IpShare

Studienrichtung
Technische Informatik

Hochschule
Augsburg University of
Applied Sciences

Fakultät für
Informatik

Jan Luis Holtmann

Betreuer: Erich Seifert

Abgabedatum: 02.02.2024

Hochschule für angewandte
Wissenschaften Augsburg

An der Hochschule 1
D-86161 Augsburg

Telefon +49 821 55 86-0
Fax +49 821 55 86-3222
www.hs-augsburg.de
info(at)hs-augsburg-de

Fakultät für Informatik
Telefon +49 821 55 86-3450
Fax +49 821 55 86-3499

An der Hochschule 1
86150 Augsburg
Telefon +49 821 55 86-3450
max@hs-augsburg.de

Inhaltsverzeichnis

Kapitel 1

Motivation und Anforderungen

Musstest du schon einmal die Ip-Adresse von einem anderen Gerät per Hand abtippen? Wahrscheinlich nicht und wenn hat es bestimmt auch nicht weh getan. Aber währe es nicht viel besser stattdessen an beiden Geräten den Browser zu öffen und auf eine Webseite zu gehen, welche die Ip-Adresse von einem Gerät zum anderen überträgt. "Nein, es dauert genauso lang" höre ich dich sagen. Wahrscheinlich hast du recht. Na ja, ich habe trotzdem einige Monate an einer Webseite entwickelt, die im Prinzip das macht. Frei nach dem Motto automatisiere 3 Monate, eine Aufgabe die 3 Sekunden dauert. Bei IpShare handelt es sich also im Prinzip um ein overengineertes Clipboard.

1.1 Projektidee

Spaß beiseite, tatsächlich habe ich allerdings einen Anwendungsfall, indem ich so etwas brauche. An unserer Hochschule bekommt man im Eduroam-WLAN eine öffentliche Ip-Adresse. Für Experimente ist das als Student natürlich ein Fest. Jedenfalls wollte ich mich von einem Raspberry Pi aus einem 5G Netz zu meinem Laptop verbinden. Dabei handelt es sich nebenbei um das Maveric-Projekt. Das funktioniert auch da ich ja eine öffentliche Ip bekomme. Nur ändert sich diese Ip gelegentlich. Leider kann ich keine Angaben dazu machen, wann oder wie oft das Vorkommt. Es wäre nicht schlecht, automatisiert die neue Ip herauszubekommen, da ich nur umständlich Zugriff auf den Raspberry Pi habe. Während ich das hier schreibe, ist das noch nicht erprobt worden, sollte aber dank Api funktionieren. Diese Problematik gab mir jedenfalls die Idee für das Projekt.

1.2 Ziel

Ob mein Projekt große Verwendung für andere hat bleibt abzuwarten. Im Wesentlichen ging es darum mich mit Webentwicklung und im speziellen Flask zu beschäftigen. Etwas zu experimentieren und dabei zu lernen. Dementsprechend sind manche Designentscheidungen auch getroffen, einfach um zu experimentieren. Manche Dinge sind daher auch nicht ganz einheitlich implementiert.

1.3 bestehende Lösungen

Es gibt bereits Webseiten, welche einem die eigene öffentliche Ip Adresse verraten. Diese währen z.B. whatismyipaddress.com[4] oder whatismyip[2]. Sie erlauben es aber nicht, die Adresse von einem anderen Gerät zu bekommen.

1.4 Anforderungen

Auf die Details möchte ich im Kapitel zu Planung und Entwurf 2 eingehen. Im groben sollen Ip-Adressen von einem Gerät geteilt werden können um sie dann von einem andern Gerät abrufen zu können. Als Rahmenbedingung für das Projekt war noch gegeben das es sich um eine multiuserfähige Webanwendung mit Datenbank handeln soll. Die Entwicklung sollte nach einem Agilen-Vorgehensmodell erfolgen.

Kapitel 2

Planung und Entwurf

2.1 Organisation

Da ich nicht im Team arbeite, ist die Organisation trivial. Um einen groben und flexiblen Plan zu haben orientierte ich mich an Scrum und verwendete einen Backlog mit User Storys. Eine unterteilung in Product Backlog und Sprint Backlog war hier nicht angebracht. Die Einträge (User Storys) im Backlog sind nicht technisch, sondern fachlich. Sie folgen einem einheitlichen Schema:

Als “Rolle” möchte ich “Ziel/Wunsch” (, um “Nutzen”).

2.1.1 User Rollen

Ich verwende drei einfache Rollen, um Personen zu klassifizieren, die mit der Anwendung interagieren:

- Visitor - jemand der die Seite nur besucht.
- User - jemand der einen Account hat.
- Admin - jemand der die Seite verwaltet.

2.1.2 User Storys

☑ Als Visitor möchte ich meine Ip-Adresse teilen, um sie auf einem andern Gerät abzurufen.

☑ Als Visitor möchte ich über die Risiken des Teilens, informiert werden.

☐ Als Visitor möchte ich über das Anlegen eines Users informiert werden.

☑ Als Visitor möchte ich einen Accout Registirern, um User zu werden.

⏩ Als Visitor möchte ich die App ohne Cookies nutzen können.

☑ Als Admin möchte ich eine requirements.txt.

⏩ Als User möchte ich eine Liste aller meiner IPs sehen und diese Sortieren können.

☑ Als User möchte ich meine Ip Passwort geschützt teilen.

⏩ Als User möchte ich sehen wann die Ip geteilt wurde.

☐ Als User möchte ich leicht herausbekommen wie ich die API verwenden kann. (Immer hinweise wie ich die Informationen mit der API abrufen könnte?!)

☑ Als User möchte ich mich anmelden.

☑ Als User möchte ich meinen account Löschen können.

☑ Als User möchte ich meinen Passwort ändern können.

☑ Als User möchte ich meinen Remember Cookie einstellen können.

☑ Als User möchte ich meine Token invalidieren können.

☒ Als User möchte ich meine Adressen veröffentlichen können.

☑ Als User möchte ich öffentliche Ips importieren/namen.

☑ Als User möchte ich Adressen manuell eingeben können.

☐ Als User Ipv4 und Ipv6 (evtl. onion-url) validierung.

☐ Als User möchte ich meine Adressen in Gruppen teilen können.

☐ Als User möchte meine Aderssen als QR-Code angezeigt bekommen.

☐ Als Admin möchte ich eine Firewall.

☐ Als Admin möchte ich AGB und Datenschutzerklärung haben.

☐ Als Admin möchte ich IPs GEO blocken bzw. differrenzeiren.

☑ Als User möchte meine Aderssen in Echtzeit aktualisiert haben.

- ⏩ Als Admin möchte Rate Limits auf den Routen.
- ☒ Als User möchte ich einen Cookie banner.
- ☐ Als Visitor möchte ich keine Cookies.
- ☐ Als User möchte ich über E-Mail informiert werden können wenn der Service nicht verfügbar ist oder sich ändert.
- ? Als User möchte ich einen Reverse VPN öffnen können.
- ☐ Als User möchte ich mein Adressen einfach mit Api abrufen können.
- ☐ Als User und Visitor möchte ich ein übersichtliches Interface.
  - ☐ Kenntlich machen von Public und Private
  - ☐ Als User und Visitor möchte ich Mehrsprachigkeint. (Englisch und Deutsch evtl. weitere)
  - ☑ Als User und Visitor möchte ich durch klicken Adressen und Namen Kopieren.
  - ☐ Teilen durch klicken auf IP ist unklar. -> Share Icon?!
  - ☐ Als User und Visitor möchte ich Tooltips.
- ☐ Als Visitor möchte ich das es Schwer oder unmöglich ist einen Link zu erstellen der automatisch/unfreiwilliges meine Adresse teilt.
- ? Als Admin möchte ich Datenbank Migrationen.
- ☑ Als Admin möchte ich Blueprints.
- ☐ Als Admin möchte ich Logging, um Angirffe und Fehler zu erkennen.

2.2 Frameworks und Entwicklungsumgebung

Ich verwende Flask als Framework und programmiere in PyCharm Professional 2023. Mein verwalte mein Python Environment und meine Pakete mit Anaconda[1]. Die verwendete Python Version ist 3.11.5. Zur Versionierung

verwende ich git und GitHub. Die verwendeten Bibliotheken und deren Versionen finden sich im environment.yml oder im requirements.txt. Hier eine Kure Übersicht:

- Flask
  - APScheduler
  - Bcrypt
  - Limiter
  - Login
  - SocketIO
  - SQLAlchemy
  - WTF
- NumPy
- OpenCV
- qrcode
- Pillow
- Requests
- PyJWT
- urllib3

2.3 Datenbankstruktur

Die Datenbank ist relativ simple es gibt eine User- und eine Address-Tabelle. Es gelten unter anderem folgende Abhängigkeiten:

$$id \dot{\rightarrow} passwort \tag{2.1}$$
$$id \dot{\rightarrow} remember \tag{2.2}$$
$$id \dot{\leftrightarrow} alternative\_id \tag{2.3}$$
$$id \dot{\leftrightarrow} name \tag{2.4}$$
$$id, device\_name \dot{\rightarrow} address \tag{2.5}$$
$$id, device\_name \dot{\rightarrow} last\_updated \tag{2.6}$$

Normalform	ok/nok	Begründung
1NF	ok	Alle Domänen enthalten nur einfache atomare Werte.
2NF	ok	address und last_updated sind voll funktional abhängig von (id, device_name).
3NF	ok	Es gibt keine transitive Abhängigkeit.
BCNF	ok	Die Determinante (id, device_name) ist auch Schlüsselkandidat.
4NF	ok	Es gib keine mehrwertige Abhängigkeit.
5NF	nok	Eine Teilung in address(id, device_namem, address) und updated(id, device_namem, last_updated) wäre möglich.

Tabelle 2.1: Normalformen shared_addresses

Abbildung 2.1 zeigt das Entity-Relationship-Modell der Datenbank. Zwischen shared_addresses und user besteht eine nur-eins-zu-null-oder-mehr-Beziehung. Der FremdschlÃ¼ssel user ist gleich der id in der user-Tabelle. Der name ist unique und daher auch SchlÃ¼sselkandidat der Tabelle Auch die alternative_id ist ein eindeutiger SchlÃ¼sselkandidat. Alle drei id, alternative_id und name determinieren sich gegenseitig.

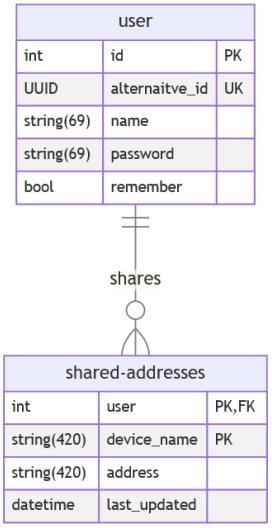

Abbildung 2.1: Entity-Relationship-Modell der DB

Wie Tabelle 2.1 und Tabelle 2.2 zeigen sind beide Relationen in 4ter Normalform. Eine überführung in die 5te Normalform ist hier nicht nötig da die Relationen keine mehrwertigen Abhängigkeiten haben. Dementsprechend lassen sich durch eine Teilung auch keine neuen Informationen gewinnen.

Normalform	**ok/nok**	**Begründung**
1NF	ok	Passwort könnte vom Salt getrennt werden.
2NF	ok	password und remember sind voll funktional abhängig von z.B. id.
3NF	ok	Es gibt transitive Abhängigkeiten nur über die Schlüsselkandidaten.
BCNF	ok	Jede der drei Determinanten ist auch Schlüsselkandidat.
4NF	ok	Es gib keine mehrwertige Abhängigkeit.
5NF	nok	Teilung in Alternative(id, alternative_id), Name(id, name), Password(id, password) und Remember(id, remember) möglich.

Tabelle 2.2: Normalformen user

Kapitel 3

Entwicklung

Beschreibung der Funktionen (z.B. mit Bildschirmfotos) sowie technischer Herausforderungen (Code-Schnippsel)

Kapitel 4

Inbetriebnahme

Schritte zur Inbetriebnahme (auf einem lokalen Rechner)

Kapitel 5

Fazit

Was wurde von den ursprÃ¼nglichen Anforderungen umgesetzt? PersÃ¶nlicher Eindruck, ErweiterungsmÃ¶glichkeiten

Literaturverzeichnis

[1] Anaconda.

[2] Whatismyip.com.

[3] Pascal Brachet. Texmaker, 9. Oktober 2022.

[4] Chris Parker. Whatismyipaddress.com.

[5] Knut Sveidqvist. Mermaid diagramming and charting tool.

[6] Vector. Sae j1939 offener standard fÃ¼r die vernetzung und kommunikation im nutzfahrzeugbereich.

[7] Wilfried Voss. Sae j1939 programming with arduino - transport protocol â€“ rts/cts session (sae j1939/21), 15. Oktober 2018.