CSC431

Download of Public-facing Data

System Architecture Specification

Team #3

Jerry Bonnell Gururaj Shriram Re Chang Heyu Yao Lixiong Liang

Version History

Version	Date	Author(s)	Change Comments
1	March 16, 2018	Jerry Bonnell and Gururaj Shriram	First Draft

Contents

List of Figures

List of Tables

1 System Analysis

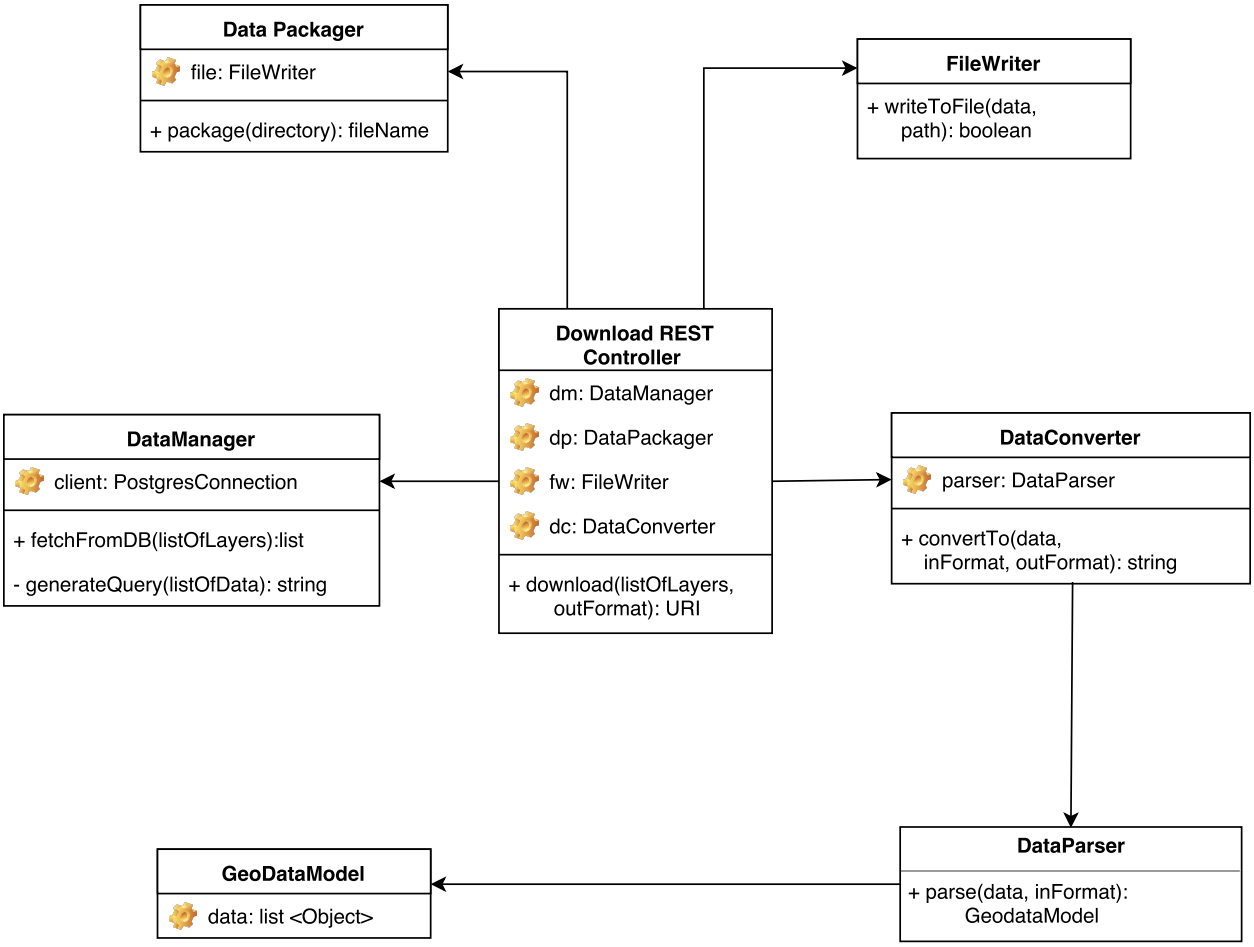
1.1 System Overview

The System will be comprised of four main parts: a data manager, data packager, data converter, and data controller. The data manager will control the connection to the Access Database and handle database queries. The data packager will package all requested pieces of data into a compressed file if multiple pieces of data are requested. The data converter will convert files from the database into the users' desired file format from the list of supported file formats. The data controller will manage all of the other aspects of the system and manage the end-to-end pipeline of the downloading process.

The type of architecture used for the application is primarily the MVC (model view controller) style. The model service will sit in the Postgres database, Express will serve as the controller, and the view will be exist as part of another team's system.

1.2 System Diagram

Figure 1: Download System Diagram



1.3 Actor Identification

There are three types of human actors: unauthorized users, authorized users, and the administrators. Unauthorized users may only download public pieces of data unless they are granted permission to specific pieces of data from an administrator. Authorized users are allowed to download any piece of data. Administrators are able to download any piece of data as well as approve requests for unauthorized users to download private pieces of data. There are currently no plans for the system to support non-human actors, i.e. with an API.

1.4 Design Rationale

1.4.1 Architectural Style

We will use the MVC, 3-tier architectural style pattern because this system is a web application which can be efficiently divided into to the model, view, and controller. The model lies on the database and manages the exchange of data. The controller receives user input, validates the input, and requests data download from the model. This system, which is part of a larger system, primarily lies on the backend and hence the "view" is only comprised of checkbox selections and download button requests from the user.

1.4.2 Design Pattern(s)

Express is a flexible Node.js framework that provides a robust set of features. We predict that the following design patterns will be primarily used:

- Factory
- Singleton
- Middleware
- Stream

1.4.3 Framework

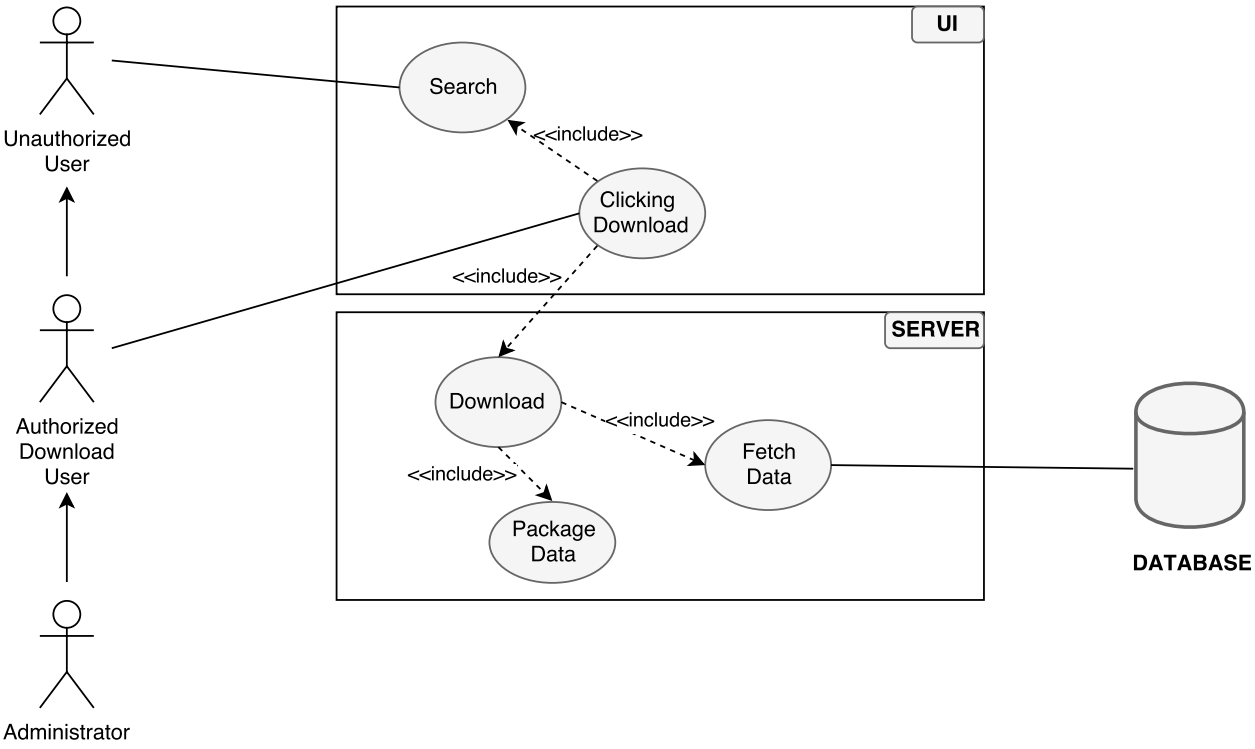
The web application will run on the `Node.js` JavaScript engine on the `Express` framework. `Express` allows for the successful implementation of a MVC on the backend and will facilitate a lightweight, efficient download pipeline.

Links:

- `https://nodejs.org`
- `https://expressjs.com`

2 Functional Design

Figure 2: Download Use Case



3 Structural Design

Identify all components and model them using class diagrams.