CHAPTER 1

1. INTRODUCTION

1.1 Problem statement

The problem of the system is that there was no web applications available for existing system, and that existing system are only accessing the database source and doesn’t provide access for external data source Excel book. And another problem is that it was not more interactive with user. In the existing system, the report is built only using these excel and it was difficult for the admin to build a report based on a set of queries. And there was no direct connection to build a report from excel and database. These problems lead to lot of human errors and to overcome these problems, this web application is developed from scratch.

CHAPTER 2

SYSTEM ANALYSIS

2.1 Existing System

In existing system only database can be accessed. The existing report system will allow data to come from tables, views, or stores procedures within the report builder. It doesn’t provide access for external data source excel book.

2.2 Proposed System

To overcome the existing system, Report Builder Application can provide the functionalities to insert a new student record, store all the student records in the user interface form and provide the data reports using Excel inputs and reports can be retrieved in various formats such as doc file, pdf file and Excel file. By this Application all the information stored in the database will be in a uniform manner.

2.3 System Architecture

System Architecture of the application starts with the super admin and staff who interacts with the server through the frontend. This server connects with the database.

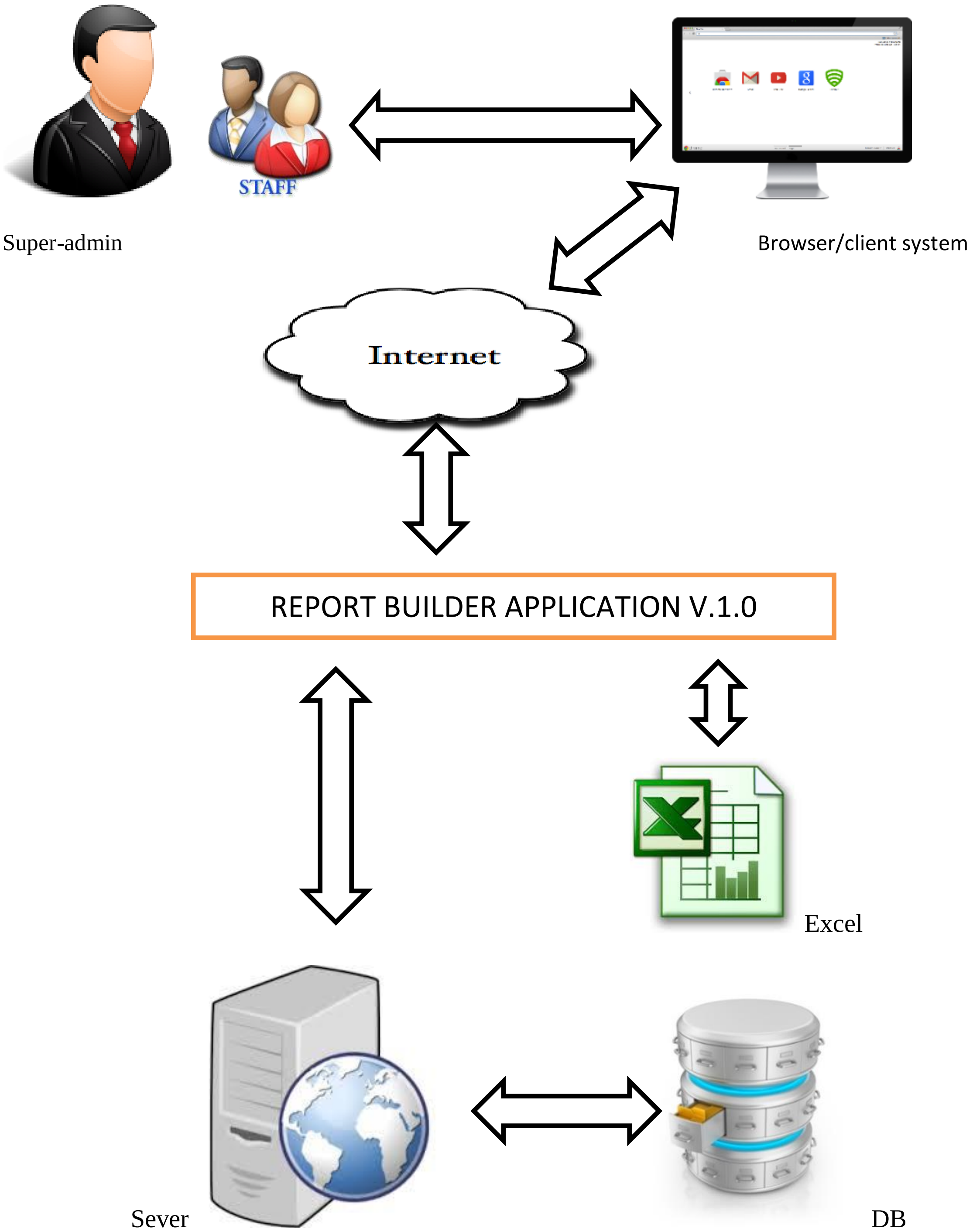


FIG 2.3.1: System Architecture

2.4 Subsystem Description

Subsystem description describes how the components is further divided into subcomponents and relationships and interaction between the subcomponents.

Subsystem of this tool are described below

- Manage students
- Reports
- Load Excel file
- Manage staff

1. MANAGE STUDENTS:

In this module super admin has full privilege of adding new student to all the departments, modifying student information, further updates or deletion can be done only by the super admin. Staffs can also view the student record of their respective department and based on the privileging rights given by the super admin to the staff. Staff can modify or delete the student record from the database.

2. REPORTS:

Reports are customized document. There are predefined categories based on that the information needed, the data's are retrieved from the database and that can be generated as a report and also based on the user requirement. The categories can be customized and the data are retrieved as per the requirement from the database and the reports are generated the different types of charts are generated from the report and the user can select type of the chart according to their wish. The generated report can be downloaded in the various formats like pdf, excel and csv. Super admin can able to view all the department records whereas staffs can view only their privileged departments. Super admin can able to generate the report from all the departments, staff can generate the report only from their department.

3. LOAD EXCEL FILE:

This module can be accessed by both super admin and staffs. Here reports are generated from the external excel file. Once the user login into the system can only able access this module. Even the data from the excel file can be customized as per the requirement and the report are generated based on the data required.

4. MANAGE STAFF:

This module fully controlled by super admin. To manage all the staff login process and giving access privilege is done super admin using in this module. Access privilege in the sense each staff may belong to one department there is no need of accessing other department process or information. So cut down those privilege super admin will give the access privilege to only the department they belong to. So the staff cannot access other department. This process is done by the super admin. Super admin may able to give access privilege to one or more department if there is need of accessing. Super admin can also change there or modify the access privilege whenever is a need.

2.5 Functional requirements

2.5.1 Data description

The data that have been used in this system are described over here.

2.5.1.1 Data objects

Data object that compile to system are:

DATA OBJECT	**ADMISSION_DETAILS**
FIELD	**DESCRIPTION**
ADMISSION_NO	Admission number
ADMISSION_DATE	Date of admission
ADMISSION_QUOTA	Admission quota

DATA OBJECT	**STUDENT PERSONAL DETAILS**
FIELD	**DESCRIPTION**
ADMISSION_NO	Admission number for student
STU_ROLLNO	Student unique roll no
STU_FIRSTNAME	Student first name
STU_LASTNAME	Student last name
STU_GENDER	Student gender
STU_DOB	Student date of birth
STU_RELIGION	Student religion
STU_COMMUNITY	Student community
STU_MOTHER MAIDEN NAME	Student mother maiden name
STU_MOTHER NAME	Student mother name
STU_FATHER NAME	Student father name
STU_PARENT_INCOME	Student parent's income
STU_NATIONALITY	Student nationality
STU_BLOOD_GROUP	Student blood group
STU_MOTHER_TONGUE	Student mother tongue
STU_LANGUAGE_KNOWN1	Student language known 1
STU_LANGUAGE_KNOWN2	Student language known 2

DATA OBJECT **FIELD**	**STUDENT CONTACT DETAILS** **DESCRIPTION**
STU_ROLLNO	Student unique roll no
STU_EMAIL	Student personal email id
STU_PARENT_EMAIL	Student parent email id
STU_MOBILE	Student personal mobile number
STU_PARENT_MOBILE	Student parents’ mobile number
STU_ALTER_MOBILE	Student alternative mobile
PRESENT_HOUSENO	Student present address house number
PRESENT_STREET	Student present address street
PRESENT_AREA	Student present address area or landmark
PRESENT_CITY	Student present address city
PRESENT_DISTRICT	Student present address district
PRESENT_STATE	Student present address state
PRESENT_COUNTRY	Student present address country
PRESENT_PINCODE	Student present address pin code
PERMANENT_HOUSENO	Student permanent address house no
PERMANENT_STREET	Student permanent address street
PERMANENT_AREA	Student permanent address area or landmark
PERMANENT_CITY	Student permanent address city
PERMANENT_DISTRICT	Student permanent address district
PERMANENT_STATE	Student permanent address state
PERMANENT_COUNTRY	Student permanent address country
PERMANENT_PINCODE	Student permanent address pin code

DATA OBJECT FIELD	**STUDENT CURRENT COURSE DETAILS** **DESCRIPTION**
STU_ROLLNO	Student unique identity roll no
STU_REGNO	Student unique university register number
STU_DEGREE	Student currently pursing degree
STU_COURSE	Student currently pursing course
STU_BRANCH	Student currently pursing branch
STU_SECTION	Student section
STU_BATCH	Student batch year
STU_COURSE_TYPE	Student course type
STU_JOIN_MODE	Student join mode for regular or lateral entry

DATA OBJECT FIELD	**STUDENT PREVIOUS ACADEMIC DETAILS** **DESCRIPTION**
STU_ROLLNO	Student unique identity roll no
PREV_DEGREE	Student previous studied degree
PREV_COURSE	Student previous studied course
PREV_BRANCH	Student previous studied branch
YEAR_OF_PASSING	Year of passing previous degree
COURSE_TYPE	Previous course type
INSTITUTE NAME	Previous institute name
BOARD OF EDUCATION/UNIVERSITY NAME	Previous degree board of education
CGPA/MARK OBTAINED	Previous degree cgpa or mark obtained
TOTAL CGPA/MARK	Total cgpa or mark for previous degree
PERCENTAGE	Percentage of previous degree

DATA OBJECT **FIELD**	**STUDENT BLACK MARK DETAILS** **DESCRIPTION**
REMARK_ID	Remark unique id
STU_ROLLNO	Student unique roll no
REMARK_DATE	Remark given date
REMARK_STAFF	Staff name who given remark for student
REMARK_REASON	Remark reason
MET_STAFF	Student met staff yes or no
DATE_OF_MET	Remark student date of met to staff

DATA OBJECT **FIELD**	**DEGREE DETAILS** **DESCRIPTION**
DEGREE_ID	Degree unique id
DEGREE_NAME	Degree name

DATA OBJECT **FIELD**	**COURSE DETAILS** **DESCRIPTION**
COURSE_ID	Course unique id
DEGREE_ID	Degree unique id
COURSE_NAME	Course name

DATA OBJECT **FIELD**	**BRANCH DETAILS** **DESCRIPTION**
BRANCH_ID	Branch unique id
COURSE_ID	Course unique id
BRANCH_NAME	Branch name

DATA OBJECT **FIELD**	**STAFF DETAILS** **DESCRIPTION**
STAFF_ID	Staff unique identity number
STAFF_NAME	Staff name
STAFF_GENDER	Staff gender
STAFF_DOB	Staff date of birth
PRESENT ADDRESS	Present address for staff
PERMANAENT ADDRESS	Permanent address for staff
STAFF_EMAIL	Staff unique personal email
STAFF_MOBILE	Staff contact mobile number
STAFF_DESIGNATION	Staff designation HOD or staffs
STAFF_DEPT	Staff department
STAFF_DOJ	Date of joins in staff
STAFF_QUALIFICATION	Qualification for staff

DATA OBJECT **FIELD**	**LOGIN_INFO DETAILS** **DESCRIPTION**
STAFF_ID	Staff unique identity id
USERNAME	Staff username for login
PASSWORD	Staff password for login

DATA OBJECT **FIELD**	**ACCESS RIGHTS_INFO DETAILS** **DESCRIPTION**
ID	Unique permission id
STAFF_ID	Unique identity id for staff
PERMISSION	Access permissions for staff
MODULE	Access permission belongs to modules

2.6 NON – FUNCTIONAL REQUIREMENTS

Non-functional requirements define the needs in terms of performance, reliability, security, maintainability, and portability

Performance requirements

Performance requirements define acceptable response times for the system functionality, this system can transfer register information from application to database in 5 seconds.

Reliability

This application can be used for any number of systems and provide good efficiency.

Security

This system provides high security by authorizing staffs using their user id

Maintainability

This application is being developed in PHP. So, it is easy to maintain.

Portability

This software is easily transferred to another environment, including install ability.

2.6.1 Hardware Specification

Client side (minimum required)

Processor	:	Pentium 4
RAM	:	1 GB
Disk space	:	40 GB

Server side (minimum required)

Processor	:	Intel Xeon
RAM	:	4 GB
Disk space	:	100 GB

2.6.2 Software Specification

Client side (minimum required)

Front End	:	HTML
Back End	:	MYSQL
Scripting Language	:	JavaScript and jQuery
Framework	:	BOOTSTRAP
Operating System	:	Windows XP SP3

Server side (minimum required)

Business Logic	:	PHP
Server	:	Apache Tomcat 6.0
Operating System	:	LINUX

CHAPTER 3

SYSTEM DESIGN

3.1 DATA MODEL

3.1.1 E-R DIAGRAM

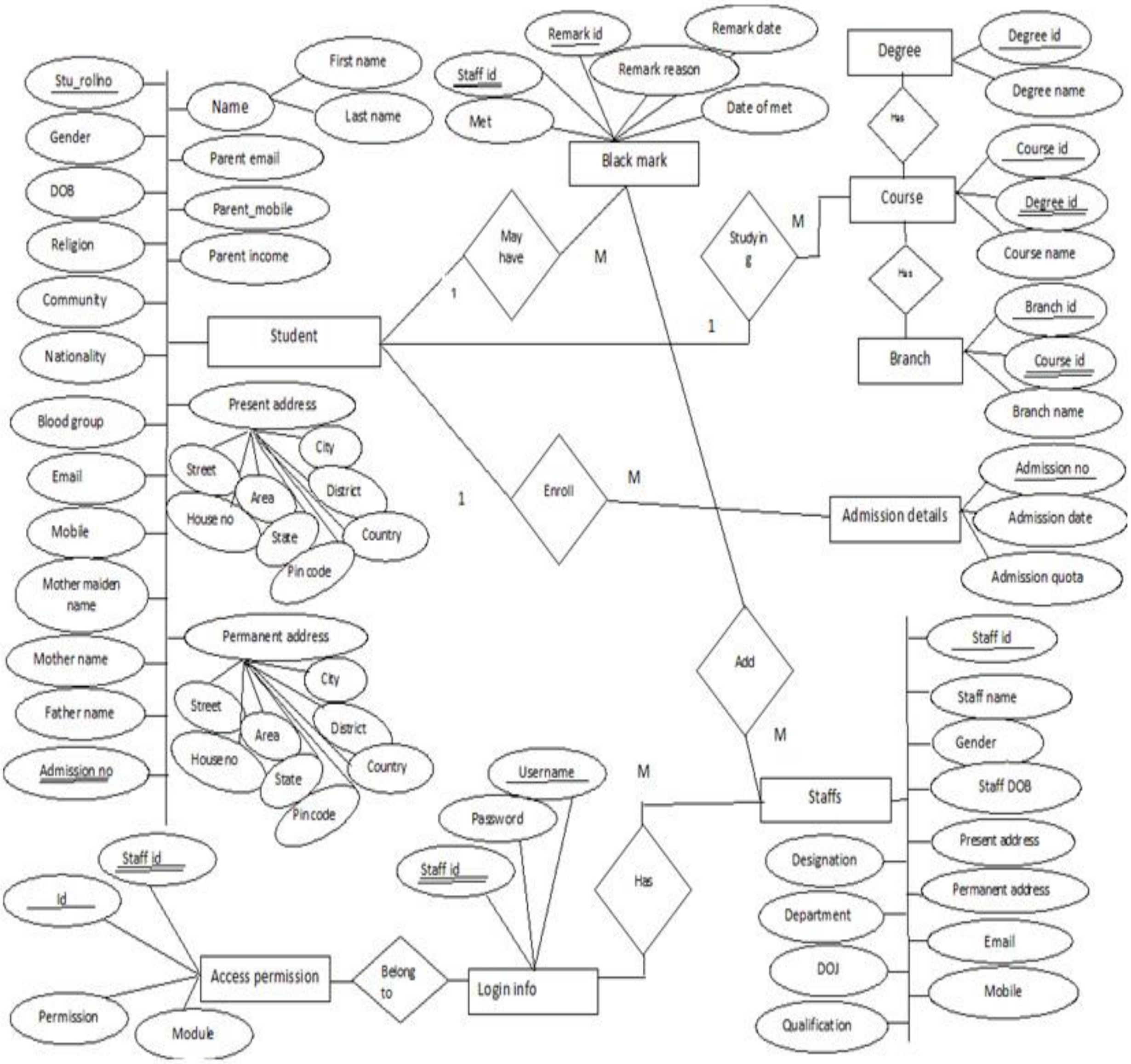


Fig 3.1.1 E-R Diagram

3.1.2 TABLE DESIGN:

3.1.2.1 ADMISSION_DETAILS:

FIELDS	DATATYPE	CONSTRAINTS
admission_no	INTEGER	PRIMARY KEY
admission_date	DATE	NOTNULL
admission_quota	VARCHAR(25)	NOTNULL

3.1.2.2 STUDENT_PERSONAL_DETAILS:

FIELDS	DATATYPE	CONSTRAINT
Admission_no	INTEGER	FOREIGN KEY (admission_details)
stu_rollno	INTEGER(11)	PRIMARY KEY
stu_firstname	VARCHAR(40)	NOTNULL
stu_lastname	VARCHAR(40)	NOTNULL
stu_gender	VARCHAR(8)	NOTNULL
stu_dob	DATE	NOTNULL
stu_religion	VARCHAR(40)	NOTNULL
stu_community	VARCHAR(40)	NOTNULL
stu_mother’s_maiden_name	VARCHAR(40)	NULL
stu_mother’s_name	VARCHAR(40)	NOTNULL
stu_father’s_name	VARCHAR(40)	NOTNULL
stu_parent_income	DOUBLE (10,2)	NOTNULL
stu_nationality	VARCHAR(40)	NOTNULL
stu_blood_group	VARCHAR(5)	NOTNULL
stu_mother_tongue	VARCHAR(40)	NOTNULL
stu_language_known1	VARCHAR(40)	NULL
stu_language_known2	VARCHAR(40)	NULL

3.1.2.3 STUDENT_CONTACT_DETAILS:

FIELDS	DATATYPE	CONSTRAINT
stu_rollno	INTEGER	FOREIGN KEY(stu_personal_details) PRIMARY KEY
stu_email	VARCHAR(50)	UNIQUE
stu_parent_email	VARCHAR(50)	NULL
stu_mobile	VARCHAR(50)	UNIQUE
stu_alter_mobile	VARCHAR(50)	NOTNULL
stu_parents_mobile	VARCHAR(50)	NOTNULL
stu_pre_houseno	VARCHAR (15)	NOTNULL
stu_pre_street	VARCHAR (50)	NOTNULL
stu_pre_area	VARCHAR(100)	NOTNULL
stu_pre_city	VARCHAR(100)	NOTNULL
stu_pre_district	VARCHAR(100)	NOTNULL
stu_pre_state	VARCHAR(100)	NOTNULL
stu_pre_country	VARCHAR(100)	NOTNULL
stu_pre_pincode	MEDIUMINT(9)	NOTNULL
stu_per_houseno	VARCHAR (15)	NOTNULL
Stu_per_street	VARCHAR(50)	NOTNULL
stu_per_area	VARCHAR(100)	NOTNULL
stu_per_city	VARCHAR(100)	NOTNULL
stu_per_district	VARCHAR(100)	NOTNULL
stu_per_state	VARCHAR(100)	NOTNULL
stu_per_country	VARCHAR(100)	NOTNULL
stu_per_pincode	MEDIUMINT(9)	NOTNULL

3.1.2.4 STUDENT_COURSE_DETAILS:

FIELDS	DATATYPE	CONSTRAINTS
stu_rollno	INTEGER	FOREIGNKEY(stu_personal_details) PRIMARY KEY
stu_univ_regno	BIGINT	UNIQUE
stu_degree	INTEGER	NOTNULL
stu_course	INTEGER	NOTNULL
stu_branch	INTEGER	NOTNULL
stu_section	VARCHAR(4)	NULL
stu_batch	VARCHAR(12)	NOTNULL
stu_course_type	VARCHAR(30)	NOTNULL
stu_joined	VARCHAR(30)	NOTNULL

3.1.2.5 STUDENT PREVIOUS ACADEMIC DETAILS:

FIELD	DATATYPE	CONSTRAINTS
id	INTEGER	PRIMARY KEY
Stu_rollno	INTEGER	FOREIGN KEY (stu_personal_details)
prev_degree	VARCHAR(20)	NOTNULL
prev_course	VARCHAR(50)	NOTNULL
prev_branch	VARCHAR(50)	NOTNULL
yr_of_passing	YEAR	NOTNULL
course_type	VARCHAR(30)	NOTNULL
institute_name	VARCHAR(150)	NOTNULL
board_of education/university name	VARCHAR(100)	NOTNULL
cgpa/mark obtained	FLOAT	NOTNULL
total mark/cgpa	INTEGER	NOTNULL
percentage	FLOAT	NOTNULL

3.1.2.6 BLACK_MARK DETAILS:

FIELD	DATATYPE	CONSTRAINTS
remark_id	INTEGER	PRIMARY KEY
stu_rollno	INTEGER	FOREGIN KEY (stu_personal_details)
remark_date	DATE	NOTNULL
remark_staff	INTEGER	FOREIGN KEY(staff_details)
remark_reason	TEXT	NOTNULL
met_staff	VARCHAR(5)	NOTNULL
date_of_met	DATE	NOTNULL

3.1.2.7 DEGREE:

FIELDS	DATATYPE	CONSTRAINTS
degree_id	SMALLINT	PRIMARY KEY
degree_name	VARCHAR(200)	NOTNULL

3.1.2.8 COURSE:

FIELDS	DATATYPE	CONSTRAINTS
course_id	SMALLINT	PRIMARY KEY
degree_id	SMALLINT	FOREIGN KEY(degree)
course_name	VARCHAR(100)	NOTNULL
course_duration	SMALLINT	NOTNULL

3.1.2.9 BRANCH:

FIELD	DATATYPE	CONSTRAINTS
branch_id	SMALLINT	PRIMARY KEY
course_id	SMALLINT	FOREIGN KEY(courses)
branch_name	VARCHAR(50)	NOTNULL

3.1.2.10 STAFF_DETAILS:

FIELD	DATATYPE	CONSTRAINTS
Staff_id	INTEGER	PRIMARY KEY
Staff_name	VARCHAR(30)	NOTNULL
Staff_gender	VARCHAR(10)	NOTNULL
Staff_dob	DATE	NOTNULL
Staff_present_address	TEXT	NOTNULL
Staff_permanent_address	TEXT	NOTNULL
Staff_email	VARCHAR(50)	NOTNULL
Staff_mobile	VARCHAR(10)	NOTNULL
Staff_date_of_join	DATE	NOTNULL
Staff_qualification	VARCHAR(50)	NOTNULL

3.1.2.11 STAFF_NON_TEACHING:

FIELD	DATATYPE	CONSTRAINTS
Id	INTEGER	PRIMARY KEY
Staff_id	INTEGER	FOREIGNKEY(staff details)
department	VARCHAR(100)	NOTNULL
designation	VARCHAR(100)	NOTNULL
Join date	DATE	NOTNULL

3.1.2.12 STAFF_TEACHING:

FIELD	DATATYPE	CONSTRAINTS
Id	INTEGER	PRIMARY KEY
Staff_id	INTEGER	FOREIGNKEY(staff details)
branch	INTEGER	FOREIGN KEY(branch)
designation	VARCHAR(100)	NOTNULL
join date	DATE	NOTNULL

3.1.2.13 LOGIN_INFO:

FIELD	DATATYPE	CONSTRAINTS
Staff_id	INTEGER	FOREIGN KEY(staff_details)
username	VARCHAR(50)	PRIMARY KEY
password	VARCHAR(32)	NOTNULL

3.1.2.14 ACCESS_RIGHTS INFO:

FIELD	DATATYPE	CONSTRAINTS
Id	INTEGER	PRIMARY KEY
username	VARCHAR(50)	FOREGIN KEY (username)
permission	VARCHAR(50)	NOTNULL
Module	VARCHAR(50)	NOTNULL
Dept_permission	VARCHAR(50)	NOTNULL

3.2 PROCESS MODEL

3.2.1 USE CASE DIAGRAM

3.2.1.1SUPER-ADMIN/STAFF

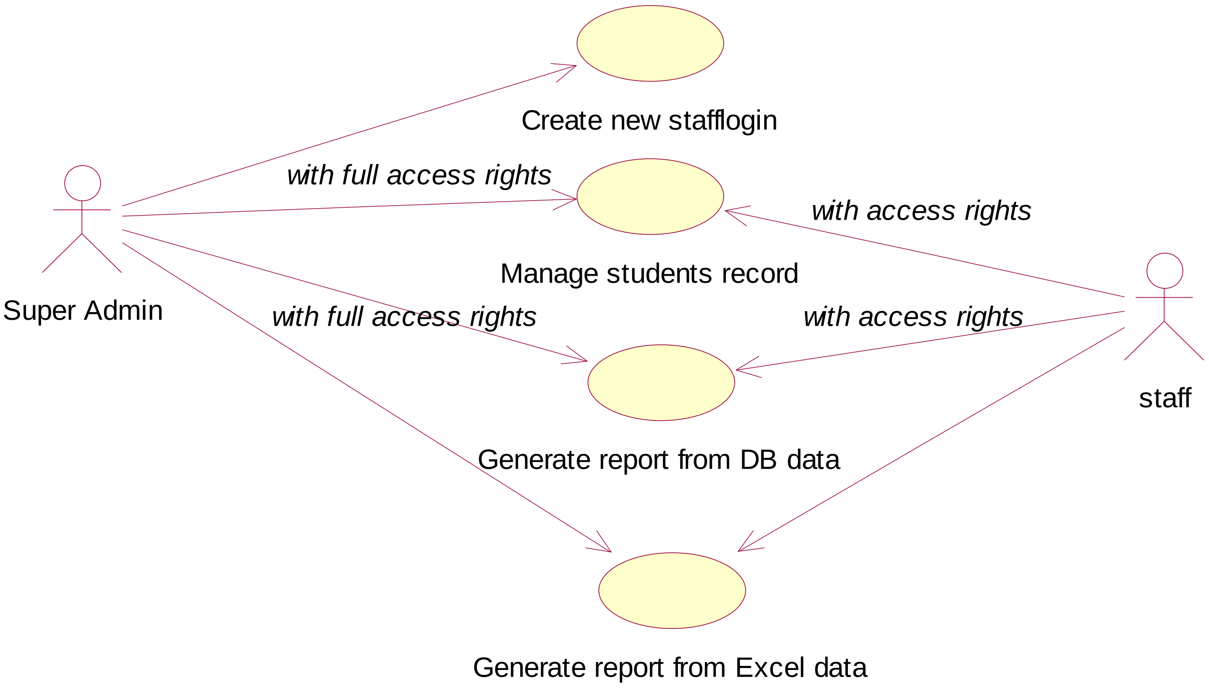


FIG 3.2.1.1:*SUPER-ADMIN/STAFF*

3.2.2 DATA FLOW DIAGRAM

3.2.2.1. Level 0:

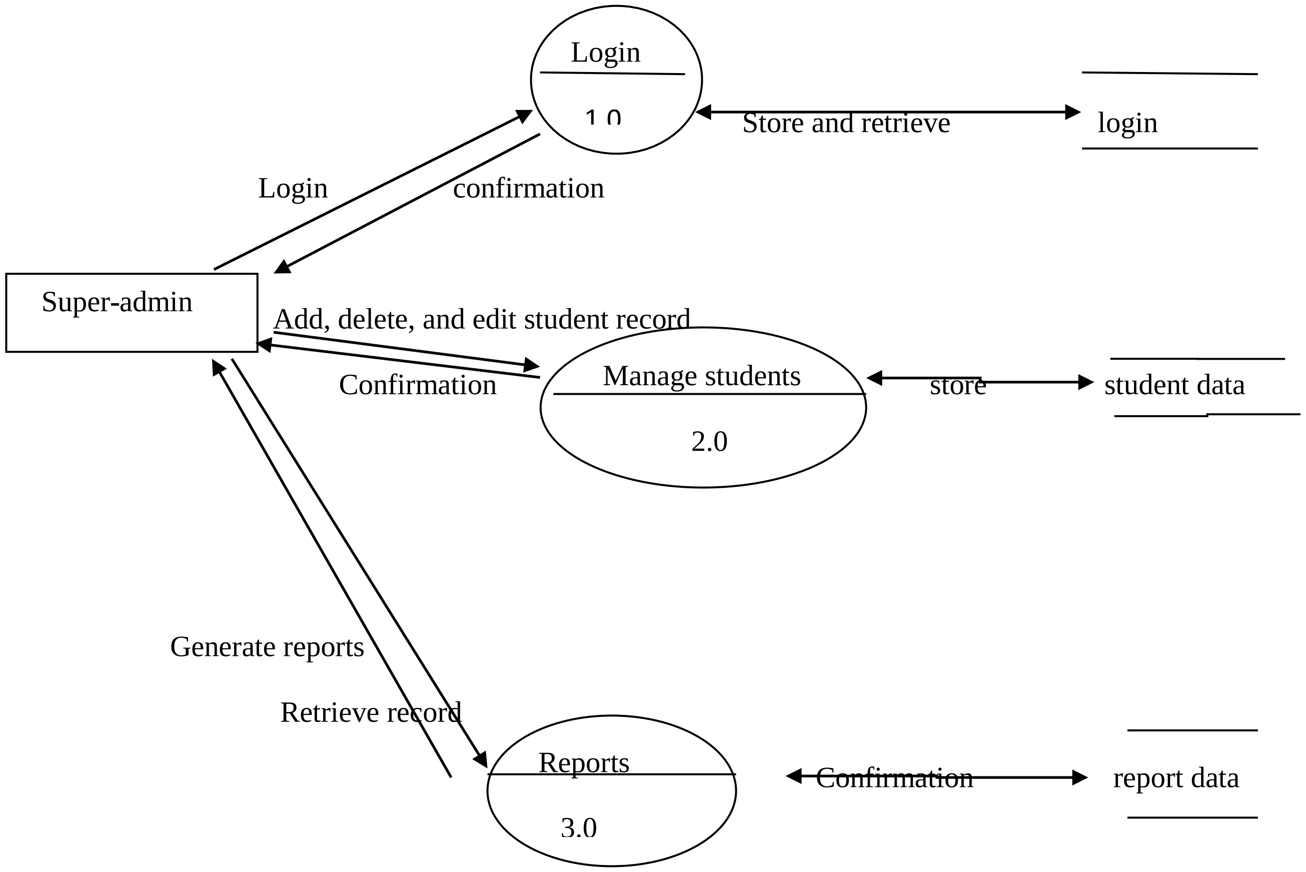


FIG 3.2.2.1 Level 0 Data Flow Diagram

3.2.2.2. Level 1:

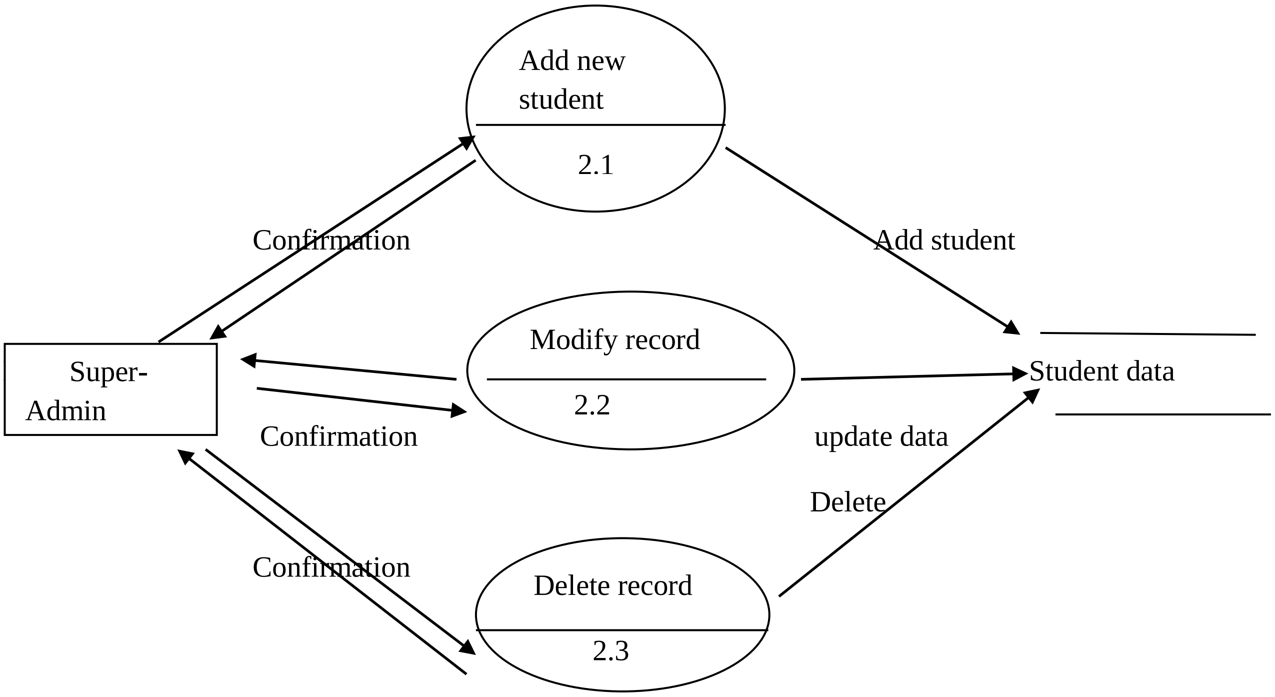


FIG 3.2.2.2 Level 1 Data Flow Diagram

3.2.2.3. Level 2:

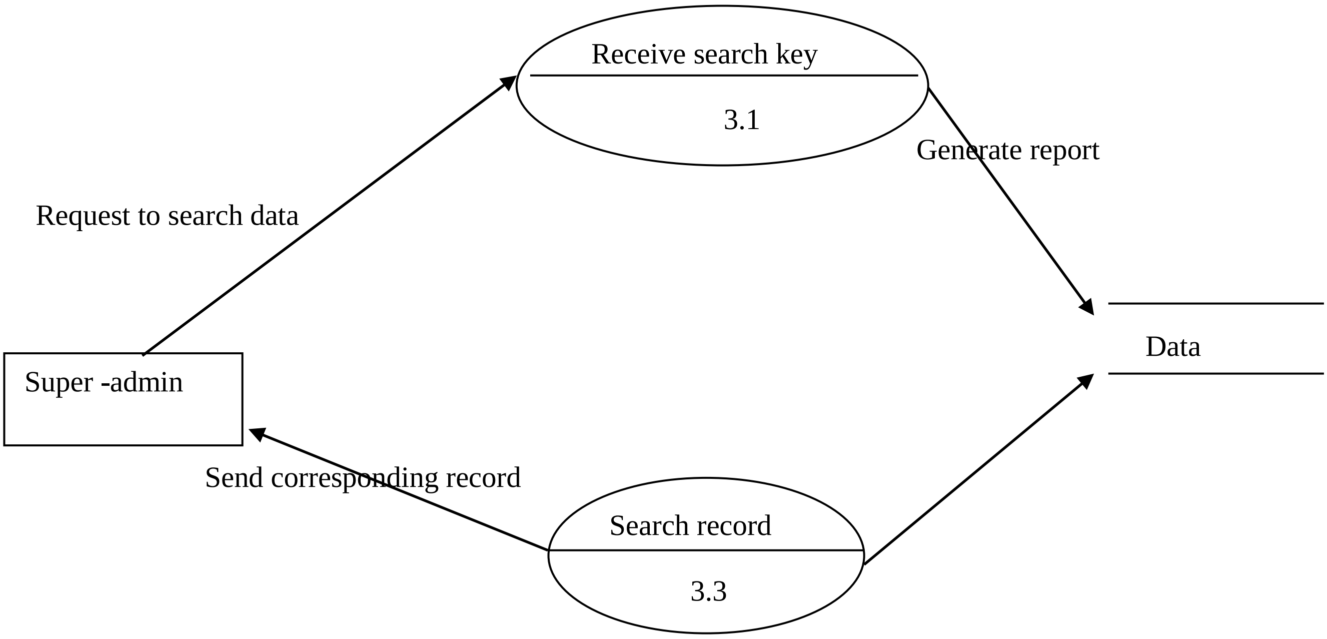


FIG 3.2.2.3 Level 2 Data Flow Diagram

3.2.3 SEQUENCE DIAGRAM

3.2.3.1 MANAGE STUDENTS

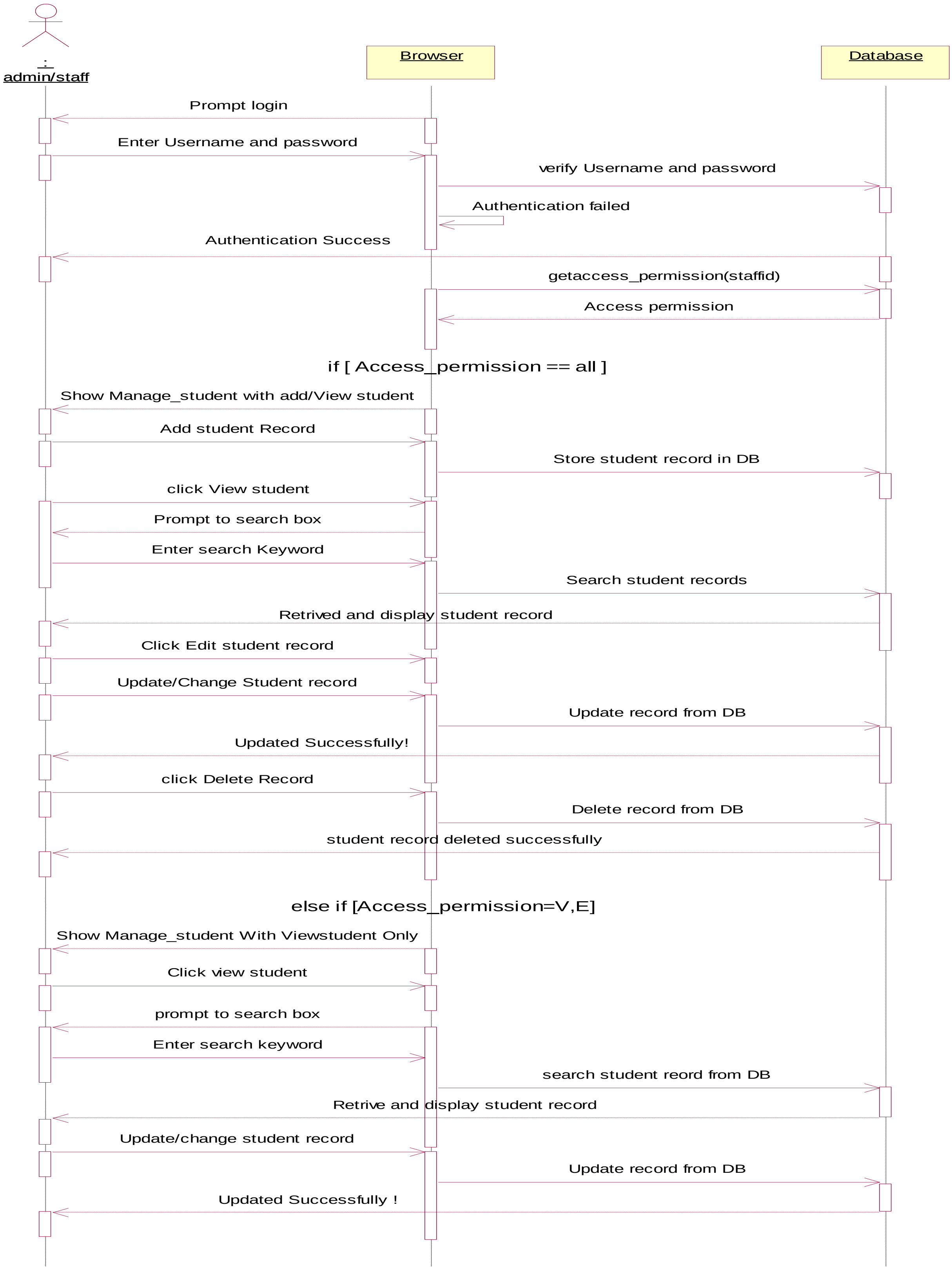

: admin/staff
Browser
Database
Prompt login
Enter Username and password
verify Username and password
Authentication failed
Authentication Success
getaccess_permission(staffid)
Access permission
if [Access_permission == all]
Show Manage_student with add/View student
Add student Record
Store student record in DB
click View student
Prompt to search box
Enter search Keyword
Search student records
Retrived and display student record
Click Edit student record
Update/Change Student record
Update record from DB
Updated Successfully!
click Delete Record
Delete record from DB
student record deleted successfully
else if [Access_permission=V,E]
Show Manage_student With Viewstudent Only
Click view student
prompt to search box
Enter search keyword
search student reord from DB
Retrive and display student record
Update/change student record
Update record from DB
Updated Successfully !

FIG 3.2.3.1: Manage Students

3.2.3.2 CREATE STAFF LOGIN

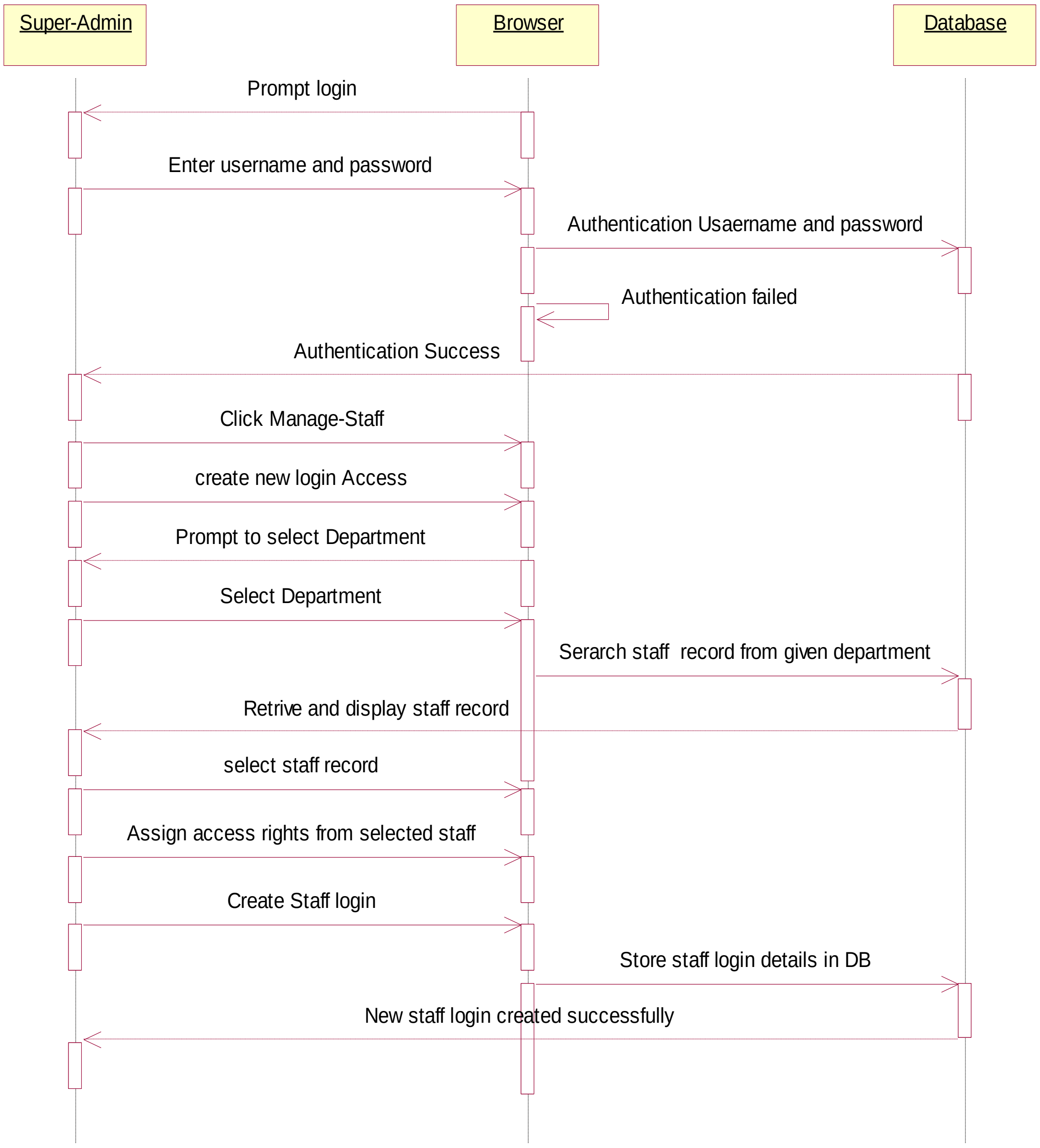

Super-Admin
Browser
Database
Prompt login
Enter username and password
Authentication Usaername and password
Authentication failed
Authentication Success
Click Manage-Staff
create new login Access
Prompt to select Department
Select Department
Serarch staff record from given department
Retrive and display staff record
select staff record
Assign access rights from selected staff
Create Staff login
Store staff login details in DB
New staff login created successfully

FIG 3.2.3.2: Create Staff Login

3.2.3.3 REPORT GENERATION FROM DB

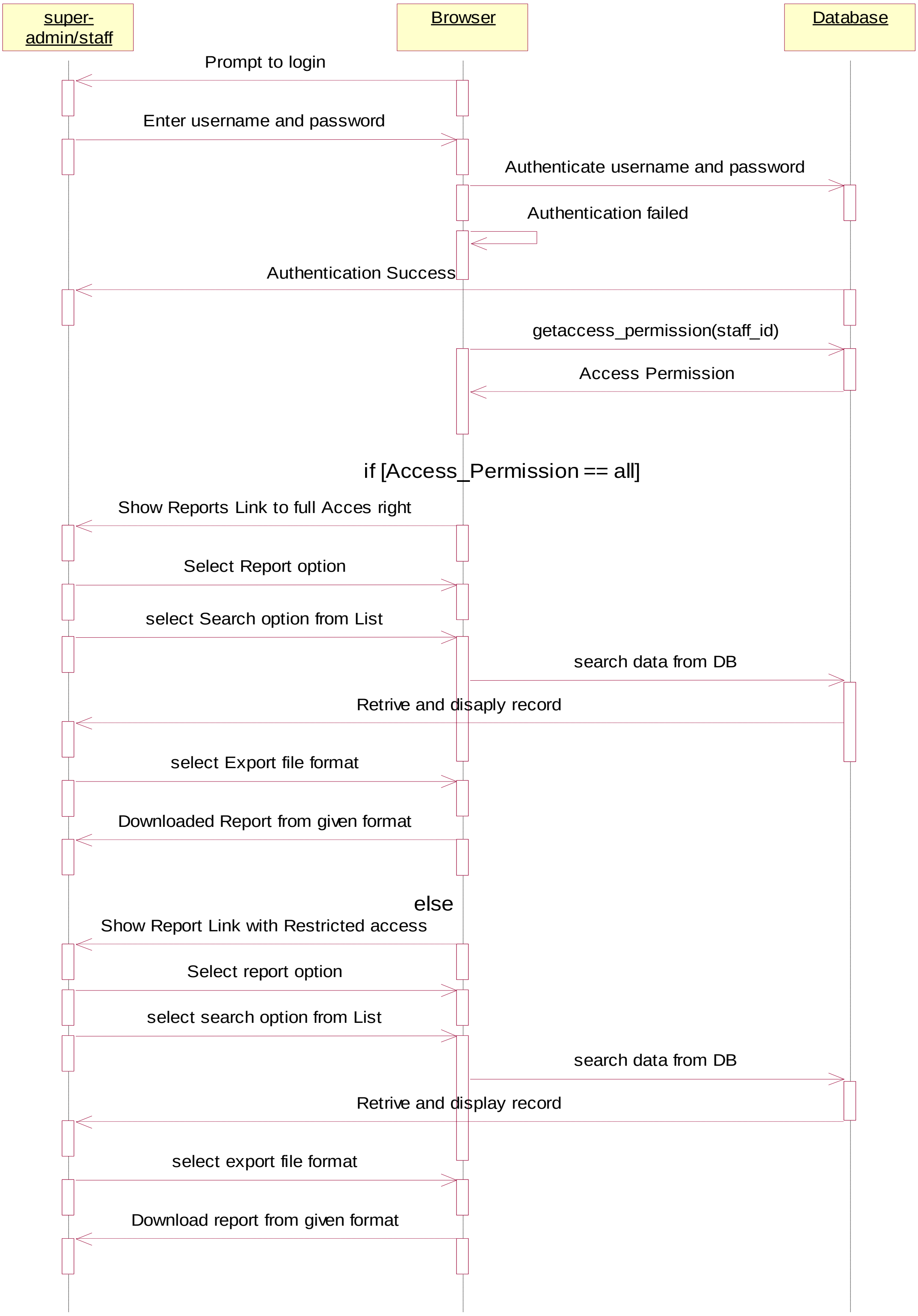

super-admin/staff
Browser
Database
Prompt to login
Enter username and password
Authenticate username and password
Authentication failed
Authentication Success
getaccess_permission(staff_id)
Access Permission
if [Access_Permission == all]
Show Reports Link to full Acces right
Select Report option
select Search option from List
search data from DB
Retrive and disaply record
select Export file format
Downloaded Report from given format
else
Show Report Link with Restricted access
Select report option
select search option from List
search data from DB
Retrive and display record
select export file format
Download report from given format

FIG 3.2.3.3: Report Generation from DB

3.2.3.4 REPORT GENERATION FROM EXCEL

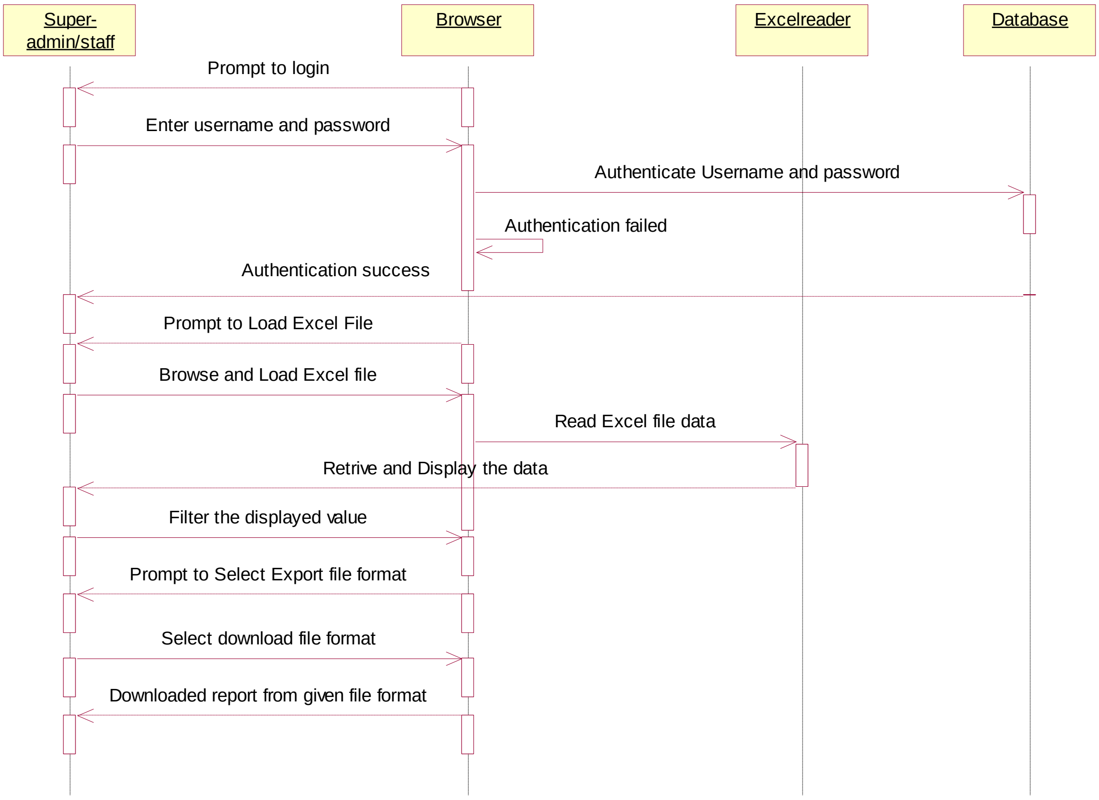

Super-admin/staff
Browser
Excelreader
Database
Prompt to login
Enter username and password
Authenticate Username and password
Authentication failed
Authentication success
Prompt to Load Excel File
Browse and Load Excel file
Read Excel file data
Retrive and Display the data
Filter the displayed value
Prompt to Select Export file format
Select download file format
Downloaded report from given file format

FIG 3.2.3.4: Report Generation from Excel

3.2.4 COLLABORATION DIAGRAM

3.2.4.1 MANAGE STUDENTS

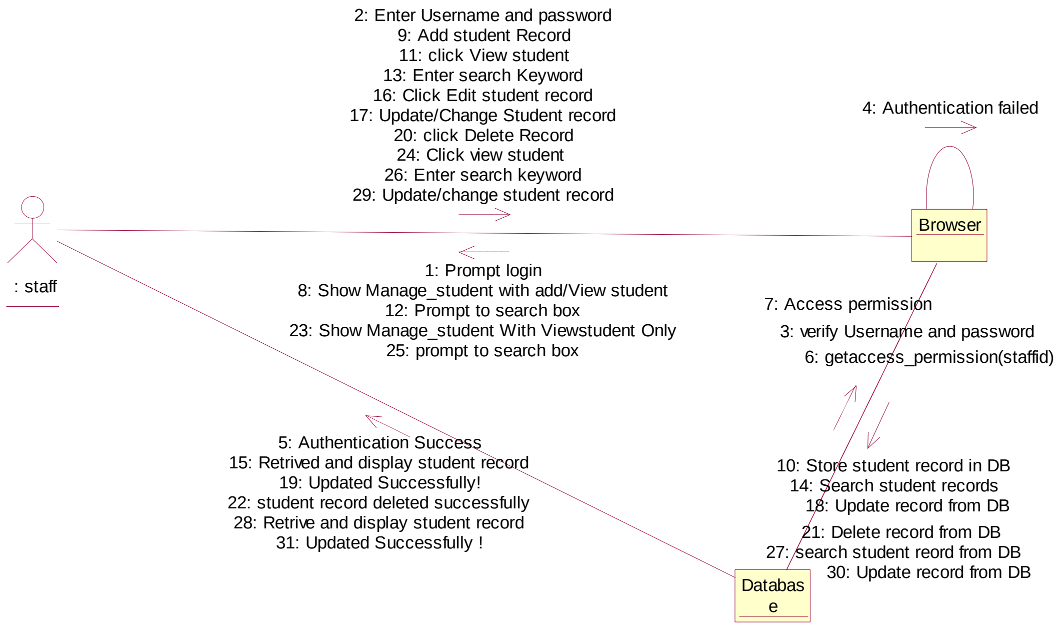

2: Enter Username and password
9: Add student Record
11: click View student
13: Enter search Keyword
16: Click Edit student record
17: Update/Change Student record
20: click Delete Record
24: Click view student
26: Enter search keyword
29: Update/change student record
4: Authentication failed
Browser
: staff
1: Prompt login
8: Show Manage_student with add/View student
12: Prompt to search box
23: Show Manage_student With Viewstudent Only
25: prompt to search box
7: Access permission
3: verify Username and password
6: getaccess_permission(staffid)
5: Authentication Success
15: Retrived and display student record
19: Updated Successfully!
22: student record deleted successfully
28: Retrive and display student record
31: Updated Successfully !
10: Store student record in DB
14: Search student records
18: Update record from DB
21: Delete record from DB
27: search student reord from DB
30: Update record from DB
Databas
e

FIG 3.2.4.1: Manage Students

3.2.4.2 CREATE STAFF LOGIN

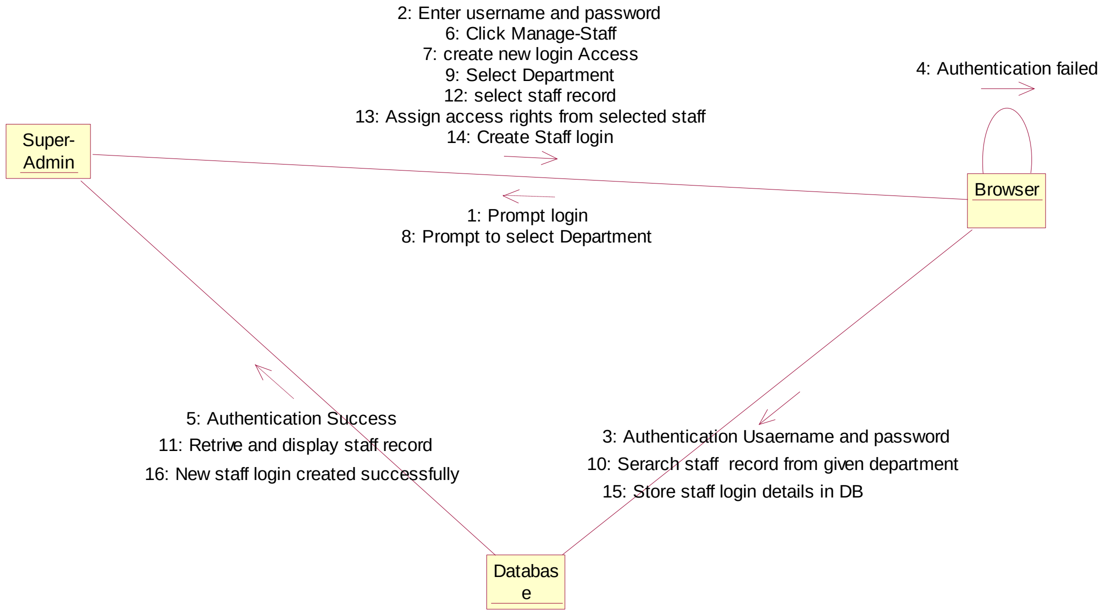

2: Enter username and password
6: Click Manage-Staff
7: create new login Access
9: Select Department
12: select staff record
13: Assign access rights from selected staff
14: Create Staff login
4: Authentication failed
Super-
Admin
Browser
1: Prompt login
8: Prompt to select Department
5: Authentication Success
11: Retrive and display staff record
16: New staff login created successfully
3: Authentication Usaername and password
10: Serarch staff record from given department
15: Store staff login details in DB
Databas
e

FIG 3.2.4.2: Create Staff Login

3.2.4.3 REPORT GENERATION FROM DB DATA

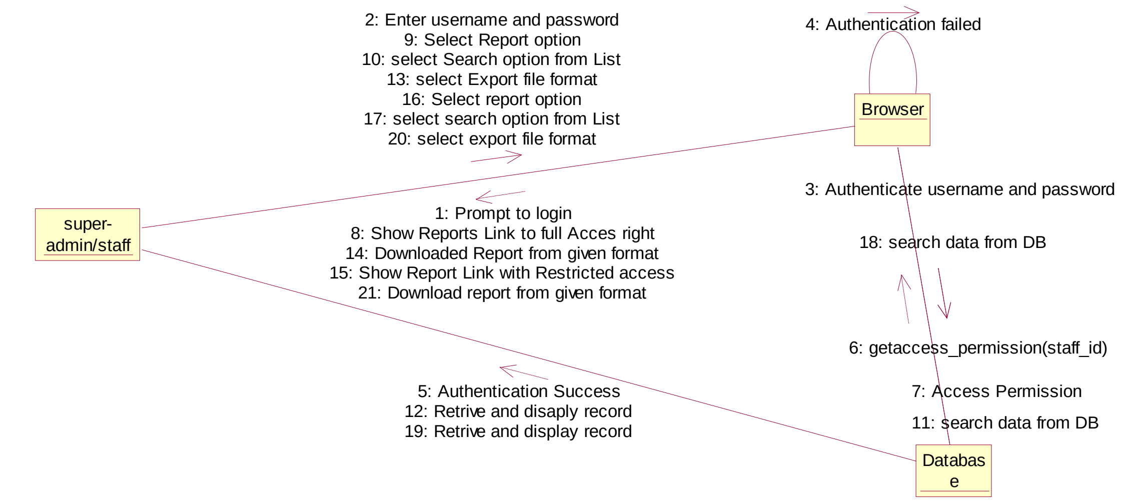

2: Enter username and password
9: Select Report option
10: select Search option from List
13: select Export file format
16: Select report option
17: select search option from List
20: select export file format
4: Authentication failed
Browser
super-admin/staff
3: Authenticate username and password
1: Prompt to login
8: Show Reports Link to full Acces right
14: Downloaded Report from given format
15: Show Report Link with Restricted access
21: Download report from given format
18: search data from DB
6: getaccess_permission(staff_id)
5: Authentication Success
12: Retrive and disaply record
19: Retrive and display record
7: Access Permission
11: search data from DB
Databas e

FIG 3.2.4.3: Report Generation from DB Data

3.2.4.4 REPORT GENERATION FROM EXCEL DATA

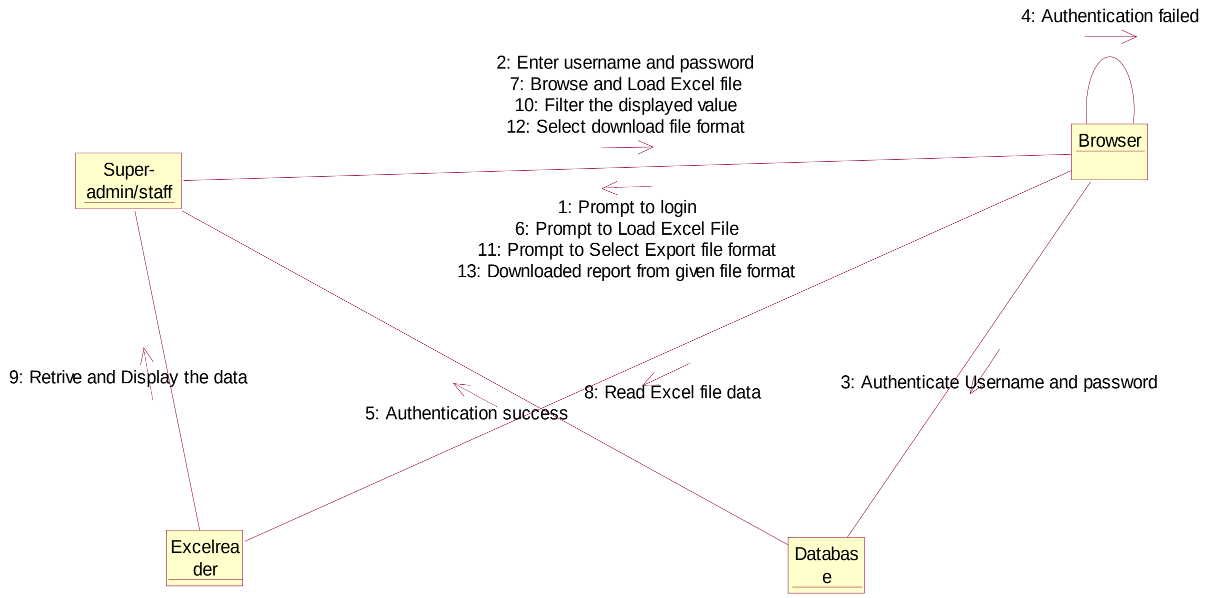

4: Authentication failed
2: Enter username and password
7: Browse and Load Excel file
10: Filter the displayed value
12: Select download file format
Browser
Super-admin/staff
1: Prompt to login
6: Prompt to Load Excel File
11: Prompt to Select Export file format
13: Downloaded report from given file format
9: Retrive and Display the data
8: Read Excel file data
5: Authentication success
3: Authenticate Username and password
Excelrea der
Databas e

FIG 3.2.4.4: Report Generation from Excel Data

3.2.5 ACTIVITY DIAGRAM

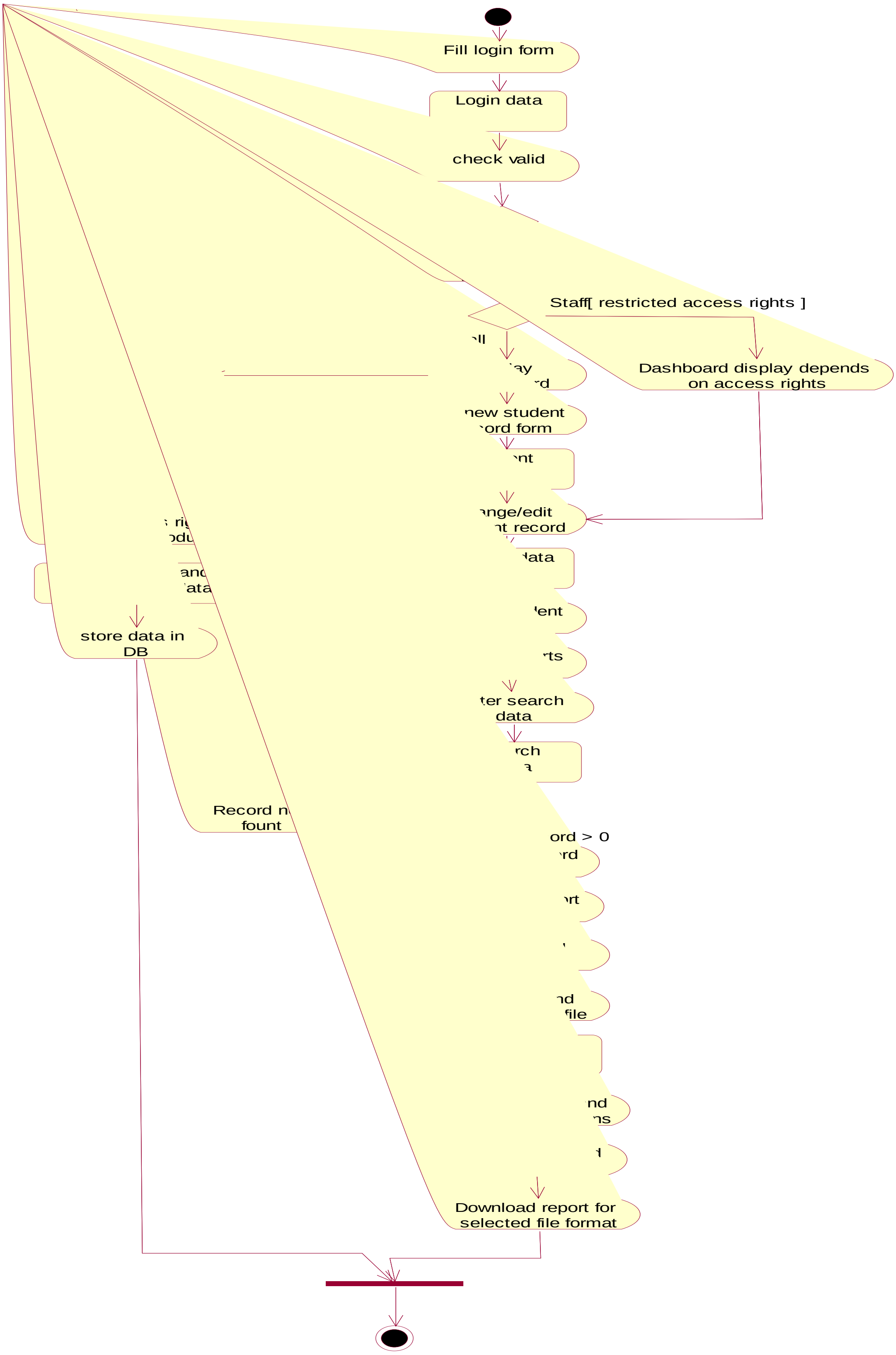


FIG 3.2.5.1: Activity Diagram

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 SAMPLE CODE

Index.php

```
<! DOCTYPE html>
<Html>
<Head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <title>AdminLTE 2 | Dashboard</title>
        <!-- Tell the browser to be responsive to screen width -->
        <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" name="viewport">
        <!-- Bootstrap 3.3.5 -->
        <link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">
</head>
<body class="hold-transition skin-blue-light sidebar-mini">
<div class="wrapper">
                <?php
                        include("header.php");
                        include("sidebar.php");
                ?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
<!-- Content Header (Page header) -->
<section class="content-header">
<h1>
        Dashboard
<small>Control panel</small>
</h1>
<ol class="breadcrumb">
<li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>
<li class="active">Dashboard</li>
</ol>
</section>
                <?php
                        include("db_con.php");
                        $query = "SELECT COUNT(*) FROM admission_details;";
                        $run_query = mysqli_query($con, $query);
                        $total_students = mysqli_fetch_array($run_query);
        $query = "SELECT COUNT(*) FROM current_course WHERE stu_degree = 3";
        $run_query = mysqli_query($con, $query);
        $ug_students = mysqli_fetch_array($run_query);
        $query = "SELECT COUNT(*) FROM current_course WHERE stu_degree = 2";
                $run_query = mysqli_query($con, $query);
                $pg_students = mysqli_fetch_array($run_query);
        ?>
        <section class="content">
                <!-- Small boxes (Total-student STUDENTS) -->
                <div class="row">
                        <div class="col-lg-3 col-xs-6">
                                <!-- small box -->
                                <div class="small-box bg-aqua">
                                        <div class="inner">
                                        <h3><?php echo $total_students[0]; ?></h3>
                                        <p>Total Students</p>
                                        </div>
                                        <div class="icon">
                                                <i class="fa fa-user"></i>
                                        </div>
                                        <a href="#" class="small-box-footer">More info <i
class="fa fa-arrow-circle-right"></i></a>
                                </div>
                                </div><!-- ./col -->

                                <div class="col-lg-3 col-xs-6">
<!-- small box (UG STUDENTS)-->
                                <div class="small-box bg-green">
                                        <div class="inner">
                                        <h3><?php echo $ug_students[0]; ?><sup
style="font-size: 20px"></sup></h3>
                                        <p>UG STUDENTS</p>
                                        </div>
                                        <div class="icon">
                                        <i class="fa fa-user"></i>
                                        </div>
                                        <a href="#" class="small-box-footer">More info <i
class="fa fa-arrow-circle-right"></i></a>
                                </div>
                                </div><!-- ./col -->

                                <div class="col-lg-3 col-xs-6">
<!-- small box (PG STUDENTS)-->
```

```
                <div class="small-box bg-green">
<div class="inner">
<h3><?php echo $pg_students[0] ?><sup style="font-size: 20px"></sup></h3>
<p>PG STUDENTS</p>
</div>
<div class="icon">
    <i class="fa fa-user"></i>
</div>
<a href="#" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>
</div>
</div><!-- ./col -->
<div class="col-lg-3 col-xs-6">
<!-- small box (UG STUDENTS)-->
        <div class="small-box bg-yellow">
        <div class="inner">
        <h3>100 <sup style="font-size: 20px"></sup></h3>
            <p>STAFFS</p>
        </div>
        <div class="icon">
            <!--<i class="ion ion-stats-bars"></i>-->
        </div>
        <a href="#" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>
        </div>
    </div><!-- ./col -->
</div><!--./row -->
</section><!-- ./section-content -->
<?php
    include("sidepane.php");
?>
</div><!-- /.content-wrapper -->
<?php
     include("footer.php");
?>
</div><!-- ./wrapper -->
</body>
</html>
```

Add.php:

```
<?php
    include("../../db_con.php");
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>Admin - Managestudents</title>
    <!-- Custom css For Form Styling -->
    <link rel="stylesheet" type="text/css" href="<?php echo admin_index;
?>dist/css/style.css" />
<!-- Select2 -->
<link rel="stylesheet" href="<?php echo admin_index; ?>plugins/select2/select2.min.css">
    </head>
<body class="hold-transition skin-blue-light sidebar-mini">
    <div class="loader"></div><!-- div from loader  -->
    <div class="wrapper">
        <?php
            //Include the header.php
            include("../../header.php");
            //Include the sidebar.php
            include("../../sidebar.php");
        ?>
        <div class="content-wrapper">
            <section class="content-header">
                <h1>Manage Students
                    <small>Add Student Record</small>
                </h1>
                <ol class="breadcrumb">
                    <li><a href="#"></a>Home</li>
                    <li><a href="#">Manage Students</a></li>
                    <li>Add Student Record</li>
                </ol>
            </section><!--./section header -->
            <section class="content">
            <div class="row">
                <div class="panel panel-info">
                    <div class="panel-heading">Add Student
Record</div>
                    <div class="panel-body" style="font-size:12px;">
                <form class="" id="student_form" name="student" method="post" action="">
<!-- Student Personal Details -->
        <div class="frm" id="sf1">
                <fieldset>
                <legend>Step 1 of 4 <span>(Admission/Personal Details)</span></legend>

                <div class="from-group">
                <label class="col-lg-2 control-label" for="admission_no">Admission No <span class="text-danger">*</span></label>
                <div class="col-lg-6">
                <input type="text" name="admission_no" class="form-control input-sm" id="admission_no" autocomplete="off" />
                </div><!-- ./lg6-->
</div><!-- ./form group for admissionno-->
<div class="clearfix" style="height: 10px;clear: both;"></div>
<!-- Admission date -->
        <div class="form-group" style="margin-bottom:0px;">
        <label class="col-lg-2 control-label" for="admission_date">Date Of Admission<span class="text-danger">*</span></label>
        <div class="col-lg-6 input-group" style="margin-left:18%;">
        <div class="input-group-addon">
        <i class="fa fa-calendar"></i>
        </div><!-- ./Input group addon -->
<input type="text" class="form-control input-sm" name="admission_date" style="width:94%;" data-inputmask="'alias':'dd/mm/yyyy'" data-mask />
        </div><!-- ./col-lg-6 and input-group-->
        <label for="admission_date" generated="true" class="error" style="margin-left:18%;"></label>
        </div><!-- ./form-group for admission_date -->
<!-- Admission qouta -->
        <div class="form-group">
        <label class="col-lg-2 control-label" for="admission_qouta">Admission-Quota <span class="text-danger">*</span></label>
        <div class="col-lg-6 col-sm-5">
        <select class="form-control select2 input-sm" name="admission_quota" data-placeholder="Select a admission_quota">
        <option value="default">--Select--</option>
        <option>Councelling</option>
        <option>Management</option>
        </select>
<label for="religion" generated="true" class="error"></label>
</div>
</div><!-- ./form-group for religion -->
<div class="clearfix" style="height: 10px;clear: both;"></div>
<div class="from-group">
<label class="col-lg-2 control-label" for="rollno">Rollno <span class="text-danger">*</span></label>
<div class="col-lg-6">
```

```
<input type="text" name="rollno" class="form-control input-sm" id="rollno"
autocomplete="off" />
</div><!-- ./lg6-->
</div><!-- ./form group for rollno-->
<div class="clear fix" style="height: 10px; clear: both ;"></div>
<div class="form-group">
<label class="col-lg-2 control-label" for="fname">FirstName <span class="text-
danger">*</span></label>
<div class="col-lg-6">
<input type="text" class="form-control input-sm" name="fname" id="fname"
autocomplete="off" />
</div><!-- ./lg-6-->
</div><!-- ./form-group for fname -->
<div class="clearfix" style="height: 10px;clear: both;"></div>
<div class="form-group">
        <label class="col-lg-2 control-label" for="lname">LastName <span class="text-
danger">*</span></label>
        <div class="col-lg-6">
        <input type="text" class="form-control input-sm" name="lname" id="lname"
autocomplete="off" />
        </div><!-- ./lg-6 -->
        </div><!-- ./form-group for lname -->
<div class="clearfix" style="height: 10px;clear: both;"></div>
<div class="form-group">
        <label class="col-lg-2 control-label" for="gender">Gender <span class="text-
danger">*</span></label>
        <div class="col-lg-6">
        <label class="Form-label--tick">
        <input type="radio" name="gender" class="Form-label-radio" value="male" />
        <span class="Form-label-text"> Male</span>
        </label>
        <label class="Form-label--tick">
        <input type="radio" name="gender" class="Form-label-radio" value="female" />
        <span class="Form-label-text"> Female</span>
        </label>
<label class="Form-label--tick">
        <input type="radio" name="gender" class="Form-label-radio" value="others" />
        <span class="Form-label-text"> Others</span>
</label>
<label for="gender" generated="true" class="error" style="font-
weight:bold;color:#FF0000;margin-left:7%;"></label>
</div><!-- ./lg-6-->
</div><!-- ./form-gropu for Gender -->
<div class="clearfix" style="height: 10px;clear: both;"></div>
<!-- Date Of birth -->
        <div class="form-group">
<label class="col-lg-2 control-label" for="dob">Date of Birth <span class="text-
danger">*</span></label>
<div class="col-lg-6 input-group" style="margin-left:18%;">
<div class="input-group-addon">
```

```
<i class="fa fa-calendar"></i>
        </div><!-- ./Input group addon -->
        <input type="text" class="form-control input-sm" name="dob" style="width:94%;"
data-inputmask="'alias':'dd/mm/yyyy'" data-mask />
</div><!-- ./col-lg-6 and input-group-->
        <label for="dob" generated="true" class="error" style="margin-
left:18%;"></label>
        </div><!-- ./form-group for DOB -->
        <!-- Religion -->
        <div class="form-group">
        <label class="col-lg-2 control-label" for="religion">Religion <span class="text-
danger">*</span></label>
        <div class="col-lg-6 col-sm-5">
<select class="form-control select2 input-sm" name="religion" data-placeholder="Select a
Religion">
<option value="default">--Select--</option>
        <option>Hindu</option>
        <option>Christian</option>
        <option>Islam</option>
        <option>Jain</option>
        <option>Sikhism</option>
</select>
<label for="religion" generated="true" class="error"></label>
</div>
</div><!-- ./form-group for religion -->
<div class="clearfix" style="height: 10px;clear: both;"></div>
        <!-- Community -->
        <div class="form-group">
        <label class="col-lg-2 control-label" for="Community">Community <span
class="text-danger">*</span></label>
                <div id="totopscroller"></div>
                <?php
                        //Include the footer page
                        include("../../footer.php");
                        //Include the sidepane page
                        include("../../sidepane.php");
                ?>
        </div><!-- ./wrapper -->
</body>
</html>
```

CHAPTER 5

TESTING

5.1 Testing Strategy

A testing strategy is a general approach to the testing process rather than a method of devising particular system or component tests.

The main objectives of testing it as follows:

- Testing is the process of executing program with the intent of finding error.
- A good test is one that has a high probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.
- To affirm the quality of the project.
- To find and eliminate any errors from previous stages.
- To validate the software and to eliminate the operation.
- Reliability of the system.

5.1.1 Unit Testing

Unit testing is one of the types of testing strategies that focuses on testing the software module and its boundaries.

Specify the minimum degree of comprehensiveness desired. Identify the techniques which will be used to judge the comprehensiveness of the testing effort. Specify any additional completion criteria. The techniques to be used to trace requirements should be specified.

Unit testing is the phase of testing that testing the basic functionality and the structure of the code. The module is testing to ensure that information flows is proper into and out of the program under unit testing. Each of the individual reports has been tested extensively to determine whether the required output has been obtained or not.

Unit testing is done every individual unit in the system.

In Report Builder, I included login and staff Login creational module as a sample for unit testing. It registers new staff and student into the system. Once the registration for the staff is successful, the details will be stored into the system and the staff Username will be generated and send mail to the staff. Open Load Excel page and Excel name will be displayed left text box Display the message as 'The file has been loaded' and data to be displayed Table.

5.1.1.1 TEST PLAN

Table 5.1 – Test Plan

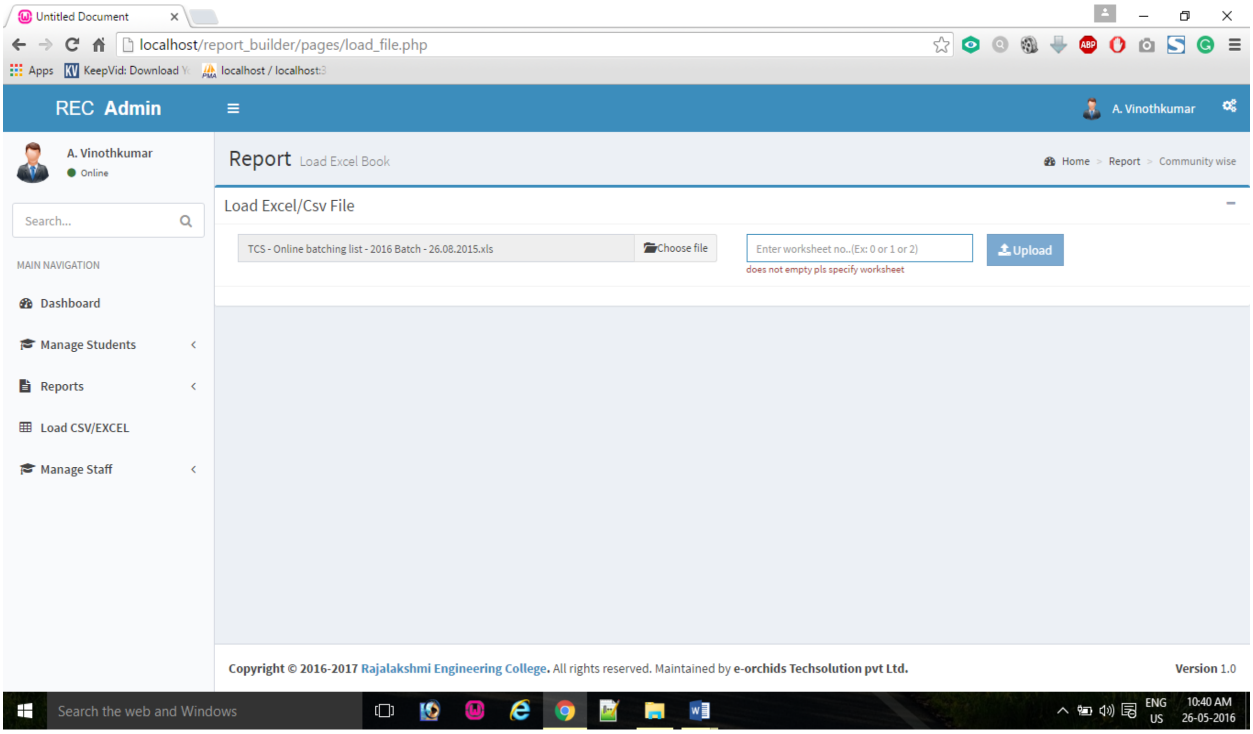
Module Name	**Type of Test**	**Description about the test**
Manage Students	Unit test, Functional test	If the super admin enters the valid student data. It will show next screen. And store data into database If invalid details given it will show the error message
Manage staff	Unit test, Functional test	If the staff enters the valid username and password. It will show next screen. If invalid details given it will show the error message
Load Excel	Unit test, Functional test	If the staff load Excel file. It will show Excel data in page. If invalid file format given it will show the error message

5.1.1.2 Test Cases:

Test Case1:

Table - 5.1 Test case – Load Excel

<table>
<tr><td colspan="2">ID</td><td>1</td></tr>
<tr><td colspan="2">Title</td><td>Load Excel Sheet</td></tr>
<tr><td colspan="2">Priority</td><td>High</td></tr>
<tr><td colspan="2">Module</td><td>Load Excel/CSV</td></tr>
<tr><td colspan="2">Execution result</td><td>🟩 Success</td></tr>
<tr><td colspan="2">Purpose</td><td>Load external Excel file for report generation</td></tr>
<tr><td colspan="2">Created by</td><td>Balakumar B, 2016-05-11 09:40 AM</td></tr>
<tr><td colspan="2">Test Environment</td><td>WAMP</td></tr>
<tr><td colspan="3">Pre-conditions: User has Excel/ CSV file with first row as a Header</td></tr>
<tr><td></td><td>Description</td><td>Expected result</td></tr>
<tr><td>Step 1</td><td>Go to http://localhost/report_builder/pages/load_file.php</td><td>Open Load Excel page</td></tr>
<tr><td>Step 2</td><td>Choose Excel file from your PC</td><td>Excel name will be displayed left text box</td></tr>
<tr><td>Step 3</td><td>Enter work sheet Number from excel sheet</td><td></td></tr>
<tr><td>Step 4</td><td>Click ‘upload’ button</td><td>Display the message as ‘The file has been loaded’ and data to be displayed Table</td></tr>
<tr><td colspan="3">Post-conditions: The Excel Data Loaded Successfully.</td></tr>
</table>



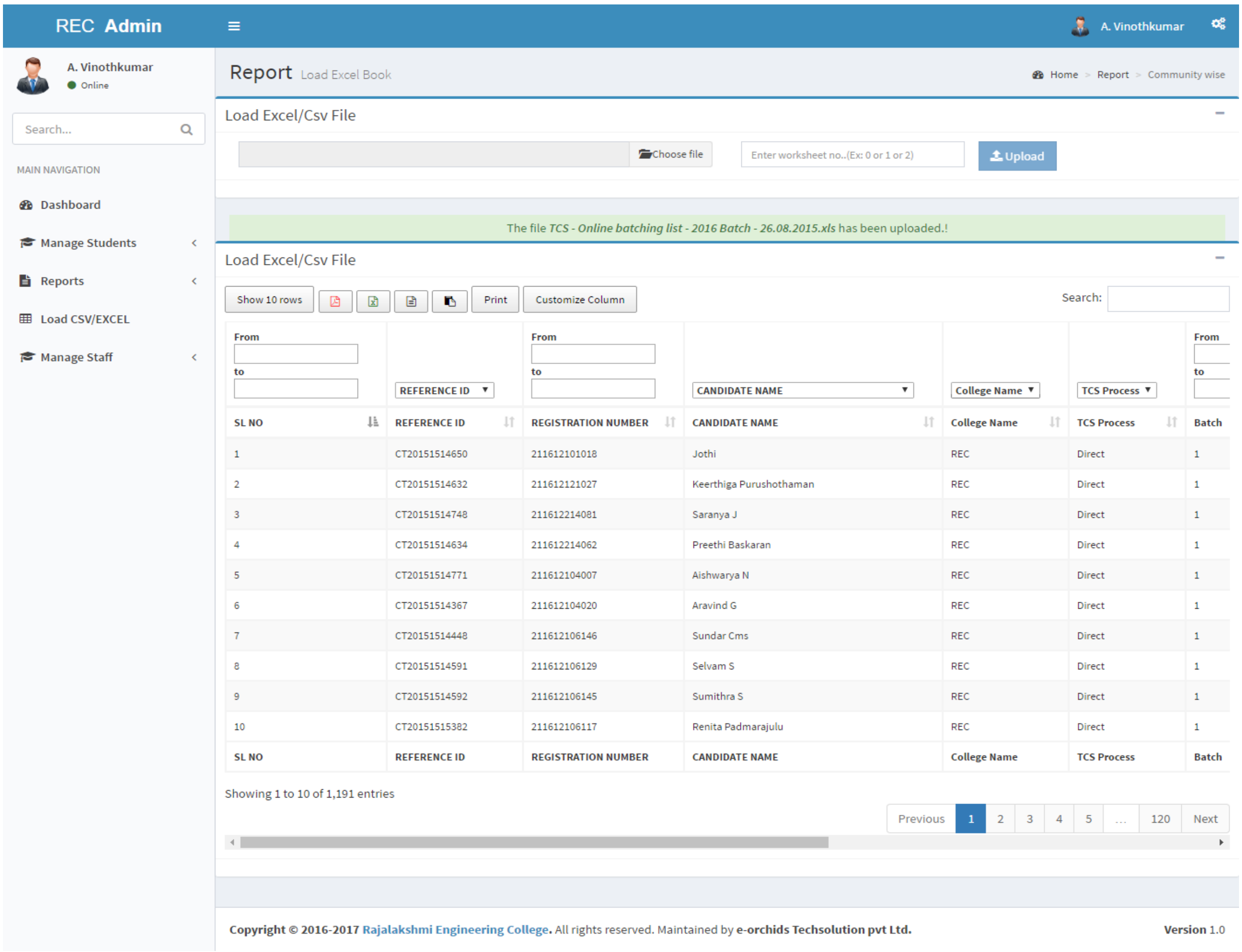
TCS - Online batching list - 2016 Batch - 26.08.2015.xls

Enter worksheet no..(Ex: 0 or 1 or 2)

does not empty pls specify worksheet

Copyright © 2016-2017 Rajalakshmi Engineering College. All rights reserved. Maintained by e-orchids Techsolution pvt Ltd. Version 1.0

Fig. 5.1(a) - Invalid Details for Load excel file



Enter worksheet no..(Ex: 0 or 1 or 2)

The file *TCS - Online batching list - 2016 Batch - 26.08.2015.xls* has been uploaded.!

Load Excel/Csv File

SL NO	REFERENCE ID	REGISTRATION NUMBER	CANDIDATE NAME	College Name	TCS Process	Batch
1	CT20151514650	211612101018	Jothi	REC	Direct	1
2	CT20151514632	211612121027	Keerthiga Purushothaman	REC	Direct	1
3	CT20151514748	211612214081	Saranya J	REC	Direct	1
4	CT20151514634	211612214062	Preethi Baskaran	REC	Direct	1
5	CT20151514771	211612104007	Aishwarya N	REC	Direct	1
6	CT20151514367	211612104020	Aravind G	REC	Direct	1
7	CT20151514448	211612106146	Sundar Cms	REC	Direct	1
8	CT20151514591	211612106129	Selvam S	REC	Direct	1
9	CT20151514592	211612106145	Sumithra S	REC	Direct	1
10	CT20151515382	211612106117	Renita Padmarajulu	REC	Direct	1
SL NO	REFERENCE ID	REGISTRATION NUMBER	CANDIDATE NAME	College Name	TCS Process	Batch

Showing 1 to 10 of 1,191 entries

Copyright © 2016-2017 Rajalakshmi Engineering College. All rights reserved. Maintained by e-orchids Techsolution pvt Ltd. Version 1.0

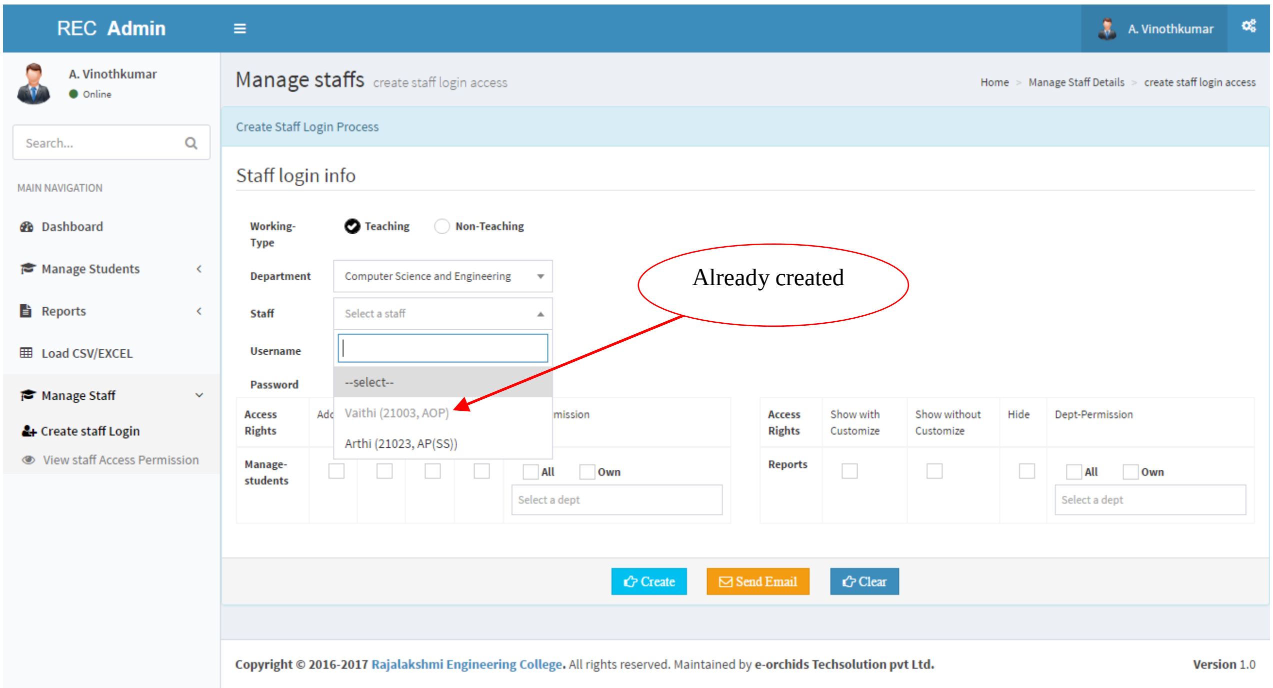
Fig. 5.1(a) - Valid Details for Load excel file

Test Case2:

Table - 5.2 Test case – create staff Login

ID	2	
Title	Create staff login	
Priority	High	
Module	Manage staff	
Execution result	🟩 Success	
Purpose	Provide username and password in individual login for staff.	
Created by	Balakumar B, 2016-05-11 03:40 PM	
Test Environment	WAMP	
Pre-conditions: super-admin only can create staff login and staff details should be there in DB		
	Description	Expected result
Step 1	Go to http://localhost/report_builder/pages/manage_staffs/add.php	Open create staff login page
Step 2	Ensure Choose Teaching or Non-Teaching	If choosing Teaching load Teaching departments in Department Dropdown List or If choosing non-Teaching Load Non-Teaching departments in Department dropdown.
Step 3	Should be Select Department as ‘Computer science and Engineering’	Load staff details in Staff Dropdown list like *Arthi (21023, (AP(SS))*, If staff already have username and password show disabled
Step 4	Ensure Select staff as ‘Arthi (21023, (AOP))’	Username and password automatically generated. username as ‘*arthi_21023*’ and password as ‘*arthi_21023*’ and automatically filled at Username and password fields
Step 5	Assign access permissions as ‘edit, delete, view’ for manage students module and select dept.-permission as	

	'own' and assign access permission as 'show without customize' for reports modules and select dept.-permission as 'own'.	
Step 6	Click 'create' button	Show response message as 'staff Login created successfully for Arthi(21023, AP(SS))'
Step 7	Click 'Send Email' button	Show response message as 'mail sent successfully'
Step 8	Click 'clear' button	Clear all selected and filled fields.
Post-conditions: Staff login details and access permission details stored in database successfully		



Staff login info

Working-Type: ● Teaching ○ Non-Teaching

Department: Computer Science and Engineering

Staff: Select a staff

--select--

Vaithi (21003, AOP)

Arthi (21023, AP(SS))

Username

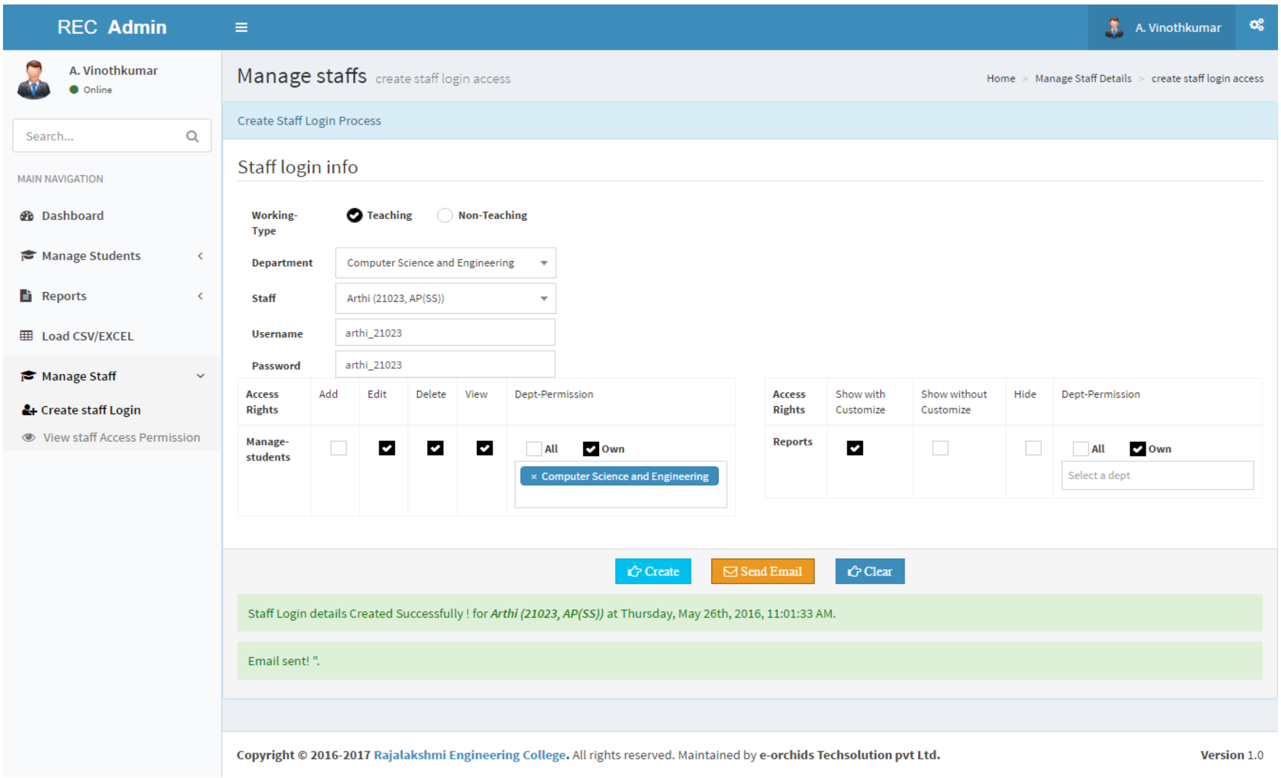
Password

Access Rights	Add				Dept-Permission
Manage-students	□	□	□	□	□ All □ Own Select a dept

Access Rights	Show with Customize	Show without Customize	Hide	Dept-Permission
Reports	□	□	□	□ All □ Own Select a dept

Copyright © 2016-2017 Rajalakshmi Engineering College. All rights reserved. Maintained by e-orchids Techsolution pvt Ltd. Version 1.0

Fig. 5.2(a) - Invalid Details for staff login creational



Working-Type: ● Teaching ○ Non-Teaching

Department: Computer Science and Engineering

Staff: Arthi (21023, AP(SS))

Username: arthi_21023

Password: arthi_21023

Access Rights	Add	Edit	Delete	View	Dept-Permission
Manage-students	□	☑	☑	☑	□ All ☑ Own × Computer Science and Engineering

Access Rights	Show with Customize	Show without Customize	Hide	Dept-Permission
Reports	☑	□	□	□ All ☑ Own Select a dept

Staff Login details Created Successfully ! for *Arthi (21023, AP(SS))* at Thursday, May 26th, 2016, 11:01:33 AM.

Email sent! ".

Copyright © 2016-2017 Rajalakshmi Engineering College. All rights reserved. Maintained by e-orchids Techsolution pvt Ltd. Version 1.0

Fig. 5.2(a) - Valid Details for staff login creational

5.1.2 Validation Testing

Validation testing is used to uncover and correct interfacing errors. It succeeds when the software functions in the manner that is reasonably expected. The various inputs in the system is checked and validated in the client side and provide response before sending the data to the server.

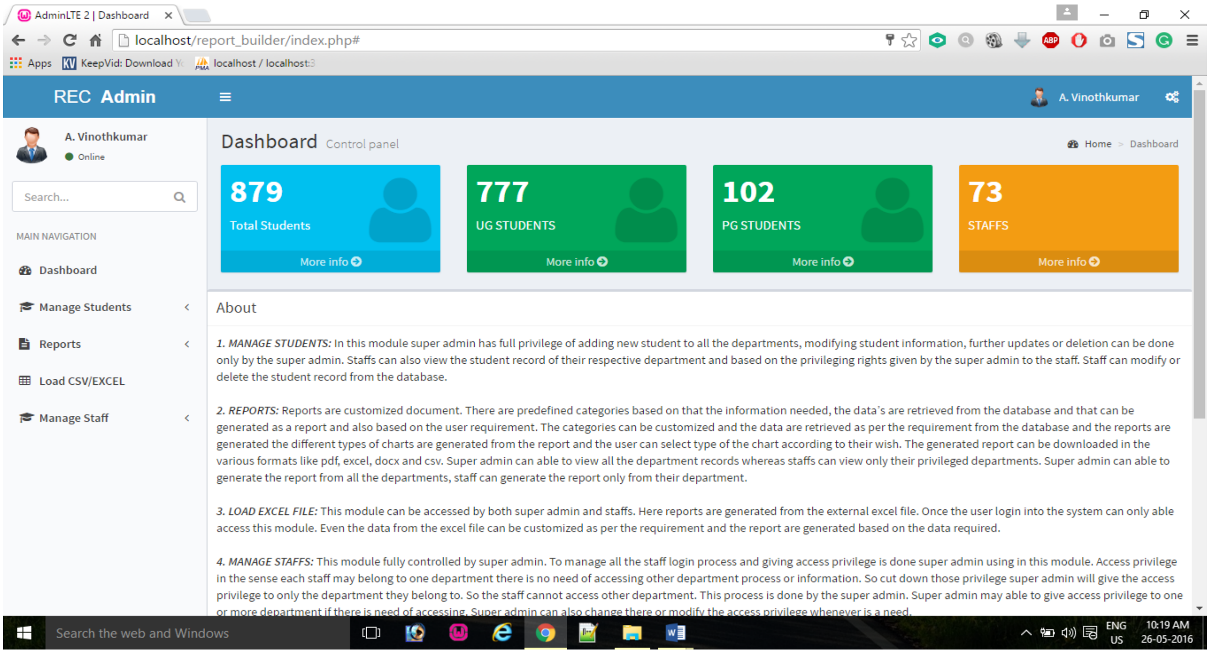
Validation testing plays a vital role in maintaining data consistency and avoiding incorrect data to store into the system.

Validation test cases for admin login is given below

Test Case 3:

Table 5.3 - Test case- Staff Login

<table>
<tr><td>ID</td><td colspan="2">3</td></tr>
<tr><td>Title</td><td colspan="2">Log in</td></tr>
<tr><td>Priority</td><td colspan="2">High</td></tr>
<tr><td>Execution result</td><td colspan="2">🟩 Success</td></tr>
<tr><td>Purpose</td><td colspan="2">Verify Login with valid login credential</td></tr>
<tr><td>Created by</td><td colspan="2">Balakumar B, 2016-05-12 10:40 AM</td></tr>
<tr><td>Test Environment</td><td colspan="2">WAMP</td></tr>
<tr><td colspan="3">Pre-conditions: User has valid username and password</td></tr>
<tr><td></td><td>Description</td><td>Expected result</td></tr>
<tr><td>Step 1</td><td>Open a Browser</td><td>The browser opens</td></tr>
<tr><td>Step 2</td><td>Go to http://localhost/report_builder/</td><td>The Login page is shown</td></tr>
<tr><td>Step 3</td><td>Enter ‘<i>a.vinothkumar_21044</i>’ as the username and ‘<i>a.vinothkumar_21044</i>’as the password.</td><td></td></tr>
<tr><td>Step 4</td><td>Click ‘sign in’ button</td><td>The System logs in to your request
Redirect Dashboard page/home page</td></tr>
<tr><td colspan="3">Post-conditions: User is validated with database and successfully login to account. The account session details are logged in database</td></tr>
</table>



About

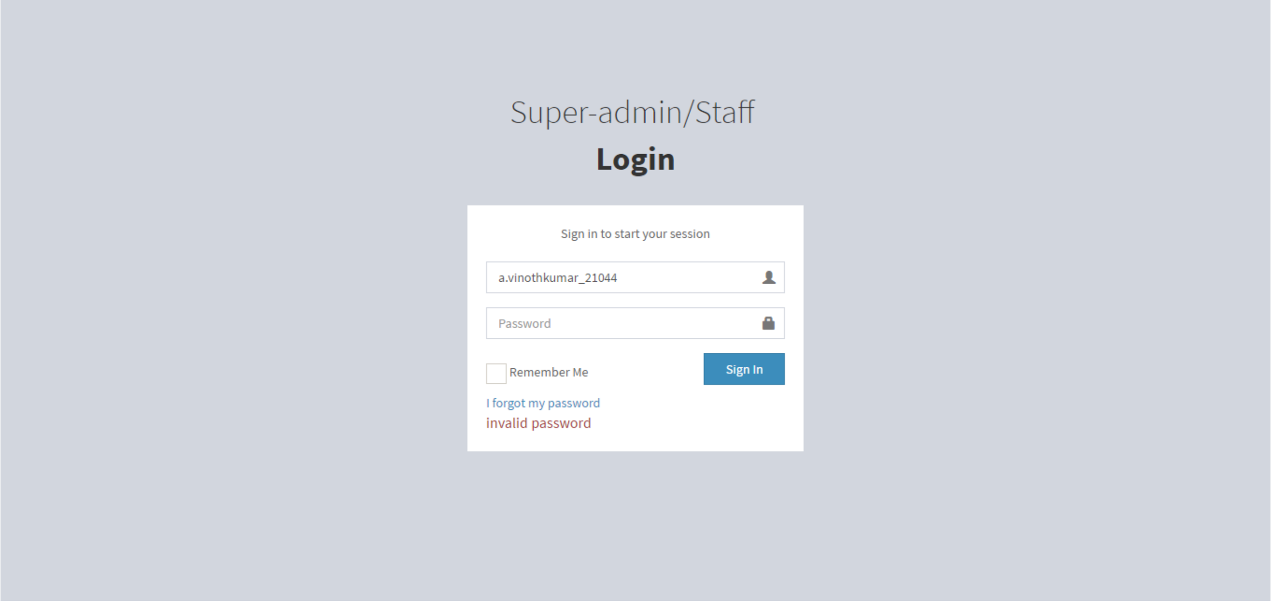
1. MANAGE STUDENTS: In this module super admin has full privilege of adding new student to all the departments, modifying student information, further updates or deletion can be done only by the super admin. Staffs can also view the student record of their respective department and based on the privileging rights given by the super admin to the staff. Staff can modify or delete the student record from the database.

2. REPORTS: Reports are customized document. There are predefined categories based on that the information needed, the data's are retrieved from the database and that can be generated as a report and also based on the user requirement. The categories can be customized and the data are retrieved as per the requirement from the database and the reports are generated the different types of charts are generated from the report and the user can select type of the chart according to their wish. The generated report can be downloaded in the various formats like pdf, excel, docx and csv. Super admin can able to view all the department records whereas staffs can view only their privileged departments. Super admin can able to generate the report from all the departments, staff can generate the report only from their department.

3. LOAD EXCEL FILE: This module can be accessed by both super admin and staffs. Here reports are generated from the external excel file. Once the user login into the system can only able access this module. Even the data from the excel file can be customized as per the requirement and the report are generated based on the data required.

4. MANAGE STAFFS: This module fully controlled by super admin. To manage all the staff login process and giving access privilege is done super admin using in this module. Access privilege in the sense each staff may belong to one department there is no need of accessing other department process or information. So cut down those privilege super admin will give the access privilege to only the department they belong to. So the staff cannot access other department. This process is done by the super admin. Super admin may able to give access privilege to one or more department if there is need of accessing. Super admin can also change there or modify the access privilege whenever is a need.

Fig. 5.3(a)–Staff Login, Valid Details



Super-admin/Staff

Login

Sign in to start your session

a.vinothkumar_21044

Password

□ Remember Me

I forgot my password

invalid password

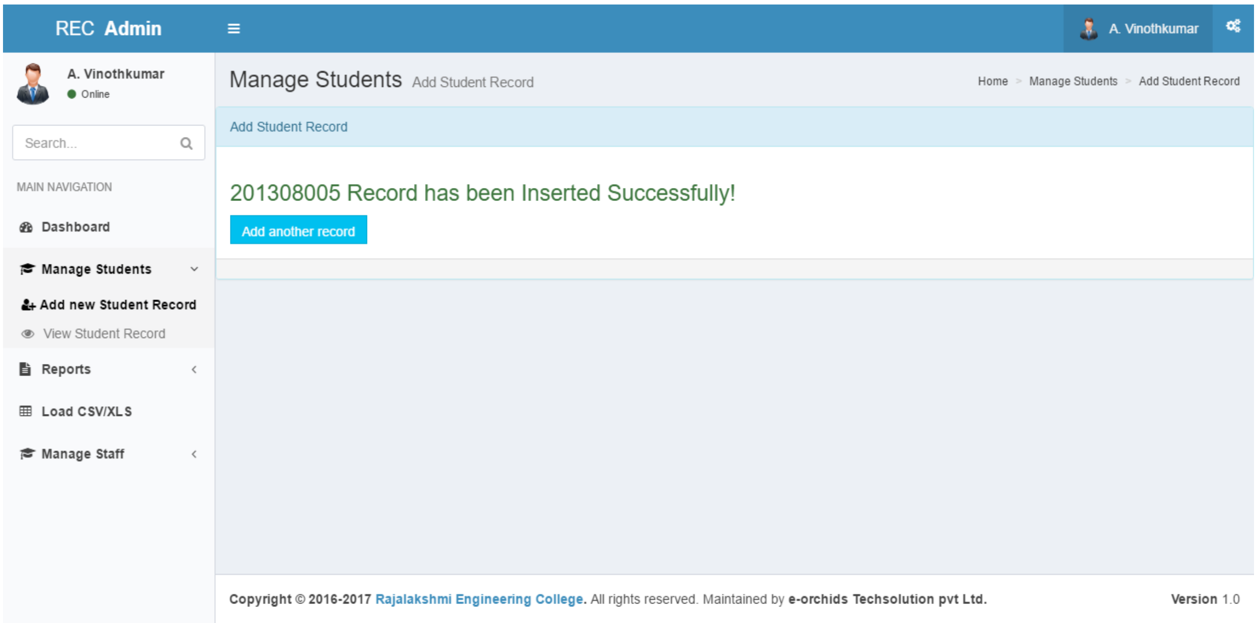
Fig. 5.3(b)–Staff login, Invalid Data

Test Case 4:

Table 5.4 - Test case – Add new Student Record

<table>
<tr><td colspan="2">ID</td><td colspan="2">4</td></tr>
<tr><td colspan="2">Title</td><td colspan="2">Add Student Individual record</td></tr>
<tr><td colspan="2">Priority</td><td colspan="2">Medium</td></tr>
<tr><td colspan="2">Module Name</td><td colspan="2">Manage Students</td></tr>
<tr><td colspan="2">Execution result</td><td colspan="2">🟩 Success</td></tr>
<tr><td colspan="2">Purpose</td><td colspan="2">Store student personal, contact and academic details</td></tr>
<tr><td colspan="2">Created by</td><td colspan="2">Balakumar B, 2016-05-11 03:40</td></tr>
<tr><td colspan="2">Test Environment</td><td colspan="2">WAMP</td></tr>
<tr><td colspan="4">Pre-conditions: User has complete record from student like personal details, contact details, academic details and user has permission for Add new student</td></tr>
<tr><td></td><td colspan="2">Description</td><td>Expected result</td></tr>
<tr><td>Step 1</td><td colspan="2">Go to http://localhost/report_builder/pages/manage_students/add.php</td><td>The Add student record page shown</td></tr>
<tr><td>Step 2</td><td colspan="2">Enter student personal details, admission details for admission no, admission date, and select admission quota then enter roll no, first name, last name, etc... And all fields are mandatory except mother’s maiden name and languages known.</td><td></td></tr>
<tr><td>Step 3</td><td colspan="2">Click ‘Next’ Button</td><td>Doesn’t show error message</td></tr>
</table>

		and show contact details fields
Step 4	Enter student contact details for email, parent email, mobile, parent mobile, and present address, permanent address contains house no, street, area, city, district, state, country and pin code. these all fields are mandatory and student email and parent email shout not be same	
Step 5	Click ‘Next’ Button	Doesn’t show error message and show current academic details fields.
Step 6	Enter student current course details like university register number, and select join mode, course-type, degree, course, and branch and enter section. All fields are mandatory except section.	
Step 7	Click ‘Next’ Button	Doesn’t show error message. And show previous academic details fields
Step 8	Default show X degree fields if you want add one more degree record click Add row button for top of table, dynamically add one more degree fields and then select degree then enter details. If not need one more degree fields click Remove row button dynamically remove bottom of the row fields	
Step 9	Click ‘submit’ Button	Doesn’t show error message and display response message for Student record successfully Inserted.
Post-conditions: Student all details stored in database		



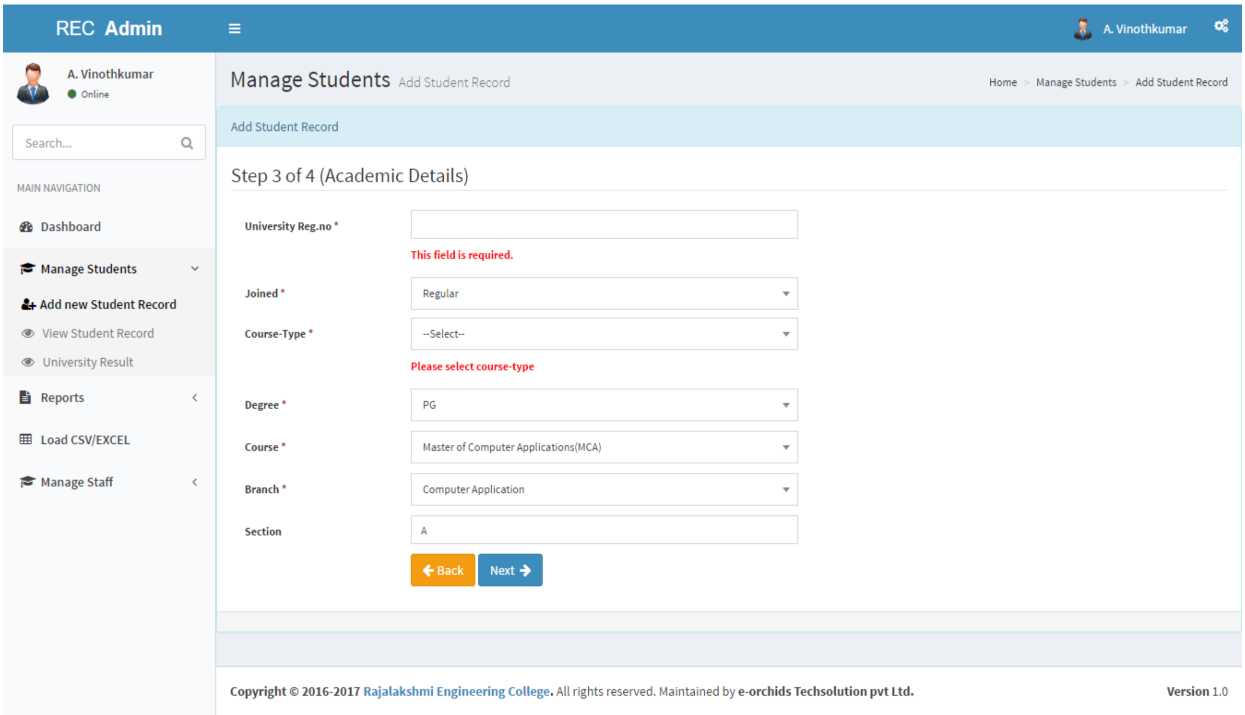
Manage Students Add Student Record

Add Student Record

201308005 Record has been Inserted Successfully!

Copyright © 2016-2017 Rajalakshmi Engineering College. All rights reserved. Maintained by e-orchids Techsolution pvt Ltd. Version 1.0

Fig. 5.4(a) – Add new student record, Valid Details



Manage Students Add Student Record

Add Student Record

Step 3 of 4 (Academic Details)

University Reg.no *	
	This field is required.
Joined *	Regular
Course-Type *	--Select--
	Please select course-type
Degree *	PG
Course *	Master of Computer Applications(MCA)
Branch *	Computer Application
Section	A

Copyright © 2016-2017 Rajalakshmi Engineering College. All rights reserved. Maintained by e-orchids Techsolution pvt Ltd. Version 1.0

Fig. 5.4(b) – Add new student record, Invalid Data

5.1.3 CYCLOMATIC COMPLEXITY

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of the software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Path testing is a structural method for unit testing. Its goal is to ensure that all statements in the program, and both sides of every (two-sided) test in the program, are executed or followed by at least one test. It can be applied to source code or to reasonably detailed pseudo code, provided that all tests and loops in the source code are explicitly shown in the pseudo code as well. Path testing includes two major steps:

1. Use the source code to produce a Flow Graph
2. Using the flow graph and source code as references, produce a set of test for the given program.

Consider the following source code which authenticate the users and provides them respective functions.

```
S1: $stu_id=$_POST [“stu_rollno”];

S2: $stu_name=$_POST [“stu_name”];

S3: $email=$_POST [“email”];

S4: $stu_dob=$_POST [“stu dob”];

S5: $address=$_POST [“address”];

S6: $state=$_POST [“state”];

S7: $pin code=$_POST [“pincode”];

S8: $country=$_POST [“country”];

S9: $district=$_POST [“district”];
```

S10: $mob_no=$_POST [“mob_no”];

C1: if (empty ($name))

S11: echo “Please enter name”;

C2: else if (empty ($address))

S12: echo “Please enter address”;

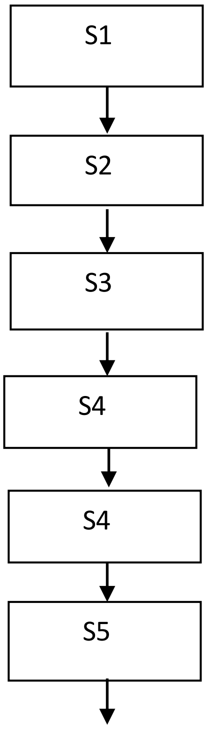
C3: else if (empty ($email))

S13: echo “Please enter email address”;

C4: else if (empty ($mobile))

S14: echo “Please enter mobile number”;

The flow – graph for the above code is shown in Fig 5.5



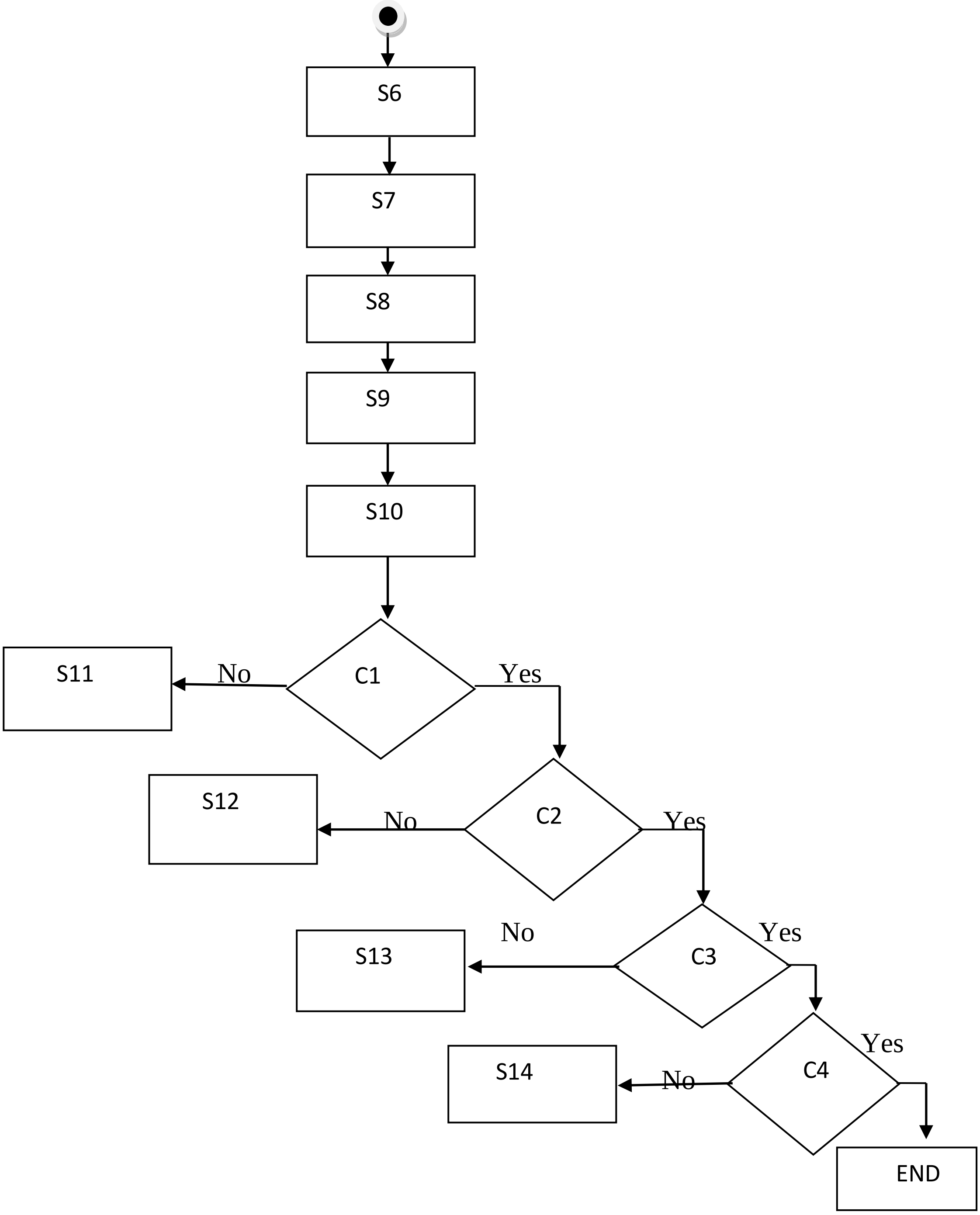


Fig 5.5 Flow Graph – Basic Path Testing

Path 1	S1-S2-S3-S4-S5-S6-S7-S8-S9-S10-C1-S11
Path 2	S1-S2-S3-S4-S5-S6-S7-S8-S9-S10-C2-S12
Path 3	S1-S2-S3-S4-S5-S6-S7-S8-S9-S10-C3-S13
Path 4	S1-S2-S3-S4-S5-S6-S7-S8-S9-S10-C4-S14

Cyclomatic complexity is a source code complexity measurement that is being correlated to a number of coding errors. It is calculated by developing a Control Flow Graph of the code that measures the number of linearly-independent paths through a program module.

Lowe the program’s cyclomatic complexity, lower the risk to modify and easier to understand. It can be represented using the below formula:

Cyclomatic complexity = E- N + P,

Where,

E = number of edges in the flow graph.

N = number of nodes in the flow graph.

P = number of nodes that have exit points.

The Cyclomatic complexity is calculated using the above control flow diagram that shows twenty nodes (shapes) and twenty one edges (lines), hence the cyclomatic complexity is derived as.

Cyclomatic complexity = 15-4+1
= 10

CHAPTER 6

CONCLUSION & FUTURE ENHANCEMENT

6.1 Conclusion:

Thus the Report builder provides an interface between User and the system. The application reads Excel book data and Database record then generate the Report from user request to the web server.

6.2 Future Enhancement

This system can further be enhanced by including the following are

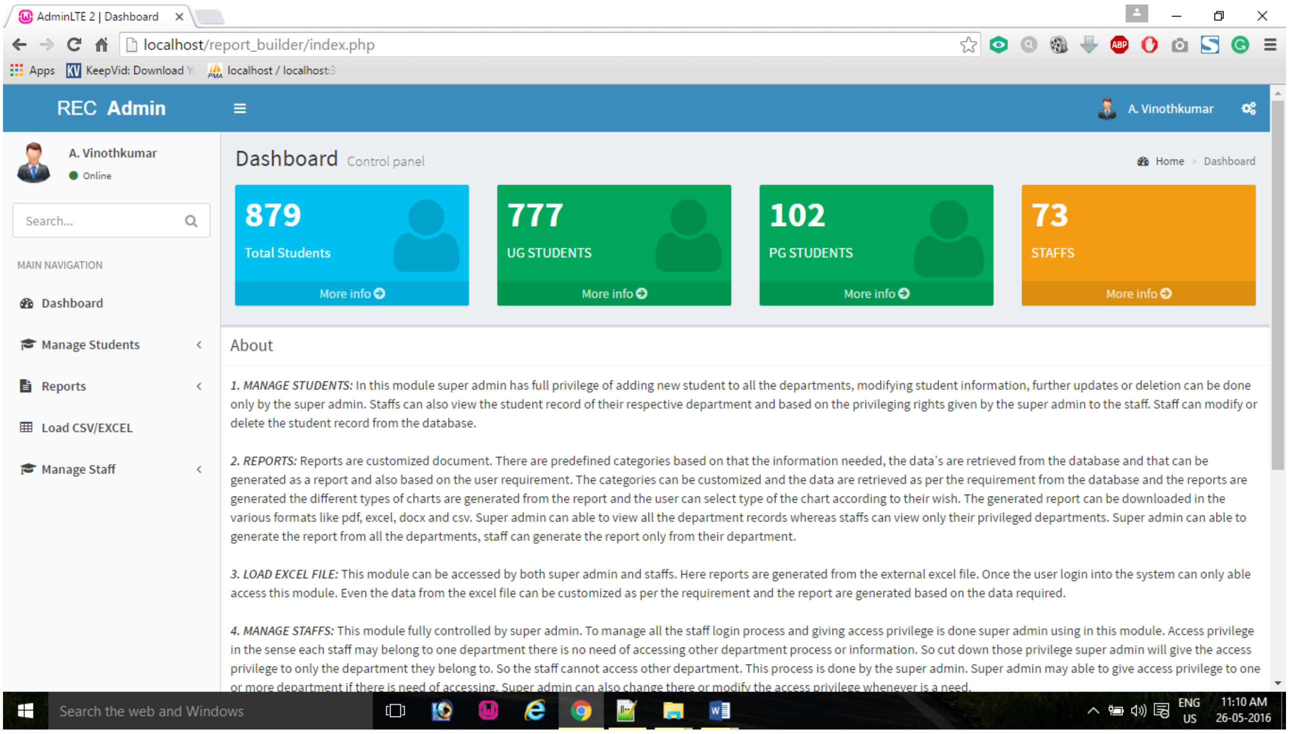
Up until now the builder is developed only for student purposes. This report builder can be extensively developed for an organization or for the whole college including staff, employees, library, and Accounts data and others can also be used in this report builder.

CHAPTER 7

APPENDICES

7.1 Sample screens

FIG 7.1.1 Index page



About

1. MANAGE STUDENTS: In this module super admin has full privilege of adding new student to all the departments, modifying student information, further updates or deletion can be done only by the super admin. Staffs can also view the student record of their respective department and based on the privileging rights given by the super admin to the staff. Staff can modify or delete the student record from the database.

2. REPORTS: Reports are customized document. There are predefined categories based on that the information needed, the data's are retrieved from the database and that can be generated as a report and also based on the user requirement. The categories can be customized and the data are retrieved as per the requirement from the database and the reports are generated the different types of charts are generated from the report and the user can select type of the chart according to their wish. The generated report can be downloaded in the various formats like pdf, excel, docx and csv. Super admin can able to view all the department records whereas staffs can view only their privileged departments. Super admin can able to generate the report from all the departments, staff can generate the report only from their department.

3. LOAD EXCEL FILE: This module can be accessed by both super admin and staffs. Here reports are generated from the external excel file. Once the user login into the system can only able access this module. Even the data from the excel file can be customized as per the requirement and the report are generated based on the data required.

4. MANAGE STAFFS: This module fully controlled by super admin. To manage all the staff login process and giving access privilege is done super admin using in this module. Access privilege in the sense each staff may belong to one department there is no need of accessing other department process or information. So cut down those privilege super admin will give the access privilege to only the department they belong to. So the staff cannot access other department. This process is done by the super admin. Super admin may able to give access privilege to one or more department if there is need of accessing. Super admin can also change there or modify the access privilege whenever is a need.

FIG 7.1.1: Index page

FIG 7.1.2 Login page

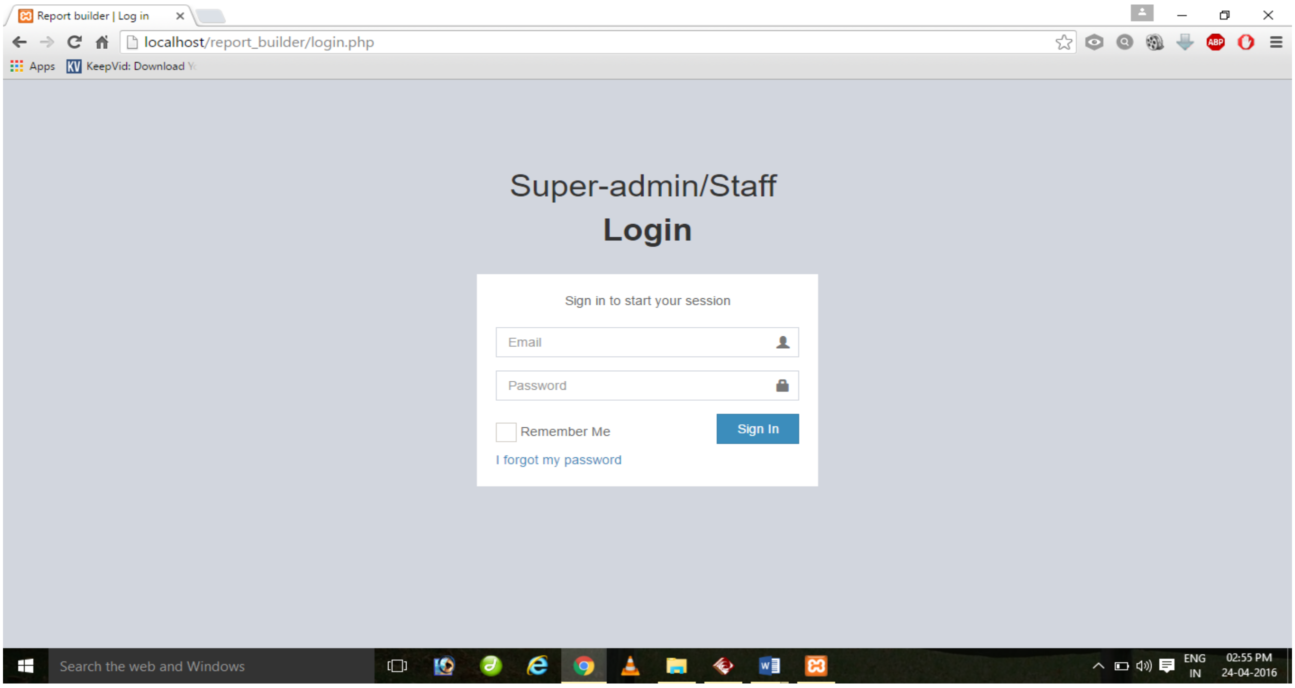
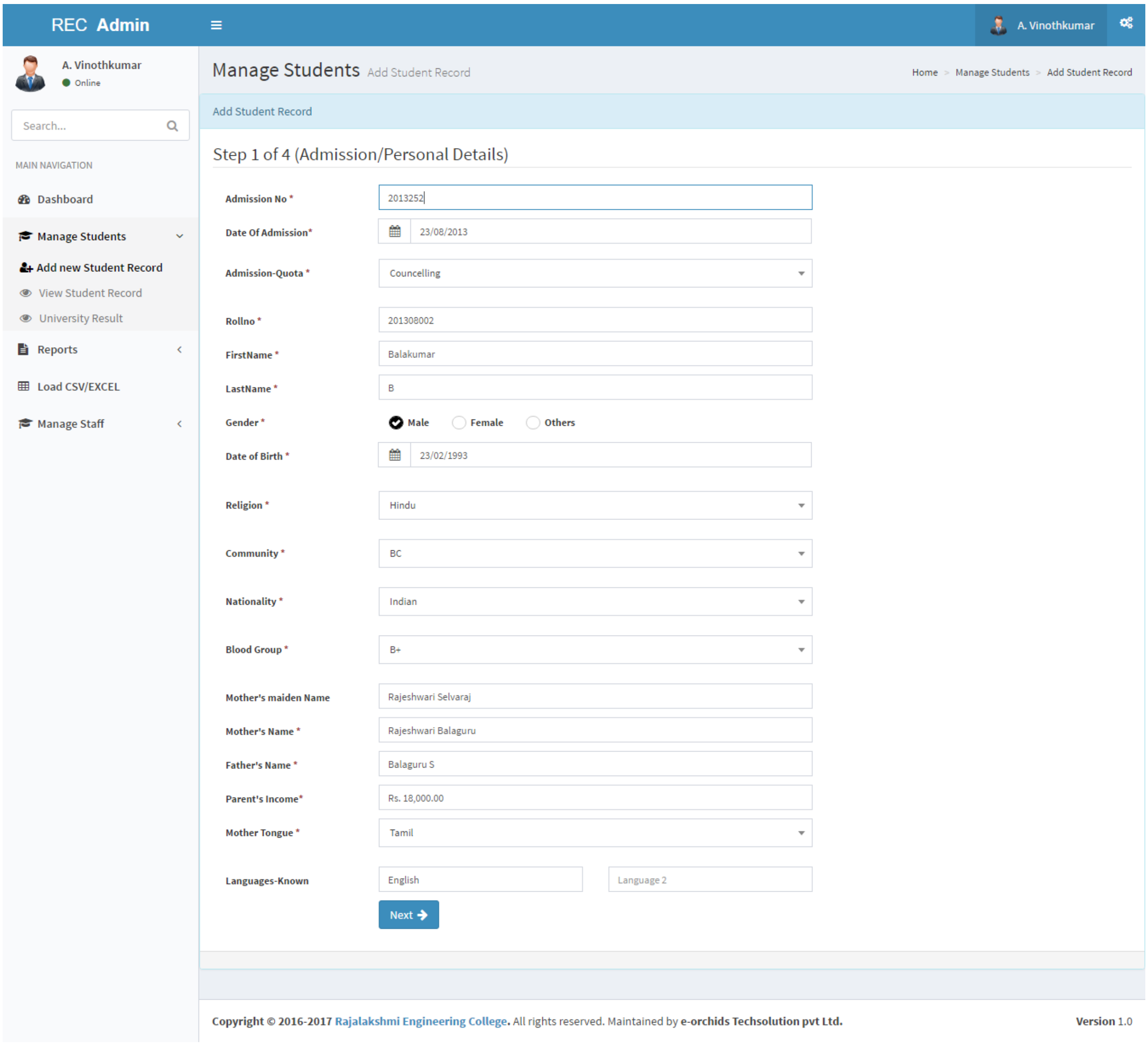


FIG 7.1.2: Login page

FIG 7.1.3 Manage student (add student record):



Manage Students Add Student Record

Add Student Record

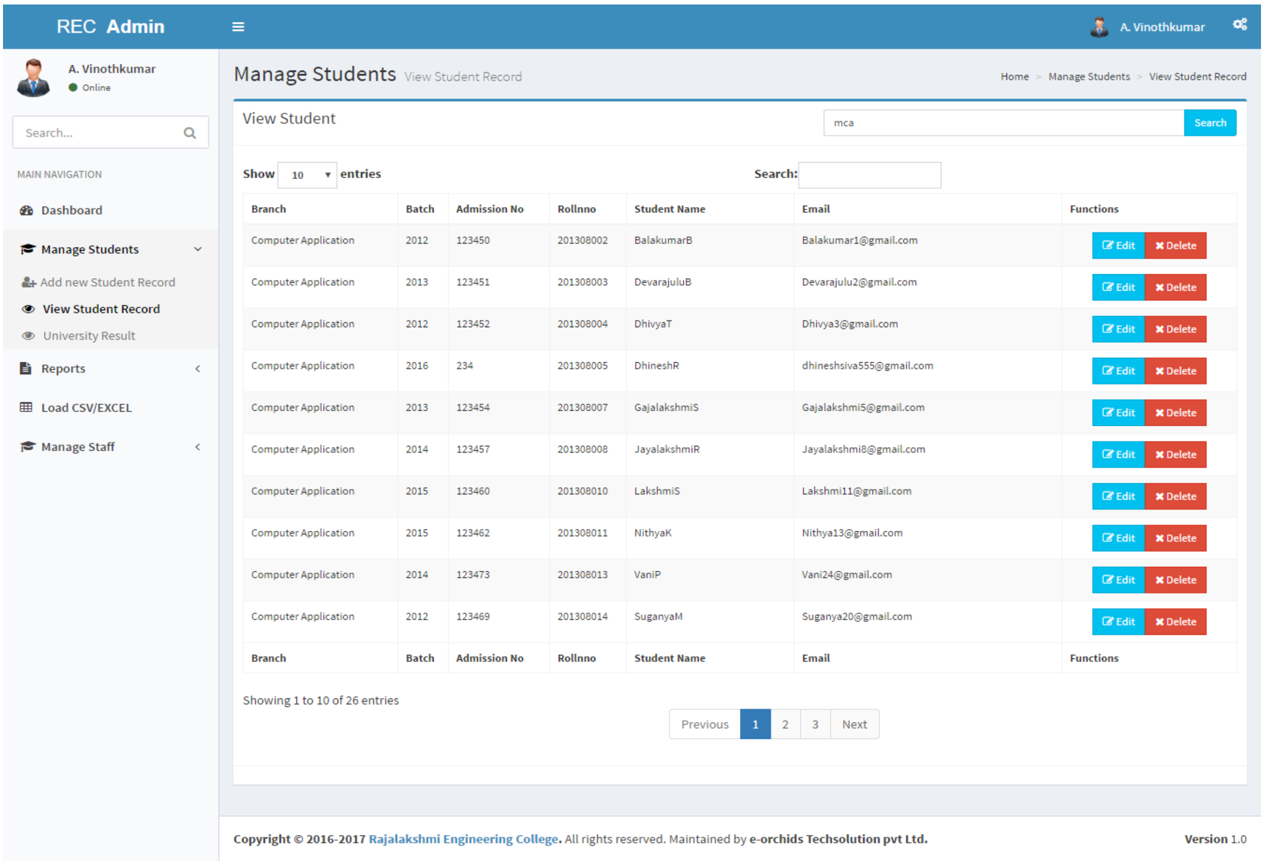
Step 1 of 4 (Admission/Personal Details)

Admission No *	2013252
Date Of Admission*	23/08/2013
Admission-Quota *	Councelling
Rollno *	201308002
FirstName *	Balakumar
LastName *	B
Gender *	● Male ○ Female ○ Others
Date of Birth *	23/02/1993
Religion *	Hindu
Community *	BC
Nationality *	Indian
Blood Group *	B+
Mother's maiden Name	Rajeshwari Selvaraj
Mother's Name *	Rajeshwari Balaguru
Father's Name *	Balaguru S
Parent's Income*	Rs. 18,000.00
Mother Tongue *	Tamil
Languages-Known	English Language 2

Copyright © 2016-2017 Rajalakshmi Engineering College. All rights reserved. Maintained by e-orchids Techsolution pvt Ltd. Version 1.0

FIG 7.1.3: Manage students (add student record)

FIG 7.1.4 View student:



Manage Students View Student Record

View Student

mca

Show 10 entries Search:

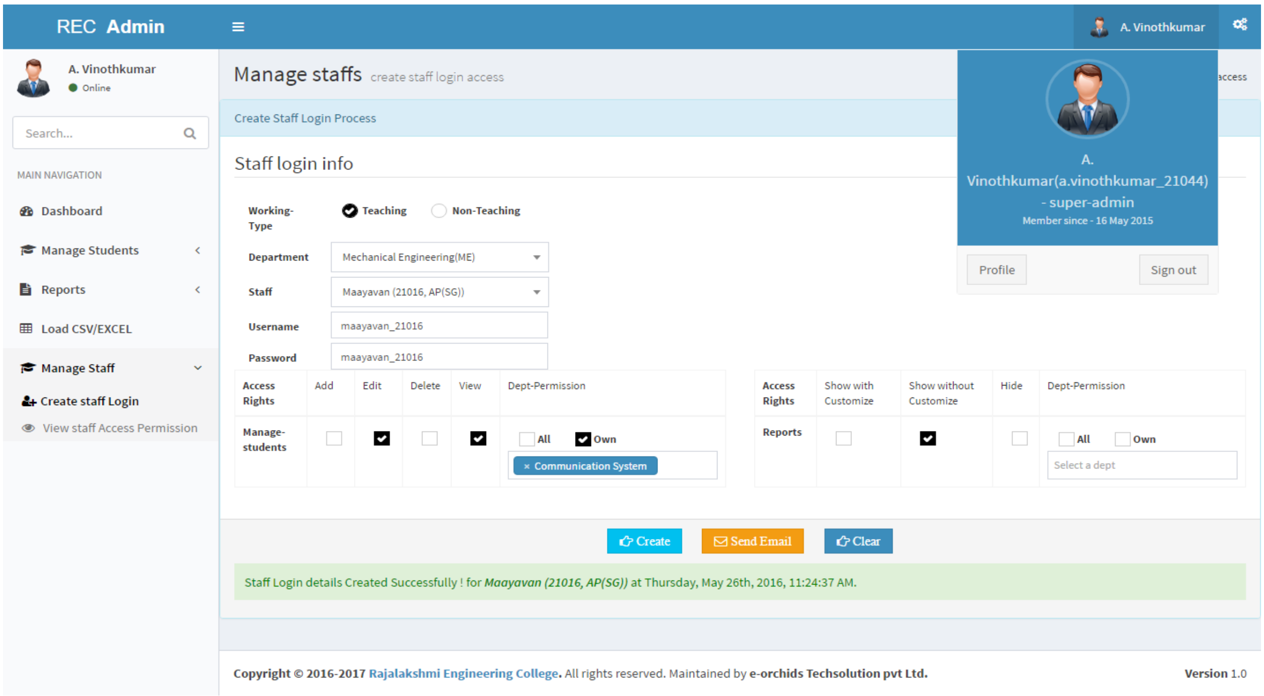
Branch	Batch	Admission No	Rollno	Student Name	Email	Functions
Computer Application	2012	123450	201308002	BalakumarB	Balakumar1@gmail.com	Edit Delete
Computer Application	2013	123451	201308003	DevarajuluB	Devarajulu2@gmail.com	Edit Delete
Computer Application	2012	123452	201308004	DhivyaT	Dhivya3@gmail.com	Edit Delete
Computer Application	2016	234	201308005	DhineshR	dhineshsiva555@gmail.com	Edit Delete
Computer Application	2013	123454	201308007	GajalakshmiS	Gajalakshmi5@gmail.com	Edit Delete
Computer Application	2014	123457	201308008	JayalakshmiR	Jayalakshmi8@gmail.com	Edit Delete
Computer Application	2015	123460	201308010	LakshmiS	Lakshmi11@gmail.com	Edit Delete
Computer Application	2015	123462	201308011	NithyaK	Nithya13@gmail.com	Edit Delete
Computer Application	2014	123473	201308013	VaniP	Vani24@gmail.com	Edit Delete
Computer Application	2012	123469	201308014	SuganyaM	Suganya20@gmail.com	Edit Delete
Branch	Batch	Admission No	Rollno	Student Name	Email	Functions

Showing 1 to 10 of 26 entries

Copyright © 2016-2017 Rajalakshmi Engineering College. All rights reserved. Maintained by e-orchids Techsolution pvt Ltd. Version 1.0

FIG 7.1.4: View student

FIG 7.1.5 Create Staff Login:



Manage staffs create staff login access

Create Staff Login Process

Staff login info

Working-Type ● Teaching ○ Non-Teaching

Department: Mechanical Engineering(ME)

Staff: Maayavan (21016, AP(SG))

Username: maayavan_21016

Password: maayavan_21016

Access Rights	Add	Edit	Delete	View	Dept-Permission
Manage-students	□	☑	□	☑	□ All ☑ Own × Communication System

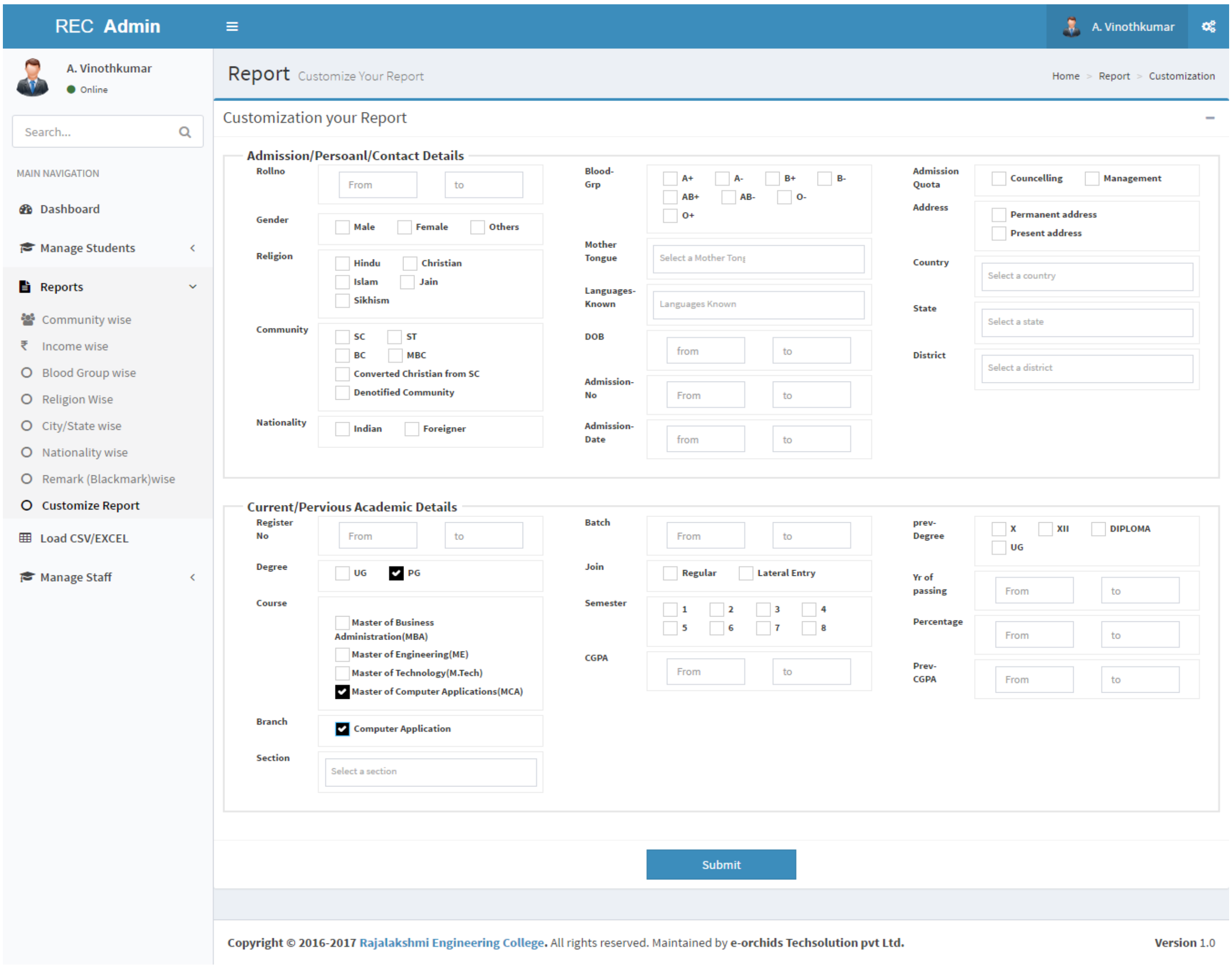
Access Rights	Show with Customize	Show without Customize	Hide	Dept-Permission
Reports	□	☑	□	□ All □ Own Select a dept

Staff Login details Created Successfully ! for *Maayavan (21016, AP(SG))* at Thursday, May 26th, 2016, 11:24:37 AM.

Copyright © 2016-2017 Rajalakshmi Engineering College. All rights reserved. Maintained by e-orchids Techsolution pvt Ltd. Version 1.0

FIG 7.1.4: Create staff Login

FIG 7.1.6 Customize Report:



Report Customize Your Report

Customization your Report

Admission/Persoanl/Contact Details

Rollno: From / to

Gender: □ Male □ Female □ Others

Religion: □ Hindu □ Christian □ Islam □ Jain □ Sikhism

Community: □ SC □ ST □ BC □ MBC □ Converted Christian from SC □ Denotified Community

Nationality: □ Indian □ Foreigner

Blood-Grp: □ A+ □ A- □ B+ □ B- □ AB+ □ AB- □ O- □ O+

Mother Tongue: Select a Mother Tong

Languages-Known: Languages Known

DOB: from / to

Admission-No: From / to

Admission-Date: from / to

Admission Quota: □ Councelling □ Management

Address: □ Permanent address □ Present address

Country: Select a country

State: Select a state

District: Select a district

Current/Pervious Academic Details

Register No: From / to

Degree: □ UG ☑ PG

Course: □ Master of Business Administration(MBA) □ Master of Engineering(ME) □ Master of Technology(M.Tech) ☑ Master of Computer Applications(MCA)

Branch: ☑ Computer Application

Section: Select a section

Batch: From / to

Join: □ Regular □ Lateral Entry

Semester: □ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7 □ 8

CGPA: From / to

prev-Degree: □ X □ XII □ DIPLOMA □ UG

Yr of passing: From / to

Percentage: From / to

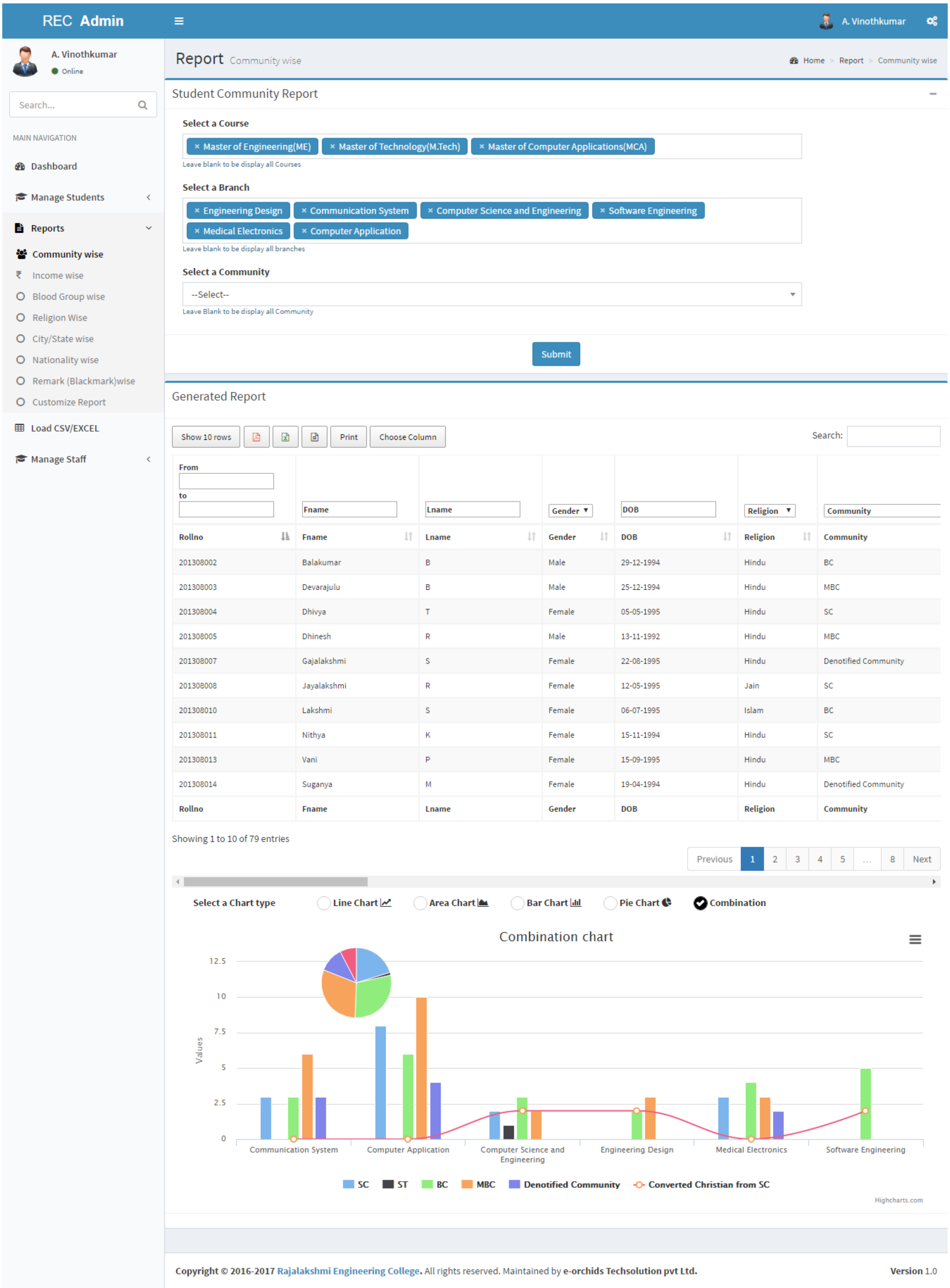
Prev-CGPA: From / to

Submit

Copyright © 2016-2017 Rajalakshmi Engineering College. All rights reserved. Maintained by e-orchids Techsolution pvt Ltd. Version 1.0

FIG 7.1.6: Customize report

FIG 7.1.7 Reports:



REC Admin ≡ A. Vinothkumar

A. Vinothkumar
● Online

Search...

MAIN NAVIGATION

- Dashboard
- Manage Students
- Reports
  - Community wise
  - Income wise
  - Blood Group wise
  - Religion Wise
  - City/State wise
  - Nationality wise
  - Remark (Blackmark)wise
  - Customize Report
- Load CSV/EXCEL
- Manage Staff

Report Community wise

Home > Report > Community wise

Student Community Report

Select a Course

× Master of Engineering(ME) × Master of Technology(M.Tech) × Master of Computer Applications(MCA)

Leave blank to be display all Courses

Select a Branch

× Engineering Design × Communication System × Computer Science and Engineering × Software Engineering × Medical Electronics × Computer Application

Leave blank to be display all branches

Select a Community

--Select--

Leave Blank to be display all Community

Submit

Generated Report

Show 10 rows | Print | Choose Column

Search:

From

to

Rollno	Fname	Lname	Gender	DOB	Religion	Community
201308002	Balakumar	B	Male	29-12-1994	Hindu	BC
201308003	Devarajulu	B	Male	25-12-1994	Hindu	MBC
201308004	Dhivya	T	Female	05-05-1995	Hindu	SC
201308005	Dhinesh	R	Male	13-11-1992	Hindu	MBC
201308007	Gajalakshmi	S	Female	22-08-1995	Hindu	Denotified Community
201308008	Jayalakshmi	R	Female	12-05-1995	Jain	SC
201308010	Lakshmi	S	Female	06-07-1995	Islam	BC
201308011	Nithya	K	Female	15-11-1994	Hindu	SC
201308013	Vani	P	Female	15-09-1995	Hindu	MBC
201308014	Suganya	M	Female	19-04-1994	Hindu	Denotified Community
Rollno	Fname	Lname	Gender	DOB	Religion	Community

Showing 1 to 10 of 79 entries

Previous 1 2 3 4 5 ... 8 Next

Select a Chart type ○ Line Chart ○ Area Chart ○ Bar Chart ○ Pie Chart ● Combination

Combination chart

Copyright © 2016-2017 Rajalakshmi Engineering College. All rights reserved. Maintained by e-orchids Techsolution pvt Ltd.

Version 1.0

FIG 7.1.6: Reports

CHAPTER 8

REFERENCES

Peter Shaw (2014). *Twitter Bootstrap 3 Succinctly*. USA: Syncfusion.inc.

Cody Lindley (2012). *jQuery Succinctly*. USA: Syncfusion.inc.

Ulf Eriksson (31 October 2012). *Learn How To Write Effective Test Cases*.

www.stackoverflow.com

http://api.highcharts.com

www.getbootstrap.com

http://www.jqueryrain.com

https://www.mysql.com