Student Tracking Monitoring System

1. INTRODUCTION:

This project helps parents to keep track of their children. It provides easy communication between school & parents. Using this, parents can easily track the position of school bus. This helps in real time location tracking about where the child is. School management reduces the risk of parents in their child's security. It reduces the police involvement, as this may avoid crimes like kidnaps etc. The additional objective of this project is, it is useful in case of vehicle theft detections.

Children lack the skills to protect themselves. It is our responsibility to safeguard children. After child's school parents are facing problems in receiving their child safe back home from near bus stops. Parents have no information about where their child is. There is need to commute between bus driver and parent. Little late in school bus make parent's feel tense. Sometimes parents wait for hours in the bus stop to receive their child. Safer transportation of school children has become a critical issue now-a-days. As it is said that technologies are made for man, child tracking system will really bring change towards school bus safety system. This also proves the reliable and effective system for children safety. Child tracking system helps to avoid the crime against children which is taking place for many school going children. This technology make use of GPS (Global positioning system) for tracking the bus location, GSM (global system for mobile communication).

This system is proposed to intimate parents about the bus location and also about child boarding to the school. Today Internet connects people through countries and continents then why can't we communicate between bus driver and parents through GPS and GSM based tracking system. So the proposed system uses GPS (Global positioning system) and GSM (Global positioning system). We are presenting a bus safety management system which is designed to control the entering and exiting of students to and from the bus. Each parent is given a specific id in order to track their children bus through the login into application.

Parent is informed of the estimated time of arrival of their child's bus before it reaches the stop before/after school. Parents can track the location of the bus in real time. In case there is a traffic jam, natural calamity or any other problem, an SMS is immediately dispatched to the parent informing the reason for delay. This will let the driver to know the number of students entered the bus and the students who departed from the bus. In addition, if the bus departs and arrive successful from the source to destination, it will inform the management through a notification about its successful departure and arrival is done. There is also one more unit proposed in this system i.e. the Parent Unit, in the Parent Unit when a child successfully arrived and departed from source to the destination the Notification is

sent to the Parent Unit (Example: “Your child is arrived into the bus” with the name and time when he arrived).

Advantages:

- Real-time location tracking.
- Not every school have this security feature.
- It is useful in case of vehicle theft detection.
- It is available at minimal cost and performs with high efficiency.

2. DESIGN

REQUIREMENT SPECIFICATION

1. Android operating system to run the application.

2. SQL database to store the information.

3. RAM more than 512mb to run the application in smart phone.

4. ROM between 2-5mb to download the application.

ANDROID OPERATING SYSTEM TO RUN THE APPLICATION: Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touch screen mobile devices such as smart phones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for Cars, and Android Wear for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.

SQL DATABASE TO STORE THE INFORMATION: A database is an organized collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information, such as (for example) modeling the availability of rooms in hotels in a way that supports finding a hotel with vacancies. A database management system is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyze data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases. A database is not generally portable across different DBMSs, but different DBMSs can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS.

RAM: The RAM works quite different on your android phones than on your PC which most of us basic users are better acquainted with. The Apps that you download and install are first loaded in the RAM and then executed. Those Apps remain in the RAM after you are no longer using them and they have

been shifted to background. The next time you use those apps they will be available for fast retrieval from RAM unless they have been removed. With continuous usage, the Apps that you use most frequently get placed on your RAM that is IF you have enough RAM.

ROM: ROM or Read Only Memory is more permanent by nature. What goes in there stays there. The ROM or read only memory forms a part of your internal storage and that part is not Accessible for users to write on and is thus referred to as Read Only Memory. The rest of your internal storage is where you store Apps and can store media files and other documents. This ROM part of your internal storage is where your Android operating system resides. Also OEM's add some preinstalled apps in this memory sections which cannot be deleted on users end either. This is the reason why you don't get full internal memory as advertised on the Box, because a part of it has been used to house O.S. and preinstalled apps. The rest of the internal storage will be used for storing and downloading Apps which will store this**.**

3.2 UML (UNIFIED MODELLING LANGUAGE)

UML stands for Unified Modeling Language. It is a third generation method for specifying, visualizing and documenting the artifacts of an object oriented system under development. Object modeling is the process by which the logical objects in the real world are represented by the actual objects in the program. This visual representation of the objects, their relationships and their structures is for the ease of understanding. This is a step while developing any product after analysis.

Some information could be intentionally omitted from the diagram, some Information represented on the diagram could have different interpretations, and some concepts of UML have no graphical notation at all, so there is no way to depict those on diagrams.

The Unified Modeling Languages emcompasses a number of models:

- Structural Diagram
- Behavioral Diagrams

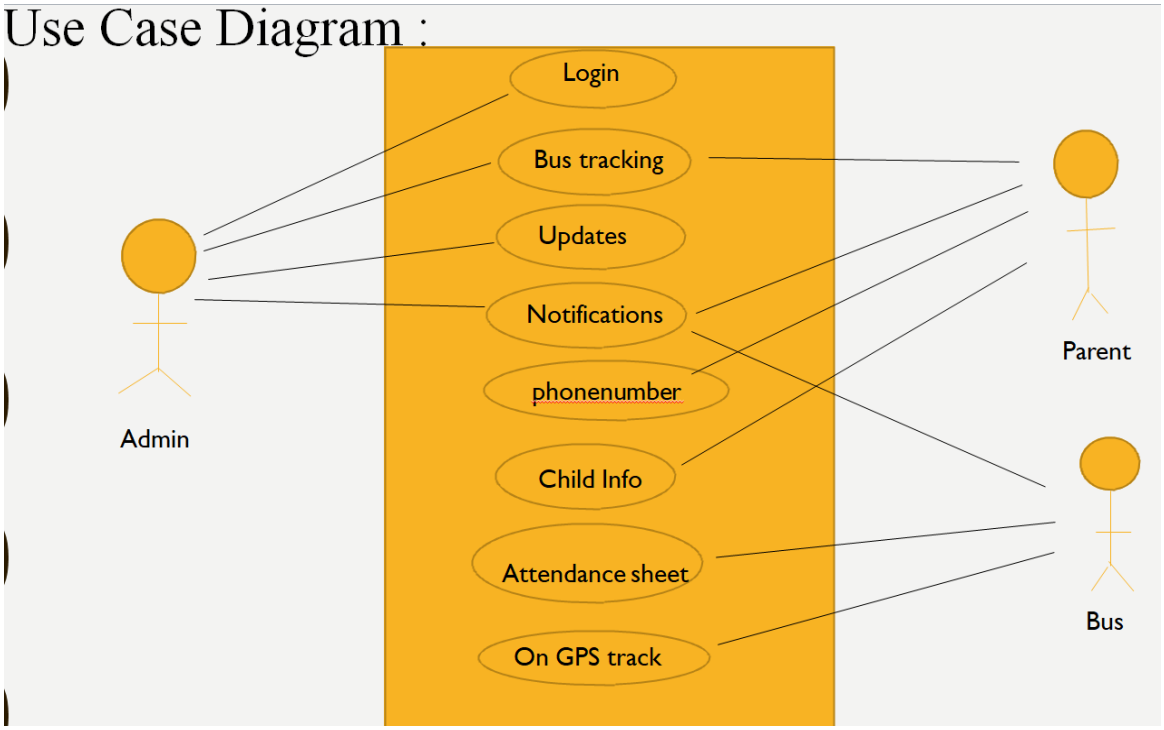
They are again classified into 9 diagrams some of them are

- Usecase Diagram
- Class Diagram
- Sequence Diagram

USE CASE DIAGRAM:

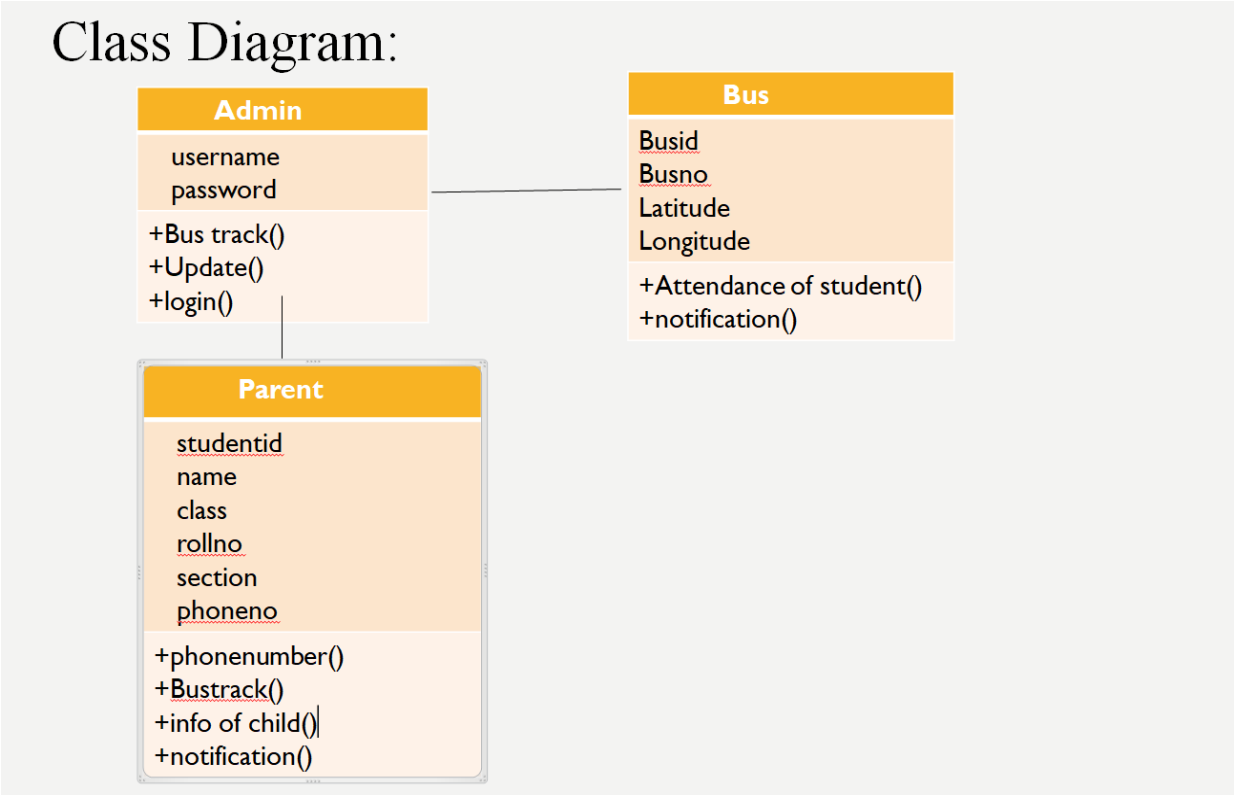
Use case diagram consists of use cases and actors and shows the interaction between them. The key points are:

- The main purpose is to show the interaction between the use cases and the actor.
- To represent the system requirement from users perspective.
- The use cases are the functions that are be performed in the module.
- An actor could be the end-user of the system or an external system.



CLASS DIAGRAM:

Class Diagram consists of the classes and the objects and the interaction between them. It mainly deals with the interaction between classes in the system,their behaviour and properties of the system. Apart from classes this also provides inheritence relationships in the project. The class diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.



SEQUENCE DIAGRAM:

The purpose of sequence diagram is to show to flow of functionality through a use case. In other words, we call it mapping process in terms of data transfers from the actor through the corresponding objects.

The key points are:

- The main purpose is to represent the logical flow of data with respect to process
- A sequence diagram displays the objects and not the classes.

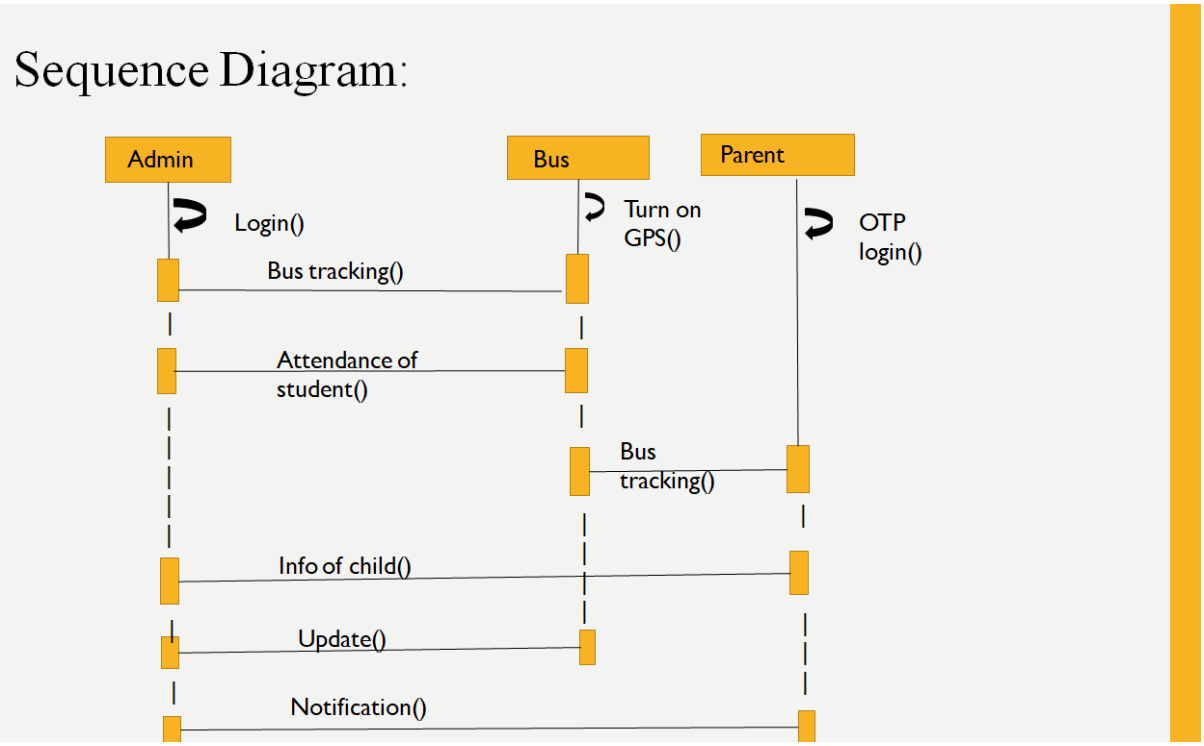


Fig 3.2.1 Sequence Diagram

3.3 ER DIAGRAMS

- The relation upon the system is structure through a conceptual ER-Diagram, which not only specifics the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
- The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modeling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions.

- The primary purpose of the ER diagram is to represent data objects and their relationships.

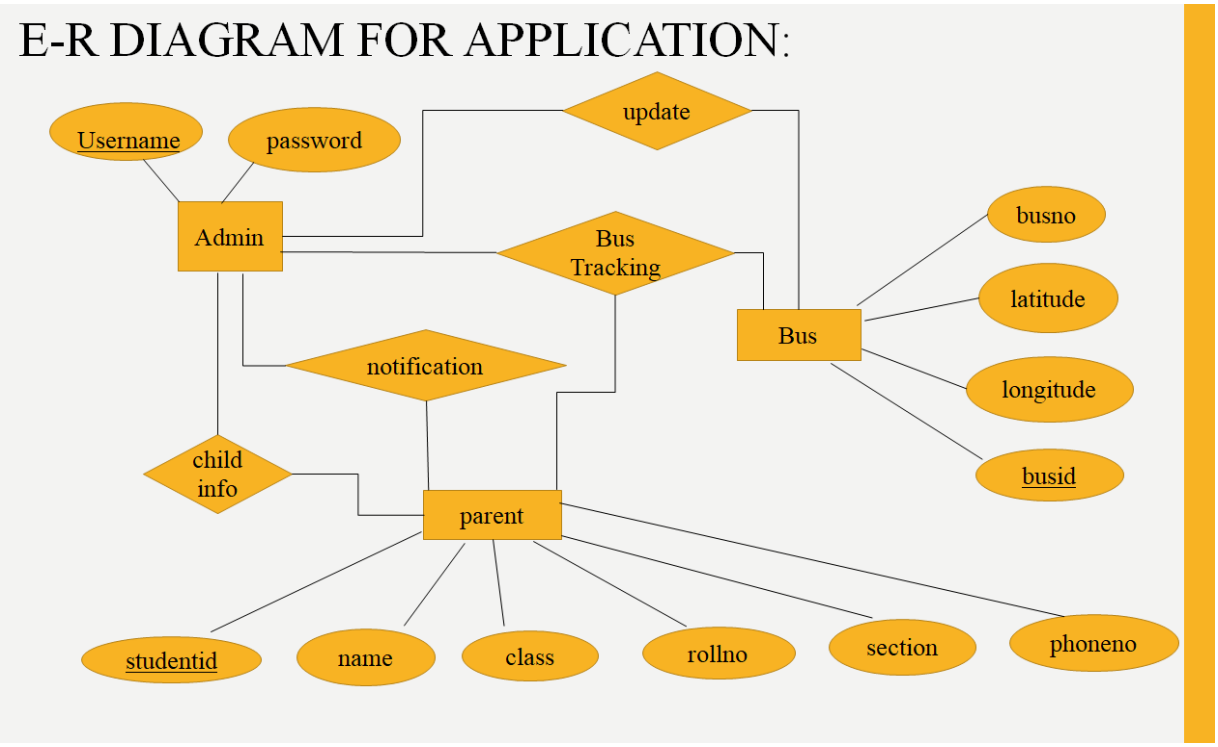


Fig 3.3.1 ER DIAGRAM

4. IMPLEMENTATION

4.1. MODULES:

MODULE DESCRIPTION:

1. **Parent module:**

The operations that are performed are:

- Login into application through their child's unique ID.
- To access the application in order to track the bus location and to get the notification of the school bus exactly where it is.

2. **Admin module:**

The operations that are performed are:

- Monitors the application.
- Controls the operations of the parents.

- It handles and provides security to the data of the application.

4. 2 SAMPLE CODE

```
public class BusHomeActivity extends Activity implements LocationListener {

        String lat="0.0",lon="0.0";
        String phno="",etype="",location="" ;


         LocationManager locationManager ;
           String provider;

           ////////////
           ListView lv;
                  Model[] modelItems;
                  Button post;

                   // Progress Dialog
        private ProgressDialog pDialog;

        //Creating JSON Parser object
        JSONParser jParser = new JSONParser();

        String recs="",att_res="",studids="";
        int n=0;

        TextView t[];
        CheckBox c[];

        ArrayList<String> sids=null;
        //url to get all products list
        private static String url_doc_search =
"http://www.helplinecontacts.com/schoolbt/bus/getstudents.php";
```

```
    //JSON Node names
    private static final String TAG_SUCCESS = "success";
    private static final String TAG_PRODUCTS = "products";
    private static final String TAG_PID = "slno";
    private static final String TAG_NAME = "phno";

    //products JSONArray
    JSONArray products = null;
    LinearLayout l1;

       //////////

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



             // Getting LocationManager object
locationManager = (LocationManager)getSystemService(Context.LOCATION_SERVICE);

// Creating an empty criteria object
Criteria criteria = new Criteria();

// Getting the name of the provider that meets the criteria
provider = locationManager.getBestProvider(criteria, false);

if(provider!=null && !provider.equals("")){

    // Get the location from the given provider
    Location location = locationManager.getLastKnownLocation(provider);

    locationManager.requestLocationUpdates(provider, 20000, 1, this);
```

```
        if(location!=null)
            onLocationChanged(location);
        else
            Toast.makeText(getBaseContext(), "Location can't be retrieved",
Toast.LENGTH_SHORT).show();

            }else{
        Toast.makeText(getBaseContext(), "No Provider Found", Toast.LENGTH_SHORT).show();
    }


    Geocoder geocoder = new Geocoder(getApplicationContext(), Locale.getDefault());
                try {
                    List<Address> listAddresses =
geocoder.getFromLocation(Double.parseDouble(lat), Double.parseDouble(lon), 1);
                    if(null!=listAddresses&&listAddresses.size()>0){
                        location = listAddresses.get(0).getAddressLine(0);
                        Toast.makeText(BusHomeActivity.this, "Location="+location ,
Toast.LENGTH_LONG).show();
                    }
                } catch (IOException e) {
                        Toast.makeText(BusHomeActivity.this, "Location can't retrive" ,
Toast.LENGTH_LONG).show();
                }


                //////////
                 new GetStudDetials().execute();
               l1=(LinearLayout)findViewById(R.id.l1);
```

```
                post=(Button)findViewById(R.id.postatt);
                post.setOnClickListener(new postclick());
               /////////



    }

    @Override
public void onBackPressed() {

  //  super.onBackPressed();
    doExit();
 }

private void doExit() {

   AlertDialog.Builder alertDialog = new AlertDialog.Builder(
         BusHomeActivity.this);

   alertDialog.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

      @SuppressLint("NewApi")
                  @Override
      public void onClick(DialogInterface dialog, int which) {
            BusHomeActivity.this.finishAffinity();
      }
   });

   alertDialog.setNegativeButton("No", null);

   alertDialog.setMessage("Do you want to exit?");
   alertDialog.setTitle("Alert");
```

```
    alertDialog.show();
  }

      ////////////
      class postclick implements OnClickListener
      {

            @Override
            public void onClick(View arg0) {
                  // TODO Auto-generated method stub
                  att_res="";

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

                  String s=t[i].getText().toString();
                  //String ss[]=s.split("-");
                              String sid=sids.get(i);
                  if(c[i].isChecked())
                   att_res+=s+"-P-"+sid+",";
                  else
                        att_res+=s+"-A-"+sid+",";


                  //studids=studids+sids.get(i)+",";
                  }


                  //Toast.makeText(BusHomeActivity.this, att_res,
Toast.LENGTH_LONG).show();

                  new PostAttendance().execute();
```

```
        }

}

/**
 * Background Async Task to Load all product by making HTTP Request
 * */
class GetStudDetials extends AsyncTask<String, String, String> {

    /**
     * Before starting background thread Show Progress Dialog
     * */
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(BusHomeActivity.this);
        pDialog.setMessage("Loading..  Please wait...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(false);
        pDialog.show();
    }

    /**
     * getting All products from url
     * */
    protected String doInBackground(String... args) {
        // Building Parameters

        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("busno",GlobalVariables.busno));
        // getting JSON string from URL
        JSONObject json = jParser.makeHttpRequest(url_doc_search, "GET", params);

        // Check your log cat for JSON reponse
```

```
    Log.d("All Products: ", json.toString());

    try {
        // Checking for SUCCESS TAG
        int success = json.getInt(TAG_SUCCESS);

        if (success == 1)
        {

                recs=json.getString("recs");

        } else {
                recs=null;

        }
    } catch (JSONException e) {
        e.printStackTrace();
    }

    return null;
}

/**
 * After completing background task Dismiss the progress dialog
 * **/
protected void onPostExecute(String file_url) {
    // dismiss the dialog after getting all products
    pDialog.dismiss();

    TextView t1;
    CheckBox c1;

    if(recs!=null)
    {
```

```
String rows[]=recs.split("#");

t=new TextView[rows.length];
c=new CheckBox[rows.length];
int i=0;
sids=new ArrayList<String>();
for(String row : rows)
{
        String ss[]=row.split("!");
        t1=new TextView(BusHomeActivity.this);
        t1.setText(ss[0]);
        c1=new CheckBox(BusHomeActivity.this);
        c1.setChecked(true);
        t[i]=t1;
        c[i++]=c1;

        LinearLayout hl=new LinearLayout(BusHomeActivity.this);
        hl.setOrientation(LinearLayout.HORIZONTAL);

        hl.addView(c1);
        hl.addView(t1);
        l1.addView(hl);

        sids.add(ss[1]);


}

}

}
```

```
        }


////////////////
class PostAttendance extends AsyncTask<String, String, String>
{
private ProgressDialog progressDialog = new ProgressDialog(BusHomeActivity.this);
InputStream is = null ;
String result = "";
protected void onPreExecute() {
progressDialog.setMessage("Fetching data...");
progressDialog.show();
}
@Override
protected String doInBackground(String... args) {



Calendar c=Calendar.getInstance();
String today=c.get(Calendar.YEAR)+"-"+(c.get(Calendar.MONTH)+1)+"-
"+c.get(Calendar.DAY_OF_MONTH);


String url_select = "http://www.helplinecontacts.com/schoolbt/bus/postattendance.php";


ArrayList<NameValuePair> param = new ArrayList<NameValuePair>();

param.add(new BasicNameValuePair("busno",GlobalVariables.busno));
param.add(new BasicNameValuePair("today",today));
param.add(new BasicNameValuePair("att_res",att_res));
```

```
JSONObject json = jParser.makeHttpRequest(url_select, "GET", param);
//Check your log cat for JSON reponse
Log.d("Student details: ", json.toString());




return null;

}
protected void onPostExecute(String file_url) {
//dismiss the dialog after getting all products
progressDialog.dismiss();
Toast.makeText(BusHomeActivity.this, "attendance posted", Toast.LENGTH_LONG).show();

}
}

/////////////////

        ////////////


        @Override
        public void onLocationChanged(Location location) {
        Log.d("Latitude", "changing location");
        //Toast.makeText(HomeActivity.this, "Hello", Toast.LENGTH_LONG).show();
        Toast.makeText(BusHomeActivity.this, "Latitude:" + location.getLatitude() + ", Longitude:"+
location.getLongitude(),Toast.LENGTH_LONG).show();
        lat=""+location.getLatitude();
        lon=""+location.getLongitude();
        new updatebuslocation().execute();
```

```
        }

        @Override
        public void onProviderDisabled(String provider) {
        Log.d("Latitude", "disable");
        }

        @Override
        public void onProviderEnabled(String provider) {
        Log.d("Latitude", "enable");
        }

        @Override
        public void onStatusChanged(String provider, int status, Bundle extras) {
        Log.d("Latitude", "status");
        }

//////////////////////
class updatebuslocation extends AsyncTask<String, String, String>
{

/**
* Before starting background thread Show Progress Dialog
* */
private ProgressDialog pDialog = new ProgressDialog(BusHomeActivity.this);
private static final String TAG_SUCCESS = "success";
int success;



@Override
protected void onPreExecute() {
super.onPreExecute();
```

```
pDialog.setMessage("Add Student...");
pDialog.setIndeterminate(false);
pDialog.setCancelable(true);
pDialog.show();
}

/**
* Storing Data
* */
protected String doInBackground(String... args) {

try {
//Building Parameters


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

params.add(new BasicNameValuePair("busno",GlobalVariables.busno));
params.add(new BasicNameValuePair("latitude",lat));

params.add(new BasicNameValuePair("longitude",lon));

//getting JSON Object
//Note that create product url accepts POST method
//JSONObject json = new
JSonParser().makeHttpRequest("http://www.navyugelectronics.com/ems/addemp.php", "POST",
params);

try {
// Checking for SUCCESS TAG
success = json.getInt(TAG_SUCCESS);

} catch (JSONException e) {
e.printStackTrace();
```

```
}



} catch (Exception e) {
//e.printStackTrace();
//Toast.makeText(AddempActivity.this, "Error"+e.toString(), Toast.LENGTH_LONG).show();
}

return null;
}

/**
* After completing background task Dismiss the progress dialog
* **/
protected void onPostExecute(String file_url) {
//dismiss the dialog once done
pDialog.dismiss();
//if(success==1)
//{

//Intent i=new Intent(AddStudentActivity.this,StudentMenuActivity.class);
//startActivity(i);

}
}
//////////////////////

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
                // Inflate the menu; this adds items to the action bar if it is present.
                getMenuInflater().inflate(R.menu.bus_home, menu);
                return true;
        }
```

}

5. TESTING

5.1 TEST APPROACH:

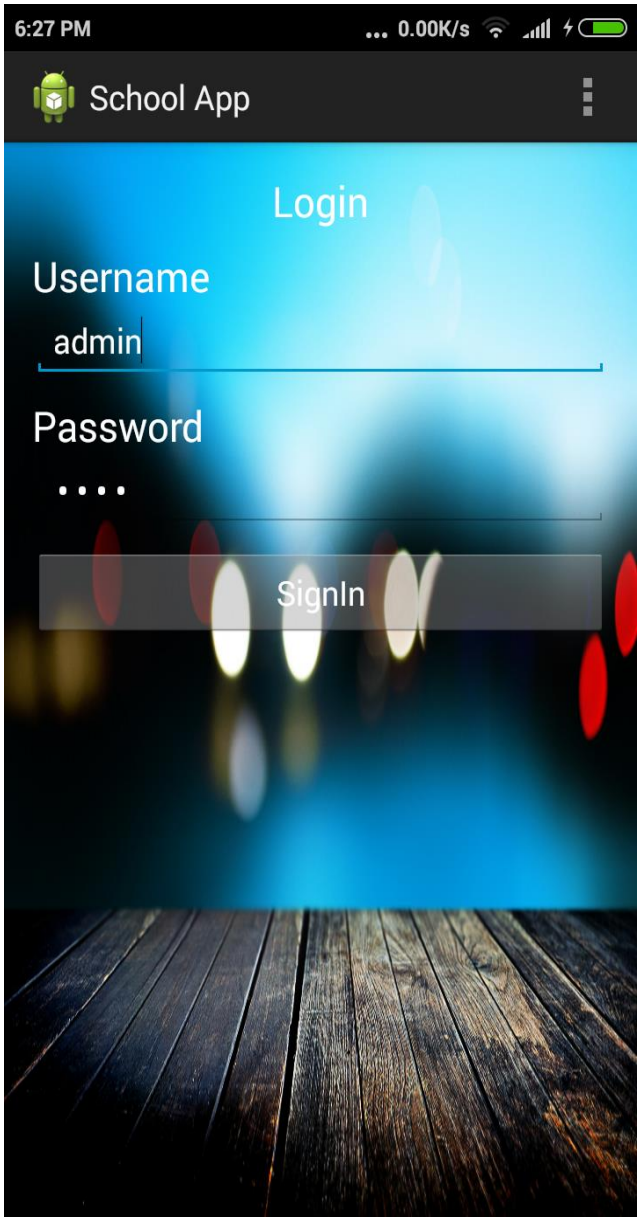
Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

5.2 TEST CASES & RESULTS:

S.no.	Screen	Input	Output	Result
1	Login Page	User Id Password	User validation	Passed
2	User Menu	User will select items from menu.	Selected items are stored in the order records.	Passed
3	User Menu	Orders option is clicked.	Ordered details are displayed.	Passed
4	User Menu	Cancel orders option is clicked and cancel order.	Cancelled orders will be cleared.	Passed
5	Admin Menu	Add items option is clicked and select items to add.	Added items are displayed.	Passed
6	Admin Menu	Order details is clicked.	Displays the orders placed by the users.	Passed
7	Admin Menu	Delivered items are clicked and the date is selected.	Displays the items that are delivered by the admin.	Passed

8	Admin Menu	Daily collections are clicked and the date is selected.	Displays the total collections of the selected date.	Passed

6. RESULT



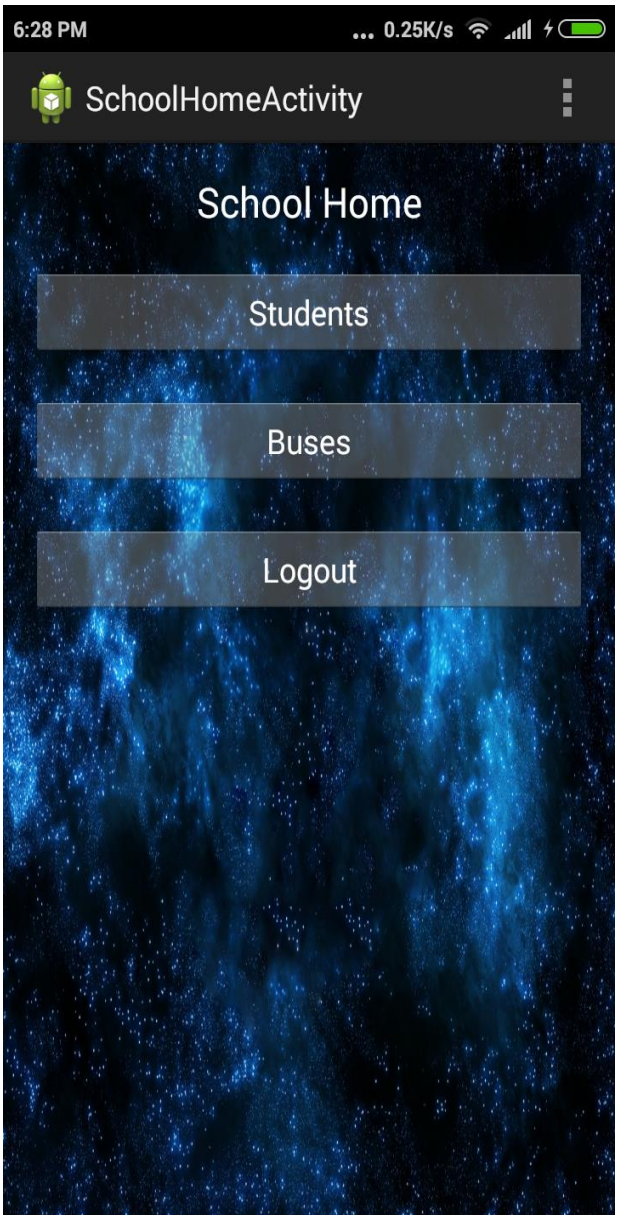
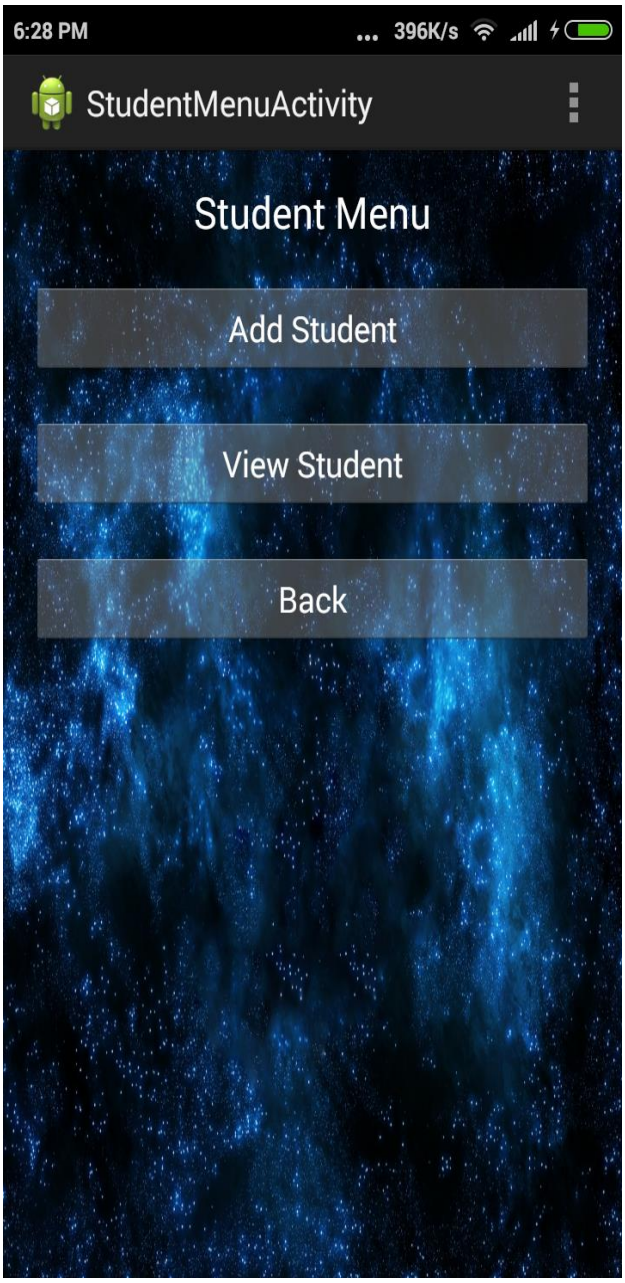


Fig 6.1a Login page

Fig 6.1b School home activity



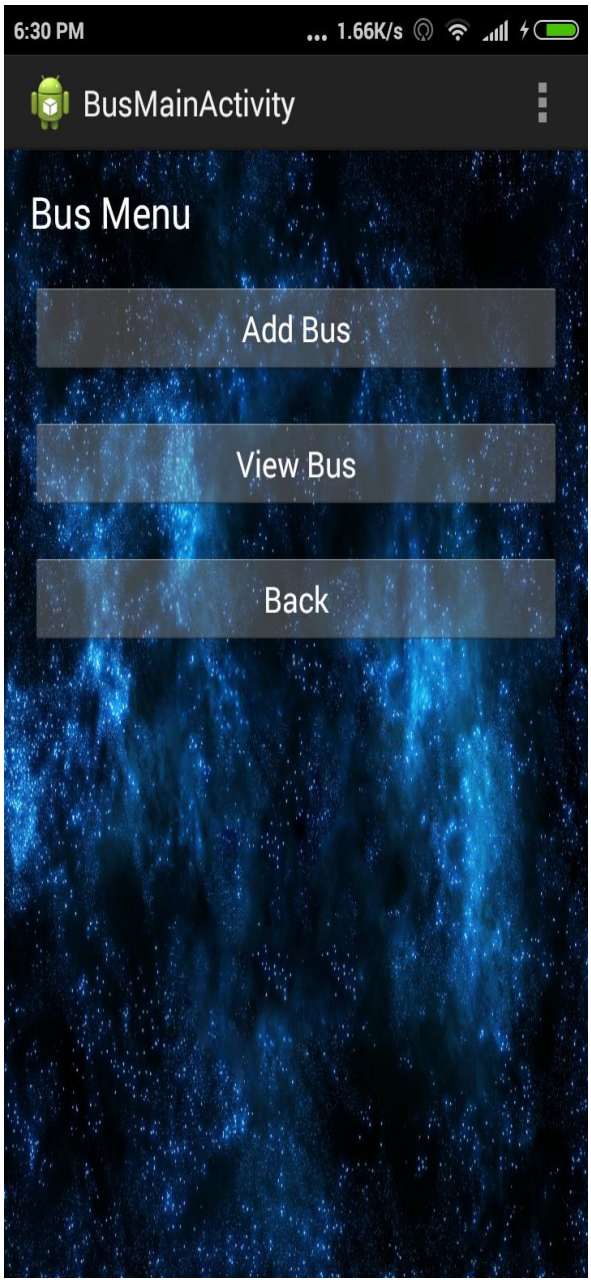


Fig 6.1c Student menu activity

Fig 6.1d Bus main activity

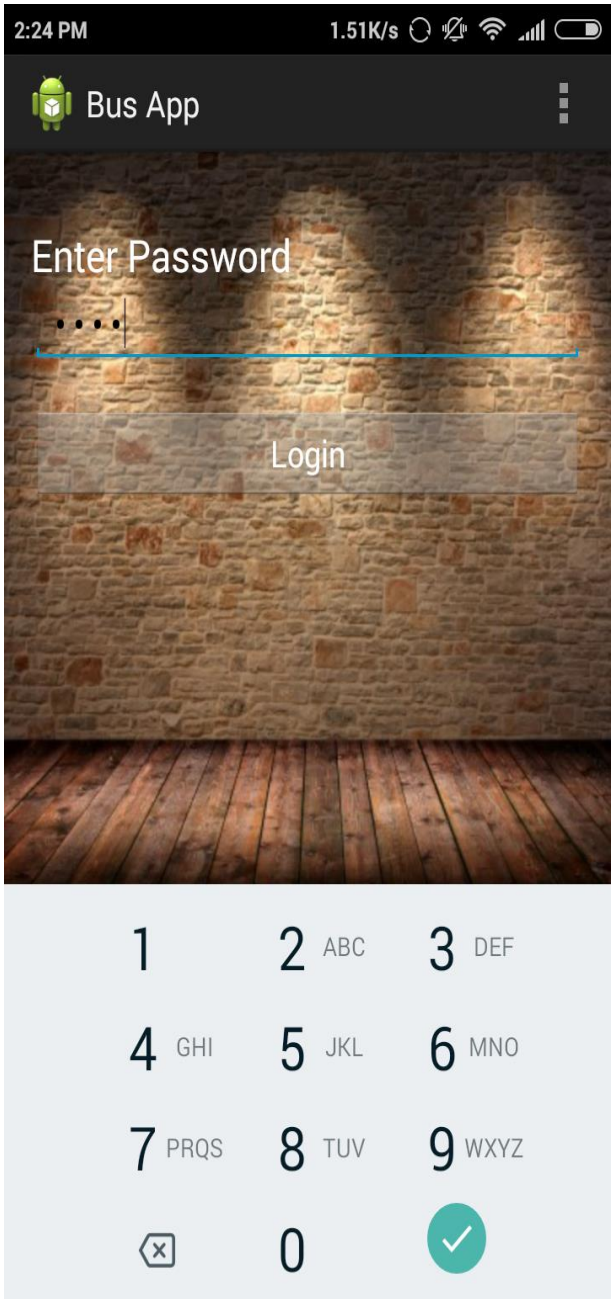


Fig 6.2a Bus App

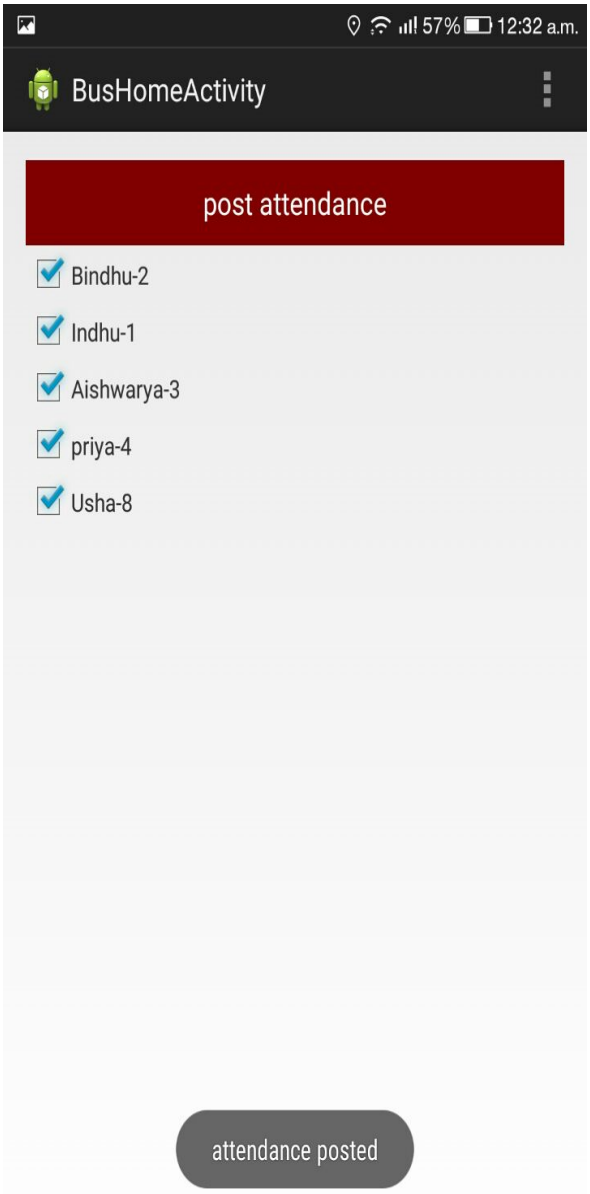


Fig 6.2b Attendance posted

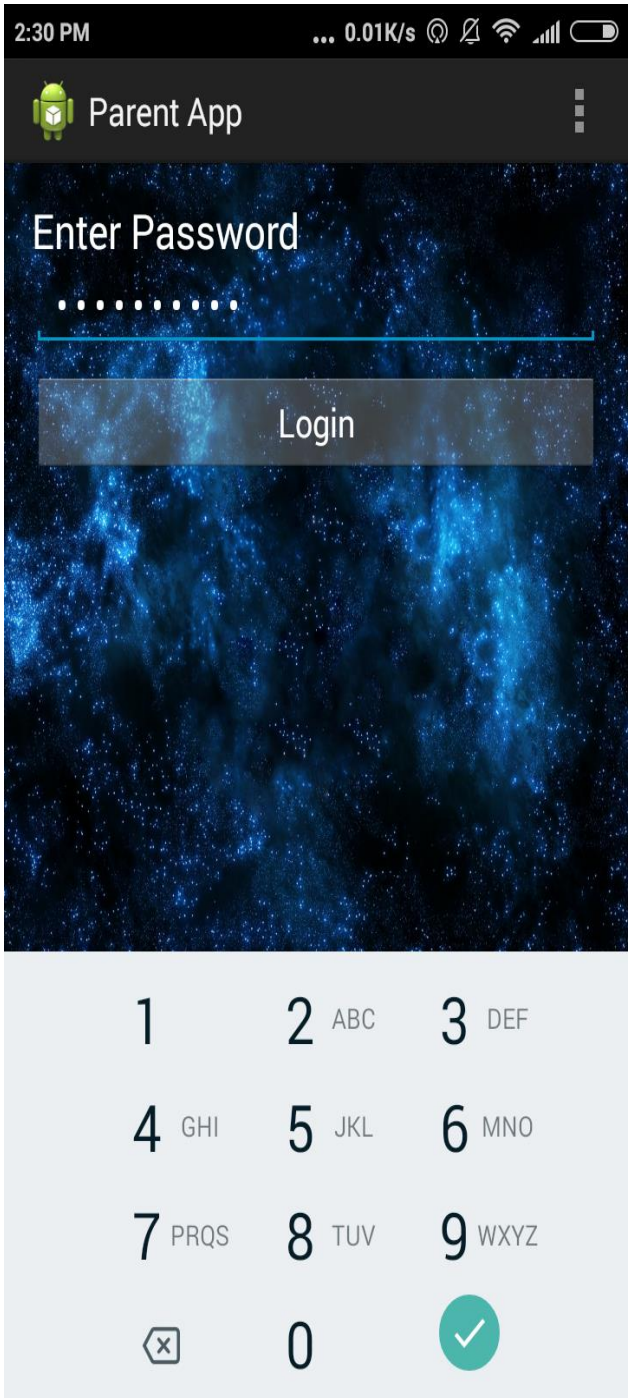


Fig 6.3a Parent app

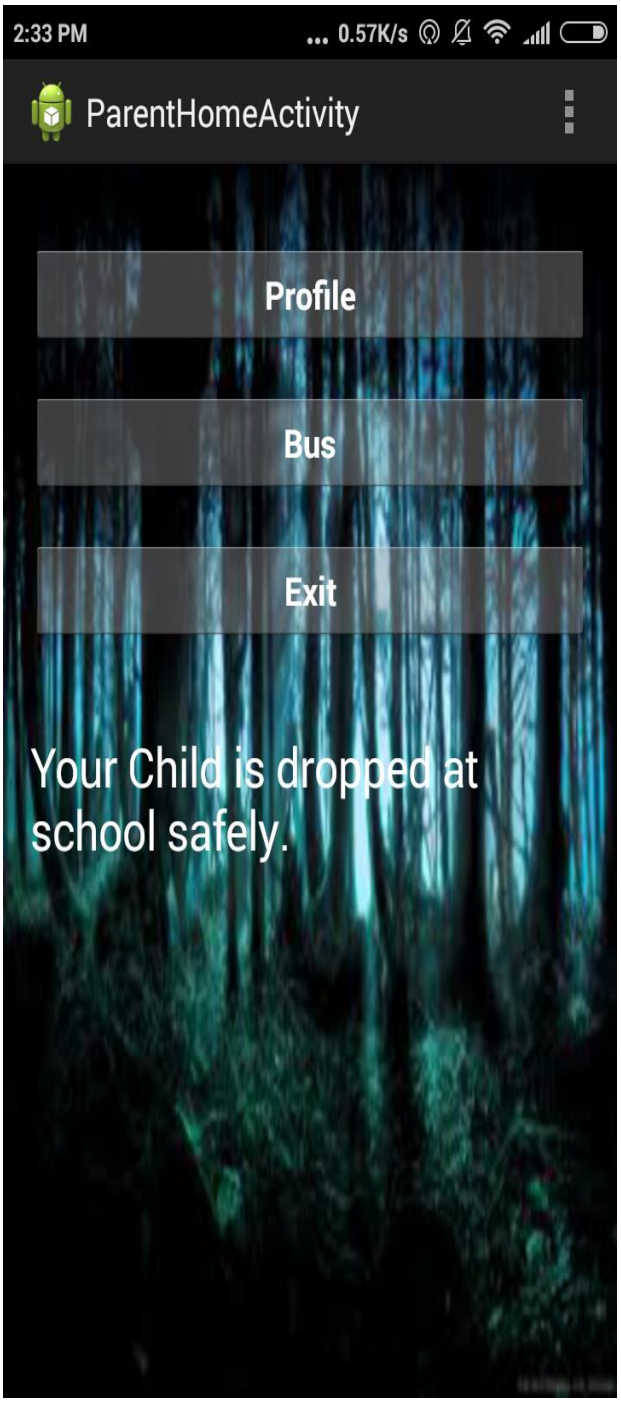


Fig 6.3b Parent home activity

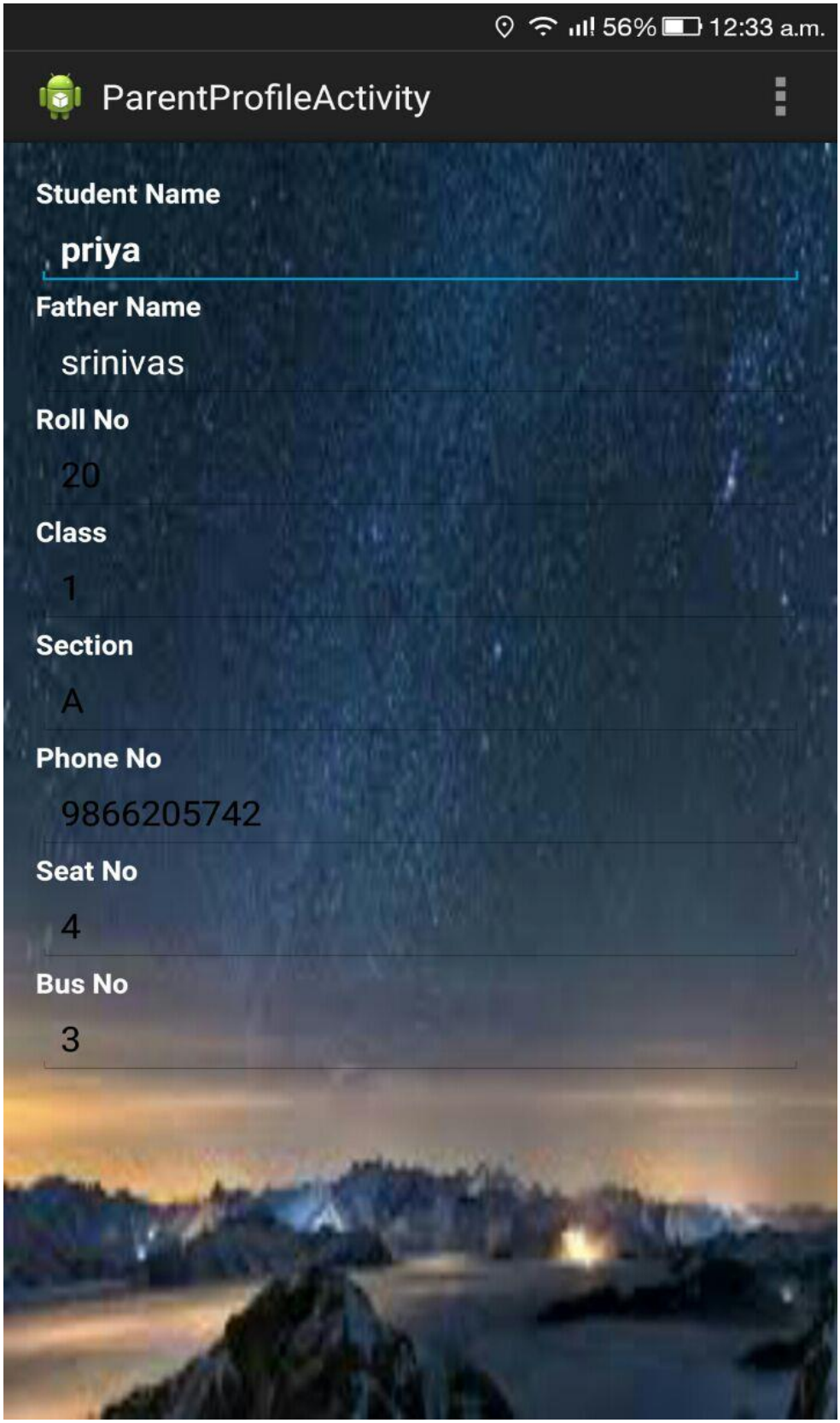


Fig 6.3c Parent profile

Fig 6.3d Location of bus

7. CONCLUSION:

Problems occurred now days for child's safety have become a major concerning issue for the parents and school too. This paper proposes to develop an application that aims at enhancing the safety of children during the daily bus trip to and from the school. The application include a bus grid which include a child name attendance is posted by the bus driver. The application updates after each action which is performed in particular application. The parent login with the help of his mobile number and check whether his/her child climbed the bus or not.

Our Future Scope includes the real time implementation of the proposed system with the additional features like parking management system for the school buses and more in an effective manner. In addition, if the bus crosses the speed limit while traveling the bus, the alert notification message will be send directly to the School Unit. School Unit will have the web application and the android application where it will notify each and every moment of the bus to the admin. In addition, when the bus is at one source or the stop then the Notification is send to the Parent Unit staying at the next source i.e. having next stop (Example: "The next stop is yours") with driver.