



McGill

Department of Electrical and Computer Engineering
ECSE 321 Introduction to Software Engineering
Fall 2020

Project overview

The art gallery in your neighbourhood has always done business the old-fashioned way: connecting local artists with potential customers by showcasing their work at the gallery. The current pandemic is forcing them to change their business model and focus their business online. They hired you to help them design a new software system for them.

The gallery wants to focus on local artists and encourage them to use their website and app to promote and sell their work. Artists register, create a profile, and present any pieces they have to sell. Interested customers can browse and buy what they like. The gallery can also promote artwork available on their premises. Artists can deliver their art to customers by mail or can leave it at the store for pickup. The gallery makes its income on commission from sold artwork.

In teams of five students, you will gather requirements, design a multi-tier software solution to satisfy those requirements, implement the system, validate that the system is satisfying the requirements, and develop a release pipeline to automate the software delivery process.

Scope of the project

Your application must support the scenarios described *for every stakeholder*. All functionality of the system needs to be accessible via the web frontend for respective stakeholders. In addition, a mobile (Android) frontend shall allow the execution of the most important functionality for the given stakeholder, i.e. it shall have both read and write access to the backend via RESTful service calls. External systems or services are not required to be integrated.

Technological constraints

Your project should adhere to the following technological constraints:

1. For each sprint, your team must
 - 1.1. Provide project backlog using GitHub Projects.
 - 1.2. Use issues in GitHub to track *development*, *release engineering*, and *documentation* tasks.
 - 1.3. Define milestones of the project for each deliverable and assign all issues created during a sprint to its corresponding milestone.
 - 1.4. Provide documentation (e.g. meeting minutes with key decisions, effort table, models, supplementary images) using the wiki pages of the GitHub repository.
2. Starting from Sprint 1 (Database), your team must
 - 2.1. Use UML to create a domain model.
 - 2.2. Implement a persistence layer using a Postgres database.
 - 2.3. Use the ORM technology Hibernate to map objects to database concepts.

- 2.4. Create a Spring/Spring Boot project.
- 2.5. Configure a build system using Gradle.
- 2.6. Use a Continuous Integration process using Travis CI to build and test the database layer.
3. Starting from Sprint 2 (Backend), your team must
 - 3.1. Implement RESTful web service using Java Spring/Spring Boot.
 - 3.2. Provide a suite of unit tests for the backend using JUnit.
 - 3.3. Deploy the project as a Heroku application in addition to the constraints above.
4. Starting from Sprint 3 (Web), your team must
 - 4.1. Implement the web frontend using Vue.js.
5. For Sprint 4 (Android), your team must
 - 5.1. Implement the mobile frontend using the Android SDK but without the need for continuous integration and deployment for the Android frontend.

A team may choose a technology different from the recommended ones in the case of items 2.1, 2.3, 2.4, 2.5, 3.1, 3.2, and 4.1, but no technical support will be provided. All other technological constraints need to be respected.

Deliverables

You will deliver the system in four main iterations during the term. The corresponding deliverables are to be submitted at checkpoints throughout the term as described below. This section gives an overview of the deliverables. More details for every deliverable will be available.

Deliverable 1 – Requirements, Domain Modeling, and Database Design (10%)

Deliverable 2 – Backend and Quality Assurance (10%)

Deliverable 3 – Web Frontend and Architecture (10%)

Deliverable 4 – Mobile Frontend and Availability (10%)

Deliverable 5 – Presentation (5%)