ST. JOSEPH'S COLLEGE
(AUTONOMOUS)

COMPUTER SCIENCE
Major Project for
(APD Exports)

Submitted By:

Renick Andrews A. (16BCA41036)

ST. JOSEPH’S COLLEGE (AUTONOMOUS)

Bangalore-560027

Certificate

This is to certify that **Renick Andrews A.** bearing register number **16BCA41036** of 3rd BCA has successfully completed the project **“ERP-Enterprise Resource Planning”** for **“APD Exports”** for the academic year 2018-2019.

Head of the department **Lab In-charge**

Examiners:

1. ______________________

2. ______________________

A.P.D. EXPORTS

GOVT. RECOGNISED EXPORT HOUSE

69/3, Byraveshwara Indl. Area, Hegganahalli, Behind Shushruthi Bank, BENGALURU- 560 091.
Tel : 91-080-28363937

Date : 15-03-2019

CERTIFICATE

This is to certify that **Renick Andrews A. (16BCA41036),** of **St. Joseph's College** has successfully completed the project - " **ERP-Enterprise Resource Planning** " for APD Exports during the period November 2018 - March 2019.

The source code and content of the project should not be shared or disclosed to anyone as per company policy .

We wish you all the best in your future endeavours.

Yours faithfully,

For APD Exports
G.M.

ACKNOWLEDGEMENT

We would like to thank God for giving us the strength, knowledge and ability to undertake this project and also for all his grace blessings.

We would like to thank APD Exports for giving us the opportunity to work and create an application for them. It was truly an enlightening learning experience for us.

We would never have been able to finish our project without the guidance of the following people, who have shown their constant support. We express gratitude to St. Joseph's college, Bangalore for giving us the opportunity.

We are also grateful to Ms. Banu and Ms. Nithya, our lab supervisors for providing invaluable insight, support, guidance and suggestions which helped us complete the project.

We are also thankful to all the people who have been a part of this, especially our parents. We feel deeply indebted to all the people who helped us to finish this project.

CONTENTS

ABSTRACT 1

INTRODUCTION 2

STUDY OF EXISITNG SYSTEM 3

STUDY OF PROPOSED SYSTEM 4

SYSTEM REQUIREMENTS HARDWARE 5

SYSTEM REQUIRMENTS SOFTWARE 5

SYSTEM CONFIGURATION 5

ANALYSIS 9

DESIGN 10

DATA FLOW DIAGRAM 10

0-Level data flow diagram 11

1-Level data flow diagram(Admin side) 12

1-Level data flow diagram(User side) 13

Use case diagram 14

CODING 16

TESTING STRATIGES 23

Functional testing 23

Non-Functional testing 25

FUTURE ENHANCEMENT 26

CONCLUSION 27

BIBILOGRAPHY 28

ABSTRACT

The Enterprise Resource Planning (ERP) application is developed for managing APD Exports. Enterprise Resource Planning, or ERP, is a large-scale software program designed for modern businesses, both large and small. A simple definition is that ERP systems aid the flow of internal business processes and allow for communication between a business's departments and its internal functions and data. If successfully implemented, ERP systems provide many benefits to the organizations that adopt them. Given that these solutions provide an integrated solution to the needs of an organization, these systems are in high demand by all organizations.

This project is made in Qt (using C++), a free and open source widget toolkit for creating graphical user interfaces as well as cross-platform applications that run on various software and hardware platforms such as Linux, Windows, Android etc. The back-end software used is SQLite, which unlike SQL is embedded into the end-program.

The main functions being automated in the initial version are employee database, attendance, design management and daily productivity.
The application performs all these tasks efficiently and saves time and manpower.

Keeping the end user in our minds, we have developed an application which provides a rich graphical UI to the users for better and efficient access to the system.

INTRODUCTION

Problem Definition:

A lot of the business function in the company are currently being done manually, which is inefficient, time consuming and also leaves room for errors to creep in. The current method involves the use of large amount of paper which makes back-tracking and audits difficult and is also not environment friendly.

Scope of the project:

Our project mainly focuses on Inventory and Material Management, Quality, Research and Development, Production, Quality, Finance, Human Resource, and service functions.

A database with all these details in maintained and can be updated whenever required.

The system can perform the following:

- Provide a secure LOGIN access
- Provide a user-friendly, graphical UI to the users.
- Access the database to display details such as attendance, designs, employees, etc.
- Update the database and display data dynamically
- Interlink each module with the other so that necessary information could be accessed for making right business decisions and for efficiently accomplishing all business functions.
- Digitize, integrate, sufficiently automate, and streamline the function of all the departments of the business.

STUDY OF EXISTING SYSTEM

In the existing system the tasks are performed manually and authenticity and security cannot be guaranteed. Certain functions like attendance and employee database are done digitally. But these systems operate independently and there is no communication between them. The other tasks like productivity and design management are done manually.

DISADVANTAGES:

- Time Consuming
- Possibility of errors occurring
- Inefficient
- Data can be modified by unauthorized personnel

STUDY OF PROPOSED SYSTEM

Our project aims to introduce a platform that makes it easy to keep a track of the employee data, attendance, production, designs, productivity etc. The system is built around a user-friendly GUI (Graphical User Interface). The proposed system possesses the following advantages over the existing system:

- Strengthen business performance and efficiency
- Enable effective reporting
- Save time
- Lower administrative cost
- Increase productivity
- Improve decision making
- Meet your customer’s needs on time
- Optimization of business processes.

SYSTEM CONFIGURATION

MINIMUM HARDWARE CONFIGURATION:

PROCESSOR	-	INTEL(R) I3 CPU@ 2.50GHz
CLOCK SPEED	-	2.50GHZ
RAM	-	4GB
MEMORY REQUIRED	-	500MB

SOFTWARE SPECIFICATION:

OPERATING SYSTEM	-	Windows 7 or higher

SYSTEM REQUIREMENTS

SOFTWARE REQUIREMENTS

Front-end: Qt (C++)

Back-end: SQLite

HARDWARE REQUIREMENTS

Approximate minimal hardware requirements for running Boot to Qt are:

256 MB of RAM

500 MHz CPU, 1 GHz preferred for 60-FPS velvet-smooth UI

OpenGL ES 2.0 support

FRONT END AND BACK END SOFTWARE DESCRIPTION

Overview of Front-End Software

Qt (pronounced "cute") is a free and open source widget toolkit for creating graphical user interfaces as well as cross platform applications that run on various software and hardware platforms such as Linux, Windows, macOS, Android or embedded systems with little or no change in the underlying codebase while still being a native application with native capabilities and speed. Qt is currently being developed by The Qt company, a publicly listed company, and the Qt project under open source governance, involving individual developers and organizations working to advance Qt. Qt is available under both commercial licenses and open source GPL 2.0, GPL 3.0, and LGPL 3.0 licenses.

Qt is used for developing graphical user interfaces (GUIs) and multi-platform applications that run on all major desktop platforms and most mobile or embedded platforms. Most GUI programs created with Qt have a native-looking interface, in which case Qt is classified as a widget toolkit. Also, non-GUI programs can be developed, such as command line tools and consoles for servers. An example of such a non-GUI program using Qt is the Cutelyst web framework.

Qt supports various compilers, including the GCC C++ compiler and the Visual Studio suite and has extensive internationalization support. Qt also provides Qt Quick, that includes a declarative scripting language called QML that allows using JavaScript to provide the logic. With Qt Quick, rapid application development for mobile devices became possible, while logic can still be written with native code as well to achieve the best possible performance.

Other features include SQL database access, XML parsing, JSON parsing, thread management and network support.

Features:

- It's cross-platform and it uses the system's resources to draw windows, controls, etc. so your application will get a native look (e.g. on a Mac your app window will be lacking the menu bar and the menu bar will appear on the system's menu bar as it is the standard behavior on the Mac platform).
- Writing in C++ gives you great control, the possibility to work with fantastic libraries like the STL, Boost, etc.; and your code is compiled to native binaries that will run at full speed without the need for a virtual machine.
- Qt is open source and is developed by the Qt Group (formerly Toltec) at Nokia so you have a very large enterprise maintaining it with the support from the community and ensuring its evolution.
- It has the best GUI designer I've ever seen (I have worked with .NET before using Windows Forms.)
- It now also has a very capable IDE in Qt Creator which again works on all platforms and hence gives you same development environment where ever you go.
- The latest 4.7 is coming with a new way of developing apps for devices using QML + JavaScript for your fluid multi touch gesture enabled interface talking directly to the business logic done in C++ Qt classes.
- The same is true for Qt Web kit based on Web kit toolkit where you can connect your JavaScript code to your core C++ classes done in Qt.

Overview of Back-End

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

Features:

- SQLite does not require a separate server process or system to operate (serverless).
- SQLite comes with zero-configuration, which means no setup or administration needed.
- A complete SQLite database is stored in a single cross-platform disk file.
- SQLite is very small and lightweight, less than 400KB fully configured or less than 250KB with optional features omitted.
- SQLite is self-contained, which means no external dependencies.
- SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.
- SQLite supports most of the query language features found in SQL92 (SQL2) standard.
- SQLite is written in ANSI-C and provides simple and easy-to-use API.
- SQLite is available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT).

ANALYSIS

Feasibility Study

A feasibility study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technological, legal and scheduling factors. A feasibility study tests the viability of an idea, a project or even a new business. The goal of a feasibility study is to place emphasis on potential problems that could occur if a Project is pursued and determines if, after all significant factors are considered, the project should be pursued.

Components of Feasibility study:

- **Economic Feasibility**

Analysis of the projects revealed that it would have a positive economic impact for the company by reducing material costs and also increasing productivity.

- **Technical feasibility**

Because the application is being developed on the Windows platform it can run on their existing systems which have a range of different hardware and software configurations.

- **Operational feasibility**

The new application will be highly expendable and will perform all the manual processes it is replacing with a higher level of accuracy and efficiency with will save both human and financial resources.

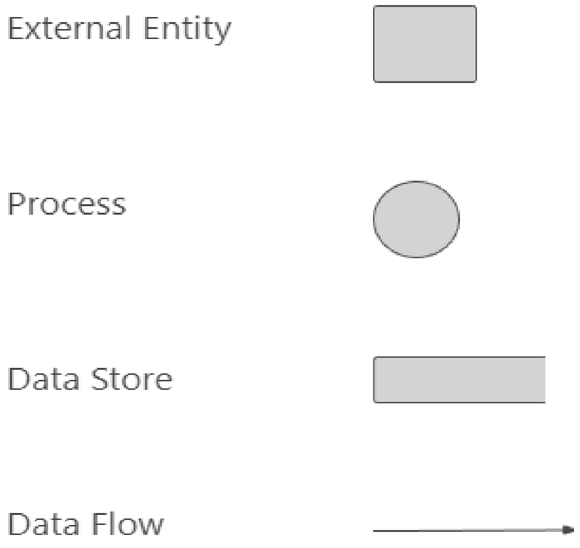
- **Schedule Feasibility**

Time duration of this project requires 61 days covering 12 days for Requiring Gathering, 15 days for Design, 20 days for Implementation (Front end coding and backend coding), 3 days for testing and 6 days for documentation which fits well within both the company’s as well as our timeframe.

DESIGN

Data Flow Diagram

Data flow symbol and their meanings: -



Entity: A source of data or a destination for data.

Source/Sink: Represented by rectangles in the diagram. Sources and Sinks are external entities which are sources or destinations of data, respectively.

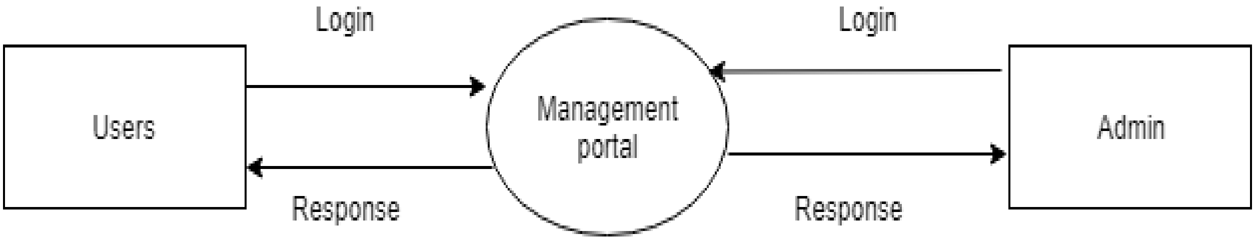
Process: Represented by circles in the diagram. Processes are responsible for manipulating the data. They take data as input and output an altered version of the data.

Data Store: Represented by a segmented rectangle with an open end on the right. Data Stores are both electronic and physical locations of data. Examples include databases, directories, files, and even filing cabinets and stacks of paper.

Data Flow: Represented by a unidirectional arrow. Data Flows show how data is moved through the System. Data Flows are labeled with a description of the data that is being passed through it.

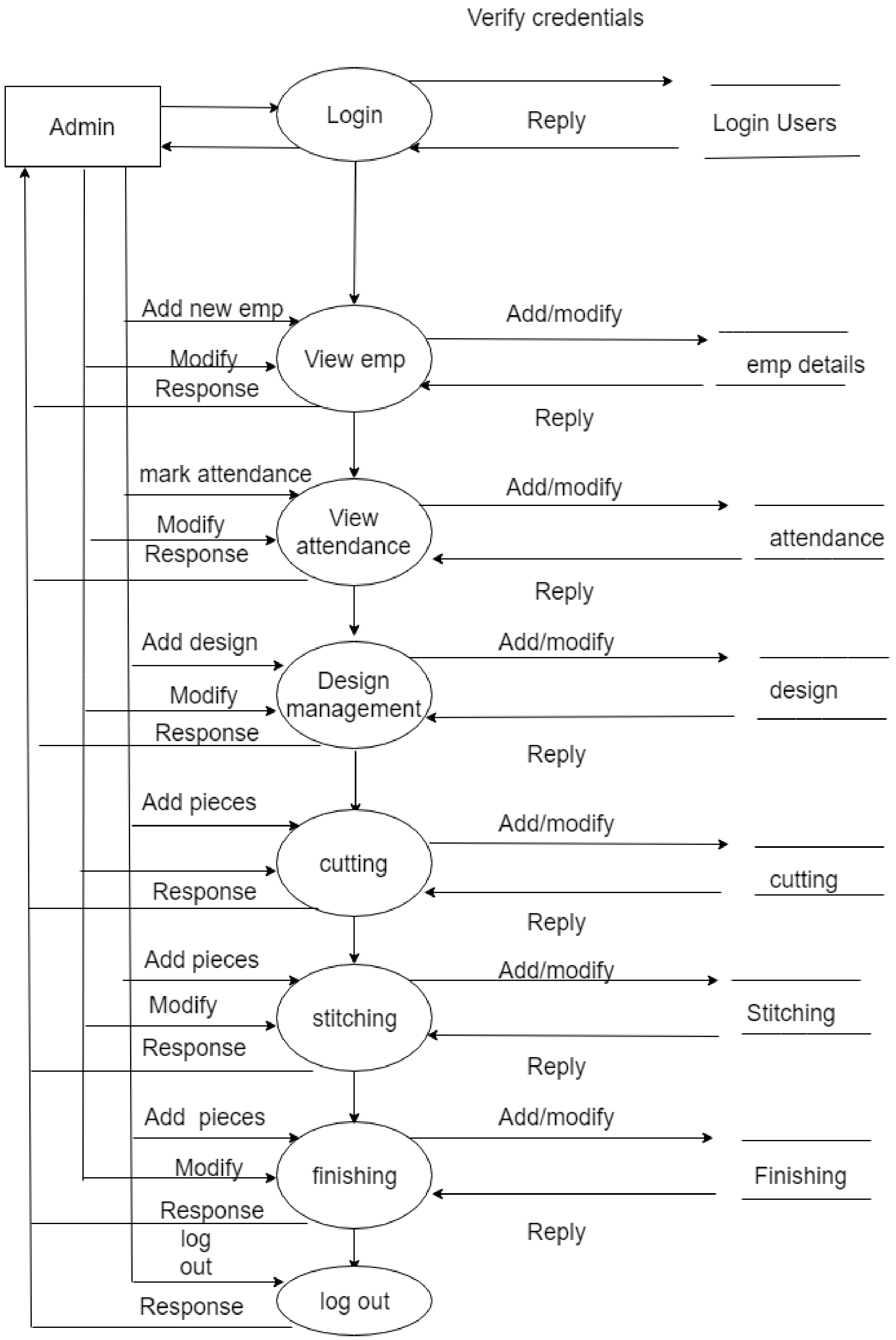
Level-0 DFD

A level-0 DFD is the most basic form of DFD. It aims to show how the entire system works at a glance. There is only one process in the system and all the data flows either into or out of this process.

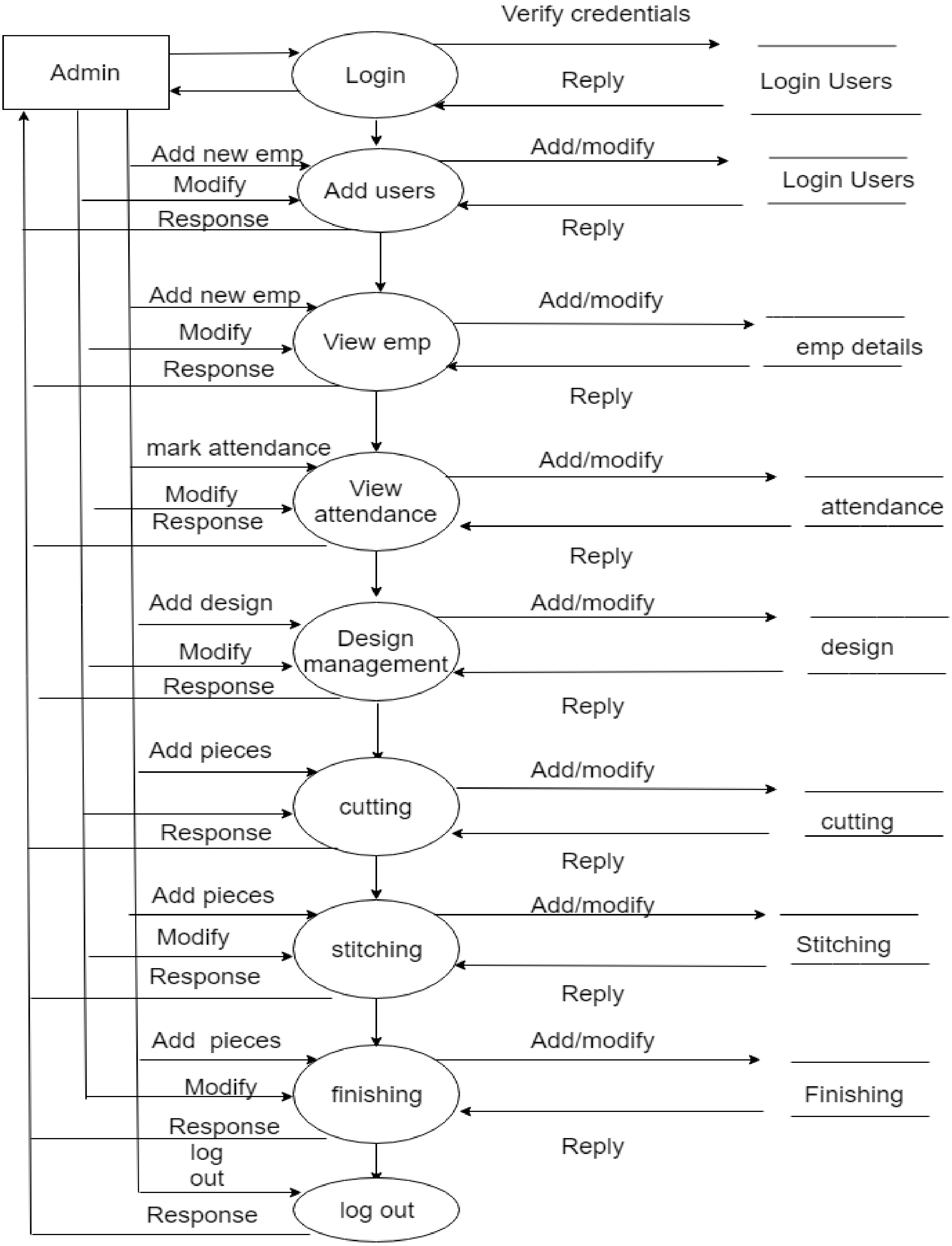


Level-1 DFD

User side:



Admin Side:


Admin
Login
Verify credentials
Reply
Login Users
Add users
Add new emp
Modify
Response
Add/modify
Login Users
Reply
View emp
Add new emp
Modify
Response
Add/modify
emp details
Reply
View attendance
mark attendance
Modify
Response
Add/modify
attendance
Reply
Design management
Add design
Modify
Response
Add/modify
design
Reply
cutting
Add pieces
Response
Add/modify
cutting
Reply
stitching
Add pieces
Modify
Response
Add/modify
Stitching
Reply
finishing
Add pieces
Modify
Response
Add/modify
Finishing
Reply
log out
log out
Response

Use Case Diagram

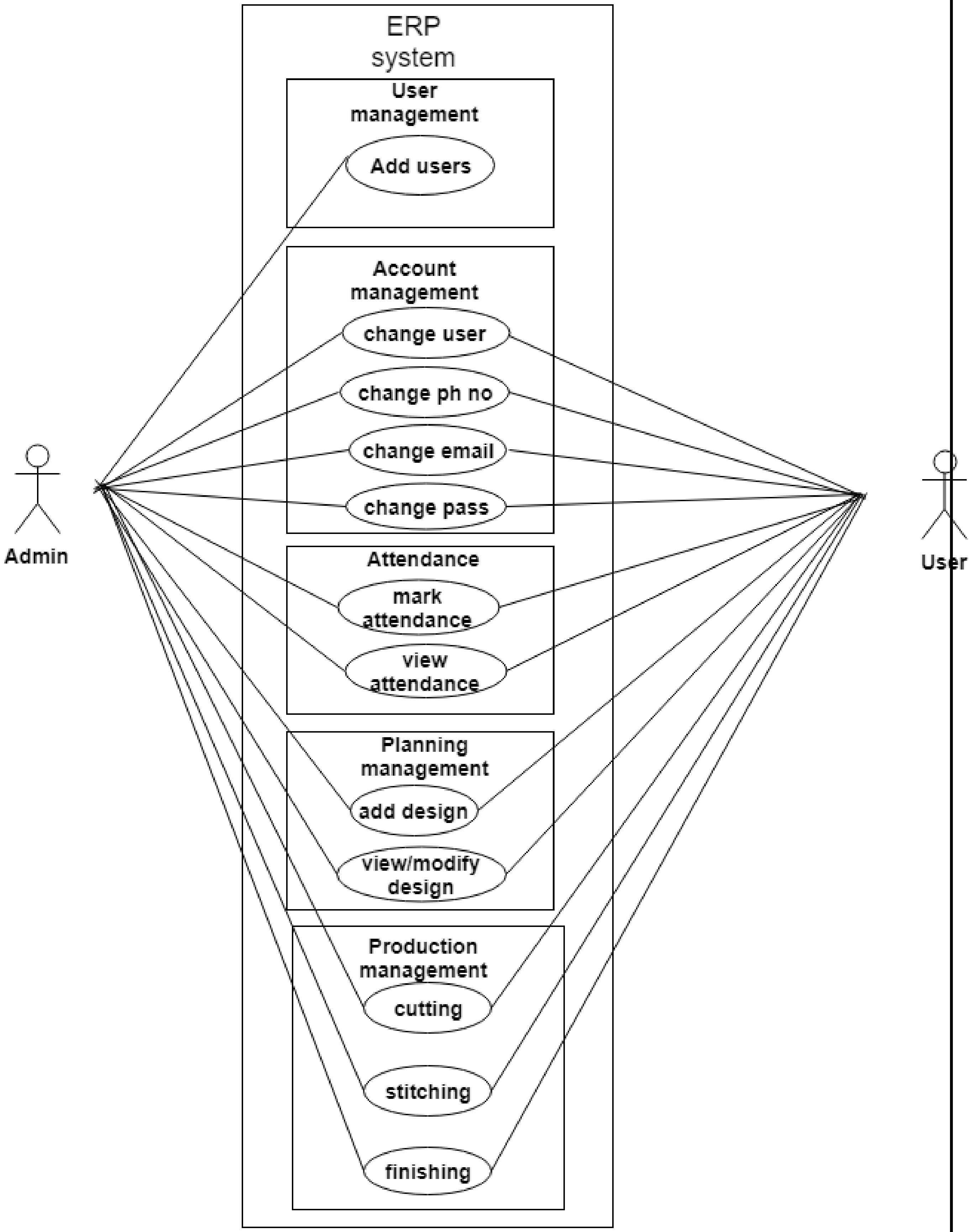
A use case diagram is a graphic depiction of the interactions among the elements of a system.

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

A use case diagram looks something like a flowchart. Intuitive symbols represent the system elements.


ERP system
User management
Add users
Account management
change user
change ph no
change email
change pass
Attendance
mark attendance
view attendance
Planning management
add design
view/modify design
Production management
cutting
stitching
finishing
Admin
User

CODING

Sample Code

Login code:

```
void Login::on_pushButton_clicked()
{
QSqlDatabase localdb= QSqlDatabase::database();
if (!localdb.open())
{
   qDebug() << "Error: connection with database fail";
}
else
{
   qDebug() << "Database: connection ok";
}
   QString username=ui->lineEdit->text();
   QString password=ui->lineEdit_2->text();
   QSqlQuery pcheck;
   if(pcheck.exec("select * from login_users where username ='"+username +"' AND
password = '" +password+"'"))
    {
      int count=0;
      while(pcheck.next())
      {
         count++;

      }
      if(count==1)
      {
         dashb=new Dash;
         dashb->show();
         this->close();
      }
      else {
         QMessageBox::critical(
           this,
           tr("ERROR"),
           tr("Invalid Username/Password") );
      }
 }
}
```

Add employee code:

```
void addemployee::on_pushButton_clicked()
{
    QSqlDatabase m_db;
    m_db = QSqlDatabase::addDatabase("QSQLITE");
    m_db.setDatabaseName("C:/Users/Acer/Desktop/SQLiteStudio/DB");
if (!m_db.open())
{
    qDebug() << "Error: connection with database fail";
}
else
{
    qDebug() << "Database: connection ok";
}
QString ename = ui->lineEdit->text();
QDate dob = ui->dateEdit->date();
QString eid = ui->lineEdit_2->text();
QString phno = ui->lineEdit_4->text();
QString salary = ui->lineEdit_5->text();
QString dep;
if(ui->radioButton->isChecked())
    dep="Cutting";
else if(ui->radioButton_2->isChecked())
    dep="Finishing";
else if(ui->radioButton_3->isChecked())
    dep="Stiching";

int val=0, phv=0;

QRegularExpression regex("\\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\\.[A-Z]{2,4}\\b");

if(!regex.match(eid).hasMatch())
{
 val++;
     }
else {
val++;
QMessageBox::critical(
  this,
  tr("ERROR"),
  tr("Invalid E-mail ID") );


}
for(int i=0;i<phno.size();i++)
{
    if(phno[i].isDigit())

        phv=0;
         else
        phv=1;
    }
if(phv==1)
{
    QMessageBox::critical(
      this,
      tr("ERROR"),
      tr("Invalid Mobile Number") );
}
```

```
else { if(phno>999999999)
    val++;
}
QSqlQuery in;
in.prepare("INSERT INTO attendance (Name,dept)" "VALUES(:ename,:dep)");
in.bindValue(":ename",ename);
in.bindValue(":dep",dep);
in.exec();
QSqlQuery ins;
ins.prepare("INSERT INTO
emp_details(Name,DOB,emailID,Mobile_no,salary,dep)"
"VALUES(:ename,:dob,:eid,:phno,:salary,:dep)");
ins.bindValue(":ename",ename);
ins.bindValue(":dob",dob);
           ins.bindValue(":eid",eid);
           ins.bindValue(":phno",phno);
           ins.bindValue(":salary",salary);
           ins.bindValue(":dep",dep);

   if(ins.exec())
{
    QMessageBox::information(
      this,
      tr("Success"),
      tr("Record Added!!") );
    foreach(QLineEdit *widget, this->findChildren<QLineEdit*>()) {
        widget->clear();
    }
    QSqlQuery d1,d2,d3,u1,u2,u3;
    d1.prepare("DROP TABLE stiching");
        d2.prepare("DROP TABLE cutting");
            d3.prepare("DROP TABLE finishing");
    d1.exec();
    d2.exec();
    d3.exec();
    u1.prepare("CREATE TABLE stiching AS SELECT * FROM emp_details WHERE
dep is 'Stiching'");

    u2.prepare("CREATE TABLE cutting AS SELECT * FROM emp_details WHERE dep
is 'Cutting'");

    u3.prepare("CREATE TABLE finishing AS SELECT * FROM emp_details WHERE
dep is 'Finishing'");

    u1.exec();
    u2.exec();
   u3.exec();
}
else {
    QMessageBox::critical(
      this,
      tr("ERROR"),
      tr("Could not save record") );
```

FORM DESIGN

Form is a tool with the message. It is a physical carrier of information.

We learnt that it is data that provides the basis of information to systems.

No systems exist without data.

Requirements of form design:

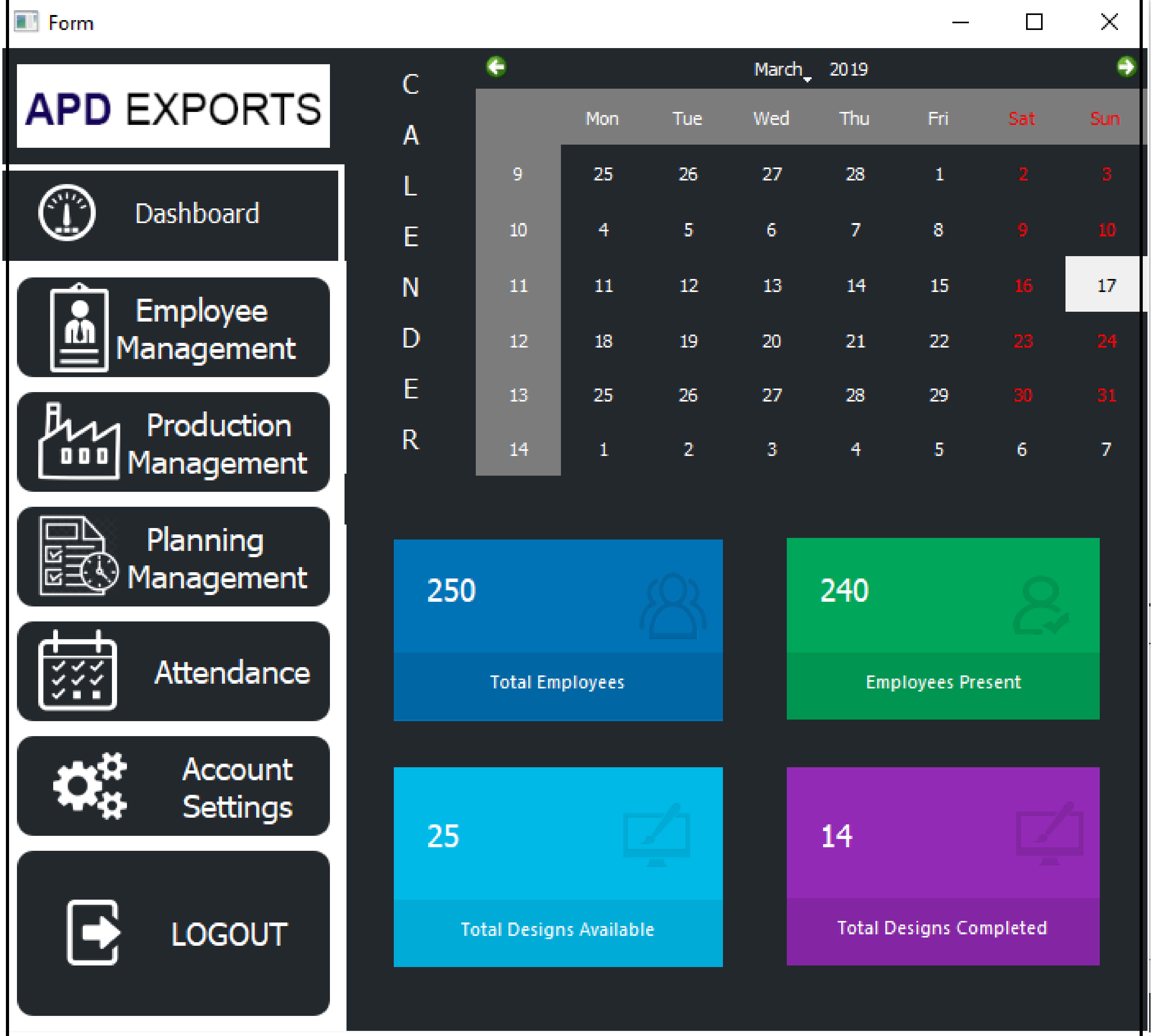
Form design follows analysis evaluating present documents and creating new forms. Since the purpose of the form is to communicate effectively through form design. There are several major requirements.

- Identification number
- Maximum readability and use
- Order of items
- Erase of data entry
- Size and arrangement
- Use of instructions
- Efficiency considerations

Sample Forms:

Intro

Dashboard



	Mon	Tue	Wed	Thu	Fri	Sat	Sun
9	25	26	27	28	1	2	3
10	4	5	6	7	8	9	10
11	11	12	13	14	15	16	17
12	18	19	20	21	22	23	24
13	25	26	27	28	29	30	31
14	1	2	3	4	5	6	7

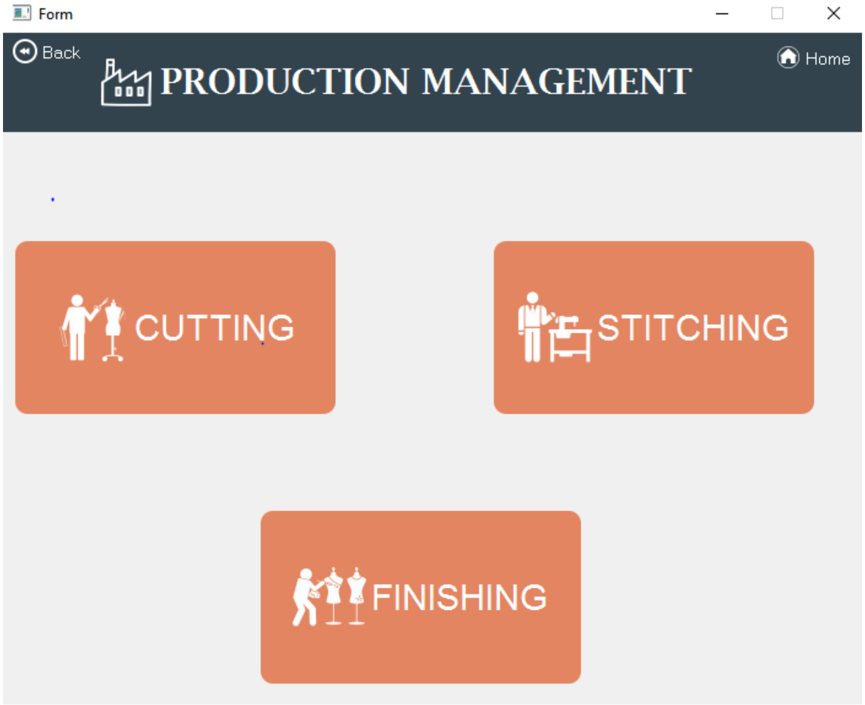
250 Total Employees

240 Employees Present

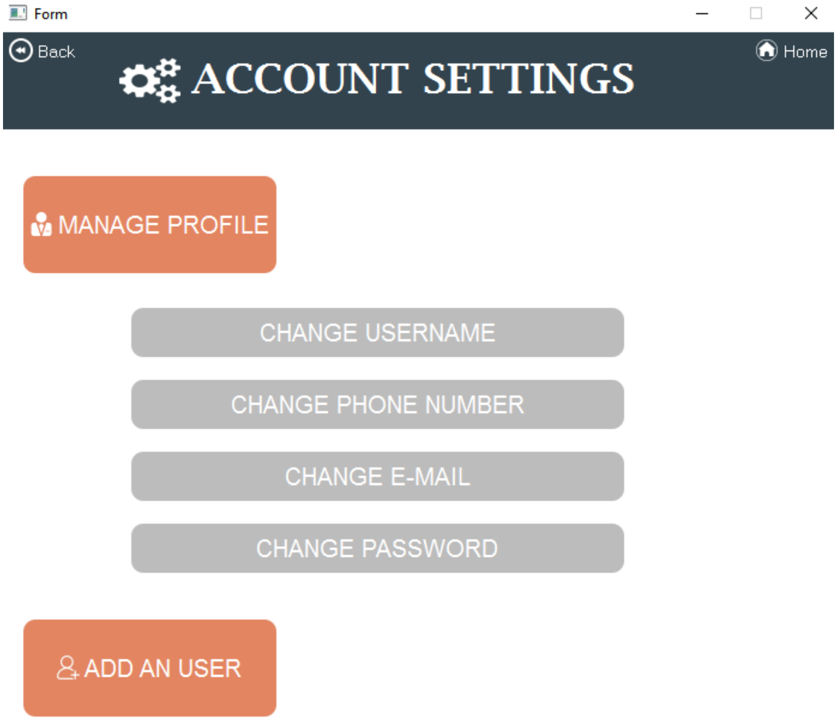
25 Total Designs Available

14 Total Designs Completed

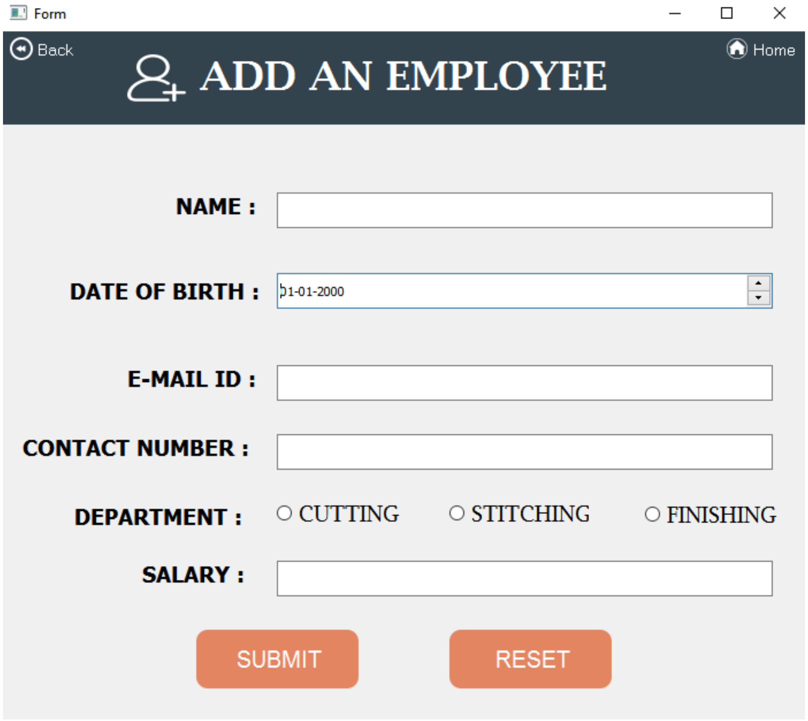
Production Management



Account Settings



Add Employee



ADD AN EMPLOYEE

NAME :

DATE OF BIRTH : 01-01-2000

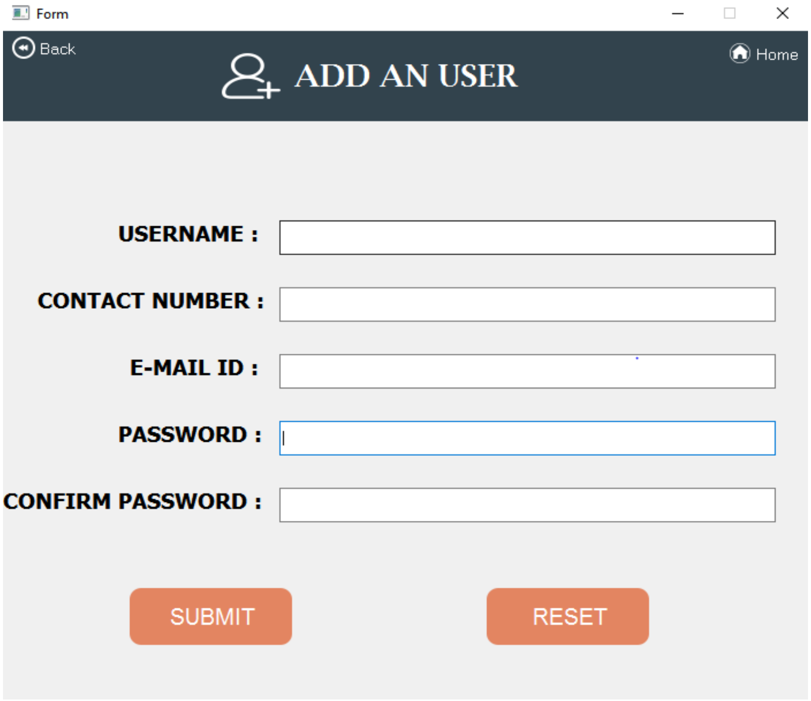
E-MAIL ID :

CONTACT NUMBER :

DEPARTMENT : ○ CUTTING ○ STITCHING ○ FINISHING

SALARY :

Add User



ADD AN USER

USERNAME :

CONTACT NUMBER :

E-MAIL ID :

PASSWORD :

CONFIRM PASSWORD :

TESTING

Testing is a vital part of software development, and it is important to start it as early as possible, and to make testing a part of the process of deciding requirements.

FUNCTIONAL TESTING

This type of testing ignores the internal parts and focus on the output is as per requirement or not. The functional testing strategies utilized in the project are:

Black box testing – Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It is sometimes referred to as specification-based testing.

Unit testing – Testing of individual software components or modules. Typically, it is done by the programmer and not by testers or final users, as it requires detailed knowledge of the internal program design and code.

Incremental integration testing – Bottom up approach for testing i.e. continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately. It was done by the programmer.

System testing – System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements. System testing takes, as its input, all of the integrated components that have passed integration testing.

Acceptance testing –The testing is done to verify if system meets the customer specific requirements. User or customers do this testing to determine whether to accept application.

TEST CASES

Test case no	Test case name	Test case description	Input data	Expected result	Status
1	Log in to the application	Attempting to log in to the system with certain empty fields.	If there is field that was left empty	There was an error message: Invalid Username /password	Fail
2		Sign in with wrong details	Ex: by entering wrong username or password	There was an error message: Invalid Username /password	Fail
3	Add an employee	Try registering using duplicate data	The data entered already exists.	Error message: the email address or the name already exists.	Fail
4		Registration done with proper details	First name, last name, mob no etc	Message generated: Record added successfully.	Success
5	Change password	Trying to change password with valid data	Valid username and password	Message: Password updated.	Success
6		Trying to change password with invalid data	Invalid username or password	Error message: Invalid username/password	Fail
7	Change username	Trying to change username with valid data	Valid username and password	Message: Username updated.	Success

8		Trying to change username with invalid data	Invalid username or password	Error message: Invalid username/password	Fail
9	Change phone number	Trying to change number with valid data	Valid username and password	Message: Number updated.	Success
10		Trying to change number with invalid data	Invalid username or password	Error message: Invalid username/password	Fail
11	Change email	Trying to change email with valid data	Valid username and password	Message: Email updated.	Success
12		Trying to change email with invalid data	Invalid username or password	Error message: Invalid username/password	Fail
13	Add a user	Try registering using duplicate data	The data entered already exists.	Error message: the email address or the name already exists.	Fail
14		Registration done with proper details	First name, last name, mob no etc	Message generated: Record added successfully.	Success

NON-FUNCTIONAL TESTING

Usability testing – User-friendliness check. Application flow is tested, can new user understand the application easily, Proper help documented whenever user stuck at any point. Basically, system navigation is checked in this testing.

FUTURE ENHANCEMENT

As every project goes, this project also has future improvements.
One of the main emphasis is to make it available on mobile platforms on the go as an application.
The future models will be designed on all the flaws and drawbacks that are experienced over the usage of the proposed system.

The following are the future enhancements that would be worked upon:

- Edit each design internally, meaning, the different parts of the clothing can be modified that is present in one single design.
- Integrate a biometric system with the ERP software
- Make the ERP system cloud based.
- Add new core business functions
- Newer accounting models (e.g. blockchain-based), and newer underlying data models (e.g. non-relational)
- More embedded business intelligence, analytic and data management features built into ERP
- Intelligence Reporting – A reporting solution that is required to access and use transaction data.
- Social platform –Add few platforms such as Facebook, LinkedIn, Skype, twitter etc. that will be operate through your ERP system.

CONCLUSION

This project work is done to provide APD exports with a working ERP system. All the activities provide a feeling like an easy walk over to the user who is interfacing with the system. Every effort has been made to represent the system in more user-friendly manner. The end-user will have no difficulties in understanding any aspects of our system.

All the disadvantages of the existing system (MANUAL) have been overcome using the present software system of "Enterprise Resource Planning (ERP) "which had been successfully implemented at the user's end. A trail run of the system has been made and is giving good results.

ERP systems can affect nearly every aspect of organizational performance and functioning, and measures of ERP system must reflect this fact.

Most of the companies were looking towards IT as a necessary expenditure: almost all including APD Exports, felt that IT was necessary to leverage their existing capabilities in a competitive world.

The Employee and customer satisfaction have increased in the past after this ERP package was implemented with increase in performance in productivity and market coverage.

This, this project would enhance the effectiveness and efficiency of the organizational functions in every aspect of its retail sector.

BIBLIOGRAPHY

Craig Stedman, “Flash! ERP works if you're careful”, Computerworld, 12/13/99

Charles Trepper, “ERP Project Management is Key to A Successful Implementation”,

http://itmanagement.earthweb.com/entdev/article/0,,11980_614681,00.html

LINKS:

http://www.webopedia.com/TERM/E/ERP.html

https://www.youtube.com/watch?v=EkjaiDsiM-Q

https://www.youtube.com/watch?v=5JVLO8yBMXA

https://wiki.qt.io/Basic_Qt_Programming_Tutorial

http://www.sqlitetutorial.net/

https://www.youtube.com/watch?v=U7nfe4adDw8

https://www.youtube.com/watch?v=SWYqp7iY_Tc

https://forum.qt.io/topic/67053/how-to-connect-to-sqlite-database