Faculty Directory for Multi Universities

**Software Requirements Specification**

Version 1.0



**Group Id: S21025A8D8 (BC180404438)**

**Supervisor Name: Saima Jamil**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| **25/05/2021** | **1** | The purpose for this report is to give a point-by-point portrayal of the prerequisites and to examine the necessities for the undertaking of "Designing and Development of Faculty Directory for Multi Universities)".  This phase drills down into details on the requirements and documents them thoroughly. It uses several different methods and sources to find and collect data. | **BC180404438** |

**Table of Contents**

1. [Scope (of the project) page # 4](#Scope" \t "_top)
2. [Functional Requirements Non-Functional requirements page # 5](#Requirements)
3. [Use Case Diagram page # 7](#UseCase)
4. [Usage Scenarios page # 8](#Usage)
5. [Adopted Methodology page # 16](#Methodology)
6. [Work Plan (Use MS Project to create Schedule/Work Plan) page # 19](#WorkPlan)

**Scope of Project**

In the time of Covid-19, many students face difficulties in learning due to the lack of the best faculty. Therefore, a large number of students tend to more towards the remote study and the first thing they do, is to find the best online faculty all over the world. The problem arises where they had to search through every university's website side by side in order to find the best faculty member.

"Faculty Directory for Multi Universities" is the solution to the above problem. The goal of this web application is to facilitate students through a user-friendly graphical user interface with the best faculty members so that they don't have to search every site instead they will find the world's best faculty member on this single platform so that they can search every faculty on this website accordingly.

On the main page, the visitor (user) will see the top faculty members list among all. Users will be able to search faculty details such as name, university, their department, their specialties, professional interest, etc. using different filters.

Faculty Members will have to register themselves in order to add their profile on this directory that will be allowed by the administrator so that visitors will see their profile.

The admin will be authorized to add and manage all the faculty details. Visitors will not need any registration in order to search faculty

**Functional and non-Functional Requirements**

* **Functional Requirements:**

1. **Faculty:**
   1. **Registration and sign in:** Faculty member will register him/herself in the directory and will be able to login after the approval of registration request by the Admin.
   2. **Add personal details:** Faculty member will have the rights of adding, updating their personal detail in the profile.
2. **Admin**
   1. **Approve Faculty Registration:** Admin can approve the registration requests from faculty members.
   2. **Manage Faculty information:** All the added faculty details can be Viewed / Updated / Deleted by the admin.
3. **Visitor/Student**
   1. **Faculty List:** All the added faculty details will be displayed to the user.
   2. **Refine / Search:** User can search a specific faculty details by different filters such as Name, University, Department, Courses, Area of Expertise, and Professional Interests etc.

* **Non-Functional Requirements**

1. **Responsive / Performance**
   1. **Responsive:** It will be fully Responsive Web App that will provide best experience on every browser on every device.
   2. **Performance:** App will work faster even on slow internet by using fastest algorithm.
2. **User Friendly**
   1. **Interactive:** App will be fully interactive and user friendly. User will interact through fully flexible and responsive GUI.
   2. **Easy to understand:** It will provide a user interface that can be used by anyone, even if they have no deep knowledge of it. This website will have elements that are easy to access, understand, and use to facilitate those actions.
3. **Efficiency**
   1. The online website should ensure that it is reliable. System should be able to complete its on-going operations in spite of any error or problem within defined response time.
   2. The ability of our system to behave consistently in a user-acceptable manner when operating within the environment for which the system was intended. For example, our system will be able to update process must roll back all related updates in the database when any update fails.
4. **Maintainability**
   1. This Web application will be very simple and easily maintainable by adding any further features in the future allow making change easier and more affordable, avoiding change that is not necessary and should allow performing a successful repair action within a given time. Also, configuration and content management will be easy.

**Use Case Diagram**



**Usage Scenarios**

|  |  |
| --- | --- |
| **Use Case Title** | **Login** |
| Use Case Id | 1 |
| Actions | Login to website |
| Descriptions | This Use Case will allow member to login into the site / system. They will have to enter his / her Email and Password to get login into the system |
| Alternative Paths | None |
| Pre-conditions | * System should be connected with the database * Application should be running properly. * User should have a Valid Username and Password. * Faculty Member must have registered account in the application before login * Faculty Member can sign in after the approval of admin |
| Post-conditions | * Member logged into the system successfully. * Member re-directed to his / her profile, based on his / her role in the system. |
| Author | **S21025A8D8 (BC180404438)** |
| Exceptions | None |

|  |  |
| --- | --- |
| **Use Case Title** | **Registration** |
| Use Case Id | 2 |
| Actions | Register to this website database |
| Descriptions | This Use Case will allow the only Users to make a registration to get facilities provided by this website. User will have to fill a form containing all the necessary information which is needed to complete the registration process. |
| Alternative Paths | None |
| Pre-conditions | * System should be connected with the database. * Application should be running properly. * User has a Valid Mobile / Email Address. |
| Post-conditions | * The Registration was successful. |
| Author | **S21025A8D8 (BC180404438)** |
| Exceptions | None |

|  |  |
| --- | --- |
| **Use Case Title** | **View Profile** |
| Use Case Id | 3 |
| Actions | Members can view their Profile |
| Descriptions | This Use Case will allow members to see their profile after they logged in to the system. User will have to click the Profile button on the top of the website in their dashboard |
| Alternative Paths | None |
| Pre-conditions | * System should be connected with the database. * Application should be running properly. * Member must have logged in with their credentials |
| Post-conditions | * Members have to click the profile element |
| Author | **S21025A8D8 (BC180404438)** |
| Exceptions | None |

|  |  |
| --- | --- |
| **Use Case Title** | **Update Profile** |
| Use Case Id | 4 |
| Actions | Members will edit the profile form showing in profile section |
| Descriptions | This Use Case will allow Member to edit their profile including their password. The Member will have to save the profile after editing. They must have to enter their previous password in order to change it. |
| Alternative Paths | None |
| Pre-conditions | * System should be connected with the database. * Application should be running properly. * Member must have logged in with their credentials |
| Post-conditions | * Profile Updated Successfully Popup will show up. |
| Author | **S21025A8D8 (BC180404438)** |
| Exceptions | None |

|  |  |
| --- | --- |
| **Use Case Title** | **Approve/Decline Registration Request** |
| Use Case Id | 5 |
| Actions | Admin will accept/reject Faculty Member’s Registration request |
| Descriptions | This Use Case will allow admin to see the registrations request made by faculty member. The admin will decide whether to accept or reject the request by click button respectively. |
| Alternative Paths | None |
| Pre-conditions | * System should be connected with the database. * Application should be running properly. * Admin must have logged in with their credentials |
| Post-conditions | * The faculty member request will be accepted/declined accordingly |
| Author | **S21025A8D8 (BC180404438)** |
| Exceptions | None |

|  |  |
| --- | --- |
| **Use Case Title** | **Manage Faculty Information** |
| Use Case Id | 6 |
| Actions | Admin can view, update, delete faculty information that was added by the faculty member |
| Descriptions | This Use Case will allow admin to view the faculty information as well as admin can update them and can delete them accordingly. All faculty members details will be shown in admin dashboard |
| Alternative Paths | None |
| Pre-conditions | * System should be connected with the database. * Application should be running properly. * Admin must have logged in with their credentials * There must be at least one faculty member’s info |
| Post-conditions | * Details will be update, delete accordingly |
| Author | **S21025A8D8 (BC180404438)** |
| Exceptions | None |

|  |  |
| --- | --- |
| **Use Case Title** | **Add Personal details** |
| Use Case Id | 7 |
| Actions | Faculty Member will add their personal details through the form that will show up in their profile |
| Descriptions | After the approval of admin, Faculty Member will add their personal information that will be shown up in the website that will be viewed by visitor/student. When faculty member will log in to the system. They will redirect to their dashboard where they can add. |
| Alternative Paths | None |
| Pre-conditions | * System should be connected with the database. * Application should be running properly. * Member must have logged in with their credentials |
| Post-conditions | * Information will be added successfully |
| Author | **S21025A8D8 (BC180404438)** |
| Exceptions | None |

|  |  |
| --- | --- |
| **Use Case Title** | **Update Personal Details** |
| Use Case Id | 8 |
| Actions | Faculty Member will update their personal details after adding through the form that will show up in their profile |
| Descriptions | After adding their personal details, Members can edit them accordingly through same form that was used to add |
| Alternative Paths | None |
| Pre-conditions | * System should be connected with the database. * Application should be running properly. * Member must have logged in with their credentials * They must have to add their details first in order to update |
| Post-conditions | * Information will be updated successfully |
| Author | **S21025A8D8 (BC180404438)** |
| Exceptions | None |

|  |  |
| --- | --- |
| **Use Case Title** | **Search Faculty** |
| Use Case Id | 9 |
| Actions | Visitor will see faculty details on website using different filters |
| Descriptions | This Use Case will allow the visitor to search for the faculty Member’s information that has been added into the website by Faculty Member. Visitor can search the faculty using different filters such as by their name, majors, department etc. |
| Alternative Paths | None |
| Pre-conditions | * System should be connected with the database. * Application should be running properly. * There must be at least One Faculty Member’s info |
| Post-conditions | * Visitor has searched the Faculty successfully |
| Author | **S21025A8D8 (BC180404438)** |
| Exceptions | None |

|  |  |
| --- | --- |
| **Use Case Title** | **Recover Password** |
| Use Case Id | 10 |
| Actions | Member can recover the password if they forgot |
| Descriptions | This Use Case will allow the Members to reset their password if they forgot it by using their registered email |
| Alternative Paths | None |
| Pre-conditions | * System should be connected with the database. * Application should be running properly. * Member must have valid email |
| Post-conditions | * Members has recovered password successfully. |
| Author | **S21025A8D8 (BC180404438)** |
| Exceptions | None |

**Adopted Methodology**

According to the requirement, for making our software we adopted the methodology **“VU process model”**. The description of which is given below:

* **VU Process Model:**

It is the combination of waterfall methodology and spiral methodology. It is also known as hybrid approach of system development. It has five phases and which are Gathering and Analyzing requirement, Planning, Analysis and Design, Development and Final report.

Essentially, Water Fall Model is a framework for software development in which development proceeds sequentially through a series of phases, starting with system requirements analysis and leading up to product release and maintenance, whereas the key characteristics of spiral model is risk management at regular stages in the entire software development cycle.

* **Diagram**



* **Why are we using VU Process Model Methodology?**

Following are the basic reasons for choosing VU Process Model for our project:

1. Our project is divided into different phases which we have to complete in sequence and submit it to our supervisor. In the requirement phase the system goals, services and constrains are established after discussion with user.
2. The planning phase is synchronized effort to find all the potential roadblocks and draft a robust plan to tackle each. This phase usually involves a lot of brainstorming within software development team as well as with respective client. It maximizes the quality of system and minimizes any risks and errors.
3. If Supervisor suggests improvement, then we will make improvement in planning phase. This process will be adopted due to the spiral nature of VU Process model. When the phase is well-improved and well worked-out, and also accepted by our supervisor then we will proceed to next phase. This will be done due to the waterfall nature of the VU Process model.

Hence both models are suppressed in single form and that will be our VU process model. It will provide error free result as each step is done in a sequence.

**Work Plan**

* **Timeline**

Start  
May 21, 2021

Finish  
October 4, 2021

June

July

August

September

October

**Software Requirements & Specifications (SRS)**  
May 21, 2021 - June 15, 2021

**Design**  
June 16, 2021 - July 15, 2021

**Test Phase**  
July 16, 2021 - August 2, 2021

**Final Deliverable**  
August 3, 2021 - October 4, 2021

**Today**

* **GANTT Chart**

