Student Grading System Project

Project Document

Name: Purushothama Rajanna
SUID: 770079902
email: prajanna@syr.edu

Table of Contents

Product Report -

Introduction:

A simple system developed using Java to monitor and track progress of students throughout the academic year.

Brief Description:

The main objective of the project is to have one place to track all student information, courses taken and at the same time add the scores of the respective courses. The system can be used to track the progress of all the students at one place. This simple tool can be useful for professors, students, course coordinators and management to track the academic progress. The system is a web-based application developed using Netbeans IDE, Java programming and MySQL for database management. The project has been completed using Agile methodology.

How to run:

Tools Required –

1. Netbeans IDE
2. XAMPP server
3. MySQL

Instructions:

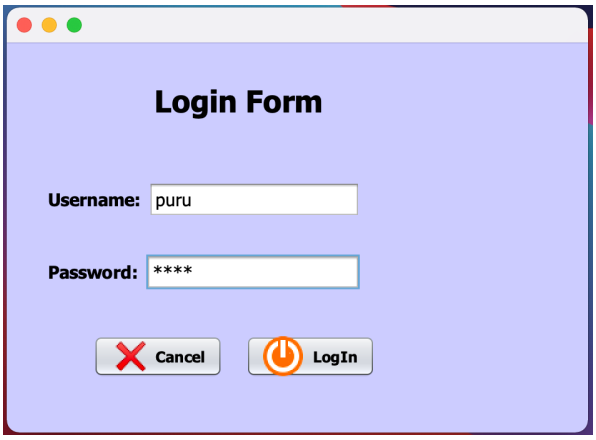
1. Once all the required tools are downloaded and installed, download the source in code uploaded in the .zip format. Extract the files.
2. Open Netbeans IDE and open the file from the option on top right corner. Open the directory in while the source code is present.
3. Open XAMPP server control manager. Make sure MySQL database and Apache web server is running. If not start these processes.
4. Run the stdmgdb.sql script in the myphpadmin SQL command line. This should generate all the required tables.
5. Build and run the code in Netbeans IDE.

6. For login credentials use – Username : puru
   Password : test123
   or write this in the SQL commandline and get all user logins - “select * from user “

Requirements–

Functional System Requirements:

UR-1. Each user will have an individual unique login access created by the Administrator. With the use of this unique ID and password, a user shall gain access to the portal. Login page is shown in below figure.



UR-2. Administrator will have root access to the portal. Admin can add, edit, and delete users.

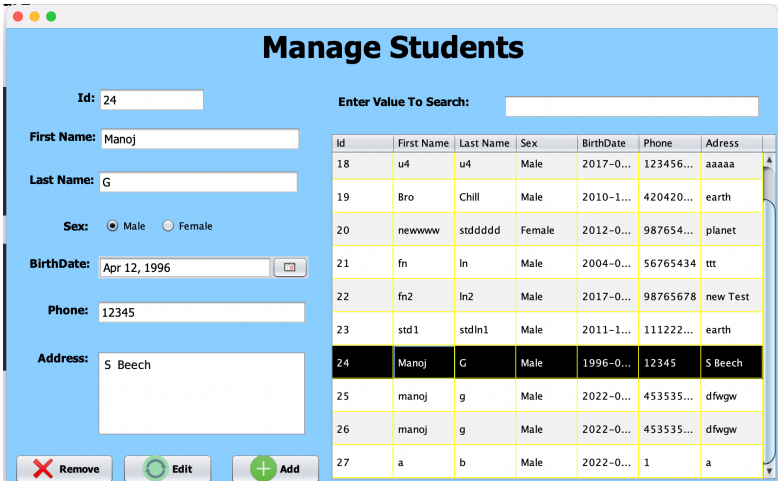
UR-3. User shall login using his unique login access provided by the admin. When user logins successfully the main page is shown. If login credentials are entered wrong in the login form, an error message is shown.

UR-4. When the user lands in the main page, he will be shown a welcome page with the total number of courses and total number of students enrolled.

UR-5. Admin also sees options on top left – Student, course, and Score.

UR-6. On click of student, add and manage options are seen. User gets add students using the add button. A new page is shown on click of add where user can enter all the details of the student such as first/last name, sex, birthday, phone no., and address. All these fields are mandatory. Add page is shown in below screenshot.

UR-7. Manage option views the details of all the students added to the list. User can click on individual student and edit the previously filled information. On click of any student, the input boxes on the left gets auto populated. Details can be edited here or in the database list directly on the right. Refer below screenshot.

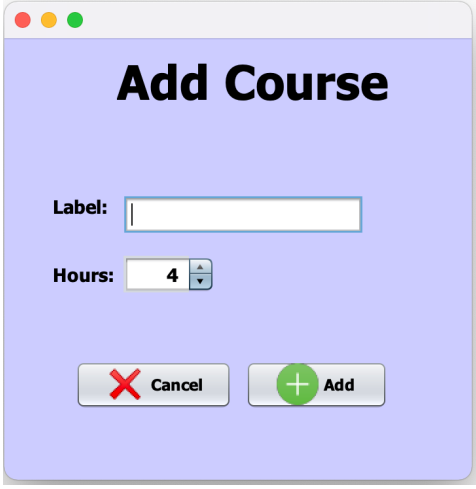


UR-8. In the manage tab, user also has the access to add a new student from this screen. It simply shows the add screen (UR-6) on click of add button. User has the option to delete any existing student from the list on click of delete button. Cancel button closes the tab.

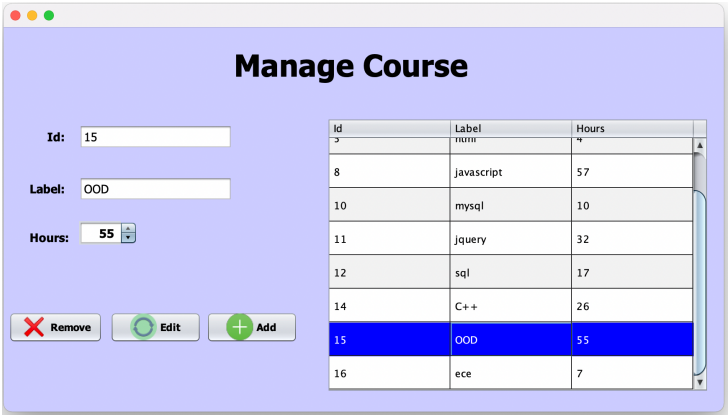
UR-9. Manage tab has a search bar on top. Any information regarding the student can be searched through key words and the matching details will be shown in the table.

UR-10. Welcome screen also has a course button on top left. On click of course, add, and manage options are shown.

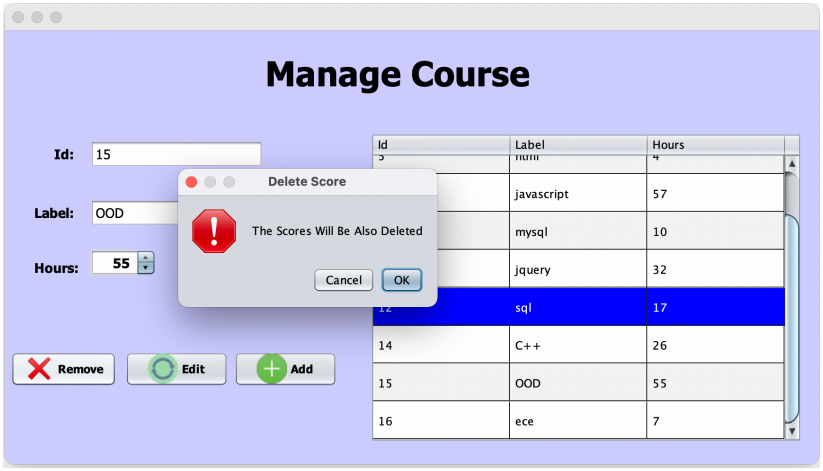
UR-11. On click of add, a new window pops up where user can add the course label and minimum number of hours required for the course. A unique ID is auto assigned to the course once its added.



UR-11. On click of manage course, a new window pops up where all the existing courses are seen on the right. User can view and select any course they would like to edit. By selecting any data, the columns on the left get auto populated. Any of these columns can be edited/deleted.

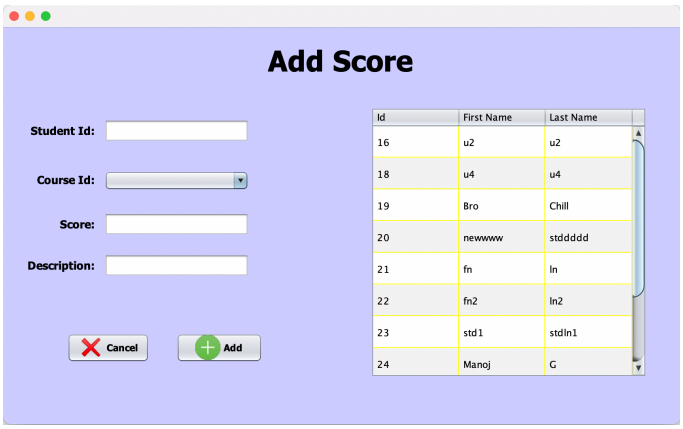


UR-12. If user decides to delete any course, this will also delete the respective scores added for the course.



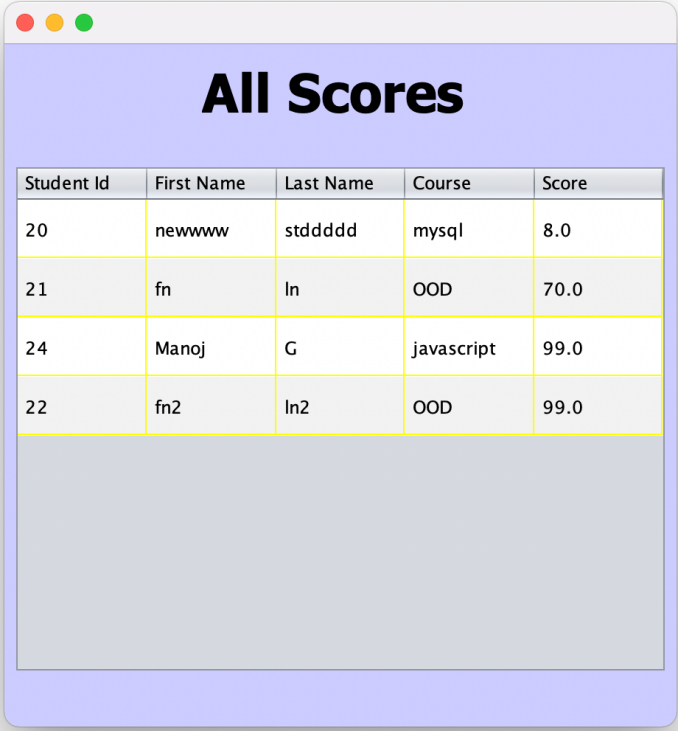
UR-13. The score tab on top left manages all the scored obtained by the students. The score tab has add, edit/delete, and show score options. User can click on any of these and make necessary changes.

UR-14. The add option in score tab opens a new pop up where you can select the students from the list. Once you click on any student, their student ID gets auto populated in the left. All the courses are shown in the drop down. User can select any course and can add score and description.



UR-15. The edit/delete option in score tab opens a new pop-up window where user can see the list of students. Any student selected will auto populate their respective details in the left. User can then decide whether to edit/delete their scores.

UR-16. The scores tab also has a view scores option. On click on view score, a new window pops up which will show all the existing score data added.



Non-Functional System Requirements:

UR-1. The system shall be available to anyone with internet access at any time.

UR-2. The system shall store the user's information securely.

UR-3. The system shall not share any personal information of the users to other a third-party organization not involved in business transaction.

UR-4. The system shall be simple enough that it requires no formal training for using it.

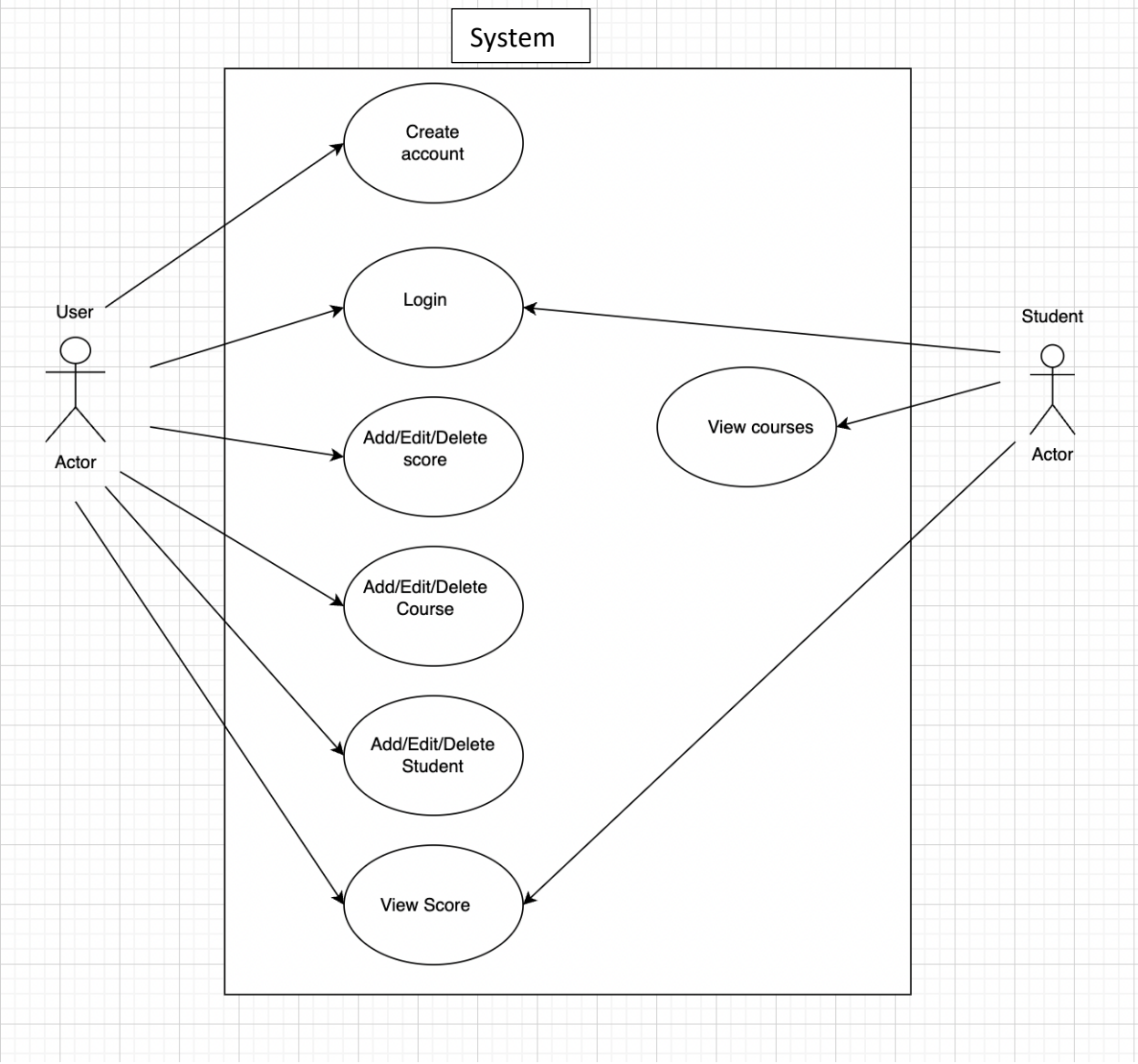
UR-5. The system shall be able to list all the student data is very less time.

UR-6. The system should not take more time to add or update student information/scores.

UR-7 The system shall not consume a lot of power for performing its operation.

Models –

Use case Diagram:



- Use Case Diagram

- Create account: Administrator can create login credentials which will be used by users.

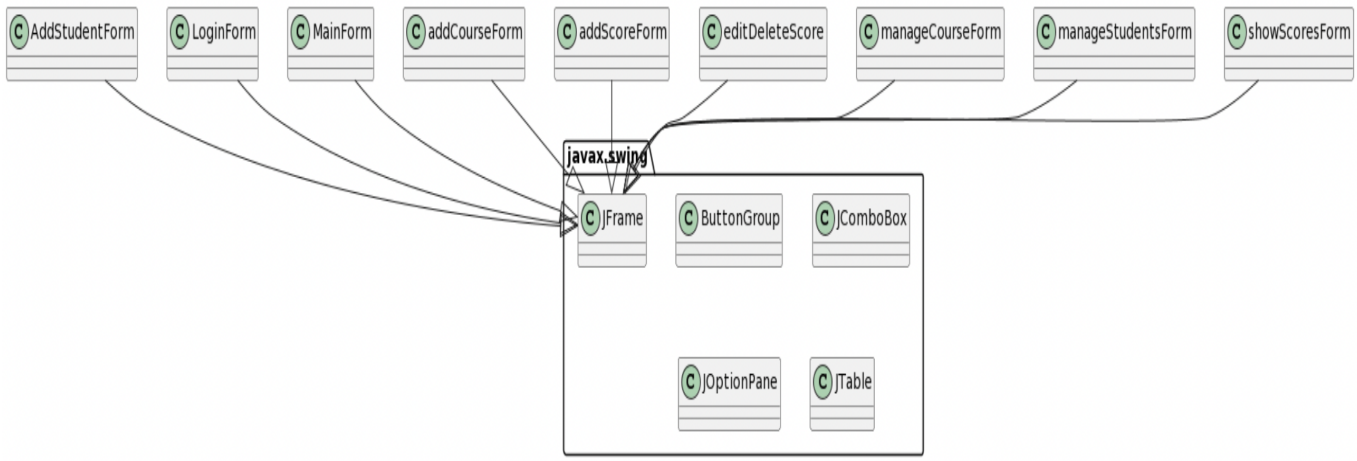
- Login: Users can login using the provided credentials. If credentials are entered wrong, user will not be allowed to go further.

- Add/edit/delete course: Users are allowed to add or edit or delete courses. If any course is deleted the subsequent scores of the same course will get auto deleted.

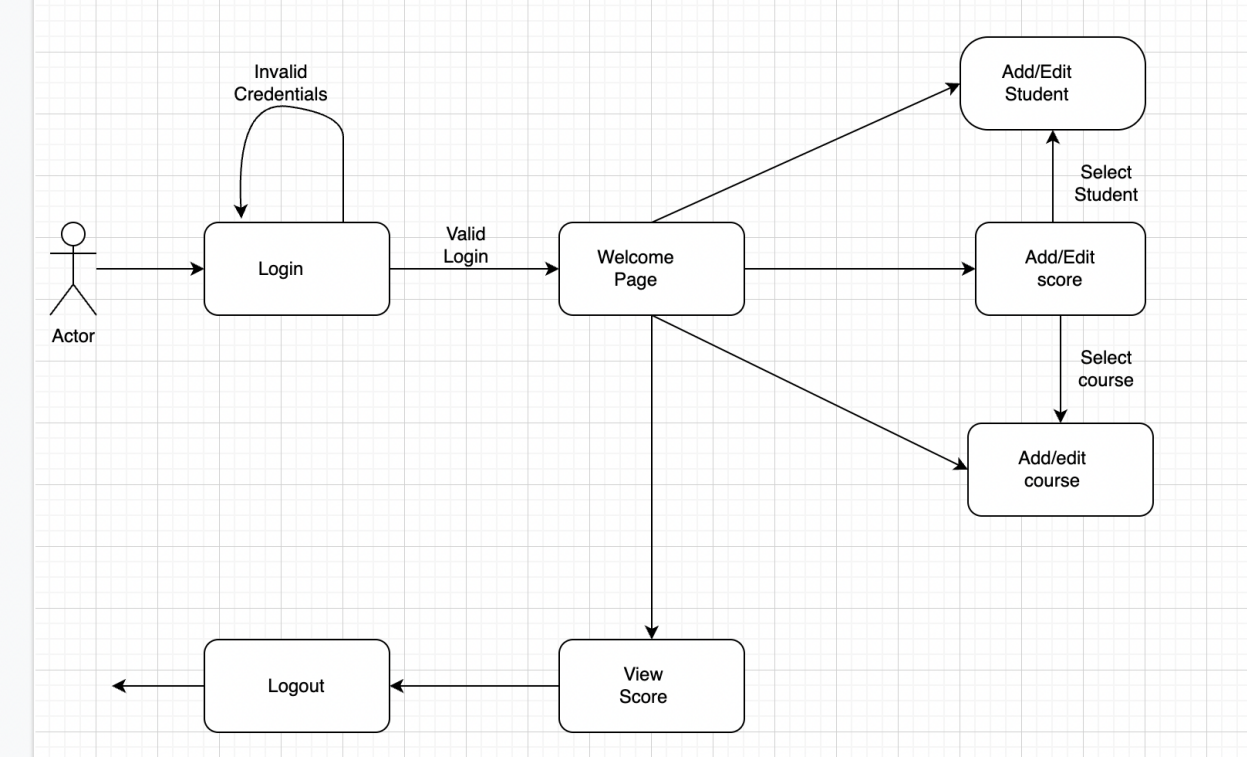
- Add/edit/delete student: Users are allowed to add or edit or delete student details
- Add/edit/delete score: Users are allowed to add or edit or delete score for the selected course.

- View Score: Admin user or student are both allowed to view scores and courses.

Sequence Diagram:



System Model-



Design Patterns-

Singleton pattern:
Used in the loginform.java class since login class will be used only once throughout the app.

Factory pattern:
Used instead of if..else condition. Instead of using if..else multiple times, factory pattern is used in manageStudentsForm.java

References:

https://netbeans.apache.org/kb/docs/java-se.html
apachefriends.org/docs/
https://dev.mysql.com/doc/