



Bilkent University

Department of Computer Engineering

Senior Design Project

Willow: Graph-Based Browsing

Project Analysis Report

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

Supervisor: Uğur Doğrusöz

Jury Members: Çiğdem Gündüz Demir, Can Alkan

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	2
2. Current System	2
3. Proposed System	3
3.1 Overview	3
3.2 Functional Requirements	3
3.2.1 Core Functionalities	3
3.2.2 Extended Functionalities	4
3.3 Nonfunctional Requirements	4
3.4 Pseudo Requirements	5
3.4.1 Implementation Requirements	5
3.4.2 Economic Requirements	5
3.4.3 Sustainability Requirements	5
3.4.4 Time Constraints	5
3.5 Use Scenarios	5
3.5 System Models	7
3.5.1 Use Case Model	7
3.5.4 Dynamic Models	14
3.5.4.1 Accessing previous sessions through Chrome's history page	14
3.5.4.2 Different states of a node	15
3.5.5 User Interface and Screen Mock-ups	15
4. Other Analysis Elements	16
4.1. Consideration of Various Factors in Engineering Design	16
4.2. Risks and Alternatives	17
4.3. Project Plan	18
4.4. Ensuring Proper Teamwork	27
4.5. Ethics and Professional Responsibilities	28
4.6. Planning for New Knowledge and Learning Strategies	28
5. Glossary	29
6. References	30

1.Introduction

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 a solution to these problems, we came up with Willow. Willow is a Chrome Extension 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 is the analysis report for Willow which will convey the design and implementation ideas for Willow in an understandable manner. Our system models are explained through various UML diagrams. The report also contains analysis of Willows' social impact, our team work and feasibility constraints. In the very end we there is a small glossary, which a reader may refer to if there is any confusion regarding the terminology we use.

2. Current System

The current system is Google Chrome's tab management system[1], which is not very flexible for users that work with many tabs open. Right now, Google Chrome only offers tab management through features such as;

- All tabs open on a Chrome window will appear on top of the window side to side based on the order in which they were opened.
- If a user wants to, they can change the order of tabs that appear on top of a specific Chrome window.
- Users can move tabs between different Chrome windows.
- Users can hover their mouse on a tab and it will show a tooltip containing a brief description of the web page such as its name and title.
- Users can retrieve closed tabs from their history.
- Users can edit their history.
- Users can bookmark the tabs they deem important and have quick access to those tabs through Bookmarks.
- Users can edit their Bookmarks.
- When users shut down the chrome app and then launch Chrome again, all web pages that were open before can be restored.
- The relationships that exist among tabs is not and can not be indicated, except by the primitive feature of tab groups, which can be constructed manually to underline chosen tabs with the same color and keep them adjacent in the order the tabs are displayed in.

Although the above features are useful they are not enough for a user that surfs the web with more than 30 tabs open on multiple Chrome windows. As the amount of open tabs grow, Google Chrome's tab management system starts to lose its use.

3. Proposed System

To address the problems that arise with the current system, we propose a new version of web surfing with a Chrome extension called Willow. Willow will introduce a new way of organizing your tabs as well as providing an intuitive visual model of the user's browsing session, which will indicate the chronological and hierarchical relationships among tabs and previously visited web pages.

3.1 Overview

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 in a browsing session as well as management features for browsing sessions.

3.2 Functional Requirements

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

3.2.1 Core Functionalities

- Visualization of the browsing session in a graph
 - A hideable side panel on the browser displays the session graph.
 - The graph is a directed forest with a node 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.
 - Nodes corresponding to pages that are open in a tab (open nodes) are visually specified. The active tab is also highlighted.
- 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 linear tab list.
- Navigation through the graph
 - The graph can be panned and zoomed in order to focus on different parts of the browsing session.
 - The pages in the graph that contain some keywords either in their title or in their content are found and presented to the user.
- Organisation and visualisation of the browsing history
 - The browsing history is portrayed as a set of sessions.
 - To navigate around the browsing history, a session may be chosen to be visualized. The graph related to this session is displayed and can be searched and traversed, just like an ordinary session graph.

- 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 it later on. The graph representation is restored along with the open tabs.
 - When a user saves the session by default that session is named with the save date. However users can change the name of sessions and save it with a different name.

3.2.2 Extended Functionalities

- Page recommendations
 - New pages are recommended to the user by analyzing their browsing history.
- Vertex clustering
 - The content of pages are 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 behaviour.
- Sticky notes
 - The user can create textual notes to be displayed on the graph as reminders or additional information.

3.3 Nonfunctional Requirements

- Performance

The response time of Willow should be at most 30 milliseconds.
- Scalability

The performance degradation of Willow with respect to the number of vertices should be sublinear.
- Usability

The user interface of Willow should be intuitive enough that the average user can use the application 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

Willow should be made available on the Chrome Web Market [2] free of charge to reach a vast amount of users.

3.4 Pseudo Requirements

3.4.1 Implementation Requirements

- Github 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.

3.4.2 Economic Requirements

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

3.4.3 Sustainability Requirements

- 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.

3.4.4 Time Constraints

- A fully functioning version of the application should be implemented early enough to allow for sufficient time to accumulate a large amount of users, receive user feedback and produce updated versions of the software before CSFair. We plan to have a fully functioning prototype by the end of the first semester.

3.5 Use Scenarios

Scenario 1 - Locating an Open Tab

- Ege launches Google Chrome to start researching Turkey.
- Ege navigates to the Wikipedia page about Turkey. Which contains links to other sources about History, Politics, Culture.
- Ege opens 3 new tabs from History, navigates to another page from Politics and backtracks to Wikipedia from Culture.
- Ege continues browsing in the same session which results in numerous tabs on top of his screen.
- Ege is sure he opened up an article about Human Rights in Turkey in one of his tabs but struggles to find it.
- Ege clicks on the extension icon for Willow near the address bar to display the overlay.
- Willow's graph structure shows Ege the structure of his browsing session and marks his open tabs on the graph.
- The graph helps Ege locate the tab containing the article through the following mechanisms:
 - He sees a preview of each tab's content on its related node.
 - He sees pages that are potentially related to his target page and follows the path of nodes to find it.

- He uses the search utility and Willow locates the tab for him.
- Ege clicks on the node to navigate to the target web page.

Scenario 2 - Restoring a Session (Loading a Session)

- After a thorough research on Turkey, Ege decides to take a break. Ege saves the current session with the name "History, Politics and Culture of Turkey" and then closes the session.
- When Ege wants to continue his research on Turkey. He realizes that he still has a lot of information he acquired during his previous and wants to continue from there.
- Ege goes to the history page and accesses the previous sessions.
- Ege is now met with a list of his previous sessions. The sessions are listed in a reverse time order in which last session saved is shown on top.
- Ege can also find the session he wants to restore by typing the name he saved it with: History, Politics and Culture of Turkey.
- After ege clicks the old session is restored along with all its tabs and graph structure.
- Ege can continue his research exactly where he left off with Willow's help.

Scenario 3 - Accessing Pages from Previous Sessions

- After a couple of days, Ege is now browsing about Politics in general.
- Again after opening a couple of tabs, he reads about an article which describes a political meeting.
- He remembers that in the research he made about History, Politics and Culture of Turkey a couple of days ago this meeting was mentioned in a web page.
- He decides that he wants to find that page and see the parts about this meeting.
- In order to do this, he again opens the previous sessions. However this time, instead of launching the old session, by using the methods told above in *Scenario 2 - Restoring a Session* he just finds the node (the web page containing the information about the meeting) in the tree and clicks on that.
- After this, the page is opened in a new tab in his browser and a node representing the page is added to his current session graph.

Scenario 4 - Displaying the Browsing Session Through the Graph (Restoring Closed tabs)

- Ege keeps researching about politics.
- Ege finds a page containing links to websites about the politics of different countries.
- He decides to read about the politics of a few countries from that website.
- After reading about three countries, he discovers other links about the politics of Russia and researches further in that area.
- After he is satisfied with his research about Russian politics, he loses the big picture of what he had been doing.
- He opens the graph representing his session and is quickly reminded of his research process.
- Inspecting the graph, he checks which countries he already has read about and understands how much he has researched each one.
- He clicks on the node for the website that contains links for different countries, which is not open in any tab anymore, and the website is opened in a new tab..
- He continues his research from that point.

3.5 System Models

We provide different use cases and an object diagram as static models. As a dynamic model, we provide an activity diagram. We also provide two mockups of our interface design.

3.5.1 Use Case Model

The use cases are split into four groups:

Visual Paradigm Online Diagrams Express Edition

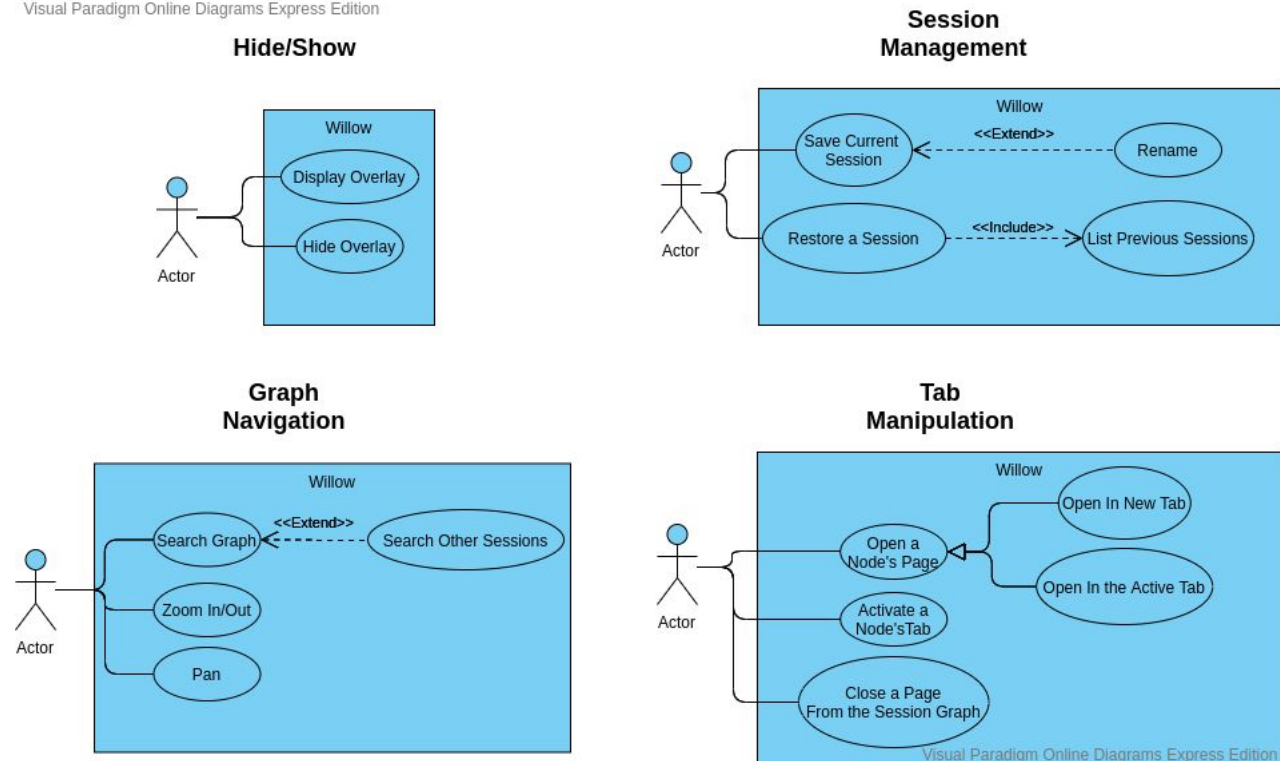


Figure 1: Use Case diagram

Use Case Descriptions

1. Hide/Show Use Case Descriptions

Use Case 1

Use Case Name: Display Overlay

Participant Actor: User

Flow of Events:

1. User presses the extension icon.
2. Willow displays the browsing graph of the current session.

Entry Condition: The overlay is not displayed.

Exit Condition: The overlay is displayed.

Use Case 2

Use Case Name: Hide Overlay

Participant Actor: User

Flow of Events:

- 1.The user presses the extension icon.
- 2.Willow removes the drawn browsing graph.

Entry Condition: The overlay is displayed.

Exit Condition: The overlay is not displayed.

2. Session Management Use Case Descriptions

Use Case 3

Use Case Name: Save Current Session

Participant Actor: User

Flow of Events:

1. The user presses the save button.
2. Willow saves the current browsing graph to the local filesystem with a name based on the date and time.

Entry Condition: No entry condition.

Exit Condition: No exit condition.

Use Case 4

Use Case Name: Rename

Participant Actor: User

Flow of Events:

- 1.While saving the session, the user labels it with a name rather than using the default name that is based on the date and time.
2. Willow saves the session with that name instead.

Entry Condition: The session has a time-based default name.

Exit Condition: The session has a user-assigned name.

Use Case 5

Use Case Name: Lists previous sessions

Participant Actor: User

Flow of Events:

- 1.The user presses the list previous sessions button.
2. Willow displays a list of the previously saved sessions.

Entry Condition: There exist saved sessions.

Exit Condition: The user sees a list of the saved sessions.

Use Case 6

Use Case Name: Restore a session

Participant Actor: User

Flow of Events:

- 1.The user lists the previous sessions and chooses a session to be restored.
- 2.Willow saves the current session and replaces the session graph with that of the session being restored.
- 3.Willow restores the open tabs from the previous session.
- 4.Willow navigates the user to the tab that was active when the previous session was saved.

Entry Condition: The current session may not be saved. The browsing graph reflects the structure of the current session. Chrome contains tabs that are open in the current session.

Exit Condition: The current session is saved. The browsing graph reflects the structure of the restored session. Chrome contains the open tabs from the restored session.

3. Graph Navigation Use Case Descriptions

Use Case 7

Use Case Name: Search Graph

Participating Actor: User

Main Flow of Events:

1. User clicks on the search icon.
2. Willow displays the search bar.
3. User writes what they want to search on the search bar.
4. Willow highlights the node that contains the searched page.

Entry Condition: User has opened the session graph.

Exit Conditions:

- The node to be searched is found and highlighted on the graph.
- The node to be searched is not found, no node is highlighted.

Alternate Flow of Events:

1. User clicks on search icon.
2. Willow displays the search bar.
3. User writes what they want to search on the search bar.
4. No node meets the search criteria and nothing happens.

Use Case 8

Use Case Name: Zoom In/Out

Participating Actor: User

Flow of Events:

1. User uses zoom in/out move on the mouse on a specific place on the session graph.
2. Willow displays the zoomed in/out version of the graph.

Entry Condition: User has opened the session graph.

Exit Condition: Graph is zoomed in/out.

Use Case 9

Name: Pan

Participating Actor: User

Flow of Events:

1. The user drags the mouse to the direction of the session graph they want to go to.
2. Willow shifts the display towards the indicated direction of the graph.

Entry Condition: User has opened the session graph

Exit Condition: Part of the graph that is in the panned direction is shown.

4. Tab Manipulation Use Case Descriptions

Use Case 9

Use Case Name: Open a Node's Page

Participating Actor: User

Flow of Events:

- 1.The user right clicks on a node.
- 2.The context menu is displayed.
- 3.The user chooses the "Open in New Tab" option.
- OR
- 4.The user chooses the "Open in the Active Tab" option.
- 5.The node's page is opened.
- 6.The node is marked as an open node in the session graph.

Entry Conditions:

- The user has opened the Willow overlay and the session graph is displayed.
- The node that the user interacts with is a closed node.

Exit Condition:

- The node's page is opened in a new tab.
- OR
- The node's page is opened in the active tab.

Use Case 10

User Case Name: Open in New Tab

Participating Actor: Inherited from Open a Node's Page use case.

Flow of Events:

- 1.The user right clicks on a node.
- 2.The context menu is displayed.
- 3.The user chooses the "Open in New Tab" option.
- 4.The node's page is opened in a new tab.
- 5.The node is marked as an open node in the session graph.

Entry Condition: Inherited from Open a Node's Page use case.

Exit Condition: Inherited from Open a Node's Page use case.

Use Case 10

Use Case Name: Open in the Active Tab

Participating Actor: Inherited from Open a Node's Page use case.

Flow of Events:

- 1.The user right clicks on a node.
- 2.The context menu is displayed.
- 3.The user chooses the "Open in the Active Tab" option.
- 4.The node's page is opened in the active tab.
- 5.The node is marked as an open node in the session graph.

Entry Condition: Inherited from Open a Node's Page use case.

Exit Condition: Inherited from Open a Node's Page use case.

Use Case 11

Name: Activate a Node's Tab

Participating Actor: Initiated by User

Main Flow of Events:

1. User right clicks on a non-active node .
2. Willow displays the context menu for the node.
3. User clicks on the "Activate node" option.
4. Willow activates the tab corresponding to that node so that the user is now on that page.

Entry Conditions: User has opened the current session graph

Exit Conditions:

- The active node is changed according to the selected node.
- The active node is not changed. Nothing extra is done.

Alternate Flow of Events:

1. User right clicks on the currently active node .
2. Willow displays the context menu for the node.
3. User clicks on the "Activate node" option.
4. Willow does not do anything.

Use Case 12

Name: Close a Page From the Session Graph

Participating Actor: Initiated by User

Main Flow of Events:

1. User right clicks on a non-active node .
2. Willow displays the context menu for the node.
3. User clicks on the "Close page" option.
4. Willow closes the web page corresponding to that node and removes that node from the session graph.

Entry Conditions: User has opened the current session graph

Exit Conditions:

- The tab(s) corresponding to the selected node is closed.
- The tab(s) corresponding to the selected node is closed and a new active node is chosen by Willow.

Alternate Flow of Events:

1. User right clicks on the currently active node.
2. Willow displays the context menu for the node.
3. User clicks on the "Close page" option.
4. Willow changes the active node and closes the web page corresponding to the selected node and removes that node from the session graph.

3.5.3 Object and Class Model

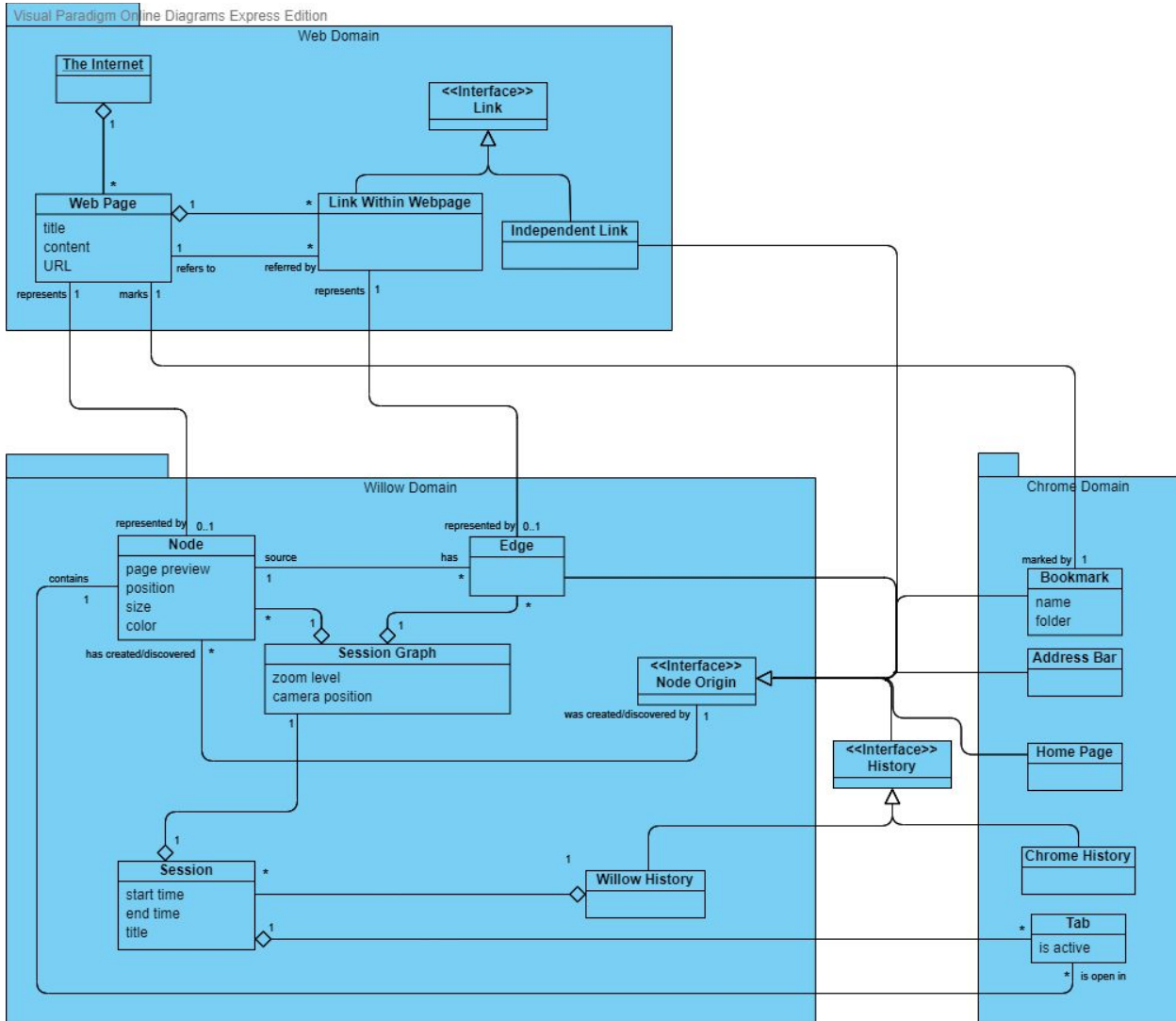


Figure 2: Class diagram

Notes and clarification:

- Nodes and edges are Willow domain representations of web pages and links. Therefore, it may seem redundant to introduce them as separate classes within our model. However, we find that this model, through the "0..1" multiplicities, illustrates the fact that only some of the web pages and links found in the web domain, namely those that have been traversed at least once, are associated with nodes and edges. Our model also separates the inherent properties of web pages from the additional attributes brought on in a willow domain representation.
- By "Independent Link", we denote links that are not contained in any webpage. For examples of such links, consider the links commonly found in documents within the user's file system or links sent to the user via chatting platforms such as Zoom or Discord, which, when clicked, automatically launch the user's browser and open the web page. The distinction between independent links and links within web pages is necessary. While both categories may cause the addition of new nodes to the session graph, links belonging to the latter category are potentially represented with an edge since they cause a transition between web pages. The former category, in contrast, can only create root nodes, for which there exists no parent page.
- Tabs are inherently in association with web pages. While we have excluded an explicit denotation of that association from our object model, it is still implicitly present in the association between tabs and nodes, which are themselves associated with web pages. This indirection signifies the fact a tab never contains a web page that is not represented as a node in the session graph.

3.5.4 Dynamic Models

3.5.4.1 Accessing previous sessions through Chrome's history page

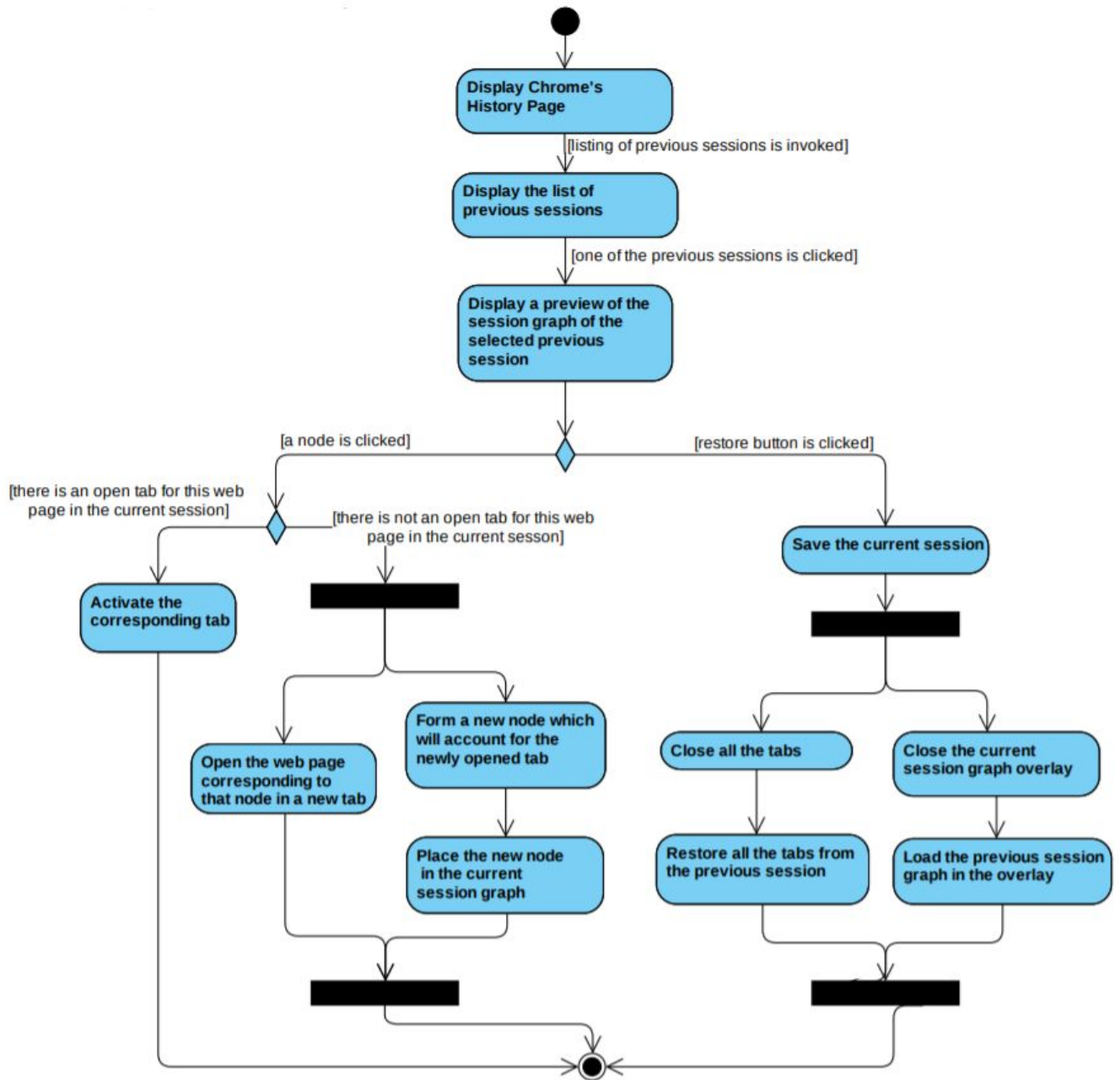


Figure 3: Activity diagram of accessing previous sessions

3.5.4.2 Different states of a node

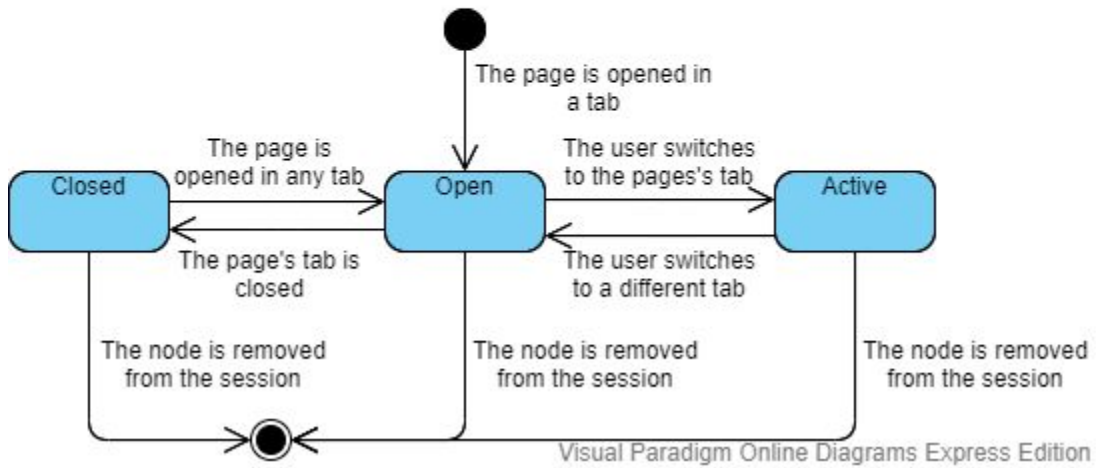


Figure 4: State diagram of a node

3.5.5 User Interface and Screen Mock-ups

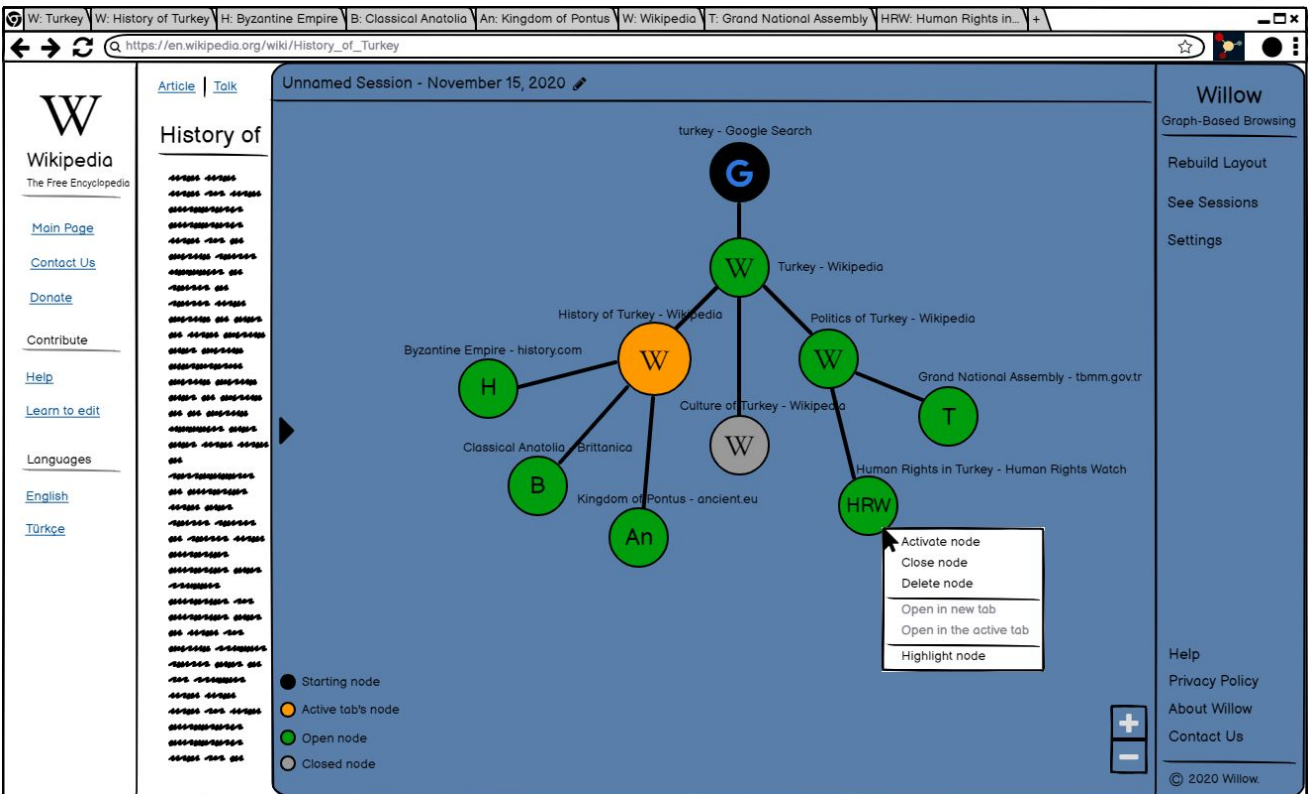


Figure 5: Mock-up of the Willow overlay displaying the session graph

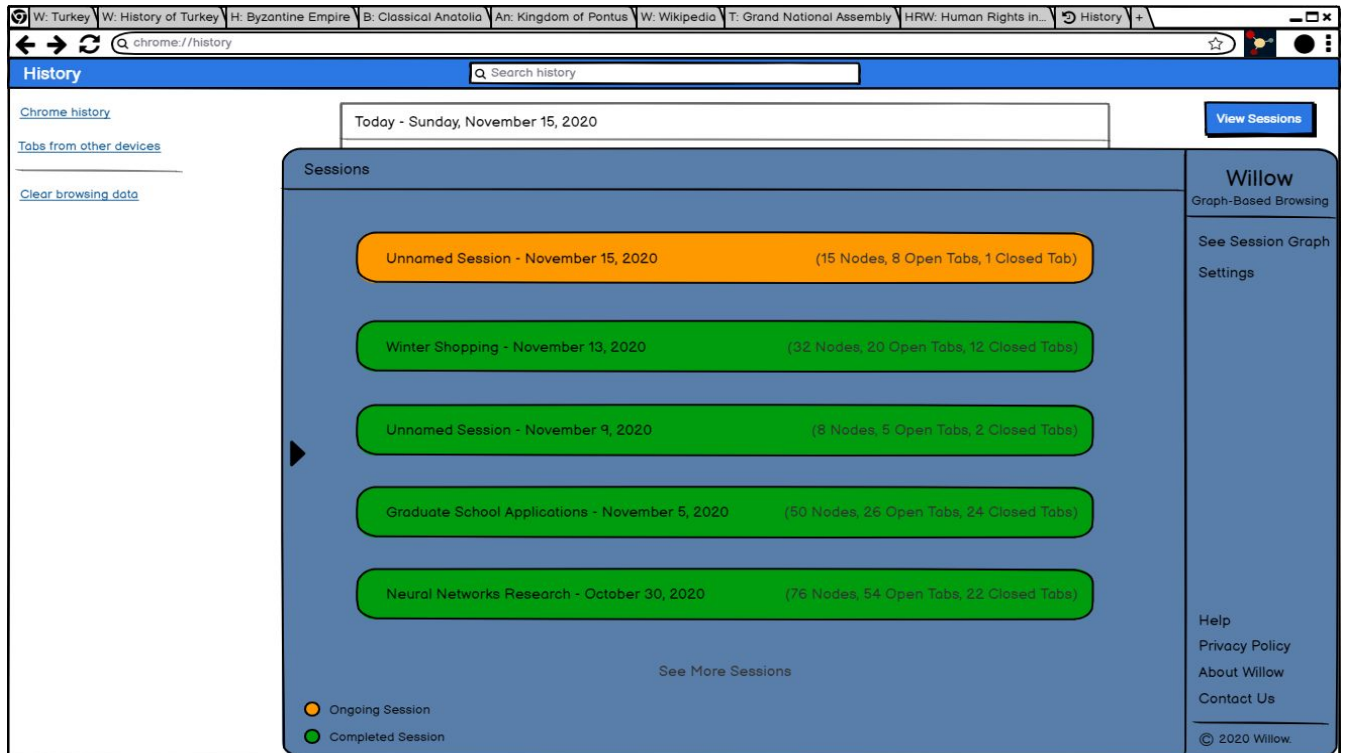


Figure 6: Mock-up of the Willow overlay displaying the session list

4. Other Analysis Elements

4.1. Consideration of Various Factors in Engineering Design

Willow is a tool designed to enhance people's internet browsing experience. Hence, it does not affect and is not affected by public health, public safety and public welfare in any way at all. Social factors are also irrelevant for the design of Willow.

Global Factors:

Willow is imagined as a software to be used by people all around the world. However, people speak many different languages all around the globe. Considering that most people that browse the internet speak English, having the interface in English will allow people from different countries to access the software.

Cultural Factors:

Different cultures have different understandings of privacy. The European Union has very strict rules about data protection. On the other hand, the Chinese are used to a much more invasive mode of surveillance. In designing Willow, the privacy boundaries of different cultures need to be respected.

Table 1: Factors

	Effect Level	Effect
Public Health	0	-
Public Safety	0	-
Public Welfare	0	-
Global Factors	4	Interface should be English
Cultural Factors	6	Privacy and Data Protection
Social Factors	0	-

4.2. Risks and Alternatives

Since we are going to introduce a new outlook on the way people surf the web, of course there will be an adaptation process where we will work with users to make this transition as easy as possible. This is why we want to finish Willow as soon as possible in order to get feedback from users and update Willow accordingly. We believe this process will minimize the risk of people not adapting to Willow and therefore lower the chances of failure.

Another risk with Willow is that we may face some synchronization issues regarding the session graph and the session itself in the browser. For example if the user tries to open multiple new tabs at the same time and the results may load late to the session graph. These kinds of issues may result in negative feedback from the users. If we are to encounter such synchronization problems we plan to use parallel computing so we can finish similar tasks synchronously.

We may also face the risk of not accumulating a large enough user base. This is a very important measure in success for us so we decided to have an alternative plan to expand our user base. As discussed in the below paragraph because the competing apps have a considerable amount of users we believe that Willow will meet the demand of many users so we think not acquiring enough users has low likelihood. However if we are to encounter this problem we plan to test Willow again, go over its faults and advertise.

There are many alternatives to Google Chrome Tab Management System but none of them are as visual as Willow. The biggest competitor is Tabs Outliner[3] Chrome extension. Currently Tabs Outliner has over 100.000 users they provide tab visualizing through a tree like structure but it's more text oriented and they also provide session management. However there is no graph-structure on Tabs Outliner and the user interface is not easy to use and also just like Google Chrome, Tabs Outliner underperforms when the amount of tabs opened on the browser increases. For these reasons Willow will be a breath of fresh air to everyone that can not manage their web surfing with existing options. We believe we have a more tidy and easy to use interface than Tabs Outliner and Tabs Outliner's success shows us there is a demand for extensions that make web surfing easier on Chrome. So after considering all these factors we believe that we can reach a good amount of active users.

Table 2: Risks

	Likelihood	Effect on the project	B Plan Summary
Users can not adapt to new way of browsing	High	Change on UI	Recreate UI from user feedback
Synchronization issues may lower quality of Willow	Low	Uncomfortable web surfing	Parallel computing
Not acquiring enough users	Low	Not getting user feedback	Advertise and fix problems that may keep users away

4.3. Project Plan

Our primary aim as Team Willow is to follow the software development life cycle properly. In order to achieve this we will analyze and design our project comprehensively and we plan on implementing a prototype based on our analysis and design before the end of the fall semester. Creating a prototype is important for us because we want to receive qualitative and quantitative feedback from users as soon as possible. This will allow us to test our product on the field and make continuous improvements in the necessary areas. After making sure our product is shaped to meet the initially planned functionalities and the demands of the users, we will move on to implementing our extended features. We plan on testing our extended features with users and receiving feedback in the same way with the prototype of the core product.

According to our plan we have 9 work packages: Analysis Report, High-Level Design Report, First Prototype, Marketing, Low-Level Design Report, Second Prototype, Marketing, Final Implementation Changes, Final Report. And we had one previous work package for Requirements Specification.

Table 3: List of work packages

WP#	Work package title	Leader	Members involved
WP0	Requirements Specification	Sezin Zeydan	All members
WP1	Analysis Report	Tuana Türkmen	All members
WP2	High-Level Design Report	Cem Cebeci	All members

WP3	First Prototype	Efe Dağdemir	All members
WP4	Marketing	Can Cebeci	Sezin Zeydan, Efe Dağdemir
WP5	Low-Level Design Report	Sezin Zeydan	All members
WP6	Second Prototype	Can Cebeci	All members
WP7	Marketing	Tuana Türkmen	Cem Cebeci
WP8	Final Implementation Changes	Cem Cebeci	All members
WP9	Final Report	Efe Dağdemir	All members

Table 4: Explanation of the work packages

WP 0: Requirements Specification			
Start date: Sep 16, 2020 End date: Oct 12, 2020			
Leader:	Sezin Zeydan	Members involved:	All members (Tuana Türkmen, Efe Dağdemir, Cem Cebeci, Can Cebeci)
Objectives: Name and briefly describe the project. Identify initial project requirements, constraints and professional & ethical responsibilities.			

Tasks:

Task 0.1 Functional Requirements : Identifying the functional requirements.

Task 0.2 Constraints: Determining the constraints.

Deliverables

DO.1: Project Specification Report

WP 1: Analysis Report

Start date: Oct 12, 2020 **End date:** Nov, 21 2020

Leader:

Tuana Türkmen

Members involved:

All members (Efe Dağdemir, Sezin Zeydan, Cem Cebeci, Can Cebeci)

Objectives: Having a comprehensive analysis of the project so that the design and implementation of the project goes as smoothly as possible.

Tasks:

Task 1.1 Research about the current system: The purpose is to know the current system. This is important for the design and implementation of the project because we would know what is already available, what should be extra implemented and what can be changed in the current system.

Task 1.2 Analyzing the proposed system: The purpose is to decide on the functionalities and the models of the system.

Task 1.3 Considering social conditions: The purpose is to decide on risks, project plan, responsibilities, etc.

Deliverables

D1.1: Analysis Report

WP2: High-Level Design Report

Start date: Nov, 21 2020 **End date:** Dec 21, 2020

Leader:

Cem Cebeci

Members involved:

All members (Efe Dağdemir, Sezin Zeydan, Tuana Türkmen, Can Cebeci)

Objectives: *Having a comprehensive design of the project so that the implementation of the project goes as smoothly as possible.*

Tasks:

Task 2.1 Architecture: *The purpose is to decide on which architecture we will use while implementing our project.*

Task 2.2 System Models: *The purpose is to have more precise system models that we will use during our implementation.*

Deliverables

D2.1: High-Level Design Report

WP3: First Prototype

Start date: Dec 21, 2020 **End date:** Before the end of fall semester

Leader:	<i>Efe Dağdemir</i>	Members involved:	<i>All members (Sezin Zeydan, Tuana Türkmen, Can Cebeci, Cem Cebeci)</i>
Objectives: <i>Implementing a presentable prototype with basic functionality.</i>			
Tasks:			
Task 3.1 Implementation: <i>Implementing the project so that it has basic functionality and can be tested by presenting to users.</i>			
Deliverables			
D3.1: <i>A basic functioning prototype</i>			
WP4: <i>Marketing</i>			
Start date: After implementation of the prototype End date: Before the demo on the fall semester			
Leader:	<i>Can Cebeci</i>	Members involved:	<i>Sezin Zeydan, Efe Dağdemir</i>
Objectives: <i>Marketing strategies for populating the product between users.</i>			
Task 4.1 Social Media: <i>Using social media platforms to introduce users with the product.</i>			

WP5: *Low-Level Design Report*

Start date: *1st week of the spring semester* **End date:** *3rd week of the spring semester*

Leader: *Sezin Zeydan*

Members involved:

All members (Efe Dağdemir, Tuana Türkmen, Can Cebeci, Cem Cebeci)

Objectives: *Having a comprehensive low-level design of the project so that the implementation of the project goes as smoothly as possible.*

Tasks:

Task 5.1 Architecture: *The purpose is to decide on which architecture we will use while implementing our project.*

Task 5.2 System Models: *The purpose is to have more precise system models that we will use during our implementation.*

Deliverables

D5.1: *Low-Level Design Report*

WP6: *Second Prototype*

Start date: *3rd week of the spring semester* **End date:** *Before the Final Report deadline*

Leader: *Can Cebeci*

Members involved:

All members (Sezin Zeydan, Tuana Türkmen, Efe Dağdemir, Cem Cebeci)

Objectives: Implementing a second prototype according to the user feedback we received which optimistically will have more features.

Tasks:

Task 6.1 Changes: *The purpose is to change the first prototype according to the feedback received.*

Task 6.1 Extended Features: *The purpose is to add the extended features to the implementation.*

Deliverables

D6.1: Second Prototype

WP7: *Marketing*

Start date: After implementation of the second prototype **End date:** Before the final changes

Leader: *Tuana Türkmen*

Members involved:

Cem Cebeci

Objectives: *Marketing strategies for populating the product between users.*

Task 7.1 Social Media: *Using social media platforms to introduce users with the product.*

WP8: Final Implementation Changes

Start date: After implementation and marketing of the second prototype End date: Before CSFair			
Leader:	<i>Cem Cebeci</i>	Members involved:	<i>All members (Efe Dağdemir, Sezin Zeydan, Tuana Türkmen, Can Cebeci)</i>
Objectives: <i>Having the final implementation of the project.</i>			
Task 8.1 Finalizing: <i>Finalizing the project implementation according to the latest user feedback and having an extensive testing procedure.</i>			
Deliverables			
D8.1: <i>Final Product</i>			
WP9: <i>Final Report</i>			
Leader:	<i>Efe Dağdemir</i>	Members involved:	<i>All members (Sezin Zeydan, Tuana Türkmen, Can Cebeci, Cem Cebeci)</i>
Objectives: <i>Provide a comprehensive final report of the project that gives complete information about the system.</i>			
Tasks:			
Task 9.1 Architecture: <i>Final architecture of the system.</i>			
Task 9.2 Maintenance Plan: <i>Provide a maintenance plan for the system.</i>			
Task 9.3 Social Impact: <i>Analyse social impact of the system.</i>			

Deliverables

D9.1: Final Report

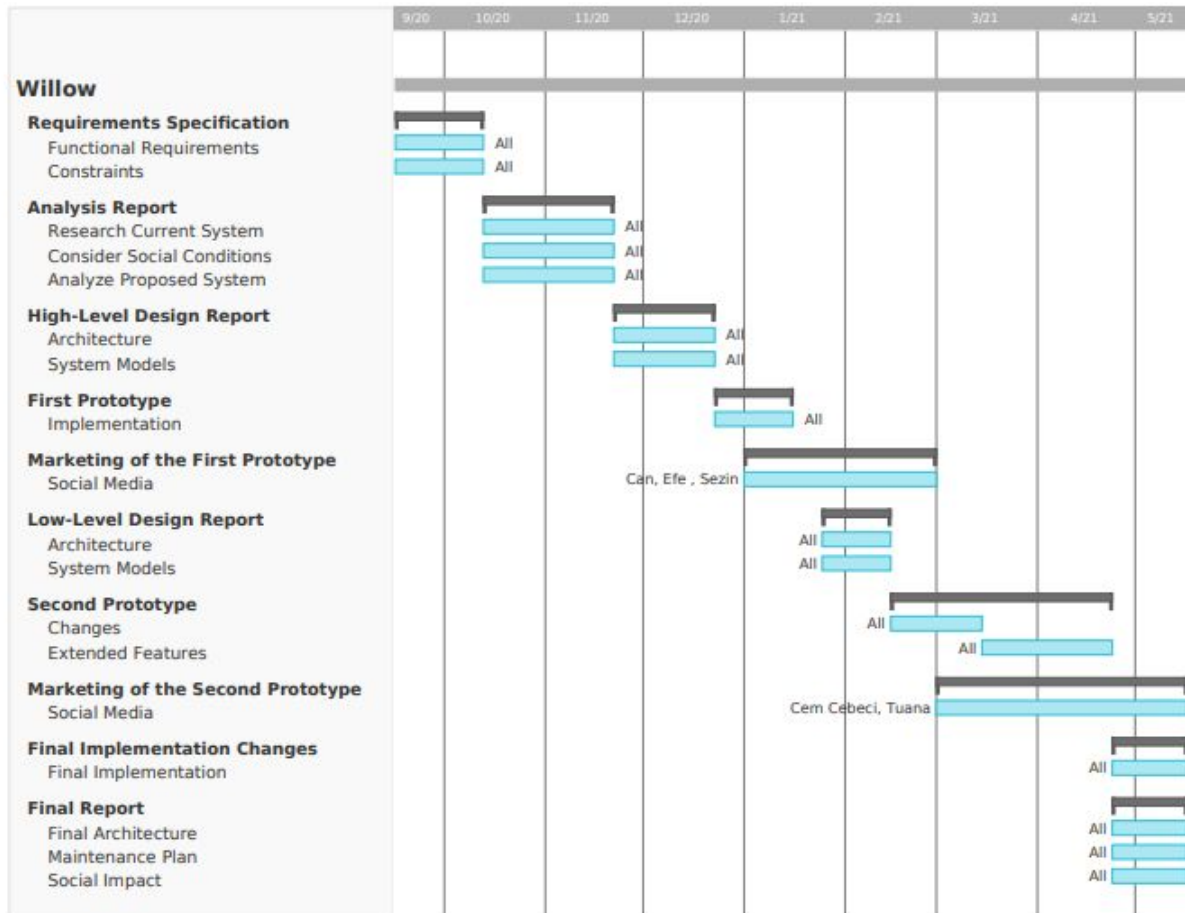


Figure 7: Gantt chart of the project plan

4.4. Ensuring Proper Teamwork

As the Willow team we believe that the road to ensuring proper teamwork is through healthy communication. To do that we meet at least once a week and discuss our progress that week. In these meetings new tasks are assigned to every group member in a fair manner and important decisions about design, implementation or feasibility are made. We do every task on a schedule. We give proper deadlines for each task and try to keep up with those deadlines and this ensures that we keep proper time tables and never get behind schedule. Also, all of our team members have previously worked on various course projects together. We believe that our familiarity with each other's strengths, weaknesses, working habits and communication styles will help us maintain a respectful and comfortable work environment and keep us on track.

Another caution we take to ensure proper teamwork is that we do not do every task individually. For example if it is a very big task we divide the task in parts and we also divide the group and then start doing it each smaller task for each group. When it's time to combine the mini tasks we come together to review and discuss the tasks done by each group. This approach is advantageous because it is very efficient and also gives everyone in the group to review the task as a whole in the end.

4.5. Ethics and Professional Responsibilities

Willow will follow ACM Code of Ethics and Professional Conduct [4] and IEE Code of Ethics [5].

In projects where the history of a user is managed, privacy is an important value. However, Willow will not store its users' data on a remote server. Users' data will be stored locally. Additionally, we will not share the data with any other third-party without informing the users.

The sources (libraries, APIs, etc.) we will use will all be used according to their licences and the correct reference to them will be done. And lastly, we give importance to the accuracy of used data in order to not deceive our users. Thus, the accuracy of visualization of the data will be given great importance [6].

4.6. Planning for New Knowledge and Learning Strategies

Firstly, while developing Willow, we plan on enhancing our JavaScript knowledge and coding skills. Most of the participants in our group have not used JavaScript very much in their life. And since Chrome extensions are implemented with JavaScript we have the chance to change this. We plan on doing this via online tools (online tutorials, YouTube videos and example codes).

Secondly, we will learn about extensions. None of us has worked with extensions before thus we do not have very broad information about them. Chrome has a very informative web site about its extensions [7] for developers. First it tells what extensions are and then goes on to explaining the separate components of extensions and how to implement them. It provides a quick tutorial and a developer guide. Then gives sample extension codes. Thus, all of these will be used to improve our knowledge about extensions.

Lastly, we will learn how to use the new libraries we will utilize. For this, we will experiment with the libraries and understand the purposes of their components. Then we will start to use them in our own implementation.

5. Glossary

- **Session:** Sessions are continuous instances of browsing activity. A new session starts every time the user launches the Chrome app. The session is terminated upon exit. Sessions can be saved and restored, allowing them to last across multiple runs of Chrome.
- **Session Graph:** A visual graph structure that shows the relationships between nodes. The session graph contains an edge $u \rightarrow v$ if and only if the first instance of access to the website associated with node v was through a link contained in the website associated with node u .
- **Node:** A vertex on the session graph which represents a web page visited within the session.
- **Open Node:** A node whose associated page is currently open in a tab.
- **Closed Node:** A node whose associated page was visited within the session and is not currently open in a tab. The tab that once contained the webpage need does not have to be closed, it may simply be navigated to another webpage.
- **Active Tab:** The tab whose content is currently displayed. At any time, the browser contains exactly one active tab and any number (possibly zero) of inactive tabs.
- **Active Node:** The node that is associated with the webpage contained in the active tab. Trivially, there is exactly one active node at all times and the active node is an open node.
- **Willow Overlay:** The hideable and extendable panel that is added to Chrome's UI when Willow is installed. The overlay contains the session graph and is the main medium of interaction with Willow.

6. References

- [1] "Keep tabs on your tabs in Google Chrome " Available:
<https://blog.google/products/chrome/manage-tabs-with-google-chrome/> [Accessed: 20-Nov-2020].
- [2] "What are extensions?," *Chrome*. [Online]. "
Available: <https://developer.chrome.com/extensions>. [Accessed: 20-Nov-2020].
- [3]"Chrome Web Store," *Google*. [Online]. Available:
<https://chrome.google.com/webstore/category/extensions>. [Accessed: 11-Oct-2020].
- [4] "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: 20-Nov-2020].
- [5] "IEEE Governing Documents," IEEE. [Online]. Available:
<https://www.ieee.org/about/corporate/governance/index.html>. [Accessed: 20-Nov-2020].
- [6] 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: 20-Nov-2020].
- [7]"Develop Extensions," *Chrome*. [Online]. Available:
<https://developer.chrome.com/extensions/devguide>. [Accessed: 21-Nov-2020].