CHAPTER 3

In this chapter we are going to discuss the requirement, design of the system and analyse them using appropriate software models.

3.1 Methodology

The most suitable and appropriate software development methodology we found for our software development is prototype software methodology model, which will give us a clear view about our project and will help us achieve our goal.

3.2 The SDLC Prototype

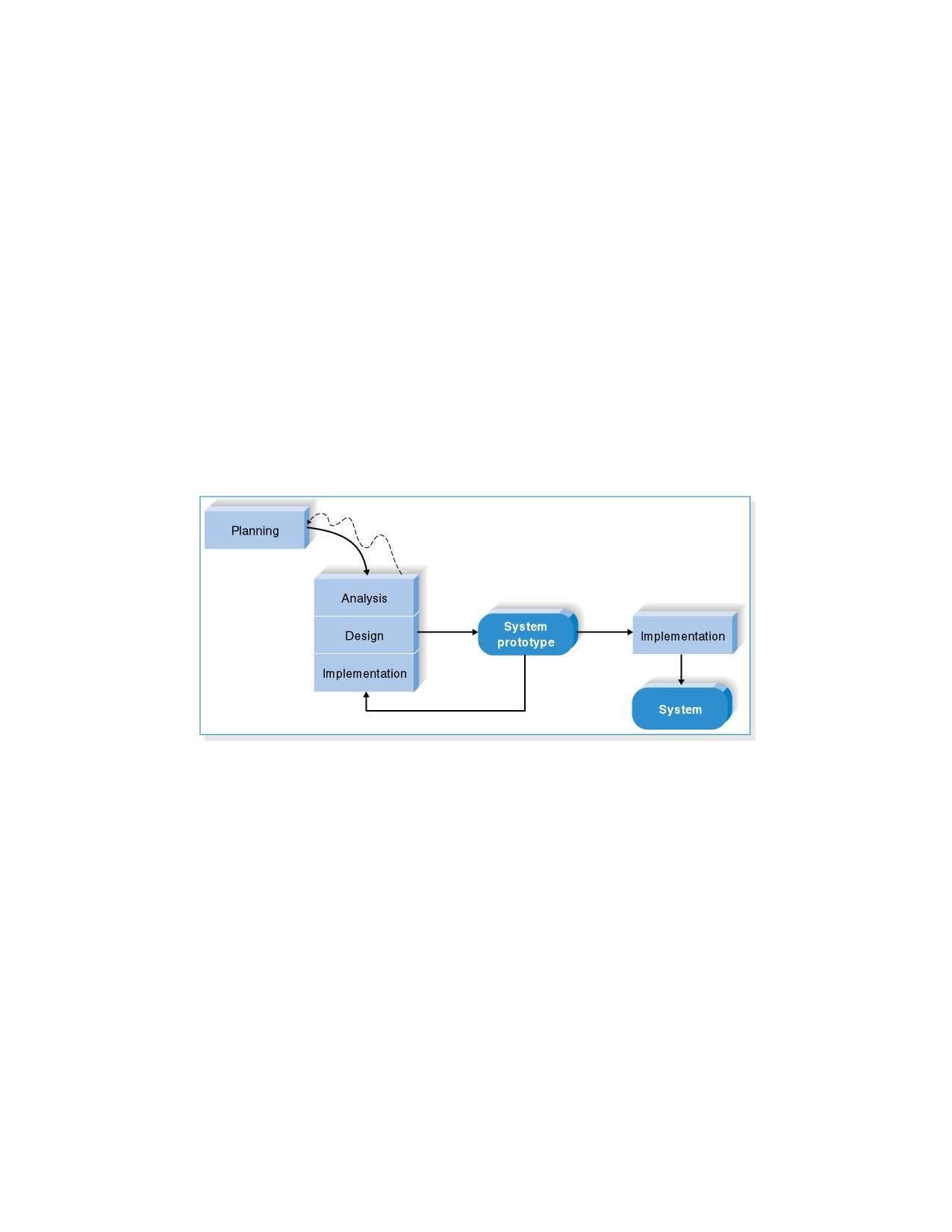
The prototype is an attractive idea for large and complicated systems for which there is no manual process or existing system to help determine the requirement.

The prototype model is a system development method in which creating a scaled-down or incomplete version of a system to demonstrate or test aspects of the system. This prototype is developed based on the currently known requirement. By using this prototype, the client gets an actual feel of the system, since the interaction with the prototype can enable the client to better understand the requirement of the desired system.

The prototype is usually not complete systems and many of the details are not built in the prototype. The goal is to provide the system with overall functionality.

3.2.1Reason to Choose Prototype Model

This approach is most appropriate for our project because all the requirement and goals of our project are very clear. Secondly it is easy to work with this prototype model because the system to be build, have a high amount of interaction with the end users. It might take a while for the system to be built that allows ease of use and need minimal training for the end users.



3.3 System Requirement

Requirement analysis is an important phase of the system development cycle which provides us all the specification of system in detail which is very essential to build the system and also provides us knowledge about the behaviour of the system. Collection of system requirements is a very critical point because the whole system is based on this knowledge and it also provides input to the next following stages of the system development life cycle. We can classify the requirements of the system in two categories as following:

* Functional Requirement
* Non Functional Requirement

3.3.1 Functional Requirement

Functional Requirements specifically defines functionalities of the system behaviour of the system and the goals to achieve it. Functions that describe the behaviour of the system are considered as behaviour requirements and will best show in the form of use cases. The importance and description of functional requirements are explained in the following table:

Functional Requirement

|  |  |  |
| --- | --- | --- |
| NO. | Functional Requirement | Description |
| 1 | Upload | To upload files unto the system |
| 2 | Download | To download file from the system |
| 3 | Search | To search out files,  . |
| 4 | Web Interface | Provides interaction between  user and database |

3.3.2 Non Functional requirement

Non-functional requirements are also knows as quality of a system. Hence, it provides us knowledge regarding the operations instead of behaviour or functionalities, contradicting with the functional requirement in this manner. Non-functional requirement are described in the system architecture helping us to achieve the quality goals and improves the functionalities of the system. Non-functional requirements are as following:

Non Functional Requirement

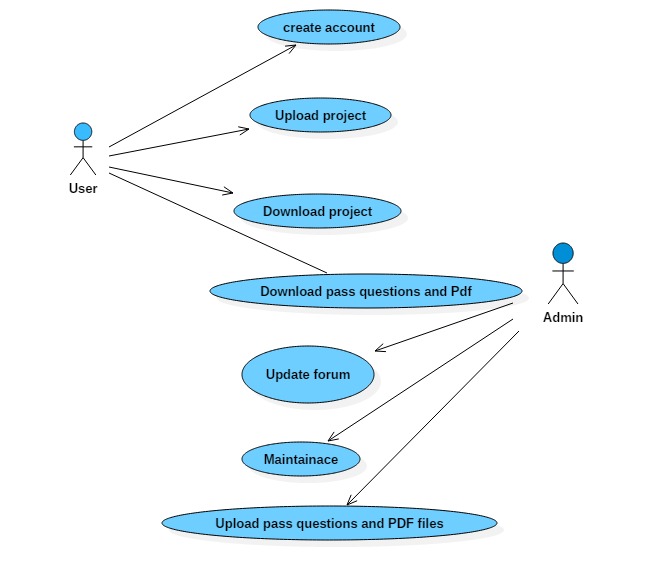
|  |  |  |
| --- | --- | --- |
| NO. | Non Functional Requirement | Description |
| 1 | Performance | Performance of system should be adequate and fast |
| 2 | User friendly environment | System interface should be simple and user friendly |
| 3 | Application scalability | System should be flexible |
| 4 | Application Maintain | Maintenance and documentation should be done thoroughly |
| 5 | Security | Access to the individual user account will be protected by a user login screen that requires a username and password. To further enhance the security of the system, all passwords will be stored in encrypted format. Also, each successful login attempt shall be recorded in a log file. |
| 6 | Platform independent | System should be able to work in any environment |

3.4 Analysis Model

To produce a model of the system which is correct, complete and consistent we need to construct the analysis model which focuses on structuring and formalizing the requirements of the system. Analysis model contains three models: functional, object and dynamic models. The functional model can be described by use case diagrams. Class diagrams describe the object model. Dynamic model can also be described in terms of sequence, state chart and activity diagrams. For the purpose of this project we have described the analysis model in terms of the functional model that is use case diagram.

**3.5.1 USE CASE DIAGRAM**.

Use cases of the system are identified to be **register student, lecturers (users) students to upload, download project**, **download** **past questions, and download PDF files.** The diagram depicted in Figure below shows the use case diagram of the system.



**3.5.2 ACTOR DESCRIPTION**

**NAME**: ADMIN

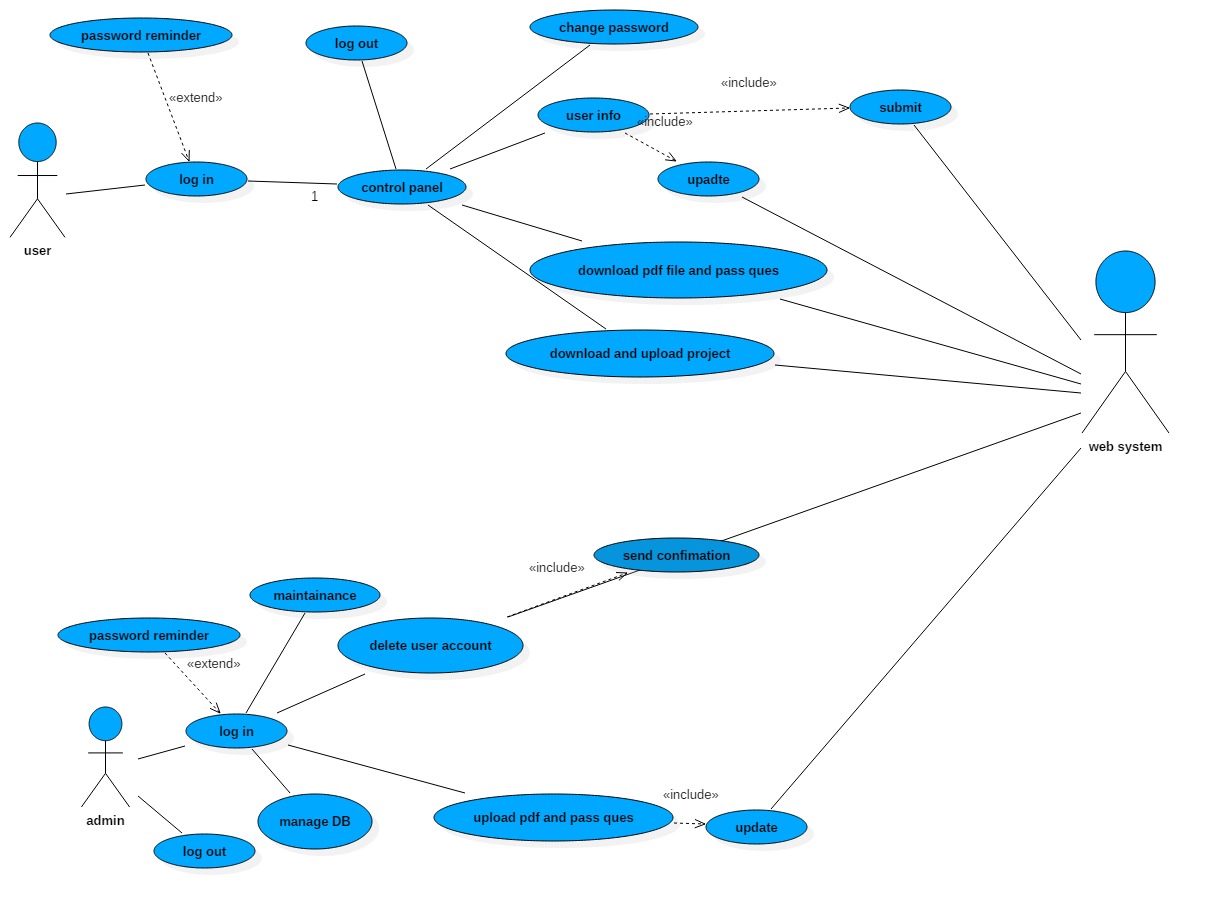
**DESCRIPTION**: The last actor is system administrator. Maintenance is one of his major responsibilities. He checks the security logs, handles the security issues, checks the application logs, handles the error massages of the application, handles the database errors and web server problems etc. The also admin keeps records of the users, update the forum, upload lectures slide, course outline and pass questions and also view their information

**NAME**: USERS

**DESCRIPTION**: The users (lecturers, student) are able to create an account, share ideas on the forum. User (student) will be able to upload and download project work, download past questions, course outline and lectures PDF.

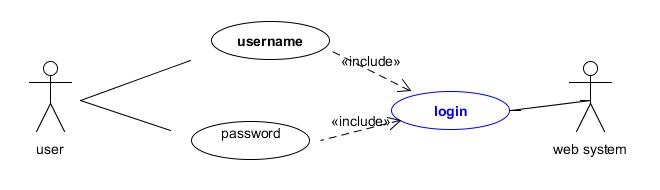
3.6.0 SYSTEM USE CASE IN DETAIL

This figure is showing the use cases of the system in detail and will be Discussed further each use case individually.



**USE CASE DIAGRAM IN DETAIL**

**3.6.1** USE CASE SCENARIO OF LOG IN

**USE CASE DIAGRAM LOGIN**

Scenario

Actors (administrator, users) want to enter the system to perform the specific operations.

Basic Flow

When any of the actors want to log-in, the system will ask the user name and password. On entering the log-in parameters the system will authenticate them. After validation it will authorized the actor to enter.

Alternative Flow

An error is generated if the log-in parameters are not valid.

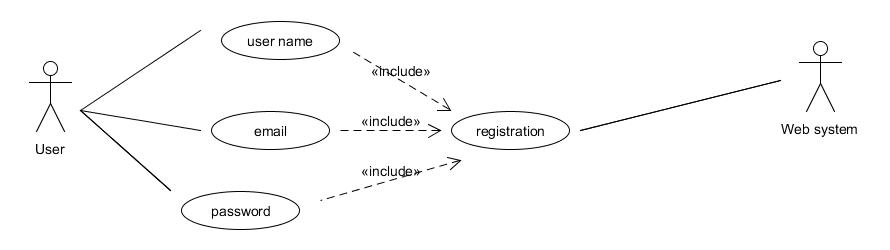
Pre-Condition

User must be registered into the system.

Post-Condition

The user will be entered into the system successfully.

3.6.2 USE CASE SCENARIO OF REGISTRATION



**Use case diagram of user registration**

Scenario

Registration is required to enter and use the system. After registration the

Actors (users) are authorized only to log-in into the system.

Basic Flow

In case of user registration the user will enter the required

Information about the user including the user name, password, email and create the new user accounts.

Alternative Flow

In case of creating new user account, if the user name already exists or if the entered information is not in a valid format, an error message is generated.

Pre-Condition

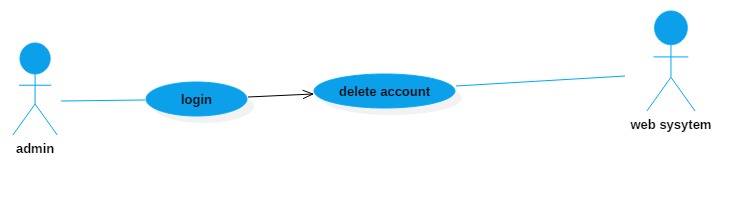
In case of new user accounts, no entries should be existing in the system for that user.

Post-Condition

In case of new user entries system will create a new user in the database.

System will update the existing user accounts in the database, in case of editing user accounts.

3.6.3 USE CASE SCENARIO OF DELETE ACCOUNT



**USE CASE DIAGRAM OF USER DELETION**

Scenario

Administrator wants to delete an existing user account.

Basic Flow

Administrator can delete the existing user accounts after logging in when

Ever it is necessary.

Alternative Flow

System will generate a message to confirm deletion of existing user

Account or give an alternative to cancel the process without deletion.

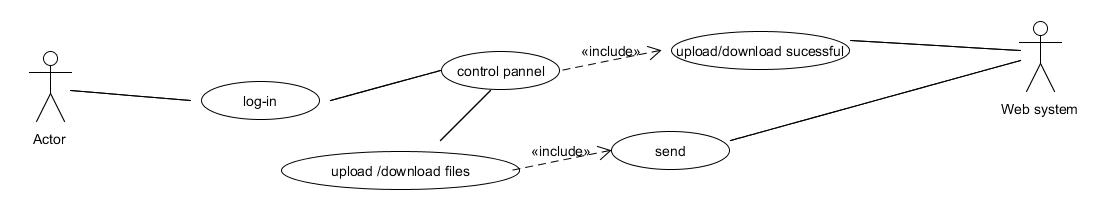
Pre-Condition

User account must exist.

Post-Condition

System will delete user account and update the database.

3.6.4 USE CASE SCENARIO FOR UPLOADING AND DOWNLOAD FILES



**USE CASE DIADGRAM FOR UPLOADING AND DOWNLOADING FILES**

Scenario

Actor (admin, users) wants to upload or download file unto the system

Basic Flow

Actor (user, administrator) will select the type of files to uploaded or downloaded and submit to server. After input validation, server will then display it on the web system.

Alternative Flow

* System will generate error message of wrong file (upload unsuccessful)
* System will generate error message of download failed

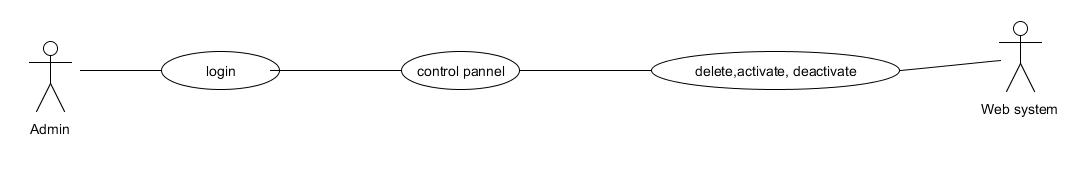
Pre-Condition

Files should be stored in the data base

Post-Condition

System should allow upload and download

3.6.5 Use Case Scenario for Activation/Inactivation/Deleting Accounts



**USE CASE DIAGRAM FOR ACTIVATION /IN ACTIVATE /DELETE ACCOUNTS**

Scenario

In case, if an actor (users) temporarily or permanently quits, graduate from the school or re-joins, then the authorized users will change his status in the system.

Basic Flow

After logging-in, the user will search for the actors (users) from the

Database by entering his ID and performing the following functions:

* The user will activate the account, if the actor (user) comes to re-join the Organization after some time.
* In-Activate the account if user wants to quite temporarily.
* Delete the user account if user quits permanently.

Alternative Flow

* The system will generate an error if the entered ID is not valid and the

User will remain on the same page.

* Pre-Condition

User must be logged-in and authorized to perform task. Account of an

User must exist.

* Post-Condition

If the basic flow is performed successfully the status a user account will be changed and updated in the database.