MULTIDISCIPLINARY PROJECT REPORT

ANDROID DEVELOPMENT

PROJECT REPORT

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

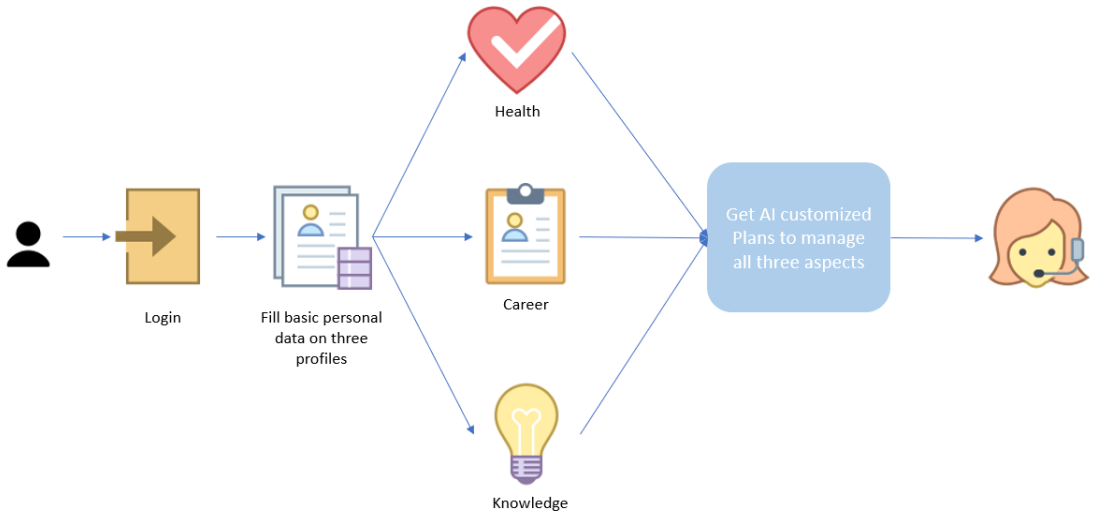
TITLE:

ANDROID APP FOR TRACKING

AND HELPING USER'S

HEALTH, CAREER AND KNOWLEDGE

ASPECTS



INTRODUCTION

Our project aims at developing an android application that aims at helping users with their Health, Career and knowledge aspects.

We focus at three basic pillars of our app:

1. Health
2. Career
3. Knowledge

In today's scenario, there is no doubt that the youth is career oriented but most of us do not know how to practically build short goals that aim at bringing us closer to our main career goal. There is a lot information on the internet but it is mostly scattered and one needs to do a lot research before actually thinking of implementing such knowledge. Our app would reduce this time. We aim at providing the user with a short-term plan for his/her career based on the details filled by the user through a very fine process.

We aim to apply the same strategy to the other two pillars of our app which are Health and knowledge.

The health segment of our app is a very basic and primitive one i.e. we aim at collecting information about user's body type, weight, height and other such aspects without going into much details to users other health issues so that we can come up with a basic plan for user's daily exercise and routine. If the user already suffers from a condition or a health issue, then we would strictly advice the user to check up with his/her doctor before taking our exercise plans. These plans include basic warming up exercises and running routines.

In the knowledge segment we would provide the user with random facts and information about the topics that the user is finds interesting.

LITERATURE SURVEY:

- INCLUDES ALREADY EXISTING APPS THAT PROVIDE HELP ON USER'S HEALTH CAREER AND KNOWLEDGE ASPECTS.
- THESE ARE INDIVIDUAL APPS
- THESE APPS DO NOT EXIST ON A COMMON PLATFORM.

What we aim to do is combine all these segments in one platform thereby giving user the best experience.

.

Conclusion

Our survey and analysis shows that there on an average 1 million to 5 million users that are using apps based on health and career but none that includes both these segments at the same platform. We aim to build such a platform that gives user opportunity to grow and learn at the same time being career oriented as well as healthy.

METHODLOGY USED

Project scope

It is a mobile app computer with a public access processing power and a limited on-chip memory. You can use it anytime, anywhere. Its compiler and IDE is built on the C/C++ framework and hence understands the same language with a little amount of electronic-specific syntaxes.

Problem identification

Most humans suffer from problems like anxiety, lost of goal, health issues etc. Amigo will keep a watch on your day to day life and will design a proper plan for you.

“Basically, we’ll design an algorithm for you!”

Proposed solution

In today’s scenario, there is no doubt that the youth is career oriented but most of us do not know how to practically build short goals that aim at bringing us closer to our main career goal. There is a lot information on the internet but it is mostly scattered and one needs to do a lot research before actually thinking of implementing such knowledge. Our app would reduce this time. We aim at providing the user with a short-term plan for his/her career based on the details filled by the user through a very fine process.

We aim to apply the same strategy to the other two pillars of our app which are Health and knowledge.

Software requirement

The Eclipse for Android Developer

Package Description

An IDE for developers creating Android applications.

This package includes:

- Eclipse Git Team Provider
- Eclipse Java Development Tools
- Maven Integration for Eclipse
- Mylyn Task List
- Code Recommenders Tools for Java Developers
- Eclipse XML Editors and Tools

2. **Android Studio**

Android Studio is the official[7] integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.[8] It is available for download on Windows, macOS and Linux based operating systems[9][10]. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

3. SQLite

SQLite is a lightweight database which comes with android. It is an Open-Source embedded SQL database engine. This provides relational database management structure for storing user defined records in the form of tables.

Key point to understand regarding SQLite: -

– SQLite is RDBMS (Relational Database Management System)
– SQLite is written in C programming language
– SQLite is embedded within the Android operating System, so you don't need anything external on Android to use SQLite
– To manipulate data (insert, update, delete) in SQLite database – we'll use SQL (Structured Query Language)

Code

The following code is divided in the complexity of program

1. For Getting data from user we have to create the database of user in the system.

```
<resources>
    <string name="app_name">Login Register</string>
    <string name="text_accounts">All Accounts</string>
    <string name="hint_name">Name</string>
    <string name="hint_email">Email</string>
    <string name="hint_password">Password</string>
    <string name="hint_confirm_password">Confirm Password</string>
    <string name="text_login">Login</string>
    <string name="text_register">Register</string>
    <string name="error_message_name">Enter Full Name</string>
    <string name="error_message_email">Enter Valid Email</string>
    <string name="error_message_age">Enter Age</string>
    <string name="error_message_password">Enter Password</string>
    <string name="success_message">Registration Successful</string>
    <string name="text_not_member">No account yet? Create one</string>
    <string name="text_already_member">Already a member? Login</string>
    <string name="error_email_exists">Email Already Exists</string>
    <string name="error_password_match">Password Does Not Matches</string>
    <string name="error_valid_email_password">Wrong Email or Password</string>
```

```
20    <string name="action_settings">Settings</string>
21    <string name="text_hello">Hello,</string>
22    <string name="text_title">Android Tutorials Hub</string>
23 </resources>
```

2. For creating user model class

```
1   package com.androidtutorialshub.loginregister.model;
2
3   public class User {
4
5       private int id;
6       private String name;
7       private String email;
8       private String password;
9
10      public int getId() {
11          return id;
12      }
13
14      public void setId(int id) {
15          this.id = id;
16      }
17
18      public String getName() {
```

```
19        return name;
20    }
21
22    public void setName(String name) {
23        this.name = name;
24    }
25
26    public String getEmail() {
27        return email;
28    }
29
30    public void setEmail(String email) {
31        this.email = email;
32    }
33
34    public String getPassword() {
35        return password;
36    }
37
38    public void setPassword(String password) {
39        this.password = password;
40    }
41 }
```

3. Creating Database helper

```
1  package com.androidtutorialshub.loginregister.sql;
2
3  import android.content.ContentValues;
4  import android.content.Context;
5  import android.database.Cursor;
6  import android.database.sqlite.SQLiteDatabase;
7  import android.database.sqlite.SQLiteOpenHelper;
8
9  import com.androidtutorialshub.loginregister.model.User;
10
11 import java.util.ArrayList;
12 import java.util.List;
13
14 public class DatabaseHelper extends SQLiteOpenHelper {
15
16     // Database Version
17     private static final int DATABASE_VERSION = 1;
18
19     // Database Name
20     private static final String DATABASE_NAME = "UserManager.db";
21
22     // User table name
```

```
23      private static final String TABLE_USER = "user";
24
25      // User Table Columns names
26      private static final String COLUMN_USER_ID = "user_id";
27      private static final String COLUMN_USER_NAME = "user_name";
28      private static final String COLUMN_USER_EMAIL = "user_email";
29      private static final String COLUMN_USER_PASSWORD = "user_password";
30
31      // create table sql query
32      private String CREATE_USER_TABLE = "CREATE TABLE " + TABLE_USER + "("
33              + COLUMN_USER_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," +
34  COLUMN_USER_NAME + " TEXT,"
35              + COLUMN_USER_EMAIL + " TEXT," + COLUMN_USER_PASSWORD + " TEXT" + ")";
36
37      // drop table sql query
38      private String DROP_USER_TABLE = "DROP TABLE IF EXISTS " + TABLE_USER;
39
40      /**
41       * Constructor
42       *
43       * @param context
44       */
45      public DatabaseHelper(Context context) {
```

```
46        super(context, DATABASE_NAME, null, DATABASE_VERSION);
47    }
48
49    @Override
50    public void onCreate(SQLiteDatabase db) {
51        db.execSQL(CREATE_USER_TABLE);
52    }
53
54
55    @Override
56    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
57
58        //Drop User Table if exist
59        db.execSQL(DROP_USER_TABLE);
60
61        // Create tables again
62        onCreate(db);
63
64    }
65
66    /**
67     * This method is to create user record
68     *
```

```
69      * @param user
70      */
71     public void addUser(User user) {
72         SQLiteDatabase db = this.getWritableDatabase();
73
74         ContentValues values = new ContentValues();
75         values.put(COLUMN_USER_NAME, user.getName());
76         values.put(COLUMN_USER_EMAIL, user.getEmail());
77         values.put(COLUMN_USER_PASSWORD, user.getPassword());
78
79         // Inserting Row
80         db.insert(TABLE_USER, null, values);
81         db.close();
82     }
83
84     /**
85      * This method is to fetch all user and return the list of user records
86      *
87      * @return list
88      */
89     public List<User> getAllUser() {
90         // array of columns to fetch
91         String[] columns = {
```

```
92              COLUMN_USER_ID,
93              COLUMN_USER_EMAIL,
94              COLUMN_USER_NAME,
95              COLUMN_USER_PASSWORD
96          };
97          // sorting orders
98          String sortOrder =
99              COLUMN_USER_NAME + " ASC";
10          List<User> userList = new ArrayList<User>();
0
10          SQLiteDatabase db = this.getReadableDatabase();
1
10          // query the user table
2           /**
10           * Here query function is used to fetch records from user table this function works like
3   we use sql query.
10           * SQL query equivalent to this query function is
4            * SELECT user_id,user_name,user_email,user_password FROM user ORDER BY
10  user_name;
5            */
10          Cursor cursor = db.query(TABLE_USER, //Table to query
6               columns,    //columns to return
                null,       //columns for the WHERE clause
```

```
10              null,       //The values for the WHERE clause
7               null,      //group the rows
10              null,      //filter by row groups
8               sortOrder); //The sort order
10
9
11          // Traversing through all rows and adding to list
0           if (cursor.moveToFirst()) {
11              do {
1                   User user = new User();
11                  user.setId(Integer.parseInt(cursor.getString(cursor.getColumnIndex(COLUMN_USER
2   _ID))));
11                  user.setName(cursor.getString(cursor.getColumnIndex(COLUMN_USER_NAME)));
3                   user.setEmail(cursor.getString(cursor.getColumnIndex(COLUMN_USER_EMAIL)));
11                  user.setPassword(cursor.getString(cursor.getColumnIndex(COLUMN_USER_PASSW
4   ORD)));
11                  // Adding user record to list
5                   userList.add(user);
11              } while (cursor.moveToNext());
6           }
11          cursor.close();
7           db.close();
```

```
11          // return user list
8           return userList;
11      }
9
12      /**
0        * This method to update user record
12       *
1        * @param user
12       */
2       public void updateUser(User user) {
12          SQLiteDatabase db = this.getWritableDatabase();
3
12          ContentValues values = new ContentValues();
4           values.put(COLUMN_USER_NAME, user.getName());
12          values.put(COLUMN_USER_EMAIL, user.getEmail());
5           values.put(COLUMN_USER_PASSWORD, user.getPassword());
12
6           // updating row
12          db.update(TABLE_USER, values, COLUMN_USER_ID + " = ?",
7                   new String[]{String.valueOf(user.getId())});
12          db.close();
8       }
```

```
12      /**
9        * This method is to delete user record
13       *
0        * @param user
13       */
1       public void deleteUser(User user) {
13          SQLiteDatabase db = this.getWritableDatabase();
2           // delete user record by id
13          db.delete(TABLE_USER, COLUMN_USER_ID + " = ?",
3                   new String[]{String.valueOf(user.getId())});
13          db.close();
4       }
13
5       /**
13       * This method to check user exist or not
6        *
13       * @param email
7        * @return true/false
13       */
8       public boolean checkUser(String email) {
13
9           // array of columns to fetch
            String[] columns = {
```

```
14              COLUMN_USER_ID
0           };
14          SQLiteDatabase db = this.getReadableDatabase();
1
14          // selection criteria
2           String selection = COLUMN_USER_EMAIL + " = ?";
14
3           // selection argument
14          String[] selectionArgs = {email};
4
14          // query user table with condition
5           /**
14           * Here query function is used to fetch records from user table this function works like
6   we use sql query.
14           * SQL query equivalent to this query function is
7           * SELECT user_id FROM user WHERE user_email = 'jack@androidtutorialshub.com';
14           */
8           Cursor cursor = db.query(TABLE_USER, //Table to query
14              columns,                //columns to return
9               selection,              //columns for the WHERE clause
15              selectionArgs,          //The values for the WHERE clause
0               null,                   //group the rows
                null,                   //filter by row groups
```

```
15              null);                  //The sort order
1           int cursorCount = cursor.getCount();
15          cursor.close();
2           db.close();
15
3           if (cursorCount > 0) {
15              return true;
4           }
15
5           return false;
15      }
6
15      /**
7           * This method to check user exist or not
15          *
8           * @param email
15          * @param password
9           * @return true/false
16          */
0       public boolean checkUser(String email, String password) {
16
1           // array of columns to fetch
            String[] columns = {
```

```
16              COLUMN_USER_ID
2           };
16          SQLiteDatabase db = this.getReadableDatabase();
3           // selection criteria
16          String selection = COLUMN_USER_EMAIL + " = ?" + " AND " +
4   COLUMN_USER_PASSWORD + " = ?";
16
5           // selection arguments
16          String[] selectionArgs = {email, password};
6
16          // query user table with conditions
7           /**
16           * Here query function is used to fetch records from user table this function works like
8   we use sql query.
16           * SQL query equivalent to this query function is
9           * SELECT user_id FROM user WHERE user_email = 'jack@androidtutorialshub.com' AND
17  user_password = 'qwerty';
0           */
17          Cursor cursor = db.query(TABLE_USER, //Table to query
1               columns,                //columns to return
17              selection,              //columns for the WHERE clause
2               selectionArgs,          //The values for the WHERE clause
                null,                   //group the rows
```

```
17              null,                   //filter by row groups
3               null);                  //The sort order
17
4           int cursorCount = cursor.getCount();
17
5           cursor.close();
17          db.close();
6           if (cursorCount > 0) {
17              return true;
7           }
17
8           return false;
17      }
9   }
18
0
18
1
18
2
18
3
```

18

4

18

5

18

6

18

7

18

8

18

9

19

0

19

1

19

2

19

3

19

4

19

5

19

6

19

7

19

8

19

9

20

0

20

1

20

2

20

3

20

4

20

5

20

6

20

7

20

8

20

9

21

0

21

1

21

2

21

3

21

4

21

5

21

6

21

7

21

8

21

9

22

0

22

1

22

2

22

3

22

4

22

5

22

6

22

7

22

8

22

9

23

0

23

1

23

2

23

3

23

4

23

5

23

6

23

7

23

8

23

9

24

0

24

1

24

2

24

3

24

4

24

5

24

6

24

7

24

8

24

9

25

0

25

1

25

2

25

3

25

4

25

5

25

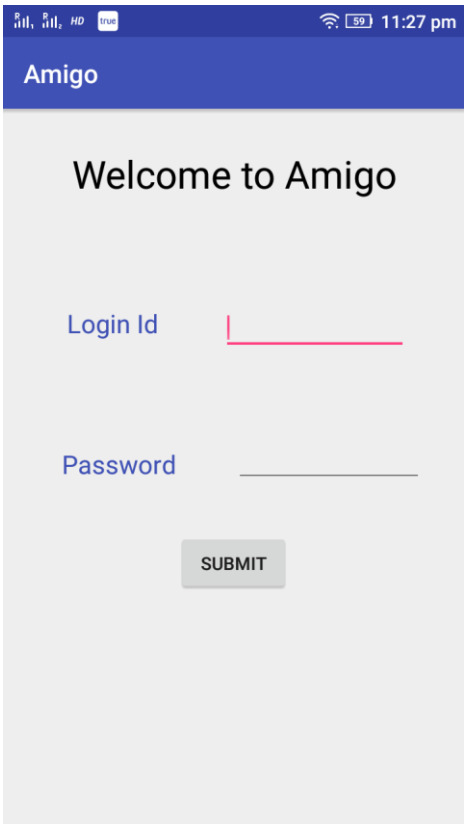
6

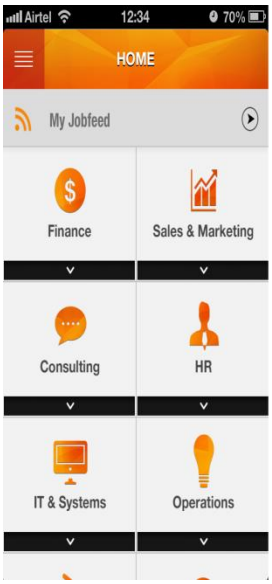
25

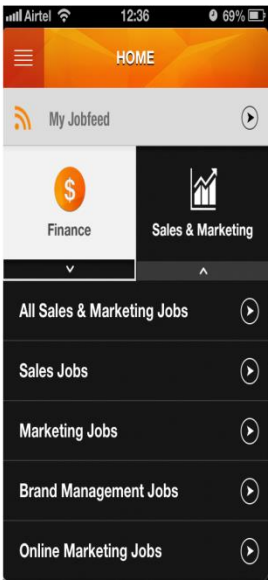
7

Results

Screenshots




Airtel
12:34
70%
HOME
My Jobfeed
Finance
Sales & Marketing
Consulting
HR
IT & Systems
Operations


Airtel
12:36
69%
HOME
My Jobfeed
Finance
Sales & Marketing
All Sales & Marketing Jobs
Sales Jobs
Marketing Jobs
Brand Management Jobs
Online Marketing Jobs

Explanation of the project

Health:-

With the help of Fitbit tracker amigo will measure user's pulse rate, and will generate the data, meanwhile user will answer another set of questions which will be related to his daily exercise and activities , our tech team will analyze the data and will make a proper diet plan and exercise plan for the user, and will give remainder every hour. At the end of every month user's activity will be again analyse and will convert it into graphs and pie charts.

Career:-

In the career part user will choose all his field of interest topics and will find suitable internships, freelance projects, workshops. Our team will analyze the progress of user and will work on his achievements.

Knowledge=

The knowledge part contains fun facts, did you know, and some quora and reddit stuff.

CONCLUSION

Our survey and analysis shows that there on an average 1 million to 5 million users that are using apps based on health and career but none that includes both these segments at the same platform. We aim to build such a platform that gives user opportunity to grow and learn at the same time being career oriented as well as healthy