Software Requirements Specification

for

WAVED

Version 2.0

Prepared By:

Sean Bluestein, Kristian Calhoun, Keith Horrocks, Steven Nguyen, Hannah Pinkos

Advisor:

Kurt Schmidt

Stakeholder:

Climate Central

Table of Contents

1. Document History

Name	Date	Reason	Version
Kristian Calhoun, Sean Bluestein, Hannah Pinkos, Steve Nguyen, Keith Horrocks	January 7, 2014	Initial Draft	1.0
Kristian Calhoun, Sean Bluestein, Hannah Pinkos, Steve Nguyen, Keith Horrocks	February 4, 2014	Updated requirements to match new paper prototype	2.0

2. Introduction

2.1 Purpose

This document provides the requirement specifications for Web App for Visualizing Environmental Data (WAVED). It specifies user interface attributes, functional and nonfunctional requirements, and long-term ideas for the evolution of the system.

2.2 Overview

Climate Central performs scientific research to collect multi-coordinate data on climate change. Climate Central creates interactive maps and other graphical widgets to display this data on their website for public consumption. For example, one such demonstration colors each state in the United States based on its recorded change in temperature over time. Clicking on a state displays a line graph showing the temperature data for that particular state. Currently, Climate Central develops each interactive visualization from scratch. This process is time and labor intensive and does not provide any mechanism for reusing common elements across visualizations.

WAVED is a web-based development environment that enables users of all technical skill levels to create interactive visualizations via a graphical user interface. WAVED allows end-users to upload data and decide how to visualize it on a map or graph. Users can add other widgets, such as buttons or sliders, to trigger user-defined events that manipulate how the data is displayed. As users build projects, they can interact with the visualization via a live preview. Once finished, projects can be exported as a collection of automatically generated HTML, JavaScript, and CSS files that can be embedded in a web page.

2.3 Scope

This document describes the software requirements for the WAVED application. The initial release will be tailored specifically to Climate Central’s immediate requirements. Developers are expected to use this document as a guide for the creation of the system and testers will use it as a basis for validating the release of the product. These requirements are also intended for end-users interested in the application’s functionality and future evolution.

2.4 Definitions

Action - A process that changes the properties of a widget, element or data set via an event

Binding Data - Making an explicit association between a data set and a widget, property, or event

Data Set - A data source or data subset

Data Source - A collection of data, such as the contents of a CSV or JSON file

Data Subset - The resulting portion of a data source that meets some specified criteria.

Element - An accessory component of a widget. The creation of elements requires a parent widget to exist; elements cannot stand on their own. For example, glyphs are elements of a map widget.

Event - Something that occurs (click, hover, etc.) to trigger an action

Glyph - A marker indicated by some shape or icon

Project - The context in which a user works to create a single visualization using the WAVED application

Property - Defines a value used for a widget attribute, such as width, height, color, name, etc.

Subwidget - A widget that is the child of another widget

Target - Part of a widget, element, or data set affected by an action

Trigger - Part of a widget or element that causes an event to fire (e.g. a component that can be clicked, hovered, etc.)

Widget - A top level component that has properties, subwidgets, elements, and bound data

Workspace - The interactive area of the WAVED user interface that displays the visualization as it is assembled from widgets

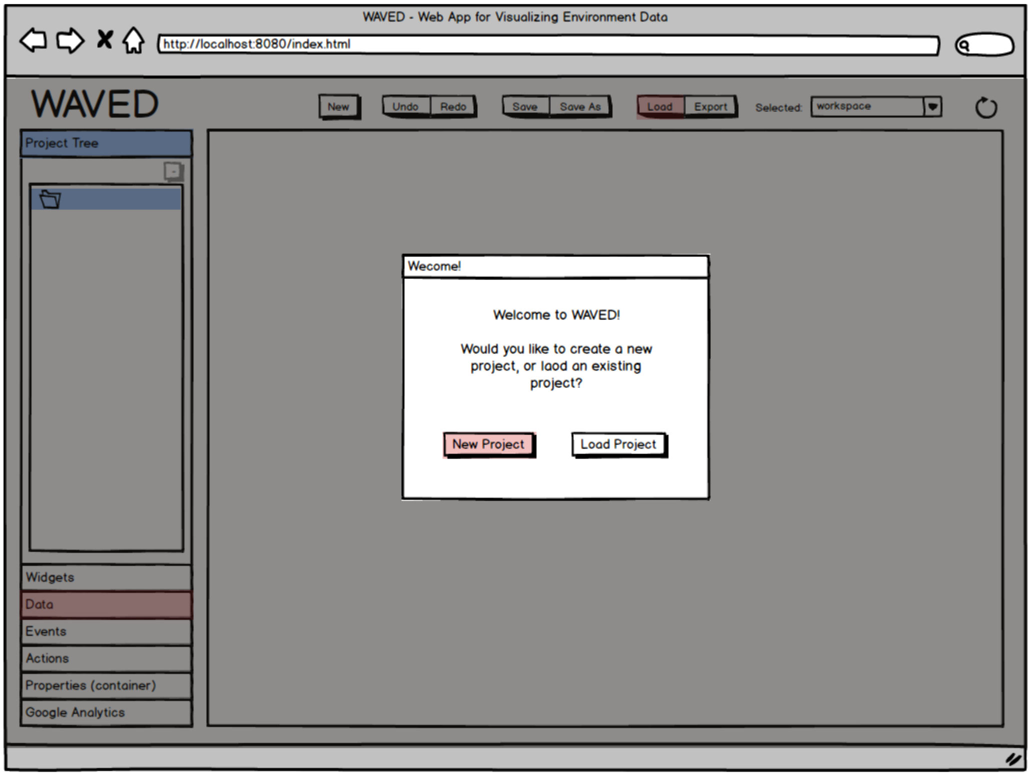
2.5 Requirements Apportioning

Priority Level	**Description**
1	Priority 1 items have the highest level of importance and are necessary for the usability of the product. The items labeled Priority 1 must be completed and verified prior to the release of the application.
2	Priority 2 items are not crucial to the usability of the application, but would enhance the user experience and are expected to be implemented in the next release of the application. Priority 2 items not completed prior to release would not inhibit functionality of the release version of the application.
3	Priority 3 items are not crucial to the usability of the application and are not within the current scope of the system. These items are expected to change with future development.

3. User Interface

3.1 Overview

The WAVED user interface is a graphical, interactive, integrated development environment. The development environment runs within a web browser, using a keyboard and pointing device for input. Upon navigating to the web address of the WAVED application, users are presented with a welcome dialog that presents two options: Create a New Project or Load a Project (Figure 1). If the user selects to create a new project, they are prompted to enter a name for the project and then be presented with a blank project (Figure 2). If the user selects to load an existing project, he or she is presented with a list of previously saved project names from which to pick. All possible methods for creating a project within WAVED are shown in the activity diagram in Figure 3.



WAVED - Web App for Visualizing Environment Data

http://localhost:8080/index.html

Selected: workspace

Wecome!

Welcome to WAVED!

Would you like to create a new project, or load an existing project?

Figure 1. The welcome splash screen

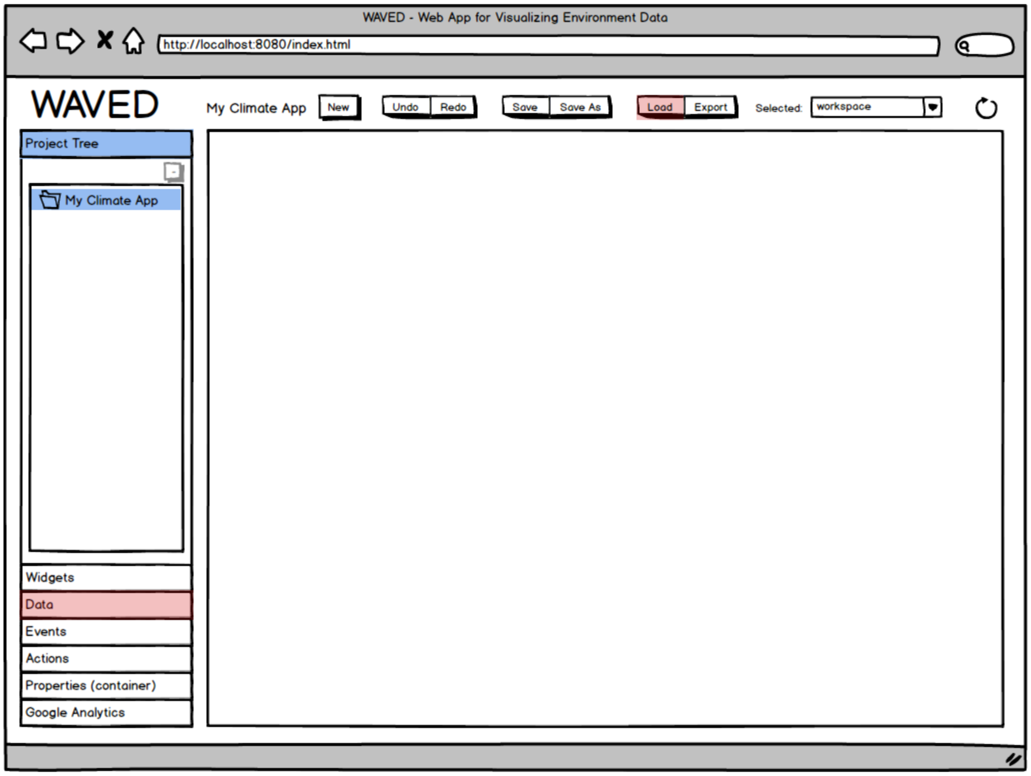


Figure 2. The WAVED interface after creating a new project

The main interface has of a series of collapsible panels on the left hand side, including the Project Tree Panel, Widgets Panel, Data Panel, Events Panel, Actions Panel, Properties Panel, and Google Analytics Panel. A menu at the top of the application displays buttons to create a new project, undo and redo changes, save and load the application, export a project, and refresh the workspace. The combobox in the menu allows the user to select a widget or element that has been added to the project. Below the menu and to right of the collapsible panels is the workspace area.

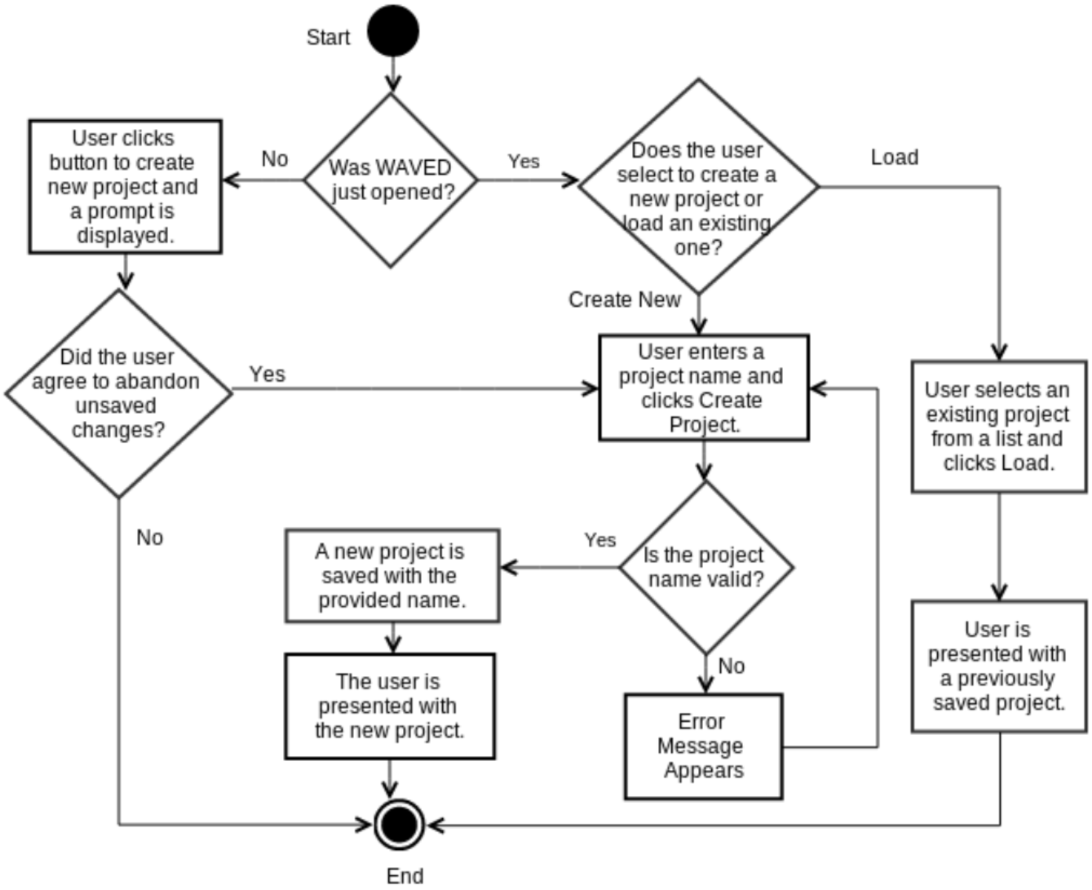

Start
Was WAVED just opened?
No
Yes
User clicks button to create new project and a prompt is displayed.
Does the user select to create a new project or load an existing one?
Load
Create New
Did the user agree to abandon unsaved changes?
Yes
User enters a project name and clicks Create Project.
User selects an existing project from a list and clicks Load.
No
Yes
A new project is saved with the provided name.
Is the project name valid?
No
User is presented with a previously saved project.
The user is presented with the new project.
Error Message Appears
End

Figure 3. An activity diagram for creating a new WAVED project

3.2 Data Panel

Opening the Data Panel allows the user to view a list of data sources that have been added to the project (Figure 4). The user can click on the plus (+) button in the source files section of the Data Panel to open a dialog and import local data files into the currently opened WAVED project. (Figure 5). The user can then remove data sources that have been added to the project by selecting a data source from the source files list and clicking the minus (-) button.

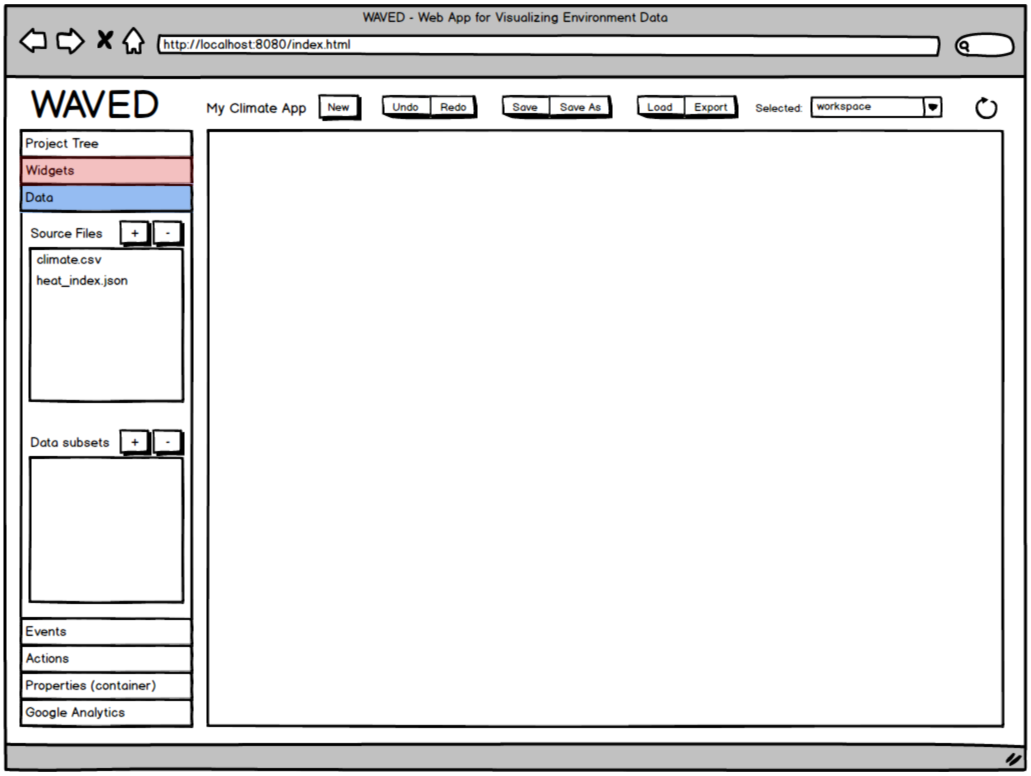


Figure 4. A populated source files list in the Data Panel

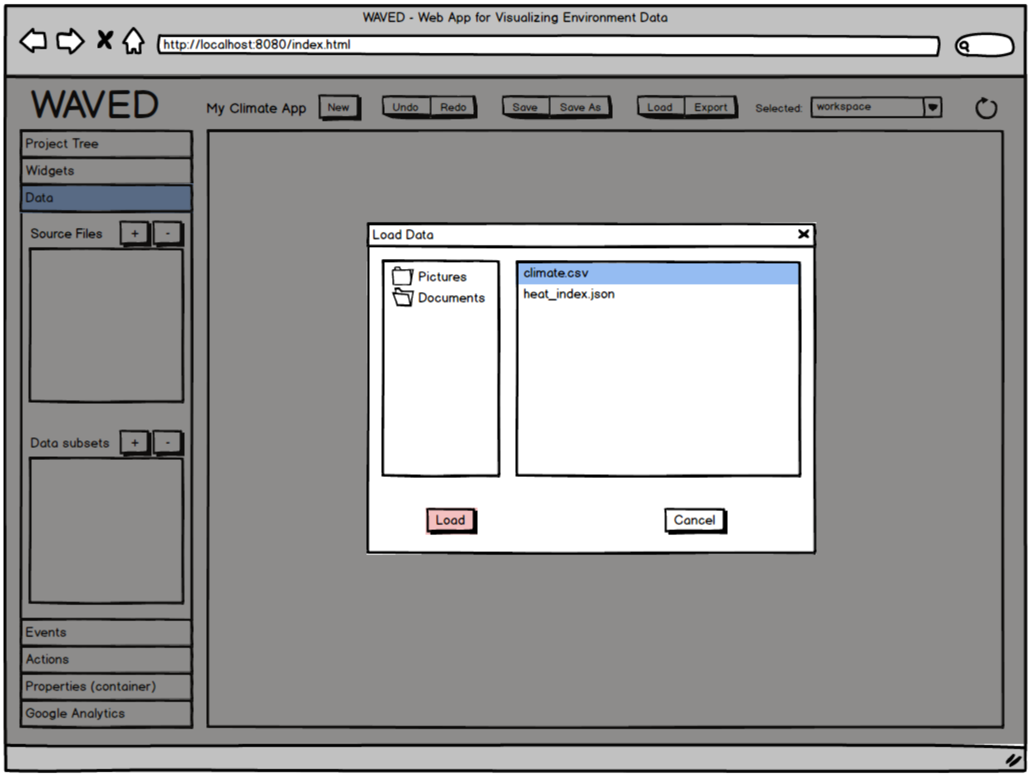
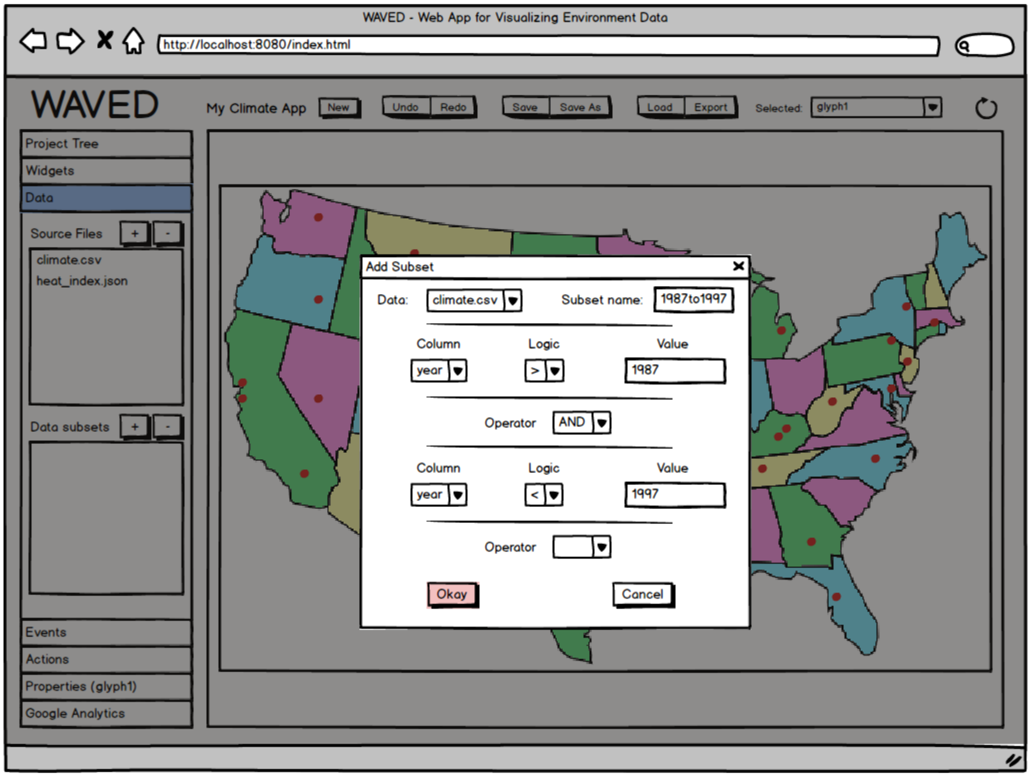


Figure 5. A dialog for importing local data files into the current project

After a data source has been added to a project, it can be used to create data subsets that contain only parts of the original data matching some specified criteria. Clicking the plus (+) button in the data subsets section of the Data Panel opens a dialog for creating a new data subset. The data subset creation dialog provides fields for the user to choose a data source file, enter a name for the new data subset, and construct a query statement by selecting options from the provided comboboxes (Figure 6). Once a data subset is created, its name appears in the Data Subsets list of the Data Panel and it can be bound to widgets and elements just like any other data source.



Data: climate.csv

Subset name: 1987to1997

Column: year | Logic: > | Value: 1987

Operator: AND

Column: year | Logic: < | Value: 1997

Operator:

Figure 6. Data Subset Creation Dialog

3.3 Widgets Panel

Opening the Widgets Panel provides the user with a selection of available widgets to add to the project (Figure 7). Selecting a widget from this panel adds it to the project workspace and automatically opens the Properties Panel, which becomes populated with properties associated with the added widget (Figure 8). The added widget can now be selected from the Project Tree Panel and the widget selection combobox in the menu.

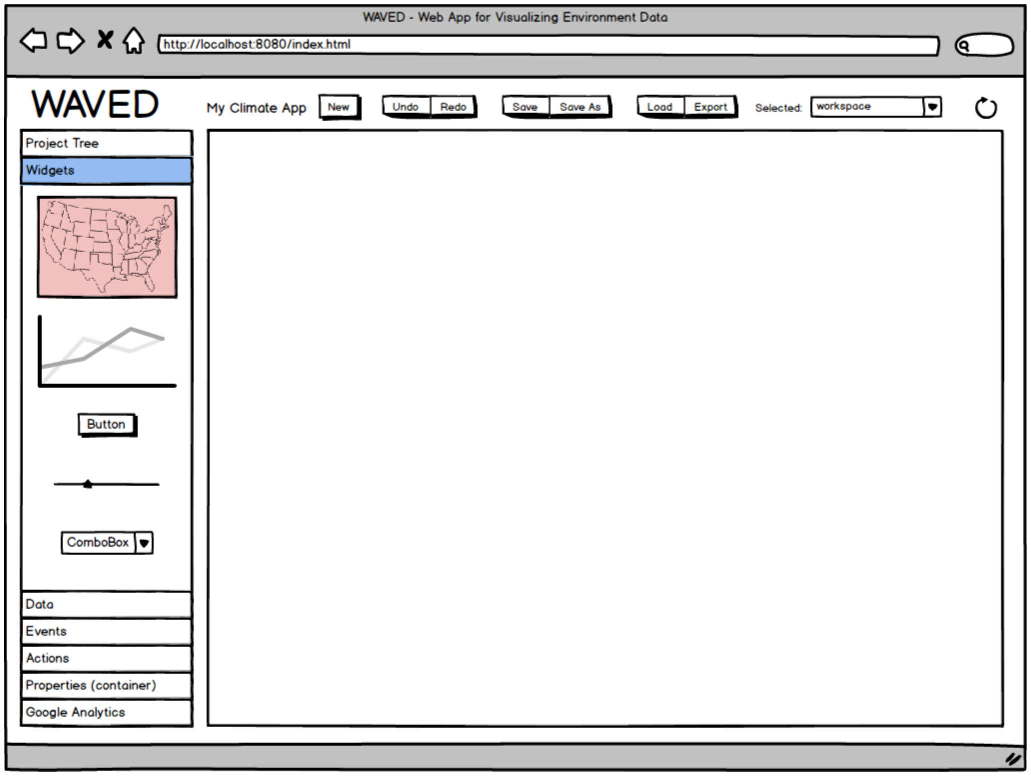
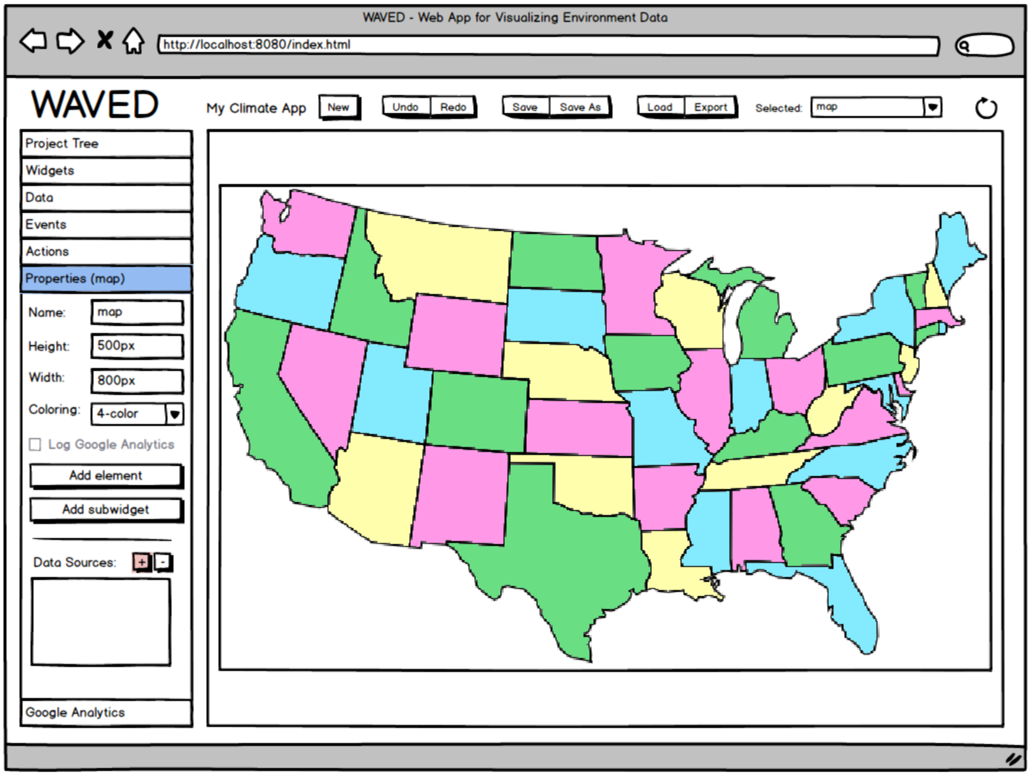


Figure 7. Widget Selection Panel

3.4 Properties Panel

The Properties Panel contains widget-specific input fields for changing widget properties and binding data sets to the currently selected widget (Figure 8). If the selected widget can have subwidgets or elements added to it, buttons for those options are also available in the Properties Panel. To bind data to a widget, the user clicks the plus (+) button in the Data Sources section of the Properties Panel to display a dialog containing the data sources available to be bound (Figure 9).



Selected: map

Name: map

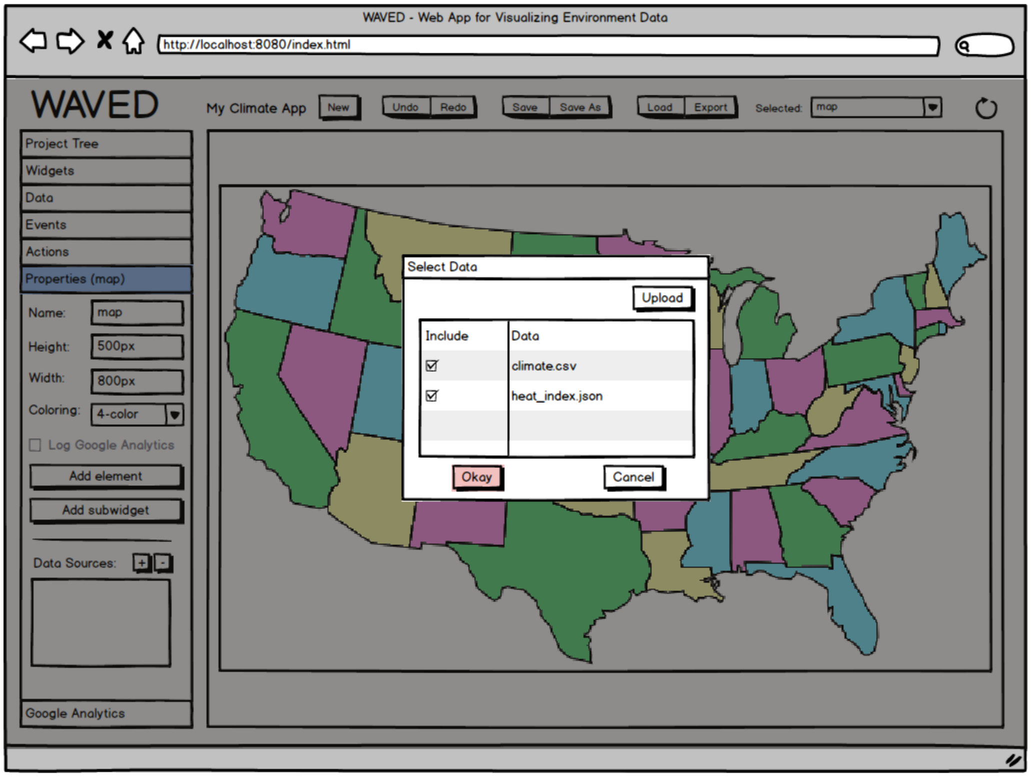
Height: 500px

Width: 800px

Coloring: 4-color

□ Log Google Analytics

Figure 8. A widget's Properties Panel



Selected: map

Name: map

Height: 500px

Width: 800px

Coloring: 4-color

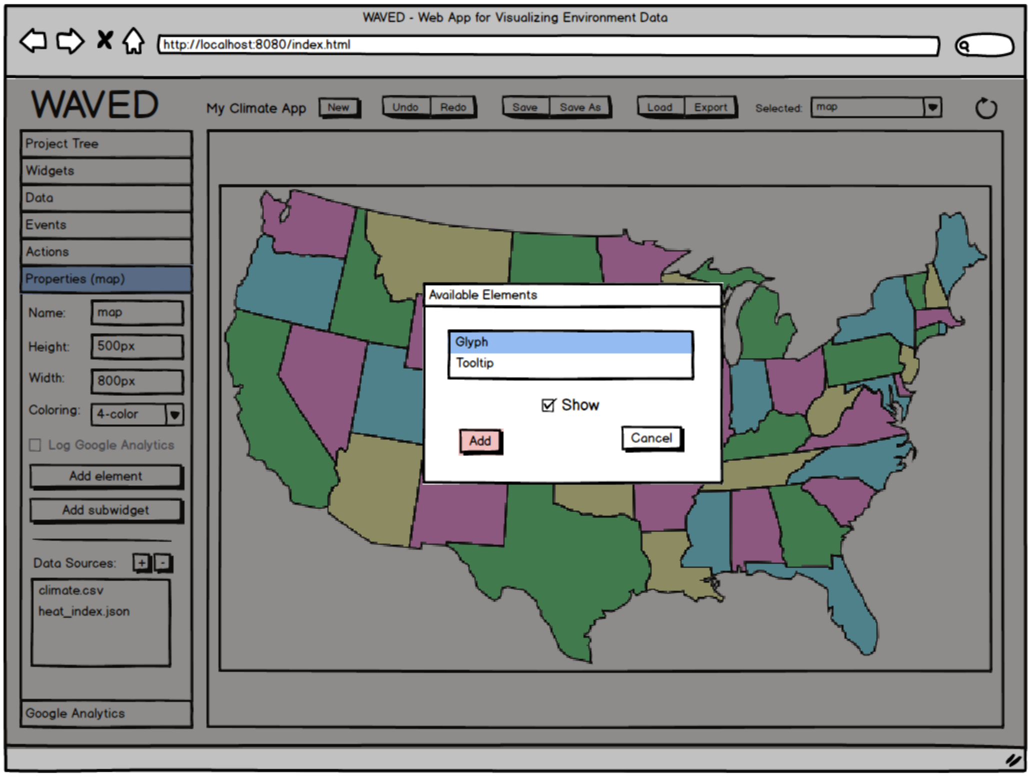
□ Log Google Analytics

Select Data

Include	Data
☑	climate.csv
☑	heat_index.json

Figure 9. A dialog for binding data to a widget

Clicking the Add Element button brings up a dialog that allows the user to select which type of element to add. In the example shown in Figure 10, the element selected to be added is a glyph. Once an element has been added to a widget, the element is selectable from the combobox in the top menu and the Properties Panel populates to allow for user manipulation of the element's properties (Figure 11).



Selected: map

Name: map

Height: 500px

Width: 800px

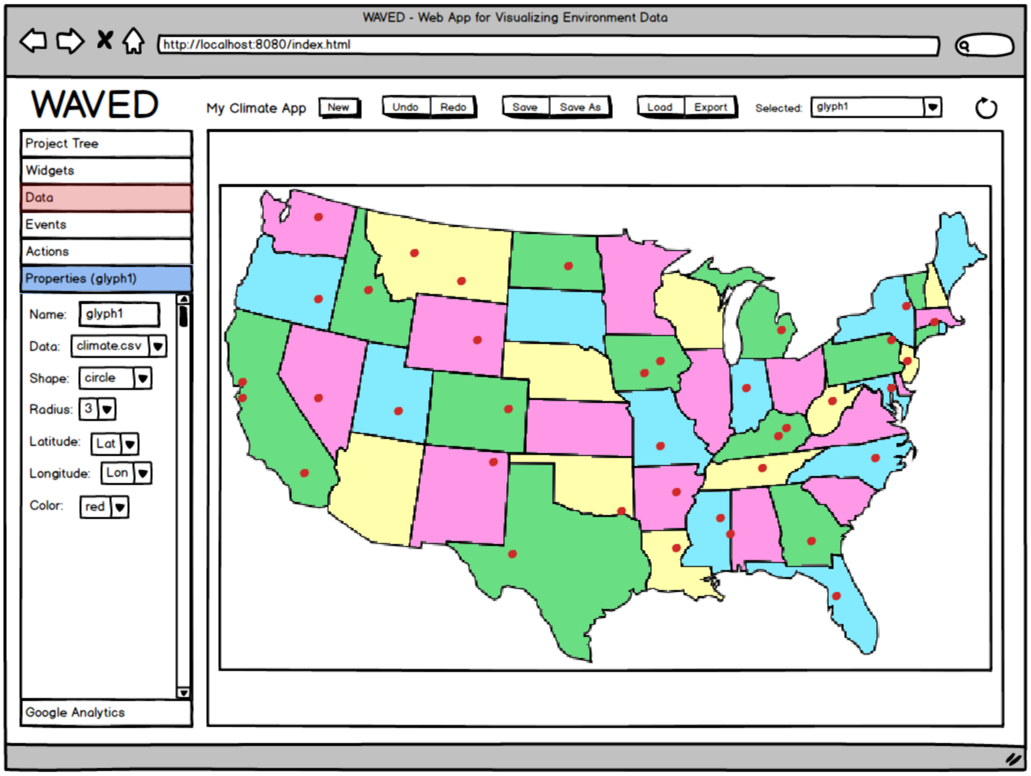
Coloring: 4-color

☐ Log Google Analytics

climate.csv
heat_index.json

☑ Show

Figure 10. Add element dialog

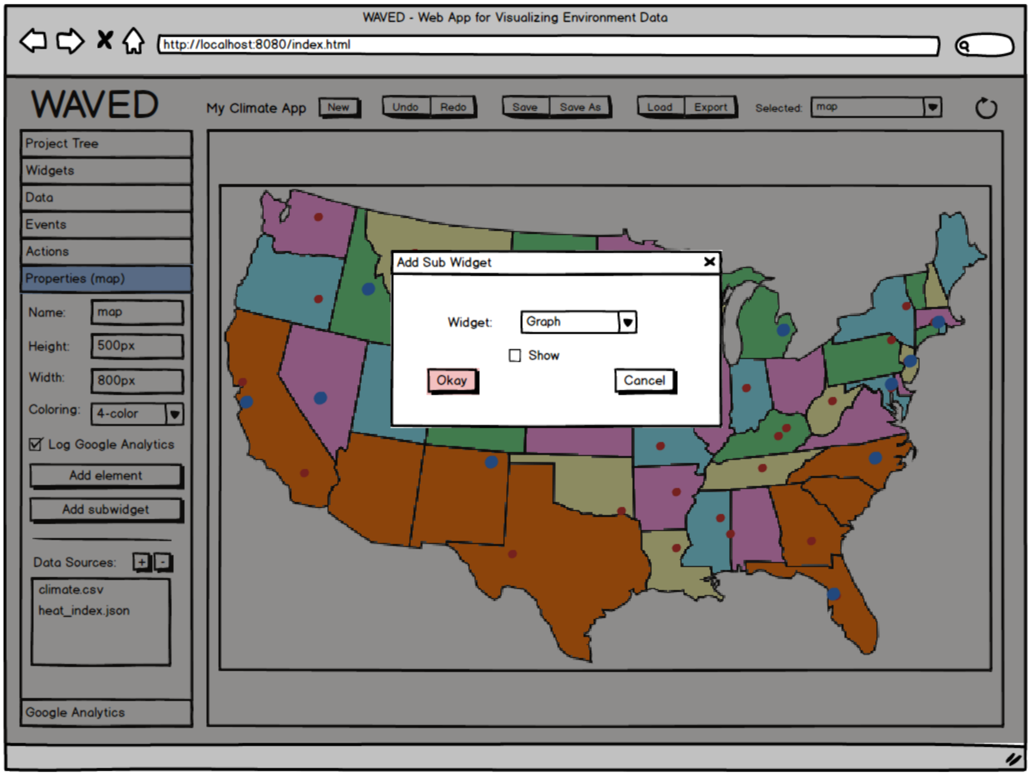


Selected: glyph1

Name: glyph1
Data: climate.csv
Shape: circle
Radius: 3
Latitude: Lat
Longitude: Lon
Color: red

Figure 11. Glyph Properties Panel and glyphs on a U.S. map widget

Subwidgets can be added as a child of the currently selected widget by clicking on the Add Subwidget button in a widget's Properties Panel. Clicking this button displays the subwidget dialog, which allows the user to select a widget type. Figure 12 demonstrates adding a graph subwidget to a U.S. map widget. Subwidgets are hidden by default in the workspace area if the "Show" checkbox on the add subwidget dialog is left unchecked. Selecting a subwidget via the Project Tree Panel or combobox in menu displays the subwidgets properties in the Properties Panel even if the subwidget is hidden (Figure 13).



Selected: map

Name: map

Height: 500px

Width: 800px

Coloring: 4-color

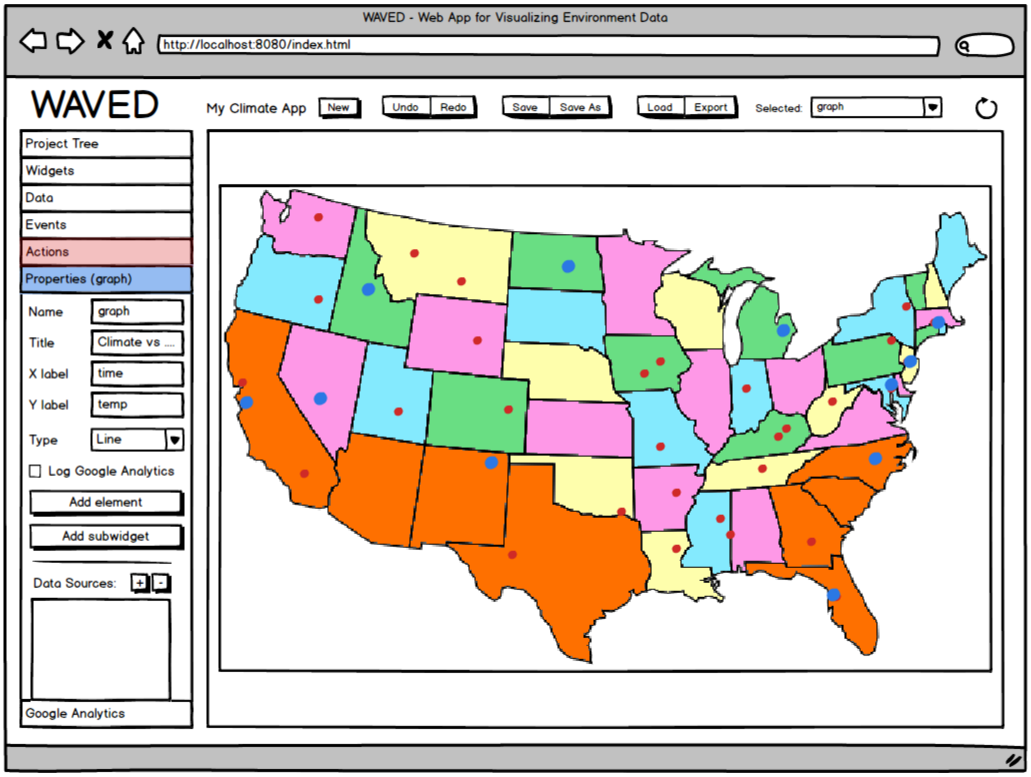
☑ Log Google Analytics

Data Sources: climate.csv, heat_index.json

Widget: Graph

□ Show

Figure 12. Add subwidget dialog



Selected: graph

Name: graph

Title: Climate vs ...

X label: time

Y label: temp

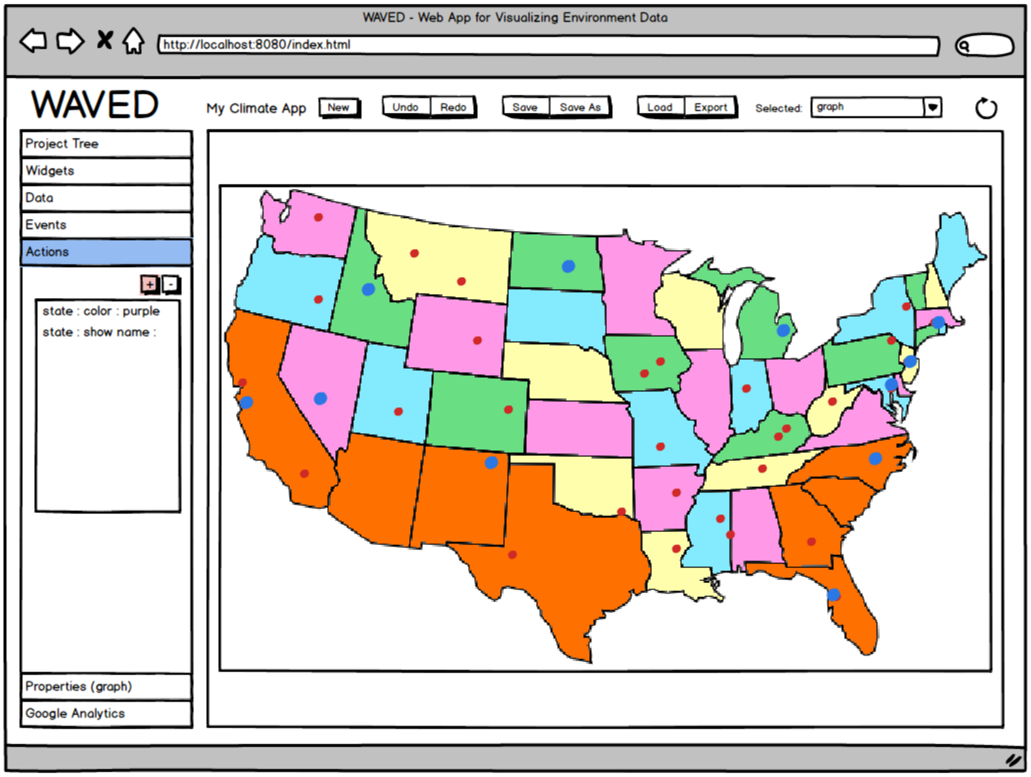
Type: Line

□ Log Google Analytics

Figure 13. The Properties Panel for a hidden graph subwidget

3.5 Actions Panel

The Actions Panel (Figure 14) allows the user to add, edit, and remove actions. The workflow for creating a property type action is illustrated in Figure 15. Clicking the plus (+) button in the Actions Panel displays the Action Editor dialog. The Action Editor dialog contains options for selecting the widget or element to be affected, the bound data set the action will use, the widget or element property to change, and a new value for the chosen property (Figure 16). If the “apply by default” option is selected in the Action Editor dialog, the workspace updates to reflect the effect of the action. Once an action is created, it is displayed in the Actions Panel and is available to be used by events. Selecting an action from the Actions Panel and clicking the minus (-) button removes the action from the project.



Selected: graph

state : color : purple
state : show name :

Figure 14. Actions Panel

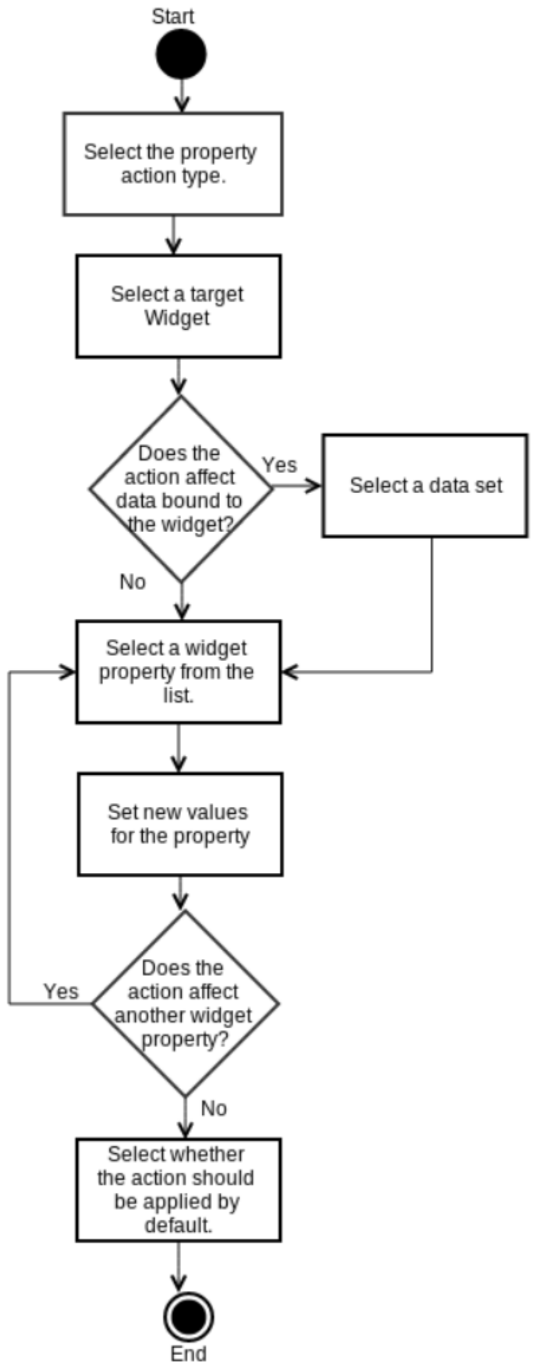
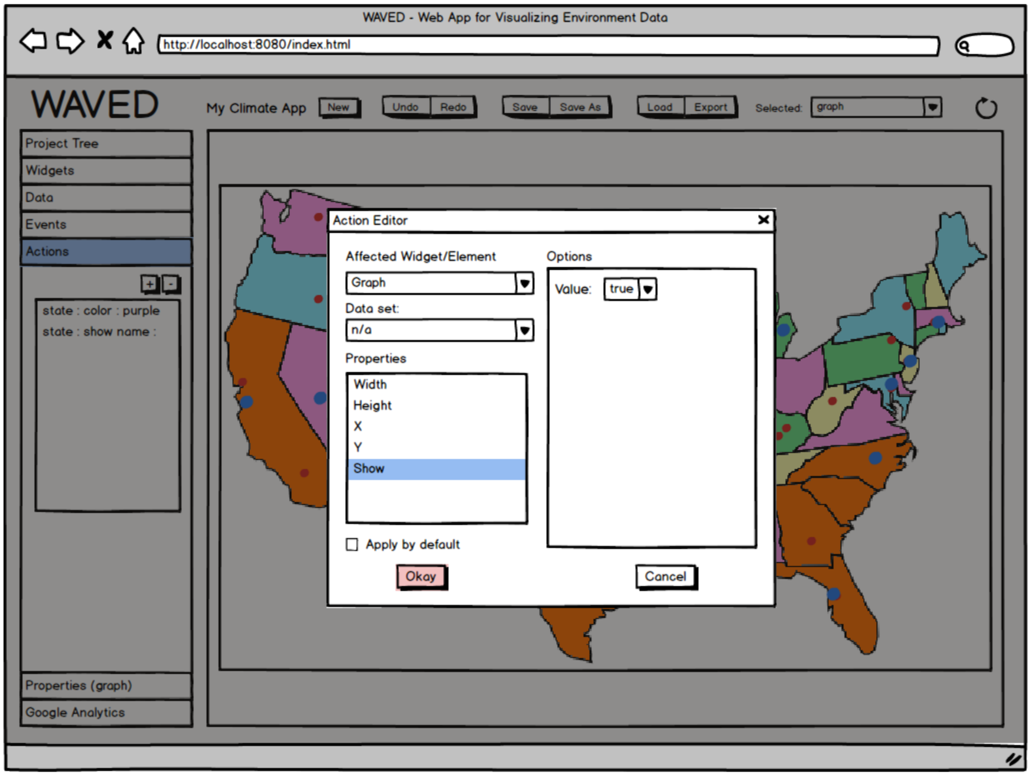

Start
Select the property action type.
Select a target Widget
Does the action affect data bound to the widget?
Yes
Select a data set
No
Select a widget property from the list.
Set new values for the property
Does the action affect another widget property?
Yes
No
Select whether the action should be applied by default.
End

Figure 15. Activity diagram for creating a property action.



state : color : purple
state : show name :

Figure 16. Action Editor dialog

3.6 Events Panel

Opening the Events Panel (Figure 17) and clicking the plus (+) button opens the event editor dialog (Figure 18). From here, the user follows the workflow described in Figure 19 to create an event. After an event has been added to the project, the user can interact with the workspace to trigger the event and fire the associated action. An exported visualization behaves in exactly the same way as the interactive preview in the workspace. Within WAVED, users can click the refresh button located at the top right side of the window to reset the workspace to the initial state of the visualization (before any user events were fired).

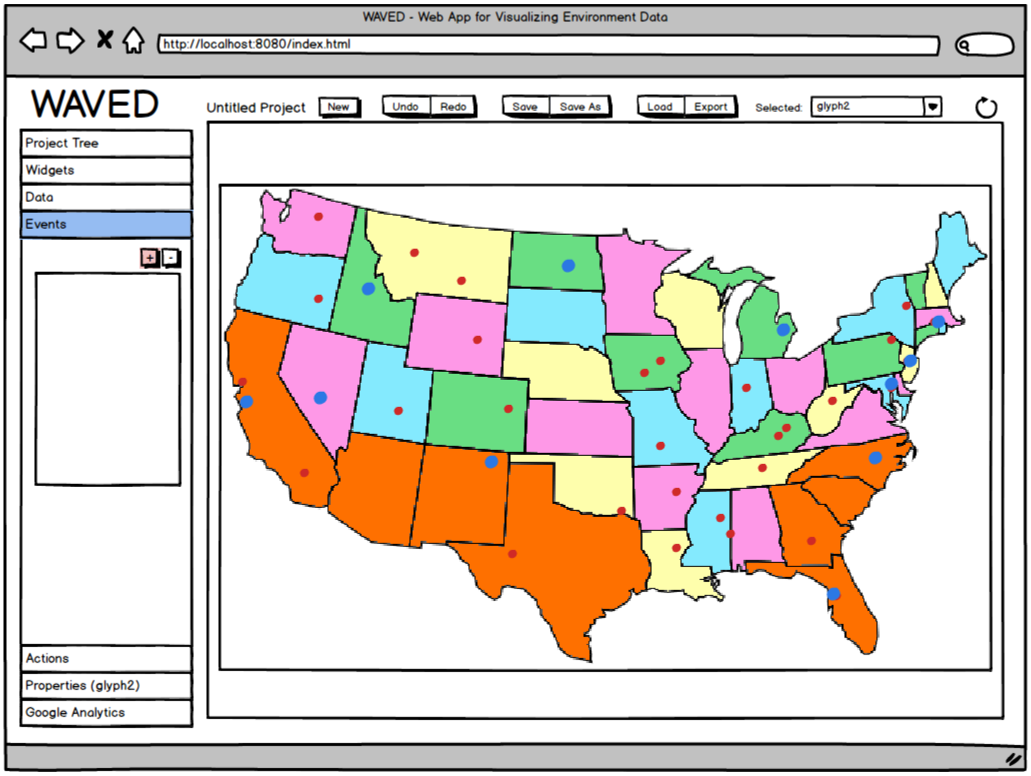
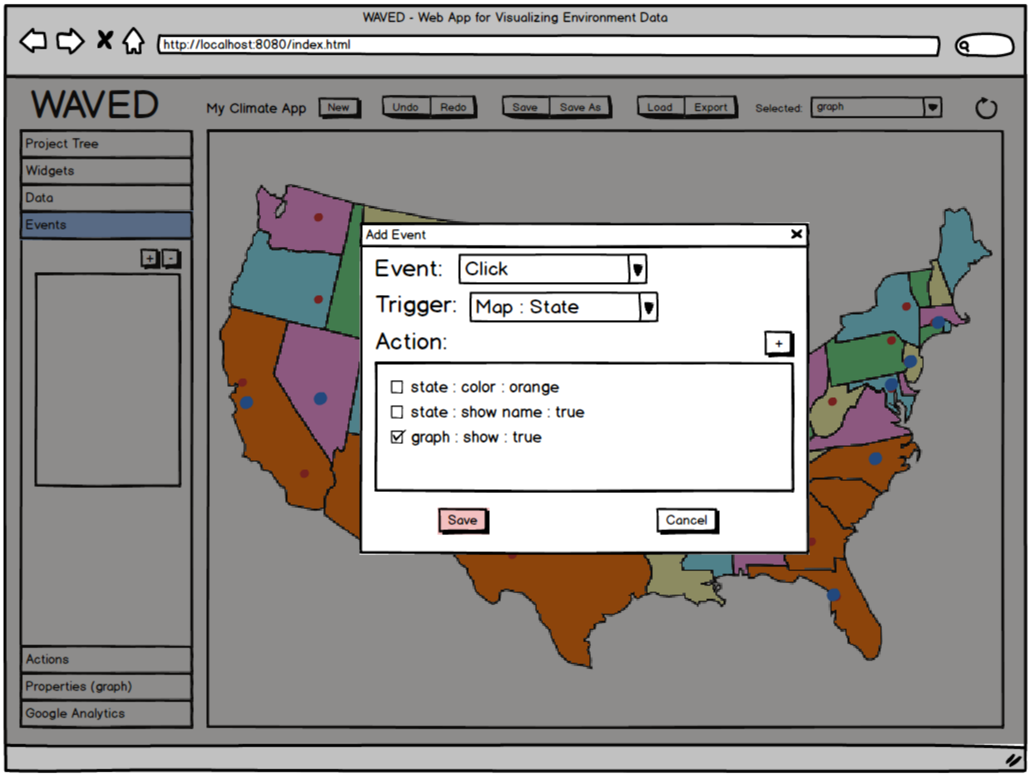


Figure 17. Events Panel



Event: Click

Trigger: Map : State

Action:

- ☐ state : color : orange
- ☐ state : show name : true
- ☑ graph : show : true

Figure 18. Event Editor Dialog

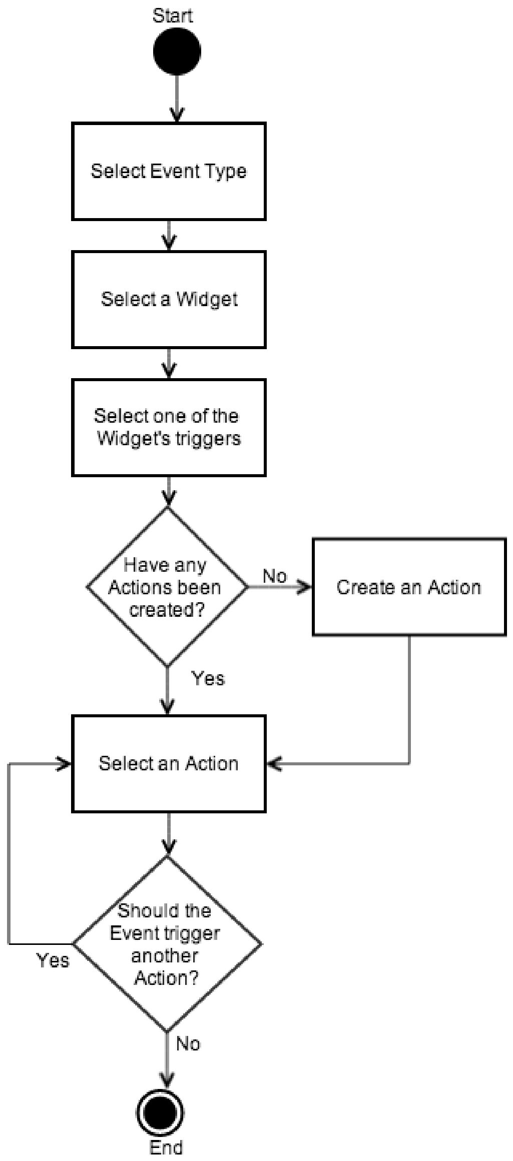
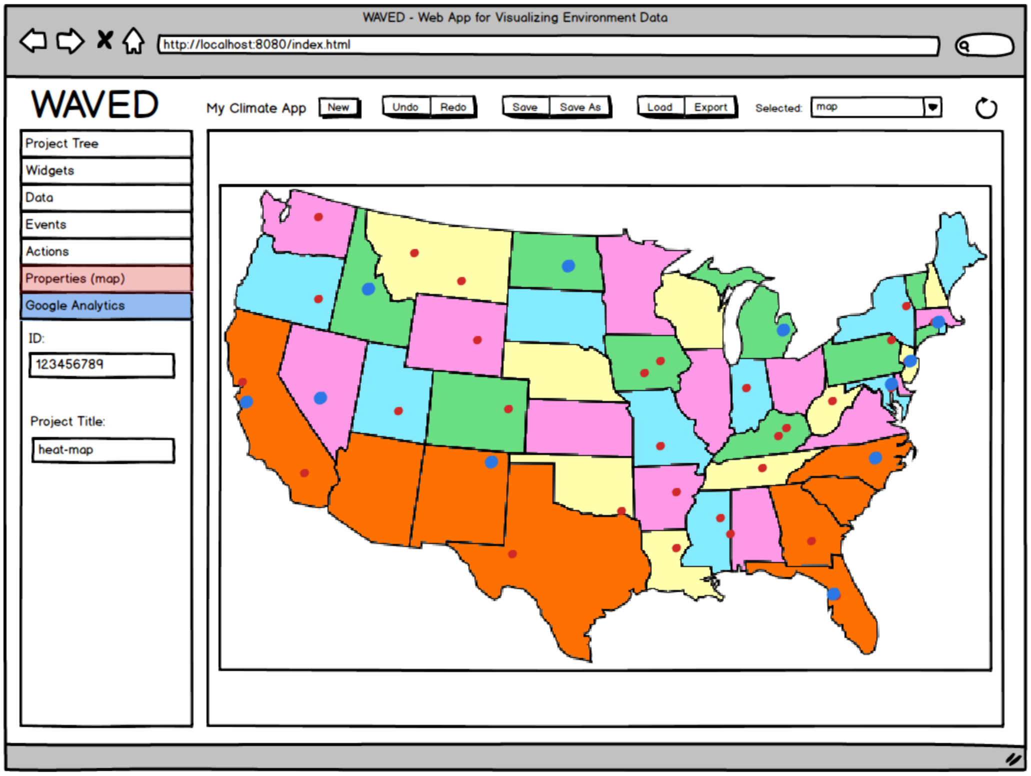

Start
Select Event Type
Select a Widget
Select one of the Widget's triggers
Have any Actions been created?
No
Create an Action
Yes
Select an Action
Should the Event trigger another Action?
Yes
No
End

Figure 19. Activity diagram for creating an event

3.7 Google Analytics Panel

Opening the Google Analytics Panel displays input fields where the user can add a Google Analytics ID and a project title used for grouping group all logged events within Google Analytics (Figure 20). The user can have an individual widget log Google Analytics click events by checking the “Log Google Analytics” checkbox in the widget’s Properties Panel (Figure 21).



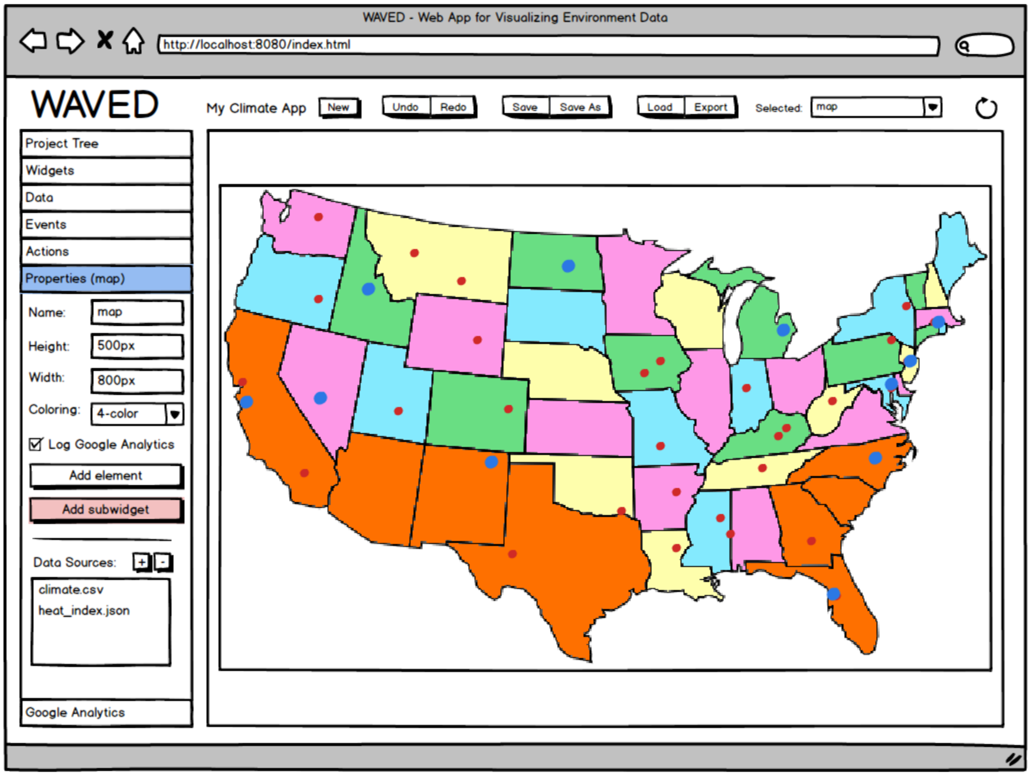
http://localhost:8080/index.html

Selected: map

ID: 123456789

Project Title: heat-map

Figure 20. Google Analytics Panel



http://localhost:8080/index.html

Selected: map

Name: map

Height: 500px

Width: 800px

Coloring: 4-color

☑ Log Google Analytics

Data Sources: climate.csv heat_index.json

Figure 21. A checked Log Google Analytics checkbox in a widget's Properties Panel

3.8 Project Tree Panel

The Project Tree Panel displays all of the widgets, subwidgets, and elements that have been added to the current project in a hierarchical fashion (Figure 22). Users can delete a widget, subwidget, or element from the project by selecting a widget from the project tree and clicking the minus (-) button located in the top right of the Project Tree Panel. Selecting the project folder from the Project Tree Panel and clicking the minus (-) button displays a dialog asking the user to confirm deletion of the project. After a project is deleted, the user is shown the welcome screen pictured in Figure 1. Additionally, if the user opens the Properties Panel after selecting a widget from the project tree, the Properties Panel is populated with the properties for the selected widget. This behavior is the same as what happens when a widget is selected using the combobox found in the main menu.

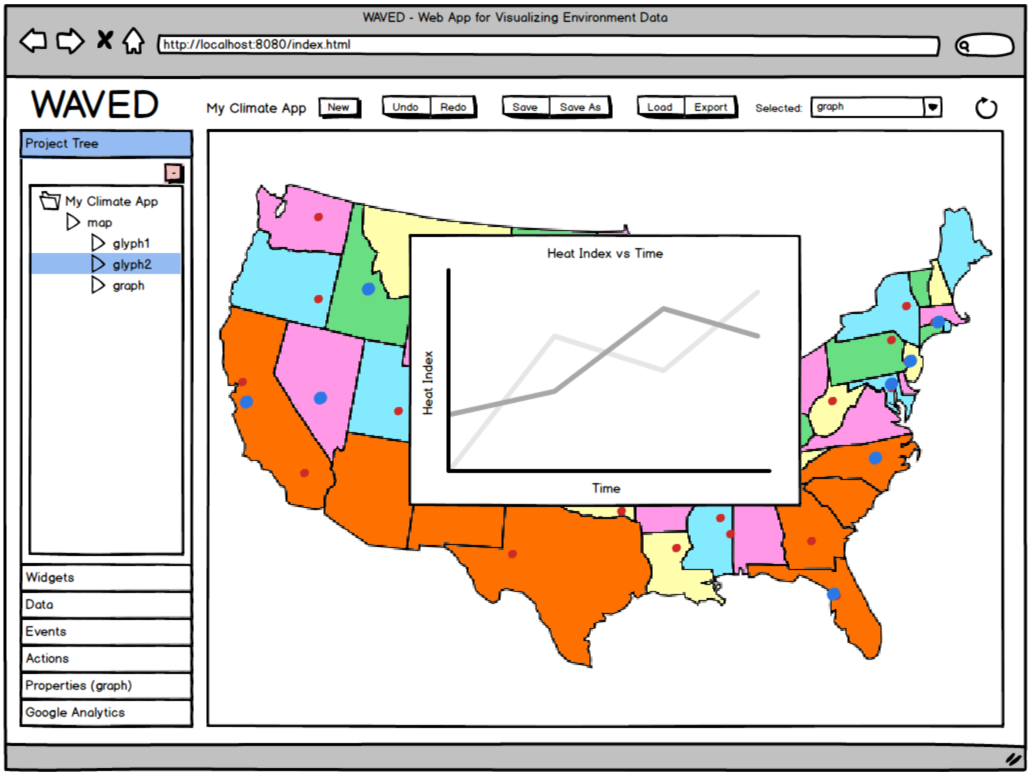
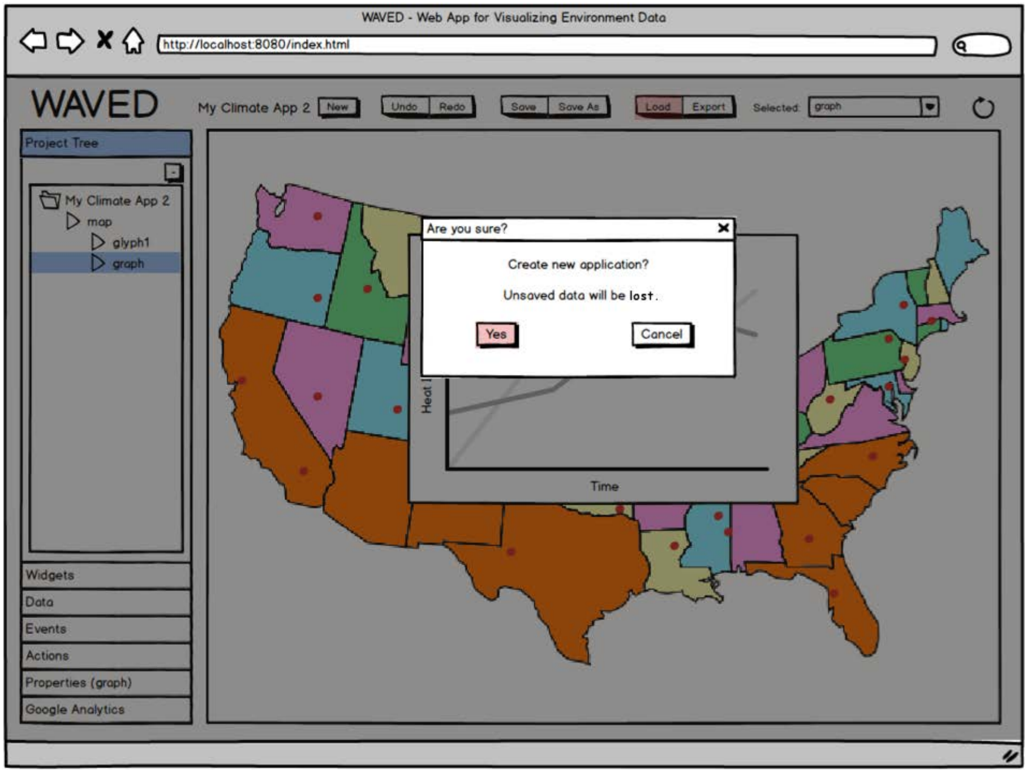


Figure 22. Project Tree Panel

3.9 Main Menu

The main menu bar located at the top of the screen contains New, Undo, Redo, Save, Save As, Load, and Export buttons. There are two ways to create a new project, as seen in the activity diagram shown in Figure 3. Aside from selecting to create a new project when WAVED is loaded, users can click the New button in the main menu to create a new project at any time. Clicking the New button displays a dialog asking the user to agree to discard any unsaved changes and to create the new project (Figure 23). Clicking yes prompts the user to enter a valid name for the new project and then opens up a blank new project as seen in Figure 2.



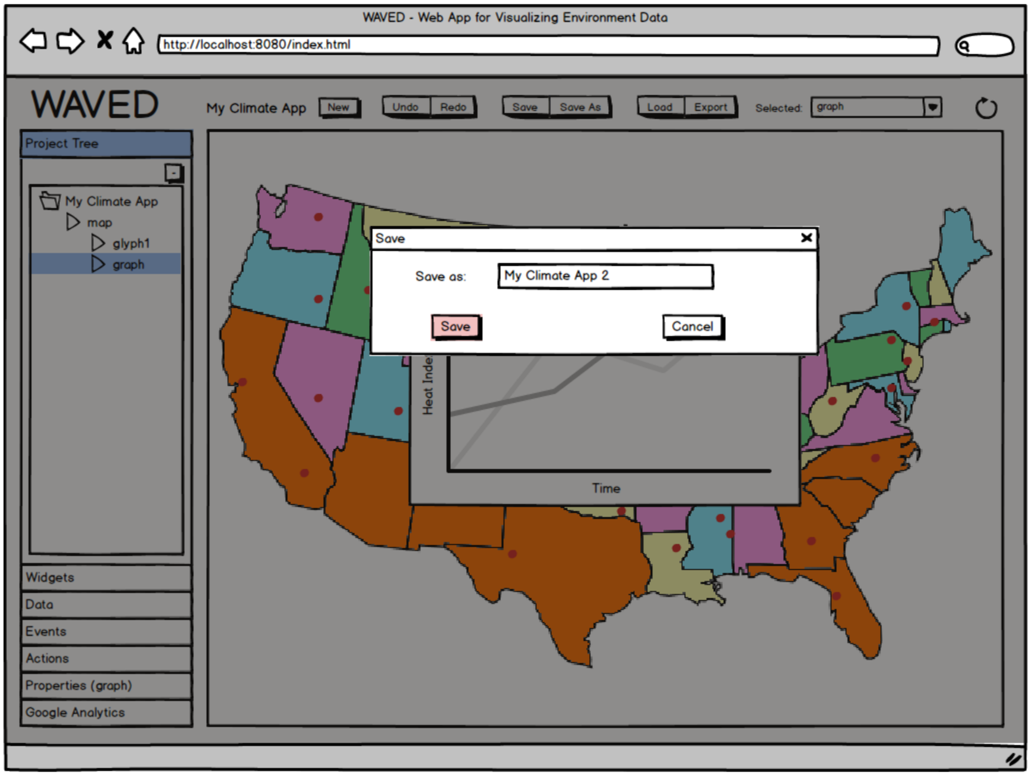
Are you sure?

Create new application?

Unsaved data will be lost.

Figure 23. New Project Dialog

Clicking the Save button saves the current state of the project under the name given to the project upon creation. The Save As button allows the user to save a copy of the project under a different project name. Clicking the Save As button opens a dialog prompting the user for a new project name (Figure 24). If the project name entered already exists, the user is prompted to confirm whether the existing project should be overwritten (Figure 24).



Save as: My Climate App 2

Figure 24. Save As Dialog

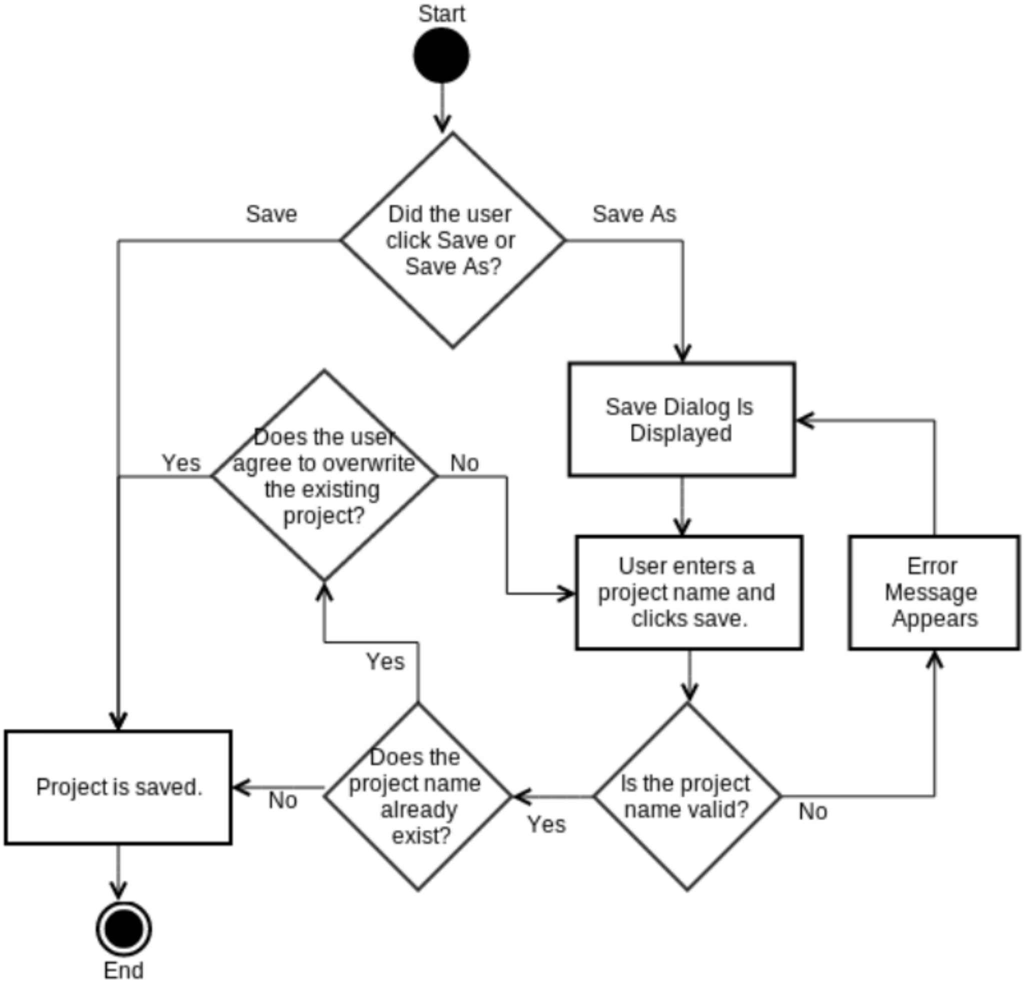
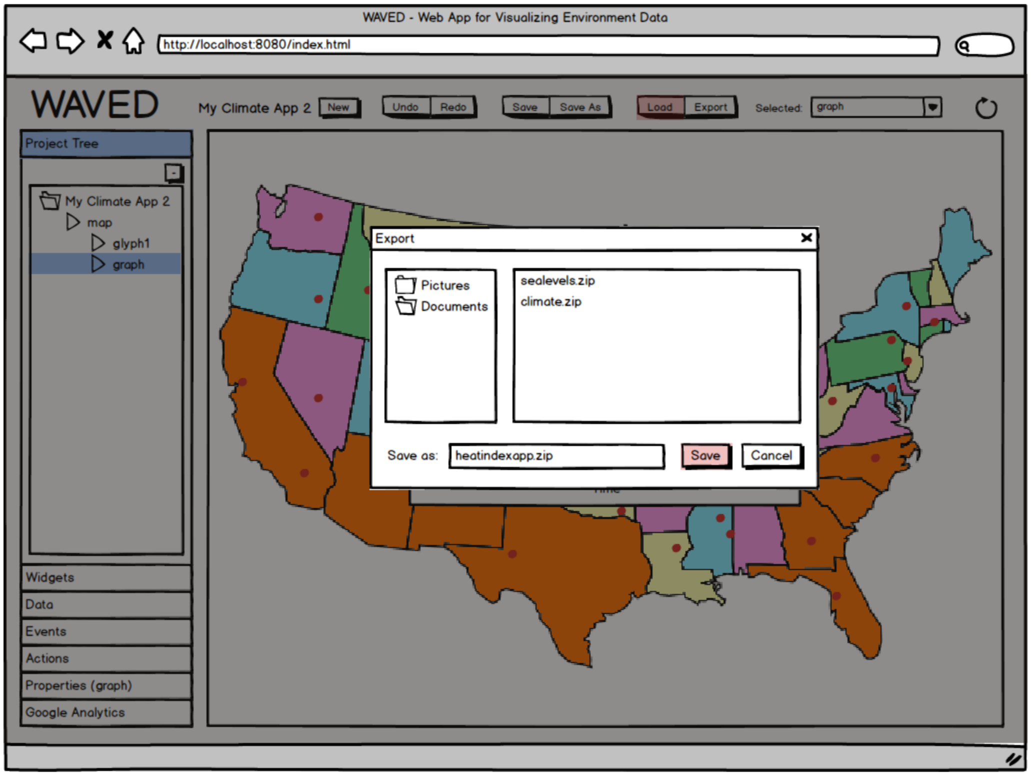

Start
Did the user click Save or Save As?
Save
Save As
Save Dialog Is Displayed
Does the user agree to overwrite the existing project?
Yes
No
User enters a project name and clicks save.
Error Message Appears
Yes
Project is saved.
No
Does the project name already exist?
Yes
Is the project name valid?
No
End

Figure 25. An activity diagram for saving a project

Clicking the Export button displays a dialog allowing the user to name the exported project file. A file browser then appears for the user to designate the location to save the file if the user's web browser doesn't have a default download location (Figure 26).



http://localhost:8080/index.html

Selected: graph

Save as: heatindexapp.zip

Figure 26. Export Dialog

Clicking the Load button displays a dialog listing all existing projects available to the user (Figure 27). Clicking the Load button within that dialog after selecting a project loads the project state into the workspace and populates the side panels with the project's widget, data, action, event, and Google Analytics information.

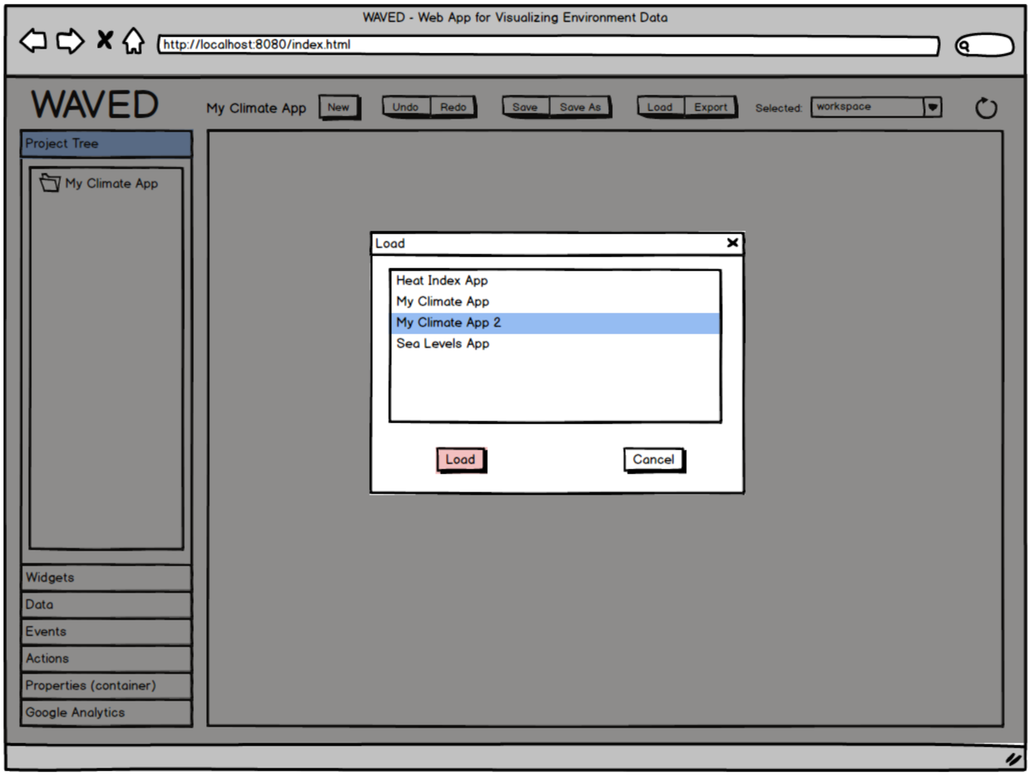


Figure 27. Load Project Dialog

4. Functional Requirements

4.1 Application

4.1.1 Upon launching the application, the user must choose one of the following options:

4.1.1.1 Create a new project **Priority 1**

4.1.1.2 Load an existing project **Priority 1**

4.2 Creating a Project

4.2.1 The user can select to create a new project when the application is launched **Priority 1**

4.2.2 The user can create a new project from the main interface. **Priority 1**

4.2.3 When a user selects to create a new project and the current project contains unsaved changes, the application prompts the user whether any unsaved changes should be abandoned. **Priority 1**

4.2.4 The user is prompted to name the project upon creation. **Priority 1**

4.2.4.1 Every project must have a unique name. **Priority 1**

4.2.4.2 A valid project name contains 1 to 50 characters. **Priority 1**

4.2.4.3 A valid project name only includes alphanumeric, hyphen (-), underscore (_) and space characters. **Priority 1**

4.2.4.4 If the project name matches the name of an existing project, the application prompts the user whether the existing project should be overwritten. **Priority 1**

4.3 Saving a Project

4.3.1 The user can save the state of the current project. **Priority 1**

4.3.2 The user can save a copy of the current project under a different name. **Priority 1**

4.3.3 The user can save a previously saved project under the same name without being prompted for a project name. **Priority 1**

4.3.4 Once a project has been saved, the application automatically saves the project every five minutes. **Priority 3**

4.4 Loading a Project

4.4.1 The user can load a previously saved project. **Priority 1**

4.4.2 When loading a project, the user selects from a list of previously saved projects. **Priority 1**

4.4.3 Once loaded, the project state matches the most recently saved version of the project. **Priority 1**

4.5 Deleting a Project

4.5.1 The user can choose to delete a project from the list of saved projects. **Priority 1**

4.5.2 The user can choose to delete the currently opened project. **Priority 1**

4.5.3 The application prompts the user to confirm the deletion of a project before it is permanently deleted. **Priority 1**

4.5.4 Once a project is deleted, the project no longer appears in the list of saved projects and can no longer be loaded. **Priority 1**

4.5.5 Once deleted, all files associated with the project are removed from the server. **Priority 1**

4.6 Workspace

4.6.1 The application automatically updates the workspace every time the user makes a change to the project. **Priority 1**

4.6.2 The application provides a mechanism for restoring the workspace to a state where no user events have been fired. **Priority 1**

4.6.3 The workspace emulates the functionality of the exported project. **Priority 1**

4.6.4 The user can specify the height of the workspace in pixels as a non-negative integer. **Priority 1**

4.6.5 The default height of the workspace is 600 pixels. **Priority 1**

4.6.6 The user can specify the width of the workspace in pixels as a non-negative integer. **Priority 1**

4.6.7 The default width of the workspace is 750 pixels. **Priority 1**

4.7 Export Visualization

4.7.1 Exported File

4.7.1.1 The user can export the current visualization as a zip file.

4.7.1.2 The zip file contains the following files:

4.7.1.2.1 An HTML file (.html) **Priority 1**

4.7.1.2.2 A JavaScript file (.js) **Priority 1**

4.7.1.2.3 A CSS file (.css) **Priority 1**

4.7.1.2.4 A folder containing all data files bound to widgets **Priority 1**

4.7.2 Visualization Functionality

4.7.2.1 The exported visualization has the same functionality as what is shown in the workspace. **Priority 1**

4.7.2.2 The exported visualization can be embedded in a web page. **Priority 1**

4.7.3 Readability of Exported Code

4.7.3.1 All exported code must comply with the following conventions:

4.7.3.1.1 Indentation and white space are used to indicate code hierarchy. **Priority 1**

4.7.3.1.2 Each line contains a single statement. **Priority 1**

4.7.3.1.3 Variable names are related to the context of the code. **Priority 1**

4.7.3.1.4 Code comments separate code for each widget. **Priority 1**

4.8 Project Revision History

4.8.1 Undo

4.8.1.1 The user can revert the 10 most recent client-side changes made to the current project. **Priority 1**

4.8.1.2 The undo functionality is disabled when no changes have been made. **Priority 1**

4.8.2 Redo

4.8.2.1 The user can restore a change that has been reverted. **Priority 1**

4.8.2.2 When the user makes a change after reverting a change, the reverted change can no longer be restored. **Priority 1**

4.8.2.3 The redo functionality is disabled when no changes have been reverted. **Priority 1**

4.9 Widgets and Elements

4.9.1 Adding a Widget

4.9.1.1 The user can add a widget to the project. **Priority 1**

4.9.1.2 The user can add a widget as a child (subwidget) of an existing widget. **Priority 1**

4.9.2 Adding an Element

4.9.2.1 The user can add an element to a supported parent widget. **Priority 1**

4.9.2.2 The user can add a subwidget as a child of an existing element. **Priority 3**

4.9.3 Modifying a Widget or Element

4.9.3.1 The user can modify any of the defined properties of a widget or element. **Priority 1**

4.9.3.2 Users can only modify the properties of the currently selected widget or element. **Priority 1**

4.9.4 Deleting a Widget or Element

4.9.4.1 The user can delete a widget or element that has been added to the project. **Priority 1**

4.9.4.2 A widget or element can only be deleted if it has first been selected. **Priority 1**

4.9.4.3 A widget cannot be deleted until no events use the widget as a trigger. **Priority 1**

4.9.4.4 A widget cannot be deleted until no actions target the widget. **Priority 1**

4.9.4.5 If the widget or element selected for deletion has any children, the user is prompted to confirm the deletion. **Priority 1**

4.9.4.6 Once a widget or element is deleted, all of its children are deleted recursively. **Priority 1**

4.9.5 Common Widget and Element Properties

4.9.5.1 Every widget and element must have the following properties:

4.9.5.1.1 Name **Priority 1**

4.9.5.1.1.1 A valid name contains between 1 and 50 characters. **Priority 1**

4.9.5.1.1.2 A name must be unique to each widget or element. **Priority 1**

4.5.8.1.1.3 A valid widget name can only include alphanumeric, hyphen (-), underscore (_) and space characters. **Priority 1**

4.9.5.1.2 Visibility **Priority 1**

4.9.5.1.2.1 A valid visibility value is true or false, representing whether the widget or element is visible or hidden. **Priority 1**

4.9.5.1.2.2 The default visibility value is true, representing that the widget or element is visible. **Priority 1**

4.9.5.1.3 Log Google Analytics. **Priority 1**

4.9.5.1.3.1 If set to true, Google Analytics are logged for the widget. **Priority 1**

4.9.6 Common Widget Properties

4.9.6.1 Every widget must have the following properties:

4.9.6.1.1 Horizontal Offset **Priority 1**

4.9.6.1.1.1 A valid horizontal offset value is a number between 0 and 100 representing a percentage of the workspace width. **Priority 1**

4.9.6.1.1.2 The horizontal offset is measured from the left edge of the workspace. **Priority 1**

4.9.6.1.1.3 The default horizontal offset value is zero. **Priority 1**

4.9.6.1.2 Vertical Offset **Priority 1**

4.9.6.1.2.1 A valid vertical offset value is a number between 0 and 100 representing a percentage of the workspace height. **Priority 1**

4.9.6.1.2.2 The vertical offset is measured from the top edge of the workspace. **Priority 1**

4.9.6.1.2.3 The default vertical offset value is zero. **Priority 1**

4.9.6.1.3 Height **Priority 1**

4.9.6.1.3.1 A valid height value is a number between 0 and 100 representing a percentage of the workspace height. **Priority 1**

4.9.3.1.3.2 The default height value is 50. **Priority 1**

4.9.6.1.4 Width **Priority 1**

4.9.6.1.4.1 A valid width value is a number between 0 and 100 representing a percentage of the workspace width. **Priority 1**

4.9.6.1.4.2 The default width value is 50. **Priority 1**

4.9.7 Available Widgets

4.9.7.1 U.S. Map

4.9.7.1.1 U.S. Map

4.9.7.1.1.1 The map must display all 50 United States. **Priority 1**

4.9.7.1.2 U.S. Map Properties

4.9.7.1.2.1 Coloring **Priority 1**

4.9.7.1.2.1.1 The individual states on the U.S. map can be colored according to one of the following schemes:

4.9.7.1.2.1.1.1 All states are colored using the same color. **Priority 1**

4.9.7.1.2.1.1.2 The states are colored using four distinguishable colors such that no two adjacent states use the same color. **Priority 1**

4.9.7.1.2.1.1.3 The intensity of the hue for each state varies based on a number valued property in a per-state data file. **Priority 2**

4.9.7.1.1.1.2 The default coloring is grey for all states. **Priority 1**

4.9.7.1.3 U.S. Map Elements

4.9.7.1.3.1 Glyph (4.9.8.1) **Priority 1**

4.9.7.1.3.2 Tooltip (4.9.8.2) **Priority 1**

4.9.7.1.4 U.S. Map Triggers

4.9.7.1.4.1 Anywhere within the boundaries of the U.S. map. **Priority 1**

4.9.7.1.4.2 Individual states on the U.S. map. **Priority 1**

4.9.7.1.4.3 Geographic regions on the U.S. map. **Priority 2**

4.9.7.2 Line Graph

4.9.7.2.1 Line Graph

4.9.7.2.1.1 The graph displays circles for data points. **Priority 1**

4.9.7.2.1.2 The graph displays lines that connect the data points. **Priority 1**

4.9.7.2.1.3 The graph has the option to display a line of best fit. **Priority 2**

4.9.7.2.2 Line Graph Properties

4.9.7.2.2.1 Title **Priority 1**

4.9.7.2.2.1.1 A valid title contains between 0 and 50 characters. **Priority 1**

4.9.7.2.2.2 X-Axis Label **Priority 1**

4.9.7.2.2.2.1 A valid x-axis label contains between 0 and 50 characters. **Priority 1**

4.9.7.2.2.3 Y-Axis Label **Priority 1**

4.9.7.2.2.3.1 A valid y-axis label contains between 0 and 50 characters. **Priority 1**

4.9.7.2.2.4 Graph Data **Priority 1**

4.9.7.2.2.4.1 The user must define which bound data set to use for the x and y axis data (4.9.7.2.2.5, 4.9.7.2.2.6). **Priority 1**

4.9.7.2.2.5 X-Axis Data Field **Priority 1**

4.9.7.2.2.5.1 The x-axis data field must be a numeric field from a data set bound to the line graph widget. **Priority 1**

4.9.7.2.2.6 Y-Axis Data Field **Priority 1**

4.9.7.2.2.6.1 The y-axis data field must be a numeric field from a data set bound to the line graph widget. **Priority 1**

4.9.7.2.3 Line Graph Elements

4.9.7.2.3.1 Tooltip (4.9.8.2) **Priority 1**

4.9.7.2.4 Line Graph Triggers

4.9.7.2.4.1 Anywhere within the boundaries of the line graph. **Priority 1**

4.9.7.2.4.2 The data points on the line graph. **Priority 1**

4.9.7.2.4.3 Vertical grid lines aligned with the data points on the line graph. **Priority 1**

4.9.7.3 Button

4.9.7.3.1 Button

4.9.7.3.1.1 The button features a clickable area with text. **Priority 1**

4.9.7.3.2 Button Properties

4.9.7.3.2.1 Label **Priority 1**

4.9.7.3.2.1.1 A valid label has between 1 and 50 characters. **Priority 1**

4.9.7.3.2.2 Color **Priority 2**

4.9.7.3.2.2.1 A valid color value is a legal CSS color value. **Priority 2**

4.9.7.3.3 Button Triggers

4.9.7.3.3.1 The button. **Priority 1**

4.9.7.4 Slider

4.9.7.4.1 Slider

4.9.7.4.1.1 A slider includes a sliding selector to choose between a range of values **Priority 3**

4.9.7.4.2 Slider Properties

4.9.7.4.2.1 Minimum Value **Priority 3**

4.9.7.4.2.1.1 A valid minimum value is a number representing the lower bound of the slider. **Priority 3**

4.9.7.4.2.1.2 The minimum value must be less than the maximum value. **Priority 3**

4.9.7.4.2.2 Maximum Value **Priority 3**

4.9.7.4.2.2.1 A valid maximum is a number representing the upper bound of the slider. **Priority 3**

4.9.7.4.2.2.2 The maximum value must be greater than the minimum value. **Priority 3**

4.9.7.4.2.3 Current Value **Priority 3**

4.9.7.4.2.3.1 A valid current value is a number representing the current position of the slider’s pointer. **Priority 3**

4.9.7.4.2.3.2 The current value of the pointer must be between the minimum and maximum values inclusively. **Priority 3**

4.9.7.4.2.3.3 The default for the current value is the minimum value. **Priority 3**

4.9.7.4.2.4 Increment **Priority 3**

4.9.7.4.2.4.1 A number that determines the amount by which the slider increments. **Priority 3**

4.9.7.4.3 Slider Triggers

4.9.7.4.3.1 The slider pointer. **Priority 3**

4.9.7.4.3.2 The slider’s scale. **Priority 3**

4.9.7.5 Text Block

4.9.7.5.1 Text Block

4.9.7.5.1.1 A text block has a rectangular area containing text. **Priority 1**

4.9.7.5.2 Text Block Properties

4.9.7.5.2.1 Text Content **Priority 1**

4.9.7.5.2.1.1 Valid text content has up to 500 characters. **Priority 1**

4.9.7.5.2.2 Text Color **Priority 2**

4.9.7.5.2.2.1 A valid color value is a legal CSS color value. **Priority 2**

4.9.7.5.2.3 Background Color **Priority 2**

4.9.7.5.2.3.1 A valid background color value is a legal CSS color value. **Priority 2**

4.9.7.5.2.4 Border Color **Priority 2**

4.9.7.5.2.4.1 A valid border color value is a legal CSS color value. **Priority 2**

4.9.7.5.2.5 Border Thickness **Priority 2**

4.9.7.5.2.5.1 A valid thickness value is a nonnegative integer representing the width of the border in pixels. **Priority 2**

4.9.7.5.2.6 Border Radius **Priority 2**

4.9.7.5.2.6.1 A valid border radius value is a nonnegative integer. **Priority 2**

4.9.7.5.3 Text Block Triggers

4.9.7.5.3.1 Any location within the text block's boundaries. **Priority 2**

4.9.7.6 Combobox

4.9.7.6.1 Combobox

4.9.7.6.1.1 A combobox has a dropdown menu with one or more options. **Priority 2**

4.9.7.6.2 Combobox Properties

4.9.7.6.2.1 Options **Priority 2**

4.9.7.6.2.1.1 The combobox must contain one or more options. **Priority 2**

4.9.7.6.2.1.2 A valid option has a label and a value. **Priority 2**

4.9.7.6.2.1.3 A valid label has text between 1 and 50 characters. **Priority 2**

4.9.7.6.2.1.4 A valid value has between 1 and 50 characters. **Priority 2**

4.9.7.6.2.2 Selected Option **Priority 2**

4.9.7.6.2.2.1 The option currently selected from the list of available options. **Priority 2**

4.9.7.6.3 Combobox Triggers

4.9.7.6.3.1 The combobox. **Priority 2**

4.9.7.6.3.2 An option from the combobox. **Priority 2**

4.9.7.7 Radio Button Set

4.9.7.7.1 Radio Button Set

4.9.7.7.1.1 A radio button set has one or more radio buttons. **Priority 2**

4.9.7.7.1.2 Each radio button has an associated label. **Priority 2**

4.9.7.7.2 Radio Button Set Properties

4.9.7.7.2.1 Labels **Priority 2**

4.9.7.7.2.1.1 A valid label is text between 1 and 50 characters. **Priority 2**

4.9.7.7.2.2 Orientation **Priority 3**

4.9.7.7.2.2.1 The radio button set can be oriented vertically or horizontally. **Priority 3**

4.9.7.7.2.3 Selected Button **Priority 2**

4.9.7.7.2.3.1 The button in the set that is currently selected. **Priority 2**

4.9.7.7.3 Radio Button Set Triggers

4.9.7.7.3.1 Any radio button in the set. **Priority 2**

4.9.7.8 Checkbox Set

4.9.7.8.1 Checkbox Set

4.9.7.8.1.1 A checkbox set has one or more checkboxes. **Priority 2**

4.9.7.8.1.2 Each checkbox has an associated label. **Priority 2**

4.9.7.8.2 Checkbox Set Properties

4.9.7.8.2.1 Labels **Priority 2**

4.9.7.8.2.1.1 A valid label is text between 1 and 50 characters. **Priority 2**

4.9.7.8.2.2 Orientation **Priority 3**

4.9.7.8.2.2.1 The checkbox set can be oriented vertically or horizontally. **Priority 3**

4.9.7.8.2.3 Checked Boxes **Priority 2**

4.9.7.8.2.3.1 The checkboxes in the set that are currently checked. **Priority 2**

4.9.7.8.3 Checkbox Set Triggers

4.9.7.8.3.1 Any checkbox in the set. **Priority 2**

4.9.8 Available Elements

4.9.8.1 Glyphs

4.9.8.1.1 Glyph Properties

4.9.8.1.1.1 Data

4.9.8.1.1.1.1 The user is required to select a data set bound to the parent widget. **Priority 1**

4.9.8.1.1.2 Color

4.9.8.1.2.2.1 A valid color value is a legal CSS color value. **Priority 1**

4.9.8.1.2.2.2 The default color value is red. **Priority 1**

4.9.8.1.1.3 Shape

4.9.8.1.1.3.1 A glyph can have one of the following shapes:

4.9.8.1.1.3.1.1 Circle **Priority 1**

4.9.8.1.1.3.1.2 Square **Priority 2**

4.9.8.1.1.3.1.3 Triangle **Priority 2**

4.9.8.1.1.3.2 The default shape is circle. **Priority 1**

4.9.8.1.1.4 Size

4.9.8.1.1.4.1 The dimension of the glyphs can be one of the following:

4.9.8.1.1.4.1.1 Each glyph is a constant size, measured in pixels. **Priority 1**

4.9.8.1.1.4.1.2 Each glyph is scaled according to a numeric data field the user chooses from the selected data file. **Priority 1**

4.9.8.1.1.5 Latitude

4.9.8.1.1.5.1 The latitude of the glyphs is determined by a user selected field in one of the data files bound to the parent U.S. Map widget. **Priority 1**

4.9.8.1.1.5.2 The latitude values must be numbers between -180 and 180. **Priority 1**

4.9.8.1.1.6 Longitude

4.9.8.1.1.6.1 The longitude of the glyphs is determined by a user selected field in one of the data files bound to the parent U.S. Map widget. **Priority 1**

4.9.8.1.1.6.2 The longitude values must be numbers between -90 and 90. **Priority 1**

4.9.8.1.2 Glyph Triggers

4.9.8.1.2.1 An individual glyph. **Priority 1**

4.9.8.1.2.2 Any glyph in the set of glyphs. **Priority 1**

4.9.8.2 Tooltip

4.9.8.2.1 Tooltip Properties

4.9.8.2.1.1 Text **Priority 1**

4.9.8.2.1.1.1 The text can contain up to 500 characters **Priority 1**

4.9.8.2.2 Tooltip Triggers

4.9.8.2.3.1 The tooltip. **Priority 1**

4.10 Data

4.10.1 Valid Data File Formats

4.10.1.1 The application accepts data files of the following formats:

4.10.1.1.1 A comma separated value (CSV) file. **Priority 1**

4.10.1.1.2 A JavaScript Object Notation (JSON) file. **Priority 1**

4.10.1.1.3 An XML file **Priority 1**

4.10.2 Uploading a Data Source

4.10.2.1 The user can select a local data file to upload and use in a project. **Priority 1**

4.10.2.2 The user cannot upload a local data file that is not of a valid file format. **Priority 1**

4.10.2.3 Each data source uploaded to a project must have a unique name. **Priority 1**

4.10.3 Viewing a Data Set

4.10.3.1 The user can view the data fields for a data set that has been added to the current project. **Priority 1**

4.10.3.2 The user can view the first 10 data entries of a data set that has been added to the current project. **Priority 1**

4.10.4 Creating a Data Subset

4.10.4.1 A data subset must have a name to identify it. **Priority 1**

4.10.4.1.1 The provided data subset name must be unique across all data set names. **Priority 1**

4.10.4.1.2 A valid data subset name consists of between 1 and 255 characters. **Priority 1**

4.10.4.1.3 A valid data subset name can only include alphanumeric, hyphen (-), underscore (_) and space characters. **Priority 1**

4.10.4.2 The user must specify the name of a data source added to the project from which the subset will be created. **Priority 1**

4.10.4.3 The user must specify one or more conditional statements, each consisting of one of each of the following:

4.10.4.3.1 A data field contained in the data source specified by 4.10.4.2. **Priority 1**

4.10.4.3.2 One of the following comparison operators:

4.10.4.3.2.1 Less Than (<) **Priority 1**

4.10.4.3.2.2 Less Than or Equal To (≤). **Priority 1**

4.10.4.3.2.3 Equal To (=) **Priority 1**

4.10.4.3.2.4 Not Equal To (≠) **Priority 1**

4.10.4.3.2.5 Greater Than (>) **Priority 1**

4.10.4.3.2.6 Greater Than or Equal To (≥) **Priority 1**

4.10.4.3.3 A user-supplied value. **Priority 1**

4.10.4.4 Each conditional statement is combined using one of the following logical operators:

4.10.4.4.1 AND **Priority 1**

4.10.4.4.2 OR **Priority 1**

4.10.4.5 The precedence of AND is greater than OR. **Priority 1**

4.10.5 Removing a Data Set

4.10.5.1 The user can remove a data set from the project. **Priority 1**

4.10.5.2 If the user attempts to delete a data source, and data subsets created from the data source exist, the user is asked to confirm deletion of the data source. **Priority 1**

4.10.5.3 When a data source is deleted, all subsets created from that data source are also deleted. **Priority 1**

4.10.5.4 When the user removes a data source from the project, the corresponding data file is deleted from the server. **Priority 1**

4.10.5.5 The user cannot remove a data set that is bound to widget. **Priority 1**

4.10.5.6 If the user attempts to delete a data set that has been bound to a widget, the user will be instructed to unbind the data set before it can be deleted. **Priority 1**

4.10.6 Binding Data to Widgets

4.10.6.1 A data set must be bound to a widget before that widget can access the data. **Priority 1**

4.10.6.2 An element only has access to data sets bound to its parent widget. **Priority 1**

4.10.7 Unbinding Data from Widgets

4.10.7.1 A bound data set can be removed from a widget. **Priority 1**

4.10.7.2 A bound data set can only be removed from a widget if it is not used by the widget or any of its elements. **Priority 1**

4.11 Actions

4.11.1 Creating an Action

4.11.1.1 To create an action, the user must supply the following information:

4.11.1.1.1 A valid action type, as specified in 4.11.2. **Priority 1**

4.11.1.1.2 The target affected when the action is fired. **Priority 1**

4.11.1.1.3 An optional data set. **Priority 1**

4.11.1.1.4 The properties or data subset to change. **Priority 1**

4.11.1.1.5 The new value of the properties or data subset query. **Priority 1**

4.11.1.1.6 Whether to apply the action immediately or in response to an event. **Priority 1**

4.11.2 Action Types

4.11.2.1 A valid action type is one of the following:

4.11.2.1.1 Modify property values of the selected widget or element. **Priority 1**

4.11.2.1.2 Modify the query of a data subset. **Priority 1**

4.11.3 Modifying an Action

4.11.3.1 The user can modify an existing action. **Priority 1**

4.11.4 Removing an Action

4.11.4.1 The user can remove existing actions. **Priority 1**

4.11.4.2 If the user attempts to remove an action used by an event, the user is alerted that the action cannot be removed until it is no longer used by any events. **Priority 1**

4.12 Events

4.12.1 Event Types

4.12.1.1 A valid event type is one of the following:

4.12.1.1.1 A click event, which fires an action when the user clicks on a trigger. **Priority 1**

4.12.1.1.2 A hover event, which fires an action when the user moves the mouse onto a trigger. **Priority 1**

4.12.1.1.3 A move event, which fires an action when the user moves the mouse while within the boundaries of a trigger. **Priority 1**

4.12.2 Creating an Event

4.12.2.1 To create an event, the user must supply the following information:

4.12.2.1.1 One of the event types specified in 4.12.1. **Priority 1**

4.12.2.1.2 The name of a target widget or element. **Priority 1**

4.12.2.1.3 A trigger of the widget or element specified in 4.12.2.1.2 that will fire the event. **Priority 1**

4.12.2.1.4 One or more actions to occur when the event is triggered. **Priority 1**

4.12.3 Modifying an Event

4.12.3.1 The user can modify an existing event. **Priority 1**

4.12.3 Removing an Event

4.12.3.1 The application must allow the user to remove an added event from a widget or element. **Priority 1**

4.13 Google Analytics

4.13.1 A Google Analytics account can be associated with a project. **Priority 1**
4.13.2 The user must supply a Google Universal Analytics (UA) tracking code and a title when associating Google Analytics with the project. **Priority 1**
4.13.3 A valid Google UA tracking code consists of alphanumeric characters and hyphens. **Priority 1**
4.13.4 A valid title can only include alphanumeric, hyphen (-), underscore (_) and space characters. **Priority 1**

4.14 Project Tree

4.14.1 All widgets and elements in the project are displayed in a hierarchical format. **Priority 1**
4.14.2 The user can change the selected widget or element from the project tree. **Priority 1**

4.15 User Accounts

4.15.1 Creating User Accounts

4.15.1.1 When creating a user account, the following information must be supplied:
4.15.1.1.1 E-mail Address **Priority 3**
4.15.1.1.1.1 A valid email address has up to 254 characters. **Priority 3**
4.15.1.1.1.2 A valid email address must be of the format `address@domain.extension` where address, domain and extension are not blank. **Priority 3**
4.15.1.1.2 Password **Priority 3**
4.15.1.1.2.1 A valid password has between 6 and 30 characters. **Priority 3**
4.15.1.1.2.2 A valid password must include at least one uppercase letter, one lowercase letter and one number. **Priority 3**
4.15.1.1.3 First Name **Priority 3**
4.15.1.1.3.1 A valid first name has between 1 and 50 characters. **Priority 3**
4.15.1.1.3.2 A valid first name can only include alphabetic, hyphen (-) or space characters. **Priority 3**
4.15.1.1.4 Last Name **Priority 3**
4.15.1.1.4.1 A valid last name has between 1 and 50 characters. **Priority 3**
4.15.1.1.4.2 A valid first name can only include alphabetic, hyphen (-) or space characters. **Priority 3**
4.15.1.1.5 Role **Priority 3**
4.15.1.5.1 A user account must have one of the following roles:
4.15.1.5.1.1 User **Priority 3**
4.15.1.5.1.2 Administrator **Priority 3**
4.15.1.2 Only a user with the Administrator role (4.1.1.5.1.2) can create a new user account. **Priority 3**
4.15.1.3 There must always be at least one user account with the Administrator role. **Priority 3**

4.15.2 Modifying User Accounts

4.15.2.1 A user with the administrator role can modify user accounts. **Priority 3**

4.15.2.2 The modified user account must continue to meet the details specified 4.15.1.1. **Priority 3**

4.15.3 Deleting User Accounts

4.15.3.1 A user with the administrator role can delete user accounts. **Priority 3**

4.15.3.2 Once a user account is deleted, that user can no longer log into the application. **Priority 3**

4.15.4 User Account Authentication

4.15.4.1 A user cannot access the contents of the application if he or she has not been authenticated. **Priority 3**

4.15.4.2 The user is authenticated if he or she supplies a valid email address and password that match an existing user account. **Priority 3**

5. Nonfunctional Requirements

5.1 Human Factors

5.1.1 Types of Users

5.1.1.1 Users that have little to no programming experience that will not be directly modify the underlying code of an exported visualization created in WAVED.

5.1.1.2 Users that have the capability to directly modify the underlying code of an exported visualization created in WAVED.

5.2 Documentation

5.2.1 Interactive Visualization Documentation

5.2.1.1 Documentation must be provided on the use of the WAVED application to create interactive visualizations.

5.2.2 Application Architecture Documentation

5.2.2.1 Documentation must be provided on the architecture of the system in order to specify how the functional requirements are implemented.

5.2.2.2 Documentation must be provided on how to modify the WAVED system to add or remove features.

5.3 Hardware

5.3.1 The client must have a computer with the following hardware:

5.3.1.1 A keyboard

5.3.1.2 A pointing device

5.3.1.3 A monitor.

5.3.2 The client's computer must have a network connection.

5.3.3 Server hardware must have a network connection.

5.4 Software

5.4.1 The WAVED application must function properly in the following Internet browsers:

5.4.1.1 Google Chrome version 31 or higher

5.4.1.2 Mozilla Firefox version 26 or higher

5.4.2 The server must support database software.

5.4.2.1 The database software must support multiple, concurrent connections.

5.4.3 The server must support web application hosting.

5.4.3.1 The hosting software must support multiple, concurrent connections.

5.5 Performance

5.5.1 Response Time

5.5.1.1 User interaction with the system that does not require the loading of data or off-client communication must be under 0.5 seconds.

5.5.1.2 The system response for an export request must be under 0.5 seconds. The actual download time of the visualization may vary depending on Internet connection speeds.

5.5.1.3 The system response time must be less than 1 second when loading and saving data.

5.6 Error Handling and Reliability

5.6.1 System Downtime

5.6.1.1 The acceptable downtime for the application can vary on a case-by-case basis, provided the end users are notified in advance by a system administrator.

5.6.1.2 The database server must not be down during times WAVED is available to users.

5.6.1.3 The database must be not be down during times when WAVED is available to users.

5.6.2 Input Errors

5.6.2.1 When an input error occurs, the system must display an error message to the user.

5.6.2.2 The user must acknowledge the error message in 5.5.2.1 before continuing to use the application.

5.7 Security

5.7.1 Access to the WAVED application is restricted to users on the network where WAVED is hosted.

5.7.2 Database access is restricted to a system administrator and any other personnel the administrator authorizes.

6. System Evolution

To narrow the scope of initial development, the current requirements are based on two expectations. The first expectation is that only a few widgets are needed to support Climate Central's immediate use cases. The second expectation is that WAVED is not public facing, but rather a tool used on an internal network, so an open permissions model will suffice. As the system evolves, changes to WAVED can add functionality beyond these assumptions.

6.1 New Widgets

Future modifications to the system include expanding the set of available widgets by implementing new plugins. New widgets may include:

- Vector maps of geographic areas other than the United States
- Widgets using third party mapping APIs such as Google Maps, Leaflet, Cesium, etc.
- Different types of graphs (e.g. bar graph, pie chart, etc.)

6.2 User Permissions

Some future modifications to WAVED will require expanding the user permissions model. Rather than using an open permissions model, users will be required to register and log in to WAVED with individual user accounts. Restrictions will be placed on the projects users are allowed to load and modify. Only users who are the owner, or who are given explicit permission from the owner, will be able to load and make changes to a given project within the WAVED development environment. Expanding the user system will also allow WAVED to associate uploaded data files with individual users such that users can use the same data files across multiple projects without having to re-upload them. Finally, stricter user permissions means that WAVED can evolve to allow users to permanently delete projects they previously created and saved to the server.