

**Table of Contents**

1. INTRODUCTION .....	1
1.1    Aim .....	1
1.2    Objectives .....	1
1.3    Scope.....	2
1.4    Types of Users .....	3
1.5    Benefits .....	3
1.6    Gaining Skills.....	4
1.7    Gantt chart- Campus Recruitment System.....	5
2. BACKGROUND STUDY .....	6
2.1    Actual System .....	6
2.2    Initial Investigation .....	6
2.3    Methods used for Initial Investigation .....	6
2.4    Interview- Open-ended questions .....	7
2.5    Observation .....	8
2.6    Document Inspection .....	8
2.7    Actual system- How recruitment are being done by companies.....	8
2.8    Actual system- How recruitment of students are being performed. ....	9
2.9    Use Case Diagram- Actual System.....	10
3 ANALYSIS.....	12
3.1    Actual system problem .....	12
3.2    Proposed System.....	14
3.3    Requirement of the proposed solution .....	15
3.4    Use case- Proposed solution .....	16
3.5    System development methodology .....	19

3.6	Hardware Requirement .....	22
3.7	Software Requirement .....	22
3.8	Feasibility Study .....	23
3.9	Cost Analysis .....	24
3.10	Conclusion- justify the worthiness of continuing with the project .....	24

### **List of Figures**

Figure 1-	Gantt Chart .....	5
Figure 2-	Use case diagram: Actual System (I): UDM Recruitment System .....	10
Figure 3-	Use case diagram: Actual System (II): Companies Recruitment System ....	11
Figure 5-	Use case after recruitment of students: Proposed Solution .....	16
Figure 6-	Use case Company recruit student via campus recruitment system: Proposed Solution .....	17
Figure 7-	Use Case Student apply to company via campus recruitment system: proposed Solution .....	18
Figure 8:-	Scrum Process Flow .....	20

### **List of Table**

Table 1-Minimum	Hardware Requirement.....	22
Table 2-	Chosen Hardware Requirement.....	22
Table 3:-	Minimum Software Requirement.....	22
Table 4-	Chosen Software Requirement .....	23
Table 5-	Feasibility Study .....	23

## 1. INTRODUCTION

Campus Recruitment System is to replace the manual operation of recruitment of the companies at Université Des Mascareignes and also will provide a platform for students to create their own identity. As recruitment happened on a yearly basics where thousands of students get employed and being recruited at the campus. Students will be able to create their own status and even be able to send their CV to the companies and companies will send be able to select the best candidate and even select students without receiving any CV from students. As soon as candidates have been selected then the interview schedule will be send by mail. Communication will be done by chat or mail. The system will be able to produce queries and managed the database during recruitment process. Recruitment is refers as to make someone be a new member of the organization or institution.

### 1.1 Aim

The aim of this project is to provide a platform to create their own identity. In broad terms there are two main parts to it – university and companies. University consists of students and lecturers where they will be able to create their own identity and to be able to be hire by companies easily. Ease the process of recruitment that are being performed by companies on the campus and give them the facilities to search for specific students. And to create a good communication channel between the university and companies.

### 1.2 Objectives

- To provide services for easy recruitment and Job seekers through the website.
- They can update their profile and express their experiences and skills.
- Design and develop user interface with the consents of the Campus
- Care about the security of their data.
- The users are provided with various job details and also specified job they are eligible for according to their skills and modules taught at university.
- To facilitate the communication between users by mail.
- To allow users to create their own identity and use them for their benefits.
- Allow students and lecturers to search different companies and job prospects.

- Complete components of the newly website by 17<sup>th</sup> March 2017.

### 1.3 Scope

#### In -scope

1. User interface – The user's interface of the website will be developed in this project.
2. Front end – Front end coding will be implemented to give a dynamic feel for the new website and make life of the users more professional and to distinguish between the current system.
3. Back end - Back end coding is necessary for connection to the database where all the data will be stored and to retrieve users information and build up the interfaces the users will see.
4. Database – A database will hold all the information about the users and displayed in the website. The data are being kept in tables where appropriate and should confirm about the structure of the database based upon investigation.
5. Administrative interface – To be able to register new users and manage their information. Able to locate appropriate access right to different users. The administrator interface will be developed within the scope of the project.
6. Website testing – The ensure the validation of the data entered, whether the website is functioning well and all the data displayed are being well displayed and test about its security and components and other features.
7. Security – To be able to introduced first level of security to secure the database and all its data.

#### Out-Scope

1. Mail implementation – Due to little time the website will not be able to perform direct email to the company or another person via the website. Will take into consideration afterwards.
2. Chat log – Due to time constraint the website will not be able to contain the chat log where everyone can chat at the same time. It might be introduced if the project is completed before its time limits.

## 1.4 Types of Users

- 1 Administrator: Administrator will registered users on the website and any update needed will be performed by the administrator.
- 2 Students: Students will login into their accounts and update their data, even view companies, view lecturer details and even send email to lecturers, companies.
- 3 Lecturers: Lecturers will login into their accounts and update their data and even view companies and student's details and send email to students and companies.
- 4 Companies: Companies will be able to register themselves on the website and search for students and lectures and send mails to students, teaching Staff.

## 1.5 Benefits

- Campus Recruitment System will help students to communicate easily with lecturers and other students and vice-versa. No need to log in on mail and search for the lectures or student mail address and send a mail. All these can be done on the system itself.
- Campus Recruitment System provides a platform to select qualified students for different job sectors. That will help the employers to save lots of time in recruitment. The new talented generation are ready to be on the job market for new challenges. So, the most challenging task will be given to these new talented students. Students can also search for the companies that they want to apply to accept the new challenge according to their skills and interest. That will help to save lots of time with job hunting.
- Campus Recruitment System is cost efficient as it saves lot of time for employers. The Recruitment teams need to visit university and spend some days to get a short list hundreds of students. The system help them to recruit on the system directly. No need to go to different places and invest lots of money in travelling, advertisements, screening, and perform different steps so that at the end to select the best candidate and it is time consuming.

- The companies conducting campus recruitment in a specific university establish strong relationship with the institute and its students. The students will prefer to train and conduct their internship in these companies. Therefore that build up a strong bond for following year's recruitment.
- There will be no need of putting up notice or emailing every student about the company coming in college. The students can search for the companies on the system itself.

## 1.6 Gaining Skills

- ✓ Human Computer Interface
- ✓ Database planning and management
- ✓ Management WAMPP
- ✓ Photoshop/ Illustrator/ Fireworks
- ✓ UML case / Data Flow Diagram (DFD)
- ✓ Object Oriented Design
- ✓ Testing
- ✓ Programming
- ✓ Introductory Level of Security

## 1.7 Gantt chart- Campus Recruitment System

Figure 1 below shows the timeline that will be used to develop the system. This chart was devised on “Creately”. [1]

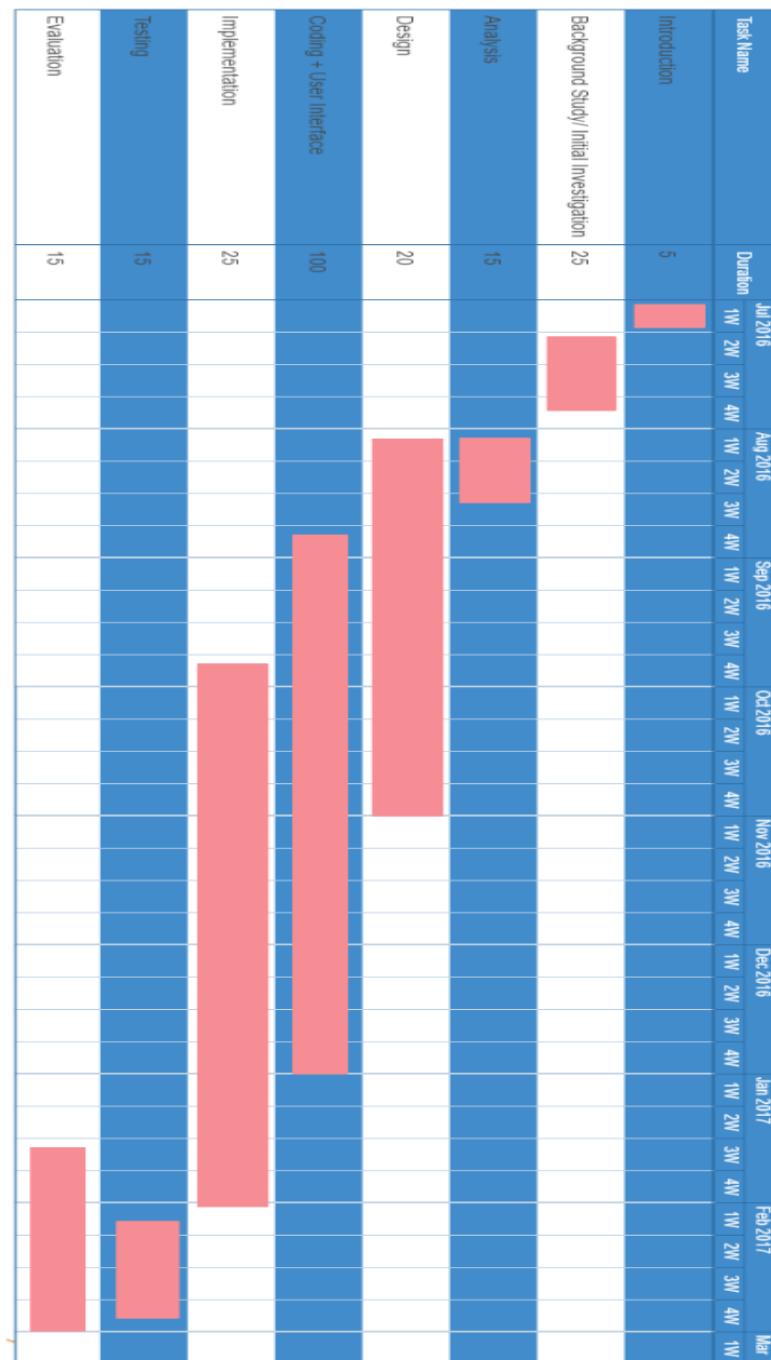


Figure 1- Gantt Chart

## 2. BACKGROUND STUDY

Université Des Mascareignes is founded in 2012 after merging between the IST (Institut Supérieur de Technologie) of Camp Levieux and Swami Dayanand Institute of Management (SDIM) of Pamplemousses and it is affiliated with Université de Limoges. Students earn two degrees. The first one from Université Des Mascareignes and second from Université de Limoges, it is a French degree.

### 2.1 Actual System

The existing system allow companies to recruit students on-site and students has to visit companies or search for companies address to be able to send CV's. There is no record about the student's qualities on the existing system and no contacts and relationship between companies and Universities.

### 2.2 Initial Investigation

Initial investigation is used to identify problems, and suggest idea to improve the current system.

Initial Investigation is needed to be carried out so as to understand the existing system and then decided rather to propose solutions or to improve the actual system.

### 2.3 Methods used for Initial Investigation

After a short review on the different methods (Interview, Questionnaire, Personal Observation, Documents Inspection, and Storytelling) available for initial Investigation, therefore I found document inspection and Interview and observation to be more efficient and effective to be used for the investigation process. Moreover, as there are different methods of interview I found that the unstructured interview is more appropriate to be used where it contains open-ended questions and observation so that more information can be collected and can added more questions if required.

## 2.4 Interview- Open-ended questions

All the questions that has been asked during the interview are listed below:

1. How the actual system at the campus operates for recruitment period?
2. How the process of recruitment by companies are performed?
3. What are the duration for Job placement?
4. How is the communication channel between students and lecturers?
5. How the actual system can be improved?

After conducting the interview with different interviewee and all the answers I received are shown below in a summaries way as every interviewee answer almost the same.

*Answers:*

1. Every year during April, Université Des Mascareignes recruits students who are school leavers or top up or Master to pursue further studies. Students gets their application form, fill it and return it to the registration office with all the necessary documents been attached to it. And then they wait for the reply if they have been qualified to get a seat at Université Des Mascareignes.
2. Every year companies visit Université Des Mascareignes for advertising about their companies so that to receive students to perform their internship in their company. Students prepare their Curriculum Vitae, Motivation letter and post it to the companies they want to perform their internship and even send Co-operate mail to the companies. Students get a job placement sheet along with a progress sheet during job placement. The progress sheet should be return when their internship is over.
3. Duration for all faculties about internship are the approximately the same amount of weeks that is 10 weeks of internship should be covered.
4. Communication is poor between lecturers and students and even between members and students of the University.
5. As we are in the new generation where there are massive improvement of technology I would like that everyone can have their own profile that will them to develop themselves and share their skills. For instance, LinkedIn, Blog.

## 2.5 Observation

Observation are being done when the analysts is present on the spot when the task are being carried out. There are two types of observations:

- *Active Observation:* It has been done when Université Des Mascareignes is recruiting students and while companies are recruiting students for internship
- *Passive Observation:* Observation about how the system operates.

## 2.6 Document Inspection

After having done with the interview, active observation and document inspection for collecting information about how the actual system works therefore a summaries of the information are narrated in the following.

## 2.7 Actual system- How recruitment are being done by companies.

Almost every January there are companies who visit our university to advertise their company and explain about their companies and what the sectors available in and the target of the companies and how it functions. In the 6<sup>th</sup> Semester students start searching for companies information and details and companies starts searching for new fresh youngsters to work for their enterprise. When students already get the companies details and info, they prepare their Curriculum Vitae, Portfolio, Motivation letter so that to post it to the company, email the company and even go and drop it themselves at the companies' office. After that they wait for the companies answer. There are some gets the work they needed, some no. There are students that did not even get reply from the companies they have applied but fortunately we have lecturers that help so that everyone get a job placement. Unfortunately, there are students that do get the post they have applied for but another post where they are less performant. Students and companies have not enough information about each other and poor communication between students and companies. Moreover, students face lots of problem for searching of details and information's about companies as well as companies also as they have trouble to search for the specific students. Once students has already got their job placement, then they are given a progress sheet for their industrial attachment and after their terms of job placement ends they have to submit a report. Once over with the job placement phase then less communication between the campus and the companies.

## **2.8 Actual system- How recruitment of students are being performed.**

There are lots of events and preparation takes place before, during and after the recruitment process. Before recruitment of students adverts are being made on social media and newspaper as well as promoting them each days on different social media site. Where it is mentioned about where the open day event will take place and when. During the even the students will get all information about the courses that are being offered, the price and qualification needed to be enroll in the university. Moreover, there are lecturers that help by giving more information about different modules and job prospect that are currently available on the market. The visitors' even gets the opportunity to visit the campus and the registration form are being given to them where all the information needed and other documents that will be needed to be able to complete the registration phase. The registration are open for a period of time approximately one month. After the registration period is over then the selection of students are being done upon their qualification of Higher School Certificate and if they are eligible for the course that they have been applied. Moreover a fees of registration is required. After the registration phase is over. All the selected students received a letter from the university, where it is mentioned the details fees of the course and the National Transport Authority fee (Bus Pass) and the date when the course will start and other details. Once the students already paid for the course, their recorded are just stored on the system on a sheet of excel where there is only the name of the students, address, group of faculty and amount paid for the course. There are no update anything about the students, not even a file about the student skills, experience and more information about the students. Moreover, the students received their university personal mail address after many weeks later that's is really bad for the campus. Not enough information about students and poor communication system.

## 2.9 Use Case Diagram- Actual System

Figure 2 below shows the use case diagram about the recruitment process of Université Des Mascareignes. [2]

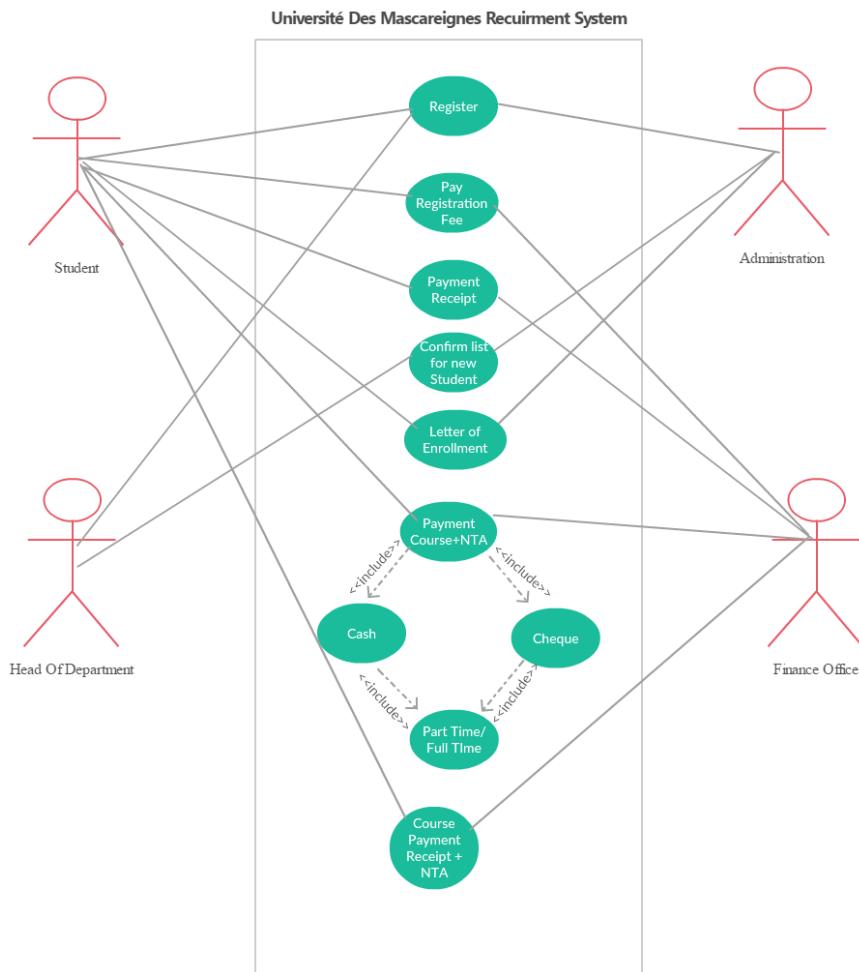


Figure 2- Use case diagram: Actual System (I): UDM Recruitment System

Figure 3 below shows the use case diagram about the recruitment process of student by companies for internship. [2]

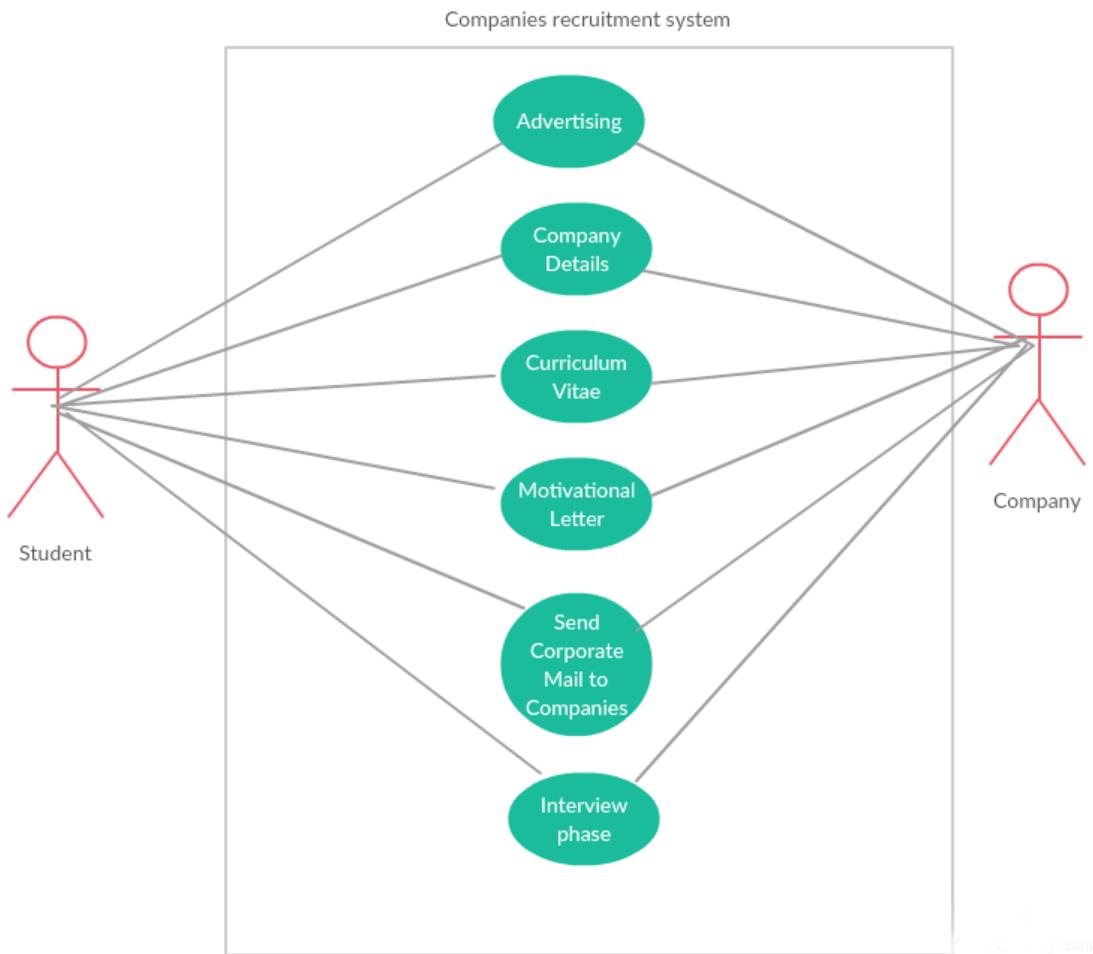


Figure 3- Use case diagram: Actual System (II): Companies Recruitment System

## 3 ANALYSIS

After the initial investigation and analyzing the information about the actual system that has been collected during the initial investigation phase therefore this phase will contain the followings:

- Actual System problem
- Proposed solution
- Requirement of the proposed system
- Context Diagram for proposed solution
- Use Case for proposed solution
- System Development Methodology
- Hardware Requirement
- Software Requirement
- Feasibility Study
- Cost benefit analysis

### 3.1 Actual system problem

After the initial investigation phase, the problem that the campus is facing are being describe as follows:

- There are a lots of issues concerning communication and personalities between lecturers, students, administration and companies. First and foremost, lecturers and students does not have an identities and is not known by anyone. Students, nobody knows their skills, experiences, capabilities, awards, hobbies, passions what they want to do, talents and other potentials. Lecturers, students does not have the enough information about the lecturers to know them well. They are not even aware of the lecturer's experience, skills. Talents, their most life challenging moment (moment can inspire any one to think and take wise decision), conference they attend and other potentials. Students are not well informed about their lecturers and that leads to a confusion when has to ask help from lecturers for any assignment or even help. Lecturers could not communicate well to their students about important issues or any urgent issues

and vice-versa. Furthermore, Students are not informed about university term scheduled and holidays.

- Lots of pain while searching for lecturers or students university email address to send a mail or even ask for an information. Students have to go online search for potential companies where they can get a chance for an internship or search via newspapers any email address or telephone number where they could send their CV's and personal information. Then waiting for reply couple of weeks and they used lots of their time to search for companies email address or even deposit their CV's manually at the companies' office. Moreover, students are not well informed about the potential companies that are available on the market and the skills that these companies required so that to be able to get an interview and even not aware about the company's history, way of work and in which fields they are recruiting members.
- Even companies have lots of trouble to search for potential candidate for internship at their places and they even go campus to campus for onsite recruitment therefore that cause lots of problem for the companies in terms of expenses and time. Their lots of information that are being escape from students for about university activities or meeting and even not a proper channel of communication between administration and lecturers and students.
- Progress records and IA sheet can be lost or damaged.

### 3.2 Proposed System

- This project Campus Recruitment System is an online website for the Université Des Mascareignes where companies have already been registered by UDM administration so that to be able to get an overview of the campus system and all its modules that are being taught at the University. Moreover, companies will be able to search for student who match their qualification and select them for an interview.
- The Campus Recruitment System facilitates the user's task as they can directly upload their CV's and apply for jobs that available in the companies. The System also give the companies the right to view the students and teaching staff profiles of interested candidates. Currently recruitment are being done manually and it was time consuming but now recruitment will be done within seconds.
- The Campus Recruitment System will allow teaching staff and students to view companies' requirement and also allow them to send mail for more information about the companies. That will avoid the use of searching the companies' mail on the internet or brochure.
- It is also possible for companies to offer any students or staff a job proposal based on their qualification, interest, skill and experience that are being shown up on their profile.
- The Campus Recruitment System will facilitate the communication channel between lectures, students and Companies. Communication will be divided into two level:
  - Email
- The Campus Recruitment System will enable every students, lecturers to have a platform where they can share their skill, experience and talents.

### 3.3 Requirement of the proposed solution

The Requirements of the proposed system is divided into two parts:

- Functional Requirement
- Non-Functional Requirement

#### *Functional Requirement*

The main functionalities of the system are:

1. The proposed system will allows users to describe their personalities and skills.
2. The proposed system will allows users to send CV and emails to companies directly.
3. The proposed system will allows companies to recruit students directly.
4. The proposed system will allows students to communicate via mail.
5. Administrator of the system will have full access rights to add, delete, modify and records on the system.
6. The proposed system will allowed company and lecturer to download CV from students.
7. Every details should be able to customized. Minimized Hardcore code.
8. A personal calendar for all users.
9. A blog or online portfolio for each lecturer and student.

#### *Non-Functional Requirement*

The non-functional requirement are as follows:

1. Availability

The proposed system should be available on a 24-hour basis and 7/7. So that students can update their records and companies can view profile of students at any time.

2. Capacity

The proposed system will have a sufficient space to store large volume of information. No problem of memory or data loss will occur due to memory overload.

### 3. Performance

The proposed system will be a performant one as the details will be up-to-date and few waiting time for processing of data and query about information's. Response time will be quickly.

#### 3.4 Use case- Proposed solution

Figure 6 shows the use case of the proposed system after the recruitment of students[2]

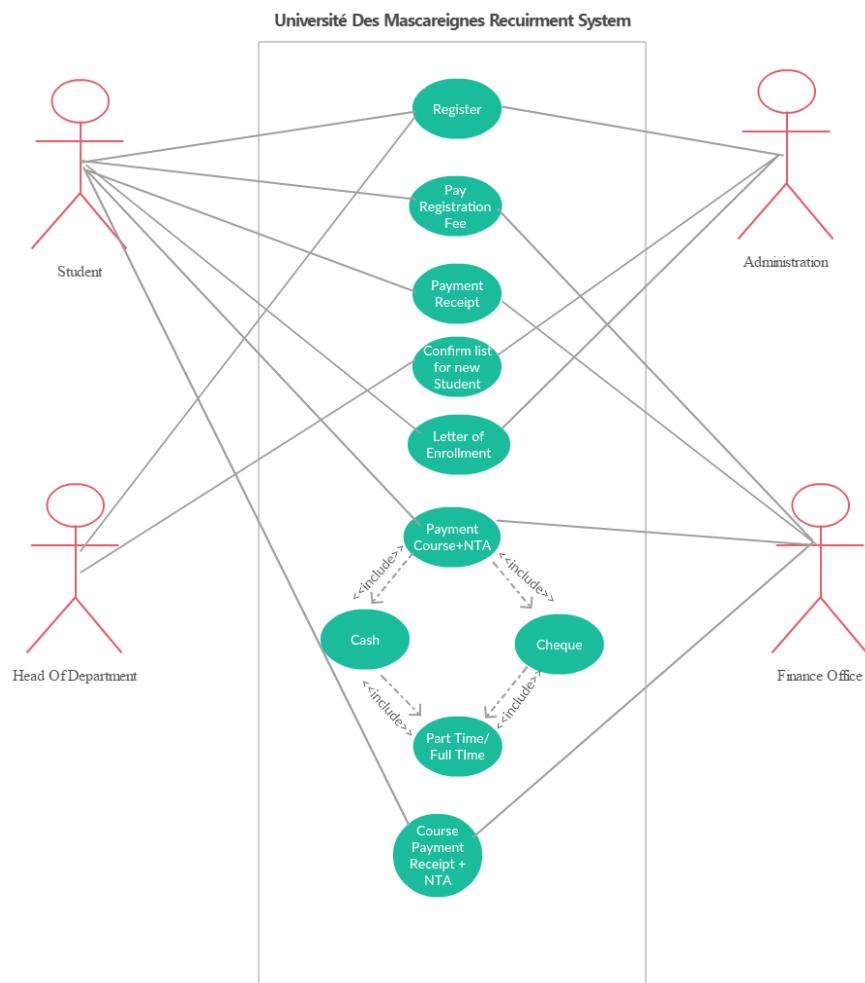


Figure 4-Use case after recruitment of students: Proposed Solution

Figure 7 shows the use case of the proposed system for the recruitment of students by companies [2]

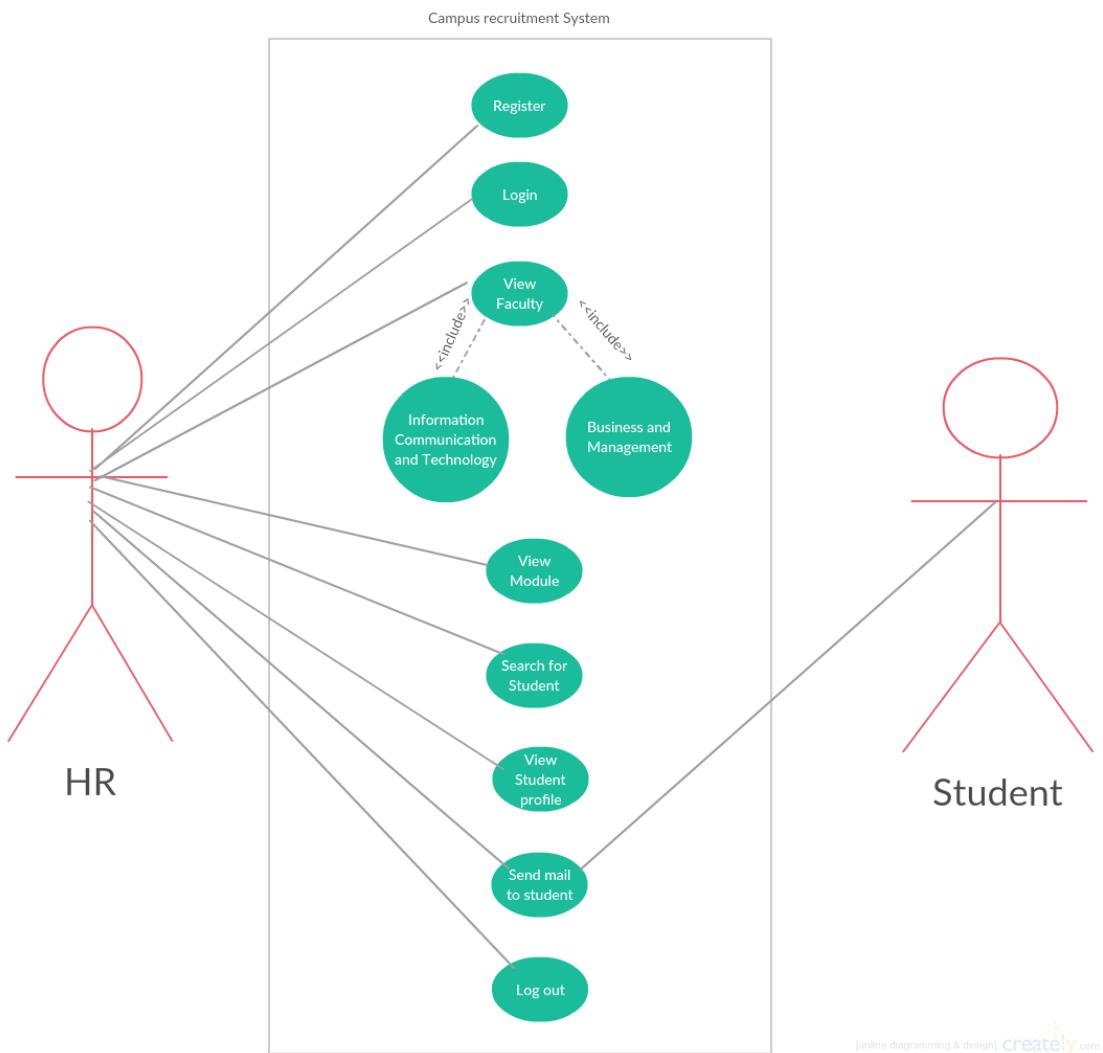


Figure 5- Use case Company recruit student via campus recruitment system: Proposed Solution

Figure 8 shows the use case of the proposed system where student search for company and apply. [2]

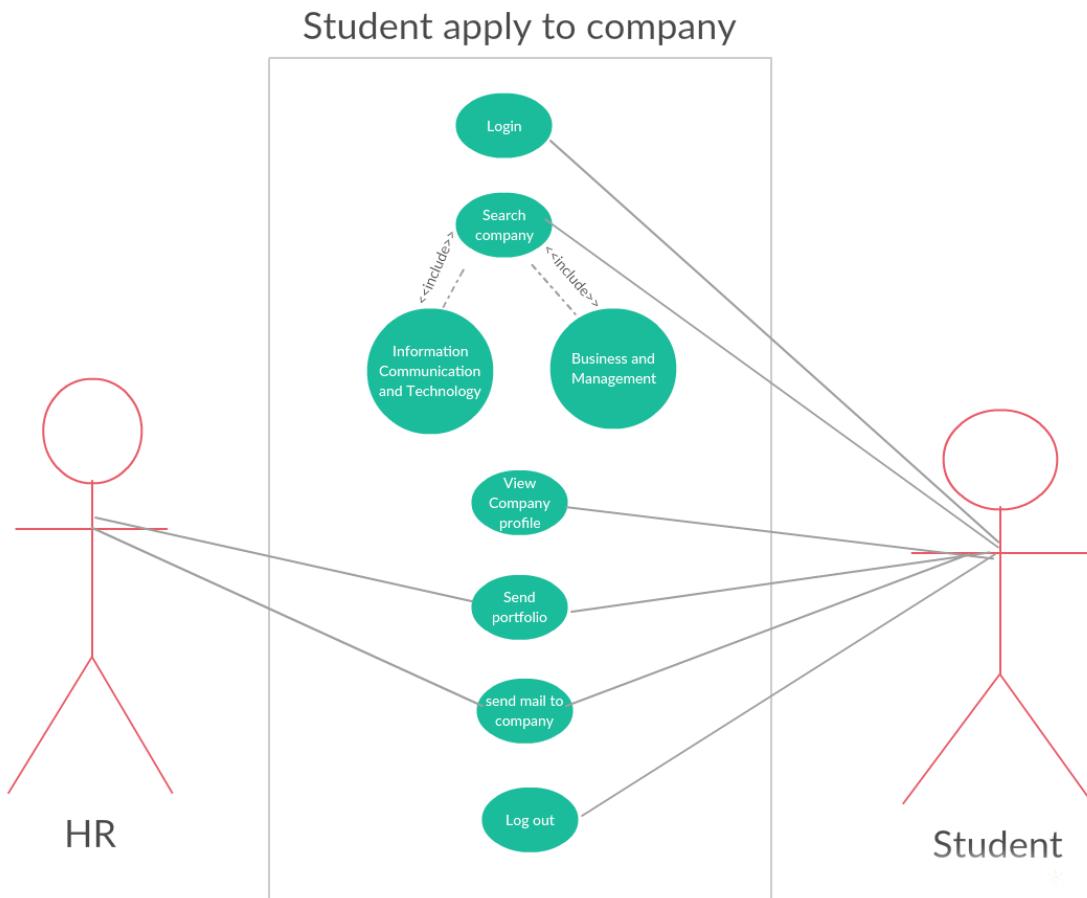


Figure 6- Use Case Student apply to company via campus recruitment system: proposed Solution

### 3.5 System development methodology

It is important to employ a methodology to ensure an actual thought process is going on in the development of the software program and not just a straight jump from having an idea to writing the code. Software engineering involves much more than just coding.

System development methodologies like waterfall model, Prototyping, Incremental, Spiral, Rapid Application Development (RAD) and Agile have become a key tool for software project managers as it aids them in delivering software products that meet requirements set out by the client. Such methodologies also help in delivering the product on time.

Each methodology has its strengths and weaknesses, thus it must be taken up by a project with careful consideration as to whether the flow of events dictated by the methodology will be useful or required. After review the different types of System development methodology, the agile software development framework will be implemented, specifically the Scrum development methodology.

Scrum has been chosen for many beneficial reasons. It allows the project to be focused around completing the requirements gathered from the client using appropriate requirement specification techniques. This is done by grouping together a set of requirements that the project team know they can fulfil, implementing the design and development of that set of requirements in a time limited ‘sprint’, and producing a working result that corresponds to the requirements initially set out for that sprint. Continuous sprints are done until the project is completed (i.e. product is complete, customer is satisfied, software is ready for release).

One of the main reasons for choosing the Scrum methodology is that it confines you in terms of time. Sprints are limited to a certain time period that is given to the team before they start. Requirements chosen to be completed in a sprint from the product backlog (known as the list of requirements in the Scrum methodology) must be completed within this given timeframe. If one is not completed, it is returned to the product backlog and is taken up in a future sprint, depending on its priority (i.e. high, medium or low).

So each aspect of the methodology can be performed without having a project team. Product backlogs (list of requirements) can easily be made by using requirement elicitation techniques; sprints can be carried out on your own by making a sprint backlog to implement, and can be kept track of by recording progress of each task in a personal journal. This would make as an equivalent to conducting daily scrum meetings where the project team come together and discuss any problems and next steps. Then at the end of the sprint, the product owner (in this case, the client) can be shown the results produced and whether they approve/would like to add any extra requirements.

The fact that this project is limited in terms of having only one person involved, apart from the project supervisor who acts as the ‘Scrum Master’, doesn’t justify not taking advantage of the organized approach that the Scrum methodology offers in getting the requirements set out to be implemented on time. It is a logical approach in the sense that tasks will be distributed into manageable portions and work from them instead of slowly going linearly through the list. For this reason it is felt that this methodology will provide this structure along with deadlines and other aiding features to successfully complete this project.

Figure 9 below shows the scrum process flow.

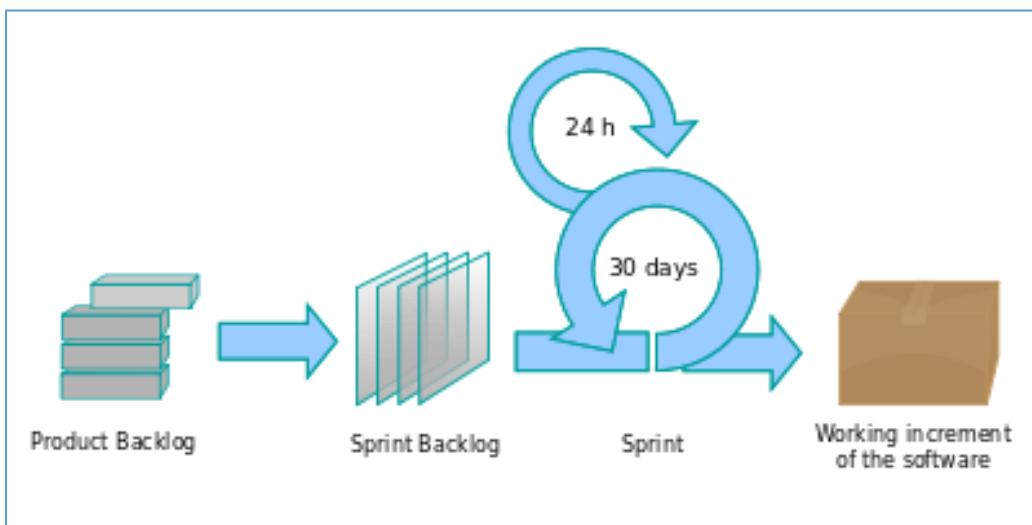


Figure 7:- Scrum Process Flow

- **Product Backlog**

The product backlog contains a prioritized list of requirements that should be included in the product. Items on this list are picked by the development team to implement in upcoming sprints.

- **Sprint Backlog**

This is the gathered list of requirements the product owner has chosen for the development team to work on in the upcoming sprint. These requirements are broken down into tasks that are given an estimate in time, which should ideally range from 4-16hrs<sup>i</sup>. In the first few sprints the development team should concentrate on the highest priority requirements in order to build the foundations of the product. Later on, sprint backlogs will contain only medium to low priority requirements to finalise the product. Before the sprint has begun, a ‘sprint planning’ meeting is held to select what work has to be done during the sprint and to assess whether the selected work can be completed within the timeframe.

- **Sprint**

A sprint is a time based development period where tasks from the sprint backlog are implemented. The time for a sprint is determined by the collective time period of all tasks contained in the sprint backlog, but normally ranges from 7-30 days. Each sprint must end on time, regardless of whether or not the selected requirements have been met successfully or not. If for some reason a requirement is not met at the end of the sprint, it is returned back to the product backlog to complete in a future sprint. Within a sprint, daily meetings are held to update team members on what they have done, what they plan to do and whether they require help in completing a task i.e. they are stuck.<sup>ii</sup> This is illustrated in the above diagram by the 24hr loop shown above the sprint cycle which also represents the tasks the team has chosen to complete for that day.

Once the sprint has ended, there are two meetings which are held. The ‘sprint review meeting’ is held to demo the results of the sprint to the client and review what work was and was not completed. The ‘sprint retrospective’ gives the project team the chance

to assess what went wrong in the process during the last sprint and what went well, in order to document and improve on future sprints.

### 3.6 Hardware Requirement

Minimum Requirements:

Screen	13.3"
Speaker	-
Keyboard	-
Storage	500 GB
Processor	Core i5
RAM	4 GB
Mouse	-
Printer	Inkjet Printer, Laser Printer

Table 1-Minimum Hardware Requirement

Chosen Hardware Requirement

Screen	15.6"
Speaker	-
Keyboard	wireless
Storage	1 TB
Processor	Core i7
RAM	6 GB
Mouse	wireless
Printer	Inkjet Printer

Table 2- Chosen Hardware Requirement

### 3.7 Software Requirement

Minimum Requirements:

<b>Operating System</b>	Windows 8.1 / Windows 10
<b>Text Editor</b>	Sublime Text 2/3, Brackets 1.6/ 1.6+/ 1.7
<b>Web Server</b>	XAMPP, WAMP, IILS
<b>Microsoft Office</b>	Version 2013
<b>Adobe Acrobat Reader</b>	Adobe Acrobat Reader DC
<b>Database</b>	MySQL
<b>Web Technologies</b>	HTML, CSS, Java Script
<b>Programming Languages</b>	PHP, Ruby, Java

Table 3:- Minimum Software Requirement

Chosen Software Requirement:

<b>Operating System</b>	Windows 10
<b>Text Editor</b>	Brackets 1.7, Visual Studio Code, Sublime text 2
<b>Web Server</b>	WAMP 5.0
<b>Microsoft Office</b>	Version 2013
<b>Adobe Acrobat Reader</b>	Adobe Acrobat Reader DC 10.0
<b>Database</b>	MySQL
<b>Web Technologies</b>	HTML, CSS, Java Script
<b>Programming Languages</b>	PHP7, Java

Table 4- Chosen Software Requirement

### 3.8 Feasibility Study

The Feasibility for the Campus Recruitment System was assessed in three principal ways, Technical, economically and operationally. The information below was collected by means of interviews and Observation.

<b>Technical Feasibility</b>	
Add on to Present System	Good storage capacity to cater for large amount of data.
Technology available to meet user's needs	All technologies are available.
<b>Economic Feasibility</b>	
Cost of system study	Not Applicable
Estimated cost of Hardware	Not Applicable
Cost of packaged software	Not Applicable
<b>Operational Feasibility</b>	
Whether the system will operate when installed	Chances are high
Whether the system will be used	Users will like to use if it is efficient and effective

Table 5- Feasibility Study

### **3.9 Cost Analysis**

#### **Cost**

The Campus Recruitment System is being developed for the University. After investigating about the existing system, it can be deduced that the users are ready for the new system and all the requirements are ready available to access to the website.

The Campus Recruitment System will be hosted on the local server at the University. No need for additional tools for the implementation of the system and no need for additional software to host the system on the local server.

### **3.10 Conclusion- justify the worthiness of continuing with the project**

After having an investigation of the actual system and identifies all its problem and all the requirement that will be used to implement the new system therefore it was decided to go ahead with the proposed system as it will resolve a real problem and will help the students to build up their own identity. Moreover every students, lecturers and companies will be beneficial from it. The project will not require huge amount of budget therefore it is worth to continue with the proposed system as it does not required huge budget and yet real problem will be solved.



**Table of Contents**

4. Design .....	27
4.1 Database Design .....	28
4.2 Relationship between tables .....	35
4.3 Database backup.....	36
4.4 Database Security .....	37
4.5 Procedural Design .....	38
4.6 Sequence Diagram.....	41
4.7 Flowchart.....	44
4.9 Form Design.....	50
4.10 NOTE.....	55

**Table of Figures**

Figure 1- Relationship of tables .....	35
Figure 2- backup phpMyAdmin.....	36
Figure 3- Entity Relationship Diagram.....	38
Figure 4- Login DFD .....	39
Figure 5- Complete DFD .....	40
Figure 6- Login user.....	41
Figure 7- Company-User .....	42
Figure 8- user-company .....	43
Figure 9- login flowchart .....	44
Figure 10- add new record .....	45
Figure 11- delete a record .....	46
Figure 12- update a record .....	47
Figure 13- search for a specific record .....	48
Figure 14- Site Map .....	49
Figure 15- home layout.....	50
Figure 16- login layout.....	51
Figure 17- forget password .....	52

Figure 18- Student and Lecturer profile.....	53
Figure 19-User dashboard.....	54
Figure 20- chat layout .....	55
Figure 216- admin dashboard .....	55

## Table of content

Table 1: company table .....	28
Table 2: Department LTSE.....	28
Table 3- Department LTGDM .....	29
Table 4- Department LPNDM .....	29
Table 5- Department BM .....	29
Table 6- Department Accounts and Finance.....	30
Table 7- Faculty .....	30
Table 8- Admin .....	30
Table 9- calendar admin.....	30
Table 10- calendar.....	31
Table 11- calendar 1.....	31
Table 12- calendar 2.....	31
Table 13- Lecturer.....	33
Table 14- student table.....	34

## 4. Design

The design phase contains of several diagram that will explain about the system and thus this will enable developer and admin and webmaster to have knowledge and idea of the system and the different phase of it. The different diagram or charts will illustrate the different module that are present in the system and the different diagrams that will be present the design phase are as follows:

Data design is a model to represent the aspects of the system in an abstract way. It involves mainly in the entities present and the link between them.

Use case Diagrams represent the user interaction with the system showing the relationship between the user and the system. In the use case diagram the different users associated with the system are shown the way they interact with the system.

Database Design is the creation of a detailed data model of a database. The database will consist of related files and records. Recording of records can be done using a unique id for the different entities of the system to avoid data redundancy.

Interface design which will display all the interface that will be available on the website and even different interface according to user's role.

Security design will show how the website is being protected as well as the database also to avoid compromise of the website and database and stealing of data.

## 4.1 Database Design

### Design Tables

#### 1. Company Table

Field Name	Data Type	Field Size	Constraints	Description
CID	int	11	primarykey	auto increment
companyID	varchar	50		Unique Key to identify company
Name	Varchar	75		Name of company
Address	varchar	100		address company
Email	varchar	75		email address
Motto	Varchar	100		saying
Telephone	Int	15		telephone number
Facebook	Varchar	255		facebook url
Skype	Varchar	25		skype name
Blog	Varchar	255		blog address
Website	Varchar	255		website address
Username	Varchar	15		username for login
Password	varchar	255		password for login
userPic	varchar	200		user picture
Docs	varchar	200		
Time_Joined	timestamp			store registered time

Table 1: company table

#### 2. Department LTSE Table

Field Name	Data Type	Field Size	Constraints	Description
MID	int	11	primary key	Unique key to identify module
Department	varchar	50		
ModuleID	varchar	15		Unique key
FacultyID	varchar	10	foreign key	used to identify faculty ID
Name	varchar	25		module name
Aims	varchar	1000		module aims
Percentage	float	4		rate of module
Hours	time	2		total module hours

Table 2: Department LTSE

### 3. Department LTGDM Table

Field Name	Data Type	Field Size	Constraints	Description
MID	int	11	primary key	Unique key to identify module
Department	varchar	50		
ModuleID	varchar	15	index	Unique key
FacultyID	varchar	10	foreign key	used to identify faculty ID
Name	varchar	25		module name
Aims	varchar	1000		module aims
Percentage	float	4		rate of module
Hours	time	2		total module hours

Table 3- Department LTGDM

### 4. Department LPNDM

Field Name	Data Type	Field Size	Constraints	Description
MID	int	11	primary key	Unique key to identify module
Department	varchar	50		
ModuleID	varchar	15	index	Unique key
FacultyID	varchar	10	foreign key	used to identify faculty ID
Name	varchar	25		module name
Aims	varchar	1000		module aims
Percentage	float	4		rate of module
Hours	time	2		total module hours

Table 4- Department LPNDM

### 5. Department BM

Field Name	Data Type	Field Size	Constraints	Description
MID	int	11	primary key	Unique key to identify module
Department	varchar	50		
ModuleID	varchar	15	index	Unique key
FacultyID	varchar	10	foreign key	used to identify faculty ID
Name	varchar	25		module name
Aims	varchar	500		module aims
Percentage	float	4		rate of module
Hours	time	2		total module hours

Table 5- Department BM

## 6. Department Accounts and Finance

Field Name	Data Type	Field Size	Constraints	Description
MID	int	11	primary key	Unique key to identify module
Department	varchar	50		
ModuleID	varchar	15	index	Unique key
FacultyID	varchar	10	foreign key	used to identify faculty ID
Name	varchar	25		module name
Aims	varchar	1000		module aims
Percentage	float	4		rate of module
Hours	time	2		total module hours

Table 6- Department Accounts and Finance

## 7. Faculty Table

Field Name	Data Type	Field Size	Constraints	Description
FID	int	11	Primary key	Unique key to identify faculty
FacultyID	varchar	10	index	unique key
Name	varchar	50	index	faculty name
Dean	varchar	15		dean name
HOD	varchar	15		Head of Department name

Table 7- Faculty

## 8. Admin table

Field Name	Data Type	Field Size	Constraints	Description
user_id	varchar	10	primary key	auto increments
user_name	varchar	15		admin username
user_email	varchar	40		admin email address
user_pass	varchar	255		admin password
Times_Joined	timestamp			time registered

Table 8- Admin

## 9. Calendaradmin

Field Name	Data Type	Field Size	Constraints	Description
id	int	11	primary key	auto increment
title	varchar	255		event title
startdate	varchar	48		event start date
enddate	varchar	48		event end date
allDay	varchar	5		either full day or not

Table 9- calendar admin

## 10. Calendar

Field Name	Data Type	Field Size	Constraints	Description
id	int	11	primary key	auto increment
studentID	varchar	50	foreign key	store data according to student id for retrieve to be easy
title	varchar	255		event title
startdate	varchar	48		event start date
enddate	varchar	48		event end date
allDay	varchar	5		either full day or not

Table 10- calendar

## 11. Calendar 1

Field Name	Data Type	Field Size	Constraints	Description
id	int	11	primary key	auto increment
LecturerID	varchar	50	foreign key	store data according to lecturerid for retrieve to be easy
title	varchar	255		event title
startdate	varchar	48		event start date
enddate	varchar	48		event end date
allDay	varchar	5		either full day or not

Table 11- calendar 1

## 12. Calendar2

Field Name	Data Type	Field Size	Constraints	Description
id	int	11	primary key	auto increment
companyID	varchar	50	foreign key	store data according to company id for retrieve to be easy
title	varchar	255		event title
startdate	varchar	48		event start date
enddate	varchar	48		event end date
allDay	varchar	5		either full day or not

Table 12- calendar 2

## 13. Lecturer Table

Field Name	Data Type	Field Size	Constraints	Description
LID	int	11	Primary key	Unique Key to identify lecturer
LecturerID	varchar	15	Unique	
FacultyID	varchar	10	foreign key	used to identify faculty ID
Name	Varchar	50		Name of lecturer
Address	varchar	100		address lecturer
Email	varchar	50		email address
Motto	Varchar	500		saying
Quote1	varchar	500		quote
Quote2	varchar	500		quote
GoodAt	varchar	1500		Good at which tasks
Hobbies	varchar	1500		Hobbies
WorkExperienceTitle1	varchar	200		workexperiencetitle
WorkExperience1	varchar	200		workexperience
WorkExperienceDateEnd1	varchar	100		start and end date
WorkExperienceAbout1	varchar	2000		about the work
WorkExperienceTitle2	varchar	200		workexperiencetitle
WorkExperience2	varchar	200		workexperience
WorkExperienceDateEnd2	varchar	100		start and end date
WorkExperienceAbout2	varchar	2000		about the work
WorkExperienceTitle2	varchar	200		workexperiencetitle
WorkExperience2	varchar	200		workexperience
WorkExperienceDateEnd2	varchar	100		start and end date
WorkExperienceAbout2	varchar	2000		about the work
Education	varchar	200		education level
EducationDateEnd	varchar	100		date start and end
Education1	varchar	200		education level
EducationDateEnd1	varchar	100		date start and end
Education2	varchar	200		education level
EducationDateEnd2	varchar	100		date start and end
OtherQualification	varchar	1500		Other qualification about user
Telephone	int	15		telephone number
LinkedIn	Varchar	25		LinkedIn url
Twitter	varchar	255		user twitter address
Skype	Varchar	25		skype name
Blog	Varchar	25		blog address
Website	Varchar	25		website address
skills	varchar	1000		lecturer skill
Department	varchar	50		Department name
AboutMe	varchar	1500		user description
Portfolio	varchar	255		user portfolio

Username	Varchar	15		username for login
Password	varchar	15		password for login
userPic	varchar	200		User picture
CV	varchar	200		user CV
Time_Joined	timestamp			take the time of registration

*Table 13- Lecturer*

## 14. Student Table

Field Name	Data Type	Field Size	Constraints	Description
SID	int	11	Primary key	Unique Key to identify lecturer
StudentID	varchar	50	Unique	
FacultyID	varchar	50	foreign key	used to identify faculty ID
Name	Varchar	50		Name of lecturer
Address	varchar	100		address lecturer
Email	varchar	50		email address
Motto	Varchar	500		saying
Quote1	varchar	500		quote
Quote2	varchar	500		quote
GoodAt	varchar	1500		Good at which tasks
Hobbies	varchar	1500		Hobbies
WorkExperienceTitle1	varchar	200		workexperiencetitle
WorkExperience1	varchar	200		workexperience
WorkExperienceDateEnd1	varchar	100		start and end date
WorkExperienceAbout1	varchar	2000		about the work
WorkExperienceTitle2	varchar	200		workexperiencetitle
WorkExperience2	varchar	200		workexperience
WorkExperienceDateEnd2	varchar	100		start and end date
WorkExperienceAbout2	varchar	2000		about the work
WorkExperienceTitle2	varchar	200		workexperiencetitle
WorkExperience2	varchar	200		workexperience
WorkExperienceDateEnd2	varchar	100		start and end date
WorkExperienceAbout2	varchar	2000		about the work
Education	varchar	200		education level
EducationDateEnd	varchar	100		date start and end
Education1	varchar	200		education level
EducationDateEnd1	varchar	100		date start and end
Education2	varchar	200		education level
EducationDateEnd2	varchar	100		date start and end
OtherQualification	varchar	1500		Other qualification about user
Telephone1	int	15		telephone number
Telephone2	int	15		telephone number

LinkedIn	Varchar	255		LinkedIn url
Facebook	varchar	255		Facebook url
Twitter	varchar	255		user twitter address
Skype	Varchar	25		skype name
Blog	Varchar	25		blog address
skills	varchar	1000		lecturer skill
Department	varchar	50		Department name
AboutMe	varchar	1500		user description
Portfolio	varchar	255		user portfolio
Username	Varchar	15		username for login
Password	varchar	15		password for login
userPic	varchar	200		User picture
CV	varchar	200		user CV
registration_time	timestamp			take the time of registration

Table 14- student table

## 4.2 Relationship between tables

The databases contains nine Entities COMPANY, STUDENT, LECTURER, LTSE, LTGDM, LPNDM, BM, ACCOUNTING AND FINANCE, and FACULTY and they are related as follows:

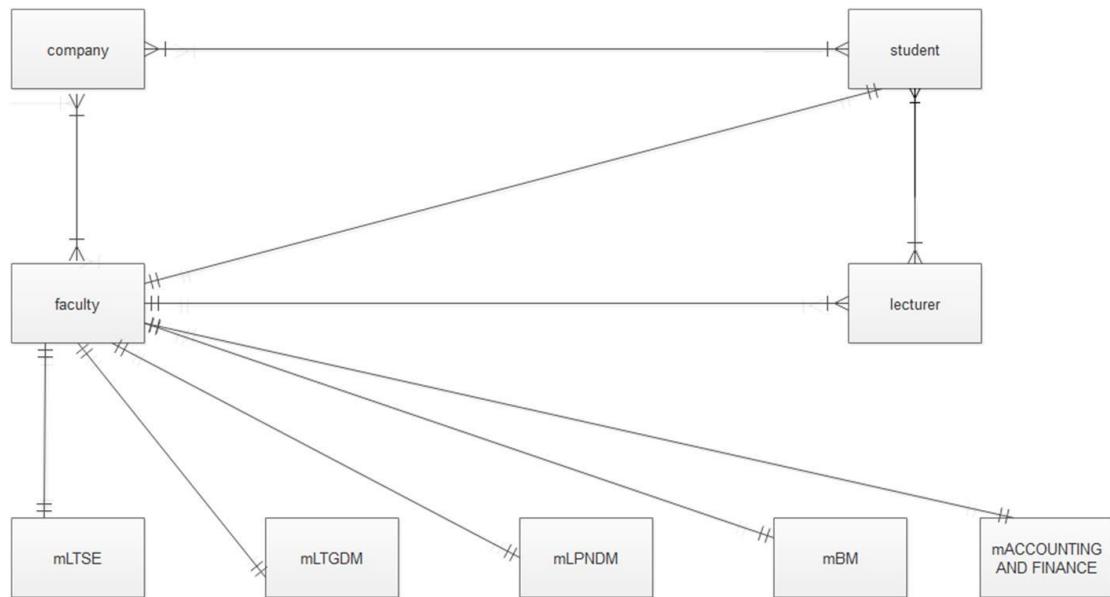


Figure 1- Relationship of tables

### Description of Entities Relationship:

- 1 Many companies can view many students
- 2 Many students can view many companies
- 3 Many student can be taught by many lecturers
- 4 Many lecturers teach many students
- 5 One faculty contain many lecturers
- 6 Many companies can view different faculties
- 7 One module belong to one faculty

### 4.3 Database backup

Backup will be done weekly so that to ensure security of the system. The figure 2 below shows how backup can be done on phpMyAdmin.

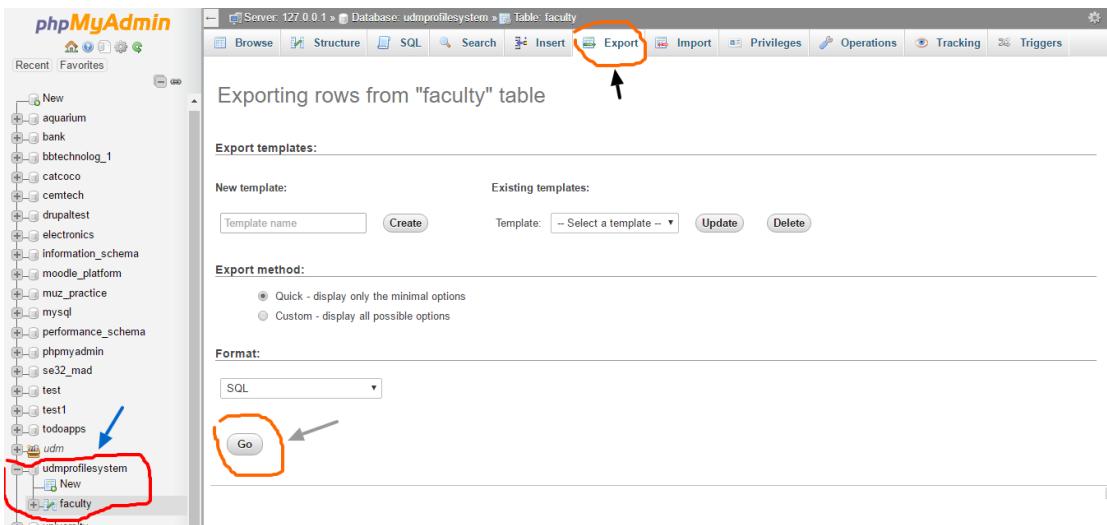


Figure 2- backup phpMyAdmin

To perform backup, follow the following step:

- Expand or select the database “udmprofilesystem” on the left sidebar like as been shown with the red circular and blue direction in figure 2
- Click on export on the tab bar on the page “phpMyAdmin” like as been shown with the red circle and black direction in figure 2
- Click on the “Go” bottom at the bottom like as been shown with red circle and grey direction.
- After clicking on “Go” the database will be downloaded as a backup call “udmprofilesystem.sql” and save in the download folder.
- Move the “udmprofilesystem.sql” from the download folder to a secure place.

## 4.4 Database Security

Prevention for Illegal access to the system and database.

- Each user will be provided with a username and password.
- Administrator will have access through the whole system including user databases.

Different ways can be used to secure a database:

- 1 The physical machine where database will be host should be kept in a secure place and under surveillance to prevent unauthorised access.
- 2 All unused or functions of the databases are removed or turn off.
- 3 Null passwords should not be allowed to use while registering users.
- 4 Backup and recovery procedures are periodically tested.

### **Firewalls**

A firewall is a network security system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both.

#### **Hardware Firewall**

A Hardware Firewall is a device to which you connect your computers or network in order to protect them from unauthorized access

#### **Software Firewall**

A Software Firewall is a piece of software that is installed on your computer in order to protect it from unauthorised access and unauthorised normal users to perform any modifications of information in the database.

## 4.5 Procedural Design

### Entity Relationship Diagram

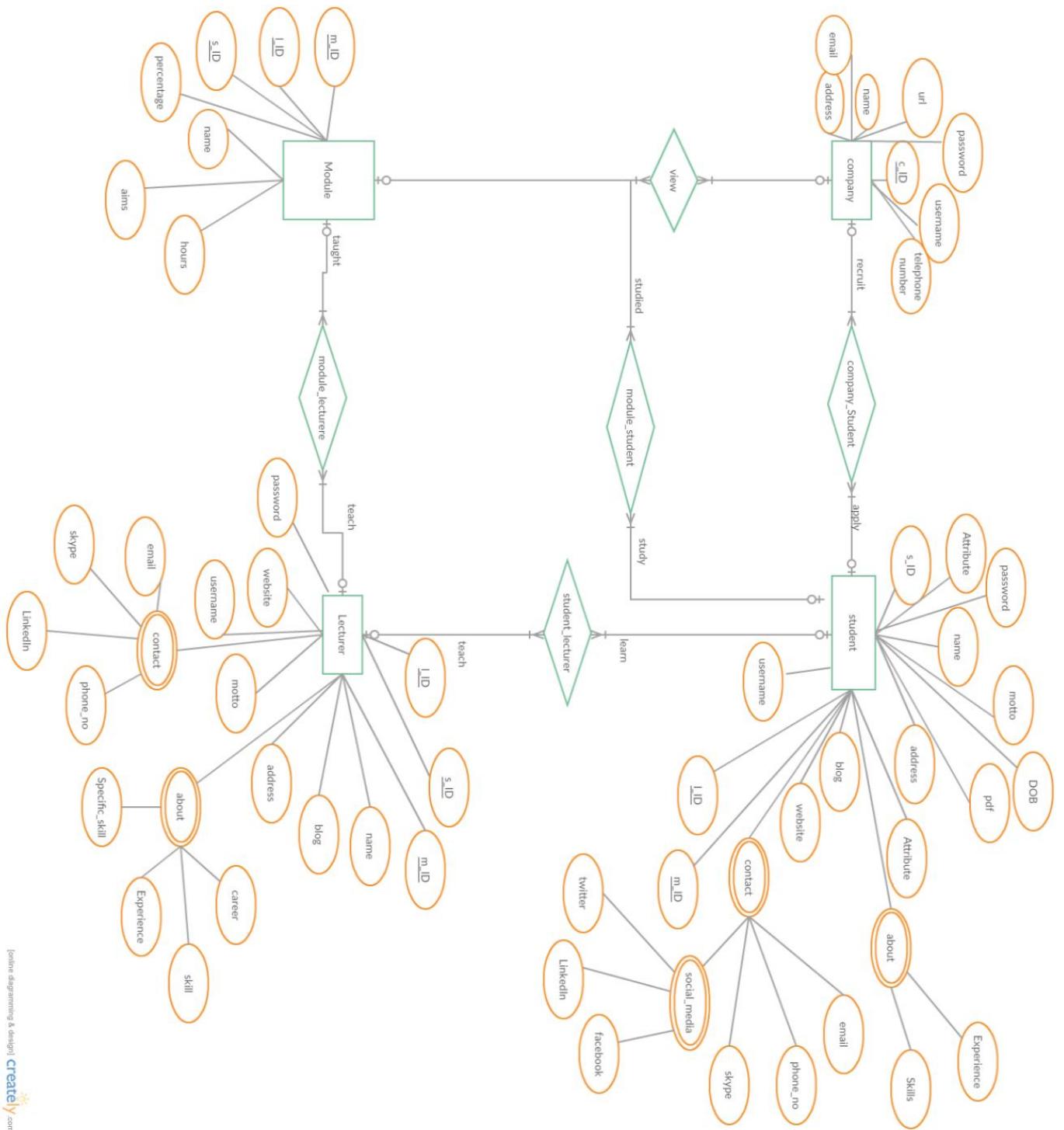
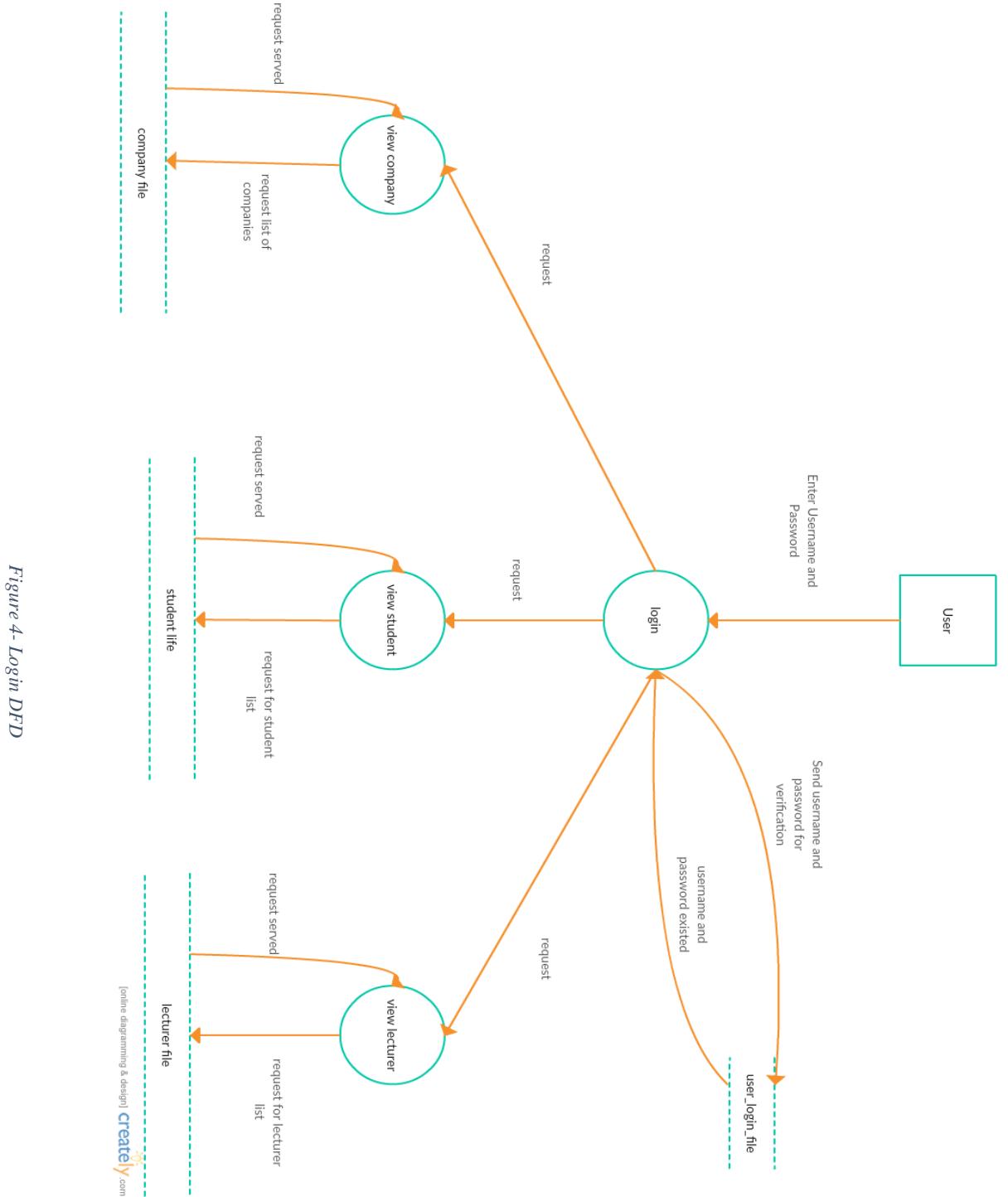


Figure 3- Entity Relationship Diagram

The figure 3 above shows the entire Entity Relationship Diagram with all its foreign keys and primary keys.

### Data Flow Diagram (DFD)

- Login DFD



- Complete DFD

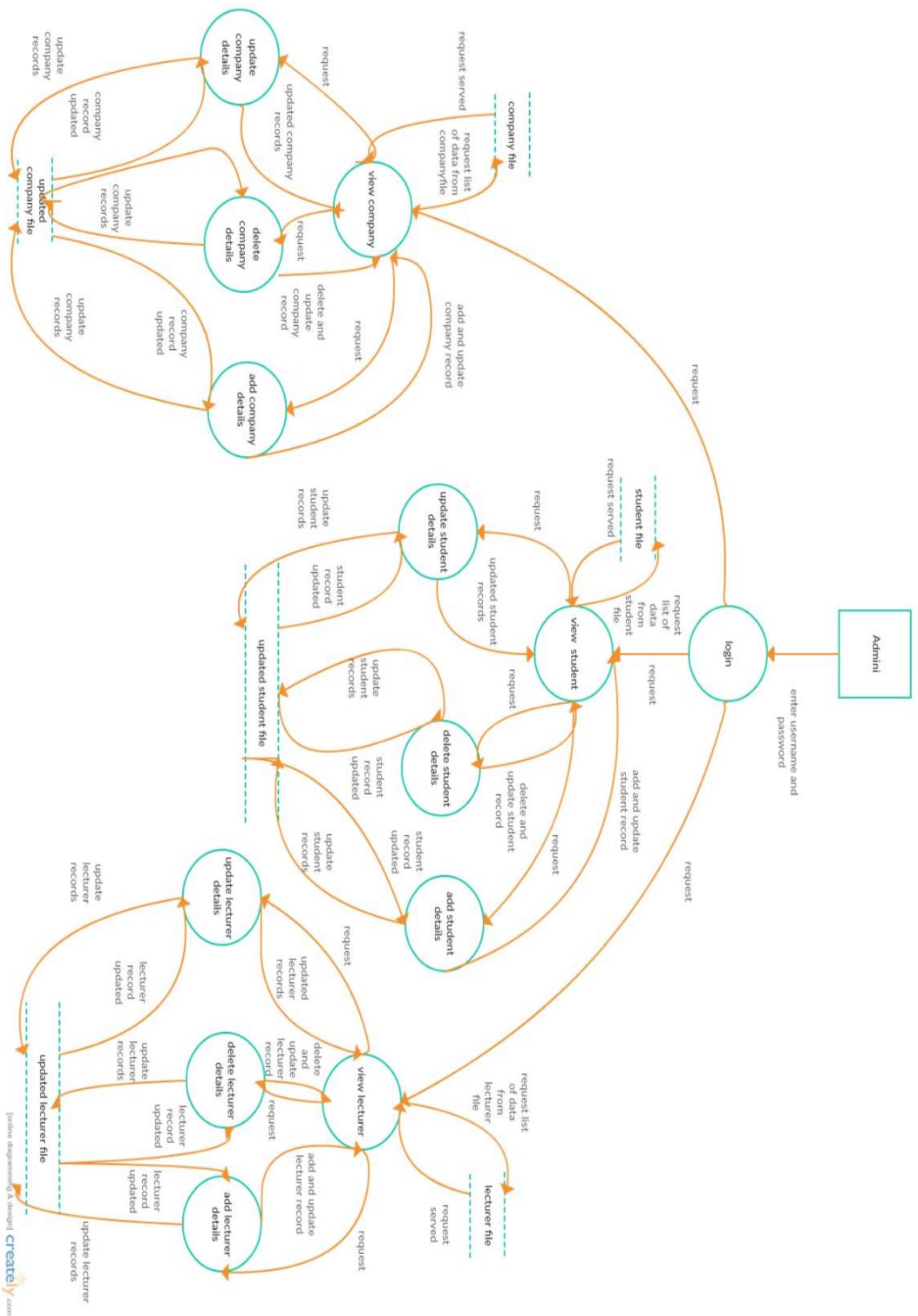


Figure 5- Complete DFD

## 4.6 Sequence Diagram

### 1. Login

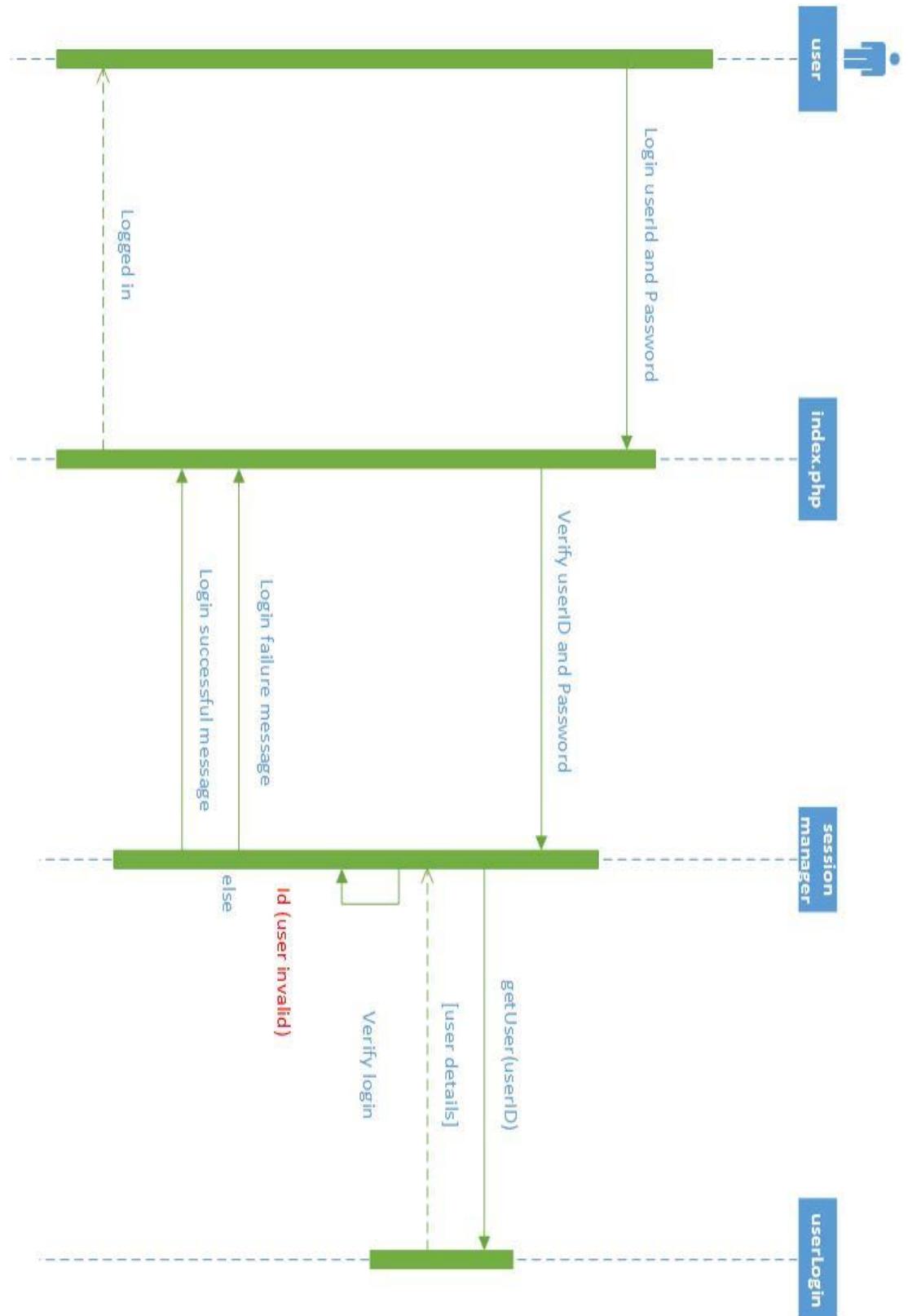


Figure 6- Login user

## 2. Company-User



Figure 7- Company-User

### 3. User-Company

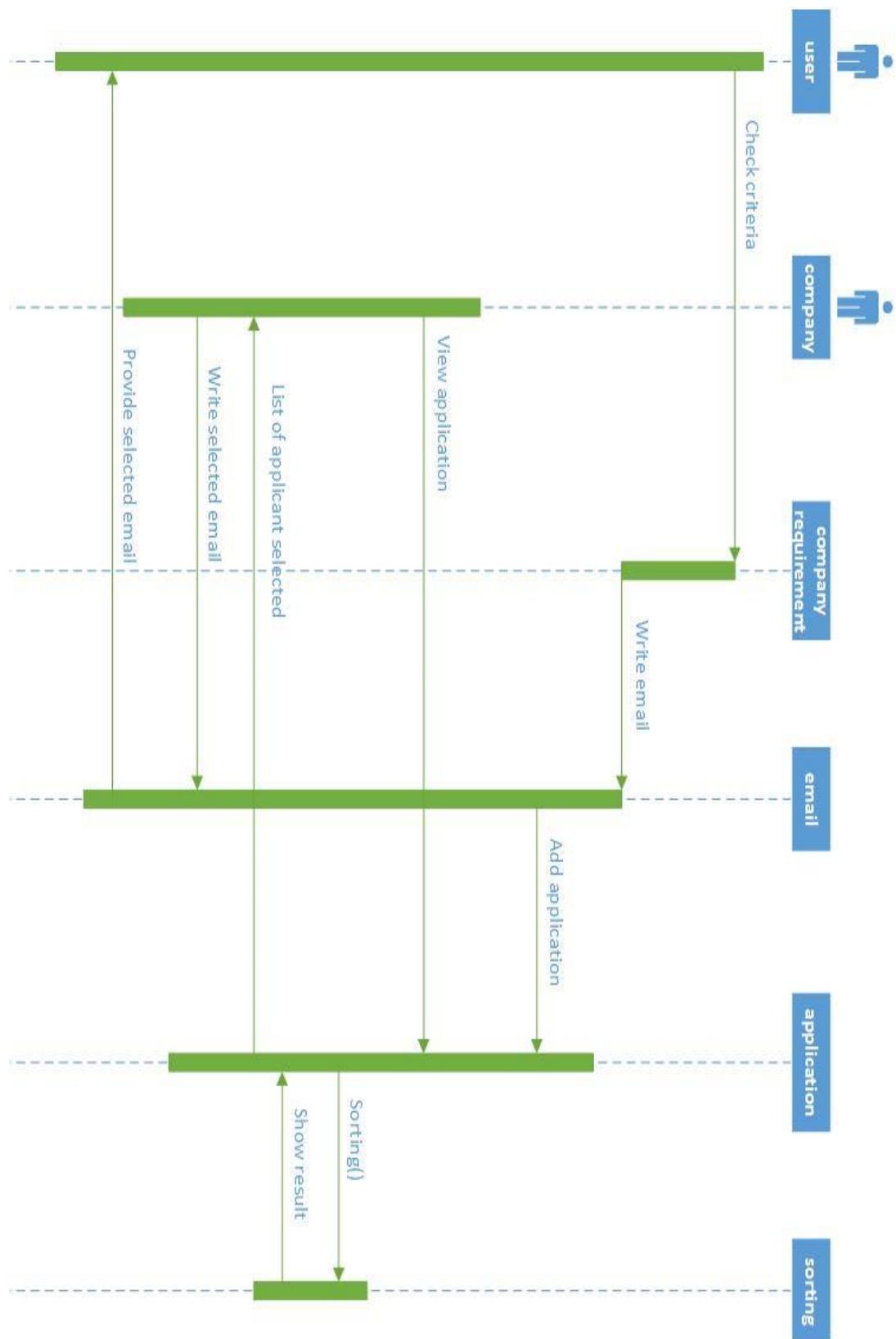


Figure 8-user-company

## 4.7 Flowchart

### Login into the website

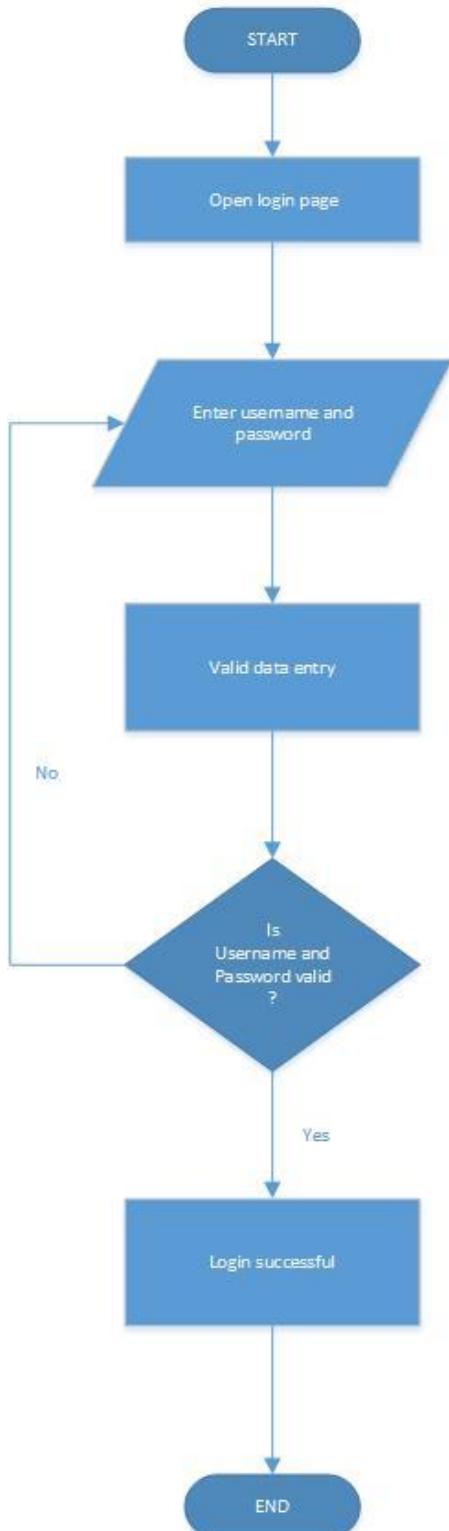


Figure 9- login flowchart

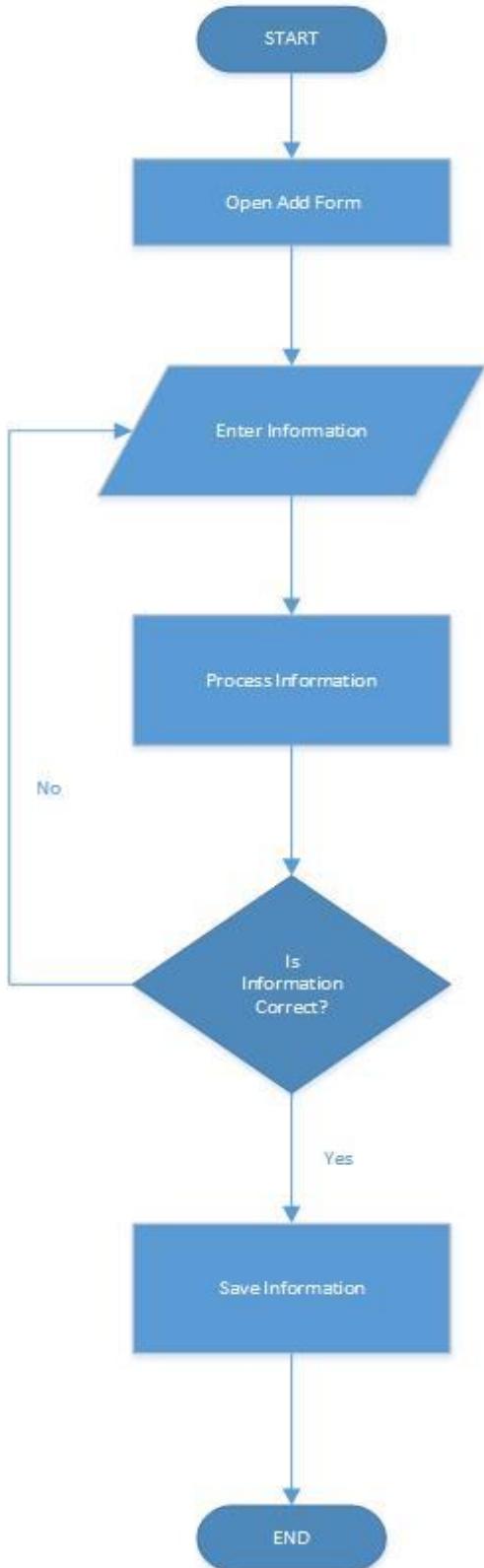
Add new record

Figure 10- add new record

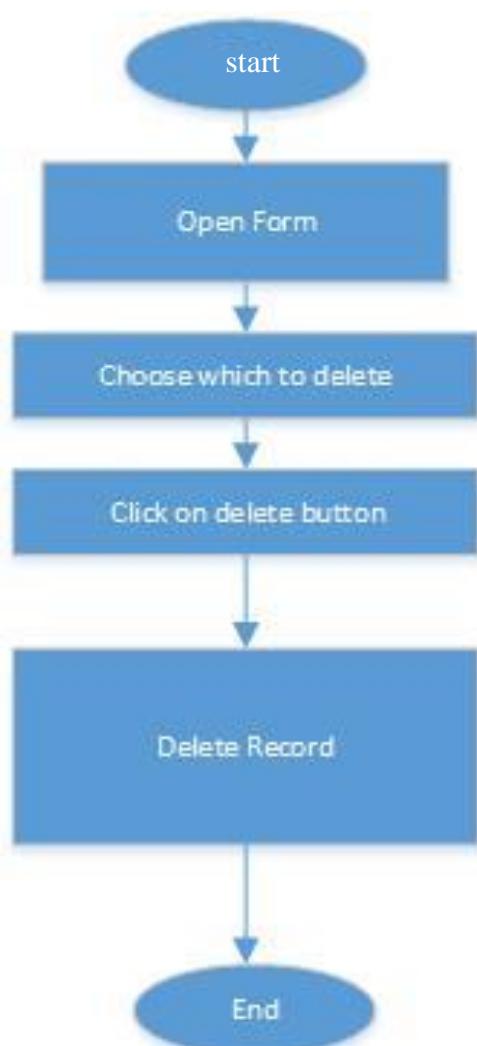
Delete a record

Figure 11- delete a record

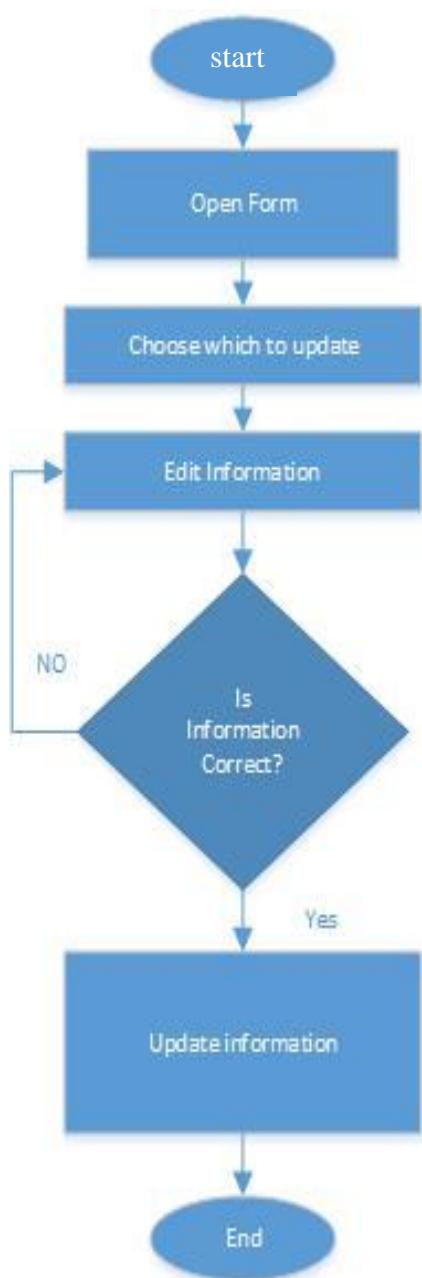
Update a record

Figure 12- update a record

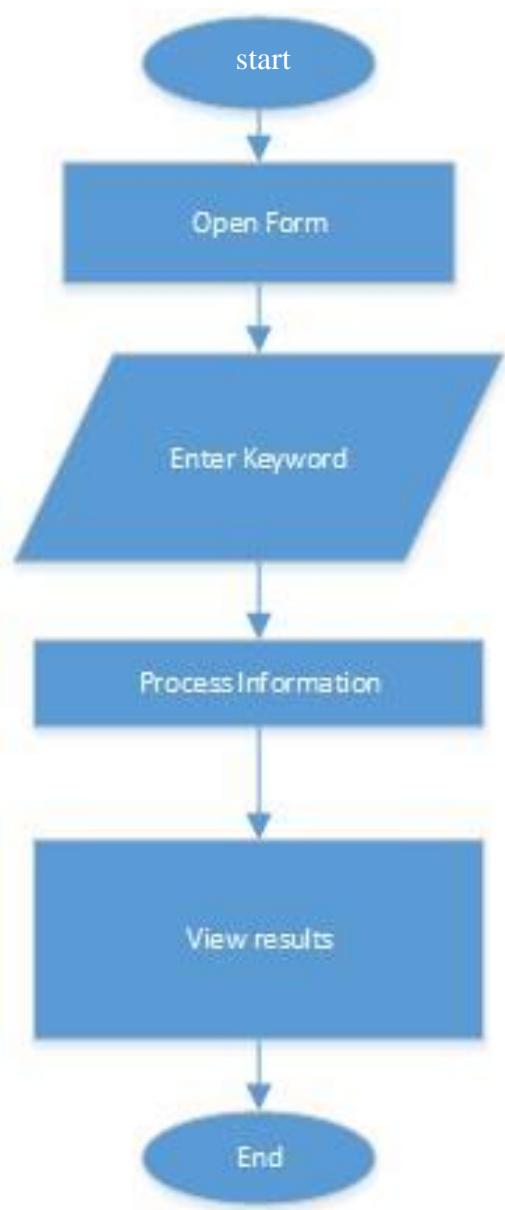
Search a record

Figure 13- search for a specific record

## 4.8 User Interface Design

### Site Map

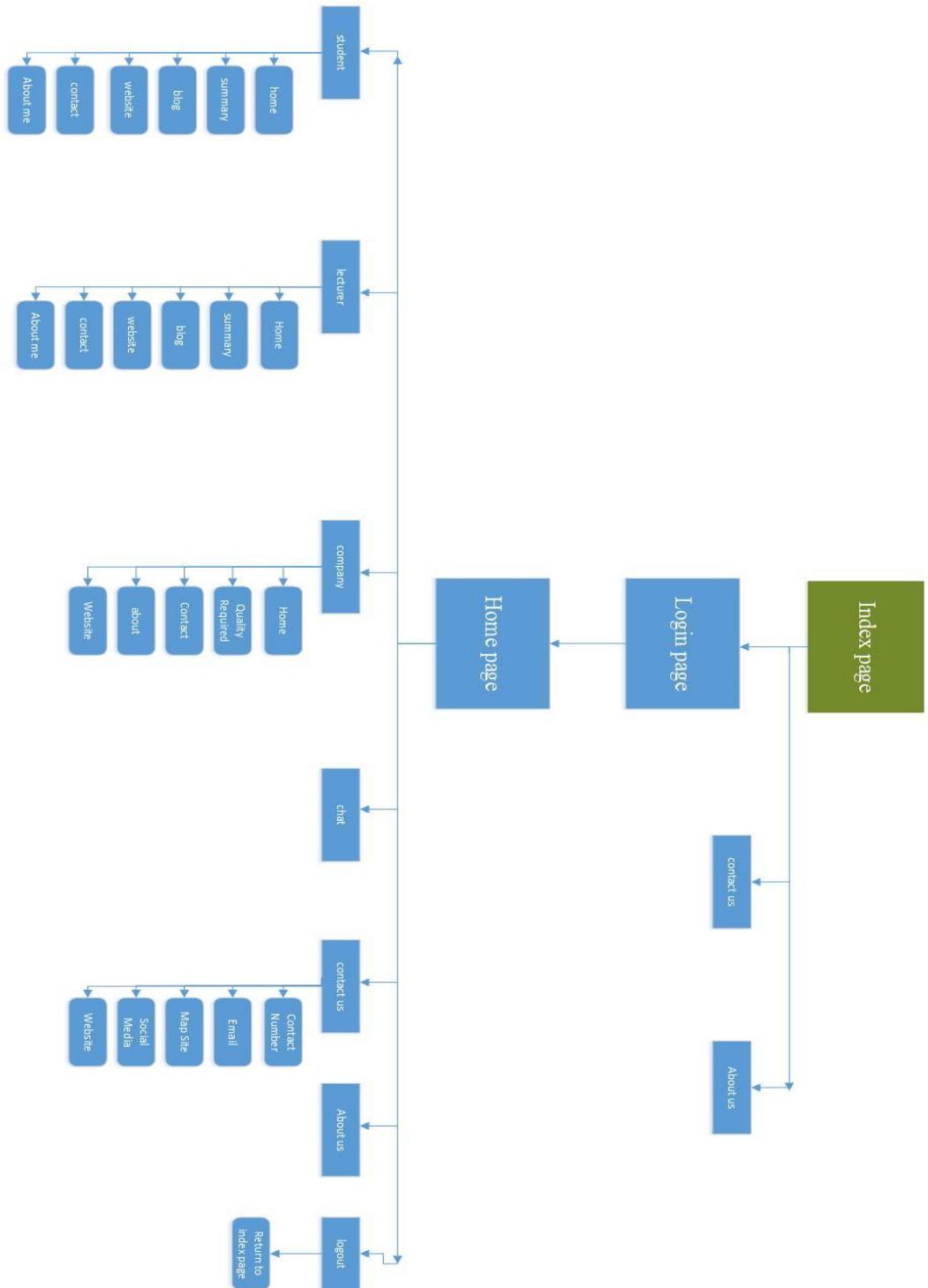


Figure 14- Site Map

## 4.9 Form Design

### Home layout

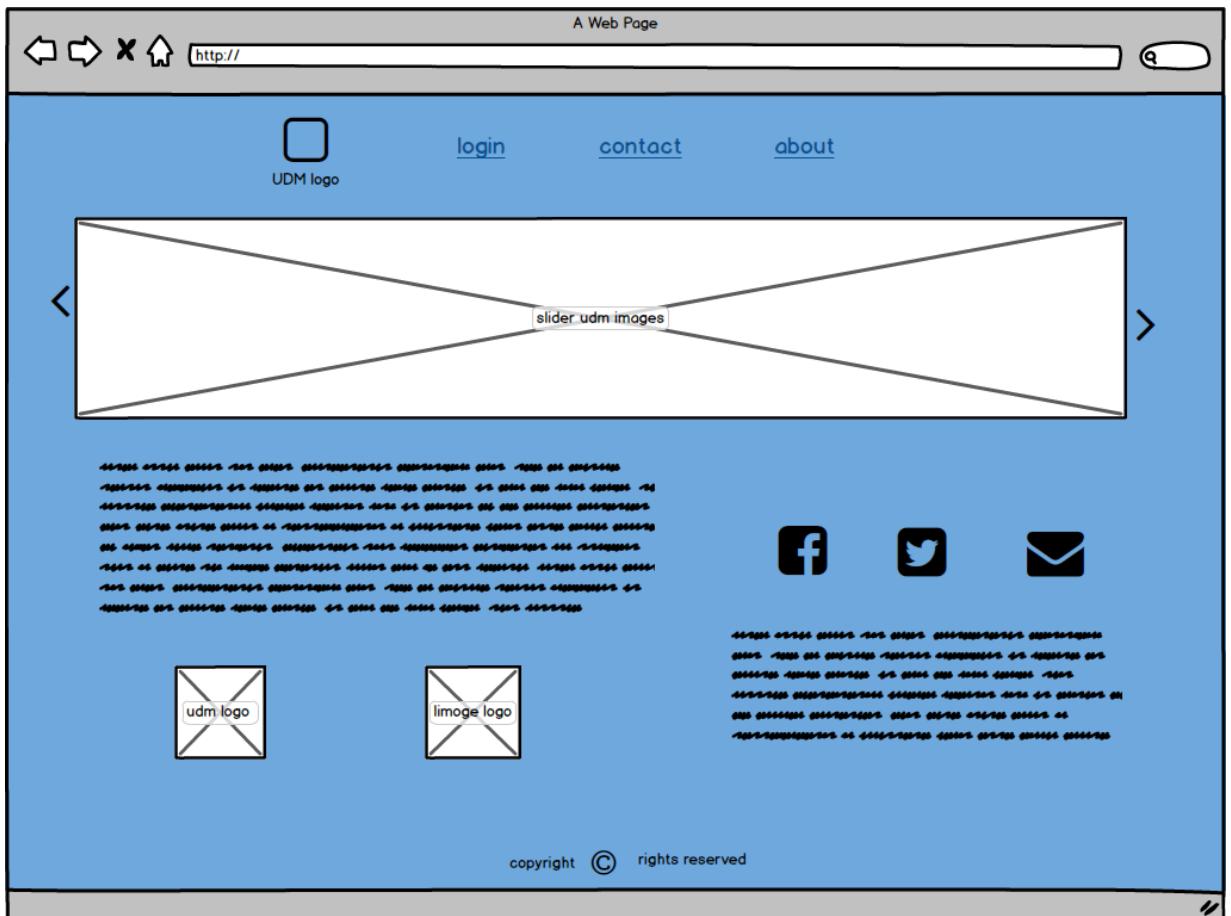


Figure 15- home layout

The above figure 11 is a home layout of the website where it will included some text and photos and even hyperlinks to the login part.

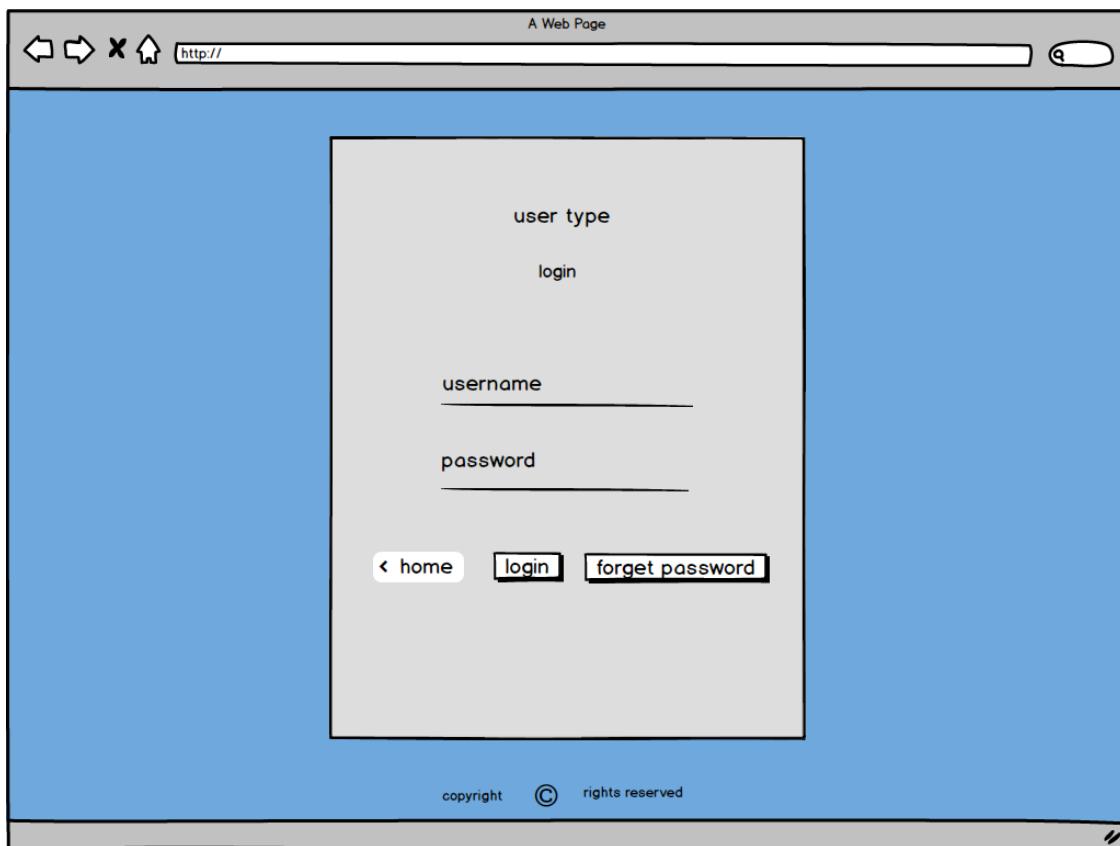
Login layout

Figure 16- login layout

The above figure 12 is a login layout for the website so that to be able to access the system.

### Forget password

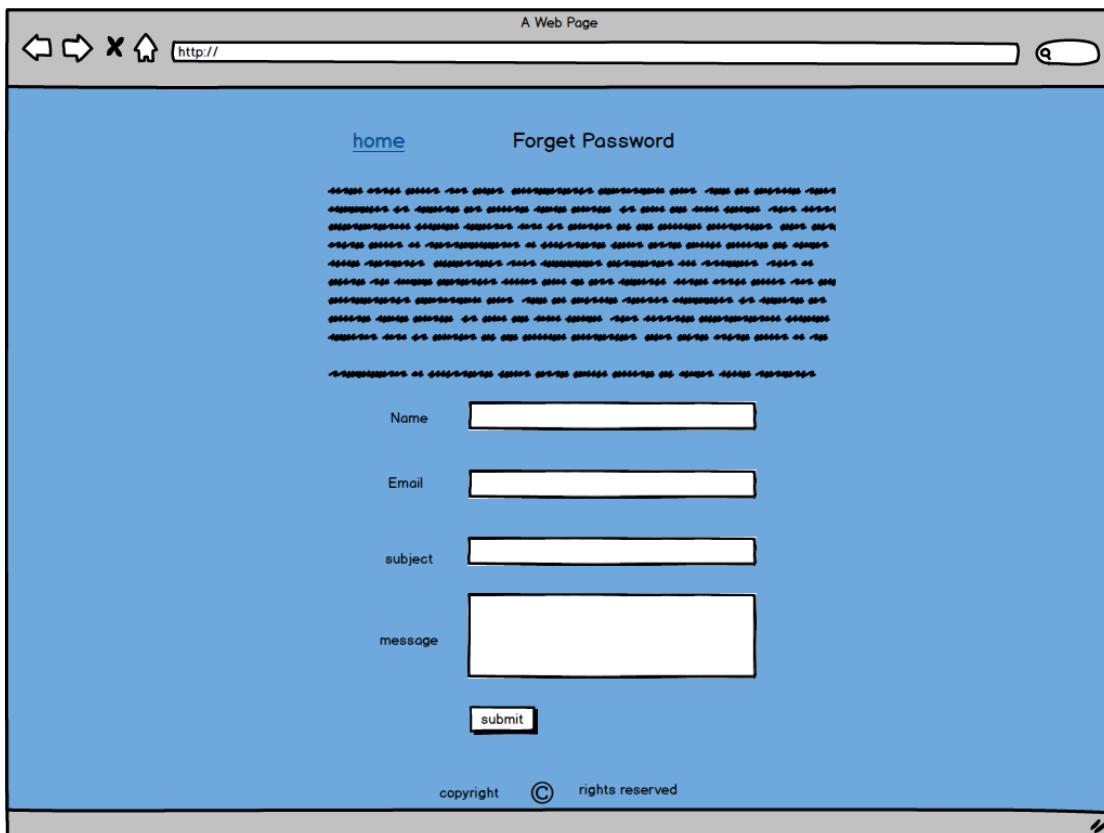


Figure 17- forget password

The above figure 13 is the forget password part whenever the user is sent to this page he has to fill up and his password will be reset during 24 hours.

### Student and Lecturer profile

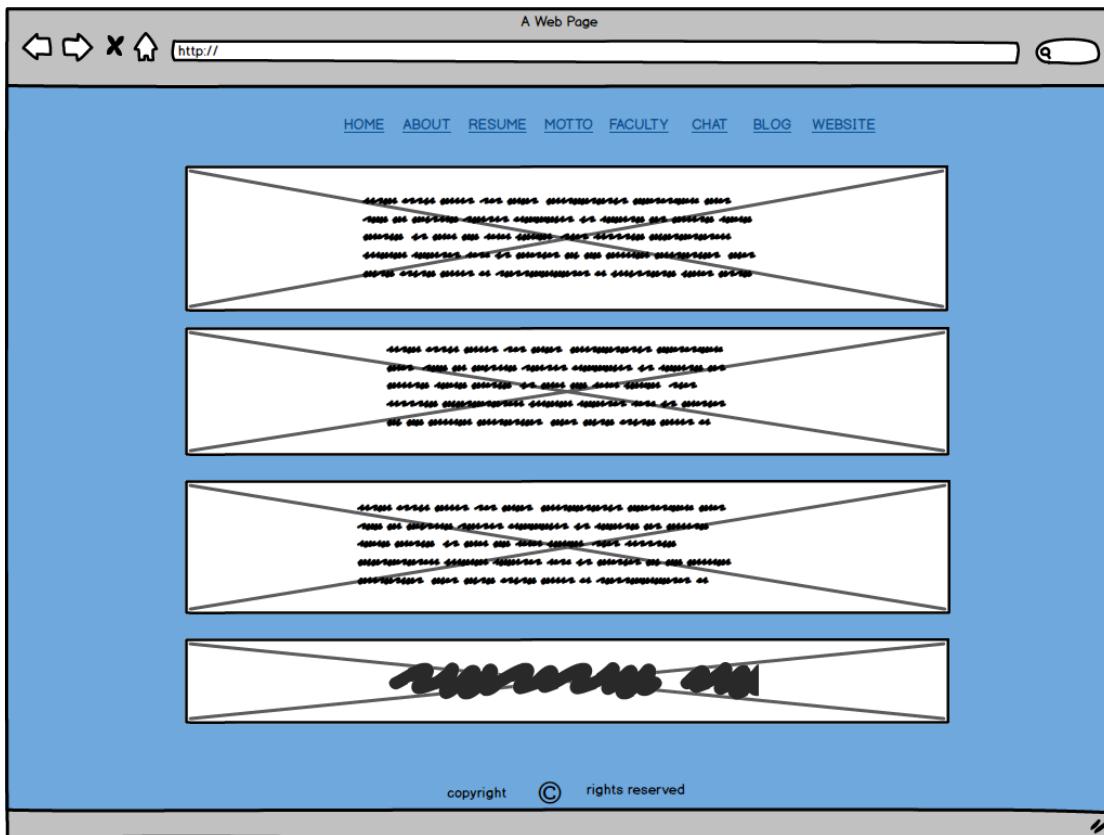


Figure 18- Student and Lecturer profile

The above figure 14 is how the student and lecturer page will look like after the system will developed.

## User Dashboard



Figure 19-User dashboard

The above figure 15 is how a dashboard for student, lecturer and company will look like so that they are able to view data and update their data.

## Admin Dashboard

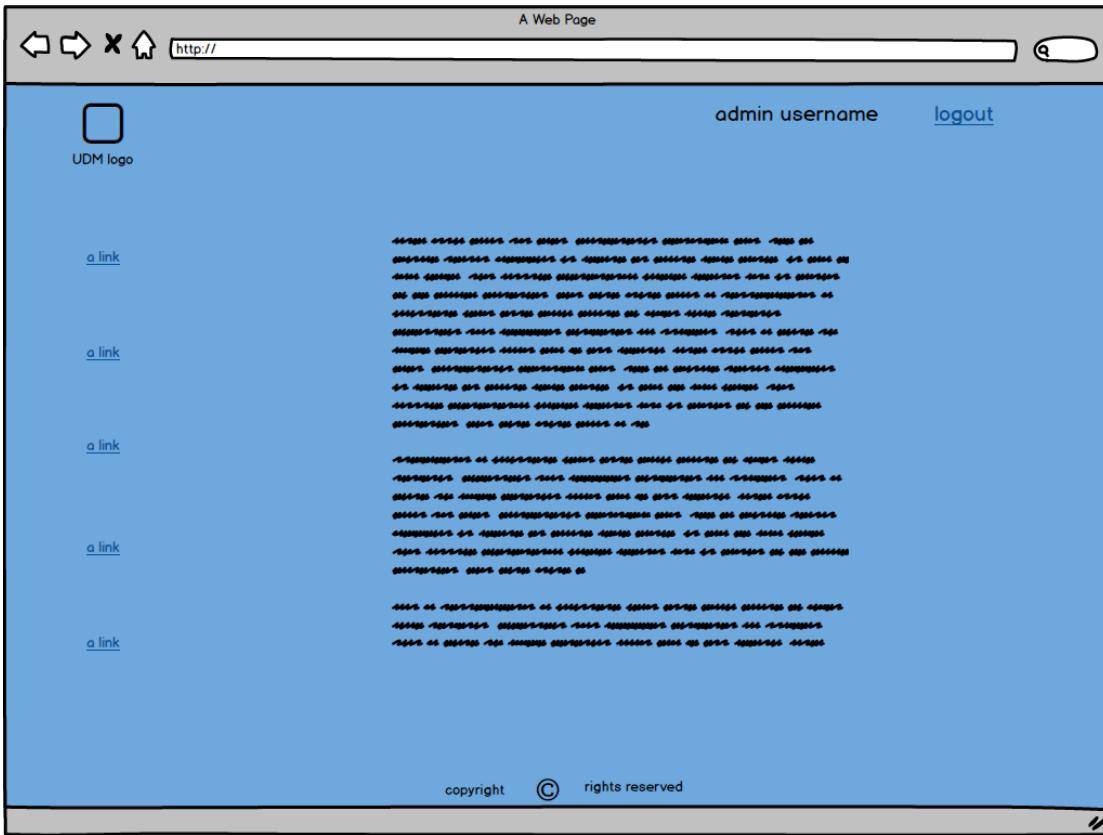


Figure 216- admin dashboard

The above figure 16 shows the admin dashboard where the admin can view, update or delete information about users.

### 4.10 NOTE

The site map (figure 10) and all the design about the interface can be change while the project is being developed. A unit testing will be perform after each sprint to ensure the functionalities are being working as expected.

There are no definite font size or font family or font color that has been used. They are being implemented as per the interface required.

Padding and Margin depends on the website but by default its 5px (custom).

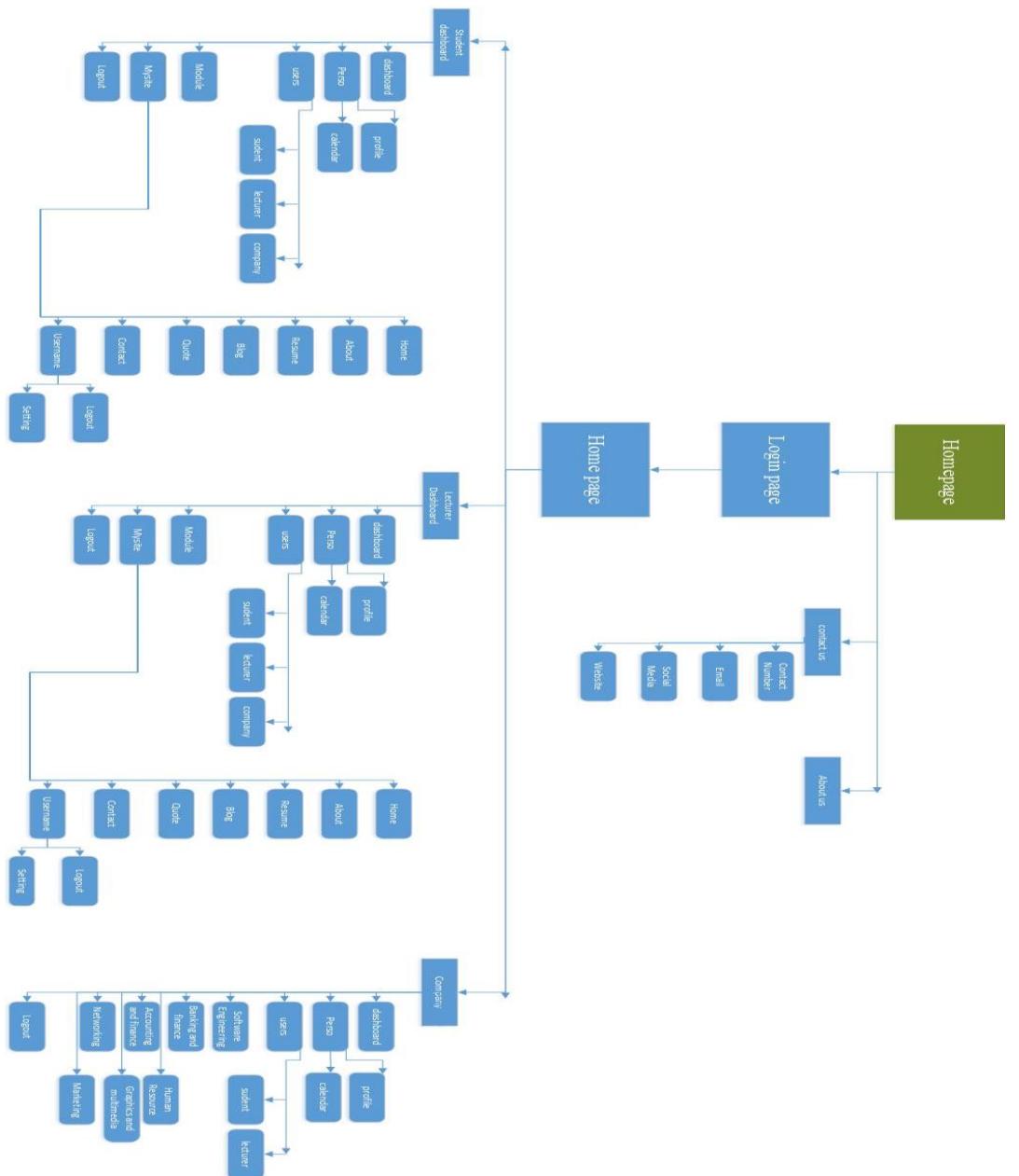
Every style has been used according the requirement of the website so that it match the website environment and keep users focus.

## Table of Contents

5 Implementation .....	57
5.1 Hardware Requirement and Software Requirement.....	58
5.2 Database server and Web server been used .....	59
5.3 How Database connection was made .....	60
5.4 Interfaces .....	61
5.4.1 Welcome at UDM.....	61
5.4.2 Administration Dashboard.....	73
5.5 Methodologies .....	81
5.6 Major Problem.....	82
5.7 Code adjustment and Implemented .....	82
5.8 Validation Code.....	100

## 5 Implementation

This section will describe how the different interfaces in the project were implemented. It will show the development tools that were used, all the steps that were followed, the codes used to implement the interfaces and what are the difficulties faced and how they were addressed.



*Figure 1 - final site map*

## 5.1 Hardware Requirement and Software Requirement

Below is a list of the hardware that have been used to make up the system:

- 1 Mouse (wireless)
- 2 Keyboard (wireless)
- 3 Laptop with screen 15.6”
- 4 Processor core i7
- 5 Hard disk 1TB
- 6 RAM 6 GB

Below is a list of the software that have used to developed the system

- 1 Web browser – google chrome, Microsoft Edge
- 2 Text editor – Sublime, Visual Studio code, Atom
- 3 Word document
- 4 Acrobat Reader (.pdf)
- 5 WAMP

Below is a list of what is needed to make the system live

- 1 FileZilla
- 2 Server
- 3 Firewall
- 4 Web browser
- 5 Text editor
- 6 XAMPP/ WAMP

## 5.2 Database server and Web server been used

### **Database Server**

- Server: Local Databases (127.0.0.1 via TCP/IP)
- Server type: MySQL
- Server version: 5.7.14 - MySQL Community Server (GPL)
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

### **Web Server**

- Apache/2.4.23 (Win64) PHP/5.6.25
- Database client version: libmysql - mysqlnd 5.0.11-dev - 20120503 - \$Id: 76b08b24596e12d4553bd41fc93cccd5bac2fe7a \$
- PHP extension: mysqli curl mbstring
- PHP version: 5.6.25

### 5.3 How Database connection was made

The connection was made using php7

```
<?php  
  
$hostname='localhost';  
  
$username='root';  
  
$password='';  
  
try {  
  
    $db =new PDO ("mysql:host=$hostname;dbname=udm", $user  
name, $password);  
  
    // set the PDO error mode to exception  
  
    $db-  
>setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
}  
  
// echo "Connected successfully";  
  
catch(PDOException $e)  
  
{echo "Connection failed: " . $e->getMessage();}  
  
?>
```

## 5.4 Interfaces

### 5.4.1 Welcome at UDM

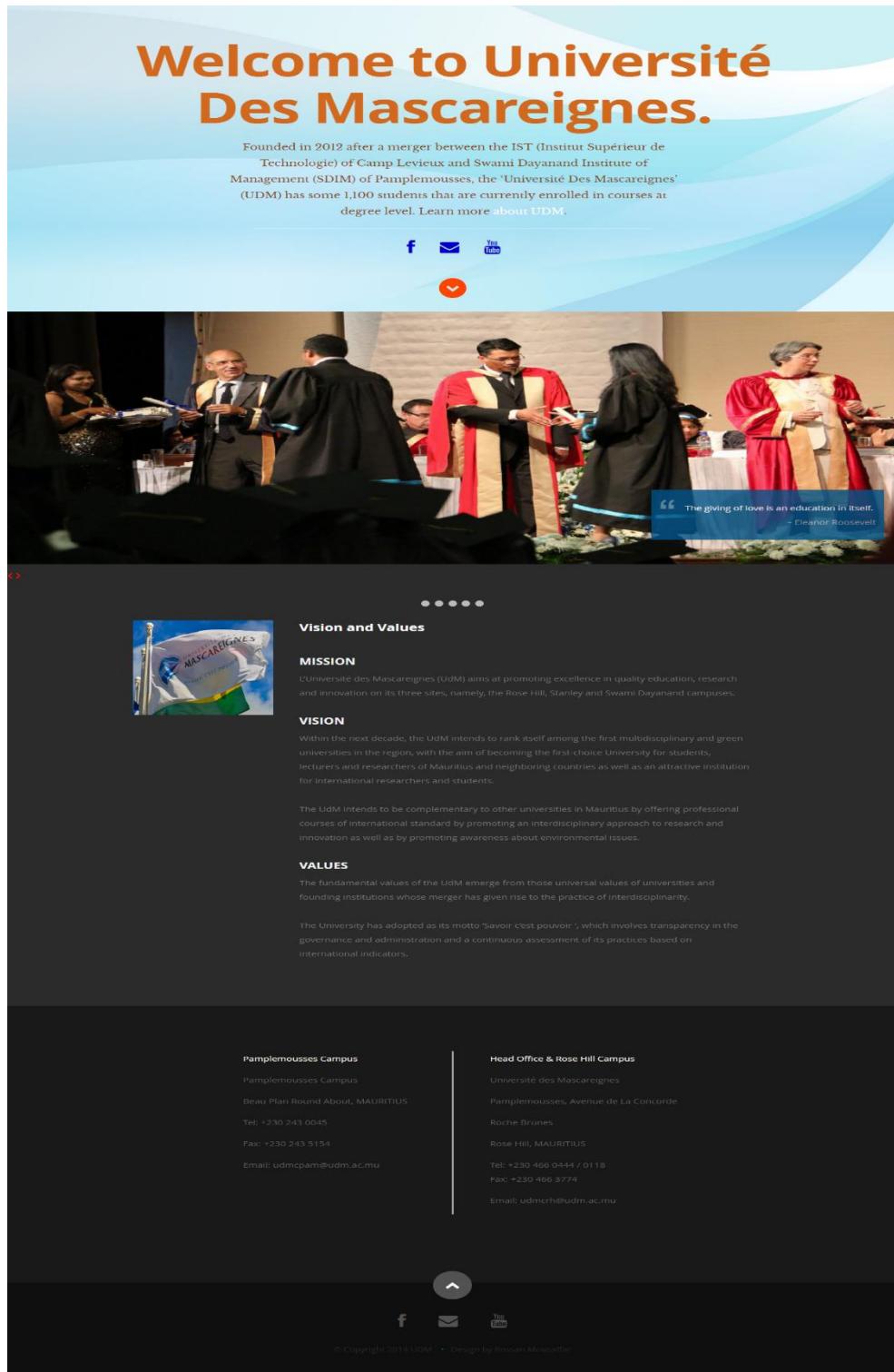


Figure 2- Homepage

After the entering the site their will have a welcome page and it's a one page where it contain different section of information like Home , About, Contact and login

So click on login then it will be directed to the login page like has been show below

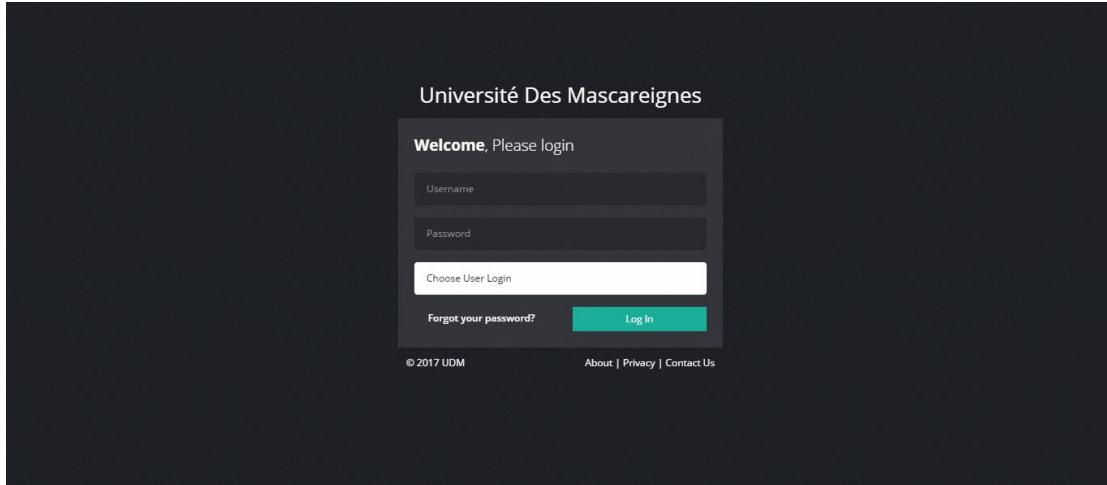


Figure 3- login

If ever user forget password just has to click on the forget password it will redirect user to the figure below:

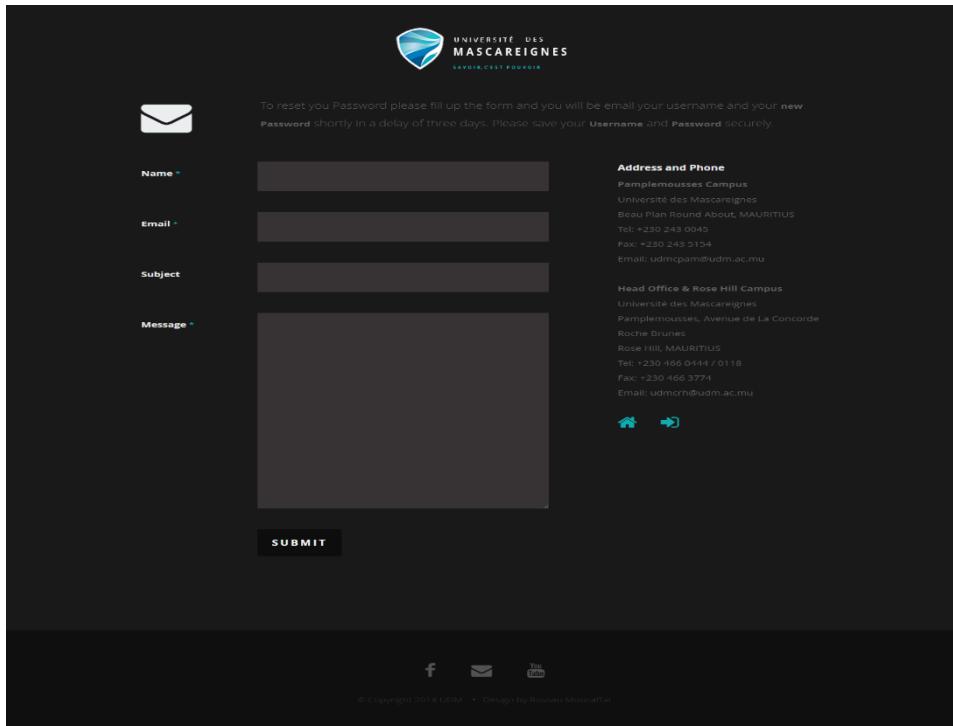


Figure 4- forget password

Before login users will have to choose between user login like has been shown below:

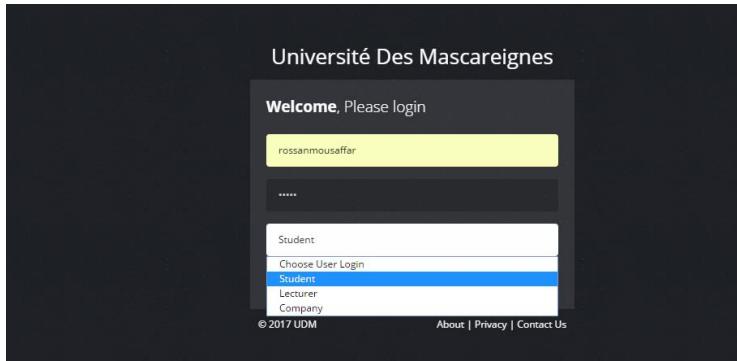


Figure 5- user role

If user log in into the system, the user dashboard will be like follows:

Figure 6- user profile/ home.php

There are different types of interfaces, find the following interfaces in below:

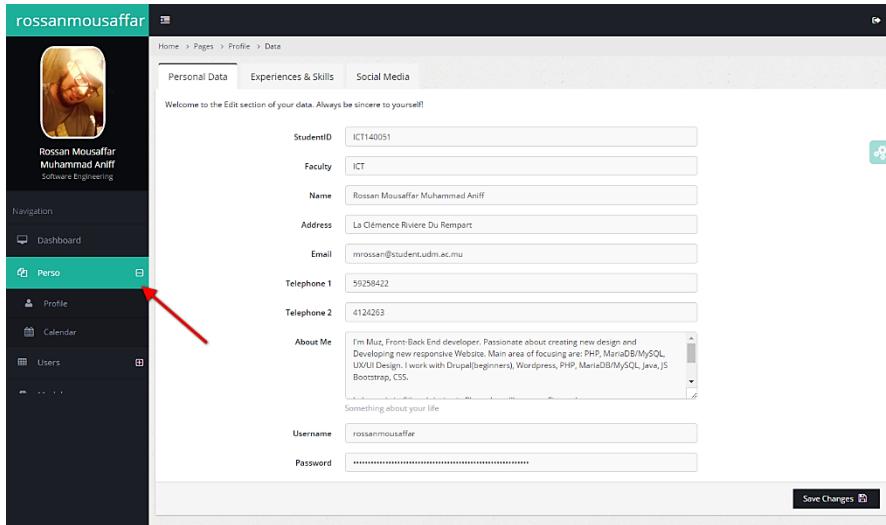


Figure 7- user profile i

The screenshot shows a user profile editing interface. At the top left is a profile picture of a person with blonde hair. Below it, the profile name is listed as "Rossan Mousaffar" and "Muhammad Aniff" under "Software Engineering". A navigation sidebar on the left includes links for "Dashboard", "Perso", "Users", "Module", and "My Site". The main content area has tabs for "Personal Data", "Experiences & Skills", and "Social Media". The "Experiences & Skills" tab is active.

**Experiences & Skills Tab Content:**

- Work Experience 1:** Department: Software Engineering; Title: Cim Finance Ltd; Experience: Customer Service Agent; Dates: November 2013 - June 2014; Description: Cim Finance Ltd a work where its really enriching for experience, Pleasant workplace and really good. Approved credit, Credit card, Leasing and others.
- Work Experience 2:** Title: North East Distribution; Experience: Stock Control; Dates: November 2015-December 2015; Description: Its was good by controlling and converse with clients. Control stock it and stock out and even sales.
- Work Experience 3:** Title: Web Ltd; Experience: Developer; Dates: December 2016 - January 2017; Description: Lots of experience, PHP developer and work on small projects using drupal and even learning a bit about seo.
- Education:** BSC Software Engineering; Dates: August 2014 - July 2017.
- Education 1:** Higher School Certificate; Dates: February 2011 - December 2013.
- Education 2:** School Certificate; Dates: January 2006 - December 2010.
- Other Qualification:** City & Guilds, Alliance Francaise, A+ Comptia.

**Skills Section:** Programming, Wordpress, HTML, SQL, CSS.

**Motto Section:** Motto: Smile; Description: Own Ideal.

**Quote Section:** Quote 1: Do Good; Description: Favorite phrase.

**Good At Section:** Good At: Programming, Team work, Decision making; Description: Best at what e.g project management, finance.

**Hobbies Section:** Hobbies: Swimming, Badminton; Description: What you like to do?

**Buttons:** Save Changes.

Figure 8- user profile ii

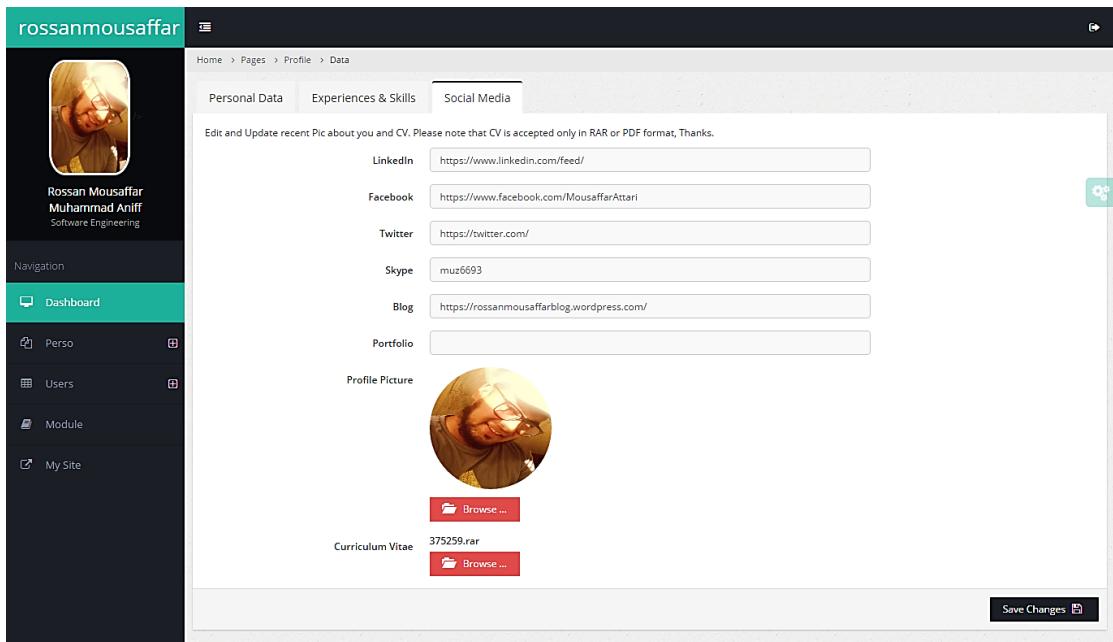


Figure 9- user profile iii

In the section “perso” there are two subsection. The user profile and calendar, in the user profile user can edit his records and save and in the calendar section users can add event on calendar like as shown below:

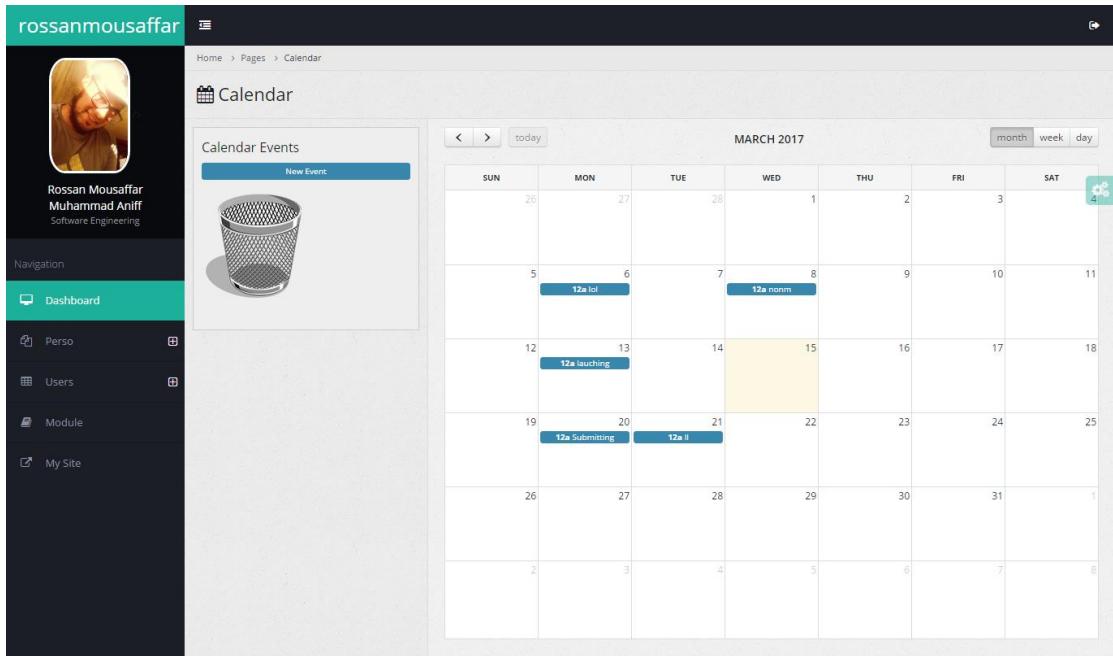
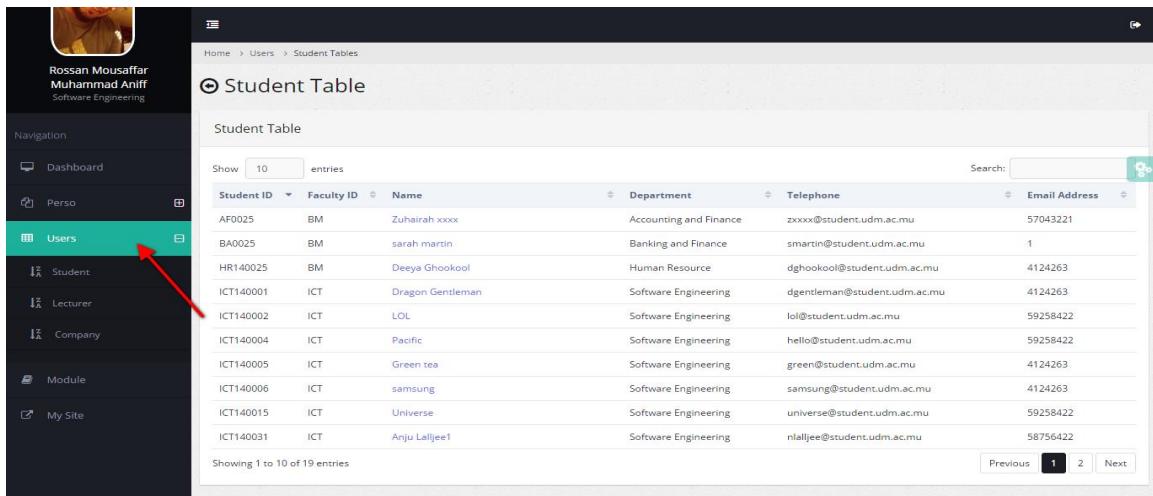


Figure 10- user calendar

Now the user section, it contains 3 subtype, student, lecturer, company.

Student section, list of all students like as been shown below:



Student ID	Faculty ID	Name	Department	Telephone	Email Address
AF0025	BM	Zuhairah xxxx	Accounting and Finance	xxxxx@student.udm.ac.mu	57043221
BA0025	BM	sarah martin	Banking and Finance	smartin@student.udm.ac.mu	1
HR140025	BM	Deeya Ghookool	Human Resource	dghookool@student.udm.ac.mu	4124263
ICT140001	ICT	Dragon Gentleman	Software Engineering	dgentleman@student.udm.ac.mu	4124263
ICT140002	ICT	LOL	Software Engineering	lol@student.udm.ac.mu	59258422
ICT140004	ICT	Pacific	Software Engineering	hello@student.udm.ac.mu	4124263
ICT140005	ICT	Green tea	Software Engineering	green@student.udm.ac.mu	4124263
ICT140006	ICT	samsung	Software Engineering	samsung@student.udm.ac.mu	4124263
ICT140015	ICT	Universe	Software Engineering	universe@student.udm.ac.mu	59258422
ICT140031	ICT	Anju Laljee1	Software Engineering	nlaljee@student.udm.ac.mu	58756422

Figure 11- user/student

Clicking on the hyperlink for e.g clicking on “dhanista” on the second page, it will redirect to the student page that has been shown below

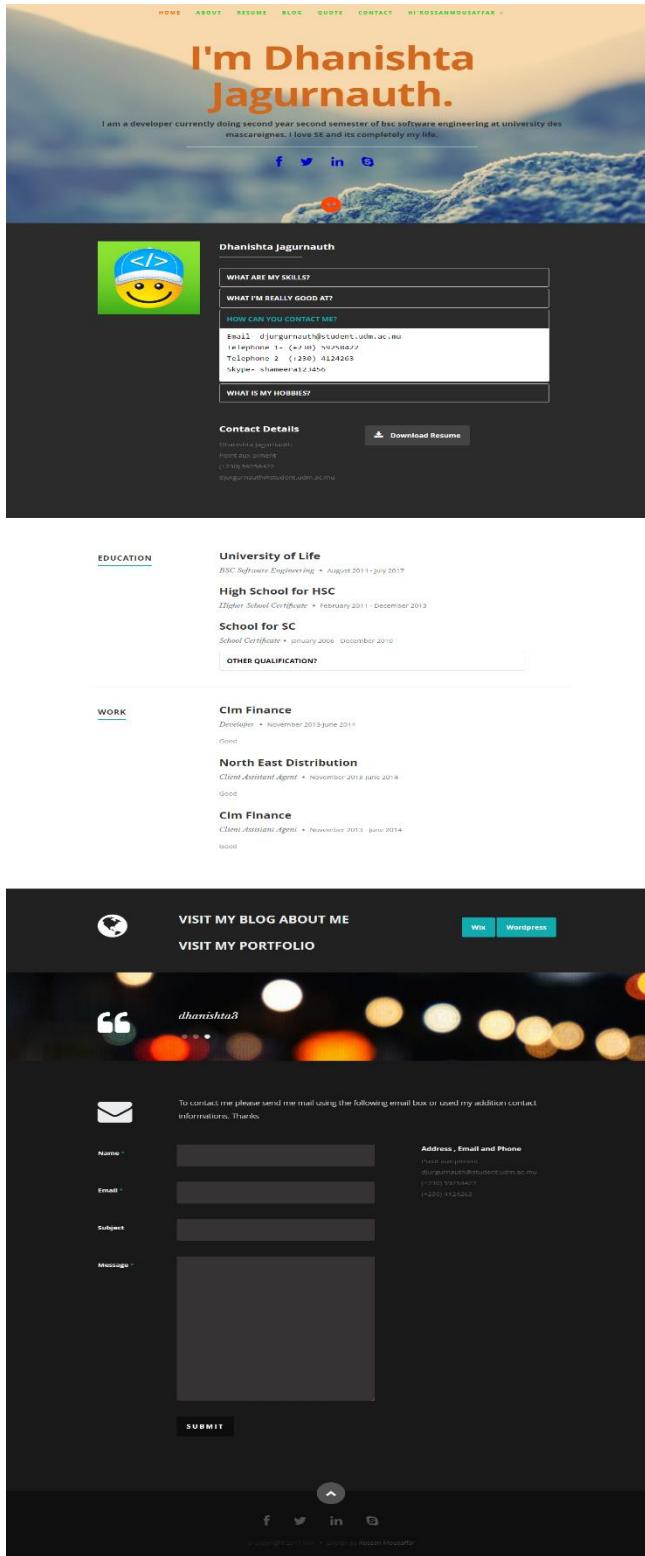


Figure 12- user site i

Now, if user want to go back just has to click on his name on the right upper corner and choose “setting” or even “logout” like has shown in the figure below

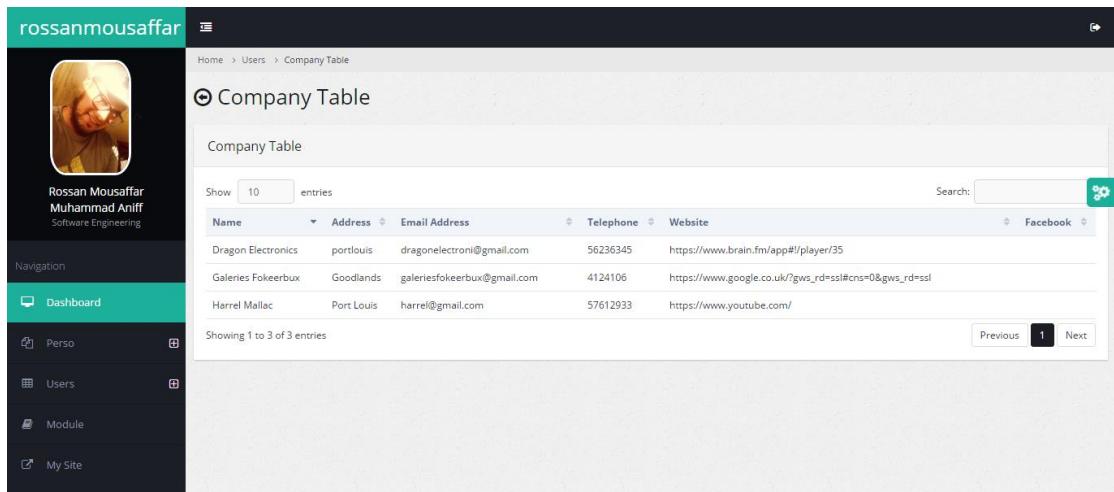


Figure 13- user site ii

Figure below shows the section of lecturer and company where student will only have access to the minimum record that has been shown on the table.

Lecturer ID	Faculty ID	Name	Department	Email Address
Ami025	ICT	Amic Seven	Software Engineering	samic@udm.ac.mu
Bai001	BM	Dr Nirmal Baichoo	Accounting and Finance	nbaichoo@udm.mail.ac.mu
Lau027	ICT	Lauthan Shameera	Software Engineering	slauthan@udm.ac.mu

Figure 14- lecturer table

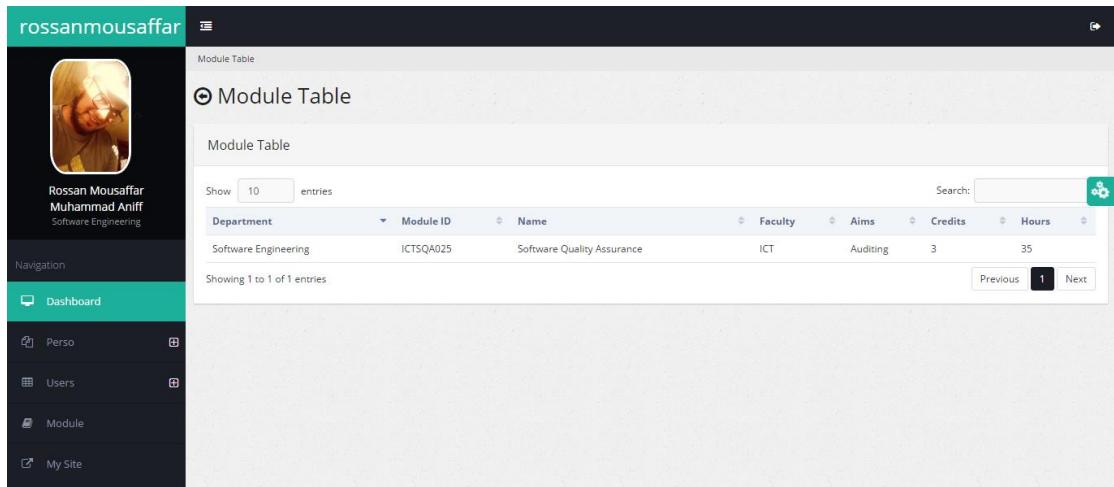


The screenshot shows a user profile for 'rossanmousaffar' at the top left. The main content area is titled 'Company Table'. It displays a table with columns: Name, Address, Email Address, Telephone, and Website. The data is as follows:

Name	Address	Email Address	Telephone	Website
Dragon Electronics	portlouis	dragonelectroni@gmail.com	56236345	<a href="https://www.brain.fm/app#/player/35">https://www.brain.fm/app#/player/35</a>
Galeries Fokeerbux	Goodlands	galeriesfokeerbux@gmail.com	4124106	<a href="https://www.google.co.uk/?gws_rd=ssl#qns=0&amp;gws_rd=ssl">https://www.google.co.uk/?gws_rd=ssl#qns=0&amp;gws_rd=ssl</a>
Harrel Mallac	Port Louis	harrel@gmail.com	57612933	<a href="https://www.youtube.com/">https://www.youtube.com/</a>

Figure 15- company table

Below shows the module table, module table depends on the department of the student.



The screenshot shows a user profile for 'rossanmousaffar' at the top left. The main content area is titled 'Module Table'. It displays a table with columns: Department, Module ID, Name, Faculty, Aims, Credits, and Hours. The data is as follows:

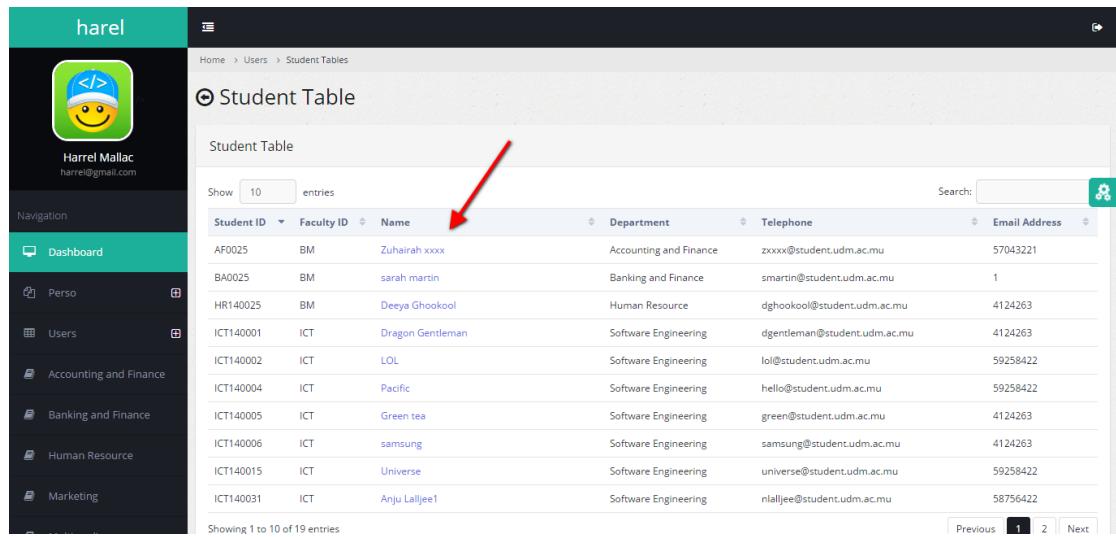
Department	Module ID	Name	Faculty	Aims	Credits	Hours
Software Engineering	ICTSQA025	Software Quality Assurance	ICT	Auditing	3	35

Figure 16- module table

And “my site” is the same as student site, just the details are different.

For the lecturer is particularly the same but lecturer have access to all students profile but cannot access lecturer profile.

About the company, company gets access to both student and lecturer profile and even have access to all modules of each department.



**Student Table**

Show 10 entries

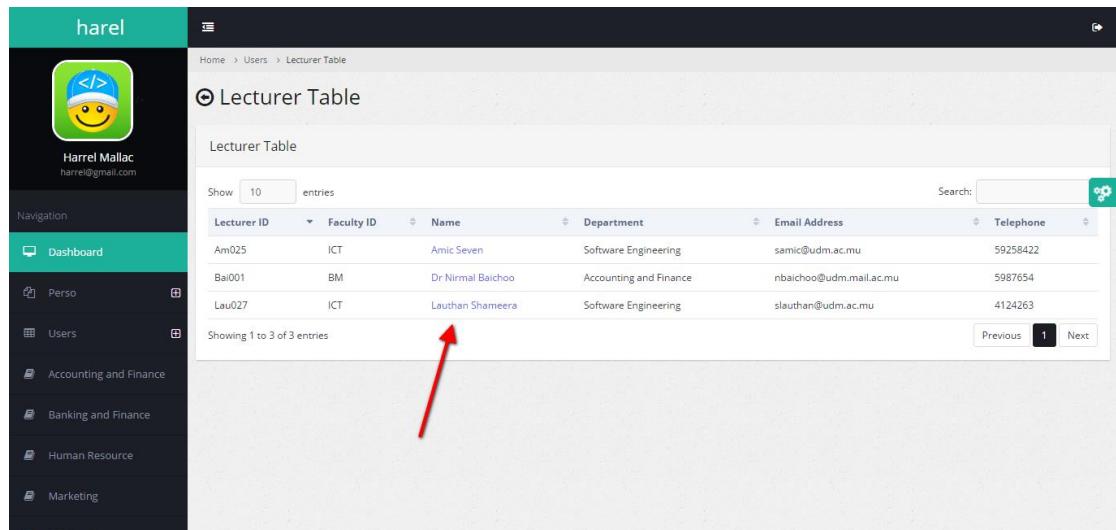
Search:

Student ID	Faculty ID	Name	Department	Telephone	Email Address
AF0025	BM	Zuhairah xxxx	Accounting and Finance	zxxxx@student.udm.ac.mu	57043221
BA0025	BM	sarah martin	Banking and Finance	smartin@student.udm.ac.mu	1
HR140025	BM	Deeya Ghookool	Human Resource	dghookool@student.udm.ac.mu	4124263
ICT140001	ICT	Dragon Gentleman	Software Engineering	dgentleman@student.udm.ac.mu	4124263
ICT140002	ICT	LOL	Software Engineering	lol@student.udm.ac.mu	59258422
ICT140004	ICT	Pacific	Software Engineering	hello@student.udm.ac.mu	59258422
ICT140005	ICT	Green tea	Software Engineering	green@student.udm.ac.mu	4124263
ICT140006	ICT	samsung	Software Engineering	samsung@student.udm.ac.mu	4124263
ICT140015	ICT	Universe	Software Engineering	universe@student.udm.ac.mu	59258422
ICT140031	ICT	Anju Lalljee1	Software Engineering	nlalljee@student.udm.ac.mu	58756422

Showing 1 to 10 of 19 entries

Previous 1 2 Next

Figure 17- company \ student list



**Lecturer Table**

Show 10 entries

Search:

Lecturer ID	Faculty ID	Name	Department	Email Address	Telephone
Am025	ICT	Amic Seven	Software Engineering	samic@udm.ac.mu	59258422
Bai001	BM	Dr Nirmal Baichoo	Accounting and Finance	nbaichoo@udm.mail.ac.mu	5987654
Lau027	ICT	Lauthan Shameera	Software Engineering	slauthan@udm.ac.mu	4124263

Showing 1 to 3 of 3 entries

Previous 1 Next

Figure 18- company \ lecturer list

Figure 16 and figure 17 shows that company “harel” has access to both user (lecturer and student) site.

Below shows the list of department that company get access to it.

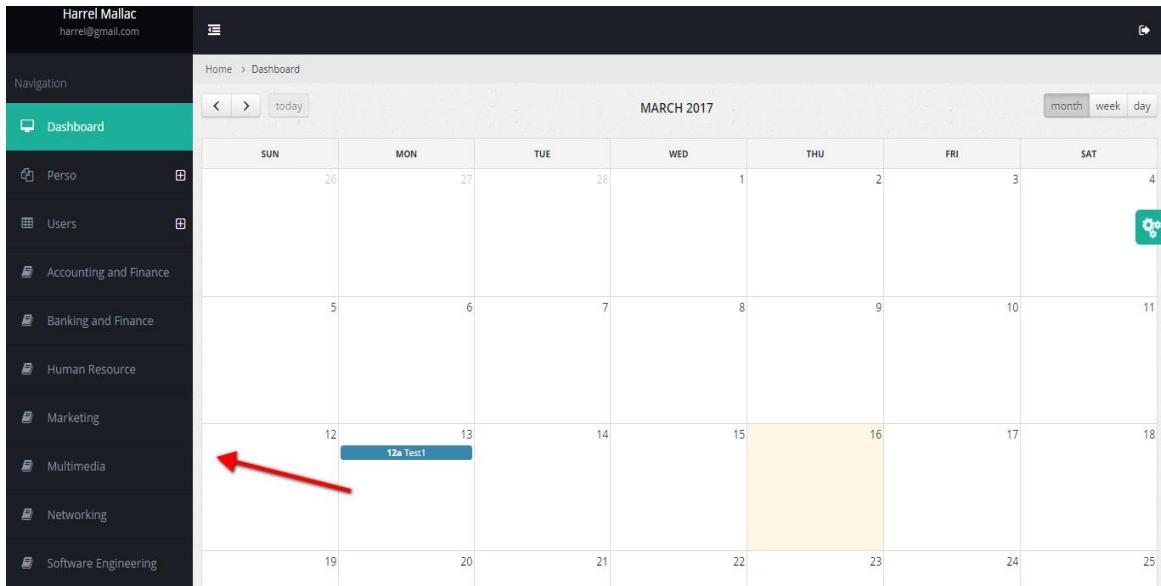


Figure 19- company \ department

Below shows the department table

The figure shows a table titled 'Module Table' for the 'Accounting and Finance' department. The table lists two entries: 'Accounting Services' and 'Information and Communication Technology'. The 'Accounting and Finance' link in the sidebar is circled in red, and a red arrow points to the same link in the module table header.

Department	Module ID	Name	Faculty	Aims	Credits	Hours
Accounting and Finance	Finan01	Accounting Services	BM	Hello ACCA	15	30
Accounting and Finance	SPM	Information and Communication Technology	BM	Hell	10	50

Figure 20- department table AF

Select Accounting and Finance on the left side bar then the module table will appear according the selection.

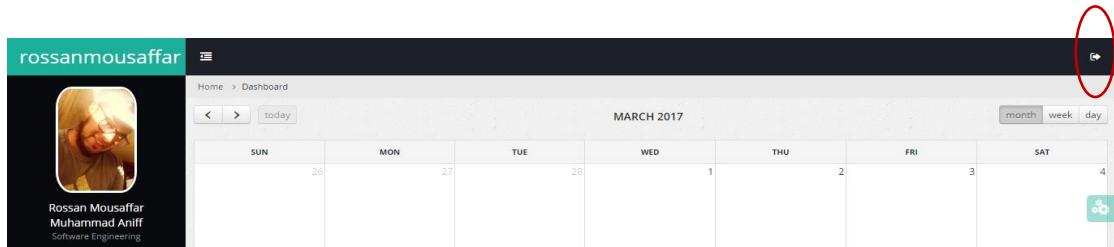
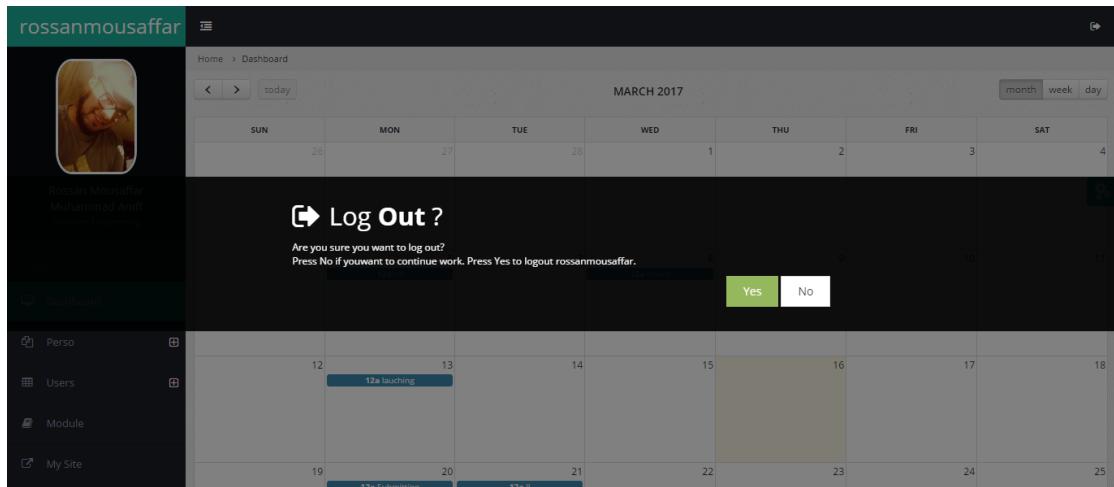


Figure 21- logout

The circle on the above pic is the logout button. Click on that a sound will be hear and its will appear as shown below:



#### 5.4.2 Administration Dashboard

Now the system even have an admin part and to get access to that part will be `localhost:8080/udm/adminudm/` then a login form will appear like have shown below

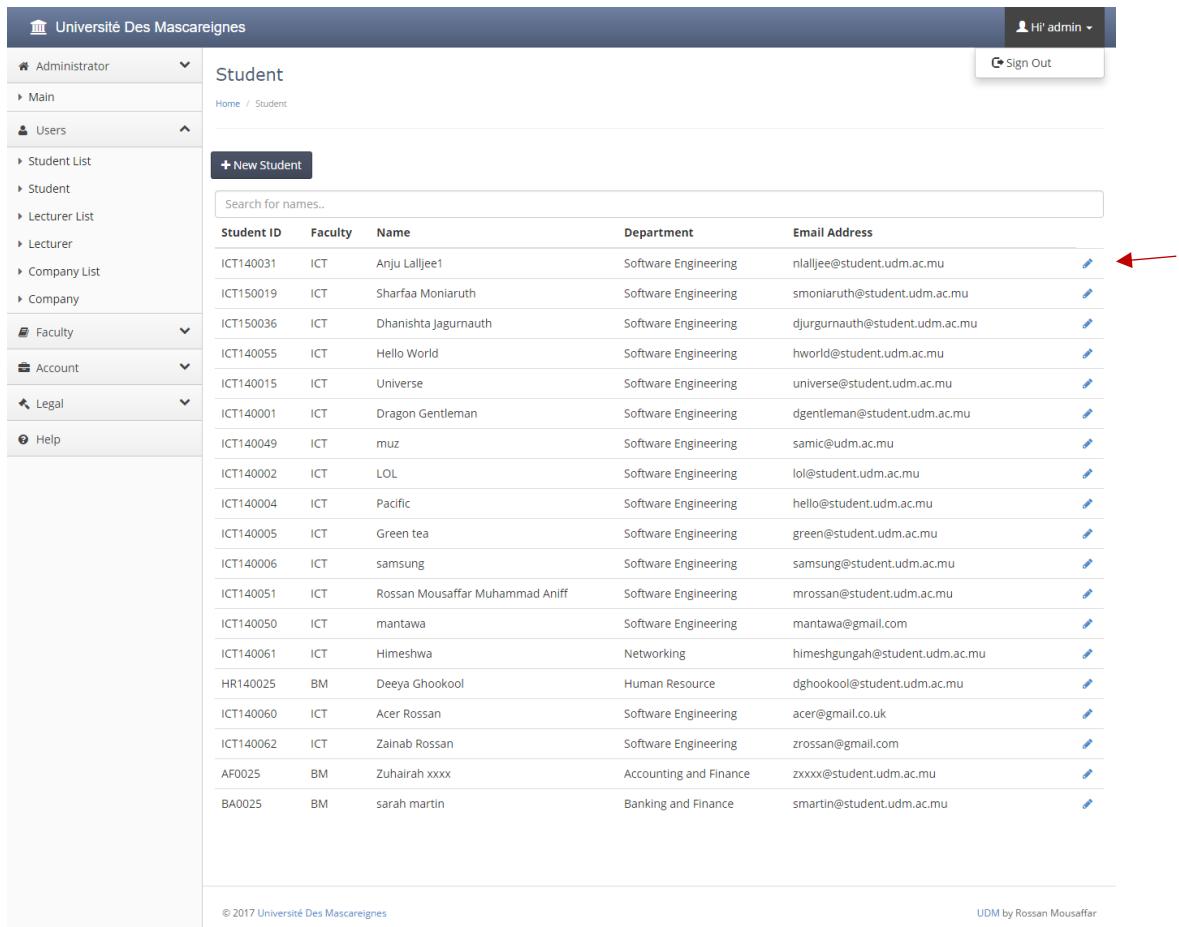


Figure 23- admin login

Once the admin access the admin part he has access to all the different users, department and faculty and can even modify records as per required or by authorized person.

The screenshot shows the Admin Dashboard of the Campus Recruitment System. The top navigation bar includes the university logo and name ('Université Des Mascareignes'), a user greeting ('Hi! admin'), and a dropdown menu for the current user ('Administrator'). The left sidebar contains a navigation menu with links for Main, Users, Faculty, Account, Legal, and Help. The main content area displays a calendar for March 2017. The calendar grid shows dates from Sunday, March 26, to Saturday, April 1. A specific date, March 15, is highlighted with a yellow background. Below the calendar is a blue button labeled 'New Event'. At the bottom of the page, there are copyright information ('© 2017 Université Des Mascareignes') and a credit ('UDM by Rossan Mousaffar').

Figure 24- admin dashboard



Student ID	Faculty	Name	Department	Email Address	Action
ICT140031	ICT	Anju Lalljee1	Software Engineering	nlalljee@student.udm.ac.mu	
ICT150019	ICT	Sharfaa Moniaruth	Software Engineering	smoniaruth@student.udm.ac.mu	
ICT150036	ICT	Dhanishta Jagurnauth	Software Engineering	djurgurnauth@student.udm.ac.mu	
ICT140055	ICT	Hello World	Software Engineering	hworld@student.udm.ac.mu	
ICT140015	ICT	Universe	Software Engineering	universe@student.udm.ac.mu	
ICT140001	ICT	Dragon Gentleman	Software Engineering	dgentleman@student.udm.ac.mu	
ICT140049	ICT	muz	Software Engineering	samic@udm.ac.mu	
ICT140002	ICT	LOL	Software Engineering	lol@student.udm.ac.mu	
ICT140004	ICT	Pacific	Software Engineering	hello@student.udm.ac.mu	
ICT140005	ICT	Green tea	Software Engineering	green@student.udm.ac.mu	
ICT140006	ICT	samsung	Software Engineering	samsung@student.udm.ac.mu	
ICT140051	ICT	Rossan Mousaffar Muhammad Aniff	Software Engineering	mrossan@student.udm.ac.mu	
ICT140050	ICT	mantawa	Software Engineering	mantawa@gmail.com	
ICT140061	ICT	Himeshwa	Networking	himeshgungah@student.udm.ac.mu	
HR140025	BM	Deeya Ghookool	Human Resource	dghookool@student.udm.ac.mu	
ICT140060	ICT	Acer Rossan	Software Engineering	acer@gmail.co.uk	
ICT140062	ICT	Zainab Rossan	Software Engineering	zrossan@gmail.com	
AF0025	BM	Zuhairah xxxx	Accounting and Finance	xxxx@student.udm.ac.mu	
BA0025	BM	sarah martin	Banking and Finance	smartin@student.udm.ac.mu	

Figure 25- admin\student list

Figure 25 shows the student list that are registered on the system. Moreover admin can click on his name on the right upper corner and a dropdown will appear where he can logout. Further user can click on the pencil for editing particular records. For e.g user click on “Rossan...” the following figures will show the edit section of it.

The screenshot shows the 'Edit Student' page of the 'Université Des Mascareignes' system. The left sidebar has a tree menu with 'Administrator' at the root, followed by 'Math', 'Users', 'Faculty', 'Account', 'Logout', and 'Help'. The main content area is titled 'Edit Student' and shows a form for a student named 'Rossan Mousaffar Muhammad Aniff'. The form includes fields for Name, Address, Email, Phone, and various experience sections like Work Experience, Education, and Skills. The 'Skills' section lists 'Programming', 'HTML/CSS', 'Precision marking', and 'Database'. The 'Work Experience' section includes titles like 'Customer Service Agent' and 'Software Engineering Intern'. The 'Education' section shows a 'BSC Software Engineering' degree from August 2014 to July 2017. The 'Skills' section also lists 'Wordpress', 'HTML/CSS', and 'JavaScript'. The 'About Me' section contains a bio about the student's interest in responsive website development.

Figure 26- admin | edit profile

It's the same for other two user's role.

The screenshot shows a web-based application for managing student records. On the left is a sidebar with various menu items: Administrator, Main, Users, Faculty, Account, Legal, and Help. The 'Administrator' item is currently selected. The main content area is titled 'Student' and shows a list of students with columns for Student ID, Faculty, Name, Department, and Email Address. At the top of this list is a button labeled '+ New Student'. Below the table is a search bar with the placeholder 'Search for names..'. Red annotations with arrows point to both the '+ New Student' button and the search bar.

Student ID	Faculty	Name	Department	Email Address
ICT140031	ICT	Anju Laljee1	Software Engineering	nlaljee@student.udm.ac.mu
ICT150019	ICT	Sharfaa Moniaruth	Software Engineering	smoniaruth@student.udm.ac.mu
ICT150036	ICT	Dhanishta Jagurnauth	Software Engineering	djurgurnauth@student.udm.ac.mu
ICT140055	ICT	Hello World	Software Engineering	hworld@student.udm.ac.mu
ICT140015	ICT	Universe	Software Engineering	universe@student.udm.ac.mu
ICT140001	ICT	Dragon Gentleman	Software Engineering	dgentleman@student.udm.ac.mu
ICT140049	ICT	muz	Software Engineering	samic@udm.ac.mu
ICT140002	ICT	LOL	Software Engineering	lol@student.udm.ac.mu

Figure 27- insert new student i

The screenshot shows the 'Add Student' form in the Campus Recruitment System. The form is titled 'Add Student' and includes the following fields:

- Profile:**
  - Student ID:** [Text input]
  - Faculty:** [Select dropdown]
  - Name:** [Text input]
  - Address:** [Text input]
  - Email:** [Text input]
  - Motto:** [Text input]
  - Quotes1:** [Text input]
  - Quotes2:** [Text input]
  - Goals1:** [Text input]
  - Goals2:** [Text input]
  - Hobbies:** [Text input]
- Work Experience:** [Section]
  - Title 1:** [Text input]
  - Start Date 1:** [Text input] (e.g. April 2007)
  - End Date 1:** [Text input] (e.g. August 2007)
  - Description 1:** [Text input]
  - Title 2:** [Text input]
  - Start Date 2:** [Text input] (e.g. April 2007)
  - End Date 2:** [Text input] (e.g. August 2007)
  - Description 2:** [Text input]
  - Title 3:** [Text input]
  - Start Date 3:** [Text input] (e.g. April 2007)
  - End Date 3:** [Text input] (e.g. August 2007)
  - Description 3:** [Text input]
- Education:** [Section]
  - Degree:** [Select dropdown] (e.g. BSC Software engineering)
  - Institution:** [Select dropdown] (e.g. August 2014 - July 2017)
  - Start Date:** [Text input] (e.g. August 2014)
  - End Date:** [Text input] (e.g. July 2017)
- Contact:** [Section]
  - Telephone 1:** [Text input]
  - Telephone 2:** [Text input]
  - LinkedIn:** [Text input]
  - Facebook:** [Text input]
  - Twitter:** [Text input]
  - Skype:** [Text input]
  - Address:** [Text input]
  - Skills:** [Text input]
- Department:** [Select dropdown] (e.g. Selected one)
- About Me:** [Text input]
- Portfolio:** [Text input]
- Username:** [Text input]
- Password:** [Text input]
- File Uploads:**
  - CV:** [Text input] (Choose File) No file chosen
  - Photo:** [Text input] (Choose File) No file chosen

At the bottom of the form are two buttons: 'Save' and 'Student List'.

Figure 28- insert new student ii

The figure below will show about the list of faculty

Faculty ID	Name	Dean	HOD
ICT	Information and Communication Technology	Mr Hossen Khan reza	Mrs S.Lauthan
BM	Business & Management	Dr Betchoo	Dr Peerbhai

Figure 29- faculty list

Click on “ new Faculty” it will redirect to adding section of adding faculty.

On the left side bar the list of department are available there and admin can choose departments to see module taught and even modify its details.

Module ID	Faculty	Name	Aims	Credits	Hours
Finan01	BM	Accounting Services	Hello ACCA	15	30
SPM	BM	Information and Communication Technology	Hell	10	50

Figure 30- list module in AF

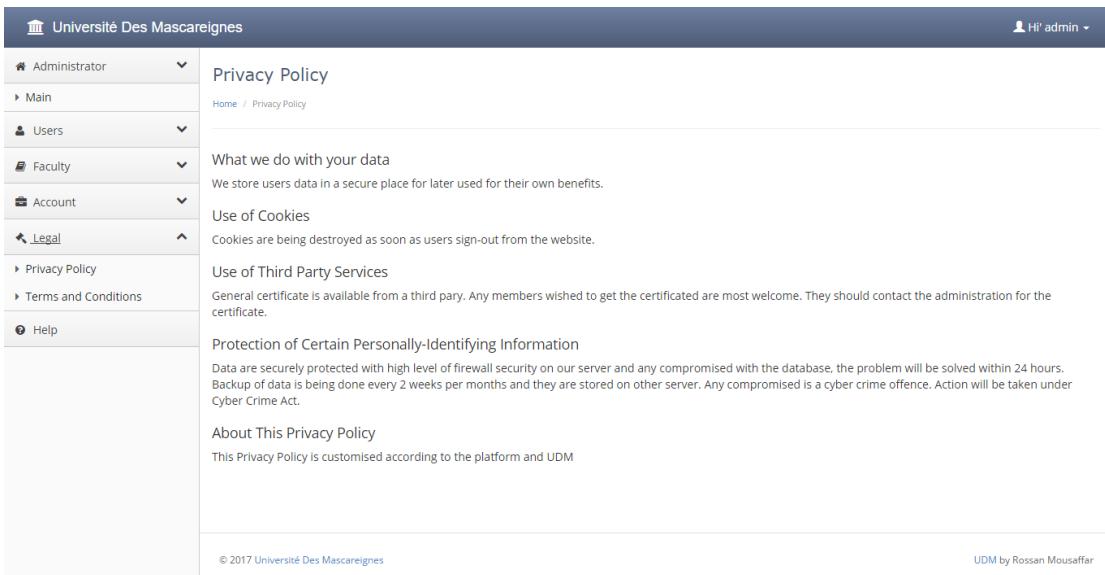
Click on the “new module” it will redirect to add new module section and click on the “pencil” on the right it will redirect to edit part of the particular module details.

Below shows the account part of the admin

The screenshot shows a user interface for updating an admin account. At the top, a header bar reads "Admin Account". Below it, a main section titled "Edit Account." contains three input fields. The first field has the value "admin", the second has "adminmuz@gmail.com", and the third is filled with a series of asterisks ("\*\*\*\*\*"). At the bottom of the form is a dark blue "UPDATE" button.

Figure 31- admin update

Change the records and click on “update” the admin records will be modify and saved.



*Figure 32- privacy section*

Figure above show the privacy section about the system and on the left sidebar under privacy there also the terms and condition about the system.

## 5.5 Methodologies

In analysis chapter 3, agile methodology would be used specifically the scrum method so while developing the system I made must to attempt the agile methodologies as it is more flexible and more rapid to develop the system.

Features have been insert into backlog according to its importance and then developed into the sprint after the sprint the developed part have been tested and added to the previous developed and tested part and have used pre-condition standard.

Moreover I have attempt to use the PHP , HTML and CSS coding standard

- Indenting and Line Length
- Control Structures
- Function Definitions
- Comments
- PHP Code Tags
- Variable Names
- Alignment of Declaration Blocks
- One Statement Per Line

- Short Methods or Functions

## 5.6 Major Problem

- Not enough resources for hosting the website on a server and even get a domain name.
- Research code need to be tested and them implement according to the website.
- Tight to follow planning as University semester is not yet finished.

## 5.7 Code adjustment and Implemented

sessionStudent.php

```
6  <?php
7
8      session_start();
9
10     require_once 'student.user.php';
11     $session = new USER();
12
13
14     if(!$session->is_loggedin())
15     {
16         // session no set redirects to login page
17         $session->redirect('home.php');
18     }
```

sessionAdmin.php

```
<?php
    session_start();

    require_once 'admin.user.php';
    $session = new USER();

    if(!$session->is_loggedin())
    {
        // session no set redirects to login page
        $session->redirect('sign-in.php');
    }
```

## User class for insertion

```

<?php
require_once('../dbconfig.php');
class USER1
{
private $conn;
public function __construct()
{
$database = new Database();
$db = $database->dbConnection();
$this->conn = $db;
}
public function runQuery($sql)
{
$stmt = $this->conn->prepare($sql);
return $stmt;
}
//-----
upload cv and pic-----
-----
// -----Student -----
-
public function registerStudent($studentID,$facultyID, $Name, $Address, $Email, $Motto, $Quote1, $Quote2, $GoodAt, $Hobbies,$WorkExperienceTitle1, $WorkExperience1, $WorkExperienceDataEnd1, $WorkExperienceAbout1, $WorkExperienceTitle2, $WorkExperience2, $WorkExperienceDataEnd2, $WorkExperienceAbout2, $WorkExperienceTitle3, $WorkExperience3, $WorkExperienceDataEnd3, $WorkExperienceAbout3, $Education, $EducationDataEnd, $Education1, $EducationDataEnd1, $Education2, $EducationDataEnd2, $OtherQualification, $Telephone1, $Telephone2, $LinkedIn, $Facebook, $Twitter, $Skype, $Blog,$Skills, $Department, $Aboutme, $Portfolio, $Username, $Password, $userpic, $cv)
{
try
{
$new_password = password_hash($Password, PASSWORD_DEFAULT);
$stmt = $this->conn->prepare("INSERT INTO student(StudentID, FacultyID, Name, Address, Email, Motto, Quote1, Quote2, GoodAt, Hobbies, WorkExperienceTitle1, WorkExperience1, WorkExperienceDataEnd1, WorkExperienceAbout1, WorkExperienceTitle2, WorkExp

```

```
erience2, WorkExperienceDataEnd2, WorkExperienceAbout2, W  
orkExperienceTitle3, WorkExperience3, WorkExperienceDataE  
nd3, WorkExperienceAbout3, Education, EducationDataEnd,  
Education1, EducationDataEnd1, Education2, EducationDataE  
nd2, OtherQualification, Telephone1, Telephone2, LinkedIn,  
Facebook, Twitter, Skype, Blog, Skills, Department, AboutM  
e, Portfolio, Username, Password, userPic, CV)  
VALUES( :studentID, :facultyID, :Name, :Address, :Email, :  
Motto, :Quotel, :Quote2, :GoodAt, :Hobbies, :WorkExperien  
ceTitle1, :WorkExperience1, :WorkExperienceDataEnd1, :Wor  
kExperienceAbout1, :WorkExperienceTitle2, :WorkExperience  
2, :WorkExperienceDataEnd2, :WorkExperienceAbout2, :WorkE  
xperienceTitle3, :WorkExperience3, :WorkExperienceDataEnd  
3, :WorkExperienceAbout3, :Education, :EducationDataEnd,  
:Education1, :EducationDataEnd1, :Education2, :EducationD  
ataEnd2, :OtherQualification, :Telephone1, :Telephone2, :L  
inkedIn, :Facebook, :Twitter, :Skype, :Blog, :Skills, :Dep  
artment, :Aboutme, :Portfolio, :Username, :Password, :use  
rPic, :CV));  
$stmt->bindparam(":studentID", $studentID);  
$stmt->bindparam(":facultyID", $facultyID);  
$stmt->bindparam(":Name", $Name);  
$stmt->bindparam(":Address", $Address);  
$stmt->bindparam(":Email", $Email);  
$stmt->bindparam(":Motto", $Motto);  
$stmt->bindparam(":Quotel", $Quotel);  
$stmt->bindparam(":Quote2", $Quote2);  
$stmt->bindparam(":GoodAt", $GoodAt);  
$stmt->bindparam(":Hobbies", $Hobbies);  
$stmt-  
>bindparam(":WorkExperienceTitle1", $WorkExperienceTitle1  
);  
$stmt->bindparam(":WorkExperience1", $WorkExperience1);  
$stmt-  
>bindparam(":WorkExperienceDataEnd1", $WorkExperienceData  
End1);  
$stmt-  
>bindparam(":WorkExperienceAbout1", $WorkExperienceAbout1  
);  
$stmt-  
>bindparam(":WorkExperienceTitle2", $WorkExperienceTitle2  
);  
$stmt->bindparam(":WorkExperience2", $WorkExperience2);  
$stmt-  
>bindparam(":WorkExperienceDataEnd2", $WorkExperienceData  
End2);
```

```
$stmt->bindparam(":WorkExperienceAbout2", $WorkExperienceAbout2);
$stmt->bindparam(":WorkExperienceTitle3", $WorkExperienceTitle3);
$stmt->bindparam(":WorkExperience3", $WorkExperience3);
$stmt->bindparam(":WorkExperienceDataEnd3", $WorkExperienceDataEnd3);
$stmt->bindparam(":WorkExperienceAbout3", $WorkExperienceAbout3);
$stmt->bindparam(":Education", $Education);
$stmt->bindparam(":EducationDataEnd", $EducationDataEnd);
$stmt->bindparam(":Education1", $Education1);
$stmt->bindparam(":EducationDataEnd1", $EducationDataEnd1);
$stmt->bindparam(":Education2", $Education2);
$stmt->bindparam(":EducationDataEnd2", $EducationDataEnd2);
$stmt->bindparam(":OtherQualification", $OtherQualification);
$stmt->bindparam(":Telephone1", $Telephone1);
$stmt->bindparam(":Telephone2", $Telephone2);
$stmt->bindparam(":LinkedIn", $LinkedIn);
$stmt->bindparam(":Facebook", $Facebook);
$stmt->bindparam(":Twitter", $Twitter);
$stmt->bindparam(":Skype", $Skype);
$stmt->bindparam(":Blog", $Blog);
$stmt->bindparam(":Skills", $Skills);
$stmt->bindparam(":Department", $Department);
$stmt->bindparam(":Aboutme", $Aboutme);
$stmt->bindparam(":Portfolio", $Portfolio);
$stmt->bindparam(":Username", $Username);
$stmt->bindparam(":Password", $new_password);
$stmt->bindparam(":userPic", $userpic);
$stmt->bindparam(":CV", $cv);
$stmt->execute();
return $stmt;
}
catch(PDOException $e)
{
echo $e->getMessage();
}
```

```

// ----- Company -----
-----
-- 
public function registerCompany($companyID,$Name, $Address, $Email, $Motto, $Telephone,$Facebook, $Skype, $Blog, $Website, $Username, $Password, $userpic, $cv)
{
try
{
$new_password = password_hash($Password, PASSWORD_DEFAULT);
$stmt = $this->conn->prepare("INSERT INTO company(companyID, name, address, email, motto, telephone, facebook,skype, blog, website,username, password, userPic,CV)
VALUES( :companyID,:Name, :Address, :Email, :Motto, :Telephone, :Facebook, :Skype, :Blog, :Website, :Username, :Password, :userPic,:CV)");
$stmt->bindparam(":companyID", $companyID);
$stmt->bindparam(":Name", $Name);
$stmt->bindparam(":Address", $Address);
$stmt->bindparam(":Email", $Email);
$stmt->bindparam(":Motto", $Motto);
$stmt->bindparam(":Telephone", $Telephone);
$stmt->bindparam(":Facebook", $Facebook);
$stmt->bindparam(":Skype", $Skype);
$stmt->bindparam(":Blog", $Blog);
$stmt->bindparam(":Website", $Website);
$stmt->bindparam(":Username", $Username);
$stmt->bindparam(":Password", $new_password);
$stmt->bindparam(":userPic", $userpic);
$stmt->bindparam(":CV", $cv);
$stmt->execute();
return $stmt;
}
catch(PDOException $e)
{
echo $e->getMessage();
}
}
// -----Lecturer -----
-- 
public function registerLecturer($lecturerID,$facultyID, $Name, $Address, $Email, $Motto, $Quote1, $Quote2, $GoodAt,$Hobbies,$WorkExperienceTitle1, $WorkExperience1, $Work

```

```

ExperienceDataEnd1, $WorkExperienceAbout1, $WorkExperienc
eTitle2, $WorkExperience2, $WorkExperienceDataEnd2, $Work
ExperienceAbout2, $WorkExperienceTitle3, $WorkExperience3
, $WorkExperienceDataEnd3, $WorkExperienceAbout3, $Educa
tion, $EducationDataEnd, $Education1, $EducationDataEnd1,
$Education2, $EducationDataEnd2, $OtherQualification, $T
elephone, $LinkedIn,$Twitter, $Skype, $Blog, $Website, $S
kills, $Department, $Aboutme, $Portfolio, $Username, $Pas
sword , $userpic, $cv)
{
try
{
$new_password = password_hash($Password, PASSWORD_DEFAULT
);
$stmt = $this->conn-
>prepare("INSERT INTO lecturer(LecturerID, FacultyID, Nam
e, Address, Email, Motto, Quotet1, Quote2, GoodAt,Hobbies,
WorkExperienceTitle1, WorkExperience1, WorkExperienceData
End1, WorkExperienceAbout1, WorkExperienceTitle2, WorkExp
erience2, WorkExperienceDataEnd2, WorkExperienceAbout2, W
orkExperienceTitle3, WorkExperience3, WorkExperienceDataE
nd3, WorkExperienceAbout3, Education, EducationDataEnd,
Education1, EducationDataEnd1, Education2, EducationDataE
nd2, OtherQualification, Telephone, LinkedIn, Twitter, Sk
ype, Blog, Website, Skills, Department, AboutMe, Portfoli
o, Username, Password, userPic,CV)
VALUES( :lecturerID,:facultyID, :Name, :Address, :Email,
:Motto, :Quotet1, :Quote2, :GoodAt,:Hobbies,:WorkExperienc
eTitle1, :WorkExperience1, :WorkExperienceDataEnd1, :Work
ExperienceAbout1, :WorkExperienceTitle2, :WorkExperience2
, :WorkExperienceDataEnd2, :WorkExperienceAbout2, :WorkEx
perienceTitle3, :WorkExperience3, :WorkExperienceDataEnd3
, :WorkExperienceAbout3, :Education, :EducationDataEnd,
:Education1, :EducationDataEnd1, :Education2, :EducationD
ataEnd2, :OtherQualification, :Telephone, :LinkedIn, :Twi
tter, :Skype, :Blog, :Website, :Skills, :Department, :Abo
utme, :Portfolio, :Username, :Password, :userPic,:CV)");
$stmt->bindparam(":lecturerID", $lecturerID);
$stmt->bindparam(":facultyID", $facultyID);
$stmt->bindparam(":Name", $Name);
$stmt->bindparam(":Address", $Address);
$stmt->bindparam(":Email", $Email);
$stmt->bindparam(":Motto", $Motto);
$stmt->bindparam(":Quotet1", $Quotet1);
$stmt->bindparam(":Quote2", $Quote2);
$stmt->bindparam(":GoodAt", $GoodAt);

```

```
$stmt->bindparam(":Hobbies", $Hobbies);
$stmt-
>bindparam(":WorkExperienceTitle1", $WorkExperienceTitle1);
);
$stmt->bindparam(":WorkExperience1", $WorkExperience1);
$stmt-
>bindparam(":WorkExperienceDataEnd1", $WorkExperienceDataEnd1);
$stmt-
>bindparam(":WorkExperienceAbout1", $WorkExperienceAbout1);
);
$stmt-
>bindparam(":WorkExperienceTitle2", $WorkExperienceTitle2);
);
$stmt->bindparam(":WorkExperience2", $WorkExperience2);
$stmt-
>bindparam(":WorkExperienceDataEnd2", $WorkExperienceDataEnd2);
$stmt-
>bindparam(":WorkExperienceAbout2", $WorkExperienceAbout2);
);
$stmt-
>bindparam(":WorkExperienceTitle3", $WorkExperienceTitle3);
);
$stmt->bindparam(":WorkExperience3", $WorkExperience3);
$stmt-
>bindparam(":WorkExperienceDataEnd3", $WorkExperienceDataEnd3);
$stmt-
>bindparam(":WorkExperienceAbout3", $WorkExperienceAbout3);
);
$stmt->bindparam(":Education", $Education);
$stmt->bindparam(":EducationDataEnd", $EducationDataEnd);
$stmt->bindparam(":Education1", $Education1);
$stmt-
>bindparam(":EducationDataEnd1", $EducationDataEnd1);
$stmt->bindparam(":Education2", $Education2);
$stmt-
>bindparam(":EducationDataEnd2", $EducationDataEnd2);
$stmt-
>bindparam(":OtherQualification", $OtherQualification);
$stmt->bindparam(":Telephone", $Telephone);
$stmt->bindparam(":LinkedIn", $LinkedIn);
$stmt->bindparam(":Twitter", $Twitter);
$stmt->bindparam(":Skype", $Skype);
$stmt->bindparam(":Blog", $Blog);
```

```

$stmt->bindParam(":Website", $Website);
$stmt->bindParam(":Skills", $Skills);
$stmt->bindParam(":Department", $Department);
$stmt->bindParam(":Aboutme", $Aboutme);
$stmt->bindParam(":Portfolio", $Portfolio);
$stmt->bindParam(":Username", $Username);
$stmt->bindParam(":Password", $new_password);
$stmt->bindParam(":userPic", $userpic);
$stmt->bindParam(":CV", $cv);
$stmt->execute();
return $stmt;
}
catch(PDOException $e)
{
echo $e->getMessage();
}
}
?
?>
```

Class for login of admin

```

<?php
require_once('../dbconfig.php');
class USER
{
private $conn;

public function __construct()
{
    $database = new Database();
    $db = $database->dbConnection();
    $this->conn = $db;
}

public function runQuery($sql)
{
    $stmt = $this->conn->prepare($sql);
    return $stmt;
}

public function register($uname,$umail,$upass)
{
    try
    {
```

```

        $new_password = password_hash($upass, PASSWORD_DEFAULT);

        $stmt = $this->conn->prepare("INSERT INTO admin(user_name,user_email,user_pass)
VALUES (:uname, :umail, :upass)");

        $stmt->bindParam(":uname", $uname);
        $stmt->bindParam(":umail", $umail);
        $stmt->bindParam(":upass", $new_password);

        $stmt->execute();

        return $stmt;
    }
    catch(PDOException $e)
    {
        echo $e->getMessage();
    }
}

public function doLogin($uname,$umail,$upass)
{
    try
    {
        $stmt = $this->conn->prepare("SELECT user_id, user_name, user_email, user_pass FROM admin WHERE user_name=:uname OR user_email=:umail");
        $stmt->execute(array(':uname'=>$uname, ':umail'=>$umail));
        $userRow=$stmt->fetch(PDO::FETCH_ASSOC);
        if($stmt->rowCount() == 1)
        {
            if(password_verify($upass, $userRow['user_pass']))
            {
                $_SESSION['user_session'] = $userRow['user_id'];
                return true;
            }
            else
            {

```

```

        return false;
    }
}
catch (PDOException $e)
{
    echo $e->getMessage();
}
}

public function is_loggedin()
{
    if (isset($_SESSION['user_session']))
    {
        return true;
    }
}

public function redirect($url)
{
    header("Location: $url");
}

public function doLogout()
{
    session_destroy();
    unset($_SESSION['user_session']);
    return true;
}
?>
```

### Check User login

```

<?php
error_reporting( ~E_NOTICE ); // avoid notice
require_once("session.php");
require_once("admin.user.php");
$auth_user = new USER();
$user_id = $_SESSION['user_session'];
$stmt = $auth_user-
>runQuery("SELECT * FROM admin WHERE user_id=:user_id");
$stmt->execute(array(":user_id"=>$user_id));
$userRow=$stmt->fetch(PDO::FETCH_ASSOC);
?>
```

### Javascript for calendar

```
<script>
$(document).ready(function() {
    var zone = "05:30"; //Change this to your timezone
    $.ajax({
        url: 'process.php',
        type: 'POST', // Send post data
        data: 'type=fetch',
        async: false,
        success: function(s) {
            json_events = s;
        }
    });
    var currentMousePos = {
        x: -1,
        y: -1
    };
    jQuery(document).on("mousemove", function (event) {
        currentMousePos.x = event.pageX;
        currentMousePos.y = event.pageY;
    });
    /* initialize the external events
    -----
    -----
    */
$('#external-events .fc-event').each(function() {
    // store data so the calendar knows to render an event upon drop
    $(this).data('event', {
        title: $.trim($(this).text()), // use the element's text as the event title
        stick: true // maintain when user navigates (see docs on the renderEvent method)
    });
    // make the event draggable using jquery UI
    $(this).draggable({
        zIndex: 999,
        revert: true, // will cause the event to go back to its original position after the drag
        revertDuration: 0 // original position
    });
});

```

```

/* initialize the calendar
-----
-----*/
$('#calendar').fullCalendar({
    events: JSON.parse(json_events),
    //events: [{"id": "14", "title": "New Event", "start": "2015-01-24T16:00:00+04:00", "allDay": false}],
    utc: true,
    header: {
        left: 'prev,next today',
        center: 'title',
        right: 'month,agendaWeek,agendaDay'
    },
    editable: true,
    droppable: true,
    slotDuration: '00:30:00',
    eventReceive: function(event) {
        var title = event.title;
        var start = event.start.format("YYYY-MM-DD[T]HH:mm:ss");
        $.ajax({
            url: 'process.php',
            data: 'type=new&title=' + title + '&startdate=' + start + '&zone=' + zone,
            type: 'POST',
            dataType: 'json',
            success: function(response) {
                event.id = response.eventid;
                $('#calendar').fullCalendar('updateEvent', event);
            },
            error: function(e) {
                console.log(e.responseText);
            }
        });
        $('#calendar').fullCalendar('updateEvent', event);
        console.log(event);
    },
    eventDrop: function(event, delta, revertFunc) {
        var title = event.title;
        var start = event.start.format();
        var end = (event.end == null) ? start : event.end.format();
        $.ajax({
            url: 'process.php',
            data: 'type=resetdate&title=' + title + '&start=' + start + '&end=' + end + '&eventid=' + event.id,
        });
    }
});

```

```

        type: 'POST',
        dataType: 'json',
        success: function(response) {
            i
f(response.status != 'success')
            revertFunc();
        },
        error: function(e) {
            revertFunc();
            alert('Error processing your request: '+e.responseText);
        }
    },
    eventClick: function(event, jsEvent, view) {
        console.log(event.id);
        var title = prompt('Event Title:', event.title, { buttons: { Ok: true, Cancel: false} });
        if (title) {
            event.title = title;
            console.log('type=changetitle&title=' + title + '&eventid=' + event.id);
            $.ajax({
                url: 'process.php',
                data: 'type=changetitle&title=' + title + '&eventid=' + event.id,
                type: 'POST',
                dataType: 'json',
                success: function(response) {

                    if(response.status == 'success')
                        $('#calendar').fullCalendar('updateEvent', event);
                    },
                    error: function(e) {
                        alert('Error processing your request: '+e.responseText);
                    }
                });
        }
    },
    eventResize: function(event, delta, revertFunc) {
        console.log(event);
        var title = event.title;
        var end = event.end.format();
        var start = event.start.format();
        $.ajax({
            url: 'process.php',

```

```
        data: 'type=resetdate&title=' + title + '&start=' + start + '&end=' + end + '&eventid=' + event.id,
        type: 'POST',
        dataType: 'json',
        success: function(response) {
            if(response.status != 'success')
                revertFunc();
            },
            error: function(e) {
                revertFunc();
                alert('Error processing your request: '+e.responseText);
            }
        },
        eventDragStop: function (event, jsEvent, ui, view) {
            if (isElemOverDiv()) {
                var con = confirm('Are you sure to delete this event permanently?');
                if(con == true) {
                    $.ajax({
                        url: 'process.php',
                        data: 'type=remove&eventid=' + event.id,
                        type: 'POST',
                        dataType: 'json',
                        success: function(response) {
                            console.log(response);
                            if(response.status == 'success'){
                                $('#calendar').fullCalendar('removeEvents');
                                getFreshEvents();
                            }
                        },
                        error: function(e){
                            alert('Error processing your request: '+e.responseText);
                        }
                    });
                }
            }
        }
    );
}
```

```

function getFreshEvents() {
    $.ajax({
        url: 'process.php',
        type: 'POST', // Send post data
        data: 'type=fetch',
        async: false,
        success: function(s) {
            freshevents = s;
        }
    });
}

$('#calendar').fullCalendar('addEventSource', JSON.parse(freshevents));
}

function isElemOverDiv() {
    var trashEl = jQuery('#trash');
    var ofs = trashEl.offset();
    var x1 = ofs.left;
    var x2 = ofs.left + trashEl.outerWidth(true);
    var y1 = ofs.top;
    var y2 = ofs.top + trashEl.outerHeight(true);
    if (currentMousePos.x >= x1 && currentMousePos.x <= x2 && currentMousePos.y >= y1 && currentMousePos.y <= y2) {
        return true;
    }
    return false;
}
}) ;
</script>
</head>
<body class=" theme-blue">
    <!-- Demo page code -->
    <script type="text/javascript">
        $(function() {
            var match = document.cookie.match(new RegExp('color=([^;]+);'));
            if (match) var color = match[1];
            if (color) {
                $('body').removeClass(function(index, css) {
                    return (css.match(/\btheme-\s+/g) || []).join(' ');
                })
                $('body').addClass('theme-' + color);
            }
            $('[data-popover="true"]').popover({
                html: true
            });
        });
    </script>

```

```

    });
</script>
<script type="text/javascript">
$(function() {
var uls = $('.sidebar-nav > ul > *').clone();
uls.addClass('visible-xs');
$('#main-menu').append(uls.clone());
});
</script>

```

### Process for calendar

```

<?php
include('config.php');
$type = $_POST['type'];
if($type == 'new')
{
    $startdate = $_POST['startdate'].'+'. $_POST['zone'];
    $title = $_POST['title'];
    $insert = mysqli_query($con,"INSERT INTO calendaradmin(title, startdate, enddate, allDay) VALUES('$title','$startdate','$startdate','false')");
    $lastid = mysqli_insert_id($con);
    echo json_encode(array('status'=>'success','eventid'=>$lastid));
}
if($type == 'changetitle')
{
    $eventid = $_POST['eventid'];
    $title = $_POST['title'];
    $update = mysqli_query($con,"UPDATE calendaradmin SET title='$title' where id='$eventid'");
    if($update)
        echo json_encode(array('status'=>'success'));
    else
        echo json_encode(array('status'=>'failed'));
}
if($type == 'resetdate')
{
    $title = $_POST['title'];
    $startdate = $_POST['start'];
    $enddate = $_POST['end'];
    $eventid = $_POST['eventid'];
}

```

```
$update = mysqli_query($con, "UPDATE calendaradmin SET title='$title', startdate = '$startdate', enddate = '$enddate' WHERE id='$eventid'");
if($update)
    echo json_encode(array('status'=>'success'));
else
    echo json_encode(array('status'=>'failed'));
}
if($type == 'remove')
{
    $eventid = $_POST['eventid'];
    $delete = mysqli_query($con, "DELETE FROM calendarAdmin WHERE id='$eventid'");
    if($delete)
        echo json_encode(array('status'=>'success'));
    else
        echo json_encode(array('status'=>'failed'));
}
if($type == 'fetch')
{
    $events = array();
    $query = mysqli_query($con, "SELECT * FROM calendaradmin");
    while($fetch = mysqli_fetch_array($query, MYSQLI_ASSOC))
    {
        $e = array();
        $e['id'] = $fetch['id'];
        $e['title'] = $fetch['title'];
        $e['start'] = $fetch['startdate'];
        $e['end'] = $fetch['enddate'];
        $allday = ($fetch['allDay'] == "true") ? true : false;
        $e['allDay'] = $allday;
        array_push($events, $e);
    }
    echo json_encode($events);
}
?>
```

### Is Login

```
<?php
session_start();
require_once("admin.user.php");
$login = new USER();
if($login->is_loggedin() != "")
{
    $login->redirect('index.php');
```

### User Picture

```
$imgFile = $_FILES['user_image']['name'];
$tmp_dir = $_FILES['user_image']['tmp_name'];
$imgSize = $_FILES['user_image']['size'];
```

```
$upload_dir = '../user_image/'; // upload directory
$imgExt = strtolower(pathinfo($imgFile,PATHINFO_EXTENSION));
// valid image extensions
$valid_extensions = array('jpeg', 'jpg', 'png', 'gif','pdf'); // valid extensions
// rename uploading image
$userpic = rand(1000,1000000).".".$imgExt;
// allow valid image file formats
if(in_array($imgExt, $valid_extensions))
{
// Check file size '5MB'
if($imgSize < 5000000)
{
move_uploaded_file($tmp_dir,$upload_dir.$userpic);
}
else
{
$error[] = "Sorry, your file is too large.";
}
}
else
{
$error[] = "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
}
```

## 5.8 Validation Code

Program name	Insert new Student
Objective	To check if content entered does not contain unwanted text or characters
	<pre> else if (\$Name != "") {     if (!preg_match("/^ [a-zA-Z ]*\$/", \$Name)) {         \$error[] = "Only letters and white space allowed in Name!";     } } else if (\$Name == "") {     \$error[] = "Name Required"; } </pre>
Logic used	Name of student cannot contain numbers by using preg match and not empty

Program name	Insert Telephone Number
Objective	To check if content entered does not contain text or characters
	<pre> \$Telephone1 = IS_NUMERIC(\$_POST['Telephone1']); \$Telephone2 = IS_NUMERIC(\$_POST['Telephone2']); if (\$Telephone1 == "") {     \$Error[] = "Telephone1 required"; } else if (\$Telephone2 == "") {     \$Error[] = "Telephone2 required"; } </pre>
Logic used	Telephone Number cannot contain text

Program name	Insert Facebook URL
Objective	To check if it is match all the characteristics of being an URL
	<pre> \$Facebook = \$_POST["Facebook"]; </pre>

	<pre>elseif(!filter_var(\$Facebook, FILTER_VALIDATE_URL) === true) {     \$error[] = "\$Facebook is not a valid address"; }</pre>
Logic used	URL should be valid so that to travel to the URL

Program name	Document type that can be upload
Objective	To check is the document upload is correct format <pre>\$valid_extensions_file= array('rar','pdf'); // valid extensions</pre>
Logic used	RAR can contain many document inside PDF cannot be modified and it keep the format that is has been written

Program name	Document type that can be upload
Objective	To check is the document upload is correct format <pre>\$valid_extensions_file= array('rar','pdf'); // valid extensions</pre>
Logic used	RAR can contain many document inside PDF cannot be modified and it keep the format that is has been written

Program name	Document type that can be upload
Objective	To check is the document upload is correct format <pre>\$valid_extensions_file= array('rar','pdf'); // valid extensions</pre>
Logic used	RAR can contain many document inside PDF cannot be modified and it keep the format that is has been written

Program name	Login
Objective	To able the users login at their specific role
	<pre> if (\$user=='Student') {     if(\$login-&gt;doLogin(\$uname,\$upass))     {         \$login-&gt;redirect('index.php');     }     else     {         \$error[] = "Wrong Details about USERNAME or PASSWORD, Please try again!!";     } }else if (\$user == 'Lecturer') {     if(\$login-&gt;doLogin1(\$uname,\$upass))     {         \$login-&gt;redirect('indexLecturer.php');     }     else     {         \$error[] = "Wrong Details about USERNAME or PASSWORD, Please try again!!";     } } else if (\$user=="Company") {     if(\$login-&gt;doLogin2(\$uname,\$upass))     {         \$login-&gt;redirect('indexCompany.php');     }     else     {         \$error[] = "Wrong Details about USERNAME or PASSWORD, Please try again!!";     } } } </pre>
Logic used	If ... else statement and logic branch



## Table of content

6. Testing .....	105
6.1 Testing of the different automations .....	105
6.1.1 Unit testing .....	105
6.1.2 Integration Strategies .....	109
6.1.3 System Testing .....	113
6.1.4 Acceptance Testing.....	113

## 6. Testing

This section outlines the different testing techniques applied to the campus recruitment system.

The test data and the results obtained for each validation and interface will be provided.

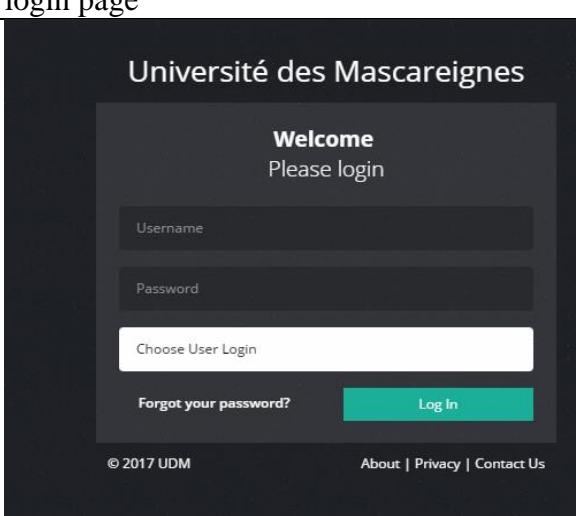
### 6.1 Testing of the different automations

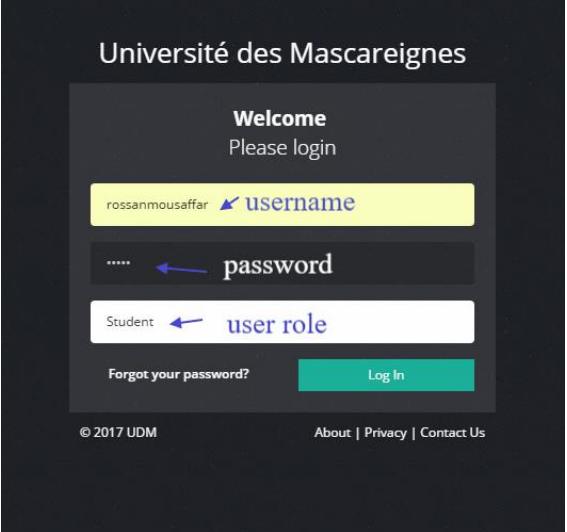
Different testing techniques have been applied to the different automations to ensure the requirements are met. Unit and integration testing will be used as testing methodologies in order to test for each component implemented in each automation as well as testing the automations when all components are integrated together.

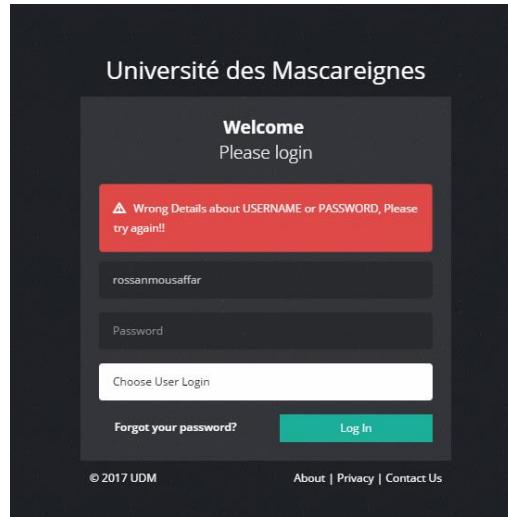
#### 6.1.1 Unit testing

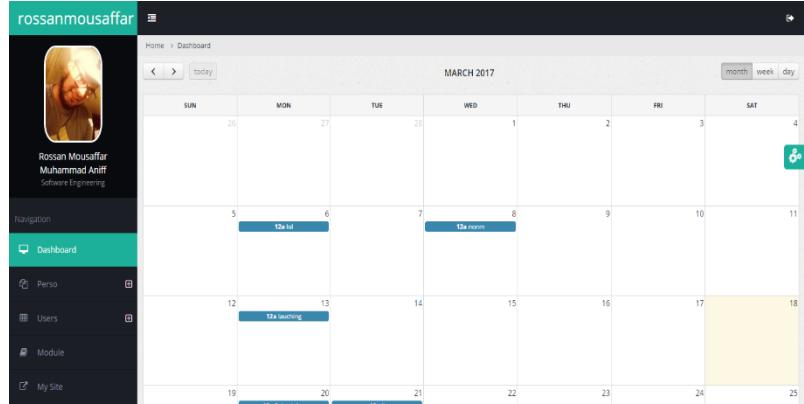
This type of testing will test the main parts of the system like the display.

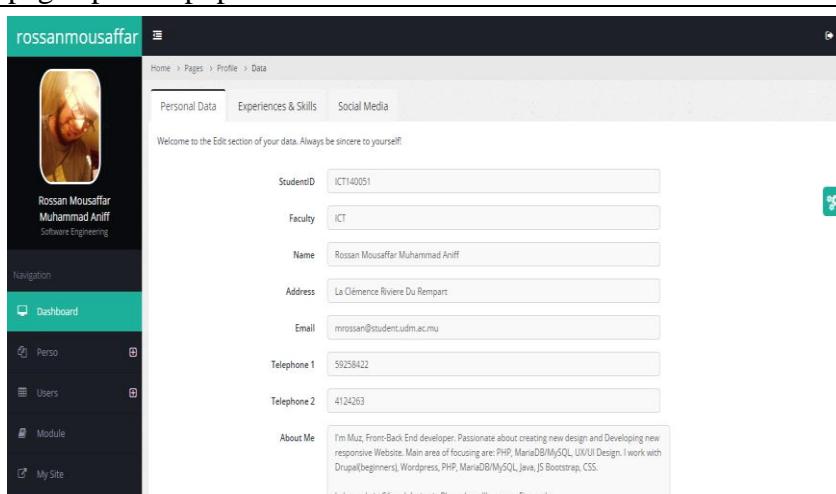
The eight following test case apply for the website

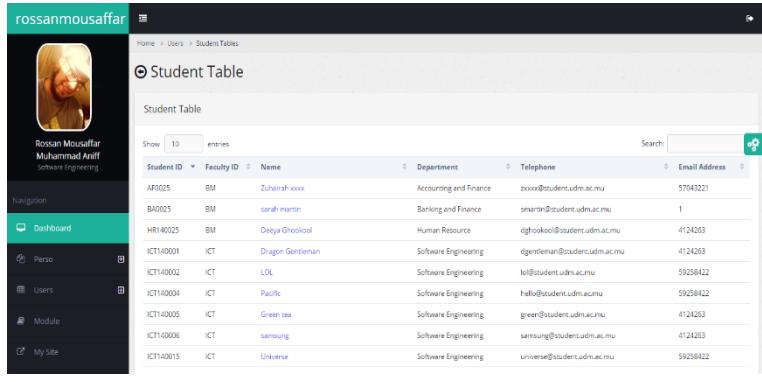
Test 1	Entering the login page
Description	Click on the log in in the index page
Expected result	login page
Actual result	
Remark test	pass

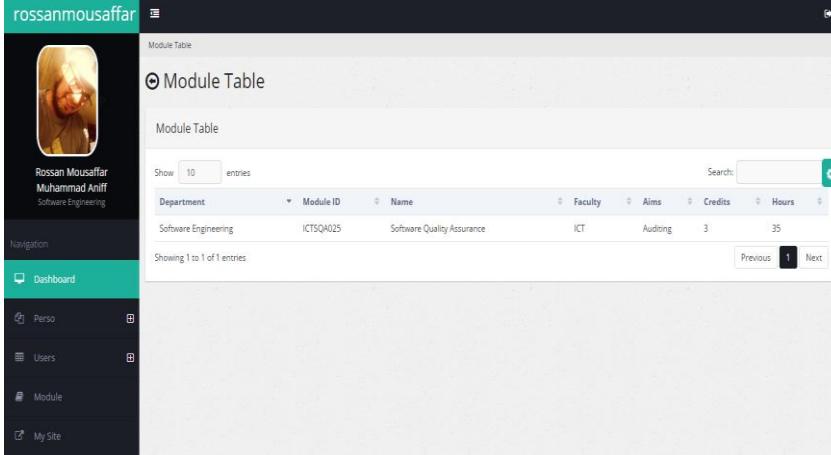
Test 2	Entering Username and password
Description	Enter correct username, password and select correct user role
Expected result	index.php
Actual result	
Remark test	pass

Test 3	Enter a wrong user name or password
Description	To check if it alert the user
Expected result	Error message display
Actual result	
Remark test	pass

Test 4	user homepage
Description	After correct username and password the user will redirect to homepage
Expected result	index.php
Actual result	
Remark test	pass

Test 5	Profile display
Description	check if the profile is displaying according to user login
Expected result	pages-profile.php
Actual result	
Remark test	pass

Test 6	check list display																																																															
Description	check if list of student display and other list will be the same but content will be different																																																															
Expected result	table-datables.php																																																															
Actual result	 <p>The screenshot shows a table titled "Student Table" with the following data:</p> <table border="1"> <thead> <tr> <th>Show</th> <th>10 entries</th> <th>Search:</th> </tr> <tr> <th>Student ID</th> <th>Faculty ID</th> <th>Name</th> <th>Department</th> <th>Telephone</th> <th>Email Address</th> </tr> </thead> <tbody> <tr> <td>AF0025</td> <td>BM</td> <td>Zuharah xxxx</td> <td>Accounting and Finance</td> <td>zxxxx@student.udm.ac.mu</td> <td>57043321</td> </tr> <tr> <td>BA0025</td> <td>BM</td> <td>sarah martin</td> <td>Banking and Finance</td> <td>smartin@student.udm.ac.mu</td> <td>1</td> </tr> <tr> <td>HR14025</td> <td>BM</td> <td>Deeya Ghoskool</td> <td>Human Resource</td> <td>dghoskool@student.udm.ac.mu</td> <td>4124063</td> </tr> <tr> <td>ICT14001</td> <td>ICT</td> <td>Dragon Gentleman</td> <td>Software Engineering</td> <td>dggentleman@student.udm.ac.mu</td> <td>4124063</td> </tr> <tr> <td>ICT14002</td> <td>ICT</td> <td>LDL</td> <td>Software Engineering</td> <td>ldl@student.udm.ac.mu</td> <td>59258422</td> </tr> <tr> <td>ICT14004</td> <td>ICT</td> <td>Pacific</td> <td>Software Engineering</td> <td>hello@student.udm.ac.mu</td> <td>59258422</td> </tr> <tr> <td>ICT14005</td> <td>ICT</td> <td>Green sea</td> <td>Software Engineering</td> <td>green@student.udm.ac.mu</td> <td>4124063</td> </tr> <tr> <td>ICT14006</td> <td>ICT</td> <td>samsung</td> <td>Software Engineering</td> <td>samsung@student.udm.ac.mu</td> <td>4124063</td> </tr> <tr> <td>ICT14015</td> <td>ICT</td> <td>Universe</td> <td>Software Engineering</td> <td>universe@student.udm.ac.mu</td> <td>59258422</td> </tr> </tbody> </table>	Show	10 entries	Search:	Student ID	Faculty ID	Name	Department	Telephone	Email Address	AF0025	BM	Zuharah xxxx	Accounting and Finance	zxxxx@student.udm.ac.mu	57043321	BA0025	BM	sarah martin	Banking and Finance	smartin@student.udm.ac.mu	1	HR14025	BM	Deeya Ghoskool	Human Resource	dghoskool@student.udm.ac.mu	4124063	ICT14001	ICT	Dragon Gentleman	Software Engineering	dggentleman@student.udm.ac.mu	4124063	ICT14002	ICT	LDL	Software Engineering	ldl@student.udm.ac.mu	59258422	ICT14004	ICT	Pacific	Software Engineering	hello@student.udm.ac.mu	59258422	ICT14005	ICT	Green sea	Software Engineering	green@student.udm.ac.mu	4124063	ICT14006	ICT	samsung	Software Engineering	samsung@student.udm.ac.mu	4124063	ICT14015	ICT	Universe	Software Engineering	universe@student.udm.ac.mu	59258422
Show	10 entries	Search:																																																														
Student ID	Faculty ID	Name	Department	Telephone	Email Address																																																											
AF0025	BM	Zuharah xxxx	Accounting and Finance	zxxxx@student.udm.ac.mu	57043321																																																											
BA0025	BM	sarah martin	Banking and Finance	smartin@student.udm.ac.mu	1																																																											
HR14025	BM	Deeya Ghoskool	Human Resource	dghoskool@student.udm.ac.mu	4124063																																																											
ICT14001	ICT	Dragon Gentleman	Software Engineering	dggentleman@student.udm.ac.mu	4124063																																																											
ICT14002	ICT	LDL	Software Engineering	ldl@student.udm.ac.mu	59258422																																																											
ICT14004	ICT	Pacific	Software Engineering	hello@student.udm.ac.mu	59258422																																																											
ICT14005	ICT	Green sea	Software Engineering	green@student.udm.ac.mu	4124063																																																											
ICT14006	ICT	samsung	Software Engineering	samsung@student.udm.ac.mu	4124063																																																											
ICT14015	ICT	Universe	Software Engineering	universe@student.udm.ac.mu	59258422																																																											
Remark test	pass																																																															

Test 7	module list display																	
Description	list display only according to student department																	
Expected result	table-module.php																	
Actual result	 <p>The screenshot shows a table titled "Module Table" with the following data:</p> <table border="1"> <thead> <tr> <th>Show</th> <th>10 entries</th> <th>Search:</th> </tr> <tr> <th>Department</th> <th>Module ID</th> <th>Name</th> <th>Faculty</th> <th>Aims</th> <th>Credits</th> <th>Hours</th> </tr> </thead> <tbody> <tr> <td>Software Engineering</td> <td>ICTSQ4025</td> <td>Software Quality Assurance</td> <td>ICT</td> <td>Auditing</td> <td>3</td> <td>35</td> </tr> </tbody> </table>	Show	10 entries	Search:	Department	Module ID	Name	Faculty	Aims	Credits	Hours	Software Engineering	ICTSQ4025	Software Quality Assurance	ICT	Auditing	3	35
Show	10 entries	Search:																
Department	Module ID	Name	Faculty	Aims	Credits	Hours												
Software Engineering	ICTSQ4025	Software Quality Assurance	ICT	Auditing	3	35												
Remark test	pass																	

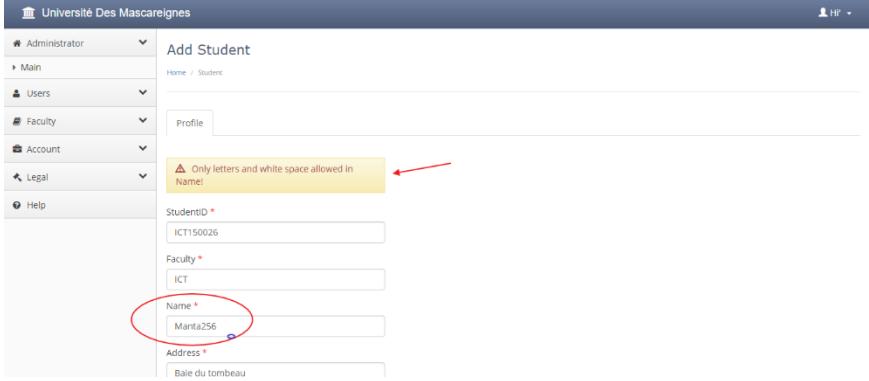
Test 8	website
Description	to check if user website is according to user profile
Expected result	home.php
Actual result	
Remark test	pass

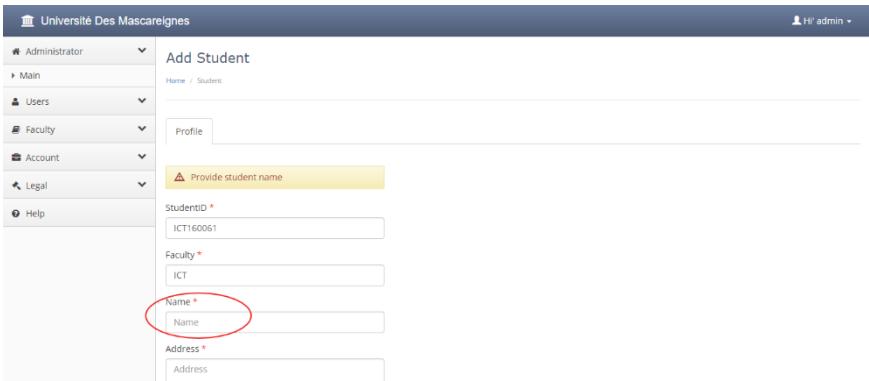
### 6.1.2 Integration Strategies

Integration strategies are done to verify the functional, performance, and reliability of the system like database, update, add, delete

The nine following test case apply for the website

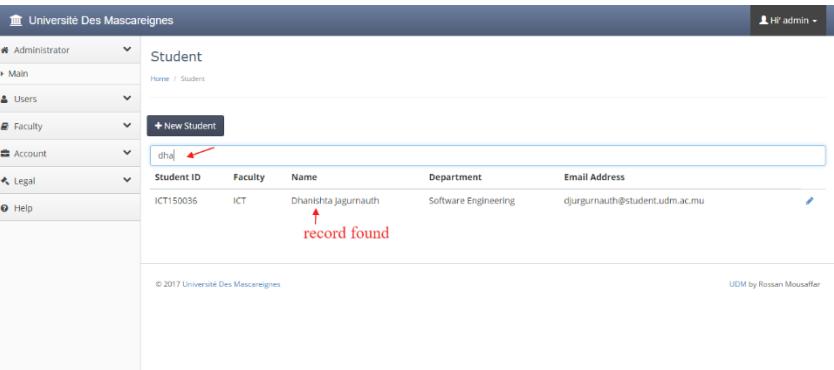
Test 9	Add a new user
Description	Able to add new user into the system
Expected result	user added
Actual result	user added
Remark test	pass

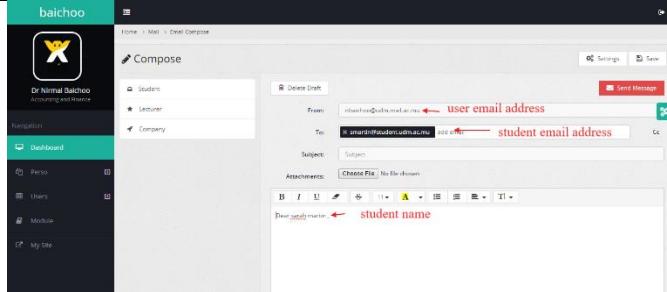
Test 10	Add a new user
Description	add new user with unwanted characters like numeric in name
Expected result	Error
Actual result	 <p>The screenshot shows the 'Add Student' form. A validation message 'Only letters and white space allowed in Name!' is displayed in a yellow box above the 'Name' field. The 'Name' field contains 'Manita256' and has a red circle around it, indicating it is the source of the error. The 'Faculty' dropdown is set to 'ICT'. The 'StudentID' field contains 'ICT150026'. The 'Address' field contains 'Baie du tombeau'.</p>
Remark test	pass

Test 11	Add a new user
Description	add new user with empty name
Expected result	Error
Actual result	 <p>The screenshot shows the 'Add Student' form. A validation message 'Provide student name' is displayed in a yellow box above the 'Name' field. The 'Name' field is empty and has a red circle around it, indicating it is the source of the error. The 'Faculty' dropdown is set to 'ICT'. The 'StudentID' field contains 'ICT160061'. The 'Address' field is empty.</p>
Remark test	pass

Test 12	Update
Description	Update user record
Expected result	updated
Actual result	updated
Remark test	pass

Test 13	Delete
Description	Delete user record
Expected result	Deleted
Actual result	Deleted
Remark test	pass

Test 14	Search										
Description	Search User record										
Expected result	Record found										
Actual result	 <p>The screenshot shows a search results page for a student named Dhanishtha Jagurnauth. A red arrow points to the search input field containing 'dhan'. Another red arrow points to the text 'record found' below the table.</p> <table border="1"> <thead> <tr> <th>Student ID</th> <th>Faculty</th> <th>Name</th> <th>Department</th> <th>Email Address</th> </tr> </thead> <tbody> <tr> <td>ICT150036</td> <td>ICT</td> <td>Dhanishtha Jagurnauth</td> <td>Software Engineering</td> <td>djagurnauth@student.udm.ac.mu</td> </tr> </tbody> </table>	Student ID	Faculty	Name	Department	Email Address	ICT150036	ICT	Dhanishtha Jagurnauth	Software Engineering	djagurnauth@student.udm.ac.mu
Student ID	Faculty	Name	Department	Email Address							
ICT150036	ICT	Dhanishtha Jagurnauth	Software Engineering	djagurnauth@student.udm.ac.mu							
Remark test	pass										

Test 15	send mail
Description	click on the email of the user it will redirect to the a mail compose to send mail
Expected result	redirect to mail compose
Actual result	
Remark test	pass

Test 16	download
Description	download user CV
Expected result	download
Actual result	downloaded
Remark test	pass

Test 17	user site
Description	Click on user name; redirect to user site
Expected result	user site
Actual result	
Remark test	pass

### 6.1.3 System Testing

System Testing is usually carried out to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

System Testing is being done while doing integration testing and unit testing on parallel.

### 6.1.4 Acceptance Testing

Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications.

Acceptance testing could not carry out as lecturer was not available

Acceptance testing could not carry out as not full-fledged due to less resources

## Table of Contents

Contents of Table .....	114
7.0 Conclusion .....	115
7.1 Achievements .....	115
7.2 Personal views .....	117
7.3 Further Upgrades.....	117

## Contents of Table

Table 1- Functionalities achievement .....	116
Table 2- Non functionalities achievement .....	116

## 7.0 Conclusion

This section of the documents will describe the achievements, strengths and weakness of the project, the difficulties encountered and the future works that can be done.

Database are closed after a particular action has been performed so that to free up the memory.

### 7.1 Achievements

The table below will show the functional requirement achievements of the project.

Functional Requirements	Status
1. The proposed system will allows users to describe their personalities and skills.	Achieved
2. The proposed system will allows companies to recruit students directly.	Achieved
3. The proposed system will allows students to communicate via mail.	Achieved
4. Administrator of the system will have full access rights to add, delete, modify and records on the system.	Achieved
5. The proposed system will allowed company and lecturer to download CV from students.	Achieved

6. A personal calendar for all users.	Achieved
7. Every details should be able to customized. Minimized Hardcore code.	Achieved
8. A blog or online portfolio for each lecturer and student.	Achieved
9. The proposed system will allows users to send CV and emails to companies directly.	Pending

*Table 1- Functionalities achievement*

The table below will show the achievements of the non-functional requirements of the project.

Functional Requirements	Status
1. The proposed system should be available on a 24-hour basis and 7/7. So that students can update their records and companies can view profile of students at any time.	Achieved
2. The proposed system will have a sufficient space to store large volume of information. No problem of memory or data loss will occur due to memory overload	Achieved
3. The proposed system will be a performant one as the details will be up-to-date and few waiting time for processing of data and query about information's. Response time will be quickly.	Achieved

*Table 2- Non functionalities achievement*

Testing was done properly and that has enable us to see where the system lacks so that we have been able to correct it. Moreover Unit testing has already started when our first sprint was done.

- Try to attempt all the functionalities and non-functionalities
- Try to attempt all the objectives.

## 7.2 Personal views

- Around 95 % the project has been completed and the remaining 5% is due to the email address as don't have enough resources.
- Testing strategies were good and help to improve the system more.
- Mostly all the objectives have been met. As Acceptance testing was not possible so the system don't have a full feedback.
- Personal achievements and benefits are met.

## 7.3 Further Upgrades

- Code Optimisation and refactoring
- Re-arrange DB
- Email address and notification
- Analyse records for login

## References

- &rarr;; View. "Advantages And Disadvantages Of Inkjet Printers". *Industrial Inkjet*. N.p., 2017. Web. 17 Mar. 2017.
- [closed], WAMP. "WAMP Vs XAMPP Pros/Cons - For Running A Local Testing Server?". *Wordpress.stackexchange.com*. N.p., 2017. Web. 17 Mar. 2017.
- "(The Only Proper) PDO Tutorial". *Phpdelusions.net*. N.p., 2017. Web. 17 Mar. 2017.
- "(The Only Proper) PDO Tutorial". *Phpdelusions.net*. N.p., 2017. Web. 17 Mar. 2017.
- Agarwal, Neeraj. "PHP PDO Insert Data Into Database | Formget". *Formget.com*. N.p., 2017. Web. 17 Mar. 2017.
- "Acer Laptop Aspire V3-572G-76EM Intel Core I7 5500U (2.40Ghz) 8GB Memory 1TB HDD FHD NVIDIA Geforce GT 840M 15.6" Windows 8.1 - Newegg.Com". *Newegg.com*. N.p., 2017. Web. 17 Mar. 2017.
- Bridge, Rachel. "Open Source Software - The Advantages & Disadvantages.". N.p., 2017. Web. 17 Mar. 2017.
- database, PDO. "PDO Get Data From Database". *Stackoverflow.com*. N.p., 2017. Web. 17 Mar. 2017.
- "Delete Record Using PDO Mysql Connection". *Plus2net.com*. N.p., 2017. Web. 17 Mar. 2017.
- "Download File From Mysql Database Using Php". *The SitePoint Forums*. N.p., 2017. Web. 17 Mar. 2017.
- "Education Quotes - Brainyquote". *BrainyQuote*. N.p., 2017. Web. 17 Mar. 2017.
- "Forbes Welcome". *Forbes.com*. N.p., 2017. Web. 17 Mar. 2017.
- "Free Availability Calendar". *PHPJabbers*. N.p., 2017. Web. 17 Mar. 2017.
- "Getting Started With WAMP". *Techiwarehouse.com*. N.p., 2017. Web. 17 Mar. 2017.

"How To Choose The Right Web Server – Which Server Is Better Apache Or IIS | Nulllunarpages Web Hosting". *Lunarpages*. N.p., 2017. Web. 17 Mar. 2017.

"How To Create A Filter/Search Table". *W3schools.com*. N.p., 2017. Web. 17 Mar. 2017.

IIS?, What. "What Is The Difference Between XAMPP Or WAMP Server & IIS?". *Stackoverflow.com*. N.p., 2017. Web. 17 Mar. 2017.

Lengstorf, Jason. "Object-Oriented PHP For Beginners". *Code Envato Tuts+*. N.p., 2017. Web. 17 Mar. 2017.

lin, jim. "Windows 8.1 - Pros And Cons". *MalwareTips.com*. N.p., 2017. Web. 17 Mar. 2017.

Linux?, What. "What Are The Benefits Of Installing XAMPP Vs [LAMP From Repository] On Ubuntu Linux?". *Serverfault.com*. N.p., 2017. Web. 17 Mar. 2017.

"Online Diagram Software To Draw Flowcharts, UML & More | Creately". *Creately.com*. N.p., 2017. Web. 17 Mar. 2017.

"PHP 5 Tutorial". *W3schools.com*. N.p., 2017. Web. 17 Mar. 2017.

"PDO Tutorial For Mysql Developers - Hashphp.Org". *Wiki.hashphp.org*. N.p., 2017. Web. 17 Mar. 2017.

"PHP 5 File Upload". *W3schools.com*. N.p., 2017. Web. 17 Mar. 2017.

PHP, Fullcalendar. "Fullcalendar CRUD With Jquery And PHP - Jquery Ajax PHP". *jQuery Ajax PHP*. N.p., 2017. Web. 17 Mar. 2017.

"PHP Connect To Mysql". *W3schools.com*. N.p., 2017. Web. 17 Mar. 2017.

"PHP Insert Data Into Mysql". *W3schools.com*. N.p., 2017. Web. 17 Mar. 2017.

PHP, Fullcalendar. "Fullcalendar CRUD With Jquery And PHP - Jquery Ajax PHP". *jQuery Ajax PHP*. N.p., 2017. Web. 17 Mar. 2017.

"PHP: Classes And Objects - Manual". *Php.net*. N.p., 2017. Web. 17 Mar. 2017.

"PHP: Mysql Database". *W3schools.com*. N.p., 2017. Web. 17 Mar. 2017.

"PHP: PDO::Exec - Manual". *Php.net*. N.p., 2017. Web. 17 Mar. 2017.

"PHP: Pdostatement::Fetchall - Manual". *Php.net*. N.p., 2017. Web. 17 Mar. 2017.

Saiyad, Asif. "DIFFERENCE BETWEEN WAMP AND XAMPP". *Baharuni\_Asif*. N.p., 2017. Web. 17 Mar. 2017.

"SQL INNER JOIN Keyword". *W3schools.com*. N.p., 2017. Web. 17 Mar. 2017.

Staff, Creative. "18 Great Text Editors For Web Designers". *Creative Bloq*. N.p., 2017. Web. 17 Mar. 2017.

"Sitepoint Smackdown: Atom Vs Brackets Vs Light Table Vs Sublime Text". *SitePoint*. N.p., 2017. Web. 17 Mar. 2017.

update, rerendering. "Rerendering Events In Fullcalendar After Ajax Database Update". *Stackoverflow.com*. N.p., 2017. Web. 17 Mar. 2017.

"What Are The Advantages And Disadvantages Of A Mac OS?". *Reference*. N.p., 2017. Web. 17 Mar. 2017.

What is the difference between MySQL, MySQLi and PDO? [closed]. "What Is The Difference Between Mysql, Mysqli And PDO?". *Stackoverflow.com*. N.p., 2017. Web. 17 Mar. 2017.

"Windows 10 Vs Windows 8 Vs Windows 7 - Should You Upgrade?". *International Business Times UK*. N.p., 2017. Web. 17 Mar. 2017.

X, Advantages/disadvantages. "Advantages/Disadvantages Of Using The Built-In Apache For Web Development On Mac OS X". *Stackoverflow.com*. N.p., 2017. Web. 17 Mar. 2017.

"XAMPP Vs WAMP: Which One To Choose For Running Websites Locally". *Qoncious.com*. N.p., 2017. Web. 17 Mar. 2017.

## Contents

Introduction.....	2
1. Login as student.....	2
1.1 How to login:.....	2
1.2 How to used user dashboard.....	5
1.3 Log out .....	11
2. Login as lecturer .....	11
3. Login as company.....	11

## Figures

Figure 1- log in.....	3
Figure 2- forget password .....	4
Figure 3- user page.....	5
Figure 4- calendar .....	6
Figure 5- profile.php .....	6
Figure 6- click .....	7
<i>Figure 7- user-website .....</i>	<i>7</i>
Figure 8- lecturer list.....	8
Figure 9- email composer .....	9
Figure 10- module.....	9
Figure 11- user own site.....	10
Figure 12- download resume .....	10
Figure 13- log out.....	11
Figure 14- company dashboard.....	11
Figure 15- module table .....	12

## Introduction

Campus Recruitment System is to replace the manual operation of recruitment of the companies at Université Des Mascareignes and also will provide a platform for students to create their own identity. As recruitment happened on a yearly basics where thousands of students get employed and being recruited at the campus. Students will be able to create their own status and even be able to send their CV to the companies and companies will send be able to select the best candidate and even select students without receiving any CV from students. As soon as candidates have been selected then the interview schedule will be send by mail. The system will be able to produce queries and managed the database during recruitment process. Recruitment is refers as to make someone be a new member of the organization or institution.

As soon as the student or lecturer has been enroll into the university. The administration should register the users on the site.

### 1. Login as student

Now, when the student or lecturer get their username and password.

#### 1.1 How to login:

- Launch a browser (recommended Google Chrome)
- Go to: localhost:80[default] / udm /
- Click on “log-in” on the navigation bar on the top

The student will be redirected to the login page as have shown on the next page:

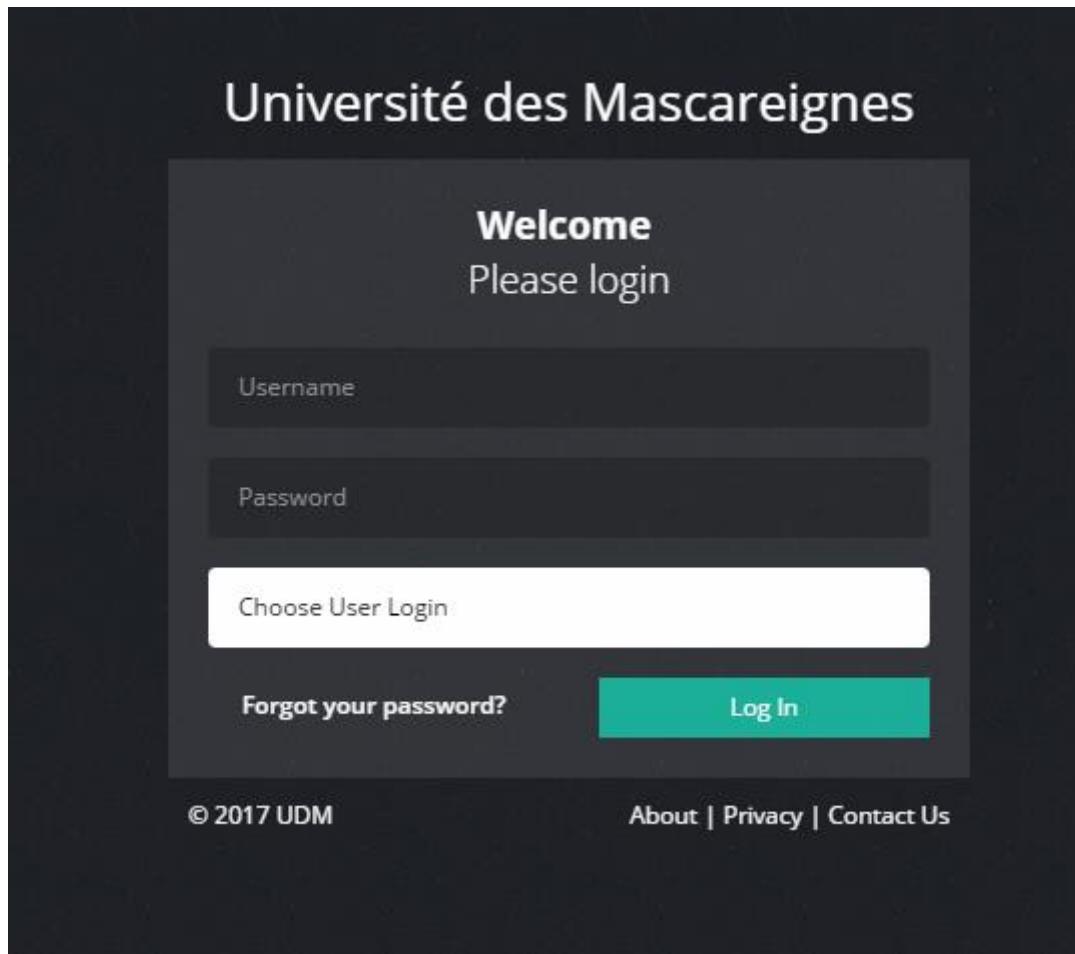


Figure 1- log in

Now, after landing of the login page the user has to input his username, password and “choose user login”[according to the its user role]

After the field has been fielded then click on “log-in” then the user will be redirected to his respective dashboard.

If happen user has forget his password then have to click on “forgot your password” then the user will be redirected to the “forgot password page” shown on the next page:



To reset your Password please fill up the form and you will be emailed your username and your **new Password** shortly in a delay of three days. Please save your **Username** and **Password** securely.

<b>Name *</b>	<input type="text"/>	<b>Address and Phone</b>
<b>Email *</b>	<input type="text"/>	Pamplemousses Campus Université des Mascareignes Beau Plan Round About, MAURITIUS Tel: +230 243 0045 Fax: +230 243 5154 Email: udmcpcam@udm.ac.mu
<b>Subject</b>	<input type="text"/>	<b>Head Office &amp; Rose Hill Campus</b>
<b>Message *</b>	<input type="text"/>	Université des Mascareignes Pamplemousses, Avenue de La Concorde Roche Brunes Rose Hill, MAURITIUS Tel: +230 466 0444 / 0118 Fax: +230 466 3774 Email: udmcph@udm.ac.mu

**SUBMIT**

f    e-mail    YouTube

© Copyright 2014 UDM • Design by Rossan Mousaffar

*Figure 2- forget password*

Once here, the user has to send a mail to the admin or respective HOD or Dean to in request for a password reset so that he can get access to his dashboard again.

Assume user has well input his username and password then he will be redirected to his dashboard shown on the next page:

## 1.2 How to used user dashboard

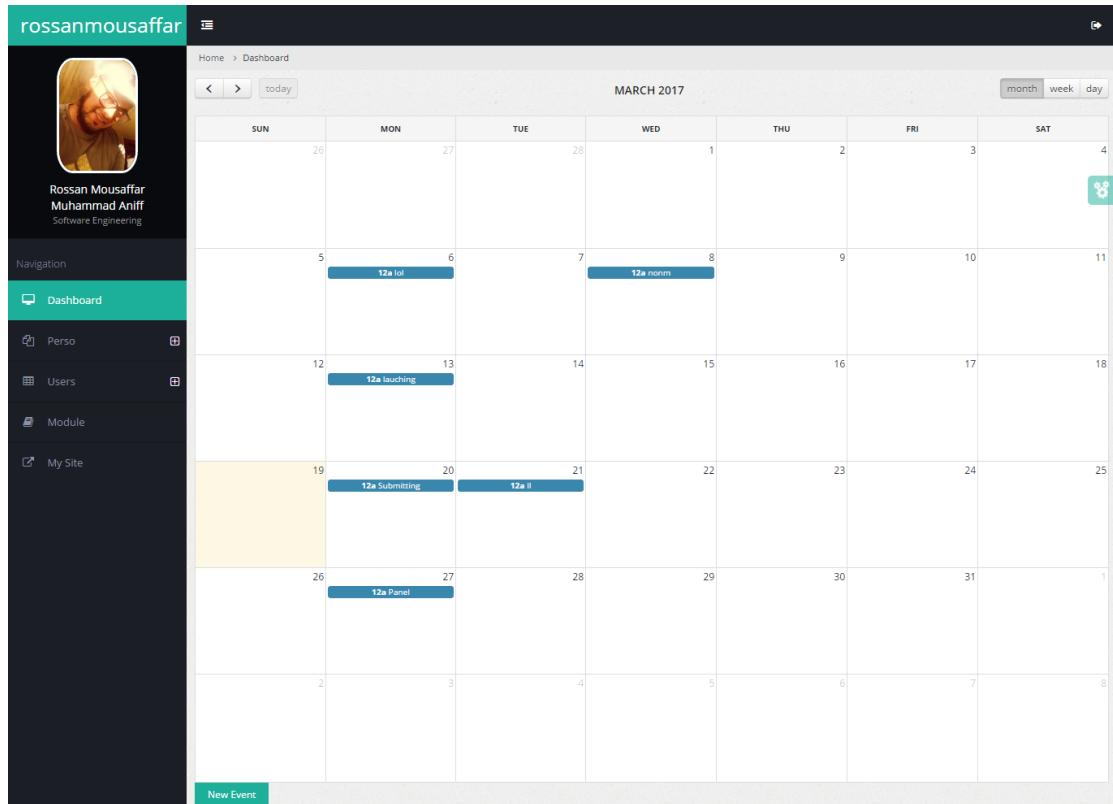


Figure 3- user page

Once here user can see the sidebar and the header bar and the content [planner]

The main page on the dashboard is a view of user planning.

If user want to add a new event he can click on the button “New Event” as shown below



Click on the “New Event” then

The calendar will appear like has been shown below:

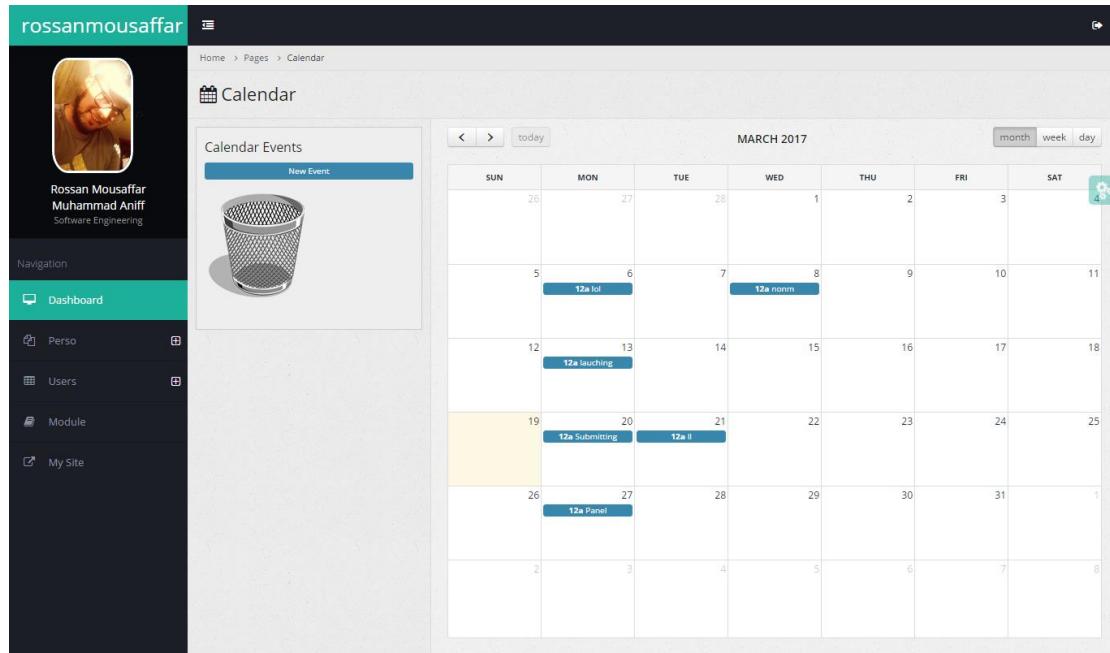


Figure 4- calendar

The new-event can be inserted by drag the button new event into the calendar. Moreover it can be deleted by drag it back to dustbin.

Click on the link “perso” like has been shown below:

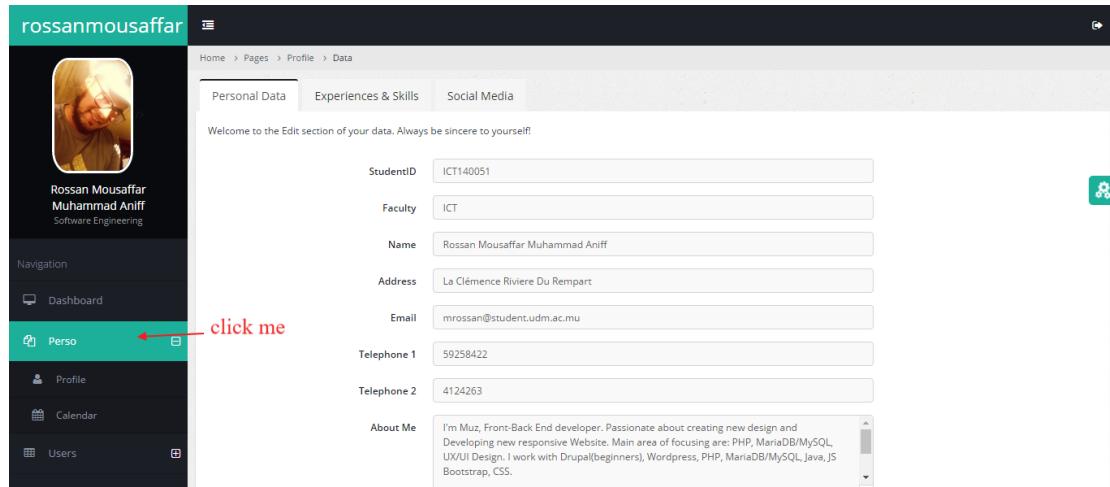


Figure 5- profile.php

In this section student can edit their informations about them that will appear on their website

Now,

- Click on “users”
- Select “students”
- The list of student will appear on the right of the screen like has been show below.
- Click on the student “Name”, user will redirect to student website like as shown below

Student Table					
Student Table					
Student ID	Faculty ID	Name	Department	Telephone	Email Address
AF0025	BM	Zuhairah xxxx	Accounting and Finance	xxxxx@student.udm.ac.mu	57043221
BA0025	BM	sarah martin	Banking and Finance	smartin@student.udm.ac.mu	57043221
HR140025	BM	Deeya Ghookool	Human Resource	dghookool@student.udm.ac.mu	4124263
ICT12001	BM	Ameesha Ahotar	Accounting and Finance	oumar@gmail.com	59258422
ICT140001	ICT	Dragon Gentleman	Software Engineering	dgentleman@student.udm.ac.mu	4124263
ICT140002	ICT	LOL	Software Engineering	lol@student.udm.ac.mu	59258422
ICT140004	ICT	Pacific	Software Engineering	hello@student.udm.ac.mu	59258422
ICT140005	ICT	Green tea	Software Engineering	green@student.udm.ac.mu	4124263
ICT140006	ICT	samsung	Software Engineering	samsung@student.udm.ac.mu	4124263
ICT140015	ICT	Universe	Software Engineering	universe@student.udm.ac.mu	59258422

Show 10 entries Search:

Showing 1 to 10 of 22 entries

Previous 1 2 3 Next

navigate to another pages →

Figure 6- click



Figure 7- user-website

How to return back from the student website?



Then it will bring the user back to his respective dashboard

When moved to lecturer will have the list of lecturer like has been shown below

Lecturer ID	Faculty ID	Name	Department	Email Address
Ami025	ICT	Amic Seven	Software Engineering	samic@udm.ac.mu
Bai001	BM	Dr Nirmal Baichoo	Accounting and Finance	nbaichoo@udm.mail.ac.mu
Lau027	ICT	Lauthan Shameera	Software Engineering	slauthan@udm.ac.mu
ramlo25	ICT	Ramlowat Dosheela	Software Engineering	ramlo@gmail.com

Figure 8- lecturer list

Click on the email address where the arrow is pointed it will redirected to the email composer like has been shown on the next page:

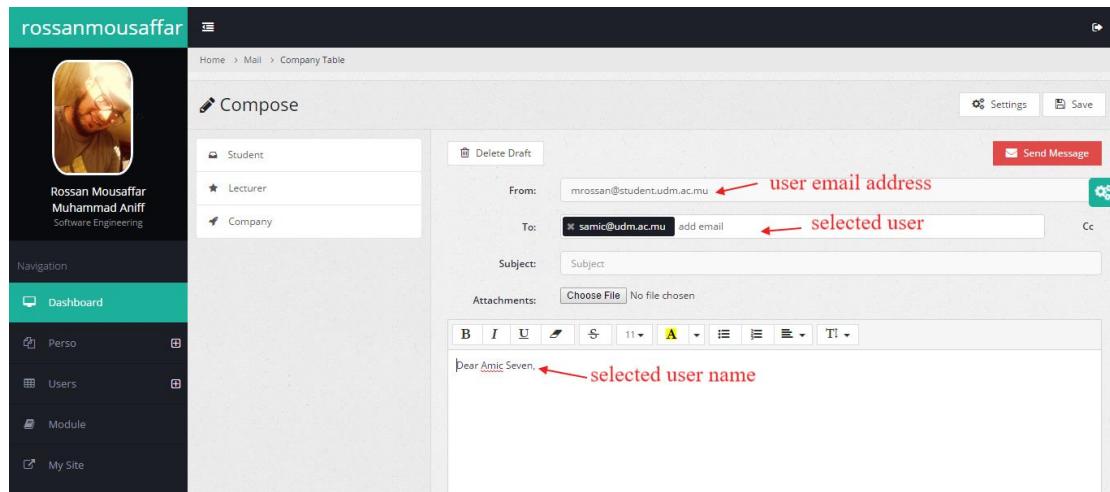


Figure 9- email composer

Note: company also has the same procedure as the lecturer List

Click the module link and it will display all the module that student is learning like has been shown below:

Module Table						
Department	Module ID	Name	Faculty	Aims	Credits	Hours
Software Engineering	ICTSQA025	Software Quality Assurance	ICT	Auditing	3	35
Showing 1 to 1 of 1 entries						

Figure 10- module

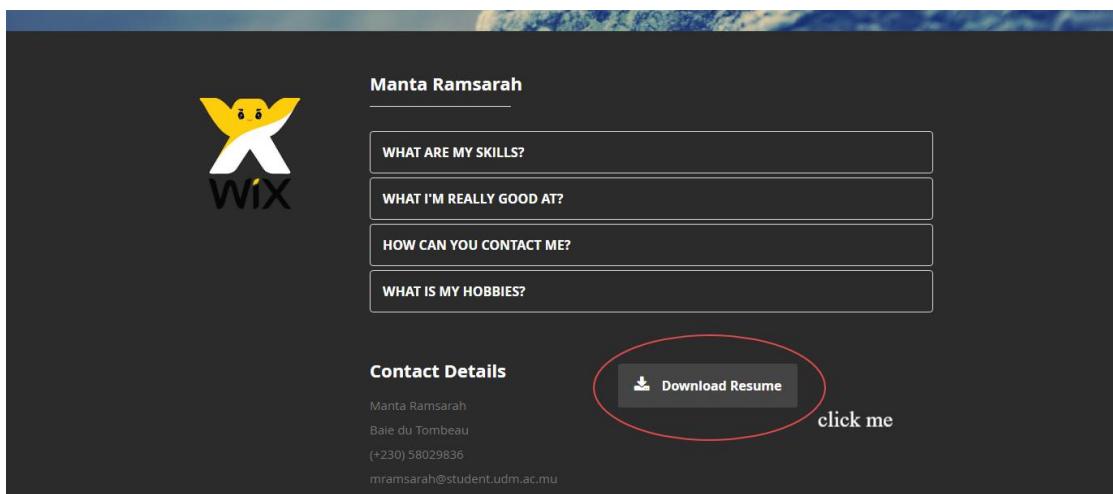
Click on “My Site” [last side bar list] that will bring the user to his own site like has been shown below:



*Figure 11- user own site*

User can upload their CV and photo on the website view tab “experience and skills, and social media for other user information.

Moreover, only student can view and download student details and lecturer can view and download student details and company can view both and download both details.



*Figure 12- download resume*

### 1.3 Log out

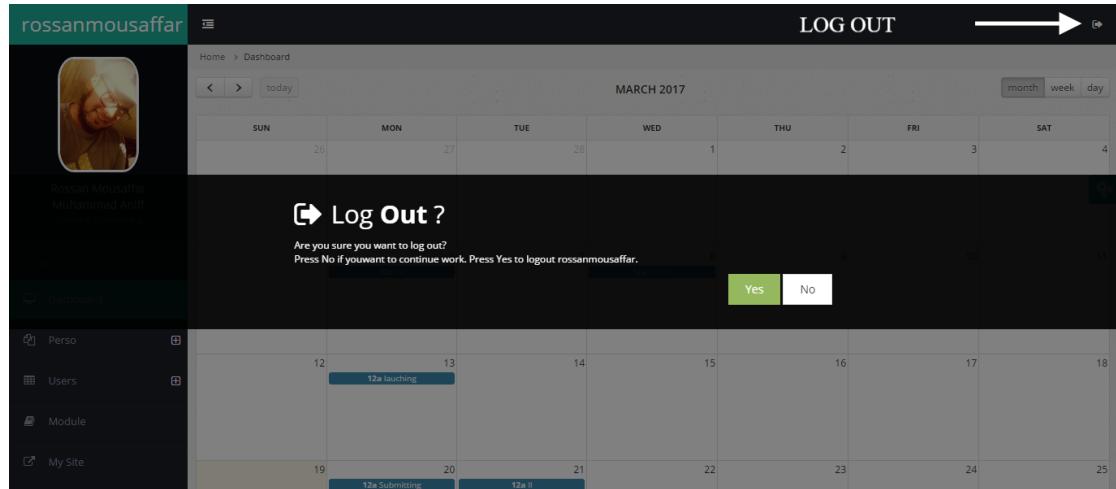


Figure 13- log out

### 2. Login as lecturer

Lecturer will be almost the same as “log in as student” till “log out”

### 3. Login as company

- Follows the steps “log in as student” will be the same as log in for company.
- No profile for company.

After login user will be redirected to his dashboard:

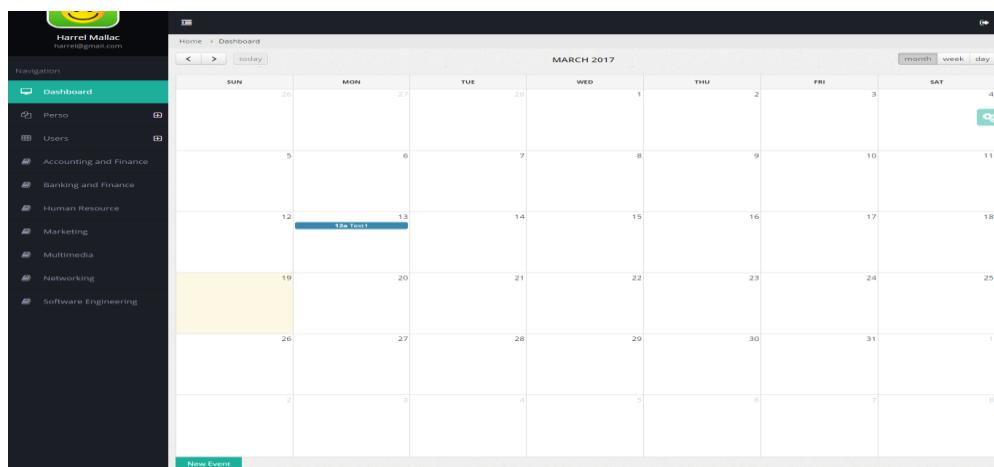
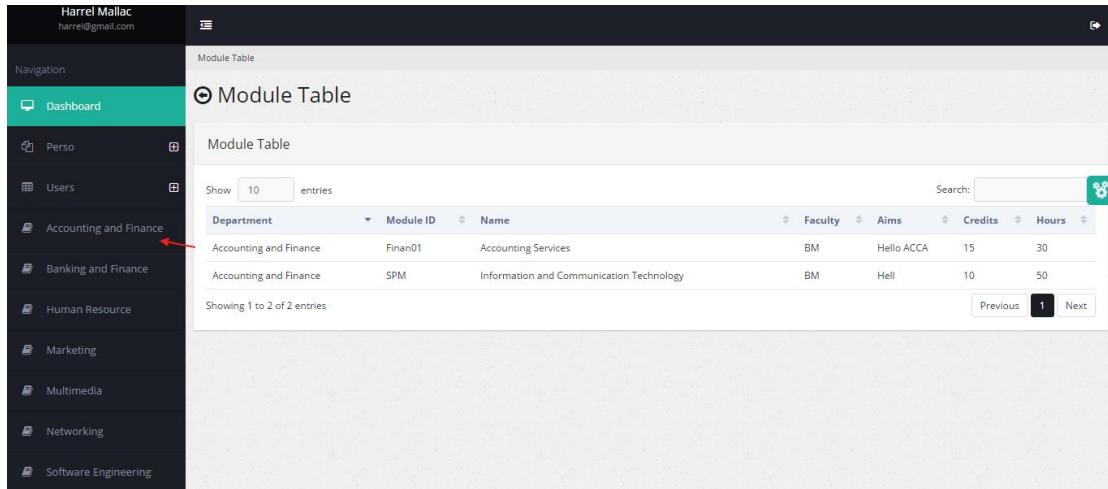


Figure 14- company dashboard

- Company has the right to view and download and send email to both ‘lecturer’ and ‘student’

On the left side bar all the department has been mentioned and each one contains their own module like has been shown below:



The screenshot shows a user interface for a campus recruitment system. On the left, there is a vertical navigation bar with the following items: Dashboard (selected), Perso, Users, Accounting and Finance (highlighted with a red arrow), Banking and Finance, Human Resource, Marketing, Multimedia, Networking, and Software Engineering. The main content area is titled 'Module Table' and displays a table of modules. The table has columns for Department, Module ID, Name, Faculty, Aims, Credits, and Hours. Two entries are listed: 'Accounting and Finance' with Module ID 'Finan01' and 'Name' 'Accounting Services', and another entry with Module ID 'SPM' and 'Name' 'Information and Communication Technology'. There are also buttons for 'Search', 'Previous', and 'Next'.

Department	Module ID	Name	Faculty	Aims	Credits	Hours
Accounting and Finance	Finan01	Accounting Services	BM	Hello ACCA	15	30
Accounting and Finance	SPM	Information and Communication Technology	BM	Hell	10	50

Figure 15- module table

- Users can logout from the website like has been shown in section 1.3

**NOTE:** For any query please contact the administrator on [admin@udm.ac.mu](mailto:admin@udm.ac.mu)

## Contents

1. How to log in as an admin .....	2
------------------------------------	---

## Figures

Figure 1- login.....	2
Figure 2- dashboard .....	3
Figure 3- add new event.....	4
Figure 4- edit calendar .....	4
Figure 5- student list .....	5
Figure 6- add student .....	5
Figure 7- search user.....	5
Figure 8- edit student .....	5
Figure 9- update admin account.....	5

## 1. How to log in as an admin

Open a web browser

Go to: localhost:80[default port]/ udm / adminudm

Then click enter

Admin will be redirected to admin log in shown below:



Figure 1- login

- Username: admin
- Password: 123456
- Then sign-In

After sign-In the admin will be redirected to admin dashboard shown below:

The screenshot shows the admin dashboard for the 'Université Des Mascareignes'. The top navigation bar includes the university logo, a user icon labeled 'Hi! admin', and a dropdown menu for 'Administrator' which is expanded to show 'Main', 'Users', 'Faculty', 'Account', 'Legal', and 'Help'. The main content area is titled 'Administrator' and shows the URL 'Home / Dashboard'. Below this is a calendar for 'March 2017' with days from Sunday to Saturday. The dates are numbered 1 through 31. A yellow box highlights the date '15' (Wednesday). At the bottom left of the calendar is a blue button labeled 'New Event'. At the bottom of the page, there is a copyright notice '© 2017 Université Des Mascareignes' and a credit 'UDM by Rossan Mousaffar'.

Figure 2- dashboard

The admin dashboard has a general planner for him. Where he has to click on the “new event” then will be redirected to the edit calendar for adding new event.

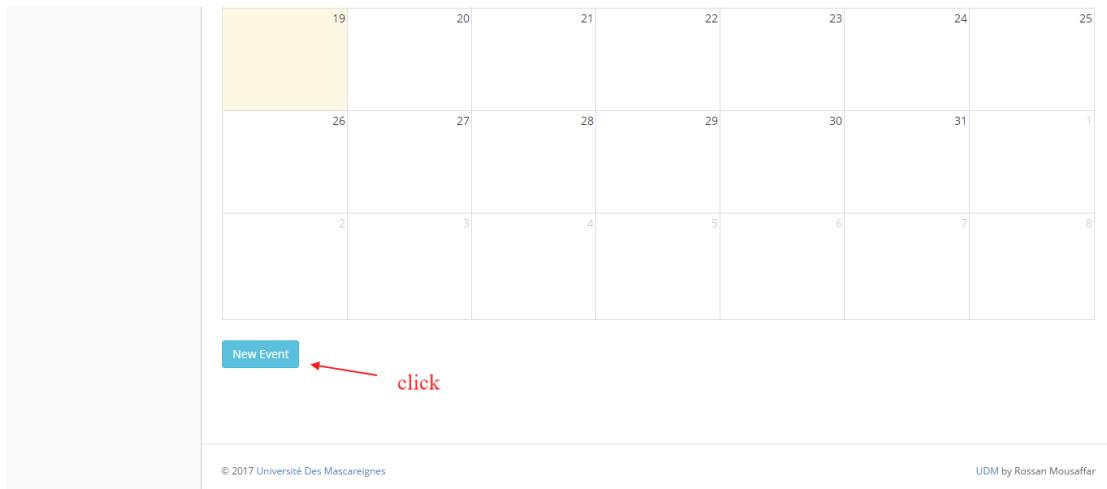


Figure 3- add new event

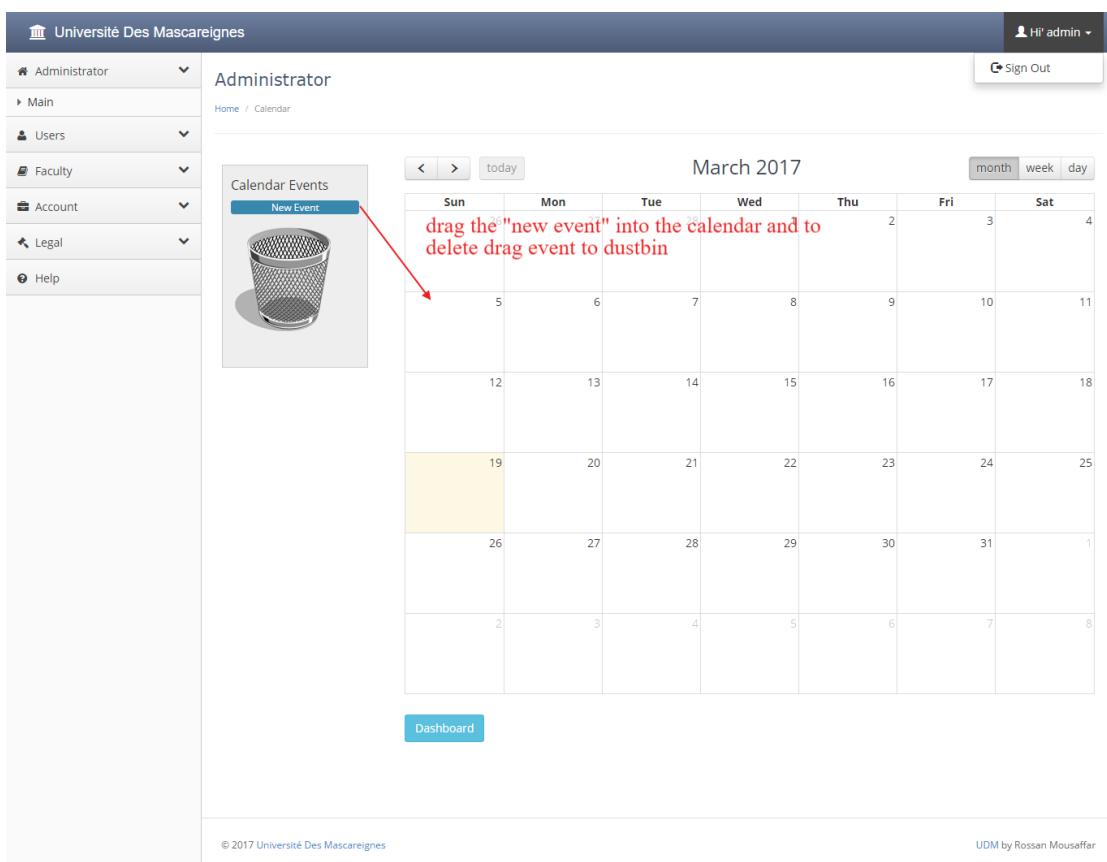
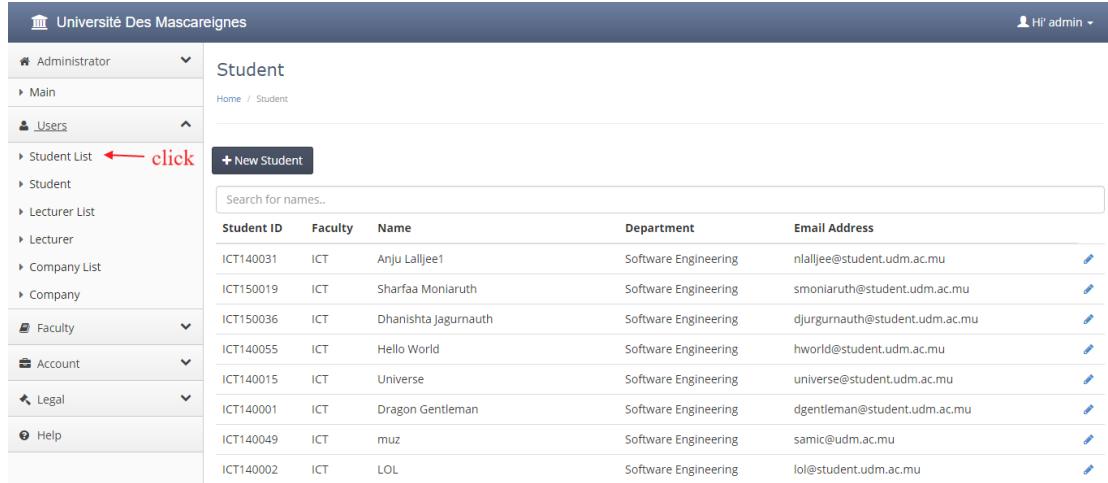


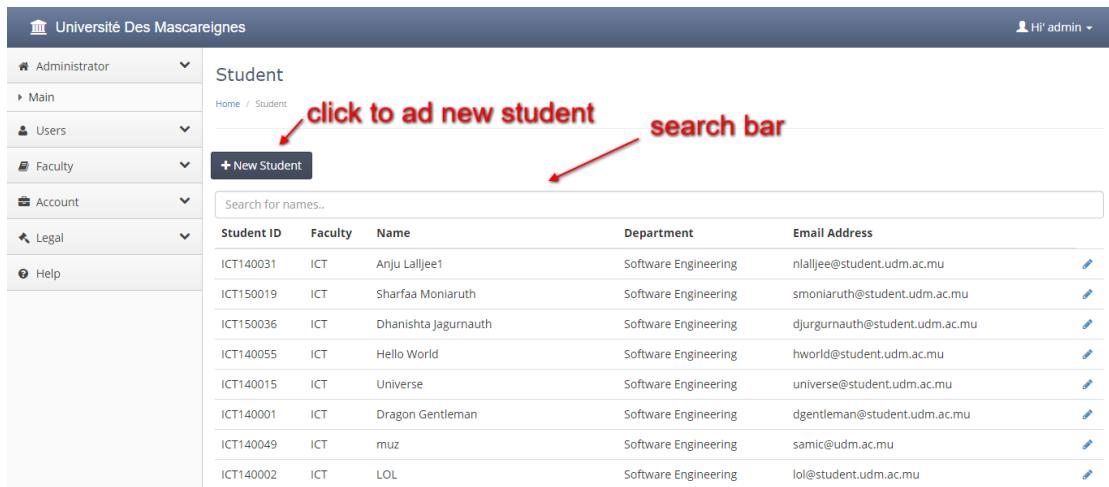
Figure 4- edit calendar

- Expand the users section
- Select student list like as shown below



Student																																																	
Home / Student																																																	
<b>+ New Student</b>																																																	
Search for names..																																																	
<table border="1"> <thead> <tr> <th>Student ID</th><th>Faculty</th><th>Name</th><th>Department</th><th>Email Address</th></tr> </thead> <tbody> <tr><td>ICT140031</td><td>ICT</td><td>Anju Lalljee1</td><td>Software Engineering</td><td>nlalljee@student.udm.ac.mu</td></tr> <tr><td>ICT150019</td><td>ICT</td><td>Sharfaa Moniaruth</td><td>Software Engineering</td><td>smoniaruth@student.udm.ac.mu</td></tr> <tr><td>ICT150036</td><td>ICT</td><td>Dhanishta Jagurnauth</td><td>Software Engineering</td><td>djurgurnauth@student.udm.ac.mu</td></tr> <tr><td>ICT140055</td><td>ICT</td><td>Hello World</td><td>Software Engineering</td><td>hworld@student.udm.ac.mu</td></tr> <tr><td>ICT140015</td><td>ICT</td><td>Universe</td><td>Software Engineering</td><td>universe@student.udm.ac.mu</td></tr> <tr><td>ICT140001</td><td>ICT</td><td>Dragon Gentleman</td><td>Software Engineering</td><td>dgentleman@student.udm.ac.mu</td></tr> <tr><td>ICT140049</td><td>ICT</td><td>muz</td><td>Software Engineering</td><td>samic@udm.ac.mu</td></tr> <tr><td>ICT140002</td><td>ICT</td><td>LOL</td><td>Software Engineering</td><td>lol@student.udm.ac.mu</td></tr> </tbody> </table>					Student ID	Faculty	Name	Department	Email Address	ICT140031	ICT	Anju Lalljee1	Software Engineering	nlalljee@student.udm.ac.mu	ICT150019	ICT	Sharfaa Moniaruth	Software Engineering	smoniaruth@student.udm.ac.mu	ICT150036	ICT	Dhanishta Jagurnauth	Software Engineering	djurgurnauth@student.udm.ac.mu	ICT140055	ICT	Hello World	Software Engineering	hworld@student.udm.ac.mu	ICT140015	ICT	Universe	Software Engineering	universe@student.udm.ac.mu	ICT140001	ICT	Dragon Gentleman	Software Engineering	dgentleman@student.udm.ac.mu	ICT140049	ICT	muz	Software Engineering	samic@udm.ac.mu	ICT140002	ICT	LOL	Software Engineering	lol@student.udm.ac.mu
Student ID	Faculty	Name	Department	Email Address																																													
ICT140031	ICT	Anju Lalljee1	Software Engineering	nlalljee@student.udm.ac.mu																																													
ICT150019	ICT	Sharfaa Moniaruth	Software Engineering	smoniaruth@student.udm.ac.mu																																													
ICT150036	ICT	Dhanishta Jagurnauth	Software Engineering	djurgurnauth@student.udm.ac.mu																																													
ICT140055	ICT	Hello World	Software Engineering	hworld@student.udm.ac.mu																																													
ICT140015	ICT	Universe	Software Engineering	universe@student.udm.ac.mu																																													
ICT140001	ICT	Dragon Gentleman	Software Engineering	dgentleman@student.udm.ac.mu																																													
ICT140049	ICT	muz	Software Engineering	samic@udm.ac.mu																																													
ICT140002	ICT	LOL	Software Engineering	lol@student.udm.ac.mu																																													

Figure 5- student list



Student																																																	
Home / Student																																																	
<b>+ New Student</b>																																																	
Search for names..																																																	
<table border="1"> <thead> <tr> <th>Student ID</th><th>Faculty</th><th>Name</th><th>Department</th><th>Email Address</th></tr> </thead> <tbody> <tr><td>ICT140031</td><td>ICT</td><td>Anju Lalljee1</td><td>Software Engineering</td><td>nlalljee@student.udm.ac.mu</td></tr> <tr><td>ICT150019</td><td>ICT</td><td>Sharfaa Moniaruth</td><td>Software Engineering</td><td>smoniaruth@student.udm.ac.mu</td></tr> <tr><td>ICT150036</td><td>ICT</td><td>Dhanishta Jagurnauth</td><td>Software Engineering</td><td>djurgurnauth@student.udm.ac.mu</td></tr> <tr><td>ICT140055</td><td>ICT</td><td>Hello World</td><td>Software Engineering</td><td>hworld@student.udm.ac.mu</td></tr> <tr><td>ICT140015</td><td>ICT</td><td>Universe</td><td>Software Engineering</td><td>universe@student.udm.ac.mu</td></tr> <tr><td>ICT140001</td><td>ICT</td><td>Dragon Gentleman</td><td>Software Engineering</td><td>dgentleman@student.udm.ac.mu</td></tr> <tr><td>ICT140049</td><td>ICT</td><td>muz</td><td>Software Engineering</td><td>samic@udm.ac.mu</td></tr> <tr><td>ICT140002</td><td>ICT</td><td>LOL</td><td>Software Engineering</td><td>lol@student.udm.ac.mu</td></tr> </tbody> </table>					Student ID	Faculty	Name	Department	Email Address	ICT140031	ICT	Anju Lalljee1	Software Engineering	nlalljee@student.udm.ac.mu	ICT150019	ICT	Sharfaa Moniaruth	Software Engineering	smoniaruth@student.udm.ac.mu	ICT150036	ICT	Dhanishta Jagurnauth	Software Engineering	djurgurnauth@student.udm.ac.mu	ICT140055	ICT	Hello World	Software Engineering	hworld@student.udm.ac.mu	ICT140015	ICT	Universe	Software Engineering	universe@student.udm.ac.mu	ICT140001	ICT	Dragon Gentleman	Software Engineering	dgentleman@student.udm.ac.mu	ICT140049	ICT	muz	Software Engineering	samic@udm.ac.mu	ICT140002	ICT	LOL	Software Engineering	lol@student.udm.ac.mu
Student ID	Faculty	Name	Department	Email Address																																													
ICT140031	ICT	Anju Lalljee1	Software Engineering	nlalljee@student.udm.ac.mu																																													
ICT150019	ICT	Sharfaa Moniaruth	Software Engineering	smoniaruth@student.udm.ac.mu																																													
ICT150036	ICT	Dhanishta Jagurnauth	Software Engineering	djurgurnauth@student.udm.ac.mu																																													
ICT140055	ICT	Hello World	Software Engineering	hworld@student.udm.ac.mu																																													
ICT140015	ICT	Universe	Software Engineering	universe@student.udm.ac.mu																																													
ICT140001	ICT	Dragon Gentleman	Software Engineering	dgentleman@student.udm.ac.mu																																													
ICT140049	ICT	muz	Software Engineering	samic@udm.ac.mu																																													
ICT140002	ICT	LOL	Software Engineering	lol@student.udm.ac.mu																																													

Admin can add student by clicking the “Add Student” and the following page will appear

The screenshot shows the 'Add Student' page of the 'Université Des Mascareignes' system. The left sidebar has a 'Users' section expanded, showing 'Student List', 'Student', 'Lecturer List', 'Lecturer', 'Company List', and 'Company'. The main content area is titled 'Add Student' and shows a 'Profile' tab selected. It contains four input fields: 'StudentID \*' (with placeholder 'Student ID'), 'Faculty \*' (with placeholder 'Faculty'), 'Name \*' (with placeholder 'Name'), and 'Address \*' (with placeholder 'Address').

Figure 6- add student

Admin can type the user name in the search bar for quick search like has been shown below:

The screenshot shows the 'Student' search results page. The left sidebar has a 'Users' section expanded. The main content area shows a table with columns: Student ID, Faculty, Name, Department, and Email Address. One row is visible: ICT150036, ICT, Dhanishta Jagurnauth, Software Engineering, djurgurnauth@student.udm.ac.mu. A red arrow points to the search bar containing 'dha'. Another red arrow points to the text 'record found' below the table.

Figure 7- search user

After viewing the list the admin can click on “pencil” on the right of the table to edit a particular student details then will be redirected to the edit of student like has been shown below:

The screenshot shows the 'Edit Student' page of the Campus Recruitment System. The top navigation bar includes the university logo and the text 'Université Des Mascareignes'. On the right, there is a user profile icon with the text 'Hi! admin ▾'. The left sidebar contains a navigation menu with items: 'Administrator' (selected), 'Main', 'Users', 'Faculty' (selected), 'Account', 'Legal', and 'Help'. The main content area is titled 'Edit Student' and shows the following form fields:  
 - StudentID \*: ICT140051  
 - Faculty \*: ICT  
 - Name \*: Rossan Mousaffar Muhammad Aniff  
 - Address \*: La Clémence Riviere Du Rempart  
 - Email \*: (field is empty)  
 Below the form, there is a 'Profile' link.

*Figure 8- edit student*

- These steps apply for the lecturer and company for editing their record as well.
- And it also apply for adding new faculty, new module, or edit faculty, edit module or deletion of records.

Admin can even change his password by to

- Accounts > Admin Account [shown below]

The screenshot shows the 'Edit Account' page for the Admin account. The title bar says 'Admin Account'. The main content area is titled 'Edit Account.' and contains the following form fields:  
 - Username: admin  
 - Email: adminmuz@gmail.com  
 - Password: (redacted)  
 At the bottom is a large 'UPDATE' button.

*Figure 9- update admin account*