OULU UNIVERSITY OF APPLIED SCIENCES

Qt GUI Interaction with an Express API

Antti Suomi, Solo project
Information Technology, Product and Device Design

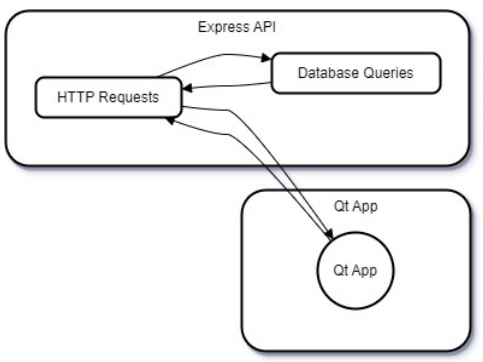
Introduction

I have been facing a lot of challenges during this year on account of my health and even though I started with a group it came apparent that I would not be able to do things in at the group's pace.

Objectives

The goal of this part of the semester is to show an understanding of QT, C++ classes and an API driven database. This project involves setting up a GitHub repository with a REST API and a Qt application. The REST API will interact with a MySQL database containing tables for books and cars. The Qt application will provide a user interface to interact with the car data in the database as shown in figure 1 below.

FIGURE 1. The System Diagram



As I am finishing the project solo my goals are first and foremost to fulfil the minimum requirements. If there is time left by 14.12, I plan on dockerizing the database and adding Github Actions to be included in the repository.

Software Application Project

ECTS Credits: 15

Date of Publication: 2023, Winter

Instructors: Pekka Alaluukas, Johanna Talvensaari

Methods

A public repository was created on GitHub and this repository was cloned to a local machine.

Following the example on the PEATutor.com website, a MySQL database was created.

A book table was created in this database, and CRUD (Create, Read, Update, Delete) operations were implemented in the REST API for this table.

These CRUD operations were tested using Postman to ensure their functionality. A new table named car was added to the database.

The table included an integer primary key and text fields for branch and model.

The SQL code used to create this table was documented in the README file of the repository.

The car table was used to store information about cars, specifically their brands and models.

CRUD operations were developed in the REST API for the car table.

These operations were tested using Postman to ensure they worked correctly.

A Qt GUI was developed to utilize these operations in QT 6.0

Results

This project provided hands-on experience in full-stack development, encompassing database and API design, as well as front-end development with Qt, demonstrating how different layers of a software application interact.

Use of Node.js and Express allowed a quick spawning for the API and testing with Postman allowed for a fluid and low overhead access to the actual parts which needed to be tested regularly

Technical Aspects

Changes were regularly committed to GitHub for tracking progress and maintaining version control.

Database connection strings and credentials were managed securely, considering the public nature of the repository.

Comprehensive testing with Postman was conducted to ensure the API handled all CRUD operations correctly and managed errors gracefully.

The Qt model-view architecture was utilized for efficient development of the user interface.

Clear documentation, including setup instructions, API endpoints, and SQL table creation scripts, was maintained in the README file.

References:

Basic database model:

https://peatutor.com/express/Examples/mysql_index.php