CycleBuddy

*Modul: PM3 Fronted-, Web und Softwareengineering*

*Bearbeitende: Lüthi Isabelle, Stuhlmann Linus  
Hochschule: ZHAW, Winterthur  
Datum: 01.12.2023*

# Aufgabenstellung

Develop a webbased client server solution with Python.

Database: Mongo DB

Webserver: Flask

Client: Webbrowser

Chart: Chart JS

Minimum Working Solution: The server reads OSM (OpenStreetMap) data (ie. Restaurants) from the DB and delivers it tot he client. The client shows a map oft he POIs as well as a chart (ie. Number of restaurants for each canton)

Presentation: due at end of course (SW13 / 14)

# Rollen

Product Owner (PO) Henrik Stormer  
Scrum Master (SM) Isabelle Lüthi  
Team Member (TM) Linus Stuhlmann, Isabelle Lüthi

# User Story

Ich als Kunde brauche eine digitale, auf die Bedürfnisse von Fahrrad fahrenden Personen (z.B. Parkplätze, Reparaturstationen, Trinkwasser) zugeschnittene Karte, welche via Internet erreichbar ist und sowohl auf dem Gerät zuhause wie auch auf dem Mobiltelefon genutzt werden kann, damit ich mehr mit dem Fahrrad und damit nachhaltiger unterwegs bin und für mich wichtige Orte, wie ein Fahrradparkplatz oder eine Reparaturstation, einfacher in meiner Nähe finden kann.

# Aufgabenformulierung / Definition of Done:

## Aufgaben Isabelle:

Am Donnerstag, 12.10.2023 ist das Repository «cycle\_buddy\_app», welches von TM Linus erstellt wurde, auf das von Isabelle verwendete Gerät geklont und fertig eingerichtet, damit die Vorgänge Push und Pull genutzt werden können.

Am Donnerstag, 12.10.2023 sind die Programme PyCharm und MongoDB auf dem Laptop installiert und für das Projekt entsprechend eingerichtet. Somit sind Grundeinstellungen und Speicherorte angepasst, damit PyCharm sowohl auf MongoDB als auch auf das GitHub-Repository «cycle\_buddy\_app» zugreifen kann.

Am Donnerstag, 12.10.2023 hat TM Isabelle die Dokumentation aktualisiert und das entsprechende Word Dokument auf GitHub zur Verfügung gestellt.

Am Donnerstag, 12.20.2023 hat TM Isabelle die Skizze der App erstellt. Zudem hat sie mindestens zwei mögliche Statistiken und eine mögliche Erweiterung der App notiert. Diese Ideen müssen noch nicht genauer bearbeitet sein, sie sollen jedoch für die Besprechung mit TM Linus ausreichend ausformuliert sein.

Am Donnerstag, 12.10.2023 hat TM Isabelle die Lernvideos und Folien der Schulwochen SW 02 «Internet communication», SW 03 «Flask», SW 04 «MongoDB» und SW 05 «HTML Introduction» geschaut und bearbeitet.

## Aufgaben Linus:

Bis zum Dienstag, 10.10.2023 erstellt TM Linus das Repositorium «cycle\_buddy\_app» auf GitHub und fügt TM Isabelle hinzu. Innerhalb des Repositoriums werden die Ordner «documents», «code» und «data» erstellt, ebenso wird ein «gitignore»-File generiert, in welchem Dokumente und Umgebungen genannt werden, welche nicht via GitHub geteilt werden dürfen. Sobald das Repositorium erstellt wurde, wird TM Isabelle per Whatsapp oder Mail informiert.

Am Donnerstag, 12.10.2023 hat TM Linus die Themen «Flask» und «MongoDB» nach seinem Ermessen repetiert / aufgefrischt, dies durch das Schauen der Lenvideos auf Moodle oder Youtube oder auch durch das Nachlesen im Internet.

Am Donnerstag, 12.10.2023 hat TM Linus den Code, welcher für das Einlesen des Json-Files in die Datenbank (hier MongoDB) benötigt wird, fertig codiert. Da das komplette File für die Datenbank zu gross ist, muss der Code eine Filterung nach den benötigten Amenities (in Liste vordefiniert) beinhalten.

Am Donnerstag, 12.10.2023 hat TM Linus das anstehende Scrum Meeting vorbereitet, die Traktanden notiert und diese in der Dokumentation (Word-Dokument) via GitHub zur Verfügung gestellt.

# Grobplanung

|  |  |
| --- | --- |
| Woche 1 | Unterlagen studieren, erste Themen bearbeiten, Ideen sammeln, |
| Woche 2 | User Story schreiben, Anforderungen festlegen, Aufgaben verteilen |
| Woche 3 | Projekt aufsetzen (Python, Flask, MongoDB) |
| Woche 4 |  |
| Woche 5 |  |
| Woche 6 |  |

Velo-App

* Reparaturstationen (bycicle\_repair\_station), Parking (bycicle\_parking), Pumpen (compressed\_air), Fahrradmiete (bycicle\_rental), evtl. Wasserspender (drinking\_water)
* Grafen / Tabellen über: Anzahl der jeweiligen Amenities in der Nähe (entweder die nächsten 5 oder Umkreis von 2-10km (evtl. anwählbar))
* Location abfragen
* Evtl. mit Tracker erweitern, für Strecke und Geschwindigkeit
* Evtl. Fahrradwege einbauen
* Evtl. Route einbauen
* Nachhaltige Fortbewegung wird unterstützt

Allgemeine Statistiken einbauen: pro Kanton, Verhältnis von Parking zu Pumpen (amenities miteinander vergleichen 🡪 evtl pro Kanton oder allgemein)

Lösung implementieren, die jemand brauchen könnte, velo app ist okey 😊 dürfen uns auch inspirieren lassen

Minimal: 3 Charts + 1 Karte

Wichtig einbauen: Filtering, Location abfragen, … (optimiert)

Json nach amenities filtern und dann in mongo db einbauen

Wir dürfen externe daten miteinbeziehen / verwenden (Dateien in Datenbank einbauen oder Dienst, welchen man anfragt und Antwort erhält)

Client Server Architecture

POIs in OSM = Node: Nodes can have different attributes. POIs are tagged with the attribute amenity. Each node has an id, a latitude (lat) and longitude (lon) value.

Anzahl Amenities?

Auswählen, welche amenitiy

Filterliste: wahl nach amenity, wahl nach kanton

Dann umrechnen pro kanton pro kopf ausrechnen (wie viele f-parkplätze hat es im kanton TG um vergleich zum kanton ZH)

Aufgaben