Software Requirements Specification

for

Firefox Desktop Browser: Improving Import Usability for Users

Version 0.2 approved

Prepared by Alvin Lew, Angel Villalobos, Ani Movsesian, Brian Pham, Bryan Macoy, Evan Liang, Nolan Ishii, Portia Wu, Steven Wang, Zachariah Harris

Sponsor: Mozilla Foundation

12/1/2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Brian Pham	12/6/22	First Revision upon SRD, major rearrange of the document in terms of requirements.	0.1
Brian Pham	12/7/22	Revisions upon sections 5.1-5.3 (Security Requirements)	0.2
Alvin Lew	12/8/22	Rewrote introduction section, Appendix A, and proofread much of the document	0.2
Portia Wu	12/9/22	Rewrote section 1.3, 1.4, 1.5, and 2.1.	0.3
Entire Team	12/10/22	Final review of first SRS draft	0.4

<Add rows as necessary when the document is revised. This document should be consistently updated and maintained throughout your project. If ANY requirements are changed, added, removed, etc., immediately revise your document.>

1. Introduction

Our project, “Firefox Desktop Browser: Improving Import Usability for Users,” refers to a series of contributions to the open-source Mozilla Firefox browser. We have added a significant amount of features to the importer of the browser, which will make it easier for users to switch to Firefox from another web browser. Specifically, we have developed new UI elements for the importer, enabled importing from more browsers to Firefox, added functionality to import favicons along with bookmarks, and simplified the overall import process to enhance user experience. All of our contributions can be found on Firefox’s public code repository, and our new features will be in the release version of Firefox. Any user with an up-to-date Firefox browser will be able to use and experience our project by simply choosing to “Import Data from Another Browser” within the Firefox browser, and our changes will be noticeable.

1.1 Purpose

The purpose of our project is to improve the existing Firefox importer. Historically, Firefox developers have chosen to prioritize other tasks over fixing bugs in the migrator. Before our project was initiated, all core features of Firefox’s importer were functional, and that’s what mattered to contributors and users. However, this has resulted in minor bugs accumulating and the UI of the migrator feeling outdated. Some of the issues we tackled have been around for up to ten years before we fixed them. Our project involves fixing a collection of many of these importer bugs as well as adding new features. We aim to redesign and improve the overall Firefox importer so that it is more user-friendly.

This document will contain information about the specific requirements for our project and the goals we aimed to achieve. It will list out all of the major changes we have contributed as of Firefox Standard Release 107.0.1.

1.2 Intended Audience and Reading Suggestions

This SRS document is intended for the community of Firefox developers and any others interested in our project, including university professors and developers from the Mozilla Foundation. Since our project is open source, all developers are welcome to reference both this document and the code in the Firefox codebase. Users of our project may only be interested in an overview of this document, whereas developers may find a more detailed read to be beneficial.

1.3 Product Scope

This product helps users transfer the information stored in the old browser to the Firefox desktop. It will consist of several features, including:

- Enabling favicon imports
- Importing profile history, bookmarks, and login information from currently unsupported browsers
- Importing bookmark favicons from all browsers that support normal bookmark imports
- Importing more data from browsers, including credit card information, addresses, and more, depending on features offered by each browser
- Fixing existing bugs in the importer
- Helping implement a redesign of UI in collaboration with Mozilla designers

1.4 Definitions, Acronyms, and Abbreviations

Please reference Appendix A for a full list of definitions, acronyms, and abbreviations.

1.5 References

How to contribute to Mozilla
https://firefox-source-docs.mozilla.org/setup/contributing_code.html

Getting started with Mozilla Open Source development
https://firefox-source-docs.mozilla.org/setup/index.html#getting-set-up-to-work-on-the-firefox-codebase

Understanding Artifact Builds
https://firefox-source-docs.mozilla.org/contributing/build/artifact_builds.html

Mercurial Guide
https://mikeconley.github.io/documents/How_mconley_uses_Mercurial_for_Mozilla_code

Bugzilla
https://bugzilla.mozilla.org/h

2. Overall Description

“Firefox Desktop Browser: Improving Import Usability for Users” is a project that adds additional functionality to the Firefox desktop, mainly on importing data from other browsers to the Firefox browser, and redesigning a modern UI. It allows users to import bookmark favicons, bookmarks, Logins, and addresses from more browsers, including Opera, DuckDuckGo, Opera Gx, Vivaldi, Internet Explorer, Safari, and Chrome. The main goal of the project is to make it easier for users to switch browsers to Firefox.

2.1 System Analysis

1. Project Goal

   The main goal of this project is to make the process of users switching browsers to Firefox more smooth. We roughly divide it into two directions:

   1.1 Improving the Firefox browser's import function. Users can transfer their bookmarks, bookmark favicons, addresses, logins, and history from DuckDuckGo, Opera, OperaGX, and Vivaldi on top of the existing supported browsers

   1.2 Implementing a redesign of the importer UI. Our project will also include the removal of unnecessary warnings that are placed, such as asking users to close the browser that is being imported.

2. Major Technical Hurdles

   The project’s technical hurdles include ensuring that the added functionality does not conflict with the existing software. This is so that our project seamlessly fits within Firefox’s standards and theme

2.2 Product Perspective

This product is part of the Firefox Desktop Browser’s importer. It adds new functionality to the current browser in order to improve the usability of the migration process.

This product shall allow users to import bookmarks, user profiles, browsing history, login information, passwords, addresses, and more. Our product will support browsers that are not currently supported, including Opera, OperaGX, and more in the future. After our project is released, there will be no need for users to manually export HTML file data from these browsers. Instead, Firefox’s new importer will remove unnecessary interactions and speed up the migration process, improving the overall user experience.

2.3 Product Functions

- Migration modal UI improvement: Users will operate the product on a redesigned and modern UI.
- Favicon import: Allow users to import bookmark favicons from various browsers.
- Opera migrator: Allow users to import data from the Opera browser.
- Opera GX migrator: Allow users to import data from the Opera GX browser.
- Vivaldi migrator: Allow users to import data from the Vivaldi browser.
- Reduce User Interactions: No need for users to interact with the migrated browser. The system will operate the migration process in the background.
- No system check needed: Eliminating the need for the system to perform checks on the bookmark toolbar to speed up the migration process.

2.4 User Classes and Characteristics

User classes will include Firefox developers and testers with advanced knowledge of the function who will test and ensure the functionality of the product, and general users of Firefox who may use the importer for their personal data.

2.5 Operating Environment

The software will operate on hardware platforms and operating systems, including Windows 7 and later versions, MacOS 10.12 and later versions, and GNU/Linux. It is supported on all platforms that support the Firefox browser itself.

The product will also operate on the developer (Nightly) and Beta versions of Firefox for testing. All features of this product will be part of the Firefox Standard Release 107.0.1 and will be accessible by users with an updated version of Firefox.

2.6 Design and Implementation Constraints

Firefox Desktop Browser: Improving Import Usability for Users is written in the same languages as Firefox and may include

- Javascript
- C
- CSS
- HTML

For Firefox developers, it may be necessary to use the Nightly full build, which can take an hour or more to build on a computer, even with all CPU focusing on the task.

2.7 User Documentation

Due to the nature of this project, user documentation will not be provided with this software. Instead, users can find updated documentation on the Mozilla website for how to use the import features that our project has implemented.

2.8 Assumptions and Dependencies

2.8.1 Device is assumed to be able to run the latest version of the Firefox Browser.

2.8.2 User is expected to have another browser installed to import data from.

2.9 Apportioning of Requirements

N/A

3. External Interface Requirements

3.1 User Interfaces

The User Interface is the Firefox Browser, which can be downloaded without financial cost via the internet. The interface itself is located on the top toolbar, which will load data into the content section of the browser.

The user can browse the internet via the search field, save and store bookmarks on the toolbar, and configure profiles and settings in the settings section.

3.2 Hardware Interfaces

Users will need a laptop, desktop, smartphone, or tablet with at least 4GB ram to run this software.

3.3 Software Interfaces

Our project will be built on top of the existing Firefox browser.

3.4 Communications Interfaces

Our software does not have any communication interface requirements at the moment.

4. Requirements Specification

4.1 Functional Requirements

4.1.1 The software shall enable the import of favicons from Internet Explorer to Firefox

4.1.2 The software shall enable the import of favicons from Google Chrome to Firefox

4.1.3 The software shall enable the import of favicons from Safari to Firefox

4.1.4 The software shall enable the import of data from Opera to Firefox

4.1.5 The software shall enable the import of data from Opera GX to Firefox

4.1.6 The software shall enable the import of data from Vivaldi to Firefox

4.1.7 The software shall retrieve relevant data from other browsers

4.1.8 The software shall store relevant data into Firefox

4.1.9 The software shall only enable imports for the browsers installed on the user's computer.

4.1.10 The software shall only display browsers that are installed on the user's computer.

4.1.11 The software shall ask the user for the necessary permissions to import the data of Safari on macOS devices.

4.2 External Interface Requirements

4.2.1 Data migrator

Description: The migrator imports data from another browser to Firefox.

Source: Input takes data from another browser, and output stores the data in Firefox.

4.2.1.1 When the user opens the migrator, the user will be able to select a browser to import from

4.2.1.2 After selecting a browser, the user will be able to select what data to import to Firefox

4.2.1.3 The software will then take the data selected and import from the selected browser

i. Bookmarks

ii. Browsing history

iii. Login information

iv. User Profiles

4.2.2 User Interfaces

4.2.2.1 Mouse/keyboard click events will be used to navigate through the migrator

4.3 Logical Database Requirements

4.3.1 Types of information

4.3.1.1 Bookmark Data

Pages that users like to frequent and have saved to the bookmark bar.

4.3.1.2 Favicon Data

Icons that represent the bookmarked link.

4.3.1.3 History Data

Browsing history of the user.

4.3.1.4 Login Data

Saved login information for websites users choose to save the login for.

4.3.1.5 Credit Card Data

Payment information to make checkout easier for the user.

4.4 Design Constraints

4.4.1 Mistaken Browser Data Storage Path

4.4.1.1 If a browser is directed to store its data somewhere other than the default directory, importing data may not work.

4.4.2 Credit card/payment information imports will not work as they are still being worked on.

4.4.3 Unsupported browsers will not be able to import data into Firefox.

4.4.4 Safari will not be able to import data if not given the correct permissions to Firefox.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 The software shall be able to import selected data such as bookmarks, logins, and history from selected browsers in a timely manner commensurate with the amount of data.

5.1.2 The importer shall finish running in under 2 seconds for a majority of users.

5.2 Safety Requirements

5.2.1 Users are expected to import data that is safe and will not harm the user or active device in any way.

5.2.2 Users should adhere to general safety guidelines when it comes to using Firefox and follow those set by Mozilla.

5.2.3 User features shall be treated the same way as currently established parts of Firefox.

5.3 Security Requirements

5.3.1 Our product is designed and built with safety in mind (for example, by eliminating the imports of cookies). However, users should refrain from using it in conjunction with malicious data or code, as it could compromise the security of the user.

5.3.2 Users should be aware that there may be existing bugs and uncaught security vulnerabilities within Firefox. By using our product, users must remember that there is always the risk of an uncaught bug out of our control.

5.3.3 Users should report any bugs to Bugzilla as soon as possible, to ensure a patch is quickly made to prevent any more bugs from affecting user experience on Firefox.

5.4 Software Quality Attributes

5.4.1 Provide availability of migrating user data from other browsers to Firefox at any time.

5.4.2 Allow user flexibility to decide which of their data to import.

5.4.3 Our project shall be up to par with standards set by QA teams at Mozilla

5.4.4 The UI of our project shall feel modern and match with the current Firefox themes

5.5 Business Rules

5.5.1 Conform to those set by Mozilla.

5.5.2 Our product will be open source and thereby be available to any interested parties.

6. Legal and Ethical Considerations

6.1 Product Description

Firefox will continue to function as intended to allow any user to access the internet.

6.2 Image and logo Trademarks

All images and logos used for branding shall be researched and deemed legally available for use and subsequently trademarked for Mozilla's purposes.

6.3 Proprietary code

All of our source code, regardless of language, shall remain open source and available for anyone to view in order to help foster the growth of Firefox.

6.4 Storing User data

All imported data as a result of features in our project shall remain in the correct user's local storage and not on any exterior server without the permission of the user.

6.5 Managing User Data

Mozilla will not sell, trade, or share any of the user's data without the user's permission. Our project does not modify the terms and conditions set by Mozilla Firefox.

Appendix A: Glossary

Bugzilla - A bug tracking system developed and used by Mozilla

Element - An open source messenger based on the Matrix protocol

Mercurial - A source control manager (SCM) used to manage and track changes across projects

Firefox - The Mozilla Firefox open-source web browser with over 200 million users

Favicon - The icon corresponding to a webpage, typically next to a bookmark or history entry

Mozilla Foundation - the group responsible for developing the Firefox browser

Build - compiled version of a program (Firefox has a full build and a faster artifact build)

Nightly - an experimental version of the Firefox browser

Beta - a version of software that is made available for testing

Phabricator - a website where Firefox developers can review and approve submitted code

Bugzilla - Issue tracker for Firefox

UI - User interface

UX - User experience

Favicon - The icon corresponding to a webpage, typically next to a bookmark or history entry

SQLite - An SQL database engine that stores data as a file on disk

Importer - The part of a browser which imports data from other browsers, including passwords, bookmarks, history, and more

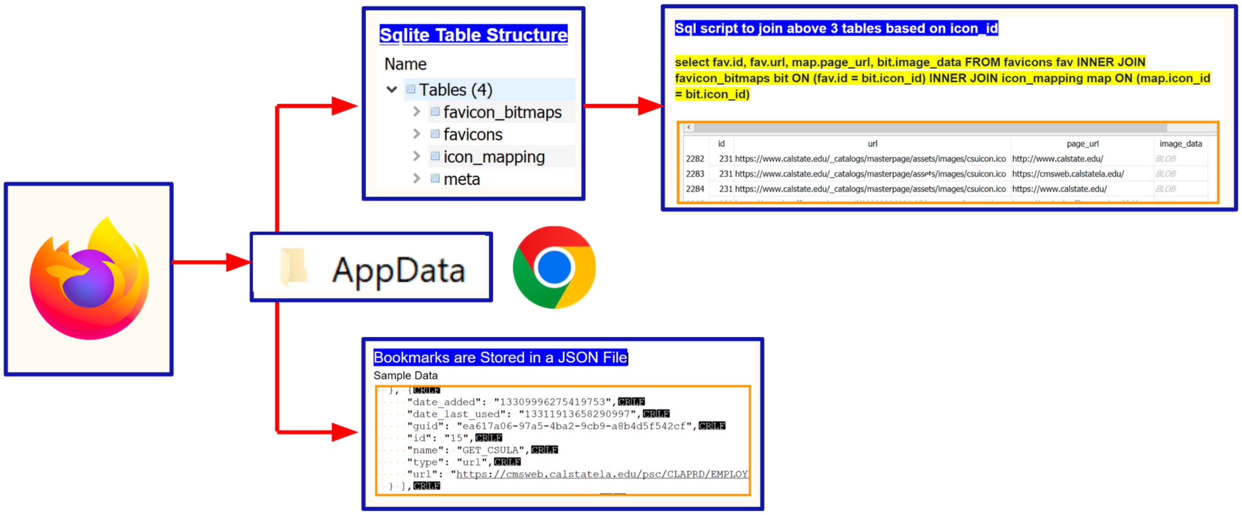
Matrix - A lightweight protocol for decentralized, real-time communication, and the officially preferred communication platform for the public Mozilla community.

Migrator - In our document, it is a synonym for importer

SRS - Software Requirements Specification

Appendix B: Analysis Models

Favicon Storage in Chrome



Sql script to join above 3 tables based on icon_id

```
select fav.id, fav.url, map.page_url, bit.image_data FROM favicons fav INNER JOIN
favicon_bitmaps bit ON (fav.id = bit.icon_id) INNER JOIN icon_mapping map ON (map.icon_id
= bit.icon_id)
```

	id	url	page_url	image_data
2282	231	https://www.calstate.edu/_catalogs/masterpage/assets/images/csuicon.ico	http://www.calstate.edu/	BLOB
2283	231	https://www.calstate.edu/_catalogs/masterpage/assets/images/csuicon.ico	https://cmsweb.calstatela.edu/	BLOB
2284	231	https://www.calstate.edu/_catalogs/masterpage/assets/images/csuicon.ico	https://www.calstate.edu/	BLOB

Bookmarks are Stored in a JSON File

Sample Data

```
  }, {CRLF
     "date_added": "13309996275419753",CRLF
     "date_last_used": "13311913658290997",CRLF
     "guid": "ea617a06-97a5-4ba2-9cb9-a8b4d5f542cf",CRLF
     "id": "15",CRLF
     "name": "GET_CSULA",CRLF
     "type": "url",CRLF
     "url": "https://cmsweb.calstatela.edu/psc/CLAPRD/EMPLOY
  } ],CRLF
```

Favicon Storage in Safari

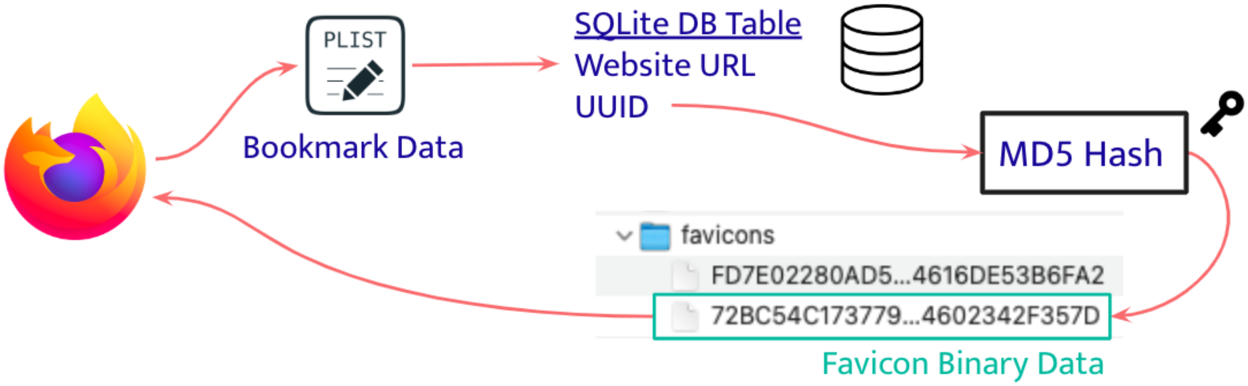
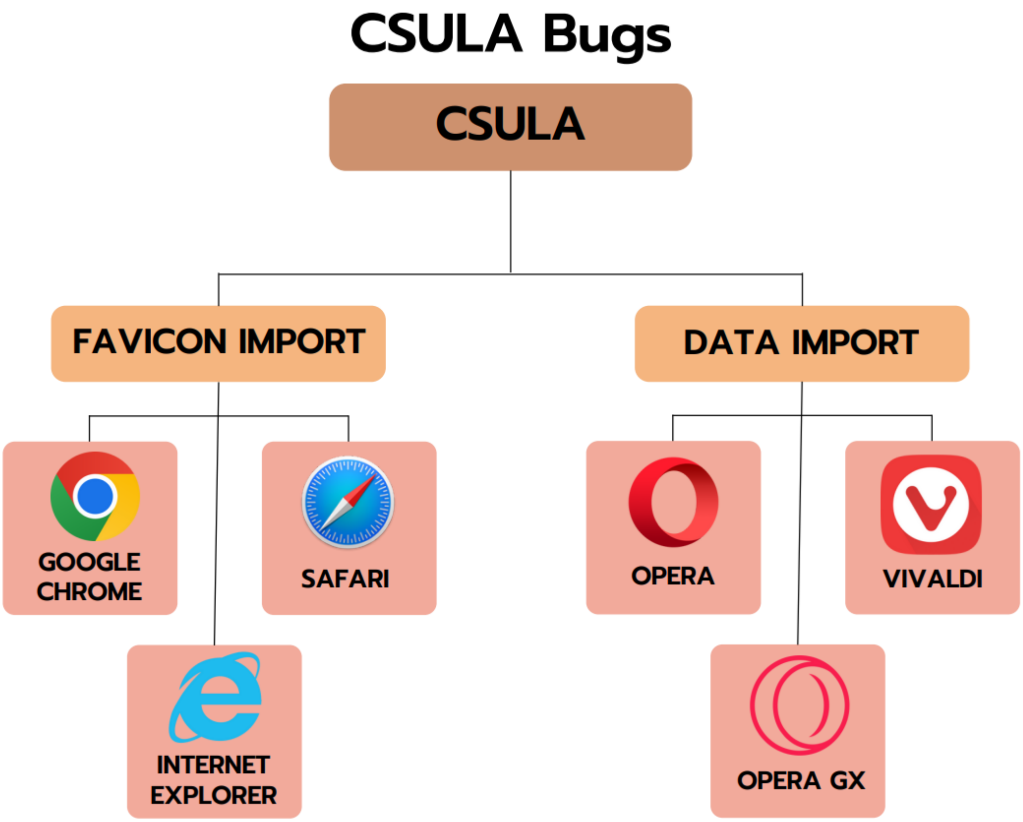


Diagram of the main features we have implemented



Appendix C: To Be Determined List

N/A