

PROJECT PROPOSAL FOR LEARNING **MANAGEMENT WEB-APP**

| Name | Roll No. |
|---------------|-------------------------|
| Aparna Ullas | <i>AM.EN.U4CSE15011</i> |
| Namratha PS | <i>AM.EN.U4CSE15040</i> |
| Bharath N Rao | <i>AM.EN.U4CSE15217</i> |

● PROJECT SUMMARY

○ Project Overview

A Lecture Notes web-app for the courses offered by Amrita Vishwa Vidyapeetham.

○ Project Scope

Primary scope of the project is to make the lecture notes of different courses, available for the faculty as well as the students.

○ Development Process

The core and a basic web-app will be developed in this sprint, with following characteristics:

- A basic web app which displays lecture notes on some selected courses.
- The courses will be categorized by departments, which offer them.
- Provision to upload lecture notes.
- Provision to add new courses for the administrator.
- Minimal GUI for the web-app.

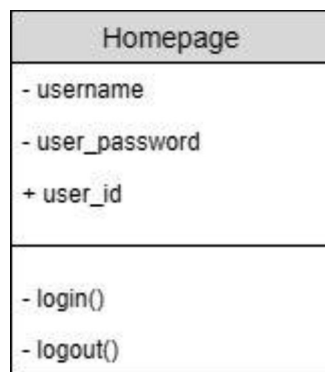
Class Diagram

In this documentation we present the static form of the system. This includes different objects, their functions and relationships. There are 4 classes namely Homepage, User Homepage, Admin Homepage and Document.

Classes

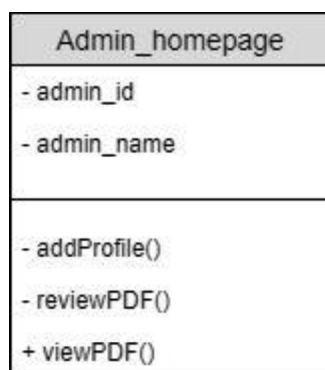
Homepage:

This class is responsible for authentication of users and admins. It has methods login and logout for the same. Username, Password and User ID are the attributes.



Admin homepage:

This class is static view of admin work field. It has attributes admin ID and admin name. The methods defined are addProfile(),reviewPDF() and viewPDF(). The admin can add new users through addProfile(), review the uploaded PDF's using reviewPDF() and he can view any PDF through viewPDF().



User homepage:

This is user homepage class where attributes are user_id and user_name. The methods include viewPDF(),search Dept() and searchCourse() to view,search department wise and to search course wise respectively.

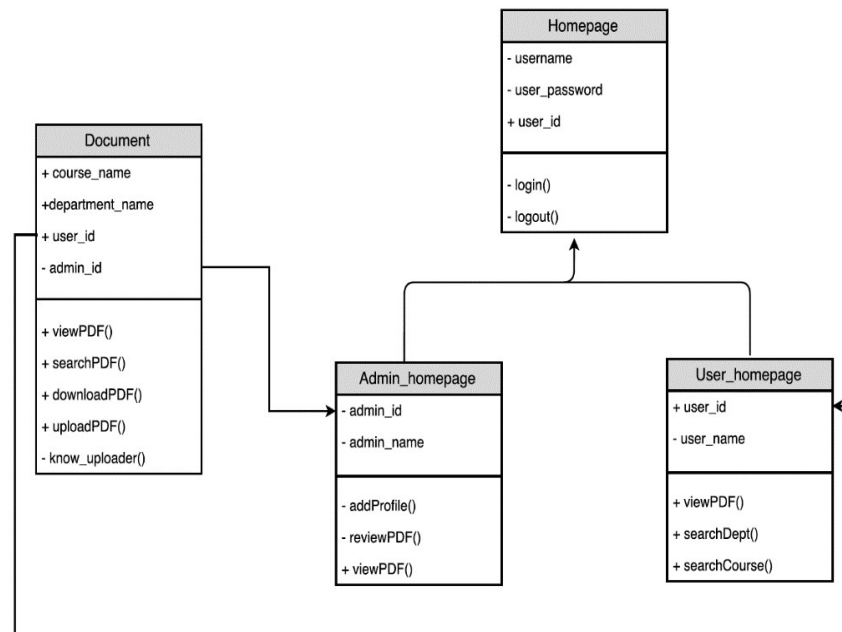
| User_homepage |
|---|
| + user_id - user_name |
| + viewPDF() + searchDept() + searchCourse() |

Document:

The document class is in a way the heart of this project. This class has all the functionalities that we implement in this Learning management web app. This class has methods viewPDF(),searchPDF(),downloadPDF(),uploadPDF(),know_uploader() . The uploadPDF() is to upload a desired PDF and know_uploader() is to keep track of who is uploading the PDF.

| Document |
|---|
| + course_name +department_name + user_id - admin_id |
| + viewPDF() + searchPDF() + downloadPDF() + uploadPDF() - know_uploader() |

The following is the complete class diagram for Lets Learn-Lecture Notes web app with relations among classes :



State Diagram

The state diagram depicts states and transitions. It illustrates how an object moves through various states within its lifetime. States are represented with rectangles with round corners that have a labelled state name. Transitions are marked with arrows. The state diagram of the system starts with a dark circle, which indicates the starting state and ends with a bordered dark circle that denotes the final state.

The proposed web app has following states:

- *Start state*
- *Main page logged out*
- *Web app logged in*
- *Error page*
- *Search result*
- *Download complete*
- *PDF Preview*
- *PDF Uploaded*
- *PDF Review*
- *Add user*
- *End state*

The state diagram for the web app system is given below:

Use Case:

The web app has two actors, any user who logs in to the web-app and administrator. Authentication of the users is done by the administrator by providing unique login id. And in turn the administrator also has his/her id.

Both the actors can search for Lecture notes, department wise and course wise. Also, they can add pdf of Lecture notes and also download them given that their authentication is complete. However, Administrator has some extended functionalities. He/ She can add courses, add users and review uploaded Lecture note.

Use case: *A web app for Lecture Notes*

Primary actors: *Primary Users, Administrator*

Objective: *To create a web app which allows primary users to preview lecture notes, upload and download them.*

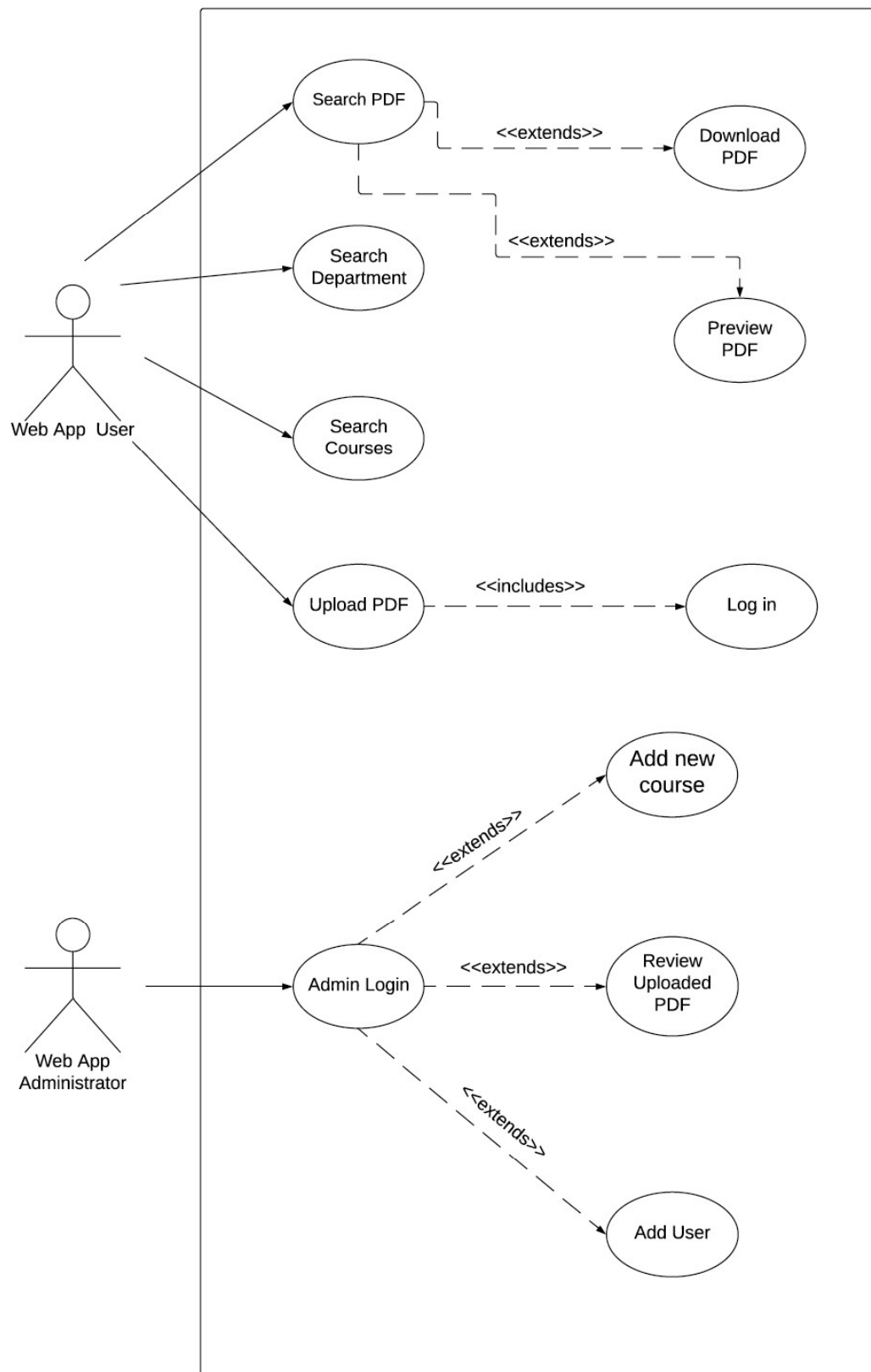
Scenario:

Primary User:

- Login*
- Search PDF*
- Preview PDF*
- Download PDF*
- Search Departments*
- Search Courses*
- Upload PDF*

Administrator:

- Login*
- Add new course*
- Add new user*
- Review uploaded PDF's*



Test Cases

In the ***Test case*** documentation it specifies input values, expected output and the preconditions for executing the test. Test case document is also a part of test deliverables, by reading test case document stakeholders get an idea about the quality of test cases written and the effectiveness of those test cases. Stakeholders can also provide inputs about the current set of test cases as well as suggest some more missing test cases.

| Test Case Id | Test Scenario | Steps | Expected Result |
|--------------|------------------------------|--|---|
| 1 | Successful login | <ul style="list-style-type: none">• Enter the user id in the username field.• Enter the password.• Click <i>Login</i>. | The user is logged in successfully. |
| | Unsuccessful login | <ul style="list-style-type: none">• Enter the user id in the username field.• Enter the password.• Click <i>Login</i>. | A display box shows "Invalid username or password" . |
| 2 | Successful search of a pdf | <ul style="list-style-type: none">• Type the name of any course in the search by course field.• Click <i>Search</i> | A new page which shows pdf notes which are available in the web app. |
| | Unsuccessful search of a pdf | <ul style="list-style-type: none">• Type the name of any course in the search by course field.• Click <i>Search</i> | A display box that shows <i>"Couldn't find the requested lecture note pdf"</i> |

| | | | |
|---|---|---|--|
| 3 | Successfully searching pdfs of the courses that certain department offers | <ul style="list-style-type: none"> • Select any department from the dropdown box. • Click <i>Search</i> | A new page which shows all the pdfs of the courses that a department offers. |
| 4 | Preview of a pdf | <ul style="list-style-type: none"> • Search for the pdf (course wise or department wise) • Click <i>Preview</i> | A new page which shows the lecture note. |
| 5 | Downloading a pdf | <ul style="list-style-type: none"> • Search for the pdf (course wise or department wise) • Click <i>Preview</i> • Click <i>Download</i> | A display box which shows <i>"Download Successful"</i> |
| 6 | Uploading a pdf | <ul style="list-style-type: none"> • Click on <i>"Upload PDF"</i> • Select path of the pdf. • Click <i>"Upload"</i> | A display box which shows <i>"Upload Successful"</i> |
| 7 | Adding a new user (<i>Admin</i>) | <ul style="list-style-type: none"> • Click on <i>"Add user"</i> • Type the user data in respective fields. • Click <i>"Authenticate"</i> | A display box which shows <i>"New user added"</i> . Furthermore, this can be checked by logging into the web app with new user account. |
| 8 | Validating an uploaded pdf (<i>Admin</i>) | <ul style="list-style-type: none"> • Click on <i>"Review pdf"</i> • Click <i>"Validate"</i> | While clicking Review pdf, a list of user uploaded pdfs shows up. The admin has to validate it so that it is shown in the web app. |
| | Revoking an uploaded pdf (<i>Admin</i>) | <ul style="list-style-type: none"> • Click on <i>"Review pdf"</i> • Click <i>"Revoke"</i> | Admin can revoke an uploaded pdf if it is not up to the standards. |
| 9 | Logging out of the web app | <ul style="list-style-type: none"> • Click on <i>"Log out"</i> | A new page shows up which displays <i>"Successfully logged out"</i> . Alongside, link for logging in is also displayed. |