Joseph Maxwell

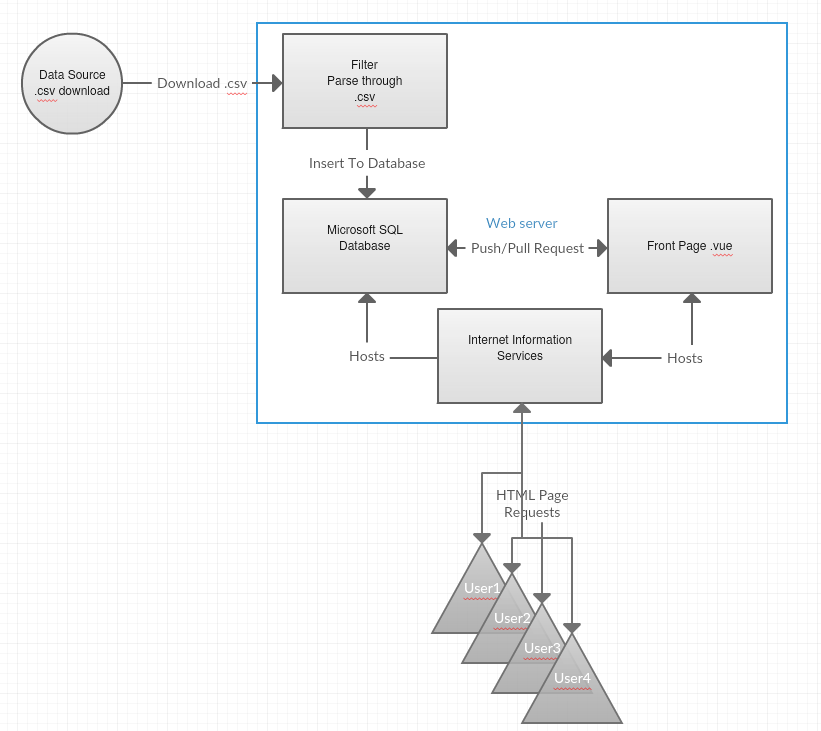
Marcus Benesch

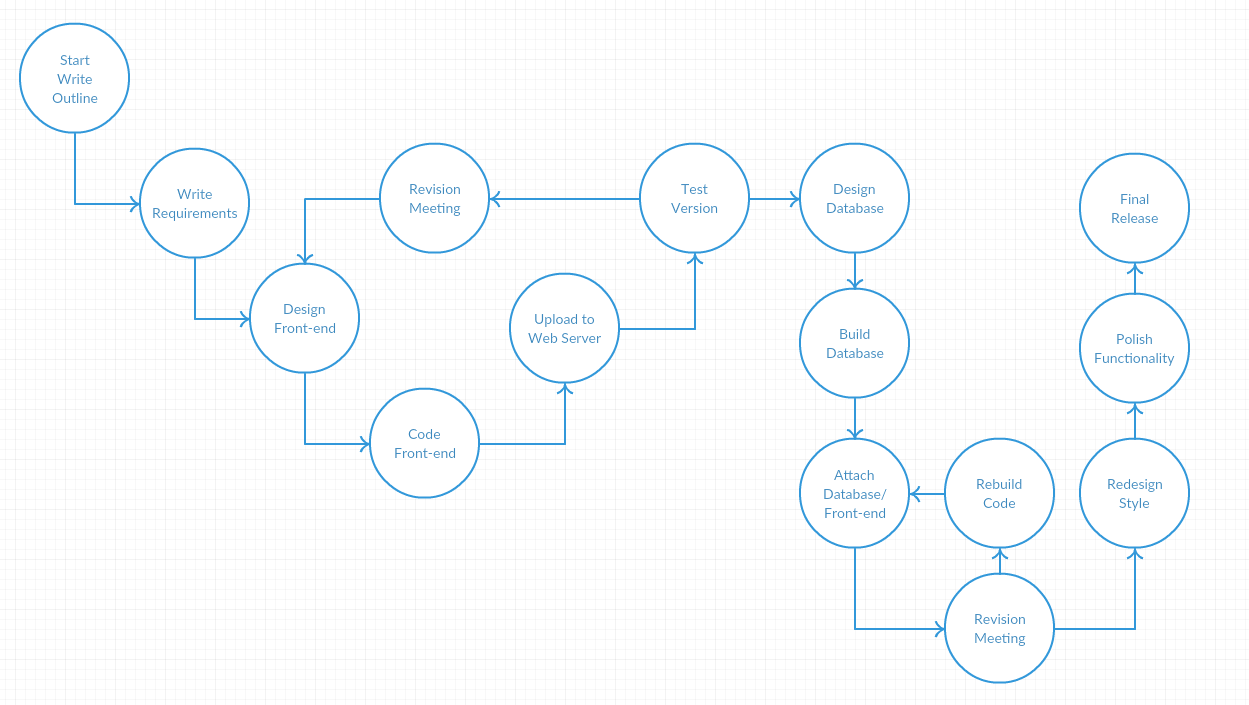
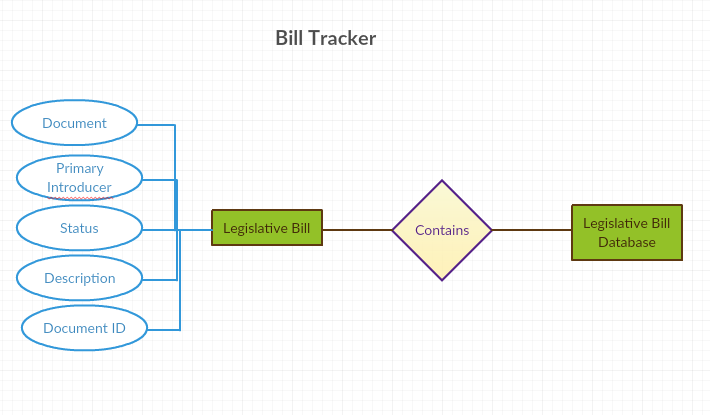
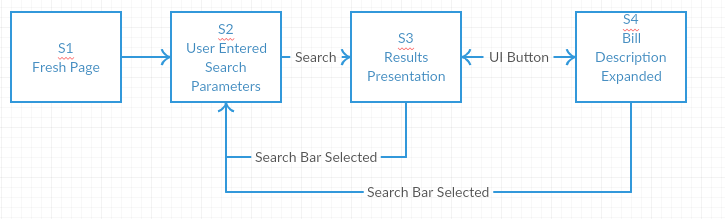
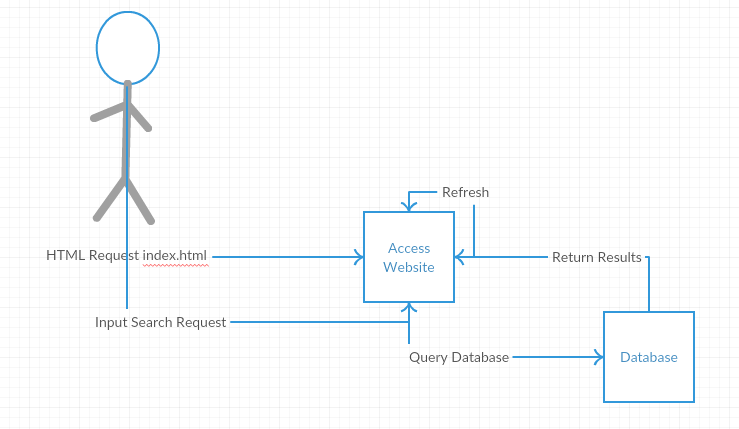
Software Design Document

1. Introduction

State Bills are often difficult to find and when they are found, this presented information is useless to the layman. This website and application aims to solve this problem by making searching for and reading bills easier. The Nebraska State Legislature allows users to download a quick list of all bills in a year session. By taking this .csv file we can parse through it and represent the descriptions of the bills to users.

1. Logical View of Architecture - explains the high level structure of the system's architecture; the rationale behind this structure as well as the rationale behind the choice of frameworks used. Useful diagrams include a box-and-line diagram of the logical view. This will show the logical components/subsystems to be implemented.



1. Decomposition Description (static models)
   1. Development View
      * How the development is expected to go, the front-end is designed and built first. This front-end lays out expected goals and requirements for the database and usability. After the front-end, the database is built and integrated to the front-end.
   2. Database View
      * An E-R diagram is useful here.
      * The database the application is using is very simplified. The “Legislative Bill Database” is a single entity that contains many “Legislative Bill(s)”. Each bill is composed of five attributes, Document, Primary Introducer, Status, Description, Document ID, and Date.
2. Dependency Description (dynamic models)
   * + Screen Navigation
       - Front Page only shows a search bar and drop-down menu. User can enter search parameters. Page is refreshed with list of resulting Legislative Bills. Selecting a bill will expand the description. At either results presentation or expanded bill view, if the search bar is selected the bills are minimized and refreshed based on search request.
     + Execution Flow
       - User access the database and is initially presented with a search bar/menu. User enters search request. Website queries the database. The database return results and refresh the page. User scrolls through results. User then selects a bill, expanding the description.
     + Database Dependencies: The front-end requires the ability to query a SQL database on the same server hosted. The database is built on Microsoft SQL server and the website is hosted using Internet Information Services.
     + External Dependencies – There is only one major external dependency. The Legislative website is the source of the .csv file necessary to update. Beyond the website, there are numerous minor dependencies, such as Java Development Kit, Internet Information Services, and Microsoft SQL Server.
3. Interface Description – The interface is a simple search bar and drop down menu.
4. Detailed Design – Java program used to pull and update the SQL server. Vuejs is used to build dynamic webpages that allow for multiple functions on a single page.
5. Design Evolution
   1. After implementing the web application it can be expanded to add more state legislatures. In order build this it would require additional tables for each state and separation between states within the search results.
   2. Another function would allow for a timeline for the bills to give a historical record of actions taken.