MANAGE ME

Project Report

Submitted By:

Hiren Koradiya (17162121011)

Dhruv Patel (17162121014)

Shashwat Silakari (17162121026)

Meet Vaghasia(17162121029)

In partial fulfilment for the Application Development Project (SEM-IV)

Of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINERRING (BDA)

at

GANPAT UNIVERSITY
॥ विद्यया समाजोत्कर्षः ॥

INSTITUTE OF COMPUTER TECHNOLOGY

2018-2019

Table of Contents

DECLARATION

I hereby declare that the project entitled "**MANAGE ME**" submitted for the B. Tech. CSE (BDA) Application Development Project is my original work.

Signature of the Student

Place:

Date:

CERTIFICATE

This is to certify that the project titled “**MANAGE ME**” submitted by **Hiren Koradiya (171621210)** of B.Tech (CSE)-SEM IV of BDA from Institute of Computer Technology, Ganpat University during the academic year 2018-19, in partial fulfilment of the requirements for the Application Development project work.

Signature of the Guide

Place:

Date:

CERTIFICATE

This is to certify that the project titled "**MANAGE ME** " submitted by **Dhruv Patel (17162121014)** of B.Tech (CSE)-SEM IV of BDA from Institute of Computer Technology, Ganpat University during the academic year 2018-19, in partial fulfilment of the requirements for the Application Development project work.

Signature of the Guide

Place:

Date:

CERTIFICATE

This is to certify that the project titled "**MANAGE ME** " submitted by **SHASHWAT SILAKRI (17162121026)** of B.Tech (CSE)-SEM IV of BDA from Institute of Computer Technology, Ganpat University during the academic year 2018-19, in partial fulfilment of the requirements for the Application Development project work.

Signature of the Guide

Place:

Date:

Annexure-3

CERTIFICATE

This is to certify that the project titled “**MANAGE ME** ” submitted by **Meet Vaghasia (17162121029)**of B Tech (CSE)-SEM IV of CBA/BDA/MA from Institute of Computer Technology, Ganpat University during the academic year 2018-19, in partial fulfilment of the requirements for the Application Development project work.

Signature of the Guide

Place:

Date:

ABSTRACT:

MANAGE ME:

Manage Me is a personal management application which will help user to manage and perform tasks like Money Management and maintaining a To-Do List. We thought of making this app to make peoples life a little bit easy than before, people can Manage their money by keeping a record where they have spent their money, by seeing the records they can even analyse where they could reduce their spending and increase their savings.

To-Do list can help to remind them which tasks are pending. In todays hectic schedule people often forgets which task they need to-do, by adding their tasks to this app, if the task gets out of their mind they can refer to this app which would help them to manage their tasks easily.

Acknowledgement:

We would like to express our special thanks of gratitude to our Application Development guides Prof. Nidhi Thacker for their able guidance and support in completing our project.

We would also like to extend our gratitude to our IBM guide Prof. Diksha Pandit for providing us with all the extras and feature additions into our project.

1. Introduction:

1.1 Problem Definition:

Our project's aim is to address the problem faced by people in day-to-day life of managing their money, as well as their day-to-day tasks. It overall reduces some calculation required and time spend to calculate the Money spent in the whole day. As well as never forget your tasks to perform.

1.2 Project Overview/Specifications:

Our project was designed keeping in mind that it can be used by all types of people, i.e. those who are not much familiar to phones can also easily understand the functionality and get used to it very easily. Our App have the following specifications:

- Login Feature
- Signup Feature
- Choice Between Money Management & To-Do List.
- Data is stored in Firebase Database

Being revolving these features, the Android App has been beautifully contoured for simple and intuitive navigation features so that each user can browse it with ease. The application is web-based, it can bee remotely accessed from anywhere on Android device

1.3 Hardware Specification:

The Android App requires the following set of minimum hardware requirements in order to function properly:

- 1GHz ARM/x64/x86 based processor
- Min. 50 MB RAM.
- Max 100 MB ROM.

1.4 Software Specification:

The Android App requires the following set of minimum software requirements in order to function properly:

- Any Android operating system above 6.0 i.e. Marshmallow or any updated version of it.

2. Requirement Analysis:

In order to make this project happen, we made several efforts to get some necessary elements pre-hand before commencing. Following list is the set of requirements we deemed for the creation the Android App under this project:

3. System Analysis & Design:

3.1 Requirement Specification:

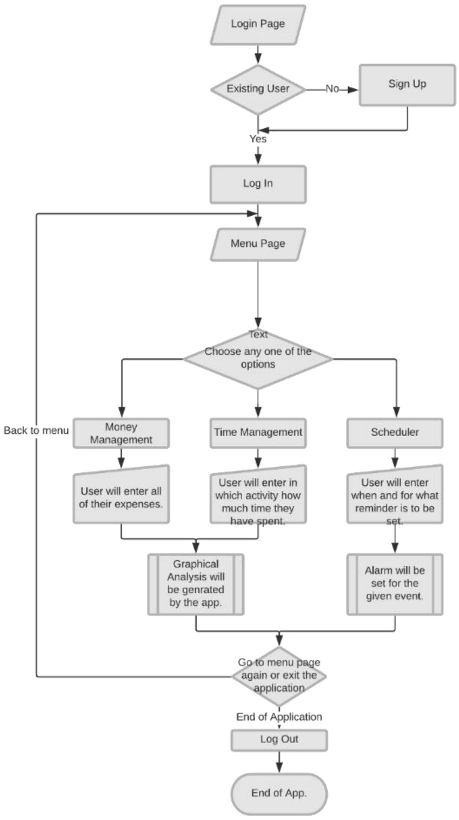
Our system should have following Specification:

- Microsoft® Windows® 7/8/10 (32- or 64-bit)/Linux/Mac OS
- 8GB RAM
- 10GB HDD
- Android Studio
- JDK 8

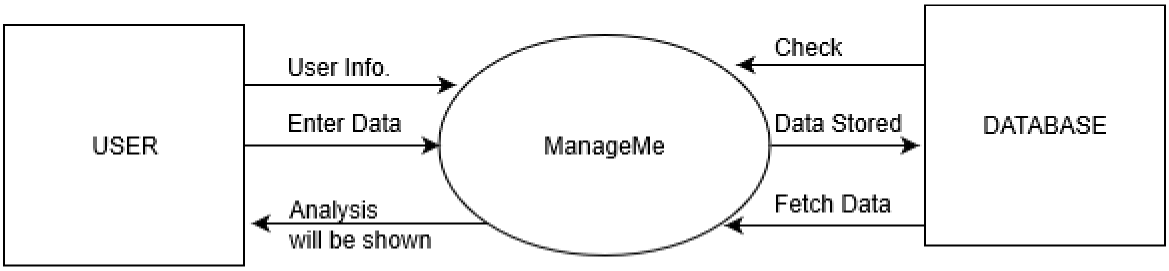
And a Configured account of Firebase for Data Storing and Retrieving purpose.

3.2 Flowcharts /DFDs

3.2.1 Flowchart:


Login Page
Existing User
No
Sign Up
Yes
Log In
Menu Page
Text
Choose any one of the options
Back to menu
Money Management
Time Management
Scheduler
User will enter all of their expenses.
User will enter in which activity how much time they have spent.
User will enter when and for what reminder is to be set.
Graphical Analysis will be genrated by the app.
Alarm will be set for the given event.
Go to menu page again or exit the application
End of Application
Log Out
End of App.

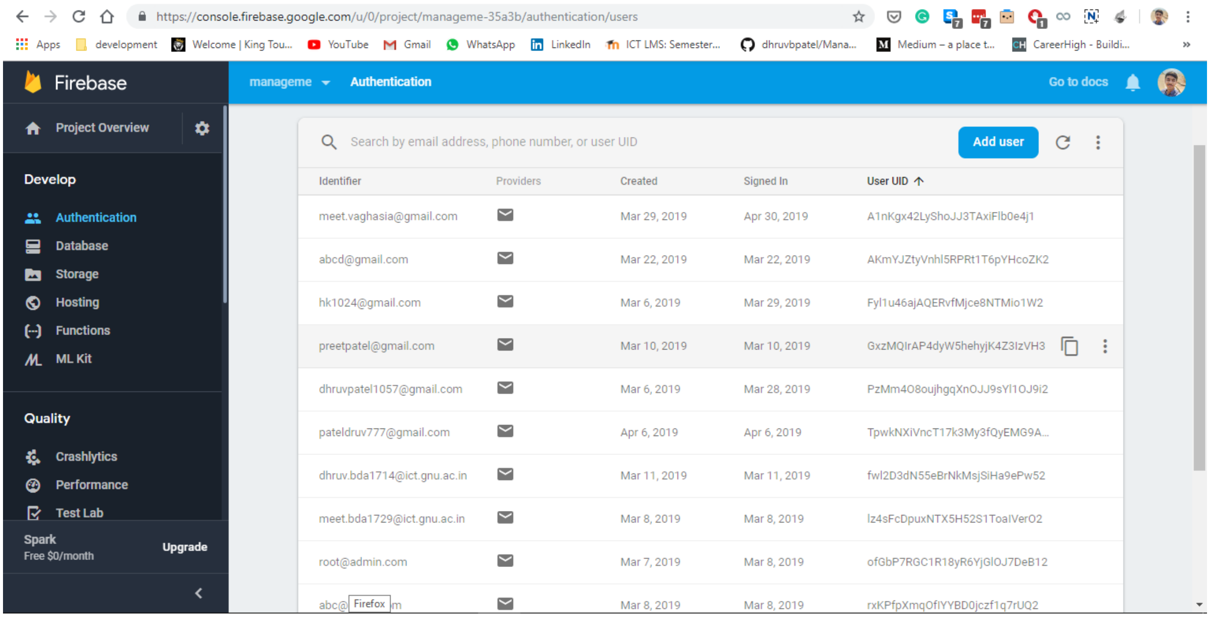
3.2.2 DFD0



3.3 Design and Test Steps / Criteria

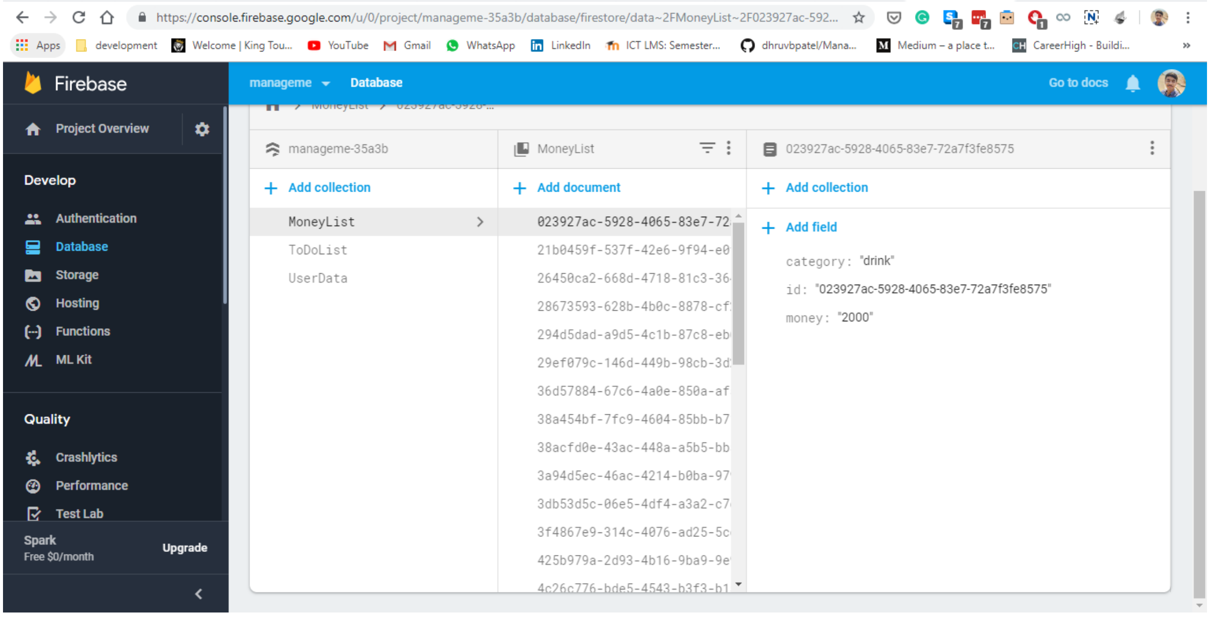
3.3.1 Firebase Database

Users Information Database:



Identifier	Providers	Created	Signed In	User UID
meet.vaghasia@gmail.com	✉	Mar 29, 2019	Apr 30, 2019	A1nKgx42LyShoJJ3TAxiFlb0e4j1
abcd@gmail.com	✉	Mar 22, 2019	Mar 22, 2019	AKmYJZtyVnhl5RPRt1T6pYHcoZK2
hk1024@gmail.com	✉	Mar 6, 2019	Mar 29, 2019	Fyl1u46ajAQERvfMjce8NTMio1W2
preetpatel@gmail.com	✉	Mar 10, 2019	Mar 10, 2019	GxzMQIrAP4dyW5hehyjK4Z3IzVH3
dhruvpatel1057@gmail.com	✉	Mar 6, 2019	Mar 28, 2019	PzMm4O8oujhgqXnOJJ9sYl1OJ9i2
pateldruv777@gmail.com	✉	Apr 6, 2019	Apr 6, 2019	TpwkNXiVncT17k3My3fQyEMG9A...
dhruv.bda1714@ict.gnu.ac.in	✉	Mar 11, 2019	Mar 11, 2019	fwl2D3dN55eBrNkMsjSiHa9ePw52
meet.bda1729@ict.gnu.ac.in	✉	Mar 8, 2019	Mar 8, 2019	lz4sFcDpuxNTX5H52S1ToaIVerO2
root@admin.com	✉	Mar 7, 2019	Mar 8, 2019	ofGbP7RGC1R18yR6YjGlOJ7DeB12
abc@ m	✉	Mar 8, 2019	Mar 8, 2019	rxKPfpXmqOfIYYBD0jczf1q7rUQ2

Money Management Module Database:

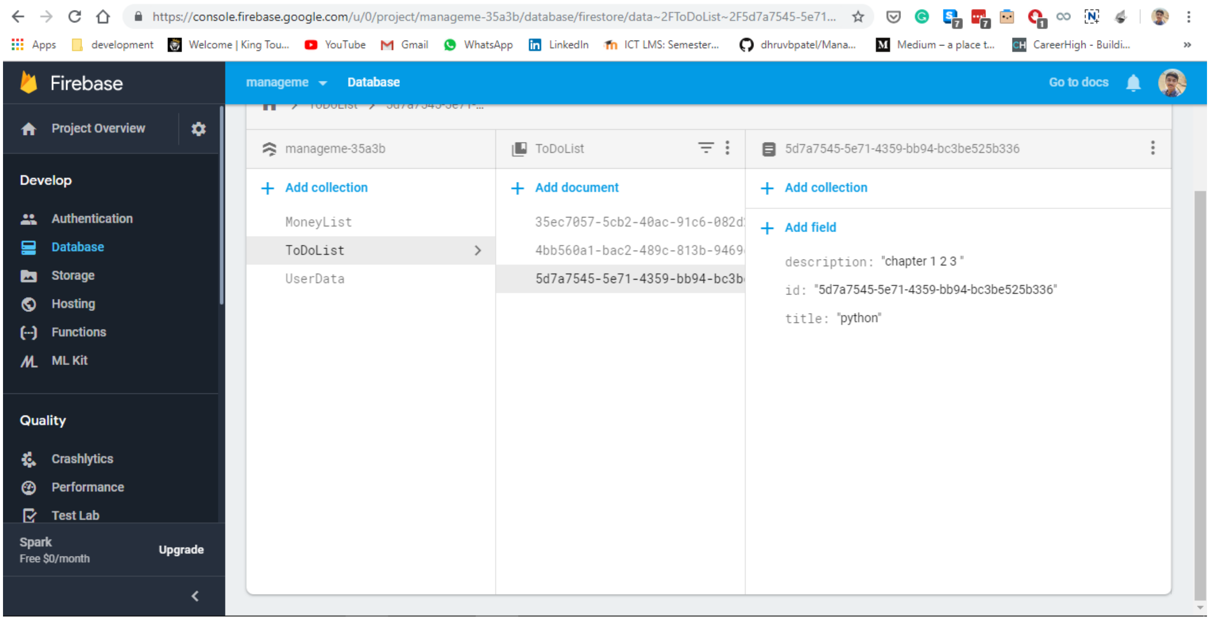


MoneyList
ToDoList
UserData

023927ac-5928-4065-83e7-72
21b0459f-537f-42e6-9f94-e0
26450ca2-668d-4718-81c3-36
28673593-628b-4b0c-8878-cf
294d5dad-a9d5-4c1b-87c8-eb
29ef079c-146d-449b-98cb-3d
36d57884-67c6-4a0e-850a-af
38a454bf-7fc9-4604-85bb-b7
38acfd0e-43ac-448a-a5b5-bb
3a94d5ec-46ac-4214-b0ba-97
3db53d5c-06e5-4df4-a3a2-c7
3f4867e9-314c-4076-ad25-5c
425b979a-2d93-4b16-9ba9-9e
4c26c776-bde5-4543-b3f3-b1

category: "drink"
id: "023927ac-5928-4065-83e7-72a7f3fe8575"
money: "2000"

To-Do List Database:



MoneyList
ToDoList
UserData

35ec7057-5cb2-40ac-91c6-082d
4bb560a1-bac2-489c-813b-9469
5d7a7545-5e71-4359-bb94-bc3b

description: "chapter 1 2 3 "
id: "5d7a7545-5e71-4359-bb94-bc3be525b336"
title: "python"

3.4 Algorithms and Pseudo Code

1.) LoginActivity:

```
  @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        userMail = findViewById(R.id.login_mail);
        userPassword = findViewById(R.id.login_password);
        btnLogin = findViewById(R.id.loginBtn);
        loginProgress = findViewById(R.id.login_progress);
        mAuth = FirebaseAuth.getInstance();
        HomeActivity = new Intent(this,HomePage.class);
        loginPhoto = findViewById(R.id.login_photo);
        loginPhoto.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent registerActivity = new
Intent(getApplicationContext(),RegisterActivity.class);
                startActivity(registerActivity);
                finish();


            }
        });

        loginProgress.setVisibility(View.INVISIBLE);
        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                loginProgress.setVisibility(View.VISIBLE);
                btnLogin.setVisibility(View.INVISIBLE);

                final String mail = userMail.getText().toString();
                final String password = userPassword.getText().toString();

                if (mail.isEmpty() || password.isEmpty()) {
                    showMessage("Please Verify All Field");
                    btnLogin.setVisibility(View.VISIBLE);
                    loginProgress.setVisibility(View.INVISIBLE);
                }
                else
                {
                    signIn(mail,password);
                }




            }
        });


    }

    private void signIn(String mail, String password) {


        mAuth.signInWithEmailAndPassword(mail,password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {


                if (task.isSuccessful()) {

                    loginProgress.setVisibility(View.INVISIBLE);
                    btnLogin.setVisibility(View.VISIBLE);
                    updateUI();

                }
                else {
                    showMessage(task.getException().getMessage());
                    btnLogin.setVisibility(View.VISIBLE);
                    loginProgress.setVisibility(View.INVISIBLE);
                }


            }
        });



    }

    private void updateUI() {

        startActivity(HomeActivity);
        finish();

    }

    private void showMessage(String text) {

        Toast.makeText(getApplicationContext(),text,Toast.LENGTH_LONG).show();
    }


    @Override
    protected void onStart() {
        super.onStart();
        FirebaseUser user = mAuth.getCurrentUser();

        if(user != null) {
            //user is already connected  so we need to redirect him to home page
            updateUI();

        }



    }
}
```

2.) Money Module

```
3.) import com.example.myapplication1.R;


    import dmax.dialog.*;


    public class MainMoney extends AppCompatActivity {
```

```
    List<MoneyModel> toDoList = new ArrayList<>();
    FirebaseFirestore db;


    RecyclerView listItem;
    RecyclerView.LayoutManager layoutManager;

    FloatingActionButton fab;

    public MaterialEditText category,money;

    public boolean isUpdate =false;  // flag
    public String idUpdate="";
        MoneyItemAdapter adapter;





    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_money);

        //Init firestore
        db= FirebaseFirestore.getInstance() ;

        // view
        AlertDialog dialog ;
        dialog= new SpotsDialog(this);
        category = (MaterialEditText)findViewById(R.id.category);
        money = (MaterialEditText)findViewById(R.id.money);
        fab = (FloatingActionButton)findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //add new
                if(!isUpdate)
                {

setData(category.getText().toString(),money.getText().toString());
                }
                else{

updateData(category.getText().toString(),money.getText().toString());
                    isUpdate = !isUpdate;  // reset flag
                }

            }
        });

        listItem = (RecyclerView)findViewById(R.id.listTodo);
        listItem.setHasFixedSize(true);
        layoutManager = new LinearLayoutManager(this);
        listItem.setLayoutManager(layoutManager);

        loadData();


    }

    @Override
    public boolean onContextItemSelected(MenuItem item) {
        if(item.getTitle().equals("DELETE") )
            deleteItem(item.getOrder());
        return super.onContextItemSelected(item);
    }
    private void deleteItem(int index) {
        db.collection("MoneyList")
                .document(toDoList.get(index).getId())
                .delete()
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        loadData();
                    }
                });
    }

    private void updateData(String title, String description){
        db.collection("MoneyList").document(idUpdate)
                .update("category",category,"money",money)
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        Toast.makeText(MainMoney.this,"Updated
!",Toast.LENGTH_SHORT).show();
                    }
                });

        // realtime update refresh data

        db.collection("MoneyList").document(idUpdate)
                .addSnapshotListener(new EventListener<DocumentSnapshot>() {
                    @Override
                    public void onEvent(DocumentSnapshot documentSnapshot,
FirebaseFirestoreException e) {
                        loadData();
                    }
                });



    }
 // for storing the database into  firebase firestore
    private void setData(String category, String money) {
        //random id
        String id = UUID.randomUUID().toString();
        Map<String,Object> todo = new HashMap<>();

        if(category.equals("") || money.equals(""))    // if content is
empty then it will not add
        {
            Toast.makeText(MainMoney.this,"Enter Title and
Description",Toast.LENGTH_SHORT).show();
        }
        else {
            todo.put("id", id);
            todo.put("category", category);
            todo.put("money", money);

            db.collection("MoneyList").document(id)
                    .set(todo).addOnSuccessListener(new
OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void aVoid) {
                    //refresh data
                    loadData();
                }
            });
        }
    }

    private void loadData() {
        //dialog.show();

        if(toDoList.size()>0)
            toDoList.clear();    // removing values

        db.collection("MoneyList")
                .get()
                .addOnCompleteListener(new
OnCompleteListener<QuerySnapshot>() {
                    @Override
                    public void onComplete(@NonNull Task<QuerySnapshot>
task) {
                        for (DocumentSnapshot doc : task.getResult()) {
                            MoneyModel mm1 = new
MoneyModel(doc.getString("id"),
                                    doc.getString("category"),
                                    doc.getString("money"));
                            toDoList.add(mm1);
                        }
                        adapter = new MoneyItemAdapter(MainMoney.this ,
toDoList);
                        listItem.setAdapter(adapter);
                        //   dialog.dismiss();
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {

Toast.makeText(MainMoney.this,""+e.getMessage(),Toast.LENGTH_LONG).show();
                    }
                });

    }
}
```

4. MainTodo:

```
5.      @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_todo);

        //Init firestore
        db= FirebaseFirestore.getInstance() ;

        // view
        AlertDialog dialog ;
        dialog= new SpotsDialog(this);
        title = (MaterialEditText)findViewById(R.id.title);
        description = (MaterialEditText)findViewById(R.id.description);
        fab = (FloatingActionButton)findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //add new
                if(!isUpdate)
                {

setData(title.getText().toString(),description.getText().toString());
                }
                else{
```

```
updateData(title.getText().toString(),description.getText().toString());
                    isUpdate = !isUpdate;  // reset flag
                }

            }
        });

        listItem = (RecyclerView)findViewById(R.id.listTodo);
        listItem.setHasFixedSize(true);
        layoutManager = new LinearLayoutManager(this);
        listItem.setLayoutManager(layoutManager);

        loadData();


    }

    @Override
    public boolean onContextItemSelected(MenuItem item) {
        if(item.getTitle().equals("DELETE") )
            deleteItem(item.getOrder());
        return super.onContextItemSelected(item);
    }

    private void deleteItem(int index) {
        db.collection("ToDoList")
                .document(toDoList.get(index).getId())
                .delete()
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        loadData();
                    }
                });
    }

    private void updateData(String title, String description){
        db.collection("ToDoList").document(idUpdate)
                .update("title",title,"description",description)
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {
                        Toast.makeText(MainToDo.this,"Updated
!",Toast.LENGTH_SHORT).show();
                    }
                });

        // realtime update refresh data

        db.collection("ToDoList").document(idUpdate)
                .addSnapshotListener(new
EventListener<DocumentSnapshot>() {
                    @Override
                    public void onEvent(DocumentSnapshot
documentSnapshot, FirebaseFirestoreException e) {
                        loadData();
                    }
                });



    }

    private void setData(String title, String description) {
        //radnom id
        String id = UUID.randomUUID().toString();
        Map<String,Object> todo = new HashMap<>();
        if(title.equals("") || description.equals(""))    // if content
is empty then it will not add
        {
            Toast.makeText(MainToDo.this,"Enter Title and
Description",Toast.LENGTH_SHORT).show();
        }
        else {
            todo.put("id", id);
            todo.put("title", title);
            todo.put("description", description);

            db.collection("ToDoList").document(id)
                    .set(todo).addOnSuccessListener(new
OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void aVoid) {
                    //refresh data
                    loadData();
                }
            });
        }



    }

    private void loadData() {
        //dialog.show();

        if(toDoList.size()>0)
            toDoList.clear();    // removing values

        db.collection("ToDoList")
                .get()
                .addOnCompleteListener(new
OnCompleteListener<QuerySnapshot>() {
                    @Override
                    public void onComplete(@NonNull Task<QuerySnapshot>
task) {
                        for (DocumentSnapshot doc : task.getResult()) {
                            ToDo todo = new ToDo(doc.getString("id"),
                                    doc.getString("title"),
                                    doc.getString("description"));
                            toDoList.add(todo);
                        }
                        adapter = new ListItemAdapter(MainToDo.this,
toDoList);
                        listItem.setAdapter(adapter);
                        //   dialog.dismiss();
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {

Toast.makeText(MainToDo.this,""+e.getMessage(),Toast.LENGTH_LONG).show();
                    }
                });

    }
}
```

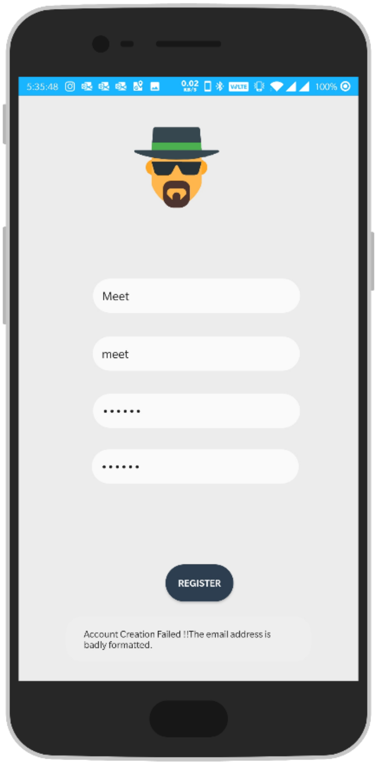
4.) Layout Design of Modules:

```
5.) <?xml version="1.0" encoding="utf-8"?>
    <android.support.design.widget.CoordinatorLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Activities.MainMoney">

    <android.support.design.widget.AppBarLayout


        android:id="@+id/appbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:fitsSystemWindows="true"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">


        <LinearLayout
            android:id="@+id/layout_info"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@color/colorPrimary"
            android:orientation="vertical"
            android:padding="16dp">

            <com.rengwuxian.materialedittext.MaterialEditText
                android:id="@+id/category"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="category"
                android:inputType="text"
                android:textColorHint="@android:color/white"
                android:textSize="30dp"
                app:met_baseColor="@android:color/white"
                app:met_floatingLabel="highlight"
                app:met_primaryColor="@android:color/white"
                app:met_singleLineEllipsis="true" />

            <com.rengwuxian.materialedittext.MaterialEditText
                android:id="@+id/money"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Money"

                android:inputType="number"
                android:textColorHint="@android:color/white"
                android:textSize="20sp"
                app:met_baseColor="@android:color/white"
                app:met_floatingLabel="highlight"
                app:met_primaryColor="@android:color/white"
                app:met_singleLineEllipsis="true" />


        </LinearLayout>


    </android.support.design.widget.AppBarLayout>

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="10dp"
        android:src="@drawable/ic_add_black_24dp"
        app:elevation="6dp"
        app:fabSize="normal"
        app:layout_anchor="@id/appbar"
        app:layout_anchorGravity="bottom|right"
        app:pressedTranslationZ="12dp" />


    <android.support.v7.widget.RecyclerView
        android:id="@+id/listTodo"
        android:layout_width="match_parent"
        android:layout_height="349dp"
        android:layout_marginTop="30dp"

        app:layout_behavior="@string/appbar_scrolling_view_behavior">

    </android.support.v7.widget.RecyclerView>


    <com.rengwuxian.materialedittext.MaterialEditText
        android:id="@+id/Total"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="650dp"
        android:hint="Total"
        android:inputType="number"
        android:text="TOTAL"
        android:textColorHint="@color/colorPrimary"
        android:textSize="20sp"
        app:met_baseColor="@color/colorPrimary"
        app:met_floatingLabel="highlight"
        app:met_primaryColor="@android:color/white"
        app:met_singleLineEllipsis="true" />


</android.support.design.widget.CoordinatorLayout>
```

3.5 Testing Process

Email ID Validation test:



Meet

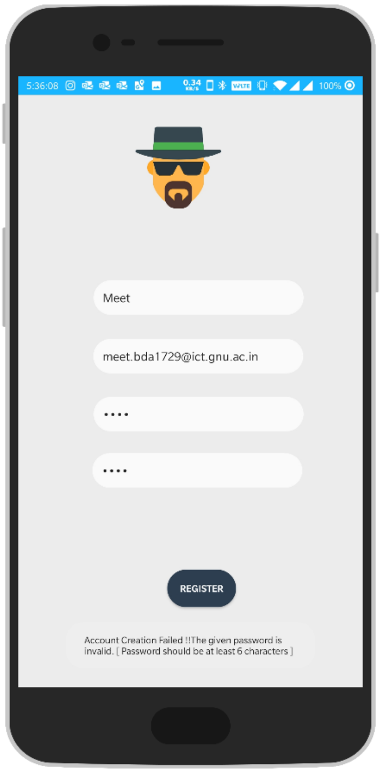
meet

••••••

••••••

Account Creation Failed !!The email address is badly formatted.

Password Validation:



Meet

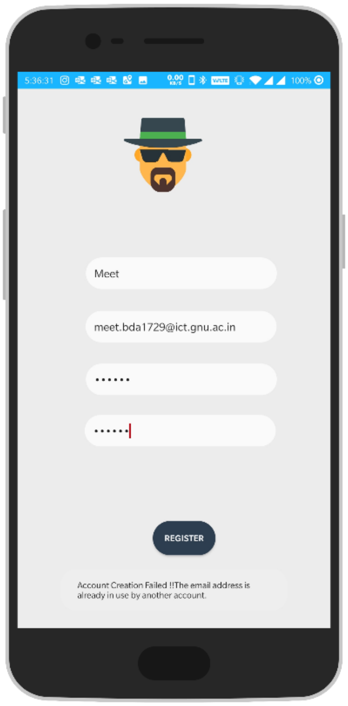
meet.bda1729@ict.gnu.ac.in

••••

••••

Account Creation Failed !!The given password is invalid. [Password should be at least 6 characters]

Account Registration Validation:



Meet

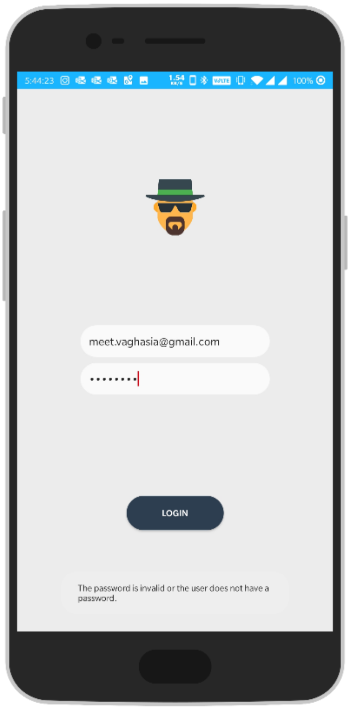
meet.bda1729@ict.gnu.ac.in

••••••

••••••

Account Creation Failed !!The email address is already in use by another account.

Login Validation:



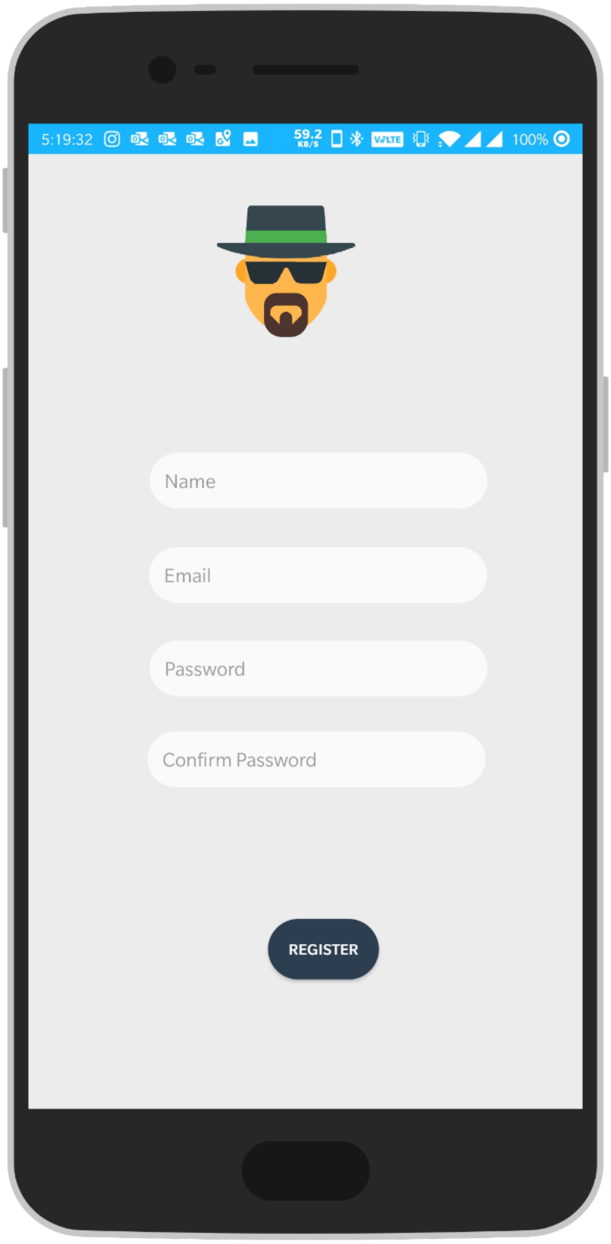
meet.vaghasia@gmail.com

•••••••••

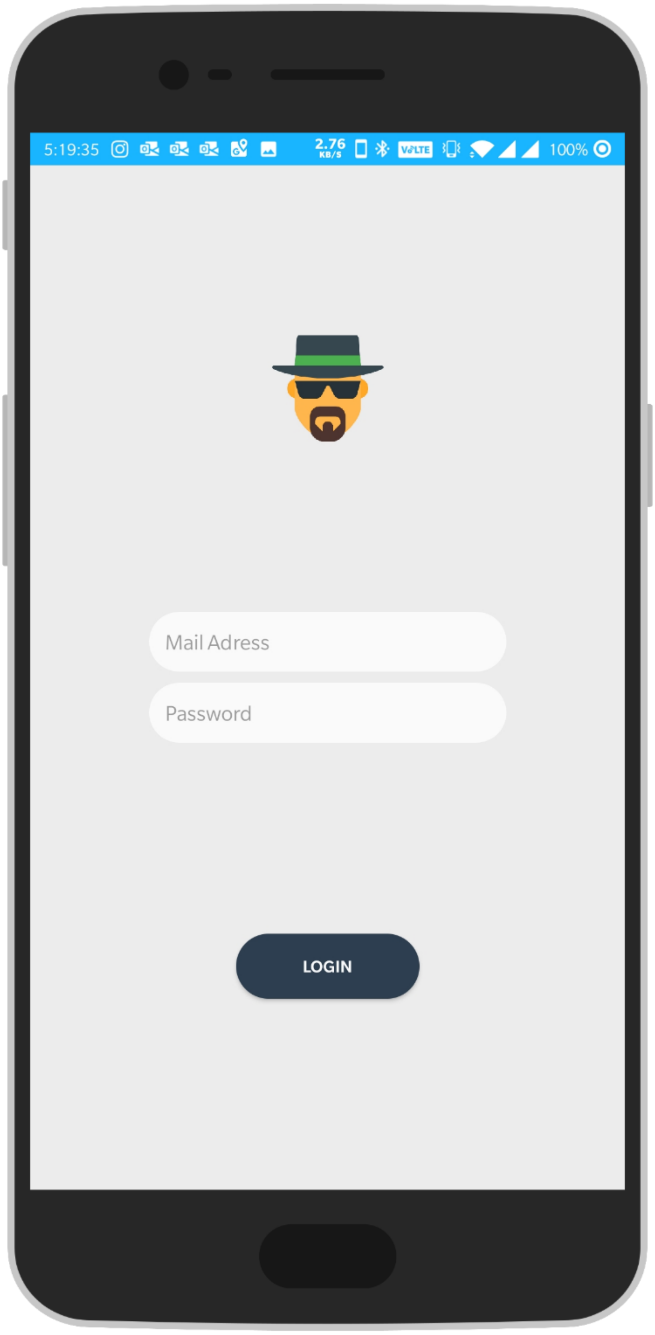
The password is invalid or the user does not have a password.

4 Results/Outputs:

Registration Page:



Login Page:



5:19:35

2.76 KB/s

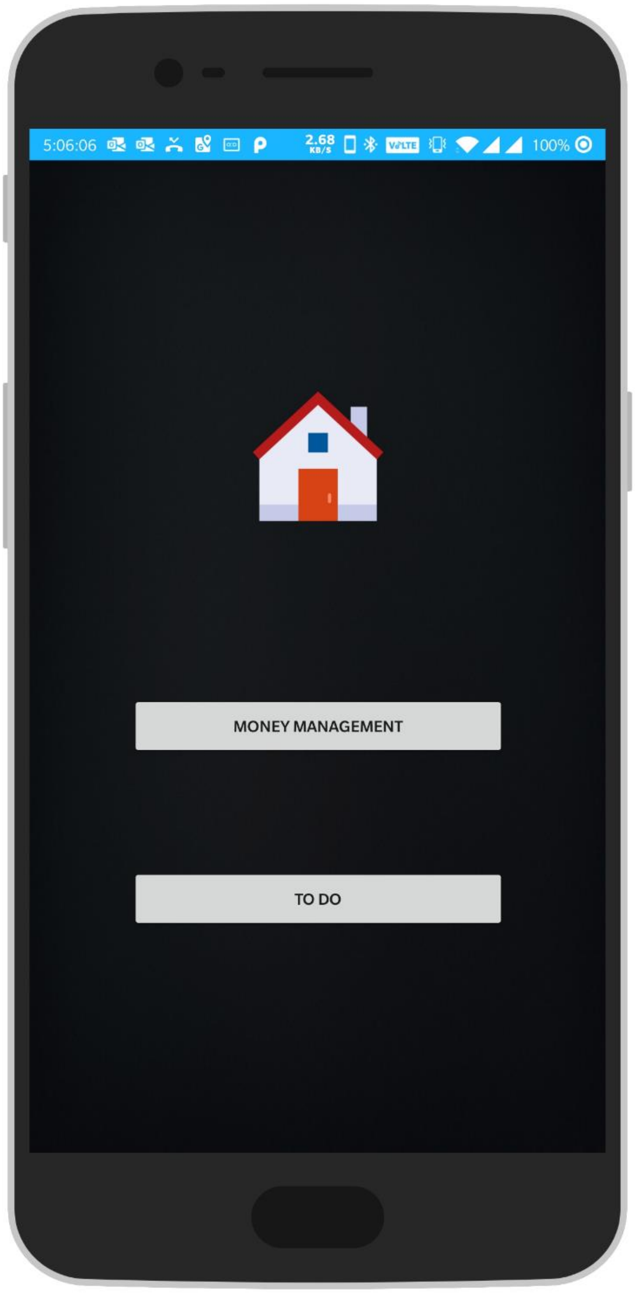
100%

Mail Adress

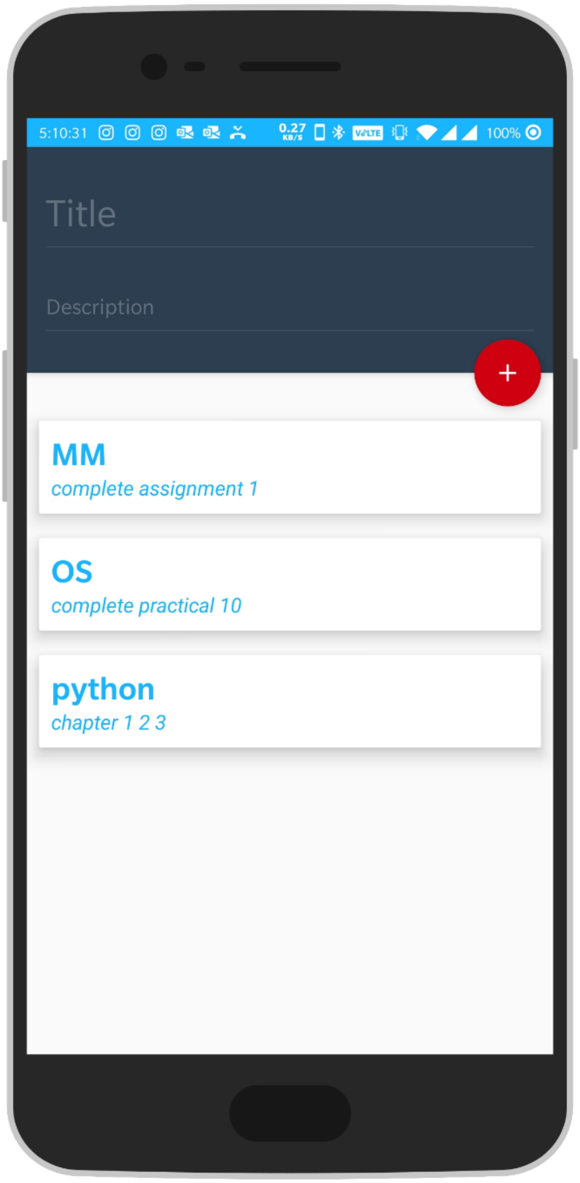
Password

LOGIN

Home Page:



To-Do List:



MM

complete assignment 1

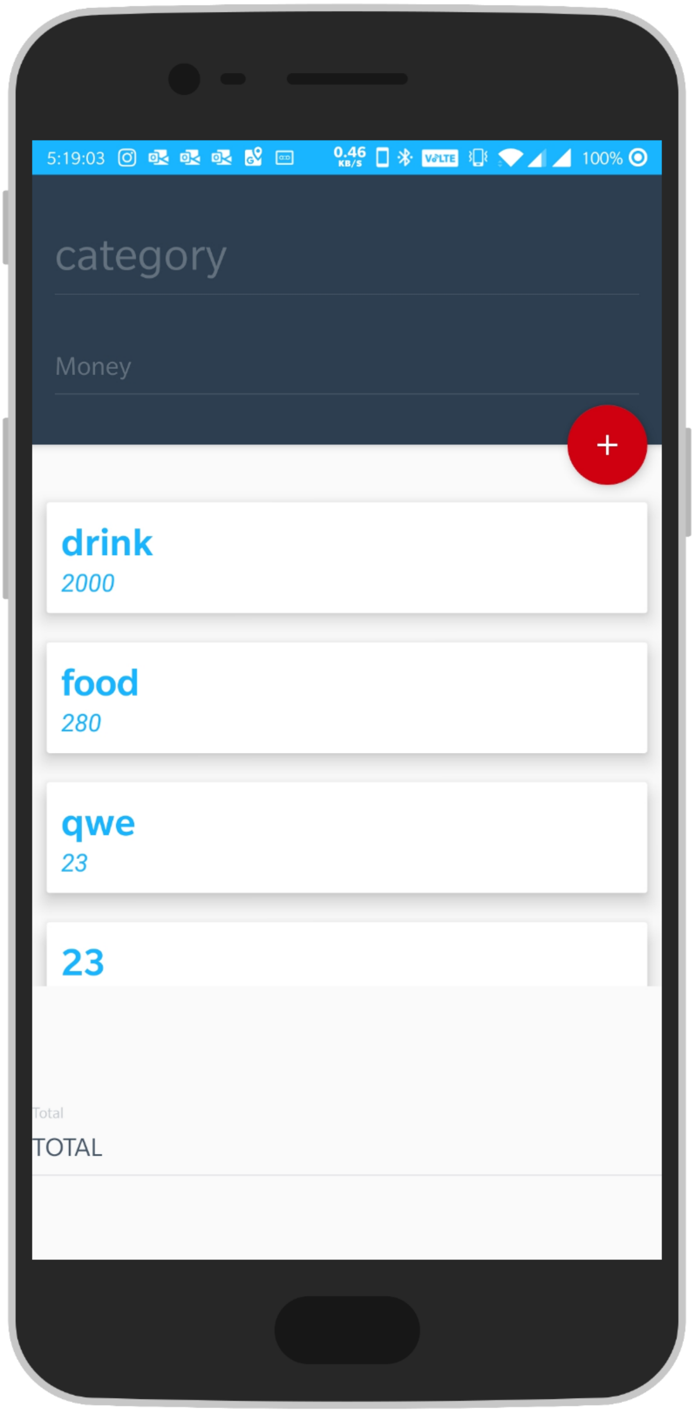
OS

complete practical 10

python

chapter 1 2 3

Money Management :



5:19:03

category

Money

+

drink
2000

food
280

qwe
23

23

Total
TOTAL

5 Conclusions:

From this Application Development Subject we learned many things

6 References:

- https://developer.android.com/docs
- https://firebase.google.com/docs

  and many other websites….