Final Project Readme

Project Summary

- The purpose of this assignment is to produce a comprehensive multi-tiered enterprise application, consisting of a persistence layer, business layer, and presentation layer, also making use of services provided by the container such as security and transactions.

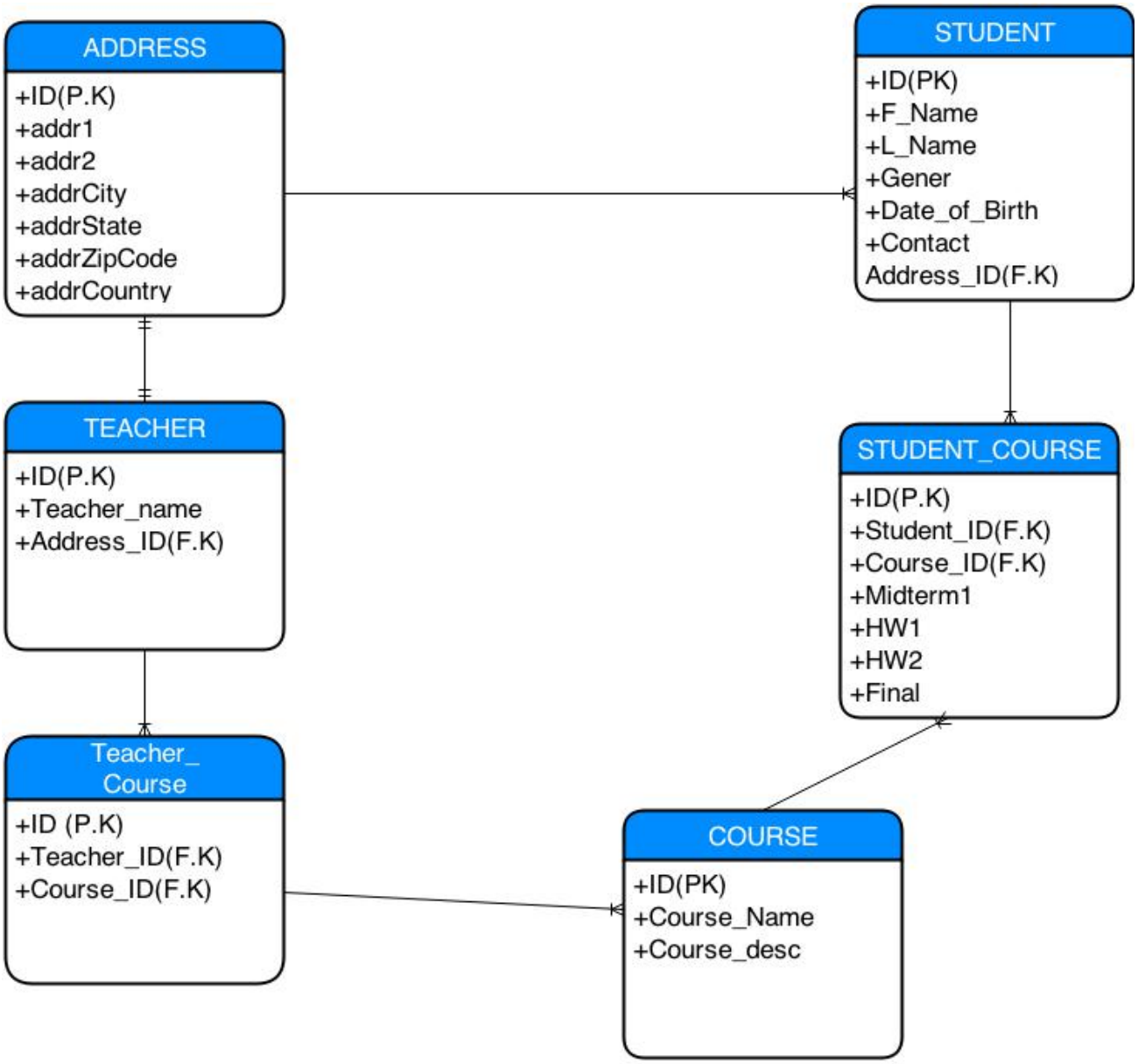
  The assignment consists of a new Web Application project, and should build on your prior work using the Glassfish application server:
  - My MP3 domain model as the persistence layer
  - My MP4 EJB or Service components as the business layer
  - My MP4 security structure and configuration

  For the final project, I have added **functionality** and **presentation** to my prior work using JSF or another MVC approach.

Design

I have designed a Student Management System where students can select the courses for which they want to enroll and the see the teachers assigned to that course. The mapping between the different entities can be illustrated as:

- Address
- Student
- Teacher
- Course
- Student_Course
- Teacher_Course



- The relationship can be explained as:
- There are four main entities: Address, Student, Teacher and Student. All have their unique primary keys to have a unique dataset.
- Address contains all the details required to find the address of a person.
- Student has its individual attributes and many students can have the same address (as in case of a siblings in the same school) and hence a One to Many relationship is maintained between Address and School.
- For a Teacher, I have assumed, that a teacher can have only one address and hence a One to One relationship is maintained between Teacher and Address.
- Course has its list of attributes such as course Name and its description.
- Now a student can have many courses and also a single course can have multiple students and hence there exists a many to many relationship between them. This is represented by having a Student_Course entity which having student_id and course_id as its foreign key from the student and course entities respectively. The many to many relationship is represented by having a One to Many between Student and Student_Course and a Many to One on Student_Course and Course.
- A Teacher can teach many courses and also a single course can have multiple teachers and hence there exists a many to many relationship between them. This is represented by having a Teacher_Course entity which having teacher_id and course_id as its foreign key from the teacher and course entities respectively. The many to many relationship is represented by having a One to Many between Teacher and Teacher_Course and a Many to One on Teacher and Course.

- On launching the project, the login page is displayed to the user where he can login or register to the application.

- Based on the login, the appropriate page gets displayed. The user can login in as a teacher ,student or an admin.

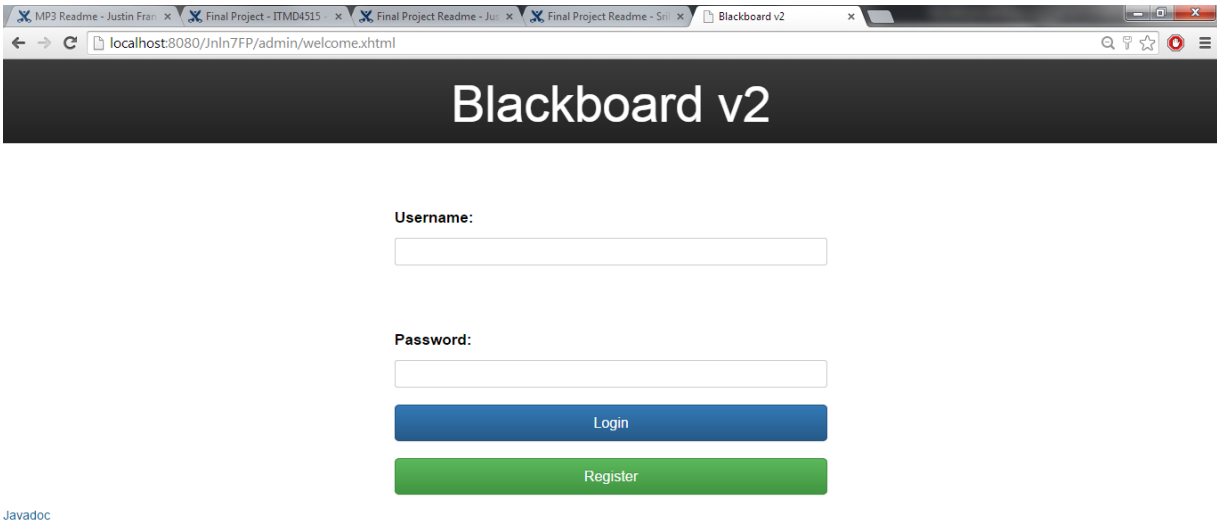
The various roles of each role is shown below:

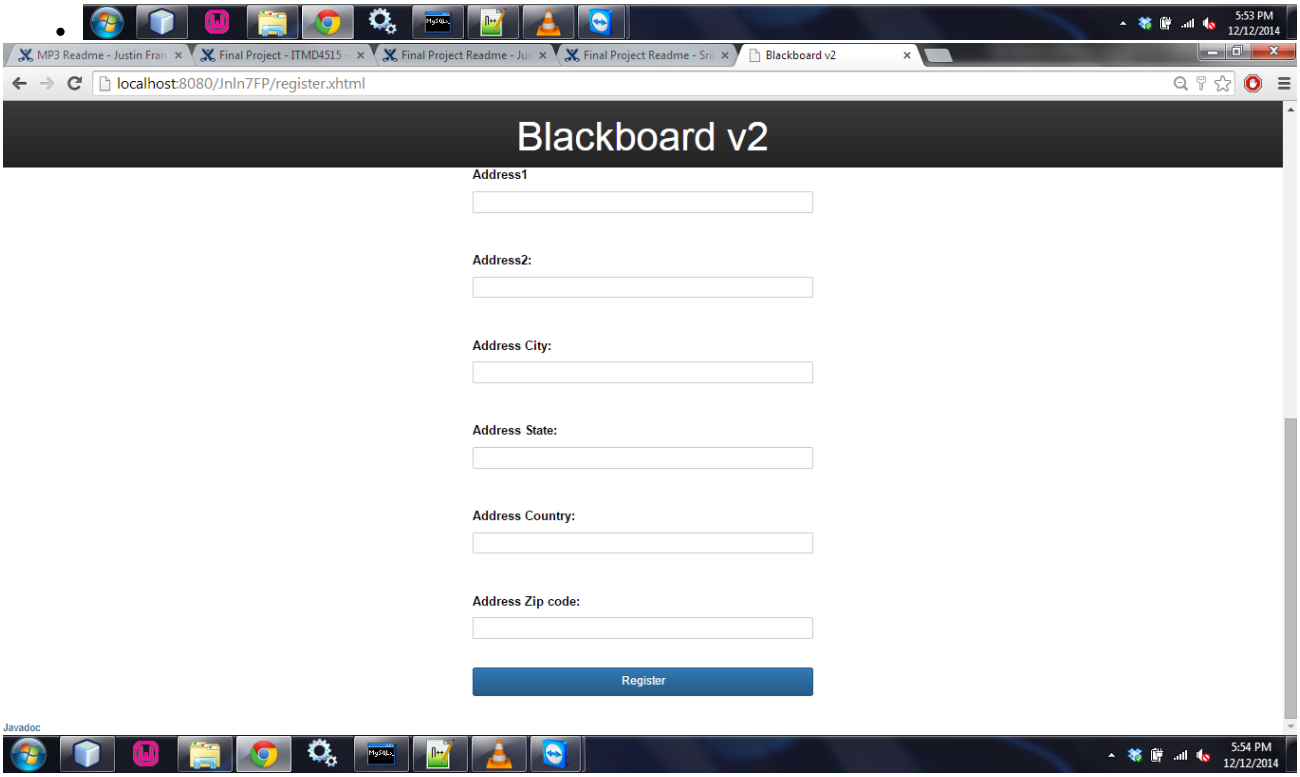
User Type	User Role
Student	-Update personal Info -Enrol for a new course View Enrolments
Teacher	-update personal info -view registered courses
Admin	-create new course -view all courses -search for an existing course

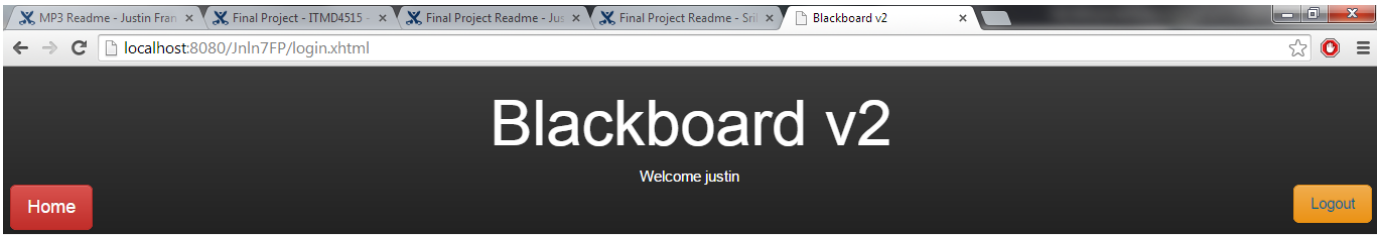
Requirements (Installation, Compile, Runtime, etc)

- Made appropriate use of MVC and its capabilities for reusability, for example Templating and Composite Components in JSF
- Create new user has been implemented. A new user can only be a student.
- An admin can search for a course based on the course name
- Provided functionality for displaying entities in tabular format where appropriate.
- Provided functionality for adding, deleting and modifying entities.
- Included appropriate navigation between pages, including the ability to return “home.”
- Server-side validation of user input has been ensured, and display appropriate messages and navigation if user input fails to validate.
- StartUpBean has been set to populate the database with all required values.
- Appropriate use of logging and exception handling.
- Javadoc has been created and a link has been added to the page.
- Apache commons codec (commons-codec-1.10-bin.zip) has been used to implement hashing passwords .
- Used bean validation to validate forms
- Implemented admin role

Screen Captures


Blackboard v2
localhost:8080/Jnln7FP/admin/welcome.xhtml
Blackboard v2
Username:
Password:
Login
Register
Javadoc

- 
5:53 PM
12/12/2014
Blackboard v2
localhost:8080/Jnln7FP/register.xhtml
Blackboard v2
Address1
Address2:
Address City:
Address State:
Address Country:
Address Zip code:
Register
Javadoc
5:54 PM
12/12/2014

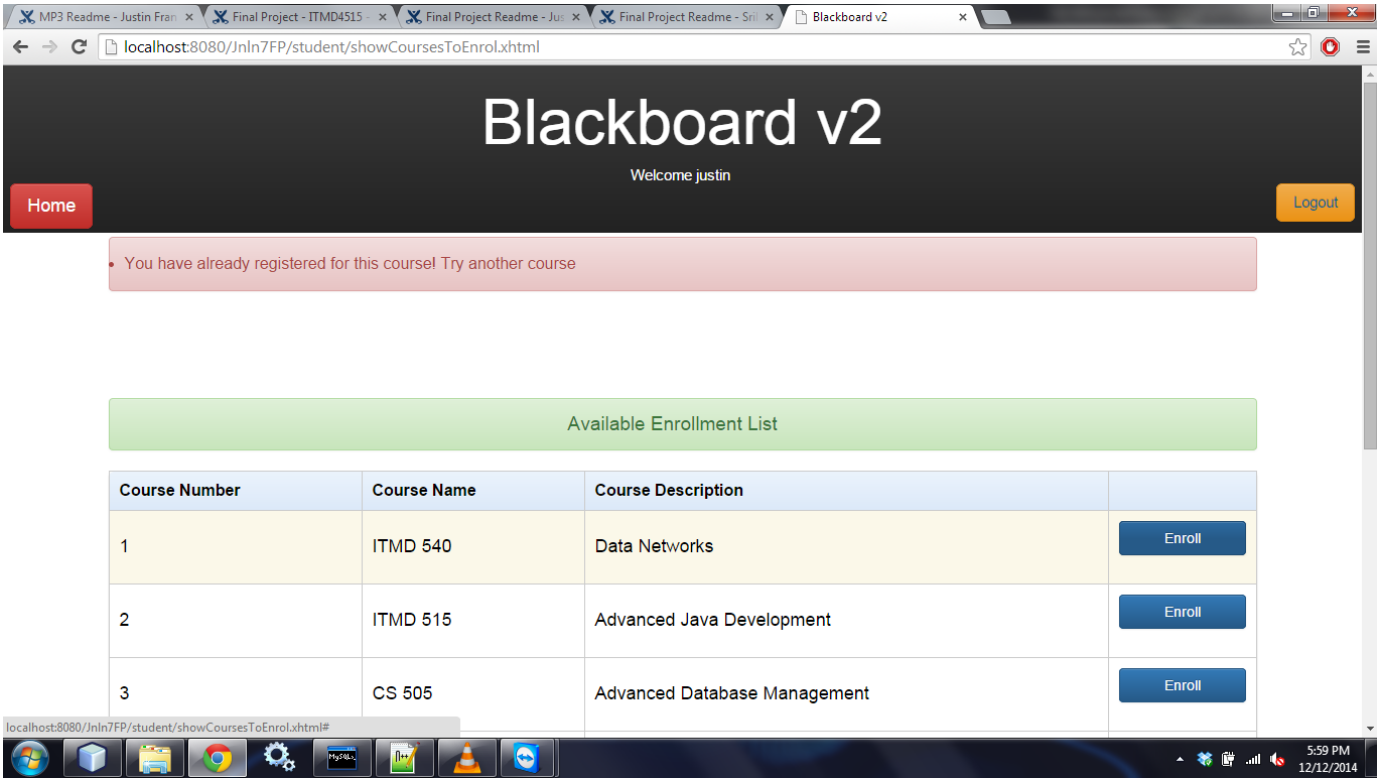


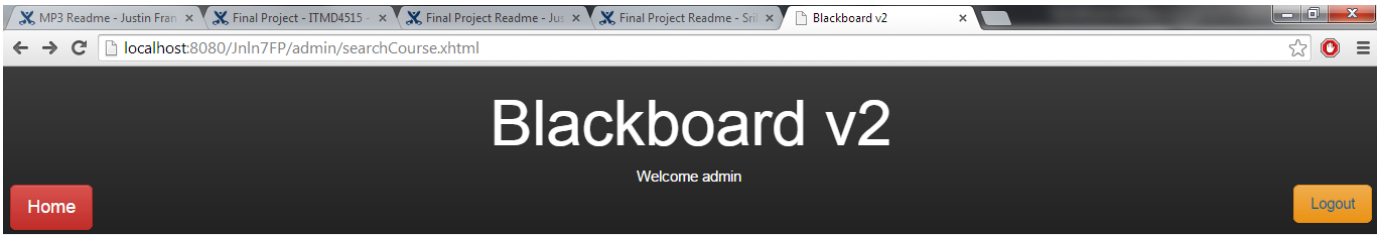
Update Personal Info

Enroll for a new Course

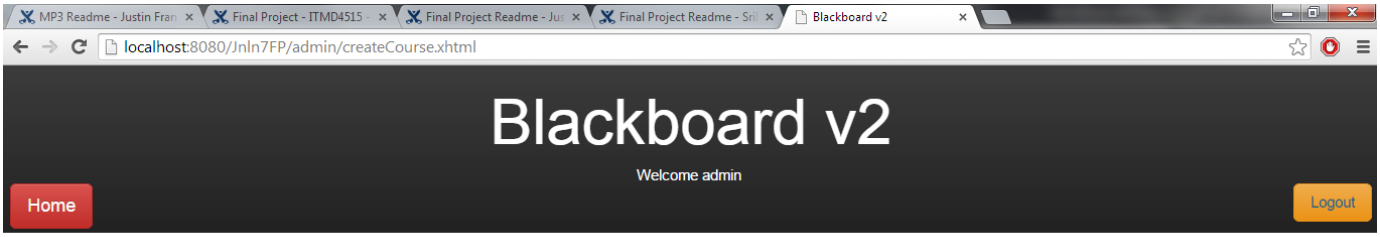
View My Enrollments

Javadoc





Javadoc



Javadoc

Expected Results/Known Issues

- Only students can be registered to the application.
    - I have created default login for different roles. Use these to test the application.

Role	Username	Password
student	justin	justin
admin	admin	admin
teacher	scott	scott