CS 4400 Phase 4 Instructions
Fall 2019

The instructions for Phase 4 of this project are similar to the Phase 3 instructions in that your team is required to implement the Atlanta Movie application as described in the original Atlanta Movie Project Description. The main difference(s) between Phase 3 and Phase 4:

- Phase 3 only requires you to develop the underlying SQL queries that are needed to manage the data
- Phase 4 requires you to develop the full application, to include an underlying data management system and back-end, along with a front-end that implements the screens and navigation between screens

Our goal for the "new" Phase 4 version of the project is to allow significantly more choices in terms of the technologies that you can select for your implementation. The main intent is that you must develop all aspect of your system – you cannot "contract out" the implementation of any component of your system to other persons. Also, we still reserve the right to disallow any technology for any reason.

Deliverables

- Each team must submit a ZIP repository that includes all of the source code needed to build your application
- Each team must also submit a list of directions (e.g., compilation commands) on how to build the application from the provided source code

Approved Technologies

Assume anything NOT on the list below is (tentatively) forbidden, but you may check with us to see if your technology request can be approved and added/allowed. You may be required to provide descriptions of how you intend to use the technology before approval is granted.

Approved Database & Data Management Systems

Code directly responsible for managing the customer data

- MySQL v8.0 or later
- If using MySQL, then the InnoDB Engine is highly recommended (and is normally the default selection)
- NoSQL/Non-Relational Key-Value Storage Systems (e.g. mongoDB)
- Other SQL/Relational Database Management Systems (e.g., SQLite, PostgreSQL, MariaDB)
- Object-Relational Mapper (ORM) Systems

Approved Back End Technologies

Code that connects to the data management system and provides business logic support

- PHP
- NodeJS (Express)
- Other packages from npm are allowed as long as they do not generate SQL code in any form
- For MySQL, use https://www.npmjs.com/package/mysql or https://www.npmjs.com/package/mysql2
- Java (JDBC, JSP, Java EE)
- Python (Flask, PyMySQL)

Approved Front End Technologies (Web, Desktop or Mobile Applications)

Code that implements the user-interface to accept user commands and render information

Web Applications

- HTML/CSS/JavaScript
- Templating engines are ok (mustache, handlebars)
- AngularJS (v1.x)
- Angular (v4.x)
- Vue.js
- React
- Bootstrap
- jQuery
- Foundation
- Sass/SCSS
- CoffeeScript/TypeScript

Desktop Applications

- Java (JavaFX/AWT/Swing)
- Python
- TkInter
- PyQt4/PyQt5 or Qt (C++) with QtCore, QtGui, and QtWidgets (and you may use QT Designer)
- Electron
- Ruby
- Rails
- Sinatra
- Django
- squel.js
- QtSql

Mobile Applications

- iOS, Android, or similar mobile platforms

Front End Design Guidelines

- You can take some “creative artistic license” with the colors and organization of the screens, but each screen must provide the same functionality as described in the project description.
- You must ensure that your user interface is clear and usable, which will be critical when performing the final demonstration and evaluation. This is especially true if you implement this as a mobile application.