



McGill

Department of Electrical and Computer Engineering
ECSE 321 Introduction to Software Engineering
Fall 2021

Project overview

The library of a small town needs your help to design a website and app to support their users. The library provides its community with access to books, movies, and music albums — all in physical form. The library would like the users to be able to browse the available titles, reserve them, and then pick them up afterwards. Some items such as newspapers and an archive can also be browsed, but not reserved or checked out. Users can checkout items for a specific period, after which the item must be either renewed or returned. In addition, the library can be used for events and booked by members of the community.

The user base is anyone living in the small town who can create a free account. Users can use the website or app to create this account, or they can ask a librarian to create an account for them. Users can choose whether they have an online account or not. If they have an online account, they must use a username/password, and provide an email address. If a user signs up with a librarian, then no username/password or email is required. All users must provide their full names and residential address, and they are issued a library card that contains a unique ID number. Users with local addresses that have been verified can access all the library resources for free. Citizens who live outside the small town can also use the library's services, however, they must pay a small fee.

The library is managed by a head-librarian who is assisted by several librarians. The head-librarian has all the privileges of a librarian in addition to a couple more. They can hire and fire librarians, assign them schedules, and decide the opening hours/days of the library.

In teams of five or six 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 GitHub Actions 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 2.3, 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%)