

Final Report

by Bilal Ashfaq

Submission date: 07-Jun-2020 05:41PM (UTC+0500)

Submission ID: 1338535324

File name: Android_Code.txt (171.56K)

Word count: 15365

Character count: 102174

```
Screen {

    Linear homenearby;
    Image imageneaby;
    ;

    Animation myAnim;

    recyler ;
    sAdapter sAdapter;
    . ;
    < s> s ;

    homenearby= ( .homeneaby);
    imageneaby= ( .backhomeneaby);
    = ( .progress_circular);
    . ( .);

    myAnim = AnimationUtils. Animation( (),R.anim.mytrans);
```

```
text. (str);

        42
DatabaseReference databaseReference =
    FirebaseDatabase.getInstance().getReference();

    databaseReference.child(str).ValueEventListener( ValueEventListener() {

        42
        Change( nNull DataSnapshot dataSnapshot) {

            Iterable<DataSnapshot> iterable= dataSnapshot.getChildren();

            for (DataSnapshot contact: iterable) {

                s s = contact.getValue( s. );

                s . ( s);

            }

            sAdapter. tifyDataSetChanged();

            . ( .);

        }

        ( nNull DatabaseError databaseError) {

            }

        });

    }

}
```

```
}

sInfoPre {

    recyler ;
    sAdapter sAdapter;
    . ;
    <s> s ;

Image back,img1;
onmap,Promocode;
main ,mainPrice, ress;

= get ();
Price = . ("Price");
= . (" ");
Prcode = . ("rating");
lat= . (" ");
lon= . ("longtech");
img= . ("Img");
back= ( .back info);
main = ( . );
```

```
ress= ( .ress);
mainPrice= ( .price);
img1 = ( .picture);
onmap = ( .onmap);
Promocode= ( .oder w);
.with( ()).asBitmap().(img).into(img1);

mainPrice. (Price);
main . ( );
= ( ());
{
List<ress> ressList = (List<ress>
.getFromLocation(Double.valueOf(lat),Double.valueOf(lon),1);
ress. ( ressList.get(0).get ressLine(0).to ());
Log.d("Bilaaaal", "onLocationChanged: "+lat+";"+lon);
} (exccccc e){
Log.d("Bilaaaal", " : "+e.getMessage());
e. ();
}
back. ( . () {
```

```
        onClick( ) {
            ();
        }
    });

onmap. ( . () {
    onClick( ) {
    }
});

Promocode. ( . () {
    onClick( v) {
        = ( (), MainPayment . );
        . (" ", );
        . ("promocode",Prcode);
        ( );
    }
});

}
}
```

```
}

ForgetPass extends  {

    ;

    ;

    submit;

EditText editText;

    ;

protected  (Bundle ){

    .();

(R. . _forget_pass);

submit =( ) ( .submit );

editText=(EditText) ( .Forgetpass);

submit. ( . () {

onClick( v){

    = .getInstance();

    = editText.getText().to ().trim();

    .sendPasswordReset ( ).( <>() {
```

```
( a ){  
    . (ForgetPass.this, "Link send", . ). ();  
    ( ( (), . ));  
    ();  
}  
}). ( () {  
  
( nNull excccc e){  
    . (ForgetPass.this, "Failed", . ). ();  
}  
});  
}  
});  
}  
}  
Main2 extends {
```

```
Stack<> Stack;  
Location location ;  
LocationListener locationListener;  
locationText;  
Linear linear ForLocation;  
Image dashboard,image ,image 1,image 2,image 3,image 4,image 5;  
;  
Linear personLinear , tification,dashboardLA,home,message_ ;  
FloatingAction favorite;  
Card LocationCard ,DimondCard ,PlatinumCard ,GoldCard ,SilverCard  
,BronezCard ;  
  
recyler ;  
ItemsAdapter mainScreenAdapter;  
. ;  
<Items> items ;
```



```
List<ress> ressList = (List<ress>) .getFromLocation(location.  
,location. 11,1);  
  
        linearForLocation. ( . );  
  
        locationText. ( ressList.get(0).getressLine(0).to () );  
  
        = ressList.get(0).getressLine(0);  
  
  
//  
        Log.d("Bilaaaal", "onLocationChanged: "+ressList.get(0));  
  
    } (excccc e){  
  
    e. ();  
  
}  
  
}  
  
});  
  
FirebaseMessaging.getInstance().subscribeToTopic("Malir5558");  
FirebaseMessaging.getInstance().subscribeToTopic("Karachi112");  
  
  
personLinear = ( .person_linear_ );  
tification = ( .tification);  
dashboard = ( .dashboard);  
dashboardLA = ( .dashboardLA);  
home= ( .home);  
message_ = ( .message_ );
```

```
favorite = ( .favorite);
LocationCard = ( .LocationBased s);
DimondCard = ( .Dimond s);
PlatiniumCard = ( .Platinium s);
GoldCard = ( .Goald s);
SilverCard = ( .Silver s);
BronezCard = ( .Bronez s);
```

```
personLinear . ( . () {
    onClick( ) {
        = ( (),SubscriptionMenue. );
        ( );
    }
});
```

```
tification. ( . () {
    onClick( ) {
        = ( (), tification . );
        ( );
    }
});
```

```
        }

    });

    home. ( . () {

        onClick( ){

            . ( (), "Home", . ). ();

        }

    });

    message_. ( . () {

        onClick( ){

            = ( (), MessagesAnd tifications. );

            ( );

        }

    });

    if( .getInstance(). () != null)

        if( .getInstance(). (). () != null)

            .with(this). ( .getInstance(). (). ()).into(dashboard);

    dashboardLA. ( . () {

        onClick( ){


```

```
= ( (), Main3 . );
    . ("value", "Dashboard");
    ( );
    // ();
}

});

favorite. ( . () {
    onClick( ) {
        = ( (), Main4 . );
        ( );
    }
});

//Hotitem1

LocationCard . ( . () {
    onClick( ) {
        = ( (), Screen. );
        . ("checker", "Foods");
        ( );
    }
});

});
```

```
DimondCard . ( . () {  
  
    onClick( ) {  
        = ( (), Screen. );  
        . ("checker","Nearby");  
        ( );  
  
    }  
});  
  
PlatinumCard . ( . () {  
  
    onClick( ) {  
        = ( (), Screen. );  
        . ("checker","Sports");  
        ( );  
  
    }  
});  
  
GoldCard . ( . () {  
  
    onClick( ) {  
        = ( (), Screen. );  
        . ("checker","Home Appliances");  
    }  
});
```

```
( );  
  
}  
});  
SilverCard . ( . () {  
  
    onClick( ) {  
        = ( (), Screen. );  
        . ("checker","Makeup");  
        ( );  
  
    }  
});  
BronezCard . ( . () {  
  
    onClick( ) {  
        = ( (), Screen. );  
        . ("checker","Wearings");  
        ( );  
  
    }  
});
```

```
items = <Items>();

items . ( Items("Nearby",R. .ic_navigation_black_24dp));

items . ( Items(" s",R. .discounted s));

items . ( Items("Top Pick",R. .ratedoffers));

items . ( Items("Local Offers",R. .local_offer));

items . ( Items("Foods",R. .restaurant));

items . ( Items("Breakfast",R. .breakfast));

items . ( Items("Bar",R. .bar));

items . ( Items("Shopping",R. .grocery_store));

items . ( Items("Card Offers",R. .creditcard_offers));

items . ( Items(" tifications",R. . tification));



recyler = ( ) ( .item );

recyler . (true);

= Linear (this, .HORIZONTAL,false);

recyler .set ( );

recyler .setItem CacheSize(4);

Animation myanim = AnimationUtils. Animation( (),R.anim.mytrans);

recyler .setAnimation(myanim);

mainScreenAdapter = ItemsAdapter(items );

recyler .setAdapter(mainScreenAdapter);
```

```
image = ( .hotItem1);  
36  
image 1 = ( .hotItem2);  
  
image 2 = ( .hotItem3);  
  
image 3 = ( .hotItem4);  
  
image 4 = ( .hotItem5);  
  
image 5 = ( .hotItem6);  
  
= ( .);  
}
```

```
onBackPressed() {  
  
.Builder(this)  
  
.set ("Quit PromoLac")  
  
. (" you want to quit PromoLac")  
  
. (" ", Interface. ()  
  
{  
  
onClick( Interface , int which) {  
  
();  
  
}  
  
})  
. (" ", null)
```

```
    . ();
}

contactsUS{
    , ;
}

contactsUS( , ) {
    this. = ;
    = ;
}

get () {
    ;
}

set ( ) {
    this. = ;
}

get () {
    ;
}
```

```
    set( ) {
        = ;
    }

}

ReadContacts(){
    Cursor phones = getContentResolver().query(ContactsContract.Comm
Kinds.Phone.CONTENT_URI, null,null,null, null);

    while (phones.moveToNext())

    {

        =phones.get (phones.getColumnIndex(ContactsContract.Comm
Kinds.Phone.DISPLAY_));

        phoneNumber = phones.get

(phones.getColumnIndex(ContactsContract.Comm Kinds.Phone.NUMBER));

        contactsUS contactsUS = contactsUS( ,phoneNumber);
    62
        FirebaseDatabase.getInstance().getReference().child("Contacts")

            .child("--" + .getInstance(). () .getDisplay ())
            .push()
            . (contactsUS);

    }

    phones. ();
}

}
Main3 extends {
```

```
Image image ,backhome,done;  
StorageReference storageReference;  
;  
Uri imageUri;  
Linear ,home,location, tificationsetting;  
;  
;  
Location location ;  
LocationListener locationListener;  
  
protected onPostExecute() {  
    if( .getInstance(). () . () != null)  
        .with(this). ( .getInstance(). () . ()).into(image );  
    .onPostExecute();  
}  
  
protected (Bundle ){  
    . ();  
    (R. . _main3);
```

```
= ( . );

tificationsetting = ( . tificationsetting);

home = ( .home_linear_);

backhome = ( .backhome);

location = ( .location);

text;

text = ( ) ( . );

final  = get ();

str = . ("value");

text. (str);

if( .getInstance(). () . () != null)

    .with(this). ( .getInstance(). () . ()).into(image );

location. ( . () {

    onClick( v) {

        final  1 =  ( (),Maps . );
```

```
    1. ("dec er","1");
    ( 1);
}

});

tificationsetting. ( . () {

    onClick( v) {
        ( ( (), tification . ));
    }
});

image . ( . () {

    onClick( v) {
        =  ();
        setType("image/*");
        . ( .ACTION_GET_CONTENT);
        ForResult( .createChooser( , "Select Picture"),1);
    }
}
```

```
});

.( .() {

    onClick( v) {
        getInstance().sig ut();
        = ( (), . );
        ( );
        ();
    }
});

= ( ) ( .pro  );
.( .getInstance(). ().getDisplay ());
done = (Image ) ( .pro  Pic);
done. ( .IN );
.( .() {

    onClick( ) {
        .setCursor (true);
        .setFocusableInTouchMode(true);
        .setInputType(InputType.TYPE__TEXT);
        .requestFocus();
    }
});
```

```
        }

    });

done. ( . () {

    onClick( ) {

        .setCursor (false);

        .setFocusableInTouchMode(false);

        done. ( .IN );

        = .getText().to ();

        UpdatePro ( );

    }

});

home. ( . () {

    onClick( v) {

        ();

    }

});

backhome. ( . () {

    onClick( v) {
```

```
        () ;

    }

});

}

protected Result(int , int , Nullable data) {

    . Result( , , data);

    if( ==1 && ==RESULT_OK){

        imageUri =data. ();

        {

            Bitmap bitmap=


MediaStore.Images.Media.getBitmap(getContentResolver(),imageUri);

        image .setImageBitmap(bitmap);

        up (bitmap);

    } (excccc e){

        . (this, "excccc"+e.to (), . ). ();

    }

}
```

```
    }

}

up (Bitmap bitmap) {
    47
    FirebaseStorage storage = FirebaseStorage.getInstance();
    StorageReference storageRef = storage.getReference("Users/");
    StorageReference mountainImagesRef = storageRef.child("images"+ .getU ());
    ByteArrayOutputStream baos =  ByteArrayOutputStream ();
    30
    bitmap.compress(Bitmap.CompressFormat.JPEG, 20, baos);
    byte[] data = baos.toByteArray();
    Up Task up Task = mountainImagesRef.putBytes(data);
    up Task. (  () {
        ( nNull excccc excccc) {
        // Handle unsuccessful up s
        . (Main3 .this, "Failed", . ). ();
    }
}). ( <Up Task.TaskSnapshot>() {
    30
    (Up Task.TaskSnapshot taskSnapshot) {
        // taskSnapshot.getMetadata() contains  metadata such as size, content-
        type, and down  URL.
```

```
Task<Uri> task1= taskSnapshot.getMetadata().getReference().getDown
Url(). OnCompleteListener( OnCompleteListener<Uri>) {
    onComplete( nNull Task<Uri> task) {
        task.getResult();
        . (Main3 .this, "Get link", . ). ();
        Log.d("down Uri", "onComplete: " +task.getResult().to ());
        UpdatePro (task.getResult());
    }
});

Uri down Url = taskSnapshot.getUp SessionUri();
}

});

}

UpdatePro (Uri down Url) {
UserPro ChangeRequest pro Updates = UserPro ChangeRequest.Builder()
.setPhotoUri(down Url)
.build();

.updatePro (pro Updates)
. OnCompleteListener( OnCompleteListener<>() {
```

```
        onComplete( null Task<> task) {  
            if (task.isSuccessful()) {  
                Log.d("Pakistan", "User pro updated.");  
            }  
        }  
    }  
  
    UpdatePro ( ) {  
        UserPro ChangeRequest pro Updates = UserPro ChangeRequest.Builder()  
            .setDisplay ( )  
            .build();  
  
        .updatePro (pro Updates)  
            . OnCompleteListener( OnCompleteListener<>) {  
  
                onComplete( null Task<> task) {  
                    if (task.isSuccessful()) {  
                        Log.d("Pakistan", "User pro updated.");  
                    }  
                }  
            }  
    }  
}
```

```
}

extends . {

;

;

login;

EditText editText1, editText2;

forgetpass, register;

, Password;

Relative backhome;

;

protected (Bundle ){

.();

(R. . _main);

getWindow().getDecor ().setSystemUiVisibility( .SYSTEM_UI_FLAG_ _STABLE |  
13  
.SYSTEM_UI_FLAG_ _FULLSCREEN);

login = ( .login);

register = ( .register );

editText1 = (EditText) ( .Login );

editText2 = (EditText) ( .Password );

forgetpass = ( ) ( .Forget );
```

```
= ( . _Login);
    . ( .IN );

= getInstance();
login. (this);
forgetpass. (this);
register. (this);

}

onClick( v) {
if (v == login) {
    = editText1.getText().to ();
    Password = editText2.getText().to ();
    if ( .isEmpty() || Password.isEmpty())
        . ( .this, "Enter Required Field", . ). ();
    else {
        . ( . );
        v. ( . );
        val ate( , Password);
    }
}
```

```
    } else if (v == forgetpass) {  
        ( (this, ForgetPass. ));  
  
    } else if (v == register) {  
        ( ( (), Registration. ));  
    }  
  
    val ate( , password) {  
        .signInWithAndPassword( , Password). ( () {  
  
            ( nNull exccccc e) {  
                login. ( . );  
                . ( . );  
                . ( .this, "Incorrect or password ", . ). ();  
            }  
        }). ( <AuthResult>() {  
  
            (AuthResult authResult) {  
                . ( .this, "Login Successfully", . ). ();  
            }  
        })  
    }  
}
```

```
        = ( (), Main2 . );
        ( );
        ();
    }

});

}

onBackPressed() {
    .Builder(this)
    .set ("Quit PromoLac")
    . (" you want to quit PromoLac")
    . (" ",  Interface. () {

        onClick( Interface , int which) {
            ();
        }
    })

    . (" ", null)
    . ();
}
```

```
}

}

Maps extends OnMapReadyCallback {

    GoogleMap mMap;
    Location location ;
    LocationListener locationListener;

protected (Bundle ){

    .();
    14
    (R. . _maps);

// Obtain the SupportMap and get tified when the map is ready to be used.

SupportMap map = (SupportMap )getSupport ()
    .find By ( .map);
    map .getMapAsync(this);

    = get ();

int dec er = Integer.parseInt( . ("dec er"));

Location lm = (Location )getSystemService( .LOCATION_SERVICE);
```

```
if(dec er == 1){

    SingleShotLocationProv er.requestSingleUpdate( (),

        SingleShotLocationProv er.LocationCallback() {

            on LocationAvailable(SingleShotLocationProv

er.GPSCoordinates location) {

                Log.d("Location", "my location is " + location.to ());

                LatLng sydney = LatLng(location. , location. 11);

                mMap. Marker( MarkerOptions().position(sydney). ("My

Location"));

                mMap.moveCamera(CameraUpdateFactory. LatLng(sydney));




                LatLng currentLocation = LatLng(location. ,location. 11);

27
                mMap.moveCamera(CameraUpdateFactory.

LatLangZoom(currentLocation,15));

                // Zoom in, animating the camera.

                mMap.animateCamera(CameraUpdateFactory.zoomIn());

                // Zoom out to zoom level 10, animating with a duration of 2 .

                mMap.animateCamera(CameraUpdateFactory.zoomTo(15), 2000,

null);

            }

        });

}
```

```
    } else {  
  
        a = .("longtech");  
        b = .(" ");  
        Log.d("OnMaps", " : "+a+" "+b);  
  
    }  
  
}
```

```
onMapReady(GoogleMap googleMap) {  
    mMap = googleMap;
```

```
    }

}

Registration extends  {

    ;

    ;

register;

EditText , ,Password;

mail,pass, ;

;

Image back_Registration;

protected  (Bundle ){

    .();

(R. . _registration);

register= ( .register r);

= ( . );

= ( . r);

Password = ( .Password r);

back_Registration= ( .back_Registration);
```

```
= ( . _registration);
. ( .IN );
= null;
pass=null;
mail=null;
back_Registration. ( . () {
onClick( v) {
();
}
});
register. ( . () {
onClick( v) {
register. ( . );
. ( . );
mail= .getText().to ();
pass=Password.getText().to ();
= .getText().to ();
= .getInstance();
if(mail.isEmpty() || pass.isEmpty()) {
```

```
. (Registration.this, "Enter Val Mail and Password", . ). ();
. ( . );
register. ( . );
}

else {
    .createUserWith AndPassword(mail, pass). ( <AuthResult>() {
        (AuthResult authResult) {
            . (Registration.this, "Registered Successfully", . ). ();
            ( ( (), Main2 . ));
            ();
        }
    }). ( () {
        ( nNull exccccc e) {
            . (Registration.this, "" + e.getMessage(), . ). ();
            . ( . );
            register. ( . );
        }
    });
}

});
```

```
}

UpdatePro ( ){

UserPro ChangeRequest pro Updates = UserPro ChangeRequest.Builder()

.setDisplay ( )

.build();

.getInstance(). ().updatePro (pro Updates)

.OnCompleteListener( OnCompleteListener<>() {

    onComplete( nNull Task<> task) {

        if (task.isSuccessful()) {

            Log.d("Pakistan", "User pro updated.");

        }

    }

});;

// .getInstance(). ().send Verification();

}

}

{

price;

;
```

```
    img;  
    longtech, ,rating ;  
  
    s( price, , img){  
        this. price = price;  
        this. = ;  
        this. img = img;  
  
    }  
  
    s( price, , img, longtech, , rating){  
        this. price = price;  
        this. = ;  
        this. img = img;  
        this.longtech = longtech;  
        this. = ;  
        this.rating = rating;  
    }  
  
    getlongtech() {  
        longtech;
```

```
}
```

```
setlongtech( longtech){  
    this.longtech = longtech;  
}
```

```
get () {  
    ;  
}
```

```
set ( ) {  
    this. = ;  
}
```

```
getRating() {  
    rating;  
}
```

```
setRating( rating){  
    this.rating = rating;  
}
```

```
s() {
```

```
}
```

```
get price() {
```

```
    price;
```

```
}
```

```
set price( price) {
```

```
    this. price = price;
```

```
}
```

```
get () {
```

```
    ;
```

```
}
```

```
set ( ){
```

```
    this. = ;
```

```
}
```

```
get img() {
```

```
    img;
```

```
}
```

```
    set img( img) {  
        this. img = img;  
    }
```

```
}
```

```
sAdapter extends .Adapter< sAdapter. s Holder> {
```

```
    <s> s ;  
    ;
```

```
    sAdapter( <s> s ) {  
        this. s = s ;  
    }
```

```
    sAdapter( <s> s ,   ) {  
        this. s = s ;  
        this. = ;  
    }
```

```
nNull
```

```
s Holder Holder( nNull , int Type) {  
    = .from( .get ()).inflate(R. .items_ s, ,false);  
    51  
    s Holder s Holder = s Holder( );  
    s Holder;  
}
```

```
onBind Holder( nNull s Holder holder, int position) {  
    s s= s .get(position);  
  
.with( ). ( s.get img()).into(holder.image );  
  
holder. . ( s.get());  
holder.Price. ( s.get price());  
}
```

```
int getItemCount() {  
    s .size();  
}
```

```
s Holder extends . Holder . {  
  
    Image image ;  
  
    , Price;  
  
    s Holder( nNull item ) {  
  
        (item );  
  
        item . (this);  
  
        //  
  
        image = (Image ) item . ( .Vcard Image);  
  
        = ( ) item . ( .Vcard );  
  
        Price = ( ) item . ( .VcardPrice);  
  
    }  
  
    onClick( ) {  
  
        // s info pre  
  
        = ( .get (), sInfoPre . );  
  
        . ("Price", s .get(get Position()).get price());
```

```
. ("Img", s .get(get Position()).get img());  
.  
. (" ", s .get(get Position()).get () );  
. (" ", s .get(get Position()).get () );  
. ("longtech", s .get(get Position()).getlongtech());  
. ("rating", s .get(get Position()).getRating());  
  
Log.d("InamInam", "onClick: "+ s .get(get Position()).getlongtech()+";" + s  
.get(get Position()).get ()+"/"+get Position());  
.get (). ( );  
}  
}  
  
}  
  
package com.example.PromoLac.Adapters;  
  
Firebase 1 {  
    ,Price,Location,Link;  
  
Firebase 1( , price, location, link){  
    = ;  
    Price = price;  
    Location = location;  
    Link = link;
```

```
}

get () {
    ;
}

set ( ) {
    =
}

getPrice() {
    Price;
}

setPrice( price) {
    Price = price;
}

getLocation() {
    Location;
}

setLocation( location) {
```

```
    Location = location;  
}  
  
getLink() {
```

```
    Link;  
}
```

```
setLink( link) {  
    Link = link;  
}  
}
```

```
Items {  
    label;  
    int Image;
```

```
Items( label, int image) {  
    this.label = label;  
    Image = image;  
}
```

```
getLabel() {  
    label;
```

```
}

setLabel( label) {
    this.label = label;
}

int getImage() {
    Image;
}

setImage(int image) {
    Image = image;
}

}

ItemsAdapter extends .Adapter<ItemsAdapter. Holder>{
    <Items> items ;

    ItemsAdapter( <Items> items ) {
        this.items = items ;
    }
}
```

```
nNull
```

```
Holder Holder( nNull , int Type) {  
    = .from( .get ()).inflate(R. .item, ,false);  
    Holder Holder =  Holder( );  
    Holder;  
}
```

```
onBind Holder( nNull Holder holder, int position) {  
    Items items = items .get(position);  
    holder.image .setImageResource(items.getImage());  
    holder. . ((CharSequence) items.getLabel());  
}
```

```
int getItemCount() {  
    items .size();  
}  
Holder extends . Holder .{
```

```
Image image ;  
;  
Holder( nNull item ) {  
    (item );  
    item . (this);  
    = ( ) item . ( .Recycle );  
    image = (Image ) item . ( .Recycle Image );  
}  
  
onClick( ) {  
    if(get Position() == 0) {  
        = ( .get (), Screen. );  
        . ("checker","Nearby");  
        .get (). ( );  
    }  
    if(get Position() == 1) {  
        = ( .get (), Screen. );  
        . ("checker","Discounted s");  
        .get (). ( );  
    }  
    else if(get Position() == 2) {  
        = ( .get (), Screen. );  
        . ("checker","Random");  
        .get (). ( );  
    }  
}
```

```
= ( .get (), Screen. );
. ("checker","Top Pick");
.get () ( );
}

else if(get Position() ==3) {
= ( .get (), Screen. );
. ("checker","Local Offers");
.get () ( );
}

else if(get Position() ==4) {
= ( .get (), Screen. );
. ("checker","Foods");
.get () ( );
}

else if(get Position() ==5) {
= ( .get (), Screen. );
. ("checker","Breakfast");
.get () ( );
}

else if(get Position() ==6) {
= ( .get (), Screen. );
. ("checker","Bar");
.get () ( );
}
```

```
    }

    else if(get Position() ==7) {

        = ( .get (), Screen. );

        . ("checker","Shopping");

        .get ()( );

    }

    else if(get Position() ==8) {

        = ( .get (), Screen. );

        . ("checker","Credit Offers");

        .get ()( );

    }

    }else if(get Position() == 9){

        //Code here for tifications

        = ( .get (), MessagesAnd tifications. );

        . ("checker"," tificationScreen");

        .get ()( );

    }

}

}

}
```

```
Messages extends {
    static ARG_ = " ";
    static ARG_ = " ";
    Input fi;
    ;
}

ObjectInput oi;
;
<Messages tifications> messages tifications ;
Linear ;
MessagesAdapter messagesAdapter;
DatabaseReference databaseReference;
<Messages tifications> ;

// TODO: Re and change types of parameters
1;
2;

On InteractionListener mListener;
```

```
Messages () {  
    // Required empty constructor  
}  
  
static Messages Instance( , ) {  
    Messages = Messages ();  
    Bundle = Bundle();  
    .put (ARG_ , );  
    .put (ARG_ , );  
    .setArguments( );  
    ;  
}  
  
(Bundle ) {  
    . ( );  
    if (getArguments() != null) {  
        1 = getArguments().get (ARG_ );  
        2 = getArguments().get (ARG_ );  
    }  
}
```

```
}

( , container,
    Bundle ){

R = .inflate(R.._messages, container, false);
=R . ( ._message );

= ( ) R . ( .Messages Recycle );
    = Linear (get (), .VERTICAL,false);
.set ( );

messages tifications = <>(); // null /// htana hy
    45
final DatabaseReference databaseReference;

databaseReference = FirebaseDatabase.getInstance().getReference();
```

```
databaseReference.child("Messages").ValueEventListener()
ValueEventListener() {
    45
    Change( nNull DataSnapshot dataSnapshot) {
        for (DataSnapshot contact: dataSnapshot.getChildren()) {
            Messages tifications m = contact.getValue(Messages tifications. );
            messages tifications . (m);
            Log.d("googlegoogle", " Change: "+m.get ()+ "\n");
        }
        messagesAdapter. tifyDataSetChanged();
        . ( . );
    }
    ( nNull DatabaseError databaseError) {
    }
});
messagesAdapter =  MessagesAdapter(messages tifications );
.setAdapter(messagesAdapter);
```

```
R ;  
}  
  
<Messages tifications> Data( ) throws IOException {  
{  
    = <Messages tifications>();  
  
    Path = get ().get sDir().getPath().to () + ;  
    Root = ( Path);  
    fi = Input ( Root);  
    oi = ObjectInput (fi);  
  
    while (true){  
        . ((Messages tifications)oi.readObject());  
    }  
}  
} ( EOFException e){  
    oi. ();  
    fi. ();
```

```
    } ( excccc e){  
        Log.d("pakistan", " Data: "+e.to());  
  
        . (get (), "Failed to Retrieve"+e.to (), . ). ();  
    }  
;  
  
}  
  
14  
on Pressed(Uri uri) {  
    if (mListener != null) {  
        mListener.on Interaction(uri);  
    }  
}  
  
( ){  
    . ();  
}  
  
() {  
    . ();
```

```
mListener = null;  
}  
  
interface OnInteractionListener {  
    // TODO: Update argument type and  
    onInteraction(Uri uri);  
}  
}  
  
tifications extends {  
    static ARG_ = " ";  
    static ARG_ = " ";  
  
    1;  
    2;  
  
    OnInteractionListener mListener;  
  
    tifications () {  
    }  
  
    static tifications Instance( , ) {  
        tifications  = tifications ();
```

```
        Bundle = Bundle();

        .put (ARG_, );
        .put (ARG_, );
        .setArguments( );
        ;
    }

(Bundle ) {
    .();
    if ( getArguments() != null) {
        1 = getArguments().get (ARG_);
        2 = getArguments().get (ARG_);
    }
}
```

14

```
( , container,
        Bundle ){

// Inflate the for this

R = .inflate(R. _main, container, false);
```

```
R ;  
}  
  
on Pressed(Uri uri) {  
    14  
    if (mListener != null) {  
        mListener.onInteraction(uri);  
    }  
}  
  
( ){  
    .();  
}  
  
{  
    .();  
    mListener = null;  
}  
  
interface OnInteractionListener {  
    on Interaction(Uri uri);  
}  
}
```

```
SingleShotLocationProv er {  
  
    static interface LocationCallback {  
        9  
        on LocationAvailable(GPSCoordinates location);  
    }  
  
    static requestSingleUpdate(final , final LocationCallback callback) {  
        final Location location = (Location ) .getSystemService(  
.LOCATION_SERVICE);  
        boolean isNetworkEnabled = location .isProv erEnabled(Location  
.NETWORK_PROV ER);  
        9  
        if (isNetworkEnabled) {  
            Criteria criteria =  Criteria();  
            criteria.setAccuracy(Criteria.ACCURACY_COARSE);  
            if ( Compat.checkSelfPermission( . . _LOCATION) != Package . _ &&  
Compat.checkSelfPermission( . . _COARSE_LOCATION) != Package . _ ) {  
                ;  
            }  
            location .requestSingleUpdate(criteria,  LocationListener() {  
                9  
                onLocationChanged(Location location) {  
                    callback.on LocationAvailable( GPSCoordinates(location.get (),  
location.get 11()));  
                }  
            });  
        }  
    }  
}
```

```
}

onStatusChanged( prov er, int status, Bundle extras) {

}

onProv erEnabled( prov er) {

}

onProv erDisabled( prov er) {

}, null);

} else {

boolean isGPSEnabled = location .isProv erEnabled(Location .GPS_PROV
ER);

9
if (isGPSEnabled) {

Criteria criteria = Criteria();
criteria.setAccuracy(Criteria.ACCURACY_ );
location .requestSingleUpdate(criteria, LocationListener()) {

onLocationChanged(Location location) {
```

```
        callback.onLocationAvailable( GPSCoordinates(location.get(),
location.get(11())));
    }

    onStatusChanged( provider, int status, Bundle extras) { }

    onProviderEnabled( provider) { }

    onProviderDisabled( provider) { }

}, null);
}

}

}

static requestLiveUpdate(final , final LocationCallback callback) {
    final Location location = (Location) .getSystemService(
.LOCATION_SERVICE);;

    boolean isNetworkEnabled = location .isProviderEnabled(Location
.NETWORK_PROVIDER);

    if (isNetworkEnabled) {
        Criteria criteria = Criteria();
        criteria.setAccuracy(Criteria.ACCURACY_COARSE);
        if ( Compat.checkSelfPermission( . . . _LOCATION) != PackageManager . _ &&
Compat.checkSelfPermission( . . . _COARSE_LOCATION) != PackageManager . _ ){
```

```
// TODO: Consider calling
//    Compat#requestS
// here to request the missing Ss, and then overriding
//    onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
//                                @NonNull int[] grantResults)
// to handle the case where the user grants the permission. See the documentation
// for Compat#requestS for more details.

;

}

location.requestLocationUpdates(Location.GPS_PROVIDER, 0, 500,
        new LocationListener() {
            @Override
            public void onLocationChanged(Location location) {
                callback.onLocationAvailable(GPSCoordinates(location.getLatitude(),
                        location.getLongitude()));
            }

            @Override
            public void onStatusChanged(String provider, int status, Bundle extras) {
            }
        }
);
```

```
    onProv erEnabled( prov er) {  
  
    }  
  
    onProv erDisabled( prov er) {  
  
    }  
});  
} else {  
    boolean isGPSEnabled = location .isProv erEnabled(Location .GPS_PROV  
ER);  
    if (isGPSEnabled) {  
        Criteria criteria =  Criteria();  
        criteria.setAccuracy(Criteria.ACCURACY_ );  
        location .requestSingleUpdate(criteria,  LocationListener() {  
  
            onLocationChanged(Location location) {  
                callback.on LocationAvailable( GPSCoordinates(location.get (),  
location.get 11()));  
            }  
        }  
    }  
}
```

```
        onStatusChanged( provider, int status, Bundle extras) { }

        onProviderEnabled( provider) { }

        onProviderDisabled( provider) { }

    }, null);

}

}

// consider using Location instead of this dummy wrapper

static GPSCoordinates {

    float lat = -1;

    float lon = -1;

    GPSCoordinates(float latitude, float longitude) {

        lat = latitude;
        lon = longitude;
    }

    GPSCoordinates(double latitude, double longitude) {

        lat = (float) latitude;
        lon = (float) longitude;
    }
}
```

```
}

}

ServiceForLocation extends Service {

    static final int tify = 180000; //interval between two services(Here Service every 5
Minute)

    Handler mHandler = Handler(); // on a ther exccc to a crash
    Timer mTimer = null; //timer handling

IBinder onBind( ) {
    throw UnsupportedOperationException("t yet implemented");
}

() {
    mTimer = Timer(); //recreate
    mTimer.scheduleAtFixedRate( TimeDisplay(), 0, tify); //Schedule task

}
```

```
onDestroy() {  
    .onDestroy();  
    .(this, "Service is Destroyed", .).();  
}  
  
// TimeDisplay for handling task  
TimeDisplay extends TimerTask {  
  
    () {  
        // on a ther exccccc  
        mHandler.post( nable() {  
  
            () {  
                // display  
                .(ServiceForLocation.this, "Service is ning", .).();  
                SingleShotLocationProv er.requestLiveUpdate( ),  
                SingleShotLocationProv er.LocationCallback() {  
                    on LocationAvailable(SingleShotLocationProv  
er.GPSCoordinates location) {  
                        Log.d("LocationService ning", "my location is " + location.to());  
                        = (());  
                    }  
                }  
            }  
        }  
    }  
}
```

```
        List<ress> ressList = (List<ress>)
        .getFromLocation(location.,location. 11,1);
        = ressList.get(0).getressLine(0);
        Up ToDatabase( );
    }

} (exccccc e){
    e. ();
}
}

});

}

});

}

}

Up ToDatabase( ){

DatabaseReference databaseReference;
13
databaseReference = FirebaseDatabase.getInstance().getReference();

databaseReference.child("UsersData").child( .getInstance(). () .getU
());
databaseReference.child("LocationsData").push(). ( Locati ( , Timestamp(System.currentTimeMillis())));
};
```

```
Log.d("Ammmmmmaaaar", "Up ToDatabase: "+  
Timestamp(System.currentTimeMillis().to ());  
}  
  
Locati {  
;  
Timestamp timestamp;  
  
Locati ( , Timestamp timestamp) {  
this. = ;  
this.timestamp = timestamp;  
}  
  
get () {  
;  
}  
  
set ( ){  
this. = ;  
}  
  
Timestamp getTimestamp() {  
timestamp;  
}
```

```
    setTimestamp(Timestamp timestamp) {  
        this.timestamp = timestamp;  
    }  
}  
}
```

Nullable

```
IBinder onBind( ) {  
    null;  
}  
  
( ) {  
    .(this, "Service of Location Starting", .).();  
  
SingleShotLocationProv er.requestLiveUpdate( ),  
    SingleShotLocationProv er.LocationCallback() {  
        on LocationAvailable(SingleShotLocationProv er.GPSCoordinates  
location) {  
            Log.d("Location", "my location is " + location.to());  
            = ( );  
            {  
                List<ress> ressList = (List<ress>) .getFromLocation(location,  
,location. 11,1);  
                = ressList.get(0).get ressLine(0);  
                unsubscribe();
```

```
        subscribe( );
        Up ToDatabase( );

//        Log.d("Bilaaaal", "onLocationChanged: "+ ressList.get(0));

    } (excccc e){
        e. ();
    }

}

});

. ();

}

onDestroy() {
    . (this, "Service For Location Destroyed", . ). ();
    .onDestroy();
}

Up ToDatabase( ) {
    DatabaseReference databaseReference;
```

13

```
databaseReference = FirebaseDatabase.getInstance().getReference();  
  
databaseReference.child("UsersData").child( .getInstance(). () .getU  
()).child("LocationsData").push(). ( Locati ( , Timestamp(System.currentTimeMillis())));  
Log.d("Ammmmmmaaaar", "Up ToDatabase: "+  
Timestamp(System.currentTimeMillis()).to () );  
}  
  
Locati {  
;  
Timestamp timestamp;  
  
Locati ( , Timestamp timestamp) {  
this. = ;  
this.timestamp = timestamp;  
}  
  
get () {  
;  
}  
  
set ( ) {  
this. = ;  
}
```

```
    Timestamp getTimestamp() {  
        timestamp;  
    }  
  
    setTimestamp(Timestamp timestamp) {  
        this.timestamp = timestamp;  
    }  
}  
  
}  
  
MessagesAdapter extends .Adapter<MessagesAdapter.Messages Holder><  
  
<Messages tifications> messages tifications ;  
  
MessagesAdapter( <Messages tifications> messages tifications) {  
    this.messages tifications = messages tifications;  
}  
  
nNull
```



```
messages tifications .size();  
}  
  
Messages Holder extends . Holder .OnLongClickListener {  
  
    Relative message_item;  
    Image image ;  
    ,Content,Time;  
    Messages Holder( nNull item ) {  
        (item );  
  
        image = (Image ) item . ( .MessagesItem_Image);  
        message_item= (Relative ) item . ( .message_item);  
        = ( ) item . ( .MessagesItem_ );  
        Content = ( ) item . ( .MessagesItem_Content);  
        Time = ( ) item . ( .MessagesItem_Time);  
        item .setOnLongClickListener( this);  
    }  
}
```

```
boolean onLongClick( View v) {  
    .(v.get (), "jjjjjjjjjjjjjjj"+get Position(), .).();  
    // message_item.setlongtechColor(Color.parseColor("#808080"));  
  
    = .Builder(v.get ())  
    .set (" you want to Delete?")  
    . (" ", Interface. () {  
  
        onClick( Interface Interface, int i) {  
            final Messages tifications messages tifications = messages tifications  
.get(get Position());  
            messages tifications .remove(get Position());  
            tifyItemRemoved(getAdapterPosition());  
            tifyItemRangeChanged(getAdapterPosition(),messages tifications  
.size());  
        //t  
        13  
        DatabaseReference ref =  
        FirebaseDatabase.getInstance().getReference();  
        Query applesQuery = ref.child("UsersData")  
        .child( .getInstance(). ().getU ())
```

```
.child("Messages");

applesQuery. ListenerForSingleValueEvent( ValueEventListener() {

    60
    Change(DataSnapshot dataSnapshot) {
        for (DataSnapshot appleSnapshot: dataSnapshot.getChildren())
    {

        Messages tifications mn =
        appleSnapshot.getValue(Messages tifications. );
        if(mn.get () == messages tifications.get ()){

            appleSnapshot.getRef().removeValue();
            ;
            messages tifications = <>();
        }
    }
}

(DatabaseError databaseError) {
    Log.e("Exexex", " ", databaseError.toexcccc());
}
});
```

```
        }

    })

    . (" ", Interface. () {

        onClick( Interface Interface, int i) {

            // message_item.setlongtechColor(Color.parseColor("E5E5E5"));

        }

    })

    . ();

true;

}

Interface. ClickListener = Interface. () {

    onClick( Interface , int which) {

        switch (which){

            case Interface. _POSITIVE:

                //Do your progress

                break;

            case Interface. _NEGATIVE:

                //Do your progress

                break;

        }

    }

}
```

```
        }

    }

};

}

}

MessagesAnd tifications extends .{

    Linear tification ,Messages , ;

    Mesage , tification ;

    Image backhome;

protected (Bundle ){

    .();

    (R. . _messages_and_ tifications);

    tification = (Linear ) ( . tification );

    Messages = (Linear ) ( . message_ );

    = (Linear ) ( . ToSet);

    Mesage =( ) ( . Messages);
```

```
backhome=(Image ) ( .backhome);

tification =( ) ( . tifications);

tification . ( .IN );

Mesage . ( .);

getSupport ().beginTransaction().replace( . ToSet, Messages ()).commit();

tification . (this);

Messages . (this);

backhome. (this);

}

onClick( ){

if( == tification ){

tification . ( .);

Mesage . ( .IN );

getSupport ().beginTransaction().replace( . ToSet, tifications ()).commit();
```

```
}

else if ( == Messages ){

    tification . ( .IN );

    Mesage . ( . );

    getSupport ().beginTransaction().replace( . ToSet, Messages ()).commit();

}

else if( ==backhome){

    ();

}

}

Messages tifications Serializable {

    ,Content,DeliveredTime,Category,forWhichLocation;

boolean isRead;

static final long serialVersionUID = -299482035708790407L;

boolean isRead() {

    isRead;

}
```

```
MarkAsRead() {  
    isRead = TRUE;  
}  
  
Messages tifications() {  
}  
  
Messages tifications( , , content, deliveredTime, category) {  
    this. = ;  
    = ;  
    Content = content;  
    DeliveredTime = deliveredTime;  
    Category = category;  
    isRead = FALSE;  
}  
  
get () {  
    ;  
}  
  
set ( ) {  
    this. = ;
```

```
}
```

```
get () {  
    ;  
}
```

```
set ( ) {  
    = ;  
}
```

```
getContent() {  
    Content;  
}
```

```
( content) {  
    Content = content;  
}
```

```
getDeliveredTime() {  
    DeliveredTime;  
}
```

```
setDeliveredTime( deliveredTime) {
```

```
    DeliveredTime = deliveredTime;  
}  
  
getCategory() {  
    Category;  
}  
  
setCategory( category) {  
    Category = category;  
}  
  
to () {  
    "Messages tifications" +  
    " ="" +  + \'\" +  
    ", ="" +  + \'\" +  
    ", Content="" + Content + \'\" +  
    ", DeliveredTime="" + DeliveredTime + \'\" +  
    ", Category="" + Category + \'\" +  
    \'\";  
}  
}  
MessagesService extends Service {
```

```
= "Messages.txt";

DatabaseReference databaseReference;
Messages tifications m;

<Messages tifications> messages tifications ;
static int loop = 0;

@SuppressLint("NewApi")
@TargetApi(16)
@Retention(RetentionPolicy.SOURCE)
@Suppress("WrongConstant")
@RequiresApi(api = Build.VERSION_CODES.KITKAT)

int onStartCommand( , int flags, int start ) {

    // Log.d("Pakistan", "onHandle : Called");

    databaseReference = FirebaseDatabase.getInstance().getReference();

    messages tifications  =  <Messages tifications>();

    //

    databaseReference. ValueEventListener( ValueEventListener() {

        @Override
        public void onDataChange( DataSnapshot dataSnapshot) {
            DataSnapshot ds1= dataSnapshot.child("Messages");
            Iterable<DataSnapshot> iterable= ds1.getChildren();
            for (DataSnapshot contact: iterable) {
                m = contact.getValue(Messages tifications. );
                messages tifications . (m);
            }
            save ( ,messages tifications );
        }
    });

    return START_STICKY;
}
```

```
    }

    ( nNull DatabaseError databaseError) {

    }

});

1//.onStartCommand( , flags, start );

}

onDestroy() {
    .onDestroy();
}

save ( , <Messages tifications> messages tifications){
{
    Path = ()get sDir().getPath().to () +  ;
    Root =  ( Path);
    Root.delete();
    if(! Root.exists()){

```

```
Root.create ();
}

Output f = Output ( Root);
ObjectOutput o = ObjectOutput (f);

// Write objects to
for (Messages tifications m : messages tifications
) {
    o.writeObject(m);
}

o. ();
f. ();
. (this, "SuccessFull Write in Messages Serv", .). ();
} (exccccc e) {
    . (this, "Failed to Save in ", .). ();
}
}
```

Nullable

```
IBinder onBind( ) {
```

```
        null;  
    }  
}  
  
Splash extends {  
    Animation animation;  
    Image image ;  
  
protected (Bundle ){  
  
    .();  
    (R. . _splash);  
  
    animation = AnimationUtils. Animation(this, R.anim.mytrans);  
    image = ( .promolac_splash);  
    image .setAnimation(animation);
```

```
excccc longtech = excccc() {  
    () {  
        {  
            // excccc will sleep for 5  
            sleep(1*1000);  
  
            // 5 to a ther  
            if( .getInstance(). () == null) {  
                ( ( (), . ));  
            } else {  
                i = (getBase (), Main2 . );  
                (i);  
            }  
            //Remove  
            ();  
        } (excccc e) {  
            }  
        }  
    };  
    // start excccc  
    longtech.start();  
}  
}
```

=====controlPanel=====

```
namespace fypPromolac.Controllers12
{
    public class controlPanelController : Controller
    {
        // GET: controlPanel
        public ActionResult controlPanel()
        {

            using (var context = new promoLacDbEntities())
            {

                int id = vendorId(User.Identity.Name);
                userPackage d = new userPackage();

6
                var result = context.userPackages.Where(x => x.vendorId == id).Select(x =>
                new userPackageModel()

                {
```

```
notificationSent = x.notificationSent,  
packageld=x.packageld,  
dividedShareOfMessages=x.dividedShareOfMessages,  
messagesRemaining=x.dividedShareOfMessages-x.notificationSent,  
  
}).FirstOrDefault();  
  
/*d.messagesSent = result.messagesSent;  
d.messagesAllowed = messagesAllowed(result.packageld);  
d.messagesRemaining = d.messagesAllowed-d.messagesSent;  
  
*/  
  
var result1 = context.vendors.Where(x => x.vendorId == id).First();  
List<DataPoint> dataPointsNot = new List<DataPoint>();  
DateTime time = result1.registerTimestamp;  
  
int y = DateTime.Compare(time, DateTime.Now);  
int count = 0;  
while (y!=1)  
{  
    var dt = time.Date;  
    var z = context.notifications.Where(a => (a.vendorId == id) &&  
(a.notificationTime == dt)).Count();  
    dataPointsNot.Add(new DataPoint(count,z));
```

```
time=time.AddDays(1);

y = DateTime.Compare(time.Date, DateTime.Now.Date);

count++;

}

ViewData["name"]= "Inam";

ViewBag.DataPointsNotification =

JsonConvert.SerializeObject(dataPointsNot);

List<DataPoint> dataPointsDeal = new List<DataPoint>();

time = result1.registerTimestamp;

count = 0;

y = DateTime.Compare(time, DateTime.Now);

while (y != 1)

{

    var dt = time.Date;

    var z = context.deals.Where(a => (a.vendorId == id) && (a.dealTime ==

dt)).Count();

    dataPointsDeal.Add(new DataPoint(count, z));

    time = time.AddDays(1);

    y = DateTime.Compare(time.Date, DateTime.Now.Date);

    count++;

}
```

```
}

ViewData["name"] = "Inam";

ViewBag.DataPointsDeal = JsonConvert.SerializeObject(dataPointsDeal);

List<DataPoint> dataPointsFence = new List<DataPoint>();

time = result1.registerTimestamp;

count = 0;

y = DateTime.Compare(time, DateTime.Now);

while (y != 1)

{

    var dt = time.Date;

    var z = context.fences.Where(a => (a.vendorId == id) &&

(DbFunctions.TruncateTime(a.startTime)== dt)).Count();

    dataPointsFence.Add(new DataPoint(count, z));

    time = time.AddDays(1);

    y = DateTime.Compare(time.Date, DateTime.Now.Date);

    count++;

}

ViewData["name"] = "Inam";

ViewBag.DataPointsFence =

JsonConvert.SerializeObject(dataPointsFence);

return View(result);
```

```
    }

}

public int messagesAllowed(int z)

{
    using (var context = new promoLacDbEntities())

    {

        var result = context.packages.Where(x => x.packagesId == z).Select(x =>

new packageModel()

        {

            messagesAllowed = x.messagesAllowed

        }).FirstOrDefault();

        return result.messagesAllowed;

    }

}

public int vendorId(string name)

{
```

```
1 using (var context = new promoLacDbEntities())
{
    var result = context.vendors.Where(x => x.vendorUserName ==
name).Select(x => new vendorModel()
{
    vendorId = x.vendorId

}).FirstOrDefault();

    return result.vendorId;
}

}

}

=====dealsController=====
```

16
namespace fypPromolac.Controllers

```
{  
    [Authorize]  
  
    public class dealsController : Controller  
  
    {  
  
        // GET: deals  
  
        public ActionResult createDeal()  
  
        {  
  
            var context = new promoLacDbEntities();  
  
            var y = context.dealCategories.ToList();  
  
            dealsModel x = new dealsModel();  
  
            x.dealCategoryLists = y;  
  
            return View(x);  
        }  
  
        [HttpPost]  
  
        public ActionResult createDeal(dealsModel d, HttpPostedFileBase dealsImage)  
  
        {  
  
            string path = uploadimage(dealsImage);  
  
            var context = new promoLacDbEntities();
```

```
deal obj = new deal();

obj.vendorId = getvendorId(User.Identity.Name);

HttpClient httpClient = new HttpClient();
string baseUrl = "https://localhost:44371/";
string path1 =
"C:\\\\Users\\\\DELL\\\\source\\\\repos\\\\fypPromolac\\\\fypPromolac\\\\Content\\\\images\\\\537570
52inam-picture.jpeg";

HttpResponseMessage response = httpClient.GetAsync(baseUrl +
"api2/fileFirebase/GetFileUpload?id=" + path + "&vId=" + obj.vendorId + "&fileName="
+ dealsImage.FileName).Result;

string stateInfo = response.Content.ReadAsStringAsync().Result;

stateInfo = stateInfo.Substring(1, stateInfo.Count()-2);

obj.dealCategory = getDealCategoryId(d.dealCategory);
```

```
obj.dealDescription = d.dealDescription;
obj.dealName = d.dealName;
obj.dealPrice = d.dealPrice;
obj.latitude = d.latitude;
obj.longitude = d.longitude;
obj.deallImage = path;
obj.dealTime = DateTime.Now.Date;

context.deals.Add(obj);
context.SaveChanges();
tempDeal ddd = new tempDeal();

ddd.dealimg = stateInfo;
ddd.dealprice = d.dealPrice.ToString();
ddd.latitude = d.latitude.ToString();
ddd.longitude = d.longitude.ToString();
ddd.rating = 2.ToString();
ddd.dealtile = d.dealName;
IFirebaseConfig mes = new FirebaseConfig
{
    AuthSecret = "CwnE0VGXMqxNxcY5YSWVNcL7IAM3hfqRuSiNjWm5",
```

```
BasePath = "https://xdadeveloperes.firebaseio.com/"

};

IFirebaseClient mfc;

mfc = new FireSharp.FirebaseClient(mes);

var data = ddd;

PushResponse r = mfc.Push(d.dealCategory + "/", data);

6
return Json(new {success=true,message="dddd"
},JsonRequestBehavior.AllowGet);

}

public int getDealCategoryId(string name)
{
    var context = new promoLacDbEntities();
    var y = context.dealCategories.Where(x => x.categoryName ==
name).FirstOrDefault();
    return y.categoryId;
}

public int getvendorId(string name)
{
7
var context = new promoLacDbEntities();
```

```
var y = context.vendors.Where(x => x.vendorUserName ==  
name).FirstOrDefault();  
  
return y.vendorId;  
}  
  
public string uploadimage(HttpPostedFileBase dealsImage)  
{  
  
Random r = new Random();  
  
string path = "-1";  
  
int random = r.Next();  
  
if (dealsImage != null && dealsImage.ContentLength > 0)  
  
{  
    string extension = Path.GetExtension(dealsImage.FileName);  
  
    if (extension.ToLower().Equals(".jpg") || extension.ToLower().Equals(".jpeg")  
|| extension.ToLower().Equals(".png"))  
  
    {
```

```
try

{
    6
    path = Path.Combine(Server.MapPath("~/Content/images"), random +
Path.GetFileName(dealsImage.FileName));

    dealsImage.SaveAs(path);

    path =
"C:/Users/DELL/source/repos/fypPromolac/fypPromolac/Content/images/" + random +
Path.GetFileName(dealsImage.FileName);

}

catch (Exception ex)

{
    31
    path = "-1";

}

}
```

```
else

{

    Response.Write("<script>alert('Only jpg ,jpeg or png formats are  
acceptable....'); </script>");

}

else

{

    Response.Write("<script>alert('Please select a file'); </script>");

    path = "-1";

}

return path;

}
```

```
    }  
}  
  
=====geoFenceController=====
```

```
19  
namespace fypPromolac.Controllers  
{  
    public class geoFenceController : Controller  
    {  
        // GET: geoFence  
        public ActionResult createFence()  
        {  
            int id = vendorId(User.Identity.Name);  
  
            fenceModel m = new fenceModel();  
            4  
            m.vendorId = id;  
  
            return View(m);  
        }  
        [HttpPost]  
        public ActionResult createFence(fenceModel f)  
        {
```

```
tempFence tf = new tempFence();
tf.Latitude = f.latitude.ToString();
tf.Longitude = f.longitude.ToString();
tf.Notification_ = f.notification;
tf.TimeStamp = DateTime.Now.AddHours(f.hr).ToString("MM/dd/yyyy
HH:mm:ss");

tf.Radius = f.radius.ToString();

IFirebaseConfig mes = new FirebaseConfig
{
    AuthSecret = "CwnE0VGXMqxNxcY5YSWVNcL7IAM3hfqRuSiNjWm5",
    BasePath = "https://xdadeveloperes.firebaseio.com/"
};

IFirebaseClient mfc;
mfc = new FireSharp.FirebaseClient(mes);
var data = tf;
PushResponse r = mfc.Push("Fences_" + "/", data);

updateFencingHours(Convert.ToInt32(f.vendorId),f.hr);

var context = new promoLacDbEntities();
```

```
fence fence_ = new fence();

fence_.latitude = f.latitude;
fence_.longitude = f.longitude;
fence_.startTime = DateTime.Now;
fence_.endTime = DateTime.Now.AddHours(f.hr);
fence_.notification = f.notification;
fence_.radius = f.radius;
fence_.vendorId = f.vendorId;
fence_.area = f.AreaName;

context.fences.Add(fence_);
12
context.SaveChanges();

return Json(new { success = true, message = "ddd" },
JsonRequestBehavior.AllowGet);

}

public void updateFencingHours(int id,int hr)
{
    int v_id= getVendorStatus(id);
7
var context = new promoLacDbEntities();
var result = context.userPackages.Where(x => x.vendorId ==
v_id).FirstOrDefault();
```

```
        result.remainingFencingHours = result.remainingFencingHours - hr;
        context.SaveChanges();
    }

    public int getVendorStatus(int id)
    {
        var context = new promoLacDbEntities();

        var result = context.vendors.Where(x => x.vendorId == id).FirstOrDefault();

        if (result.isHead == "Y")
        {
            return result.vendorId;
        }
        else
        {
            id = Convert.ToInt32(result.headId);

            return id;
        }
    }

    public int vendorId(string name)
    {
        1 using (var context = new promoLacDbEntities())
        {
```

```
var result = context.vendors.Where(x => x.vendorUserName ==  
name).Select(x => new vendorModel()  
{  
    vendorId = x.vendorId  
  
}).FirstOrDefault();  
  
return result.vendorId;  
}  
  
}  
}  
}
```

=====login=====

```
{  
  
6 public class loginController : Controller  
{  
  
    // GET: login  
8    public ActionResult login()
```

```
{  
    return View();  
}  
  
[HttpPost]  
  
public ActionResult login(vendorModel model)  
{  
    using(var context= new promoLacDbEntities())  
    {  
        bool isvalidVendor = context.vendors.Any(x => x.vendorUserName ==  
model.vendorUserName && x.vendorPassword == model.vendorPassword);  
  
        if (isvalidVendor)  
        {  
            FormsAuthentication.SetAuthCookie(model.vendorUserName, false);  
            int id = vendorId(model.vendorUserName);  
            var result = context.userPackages.Where(x => x.vendorId ==  
id).FirstOrDefault();  
  
            int comp = DateTime.Compare(result.packageEndTime, DateTime.Now);  
            if (comp<0)  
            {  
                result.packageStatus = "NotActive";  
                context.SaveChanges();  
                ModelState.AddModelError("", "Package Expired");  
            }  
        }  
    }  
}
```

```
46
else

{
    return RedirectToAction("controlPanel", "controlPanel");
}

}

else
{
    ModelState.AddModelError("", "Invalid User Name and Password");
}

}

return View();
}

public int vendorId(string name)
{
    1
    using (var context = new promoLacDbEntities())
    {
```

```
var result = context.vendors.Where(x => x.vendorUserName ==  
name).Select(x => new vendorModel()  
{  
    vendorId = x.vendorId  
  
}).FirstOrDefault();  
  
return result.vendorId;  
}  
  
}  
  
8  
public ActionResult logout()  
{  
    FormsAuthentication.SignOut();  
    return RedirectToAction("login");  
}  
  
}  
}  
  
=====notification=====
```

```
namespace fypPromolac.Controllers

{
    public class notificationController : Controller
    {

        /*IFirebaseConfig config = new FirebaseConfig
        {
            AuthSecret = "XrO8HZD-SsUSId64BI62NgD7SabTXOMA_k6z_kq1ORM",
            BasePath = "https://xdadeveloperes.firebaseio.com/"
        };

        // GET: notification

        public ActionResult notify()
        {

            var id = vendorId(User.Identity.Name);

            notificationModel tempmodel = new notificationModel();
            tempmodel.detail = Getarea(id);
            tempmodel.vendorId = id;
        }
    }
}
```

```
// return View(detail);  
return View(tempmodel);  
  
}  
  
[HttpPost]  
public ActionResult notify(notificationModel n)  
{  
    //string path = uploadimage(notimage);  
    39  
    using (var context = new promoLacDbEntities())  
    {  
        if (ModelState.IsValid)  
        {  
            foreach (var x in n.areaName)  
            {  
                notification not = new notification()  
                {  
                    notificationTitle = n.notificationTitle,  
                    notificationDescription = n.notificationDescription,  
                };  
                context.notifications.Add(not);  
            }  
            context.SaveChanges();  
        }  
    }  
    return View("Success");  
}
```

```
        notificationAreaId = getAreaId(x),  
        notificationStatus = "s",  
        vendorId = n.vendorId,  
        notificationTime = DateTime.Now  
    );  
  
    string towards = getHashCode(not.notificationAreaId);  
  
    context.notifications.Add(not);  
    updateMessagesent();  
  
    context.SaveChanges();  
  
  
  
  
    string SERVER_API_KEY =  
"AAAAtICjlu4:APA91bGGwQ8HSMWLf7dVoD3c1XB3O7IBmhet0im9YCbFuxDF5pVui  
8IBZ8iCXcnUv2K32XVLnoAFwU17CdeYs3SGW6lX01dKeRSXAht8cS7WPg21fU_WIr  
kX8o-1FpWyQjlPxJ_mp8R";  
  
    var SENDER_ID = "783036946158";  
  
    3  
    WebRequest tRequest;  
  
    tRequest =  
    WebRequest.Create("https://fcm.googleapis.com/fcm/send");
```

```
tRequest.Method = "post";
tRequest.ContentType = "application/json";
tRequest.Headers.Add(string.Format("Authorization: key={0}",
SERVER_API_KEY));
tRequest.Headers.Add(string.Format("Sender: id={0}",
SENDER_ID));

var data = new
{
    notification = new
    {
        body = not.notificationDescription,
        title = not.notificationTitle
    }
    ,
    to = "/topics/" + towards
};

33 var serializer = new JavaScriptSerializer();
var json = serializer.Serialize(data);
//logger.Info("json: " + json);
3 Byte[] byteArray = Encoding.UTF8.GetBytes(json);
```

```
tRequest.ContentLength = byteArray.Length;

Stream dataStream = tRequest.GetRequestStream();
dataStream.Write(byteArray, 0, byteArray.Length);
dataStream.Close();

WebResponse tResponse = tRequest.GetResponse();
dataStream = tResponse.GetResponseStream();
StreamReader tReader = new StreamReader(dataStream);
String sResponseFromServer = tReader.ReadToEnd();

tReader.Close();
dataStream.Close();
tResponse.Close();

IFirebaseConfig mes = new FirebaseConfig
{
    AuthSecret =
    "CwnE0VGXMqxNxcY5YSWVNcL7IAM3hfqRuSiNjWm5",
    BasePath = "https://xdadeveloper.es.firebaseio.com/"
}
```

```
};

IFirebaseClient mfc;
mfc = new FireSharp.FirebaseClient(mes);

var set = mfc.Set(@"Messages/" + not.notificationId, not);

}

}

}

}

12 return Json(new { success = true, message = "dddd" }),

JsonRequestBehavior.AllowGet);

}

17 public void updatemessagesent()

{

using (var context= new promoLacDbEntities())

{

17 int id = vendorId(User.Identity.Name);
```

```
var message = context.userPackages.FirstOrDefault(x => x.vendorId == id);

if(message != null)
{
    message.notificationSent++;
}

context.SaveChanges();

}

}

public string uploadimage(HttpPostedFileBase notimage)
{
    Random r = new Random();

    string path = "-1";

    int random = r.Next();

    if (notimage != null && notimage.ContentLength > 0)

    {

```

```
string extension = Path.GetExtension(notimage.FileName);

if (extension.ToLower().Equals(".jpg") ||
extension.ToLower().Equals(".jpeg") || extension.ToLower().Equals(".png"))

{

try

{

6
path = Path.Combine(Server.MapPath("~/Content/images"), random
+ Path.GetFileName(notimage.FileName));

notimage.SaveAs(path);

path = "~/Content/upload/" + random +
Path.GetFileName(notimage.FileName);

}

catch (Exception ex)

{
```

```
31
path = "-1";  
  
}  
  
}  
  
else  
  
{  
  
    Response.Write("<script>alert('Only jpg ,jpeg or png formats are  
22  
acceptable....'); </script>");  
  
}  
  
}  
  
else  
  
{  
  
    Response.Write("<script>alert('Please select a file'); </script>");  
  
    path = "-1";  
  
}
```

```
        return path;  
    }  
  
/* public List<area> Getarea(int id)  
{  
    1  
    using (var context = new userAuthenticationEntities())  
    {  
        var result = context.Areas.Where(x => x.areaId == id).Select(x => new  
area()  
        {  
            areaName = x.areaName  
  
        }).ToList();  
  
        return result;  
    }  
*/
```

```
public List<areaAssignedModel> Getarea(int id)

{

    1 using (var context = new promoLacDbEntities())

    {

        var result = context.areaAssigneds.Where(x => x.vendorId == id).Select(x =>

new areaAssignedModel()

    {

        areald = x.areald

    }).ToList();

        foreach (areaAssignedModel var in result)

    {

        var.areaName = getAreaName(var.areald);

    }

    2 return result;

}

}

public string getAreaName(int id)

{
```

```
using (var context = new promoLacDbEntities())
{
    var result = context.Areas.Where(x => x.areaId == id).Select(x => new
areaModel()
{
    areaName = x.areaName

}).FirstOrDefault();

    return result.areaName;
}
}

public string getHashCode(int id)
{
    using (var context = new promoLacDbEntities())
    {
        var result = context.Areas.Where(x => x.areaId == id).Select(x => new
areaModel()

{
    areaHashCode=x.areaHashCode

}).FirstOrDefault();
    }
}
```

```
        return result.areaHashCode;
    }

}

public int vendorId(string name)
{
    1
    using (var context = new promoLacDbEntities())
    {
        var result = context.vendors.Where(x => x.vendorUserName ==
name).Select(x => new vendorModel()
{
    vendorId = x.vendorId
}).FirstOrDefault();

        return result.vendorId;
    }
}

public int getAreaId(string name)
{
    1
    using (var context = new promoLacDbEntities())

```

```
{  
    var result = context.areas.Where(x => x.areaName == name).Select(x =>  
        new areaModel()  
    {  
        areald = x.areald  
  
    }).FirstOrDefault();  
  
    return result.areald;  
}  
}  
}  
}
```

=====package=====

namespace fypPromolac.Controllers12

```
{  
    public class packageController : Controller  
    {  
        // GET: package  
        public ActionResult packageDetails()  
        {  
            using (var context = new promoLacDbEntities())  
            {  
                int id = vendorId(User.Identity.Name);  
                packageModel details = new packageModel();  
                var subresult = context.userPackages.Where(x => x.vendorId == id).Select(y  
                => new packageModel())  
                {  
                    packagesId=y.packageId  
  
                }).FirstOrDefault();  
                //details.detail = Getarea(1);  
                var result= context.packages.Where(x => x.packagesId ==  
                subresult.packagesId).Select(x => new packageModel())  
                {  
                    packageName=x.packageName,  
                    packageDurationDays=x.packageDurationDays,  
                }  
            }  
        }  
    }  
}
```

```
        subUsersAllowed=x.subUsersAllowed,  
        messagesAllowed=x.messagesAllowed,  
        packageDescription=x.packageDescription,  
        fencingHours=x.fencingHours,  
  
    }).FirstOrDefault();  
  
var r = context.userPackages.Where(x => x.vendorId == id).FirstOrDefault();  
  
result.endtime = r.packageEndTime;  
  
return View(result);  
}  
  
}  
  
public int vendorId(string name)  
{  
  
    1   using (var context = new promoLacDbEntities())  
    {  
        var result = context.vendors.Where(x => x.vendorUserName ==  
name).Select(x => new vendorModel())
```

```
{  
    vendorId = x.vendorId  
  
}).FirstOrDefault();  
  
return result.vendorId;  
}  
  
}  
  
public List<areaAssignedModel> Getarea(int id)  
  
{  
    1   using (var context = new promoLacDbEntities())  
    {  
        var result = context.areaAssigneds.Where(x => x.vendorId == id).Select(x =>  
new areaAssignedModel()  
        {  
            areaid = x.areaid  
  
        }).ToList();  
        foreach (areaAssignedModel var in result)  
        {  
    }
```

```
        var.areaName = getAreaName(var.areald);

    }

2
return result;

}

}

public string getAreaName(int id)

{

    using (var context = new promoLacDbEntities())

    {

        var result = context.Areas.Where(x => x.areald == id).Select(x => new

areaModel()

        {

            areaName = x.areaName

        }).FirstOrDefault();

        return result.areaName;
    }
}
```

```
}
```

```
=====UserInfoes=====
```

```
namespace fypPromolac.Controllers
```

```
{
```

```
    [Authorize]
```

```
    28
```

```
    public class userInfoesController : Controller
```

```
{
```

```
    private promoLacDbEntities db = new promoLacDbEntities();
```

```
// GET: userInfoes
```

```
    public ActionResult Index()
```

```
{
```

```
    return View(db.vendors.ToList());
```

```
}
```

```
/*=====GET MEMBERS UNDER VENDOR=====*/
```

```
    public ActionResult memberdetails()
```

```
{
```

```
    /*if (id == null)
```

```
{
```

```
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
```

```
    }

    6
    userInfo userInfo = db.userInfo.Find(1);

    if (userInfo == null)

    {

        return HttpNotFound();

    }

    return View(userInfo);*/



using (var context = new promoLacDbEntities())

{

    28
    int id = vendorId(User.Identity.Name);

    // subUserModel details = new subUserModel();

    6
    var myresult = context.vendors.Where(x => x.headId == id).Select(x => new

userPackageModel()

    {

        vendorId = x.vendorId,

        userName=x.vendorUserName

    }).ToList();




foreach (userPackageModel x in myresult)

{
```

```
        var t = context.userPackages.Where(z => z.vendorId ==  
x.vendorId).FirstOrDefault();  
  
        x.packageId = t.packageId;  
        x.dividedShareOfMessages = t.dividedShareOfMessages;  
        x.packageStartTime = t.packageStartTime;  
        x.packageEndTime = t.packageEndTime;  
        x.packageStatus = t.packageStatus;  
        x.notificationSent = t.notificationSent;  
        x.detail = Getarea(x.vendorId);  
  
    }  
    return View(myresult);  
  
}  
}  
public ActionResult changePassword()  
{  
    using (var context=new promoLacDbEntities())  
    {
```

```
10
var myresult = context.vendors.Where(x => x.vendorUserName ==
User.Identity.Name).Select(x => new passwordChange()
{
    password = x.vendorPassword

}).FirstOrDefault();

ViewBag.password = myresult.password;
return View();
}

}

public ActionResult notificationHistory(int id)
{
    var context = new promoLacDbEntities();
    var result = context.notifications.Where(x => x.vendorId == id).Select(y => new
notificationModel() {
        notificationId = y.notificationId,
        notificationDescription = y.notificationDescription,
        notificationTitle = y.notificationTitle,
        notificationAreaId=y.notificationAreaId,
        vendorId=Convert.ToInt32(y.vendorId),
        notificationTime=y.notificationTime
    }
}
```

```
}).ToList();

foreach(var x in result)
{
    x.areas = getAreaName(x.notificationAreaId);
    x.userName = getUserName(Convert.ToInt32(x.vendorId));
}
return View(result);
}
```

```
[HttpPost]
public ActionResult changePassword(passwordChange p)
{
    7
    using (var context = new promoLacDbEntities())
    {
        var myresult =
            context.vendors.Where(x=>x.vendorUserName==User.Identity.Name).First();
        myresult.vendorPassword = p.newPassword;
    }
}
```

```
        context.SaveChanges();

        return RedirectToAction("../ControlPanel/ControlPanel");

    }

}

/* public int mesallowed(int id)
{
    using (var context = new userAuthenticationEntities())
    {
        var result1 = context.messages.Where(x => x.messageid == id).Select(x =>
new message())
        {

            m_allowed = x.m_allowed

        }).FirstOrDefault();

        return result1.m_allowed;
    }
}*/
```

```
//get vendorId through vendor name from vendor table

public int vendorId (string name){

    1   using (var context = new promoLacDbEntities())

    {
        var result = context.vendors.Where(x => x.vendorUserName ==
name).Select(x => new vendorModel()

        {
            vendorId=x.vendorId

        }).FirstOrDefault();

        return result.vendorId;
    }

}

public List<areaAssignedModel> Getarea(int id)

{
    1   using (var context = new promoLacDbEntities())
```

```
{  
    var result = context.areaAssingeds.Where(x => x.vendorId == id).Select(x =>  
        new areaAssignedModel()  
    {  
        areald=x.areald  
  
    }).ToList();  
    foreach(areaAssignedModel var in result)  
    {  
        var.areaName = getAreaName(var.areald);  
    }  
    return result;  
}  
  
}  
  
/*=====Return Area Name to function Get Area =====*/  
  
public string getAreaName(int id)  
{  
    1 using (var context = new promoLacDbEntities())  
    {
```

```
var result = context.Areas.Where(x => x.AreaId == id).Select(x => new
areaModel()

{
    areaName = x.areaName

}).FirstOrDefault();

return result.areaName;
}

}

public int getAreaId(string name)
{
    using (var context = new promoLacDbEntities())
    {
        var result = context.Areas.Where(x => x.areaName == name).Select(x =>
new areaModel()

{
    areaId = x.areaId

}).FirstOrDefault();

        return result.areaId;
    }
}
```

```
}

/*=====Return Messages Allowed to Package Details
function=====
public int messagesAllowed(int z)

{
    1
    using (var context = new promoLacDbEntities())

    {
        var result = context.packages.Where(x => x.packagesId == z).Select(x =>
new packageModel()

    {
        messagesAllowed = x.messagesAllowed

    }).FirstOrDefault();

    return result.messagesAllowed;

}

}
```

```
/*=====Return Messages Sent to Package Details

function=====
public int messagesSent(int z)

{
    1
    using (var context = new promoLacDbEntities())

    {
        var result = context.userPackages.Where(x => x.vendorId == z).Select(x =>
new userPackageModel()

        {

            notificationSent=x.notificationSent


        }).FirstOrDefault();

        return result.notificationSent;

    }

}

// GET: userInfoes/Details/5

/*=====Provide list of areas to be added for Subuser under
Vendor=====*/

```

```
public ActionResult Addarea(int id)

{
    areaAssignedModel obj = new areaAssignedModel();
    using (var context = new promoLacDbEntities())
    {

        obj.detail = Getarea(vendorId(User.Identity.Name));
        obj.vendorId = id;

        8
        return View(obj);
    }

}

[HttpPost]
public ActionResult Addarea(areaAssignedModel u)
{
    using (var context = new promoLacDbEntities())
    {
        areaAssigned a = new areaAssigned()
        {
```

```
        vendorId=u.vendorId,  
  
        areaId=getAreaId(u.areaName)  
  
    };  
  
    context.areaAssigneds.Add(a);  
    16  
    context.SaveChanges();  
  
}  
  
return RedirectToAction("../controlPanel/controlPanel");  
  
}  
  
// GET: userInfoes/Create  
  
public ActionResult Create()  
  
{  
  
    var context = new promoLacDbEntities();  
  
    int id = vendorId(User.Identity.Name);  
  
    var result2 = context.userPackages.Where(x => x.vendorId ==  
id).FirstOrDefault();  
  
    vendorModel v = new vendorModel();  
  
    var detail = context.areaAssigneds.Where(x => x.vendorId == id).Select(y=>  
new areaAssignedModel()  
  
{
```

```
areald=y.areald  
}).ToList();  
  
foreach(var z in detail)  
{  
    z.areaName = getAreaName(z.areald);  
}  
v.detail = detail;  
v.messagesStock = result2.dividedShareOfMessages;  
8  
return View(v);  
}
```

```
[HttpPost]  
[ValidateAntiForgeryToken]  
public ActionResult Create(vendorModel user)  
{  
  
    using (var context = new promoLacDbEntities())  
    {
```

```
{  
    vendor info = new vendor()  
    {  
        59  
        firstName = user.firstName,  
        lastName = user.lastName,  
        vendorEmail = user.vendorEmail,  
        vendorPassword = user.vendorPassword,  
        registerTimestamp = DateTime.Now,  
        phoneNumber = user.phoneNumber,  
        vendorUserName = user.vendorUserName,  
        headId = vendorId(User.Identity.Name),  
        isHead = "N",  
        vendorStatus = "A",  
        vendorAddress = "MC 216 C GREEN TOWN",  
        vendorCompanyName = "FAST NUCES"  
  
    };  
  
    context.vendors.Add(info);  
    var result = context.userPackages.Where(x => x.vendorId ==  
        info.headId).FirstOrDefault();  
    userPackage u = new userPackage();
```

```
        u.vendorId = user.vendorId;
        u.packageStartTime = result.packageStartTime;
        u.packageEndTime = result.packageEndTime;
        u.packageId = result.packageId;
        u.dividedShareOfMessages = user.messagesAllowed;
        u.packageStatus = "active";
        u.notificationSent = 0;
        context.userPackages.Add(u);
        context.SaveChanges();

        foreach (var x in user.areas)
        {
            areaAssigned ar = new areaAssigned()
            {
                vendorId = vendorId(user.vendorUserName),
                areaId = Convert.ToInt32(x)

            };
            context.areaAssigneds.Add(ar);
            context.SaveChanges();

        }
    }
```

```
var result1 = context.userPackages.Where(x => x.vendorId ==  
info.headId).FirstOrDefault();  
  
result1.dividedShareOfMessages = result1.dividedShareOfMessages -  
user.messagesAllowed;  
6  
context.SaveChanges();  
  
}  
  
}  
  
}  
  
return RedirectToAction("../ControlPanel/ControlPanel");  
}  
  
}  
  
}  
  
// GET: userInfoes/Edit/5  
10  
public JsonResult IsUserNameAvailable(string vendorUserName)  
{  
  
return Json(!db.vendors.Any(u => u.vendorUserName == vendorUserName),  
JsonRequestBehavior.AllowGet);  
  
}  
21  
public ActionResult Edit(int? id)
```

```
{  
    if (id == null)  
    {  
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  
    }  
    //userInfo userInfo = db.userInfo.Find(id);  
    using (var context = new promoLacDbEntities())  
    {  
        //subUserModel details = new subUserModel();  
        var myresult = context.vendors.Where(x => x.vendorId==id).Select(x => new  
        vendorModel()  
        {  
  
            vendorUserName=x.vendorUserName,  
            headId=x.headId,  
            vendorId=x.vendorId,  
            vendorPassword=x.vendorPassword  
  
        }).FirstOrDefault();  
  
        myresult.detail = GetareaSubUser(myresult.vendorId);  
    }  
}
```

```
//myresult.detail = Getarea(myresult.userId);

return View(myresult);

}

//return View(userInfo);

}

public List<areaAssignedModel> GetareaSubUser(int id)

{

    1
    using (var context = new promoLacDbEntities())

    {

        var result = context.areaAssigneds.Where(x =>x.vendorId== id).Select(x =>

new areaAssignedModel()

        {

            areaid = x.areaid

        }).ToList();

    }

}
```

```
foreach (areaAssignedModel var in result)
{
    var.areaName = getAreaName(var.areald);
}

return result;
}

/* public ActionResult Edit(user u)
{
    using (var context = new userAuthenticationEntities())
    {
        user details = new user();
        var myresult = context.userInfo.Where(x => x.userAdmin ==
User.Identity.Name).Select(x => new user()
{
```

```
userId = x.userId,  
userName = x.userName,  
}).ToList();  
  
foreach (user x in myresult)  
{  
    x.detail = Getarea(x.userId);  
    x.m_Allowed = mesallowed(x.userId);  
    x.m_sent = messent(x.userId);  
}  
16  
return View(myresult);
```

```
public ActionResult RemoveArea(string areaa,int id )  
{  
    using (var context = new promoLacDbEntities())  
    {  
        int area_Id = getAreaId(areaa);  
        var myresult =  
            context.areaAssigneds.Where(x=>(x.areaid==area_Id)&&(x.vendorId==id)).First();  
        //Where(o=>(o.areaid == id )&&(o.areaName==areaa)).SingleOrDefault();  
  
        context.areaAssigneds.Remove(myresult);  
        context.SaveChanges();  
        return RedirectToAction("../ControlPanel/ControlPanel");  
    }  
}  
2  
public string.getUserName(int id)  
{  
    var context = new promoLacDbEntities();  
    var result = context.vendors.Where(x => x.vendorId == id).FirstOrDefault();
```

```
        return result.vendorUserName;

    }

// GET: userInfoes/Delete/56
```



```
protected override void Dispose(bool disposing)

{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
```



```
}
```

AdminSide

=====deal=====

```
namespace fypPromolacAdmin.Controllers
```

```
{  
    16  
    public class dealsController : Controller  
  
    {  
  
        // GET: deals  
  
        public ActionResult createDeal()  
  
        {  
  
            var context = new promoLacDbEntities();  
  
            var y = context.dealCategories.ToList();  
  
            dealsModel x = new dealsModel();
```

```
x.dealCategoryLists = y;

return View(x);
}

[HttpPost]
public ActionResult createDeal(dealsModel d, HttpPostedFileBase dealsImage)
{
    string path = uploadimage(dealsImage);

    var context = new promoLacDbEntities();
    deal obj = new deal();

    obj.adminId = getAdminId(User.Identity.Name);

    HttpClient httpClient = new HttpClient();
    string baseUrl = "https://localhost:44371/";
```

```
string path1 =  
"C:\\\\Users\\\\DELL\\\\source\\\\repos\\\\fypPromolac\\\\fypPromolac\\\\Content\\\\images\\\\537570  
52inam-picture.jpeg";  
  
HttpResponseMessage response = httpClient.GetAsync(baseUrl +  
"api2/fileFirebase/GetFileUpload?id=" + path + "&vId=" + obj.adminId + "&fileName=" +  
dealsImage.FileName).Result;  
  
string stateInfo = response.Content.ReadAsStringAsync().Result;  
  
stateInfo = stateInfo.Substring(1, stateInfo.Count() - 2);  
  
obj.dealCategory = getDealCategoryId(d.dealCategory);  
obj.dealDescription = d.dealDescription;  
obj.dealName = d.dealName;  
obj.dealPrice = d.dealPrice;  
obj.latitude = d.latitude;  
obj.longitude = d.longitude;  
obj.deallImage = path;  
obj.area = d.AreaName;  
obj.dealTime = DateTime.Now.Date;
```

```
context.deals.Add(obj);
context.SaveChanges();
tempDeal ddd = new tempDeal();

ddd.dealimg = stateInfo;
ddd.dealprice = d.dealPrice.ToString();
ddd.latitude = d.latitude.ToString();
ddd.longitude = d.longitude.ToString();
ddd.rating = 2.ToString();
ddd.dealtitle = d.dealName;
IFirebaseConfig mes = new FirebaseConfig
{
    AuthSecret = "CwnE0VGXMqxNxcY5YSWVNcL7IAM3hfqRuSiNjWm5",
    BasePath = "https://xdadeveloper.es.firebaseio.com/"
};
IFirebaseClient mfc;
mfc = new FireSharp.FirebaseClient(mes);
var data = ddd;
```

```
PushResponse r = mfc.Push(d.dealCategory + "/", data);

6
return Json(new { success = true, message = "dddd" },
JsonRequestBehavior.AllowGet);

}

public int getDealCategoryId(string name)
{
    var context = new promoLacDbEntities();
    var y = context.dealCategories.Where(x => x.categoryName ==
name).FirstOrDefault();
    return y.categoryId;
}

public int getAdminId(string name)
{
    var context = new promoLacDbEntities();
    var y = context.mainAdmins.Where(x => x.username ==
name).FirstOrDefault();
    return y.id;
}

public string uploadimage(HttpPostedFileBase dealsImage)
```

```
{  
  
    Random r = new Random();  
  
    string path = "-1";  
  
    int random = r.Next();  
  
    if (dealsImage != null && dealsImage.ContentLength > 0)  
  
    {  
        string extension = Path.GetExtension(dealsImage.FileName);  
  
        if (extension.ToLower().Equals(".jpg") || extension.ToLower().Equals(".jpeg")  
        || extension.ToLower().Equals(".png"))  
  
        {  
            try  
  
            {  
                6  
                path = Path.Combine(Server.MapPath("~/Content/images"), random +  
                Path.GetFileName(dealsImage.FileName));  
            }  
        }  
    }  
}
```

```
    dealsImage.SaveAs(path);

    path =
"C:/Users/DELL/source/repos/fypPromolac/fypPromolac/Content/images/" + random +
Path.GetFileName(dealsImage.FileName);

}

catch (Exception ex)

{

    path = "-1";31

}

}

else

{
```

```
        Response.Write("<script>alert('Only jpg ,jpeg or png formats are  
acceptable....'); </script>");  
  
    }  
  
}  
  
else  
  
{  
    Response.Write("<script>alert('Please select a file'); </script>");  
  
    path = "-1";  
}  
  
return path;  
}  
}  
}  
  
=====fence=====
```

```
namespace fypPromolacAdmin.Controllers

{

    public class fenceController : Controller

    {

        // GET: fence

        public ActionResult createFenceAdmin()

        {

            int id = adminId(User.Identity.Name);

            fenceModel m = new fenceModel();

4            m.adminId = id;

            return View(m);

        }

        [HttpPost]

        public ActionResult createFenceAdmin(fenceModel f)

        {

            tempFence tf = new tempFence();

            tf.Latitude = f.latitude.ToString();

            tf.Longitude = f.longitude.ToString();

            tf.Notification_ = f.notification;
```

```
tf.TimeStamp = DateTime.Now.AddHours(f.hr).ToString("MM/dd/yyyy  
HH:mm:ss");  
  
tf.Radius = f.radius.ToString();  
  
IFirebaseConfig mes = new FirebaseConfig  
{  
    AuthSecret = "CwnE0VGXMqxNxcY5YSWVNcL7IAM3hfqRuSiNjWm5",  
    BasePath = "https://xdadeveloperes.firebaseio.com/"  
};  
  
IFirebaseClient mfc;  
mfc = new FireSharp.FirebaseClient(mes);  
var data = tf;  
PushResponse r = mfc.Push("Fences_" + "/", data);  
  
//updateFencingHours(Convert.ToInt32(f.vendorId), f.hr);  
  
var context = new promoLacDbEntities();  
fence fence_ = new fence();  
fence_.latitude = f.latitude;  
fence_.longitude = f.longitude;  
fence_.startTime = DateTime.Now;  
fence_.endTime = DateTime.Now.AddHours(f.hr);
```

```
fence_.notification = f.notification;
fence_.radius = f.radius;
fence_.adminId = f.adminId;
fence_.area = f.Area;

context.fences.Add(fence_);
12
context.SaveChanges();

return Json(new { success = true, message = "dddd" },
JsonRequestBehavior.AllowGet);

}

/* public void updateFencingHours(int id, int hr)
{
    int v_id = getVendorStatus(id);
7
var context = new promoLacDbEntities();
var result = context.userPackages.Where(x => x.vendorId ==
v_id).FirstOrDefault();
result.remainingFencingHours = result.remainingFencingHours - hr;
context.SaveChanges();
} */
/* public int getVendorStatus(int id)
```

```
{  
    var context = new promoLacDbEntities();  
  
    var result = context.vendors.Where(x => x.vendorId == id).FirstOrDefault();  
  
    if (result.isHead == "Y")  
    {  
        return result.vendorId;  
    }  
    else  
    {  
        id = Convert.ToInt32(result.headId);  
        return id;  
    }  
}  
*/  
  
public int adminId(string name)  
{  
  
    1 using (var context = new promoLacDbEntities())  
    {  
        var result = context.mainAdmins.Where(x => x.username == name).Select(x  
=> new adminModel()  
        {  
            id=x.id  
        };  
    }  
}
```

```
        }).FirstOrDefault();  
  
        return result.id;  
    }  
  
}  
}  
}
```

=====home=====

```
19  
namespace fypPromolacAdmin.Controllers  
{  
    public class fenceController : Controller  
    {  
        // GET: fence  
        public ActionResult createFenceAdmin()  
        {  
            int id = adminId(User.Identity.Name);  
  
            fenceModel m = new fenceModel();  
            4  
            m.adminId = id;
```

```
        return View(m);
    }

    [HttpPost]
    public ActionResult createFenceAdmin(fenceModel f)
    {

        tempFence tf = new tempFence();
        tf.Latitude = f.latitude.ToString();
        tf.Longitude = f.longitude.ToString();
        tf.Notification_ = f.notification;
        tf.TimeStamp = DateTime.Now.AddHours(f.hr).ToString("MM/dd/yyyy
HH:mm:ss");

        tf.Radius = f.radius.ToString();
        IFirebaseConfig mes = new FirebaseConfig
        {
            AuthSecret = "CwnE0VGXMqxNxcY5YSWVNcL7IAM3hfqRuSiNjWm5",
            BasePath = "https://xdadeveloperes.firebaseio.com/"
        };

        IFirebaseClient mfc;
        mfc = new FireSharp.FirebaseClient(mes);
```

```
var data = tf;

PushResponse r = mfc.Push("Fences_" + "/", data);

//updateFencingHours(Convert.ToInt32(f.vendorId), f.hr);

var context = new promoLacDbEntities();

fence fence_ = new fence();

fence_.latitude = f.latitude;

fence_.longitude = f.longitude;

fence_.startTime = DateTime.Now;

fence_.endTime = DateTime.Now.AddHours(f.hr);

fence_.notification = f.notification;

fence_.radius = f.radius;

fence_.adminId = f.adminId;

fence_.area = f.Area;

context.fences.Add(fence_);

12
context.SaveChanges();

return Json(new { success = true, message = "dddd" },
JsonRequestBehavior.AllowGet);

}
```

```
/* public void updateFencingHours(int id, int hr)

{
    int v_id = getVendorStatus(id);
    7
    var context = new promoLacDbEntities();

    var result = context.userPackages.Where(x => x.vendorId ==
v_id).FirstOrDefault();

    result.remainingFencingHours = result.remainingFencingHours - hr;
    context.SaveChanges();

}*/ /* public int getVendorStatus(int id)

{
    var context = new promoLacDbEntities();

    var result = context.vendors.Where(x => x.vendorId == id).FirstOrDefault();
    if (result.isHead == "Y")
    {
        return result.vendorId;
    }
    else
    {
        id = Convert.ToInt32(result.headId);
        return id;
    }
}
```

```
 */  
  
public int adminId(string name)  
{  
  
    1   using (var context = new promoLacDbEntities())  
    {  
  
        var result = context.mainAdmins.Where(x => x.username == name).Select(x  
        => new adminModel()  
        {  
  
            id=x.id  
  
        }).FirstOrDefault();  
  
        return result.id;  
    }  
  
}  
}  
  
=====login=====
```

6
namespace fypPromolacAdmin.Controllers

```
{  
  
    public class loginController : Controller  
  
    {  
  
        // GET: login  
        [8]  
        public ActionResult login()  
  
        {  
  
            return View();  
        }  
  
        [HttpPost]  
  
        public ActionResult login(adminModel model)  
  
        {  
  
            using (var context = new promoLacDbEntities())  
  
            {  
  
                bool isvalidVendor = context.mainAdmins.Any(x => x.username ==  
model.username && x.password_ == model.password_);  
  
                if (isvalidVendor)  
  
                {  
  
                    FormsAuthentication.SetAuthCookie(model.username, false);  
  
                    return RedirectToAction("addVendor", "Vendor");  
                }  
  
                ModelState.AddModelError("", "Invalid User Name and Password");  
  
            }  
        }  
    }  
}
```

```
        return View();
    }
}

=====notification=====

namespace fypPromolacAdmin.Controllers
{
    public class notificationController : Controller
    {

        /*IFirebaseConfig config = new FirebaseConfig
        {
            AuthSecret = "XrO8HZD-SsUSId64BI62NgD7SabTXOMA_k6z_kq1ORM",
            BasePath = "https://xdadeveloperes.firebaseio.com/"

        };*/
        // GET: notification

        public ActionResult notify()
        {

```

```
var id = adminId(User.Identity.Name);

notificationModel tempmodel = new notificationModel();

tempmodel.detail = Getarea();
2
tempmodel.adminId = id;
// return View(detail);

return View(tempmodel);

}
```

```
[HttpPost]

public ActionResult notify(notificationModel n)

{
    //string path = uploadimage(notimage);
39
    using (var context = new promoLacDbEntities())

    {
        if (ModelState.IsValid)
        {
            {
```

```
foreach (var x in n.areaName)
{
    notification not = new notification()
    {

        notificationTitle = n.notificationTitle,
        notificationDescription = n.notificationDescription,
        notificationAreaId = getAreaId(x),
        notificationStatus = "s",
        adminId=n.adminId,
        notificationTime = DateTime.Now
    };
    string towards = getHashCode(not.notificationAreaId);

    context.notifications.Add(not);

    context.SaveChanges();

    string SERVER_API_KEY =
"AAAAAtICjlu4:APA91bGGwQ8HSMWLf7dVoD3c1XB3O7IBmhet0im9YCbFuxDF5pVui
```

```
8IBZ8iCXcnUv2K32XVLnoAFwU17CdeYs3SGW6lX01dKeRSXAht8cS7WPg21fU_WIr
kX8o-1FpWyQjlPxFj_mp8R";
var SENDER_ID = "783036946158";

3
WebRequest tRequest;
tRequest =
WebRequest.Create("https://fcm.googleapis.com/fcm/send");

tRequest.Method = "post";
tRequest.ContentType = "application/json";
tRequest.Headers.Add(string.Format("Authorization: key={0}",
SERVER_API_KEY));
tRequest.Headers.Add(string.Format("Sender: id={0}",
SENDER_ID));

var data = new
{
notification = new
{
body = not.notificationDescription,
title = not.notificationTitle
}
}
```

```
        ,  
        to = "/topics/" + towards  
    };  
  
    33  
    var serializer = new JavaScriptSerializer();  
  
    var json = serializer.Serialize(data);  
  
    //logger.Info("json: " + json);  
    3  
    Byte[] byteArray = Encoding.UTF8.GetBytes(json);  
  
    tRequest.ContentLength = byteArray.Length;  
  
    Stream dataStream = tRequest.GetRequestStream();  
  
    dataStream.Write(byteArray, 0, byteArray.Length);  
  
    dataStream.Close();  
  
    WebResponse tResponse = tRequest.GetResponse();  
  
    dataStream = tResponse.GetResponseStream();  
  
    StreamReader tReader = new StreamReader(dataStream);  
  
    String sResponseFromServer = tReader.ReadToEnd();  
  
    tReader.Close();  
  
    dataStream.Close();  
  
    tResponse.Close();
```

```
IFirebaseConfig mes = new FirebaseConfig
{
    AuthSecret =
    "CwnE0VGXMqxNxcY5YSWVNcL7IAM3hfqRuSiNjWm5",
    BasePath = "https://xdadeveloper.es.firebaseio.com/"
};

IFirebaseClient mfc;
mfc = new FireSharp.FirebaseClient(mes);

var set = mfc.Set(@"Messages/" + not.notificationId, not);

}

}

}

}
```

```
12     return Json(new { success = true, message = "dddd" },  
13     JsonRequestBehavior.AllowGet);  
14 }  
15  
16 public void updatemessagesent()  
17 {  
18     using (var context = new promoLacDbEntities())  
19     {  
20         int id = adminId(User.Identity.Name);  
21  
22         var message = context.userPackages.FirstOrDefault(x => x.vendorId == id);  
23  
24         if (message != null)  
25         {  
26             message.notificationSent++;  
27         }  
28         context.SaveChanges();  
29     }  
30 }  
31  
32 public string uploadimage(HttpPostedFileBase notimage)  
33 {
```

```
Random r = new Random();

string path = "-1";

int random = r.Next();

if (notimage != null && notimage.ContentLength > 0)

{

    string extension = Path.GetExtension(notimage.FileName);

    if (extension.ToLower().Equals(".jpg") || extension.ToLower().Equals(".jpeg")

|| extension.ToLower().Equals(".png"))

    {

        try

        {

            6
            path = Path.Combine(Server.MapPath("~/Content/images"), random +

Path.GetFileName(notimage.FileName));

            notimage.SaveAs(path);

        }
    }
}
```

```
path = "~/Content/upload/" + random +  
Path.GetFileName(notimage.FileName);  
  
}  
  
catch (Exception ex)  
  
{  
    path = "-1";  
  
}  
  
}  
  
else  
  
{  
  
    Response.Write("<script>alert('Only jpg ,jpeg or png formats are  
acceptable....'); </script>");  
  
}  
}
```

```
        }

        else
        {

            Response.Write("<script>alert('Please select a file'); </script>");

            path = "-1";

        }

        return path;
    }

/* public List<area> Getarea(int id)
{
    1
    using (var context = new userAuthenticationEntities())
    {
        var result = context.areas.Where(x => x.areaid == id).Select(x => new
area()

    }

    areaName = x.areaName

}).ToList();
}
```

```
        return result;  
    }  
  
    }*/  
  
public List<area> Getarea()  
  
{  
    7 using (var context = new promoLacDbEntities())  
    {  
        var result = context.Areas.ToList();  
  
        2  
        return result;  
    }  
  
}  
  
public string getAreaName(int id)
```

```
{  
    using (var context = new promoLacDbEntities())  
    {  
        var result = context.Areas.Where(x => x.areaId == id).Select(x => new  
areaModel()  
        {  
            areaName = x.areaName  
  
        }).FirstOrDefault();  
  
        return result.areaName;  
    }  
}  
public string getHashCode(int id)  
{  
    1  
    using (var context = new promoLacDbEntities())  
    {  
        var result = context.Areas.Where(x => x.areaId == id).Select(x => new  
areaModel()  
        {  
            areaHashCode = x.areaHashCode  
  
        }).FirstOrDefault();  
    }  
}
```

```
        return result.areaHashCode;
    }

}

public int adminId(string name)
{
    1 using (var context = new promoLacDbEntities())
    {
        var result = context.mainAdmins.Where(x => x.username == name).Select(x
=> new adminModel()
{
    id = x.id

}).FirstOrDefault();

        return result.id;
    }
}

public int getAreaId(string name)
{
```

```
    }

    using (var context = new promoLacDbEntities())
    {
        var result = context.Areas.Where(x => x.areaName == name).Select(x =>
new areaModel()
{
    areald = x.areald

}).FirstOrDefault();

        return result.areald;
    }
}

}

}

=====package=====
```

12

```
namespace fypPromolacAdmin.Controllers

{

    public class packageController : Controller

    {

        // GET: package

        public ActionResult viewAllPackage()

        {

            using (var context = new promoLacDbEntities())

            {

                List <packageModel> package_List= new List<packageModel>();

                package_List = context.packages.Select(x=> new packageModel

                {

                    packageName=x.packageName,

                    packagesId=x.packagesId,

                    packageDescription=x.packageDescription,

                    packageDurationDays=x.packageDurationDays,

                    messagesAllowed=x.messagesAllowed,

                    subUsersAllowed=x.subUsersAllowed,

                    fencingHours=x.fencingHours

                }).ToList();  
17
```

```
        return View(package_List);
    }

}

public ActionResult editPackage(int id)
{
    var context = new promoLacDbEntities();
    var package=context.packages.Where(x=>x.packagesId==id).Select(x=> new
packageModel
    {
        packageName=x.packageName,
        packagesId=x.packagesId,
        packageDescription=x.packageDescription,
        subUsersAllowed=x.subUsersAllowed,
        packageDurationDays=x.packageDurationDays,
        fencingHours=x.fencingHours
    }).FirstOrDefault();

    return View(package) ;
}
```

```
[HttpPost]  
  
public ActionResult editPackage(packageModel pck)  
{  
    7 using (var context = new promoLacDbEntities())  
    {  
  
        var myresult = context.packages.Where(x => x.packagesId ==  
pck.packagesId).First();  
  
        myresult.packageName = pck.packageName;  
        myresult.packageDescription = pck.packageDescription;  
        myresult.messagesAllowed = pck.messagesAllowed;  
        myresult.packageDurationDays = pck.packageDurationDays;  
        myresult.subUsersAllowed = pck.subUsersAllowed;  
        myresult.fencingHours = pck.fencingHours;  
        context.SaveChanges();  
        List<packageModel> package_List = new List<packageModel>();  
        package_List = context.packages.Select(x => new packageModel  
        {  
            packageName = x.packageName,  
            packagesId = x.packagesId,  
            packageDescription = x.packageDescription,  
            packageDurationDays = x.packageDurationDays,  
        }).ToList();  
        return View(package_List);  
    }  
}
```

```
        messagesAllowed = x.messagesAllowed,  
        subUsersAllowed = x.subUsersAllowed,  
        fencingHours=x.fencingHours  
    }).ToList();  
  
    return View("viewAllPackage", package_List);  
  
}  
}  
8  
public ActionResult createPackage()  
{  
    return View();  
}  
[HttpPost]  
public ActionResult createPackage(packageModel pckg)  
{  
    using (var context= new promoLacDbEntities())  
    {  
        package p = new package();  
        p.packageName = pckg.packageName;  
        p.packageDescription = pckg.packageDescription;  
        p.packageDurationDays = pckg.packageDurationDays;  
        p.subUsersAllowed = pckg.subUsersAllowed;  
    }  
}
```

```
p.messagesAllowed = pckg.messagesAllowed;
p.fencingHours = pckg.fencingHours;

context.packages.Add(p);
context.SaveChanges();

/* List<packageModel> package_List = new List<packageModel>();
package_List = context.packages.Select(x => new packageModel
{
    packageName = x.packageName,
    packagesId = x.packagesId,
    packageDescription = x.packageDescription,
    packageDurationDays = x.packageDurationDays,
    messagesAllowed = x.messagesAllowed,
    subUsersAllowed = x.subUsersAllowed
}).ToList();*/
return RedirectToAction("~/Vendor/viewAllVendors");
}

}

}
```

```
=====vendor=====
```

```
23  
using System;
```

```
using System.Collections.Generic;
```

```
using System.Data;
```

```
using System.Data.Entity;
```

```
using System.IO;
```

```
using System.Linq;
```

```
using System.Net;
```

```
using System.Security.Cryptography;
```

```
using System.Threading.Tasks;
```

```
using System.Web;
```

```
using System.Web.Mvc;
```

```
using fypPromolacAdmin.Models;
```

```
namespace fypPromolacAdmin.Controllers
```

```
{
```

```
16  
public class VendorController : Controller
```

```
{
```

```
// GET: Vendor

private promoLacDbEntities db = new promoLacDbEntities();

public ActionResult addVendor()

{
    using (var context = new promoLacDbEntities())
    {
        vendorViewModel vendorObject = new vendorViewModel();
        var areasList = context.areas.Select(x => new areaModel
        {
            areald = x.areald,
            areaName = x.areaName,
            areaHashCode = x.areaHashCode
        }).ToList();
        vendorObject.detail = areasList;
        return View(vendorObject);
    }
}

[HttpPost]
```

```
public ActionResult AddCompleteVendorInfo(vendorViewModel v)

{
    vendorViewModel v1 = TempData["vendorObject"] as vendorViewModel;
    v1.vendorPackageTaken = v.vendorPackageTaken;

    using (var context = new promoLacDbEntities())
    {

        vendor ven = new vendor()
        {
            firstName = v1.firstName,
            lastName = v1.lastName,
            phoneNumber = v1.phoneNumber,
            vendorEmail = v1.vendorEmail,
            vendorAddress = v1.vendorAddress,
            registerTimestamp = DateTime.Now,
            vendorUserName = v1.vendorUserName,
            vendorPassword = v1.vendorPassword,
            vendorStatus = "A",
            vendorAdminId = getAdminId(User.Identity.Name),
            isHead = "Y",
            vendorCompanyName = v1.vendorCompanyName
        };
        context.vendors.Add(ven);
        context.SaveChanges();
    }
}
```

```
};

context.vendors.Add(ven);
context.SaveChanges();

int id = getVendorId(v1.vendorUserName);
addPackage(v.vendorPackageTaken, id);
foreach (var item in v1.area_vendor)
{
    areaAssigned ar = new areaAssigned()
    {
        vendorId = id,
        areaId = Convert.ToInt32(item)
    };
    context.areaAssigneds.Add(ar);
    context.SaveChanges();
}

return RedirectToAction("viewAllVendors");
}
```

```
    }

    public void addPackage(int x,int y)
    {
        userPackage u = new userPackage();
        u.packageId = x;
        u.vendorId = y;

        var context = new promoLacDbEntities();
        var result = context.packages.Where(z => z.packagesId == x).FirstOrDefault();

        u.dividedShareOfMessages = result.messagesAllowed;
        u.notificationSent = 0;
        u.packageStartTime = DateTime.Now;
        u.packageEndTime = DateTime.Now.AddDays(result.packageDurationDays);
        u.packageStatus = "active";
        context.userPackages.Add(u);
        context.SaveChanges();

    }

    [HttpPost]
    public ActionResult addVendor(vendorViewModel v)
```

```
{  
    1   using (var context = new promoLacDbEntities())  
  
    {  
  
        var packagesdetails = context.packages.ToList();  
  
        v.package_details = packagesdetails;  
  
        TempData["vendorObject"] = v;  
  
        return View("AddCompleteVendorInfo", v);  
  
    }  
  
}  
  
public ActionResult viewAllVendors()  
  
{  
    1   using (var context = new promoLacDbEntities())  
  
    {  
  
        var v = context.vendors.Where(x=>x.isHead=="Y").Select(x => new  
vendorViewModel()  
  
        {  
  
            vendorId = x.vendorId,  
  
            firstName = x.firstName,  
  
            lastName = x.lastName,  
  
            phoneNumber = x.phoneNumber,  
  
            vendorEmail = x.vendorEmail,  
  
            vendorAddress = x.vendorAddress,  
    }
```

```
registerTimestamp = x.registerTimestamp,  
vendorUserName = x.vendorUserName,  
vendorCompanyName = x.vendorCompanyName,  
  
}).ToList();  
  
foreach (vendorViewModel x in v)  
{  
    x.vendorpackage = getVendorPackage(x.vendorId);  
    x.vendorAreaDetail = getAreas(x.vendorId);  
    if (x.vendorpackage!=null)  
    {  
        x.packageName = getPackageName(x.vendorpackage.packageId);  
    }  
}  
 TempData["VendorDetails"] = v;  
58  
return View(v);  
}  
}
```

```
public ActionResult vendorTeamDetail(int id)
{
    var context = new promoLacDbEntities();

    var r = context.vendors.Where(v => (v.vendorId == id) || (v.headId ==
id)).Select(x => new vendorViewModel()
{
    vendorId=x.vendorId,
    vendorUserName=x.vendorUserName
}).ToList();

foreach(var item in r)
{
    item.vendorpackage = getVendorPackage(item.vendorId);
    if (item.vendorpackage != null)
    {
        item.packageName = getPackageName(item.vendorpackage.packageId);
    }
}

return View(r);
}
```

```
21
public ActionResult vendorFenceList(int id)

{
    var f= getFenceList(id,'v');

    return View(f);
}

21
public ActionResult vendorDealList(int id)

{
    var f = getDealList(id, 'v');

    return View(f);
}

21
public ActionResult vendorNotificationList(int id)

{
    var f = getNotificationList(id, 'v');

    return View(f);
}

public List<fence> getFenceList(int id,char status)

{
    var context = new promoLacDbEntities();

    if (status == 'v')

    {
        var r = context.fences.Where(x => x.vendorId == id).ToList();
2
        return r;
    }
}
```

```
        else
        {
            var r = context.fences.Where(x => x.adminId == id).ToList();
            return r;
        }

    }

public List<deal> getDealList(int id, char status)
{
    var context = new promoLacDbEntities();
    if (status == 'v')
    {
        var r = context.deals.Where(x => x.vendorId == id).ToList();
        return r;
    }
    else
    {
        var r = context.deals.Where(x => x.adminId == id).ToList();
        return r;
    }
}

public List<notification> getNotificationList(int id, char status)
```

```
{  
    var context = new promoLacDbEntities();  
    if (status == 'v')  
    {  
        var r = context.notifications.Where(x => x.vendorId == id).ToList();  
        return r;  
    }  
    else  
    {  
        var r = context.notifications.Where(x => x.adminId == id).ToList();  
        return r;  
    }  
  
}  
  
/*50*/  
public ActionResult vendorDetail(int id)  
{  
    if (id.Equals(null))  
    {  
        viewAllVendors();  
        return View();  
    }  
    else
```

```
{  
  
    using (var context = new promoLacDbEntities())  
    {  
        var v = context.vendors.Where(y=>y.vendorId==id).Select(x => new  
        vendorViewModel()  
        {  
            vendorId = x.vendorId,  
            firstName = x.firstName,  
            lastName = x.lastName,  
            phoneNumber = x.phoneNumber,  
            vendorEmail = x.vendorEmail,  
            vendorAddress = x.vendorAddress,  
            registerTimestamp = x.registerTimestamp,  
            vendorUserName = x.vendorUserName,  
            vendorCompanyName = x.vendorCompanyName,  
  
        }).FirstOrDefault();  
        v.vendorpackage = getVendorPackage(v.vendorId);  
        v.messagesSent = getMessagesSent(v.vendorId);  
        v.vendorAreaDetail = getAreas(v.vendorId);  
    }  
}
```

```
v.messagesAllowed = getMessagesAllowed(v.vendorpackage.packageId);

int count = v.messagesSent;

count = count + v.vendorpackage.notificationSent;

v.messagesSent = count;

TempData["VendorDetails"] = v;
49
return View(v);

}

}

}*/



public ActionResult EditVendor(int id)

{

if (id.Equals(null))

{

viewAllVendors();

return View();

}

else

{
```

```
var context = new promoLacDbEntities();

// List<vendorViewModel> V = TempData["VendorDetails"] as
List<vendorViewModel>;
//TempData["VendorDetails"] = V;
vendorViewModel z = new vendorViewModel();
z = context.vendors.Where(x => x.vendorId == id).Select(y => new
vendorViewModel
{
    vendorId = y.vendorId,
    vendorEmail = y.vendorEmail,
    vendorUserName = y.vendorUserName,
    vendorAddress = y.vendorAddress,
    phoneNumber = y.phoneNumber
}).FirstOrDefault();
z.vendorPackageTaken = getVendorPackage(z.vendorId).packageld;
z.vendorAreaDetail = getAreas(z.vendorId);
return View(z);
}
}

public ActionResult removeArea(int area_id, int vendor_id)2
{
```

```
using (var context = new promoLacDbEntities())
{
    var myresult = context.areaAssigneds.Where(x => (x.areald == area_id) &&
(x.vendorId == vendor_id)).First();

    context.areaAssigneds.Remove(myresult);
17
    context.SaveChanges();

}

return RedirectToAction("viewAllVendors");

}

public ActionResult addArea(int vendor_id)
{
    using (var context = new promoLacDbEntities())
    {
        vendorViewModel vendorObject = new vendorViewModel();

        var areasList = context.areas.Select(x => new areaModel
        {

            areald = x.areald,
            areaName = x.areaName,
            areaHashCode = x.areaHashCode

        }).ToList();
    }
}
```

```
List<areaModel> area_ids = getvendorAreas(vendor_id);
```

```
foreach(areaModel x in areasList.ToList())
```

```
{
```

```
foreach(var z in area_ids)
```

```
{
```

```
if (x.areaName.Equals(z.areaName))
```

```
{
```

```
areasList.Remove(x);
```

```
}
```

```
}
```

```
}
```

```
vendorObject.detail = areasList;
```

```
vendorObject.vendorUserName=getVendorUserName(vendor_id);57
```

```
return View(vendorObject);
```

```
}
```

```
    }

    public ActionResult updateVendorPackage(int vendor_id)

    {
        using (var context = new promoLacDbEntities())

        {

            var packagesdetails = context.packages.ToList();

            vendorViewModel v = new vendorViewModel();

            v.vendorId = vendor_id;

            v.package_details = packagesdetails;

            return RedirectToAction("viewAllVendors");

        }
    }

    [HttpPost]

    public ActionResult updateVendorPackage(vendorViewModel v)

    {
        7
        using (var context = new promoLacDbEntities())

        {

            var result = context.vendors.Where(x => x.vendorId ==

v.vendorId).FirstOrDefault();

            addPackage(v.vendorPackageTaken,v.vendorId);

            56
            context.SaveChanges();
        }
    }
}
```

```
        return RedirectToAction("viewAllVendors");

    }

}

[HttpPost]

public ActionResult addArea(vendorViewModel vendor)

{

    int id = getVendorId(vendor.vendorUserName);
17
    using (var context = new promoLacDbEntities())

    {

        foreach (var item in vendor.area_vendor)

        {

            areaAssigned ar = new areaAssigned()

            {

                vendorId = id,
                areaid = Convert.ToInt32(item)

            };

            context.areaAssigneds.Add(ar);

            context.SaveChanges();

        }

    }

    return RedirectToAction("viewAllVendors");
}
```

```
}

public JsonResult IsUserNameAvailable(string vendorUserName)

{
    return Json(!db.vendors.Any(u => u.vendorUserName == vendorUserName),
JsonRequestBehavior.AllowGet);

}
```

```
/* =====
```

Getter Setter functions

```
===== */
```

```
public List<areaAssignedModel> getAreas(int x)

{
    7
    using (var context = new promoLacDbEntities())

    {
        var a = context.areaAssigneds.Where(y => y.vendorId == x).Select(z => new
areaAssignedModel

        {
```

```
areaid = z.areaid

}).ToList();

foreach (areaAssignedModel j in a)
{
    j.areaNames = getAreaName(j.areaid);
}

return a;

}

}

public int getMessagesAllowed(int x)
{
    var contex = new promoLacDbEntities();

    var result = contex.packages.Where(y => y.packagesId == x).FirstOrDefault();

    return result.messagesAllowed;
}

public int getMessagesSent(int x)
{
    var contex = new promoLacDbEntities();
```

```
        var result = contex.vendors.Where(y => y.headId == x).Select(z => new
vendorViewModel()
{
    vendorId=z.vendorId
}).ToList();
int count = 0;
foreach(var item in result)
{
    var result1 = contex.userPackages.Where(y => y.vendorId ==
item.vendorId).FirstOrDefault();
    count = count + result1.notificationSent;
}

return count;
}

public userPackageModel getVendorPackage(int x)
{
    var contex = new promoLacDbEntities();
    var result = contex.userPackages.Where(y => y.vendorId == x).Select(z =>
new userPackageModel()
{
    packageId=z.packageId,
    packageStartTime=z.packageStartTime,
```

```
        packageEndTime=z.packageEndTime,  
        packageStatus=z.packageStatus,  
  
    }).FirstOrDefault();  
  
    return result;  
}  
  
public int getAdminId(string name)  
{  
    1  
    using (var context = new promoLacDbEntities())  
  
    {  
        var v = context.mainAdmins.Where(x => x.username ==  
name).FirstOrDefault();  
  
        return v.id;  
  
    }  
}  
  
public List<areaModel> getvendorAreas(7 int id)  
{  
    using (var context = new promoLacDbEntities())  
  
    {  
        var v = context.areaAssigneds.Where(x => x.vendorId == id).Select(y =>  
new areaModel
```

```
        {

            areald = y.areald,

        }).ToList();

        foreach(areaModel item in v)

        {

            item.areaName = getAreaName(item.areald);

            item.areaHashCode = getAreaHashCode(item.areald);

        }

        return v;

    }

}

public string getAreaName(int id)
{
    using (var context = new promoLacDbEntities())
    {

        var x = context.Areas.Where(y => y.areald == id).FirstOrDefault();

        return x.areaName;

    }
}
```

```
}

public string getAreaHashCode(int id)
{
    using (var context = new promoLacDbEntities())
    {
        var x = context.Areas.Where(y => y.areaId == id).FirstOrDefault();
        return x.areaHashCode;
    }
}

public string getPackageName(int id)
{
    using (var context = new promoLacDbEntities())
    {
        var v = context.Packages.Where(x => x.packageId == id).FirstOrDefault();
        var pName = v.packageName;
        return pName;
    }
}

public int getVendorId(string name)
```

```
{  
    7  
    using (var context = new promoLacDbEntities())  
  
    {  
  
        var v = context.vendors.Where(x => x.vendorUserName ==  
name).FirstOrDefault();  
  
        int id = v.vendorId;  
  
        return id;  
    }  
}  
  
public string getVendorUserName(int id)  
  
{  
  
    using (var context = new promoLacDbEntities())  
  
    {  
  
        var v = context.vendors.Where(x => x.vendorId==id).FirstOrDefault();  
  
        string name = v.vendorUserName;  
  
        return name;  
    }  
}  
  
public int getAreaId(string name)  
  
{  
    7  
    using (var context = new promoLacDbEntities())
```

```
{  
    var v = context.Areas.Where(x => x.areaName == name).FirstOrDefault();  
    int id = v.areaId;  
    return id;  
}  
}  
}  
}
```

53

National University of Computer & Emerging Sciences (FAST-NU)

PromoLac

Location-Based E-Marketing App

Project Supervisor

Sir Farrukh Hassan

Project Team

Inam Gul 16k – 3607

Bilal Ashfaq 16k – 3636

Syed Ammar Mahmood 15k – 3629

10

Submitted in the partial fulfilment of the requirements for the degree of Bachelor of
Science

Submitted on: May 30th, 2019

10

Department of Computer Science

National University of Computer & Emerging Sciences (FAST-NU) Main Campus,

Karachi

May 2020

55

National University of Computer & Emerging Sciences (FAST-NUCES)

Project Supervisor Mr. Farrukh Hasan

Project Manager

Project Team Inam Gul, Muhammad Bilal Ashfaq, Syed Ammar Mahmood

Submission Date

Supervisor

Mr. Farrukh Hasan _____

Head of Department

Dr. Muhammad Atif Tahir

10

Department of Computer Science

National University of Computer & Emerging Sciences (FAST-NU) Main Campus,

Karachi

34

Contents

1. Introduction 3
2. Significance of the Project 3
3. Need for Product 3
4. Scope 3
5. Benefits to Users 3

6. Gap Analysis with the existing solution 4

 Functional Requirements 5

 6.1.1. User Requirements 5

 6.1.1.1. Profile 5

 6.1.1.2. Deals Menu 5

 6.1.1.3. Push Notification 5

 6.1.1.4. Geo-Fencing 5

 6.1.1.5. Messages 5

 6.1.1.6. Payment 5

 6.1.1.7. Target Marketing 5

 6.1.1.8. Map Navigations 5

 38
 Non-Functional Requirements 7

 6.1.2. Performance Requirements 7

 6.1.3. Safety Requirements 7

 6.1.4. Security Requirements 7

 6.1.5. User Documentation 7

OVERALL SYSTEM DESCRIPTION 8

 1.1.1. Geo-Fencing 8

 1.1.2. Geo-Targeting: 8

 1.1.3. Geo-Conquesting 9

 1.1.4. Beacons-Technology 9

Database Design 10

ER Diagram 10

Hardware :	11
Software:	11
7.1. Repository Architecture	12
7.2. Client-Server Architecture	12
8. Use Cases	13

ACKNOWLEDGEMENT

32

In the name of Almighty Allah, Most Gracious, Most Merciful.

All praises are to Allah, for with His grace and mercy that we were able to complete this Final Year Project successfully.

61

We would like to express our deepest appreciation to the National University of Computer and Emerging Science (FAST). Thanks to Mr. Farrukh Hassan, Assistant Professor, and our Final Year Project coordinator for their hard work to ensure the success of this project.

Heartiest gratitude to our supervisor for this project. Sir Farrukh Hassan whose advice, encouragement, knowledge, and support gave me the strength to carry on with this study until its completion.

15
Last not least, our deepest gratitude and thankfulness are dedicated to our family for their prayers, support, and their encouragement to construct the project successfully. Not forgetting all of our friends and all the people who have helped, support and contributed to complete this project. Without help of particular that mentioned above, we faced difficulties while doing out FYP

Thank you to all that help us to finish this Project Promolac. May Allah bless you.

Document Information

Category Information

Customer Public

Project Title Promolac

Document Version 5.1

Author(s) Inam Gul, Bilal Ashfaq, Ammar Mahmood

Approver(s) Farrukh Hasan

Issue Date

Definition of Terms, Acronyms, and Abbreviations

Term Description

Firebase Realtime Database Database for Android

Firebase Authentication User Authentication for Android

Firebase Cloud Messaging Push Notification for Android

Firebase Storage Storing images of data

Android The mobile application of the Android operating system

Activity Refers to Android application screen

Service Refers to Android background running task

SQL Server Database for Web Application

ABSTRACT

In this age of globalization, everyone needs to be aware of events occurring around them and tech minds are working on ways to provide quick and effective information to people with less hassle. If we talk about businesses nowadays, the success of their business and the popularity of their products is highly dependent on promotional activities. So, every vendor nowadays is looking for efficient ways of promoting their products. As we are moving ahead in this age of technology new inventions are being made and technology is becoming easily accessible to everyone. So, taking advantage of this point, this FYP is motivated to bring modernity and efficiency in marketing techniques used nowadays. We are working on context-based marketing which will target specific people based on their location. If we have a look at marketing strategies nowadays, although they are much effective and provide required results, on the other hand, these marketing strategies irritate customers at some point. Like if we take one strategy which is commonly used nowadays is that brands save the customers data and when they want to convey any notification to their customers they push messages to all customers in their database irrespective of whether it is in the interest of that particular person or not, thus creating spam messages. Another technique used is to through advertisements on different social websites which sometimes create irritation to users.

The major advantage of our product is the reduced ratio of spam messages. We will be targeting specific people inside certain geo locations thus reducing the factor of spamming. Secondly, this app can be used by government bodies to notify the public about any serious event like a pandemic alert of COVID 19, blood banks, etc. in order to update people about any emergency or any big events. We are trying to implement some efficient marketing strategies implemented by some vendors in different scenarios individually, in one project. Thus, a compact marketing package will be seen after completion of this project.

Chapter 1-Background

1. Introduction

Promolac is a location-based E-Marketing System consisting of Web and Android Application. The basic business model of Promolac is to minimize spamming and make marketing effective using the location of every user. Currently, every marketing company is purchasing user's data from some organizations at heavy prices and running their marketing campaigns using information from that data. It turns out to be a low ROI (Return over Investment) in this traditional way. Our System is built upon two different pillars Geofencing and Geotargeting which does not take a huge amount of data. It just uses a real-time location of users and connects vendor and user through a common platform "Promolac". It will not only increase ROI for vendors but also it will help users to decide whether they are interested in these types of marketing campaigns or not.

2. Significance of the Project

If we have a look at marketing strategies nowadays, although they are much effective and provide required results, on the other hand, these marketing strategies irritate customers at some point. Like if we take one strategy which is commonly used nowadays is that brands save the customers data and when they want to convey any notification to their customers they push messages to all customers in their database irrespective of whether it is in the interest of that particular person or not, thus creating spam messages. Another technique used is to through advertisements on different social websites which sometimes create irritation to the user. The major advantage of our product is a reduced ratio of spam messages. We will be targeting specific people inside certain geo-location thus reducing the factor of spamming. Secondly, this app can be used by government bodies, blood banks, etc. to update people about any emergency or any other scenario. We are trying to implement some efficient marketing strategies implemented by some vendors in different scenarios individually, in one project. Thus, a compact marketing package will be seen after completion of this project.

3. Need for Product

The idea of the Promolac is to revolutionize the industry of marketing and promotion. Most companies use to purchase or spend huge amounts of money and time through

different marketing strategies to sell their products and to run their business. Now it's crucial for every company to adopt a platform that provide e-marketing through mobile application because now things are changed day by day everyone is using phone to stay connect with the world. Conventional media is upgrading to digital where it's necessary to get a platform to provide infotainment.

4. Scope

This project produced an android app and web application. The website covers two aspects i.e. admin control and vendor control. The vendor will be provided functionalities for generation and management of notification. Admin panel will be for the provision of services to vendors and regulating the whole process. Android app will be used for receiving notification and viewing deals.

5. Benefits to Users

Users have some special benefits in this system. They will receive promotions on different products and can avail that promotion. Also, they will only receive notifications about their interest so that no irrelevant notification will not be received by any user.

6. Gap Analysis with the existing solution

CATEGORY PromoLac OTHERS

Description Location Based E Marketing App Server Based Promotion App

Interface Web and Android based interface Mostly App based

Features Runtime conversion, Interactive interface, remote database Static software, runtime conversion with less accuracy

Constraint One end should have Leap motion connected Other person should be in front of view so the conversion will work

Constraint Solution NGO can support special person to have this device. OpenCV solution will be provided in future with accuracy They added that improve accuracy with current program.

Tools Python, MongoDB, JAVA, NodeJS Don't know, mostly use Python

Devices Leap Motion Hand Glove uses in some system

Chapter 2 - Requirements Analysis

7. 54 Functional Requirements

7.1.1. User Requirements

7.1.1.1. Profile

User's profile will be maintained properly with authentic data gathered at the time of signup. It is editable by the user after signing up whenever he wants.

7.1.1.2. Deals Menu

Users have lists of options to select the latest deals provided by different vendors.

Deals are divided into Categories so that User has proper easy to use Interface.

7.1.1.3. Push Notification

User Application will receive important updates with the help of push notifications provided by Firebase Cloud Messaging

7.1.1.4. Geo-Fencing

User has a complete view of all the fences around him\her created by different vendors. He can see which fences are nearby to him/her and what they offer. Also when user will enter in a particular Fence, He will receive notification about it like "You are in a Danger Zone"

7.1.1.5. Messages

User will receive some message from vendors related to the latest promotions and sales available on different brands

7.1.1.6. Payment

Users can avail certain deals of their own interest with the secure payment method implemented in the application.

7.1.1.7. Target Marketing

Each User is subscribed to some specific Topics which will be used to send Target Marketing to Specific People. If he is not interested in some marketing, He can Unsubscribe from them.

7.1.1.8. Map Navigations

Users have a proper navigation system that guides users to the target franchise of interesting deals. It requires Google Maps to guide to the proper location and the way towards it by starting directions on to Google Maps

7.1.1.9. Offline Support

User application has also offline support for Geofencing means if you are running out of data or you have no internet connection, User Application is capable of doing all the work offline. It will send you all Fence Notifications even though users do not have an active internet connection.

7.1.2. Vendor Requirements

Dashboard

Vendors have a dashboard that enables vendors to get a summary report of their activities on a monthly basis so that it is easier to get information from one page only. The dashboard also comprises of information of his packages and remaining buckets

Packages

Vendors will be subscribed to different packages created by Administrator. They will have a bucket about their subscribed packages. After expiration of the package, vendors have to contact the administrator to assign another package for him. However, he will not be able to subscribe to more than one package at the same time.

Target Notification

Vendors can send different notifications to the target audience based on their real-time locations. It helps users to get rid of spamming of every irrelevant notification and the user can use filters in order to avoid such notifications if he is not interested.

Geofencing

Geofencing helps vendors to create a fence in a particular area. Only users going through that fence will be notified others will not get any notification about that particular Fence Notification. It also consumes some time from the package which is limited by the subscribed package

Messaging

Vendors can send messages along with notifications. They are just low priority as they will not create an alert on mobile phones, but users will still get messages about promotions and sales that reside nearby to them.

Deals

Vendors can create deals with certain promotions as users will get it on the mobile application. Deals then will be availed through a password set by Vendor itself so that he can verify that either the user is authenticated or not.

52

8. Non-Functional Requirements

8.1.1. Performance Requirements

The performance characteristics of the system that are required by the business should be outlined in this section. Performance characteristics include the speed, precision, concurrency, capacity, safety, and reliability of the software. These characteristics define the performance of the project.

35

8.1.2. Safety Requirements

Specify the requirements that are concerned with possible loss, damage, or harm that could result from the use of the system. Define any safeguards or actions that must be

5

taken, as well as potentially dangerous actions that must be prevented. Identify any safety certifications, policies, or regulations to which the system must conform.

8.1.3. Security Requirements

Specify any requirements regarding security, integrity, or privacy issues that affect the use of the system and protection of the data used or created by the system. Define all user authentication or authorization requirements, if any. Identify any security or privacy policies or certifications the system must satisfy.

5

8.1.4. User Documentation

[List the user documentation components that will be delivered along with the software, such as user manuals, online help, context-sensitive help and tutorials.]

9. Features

This project provides location-based marketing depending on customer choice. This app provides easy access to the brand's new deals that connect customers on the bases of their location. This will provide information about the best places nearby along with the best deals. The Company and Consumer relationship will be maintained by maintaining rewards points.

Main Significant Feature

The main feature and the motivation of making our project are that it can be used in various contexts, not only for marketing. Like announcements in targeted areas etc. other than that, it will completely modernize the ways of marketing too.

Other features are

- Can create a push notification for a targeted audience
- Can add more members of their marketing team as well as their dashboard
- Can settle down all the permissions for their own members.
- Can view their limit reaching the purchased amount like push messages will be allowed 10,000 in one month at some price, and 20,000 for more price etc. (Different packages)
- Have access to stop the push notification created from any company.
- Have rights to take a decision for clash of the same marketing place
- Can block any user in case of security threats
- Increase or decrease the area of changing location
- Can access the personal data of users of better experience and personalized suggestion
- Have a threshold marketing deal that every company should stay on their limit of offering
- Location update via backend service
- Search option for the best options in any categories (like restaurants, shops, market etc.) nearest to customers.

10. OVERALL SYSTEM DESCRIPTION

10.1 PROJECT

The current advancement in technology of mobile development, the social media that everyone is using is to express their social life and educational media for education. No app currently in the market is available that will connect education with the social media environment and will restrict the content that will overfit the subject. This project is intended to bring advancement in marketing strategies used nowadays. As far as location-based marketing is concerned there are four main pillars that lay down a platform for this marketing strategy. These components are discussed below:

1.1.1. Geo-Fencing

It is a location-based advertisement technique in which the user's location is watched and only people in a specific location target are notified. The targeted users use any applications or participate in any programs where they are prompted to enter their location or allow a web service to access their location.

Facebook and Google (Google AdWords) have provided this feature to their user, to create a geofence and notify about anything to your targeted audience [1][2].

1.1.2. Geo-Targeting:

The practice of delivering content/advertisements to people based on their geographic location is termed as geo-targeting. Google Ads (formerly known as AdWords) has a feature that allows search advertisers to specify a location, or a set of locations, as the only area(s) in which they want their ads to show. This technique facilitates those restaurants and markets that have branches in different areas and they want to notify only people of that particular area. This saves people from a lot of spamming and the cost of an advertiser is also reduced.

1.1.3. Geo-Conquesting

This idea is somehow close to above mentioned techniques. The core idea of this technique is to direct potential customers to your business when they are close to your competitors. This technique is used to increase brand awareness and attract customers towards your shop from your competitors. For example, if you are an independent coffee shop, you could introduce geo Conquesting to your marketing plans by targeting users who are at nearby big chains like Starbucks or Costa with ads about the great menu items you offer, cheaper coffee, fantastic loyalty scheme or the importance of supporting small, local, independent businesses.

1.1.4. Beacons-Technology

This technique is used at a lower level to market products individually. The core idea behind this technique is when a customer passes by any product in shop the wireless transmitters notify him about any special packages offered by that shop on that product. This technique is considered to be the latest development in proximity marketing.

2.3 PRODUCT FUNCTIONS

- User account: System will allow creating an account on the portal.
- Number of users being supported by the system: System will be efficient to handle a large number of users.
- Search: search is simply a local search engine based on keywords.

- Discussion Forum: Provides users with a platform to discuss and help each other with their problems
- Nearby: nearby will help to find nearby friends in a circle

11. DESIGN DETAILS

Database Design

ER Diagram

//Completed in Word. Shown After integration

⁴⁴
Sequence Diagram

Sequence Diagram 1

Sequence Diagram 2

State Diagram

State Diagram1

State Diagram 2

12. IMPLEMENTATION DETAILS

7.1 DEVELOPMENT TOOLS

Development tools used in this project are mentioned below:

7.1.1.

7.1.2.

7.1.3

7.1.4

7.2 UI's

UI used in this project are included with frameworks, here are some:

1. .NET Web Interface
2. Material UI Android

Hardware:

- Processor: Intel Core i3 1 generations and above
- Memory: 4 GB and above
- Storage: 10 GB and above (SSD will be preferable)

Software:

Software required to complete this project includes:

- OS: 64-bit Windows 7/10
- JDK, Android Studio, Visual Studio, Oracle VMWare, SQL Server 2017, Anaconda Distribution

7.4 ARCHITECTURE

Hybrid Architecture with the combination of Repository and Client Server is used for increasing flexibility and efficiency.

71. Repository Architecture

Android Application and Web Application both must exchange data in real-time. This may be done by using Repository Architecture. Repository Architecture will be the shared part between two servers used by our project team.

Each Application maintains its own database and passes data explicitly to each other and will be responsible for changes on the other hand too so that synchronization will take place.

7.1.1. Advantages of Repository Architecture

- Efficient way to share large amounts of data
- Centralized Management that takes control of backup, integrity and security
- Integrate into a different database and maintain the workflow

7.2. Client-Server Architecture

Data processing is distributed across multiple clients and Application Servers and also allows clients to access the server from any point of connection, and privilege levels will be maintained at the server level so that data can be confidential and secure.

24

7.2.1. Advantages of Client-Server Architecture

- Organize the Client and Server in a set of layers each provides a set of services
- Data Encryption attain successfully
- Message broadcast will be easy through a server

7.2.2. Diagram Client and Server + Repository Architecture:

8. Use Cases

TEST PLAN 37

VERSION HISTORY

Version

Implemented

By Revision

Date Approved

By Approval

Date Reason

1.0 Promolac team 06/06/20 Mr. Farrukh Hasan 07/06/20 Test Plan

draft

48
Contents

1.	INTRODUCTION	3
1.1.	Purpose	3
2.	ENVIRONMENTAL NEEDS	3
3.	TEST STRATEGY	4
3.1.	Test Objectives	4
3.2.	Test Assumptions	4
3.3.	Test Principles	5
3.4.	Data Approach	5
3.5.	Scope and Levels of Testing	5
3.5.1.	Exploratory	5
3.5.2.	Functional Test	6
3.5.3.	User Acceptance Test (UAT)	7
4.	EXECUTION STRATEGY	7
4.1.	Entry and Exit Criteria	7
4.2.	Test Cycles	8
4.3.	Validation and Defect Management	9
4.4.	Test Metrics	10
4.5.	Defect tracking & Reporting	10

5.	TEST MANAGEMENT PROCESS	11
5.1.	Test Design Process	11
5.2.	Test Execution Process	12
5.3.	Test Risks and Mitigation Factors	13
5.1.	Communications Plan and Team Roster	14
5.2.	Role Expectations	14
5.2.1.	Project Supervisor	14
5.2.2.	Test Planning (Test Lead)	14
5.2.3.	Test Team	15
5.2.4.	Test Lead	15
5.2.5.	Development Team	15
6.	TEST ENVIRONMENT	16
7.	TEST CASES	16

No.	4 STEPS	EXPECTED RESULTS	ACTUAL RESULTS	PASS/ FAIL
1.	Correct id and password	Successful Login	Successfully login	Pass
2.	Correct id, Wrong password	41 Show Error (Wrong Id Exception)	Login denied with error message	Pass
3.	Wrong id and password	41 Show Error (Wrong id and password Error)	Login denied with error message	Pass

- | | | |
|----|--------------------------------------|-------------------------------------|
| 4. | Wrong id, Correct password | Show Error (Wrong password Error) |
| | Login denied with error message Pass | |
| 5. | Empty Password or Username | Empty Field Prompt App Crashed Fail |
| 6. | Empty Password or Username (updated) | Empty Field Prompt |
| | Empty Field Prompt Pass | |

Test 2:

TEST CASE ID: PRL002

DESCRIPTION: User Signup Check (Android Application)

No.	STEPS	EXPECTED RESULTS	ACTUAL RESULTS	PASS/ FAIL
1.	Register new account	Register Successful	Register Successfully Pass	Pass
2.	Register by using used Email	Show Error (Already Register Prompt) Successfully Registered	Fail	Fail
3.	Register by using used Email	Show Error (Already Register Prompt) Login denied with error message	Pass	Pass
4.	Register by using wrong email without domain address Address is in wrong pattern	Email Nothing happened	User Register Successfully	Fail

5.	Register by using wrong email without domain address	Email
	Address is in wrong pattern	Nothing happened Fail
6.	Register by using wrong email without domain address	Email
	Address is in wrong pattern prompt	Email Address is in wrong pattern prompt
	Pass	
7.	Empty Password or Name or Email	Empty Field Prompt App
Crashed	Fail	
8.	Empty Password or Name or Email	Empty Field Prompt Empty Field
Prompt	Pass	

Test 3:

TEST CASE ID: PRL003

DESCRIPTION: Upload Profile (Android Application)

No.	STEPS	EXPECTED RESULTS	ACTUAL RESULTS	PASS/ FAIL
1.	Upload Profile	Upload Successful	Upload Successfully	Pass
2.	Checking Upload on Firebase	Upload Successfully	No Record	
	Found Fail			

3. Checking Upload on Firebase (again) Upload Successfully
 Uploading Error in App Fail
4. Checking Upload on Firebase (again) Upload Successfully App
 Crashed Fail
5. Checking Upload on Firebase (again) Upload Successfully
 Upload Successfully Pass

Test 4:

TEST CASE ID: PRL004

DESCRIPTION: Deal List View Test (Android Application)

No.	STEPS	EXPECTED RESULTS	ACTUAL RESULTS	PASS/ FAIL
1.	Deals Added	Deals will be Show Correctly completely	Last Deal is not shown Fail	
2.	Deals Added	Deals will be Show Correctly completely	First Deal is not shown Fail	
3.	Deals Added	Deals will be Show Correctly	Deal Show Correctly Pass	
4.	Deals Image Downloading	Download Successfully	App Crashed Fail	

5. Deals Image Downloading Download Successfully Downloaded
Successfully Pass
6. Checking Upload on Firebase Upload Successfully App
Crashed Fail
7. Checking Upload on Firebase Upload Successfully Upload
Successfully Pass

Test 5:

TEST CASE ID: PRL005

DESCRIPTION: Message List View Test (Android Application)

No.	STEPS	EXPECTED RESULTS	ACTUAL RESULTS	PASS/ FAIL
1.	Messages Added	Messages will be Show Correctly Last Message is not shown completely	Fail	
2.	Messages Added	Messages will be Show Correctly First Message is not shown completely	Fail	
3.	Message Added	Message will be Show Correctly	Message Show Correctly	Pass
4.	Message Downloading	Download Successfully	App Crashed	Fail

5.	Message Downloading	Download Successfully	Downloaded
Successfully Pass			
6.	Checking Upload on Firebase	Upload Successfully	App
Crashed	Fail		
7.	Checking Upload on Firebase	Upload Successfully	Upload
Successfully Pass			
8.	Checking save in file	Saved in File	Not saved in file
9.	Checking save in file	Saved in File	Saved in File
10.	Delete Message	Delete Successfully	Not Deleted
11.	Delete Message	Delete Successfully	Not Deleted
			Pass

Vendor/Admin Test Cases

Test Case 1

TEST CASE ID: PRLV001

DESCRIPTION: Vendor/Admin login on Web Portal

No.	4 STEPS	EXPECTED RESULTS	ACTUAL RESULTS	PASS/ FAIL
1.	Authentic Id and Password	Successful Login	Successfully Login	pass
2.	Correct Id but wrong Password	Access Denied with Error message Access Denied	Access Denied with Error message pass	

3. Correct password with wrong ID Access Denied with Error message
 Access Denied pass
- 19
4. Both Id and password wrong Access Denied with Error message
 Access Denied pass
- 19
5. Empty Id/Password Field Required Prompt Fields Required Prompt
 40
 pass

Test Case 2

TEST CASE ID: PRLV002

DESCRIPTION: Package and Trying to Access

No.	STEPS	EXPECTED RESULTS	ACTUAL RESULTS	PASS/ FAIL
1.	Package available	Successful Login	Successfully Login	pass
2.	Package expired Denied	Access Denied with Error message	Access 40 pass	Access

Test Case 3

TEST CASE ID: PRLV003

DESCRIPTION: Deals Create

No.	STEPS	EXPECTED RESULTS	ACTUAL RESULTS	PASS/ FAIL
1.	Deal Create with Package available	Successful Create	Successfully Created	pass
2.	Deal Create with Package bucket expired	Create Deny Create	Denied	pass

Test Case 4

TEST CASE ID: PRLV004

DESCRIPTION: Geofence Create

No.	STEPS	EXPECTED RESULTS	ACTUAL RESULTS	PASS/ FAIL
-----	-------	------------------	----------------	------------

1.	Fence Create with Package available	Successful Create
	Successfully Created	pass
2.	Fence Create with Package bucket expired	Create Deny Create

Denied pass

Test Case 5

TEST CASE ID: PRLV005

DESCRIPTION: Send Notification

No.	STEPS	EXPECTED RESULTS	ACTUAL RESULTS	PASS/ FAIL
25.	Send Notification with Package available	Successful Notification		
send	Successfully Sent	pass		

26.	Send Notification with Package bucket expired	Send Notification Deny
	Send Notification Denied	pass

Test Case 6

TEST CASE ID: PRLV006

DESCRIPTION: Send Message

No.	STEPS	EXPECTED RESULTS	ACTUAL RESULTS	PASS/ FAIL
33.	Send Message with Package available	Successful Message send		
	Successfully Sent	pass		
34.	Send Message with Package bucket expired	Send Message Deny		
	Send Message Denied	4 pass		

Test Case 7

TEST CASE ID: PRLV007

DESCRIPTION: Create Sub Vendor

No.	STEPS	EXPECTED RESULTS	ACTUAL RESULTS	PASS/ FAIL
41.	Create Sub Vendor limited by Package	Successful Create		
	Successfully Created	pass		
42.	Create Sub Vendor in Empty bucket	Create Deny Denied	pass	

7. LIMITATION OF SYSTEM

For now, our system is limited to Karachi users. Internet connectivity for once is required along with location services kept turned on.

8. FUTURE WORK

In future we will implement location-based group chat in mobile application and two of our work Geo-Conquesting and Beacons-Technology. User experience will get major upgrade by applying Machine Learning to learn user pattern.

9. CONCLUSION

A software which can help companies to get marketing services of their product's in real life. Our goal is to make something for them that can make their life easy. Our aim is to target that kind of people and contribute in society with our product. Promolac system for e-marketing using location details was successfully executed with high accuracy. The method gives output in push notifications and message updates that helps to reduce the marketing gap between companies and end users. Thus, the implementation of system output is done in Android and Python.

Final Report

ORIGINALITY REPORT



PRIMARY SOURCES

- | | | |
|---|---|-----|
| 1 | Submitted to National University of Ireland, Galway | 3% |
| 2 | Submitted to Royal Business College | 1 % |
| 3 | jsblorh.com | 1 % |
| 4 | Submitted to University of Hertfordshire | 1 % |
| 5 | www.pitb.gov.pk | 1 % |
| 6 | Submitted to Napier University | 1 % |
| 7 | Submitted to University of Sydney | 1 % |
| 8 | dotnettutorials.net | 1 % |
| 9 | Submitted to Study Group Worldwide | |
-
- 1 Submitted to National University of Ireland, Galway 3%
2 Submitted to Royal Business College 1 %
3 jsblorh.com 1 %
4 Submitted to University of Hertfordshire 1 %
5 www.pitb.gov.pk 1 %
6 Submitted to Napier University 1 %
7 Submitted to University of Sydney 1 %
8 dotnettutorials.net 1 %
9 Submitted to Study Group Worldwide

10	Submitted to Higher Education Commission Pakistan	1 %
11	appinstitute.com Internet Source	<1 %
12	Submitted to Southern New Hampshire University - Continuing Education	<1 %
13	Submitted to Nottingham Trent University Student Paper	<1 %
14	Submitted to KDU College Sdn Bhd Student Paper	<1 %
15	Submitted to Universiti Teknologi MARA Student Paper	<1 %
16	Submitted to KTH - The Royal Institute of Technology Student Paper	<1 %
17	Submitted to 8936 Student Paper	<1 %
18	Submitted to American Intercontinental University Online Student Paper	<1 %

19	Submitted to University of Ulster Student Paper	<1 %
20	Submitted to National College of Ireland Student Paper	<1 %
21	mahedee.net Internet Source	<1 %
22	parasbabbar.in Internet Source	<1 %
23	www.c-sharpcorner.com Internet Source	<1 %
24	Submitted to University of Technology, Sydney Student Paper	<1 %
25	Submitted to Udayana University Student Paper	<1 %
26	www.wordstream.com Internet Source	<1 %
27	maxprokurnia.com Internet Source	<1 %
28	Submitted to Chichester College of Arts, Science and Technology, West Sussex Student Paper	<1 %
29	www.itstrike.cn Internet Source	<1 %

30	www.flipandroid.com Internet Source	<1 %
31	Submitted to Chester College of Higher Education Student Paper	<1 %
32	Submitted to University of Reading Student Paper	<1 %
33	www.nopcommerce.com Internet Source	<1 %
34	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Student Paper	<1 %
35	Submitted to University of Westminster Student Paper	<1 %
36	oldmastersgallery.co.uk Internet Source	<1 %
37	www.coursehero.com Internet Source	<1 %
38	Submitted to University of Wollongong Student Paper	<1 %
39	Submitted to University of Greenwich Student Paper	<1 %
40	Submitted to Melbourne Institute of Technology Student Paper	<1 %

41	Submitted to Universiti Teknikal Malaysia Melaka Student Paper	<1 %
42	Submitted to University of Central Lancashire Student Paper	<1 %
43	Submitted to University of Wisconsin System Student Paper	<1 %
44	Submitted to University of East London Student Paper	<1 %
45	Submitted to Middle East College of Information Technology Student Paper	<1 %
46	Submitted to Oxford Brookes University Student Paper	<1 %
47	Laurence Moroney. "The Definitive Guide to Firebase", Springer Science and Business Media LLC, 2017 Publication	<1 %
48	docplayer.net Internet Source	<1 %
49	forums.asp.net Internet Source	<1 %
50	Submitted to Glyndwr University Student Paper	<1 %

51	stat.fi Internet Source	<1 %
52	Submitted to University of Bradford Student Paper	<1 %
53	Tariq Mahmood, Khadija Rohail. "Analyzing terrorist incidents to support counter-terrorism - Events and methods", 2012 International Conference of Robotics and Artificial Intelligence, 2012 Publication	<1 %
54	Submitted to Cape Peninsula University of Technology Student Paper	<1 %
55	journals.sagepub.com Internet Source	<1 %
56	Submitted to Laureate Education Inc. Student Paper	<1 %
57	Submitted to Informatics Education Limited Student Paper	<1 %
58	Submitted to Louisiana Tech University Student Paper	<1 %
59	stackoverflow.com Internet Source	<1 %
60	inducesmile.com Internet Source	<1 %

61

Muhammad Ramzan, Arfan Jaffar, Amjad Iqbal,
Sajid Anwar, Abdul Rauf, Arshad Ali Shahid.

<1 %

"Project scheduling conflict identification and
resolution using genetic algorithms (GA)",
Telecommunication Systems, 2011

Publication

62

Submitted to University of Missouri, Kansas City

<1 %

Student Paper

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

Off

Final Report

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

PAGE 69

PAGE 70

PAGE 71

PAGE 72

PAGE 73

PAGE 74

PAGE 75

PAGE 76

PAGE 77

PAGE 78

PAGE 79

PAGE 80

PAGE 81

PAGE 82

PAGE 83

PAGE 84

PAGE 85

PAGE 86

PAGE 87

PAGE 88

PAGE 89

PAGE 90

PAGE 91

PAGE 92

PAGE 93

PAGE 94

PAGE 95

PAGE 96

PAGE 97

PAGE 98

PAGE 99

PAGE 100

PAGE 101

PAGE 102

PAGE 103

PAGE 104

PAGE 105

PAGE 106

PAGE 107

PAGE 108

PAGE 109

PAGE 110

PAGE 111

PAGE 112

PAGE 113

PAGE 114

PAGE 115

PAGE 116

PAGE 117

PAGE 118

PAGE 119

PAGE 120

PAGE 121

PAGE 122

PAGE 123

PAGE 124

PAGE 125

PAGE 126

PAGE 127

PAGE 128

PAGE 129

PAGE 130

PAGE 131

PAGE 132

PAGE 133

PAGE 134

PAGE 135

PAGE 136

PAGE 137

PAGE 138

PAGE 139

PAGE 140

PAGE 141

PAGE 142

PAGE 143

PAGE 144

PAGE 145

PAGE 146

PAGE 147

PAGE 148

PAGE 149

PAGE 150

PAGE 151

PAGE 152

PAGE 153

PAGE 154

PAGE 155

PAGE 156

PAGE 157

PAGE 158

PAGE 159

PAGE 160

PAGE 161

PAGE 162

PAGE 163

PAGE 164

PAGE 165

PAGE 166

PAGE 167

PAGE 168

PAGE 169

PAGE 170

PAGE 171

PAGE 172

PAGE 173

PAGE 174

PAGE 175

PAGE 176

PAGE 177

PAGE 178

PAGE 179

PAGE 180

PAGE 181

PAGE 182

PAGE 183

PAGE 184

PAGE 185

PAGE 186

PAGE 187

PAGE 188

PAGE 189

PAGE 190

PAGE 191

PAGE 192

PAGE 193

PAGE 194

PAGE 195

PAGE 196

PAGE 197

PAGE 198

PAGE 199

PAGE 200

PAGE 201

PAGE 202

PAGE 203

PAGE 204

PAGE 205

PAGE 206

PAGE 207

PAGE 208

PAGE 209

PAGE 210

PAGE 211

PAGE 212

PAGE 213

PAGE 214

PAGE 215

PAGE 216

PAGE 217

PAGE 218

PAGE 219

PAGE 220

PAGE 221

PAGE 222

PAGE 223

PAGE 224

PAGE 225

PAGE 226

PAGE 227

PAGE 228

PAGE 229

PAGE 230

PAGE 231

PAGE 232

PAGE 233

PAGE 234

PAGE 235

PAGE 236

PAGE 237

PAGE 238

PAGE 239

PAGE 240

PAGE 241

PAGE 242

PAGE 243

PAGE 244

PAGE 245

PAGE 246

PAGE 247

PAGE 248

PAGE 249

PAGE 250

PAGE 251

PAGE 252

PAGE 253

PAGE 254

PAGE 255
