1. **Vorarbeit**
   1. **Recherche zu Browser Extensions**
      1. **Extension-Programmierung allgemein**
      2. **Existieren bereits vergleichbare Extension**
      3. **Vergleich führender Browser als Plattform für die Extension**
   2. **Projekt PGuard**
      1. **Vorstellung**
      2. **API-Anbindung für die Extension**
   3. **Implementierung einer Google Chrome Extension**
      1. **Eigenschaften**
      2. **Funktionsumfang**
      3. **Darstellung im Browser**
   4. **Caching-Methoden**
      1. **Welche Rolle spielt die Performance?**
      2. **Verwendete Methoden und deren Eigenschaften**

**Vorarbeit:**

*The following chapter provides the interested reader with the basic information that is needed to understand this work. All topics are described in no more detail than needed to follow this work. If the reader is interested more in a specific topic necessary references are provided at the respective places to the respective textbooks and articles.*

*This chapter starts with a general view on ... . After that, the ... is explained. ... as an important method is then described in detail. It follows an overview over state-of-the art algorithms that will be studied in this work. This chapter ends with a section about how the performance of algorithms could be quantified.*

*Also state in this chapter related work and the state-of the art.*

Recherche zu Extensions.

Allgemein:

*Was ist eine Extension? Begriffsklärung? Welche Browser? Grober Aufbau, verwendete Sprache, etc.*

Unter einer Extension versteht man ein Programm, welches den Browser um neue Funktionen ergänzt. Durch eigene Oberflächen oder Manipulation der Website erleichtern diese Erweiterungen das Nutzen des Browser.

Im Gegensatz zu Plug-Ins haben Extensions Zugriff auf Browser-spezielle Funktionen und sind in der Lage über die Webseite hinaus zu agieren. Plug-Ins werden direkt in eine Webseite eingebettet und sind auf diese beschränkt. Der Oberbegriff „Add-on“ wird heutzutage Hauptsächlich als Synonym für Extension verwendet.

Jeder größere Browser stellt eine Plattform zur Verfügung auf denen Extensions angeboten und installiert werden können. In der Regel sind Diese kostenlos. Können auch von außerhalb installiert werden zu Entwicklungszwecken oder wenn nicht auf der Plattform angeboten.

Extensions werden in HTML, JavaScript und CSS implementiert. Dabei können alle Bibliotheken verwendet werden, welche den Browserstandards für Extensions entsprechen. Kapitel 2.3.2 befasst sich genauer damit, welche Bedingungen für diese Bibliotheken in Google Chrome gelten.

Bekannte Beispiele sind Werbeblock wie UBlock Origin und VPN-Anwendungen wie Hola.

Vergleichbare Extensions:

Datum 13.3, Dauer ca. 5h

Gesucht wurde nach einer Extension die auf der Play Store Seite den Nutzer datenschutzrelevante Informationen zu den angebotenen Apps liefert, Eine Datenschutzwertung im Playstore vergibt oder den Nutzer Apps nach Berechtigungen die Apps vorschlägt.

Extensions werden nach ihrer Kurzbeschreibung in den Suchergebnissen überprüft und bei nicht eindeutig Aufgabenbeschreibung die Infoseite aufgerufen(Bsp. Safe.ad im Web Store „ecosystem“).Nur deutsche und englische Ergebnisse werden berücksichtigt.

Die Recherche hat ergeben, dass unter den genannten Suchkriterien keine Chrome oder Firefox Extension gefunden wurde die Aufgabenbereich der geplanten Extension abdeckt. Einige aufgeführte Beispiele implementieren einen Teil der geplanten Funktion (Umsortierung, Tracker checken) , aber keine Extension erfüllt alle gewünschten Aufgaben.

Platform: Web Store von Google Chrome:

Suchwörter: “save”,”safety”, “play store”, “privacy”,“check“, “berechtigung”, “permission” und deren Kombinationen ergaben keine Ergebnisse.

Plattform: Github

Extensions im topic Chrome-Extensions:  
<https://github.com/topics/chrome-extension>

„play store“, „app“, „permission“, „safety“ , „save” keine Ergebnisse.  
„privacy“: 1 Ergebnis:  
 <https://github.com/FacettsOpen/exodify>  
 Prüft App auf Tracker

In der Gesamtsuche mit Schlagwort „chrome extension“ + „play store“,: keine Ergebnisse

**„store“ 3 Ergebnisse:  
(prüft auf mangelhafte apps/extensions)**[**https://github.com/uku/malicious-chrome-extensions**](https://github.com/uku/malicious-chrome-extensions) **(crawlt extensions aus dem web store)**[**https://github.com/logicalhacking/ExtensionCrawler**](https://github.com/logicalhacking/ExtensionCrawler) **(sortiert den Store neu)https://github.com/mathewilson/chromext-play-store-sorter**

**Plattform: Firefox addons  
  
 gleiche Suchbegrife, keine Ergebnisse**

Suchablauf, kriterien „vergleichbar“, Ergebnisse Abtrennung VORHER PGUARD?!

Vergleich führender Browser:

Vorauswahl durch Marktanteile. Eigenschaften einzelner Browser. Ausschlusskriterien. Entscheidung der Auswahl. Polyfill erwähnen

Projekt PGuard:

Vorstellung:

Was ist das für ein Projekt. Leitung/Teams. Ziele, Umsetzungen. Ergebnisse?

API-Anbindung für die Extension:

API erklären. Welche Informationen. Was sind Infoboxes Auswahl. Realisierung.

Implementierung einer Google Chrome Extension:

Begriffsklärung: Popup, page\_action/browser\_action

Policies erklären

Eigenschaften:

Aufbau der Extension:

Die Architektur einer Chrome Extension stellt ein Paket aus mehreren Dateien dar und ist vergleichbar mit anderen Web-Technologien wie zum Beispiel Webseiten.

Grundvoraussetzung für eine funktionierende Extension ist die manifest.json, welche grundlegende Informationen für den Browser bereitstellt und festlegt mit welchen Dateien und Rechten die Extension aufgebaut ist. Hinzu kommt mindestens eine HTML-Datei zur Darstellung der Inhalte und mindestens ein Skript zur Umsetzung der Funktionalität. Erweitert werden diese oft durch CSS-Dateien.  
Externe Bibliotheken wie JQuery können ebenfalls eingebunden werden, müssen aber aufgrund der Policies von Google Chrome vollumfänglich lokal vorliegen. Mehr dazu Im nächsten Abschnitt.

Die Manifest-Datei ist im JSON-Format aufgebaut und beinhaltet sämtliche Informationen über die Extension. Wichtige Punkte sind Name der Extension, Beschreibung, Rechte und Aufbau. Unter Rechten oder „permissions“ werden alle APIs aufgelistet, welche die Extension benötigt um ordnungsgemäß zu funktionieren. Bevor ein Nutzer später die Extension installiert, muss er diesen „permissions“ zustimmen. Mehr zu den APIs im entsprechenden Abschnitt.  
Der Aufbau wird unter „content\_scripts“ (BILD?) in drei Eigenschaften unterteilt: unter welchem URL sind die Skripte aktiv, welche Skripte sind dort aktiv und welche CSS-Dateien werden dort von der Extension eingesetzt.

HTML-Dateien werden als „User-Interface Elemente“ zusammengefasst und beinhalten im Normalfall eine popup.html zur Darstellung des Fensters der Extension in der oberen rechten Ecke des Browser-Fensters (BILD?). Je nach Funktionsumfang können weitere UI-Elemente eingebunden sein, um zum Beispiel die besuchte Webseite zu erweitern.  
  
Die vorhandenen Skripte werden normalerweise in zwei Kategorien eingeteilt. Das sogenannte „Background-Skript“ dient als Event-Handler und kommuniziert zwischen Extension und Browser. Alle restlichen Skripte sind „Content-Skripte“. Sie beinhalten die eigentliche Funktionalität der Extension.

Quelle: https://developer.chrome.com/extensions/overview#pages

Policies

APIs

Policies von Google Chrome, Developing. Einsatz.

Funktionsumfang:  
 Anforderungen an die Extension. Browserspezifische Limitierungen?

Darstellung:

??

Caching-Methoden:

Welche Rolle spielt die Performance?

Ladezeiten. Nutzerfreundlichkeit. Entlastung der Server

Verwendete Methoden:

Auflistung Browser und ihre Methoden/APIs.