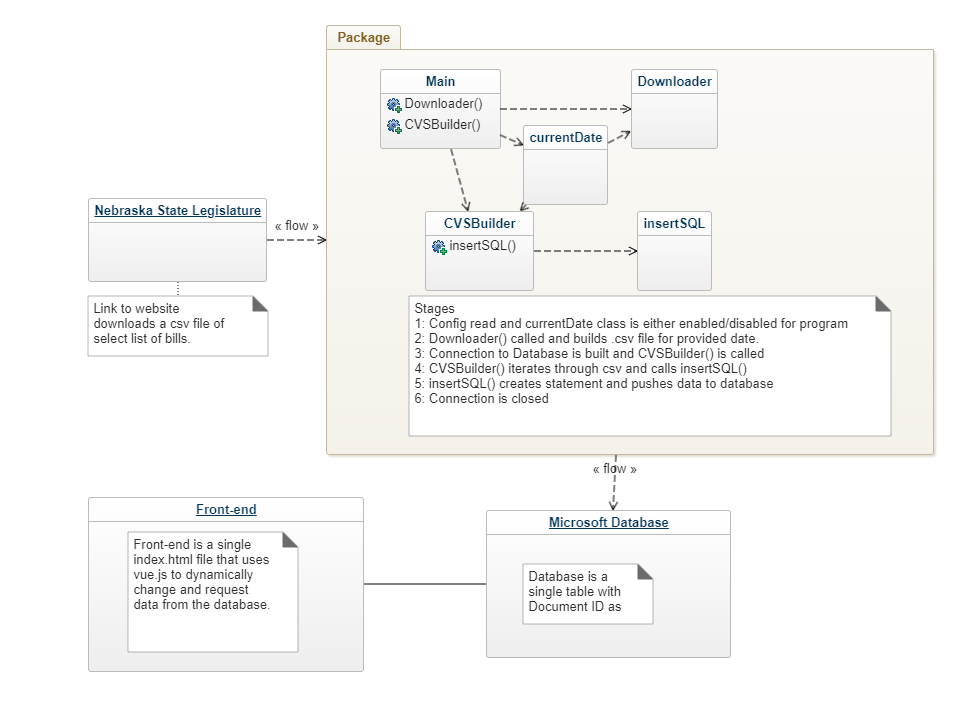
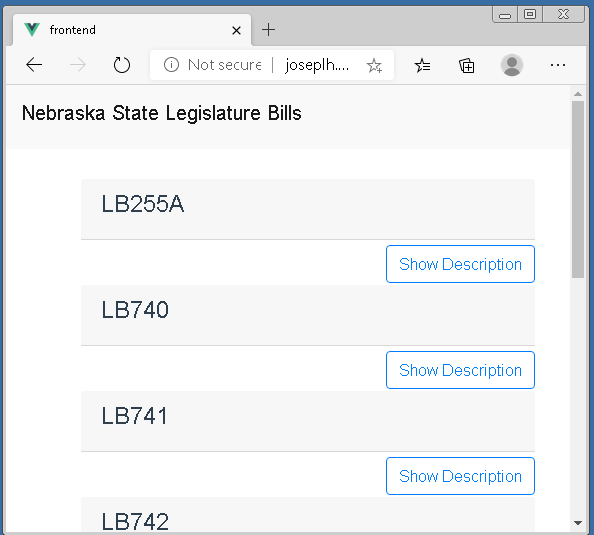
Joseph Maxwell

Marcus Benesch

State Bill Tracking

The state Bill Tracking system operates on a single windows 7 ultimate edition server housed in the basement of Joseph’s home. This server was previously used to host Joseph’s personal website which he used as a demonstration of his ability to host and maintain a live server to prospective employeers. Because of this, the front facing website is hosted on IIS (Internet Information Services), which was used previously for the personal website. The database used for the website is hosted using MSSQL (Microsoft SQL) and maintained through SSMS (SQL Server Management Studio). The website uses VUE.js to create dynamic webpages and using drivers available from Microsoft, it can pull requested data from the database. Beyond the website, on the back-end a standalone java program is used to scrape data from the state legislature website and submit it to the database. It is designed to be semi-modular and includes a config.ini file to allow for custom requests and integration.

1. Decomposition Description (static models)
2. Dependency Description (dynamic models)
   1. Process Views
      * Screen Navigation
      * Execution Flow
        + For the user, they will initially be presented a blank webpage that contains a search bar and filters. From there they can request specific information or bills. Using that data, the website then requests the data from the database and changes the page to a list of bills and description.
        + An environmental activist searches for “environment” to find what recent actions have been taken that reference the environment. The webpage filers the descriptions for “environment” and compiles a list of bills referencing the environment.
        + A concerned citizen wants to know what recently was done. Instead of entering a search term, they select latest within the filters. The webpage pulls all bills and presents the top 100 most recent based on the timestamp for the bill.
      * Database Dependencies - discusses dependencies between code modules and database tables.

There are three major dependencies that must work properly for the system to operate normally. The Nebraska State Legislature website can be redesigned at anytime and this would break the java program’s ability to pull data. The result would be that the download file would be empty and no statements would be inserted to the database. The second dependency is the connection from the java program to the database. If the connection is broken/changed, statements made from the java program would fail to upload. Finally, the front-end vue scripts also share the same dependency with the connection to the database.

1. Detailed Design - describe any nontrivial algorithms used.

The java program used has multiple modules to pull data from the Nebraska legislature and push the data to the database. The Downloader() method pulls data from the website and writes the results to a .csv file in the same folder as the java program. That .csv file is then read by the CVSBuilder() method and statements are parsed from it. The statements are sent to the insertSQL(string) method. InsertSQL then pushes the sql statement to the database. Finally the java program closes all connections and files.

1. Design Evolution
   1. Currently the design is a modular system where if we were to add more State government bills, it would be possible to redirect the java program to read from multiple sources and modify the database to organize hold multiple tables for each State Legislature.
   2. Within the java program, it reads from a config.ini file to fill out critical information needed to operate. I can insert a loop where the final lines can list all sources to download data from. There is an unused class that appends .csv files together and with the multiple sources, this class can append multiple sources together for the program to then insert into the database.