# REQUIREMENT ANALYSIS AND DESIGN

## Introduction

This chapter discusses the requirement analysis and design of the FC Research Grant Finder System. This part consists of the functional and non-functional requirements of the system. This document will describe the Software Requirement Specification (SRS), Software Design Documentation (SDD) and Software Testing Document (STD) in short. Various diagram such as Use Case, Sequence diagram and activity diagram are drawn for each use case in details. Each use case also has its specification where the details description of the use cases are given. Finally the interface of the system is designed to provide a clear idea about the system.

## Requirement Analysis

By doing consecutive meetings with the stakeholders, several functional and non-functional requirements of the system were identified which are discussed below:

### Functional Requirements

1. The application allows user to login to the system.
2. The admin can insert user data into the system.
3. The admin can modify and delete user data.
4. The admin can view the extracted data from the web scraping.
5. The admin can filter the extracted data.
6. The admin can upload the data to database.
7. The admin can transform and operate on the data.
8. The users can view the dashboard.
9. The users can search for grants.
10. The users can save grants information.

#### Use Case Diagram

The use case consists of all the functionalities and the actors that are involved in the system. It gives a clear idea of how the system works and what are the roles of each actors in the system. The figure below represents the use case diagram of the FC Research Grant Finder system.

**A picture containing text, diagram, line

Description automatically generated**

Figure 4.1 Use Case Diagram of FC Research Grant Finder system.

#### Use Case Actor Description

Table 4.1: Actor Description

|  |  |
| --- | --- |
| User | Characteristics |
| Research Officer | View the extracted data, clean the data, edit user info. |
| Research Manager | View, modify the extracted data, channel between the research head and officer. |
| Research Head | Oversees the changes made by the admin, confirms, and verified the changes and manage the scrapping. |
| Researcher/ End user | Views the dashboard, access the data, search for grants, view and save them. |

#### Use Case Description

Fig 4.2: Use Case Description

|  |  |  |
| --- | --- | --- |
| **Module** | **Use cases** | **Description** |
| Authentication | Login | Allows the users to access the system by entering their valid details. |
| Data Management | Manage User Data | Allows the admin to insert, create and update the user information or delete them. |
| View the Extracted Data | This use case allows the admin to view the raw data that has been found by scraping the target websites. |
| Arrange and filter data | Allows the research heads and manager to reorganize the extracted data, filter it according to the necessary categories ant make it ready for preprocessing. |
| Data Preprocessing & Dashboard Creation | Allows user(admin) to process the data and then upload it to the database for visual representation in the form of a dashboard. |
| Manage Scraping | Manage Scraping | Allows research head to manage the web scraping process by letting him/her modify the different parameters that control the scrapping. |
| View Dashboard | View Dashboard | Allows all the actors in the system to view the final product which contains the visualization of all the extracted data. |
| Search Grants | Search for grants | Allows the user to search for grant using specific keywords. |
| Filter and sort grants | Allows to narrow the search by selecting specific categories form the data. |
| View Grants | View Grant Details | Allows the user to view each grants information in details. |
| Save Grants | Allows user to save grants and integrate their data with their profile. |
| Deadline Notification | Allows the registered user to receive notification about the deadline of their saved grants. |

#### Use Case Specification

Table 4.3: Use Case specification

|  |  |
| --- | --- |
| **Use Case ID** | UC |
| **Use Case Name** | **Data Preprocessing and Dashboard Creation** |
| **Description** | The research officer, research manager and the research head get the extracted data from the web scraping model, clean it and then do necessary modifications and then upload the clean processed data to the database. |
| **Actor(s)** | Research Head, Research Manager, Research Officer. |
| **Pre-conditions** | The actors must be logged in to the system. |
| **Normal Flow** | 1. View the extracted data.    1. System looks for extracted data in the web scraping module.    2. System retrieves the data for preprocessing. AF1 will be executed.    3. System does not find any data, EF1 will be executed. 2. Clean the data.    1. The admin cleans the data that is extracted.    2. The admin removes the data that is not needed for the system.    3. Separate dependent and independent variables.    4. Transform data. 3. Modify the data.    1. Admin changes the data that is not understandable into an easier form.    2. Connect data with the independent variables. AF2 will be executed. 4. Confirm the changes.    1. The admin views the changed data.    2. The data is not proper. EF2 will be performed.    3. The admin confirms the data and clicks on the save button.    4. The data is uploaded to the database. |
| **Alternative Flow** | 1. If the admin is the research manager go to NF3. 2. If the admin is not Research Head, stop at NF3 |
| **Exception Flow** | 1. The system shows an error message, and the process ends. 2. Admin clicks on discard changes.    1. System displays error message.    2. Data not saved in database. |
| **Post Conditions** | The modified preprocessed data is stored in the system database. |
| **Related Requirements** |  |

### Non-Functional Requirements

The non-functional requirements of the system include Usability, reliability, maintainability, and security. The SRS provides a clearer description of the non-functional requirements of the system.

### Sequence DiagramsA picture containing screenshot, black, design Description automatically generated

Figure 4.2: One of the sequence diagram of FC Research Grant Finder system

### Activity Diagram

**A black background with white rectangles

Description automatically generated with low confidence**

Figure 4.3: One of the activity diagram of FC Research Grant Finder system.

## Project Design

The system architecture that will be used for this system will be MVC or Model-View-Controller. This architecture contains 3 layers which are model, view and controller.

A screenshot of a computer screen

Description automatically generated with low confidence

Figure 4.4: Architecture model of FC Research Grant Finder System

.

**Model:** The Model represents the application's data and business logic. It contains the data of the application and defines how it may be accessed, edited, and validated. It's in charge of data manipulation, storage, and retrieval. The Model frequently interacts with the database to fetch or update data in the context of a database-driven application. It should, however, be independent of the specific data storage implementation.

**View:** The View is in charge of displaying the application's user interface to end users. It specifies how data will be displayed and how users will interact with it. The Model sends data to the View, which formats it for presentation by producing HTML templates or showing GUI elements. It should not include any business logic and should instead concentrate on visualising the data provided by the Model.

**Controller:** The Controller functions as a go-between for the Model and the View. It processes user input and events, changes the Model, and picks the appropriate View to present the changed data. The Controller takes user input, such as button clicks or form submissions, and converts it into actions that change the state of the Model. It also connects with the View in order to adjust the user interface based on Model changes.

## Database Design

This section describes how the database of the FC Research Finder System was designed. Database design is very important in developing and implementing the system and for the system to work properly.

*A picture containing screenshot, text, design

Description automatically generated*

Figure 4.5: Database Diagram of FC Research Grant Finder system.

### Data Dictionary

A data dictionary provides information about the data that are stored in the database. The data are discussed below:

Table 4.4: Data dictionary of the system

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Constraints** | **Description** |
| Research Manager | | | |
| ManagerEmailID | String | Primary Key | Unique ID for research manager |
| Name | String | Not Null | Name of the research Manager |
| Password | String | Not Null | Password that the research manager uses to log into the system. |
| Research Head | | | |
| HeadEmailID | String | Primary Key | Unique ID for research head |
| Name | String | Not Null | Name of the research Head |
| Password | String | Not Null | Password that the research head uses to log into the system. |
| Research Officer | | | |
| OfficerEmailID | String | Primary Key | Unique ID for research officer |
| Name | String | Not Null | Name of the research officer |
| Password | String | Not Null | Password that the research officer uses to log into the system. |
| Field Type | | | |
| FieldID | Int | Primary Key | Unique ID for each field of research. |
| Name | String | Not Null | The name of the fields. |
| GrantID | Int | Foreign Key | The grantID to get the grants related to the field. |
| FundingOrgID | Int | Foreign Key | The FundingOrgID to get the sponsors related to the field. |
| Grant | | | |
| GrantID | Int | Primary Key | Unique ID for each grants. |
| Name | String | Not Null | The title of each grants. |
| grantType | String | Nullable | The type of grants. |
| Description | String | Nullable | The details of each grant. |
| grantAmount | Int | Not Null | The amount offered in each grants |
| grantStatus | Bool | Not Null | Determines if the grant is over or ongoing. |
| FieldID | Int | Foreign Key | Access the field of research related to grants |
| FundingOrganizationID | Int | Foreign Key | Access the funding organizations that provide grants. |
| Funding Organization | | | |
| FundingOrganizationID | Int | Primary Key | Unique ID for representing the funding organization. |
| Name | String | Not Null | Name of the funding organization |
| GrantID | Int | Foreign Key | Access the grant information that have the same funding organization. |
| FieldID | Int | Foreign Key | Access the number of fields an organization is funding. |
| Faculty | | | |
| FacultyID | Int | Primary Key | Unique ID for representing faculty. |
| FacultyName | String | Not Null | Provided the name of the faculty. |
| LecturerID | Int | Foreign Key | Access the lecturers in a faculty |
| GrantID | Int | Foreign Key | Access the grants by each faculty. |
| Grant Status | | | |
| GrantStatusID | Bool | Primary Key | Stores the true or false value if the grant exist or not. |
| startDate | Date | Not Null | The start date of the grant |
| endDate | Date | Not Null | The ending date of grants. |
| GrantID | Int | Foreign Key | Access the grant information to update the status. |
| Lecturer | | | |
| LecturerID |  | Primary Key | Unique ID to represent the lecturer of faculty. |
| Name | String | Nullable | Name of the lecturer. |
| FacultyID | Int | Foreign Key | Access the faculty details the lecturer belongs to. |
| FieldID | Int | Foreign Key | Access the field of research of the grants |
| GrantID | Int | Foreign Key | Access the grant information. |

## Interface Design

The interface design helps us to get a blueprint of how the system works and how it will look like once it is developed. It is a very important part. It shows the stakeholders about the proposed system and gives them more room for suggestions during development.

## Chapter Summary

Finally, the full architecture and design of the system have been demonstrated and planned in this chapter. Each use case, class, and component diagram has been thoroughly described. All of the demands for the FC Research Grant Finder system, including functional and non-functional requirements, have been outlined in the use case diagram for the complete system. Furthermore, the full explanation will be included in the Software Requirements Specification (SRS) and Software Detailed Design (SDD).