Software Requirements Specification

Interactive Timeline visualisation

CMPT 371- Team 2

Table of Contents

Executive Summary:

‘*Timelines*’ is an interactive visual environment intended to visually communicate moderate to large quantities of temporal data that relate to different entities. The user is able to specify and customize individual data sources, as well as interact with the timeline itself.

This document outlines the requirements for the successful implementation of a web-based system where the user is able to visualize, interact with, and process large volumes of data. The overall goal is to make an efficient, and aesthetically pleasing piece software based on stakeholder specifications, that the user can take effective advantage of.

1. Introduction

1.1 Purpose

The purpose of this document is to provide a comprehensive outline of the software specifications. This document entails specific attributes and functions that will be incorporated into the program, following the completion of the development cycle. This document will be prone to change, based on any additional information from the stakeholder(s) that implies functional and/or structural changes to the existing specifications.

1.2 Intended Audience

The target audience of this software is users with large amounts of temporal data, looking to either explore the data through a web interface, or to present large datasets in an informative way. The typical end user is assumed to have general computing skills, and some experience dealing with databases and editing .csv files, but is not expected to be a programmer, or even necessarily a data-scientist. The project aims to meet the needs of the end user by allowing a customizable, and clean interface that facilitates the visualisation of large data sets.

1.3 Project Scope

The project aims to implement a fully functional system, fulfilling the requirements laid out in the rest of this document by December 5th, 2019. The document outlines various features, and use-cases of the software, which are broken down into “required”, and “nice to have”. Both of these use-case categories are subject to change as per stakeholder request. Acceptance of the project into a finished state is outlined in the nonfunctional requirements part of the document. These acceptance criteria are also subject to change, as per the stakeholder request, but in general are derived from the functional components of the system. A brief description of features that will be considered entirely out-of-scope for the project are as follows.

- Database support
- Secure login
- Mobile support

Although these are out of scope for the current final deadline, they could come into scope in future iterations of the project.

1.4 Use-Cases

Required Use-Cases:

1. **The ability to upload data**
   a. This will allow the user to select a local file and have access to its columns/rows for visualisation. The user will be able to have multiple files associated with a workspace, and multiple rows associated with a specific timeline.

2. **The ability to create a timeline associated with uploaded data**
   a. At any point, the user should be able to create a new, timeline in the workspace, and associate any existing rows to that timeline.

3. **The ability to explore uploaded data via operations on a specific timeline**
   a. On a specific timeline, the user will have the ability to pan across the data, zoom into specific regions, and zoom out to large regions of the data set.
   b. Users can select or hover over individual points in their datasets to reveal more information about the data point.

4. **The ability to choose how a specific row is represented in a given timeline**
   a. The user will have the ability to choose the specific style of data representation for an associated column on a timeline
   b. The user has the ability to choose the color and legend style that a specific data-set is using.
   c. The user has the ability to either turn labels for data on or off. Or the ability to toggle this only for a specific subset of the selected data.

5. **The ability to select which data is displayed on a specific timeline.**
   a. The user has the ability to choose which data is displayed on a specific timeline, allowing them to declutter their workspace when dealing with very dense data.

6. **The ability to have multiple timelines, and switch between them in a given workspace.**
   a. The user has the ability to add multiple timelines to the workspace in order to compare and contrast datasets.
   b. The user will also have the ability to switch between different timelines within this workspace.

7. **The ability to export a workspace as a configuration file, allowing the user to pick up where they left off in a previous session.**
    a. The user can export a config. file that will store the state of their current working session.
    b. This config. file can be uploaded again and the user will be able to pick up from where they left off.

8. **The ability to choose how densely labels are displayed.**
    a. On a given timeline, the user has the ability to choose the granularity of label visibility. For example, in very dense timelines the user may only want to see a label every few thousand points or more (allowing them to actually read the labels).

9. **The ability to scale sets of data along their Y-axis**
    a. With a specific row on the timeline, the user will have the ability to scale every point by some factor. This allows comparison of radically different, but related quanta (for example GDP vs Percent Growth).

10. **The ability to select subsets of the data**
    a. A tool will exist for the user to select large subsets of data across the entire timeline, rather than just an individual point. This tool essentially operates as a typical marquee tool from most image editing software.
    b. The user will also have the ability to select individual (singleton) points of data along the timeline.

11. **The ability to view a summary of selected data, either an individual point on the timeline or a selection of many points.**
    a. When the user has a singular point selected, they are given a contextual menu showing information about that specific point, and how the point relates to the rest of the dataset as a whole.
    b. When the user has multiple points selected, they are given a contextual menu showing information about the entire selection, and how that selection relates to the rest of the dataset.

12. **The ability to see a legend which shows the user what data they're looking at**
    a. A legend will be visible (either at all times, or through a context menu) which will inform the user of what each dataset on a timeline represents.

“Nice to have” Use-cases:

1. **The ability to run analytical tools on datasets:**
    a. Things like line-of-best-fit, interpolating polynomials, splines, and other analytical features to help facilitate user analysis of data.

2. **The ability to display multiple timelines at the same time**
    a. The user can show, and compare multiple timelines at the same time within the same workspace.

3. **The maximization of screen real-estate**
    a. The user can hide any contextual menus and toolbars to maximize their ability to view data along the timeline.

4. **Support for specific hardware**
    a. The user can use peripherals like mousewheels, or trackpad gestures as default ways of interacting with the data (rather than needing to select specific tools from a toolbar).

2. Overall Description

2.1 Product Perspective

This website is an open-source, data-visualisation tool that allows users to look at large temporal datasets in an effective, and elegant way. The tool addresses the issue of big-data visualisation by allowing the end user full control over what is seen, and what isn't, giving it the power to be used either as an analytical tool, or as an aid in presentations.

2.2 Product Features

Major Features:

- **Well defined default behavior when importing data:** allowing users to get expected results when uploading a new file for the first time.

- **Full user control over what is displayed:** which colors are used, which data is visible, what type of representation a data-set has, what scale data is displayed at, and how labels for datasets are displayed.

- **Support for extremely large datasets:** along with basic analytic tools to help the user understand the relationship between events, points in time, and other datasets.

- **Web-driven design:** Allows the software to be operating system agnostic, allowing for support across all major platforms.

- **Locally stored datasets, locally performed analysis:** The system doesn't store any of your personal data server side. Files are loaded locally, and workspaces are saved to your machine so you can pick up where you left off without worrying about someone else looking at your data.

2.3 Operating Environment

This web application is intended to run on a laptop or desktop computer with any of the major web browsers: Chrome, Firefox, Opera, Safari, Edge.

2.4 Design and Implementation Constraints

The system will have little reliance on outside resources and should remain functional for the foreseeable future. Performance considerations should be given a high priority as the execution speed and memory requirements of javascript can be a limiting factor for such applications, especially if the dataset is particularly large.

2.5 Assumptions and Dependencies

- The system only needs to support .csv files at the given moment.
- No major feature of Javascript, HTML, or CSS becomes deprecated in such a way that future web browsers drop all support, similar to Adobe Flash, albeit, this is highly unlikely in the foreseeable future.

3. External Interface Requirements

3.1 User Interfaces

The general user interface will display a timeline for the user, allowing them to pan, zoom in, and zoom out over a temporal data set. Note that none of these buttons speak specifically to where they will exist on an interface, some may appear in context menus, others may be visible at all times, but all of these buttons will be required.

1. **An upload button**
    a. This will allow the user to directly upload a .csv for visualisation
2. **An add timeline button**
    a. This will allow a user to add a timeline to a workspace
3. **A hide timeline button**
    a. Allows the user to collapse the timeline they're currently working in
4. **A save button**
    a. Saves a config file
5. **A toolbar that shows the basic exploration tools**
    a. Zoom in
    b. Zoom out
    c. Pan
    d. Selection
6. **Labels for data**
    a. Strings for discrete events
    b. Values for quantified events
    c. Time-stamps for any event
7. **A legend which specifies which data-set is bound to a specific representation**
    a. Shows which color is associated with which row
8. **Mouse-hovering tool-tips**
    a. To provide additional details about the data point, for instance in temporal data this would include the time and date of the event
9. **Axis Information**
    a. Shows the current scale for the data along both the x and y axis
10. **Selected data information**
    a. Shows information about the currently selected data. Also shows information about how that selected data relates to other data in the dataset.

3.2 Hardware Interfaces

The application will be handled entirely by standard computer interface devices, specifically by keyboard and mouse (or trackpad).

4. Nonfunctional Requirements

4.1 Performance Requirements

The system will be able to handle large amounts of data while remaining reliably performant. With extremely large datasets, some frame rate drops may occur during the initial parsing of the data. The general goal will remain to display it in a smooth and seamless way, even when showing many discrete points.

4.2 Safety Requirements

The primary safety issue for this website is server/host failure. No user data is stored so no data will need to be recovered. Multiple possible web hosts will be identified and ranked in terms of performance per dollar, should the primary host experience a failure, the next best host will be used. Further, a CDN like Cloudflare can be used to cache copies of the website around the world, given that this website will have no backend, it's possible that a properly configured CDN would allow the website to remain completely functional during a primary host failure.

4.3 Security Requirements

Since user data will not be stored, there are no immediate security concerns other than making sure our web certificates are up to date.

4.4 Reliability Requirements

High reliability is required for the system, both in terms of robustness, and availability. Firstly, the system should be robust with respect to bad, corrupted, or improperly formatted data. None of these should cause any system failure, other than the intended consequence of that data not being shown.

A failure rate of 1 out of 1000 user sessions is our current aim, where failure is defined as the end-user being unable to use the system. During initial phases of implementation, failure rates may be higher, but regression of failure rate should be seen in all cases as a failed build.