CS 3200 - Semester Project Overview

Khoury College of Computer Sciences | Fall 2024

Imagine...

Northeastern has approached your team to develop a data-driven application to improve some aspect of the Experiential Education space here. Here are some examples of broad areas your team could choose from (but you could also have other ideas that are in the experiential education space, and that is ok. Just run it by me.)

- anything dealing co-op/internship process or management (before, during and/or after coops),
- individualized professional development (skills, knowledge, etc, developed outside of coursework)
- connecting students with potential employers in new ways,
- connecting students with NU current students and alumni associated with a company/position of interest,
- connecting students with other students while on coop/internship (as in virtual community building)
- any combination of the areas above

You and your team are going to go all out and change the experiential learning space at Northeastern!!!

Designing and Building a Data-Driven Application

A **data-driven application** is one in which data serves as the primary component used in crafting the user experience, directing functionality, and aiding in decision making about the application itself or the related business context. In many ways, the vast majority of the applications you interact with on a daily basis could be considered "data driven".

50,000 ft. Overview of the Complete Project

Over the next ~month, in a team of 4 or 5 students , you will conceptualize, design, and implement a data-driven application. Your team will:

- identify a general data-centric software product idea in the context of supporting experiential learning for your project (some themes/ideas listed above)
- outline six major features of four different user personas of the product. (more detail in the Phase 1 handout)
- develop a localized conceptual data model (as ER Diagram) for each user persona
- incorporate the localized data models into a global data model (output is single ER Diagram)
- convert the global ER diagram into a relational model and then implement it in MySQL.
- bootstrap the database with realistic auto-generated data from Mockaroo, Python Faker Library, and/or other data sets
- develop robust queries to support each feature of each user persona
- design and implement a REST API as a data access layer using Python + Flask
- implement a proof-of-concept user interface for your application using the Streamlit framework[1]

[1] You may implement your UI in a different framework. At LEAST two of your team members must be familiar with your chosen framework to the level of being able to answer questions directly about the code that is part of your UI. If you choose a different framework on your UI (React for example), you are responsible for all debugging and the requirements stay the same. The TAs and I will stick to answering questions about and debugging Streamlit issues.

Teaming

Each team should consist of **no fewer** than 4 students **and no more than** 5 students. You are responsible for choosing your own teams. If you don't know many people in the class, you need to put yourself out there on Slack and CampusWire

A Word About Teaming: I advise you not to choose a team based solely on your friendships with people. Choose a team based on your direct knowledge of their working standards, availability, and seriousness regarding the project. Choosing only friends can (and has) ruined friendships (and other relationships)

High-Level Project Architecture

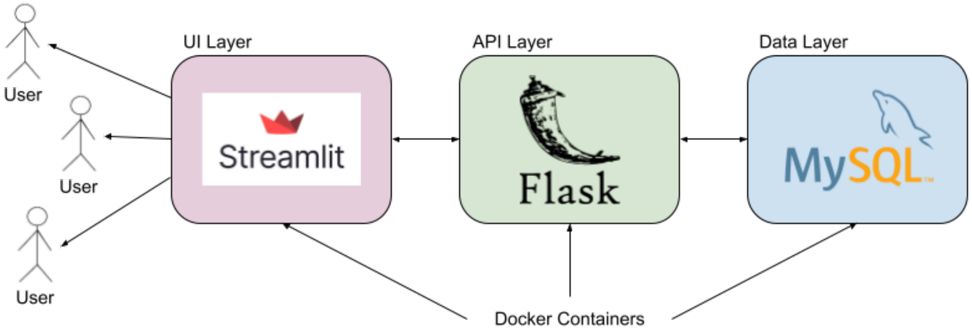

Figure 1 - General Project Architecture

You'll run three Docker containers on your local machine/laptop: a MySQL Server Container, a Python Flask API service Container, and a UI Container running the Streamlit app. A boilerplate project repository will be provided with the infrastructure already set up for you.

- AWS Docker documentation describes it: *"Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers with everything the software needs to run, including libraries, system tools, code, and runtime."* We won't be deploying on AWS, but this description is on point. Once you have a system set up to run in Dockerized containers on your development platform, they can easily be deployed to production to a system like AWS, Azure, etc. It also provides a consistent development environment for all team members.
- MySQL …. you know what this is 🙂
- Flask is a lightweight framework written in Python for building web apps, REST APIs, and other applications. It is very lean by design, but extensions provide additional functionality.
- "Streamlit's tagline is "*a faster way to build and share data apps.*" Streamlit allows you to build powerful web apps using pure Python with much less coding overhead than a framework like React or NextJS. It features a rich collection of UI elements such as text areas, select boxes, check boxes, and more advanced widgets such as graphing and mapping.

Important note: While your MVP UIs will be built for different user types/personas, ***you don't need to worry about implementing user login/authentication systems***.

The project will be completed in 3 major Phases:

- Phase 1: Ideation and Design of User Personas, features, and wireframes.
- Phase 2: Generating localized ER models, global ER model, and Physical model along with auto-generated data; Design of the REST API.
- Phase 3: Implementation of the REST API and User Interface supporting multiple personas (more to come)