

Product Requirements for Data Dwarf

Prepared by

Group Name: Mercury Squad
Sean Spearman seanmspeaman@gmail.com
Cody Brown codybrwn551@aol.com
Ray Smets rayjsmets@gmail.com
Aimee Galang aimeegalang@gmail.com
Tim Shen timshen1116@gmail.com



Revisions

Version	Primary Author(s)	Description of the Version	Date Completed
1.0	Sean Spearman, Cody Brown, Ray Smets, Aimee Galang, Tim Shen	Initial Version	01/29/2015

1 Introduction

1.1 Document Purpose

The purpose of this document is to specify the requirements for the web application *Data Dwarf*. This document is intended to be used as a reference for the development team as they develop the initial prototype of the application.

1.2 Product Scope

Given the amount of structured and unstructured data available today, there is a huge amount of potential knowledge waiting to be analyzed. *Data Dwarf* offers a fast solution to improve the analysis of data by allowing a user to dynamically compare their own data set to a vast set of data stored in our server, removing the need for users to mine, analyze, and visualize arbitrary, but potentially insightful, data. *Data Dwarf* intends to offer a user multiple sets of data that have the strongest correlation to the user's input data. It will additionally allow users to select these data sets that have already been gathered and stored on a server for custom comparison to their input data.

The user will then have the option to choose from among multiple styles of graphs or figures, so that they can find a relevant way of displaying their

information. The user will be able to select given correlated data sets to be displayed on their selected graph as well.

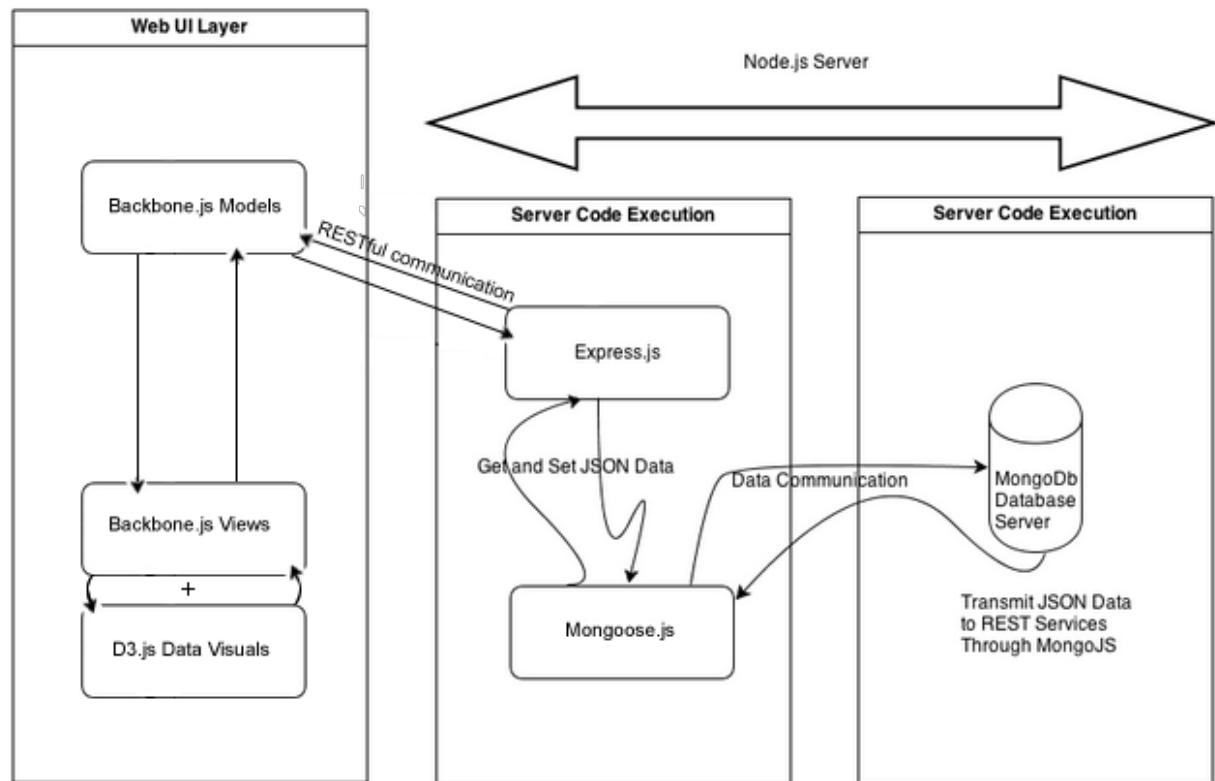
1.3 Intended Audience and Document Overview

Data Dwarf is an open source project. However, those who will likely find the most use out of it will be those who are studying or doing research on correlations between different events. For example, if one were studying the change in the amount of pollution over time, *Data Dwarf* could supply any data sets that show similar changes over the same period of time.

2 Glossary of Terms

Angular.js	Framework used to simplify development and testing web applications
API	Application program interface (API) is a set of routines, protocols, and tools for building software applications
Correlation	The mutual relation of mathematical or statistical variables which tend to occur together
Express.js	Node.js's web application framework
Heroku	Cloud computing services that provides a platform to develop, run and manage Web applications.
HTTP	The Hypertext Transfer Protocol (HTTP) is an application protocol and the foundation of data communication for the World Wide Web
MongoDB	Database designed to store document-oriented information
Mongoose.js	MongoDB wrapper for Node
Node.js	JavaScript runtime environment for server-side and networking applications
Open Source Project	Development model promoting universal access to a product's design and its subsequent improvements by anyone
R + OpenCPU	HTTP API for data analysis

3 System Architecture



4 Requirements

4.1 User Cases

Use Case 1: Generate Charts

Actors: User

Precondition: User is at the front page of the website and none of the data sets are selected

Flow of Events:

1. User clicks and selects a data set on the left toolbar
2. Charts with the data are generated for the user to select

Use Case 2: User Uploads Data

Actors: User

Precondition: User is at the front page and there is an empty column to put data in

Flow of Events:

- *Basic Path*

1. User copies and pastes data from a spreadsheet (or manually enters in data)
 2. User titles their data set
 3. User hits "save"
 4. Data set saved and added to the list of other data sets from which the user can now select
- *Alternative Path*
 1. User's list of data points is too long for column, but user tries to paste it into the empty column anyways
 2. Column expands to fit all data
 3. Scroll bar appears to view all data entered

Use Case 3: User chooses length of each selected data set

Actors: User

Precondition: Each data set is customizable/editable

Flow of Events:

1. User selects first data set
2. Data set is added to graph
3. "Edit" button appears next to selected data set
4. User selects "Edit"
5. new menu appears
6. one of the options in the "Edit" menu allows the user to enter in the first and last points on the x-axis for that particular data set
7. User exits "Edit" menu
8. the selected data set scales appropriately on the graph

Use Case 4: User adjusts multiple selected data sets at a time

Actors: User

Precondition: Each data set has a check box next to it that the User can check and uncheck somewhere at the top there's an "Edit" menu which allows data set customization.

Flow of Events:

1. User selects multiple data sets
2. User clicks "graph" button
3. selected data sets all scale to fit on the same graph
4. the only check boxes that can be selected are those of the selected data sets
5. User clicks on (adds check to) one or more check boxes
6. User clicks on the "Edit" button somewhere at the top of the screen
7. "Edit" menu appears
8. User changes of the "Edit" menu options
9. User exits "Edit" menu

10. changes get applied to all checked data sets and the graph updates accordingly

Use Case 5: Query for data sets

Actors: User

Precondition: a text window is available to type in possible names of data sets
not all data sets are shown on front page of website

Flow of Events:

1. User selects the text window
2. User begins typing the name of a possible data set
3. new window appears just below the text window and it has relevant available titles of data sets
4. User continues typing and the relevant titles window adjusts accordingly (kind of like when you type something into Google)
5. User finishes typing and clicks "enter"
6. all possible matches appear in a temporary box on the web page (it may be scrollable)
7. User selects one of the new data sets
8. new data set gets added to the main list of data sets on the web page

4.2 User Stories

1. As a user, I can access the website from a desktop
2. As a user, I can change which data sets are shown on the charts
3. As a user, I can change which type of chart the data is shown on
4. As a user, I can filter selected data sets
5. As a user, I can search through available data sets
6. As a user, I can change the scope of data sets
7. As a user, I can upload my own data
8. As a user, I can compare my data with website's data
9. As a user, I can use previously updated data if I have registered and logged in
10. As a user, I can log in to site
11. As a user, I can log out of the site
12. As a user, I can register with the site
13. As a user, I can access the website from a phone or tablet

Appendix: Technologies Employed

- MongoDB
- Express.js
- Mongoose.js
- Backbone.js

- Node.js
- R + OpenCPU
- Heroku