REQUIREMENTS DOCUMENTATION

Project : IRI 1 -- Hydroviz

Team Members : Christopher Boudreaux, Tory Hebert, Phillip Huval, Jacob LeCoq, and Andrew Wise

Version Published : October 11, 2015

ABSTRACT

Our team has been contracted to build an extension for the Hydroviz website. This extension will allow the visualization of data in the form of a graph. As the project goes on, more information will be given, and more requests will be made from the client, where we will even further extend the functionality of the product.

The current outline of the extension will allow the selection of different options via dropdown menus that reflect the scenario related to the relevant information, the extent of the data to be plotted, the date range that the data will be pulled from, the independent variable of the graph, and the display type of the data. This data will be used in a call to an api provided by our contact. The results of the call will provide use with a Json object containing the data needed to populate a graph. This graph will be created using a web application called HighCharts.

This extension, after completion, will be accessible from the main Hydroviz website, and anyone interested will be able to use it to pull a graph with their requested information.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

Introduction to the entire document

- The pages of this document contain:
    - Use case models for the functional requirements for the project, which will be represented by both visually and with textual descriptions
    - Rationale for the use case models
    - Non-functional requirements
    - Evidence that the requirements have been placed under configuration management.

Purpose and scope of the document

- The scope of this document is merely the requirements and use case models.
- The purpose of this document is to ensure that the project members know the use cases and use scenarios before they start coding, so that they can code according to the scenarios.

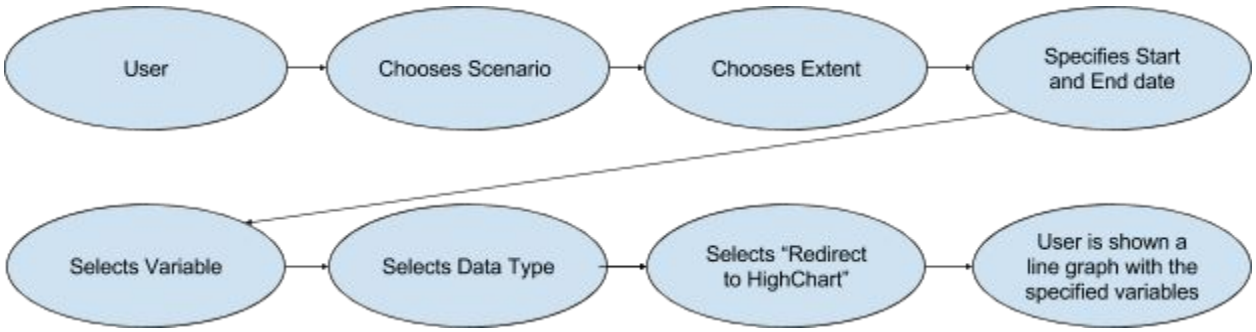
Structure of the document

- This document is divided into 4 parts:
    - A use case model for functional requirements
    - The rationale for the use case models
    - A list of non-functional requirements
    - Evidence the requirements have been placed under configuration management.
- The use case model for functional requirements is divided into two subsections: graphic use case model and textual description. Each use case will contain the name, participating actors, entry conditions, normal flow of events, exit conditions, exceptions, and special requirements.

USE CASE MODEL FOR FUNCTIONAL REQUIREMENTS

The format of this use case model is adapted from [1].

Figure 1 : Sunny Day Use Case



TEXTUAL DESCRIPTION

- Participating Actors:
    - The user

- Entry Conditions:
    - The user must be on the HydroViz website and must have selected a cell in interactive mode.

TEXTUAL DESCRIPTION (Cont.)

- Normal Flow of Events:
    - The user chooses a scenario from the drop down box. After this action, a new drop down box appears that's labeled "Extent". The user then selects an extent from the drop down box. The user then selects a start and end date for the data, as well as the variable that they want displayed in the HighChart. Finally, the user selects the data type and then selects "Redirect to High Chart", and then is shown a line graph with the behavior of the graph depending on their selections.

- Exit Conditions:
    - The user closes the HighChart page

- Exceptions:
    - The HydroViz database is down or the HighChart website is down. If this happens, the user is given an error message and must wait until the database or website comes back online.

RATIONALE FOR THE USE CASE MODEL

- Reason for ‘Entry Condition’:
    - The owner is HydroViz, so their website will be hosting the project. The reason for the user having to select a cell in interactive mode, is because our project displays a geographical graph for a user in a selected location.

- Reason for ‘Normal Flow of Events’:
    - The reasons for our normal flow of events is that in-order for interactive graph to display correct geographical data for a select location, we must first gather all the information needed.

- Reason for ‘Exit Conditions’:
    - The reason for the exit condition on our project is that after the user is done with whatever use they had for the graph, all they have to do is close the page opened up.

- Reason for ‘Exceptions’:
    - The reason for our exceptions would be that even though we hate running into issues, no database or website is up 100 percent of the time. We must account for the downtime of servers hosting the website and/or databases.

NON-FUNCTIONAL REQUIREMENTS

Product Requirements

- The HydroViz interface shall be available to all clients whenever they want to use it, whether it is during normal working hours or not.
- The event of the interface going down is extremely rare.
- The performance time bottleneck will be communication with the database and the highcharts library.
- All other computations will need to be nearly instantaneous, therefore we need the software to be relatively efficient because the response to a request should not leave the user waiting for an unreasonable amount of time.

Organizational Requirements

- Any user with an internet connection can access the HydroViz interface, therefore the product we are developing needs to work as long as Hydroviz is currently up.

External Requirements

- The interaction with the HydroViz API that will provide the data needed to create the charts.
- The project members will develop code that is of the highest professional standard.
- The project members will be mindful of intellectual property and copyrights when developing code.

REFERENCES

[1] Darren Levy, “User Case Examples -- Effective Samples and Tips”, http://www.gatherspace.com/static/use_case_example.html, 2015.