Kingdom of Saudi Arabia
Majmaah University
Ministry of Higher Education
College of Science in Zolfi
Dept. of Computer Science

Quick Center

Dept. of Computer Science

By:
Sara Faleh AL-Habardi

Graduation project

Submitted in partial fulfilment of the requirements

for the award of

Bachelor degree

of the

Majmaa University

Supervisor: T.Chafika Laabidi Ouni

Faculty of science
Dept. of Computer Science

2018

Abstract

Quick Center is a mobile application that allows students to place order for printing request and following teacher and notify if any new files added. It allows teacher adding new files and student can easy order it to print and be available in suitable time.

Through this app the center manager can add new item and remove them as well as can manage orders of printing in easy way.

Students will not need to spend too much time waiting for documents to print. Simply, all they have to do is order for what they want from the quick center app instead of going to the center and waiting for a lot of time and causing congestion inside the center to do that.

The application that will be developed will provide and facilitate the procedures for printing through simple and user-friendly interfaces.

Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I would like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Also my supervisor T.chafika Laabidi Ouni for her help.

MAJMAAH UNIVERSITY,
COLLEGE OF SCIENCE AL ZULFI,
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION

(CERTIFICATE BY STUDENT)

This is to certify that the project titled **Quick Center** submitted by **Sara Faleh AL-Habardi 351202114** under the supervision of **T.Chafika Laabidi Ouni** for award of Bachelor degree of the Majmaah University carried out during the Semester 1, 2018-19 embodies my original work.

Signature in full: ----------

Name : Sara Faleh AL-Habardi

Student ID: 351202114

Date:

Contents

Table of Figures

Table of Tables

1. Introduction

1.1 Overview

Technology became a main part of our daily bases practices, that you hardly find anyone doesn't use one or more types of technology such as using mobile and iPads in all of their times and every day. Quick Center is developing an android application to replacing method of dealing between students and center from tradition method to modern technology, so the student can order print requests to center using modern technology. Students will not need to spend so much time waiting for printing documents. Simply, all they must do is ordering whatever they want from app instead of going to center and waiting a lot of time and causing congestion inside the center to print the documents.

1.1.1 Abstract system description

In Quick Center app there is center admin for application to manage all request in application, and to check student orders and execute the orders, then the student will notify if the order is ready.

The students must register into the system and then can ordering requests. The student can also follow the teacher that register in system and follow the documents that added by teacher and make order to print it if needed to that and can also track the status of orders in addition system allows users to check for various items in available in the center.

1.2 Problem definition

In University Center that depend in traditional methods and with growth in the number of students, and increasing the daily operations of print request, it will be leading to difficulty for employee of center to manage all orders and operations, the center facing problem, such as queue and overcrowding issues، which consider very time consuming and inefficient method of working, it makes the management of student printing requests quite difficult and leading to delaying. In addition, there are many worksheets that are placed in the center for students to use for their upcoming lectures, if teachers forgot to tell the students in the previous lectures about need some worksheets in coming lecture will leading to delay in bringing the worksheets in lecture.

The student does not want to wait too long to get needs and print documents in the center she wants take ordering on time without delay. In order to address this problem, w Quick Center app come up to avoid all problems faced center and students also the teacher prepares files for printing in the center and adding this through the system. System notify the students who followed them, so they can print it before the next lecture in addition students can easily order printing requests through the app.

Quick Center app will be to get rid of these problems and improve work by minimization time consuming by make all operations in center is automated.

1.2.1 Goals

The main goal of the app is to ease communication between students and center and enables students to request order from any place in any time. also solve the problem of waiting and queues of congestion and making the work in the center easier and more effective by managing the order of printing requests and not rely on the early morning and breaks to receive printing requests.

1.2.2 Data collection

Although there are many methods that can be used to collect information for use in information analysis, one of the most commonly used methods is the questionnaire. The questionnaire method is one of the most common methods used to extract information by asking questions to users to identify the needs to meet them.

The questionnaire was used to survey people's opinions. The questionnaire contains many questions. The results were collected and analyzed.

A random sample of (36) people were selected. The questionnaire was distributed through GOOGL Drive within the Kingdom of Saudi Arabia.

The answers to the questions were as follows:

1-The print center branch at the college service is quite perfect؟

63.9% of them selected no

2-Do you think the center needs more development؟

100% of them selected yes

3-Did the waiting line at the center is a bit long؟

97.1% of them selected yes

4-Did you arrive late in lecture because you wanted for printing؟

94.4% of them selected yes

5-Did you west your break time in waiting for printing ؟

97.2% of them selected yes

6-What if you sent your file or document before in an application, then you find you need ready to pay in time you want without waiting in line؟

97.2% of them selected yes

1.2.3 Literature review

In this section present a literature review the review information about similar systems in the field of services for student. Finally, a comparison is made between the systems studied and the current system.

1-Registration program in in the Islamic university in Palestine:

A program designed for the student of the Islamic university to review all students needs such as access to lectures, view results, print documents, and submit fees.

Features:

- ✓ Works with windows, links, and Macintosh systems.
- ✓ Supports Arabic language.

- ✓ Accessible by private computers, local net, and internet.
- ✓ Safe.
- ✓ Education.
- ✓ Print and share documents.

Shortcomings:

- ✓ Printing needs a permission form the registrar.
- ✓ Home page is Arabic only.

2-Registration program in Taibah university in Saudi Arabia:

A program that assures the huge effort from the university community and student services.

Features**:**

- ✓ Easy to use and does not require high level of computer skills.
- ✓ Supports Arabic language.
- ✓ Accessible by private computers, local net, and internet.
- ✓ Electronic education.
- ✓ Print and share documents.

Shortcomings:

- ✓ Supports Arab students only because it is in Arabic.
- ✓ Homepage is Arabic only.

3-The program of al-hashmyah university in Gordon:

The program allows students request all of the services from home and have it ready in all forms. Also, decreases the pressure upon the registrars and the students.

Features:

- ✓ Supports Arabic language.
- ✓ Accessible by private computers, local net, and internet.
- ✓ Safe.
- ✓ Electronic education.
- ✓ High qualities.
- ✓ Print and share documents.

Shortcomings:

- ✓ Printing needs permitting from the registrar.
- ✓ Difficult to use for new students.

Features and shortcomings of the previous programs in compared to the suggested one:

Table 1-1 comparison

program comparison	Our program	The program of al hashmyah university in Gordon	Registration program in Taibah university in Saudi Arabia	Registration program in in the Islamic university in Palestine
multilingual	✓	×	×	×
Works with windows, links, and Macintosh systems.	✓	×	✓	✓
Accessible by private computers, local net, and internet	✓	×	×	✓
Easy to use	✓	✓	×	✓
accessibility	✓	×	✓	✓
Print and share documents	✓	✓	✓	✓
Printing permission	✓	✓	✓	✓

I reviewed the features and the shortcomings of the previous programs and compared them with my suggested program as demonstrated in the given table, which clarifies the most important advantages included in my project.

1.2.4 Objectives

Quick Center App will be able to achieve the following objectives: -

- Organization of printing request for fast and flexible way using mobile app.
- Students can upload their files and send it to the center with adding how many copies and time which saving time and effort of students

- Student can follow teachers and will have a notification if their teacher they follow upload files
- Notify Students to receive the orderings when is be ready.
- Reduce overcrowding at the center.

1.2.5 Critical success factors

- The application will save time and effort to the center manager. By reducing congestion and managing print requests electronically, allowing more time for staff at the center to focus on printing orders without wasting time in the congestion.
- The system will save time on students by canceling the waiting and going directly to the center within their break time and enable the possibility of ordering at any time easily through the application, in addition to facilitate the follow-up teacher and the files that uploads.
- The application will provide teachers with the easy way to upload files that students need to print.

1.2.6 Organization chart and responsibilities

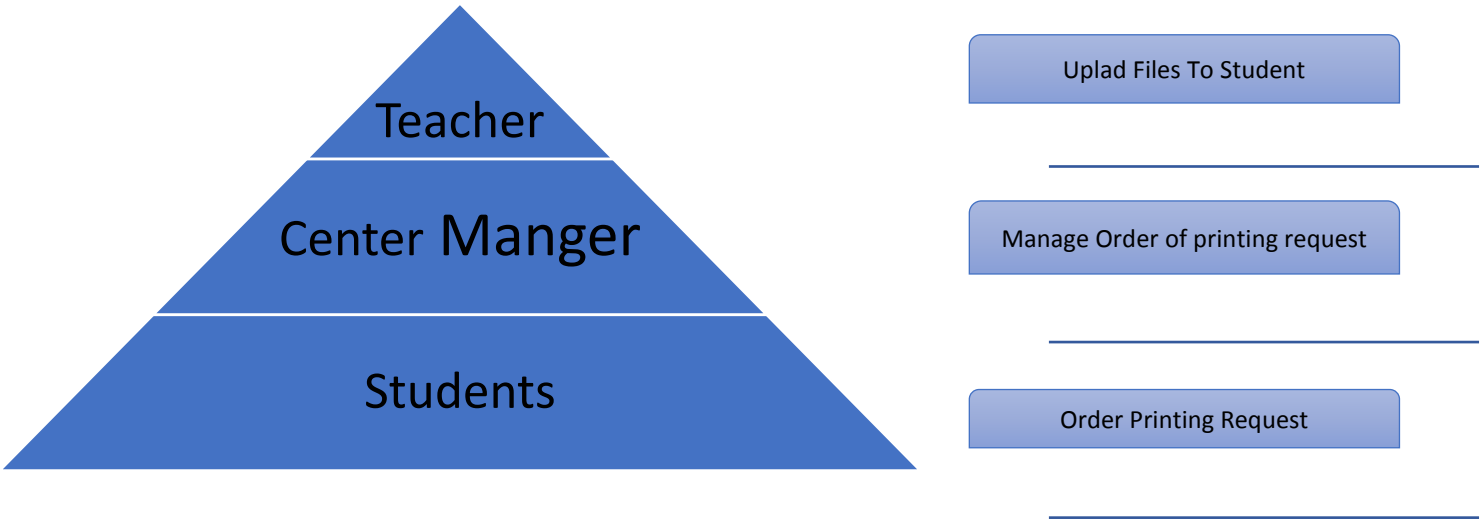


Figure 1-1 Organization chart

1.3 General rules (assumptions)

Requires both students and teachers to know how to use mobile applications in their primary form. Also the application need to:-

- **Availability:** The application should always be available for access at 24 hours and 7 days a week.
- **Reliability:** The application should provide the services immediately in response to the user needs.
- **Security:** The Information should be secure; there should not be any kind of malfunctioning, so the application will support providing username and password to prevent the application from unauthorized access.
- **Usability**: The user can use the services offered by the application through an easy to use and simple interface. The application must be easy to use by all users.

2. System analysis

2.1.Introduction

System analysis describes what a system should do to meet the information needs of users. Analysis is the most important phase in systems development life cycle. The purpose of doing analysis is to transform the system's major inputs into structured specification.

2.2.Description of Data Flow Diagram (DFD)

A data flow diagram shows the logical flow of the system. It is often used to clarifying system requirements and identifying major transformations. It shows the flow of data through a system. DFDs can also use for the visualization of data processing, shows what kind of information will be input to, and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel **[1].**

2.2.1. Context Diagram

A context diagram is a data flow diagram that only shows the top level, At this level, there is only one visible process node that represents the functions of a complete system in regard to how it interacts with external entities **[2].**

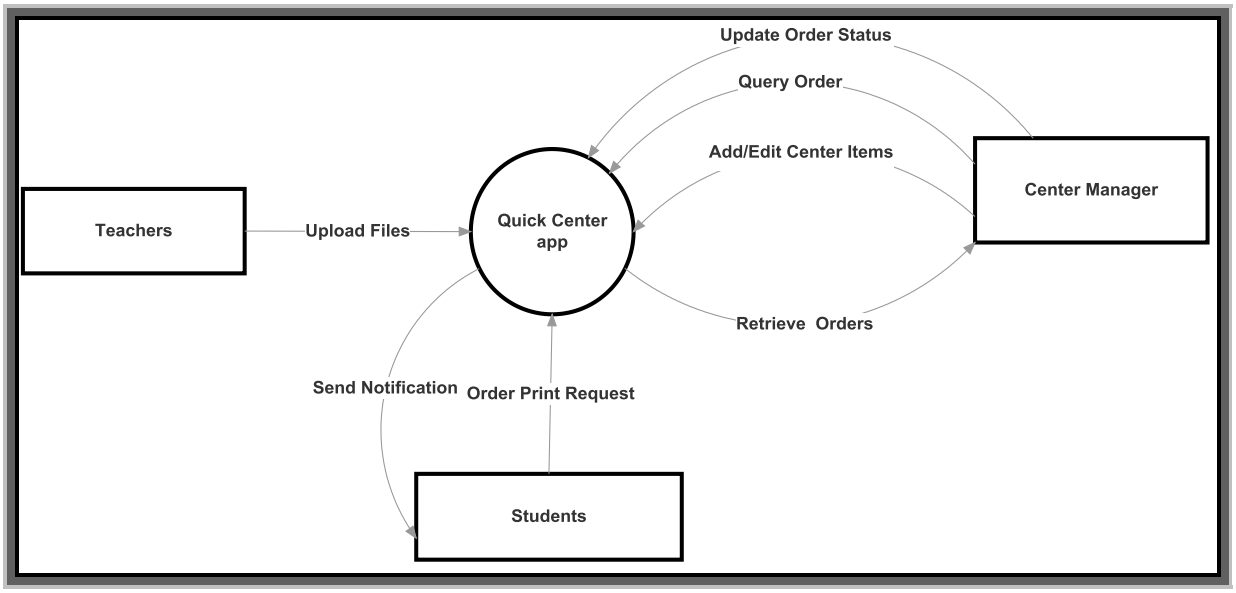


Figure 2-1 Context Diagram

2.2.2. Overview diagram (level 0)

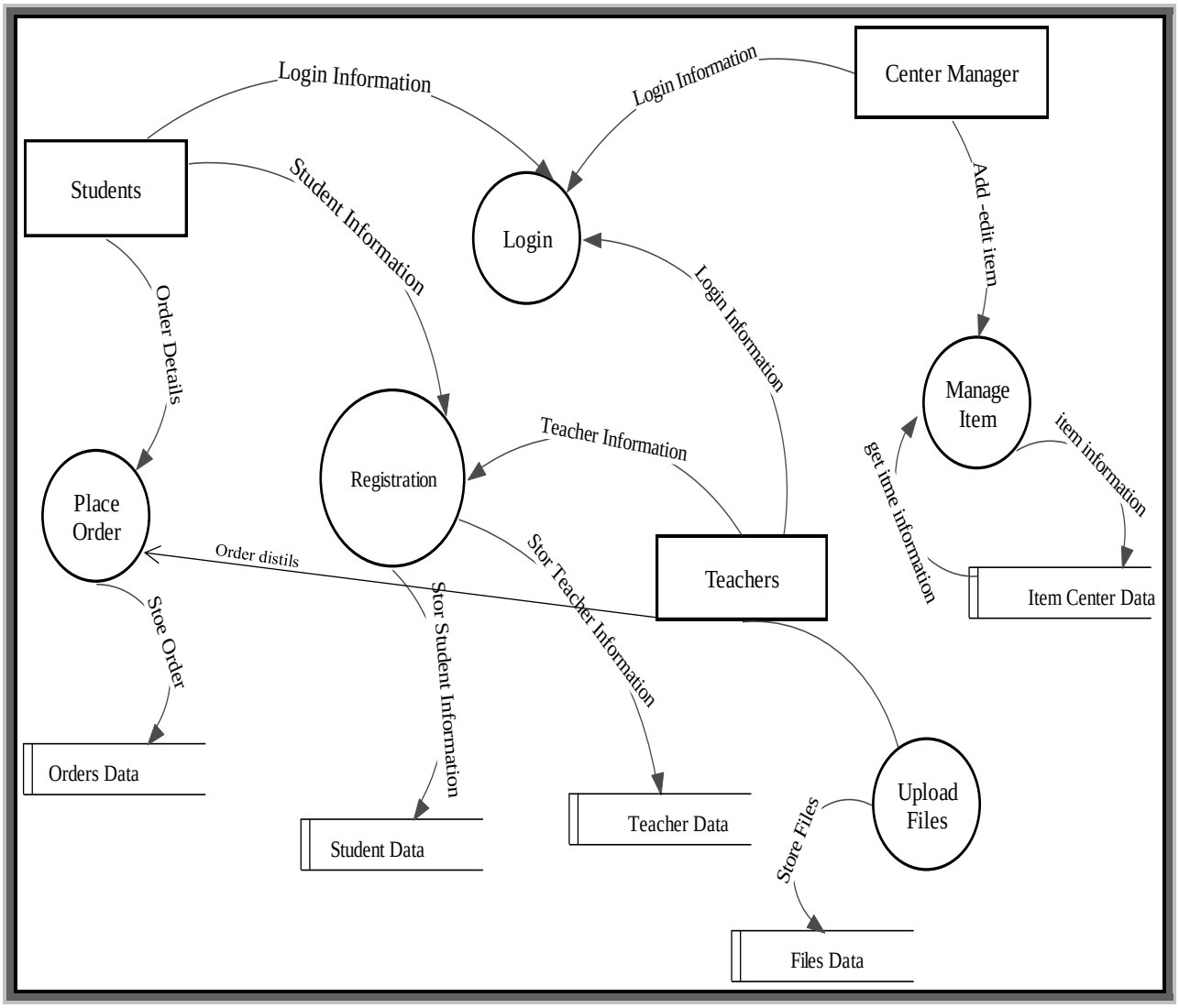


Figure 2-2 Overview diagram (level 0)

2.2.3. Detailed DFDs

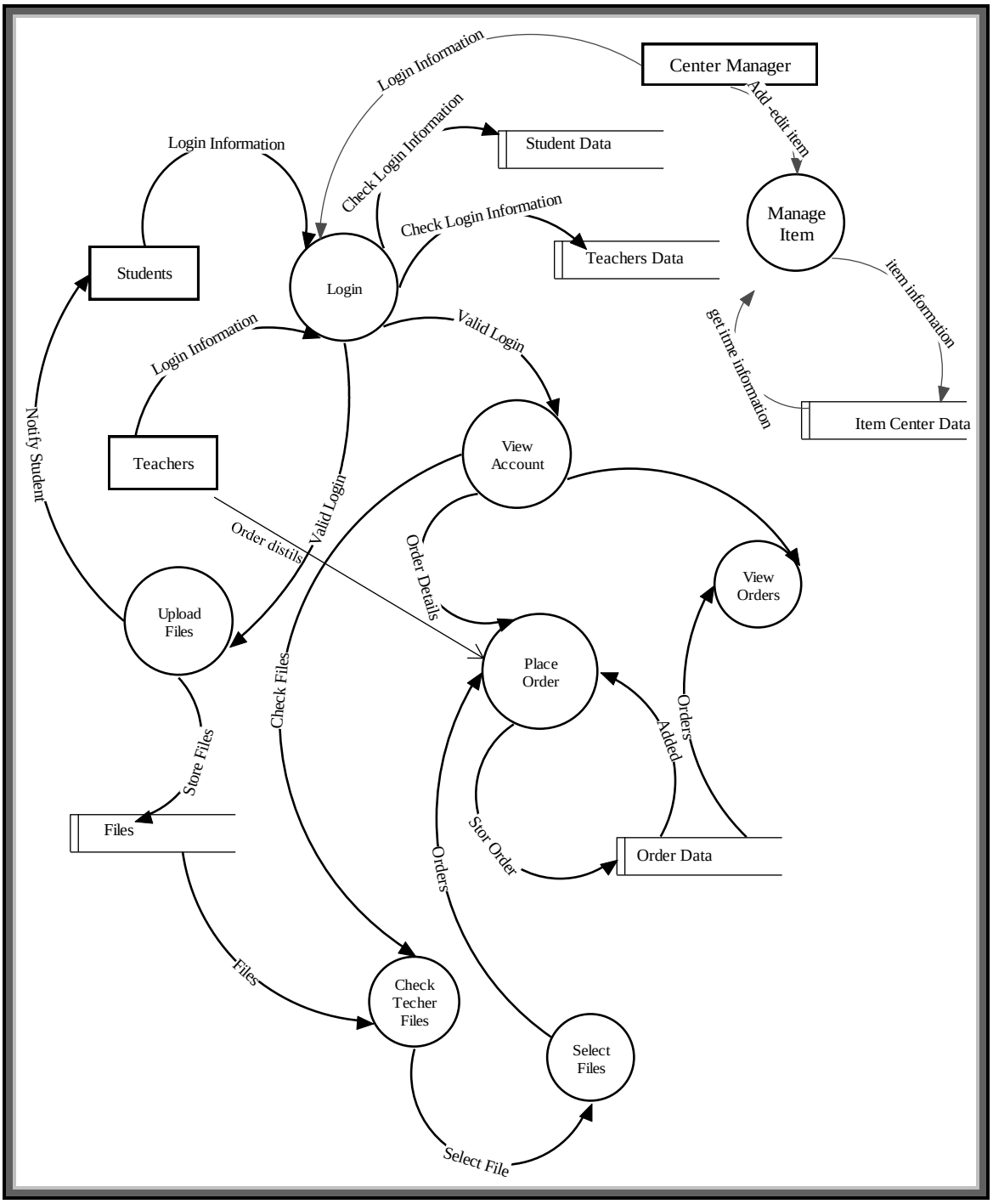


Figure 2-3 Detailed DFDs

2.3.Entity Relationship Diagram (ERD)

The entity relationship (ER) model was developed to facilitate database design and represents the overall logical structure of a database. The ER data model is one of the several semantics data models; the semantic aspect of the model lies in its representation of the meaning of the data. The ER data model has three basic notions: entity-sets, relationship sets and attributes.

2.3.1. Description of Entities

1- **Name:** Students

Attributes: Name: The name of the Student

Email: The Email of the Student

User Name: The User Name Which is determined by the student to login

Password: Student password to login to application

ID: Unique ID for student

2- **Name:** Teachers

Attributes: Name: The name of the Teacher

Email: The Email of the Teacher

User Name: The User Name Which is determined by the Teacher to login

Password: Teacher password to login to application

ID: Unique ID for Teachers

3- **Name:** Center Manger

Attributes: Name: The name of the Manger

Email: The Email of the Manger

User Name: The User Name Which is determined by the Manger to login

Password: Manger password to login to application

ID: Unique ID for Center Manger

4- **Name:** Orders

Attributes: Order Id: Unique ID

File: The File to printed

Number of copies

Time desired

Note Status

5- **Name:** Files

Attributes: File ID: Unique ID

Description: File Description

File: File that uploading

6- **Name:** Center Item

Attributes: Item ID: Unique ID

Name: Item Name

Description: Item Description

Price: item price

2.3.2. Description of relations

- The relationship between Students and ordering request is one to many so the student can order many print request and the order is for one student.
- The relationship between Students and Teachers is many to many so the student can follow many teacher and the teacher is followed by many students.
- Center Manager can add many item
- The Teachers can add one or many files, file is for one Teacher.

2.3.3. Drawing ERD.

ER diagram an entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes **[3].**

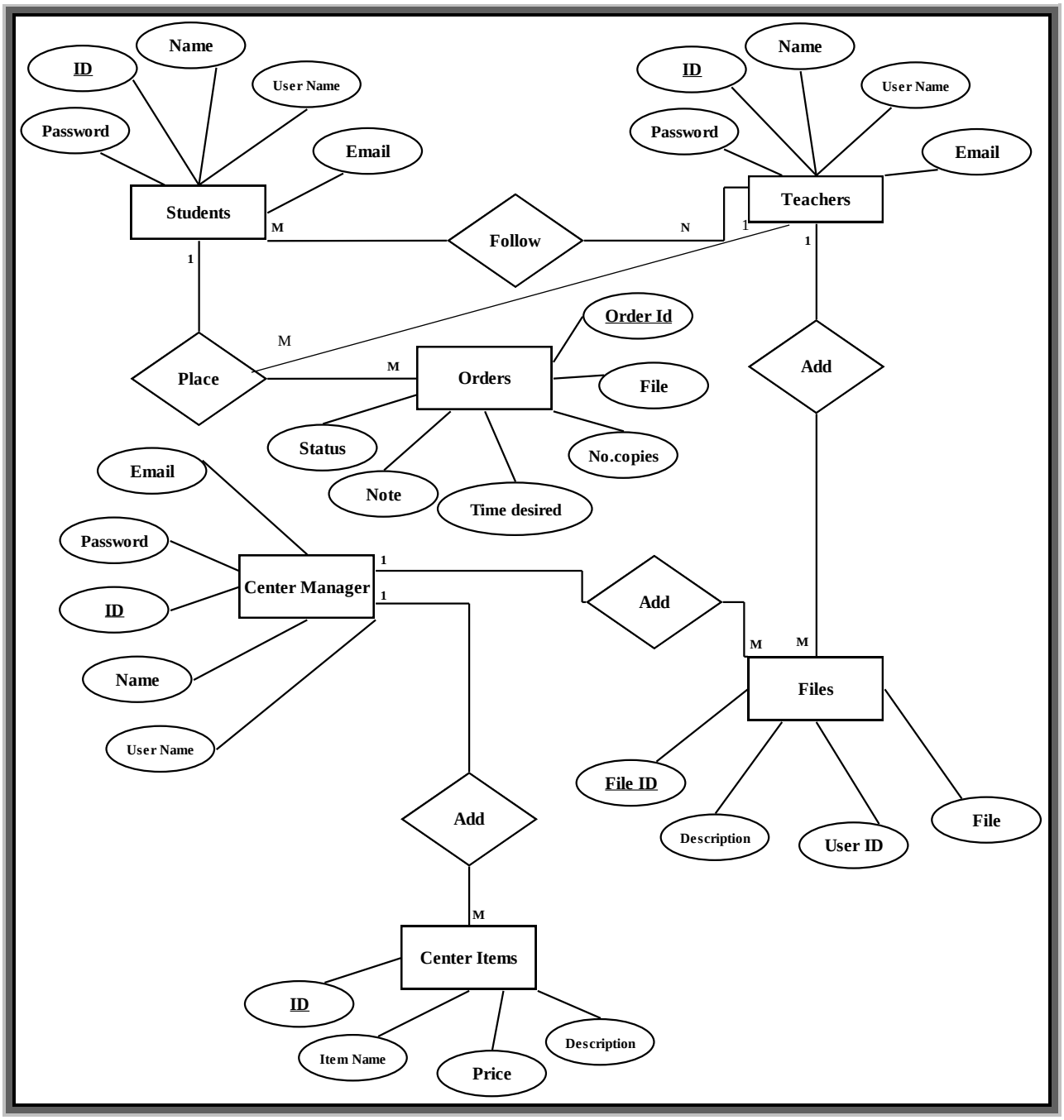


Figure 2-4 ERD diagram

2.4.Structure Diagram

2.4.1. Class Diagram

The Class Diagram is used extensively to describe the types of objects in the system and their relationships with each other[8].

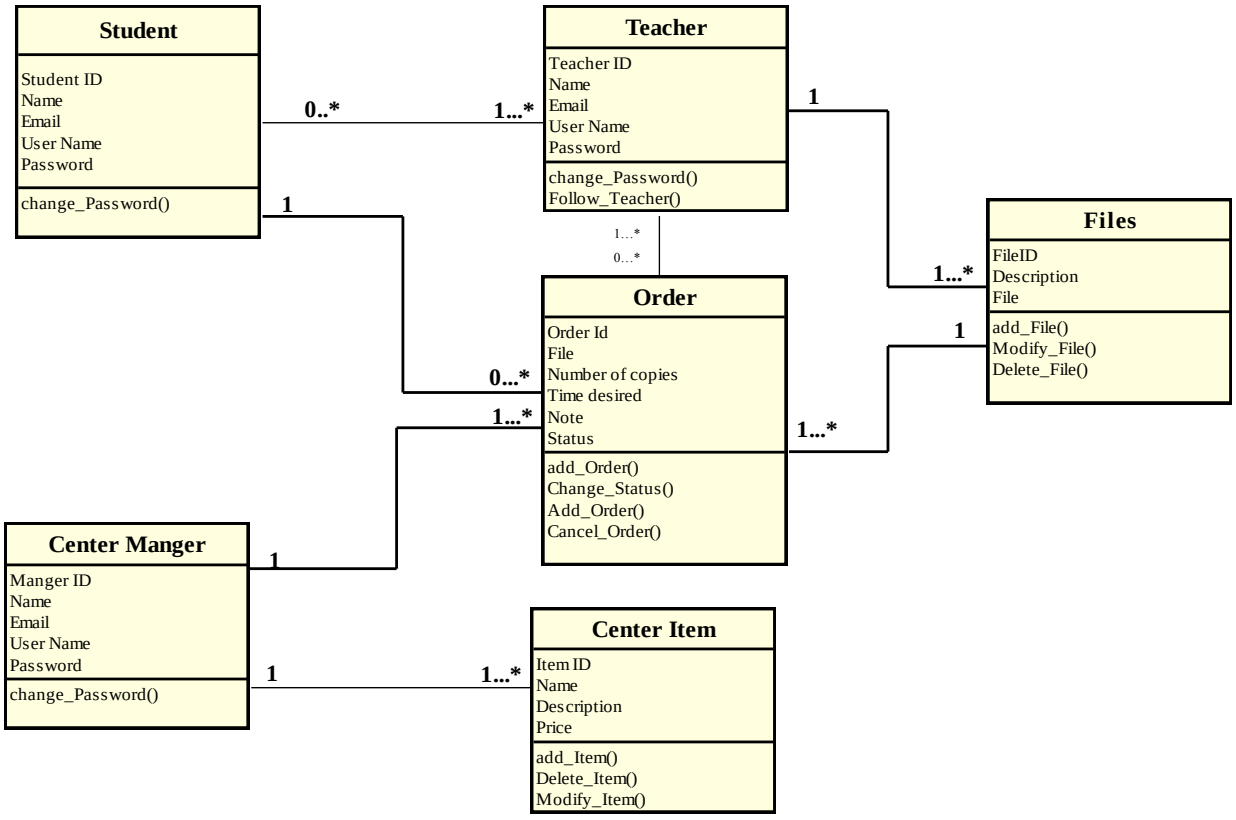


Figure 2-5 Class diagram

2.5.Behavior diagram

2.5.1. Use case diagram

One way to describe the system from the analysis point of view is the use case diagram, which is a collection of use cases initiated by specified actors. It is used to describe what a system does without describing how the system does. the actors (is a person, organization, or external system that plays a role in one or more interactions with system.

To identify use-cases performed in the application, I identify the users of the application; this is done by means of actors. The main actors are the Students, Teachers and center manager[9].

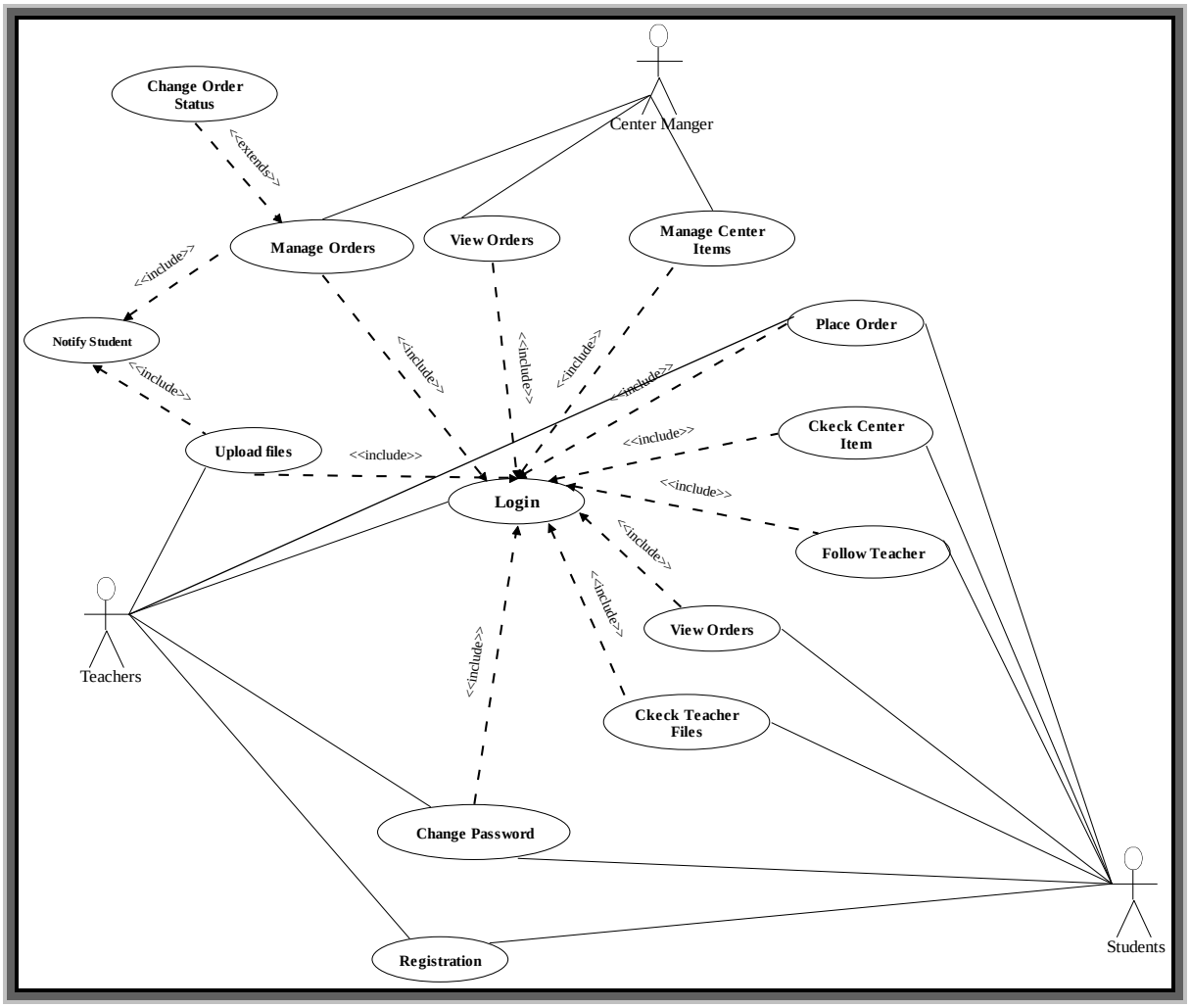


Figure 2-6 General Use Case Diagram

1- Teacher Use case diagram

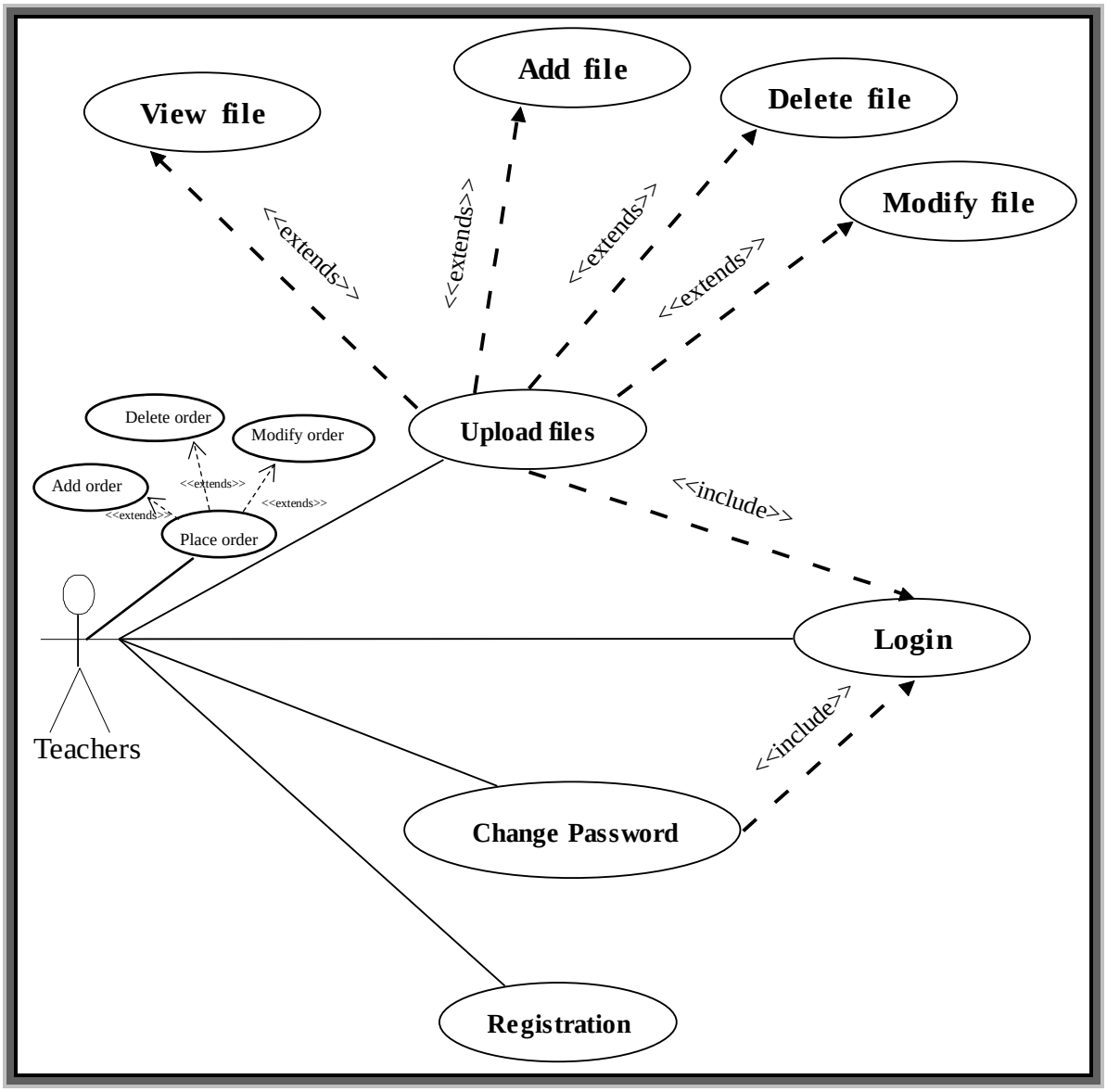


Figure 2-7 Teacher Use Case Diagram

2- Student Use case diagram

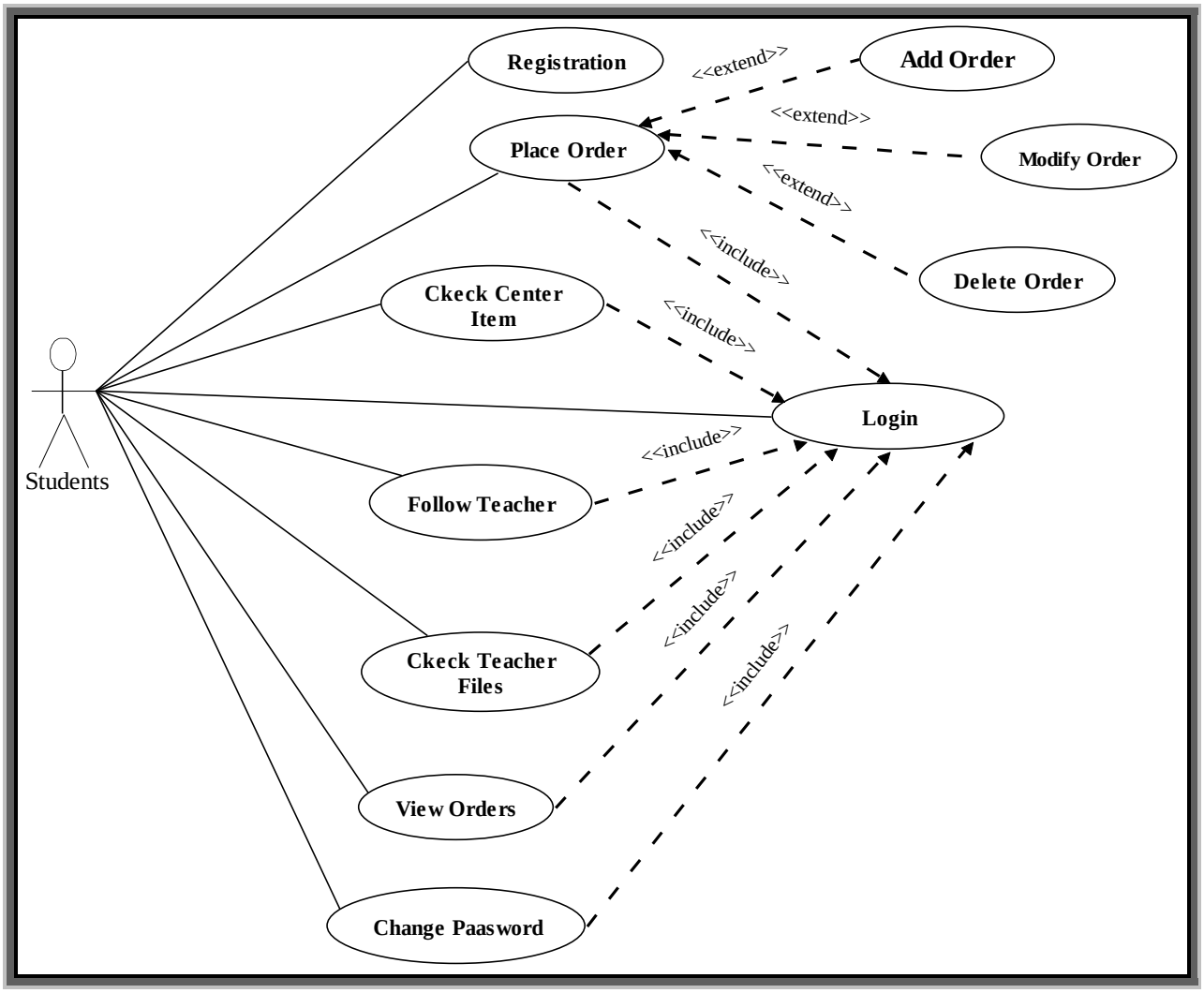


Figure 2-8 Student Use Case Diagram

3- Center Manager Use case diagram

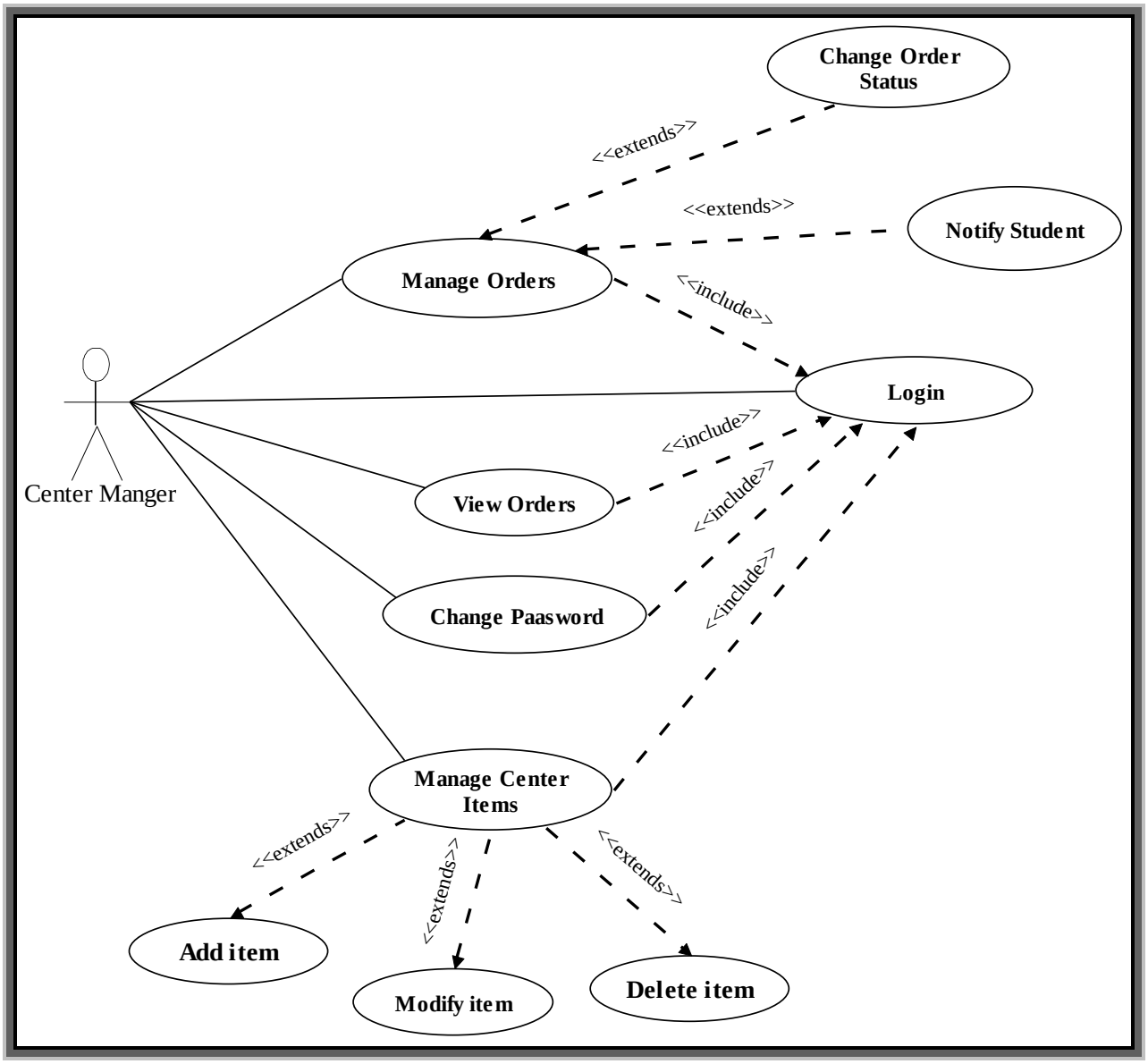


Figure 2-9 Center Manger Use Case Diagram

2.5.2. Activity Diagram

An activity diagram visually presents a series of actions or flow of control in a system like a flowchart or a data flow diagram[6].

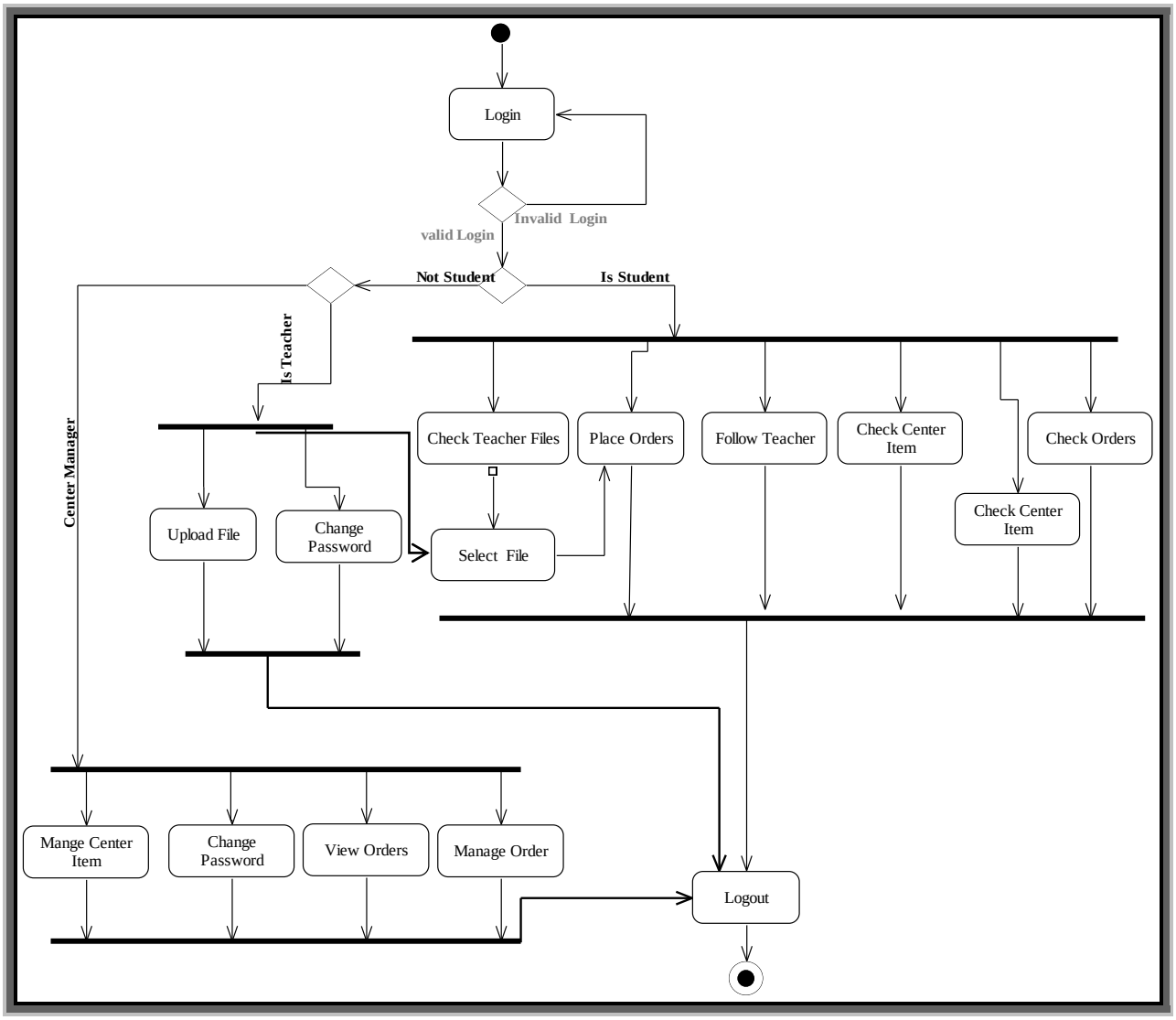


Figure 2-10 Application Activity Diagram

2.5.3. State Diagram

A state diagram is the graphical representation of a state machine and one of UML diagram types for software and systems. State diagrams show a behavioral model consisting of states, state transitions and actions[10].

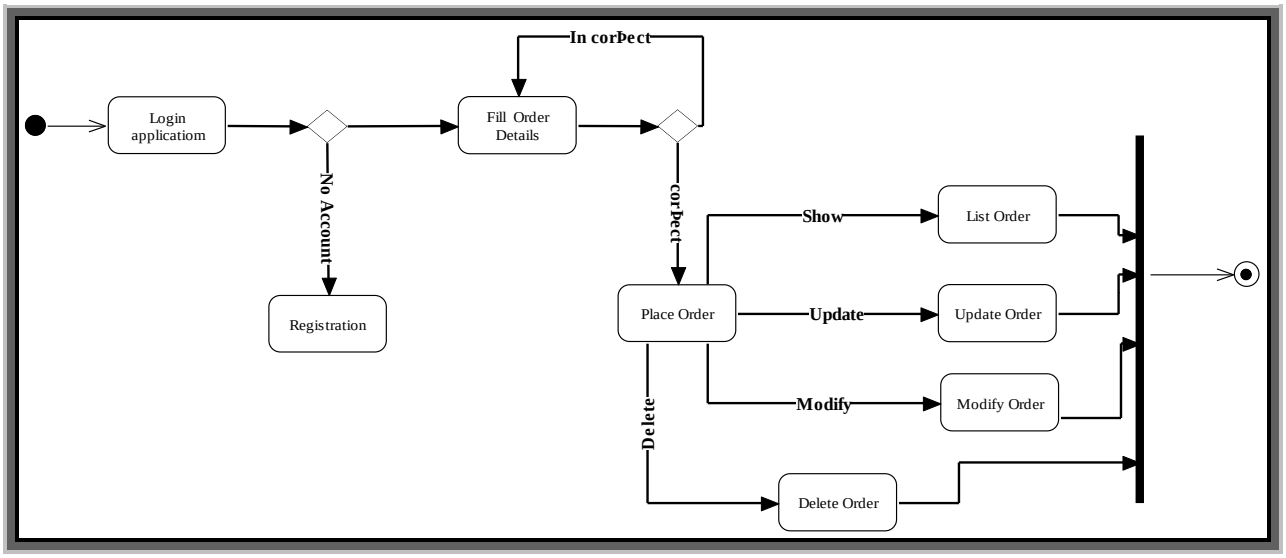


Figure 2-11 State Diagram

2.6.Interaction Diagram

2.6.1. Sequence Diagram

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order, it describes the sequence of action that need to be performed to complete task .The place order sequences diagram where the login, and logs into the app using his username and password and submits the information. The app attempts to validate the username and password against the database if his/her username & password is correct, he/she is logged into the system and taken to account screen or displayed invalid username or password warning message then the student can fill order details to place order[11].

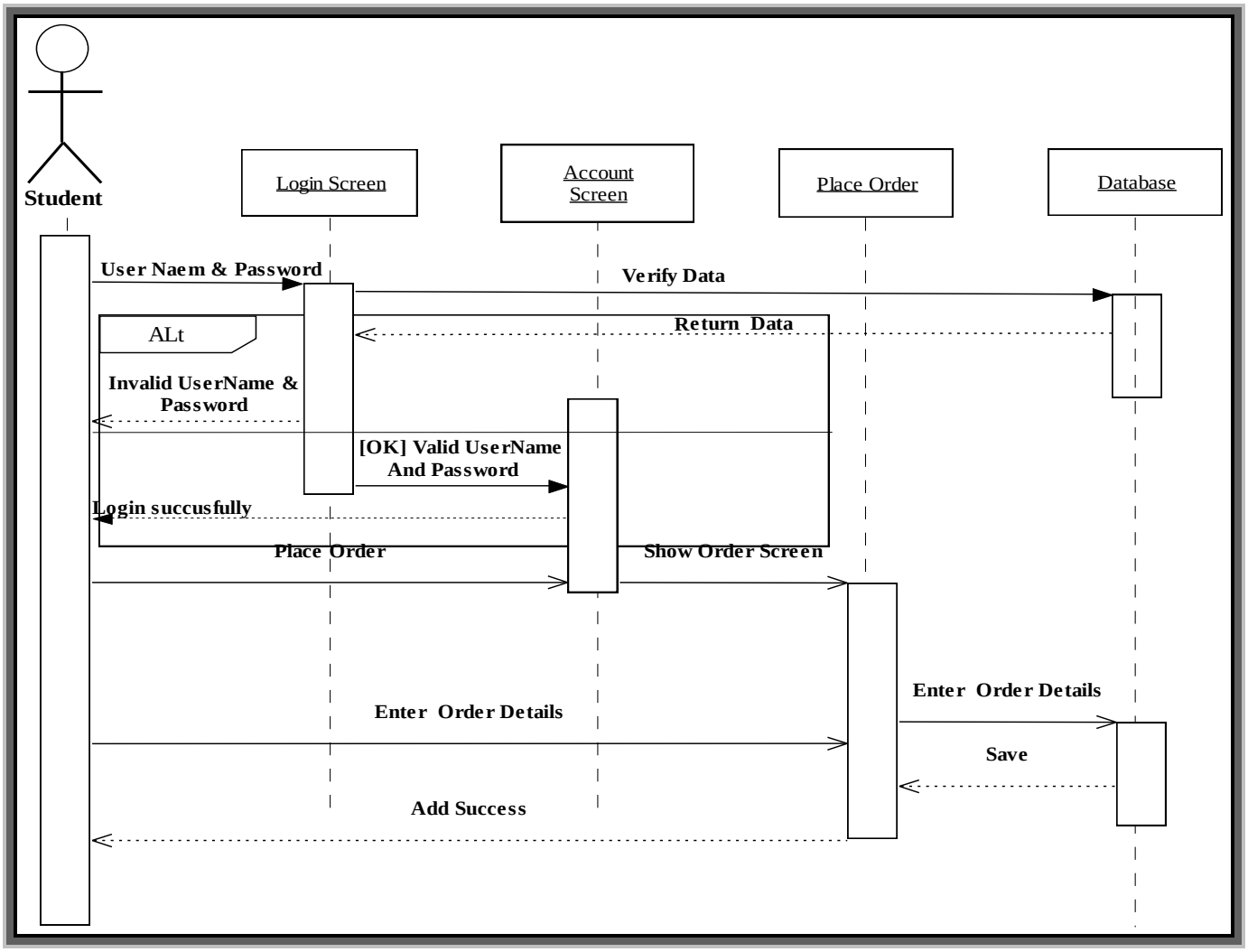


Figure 2-12 sequence diagram

3. System design

3.1.Description of procedures and function

1- Registration

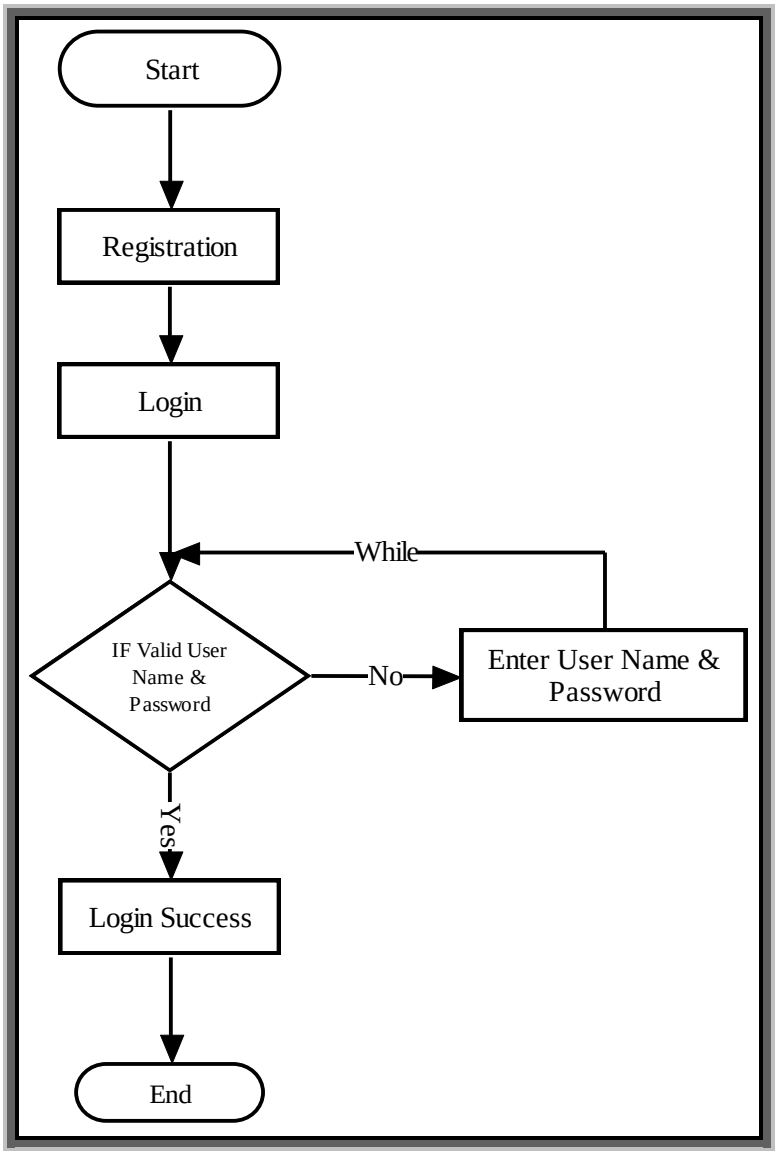


Figure 3-1 Registration

2-Place order

- Student Login to the application by valid username and password.
- The student can place order by uploading new file.

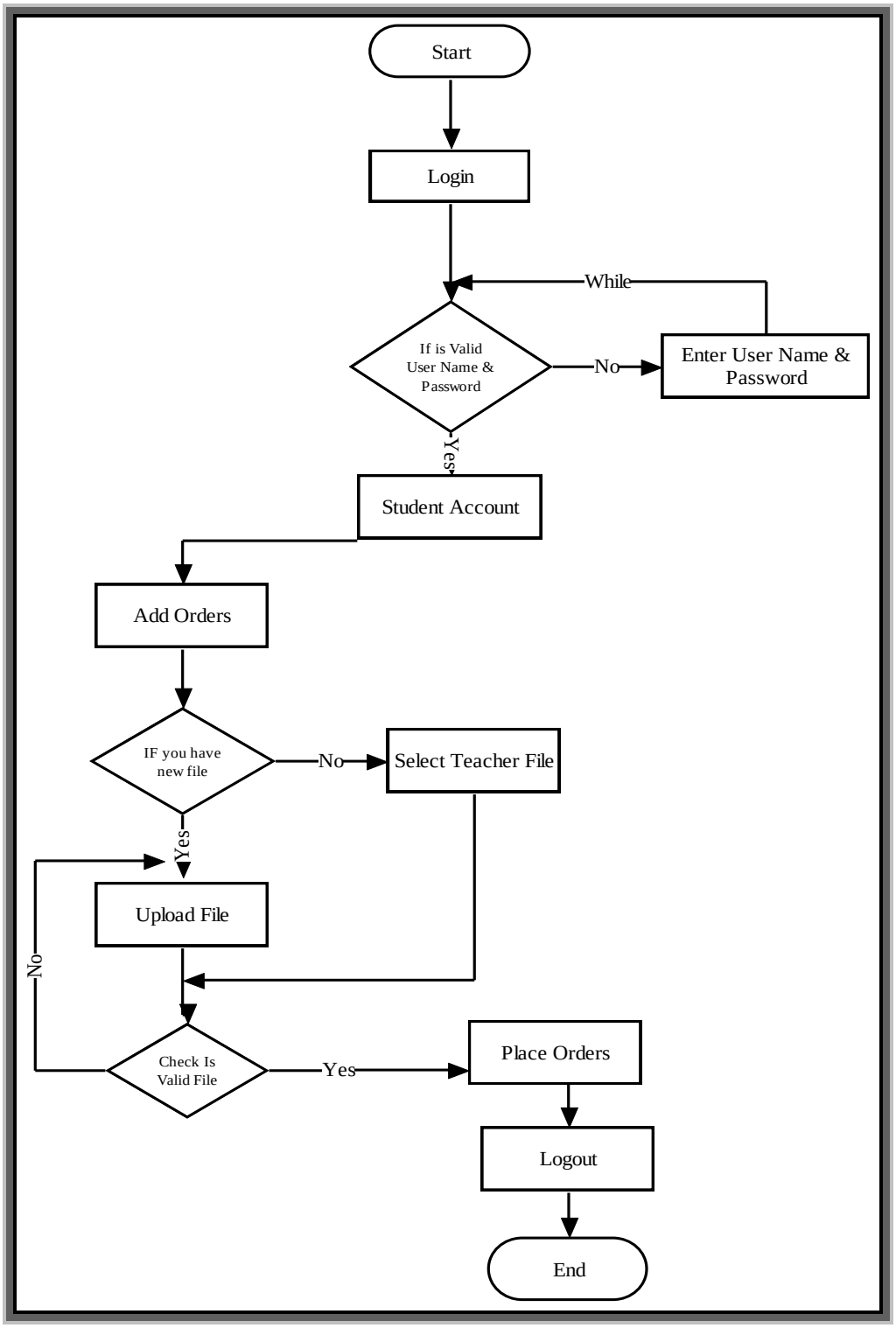


Figure 3-2 Place Order

3- Manage Orders center manager:

Login to the application by username and password.

- Show Orders of printing request
- Change Orders status
- Notify Student

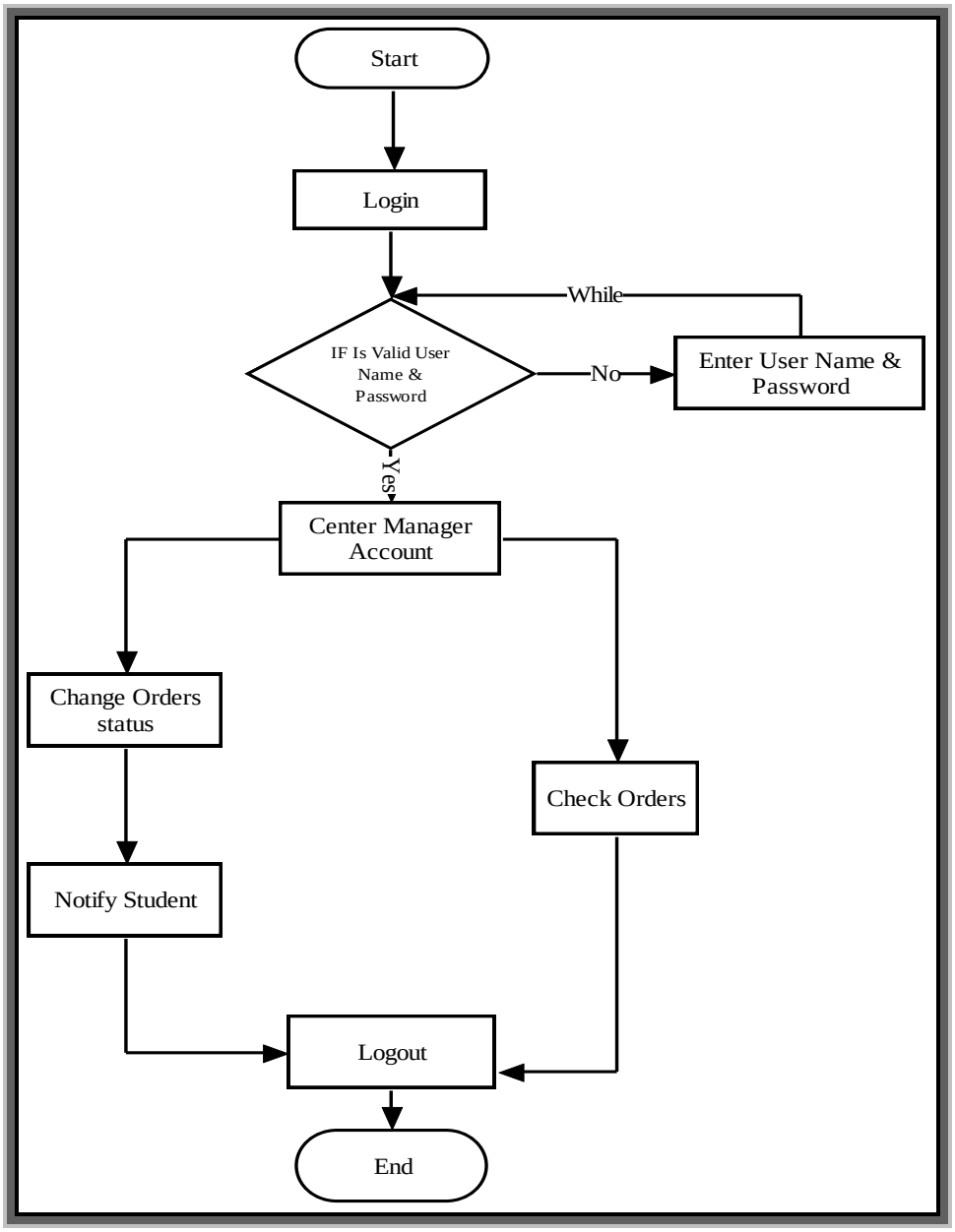


Figure 3-3 Manage Orders center manager

4-Application flow chart for Teachers:

- Registration in application
- Login to the application by username and password.
- Upload files

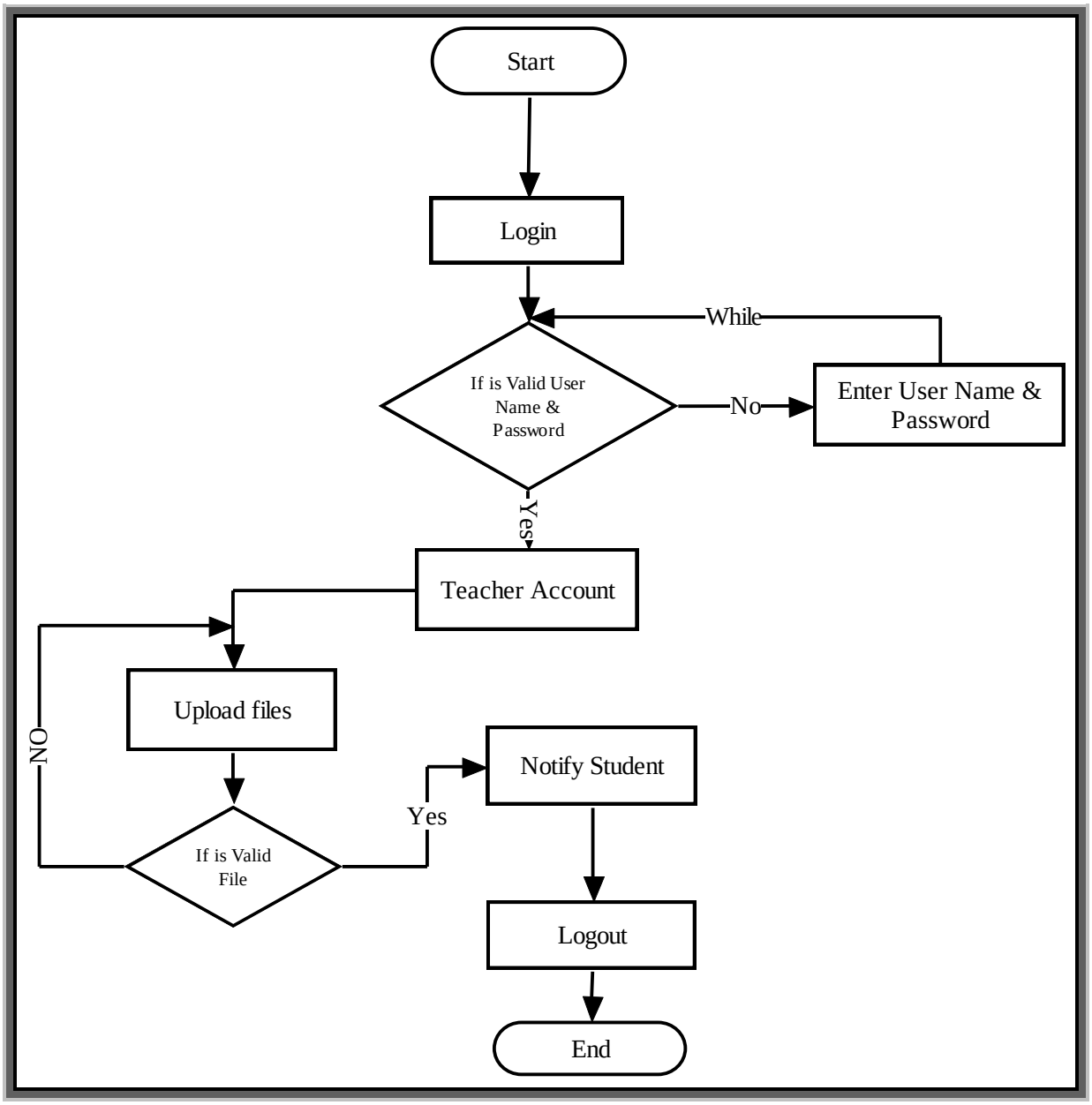


Figure 3-4 Application flow chart for Teachers:

3.2.Relation database schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful [4].

3.2.1. Tables

Several tables would be required to implement our system the following is the list tables:

- Student Table: The student table contains student information and login information for student
- Teacher Table: contains teacher information and login information for Teacher
- Center Manger Table: contains center managers information and login information for
- Center Item Table: Contains information about the items in the center and their details
- Files Table: Contains information for documents and files to be printed
- Order Table: Contains details of the print requests that were requested by students
- Following Teacher Table : Contains a list of student follow-up teachers

3.2.2. Attributes

Student Table

Table 3-1 Student Table

Field Name	Data Type	Key
ID	Number	Primary Key
Name	Text	
Email	Text	
User Name	Text	
Password	Text	

Teacher Table

Table 3-2 Teacher Table

Field Name	Data Type	Key
Teacher ID	Number	Primary Key
Name	Text	
Email	Text	
User Name	Text	
Password	Text	

Center Manger Table

Table 3-3 Center Manger Table

Field Name	Data Type	Key
Manger ID	Number	Primary Key
Name	Text	
Email	Text	
User Name	Text	
Password	Text	

Center Item Table

Table 3-4 Center Item Table

Field Name	Data Type	Key
Item ID	Number	Primary Key
Name	Text	
Description	Text	
Price	Number	
Manager id	Number	Foreign key

Files Table

Table 3-5 Files Table

Field Name	Data Type	Key
File ID	Number	Primary Key
Description	Text	
File	Text	
MangerID	Number	Foreign Key
Teacher_ID	Number	Foreign Key

Order Table

Table 3-6 Order Table

Field Name	Data Type	Key
Order Id	Number	Primary Key
File	Number	Foreign Key
Number of copies	Text	
Time desired	Date -Time	
Note	Text	
Status	Text	
Student ID	Number	Foreign Key

Following Teacher Table

Table 3-7 Following Teacher Table

Field Name	Data Type	Key
Teacher ID, Student ID	Number	Primary Key
Student ID	Number	Foreign key
Teacher ID	number	Foreign key

3.3.Hardware and software requirements

3.3.1. Software Requirements:

Various Software that is required to do our application is discussed here.

- **MySQL:** In this project, MySQL is used as the backend database. MySQL is an open source database management system. The features of MySQL are given below:

- MySQL is a relational database management system. A relational database stores information in different tables. These tables can be referenced to each other, to access and maintain data easily.
- MySQL is open source database system. The database software can be used and modify by anyone according to their needs.
- It is fast, reliable and easy to use. To improve the performance, MySQL is multithreaded database engine. A multithreaded application performs many tasks at the same time as if multiple instances of that application were running simultaneously [7].

- **JDBC Drivers** MySQL database is connected to android using MySQL JDBC Connector, Java Database Connectivity (JDBC) is a widely accepted application-programming interface (API) , it contains specification for connecting programs written in Java to the data in popular databases. for database access [5].
- **Android Studio:** We well use it to implement our application, we write code and using built in function. Android Studio provides the fastest tools for building apps on every type of Android device.

3.3.2. Hardware Requirements:

- Server for Database
- Mobile Device with android os

3.4. Screen

The most important Screens:

3.4.1.Login Screen

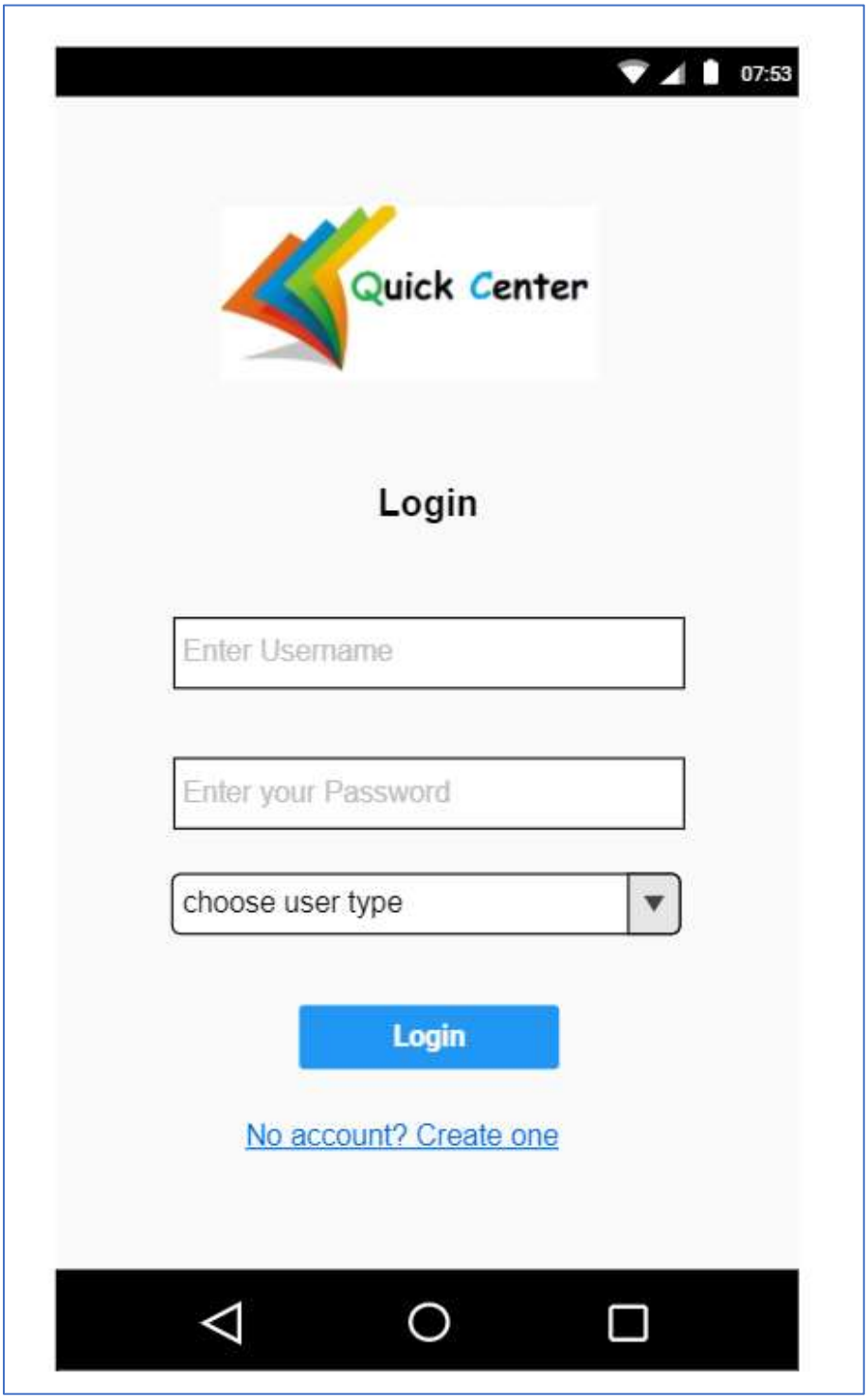


Figure 3-5 Login Screen

3.4.2.Registration Screen

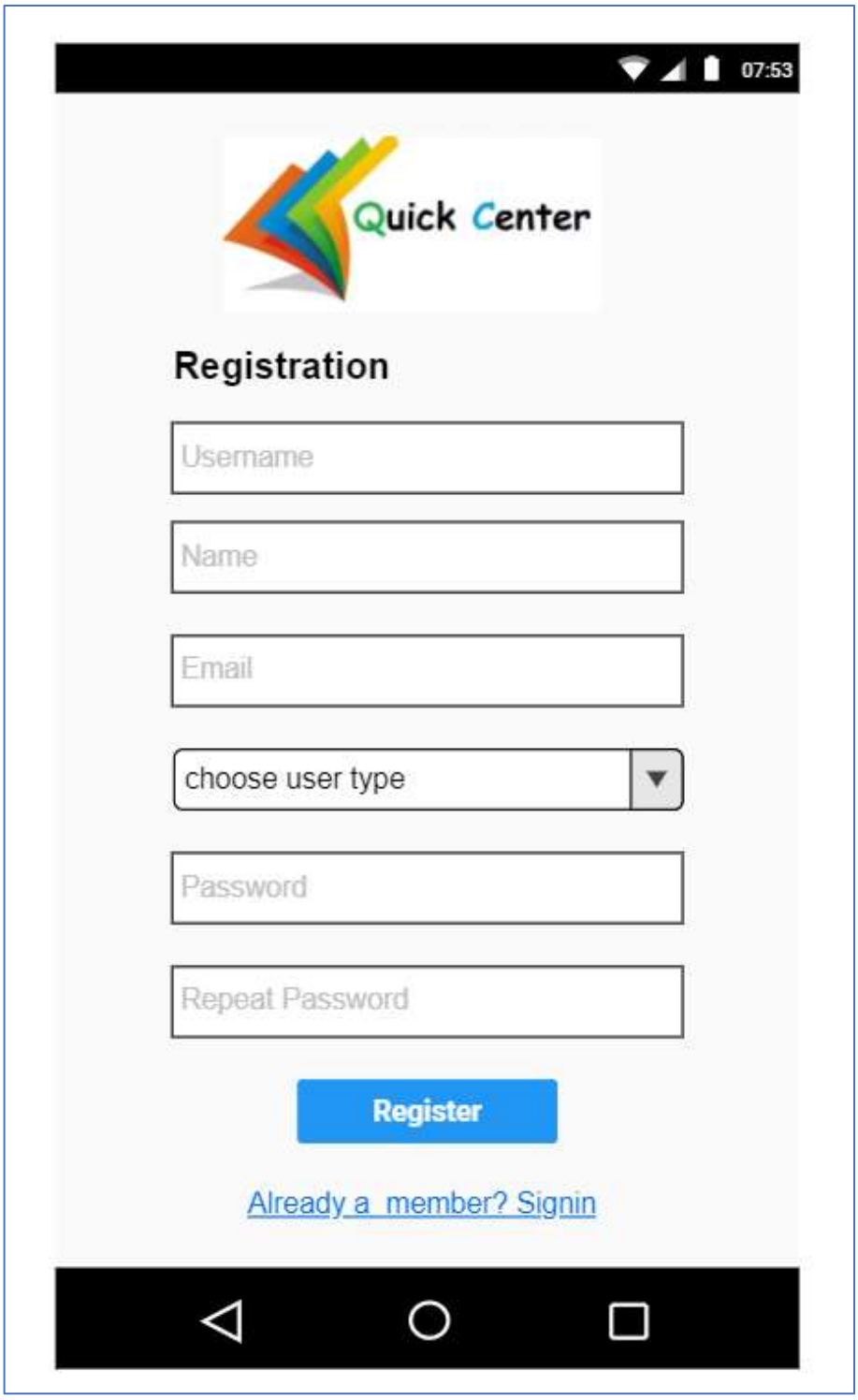


Figure 3-6 Registration Screen

3.4.3.Student Menu

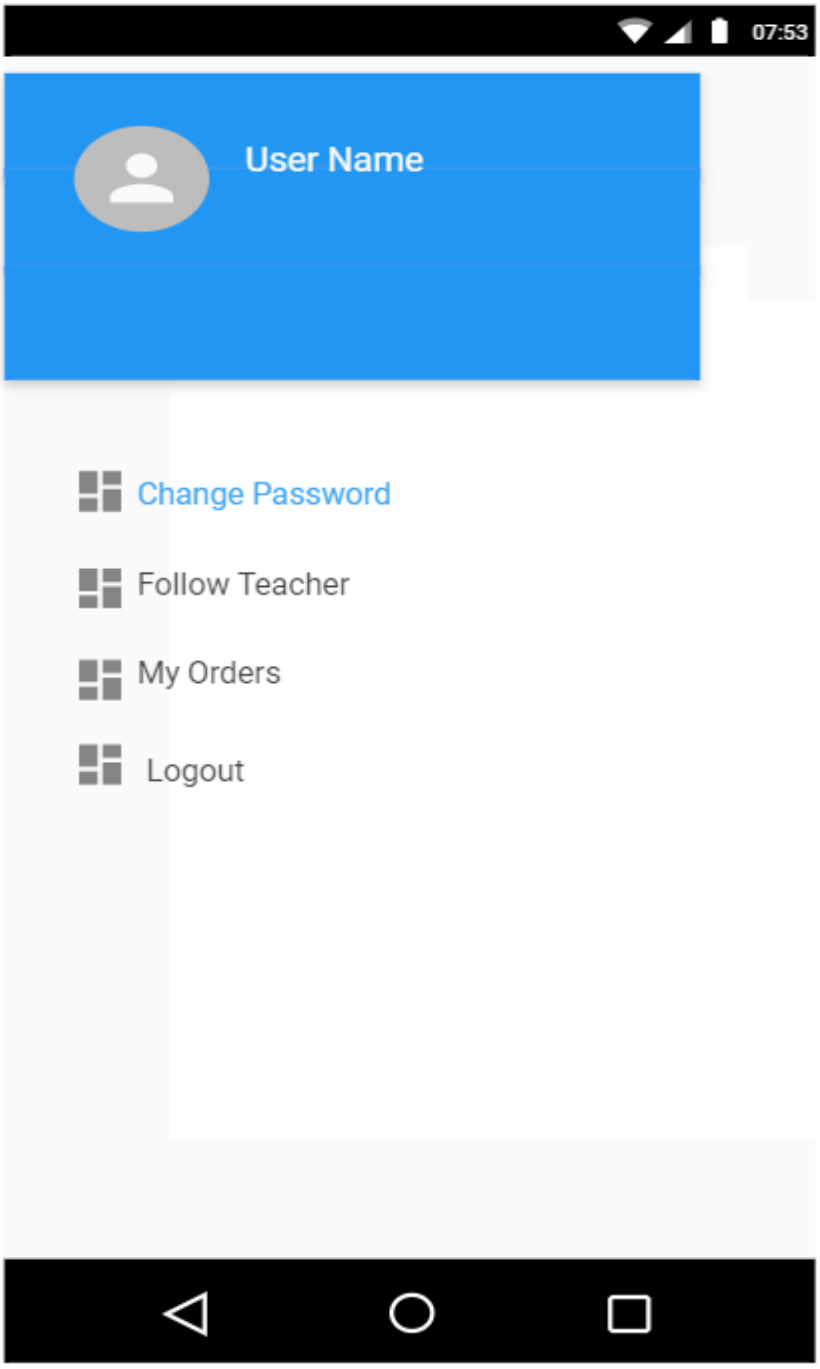


Figure 3-7 Student Menu Screen

3.4.4.Screen center manager

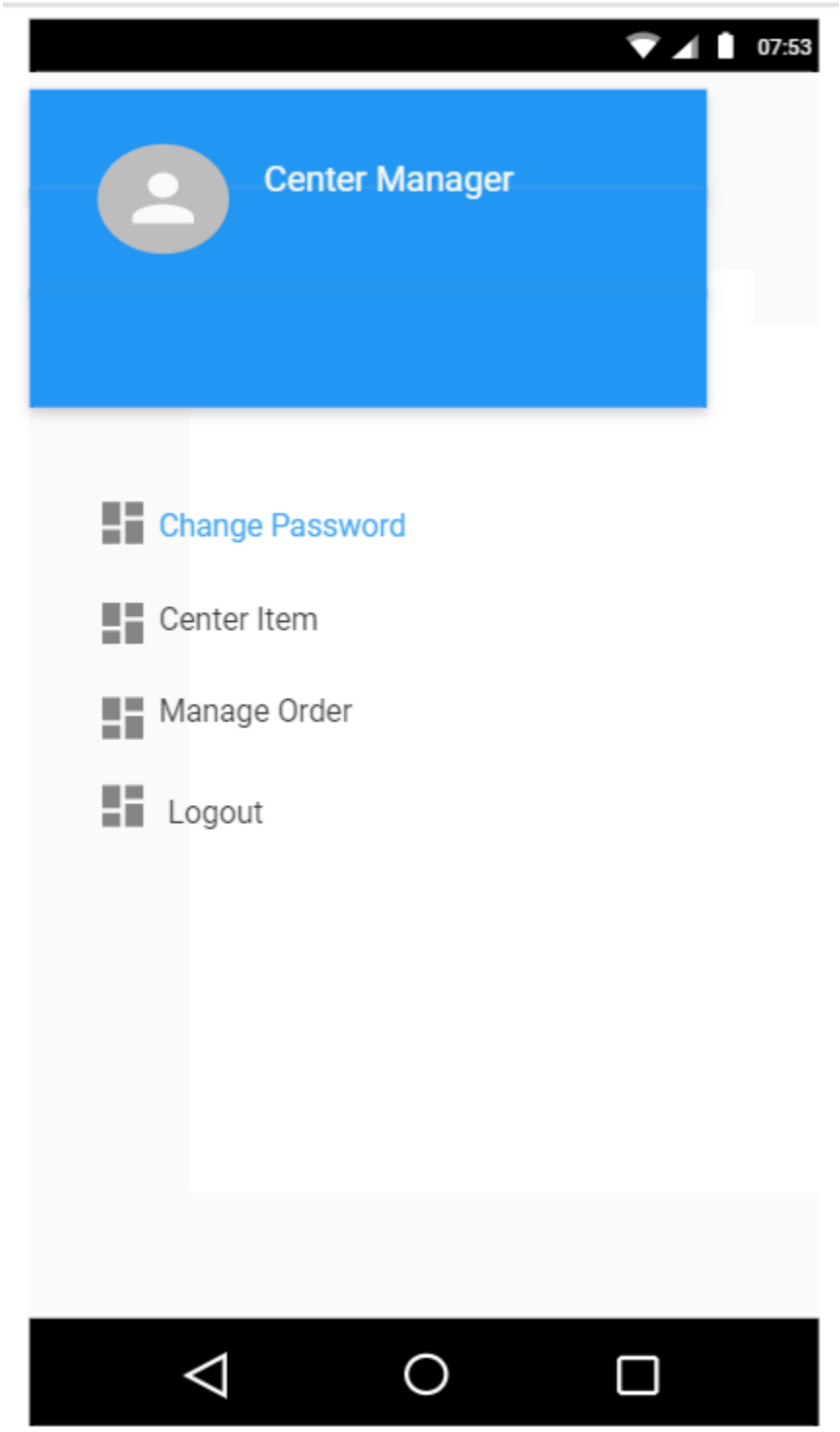


Figure 3-8 Center Manager Menu Screen

4. Implementation

4.1Introduction

The android studio is use to build up the interfaces and to writing java language, As well the MySQL to build the Databases.

4.2 Procedures

- **Login procedure**

```
public void login(View view) {

        System.out.println("**1**");
        mUserlView.setError(null);
        mPasswordView.setError(null);

        // Store values at the time of the login attempt.
       final String UserName =
mUserlView.getText().toString();
       final String password =
mPasswordView.getText().toString();

        System.out.println("**2**");
        boolean cancel = false;
        View focusView = null;

        // Check for a valid password, if the user entered
one.
        if (TextUtils.isEmpty(password)) {

mPasswordView.setError(getString(R.string.error_field_required
));
            focusView = mPasswordView;
            cancel = true;
        }


       if (TextUtils.isEmpty(UserName)) {

mUserlView.setError(getString(R.string.error_field_required));
            focusView = mUserlView;
            cancel = true;
        }

        if (cancel) {
            // There was an error; don't attempt login and
focus the first
```

```
            // form field with an error.
            focusView.requestFocus();
        } else
            {

            mUserlView.setError(null);
            mPasswordView.setError(null);
            new Thread(new Runnable() {

            @Override
            public void run() {
            try {

                Class.forName("com.mysql.jdbc.Driver");
                DB_Connection conn = new DB_Connection();
                conn.DB_Connection_open();
                String sql;
                String[] Userss =
getResources().getStringArray(R.array.Users_array);
                System.out.println(Userss[0]);
String itemNew;
                System.out.println("**************");
                System.out.println(item);
                System.out.println("**************");
                if (item.equals(Userss[0])) {
                    sql = "SELECT ID,UserName,password  FROM
student   where Username=? and Password=?";
                    itemNew="Student";

                } else if (item.equals(Userss[1])) {
                    sql = "SELECT TeacherID,UserName,password
FROM teacher  where Username=? and Password=?";
                    itemNew="Teacher";

                } else {
                    sql = "SELECT MangerID,UserName,password
FROM manger   where Username=? and Password=?";
                    itemNew="Center Manager";

                }

                PreparedStatement prest =
conn.connection.prepareStatement(sql);

                prest.setString(1,UserName);
                prest.setString(2, password);

                ResultSet rs = prest.executeQuery();
```

```
                if (rs.next()) {

                    SharedPreferences.Editor editor =
loginPrefs.edit();
                    editor.putInt("login_username",
rs.getInt(1));
                    editor.putString("login_password",
rs.getString(3));
                    editor.putString("fullname",
rs.getString(2));
                    editor.putString("logged", "logged");
                    editor.putString("login_type", itemNew);
                    editor.apply();


                        if (item.equals(Userss[0]))
                        {

                            Intent i = new
Intent(LoginActivity.this, Home.class);

i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_NEW_TASK);
                            startActivity(i);

                        }
                        else if (item.equals(Userss[1])) {

                            Intent i = new
Intent(LoginActivity.this, TeacherMain.class);

i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_NEW_TASK);
                            startActivity(i);

                        } else
                            {
                                Intent i = new
Intent(LoginActivity.this, ManagerMainActivity.class);

i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_NEW_TASK);
                                startActivity(i);
                        }
```

```
                } else {

                    runOnUiThread(new Runnable() {
                        @Override
                        public void run() {

mPasswordView.setError(getString(R.string.error_incorrect_pass
word));
                            mPasswordView.requestFocus();
                        }
                    });

                }

                prest.close();
                conn.DB_Close();


            } catch (Exception s) {
                System.out.println(s.getMessage());
                System.out.println("SQL statement is not
executed opining!");

            }


             }
             }).start();


        }









    }
```

- **Logout procedure**

```
public void imageClickOut(View view) {
    //Implement image click function

    new AlertDialog.Builder(this)
            .setIcon(R.drawable.exit)
            .setTitle("الخروج !")
            .setMessage("هل أنت متأكد من تسجيل خروجك ؟")
            .setPositiveButton("نعم", new
```

```
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog,
int which) {
                    finish();
                    Intent intent = new
Intent(LoginActivity.this, LoginActivity.class);

intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    intent.putExtra("EXIT", true);
                    startActivity(intent);


                }

            })
            .setNegativeButton("لا", null)
            .show();
}
```

- **Register procedure**

```
public void Registration(View view) {

    email.setError(null);
    password.setError(null);
    Repassword.setError(null);
    Name.setError(null);
    UserName.setError(null);
    String pass=password.getText().toString();
    String cpass=Repassword.getText().toString();
    boolean cancel = false;
    View focusView = null;
    if (email.getText().toString().isEmpty() )
    {

email.setError(getString(R.string.error_field_required));
        focusView = email;
        cancel = true;
    }

    if (Name.getText().toString().isEmpty() )
     {

Name.setError(getString(R.string.error_field_required));
                focusView = Name;
                cancel = true;
    }

    if (UserName.getText().toString().isEmpty() )
```

```
    {

UserName.setError(getString(R.string.error_field_required));
        focusView = UserName;
        cancel = true;
    }

    if (password.getText().toString().isEmpty() )
    {

password.setError(getString(R.string.error_field_required));
        focusView = password;
        cancel = true;
    }



    if (email.getText().toString().isEmpty() )
    {

email.setError(getString(R.string.error_field_required));
        focusView = email;
        cancel = true;
    }



    if(!pass.equals(cpass)){
        password.setError("Wrong Password");
        focusView = password;
        cancel = true;

    }

    if (cancel) {
        // There was an error; don't attempt login and focus
the first
        // form field with an error.
        focusView.requestFocus();

    }
    else
    {
        new Thread(new Runnable() {

            @Override
            public void run() {
                try {
```

```
                    Class.forName("com.mysql.jdbc.Driver");
                    DB_Connection conn = new DB_Connection();
                    conn.DB_Connection_open();
                    String sql;
                    String[] Userss =
getResources().getStringArray(R.array.Users_array);

                    if (
UserType.getSelectedItem().toString().equals(Userss[0]))
                    {
                        sql ="INSERT INTO
student(Name,Email,UserName, Password) VALUES"+"(?,?,?,?)";

                    }
                    else if (
UserType.getSelectedItem().toString().equals(Userss[1]))
                    {
                        sql ="INSERT INTO
teacher(Name,Email,UserName, Password) VALUES"+"(?,?,?,?)";

                    }
                    else
                    {
                        sql ="INSERT INTO
manger(Name,Email,UserName, Password) VALUES"+"(?,?,?,?)";
                    }

                    PreparedStatement prest =
conn.connection.prepareStatement(sql);
                    prest.setString(1,
Name.getText().toString() );
                    prest.setString(2,
email.getText().toString() );
                    prest.setString(3,
UserName.getText().toString() );
                    prest.setString(4,
password.getText().toString() );

                    int res = prest.executeUpdate();
                    if(res>0)
                    {
                        runOnUiThread(new Runnable() {
                            @Override
                            public void run() {

                                LayoutInflater inflater =
getLayoutInflater();
                                View layout =
inflater.inflate(R.layout.custom_toast,
```

```
                                    (ViewGroup)
findViewById(R.id.custom_toast_container));

                                    TextView text = (TextView)
layout.findViewById(R.id.text);
                                    text.setText("Successfully
Registered ....");
                                    Toast toast = new
Toast(getApplicationContext());

toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);

toast.setDuration(Toast.LENGTH_LONG);
                                    toast.setView(layout);
                                    toast.show();
                                }
                            });
                        }
                        else{

                            //failed
                        }

                        prest.close();
                        conn.DB_Close();



                    } catch (Exception s) {
                        System.out.println(s.getMessage());
                        System.out.println("SQL statement is not
executed opining!");

                    }


                }
            }).start();

    }


}
```

- **Change Password procedure**

```
public void change_password() {

```

```
    password.setError(null);
    oldpassword.setError(null);
    Repassword.setError(null);
    boolean cancel = false;
    View focusView = null;

    if (password.getText().toString().isEmpty() )
    {

password.setError(getString(R.string.error_field_required));
        focusView = password;
        cancel = true;
    }

    if (oldpassword.getText().toString().isEmpty() )
    {

oldpassword.setError(getString(R.string.error_field_required))
;
        focusView = oldpassword;
        cancel = true;
    }

    if (Repassword.getText().toString().isEmpty() )
    {

Repassword.setError(getString(R.string.error_field_required));
        focusView = Repassword;
        cancel = true;
    }


    if (cancel) {
        // There was an error; don't attempt login and focus
the first
        // form field with an error.
        focusView.requestFocus();

    }
    else
    {

        password_intent =
loginPrefs.getString("login_password", null);


        if
(!password_intent.equals(oldpassword.getText().toString()))
```

```
        {

oldpassword.setError(getString(R.string.error_incorrect_passwo
rd));
            focusView = oldpassword;
            cancel = true;

        }


if(!password.getText().toString().equals(Repassword.getText().
toString())){
            Repassword.setError("Password does not match");
            focusView = Repassword;
            cancel = true;
        }

        if (cancel) {
            // There was an error; don't attempt login and
focus the first
            // form field with an error.
            focusView.requestFocus();

        }
        else {

            new Thread(new Runnable() {

                @Override
                public void run() {
                    try {


Class.forName("com.mysql.jdbc.Driver");
                        DB_Connection conn = new
DB_Connection();
                        conn.DB_Connection_open();
                        String sql;

                        if (
loginPrefs.getString("login_type",
null).toString().equals("Student"))
                        {
                            sql ="update student set
Password=? where ID=?";

                        }
                        else if (
loginPrefs.getString("login_type",
```

```
null).toString().equals("Teacher"))
                        {

                            sql ="update teacher set
Password=? where TeacherID=?";

                        }
                        else
                        {
                            sql ="update manger set Password=?
where MangerID=?";
                        }


                        PreparedStatement prest =
conn.connection.prepareStatement(sql);
                        prest.setString(1,
password.getText().toString() );
                        prest.setInt(2,
loginPrefs.getInt("login_username", 0));

                        int res = prest.executeUpdate();


                        new
Handler(Looper.getMainLooper()).post(new Runnable() {
                            @SuppressLint("SetTextI18n")
                            @Override
                            public void run() {
                                LayoutInflater inflater =
getLayoutInflater();
                                View layout =
inflater.inflate(R.layout.custom_toast,
                                        (ViewGroup)
getView().findViewById(R.id.custom_toast_container));

                                TextView text = (TextView)
layout.findViewById(R.id.text);

text.setText(R.string.change_succ_pass);
                                Toast toast = new
Toast(getContext());

toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);

toast.setDuration(Toast.LENGTH_LONG);
                                toast.setView(layout);
                                toast.show();
```

```
                                    if
(loginPrefs.getString("login_type",
"").toString().equals("Student"))

                                    {
                                        Intent intent = new
Intent(getActivity(), Home.class);
                                        startActivity(intent);

                                    }

                                    else
                                    {
                                        if
(loginPrefs.getString("login_type",
"").toString().equals("Teacher"))

                                        {
                                            Intent intent = new
Intent(getActivity(), TeacherMain.class);
                                            startActivity(intent);

                                        }

                                        else

                                        {

                                            Intent intent = new
Intent(getActivity(), ManagerMainActivity.class);
                                            startActivity(intent);

                                        }

                                    }


                                        }
                            });



                            prest.close();
                            conn.DB_Close();


                        } catch (Exception s) {
                            System.out.println(s.getMessage());
                            System.out.println("SQL statement is
```

```
not executed opining!");

                        }


                    }
                }).start();

            }
        }



}
```

- **Search Iteam procedure**

```
public void searchItem(String textToSearch){

    Iterator<listitme> iter = listitmes.iterator();
    while(iter.hasNext()){

        if(!iter.next().name.contains(textToSearch)){

            iter.remove();

        }

    }



    adapter.notifyDataSetChanged();

}
```

4.3 Reports

The data saved in Database

- **Files Table**

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	**FileID**	int(11)			No	*None*		AUTO_INCREMENT
☐	2	**Description**	varchar(200)	latin1_swedish_ci		No	*None*		
☐	3	**File**	longblob			No	*None*		
☐	4	**MangerID**	int(11)			Yes	*NULL*		
☐	5	**Teacher_ID**	int(11)			Yes	*NULL*		
☐	6	**Student_ID**	int(11)			Yes	*NULL*		

Figure 3-4 Login Screen

- **following teacher Table**

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**StudentID**	int(11)			No	*None*			Change Drop P
☐	2	**TeacherID**	int(11)			No	*None*			Change Drop P

Figure 3-5 following teacher

- **Item Table**

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	**itemID**	int(11)			No	*None*		AUTO_INCREMEN
☐	2	**Name**	text	latin1_swedish_ci		No	*None*		
☐	3	**Description**	text	latin1_swedish_ci		No	*None*		
☐	4	**Price**	decimal(10,0)			No	*None*		
☐	5	**ManagerId**	int(11)			No	*None*		

Figure 3-6 Item table

- **Manger Table**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**MangerID**	int(11)			No	*None*		AUTO_INCREMENT
2	**Name**	varchar(255)	latin1_swedish_ci		No	*None*		
3	**Email**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
4	**UserName**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
5	**Password**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		

Figure 3-7 Manager Table

- **Notification Table**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**ID**	int(11)			No	*None*		AUTO_INCREMENT
2	**Description**	varchar(100)	latin1_swedish_ci		No	*None*		
3	**DateCreated**	datetime			No	*None*		
4	**TeacherID**	int(11)			Yes	*NULL*		
5	**StudentID**	int(11)			Yes	*NULL*		
6	**ReadFlag**	varchar(1)	latin1_swedish_ci		No	N		

Figure 3-8 Notification Table

- **orders Table**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**OrderId**	int(11)			No	*None*		AUTO_INCREMEN
2	**File**	int(11)			No	*None*		
3	**Numofcopies**	int(11)			No	*None*		
4	**PaperType**	varchar(50)	latin1_swedish_ci		No	*None*		
5	**PrintType**	varchar(50)	latin1_swedish_ci		No	*None*		
6	**TimeDesired**	datetime			No	*None*		
7	**Note**	text	latin1_swedish_ci		No	*None*		
8	**Status**	varchar(100)	latin1_swedish_ci		No	*None*		
9	**StudentID**	int(11)			Yes	*NULL*		
10	**DateCreate**	datetime			No	*None*		
11	**Teacher_ID**	int(11)			Yes	*NULL*		

Figure 3-9 Orders Table

- **Student Table**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**ID**	int(11)			No	*None*		AUTO_INCREMENT
2	**Name**	varchar(255)	latin1_swedish_ci		No	*None*		
3	**Email**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
4	**UserName**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
5	**Password**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		

Figure 3-10 Student Table

- **Teacher Table**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**TeacherID**	int(11)			No	*None*		AUTO_INCREMENT
2	**Name**	varchar(255)	latin1_swedish_ci		No	*None*		
3	**Email**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
4	**UserName**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
5	**Password**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		

Figure 3-11 Teacher Table

4.4 Layouts

- **Main Screen**

  This Screen is the main and first interface in the application which is used to login to the system by choosing the user type and entering the user name and password.

Figure 3-12 Main Screen

- **Register Screen**

  This Screen is used for make registration for the user if he/she is new visitor to the application and the registration happens by entering the personal information and making username and password.

Figure 3-13 Register Screen

- **Center Item Screen**

This Screen is using to display all items in the center with short details

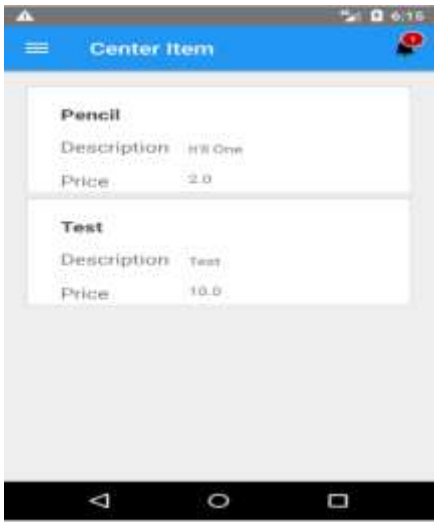


Figure 3-14 Item for print Screen

- **Menu for student Screen**

This Screen is for student that he/she can manage his/her account and make some processes as viewing his/her orders, follow teacher, and place order.

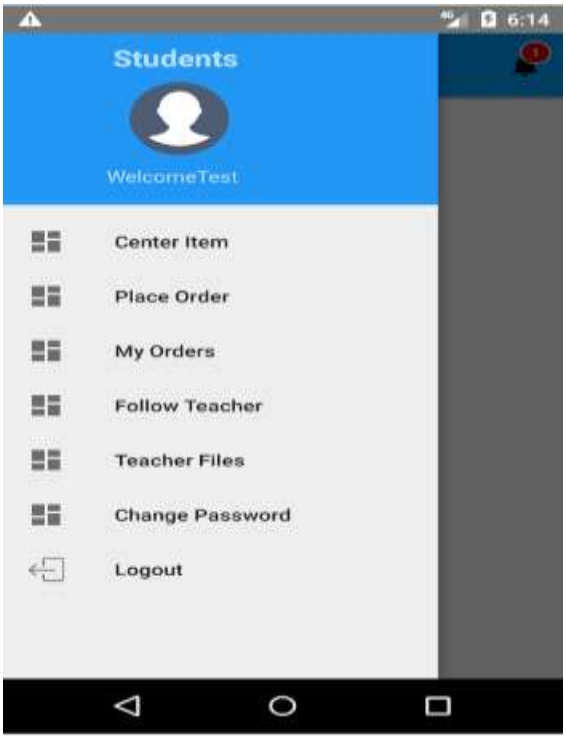


Figure 3-15 menu for student Screen

- **Place order Screen**

  This Screen is enabling the user from making order by specifying the order contents.

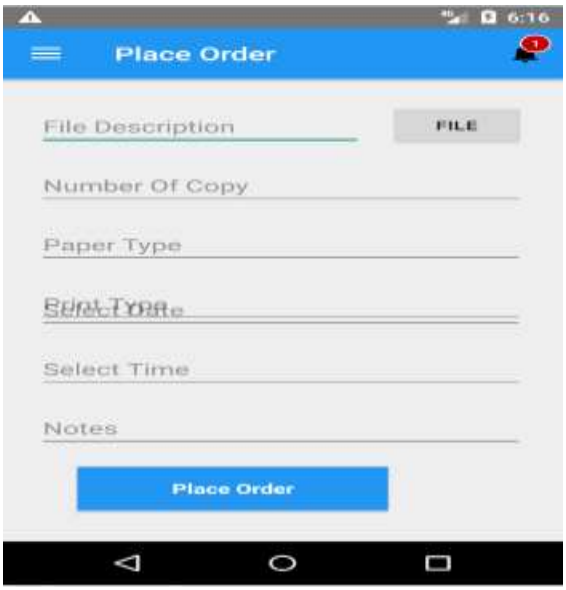


Figure 3-16 Place order Screen

- **Notify Screen**

  This Screen is giving the users message about changing the order.

Figure 3-17 Notify Screen

- **My order Screen**

  This Screen is presenting all orders for specify user.

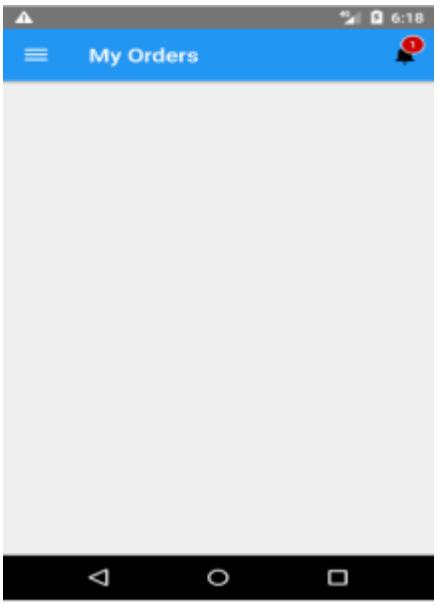


Figure 3-18 my order Screen

- **Follow teacher Screen**

  This Screen enable the student to make following for special teacher and canceling the following.

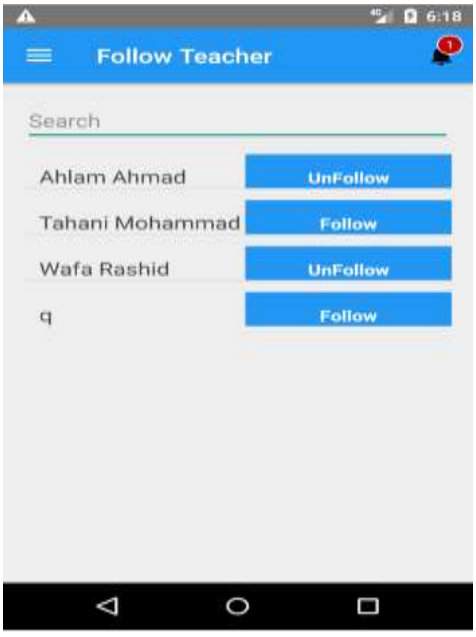


Figure 3-19 Follow teacher Screen

- **Teacher files Screen**

  This Screen enable the student to make searching for specified teacher to make following for him.

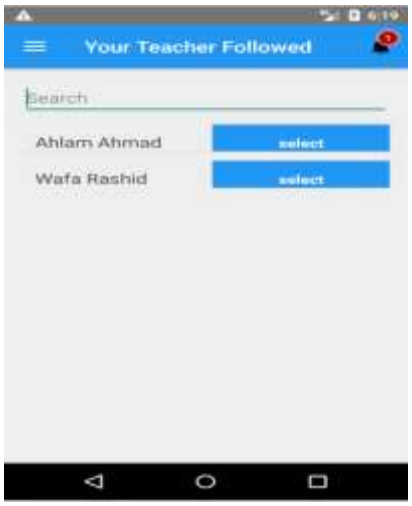


Figure 3-20 Teacher file Screen

- **Change password Screen**

  This Screen enable the user from changing his password to save his/her account from any probable attacking in future.

Figure 3-21 Change Password Screen

4.5 Reports layouts

-In this Screen, Teacher can select file and enter description then press upload button to upload file in project.

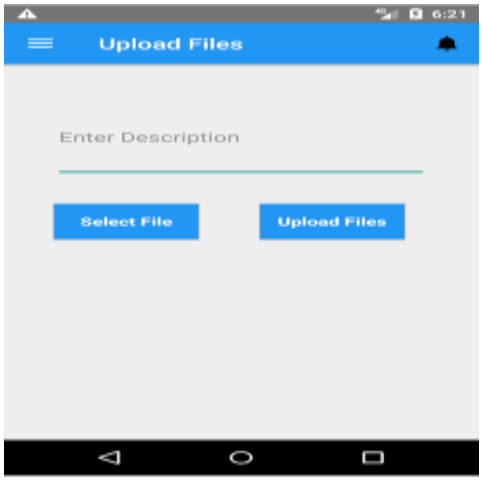


Figure 3-19 Upload file Screen

-Student and Teacher can insert new order for print by input all fields in this interface and select file then press place order.

Figure 3-20 Place order Screen

5. Conclusion

Quick center application implements some services for student, teachers and center manager electronically without the need to return to traditional manual procedures that take a lot of time. The proposed framework follows are layered architecture approach that facilitates deployment of new technologies, with a view to minimizing the loss of time and effort.

Quick Center is a mobile application will be designed to provide complete system from ordering printing requests to execute the order and notify the student in easy and simple way using a mobile device.

6. References

[1] http://www.conceptdraw.com/How-To-Guide/data-flow-diagram-symbols 01-03-2018

[2] https://circle.visual-paradigm.com/food-ordering-system-context-dfd/ 01-03-2018

[3] https://www.lifewire.com/entity-relationship-diagram-1019253

[4] https://www.tutorialspoint.com/dbms/dbms_data_schemas.htm 18-03-2018

[5] https://www.quora.com/What-is-JDBC 20-03-2018

[6] https://www.smartdraw.com/activity-diagram/ 21-03-2018

[7] https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html 21-03-2018

[8] https://creately.com/blog/diagrams/class-diagram-relationships/21-03-2018

[9] https://www.uml-diagrams.org/use-case-diagrams.html 21-03-2018

[10] https://www.uml-diagrams.org/state-machine-diagrams.html 21-03-2018

[11] https://www.smartdraw.com/sequence-diagram/ 21-03-2018

7. Appendixes A:Functional Requirements

Functional requirements are concerned with the set of services our app should provide, they show how the app will react to situations and when provided with various inputs, so in this step we establish a set of overall objectives that the app should meet.

This system will be used by the following users:

- Administrator (Center Manager)
- Teachers
- Students

as all of these have different requirements to meet their needs and avoid any type of confusion. The functional requirements of all users have been described below.

Administrator will need the following functional requirements:

- Login to the application by username and password.
- Show Orders of printing request
- Change Orders status
- Add Center Items

Students will need the following functional requirements:

- Registration in application
- Login to application.
- Add New Orders

Teachers will need the following functional requirements:

- Registration in application
- Login to application.
- Upload files

Appendixes B:Questions

1-The print center branch at the college service is quite perfect؟

- yes
- No

2-Do you think the center needs more development؟

- Yes
- No

3-Did the waiting line at the center is a bit long؟

- Yes
- No

4-Did you arrive late in lecture because you wanted for printing؟

- Yes
- No

5-Did you west your break time in waiting for printing ؟

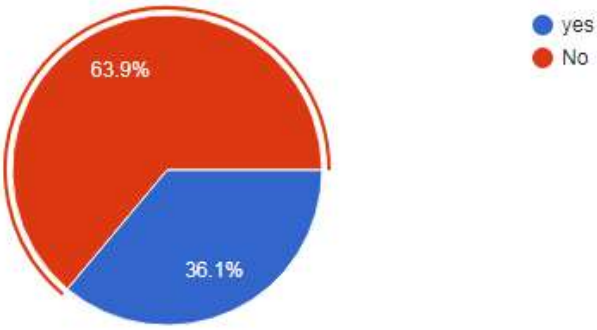
- Yes
- No

6-What if you sent your file or document before in an application, then you find you need ready to pay in time you want without waiting in line؟

- Yes
- No

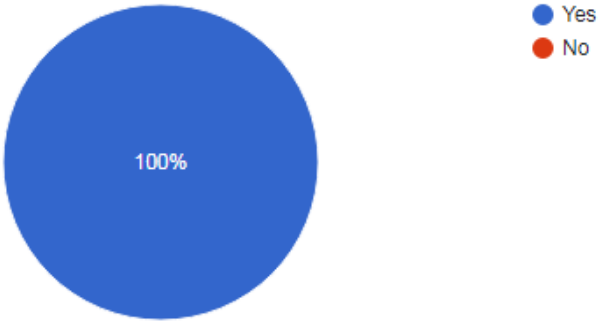
?The print center branch at the college service is quite perfect-1

٣٦ ردًا



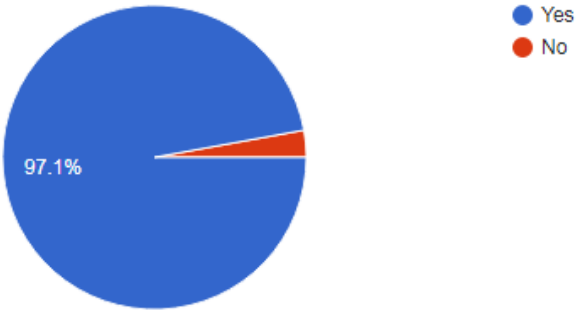
?Do you think the center needs more development -2

٣٥ ردًا



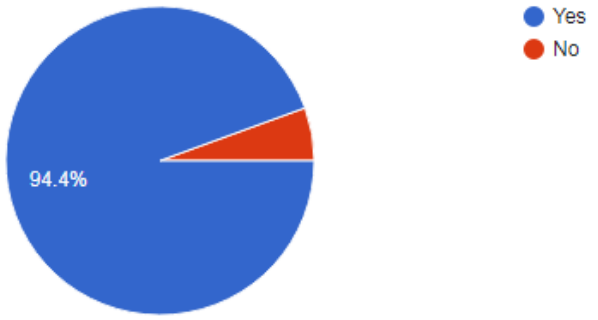
?Did the waiting line at the center is a bit long-3

٣٥ ردًا



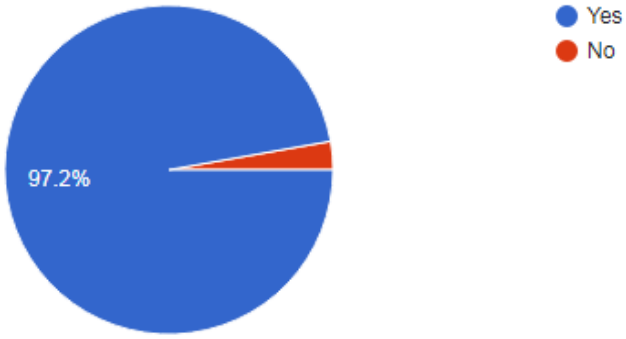
?Did you arrive late in lecture because you wanted for printing-4

٣٦ ردًا



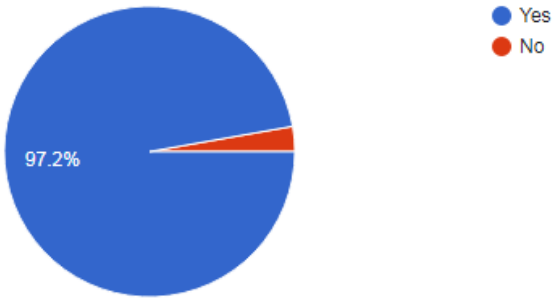
?Did you wast your break time in waiting for printing-5

٣٦ ردًا



What if you sent your file or document before in an application, then-6
?you find you need ready to pay in time you want without waiting in line

٣٦ ردًا



Appendixes C:Feasibility Study

Technical Feasibility Study

The technical requirements for the project work are:

1.Mobile Device.

2. Connection to the Internet.

3. Rent a server with the specifications specified in Table (3.1).

6. Programming languages such Java and Android Studio.

7. Databases My SQL

Operational Feasibility Study

There are no operating requirements just to tell users how to work on mobile applications

Economic Feasibility Study

The economic costs It is only a database hosting . the software is open source software and free.