



Bilkent University

Department of Computer Engineering

Senior Design Project

Willow: Graph-Based Browsing

Project Specification Report

Efe Dağdemir, Tuana Türkmen, Sezin Zeydan, Can Cebeci, Cem Cebeci

Supervisor: Uğur Doğrusöz

Jury Members: Dr. Can Alkan, Dr. Çiğdem Gündüz-Demir

Progress/Requirements Specification

October 12, 2020

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

1.Introduction	3
1.1 Description	3
1.2 Constraints	4
1.2.1 Implementation Constraints	4
1.2.2 Economic Constraints	4
1.2.3 Ethical Constraints	4
1.2.4 Sustainability Constraints	5
1.2.5 Time Constraints	5
1.3 Professional and Ethical Issues	5
2.Requirements	6
2.1 Functional Requirements	6
2.1.1 Core Functionalities	6
2.1.2 Extended Functionalities	7
2.2 Non-functional Requirements	7
3.References	9

Project Specification Report

Willow: Graph-Based Browsing

1. Introduction

This report contains information about the description, the surrounding constraints, professional and ethical issues and the functional and non-functional requirements of the project. The internet has become an essential part, almost a necessity, in our everyday lives. We make use of the internet in so many different ways but one of the most important ones is through surfing the web. Millions of people are accessing the World Wide Web every single minute and web browsers are helping them do it. Although reaching the web by a browser is simple and easy, navigating between numerous tabs and managing different browsing sessions are not. One can easily get lost in between abundant tabs and lose track of the tasks in hand. As the number of tabs get larger, the amount of time lost due to confusion increases. Furthermore, backtracking on multiple tabs gets troublesome on the current model offered by the browsers. The features offered by web browsers for such tasks are limited and outdated. For instance, the linear structure of the tabs becomes hard to use as the size of the tabs gets smaller and identifying the desired tab between many others becomes burdensome with each new tab. Additionally, the concept of not being able to access a collection of tabs from previous browsing experiences is a flexibility limitation for the users.

In the search for finding a solution to these problems, we came up with Willow. Willow is a software project that could help to ease the process of browsing via understandable and intuitive visual navigation, filtering techniques and session management features. Willow aims to make users save time and feel comfortable during browsing.

This report contains information about the description, the surrounding constraints, professional and ethical issues and the functional and non-functional requirements of the project.

1.1 Description

Willow will be a browser extension developed for the widely used web browser Google Chrome [1]. Willow will enable users to efficiently visualize and manage their browsing sessions in a graph-based, interactive structure in order to ease the process of

browsing. The goal of our project is to take the current method of browsing to the next level by providing a visual structure that allows for smooth and intuitive navigation between the web pages browsed by the users as well as management features for browsing sessions. The tool will visualize the session history and the currently open tabs of the users as a graph, from which users will be able to navigate to the websites with simple clicks. The graph structure itself will be navigable and zoomable to provide the users with the ability to target any part of the graph to their liking. Users will be able to modify the graph via trivial features such as deleting vertices and non-trivial features such as changing the position, size and color of individual vertices. Finally, the tool will allow users to save and restore their browsing sessions. Sessions will be saved/restored with that session's browsing history and open tabs along with their graph representations. Extended functionalities outside of the main features are planned for development in the case that required resources such as time are available. These extended functionalities include page recommendations for users based on their browsing history, clustering vertices by their relevancy through analyzing the content of the pages, automatic vertex size determination based on an estimation of the importance of the pages through user behavior and the analysis of the graph, vertex searching by keywords in the title or the content of the pages and lastly, sticky notes to allow users to take textual notes on the graph.

1.2 Constraints

1.2.1 Implementation Constraints

- Github and Git will be used for version control and to track the implementation process.
- JavaScript, HTML and CSS will be used for extension development.
- Chrome Extension API will be used to get and manipulate user data.
- Chrome Storage API will be used to store user data locally.

1.2.2 Economic Constraints

- The application will be free to all users.
- Publishing the extension on Chrome Web Store will require a fee.
- Development tools are available for free usage.

1.2.3 Ethical Constraints

- Users will have to allow their history data to be accessed by us.

1.2.4 Sustainability Constraints

- The application should be able to handle large and complex browsing sessions.
- As the application grows, feedback from users should be taken into account to adapt to common browsing practices.

1.2.5 Time Constraints

- A fully functioning version of the application should be implemented in advance of CSFair in order to get useful user feedback.

1.3 Professional and Ethical Issues

- Willow will conform to the rules of ACM Code of Ethics and Professional Conduct [2] and IEEE Code of Ethics [3].
- Users may be concerned about data privacy because Willow works with the histories of the users. However, we will not store the data in any remote server. The data will be stored locally.
- The licenses of all the used libraries, APIs and extra resources will be checked and they will be used in accordance with rules. All of them will be referenced.
- We attach great importance to the consent issue. Chrome explicitly informs the users of the permissions each extension requires which also applies to our software.
- The guarantee of not sharing the user data with any other third-party without the information of the user will be given.
- To address security and privacy vulnerabilities we rely on the security mechanisms implemented by the Windows file system and Google Chrome.
- Willow will be a closed source project.
- Since the usefulness of the tools and features and attractiveness of the interface are inherently subjective, user demand is important to us. The feedback of the users will be given importance for later implementations and development.
- The accuracy of the visualization of data is crucial so that it does not cause confusion and misleading [4].

2. Requirements

2.1 Functional Requirements

The functionalities proposed for Willow are divided into the following two categories based on essentiality.

2.1.1 Core Functionalities

- Visualization of the browsing session in a graph
 - A hideable side panel on the browser displays the graph.
 - The graph is a directed forest with a vertex for each web page visited in the current browsing session. An edge $u \rightarrow v$ signifies that page v was discovered (first accessed during the session) through page u . This allows for easy backtracking.
 - Vertices corresponding to pages that are open in a tab are visually specified. The active tab is also highlighted.
- Organization and visualization of the browsing history
 - The browsing history is portrayed as a set of sessions.
 - Any number of sessions may be chosen to be visualized and displayed as a merged graph.
- Navigation among websites using the graph
 - Clicking on a vertex in the graph opens the corresponding page. If the page is currently open in a tab, the tab is activated. If not, the page may be opened either in a new tab or in the active tab.
 - Tabs can be closed through the graph as well, eliminating the need to search for the tab to be closed in a crowded tab list.
- Navigation through the graph
 - The graph can be panned and zoomed in order to focus on different parts of the browsing session.
- Modification/customization of the graph
 - While the application provides an intuitive and attractive default layout, the user can customize it and express the relationship between vertices by changing the size and position of each vertex or their colors.
 - Vertices can be deleted, appending their outgoing edges to their parent. This allows the user to get rid of irrelevant pages stumbled upon during the session.

- Session saving and restoration
 - The user can save their current session and restore them later on. The graph representation is restored along with the open tabs.

2.1.2 Extended Functionalities

- Page recommendations
 - New pages are recommended to the user by analyzing their browsing history.
- Vertex clustering
 - The content of pages is analyzed to find relevant vertices and relate them visually.
- Automatic vertex size determination
 - The sizes of vertices are determined by estimating their importance through the analysis of the session graph, the page content or user behavior.
- Vertex search
 - The pages in the graph that contain some keywords either in their title or in their content are found and presented to the user.
- Sticky notes
 - The user can create textual notes to be displayed on the graph as reminders or additional information.
 -

2.2 Non-functional Requirements

- **Performance**

The response time of the extension should be at most 30 milliseconds.

- **Scalability**

The performance degradation of the extension with respect to the number of vertices should be sublinear.

- **Usability**

The User Interface of the application should be intuitive enough that the average user can use the extension to its full potential within 10 minutes of exploration.

- **Extensibility**

The architecture of the software should be extensible enough to enable us to add new features based on user feedback before CSFair.

- **Availability**

The extension should be made available on the Chrome Web Market [5] free of charge to reach a vast amount of users.

3. References

- [1] "Google Chrome - Download the Fast, Secure Browser from Google," *Google*. [Online]. Available: <https://www.google.com/chrome/?brand=CHBD>. [Accessed: 11-Oct-2020].
- [2] "The Code affirms an obligation of computing professionals to use their skills for the benefit of society.," Code of Ethics. [Online]. Available: <https://www.acm.org/code-of-ethics>. [Accessed: 10-Oct-2020].
- [3] "IEEE Governing Documents," IEEE. [Online]. Available: <https://www.ieee.org/about/corporate/governance/index.html>. [Accessed: 10-Oct-2020].
- [4] T. Bisoux, "The Ethics of Data Visualization," The Ethics Of Data Visualization | BizEd Magazine, 28-Dec-2019. [Online]. Available: <https://bized.aacsb.edu/articles/2020/january/the-ethics-of-data-visualization>. [Accessed: 09-Oct-2020].
- [5] "Chrome Web Store," *Google*. [Online]. Available: <https://chrome.google.com/webstore/category/extensions>. [Accessed: 11-Oct-2020].