My student life

SE452 – Object Oriented Enterprise Application Development

Contents

[Team members 2](#_Toc352544209)

[Overview 2](#_Toc352544210)

[Requirements 2](#_Toc352544211)

[Use Case 2](#_Toc352544212)

[Description of problem 2](#_Toc352544213)

[Design 3](#_Toc352544214)

[Sequence of major functionality 3](#_Toc352544215)

[Web UI (Common case) 3](#_Toc352544216)

[Table layout 3](#_Toc352544217)

[Deployment 3](#_Toc352544218)

[Discussion of how your design met the requirements 3](#_Toc352544219)

[Discussion of lessons learned 4](#_Toc352544220)

[Decision Log 5](#_Toc352544221)

# Overview

A social networking app that allows students to rate professors. The student might sign up or login and rate the professor’s easiness, etc.

# Requirements

## Use Case

A student will log into “My Student life” and send messages, read their messages and rate professors.

## Description of problem

A centralized place that allows a student to check their messages, send messages and rate professors.

## Run Locally:

Right click on the project “**DepaulMyStudentLife**” and select “**Run As**”-> “**Spring Boot App**”

Spring Boot’s autoconfigure is already wired to set up the application’s configuration. Seeing as **application.properties** file is where most of the application’s configuration occurs. Spring Boot comes packaged with an embedded Tomcat Server. There is no need to upload a separate server distribution which will reduce the size of the project.

# Design

## Sequence of major functionality

### Web UI (Common case)

* Student signs up for application
* Student logs into application
* Student loads their profile
* Student sends messages to peers and rates professors
* User views requested information

The Web UI technology will implement HTML, JavaScript(JQuery Datatables) and CSS. JQuery Datatable implements an organized and precise way to access your data. You can search particular data in the search box. Front end pagination is implemented.

Registering will allow your credential to be saved in the database. You can create your own account by going to [http://localhost:8080/register](http://localhost:8080/studentLife/register) :

A screenshot of a social media post

Description generated with very high confidence

If you don’t feel like registering, fee free to to use any of the credentials I created via the register functionality. They are located at: <http://localhost:8080/students>

A screenshot of a social media post

Description generated with very high confidence

Please feel free to use any of the credentials:

[ {

"studentId" : 13,

"email" : "jenny87@gmail.com",

"password" : "1234",

"userName" : "jenny87",

"createdDate" : "2018-05-21"

}, {

"studentId" : 14,

"email" : "bobby87@gmail.com",

"password" : "456",

"userName" : "bobby87",

"createdDate" : "2018-05-21"

}, {

"studentId" : 15,

"email" : "anna87@gmail.com",

"password" : "789",

"userName" : "anna87",

"createdDate" : "2018-05-21"

}, {

"studentId" : 16,

"email" : "joe88@gmail.com",

"password" : "123",

"userName" : "joe88@gmail.com",

"createdDate" : "2018-05-21"

}, {

"studentId" : 17,

"email" : "erika1@gmail.com",

"password" : "890",

"userName" : "erika1",

"createdDate" : "2018-05-21"

}, {

"studentId" : 18,

"email" : "haywood@gmail.com",

"password" : "123456",

"userName" : "haywood0709",

"createdDate" : "2018-06-04"

} ]

Here is the page for <http://localhost:8080/login> :

If you login with the incorrect credentials or you don’t exist, you will receive validation errors.

A screenshot of a social media post

Description generated with very high confidence

Notice the envelops near each student’s username. Clicking on one should cause a modal popup.

A screenshot of a computer

Description generated with very high confidence

Click the envelope to send a message to use the user in the row. Hit the send button and the message will be saved in a document named   
“**messages**” within the MongoDB NoSQL database. Something to note is these messages are messages in which the logged in user is the recipient. If you log into another’s user account, you will see messages where they are the message recipient.

A screenshot of a computer

Description generated with very high confidence

Navigate to the Teachers link and click on blue profile icon to go a professor’s page:

A screenshot of a social media post

Description generated with very high confidence

Type in the text area to leave a comment. The comments are saved in a document named “comments” within the MongoDB NoSQL database read a comment regarding the professor:

A screenshot of a social media post

Description generated with very high confidence

## Table layout – High Level

Here is the relation data model:

A screenshot of a map

Description generated with very high confidence

NoSQL structure for Comments and Messages document:

A screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidence

## Deployment

The application is bootstrapped in an embedded Tomcat Server courtesy of Spring Boot. This allows more coding and less configuration of the project. Also the portability of the application should be high as there aren’t many configuration files such as server configuration files. For ease, the database configurations are in a **application.properties** file. I implemented remote database hosting for portability of the application. There is even a cloud version of this app that I am migrating.

# Discussion of how your design met the requirements

The sample project is expected to have enterprise functionality in it beyond logging in/logging out. Messaging peers and professors provides messaging functionality which is common in enterprise applications. Social networking functionality is often common used in enterprise applications.

# Discussion of lessons learned

I learned that implementing a mockup may help with the desing of the UI. I learned about how to pool to datasources at the same time-nosql and sql. I wish I had really deep dived when implementing my data. I got pretty involved with the Spring Boot. I am knowledgeable of Spring without Spring Boot, however I wanted to implement Spring Boot’s autoconfiguration to allow high portability.

# Decision Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem** | **What was decided** | **Alternatives considered** | **Rationale** |
| Which IDE to use | IntelliJ/Eclipse | Eclipse/ Eclipse STS | IntelliJ Ultimate has some great plugins packaged with it. The UI is great and doesn’t require a lot of plugins |
| Collaboration | It is just me | Course provided | We worked like one anyway |
| Code repo | Laptop/Google Drive. | Github | I am using Github as it is cleaner. |
| MVC Framework | Spring MVCs | Struts 2/ Basic Servlet | Spring is a popular framework that provides more security than struts. |
| Persistence | Hibernate, JPA | Basic JDBC framework | Hibernate, JPA provides more flexibility when using database entities within POJOs. |