SFDS: STUDENT FACULTY DOCUMENT SHARING SYSTEM

A project report submitted to University of Calicut

In partial fulfillment of requirements for the award of

BACHELOR DEGREE

In

COMPUTER APPLICATION

By

SREELAKSHMI K R

REG. NO: EKARBCA023

(2017-2020)

DEPARTMENT OF COMPUTER SCIENCE

SAHRDAYA

COLLEGE OF ADVANCED STUDIES

KODAKARA, THRISSUR

SFDS: STUDENT FACULTY DOCUMENT SHARING SYSTEM

A project report submitted to University of Calicut

In partial fulfillment of requirements for the award of

BACHELOR DEGREE

In

COMPUTER APPLICATION

By

SREELAKSHMI K R

REG. NO: EKARBCA023

(2017-2020)

Under the supervision of

MS.VARSHA M M

DEPARTMENT OF COMPUTER SCIENCE

SAHRDAYA

COLLEGE OF ADVANCED STUDIES

KODAKARA, THRISSUR

SFDS: STUDENT FACULTY DOCUMENT SHARING SYSTEM

PROJECT REPORT

UNIVERSITY OF CALICUT

CENTRE: SAHRDAYA COLLEGE OF ADVANCED STUDIES, KODAKARA

BCA

(2017-2020)

Name : SREELAKSHMI K R

Reg. No : EKARBCA023

Semester : Sixth

EVER TO EXCEL

SAHRDAYA

COLLEGE OF ADVANCED STUDIES

PB No. 18, Kodakara, Thrissur Dist., Kerala-680684
Phone: 0480 27113713, Web: www.sahrdayacas.ac.in

Affiliated to University of Calicut, Approved by Govt. of Kerala
Managed by Irinjalakuda Diocesan Educational Trust (IDET)

DEPARTMENT OF COMPUTER SCIENCE

This is to certify that the report of the project entitled ..."SFDS"...
... is a bonafide record of the original work done by ..Sreelakshmi K.R..... (Reg.No.) ..EKARBCA023.... during the year 2017-2019..... in partial fulfilment of the requirements for the award of Bachelor Degree in Computer Applications under the University of Calicut, Kerala.

Date : 26/10/2019

Dr. JENSON PO
PRINCIPAL
SAHRDAYA
COLLEGE OF ADVANCED STUDIES
KODAKARA-680 684

.....
Project Guide

H.O.D
DEPARTMENT OF C S & I T
SAHRDAYA
COLLEGE OF ADVANCED STUDIES
KODAKARA - 680 684
Head of the Department

13/8/2019

.....
External Examiner

SAHRDAYA
COLLEGE OF ADVANCED STUDIES

CERTIFICATE

This is to certify that **Ms. SREELAKSHMI K R** Reg.No. **EKARBCA023** Department of Computer Science, Sahrdaya College of Advanced Studies, Kodakara has successfully completed the main project entitled **"SFDS"** in the period from October 2019 to March 2020 under the supervision of IT Development Team, SCAS.

During this period, she was found to be very honest, sincere and hardworking and she performed well

For IT Development team, SCAS

HoD

H.O.D
DEPARTMENT OF C S & I T
SAHRDAYA
COLLEGE OF ADVANCED STUDIES
KODAKARA-680 684

+91 480 2713713, 9497 506 109
+91 9497 233 713
Kodakara, P.B.No.18, Thrissur (Dt).
Pin 680684, Kerala State, India
info@sahrdayacas.ac.
www.sahrdayacas.ac.

DECLARATION

I hereby declare that the project entitled **"SFDS"** submitted to the University of Calicut in partial fulfillment of the requirement for the award of the **BACHELOR OF COMPUTER APPLICATION** is a record of original work done by me during the period of my study in the Department of Computer Science, Sahrdaya College of Advanced Studies, Kodakara.

Place : Kodakara SREELAKSHMI K R

Date : 13/ 03/2020

ACKNOWLEDGEMENT

The effort taken by the completion of this project would not have been possible without the kind of support and help rendered by many individuals and organizations. I would like to acknowledge my sincere thanks to all of them.

First I would like to thank God Almighty for his glorious blessings which have been accompanied throughout the course of my work. I firmly believe, it was with His help that I was able to complete this project successfully.

I am highly indebted to Fr. Davis Chenginiyadan, Executive Director and Dr.P. O. Jenson, Principal and Dr. Rani M J, Vice Principal and Dean of Science for their guidance and constant supervision as well as providing all facilities to carry out my project as part of the curriculum.

I extend my deepest sense of gratitude to my guide Ms.Varsha M M Department of Computer Application for his/her patience and support together with proper guidance throughout the completion of my project.

I record sincere thanks to Ms. Sandhya Sharma, Head of the Department of Computer Application and all other teachers of the department for their strong support and help.

I would like to express my special gratitude and thanks to project support team, SCAS for giving me such attention and time.

I also express my gratitude to the Computer Lab Assistants for providing various facilities for this project.

I am very thankful to all my classmates and friends for their whole hearted support.

SREELAKSHMI K R

ABSTRACT

Student-Faculty Document Sharing System is an Android application that provides a generalized solution to monitor the various works that are carried out by a department of college for managing it.This provides a simple interface for the maintenance of student information.This app deals with all kind of student details, academic related reports, department details/notifications (also internal marks, time table, etc.) in department wise.In this app both the student and faculty can login. Faculty may upload documents of subjects, timetable, notifications etc., through their provided login. The documents are uploaded by faculty to different corresponding departments. The students can view and download required documents. In this app, the facility of broadcast messaging is available between students and faculties.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- DFD DATA FLOW DIAGRAM
- ER ENTITY RELATIONSHIP
- XML EXTENSIBLE MARKUP LANGUAG'E
- SDK SOFTWARE DEVELOPMENT KIT
- MVC MODEL VIEW CONTROLLER
- IDE INTEGRATE DEVELOPMENT ENVIRONMENT
- SRS SOFTWARE REQUIREMENT SPECIFICATION

1. INTRODUCTION

Today's world runs on computers and smartphones. Nearly in every aspect of modern life Involves computers in some form or fashion. With the growing speed of technological advancement, smartphones have become the essential components of our daily performance. As we look for convenience, we also respect the devices, which can combine multiple features and which give us more mobility and entertainment. As the whole world is going into the new phase of technological performance, our needs become more sophisticated. On the one hand, we need speed, quality, and effectiveness, on the other hand, these features should be combined in a solution small enough to carry it in the pocket. There is a tremendous increase in users with an Android-based phone, Smartphone and tablet computers. This application provides a generalized solution to monitor the various works that are carried out by a Department of College for managing it. The design and implementation of an ambient student information system and user interface is to replace the current paper records. Department Staffs are able to directly access all aspects of a student's academic progress through a secure, online interface embedded using.

1.1 PROJECT OVERVIEW

This project is about sharing documents between students and faculty. This system allows department faculty to share important data as well as notifications of exam with the students. It consists of a faculty login and student login. Faculty may upload documents of internal mark, timetable document, notifications, notes and mark list through their provided login. The documents are uploaded by faculty. This system is built on an online server by allowing faculty to upload documents and students may download required documents through their android device. Here students only access and download data. Faculty may access and upload/edit documents. There is Broadcast messaging corner for the student and faculty to interact. Only the registered students and faculty can perform the desired operations.

1.2 OBJECTIVE

The primary objective of the project is to introduce a highly organized android application which can upload, download and view the documents.

- Faculties can easily find the students that they were under them
- The documents and other updates are uploaded by the faculties to their students
- Students can access the data uploaded by the faculty and view it
- Students can only view and download the data. They were not able upload any kind of data
- Faculties can send message to the selected students
- Sharing of important documents without any delay

2. SYSTEM DESCRIPTION

2.1 HARDWARE SPECIFICATION

The hard ware is the place where all the information and data are store permanently. So hardware must be reliable and cost effective. The hardware must suit all the application development. It is fast enough to complete and do all the jobs and executions.

Processor: Intel Core i3 processor

Android device: ARMv7 Processor

Phone memory: 15MB

SD card: 500MB

2.1 SOFTWARE SPECIFICATION

The software specification means the operating system and all other application or tools used for the development of the proposed system. It includes the operating system, and the software which we are going to use.

Operating System: Windows 10

Front end: Android

Back end: Java, XML

Framework: Android Studio 1.5.1

Android SDK: API 17 or higher

2.3 DEVELOPING TOOL

Android Studio is the official integrated (IDE) for Android platform development. It was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache License 2.0. Android Studio was in early access preview stage starting from

version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. Based on JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows, Mac OS X and Linux, and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

3 SOFTWARE DESCRIPTION

3.1 DEVELOPING PLATFORM

WINDOWS 10

Windows 10 is a personal computer operating system developed and released by Microsoft as a part of Windows NT family of operating systems. It was officially unveiled in September 2014 following a brief demo at Build 2014. The first version of the operating system entered a public beta testing process in October 2014, leading up to its consumer release on July 29, 2015

Windows 10 introduces what Microsoft described as "universal apps"; expanding on Metro-style apps, these apps can be designed to run across multiple Microsoft product families with nearly identical code including PCs, tablets, smartphones, embedded systems, Xbox ones, surface hub and Windows Holographic. The Windows user interface was revised to handle transitions between a mouse-oriented interface and a touch-screen optimized interface based on available input devices particularly on 2-in-1 PCs; both interfaces include an updated start menu which incorporates elements of windows 7's traditional start menu with the tiles of Windows 8

Microsoft described Windows 10 as an "Operating system as a service" that would receive or-going updates to its features are functionality, augmented with the ability for enterprise environments to receive non-critical updates, at a slower pace, or use long-term support milestones that will only receive critical updates, such as security patches, over their five-year lifespan of mainstream support

Windows 10 received mostly positive reviews upon its original release in July 2015; critics praised Microsoft's decision to downplay user-interface mechanics introduced by Windows 8 (including the full screen apps and start screen) in nor-touch environments to provide a desktop-oriented interface in line with previous version of Windows, although Windows 10's touch-oriented user interface mode was panned for containing regressions upon the touch-oriented interface of Windows 8.

Windows 10 was criticized for limiting how users can control its operation, including limited controls over the installation of updates on the main consumer-oriented edition in comparison to previous versions. Privacy concerns were also voiced by critics and advocates, as the operating system's default settings and certain features require the transmission of user data to Microsoft or its partners. Microsoft has also received criticism for how it has included the automatic downloads of installation files to computers, the recurring display of pop-ups advertising the upgrade, and allegations of the installation process being scheduled or initiated automatically without expressed user consent.

3.2 FRONT END TOOL

ANDROID

Android is a mobile operating system that is based on a modified version of Linux. It was originally developed by a start-up of the same name, Android, Inc. In 2005, as a part of its strategy to enter the mobile space, Google purchased Android and took over its development work (as well as its development team). Google wanted Android to be open and free; hence, most of the Android code was released under the open source Apache License, which means that anyone who wants to use Android can do so by downloading the full Android source code. This simple development model makes Android very attractive and has thus piqued the interest of many vendors. Tis has been especially true for companies affected by the phenomenon of Apple's iPhone, a hugely successful product that revolutionized the smartphone industry. Such companies include Motorola and Sony Ericsson, Which for many years have been developing their own mobile operating systems. When the iPhone was launched, many of these manufactures hand to scramble to fire new ways of revitalizing their products. These manufactures see Android as a solution they will continue to design their own hardware and use Android as the operating system that powers it. The main advantage of adopting Android is that it offers a unified approach to application development. Developers need only develop for Android, and their applications should be able run on numerous different devices, as long as the devices are powered using Android. In the world of smart phones, applications are the most important part of the success chain. Device manufactures therefore see

Android as their best hope to challenge the onslaught of the iPhone, which already commands a large base of applications.

FEATURES OF ANDROID:

Because Android is an open source and freely available to manufacturers for customization, there are no fixed are no fixed hardware or software configurations.

However, Android itself supports the following features:

- Storage: Uses SQLite, a lightweight relational database, for data storage.
- Connectivity: Supports GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth (includes A2DP and AVRCP), Wi-Fi, LTE, and WiMAX.
- Messaging: Supports both SMS and MMS.
- Web browser: Based on the open source Web Kit, together with Chrome's V8 JavaScript engine
- Media support Includes support for the following media:H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, HE-AAC(in MP4 or 3GP container), MP3, MIDI, WAV, JPEG, PNG, GIF, and BMP
- Hardware support: Accelerometer sensor, Camera, Digital Compress, Proximity Sensor, and GPS.
- Multi-tasking: Supports multi-tasking applications
- Tethering: Supports sharing of Internet Connections as a wired/wireless hotspot

3.3 BACK END TOOL

JAVA:

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte code that can run on any Java

virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licences. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (byte code compiler), GNU Class path (standard libraries), and IcedTea-Web (browser plugin for applets).

XML:

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is defined by the W3C's XML 1.0 Specification and by several other related specifications, all of which are free open standards.

The design goals of XML emphasize simplicity, generality and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while many application programming interfaces (APIs) have been developed to aid the processing of XML data.

4 SYSTEM ANALYSIS

4.1 EXISTING SYSTEM

There are so many student-faculty management applications available in android market. Some of them provide basic management features. Others provide facilities for only the few of management features. None of them provides all these features together.

4.2 PROPOSED SYSTEM

It is a highly organized android application which is developed for the easy document sharing between students and faculties. With this application, faculties can easily share the important documents like internal marks notes, time table, and mark list to the students. And students were able to download and view these documents. This application also provide the facility of broad cast messaging.

4.3 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of the data processing (structured design)

A DFD 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 whether process will operate sequence or parallel (which is shown on a flowchart).

Data flow diagrams were proposed by Larry Constantine, the original developer of structured design, based on Martin and Estrin's "Data Flow Graph" model of computation. Stating in the 1970's, data flow diagrams (DFD) became a popular way to visualize the major steps and data involved in software system processes.

DFDs were usually used to show data flow in computer system, although they could in theory be applied to business process modeling. DFD were useful to document the major data flows or to explore a new high- level design in terms of data flow.

It is common practice to draw the context- level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. This helps to create an accurate drawing in the context diagram. The system's interactions with the outside world are modeled purely in terms of data flows across the system boundary. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization. Data flow diagrams can be used in both Analysis and Design phase of the SDLC. There are different notation to draw data flow diagrams (Yourdon & Coad and Gane & Sarson, defining different visual representation for processes, data stores, data flow, and external entities.

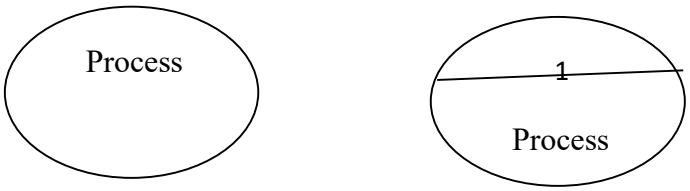
Data Flow Diagram Notations

You can use two different types of notations on your data flow diagram:

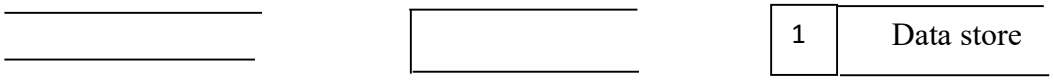
Yourdon & Coad or Gane & Sarson

- **Process Notation**

A process transforms incoming data flow into outgoing dataflow



- **Data Store Notations**



- **Data Flow Notations**

The proper process.

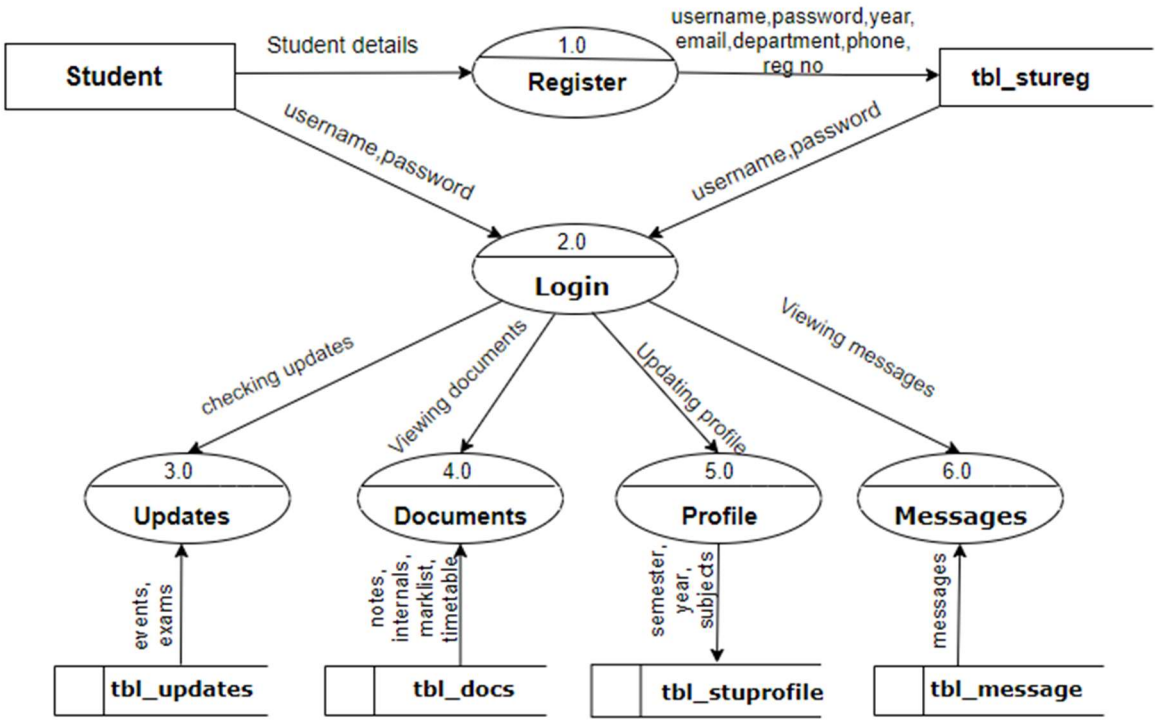
- **External Entity Notations**

This is known as actors, sources or sinks. They produce and consume data that flows between the entity and the system being diagrammed.

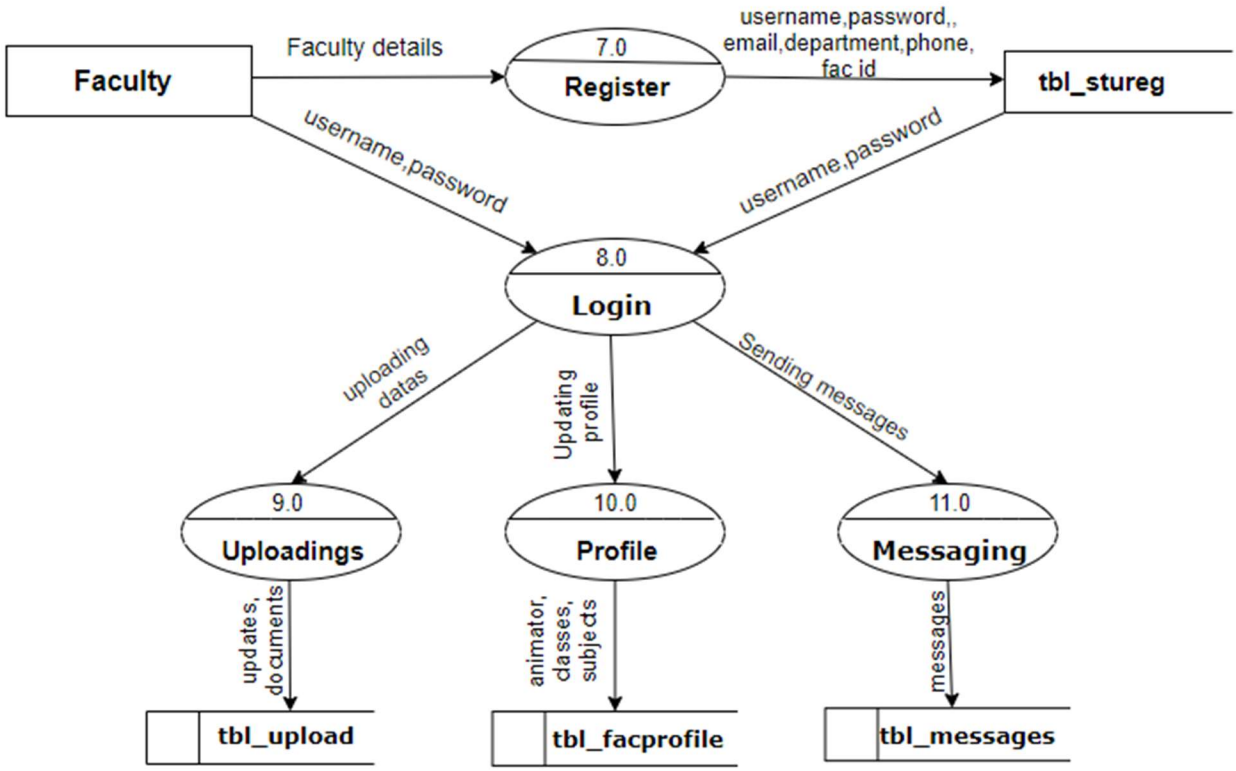
DATA FLOW DIAGRAM (DFD)

LEVEL 0

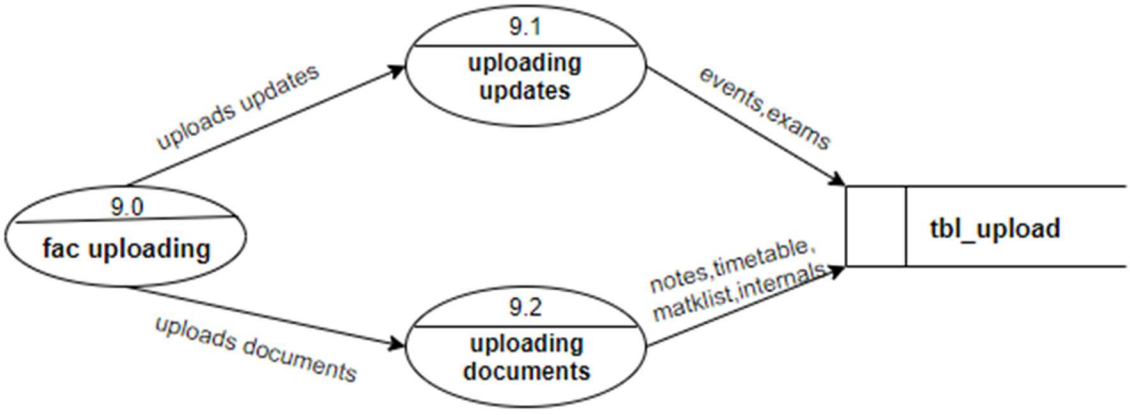
LEVEL 1 (STUDENT)



LEVEL 1 (FACULTY)



LEVEL 2



4.4 ER DIAGRAM

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

The elements of an ERD are:

- Entities
- Relationships
- Attributes

An entity-relationship diagram (ERD) is crucial to creating a good database design. It is used as a high-level logical data model, which is useful in developing a conceptual design for databases.

An entity is a real-world item or concept that exists on its own. Entities are equivalent to database tables in a relational database, with each row of the table representing an instance of that entity.

Relationship is the association that describes the interaction between entities. Cardinality, in the context of ERD, is the number of instances of one entity that can, or must, be associated with each instance of another entity. In general, there may be one-to-one, one-to-many, or many-to-many relationships.

For example, let us consider two real-world entities, an employee and his department. An employee has attributes such as an employee number, name, department number, etc. Similarly, department number and name can be defined as attributes of a department. A department can interact with many employees, but an employee can belong to only one department, hence there can be a one-to-many relationship, defined between department and employee.

In the actual database, the employee table will have department number as a foreign key, referencing from department table, to enforce the relationship.

An ER diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

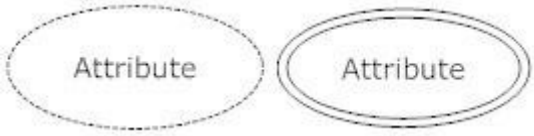
- **Entities**, which are represented by rectangles. An entity is an object or concept about which you want to store information.

A weak entity is an entity that must be defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

- **Actions**, which are represented by diamond shapes, show how two entities share information in the database. In some cases, entities can be self-linked.

- **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.

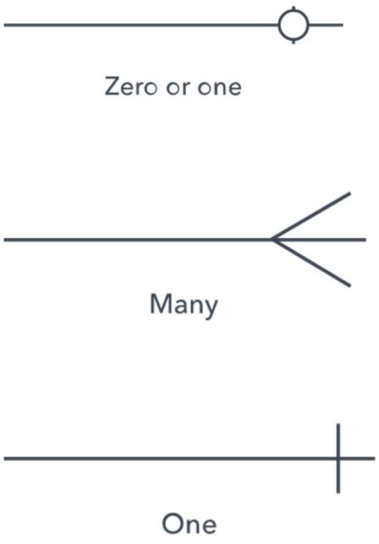
- A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values. A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

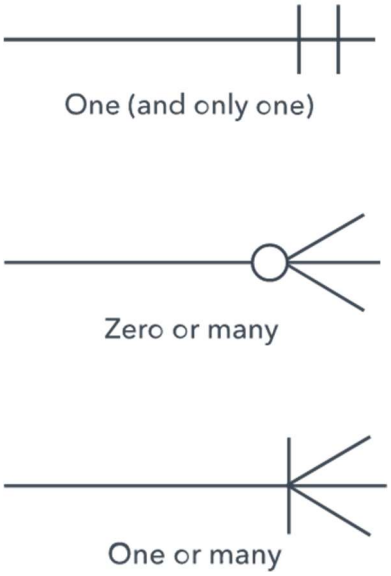


- **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the diagram.

Cardinality

Defines the numerical attributes of the relationship between two entities or entity sets. The three
main cardinal relations are one-to-one, one-to-many, and many-to-many. A **one-to-one example** would be one student associated with one mailing address. A **one-to-many example (or many-to-one, depending on the relationship direction):** One student registers for multiple courses, but all those courses have a single line back to that one student. **Many-to-many example:** Students as a group are associated with multiple faculty members, and faculty members in turn are associated with multiple students.

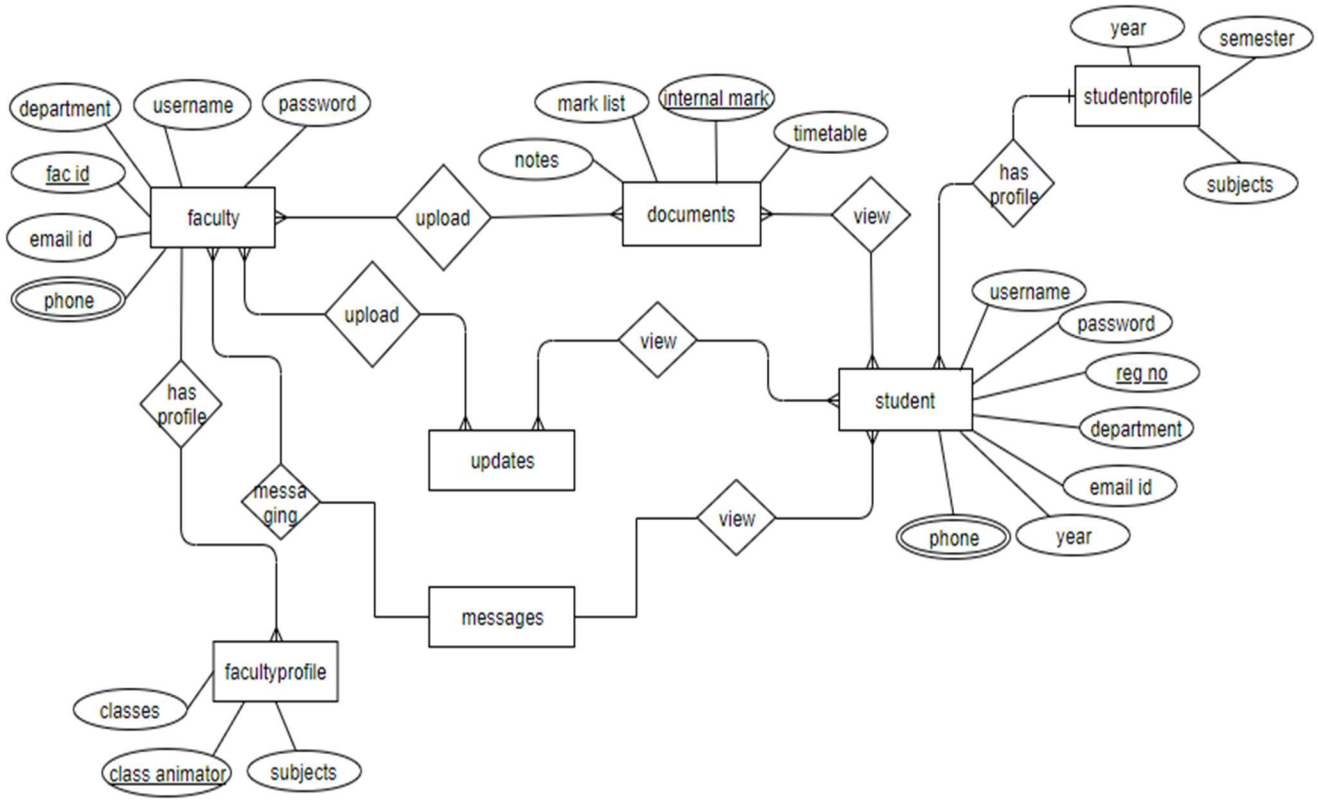




Cardinality views: Cardinality can be shown as look-across or same-side, depending on where the symbols are shown.

Cardinality constraints: The minimum or maximum numbers that apply to a relationship.

ER DIAGRAM


year
semester
studentprofile
subjects
has profile
department
username
password
fac id
email id
phone
faculty
upload
mark list
internal mark
timetable
notes
documents
view
upload
view
username
password
reg no
department
email id
phone
year
student
has profile
updates
messa ging
view
messages
facultyprofile
classes
class animator
subjects

5 PROJECT DESCRIPTION

5.1 MODULE DESCRIPTION

5.1.1 Uploading Documents

After the successful login of faculties, they were able to upload timetables, internal marks, notes, mark list, notifications on exams and events. These uploaded data's will send to the students those who were under the corresponding faculties. Students were not able to upload any kind of data.

5.1.2 Downloading documents

After the successful login of students, they were able to download the documents that is send by the faculties and view it.

5.1.3 Profile

After the login, both students and faculties can update their profile. In this, students and faculties will provide their department and other important details regarding them. This will help in sharing and accessing the data.

5.1.4 Messaging

In this, the faculties are able to send messages to the students. In this broadcast messaging only faculties are able to send messages to students.

5.2 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data in the database is safe and easily accessed.

The database design is two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS. In the second step, this Information Level Design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs

parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives:

- Data Integrity
- Data Independence

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this system for the following reasons:

- To structure the data so that here is no repetition of the data, this helps in saving space.
- To permit simple retrieval of the data in response to query and report request.
- To simplify the maintenance of the data through updates, insertions and deletions.
- To reduce the need to restrict or recognize data which new application requirement arise.

Relational Database Management System (RDBMS):

A relational model respects the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of collection of tables, each of which is assigned a unique name. A row in a table represents a set of related values.

5.2.1 Faculty

FIELD	DATATYPE	CONSTRAINTS
Fac_id	Varchar (50)	Primary key
Fac_name	Varchar (50)	Not null
Password	Varchar (50)	Not null
Email	Varchar (50)	Not null
Ph_no	Bigint	Not null
Department	Varchar (50)	Not null

5.2.2 Student

FIELD	DATATYPE	CONSTRAINTS
Reg_no	Varchar (50)	Primary key
stud_name	Varchar (50)	Not null
Password	Varchar (50)	Not null
Email	Varchar (50)	Not null
Ph_no	Bigint	Not null
Department	Varchar (50)	Not null
Year	Varchar (50)	Not null

5.2.3 Faculty_Profile

FIELD	DATATYPE	CONSTRAINTS
Class_animator	Varchar (50)	Primary key
Subjects	Varchar (50)	Not null
Classes	Varchar (50)	Not null

5.2.4 Student_Profile

FIELD	DATATYPE	CONSTRAINTS
Sem	Varchar (50)	Primary key
Subjects	Varchar (50)	Not null
Year	Varchar (50)	Foreign key

5.2.5 Messages

FIELD	DATATYPE	CONSTRAINTS
Msg_id	Varchar (50)	Primary key
Content	Varchar (50)	Not null

5.2.6 Documents

FIELD	DATATYPE	CONSTRAINTS
Internals	Varchar (50)	Primary key
Notes	Varchar (50)	Not null
Mark_list	Varchar (50)	Not null
Time_table	Varchar (50)	Not null

5.2.7 Updates

FIELD	DATATYPE	CONSTRAINTS
Event	Varchar (50)	Primary key
Exam	Varchar (50)	Not null

5.3 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing data entry. The activity of putting data into the computer for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The

design of input focuses on controlling the amount of input required,contolling errors,avoiding delays,avoiding extra steps and keeping the process simple.

Objectives

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that input acceptable and understood by the user.

Input design is used to enter all the values which are normally carried by the user and the admin. In this application the inputs are given by the user (faculty) itself. Forms are used for entering the values, which will be in the pre-determined formats for later analysis. Input design involves determining the record medias, method of input,speed of capture and entry to the system. The design decisions for input design specify how data are accepted for computer processing. Data can be entered directly or through some source files.

The input design is the process of converting the user-oriented inputs into the computer-based format. Errors entered by data entry operators can be controlled by input design. The data is fed into system using simple interactive forms. The forms have been supplied with messages so that user can enter data without facing difficulty. The data is been incorporated into the system. The goal for designing input data is to make entry as easy, logical and free from errors.

Points to be noted while designing input screen:

- Allow only information needed by user.
- Don’t over crowd the input screen.
- Keep the same style between screens.
- Ask for confirmation of critical data.
- Validate data as soon as possible on the input, thereby preventing storing the incorrect data into the system.
- The elements of the data that uniquely identifies the record being processed.

XML in Android

XML stands for Extensible Markup Language. XML is a markup language much like HTML used to describe data. XML tags are not predefined in XML. We must define our own tags. XML

as itself is well readable both by human and machine. Also, it is scalable and simple to develop. In Android we use XML for designing our layouts because XML is lightweight language so it doesn't make our layout heavy.

In this article we will go through the basic concepts of XML files used for different purpose in Android. This will help you in writing a UI code to design your desired user interface.

Different XML files used in Android:

In Android there are several XML files used for several different purposes.

1. **Layout XML Files:** Layout XML files are used to define the actual UI (User Interface) of our application. It holds all the elements (views) or the tools that we want to use in our application. Like the TextView's, Button's and other UI elements.
2. **Manifest XML File (manifest.xml):** This XML is used to define all the components of our application. It includes the names of our application packages, our activities,recievers,services and the permissions that our application needs. For example: - Suppose we need to use internet in our app then we need to define Internet permission in this file.
3. **Strings XML File (strings.xml):** This XML file is used to replace the Hard-Coded strings with a single string. We define all the strings in this XML file and then access them in our app (Activity or in Layout XML files) from this file. This file enhances the reusability of the code.
4. **Styles XML File (styles.xml):** This XML is used to define different styles and looks for the UI (User Interface) of application. We define our custom themes and styles in this file.
5. **Drawable XML File:** These are those XML files that are used to provide various graphics to the elements or views of application. When we need to create a custom UI we use drawable XML files. Suppose if we need to define a gradient color in the background of Button or any custom shape for a view then we create a Drawable XML file and set it in the background of view.
6. **Color XML File (colors.xml):** This file is used to define the color codes that we used in our app. We simply define the colors in this file and used them in our app from this file.

7. **Dimension XML File (dimens.xml):** This XML file is used to define the dimensions of the View's. Suppose we need a Button with 50dp (density pixel) height then we define the value 50dp in dimens.xml file and then use it in our app from this file.

5.4 OUTPUT DESIGN

One of the most important features of an information system for users is the output it produces. Output is information delivered to users through the information system. Users generally merit the system solely by its output. Hence the system analysts work closely with user through an interactive process, until the result is considered to be satisfactory.

Objective of Output Design:

- Design output to serve the intended purpose.
- Design output to fit the user.
- Deliver the appropriate book.
- Assure that output is where it is needed.
- Provide output on time.
- Choose the right output method.

Two phases of output design are

1. Output Definition
2. Output Specification

Output definition takes into account the type of output contents, it's frequently and its volume. The appropriate output media is determined for outputs. Once the output required from the proposed system is determined during the logical design stage itself. The physical design stage takes the outline of the output from the logical design and produces the output as specified during the logical design phase. The necessary reports are generated which provide the complete information required by the user.

User Interface

All User Interface elements in an Android app are built using View and ViewGroup objects.A View is an object that draws something on the screen that the user can interact with. A ViewGroup

is an object that holds other View (and ViewGroup) objects in order to define the layout of the layout of the interface.

Android provides a collection of both View and ViewGroup subclasses that offer you common input controls (such as buttons and text fields) and various layout models (such as linear or relative layout).

User Interface Layout

The User Interface for each component of your app is defined using a hierarchy of View and ViewGroup objects. Each view group is an invisible container that organizes child views, while the child views may be input controls or other widgets that draw some part of the UI. This hierarchy tree can be as simple or complex as you need it to be (but simplicity is best for performance).

To declare your layout, you can instantiate View objects in code and start building a tree, but the easiest and most effective way to define your layout is with an XML file. XML offers a human-readable structure for the layout, similar to HTML.

The name of an XML element for a view is respective to the Android class it represents. So a <TextView> element creates a TextView widget in your UI, and a <LinearLayout> element creates a LinearLayout viewgroup.

When you load a layout resource in your app, Android initializes each node of the layout into a runtime object you can use to define additional behaviors, query the object state, or modify the layout.

For a complete guide to creating a UI layout, see XML layout.

User Interface Components

You don't have to build all your UI View and ViewGroup objects. Android provides several app components that offer a standard UI layout for which you simply need to define the content. These UI components each have a unique set of APIs that are described in their respective documents, such as Adding the App Bar, Dialogs, and Status Notification.

6 SYSTEM TESTING

In a software development project, errors can be injected at any stage during the development phase. For each phase we have discussed, there are different methods and techniques that are available for eliminating errors. However, no technique is perfect, and it is expected that some of the errors of the earlier phase will manifest themselves in the code. Hence, the code developed during the coding activities is likely to have some requirements errors and design errors, in addition to introduce during the coding activity.

Testing is an important and critical stage in software development. Testing plays an important role in determining the quality and reliability of the application. With this process, several test cases are devised. A test case means a set of data that the system will process as the normal input. System testing consists of several key activities and steps for program testing.

6.1 WHITE BOX TESTING

White box is a testing technique that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

Advantages of white box testing:

- Forces test developer to reason carefully about implementation.
- Reveals errors in “hidden” code.
- Spots the Dead Code or other issues with respect to best programming practices.

Disadvantages of white box testing

- Expensive as one has to spend both time and money to perform white box testing.
- Every possibility that few lines of code are missed accidentally.
- In-depth knowledge about the programming language is necessary to perform white box testing.

6.2 BLACK BOX TESTING

Black box testing, also known as Behavioral testing, is a software testing method in which the internal structure design/ implementation if the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method to find errors in the following categories:

- Incorrect to missing functions.
- Interface errors.
- Errors in data structures or external database access.
- Behavior or performance errors.
- Initialization and termination errors.

6.3 UNIT TESTING

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

Advantages:

- Reduces defects in the newly developed features or reduces bugs when changing the existing functionality.
- Reduces cost of testing as defects are captured in very early phase.
- Improves design and allows better refactoring of code.
- Unit tests, when integrated with build gives the quality of the build as well.

6.4 INTEGRATION TESTING

Integration testing, also known as integration and testing (I&T), is software development process which program units are combined and tested as groups in multiple ways. In this context, a unit is defined as the smallest testable part of an application. Integration testing can expose problems with the interfaces among program components before trouble occurs in real-world

program execution. Integration testing is a component of Extreme Programing (XP), a pragmatic method of software development that takes a meticulous approach to building a product by means of continual testing and revision.

There are two major ways of carrying out an integration test, called the bottom-up method and the top-down method. Bottom-up integration testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds. In top-down integration testing, highest-level modules are tested first and progressively lower-level modules are tested after that. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic those it will encounter in customers' computers, systems and networks.

6.5 ACCEPTANCE TESTING

Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it has met the required criteria for delivery to end users.

There are various forms of acceptance testing:

- User acceptance testing
- Business acceptance testing
- Alpha testing
- Beta testing

6.6 TEST CASES

Test Case Number	Test Data	DB table name(s) Influenced	Activity Involved	Expected Result	System Result	Remarks
1	Registration of faculty with faculty details	Faculty	Activity_fregister	Faculty should be registered to the application	Registered to the application	Successfully registered
2	Registration of student with student details	Student	Activity_sregister	Student should be registered to the application	Registered to the application	Successfully registered
3	Faculty Username & password	Faculty	Activity_faculty	Faculty should be login to the home page	Login to home page	Successfully login
4	Student Username & password	Student	Activity_student	Student should be login to the home page	Login to home page	Successfully login
5	Uploading of documents/ updates by faculty	Documents Updates	Activity_uploads	Faculty should be able to upload the data	Able to upload the data	Successfully uploaded

6	Updating the profile of faculty	Faculty_Profile	Activity_profilefac	Details should be updated	Details updated	Successfully updated
7	Sending messages to the students	Messages	Activity_fm	Should be send messages	Send messages	Successfully send
9	Updating the profile of student	Student_Profile	Activity_profile	Details should be updated	Details updated	Successfully updated

7 SYSTEM IMPLEMENTATION

7.1 MAINTENANCE

This phase occurs as a result of deploying the whole system for the organization. They will perform the beta testing at the end users and inform the developers about any further modification to the application .They records all the problems that are encountered during the beta testing and reports these to the developer at regular intervals. Any problem encountered during the use of the image editing application will be recorded and will be reported at regular intervals. Maintenance has not yet been performed and is a part of future plans.

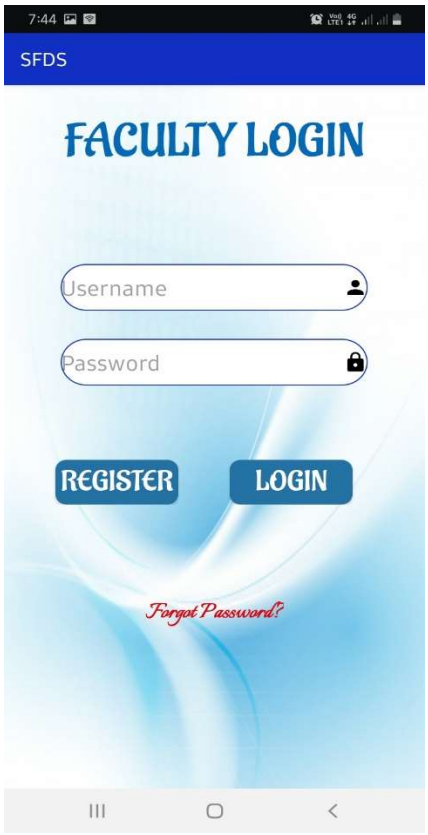
Even with the best quality assurance activities, it is likely that the organization will uncover defects in the software. Corrective maintenance changes the software to correct the defects. Corrective Maintenance activity may consist of repair, restoration or replacement of equipment. This activity will be the result of a regular inspection, which identifies the failure in time for corrective maintenance to be planned and scheduled, then performed during a routine maintenance shutdown.

8 SCREENSHOTS

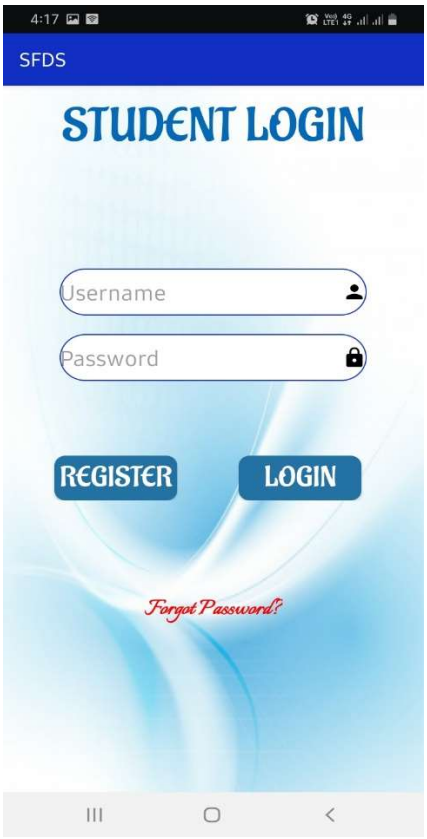
8.1 SPLASH SCREEN

8.2 MEMBER LOGIN

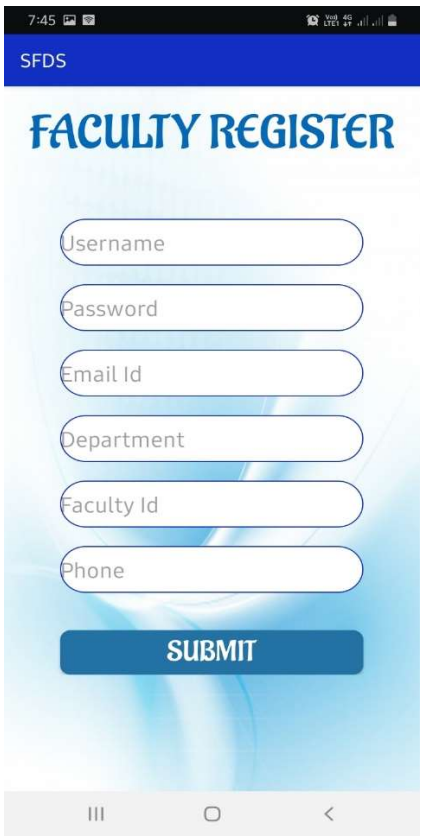
8.3 FACULTY LOGIN



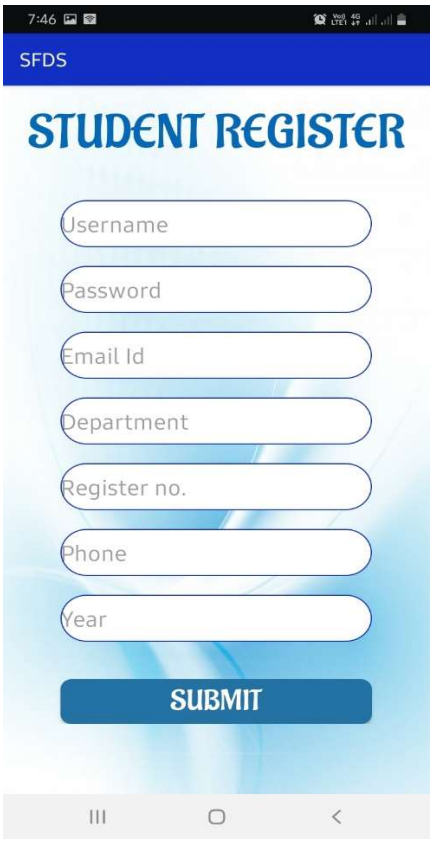
8.4 STUDENT LOGIN



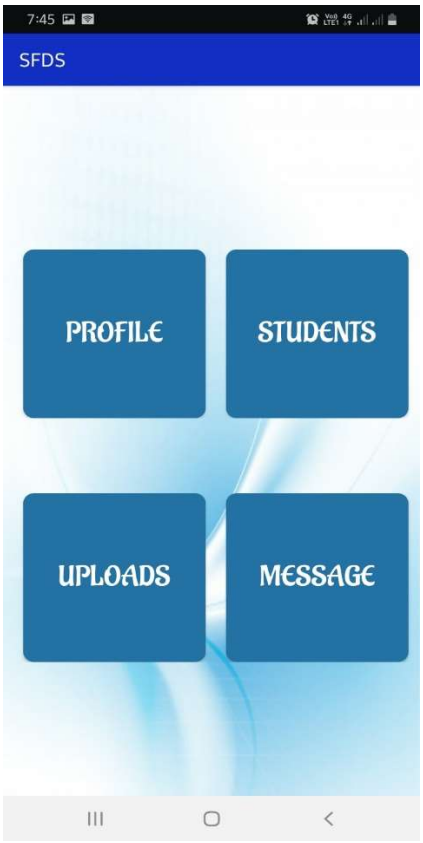
8.5 FACULTY REGISTER



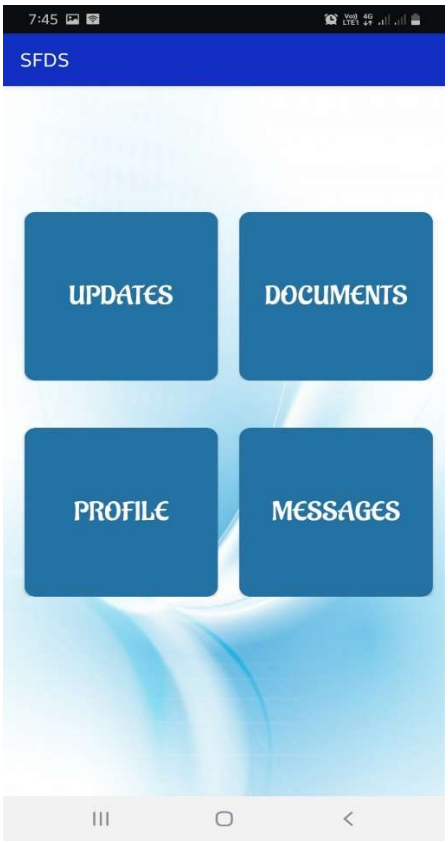
8.6 STUDENT REGISTER



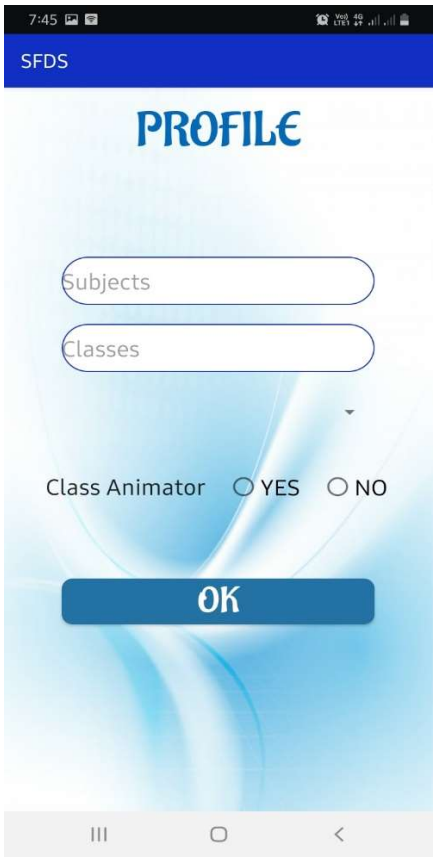
8.7 FACULTY HOME



8.8 STUDENT HOME



8.9 FACULTY PROFILE



8.10 UPLOADING

8.11 UPDATES

8.12 DOCUMENTS

8.13 STUDENT PROFILE

9 CONCLUSION

9.1 FUTURE ENHANCEMENT

There is a big demand for different types of software which is because IT has become the main part of our world. The document sharing application "SFDS – Student Faculty Document Sharing System" provides an easy way to share the important documents to students. The future implementation of this project involves the integration of IOT concept to each and every staff and students to interact with each other in real time and do the data analysis part for giving them the right kind of tool which will enhance the overall performance of the department.

9.2 CONCLUSION

SFDS (Student Faculty Document Sharing System) is an Android application that is mainly developed for two users: Students and Faculties. It gives a platform for the users to get involved in this application and get the required results. This system allows department faculty to share important data as well as notifications of exam with the students. The system will allows the faculty to upload documents and students may download required documents through their android device. Here students only access and download data. Faculty may access and upload/edit documents. There is Broadcast messaging corner for the student and faculty to interact. Only the registered students and faculty can perform the desired operations.

While paper records are the traditional way for the sharing of important documents to students but it is difficult to manage and track. The physical stress required to retrieve, alter, and re-file the paper records are all non-value added activities. This system provides a simple interface to maintain the all the records\documents without any difficult.

REFERENCES

[1] Wei-Meng Lee, Beginning Android 4 Application Development, Wiley publications, 2012

[2] Roger. S Pressman, Software Engineering- A Practitioners Approach, McGraw Hill International Edition, 7th Edition, 2010

[3] Tutorials on Android Programming. Retrieved from *http://developer.android .com/training/*

[4] Tutorials on Android Programming. Retrieved from *https://tutorialspoint.com* /android/

[5] Tutorials on Android Programming. Retrieved from *https://abhiandroid.com*