

    pDialog.show();

}

protected void onPostExecute(String file_url) {

    // dismiss the dialog once done

    pDialog.dismiss();

}

}
```

6. TESTING

6.1 TEST PLAN

Software testing can be stated as the process of validating and verifying that a computer program/application/product:

- Meets the requirements that guided its design and development,
- Works as expected,
- Can be implemented with the same characteristics,
- Satisfies the needs of stakeholders.

Testing methods used in the development of this project are:-

- 1) Black-Box Testing
- 2) White-Box Testing

BLACK-BOX TESTING

Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings (white-box testing). This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

WHITE-BOX TESTING

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.

6.2 TEST STRATEGIES

The Syntactic Test

1. Are all mandatory text fields are entered?
2. Buttons conduct desired functions?
3. Is the data fetched correct?
4. Is the data stored correct?
5. Is there enough screen space to display all contents?
6. Is everything displayed in the correct manner?
7. Are all buttons aligned with their labels?
8. Are all labels on command buttons correct?
9. Are similar buttons named the same?
10. Does the program switch quickly between different windows?
11. Are objects in all related screens similarly placed?
12. Is there proper connectivity between the various forms?
13. Do all fields meet the description what is asked to do?

The Semantic Test

1. When users login with correct information, the user gets a dashboard screen.
2. When the users fail in the authentication process, they are shown an error message.
3. If all mandatory fields are not entered, the user gets an error message indicating him to fill up all details.
4. The user must sign out to exit the application.
5. The user can change its password by going to My Account

6.3 TEST CASES

The following test cases were used on the respective pages in order to test their behaviour under different circumstances:

Table 6.1: Test Cases

Sl No	Screen Reference	Description	Expected Result	Actual Result	Passed/Failed	Remarks
1	Registration	Incomplete user information	Toast indicating empty fields	Message box pops	Passed	
2	Registration	Complete user information	Store details in database and open Dashboard	Successfully registered and obtained Dashboard	Passed	
3	Registration	No Internet Connection	Toast indicating no internet	Toast	Passed	
4	Notes	New note	Create new note	Intent to create new note	Passed	
5	Notes	Rename note	Check for existing note names and update	Successful validation and updating	Passed	

6	Notes	Delete note	Remove note and delete contents	Successful deletion and removal of contents	Passed	
7	Map	No input for place search	Toast indicating empty field	Toast	Passed	
8	Map	No internet	Toast indicating no connection	Toast	Passed	
9	Currency	Incomplete field	Place default value - 0	Placed 0.00	Passed	
10	Currency	No Internet	Load last retrieved values	Loaded saved values	Passed	
11	Translator	Incomplete fields	Toast indicating incomplete fields	Toast	Passed	
12	Translator	No Internet	Toast indicating no connection	Toast	Passed	

6.4 TEST REPORT

The following is a report of the behaviour of the system when exposed to the above mentioned test cases:



Fig 6.1: Note Creation

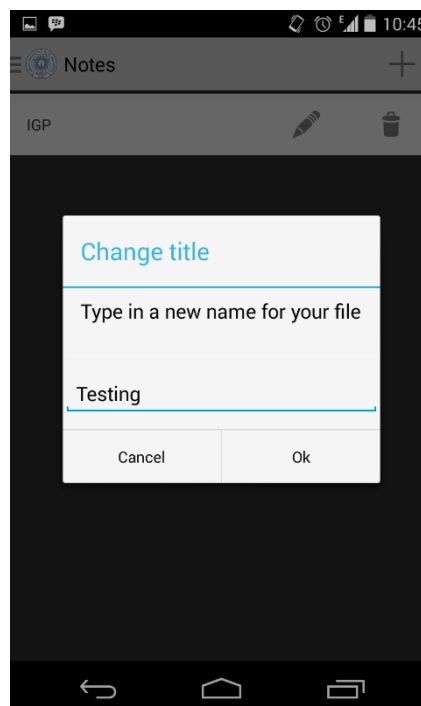


Fig 6.2: Note Rename

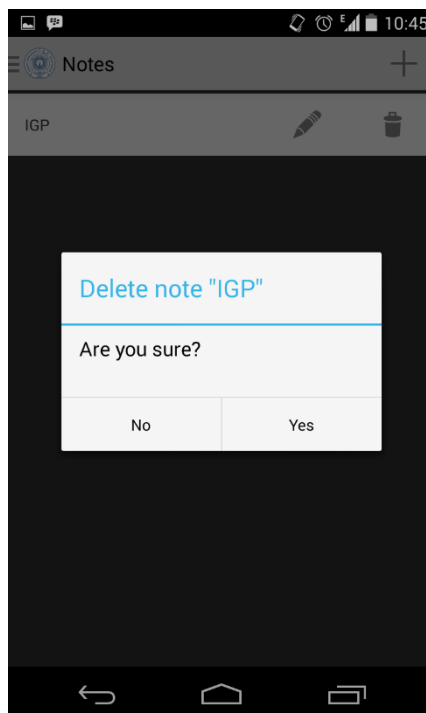


Fig 6.3: Note Deletion

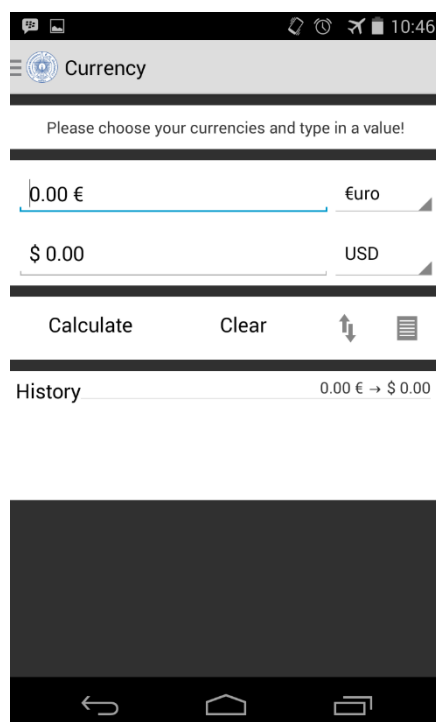


Fig 6.4: Currency Blank Field

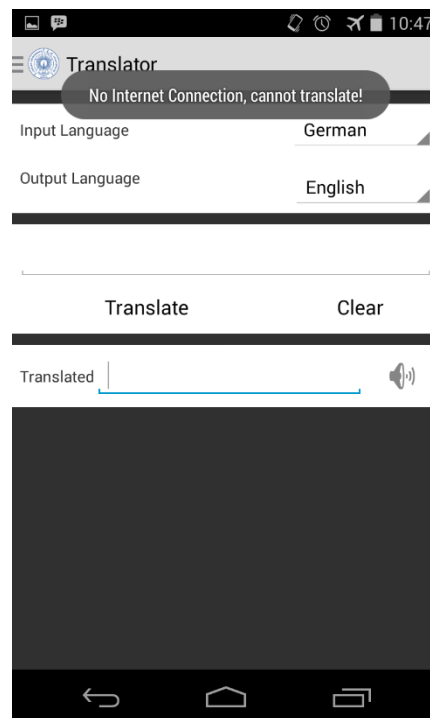


Fig 6.5: Translator- No Internet

7. CONCLUSION

@CU helps users in dealing with daily life at Christ University, Bangalore by providing a number of utility tools which can solve their problems of surviving in a foreign country. With modules such as Maps, Notes, Translator, Events, Contacts, Gallery, Currency, Weather Forecast and Scheduler, @CU is an integrated application covering a wide range of required information and services.

A one-time registration helps maintain online backups of notes and get push notifications about events happening in Christ University. A user need not open the application to browse through the events.

7.1 ADVANTAGES

@CU solves the various anomalies faced by the international students by incorporating all necessary and relevant information on a mobile application which is built on the Android platform. The Android OS is currently the leading operating system in the market with a huge population of crowd using it. Therefore the development of @CU on Android was the preferred choice and it can reach a large audience. The application is meant to aid students in travelling and exploring in-campus areas as well as areas around the campus.

The features of @CU include news and updates pertaining to the university, events which are directly synchronized with the built-in calendar and aided by push notifications, multimedia gallery for photos and videos of the university. Apart from the in-campus features, the application will also have features such as transportation finder to help find the easiest and best possible route to and from your location.

7.2 LIMITATIONS

@CU is currently available only in English which can be a problem for some international students whose first language is not English. The proposed system needs additional language support and also needs porting to other operating systems since currently it is exclusively available for the Android OS.

7.3 FUTURE ENHANCEMENTS

- Backward compatibility for phones running Android OS level below API 14 (ICS)
- Expand beyond just Android OS
- Multiple Language Support
- Food points in and around campus
- Bus routes
- Cloud backup of information

REFERENCES

- [1] “Stack Overflow” <<http://www.stackoverflow.com>>
- [2] “Android Developer” <<http://developer.android.com>>
- [3] “Android Hive” <<http://www.androidhive.com>>
- [4] “Android Example” <<http://www.androidexample.com>>
- [5] Retro Meier. <Professional Android 4 Application Development>