

Full Stack Group MVC Project

Technologies

- Spring MVC
- JPA
- MySQL
- HTML/CSS

Overview

To wrap up the program's Java/SpringMVC/JPA materials we are going to have you build a full stack app in a group. In the workforce you will be working in groups on a daily basis, and it is good practice to learn how to contribute, plan, and communicate as part of a group.

If issues arise Kris, Andrew, Steve, and Rob will be acting as project owners for this project. You are welcome to approach one of us for advice on feature issues, as well as problems regarding individuals within your team.

You will be given time to meet with your group and come up with an idea that fits with the project requirements. From here you must pitch the idea to Steve, Rob, Kris, and Andrew. If we do not think your idea is feasible we will not allow you to move forward with the project. We are requiring all groups configure their GitHub repository as well as a Kanban, in our case Trello, on day 1.

Requirements

Below are a list of features for your group project. The scope of the project is not limited to, but must contain these features. It is important to get the basic functionality down before moving on to the more advanced topics.

Feature sets include:

- A fully developed database schema. The database must be connected to your Java code using JPA.
- Our friend C.R.U.D, this time manipulating objects and saving them in an appropriate database table.
- A user log in, which is authenticated by a mySQL table. There should be different permission levels for the users, one admin that can add/update/delete content present on the site.
- A correct implementation of a session object, this can be used to track a logged in user and any changes they make (for example, a shopping cart or favorites list).

- A way to track an individual users previous experiences on the site. For example if it is a food ordering system, a list of previous orders.

Milestones

Milestone #1: Set roles, configure tools and determine a plan. (Do this together)

- Complete the [project setup](#) & Configure with help of the [MVC Project Walkthrough](#)
- Create your project STS workspace
- Stub out your projects essential files and configure the basic gradle files.
- (One of you) create a public Github repository and invite your teammates as collaborators.
- Write a README.md file for your repository ([Markdown Cheatsheet](#)). Include:
 - Your names (or at least usernames).
 - The technologies you want to use.
 - Why you are building the application.
 - The methodologies you are using to build the application (group project, agile etc).
- Push your project up to GitHub for the 'init commit'
- Grab a white board and create a Kanban with post it notes OR create a [Trello](#) board and invite your teammates to it.
- Develop user stories from the requirement and add them to your Trello icebox.
- Build a wireframe for your site together. (So that you are all on the same page -- [more info on how to do that here](#))
- Slack a link to your Trello board, repository and your wireframe to Andrew. (This counts as submitting your proposal)

Milestone #2: Schema developed and tested

- Determine the relationships you will need for your database. Draw a schema that you will use for reference as you develop your application and distribute it to your team.
- Create and populate your database.
- Check that your relationships are correct with queries.
- Create a dump file to roll your database back to in the event that your lose data integrity.

Milestone #3: JPA mappings and config

- Create a separate JPA project
- Map your SQL tables to your Java entities
- Map your relationships
- Add your JPA project as a JPA to your Dynamic Web projects build path
- Configure your dynamic web projects -servlet.xml to create an entity manager, entity manager factory, and a transaction manager

Milestone #4: Basic UI functionality (MVP)

- Pages are served by a SpringMVC application.
- Pages link together according to requirement.
- Data is accessed via a DAO
- Basic styles and site architecture are present.
- Users can login.
- There are different user permission levels (one admin).
- Application is deployed to AWS.

Milestone #5: Dynamic application functionality

- Content is served from a mySQL database via your Spring MVC Controller.
- You have a Dao interface and JpaDao class that implements it.
- Content can be created, read, updated, and deleted (remember CRUD?) and be persisted to your database.
- Application is deployed to AWS.

Group Checklist

Date	Requirement	In Progress	Complete
Wed AM	Pitch Idea		
Thur AM	Milestone #1		
Thur PM	Milestone #2		
Fri PM	Milestone #3		
Tue PM	Milestone #4		
Thursday 14:00+	<u>Prepare Presentation</u>		
Friday ~12:00	Present Project		

Expectations

Your team will be given until next Friday to complete the necessary feature sets provided to you in the project specifications. Upon completion, your team should have the updated code in a git repository (can be on team members personal github account), as well as have the project hosted on each group members AWS server. Each group will have to discuss their project in a 8 minute or less presentation touching on feature sets, issues encountered, and the overall experience. After the presentation your group will field any questions that arise. As part of your grade, we will factor in your presentation skills (practice!!), as well as look at your github logs to make sure everyone is contributing and using git appropriately for version control.