Software Requirements Specification

Version 3.0

24th March 2016

Financial Grant Management System

Submitted in partial fulfillment

of the requirements of

CS 223 Software Engineering

This work is based upon the submissions of the course Software Engineering (CS223). The students who submitted this team projects were :

* Abhay Kumar Singh (UG201310003)
* Deepshi Garg (UG201313008)
* K. Sai Raghava (UG201310016)

# Table of Contents

[Table of Contents](#h.1fob9te)

[List of Figures](#h.2et92p0)

[1.0. Introduction](#h.tyjcwt)

[1.1. Purpose](#h.3dy6vkm)

[1.2. Scope of Project](#h.1t3h5sf)

[1.3 Constraints](#h.4d34og8)

[1.4 Assumptions and Dependencies](#h.2s8eyo1)

[1.3. Glossary](#h.17dp8vu)

[1.4. References](#h.3rdcrjn)

[1.5. Overview of Document](#h.26in1rg)

[2.0. Overall Description](#h.lnxbz9)

[2.1 System Environment](#h.35nkun2)

[2.2 Functional Requirements Specification](#h.1ksv4uv)

[2.2.1 Use case 1](#h.44sinio)

[Use case:](#h.2jxsxqh)

[2.3 User Characteristics](#h.z337ya)

[2.4 Non-Functional Requirements](#h.3j2qqm3)

[3.0. Requirements Specification](#h.1y810tw)

[3.1 Functional Requirements](#h.4i7ojhp)

[3.1.1 << Name of the first feature>>](#h.2xcytpi)

[3.1.2 << Name of the second feature>>](#h.1ci93xb)

[3.3 Detailed Non-Functional Requirements](#h.3whwml4)

[***3.4*** ***Logical Structure of the Data***](#h.2bn6wsx)

[4.0 Supporting information](#h.qsh70q)

[4.1 Table of contents and index](#h.3as4poj)

[4.2 Appendixes](#h.1pxezwc)

# List of Figures

|  |  |  |
| --- | --- | --- |
| **S. No** | **Diagram Name** | **Page No.** |
| 1 | Use Case Diagram | 9 |
| 2 | Class Diagram | 10 |
| 3 | Sequence Diagram : Request Grant | 10 |
| 4 | Sequence Diagram : Settle Grant | 11 |
| 5 | Sequence Diagram : Access Permission | 11 |
| 6 | Sequence Diagram : Ask For Help | 12 |
| 7 | Sequence Diagram : Cancel Grant (By Institute People) | 12 |
| 8 | Sequence Diagram : Print Grant Log | 13 |
| 9 | Sequence Diagram : Verify Request (By Admin2) | 13 |
| 10 | Sequence Diagram : Cancel Request (Admin2) | 14 |
| 11 | Sequence Diagram : Add/Delete User (Admin2) | 14 |
| 12 | Sequence Diagram : Approve Request (Admin1) | 15 |
| 13 | Sequence Diagram : Reject Request (Admin1) | 15 |
| 14 | Activity Diagram : Request Grant | 16 |
| 15 | Activity Diagram : Settle Grant | 16 |
| 16 | Activity Diagram : Request Access Permission | 17 |
| 17 | Activity Diagram : Ask For Help | 17 |
| 18 | Activity Diagram : Cancel Grant | 18 |
| 19 | Activity Diagram : Print Activity | 18 |
| 20 | Activity Diagram : Registration Activity | 19 |

# 1.0. Introduction

## 1.1. Purpose

The purpose of this document is to manage the records of financial grants of Institute people. It will allow research professionals to spend more time on research and discovery and less on administration. Using software for this purpose will avoid any duplicity or manipulation of records and thus stop corruption. It will track pre and post-award information at institution with a significant number of proposals, grants or research projects. It will provide privacy and reliability.

## 1.2. Scope of Project

This software system will be designed for 3 kind of users :

* **Admin1**
* **Admin2**
* **Institute People (Faculty, Phd, MTech.)**

**Role of Admin1** : Admin1 is a super user who can modify (update) all database. He can grant permission to other users if they want to access other user’s information. He adds new users, and also deletes existing ones.

**Role of Admin2 :** Admin2 verifies the requests for new grants from Institute People and rejects them, or forwards them to Admin1 for final approval. Also, he verifies the settlement requests by manually looking at the bills.

**Role of Institute People** : It includes **Faculty, Phd & MTech.** These users can log into their account using their unique institute email id as username and their own password. These users can access (read) their own database only. They can enter a new grant request but can’t make changes in older one. If they want to make change, they will have to cancel the previous grant till it is pending and request for a new one. Limits on request will be there based on the roles.

**Apart from this, there will be different types of grants which a user can request for**.

**Types of Grants** :

* Resources
* Travelling Allowances (Different based on roles)
* Medical Grants
* Other (Conference Fee, Accommodation Charges, Stationary etc)

## 1.3 Constraints

* Only the person with valid email id and valid role can access this software.
* Users will have constraints depending on their roles. For eg. Limits of different users are different.
* A user can access own database only. Only faculty can request to access the grants log of PhD or MTech. Moreover, user cannot make any changes in already requested grant. If he wants to do so, he will have to cancel it and ask for it again.

## 1.4 Assumptions and Dependencies

* There is an **Admin1 and an Admin2** who manages all the grant request. Every user, who request for grant, his request is sent to the Admin and he is the only authority to grant it.
* **BTech.** students cannot apply for grants. If they wish to, they can ask their project mentor to do so.
* **Admins** are not corrupt.
* Admin2 verifies all the bills by personally looking at them.

## 1.3. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Institute People | Faculty, Phd, MTech. These are the people who request for the grants. |
| Admin1 | Any higher authority who manages the complete database of grants i.e he approve, reject requests and add, delete users. |
| Admin2 | A person who manually verifies every request before forwarding it to the Admin2. |
| Database | Collection of all information monitored by this system. |
| User | Admins & Institute People |
| Request Grant Window | It’s a screen comprises of different option of grants from where a user can request for grant. |
| Login Screen | It’s the first opening window that appears on the screen when someone starts this software. A user enter his username, password and select his role to get access to his database. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Roles | Faculty, Phd, MTech., Admin1, Admin2 |
| Valid Email ID | Email ID provided by the institute to the user. |
| Limits | A class which will contain all the details regarding the limits on grant requests of each type |
| Notification | A class whose attribute is a string which will contain a message |
| Pending Grants | Grant requests which are yet to be approved. |
| Unsettled Grants | Grants which have been approved but has not being settled |
| Settled Grants | All settled and cancelled grants. |

## 1.4. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## 1.5. Overview of Document

The rest of the document is designed in the following way:

# 2.0. Overall Description

## 2.1 System Environment

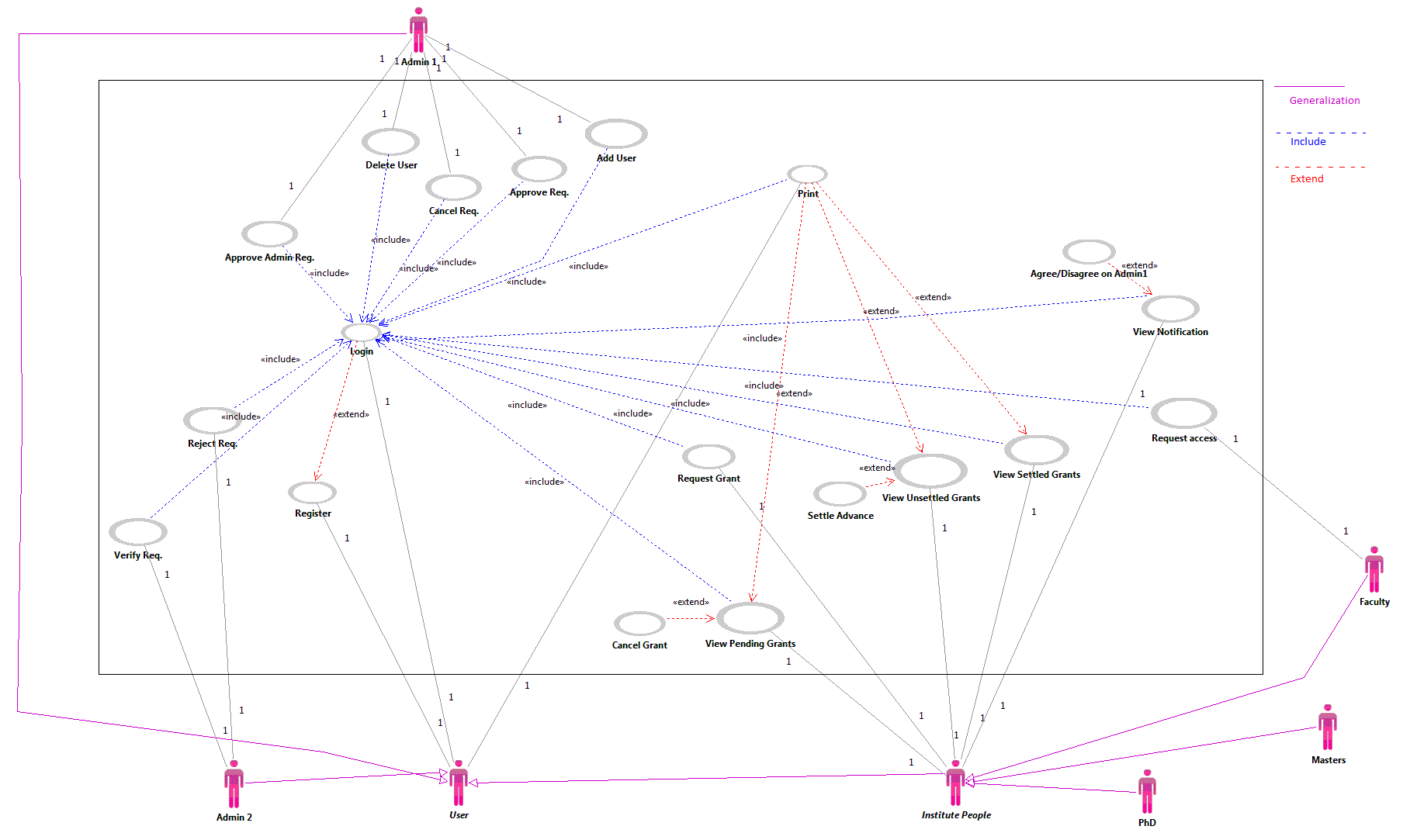
* Backend: PHP
* Database: MYSQL
* Frontend: Bootstrap, HTML, CSS
* Operating system: Linux
* Local server : Xampp

## 2.2 Functional Requirements Specification

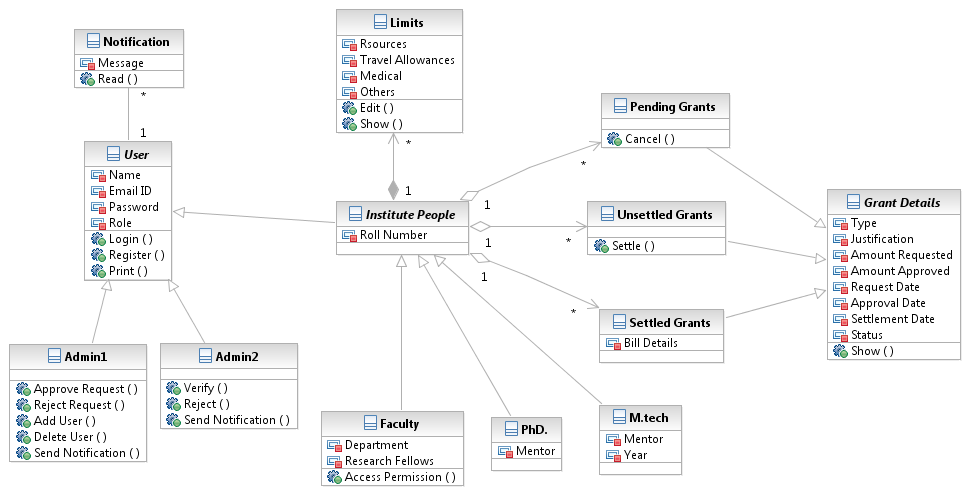
* phpMyAdmin
* Browser

### 

### ***2.2.1 Use Case Diagram***



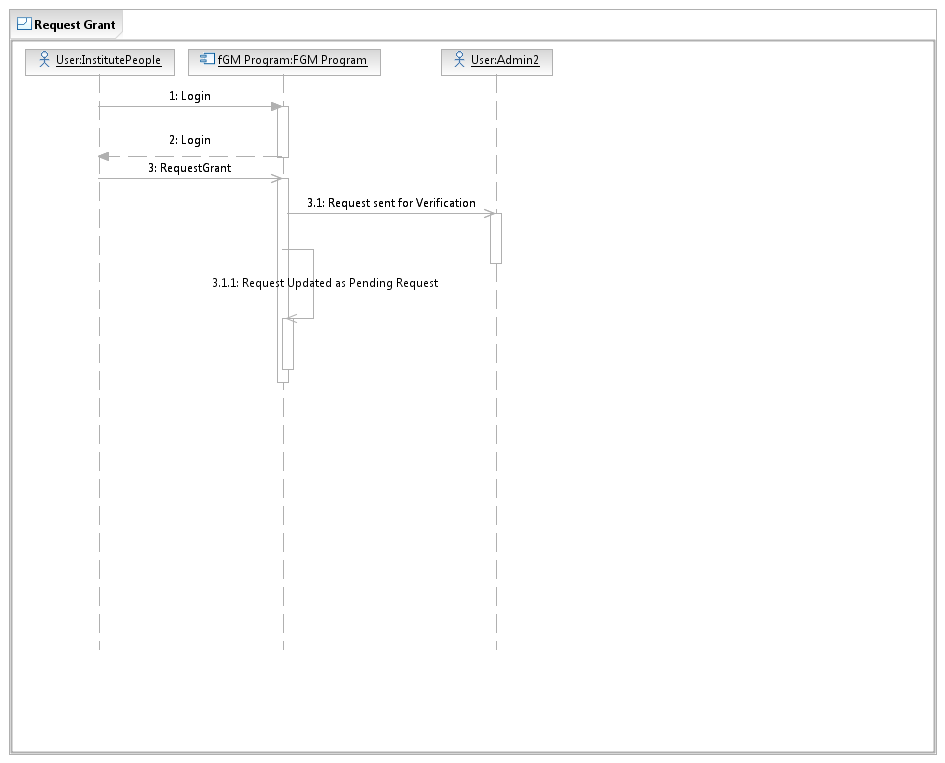
***2.2.2 Class Diagram***



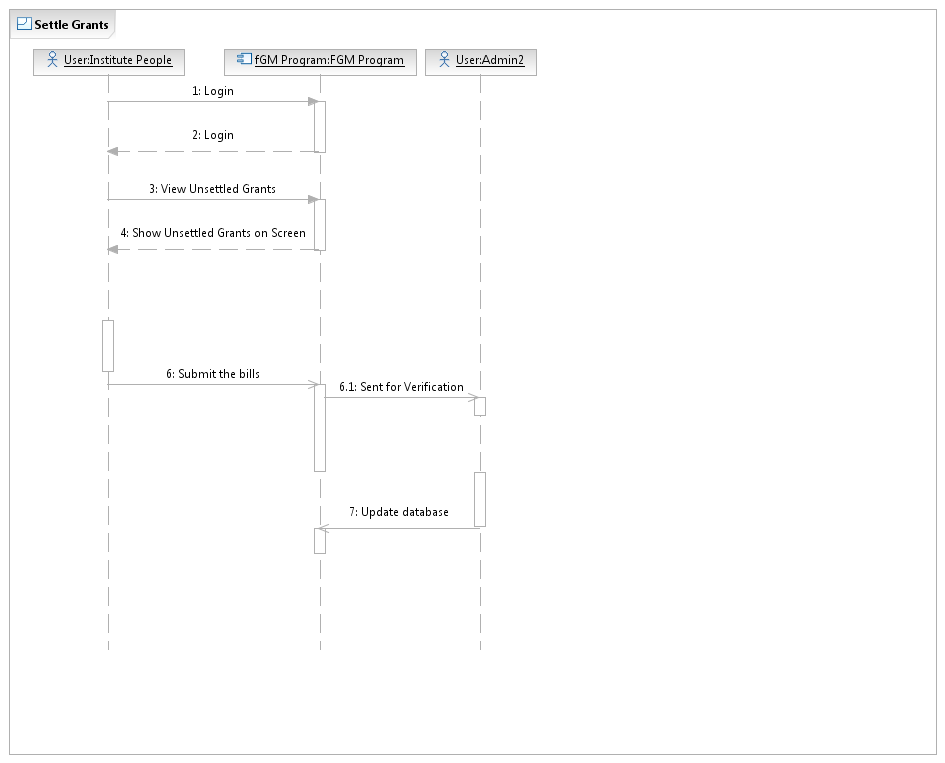
## 

## 2.2.3 Sequence Diagram

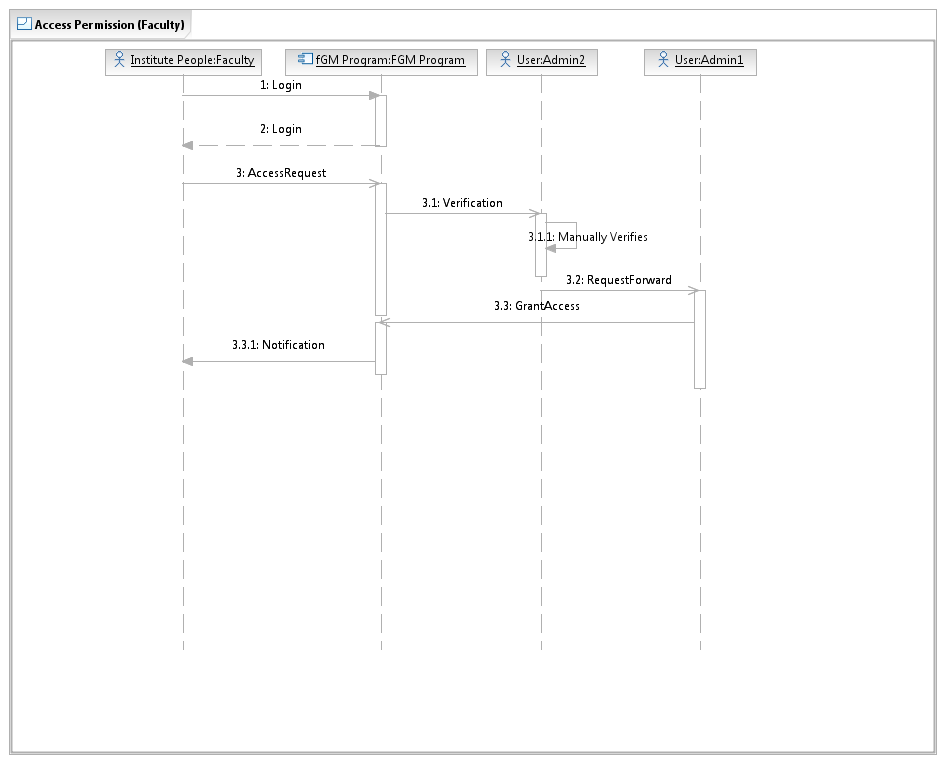
1. **Request Grant**



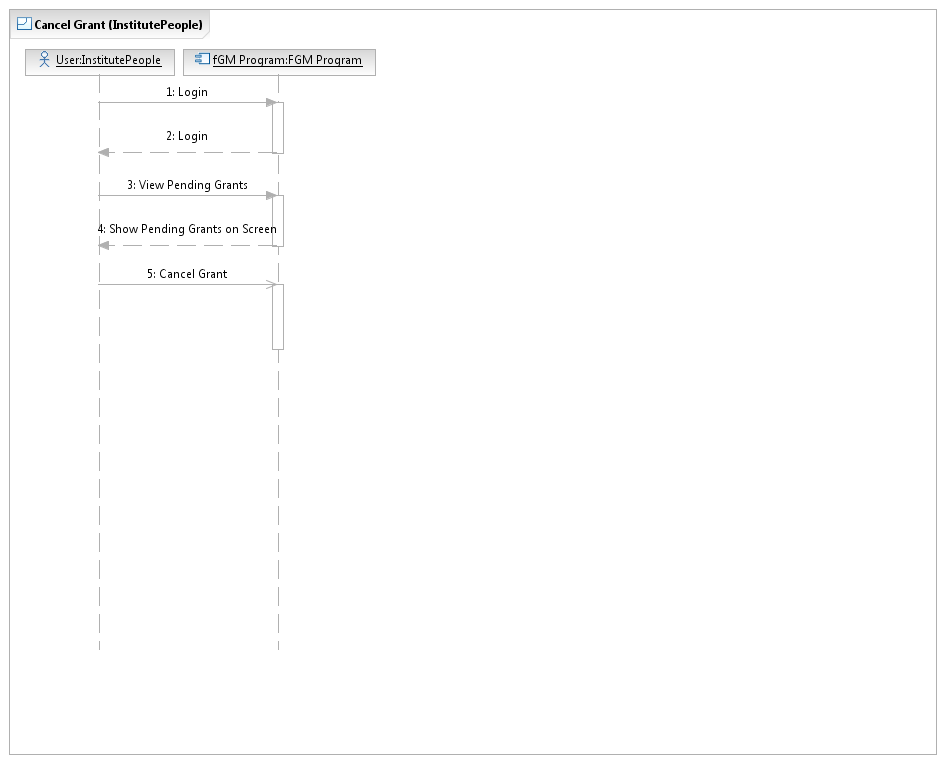
**2. Settle Grant**



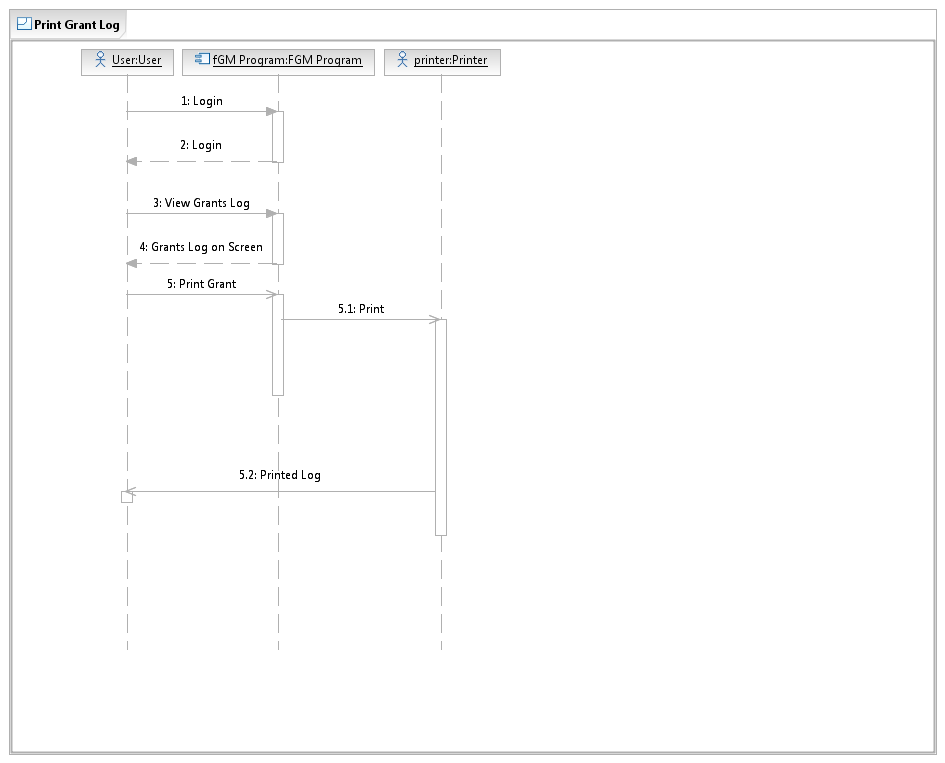
**3. Access Permission**



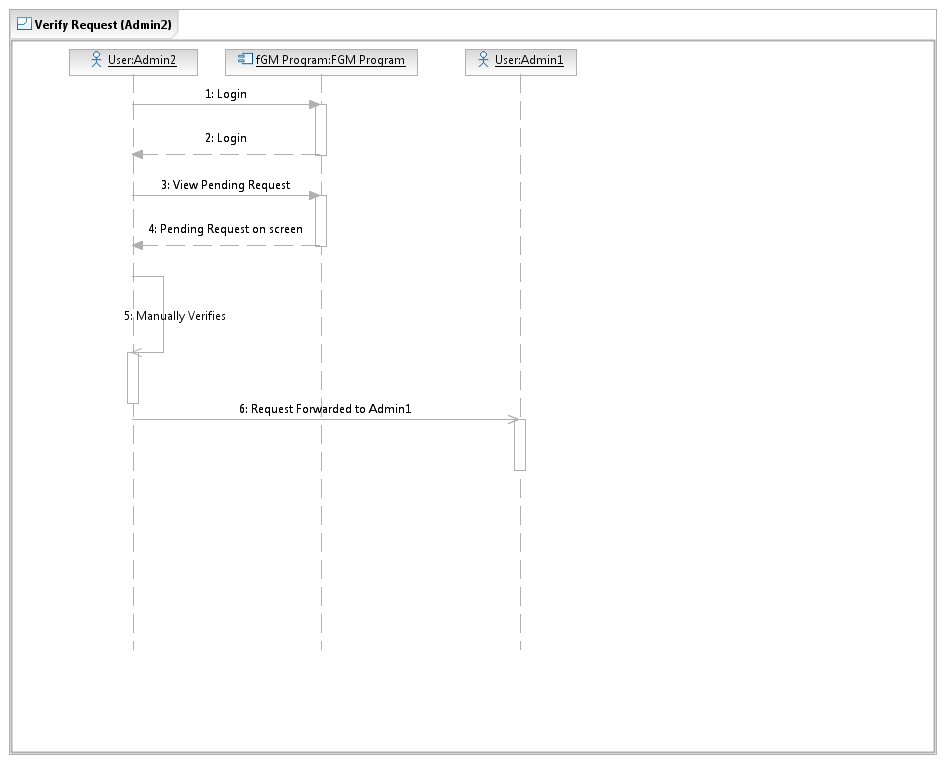
**5. Cancel Grant (By Institute People)**



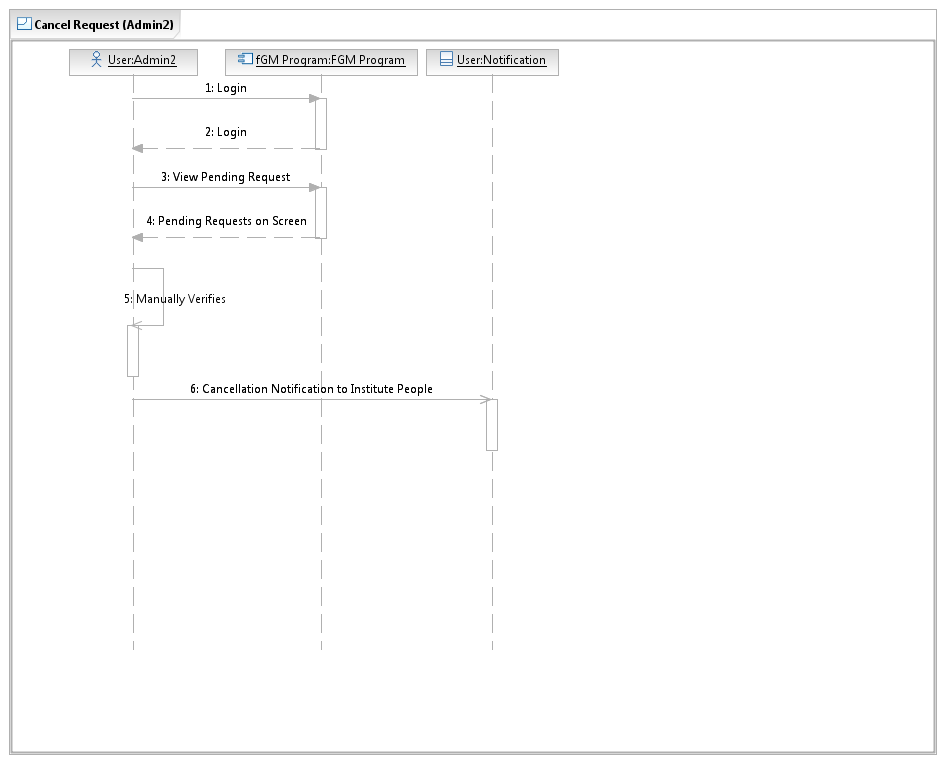
**6. Print Grant Log**



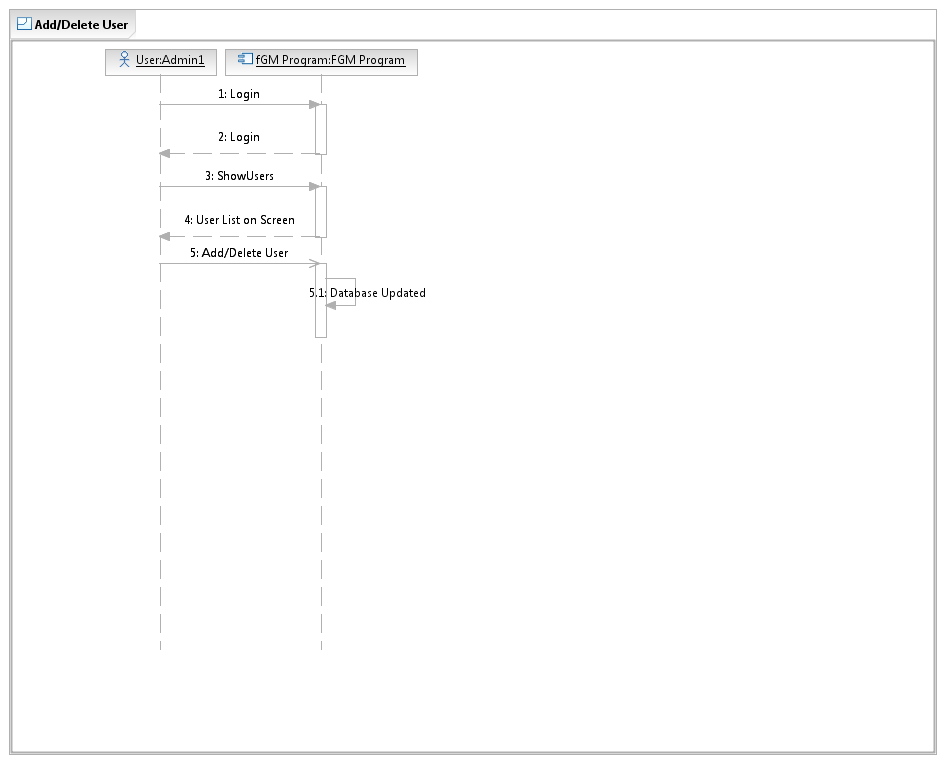
**7. Verify Request (By Admin2)**



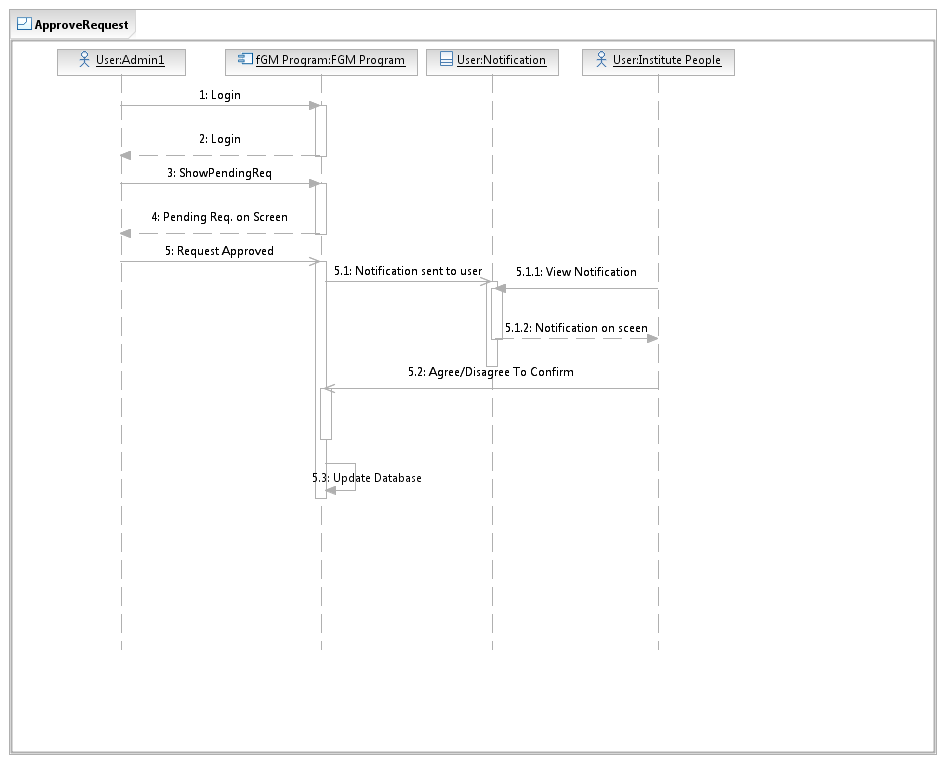
**8. Cancel Request (Admin2)**



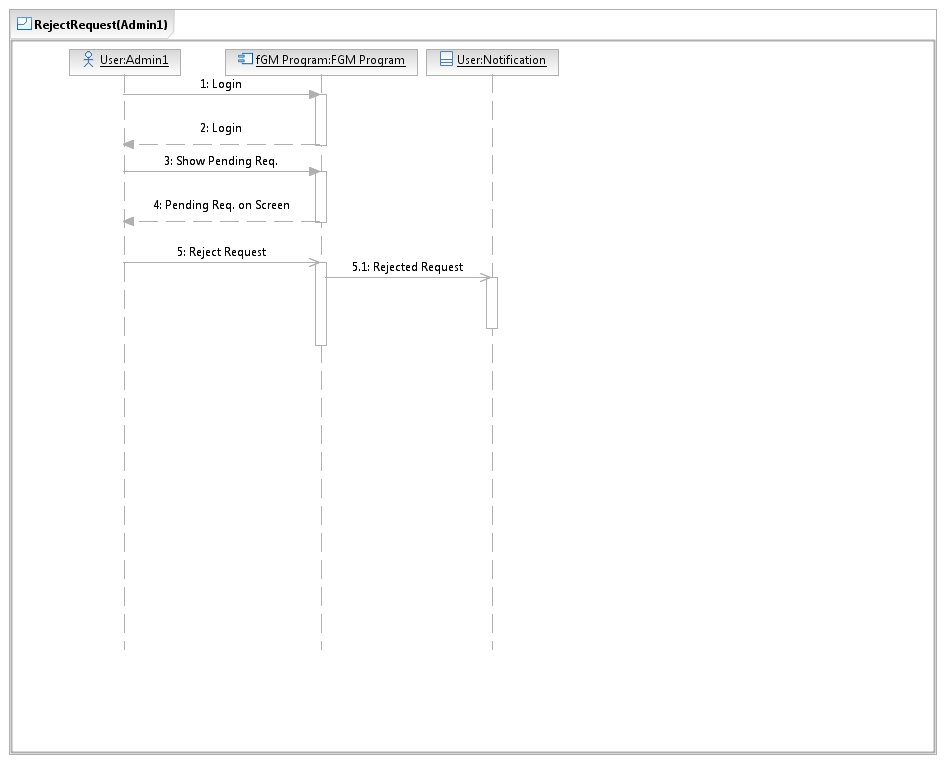
**9. Add/Delete User (By Admin1)**



**10. Approve Request (By Admin1)**

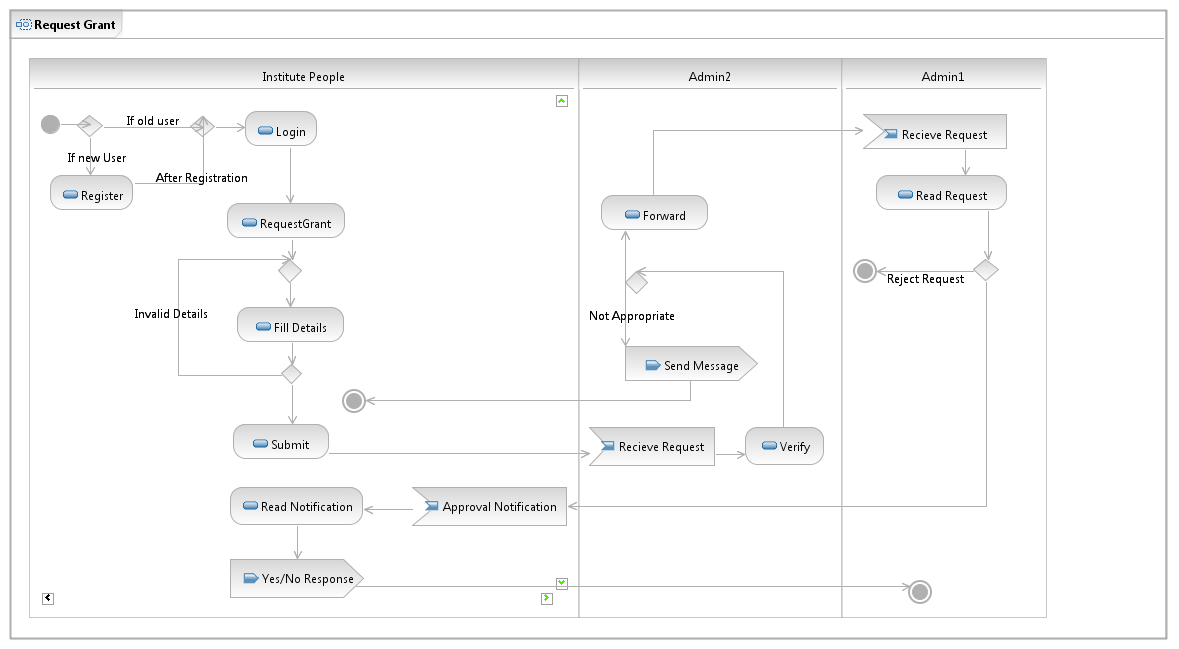


**11. Reject Request (By Admin1)**

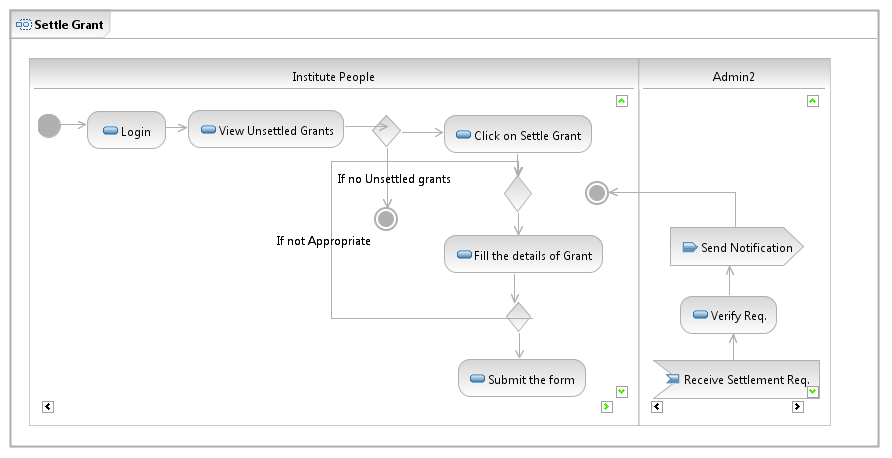


***2.2.4 Activity Diagrams***

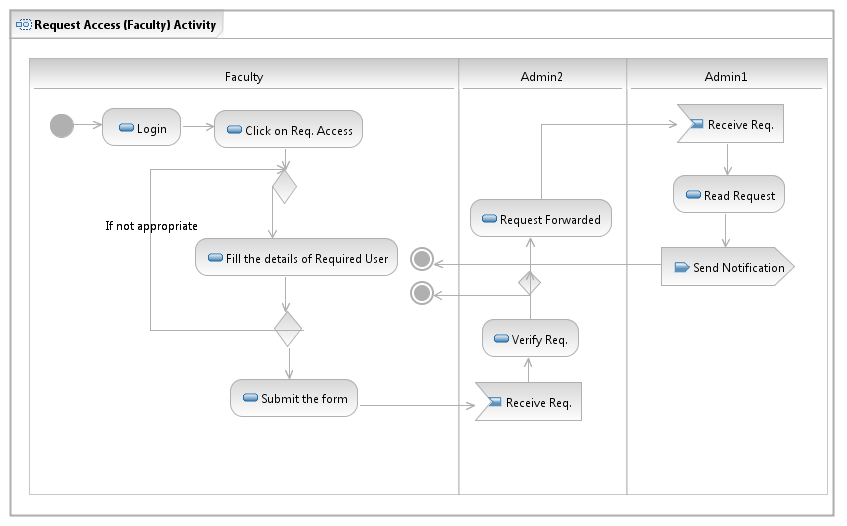
1. **Request Grants**



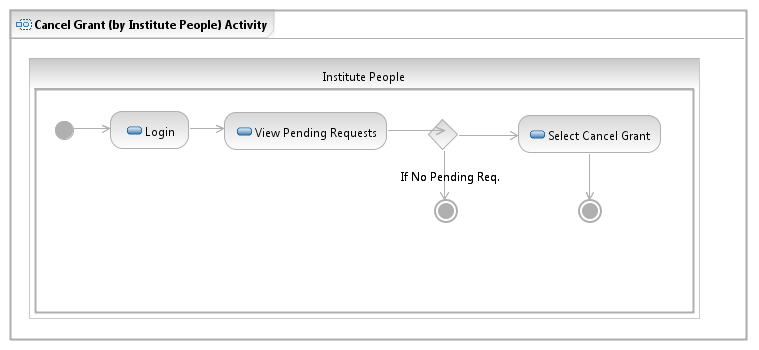
**2. Settle Grants**



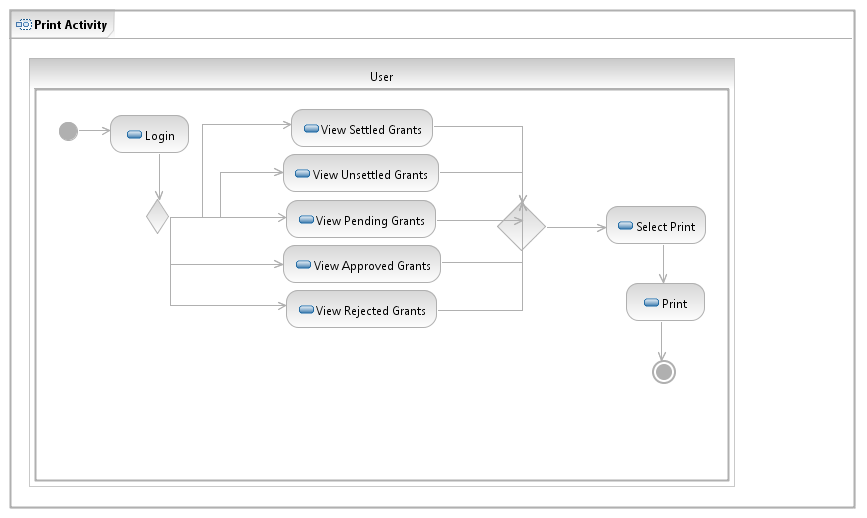
**3. Request Access Permission**



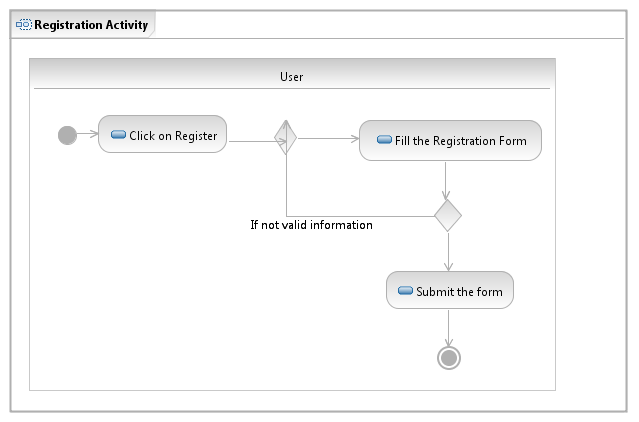
**5. Cancel Grants**



**6. Print Activity**



**7. Registration Activity**



## 2.3 User Characteristics

The user of this software system requires the following skills to use this software:

* User must be able to handle a computer. Basic knowledge of computer is required.
* He must have clear idea of his roles.
* He must be familiar with software where a person has to enter username and password to get access of information.
* He must be able to understand the error messages that appear on the screen when he exceed limits on the requests.

## 2.4 Non-Functional Requirements

# 3.0. Requirements Specification

## 3.1 Functional Requirements

### 3.1.1 Login

|  |  |
| --- | --- |
| **Use Case Name** | Login |
| **Trigger** | User open the software |
| **Precondition** | Login screen is in front of him |
| **Basic Path** | 1. User enter his username 2. User enter his password 3. User selects his role 4. System verifies the above details 5. After verification, user is logged into his account |
| **Alternative Paths** | NO Alternative Path |
| **Postcondition** | User is able to access his database |
| **Exception Paths** | The user may abandon the process at any time. |
| **Other** | Nothing |

### 

### 3.1.2 Register User

|  |  |
| --- | --- |
| **Use Case Name** | Register User |
| **Trigger** | The new user (unregistered) selects to register button on the login screen |
| **Precondition** | The new user (unregistered) has accessed the main login screen and has a valid email id for registration |
| **Basic Path** | 1. The system presents a screen with blank grids to fill details 2. The new user enters the information and submits the form. 3. The system checks if the email id is valid or not. 4. The system checks that the name and the email address fields are not blank and updates the database. |
| **Alternative Paths** | NO Alternate Path |
| **Postcondition** | The new user is registered |
| **Exception Paths** | The new user may abandon the process at any time. |
| **Other** | For registration, the new user must have a valid email id (provided by institute) & he must have valid roles (faculty,Phd’s,Mtech.) |

### 3.1.3 Print Grant (User)

|  |  |
| --- | --- |
| **Use Case Name** | Print Grant |
| **Trigger** | User click on the print grant button |
| **Precondition** | User must be logged into his account and there must be any grant request in his record |
| **Basic Path** | 1. User searches for his grant request. 2. Once found, he selects it. 3. Click on print button |
| **Alternative Paths** | NO Alternative Path |
| **Postcondition** | The selected grant request is printed |
| **Exception Paths** | The user may abandon the process at any time. |
| **Other** | Nothing |

### 

### 

### 3.1.4 Add User (Admin1)

|  |  |
| --- | --- |
| **Use Case Name** | Add User |
| **Trigger** | The Admin1 selects add user button on the login screen |
| **Precondition** | The admin has accessed the main login screen and has a valid email id of new user to be added |
| **Basic Path** | 1. The system presents a screen with blank grids to email id 2. The Admin1 enters the information and submits. 3. The system checks if the email id is valid or not. |
| **Alternative Paths** | NO Alternate Path |
| **Postcondition** | The system updates this new email id into the database |
| **Exception Paths** | The new user may abandon the process at any time. |
| **Other** | Nothing |

### 3.1.5 Remove User (By Admin1)

|  |  |
| --- | --- |
| **Use Case Name** | Remove User |
| **Trigger** | Admin user clicks on the remove user button |
| **Precondition** | The user have left the institute |
| **Basic Path** | 1. Admin user enter the details of the user to be removed 2. System prompts for verification of entered details 3. Admin verifies and delete the user |
| **Alternative Paths** | In step 1, instead of entering details, admin can search manually for removal of user. |
| **Postcondition** | Selected user is deleted from the record |
| **Exception Paths** | The admin may abandon the process at any time. |
| **Other** | Admin must know the email ID of the user to be removed |

### 

### 3.1.6 Approve Request (Admin1)

|  |  |
| --- | --- |
| **Use Case Name** | Approve Request |
| **Trigger** | Admin click on pending or new grant request button |
| **Precondition** | Admin must be logged into his account and there must be any request in the database. |
| **Basic Path** | 1. Admin look at the request 2. If found appropriate, click on the Approve button 3. After approval, notification is sent to the user |
| **Alternative Paths** | In step 2, if the request is inappropriate, Admin cancels that request and look at the another one. |
| **Postcondition** | Notification is sent to the user requesting for grant |
| **Exception Paths** | The user may abandon the process at any time. |
| **Other** | Nothing |

### 

### 3.1.7 Reject Request (By Admin1)

|  |  |
| --- | --- |
| **Use Case Name** | Reject Request (By Admin1) |
| **Trigger** | Admin click on the Pending Request button |
| **Precondition** | Admin must be logged into his account and there must be a request in pending or new request table. |
| **Basic Path** | 1. Admin searches for the inappropriate request 2. Admin selects that request 3. Admin click on cancel request button |
| **Alternative Paths** | NO Alternate Path |
| **Postcondition** | That particular grant request is cancelled and a notification is sent to the user |
| **Exception Paths** | The user may abandon the process at any time. |
| **Other** | Nothing |

### 

### 3.1.8 Verify/Reject Request (By Admin2)

|  |  |
| --- | --- |
| **Use Case Name** | Verify/Reject Request (By Admin2) |
| **Trigger** | Admin click on the Pending Request button |
| **Precondition** | Admin must be logged into his account and there must be a request in pending or new request table. |
| **Basic Path** | 1. Admin click on the request 2. Admin verifies that request 3. Admin click on reject request button if request is inappropriate or click on verified button if it is appropriate |
| **Alternative Paths** | NO Alternate Path |
| **Postcondition** | That particular grant request is cancelled and a notification is sent to the user as a note. |
| **Exception Paths** | The user may abandon the process at any time. |
| **Other** | Nothing |

### 

### 3.1.9 Request Grant (By Institute People)

|  |  |
| --- | --- |
| **Use Case Name** | Request Grant |
| **Trigger** | The user selects to grant request button on the screen. |
| **Precondition** | The user has logged into the software using his email id and password |
| **Basic Path** | 1. The system presents a screen with various option to click (types of grants) 2. The user click on one of the type. 3. The user enter all the details (pre-award information) regarding a particular grant. 4. The system verifies if the request is valid or not. 5. The user click on submit button to finish the process. |
| **Alternative Paths** | In step 4, if system invalidate a request (if cost is too much), user can go back to step 2 and directly request for the hardware instead of cash |
| **Postcondition** | Request for the grant is sent |
| **Exception Paths** | The user may abandon the process at any time. |
| **Other** | There are limits on the grant quantity for different kind of users. Faculties are provided with maximum limit and Mtech. with minimum limit. |

### 

### 

### 

### 3.1.10 Cancel Grant (By Institute People)

|  |  |
| --- | --- |
| **Use Case Name** | Cancel Grant (By Institute People) |
| **Trigger** | User click on the View Pending Grant button |
| **Precondition** | User must be logged into his account and there must be a pending grant request in his database |
| **Basic Path** | 1. User searches the grant request to be cancelled 2. User select the particular request 3. He click on the cancel grant button |
| **Alternative Paths** | NO Alternate Path |
| **Postcondition** | That particular grant request is cancelled |
| **Exception Paths** | The user may abandon the process at any time. |
| **Other** | Nothing |

### 

### 

### 

### 

### 3.1.11 Settle Grant (By Institute People)

|  |  |
| --- | --- |
| **Use Case Name** | Settle Grant (Institute People) |
| **Trigger** | Admin click on the View Unsettled Grant button |
| **Precondition** | Admin must be logged into his account and there must be a unsettled grant |
| **Basic Path** | 1. User searches for an unsettled grant. 2. User fill all the details and upload required documents. 3. User click on submit button. |
| **Alternative Paths** | NO Alternate Path |
| **Postcondition** | Query for settlement of that grant is sent to the Admin2 |
| **Exception Paths** | The user may abandon the process at any time. |
| **Other** | Nothing |

### 

### 3.1.12 Permission to access other’s database (By Faculty)

|  |  |
| --- | --- |
| **Use Case Name** | Access Permission |
| **Trigger** | User click on the access permission button for a particular user |
| **Precondition** | Both the users must be there in the database |
| **Basic Path** | 1. Admin click on the Pending Request 2. If any request present, he verifies it 3. If appropriate, he grants permission |
| **Alternative Paths** | No Alternative Path |
| **Postcondition** | Requesting user is able to see other’s user database |
| **Exception Paths** | The user may abandon the process at any time. |
| **Other** | Now, that user can only see the database of that user whom he requested for, not all the users. |

### 

### 3.1.13 Change Password

### 

|  |  |
| --- | --- |
| **Use Case Name** | Change Password |
| **Trigger** | User click on change password in his home screen |
| **Precondition** | Both the users must be there in the database |
| **Basic Path** | 1. User login to his account 2. User click on change password option 3. New screen appears where user enter his old and new password 4. System verifies old password and therefore update the password |
| **Alternative Paths** | No Alternative Path |
| **Postcondition** | User is able to change his password |
| **Exception Paths** | The user may abandon the process at any time. |
| **Other** | Now, that user can only see the database of that user whom he requested for, not all the users. |

### 

## 3.3 Detailed Non-Functional Requirements

### ***3.4 Logical Structure of the Data***

<< Keep this blank for the time being>>

# 4.0 Supporting information

## 4.1 Table of contents and index

## 4.2 Appendixes