

SCHOOL MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted By

VISHAL CHATURVEDI

2100290140150

SHAGUN

2100290140119

VISHAL RAJPUT

2100290140151

**Submitted in partial fulfilment of the
Requirements for the Degree of**

MASTER OF COMPUTER APPLICATION

**Under the Supervision of
Ms. Divya Singhal
Assistant professor**



Submitted to

**DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206**

(DEC 2022)

CERTIFICATE

Certified that **Vishal Chaturvedi 210029014087298, Shagun 210029014070590, Vishal Rajput 2100290140187698** has/ have carried out the project work having “**School Management System**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Vishal Chaturvedi (2100290140150)

Shagun (2100290140150)

Vishal Rajput (2100290140150)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

**Ms. Divya Singhal
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad**

Signature of Internal Examiner

Signature of External Examiner

**Dr. Arun Kumar Tripathi
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad**

ABSTRACT

The main objective of School Management System is to automate all functionalities of a school or colleges. Using this system you can manage all school management work like admission, fees submission, time table management and result declaration. Using this school management system, you can view or update data and information about students and staff easily. This system helps in managing the activity like student admission, student registration, fees submission. Admin can also retrieve information of employee student.

ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Ms. Divya Singhal** for her guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi**, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions. Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Vishal Chaturvedi

Shagun

Vishal Rajput

TABLE OF CONTENT

Title Page	i
Certificate	ii
Abstract	iii
Acknowledgement	iv
Table Of Content	
1. INTRODUCTION	1
1.1 Problem Definition	4
1.2 Project Overview/Specifications	5
1.3 Hardware Specification	8
1.4 Software Specification	8
2. LITERATURE SURVEY	
2.1 Existing System	10
2.2 Proposed System	12
2.3 Feasibility Study	14
3. SYSTEM ANALYSIS & DESIGN	19
3.1 Data Flow Diagrams	25
3.1.1 Zero Level DFD	28
3.1.2 One Level DFD	29
3.2 Sequence Diagram	30
3.3 E-R Diagram	31
4. ALGORITHMS AND PSEUDO CODE	32
5. TESTING PROCESS	45
6. RESULTS / OUTPUTS	51
7. FUTURE SCOPE OF THE PROJECT	55
8. CONCLUSIONS / RECOMMENDATIONS	56
9. BIBLIOGRAPHY	57

INTRODUCTION

INTRODUCTION

The main objective of School Management System is to automate all functionalities of a school or colleges. Using this system you can manage all school management work like admission, fees submission, time table management and result declaration. Using this school management system, you can view or update data and information about students and staff easily. This system helps in managing the activity like student admission, student registration, fees submission.

Admin can also retrieve information of employee student.

The SCHOOL MANAGEMENT SYSTEM can be used to store student information like attendance, fees, and student result etc. admin can create report regarding any student any time using this system. Using this system, you can register new student and their course details. You can submit students' fees and can check fees details anytime. You can create exam result and submit in this system. Student can check their result online by logging to the system. You can also add new employee in the system and can check details of the employee easily. Student can also check course detail online from this system.

Using this system you can manage all information of all aspects of a college, its students, faculties, Departments, marks and other curricular activities. School Management System provides the easiest way to manage all functionalities of a college. This system facilitates colleges to maintain the functionality related to school employees and their students.

School Management System can store and manage all data of the various departments of a school like Administration, Attendance, Staff details etc. using this system user can retrieve any information related to student, teacher and fees. Using this system teacher can check student attendance anytime. This system also help teacher to announce the result. School administration can also manage school work easily. Admin can check leave, salary and other details of teacher any time. They can also create time table of classes from this system. The Library module is used for the data process of library and book accessing for students and staffs.

School management system using Java and Firebase is an android application. School Management Project is software that is helpful for students as well as the school authorities. In the current system, all the activities are done manually. It is very time-consuming and costly. Our online School Management System deals with the various activities related to the students.

Project Name	Student Management System
Language Used	Java
Database	Firebase
User Interface Design	XML
Operating System	Android
Software	Android Studio

Problem Definition

Students is a main part of any institution that concerns with. But the institutions find it difficult to keep details of so many students of the organization just in one stretch. It will involve a lot of pens paper work. Sometimes there will be some huge heap of files bundled up and kept together in some corner of the office. If you want any information regarding the particular student then it can be obtained by just entering the roll number or the name of the student to be searched. This School Management System will make the work of storing the data in an organized way. The School Management System application will help in managing the student's information will become easier with one such system. It will also help in saving time and effort. The user interface of this system is user friendly and easy to understand. The information of the particular student will be obtained in just one click.

Here some problem point:

- The existing system which we using in our college is traditional process is a complete manual process.
- Now-a-days, education is playing very significant role in the society. Day-by-day, the percentage of illiterates are decreasing and the percentage of literates is increasing.
- Education will change the society in all the aspects and everyone wants to study higher professional degrees.
- Admissions are increasing day by day so there by ratio of establishment new colleges and schools are also increasing.
- But the actual challenge is starting from now. Most of the schools and colleges are maintain student information in records.
- When the number of records increased, it is difficult to maintain the information of each student in the old manual system.
- Maintaining the records manually leads to error prone and required more man power and it consumes more time for processing the records.

Project Overview/Specification

School Management System (also known as a student information system or SIS) helps a school manage data, communications, and scheduling. A school system generates and uses a large amount of data. This data must be communicated appropriately to students, faculty, and parents. **School Management System Project Modules**

The three main users involved in this system are

1. User (i.e., Students)
2. Teacher
3. Admin

Admin:

1. **Dashboard:** In this section, admin can see all detail in brief like Total Classes, Total Students, Total Class Notices and Total Public Notices.
2. **Class:** In this section, admin can manage class (Add/Update/Delete).
3. **Students:** In this section, admin can manage the students (Add/Update/Delete).
4. **Notices:** In this section, the admin can manage notices (Add/Update/Delete).
5. **Public Notices:** In this section, the admin can manage public notices.
6. **Pages:** In this section admin, can manage about us and contact us page of administration
7. **Search:** In this section admin, can search students by their student's name.
8. **Reports:** In this section admin, can view how much students have been registering in particular period.
9. Admin can also update his profile, change the password and recover the password.

Teacher:

1. **Dashboard:** In this section, admin can see all detail in brief like Total Classes, Total Students, Total Class Notices and Total Public Notices.
2. **Class:** In this section, Teacher can manage class (Add/Update/Delete).
3. **Students:** In this section, Teacher can manage the students (Add/Update/Delete).
4. **Notices:** In this section, the Teacher can manage notices (Add/Update/Delete).
5. **Search:** In this section Teacher, can search students by their student's name.
6. Teacher can also update his profile, change the password and recover the password.

User (Students):

1. **Dashboard:** It is welcome page for students.
2. **View Notices:** In this section, user can view notices which are announced by administrator.
3. Student can also view his profile, change the password and recover the password.

MODULES USED IN THE PROJECT

This project includes the following modules for development of the project. These are as follows: -

1. SPLASH SCREEN

This is a first screen that displays the welcome screen for the user and also shows the information of developer or version etc.

2. LOGIN SCREEN

This login shows the Login name and password when user enter a valid user name and password then he/she can operate the application.

3. REGISTER SCREEN

User and teacher will register to operate the application. If the student and teacher will be verified by the admin, then the application will be able to run.

4. STUDENT FORM SCREEN

This screen provides the option to add, modify, delete or find the information of a student who seeks the admission in the school.

5. ADD NOTICES SCREEN

This screen provides the option to add, delete, search and delete the information of notices.

5. ADD NOTES OR STUDY MATERIAL SCREEN

From this screen the teacher can upload and delete his notes and study material

6. FEE PAYMENT SCREEN

Through this screen you can deposit school fees. From this screen you can see your dues and last payment

7. MEETING SCREEN

Student can attend the online classes by using android app. Student put question by raise your hand option.

Hardware Specification

Hardware Requirements of Student Management System The hardware required for the development of the project is:

- **Processor:** Intel i3 8th gen System
- **Processor Speed:** 2GHz or more
- **Ram:** 4 GB Ram
- **Hard Disk:** 40 GB

Software Specification

Software Requirements of Student Management System The software required for the development of the project is:

- **Operating System:** Windows 10
- **Software:** Android Studio
- **Language:** Java
- **Back-end:** Firebase

LITERATURE SURVEY

LITERATURE SURVEY

EXISTING SYSTEM

- The main problem projected is that pupil's particulars are reported manually in distinct records, which is a laborious job. Handling and updating these records manually increase the chances of mistakes. It takes a lot of time and needs many employees to accomplish the task. It even lacks security and disability to produce various types of reports.
- They imply a high entry barrier for educators and students who have to learn the new tools. This is a well-known problem related to the usage of IT solutions as education- supporting elements
- In this section, we will discuss the existing system and some of its drawbacks which force us to plan this whole idea of developing an online college management system. Let us take an example Initially colleges appoint many employees to maintain the record of the student, teachers, other staff, employees' salaries, and student's fees.
- If any student wants to submit the fee, he/she need to go to the fee counter for paying their fees and wait for one to two days for updating and again went to the fee office to know the status, and this whole record uses to maintain on card and files so there might be chances of missing.
- Suppose if some student comes and asks for the status of his fee after one month then the worker will open his file and search it manually for the student's name which will take time. Suppose we want to organize an event in college we need to create posters, banners and make an announcement in classes so that students participate but if any student got absent on that day, then he will remain unnotified.

LIMITATION OF EXISTING SYSTEM:

- It was limited to a single system.
- It was less user-friendly.
- It has a lot of manual work (Manual system does not mean that we are working with pen and paper, it also includes working on spread sheets and other simple software's)
- It requires more no of employees need to work.
- It was time consuming process.
- The present system was very less secure.
- It is unable to generate different kinds of report.
- There is a threat to the record of the student and teachers, in this case, there might be a chance that a person makes entry on someone else record.
- There is no proper way of getting to know about the events and extra curriculum activities happening in colleges.
- It might be the case that student tries to bribe the teachers to avoid the long queue.

PROPOSED SYSTEM

In our proposed system we have the provision for adding the details of the students by themselves. So, the overhead of the school authorities and the teachers is become less. Another advantage of the system is that it is very easy to edit the details of the student and delete a student when it found unnecessary. The marks of the student are added in the database and so students can also view the marks whenever they want.

- A student can register himself directly on the college application for any notification of the event and for depositing fee and students can also post their queries and check their fee status.
- If the students are not aware of policies, then the admin will make him contact with this application and help them to complete the whole process.
- Admin will maintain the record and atomicity by using the proper table to save records so that no one can access it and data remain safe.
- Students need not bother about his/her fee deposit and without moving from their place or making so many phone calls they can make the fee payment.
- No need of worrying about status you can check that from your place.
- Students can check new information and achievements of the college students through the application which will motivate them and all things would be uploaded by the admin on the application.

The whole process of getting work done will become faster than the conventional way of getting insurance. Let us look at the different aspects which we have designed to make the website work properly. Here we are not discussing the admin module because all other modules will cover the admin module part so to remove the same content from the report, we are ignoring the admin part.

Our proposed system has several advantages

- User friendly interface
- Fast access to database
- Less error
- More Storage Capacity
- Search facility
- Look and Feel Environment
- Quick transaction

All the manual difficulties in managing the student details in a school or school have been rectified by implementing computerization.

We resolve all the disadvantages of existing system in our proposed system, CMS. Old students can share their experience, ideas, motivations to their juniors in the college. They can provide information regarding higher studies. RESULT Automation which yields aggregate percentage up to that instant. Time will be saved, no chance of getting error.

FEASIBILITY STUDY

FEASIBILITY STUDY

For all new systems, the requirements engineering process starts with feasibility study. The input to the feasibility study is an outline description of the system and how it will be used within an organization. The result of the feasibility study should be a report which recommends whether or not it is worth carrying on with the requirements engineering and system development process.

The Requirement Engineering Process:

Based on these requirements we have carried out a feasibility study from three perspectives viz. Economic, Technical and Social Feasibility. The following pages describe these studies.

Economic Feasibility

One of the aims of developing the School Management System project was to provide such an alternative to the current system, which is highly productive while remaining economically feasible. School Management System is an economically feasible alternative to the current system.

School Management System can perform all the tasks of management at Products in an efficient manner.

It is true that a significant amount of investment is needed to setup the automated system. This cost includes the computer hardware and software expense. The college management System may even require a trained person to operate the system. This cost withstanding School Management System can make up for the expense in a short period of time.

The old system required for manual maintenance of registers. School Management System does away with the need for these registers. Thus, a significant amount of money can be saved in the form of reduced stationary cost. Thus, this can give a boost to their environment friendly image. School Management System with its fast and accurate transaction handling can significantly improve the productivity of the school management System. The manual system is prone to human errors which can cost the Items loss in terms of money and goodwill. Thus, we can conclude that School Management System is an economically feasible alternative to the present system.

Technical Feasibility

With the use of the tools and resources which were available with us, it was highly feasible to develop school management system. We did a detailed study of the various requirement of the system and on in-depth analysis concluded that it was possible to technically develop the project.

All the requirements of the School Management System are satisfied by Manager. It is able to efficiently mimic the present system with many added features and productivity tools. Thus, we can conclude that School Management System project is technically feasible to develop and implement.

Social Feasibility

For any proposed system to be successful, it has to be accepted by the people who work on that system. A system which is readily accepted by its users as a credible and workable environment is a successful system.

One of our very basic aims when setting out to develop this project was to gain user acceptance. With the present system being in use for eternal time, it is difficult to convince the Inventory Control System to leave the present system and move on to a computerized system.

This problem is not so prominent in case of new Stock being setup. We have tried to add all such features to Stock manager which make it easier for the management to migrate to Stock Manager. One of the basic necessities was to provide a simple and attractive user interface.

Stock Manager is successful in doing so. The other aim was to make the working as simple and straight forward as possible. Stock Manager is successful in meeting this requirement also.

SYSTEM ANALYSIS AND DESIGN

SYSTEM ANALYSIS AND DESIGN

The following requirements are raised during the analysis of the needs of the users:

- A Person Should be able to login to the system through the first page of the Application.
- The Administrator can create users as per user requirement.
- Admin can upload the data for a particular Student. On successful completion of upload, user (Student/Parent/Faculty) can view reports.
- A general user will have access to see the status of particular Student id number.
- Student (user) can use all the facilities, same as which are provided to him in the college.
- Student can see attendance, notices, grades, report and other facilities in updated manner.
- There will be a separate page for every student as his account in which he can get notices, attendance, grades, assignments etc.
- Parent can just view the record of student with the username and password provided.
- Faculty can give the attendances and notices for the students.
- The administrator verifies all these reports and generates them for users to view them.

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

During this, the developer focuses on three tasks: -

- Defining the problem and deciding whether to proceed.
- Analyzing the current system in depth and developing possible solution to the problem.
- Selecting the best solution and defining its function.

Identification of needs is done in order to understand the problem the software system is to solve. The problem could be automating an existing manual process, developing a new automated system, or a combination of the two. For large systems that have many features, and that need to perform many different tasks, understanding the requirements of the system is a major task. The emphasis in requirements analysis is on identifying what is needed from the system, not how the system will achieve its goals. This task is complicated by the fact that there are often at least two parties involved in software development- a client and a developer. The developer has to develop the system to satisfy the client's needs. The developer usually does not understand the client's problem domain, and the client does not understand the issues involved in the software systems. In most software projects, the need ends with a document describing all the requirements. In other words, the goal of the requirements specification is to produce the software requirements specification. The person responsible for requirements needed is often called the analyst. The knowledge gathered regarding the current system is documented in several different ways. Some analysts use data flow diagrams, which show the flow of data

Analysis Model

This document plays a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure:
 1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
 2. Defining the requirements of the second prototype.
 3. Planning and designing the second prototype.
 4. Constructing and testing the second prototype.

- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
- The final system is constructed, based on the refined prototype.
- The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

STUDY OF THE SYSTEM

Graphical user interface

In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a browser interface. The GUI'S at the top level have been categorized as

1. Administrative user interface
2. The operational or generic user interface

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The interfaces help the administrations with all the transactional states like Data insertion, Data deletion and Date updating along with the extensive data search capabilities.

The operational or generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.

Number of Modules

The system after careful analysis has been identified to be presented with the following modules:

The modules involved are:

1. School information: Through this service one can access the complete information about the school campus such as courses available, admission procedure, placements, school events, achievements etc.
2. Student tracking: Any company or any organization that want to check the summary about the student of the college, so that they will be able to choose the particular students for their campus placement and for that purpose they will be given a particular link through which they can access the information required.
3. Student attendance status: It gives the attendance status of students. Faculty will update the attendance periodically and can be seen by students and parents.
4. Student's performance in exams: This facility provides the performance of the student in each exam which is conducted by university or school such as midterm performance. Marks obtained by students in exams will be updated by faculties that can be access by students and parents.
5. Exam Notification: This facility notifies students and parents about examination schedule.
6. Events: it will give information about different events that will be conducted by school time to time. Information about these events will be updated by administrator.

DATA FLOW DIAGRAMS

The development of DFD's is done in several levels. Each process in lower-level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consists a data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagram shows the actual implements and movement of the data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

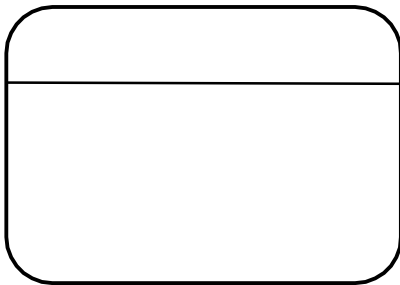
The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for an analyst to understand the process. Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this led to the modular design.

A DFD is also known as 'Bubble Chart' has the purpose of clarifying system requirements and identifying major transformation that will become programs in system design. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

DFD SYMBOLS:

In the DFD, there are four symbols

1. A square defines a source (originator) or destination of system data.
2. An arrow identifies data flow. It is the pipeline through which the information flows.
3. A circle or a bubble represents a process that transforms incoming dataflow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data.



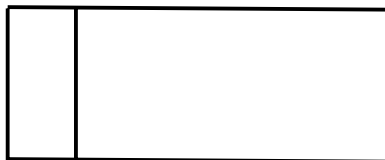
Process that transforms data flow



Source or Destination of Data



Data Flow



Data Store

CONSTRUCTING A DFD:

Several rules of thumb are used in drawing DFD's:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower-level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each word capitalized.

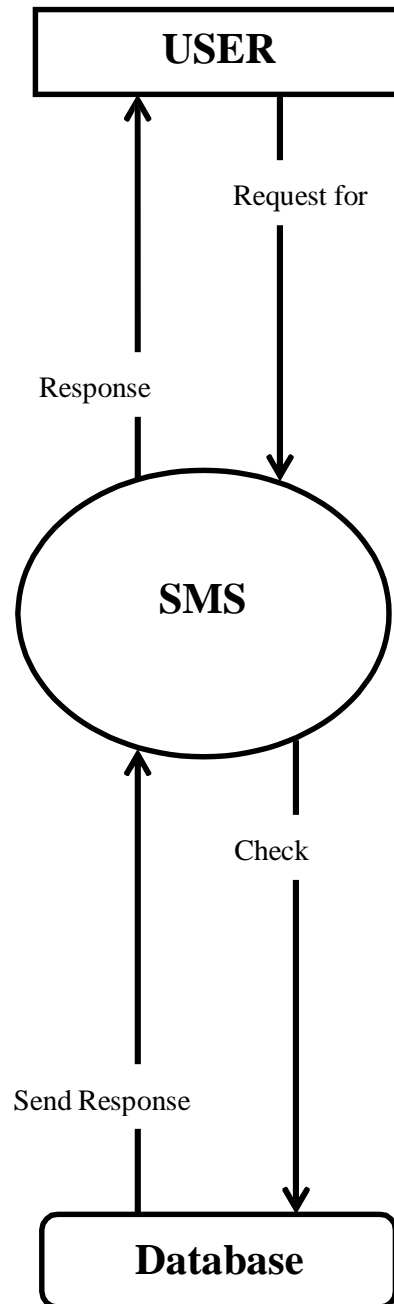
A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out. Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

SAILENT FEATURES OF DFD's

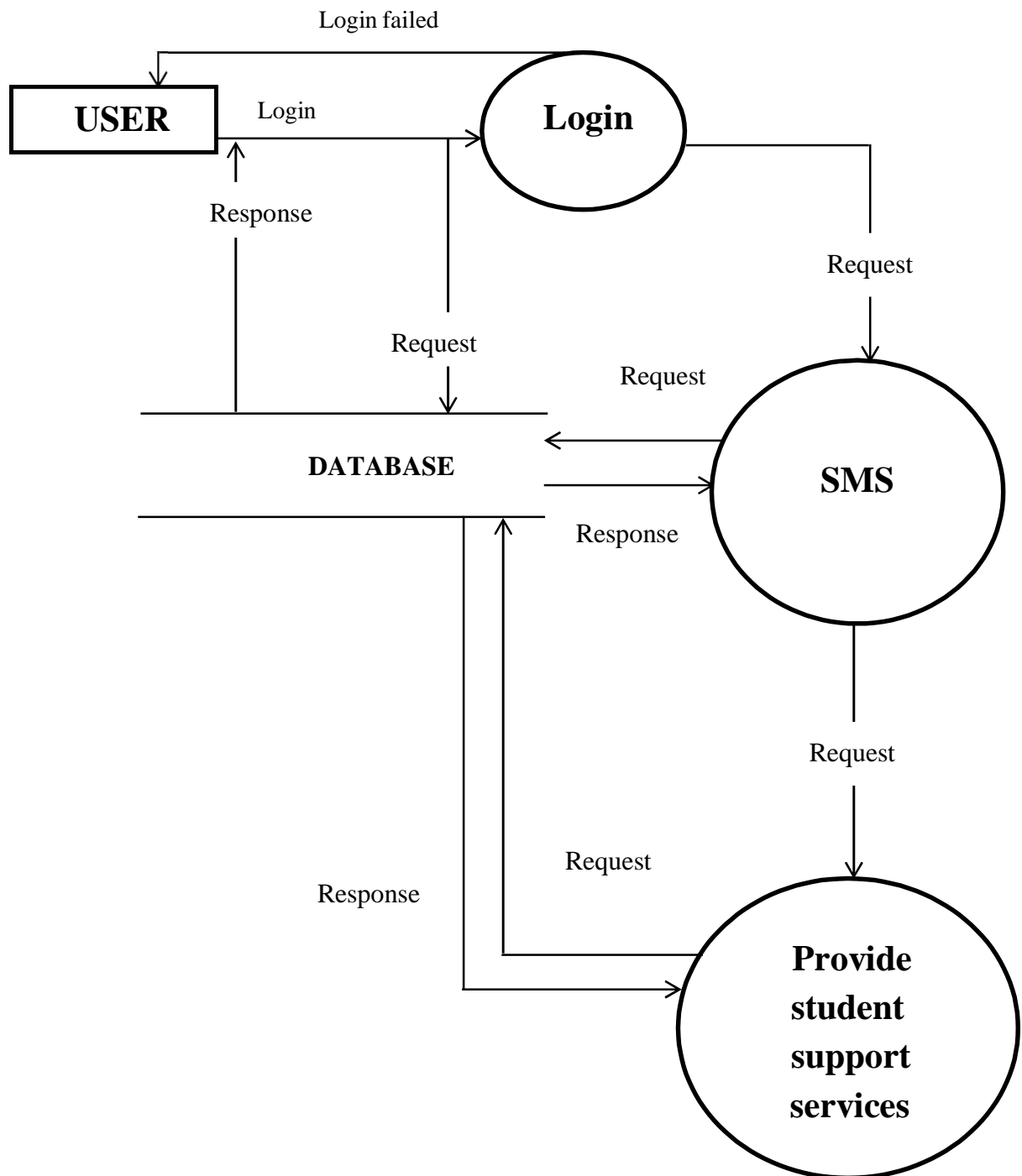
1. The DFD shows flow of data, not of control loops and decisions are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process whether the dataflow take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

DATA FLOW DIAGRAM

- **0(Zero) Level DFD**



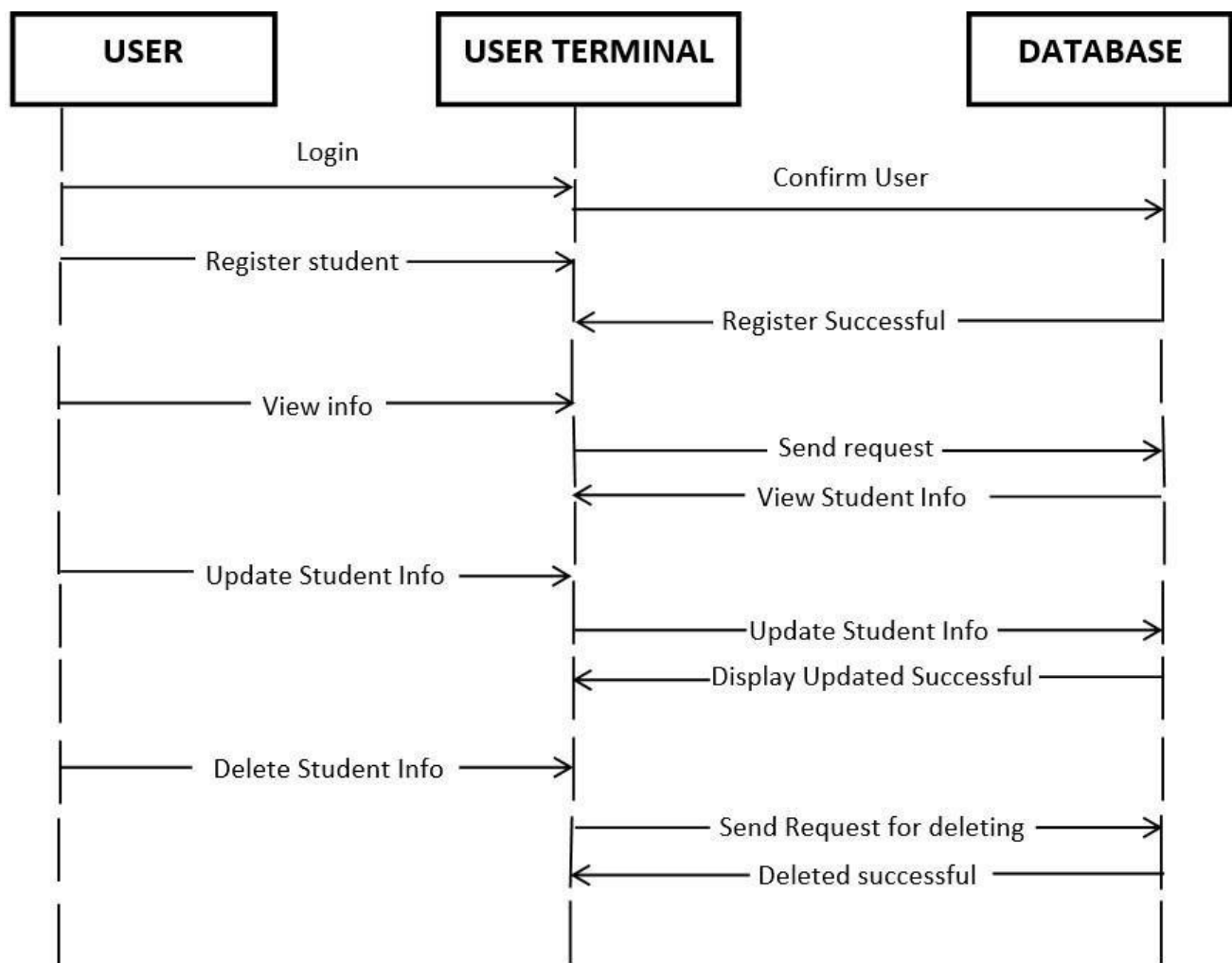
1 (One) Level DFD



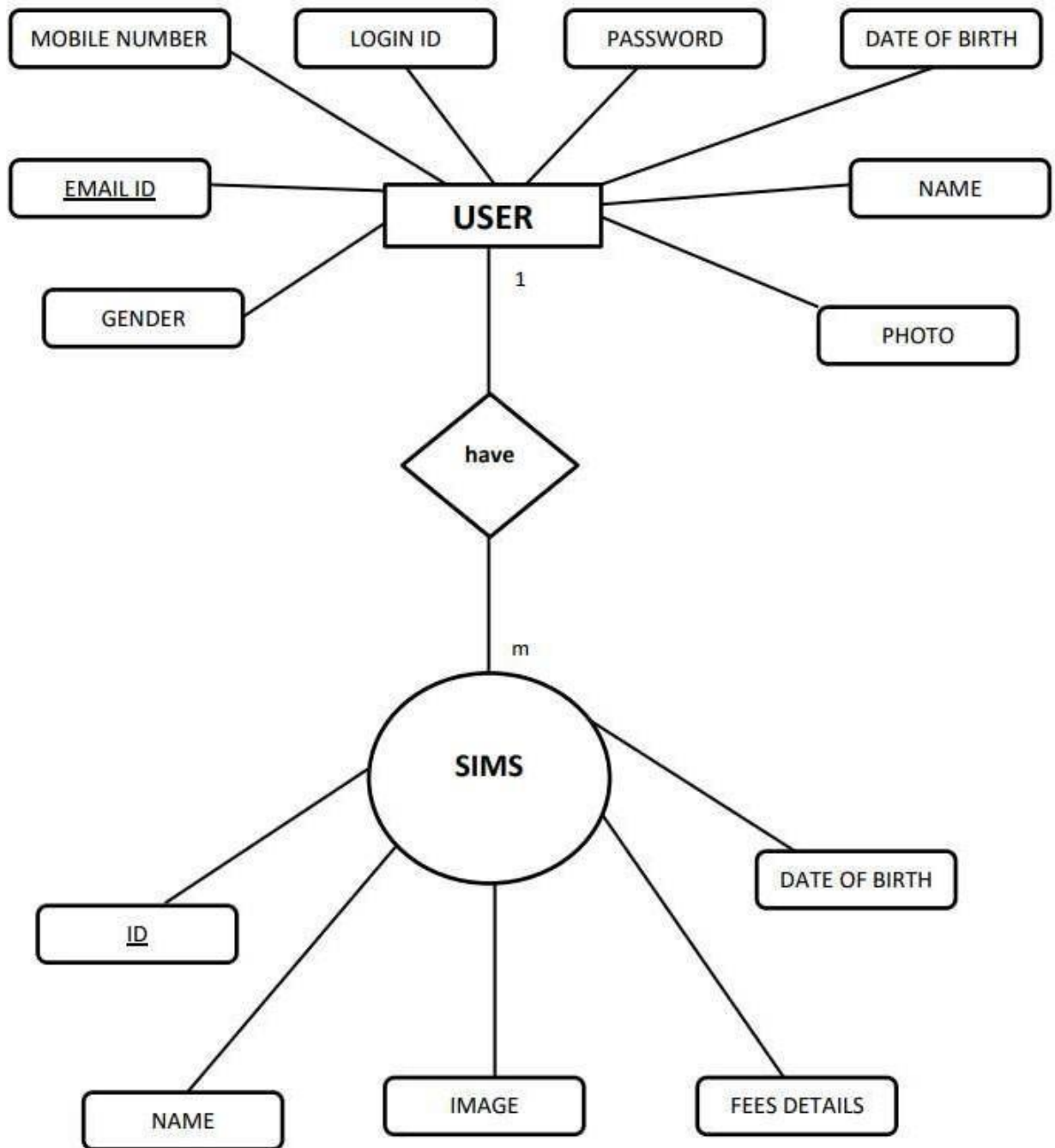
SEQUENCE DIAGRAM:

UML sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system.

A sequence diagram is a two dimensional in nature. On the horizontal axis, it shows the life of the object that it represents, while on the vertical axis, it shows the sequence of the creation or invocation of these objects.



ENTITY RELATIONSHIP DIAGRAM:



ALGORITHMS AND PSEUDO CODE

ALGORITHMS AND PSEUDO CODE

Here some main modules code:

1. Login Activity

```
public class LoginActivity extends AppCompatActivity {
    private EditText email, passowrd;
    private Button login, register;
    private ProgressBar progressbar;
    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        mAuth = FirebaseAuth.getInstance();
        email = findViewById(R.id.email);
        passowrd = findViewById(R.id.password);
        login = findViewById(R.id.login);
        register = findViewById(R.id.register);
        progressbar = findViewById(R.id.progressBar);
        register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(getApplicationContext(), RegisterActivity.class));
            }
        });
        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                loginUserAccount();
            }
        });
    }
}
```

```
});  
}
```

```
private void loginUserAccount() {  
    progressbar.setVisibility(View.VISIBLE);  
    String emailLogin, password;  
    emailLogin = email.getText().toString();  
    password = passowrd.getText().toString();  
    if (TextUtils.isEmpty(emailLogin)) {  
        Toast.makeText(getApplicationContext(),  
            "Please enter email!!",  
            Toast.LENGTH_LONG)  
            .show ();  
        return;  
    }  
  
    if (TextUtils.isEmpty(password)) {  
        Toast.makeText(getApplicationContext(),  
            "Please enter password!!",  
            Toast.LENGTH_LONG)  
            .show();  
        return;  
    }  
    mAuth.signInWithEmailAndPassword(emailLogin, password)  
        .addOnCompleteListener(  
            new OnCompleteListener<AuthResult>() {  
                @Override  
                public void onComplete(  
                    @NonNull Task<AuthResult> task) {  
                    if (task.isSuccessful()) {  
                        Toast.makeText(getApplicationContext(),  
                            "Login successful!!",  
                            Toast.LENGTH_LONG)
```

```

        .show();
        progressbar.setVisibility(View.GONE);
        Intent intent
            = new Intent(LoginActivity.this,
                MainActivity.class);
        startActivity(intent);
    } else {
        Toast.makeText(getApplicationContext(),
            "Login failed!!",
            Toast.LENGTH_LONG)
            .show();
        progressbar.setVisibility(View.GONE);
    }
}

@Override

protected void onStart() {
    super.onStart();
    if (mAuth.getCurrentUser() != null) {
        Intent i = new Intent(LoginActivity.this, MainActivity.class);
        startActivity(i);
        finish();
    }
}
}

```


2. Registration Activity

```
public class RegisterActivity extends AppCompatActivity {
    public static final String TAG = "TAG";
    AutoCompleteTextView Class,role;
    EditText fName, email, password, phone;
    Button mRegisterBtn;
    FirebaseAuth fAuth;
    FirebaseFirestore fStore;
    String userID;
    String [] classArr = {"1st", "2nd", "3rd", "4th", "5th", "6th", "7th", "8th", "9th", "10th", "11th",
"12th"};
    String[] roleArr = {"Teacher","Student"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        fName = findViewById(R.id.fullName);
        email = findViewById(R.id.email);
        password = findViewById(R.id.password);
        phone = findViewById(R.id.phone);
        role = findViewById(R.id.role);
        Class = findViewById(R.id.Class);
        mRegisterBtn = findViewById(R.id.teacherRegister);
        fAuth = FirebaseAuth.getInstance();
        fStore = FirebaseFirestore.getInstance();
        if (fAuth.getCurrentUser() != null) {
            startActivity(new Intent(getApplicationContext(), MainActivity.class));
            finish();
        }
        ArrayAdapter<String> classAdapter = new ArrayAdapter<String>
            (this, android.R.layout.select_dialog_item, classArr);
```

```

Class.setThreshold(2);
Class.setAdapter(classAdapter);

ArrayAdapter<String> roleAdapter = new ArrayAdapter<String>
    (this, android.R.layout.select_dialog_item, roleArr);
role.setThreshold(2);
role.setAdapter(roleAdapter);

mRegisterBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        final String emailString = email.getText().toString().trim();
        String passwordString = password.getText().toString().trim();
        final String fNameString = fName.getText().toString();
        final String phoneString = phone.getText().toString();
        if (TextUtils.isEmpty(emailString)) {
            email.setError("Email is Required.");
            return;
        }
        if (TextUtils.isEmpty(passwordString)) {
            password.setError("Password is Required.");
            return;
        }
        if (password.length() < 6) {
            password.setError("Password Must be >= 6 Characters");
            return;
        }

        fAuth.createUserWithEmailAndPassword(emailString,
passwordString).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override

```

```

public void onComplete(@NonNull Task<AuthResult> task) {
    if (task.isSuccessful()) {

        FirebaseUser fuser = FirebaseAuth.getCurrentUser();
        fuser.sendEmailVerification().addOnSuccessListener(new
OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Toast.makeText(RegisterActivity.this, "Verification Email Has been
Sent.", Toast.LENGTH_SHORT).show();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Log.d(TAG, "onFailure: Email not sent " + e.getMessage());
            }
        });

        Toast.makeText(RegisterActivity.this, "User Created.",
Toast.LENGTH_SHORT).show();
        userID = FirebaseAuth.getCurrentUser().getUid();
        DatabaseReference documentReference =
fStore.collection("users").document(userID);
        Map<String, Object> user = new HashMap<>();
        user.put("fName", fNameString);
        user.put("email", emailString);
        user.put("phone", phoneString);
        user.put("feesAmount", 0);
        user.put("classes", Class.getText().toString().trim());
        user.put("role", role.getText().toString().trim());
        user.put("status", "pending");
    }
}

```

```

        user.put("uID", userID);
        documentReference.set(user).addOnSuccessListener(new
OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Log.d(TAG, "onSuccess: user Profile is created for " + userID);
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure( @NonNull Exception e) {
                Log.d(TAG, "onFailure: " + e.toString());
            }
        });
        startActivity(new Intent(getApplicationContext(), MainActivity.class));

    } else {
        Toast.makeText(RegisterActivity.this, "Error! " +
task.getException().getMessage(), Toast.LENGTH_SHORT).show();

    }
}

});

}

});

}

```

3. Meeting Fragment:

```
public class Meeting extends Fragment {
    Button joinBtn;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View v= inflater.inflate(R.layout.fragment_meeting, container, false);
        joinBtn = v.findViewById(R.id.joinBtn);
        try {
            JitsiMeetConferenceOptions options = new JitsiMeetConferenceOptions.Builder()
                .setServerURL(new URL(""))
                .setWelcomePageEnabled(false)
                .setFeatureFlag("invite.enabled",false)
                .setFeatureFlag("help.enabled",false)
                .build();
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        joinBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                EditText editText;
                editText = v.findViewById(R.id.customAmount);
                String text = editText.getText().toString();
                if (text.length() > 0) {
                    JitsiMeetConferenceOptions options
                        = new JitsiMeetConferenceOptions.Builder()
                            .setRoom(text)
                            .setFeatureFlag("invite.enabled",false)
                            .setFeatureFlag("help.enabled",false)
```

```
        .build();
        JitsiMeetActivity.launch(getContext(), options);
    }
}
});

return v;
}
}
```

4. Fees Payment Fragment

```
public class Fees extends Fragment implements PaymentResultListener {
    TextView sName, sClass, sPhone, amount, PaidAmount;
    EditText customAmount;
    Button PayNow;
    FirebaseAuth mAuth;
    FirebaseFirestore fStore;
    String uID, feesClass,samount;

    Double leftAmount, totalAmount, feesPaid;
    public static final String TAG1 = "TAG";

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View v = inflater.inflate(R.layout.fragment_fees, container, false);

        // sName = v.findViewById(R.id.sName);
        // sClass = v.findViewById(R.id.sClass);
        amount = v.findViewById(R.id.amount);
        customAmount = v.findViewById(R.id.customAmount);
        PayNow = v.findViewById(R.id.PayNow);
        PaidAmount = v.findViewById(R.id.PaidAmount);
        mAuth = FirebaseAuth.getInstance();
        fStore = FirebaseFirestore.getInstance();
        uID = mAuth.getCurrentUser().getUid();
        DocumentReference documentReference = fStore.collection("users").document(uID);
        documentReference.addSnapshotListener(new EventListener<DocumentSnapshot>() {
            @Override
            public void onEvent(@Nullable DocumentSnapshot value, @Nullable
```

```

FirebaseFirestoreException error) {
    // sName.setText(value.getString("fName"));
    String fName = value.getString("fName");
    String sClass = value.getString("classes");
    String sPhone = value.getString("phone");
    String sEmail = value.getString("email");
    feesPaid = value.getDouble("feesAmount");
    PaidAmount.setText(feesPaid.toString());

    DocumentReference documentReference1 =
fStore.collection("fees").document(sClass);
    documentReference1.addSnapshotListener(new EventListener<DocumentSnapshot>()
{
    @Override
    public void onEvent(@Nullable DocumentSnapshot value, @Nullable
FirebaseFirestoreException error) {
        totalAmount = value.getDouble("fees");
        leftAmount = totalAmount - feesPaid;
        amount.setText(leftAmount.toString());
        customAmount.setText(leftAmount.toString());
        PayNow.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                samount = customAmount.getText().toString();
                int amount = Math.round(Float.parseFloat(samount) * 100);
                Checkout checkout = new Checkout();
                checkout.setKeyID("rzp_test_Tt8zRSNLajSsNk");
                JSONObject object = new JSONObject();
                try {
                    object.put("name", fName + " (" + sClass + ")");
                    object.put("description", "New St. Stephen's Public School");
                    object.put("theme.color", "");

```



```

        object.put("currency", "INR");
        object.put("amount", amount);
        object.put("prefill.contact", sPhone);
        object.put("prefill.email", sEmail);
        checkout.open(getActivity(), object);
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

});

}

});

}

});

return v;
}

@Override
public void onPaymentSuccess(String s) {
    Intent i = new Intent(getActivity(), FeesTransactionActivity.class);
    getActivity().startActivity(i);
}

@Override
public void onPaymentError(int i, String s) {
    // on payment failed.
    Toast.makeText(getActivity(), "Payment Failed due to error : " + s,
Toast.LENGTH_SHORT).show();
}
}

```

TESTING

TESTING

Testing is more than just debugging. The purpose of testing can be quality assurance, verification and validation, or reliability estimation. Correctness testing and reliability testing are two major areas of testing. Software testing is a trade-off between budget, time and quality. Software Testing Software testing is the process of executing a program or system with the intent of finding errors. Or, it involves any activity aimed at evaluating an attribute or capacity of a program or a system and determining that it meets its required result. Software is not unlike other physical processes where inputs are received and outputs are produced. Where software differs is in the manner in which it fails. Unlike most physical systems, most of the defects in software are design errors, not manufacturing defects. To improve quality as computers and software are used in critical applications, the outcome of a bug can be severe. Bugs can cause huge losses. For Verification & Validation (V&A). It is heavily used as a tool in the V&V process. Testers can make claims based on interpretations of the testing result, which either the product works under certain situations, or it does not work.

SOFTWARE TESTING TYPE

The completion of a system will be achieved only after it has been thoroughly tested. Though this gives a feel the project is completed, there cannot be any project without going through this stage. Hence in this stage it is decided whether the project can undergo the real time environment execution without any break downs, therefore a package can be rejected even at this stage.

Testing methods

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

- 1) **Black box testing** - Black box testing treats the software as a "black box," without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.
- 2) **White box testing** - White box testing, by contrast to black box testing, is when the tester has access to the internal data structures and algorithms (and the code that implement these). White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.
- 3) **Grey Box Testing** - Grey box testing involves having access to internal data structures and algorithms for purposes of designing the test cases, but testing at the user, or black- box level. Manipulating input data and formatting output do not qualify as "grey box," because the input and output are clearly outside of the "black-box" that we are calling the system under test. This distinction is particularly important when conducting integration testing between two modules of code written by two different developers, where only the interfaces are exposed for test. Grey box testing may also include reverse engineering to determine, for instance, boundary values or error messages.

4) **Acceptance testing** - Acceptance testing can mean one of two things:

- A smoke test is used as an acceptance test prior to introducing a build to the main testing process.
- Acceptance testing performed by the customer is known as user acceptance testing(UAT).

5) **Regression Testing** - Regression testing is any type of software testing that seeks to uncover software regressions. Such regression occurs whenever software functionality that was previously working correctly stops working as intended. Typically regressions occur as an unintended consequence of program changes. Common methods of regression testing include re-running previously run tests and checking whether previously fixed faults have re-emerged.

6) **Non-Functional Software Testing - Special** methods exist to test non-functional aspects of software.

- Performance testing checks to see if the software can handle large quantities of data or users. This is generally referred to as software scalability. This activity of Non Functional Software Testing is often times referred to as Load Testing.
- Stability testing checks to see if the software can continuously function well in or above an acceptable period. This activity of Non Functional Software Testing is often times referred to as indurations test.
- Usability testing is needed to check if the user interface is easy to use and understand.
- Security testing is essential for software which processes confidential data and to prevent system intrusion by hackers.
- Internationalization and localization is needed to test these aspects of software, for which a pseudo localization method can be used.

TEST CASES

TEST CASES

NO	SCREEN	INPUT GIVEN	EXPECT ED OUTPUT	ACTUAL OUTPUT	TEST PASS
1	Admin Login Screen	Email: vishalchaturvedi0862@gmail.com Password – admin@ @	Admin Dashboard Screen Appears.	Admin Dashboard Screen Appear	YES
2	Student Login Screen	Email: vishalchaturvedi0862@gmail.com Password: Student@ @	Student Dashboard Screen Appears	Student Dashboard Screen Appear	Yes
3	Admin Login Screen	Email: vishalchaturvedi0862@gmail.com Password – admin	Admin Dashboard Screen Appears.	Invalid Password	YES
4	Register Screen	9760041381434	Student registrat ion success ful	Please enter a valid phone number	No
5	Register Screen	Vishal.2124mca1037@gbkljrteu	Student register successful	Please enter a valid email id	No

SCREEN LAYOUTS

SCREEN LAYOUTS

1. Login Screen

9:49 AM | 1.5MB/s

New St. Stephen Public School



Email

Password

LOGIN

REGISTER

2. Registration Screen

9:51 AM | 19.1KB/s

Registration



Full Name

Class

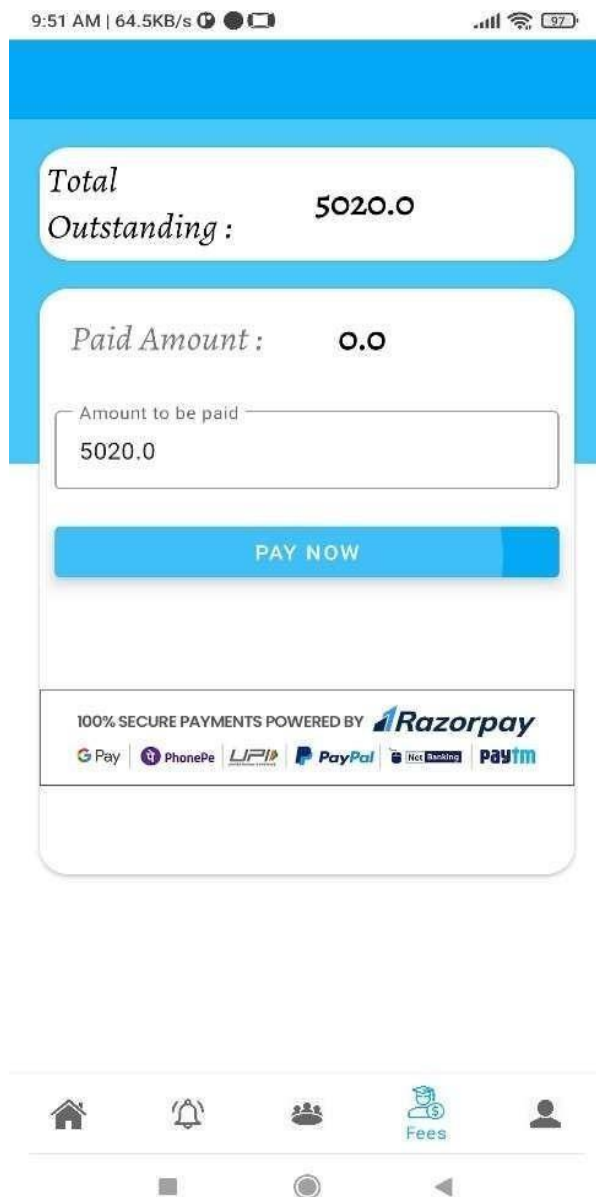
Role

Phone

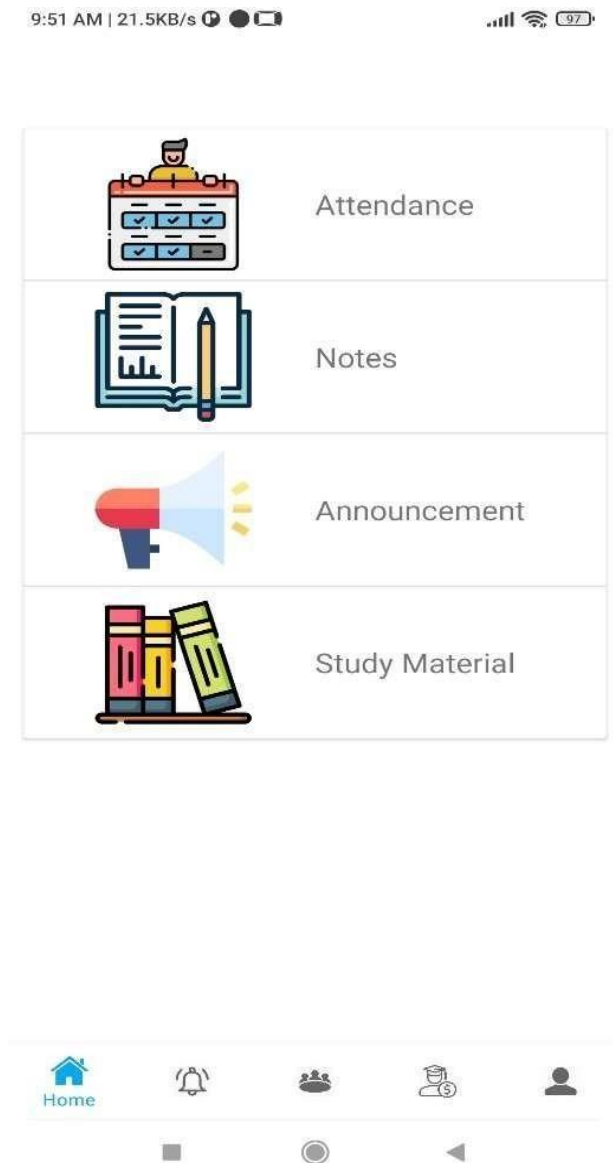
Email

Password

REGISTER



3. Fees Payment Screen



4. Home Screen

FUTURE SCOPE

- In the future, students can also be able to upload or download notes.
- In future our system can include online accounting system, good backup and restore facility.
- System is so much flexible so in future it can increase easily and new modules can be added easily.
- You can add online student admission.
- The School Management System (SMS) can be enhanced to include some other functionality like marks.
- Talent management of students based on their performance evaluation can be added.
- Social networking can also be added wherein students can interact with each other.
- Online examination module would be introduced to conduct online examination.
- Further, the faculty can upload the videos of their lectures on this site and students who had missed those classes can view those videos

CONCLUSION

- The application provides appropriate information to users according to the chosen service.
- The project is designed keeping in view the day-to-day problems faced by a school.
- Deployment of our application will certainly help the school to reduce unnecessary wastage of time in personally going to each department for some information.
- Awareness and right information about any school is essential for both the development of student as well as faculty.
- Helpful to perform paperless work and manage all data
- Provides easy, accurate, unambiguous and faster data access.
- This software is very easy to use so all educational institute can use this frequently.
- It was a wonderful and learning experience for us while working on this project.

BIBLIOGRAPHY

Books:

- The Complete Reference Java 2 (Herbert Schildt).
- Beginning Android Programming with Android Studio (Wrox; 4th edition)

Websites:

- www.geeksforgeeks.org
- www.javatpoint.com
- developer.android.com/
- www.stackoverflow.com