CS 426: Software Engineering

Project Assignment 4

University of Nevada, Reno
Department of Computer Science & Engineering
Instructors: David Feil-Seifer, Devrin Lee
Team 13
Authors: Matthew Alighchi, Guillermo Del Valle, Sherman Lee, Kyle Respicio
External Advisors: Dr. Casey Lynch, Vivan Zavataro, Tyler Brownlow-Calkin
March 22, 2021

0 Table of Contents

1 List of Use Cases and/or Requirements Demonstrated

1.1 Robot Interactions with Museum Goers

- Mobile app displays questions for the appropriate piece.
- Mobile app queries the database for new questions when needed.
- Completed UI elements: (Clickable Questions button displaying answers, button to make questions visible, general structure).

1.2 Robot Servers

- The robot will query the database.
- The robot will refresh the mobile website so that museum goers can ask questions.
- The robot will navigate through the different pieces of the tour.
- The robot will broadcast the information about these pieces.

1.3 Front-end Map Editor

- Updated sidebar UI from expanding accordion to clickable cards
- Included add button on the sidebar for new exhibits to update the state of exhibit lists
- Replaced accordion form fields with modal form fields
- Modified UI design of modals to look more modern
- Allows saving new exhibits and updating current exhibits to ping the server and update the database

1.4 Database and Server Management

- Login Authentication
- Database Persistence
- API Documentation
- Backend Security

2 List of Use Cases and/or Requirements Intended for Final Demo

2.1 Robot Interactions with Museum Goers

- Mobile app interrupts robot
- Robot has text-to-speech for when user clicks question
- Mobile app will have a togglable map of the museum

2.2 Robot Servers

- Robot will go into an “error” mode when an error occurs in navigation.
- Robot will be able to activate tours by time
- Robot will be able to send its maps to the database for the web application.
- Robot will be able to keep track of user interactions.
- Unit tests

2.3 Front-end Map Editor

- Incorporate working card list component for questions in exhibit modal
- Modify UI floating action button modal for adding pieces
- Add similar card list component for pieces in exhibit modal
- Map Editor UI
- Analytics Page
- Profile Page

2.4 Database and Server Management

- Unit Tests
- Integration Tests
- Analytic preprocessing
- Refactoring questions for exhibits and pieces
- Refactoring Photo Storage

3 Current Project Status

3.1 Robot Interactions with Museum Goers

By our December progress demo, we had only made a textbox, in simulation, that displayed text associated with each piece, and queried this information all at once. Now, the robot communicates with a mobile website. Currently, the mobile website provides museum goers a means of displaying appropriate questions and answers from the associated pieces, queries from the robot in real-time, and has most of its UI components completed.

3.2 Robot Servers

In regards to the robot servers, we have completely redone the way that data is queried from the database. Before, the robot queried everything at once from the database. Now, the robot queries one piece at a time and publishes the information to specific topics. Different modes of operation were also created for the robot, including navigation, error, presentation and interactive states.

3.3 Front-end Map Editor

The sidebar has been updated from an expanding accordion component to be a list of card components that prompt a modal with fields to fill. Information is now able to pull and save to the database since the last demo where the data was populated from a dummy file. More styles were also incorporated to make the design more modern to our standards.

3.4 Database and Server Management

The server and database underwent refactoring to ease code maintenance in the long run. First, the server side code was split up into different routes and now follows a more traditional flask app structure. A postman documentation website was also created to provide clean documentation for the rest API. The database added additional tables to support user authentication along with relationship tables for complex relations such as tours and pieces. Database entries now have a randomly generated UUID. New flask routes safeguard edits and updates to the database; moreover, these routes are protected by an auth guard as an added security feature. Prior to this, the backend was essentially a single file with poorly named and insecure routes.

4 Time Worked on Project Concept

Matthew worked a total of 31 hours on robot interactions with museum goers
Guillermo worked a total of 30 hours on robot servers
Sherman worked a total of 32 hours on front-end map editor
Kyle worked a total of 33 hours on database and server management