Core Lecture: Software Engineering 2013

Contract

Printerface 2.0 (Backend)

Customer:
Clemens Hammacher
03.02.2013

Tutor:
David Poetzsch-Heffter

Team:
Dominic Seyler
Equar Hishe Gebre
Jacob Ambat
Maximilian Reinert
Md Salahuddin Pasha
Sugavanesh Sadasivam Nagarathinam
Taner Arslan

Overview:

The printerface system will enable students and staff to send print jobs to a central backend server, which distributes the job queue to terminals which are located next to a printer. After sending a print job, the user will be able to select his/her job from a job list that will be shown in the display of the frontend terminal. To start the printing process, the user will authenticate himself/herself by placing his/her UdS card in front of a card reader (connected to the frontend terminal). Besides sending a print job directly from a user's computer, the user will be able to browse a list of course material, either on the frontend terminal or via an Internet browser (may have) on his/her computer. The course material will be provided by university staff and will be taken care of by the system's administrator.

The system is going to consist of a central linux server which can basically be divided into two components. Component one is the actual print server which receives print jobs from members of the University (print server). The idea is to effectively manage the printing jobs from students as well as from staff. The printing server will be implemented using Common Unix Printing System (CUPS). The print jobs given by the user will be queued in the database of the backend server, which directs the list of print jobs to the frontend terminals. After the user has made a selection at the terminal, the backend server receives the job-id and the user-id and sends the corresponding print job to the front end terminal.

The second component contains a database of official course material for printing (document library). The document library stores the files, uploaded by the document library admin or staff members, that can be printed by the students. The admin can decide whether the page quota of the student will be altered. At the frontend terminal, or in a web browser (may have), the user can browse the documents in the library and start the printing by selecting the desired document.

Must-have Features:

- **CUPS-Backend Server**

    - *Communication with frontend Team (Communication Protocol):* The communication with the frontend terminals will be realized using a reliable protocol. Using the protocol the backend server should be able to distribute a job list to every frontend terminal, receive a request for a print job from the terminal, send print data and send notifications (quota or authentication errors) to the terminal. [#1]
    - *Authentication via UdS Card*: After a print job is selected and the UdS card is placed in front of the card reader, the University's central authentication server will send a notification to the backend server which consists of the identifier of the card reader terminal, a unique user-id, and a permission bit (user is eligible for printing). The backend server has an internal mapping between card reader terminal and printerface frontend terminal to determine the frontend terminal that belongs to a certain card reader. [#2]
    - *Queuing and storing print jobs in database:* Print jobs by the user will be queued and stored in the database of the system. Each entry in the database will represent a job and will contain a job-id, the number of pages, user-id and the username for easy identification. Furthermore, the entry will contain the actual print data of the document. [#3]
    - *Web based Admin Interface:* The interface that is part of CUPS will be used as the admin interface for the CUPS-backend server. For all platforms CUPS web based interface runs on port 631 providing a simple class, job and printer administration interface allowing the user to modify delete, classify, control classes jobs and printers. [#4]
    - *User database:* The user database library will be designed for the groups “staff”, “students” and “admin” respectively. It will basically consist of entries with the user-id (unique identifier from UdS card), the user group, ther UdS user name and the remaining page quota. [#5]
    - *Managing the page quota per user:* The print quota will be stored, for each user group, in the database. If a print job by a new user reaches the backend, the user will be placed in a default group, which has a certain page quota. If a certain user belongs to another group (e.g. staff), the user account can be manually placed in the corresponding group. [#6]
    - *Log*: Print jobs have to be logged in the system’s database for reference, so they can be used for statistical analysis. Monthly logs will be created that state the number of pages printed by each user. [#7]

    - *Error handling:* If print jobs fails, the frontend terminal will send an error notification which will have the effect that the user's page quota will not be altered. [#8]

- **Document Library Server**

    - *Web based admin interface:* Using a web-based administration interface, a privileged person can add users or other administrators to the system. One of the Library Server's admins can create folders and assign privileged users to each folder. The users can be assigned to one or multiple folders where they will be able to upload and manage their documents. [#9]
    - *Document database*: All documents will be stored with additional metadata (e.g. number of pages) in a database. The documents will be arranged in an arbitrary folder structure which can be managed by the assigned user(s) or one of the admins. [#10]

May-have Features:

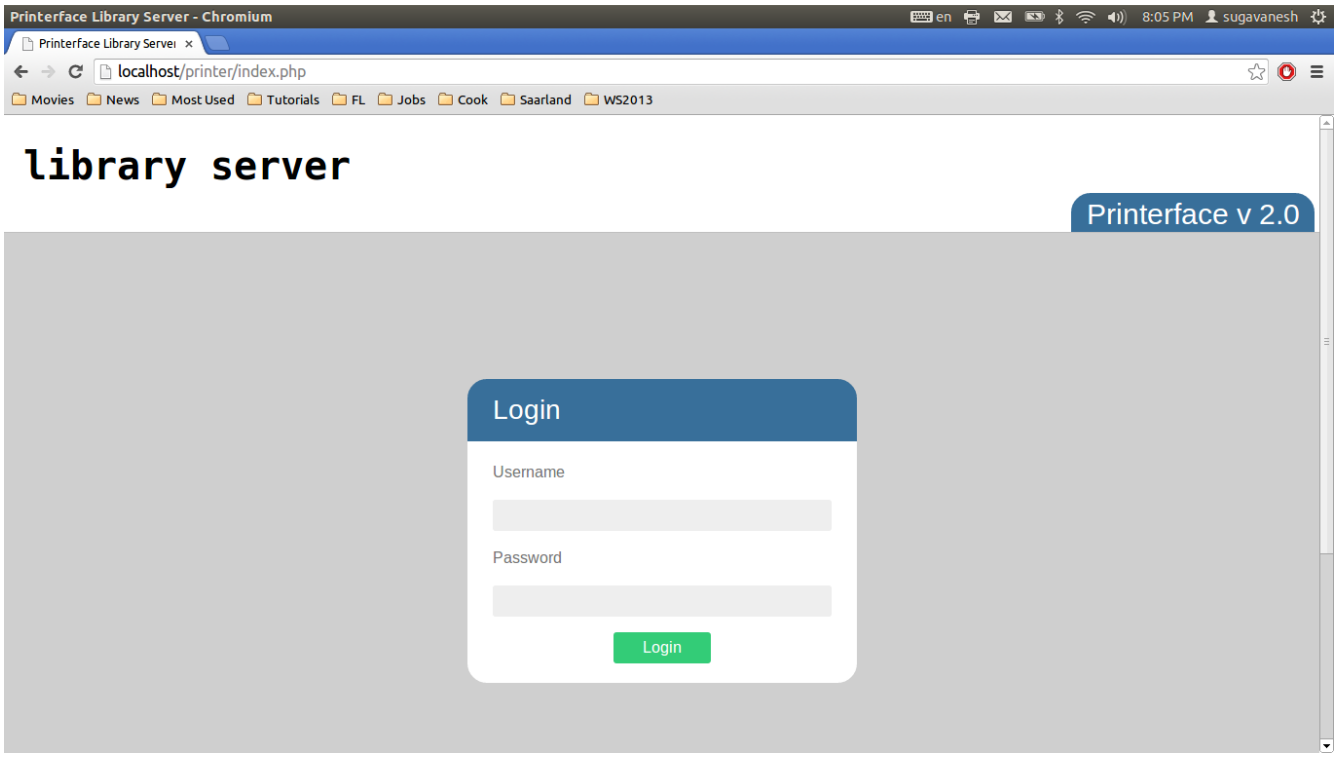
- *Browse Document Library*: The students will be able to browse and print documents from the document library using their Internet browser. [#11]
- *Time Management:* Users can postpone the print job to get their documents printed at a given time. [#12]
- *Printer problems:* When a printer is out of paper, the cartridge has to be replaced or other physical printer errors arise, the frontend terminal will notify the backend server, which will notify the Admin by email. [#13]
- *Assign quota to a certain document:* The admin of the document library server will also be able to assign quota to a document in the document library server. [#14]

Must-not-have Features:

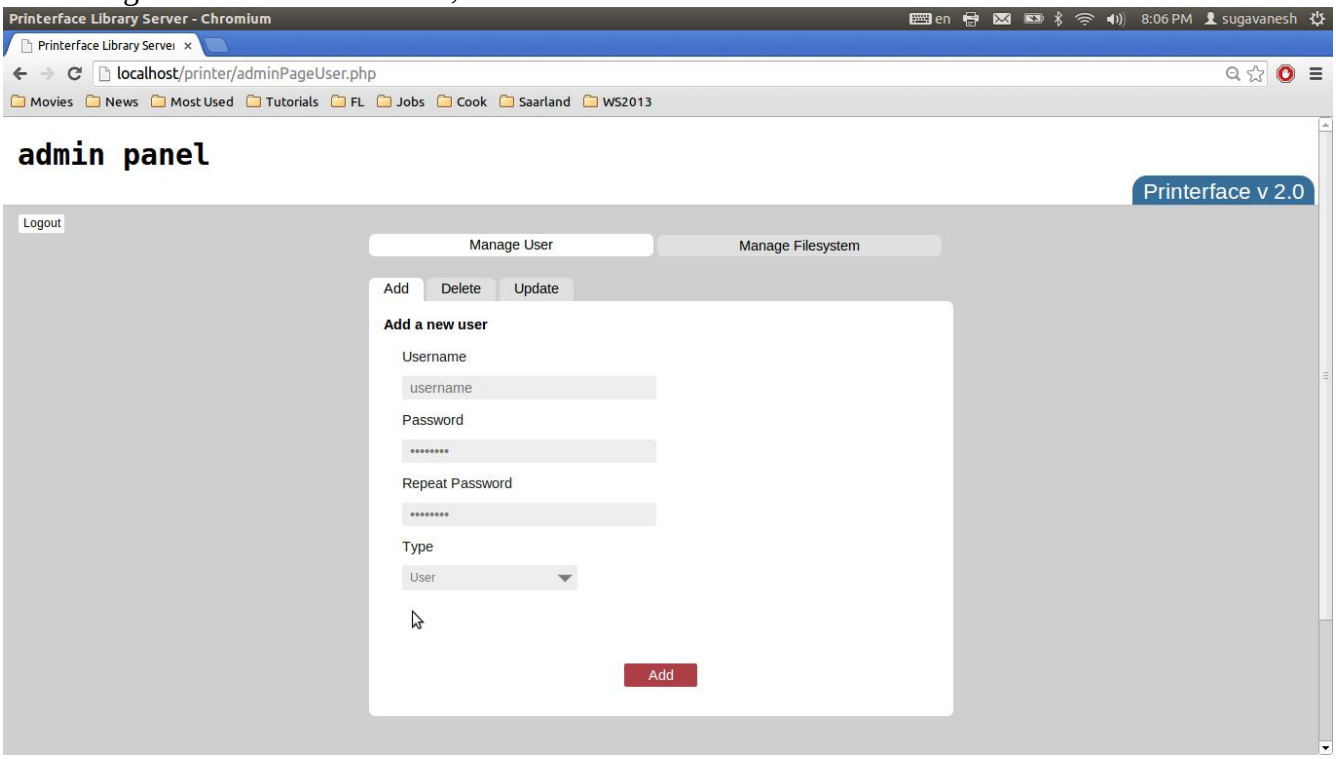
- *Language Support:* The system should not support other than English language.
- *Network Issues:* Users can order print jobs from outside of the Saarland University Network.
- *Statistical Analysis*: The statistical analysis procedure of print jobs, as mentioned above, is not part of this project.

DOCUMENT LIBRARY

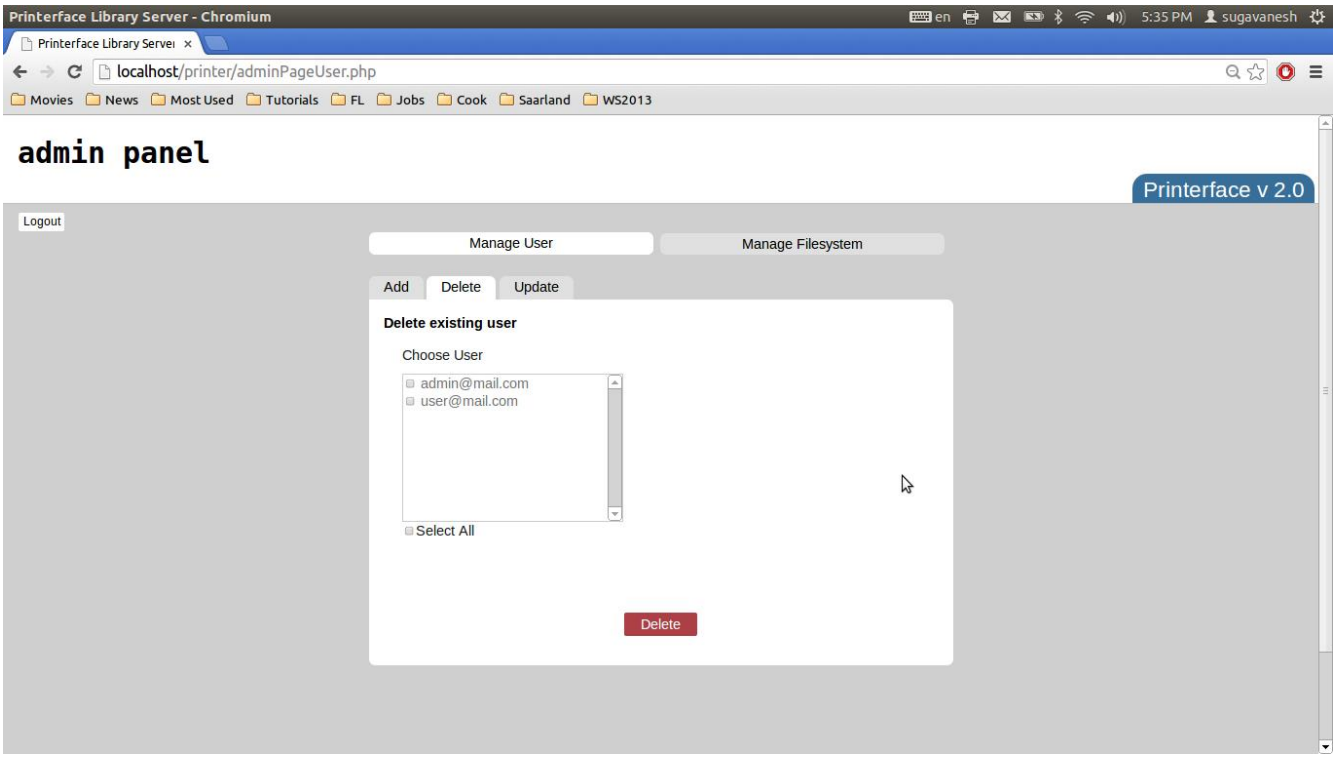
1.Login panel for Library server for both admin and user.



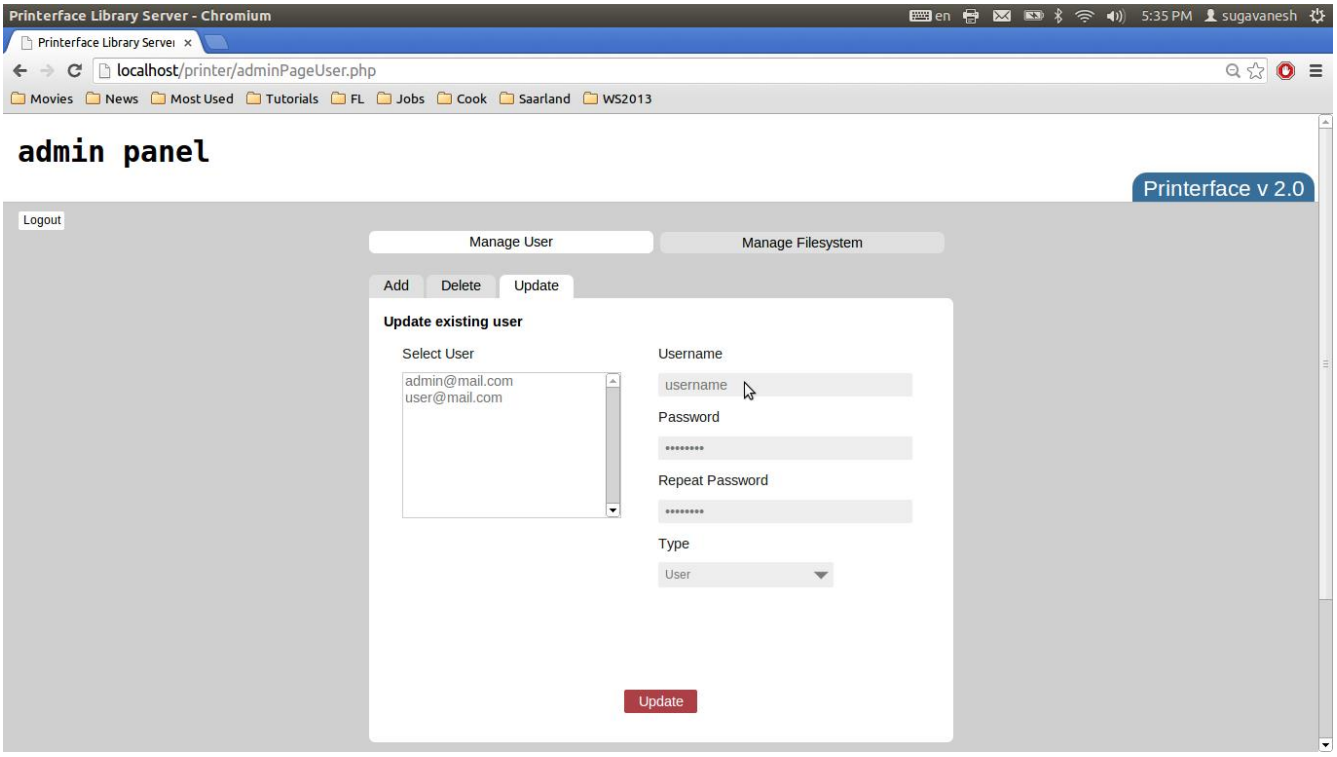
2. In the admin panel, we have to sections. One to manage user and the other to Manage Filesystem. In the Manage User section ADD tab , we can add new admin and new user credentials.



In the Delete tab, we have the choose User window in which all the added users and admin appear. We can choose from this list the id to be deleted.



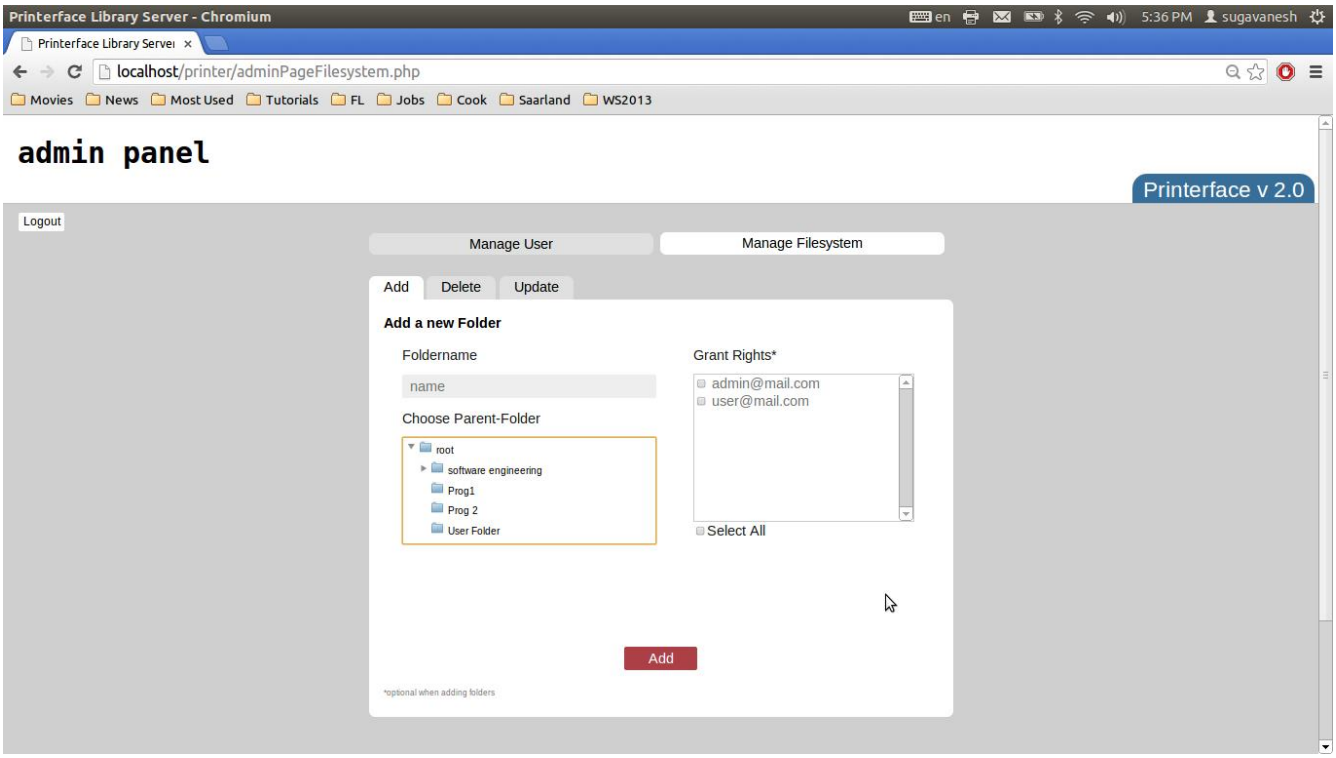
In the update tab , we have the select user window in which we can select the user to change his/her credentials



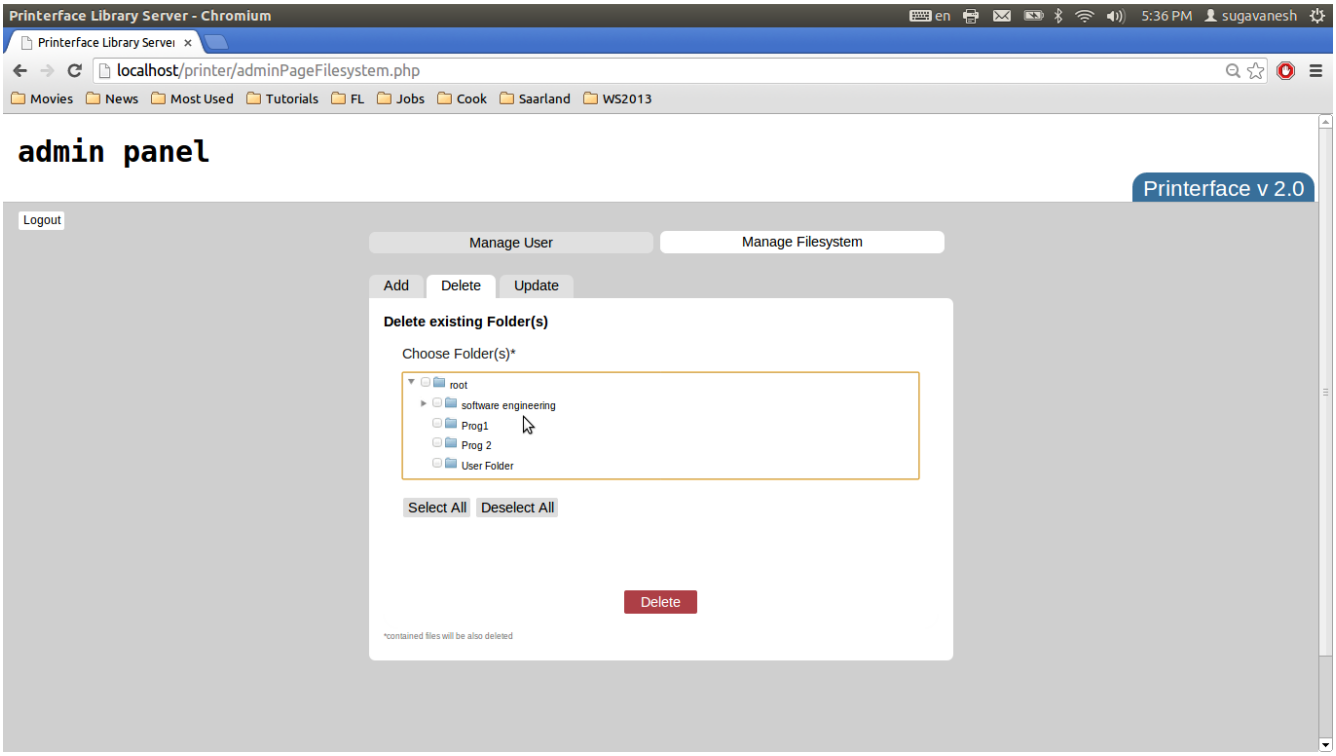
The next section in the admin panel is the Manage Filesystem, where again we have the ADD tab. Here

DOCUMENT LIBRARY

we have the file structure listing all the folders in a hierarchy. We can select the corresponding folder and add the sub-folders.

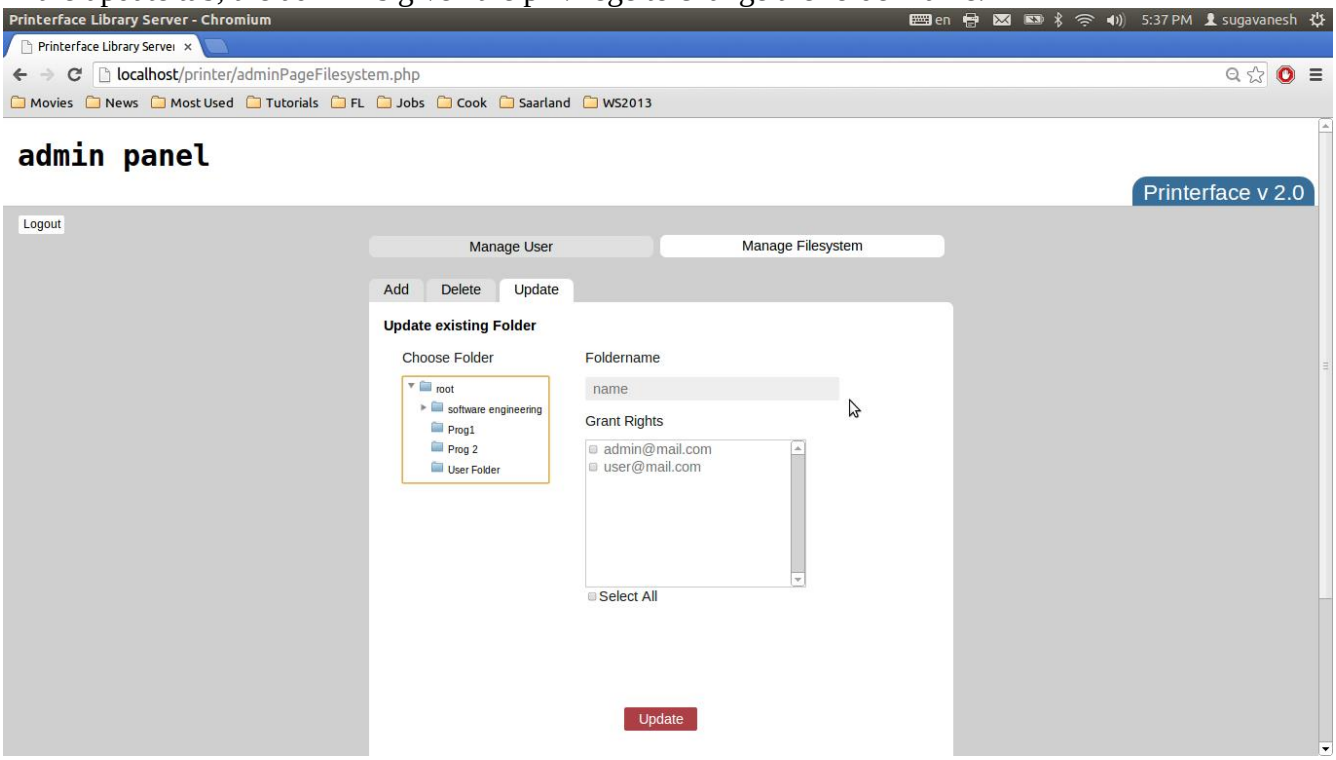


In the delete tab, we can delete the folders/sub-folders from the list.



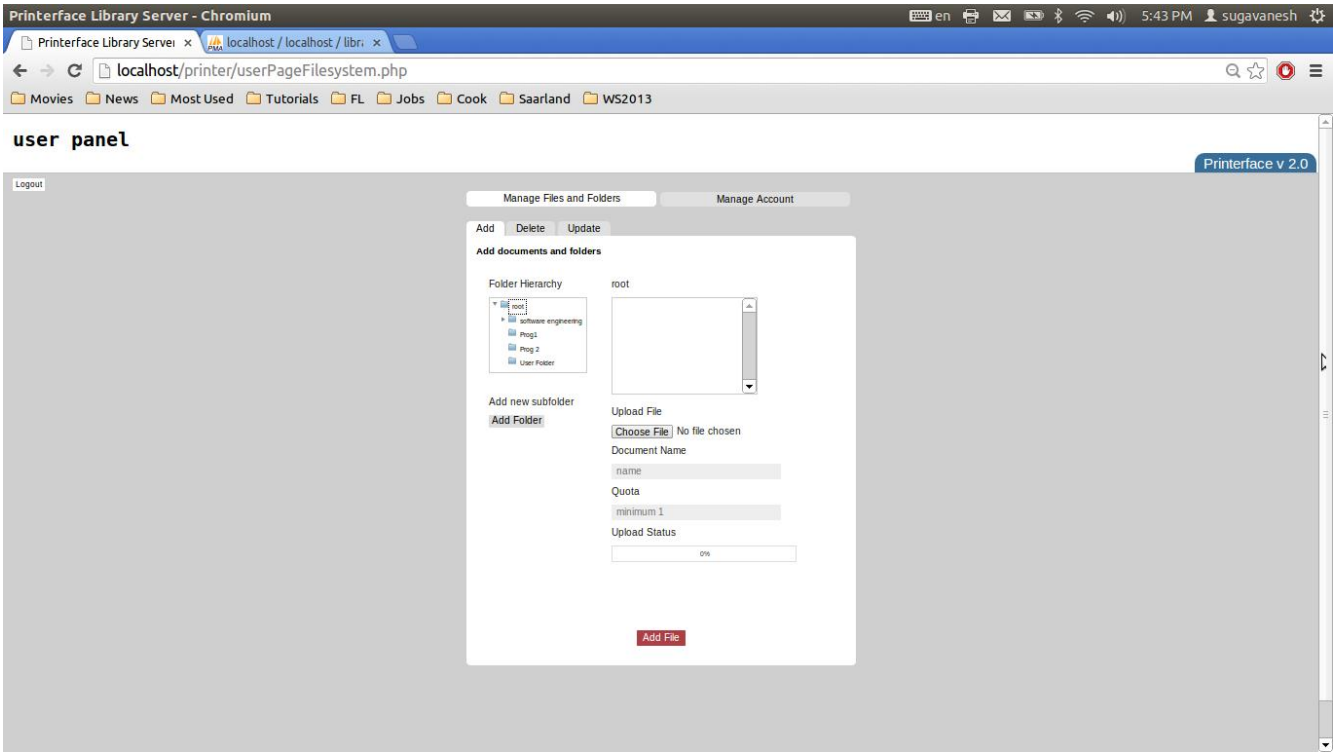
DOCUMENT LIBRARY

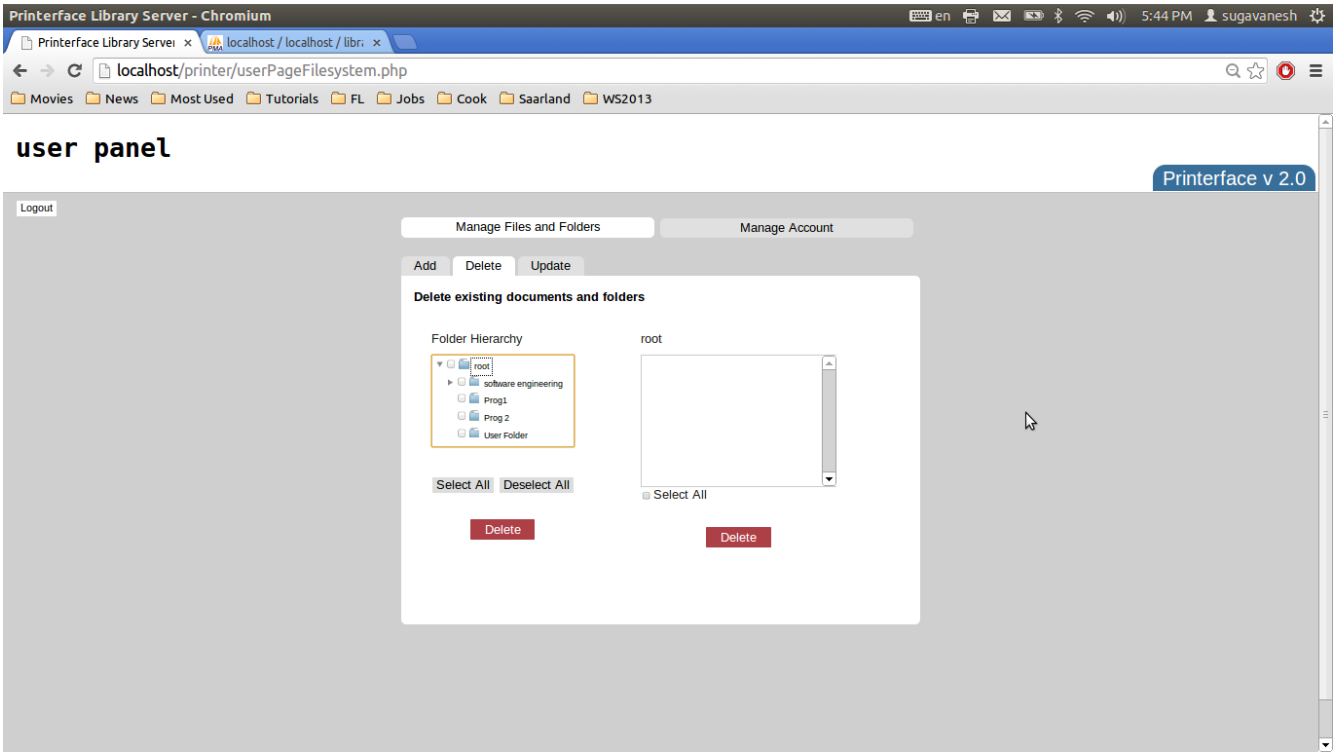
In the update tab, the admin is given the privilege to change the folder name.



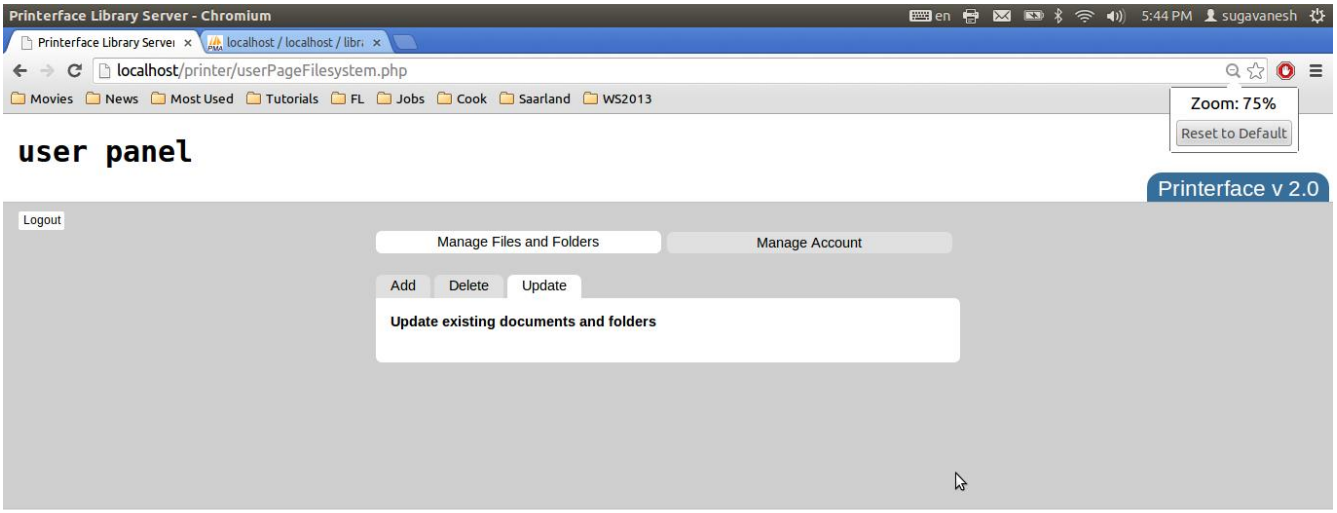
The user logs into the system and

DOCUMENT LIBRARY





DOCUMENT LIBRARY



FAQ User