**International Institute Of Professional Studies**

**Devi Ahilya Vishwa Vidhayalaya**

****

**Project Report**

**On**

**Computerization**

**Of**

**This Project is submitted for V-semester**

For degree of

**Master in Information Technology**

Guided By:- Submitted By:-

Mr. Shaligram Prajapat Karishma Yadav

IT-2K19-23

**BONAFIDE CERTIFICATE**

**This is to certify that the project report entitled "Minor project on Open Source(Mozilla firefox)" submitted to Devi Ahilya University in partial fulfillment of the requirement for the award of the degree of MASTER IN COMPUTER INFORMATION TECHNOLOGY (M.tech), is an authentic and original work carried out by Ms. Karishma Yadav (IT-2k19-23) and under my guidance.**

**The matter embodied in this project is genuine work done by the student and has not been submitted whether to this University or to any other UNIVERSITY/ Institute for the fulfillment of the requirements of any course of study.**

**Internal Examiner: External Examiner:**

**RECOMMENDATION**

**The Project wok entitled " Minor project on Open Source(Mozilla firefox)" submitted by Karishma yadav is satisfactory account of the bona fide work under my supervision and is recommended towards the end of their III year of M.Tech 2021.**

**Guided By: Mr. Shaligram Prajapat**

**ACKNOWLEDGEMENT**

**We acknowledge our sincere thanks to those who have contributed significantly to this project. It is a pleasure to extend deep gratitude to our internal guide Mr. Shaligram Prajapat, IIPS, for his valuable guidance and support and to continuously prompt us for the progress of the project. We thank him for his valuable suggestions towards our project, which helped us in making this project more efficient and user friendly.**

**We thank and acknowledge each and every ones efforts that helped us in some or the other way for small and significant things.**

**ABSTRACT**

**The open-source software in discussion is Mozilla Firefox, or simply known as just Firefox. Two key aspects of this software shall be discussed, namely the key characteristics and its architecture. We will first discuss about the key characteristics that make or break Mozilla Firefox as a web browser. We shall also briefly touch on what sets Mozilla Firefox apart from proprietary web browsers and its position its today's climate. Next, we shall delve into the conceptual architecture of the software and understand the ins and outs of the software. We will discuss about components and its role in the software and how the software was created.**

TABLE OF CONTENT

1. **INTRODUCTION**
   1. Background
   2. Objectives
   3. Purpose
   4. Scope
   5. Application
   6. Achievements
   7. Organisation of report
2. **SURVEY OF RELATED TECHNOlOGIES**
3. **REQUIREMENTS AND ANALYSIS**
   1. Problem Defination
   2. Requirements specification
   3. Planning and Scheduling
   4. Software and hardware requirements
   5. Preliminary Product Description
   6. Conceptual Models
4. **SYSTEM DESIGN**
   1. Basic Modules
   2. SDD
   3. Schema Design
   4. STP
   5. SPMP
   6. Logic diagram
   7. Data structure
   8. Algorithms Designs
   9. User interface designs
   10. Security Issues
   11. SDLC
5. **IMPLEMENTATION & TESTING**
   1. Implementation Approaches
   2. Coding Details and Code Efficiency
   3. Testing Approach
   4. Modification and Improvements
6. **CONCLUSION**
7. **REFERENCES**

INTRODUCTION

* 1. **Background:-**

Mozilla Firefox is nothing but a Web Bowser, withwhich one can access the internet. The web browser lets one access information in form of text, audio, images, and videos from all around the world. Mozilla Firefox was developed by Mozilla Foundation in 2002 under the Phoenix community. Nowadays, it is called Firefox only as it is derived from Mozilla Web Browser it is also known as Mozilla Firefox.

* 1. **Objectives:-**

Mozilla Firefox is considered a browser that assists users with an intelligent address bar anda simplified user interface. Tabbed Browsing allows you to open a lot of websites in multiple tabs at the same time, between which you can switch back and forth easily.

* 1. **Purpose:-**

Firefox is a web browser- a platform, which is used to browse the information via the internet. When we search for something, a web browser tries to fetch the data from various internet-connected servers. Data fetching is done by rendering engine which is the software itself and translates the data for user interpretation. The data is interpreted in HTML that is Hypertext Markup Language. Browser further reads this HTML code to the user, the way we see it. It could be text, image, audio, or video. All this information has a unique URL or web address that can directly be accessed using browser like Mozilla Firefox. There are a number of other extensions are available nowadays like css, RSS, XHTML, mng, etc and almost all of them are supported by Firefox.

* 1. **Scope:-**

Mozilla Firefox is known for its speed. Though the Firefox browser needs a lot of memory for operating efficiently. It may limit the multiple tasking of computers. However, It provides better network security. It has advanced security options that protect your system from spyware and malwares. It has strong popup broker and authentication protocols which makes it safe from potential attackers using any unauthorized codes. Further to enhance security users can use enhanced security options like NoScript and Flashblock. It enables user to execute advanced code so that certain new features which can make the browser more Intutive.

* 1. **Application:-**

Firefox has an interface which is very user friendly and the user can use a number of add-ons on top of that user can customize the browsing also. It has more than 6000 extensions, user can customize the browser with more than 500 themes. Mozilla offers Tabbed Browsing which can let the user open unlimited tabs in a single window. It also has got embedded memory which makes it capable of remembering pages, in case if your systems is turned off by mistake, you can recall all the open pages.

* 1. **Achievements:-**

The open, global internet is the most powerful communication and collaboration resource we have ever seen. It embodies some of our deepest hopes for human progress. It enables new opportunities for learning, building a sense of shared humanity, and solving the pressing problems facing people everywhere.Over the last decade we have seen this promise fulfilled in many ways. We have also seen the power of the internet used to magnify divisiveness, incite violence, promote hatred, and intentionally manipulate fact and reality. We have learned that we should more explicitly set out our aspirations for the human experience of the internet. We do so now.

* 1. **Organisation of report:-**

The Mozilla Public License 2.0 provides a number of benefits for users of the licensed code.For example, the express grant of patent rights protects users against certain patent lawsuits from contributors. An additional benefit is the ability to combine MPL’d code with proprietary works. The license encourages companies to be part of and give back to the OSS community without completely sacrificing the competitive advantages that come with closed-source software.

****

SURVEY OF RELATED TECHNOlOGIES

Web browsers provide the primary interface by which we access the Internet, and the choice of which browser to use can be very personal and can significantly change your online experience for better or worse. Historically, screen reader users have been slower to adopt new web browsers. The reasons for this are numerous, but can include:

* Early versions of newer web browsers generally provide less support for accessibility than their more developed alternatives.
* Assistive technology vendors have not yet updated their applications to be compatible with new browsers.
* Information and resources on using new browsers with assistive technologies may be limited or unavailable
* The learning curve associated with the move to a new browser can be steep, especially when factoring in assistive technology like a screen reader or voice recognition software.

The impact of these barriers to the adoption of modern web browsers is evident when you examine web browser usage among screen reader users. [Web Accessibility in Mind (webAIM),](https://webaim.org/) a non-profit based out of Utah State University’s Center for Persons with Disabilities, conducts a bi-annual survey of screen reader users and their use of assistive technologies, web browsers, and the Internet. When WebAIM first conducted this survey in 2008, they found that screen reader users used browsers at the following rates:

The most recent WebAIM Screen Reader survey, which was conducted in October 2017, found the following:

* Firefox: 41.0%
* Internet Explorer: 23.3%
* Chrome: 15.5%
* Safari: 10.5%
* Internet Explorer 6, 7, or 8: 4.1%
* Internet Explorer 9 or 10: 4.0%
* Microsoft Edge: 0.5%

REQUIREMENTS AND ANALYSIS

* 1. **Problem Defination:-**

**There are several compatibility issues.**  
Firefox doesn’t work well on all platforms. This is especially true for mobile platforms. Not every website is compatible with this browser either, which means some sites may not show properly on the screen if they even show at all. Because this problem is hit or miss, it may not be a disadvantage for some users at all. Others may find this problem to be a deal-breaker.

**It consumes a lot of a computer’s memory.**  
Firefox consumes an extensive amount of available memory in order to operate. This makes it difficult for many low-level computers to run anything but basic sites because of this requirement. Videos are slow to stream if they ever do and the tabs are prone to freezing or failing and requiring a reboot. This is especially true if numerous plugins must be used simultaneously.

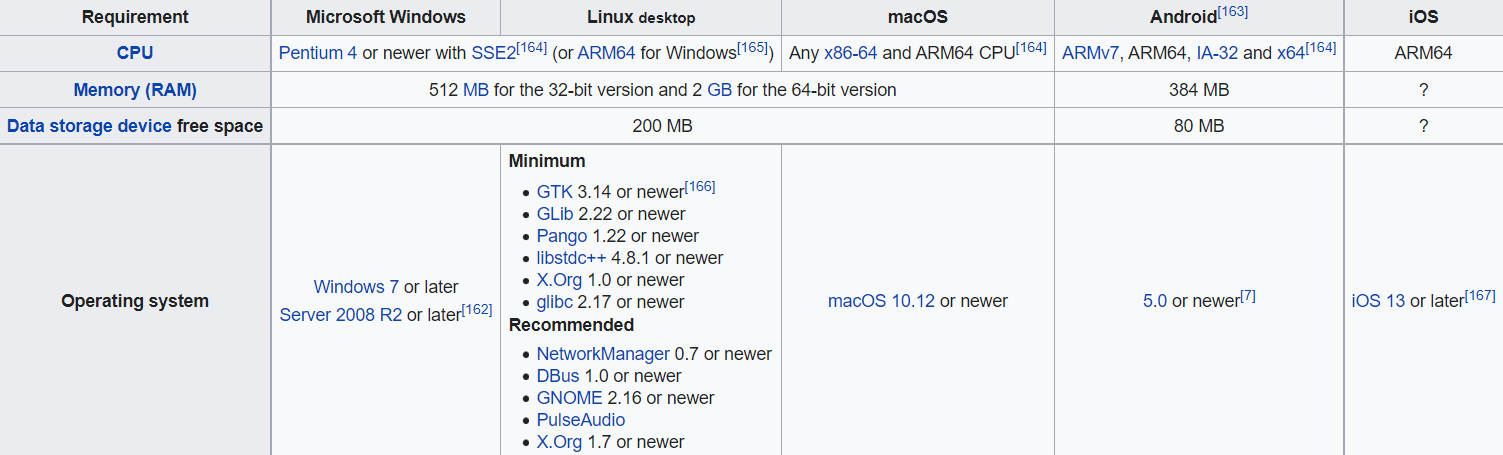
**It does not automatically resume downloads.**  
If you are downloading a file and browsing at the same time on Firefox, then there is a good chance that the browser will freeze. When this occurs, the downloaded file will stop downloading instead of pausing the download like other browsers allow. This means that the downloaded file has to start over once again. For large file downloads, this can become a tremendous headache. There are no automatic updates for plugins that are being used either, so everything must be checked manually on a regular basis to make sure the browser is performing at the highest possible level.

**It struggles with HTML 5 quite a bit.**  
The amount of trouble that Firefox has in reading HTML5 is incredible. Some websites will fail to load on this browser or fail to load certain components of the site based solely on its coding. Add-on compatibility issues are also frequent and this leads to delays on page loading and encourages added memory consumption. This issue is found on multiple computer brands and configurations, so delays in browsing tend to occur instead of increased speed as in previous versions.

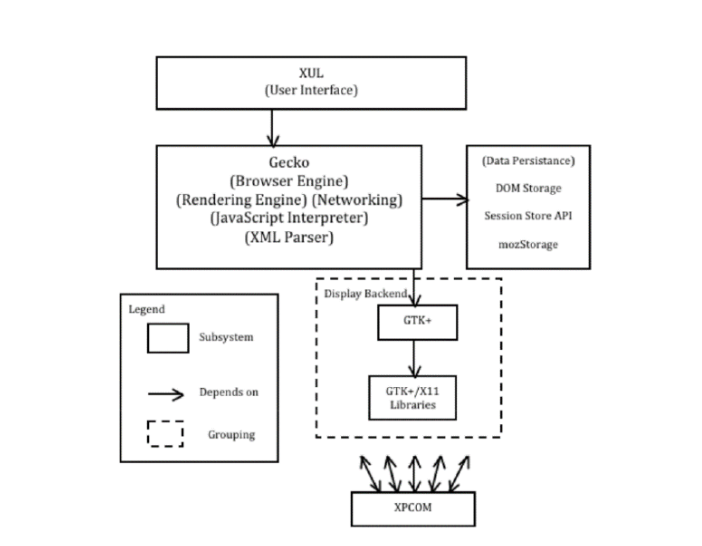
* 1. **Requirements specification:-**

Mozilla has finally announced the minimum requirement for mobile phones that will use a proprietary mobile operating system, [Firefox OS](https://www.gizmochina.com/?s=Firefox+OS). And, from a list of the specifications mentioned by Mozilla, appears low end mobile phone of Firefox is not too different from other platforms.To run[Firefox OS](https://www.gizmochina.com/?s=Firefox+OS) operating system, Mozilla set some standards, such as minimum processor speed of 800 MHz, 256MB RAM and QVGA screen with 320 x 240 pixels resolution. Compared with other platforms, the minimum requirement is similar to Firefox OS minimum standards for Windows Phone device.Some Firefox-based mobile OS that has emerged into the public also seems to use the minimum specifications set by Mozilla. The mobile-phone include ZTE and Alcatel One Touch Open Fire. In addition, Sony Mobile is also testing the Firefox OS on Sony Xperia E which also has a specification is not much different.

* 1. **Software and hardware requirements:-**

****

* 1. **Conceptual Models:-**

****

The Layered Architecture of Firefox plainly shows the conditions and dependencies of

modules within Firefox . The top layer comprises of the UI, which relies on the remainder of

the framework to render the program. This layer fills in as the fundamental setting through

which the client communicates with the Firefox framework. The UI calls upon the subsequent

layer, which administration the UI layer with rendering and information get to. This layer

contains both the Gecko and Necko subsystems, which are likewise actualized as a Virtual

Machine layered structural style with funnel and channel components. At last, the most

reduced layer is the Display Backend. The Display Backend gives proper GUI formatting

info to Gecko with the goal that pages can be correctly rendered on any stage as required.

The idea of a layered engineering benefits the framework by permitting it good portability as

the last layer gives platform‐specific rendering to the framework. The framework is likewise

modifiable since there are a wide range of components and each plays out a particular task.

• XUL - XUL is basically XML (extensible markup language), taking into account

HTML components, including JavaScript, to be incorporated to Firefox and has a

particular definition for a few element types. This provides the basis for the user

interface of Firefox. Mozilla uses this to build cross‐platform applications such as

web browsers and mail clients. XULRunner is the runtime envirenment utilized for

the sending of XUL applications. The user interface layer needs Gecko for its

rendering and parsing capabilities and for Necko, the networking component.

• Gecko – The Gecko subsystem serves more than just a search engine and rendering

engine. Its components now include the JavaScript Interpreter, SpiderMonkey, and

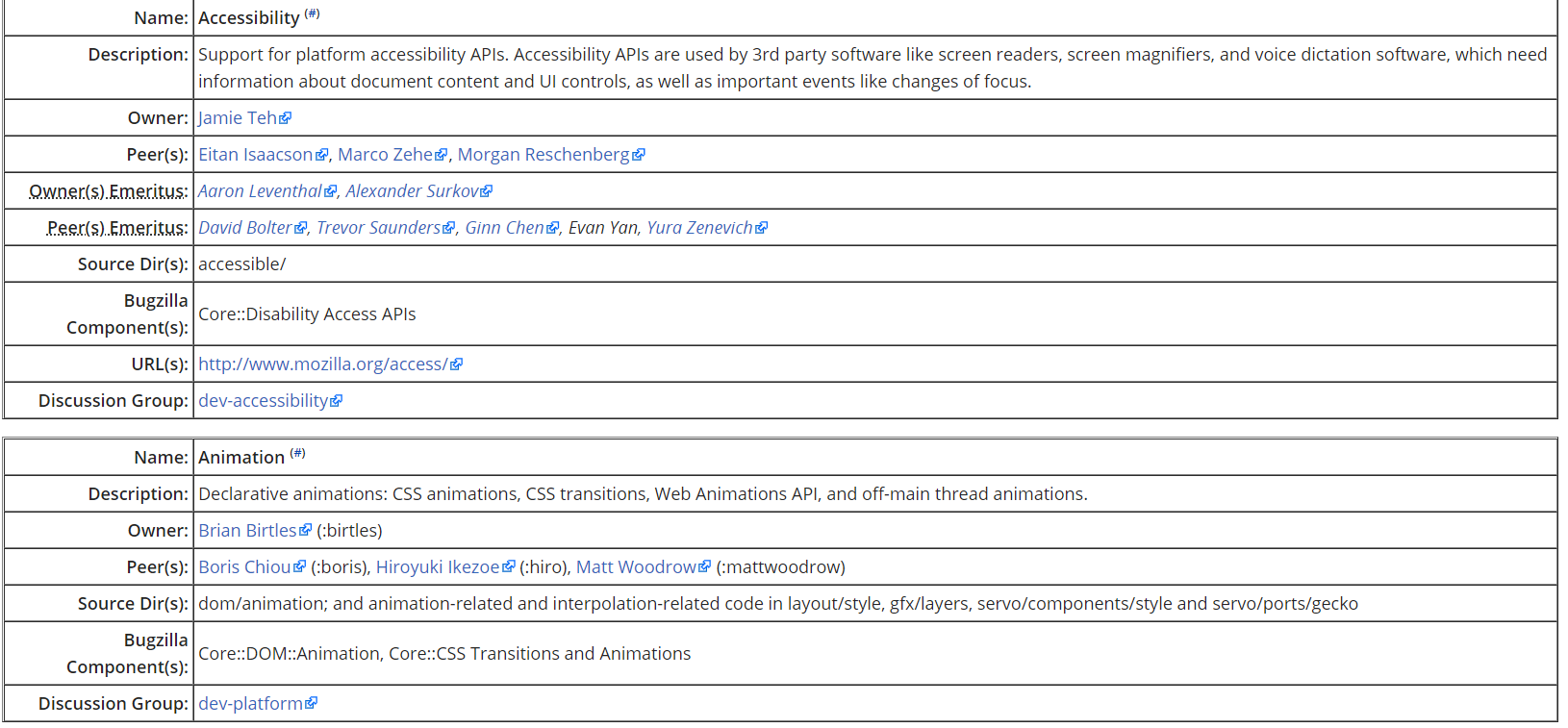
the Necko. It is highly standard compliant applying various standards and is widely

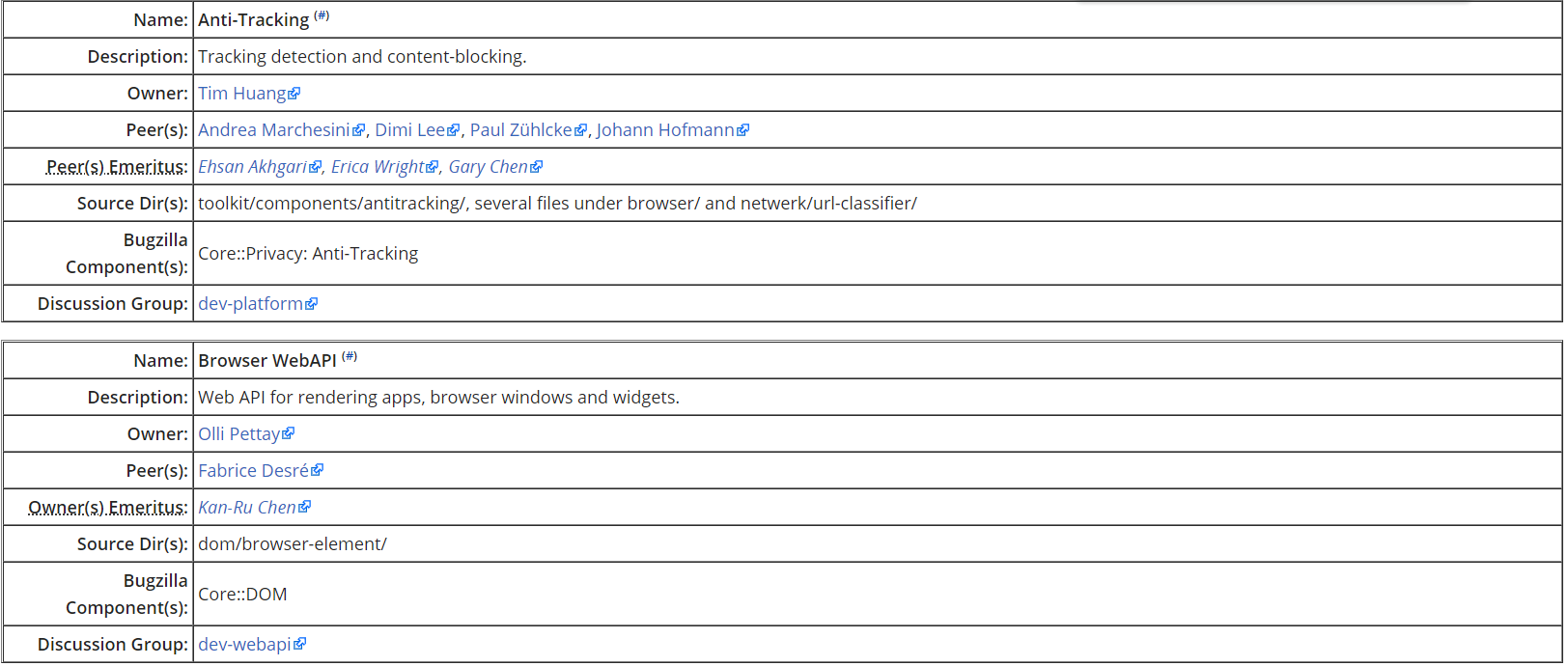
used in Mozilla products due to its capabilities and strengths in fast and effective

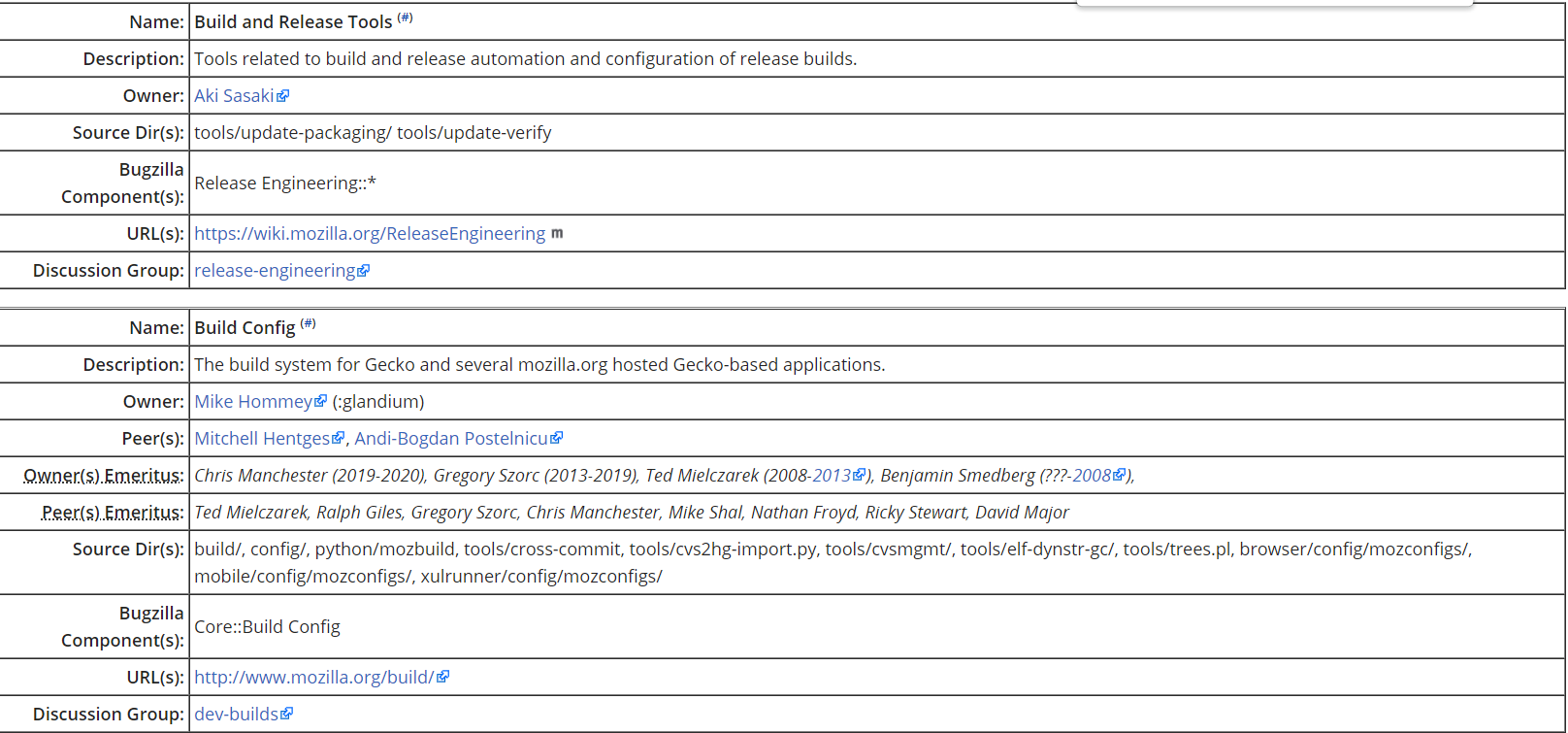
rendering.

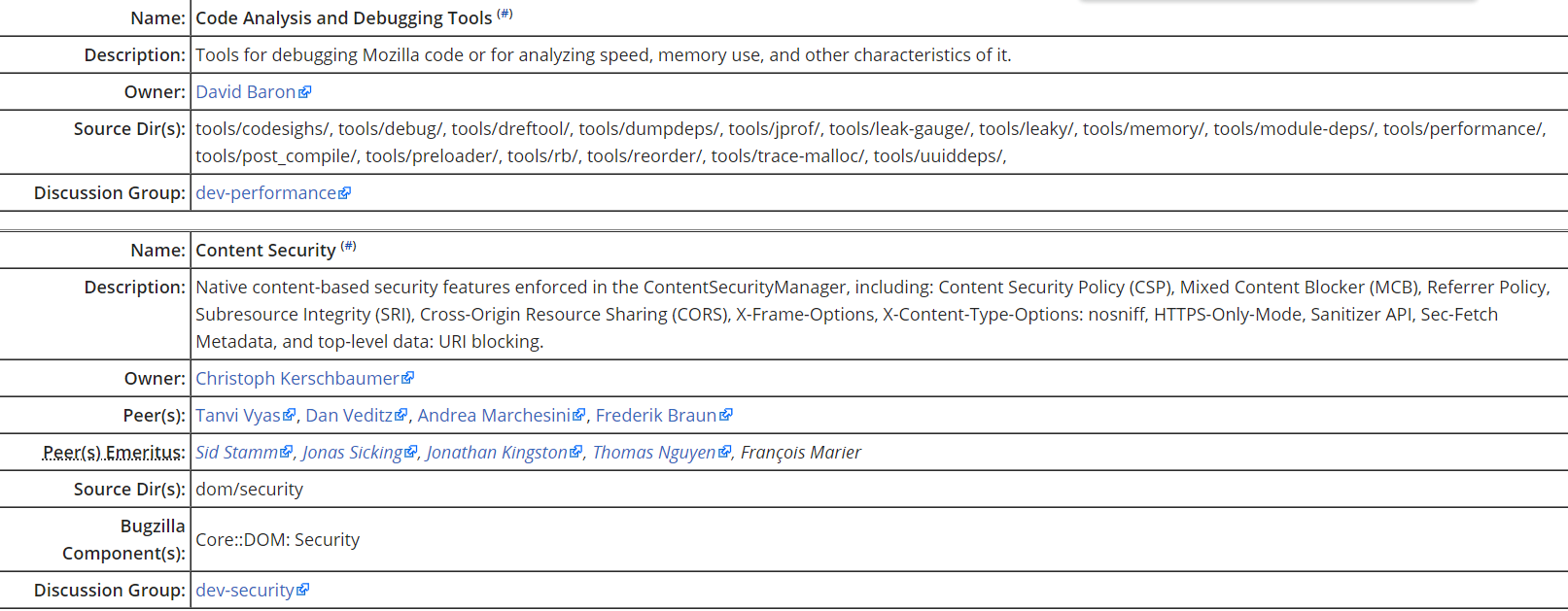
SYSTEM DESIGN

**4.1 Basic Modules:-**

****

****

****

****

****

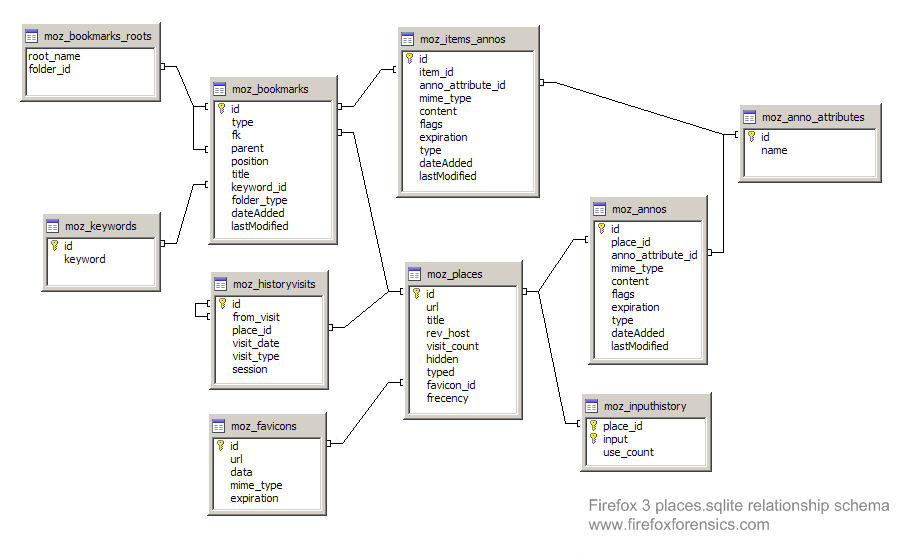
**4.2 SDD:-**

The SDD usually contains the following information:

1. The [*data design*](https://en.wikipedia.org/wiki/Data-driven_design) describes structures that reside within the software. Attributes and relationships between [data objects](https://en.wikipedia.org/wiki/Data_object) dictate the choice of [data structures](https://en.wikipedia.org/wiki/Data_structures).
2. The [*architecture design*](https://en.wikipedia.org/wiki/Software_architecture) uses information flowing characteristics, and maps them into the program structure. The transformation mapping method is applied to exhibit distinct boundaries between incoming and outgoing data. The data flow diagrams allocate control input, processing and output along three separate modules.
3. The [*interface design*](https://en.wikipedia.org/wiki/Interface_(computing)) describes internal and external program interfaces, as well as the design of the [human interface](https://en.wikipedia.org/wiki/User_interface_design). Internal and external interface designs are based on the information obtained from the analysis model.
4. The [*procedural design*](https://en.wikipedia.org/wiki/Procedural_design) describes structured programming concepts using graphical, tabular and textual notations.

These design mediums enable the designer to represent procedural detail, that facilitates translation to code. This blueprint for implementation forms the basis for all subsequent software engineering work

**4.3 schema design:-**



**4.4 STP:-**

|  |  |
| --- | --- |
| **Segment** | Computer users using the internet |
| **Target Group** | Computer users assessing internet for various purposes |
| **Positioning** | Positioned as one of the most efficient and speedy web browser |

**4.5 SPMP:-**

Software project manager prepare a document on the basis of decision finalized during the project planning. This **document**is known as **Software Project Management Plan Document** or **SPMP document**.

**SPMP document** is a well organized document that contains the project planning in detail.

 It would have details about project objective, project estimates, project schedules, project resources, project staffing, risk management plans, project monitoring, project control and other miscellenous activities.

A SPMP document is prepared and organized in structure as shown below:

**Introduction**

* Objectives
* Functions
* Performance issues
* Constraints

**Project estimates**

* Historical data used
* Estimation techniques details
* Cost, duration, effort estimates

**Project Schedule**

* Work breakdown
* Gantt and PERT chart

**Project resource**

* Manpower
* Hardware and Software
* Highly skilled professionals

**Staff organization**

* Team formation and structure
* Management reporting

**Risk Management**

* Risk analysis
* Risk identification
* Risk abatement methods

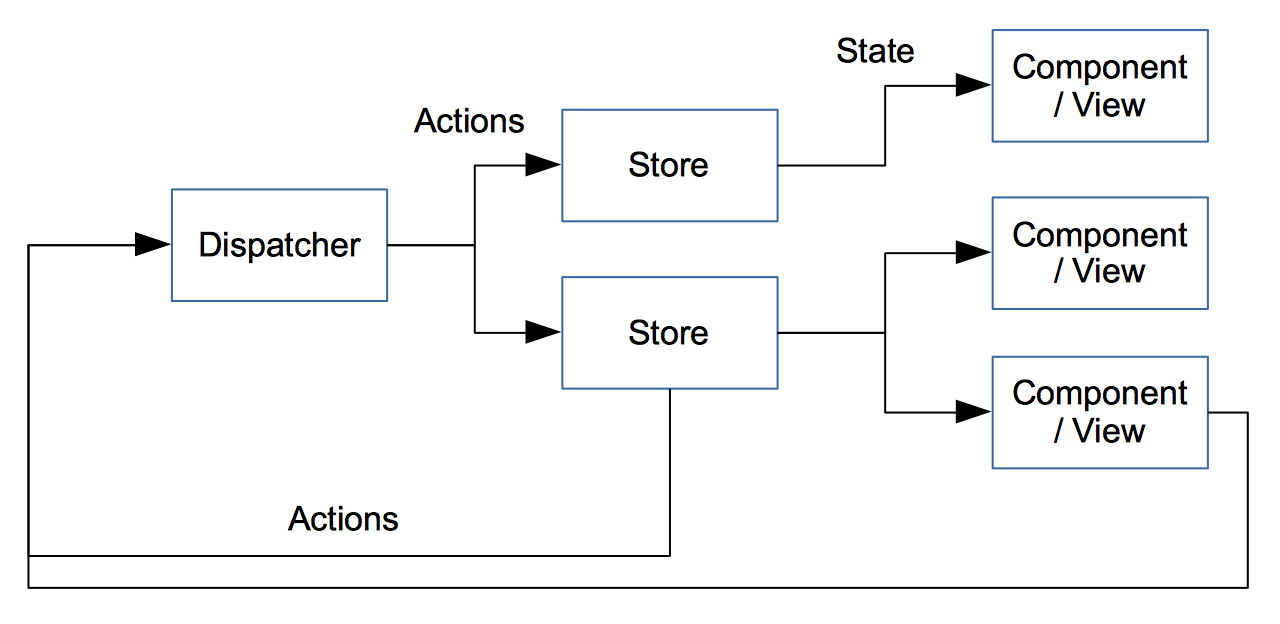
**Project tracking**

**Project Control**

**Miscellenous activities**

All the above activities are documented in **SPMP document** by project manager.

**4.6 Logic diagrm :-**



The main parts of a flux system are stores, components and actions. Some of this is a bit like an MVC system, but I find there’s better definition about what does what.  
An action is effectively a result of an event, that changes the system. For example, in Loop, we use actions for user events, but we also use them for any data incoming from the server.

A store contains the business logic. It listens to actions, when it receives one, it does something based on the action and updates its state appropriately.

A component is a view. The view has a set of properties (passed in values) and/or state (the state is obtained from the store’s state). For a given set of properties and state, you always get the same layout. The components listen for updates to the state in the stores and update appropriately.

We also have a dispatcher. The dispatcher dispatches actions to interested stores. Only one action can be processed at any one time. If a new action comes in, then the dispatcher queues it.

Actions are always synchronous – if changes would happen due to external stimuli then these will be new actions. For example, this prevents actions from blocking other actions whilst waiting for a response from the server.

**4.7 Data structure:-**

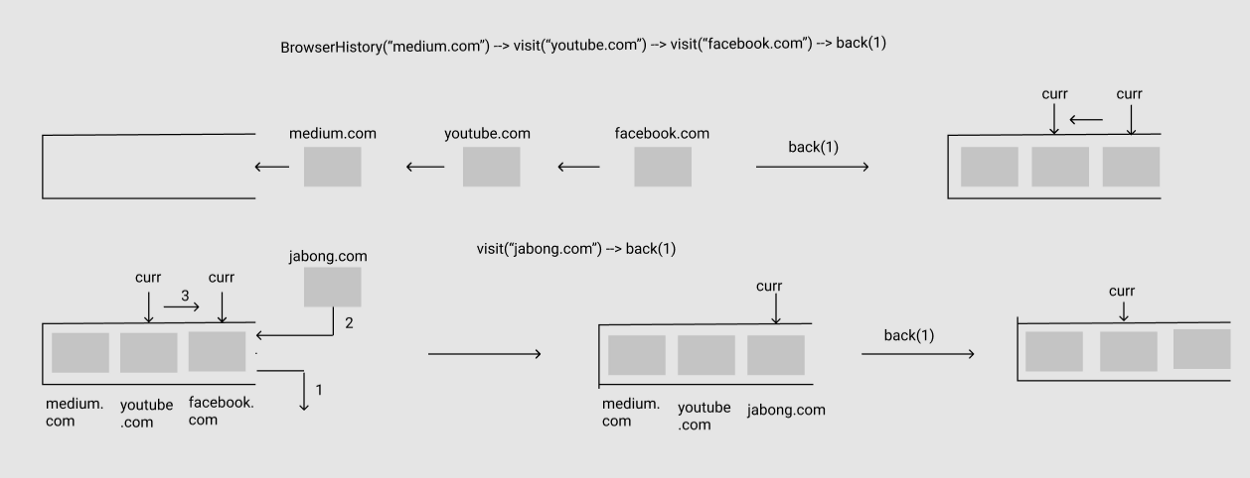
These data structures are:

* *Frame trees*, which are composed of local and remote nodes that represent which web documents are in which render process and which Blink renderer.
* The *immutable fragment tree*, represents the output of (and input to) the layout constraint algorithm.
* *Property trees*, which represent the transform, clip, effect, and scroll hierarchies of a web document, and are used throughout the pipeline.
* *Display lists and paint chunks* are the inputs to the raster and layerization algorithms.
* *Compositor frames* encapsulate surfaces, render surfaces, and GPU texture tiles that are used to draw using the GPU.

**4.8 algorithm design:-**

**BrowserHistory**(“*medium.com*”) → **visit**(“*youtube.com*”) → **visit**(“*facebook.com*”) → **back**(*1*) → **visit**(“*jabong.com*”) → **back**(*1*)

Diagrammatic explanation



The other words, an algorithm is a means of describing a way to solve a problem so that it can be solved repeatedly, by humans or machines. Computer scientists compare the efficiency of algorithms through the concept of "Algorithmic Complexity" or "Big O" notation.

For example:

* A cooking recipe is a simple algorithm for humans.
* A sorting algorithm is often used in computer programming to explain a machine how to sort data.

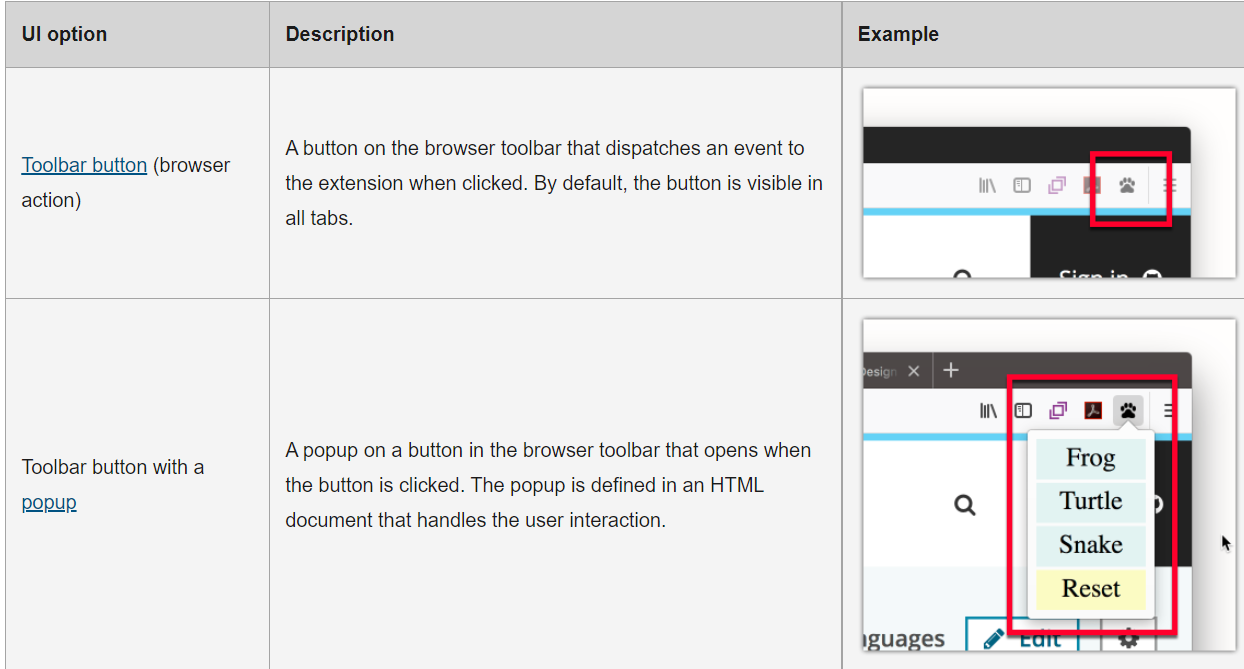
Common algorithms are Pathfinding algorithms such as the Traveling Salesman Problem, Tree Traversal algorithms and so on.

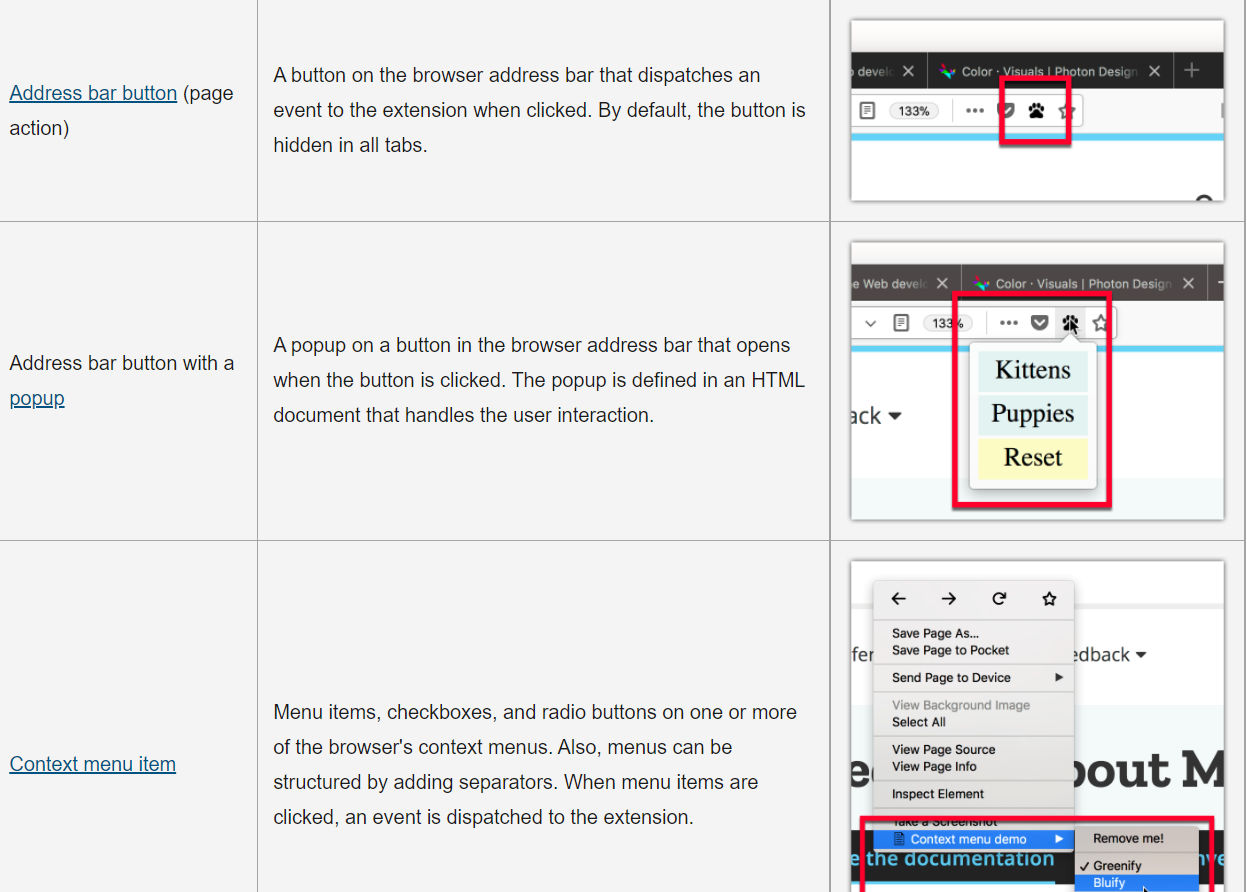
There are also Machine Learning algorithms such as Linear Regression, Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, Recurrent Neural Network (RNN), Long Short Term Memory (LSTM) Neural Network, Convolutional Neural Network (CNN), Deep Convolutional Neural Network and so on.

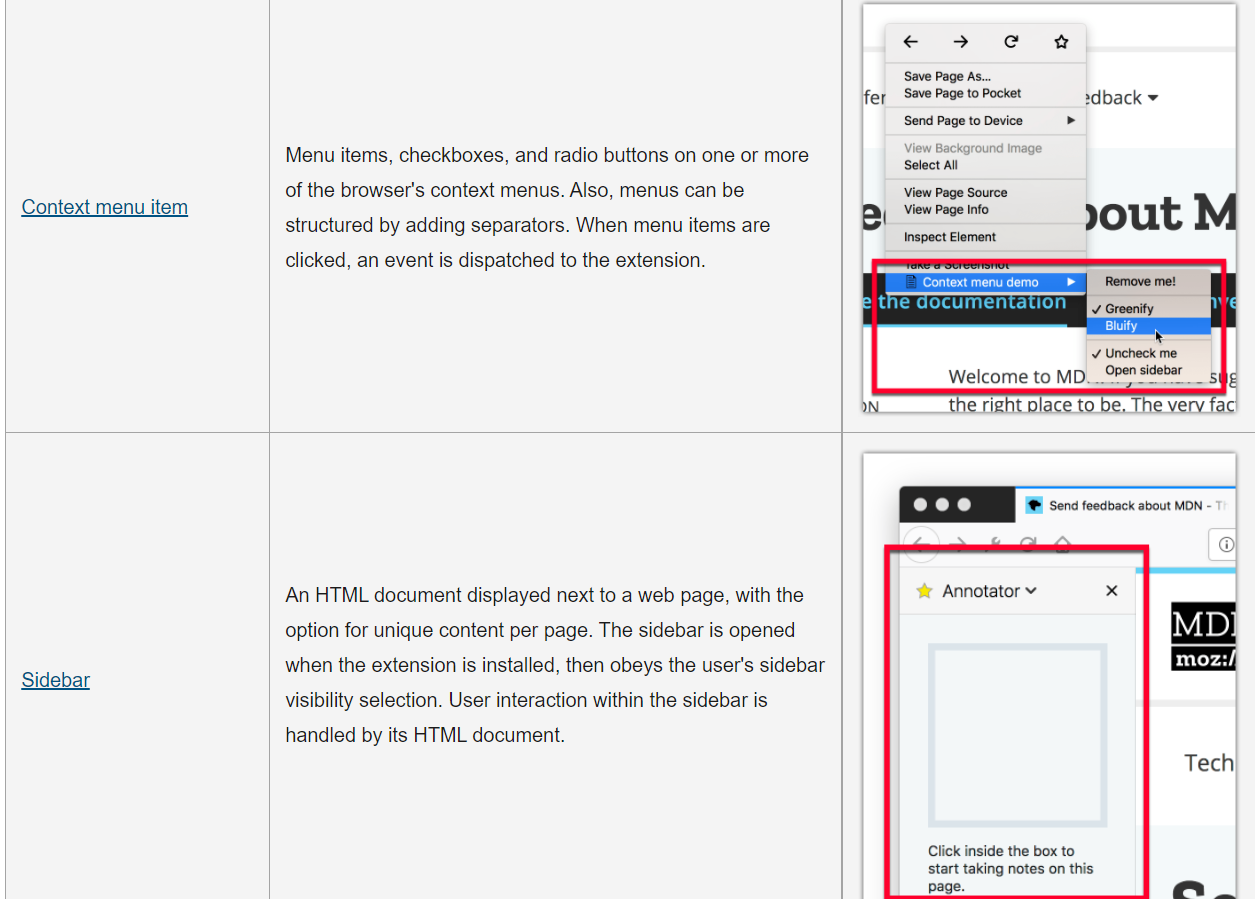
**4.9 User interface design:-**

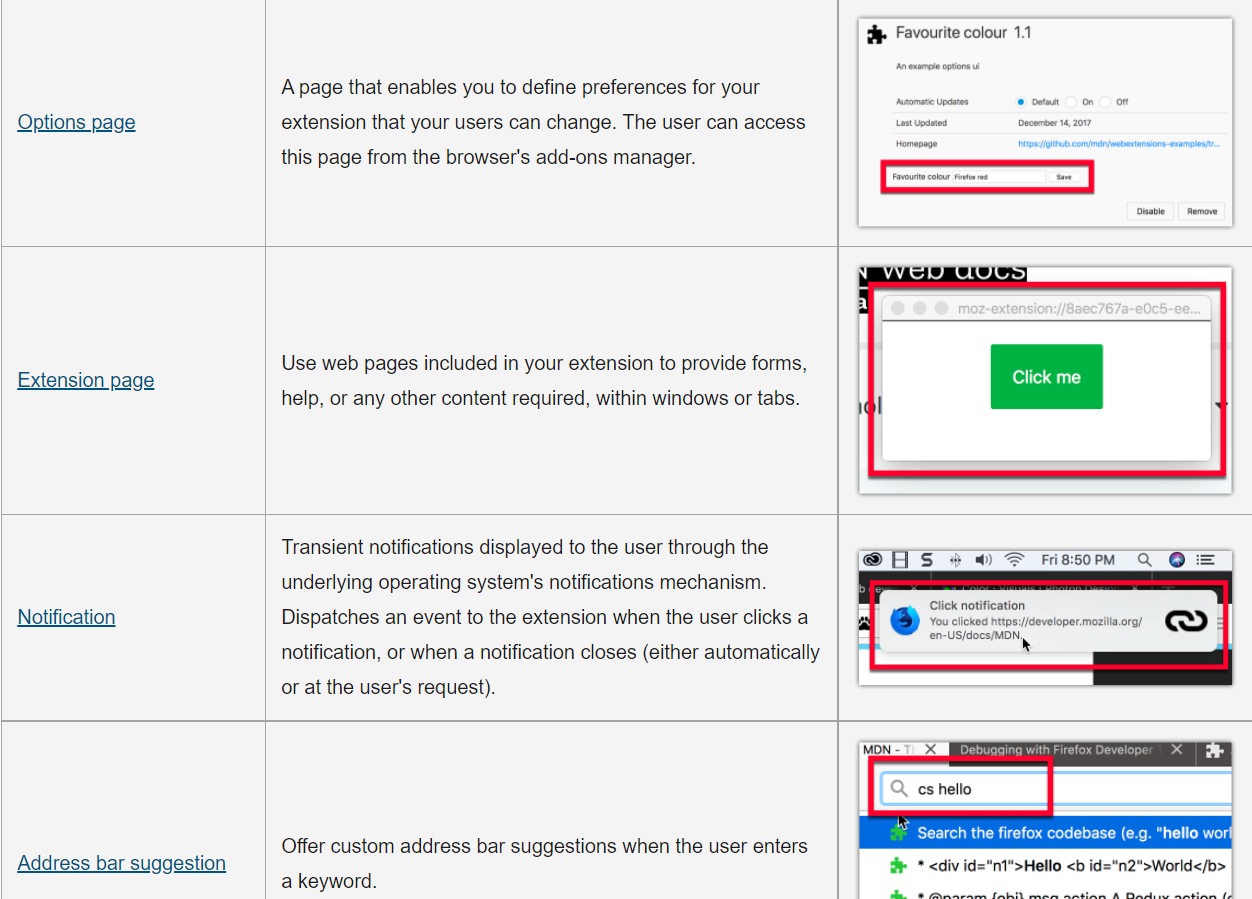
# User interface

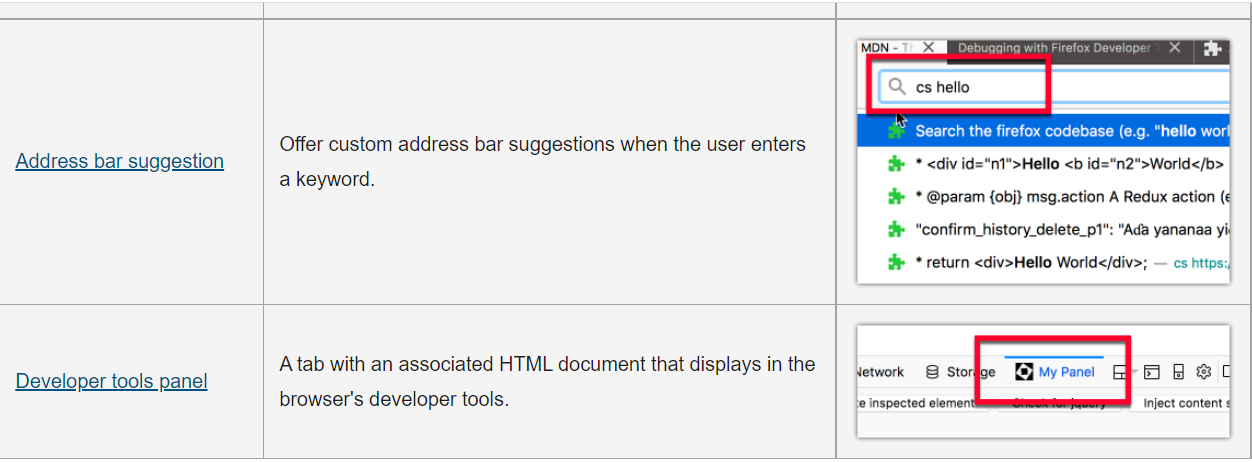
Extensions that use WebExtension APIs are provided with several user interface options so that their functionality can be made available to the user. A summary of those options is provided below, with a more detailed introduction to each user interface option in this











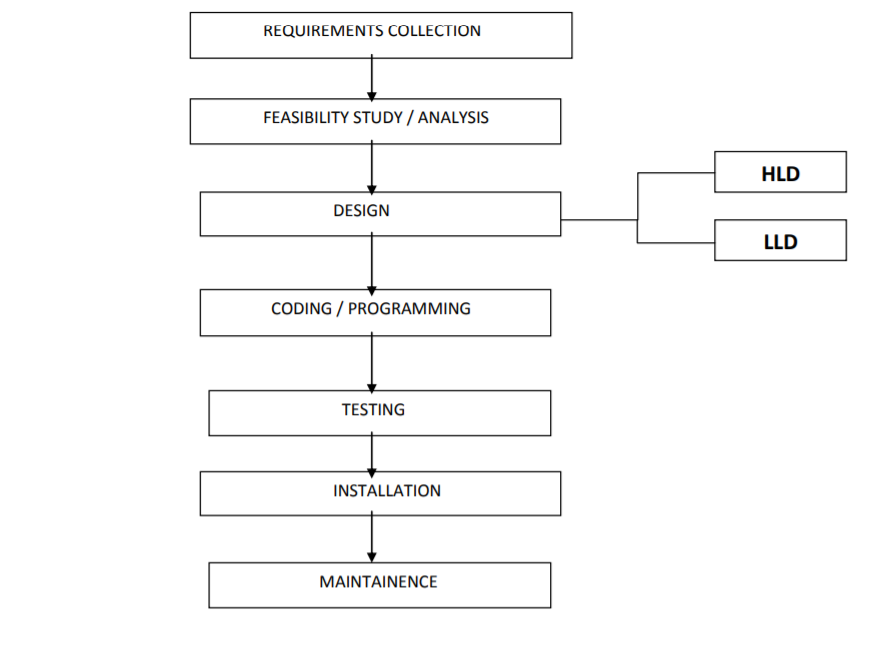
**4.10 Security issues:-**

* **Critical**Vulnerability can be used to run attacker code and install software, requiring no user interaction beyond normal browsing.
* **High**Vulnerability can be used to gather sensitive data from sites in other windows or inject data or code into those sites, requiring no more than normal browsing actions.
* **Moderate**Vulnerabilities that would otherwise be High or Critical except they only work in uncommon non-default configurations or require the user to perform complicated and/or unlikely steps.
* **Low**Minor security vulnerabilities such as Denial of Service attacks, minor data leaks, or spoofs. (Undetectable spoofs of SSL indicia would have "High" impact because those are generally used to steal sensitive data intended for other sites.)

**4.11 SDLC:-**

It is a procedure to develop the software. It is a process of creating or altering systems and the models and methodologies that people use to develop these systems. Any SDLC should result in a high quality system that meets or exceeds customer expectations, reaches completion within time and cost estimates, works effectively and efficiently and is inexpensive to maintain and cost effective to enhance. Different procedures / models are available to develop a software namely

* 1. Waterfall model It is a traditional model It is a sequential design process, often used in SDLC, in which the progress is seen as flowing steadily downwards (like a waterfall), through the different phas es as shown in the figure,



* 1. Requirements Collection :- - done by Business Analysts and Product Analysts - gathering requirements - translates business language into software language For ex, let us consider the example of a banking software
  2. . Feasibility Study :- - done by software team consisting of project managers, business analysts, architects, finance, HR, developers but not testers - architect – is the person who tells whether the product can be developed and if yes, then which technology is best suited to develop it. - here we check for, - technical feasibility - financial feasibility - resource feasibility
  3. Design :- There are 2 stages in design, HLD – High Level Design LLD – Low Level Design HLD – gives the architecture of the software product to be developed and is done by architects and senior developers LLD – done by senior developers. It describes how each and every feature in the product should work and how every component should work. Here, only the design will be there and not the code. For ex, let us consider the example of building a house.
  4. Coding / Programming :- - done by all developers – seniors, juniors, freshers - this is the process where we start building the software and start writing the code for the product.
  5. Testing :- - done by test engineers - it is the process of checking for all defects and rectifying it.
  6. Installation :- - done by installation engineers - to install the product at a client‟s place for using after the software has been developed and tested. For ex, consider the example of a software to be developed and installed at Reliance petrol bunk.
  7. Maintenance:- - here as the customer uses the product, he finds certain bugs and defects and sends the product back for error correction and bug fixing. - bug fixing takes place - minor changes like adding, deleting or modifying any small feature in the software product

IMPLEMENTATION & TESTING

**5.1 Implementation Approaches:-**

## Keep it simple

## Validate your code

## Avoid browser quirks modes

## Use CSS reset rules

## Develop in Firefox

## Test in as many browsers as possible

## Fix IE issues by using conditional comments

## Make IE6 work with transparent PNGs

## Provide fallbacks

## Code by hand

**5.2 coding details and code efficiency:-**

Code efficiency is a broad term used to depict the reliability, speed and programming methodology used in developing codes for an application. Code efficiency is directly linked with algorithmic efficiency and the speed of runtime execution for software. It is the key element in ensuring high performance. The goal of code efficiency is to reduce resource consumption and completion time as much as possible with minimum risk to the business or operating environment. The software product quality can be accessed and evaluated with the help of the efficiency of the code used.

Code efficiency plays a significant role in applications in a high-execution-speed environment where performance and scalability are paramount.

One of the recommended best practices in coding is to ensure good code efficiency. Well-developed programming codes should be able to handle complex algorithms.

Recommendations for code efficiency include:

* To remove unnecessary code or code that goes to redundant processing
* To make use of optimal memory and nonvolatile storage
* To ensure the best speed or run time for completing the algorithm
* To make use of reusable components wherever possible
* To make use of error and exception handling at all layers of software, such as the user interface, logic and data flow
* To create programming code that ensures data integrity and consistency
* To develop programming code that's compliant with the design logic and flow
* To make use of coding practices applicable to the related software
* To optimize the use of data access and data management practices
* To use the best keywords, data types and variables, and other available programming concepts to implement the related algorithm.

**5.3 Testing Approaches:-**

A test approach is the test strategy implementation of a project, defines how testing would be carried out. Test approach has two techniques:

* **Proactive -**An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.
* **Reactive -**An approach in which the testing is not started until after design and coding are completed.

## Different Test approaches:

There are many strategies that a project can adopt depending on the context and some of them are:

* Dynamic and heuristic approaches
* Consultative approaches
* Model-based approach that uses statistical information about failure rates.
* Approaches based on risk-based testing where the entire development takes place based on the risk
* Methodical approach, which is based on failures.
* Standard-compliant approach specified by industry-specific standards.

## Factors to be considered:

* Risks of product or risk of failure or the environment and the company.
* Expertise and experience of the people in the proposed tools and techniques.
* Regulatory and legal aspects, such as external and internal regulations of the development process.
* The nature of the product and the domain

**5.4 modification and improvements:-**

Add-on  allow you to add extra features and functionality to Firefox, modify the Firefox user interface and change its appearance. There are several types of add-ons but extensions are the most common. Anyone can create an extension and make it available for download.In the past, extensions often stopped working each time a new version of Firefox was released, because developers had to update them every six weeks to keep them compatible. Since extensions could also modify Firefox internal code directly, it was possible for bad actors to include malicious code in an innocent-looking extension.To address these issues, and as part of broader efforts to [modernize Firefox](https://blog.nightly.mozilla.org/) as a whole, we’ve transitioned to a [new framework](https://developer.mozilla.org/Add-ons/WebExtensions) for developing Firefox extensions. Extensions created with the new standard are safer, more secure, and won’t break in new Firefox releases. You can still personalize Firefox with extensions the same way you do now, except they won’t break in new Firefox releases.

CONCLUSION

Firefox was once the top one web browser for users during the late 00's. It replaced

Internet Explorer and now, it has officially replaced by Google Chrome. One main reason

why people opt for Google Chrome over Firefox is because Google is the top 1 search

engine in this era, not to mention that Google has provided a plethora of useful apps

which ease people's lives. Hence, Google Chrome serves as a platform to access to all

these apps thus it gains its popularity. Despite the impressive Firefox has to offer, it is still

unable to clinch top spot in browsers.

REFERENCES

(https://opensource.com/resources/what-open-source)

[2] Mozila.org, Firefox Features

( https://www.mozilla.org/en-US/firefox/features/memory/)

[3] James Brereton, Gordon Krull, Rob Staalduin, Katie Tanner (October, 2011)Conceptual

Architecture of Mozilla Firefox 6

[4] Dr. Ahmed E. Hassan (July 2007) , Concrete Architecture of Mozilla Firefox

[5] N Safie (2008). Usage of OSS in multimedia courses: OUM experience.

FORMAMENTE, No 3-4, Anno 3. 249-260.

[6] The Linux Information Project (July 2005), Proprietary Software Definition

[7] support mozilla.org, Find and install add-ons to add features to Firefox

(https://support.mozilla.org/en-US/kb/find-and-install-add-ons-add-features-to-firefox)