

SOEN 387: WebBased Enterprised Applications Design

Assignment 3 on Architectural Patterns and TDD

Fall 2021, sections F

November 22, 2021

Contents

1	General Information	2
2	Introduction	2
3	Ground rules	2
4	Overview	2
5	Your Assignment	3
5.1	User Management Layer	3
5.2	The Test Project	4
5.3	The Business Layer	4
5.4	The Web Front-End	5
5.5	Documentation and Source Control	5
6	What to Submit	6
7	Grading Scheme	7

1 General Information

Date posted: Monday November 22nd, 2021.

Date due: Friday December 10th, 2021, by 23:59¹.

Weight: 9% of the overall grade.

2 Introduction

This assignment targets 1) architectural patterns and 2) test driven development.

3 Ground rules

You are allowed to work on a team of 3 students at most (including yourself). Each team should designate a leader who will submit the assignment electronically. See Submission Notes for the details.

ONLY one copy of the assignment is to be submitted by the team leader. Upon submission, you must book an appointment with the marker team and demo the assignment. All members of the team must be present during the demo to receive the credit. Failure to do so may result in zero credit.

4 Overview

This assignment is the third iteration on the “Poll System”, we created in assignments I and II. You are free to use the code base you developed in the previous assignments, refactor it, or completely rewrite it as a new project. In either case, make sure the resulting application is fully functional.

In this assignment we refactor the Poll system and implement the following: 1) User Signup and Registration using TDD, 2) Use of Base Patterns, and 3) Implementing View Patterns.

Note: The detailed description of the functionality of the system is specified in assignment 2. Section ?? lists the changes as well as the additional features.

¹see submission notes

5 Your Assignment

This assignment extends the web-based poll application system that were created in the previous assignment. The assignment consists of the following parts: 1) User Management Plugin, 2) The Test Project, 3) The Business Layer, 4) The Web Front-End, 5) Documentation and Source Control.

5.1 User Management Layer

In this assignment, we move the “user management” in a separate layer. To do so, we create a separate class library. The user management is then **plugged-in** to the business layer.

The user management core supports the following business functions:

1. *Sign-up*: A new user may be signed-up to the system by providing name, email address, and other necessary fields, used in the previous assignments. Upon sign-up, a random verification token is generated and attached to the record. The verification token is then plugged into a verification link which is **emailed** to the user for email verification. The password is then entered during the verification (see: “Email Verification”).
2. *Forgot Password*: This function is reserved to generate a change-password token. The change-password token is similarly emailed to the user. Only one change password token is allowed per user. Requesting another password change, overwrites the previous requests. There is no expiry on the password change. Once the record is marked for change, the previous password must be invalidated (i.e account is temporarily disabled).
3. *Email Verification*: This function is reserved to enable the user record (mark the record as validated) when the user clicks on the verification link. It may be used during both sign-up and forgot password processes, upon which, the user must choose a new password.
4. *Change Password*: Upon receiving the previous valid password, the user may choose and set a new one.

5.1.1 The User Management Plugin

As specified in the above, the User Management is implemented in the separate layer. The user management functionality is however part of the Poll System's business layer. For this implementation you need to apply the **Separated Interface (476)** Base Pattern.

Additionally, the User Management plugin is **NOT** aware of any UI setup. Hence, no assumptions on web platform nor the link format must be made in the implementation. The user management uses an **Email Gateway** to send the verification token to the user.

To compose the message text, a “message template” may be used as a resource. Use **Transform View (361)** to render the message.

5.1.2 The Email Gateway

The email gateway is used to send emails to the user. It is implemented using **Gateway (466)** and **Plugin(499)** patterns. See 2 to see how to send emails in java².

You may also use **Service Stub (504)** during the development, testing, or demoing your application.

5.2 The Test Project

In this assignment You will be using Test Driven Development (TDD) approach to implement the user management plugin. To do so, create a proper Test Project and write appropriate test cases. After finishing the project, run the tests and submit the report.

5.3 The Business Layer

Make the necessary changes to the business layer so that the user management is properly plugged in to the system.

IMPORTANT: No references to the “user management” nor to “email gateway” must be used. See **Separated Interface (476)** for the implementation.

²You may need to use a valid account for sending emails. While you need to demo the feature, please do not store your credentials in the project submission.

5.4 The Web Front-End

Implement the new UI component to support additional features as listed in the previous sections.

Demonstrate using one of the web presentation patterns in the Web-UI layer.

5.5 Documentation and Source Control

5.5.1 Source Control

This assignment too must be done using a source control system. It is recommended that a new branch is created for this assignment by forking the code from the previous workspace. Document changes and store them in the source control. Include the following diagrams in the source control as well.

5.5.2 The README File

The **README.md** file contains the team information, the description of the application, and the release notes (instructions on how to install the software components);

5.5.3 UML Class Diagrams

After the project is done, draw the complete updated UML class diagram of the Business, the User Management, and the Email Gateway layers and include them in your submission. Separate classes into logical packages and include the dependencies among classes.

5.5.4 Sequence Diagrams

Provide the **Sequence Diagrams** for the following scenarios. Include all classes from all layers that are involved in each case.

1. A user sign-up scenario.
2. A user change-password scenarios.

6 What to Submit

The whole assignment is submitted by the due date under the corresponding assignment box. It has to be completed by ALL members of the team in one submission.

Submission Notes

Clearly include the names and student IDs of all members of the team in the submission. Indicate the team leader.

IMPORTANT: You are allowed to work on a team of 3 students at most (including yourself). Any teams of 4 or more students will result in 0 marks for all team members. If your work on a team, **ONLY** one copy of the assignment is to be submitted. You must make sure that you upload the assignment to the correct assignment box on Moodle. No email submissions are accepted. Assignments uploaded to the wrong system, wrong folder, or submitted via email will be discarded and no resubmission will be allowed. Make sure you can access Moodle prior to the submission deadline. The deadline will not be extended.

Naming convention for uploaded file: Create one zip file, containing all needed files for your assignment using the following naming convention. The zip file should be called a#_studids, where # is the number of the assignment, and studids is the list of student ids of all team members, separated by (_). For example, for the first assignment, student 12345678 would submit a zip file named `a1_12345678.zip`. If you work on a team of two and your IDs are 12345678 and 34567890, you would submit a zip file named `a1_12345678_34567890.zip`. Submit your assignment electronically on Moodle based on the instruction given by your instructor as indicated above: <https://moodle.concordia.ca>

Please see course outline for submission rules and format, as well as for the required demo of the assignment. A working copy of the code should be submitted for the tasks that require them. Archive all files with any archiving and compressing utility, such as WinZip, WinRAR, tar, gzip, bzip2, or others. You must keep a record of your submission confirmation. This is your proof of submission, which you may need should a submission problem arises.

7 Grading Scheme

Full Functionality (features in A1, A2, and A3)	20 marks
The User Manager Plugin	15 marks
The Email Gateway	10 marks
Configuration and Resources	5 marks
The Web-UI (changes and patterns)	15 marks
Test	15 marks
Diagram and Documentation	20 marks

Total: 100 marks.

References

1. Martin Fowler, “Patterns of Enterprise Application Architecture (EAA)”, 2002, ISBN-10: 0321127420, ISBN-13: 978-0321127426
2. Sending Emails: <https://www.baeldung.com/java-email>
3. Online UML tools:
 - <https://yuml.me/diagram/scruffy/usecase/draw>
 - <https://yuml.me/diagram/scruffy/class/draw>
 - <https://plantuml.com/state-diagram>
 - <https://sequencediagram.org/>
 - <https://plantuml.com/ie-diagram>