**Cílová skupina**

### Uživatel

* Vícero uživatelů, jejich věci na míru = uživatelský profil

### Problémy k řešení + potřeby

* Získat přehled o poslaných
* Zpětná vazba - analýza
* Usnadnění hledání, při příliš mnoho webech a zdrojích (i lokální weby firem)

### Features

* Přehled o tom kam bylo napsáno s jakým výsledkem
  + Kdy, kdy odpověď na které kolo
  + Hodnocení kola (osobní pocit šance)
  + Kolo 1-X, prescreening, screening, pohovor 1-X) - tohle asi spíš jako faktovka, protože kol a jejich hodnocení může být mrtě, údaje o firmě budou asi vždy stejné
  + Zda byl či nebyl test a jaký (doma, na místě, technologie...)
* Nejčastější technologie (klíčová slova) inzerátů
  + Mezifaktovka, dimenze a faktovka
  + Klíčová slova technologií (z inzerátu)
  + Klíčová slova o pozici/benefitech (HO, remote, dog-friendly office, flexibilní...)
* Okopírovaný popis pozice
* Okopírovaný motivák
* Analýza hledání práce
  + Kolik obepsáno za vybrané období
  + Jaké výsledky do pohovoru, do nabídky
  + Analýza klíčových slov kam bylo psáno
* Blacklist firem
* Kontaktní údaje
  + Lidí, kterým jsem psala
  + Lidí, se kterými se mám sejít etc. Asi stačí i jméno (LI, reference v budoucnu)
* Firma, pozice
  + Název pozice plný
  + Název firmy (asi dimenze s údaji firmy)
    - Velikost, typ, údaje
  + Typ pozice (on-site, hybrid, remote)
  + Město, stát
* Reakce
  + Kdy na jaké kolo
  + Osobní hodnocení, pocit z kola, z firmy
* Nabídnutá částka
* ----------------------- Tohle už je hudba budoucnosti:
* Výběr inzerátů z webů podle zadaných kritérií (pokud půjde)
  + API nebo crawler (24 h)
  + Vícero webů
  + Možnost přidat si vlastní weby
* Vyhledávání v poslaných atd.

### Očekávané používání

* Android (primární)
* Web (sekundární - zabezpečení? Přihlášení?)
* Desktop win (asi až jako poslední)

### Výběr technologií

* Python
  + Framework?
  + Knihovny?
* Databáze
  + Jaký dialekt SQL?
  + Vývoj lokálně, ale podle cloudové databáze, aby dialekt a nastavení databáze si sedělo (tedy nejprve vybrat cloud DB a pak podle toho local)
    - Oracle Cloud free tier, Mongo DB Atlas
    - Jde použít i google tabulky, airtable
    - Microsoft omezeně: Azure Cosmos DB, na rok: Azure database, Azure SQL database

### Persony - imaginární typičtí uživatelé

* Potřeby a preference
* Očekávání
* Potřeby a chování

### Scénáře a use cases

* Interakce uživatelů s app
* Hlavní účely a akce

### Návrh UI/UX

* Tak na měsíc zaplatit illustrator a nakreslit prvky v tom
* Najít aplikaci podobnou adobe UX aplikaci
* Přehled o napsaných
  + Zapsat údaje
* Přehled nabídnutých
  + Přejít na nabídku (primárně odkázat na web, ale později třeba i napsat rovnou?)
* Vyhledávání
* Analýzy

### Architektura

* Architektura aplikace, rozložení komponent, datových toků, způsobu komunikace mezi částmi systému
* Škálovatelnost, bezpečnost, udržovatelnost

### Návrh databáze

* Struktura, tabulky, indexy
* Vhodný databázový systém
* SQLite běží na mobilech a na android, databázi instalovat lokálně, každý bude mít svou vlastní, bez účtů

**Bezpečnost, testování, optimalizace a škálovatelnost**

**Základní grafiky jdou udělat s pomocí Canva, MS designer**

**Zvážit i Phind**

### **Android:**

* **SQLite Database:**
  + Android commonly uses SQLite as a local database solution. You can create a SQLiteOpenHelper or Room Database class in your Android app to handle database creation and setup.
* **Asset Folder:**
  + Include an initial SQLite database file in the "assets" folder of your Android app. This file will be copied to the device during the installation process.
* **Database Helper Class:**
  + Create a Database Helper class that extends **SQLiteOpenHelper**. In the **onCreate** method, you can execute SQL statements to set up tables and initial data.
* **Install-Time Database Initialization:**
  + During the app's first launch, you can check if the database file exists. If not, copy the initial database from the assets folder and perform any necessary setup.

Trénink data scraper a parser o pohybech na burze do databáze.

Trénink AdventtureWorks výzkumné otázky přes Bing Copilot (dokáže i říct odpovědi.

* Zpočátku asi cílený scrape (zadej url inzerátu)
* Začít rozkreslovat vztahy a db na základě již existujícího a toho, co chci dělat

## Vyhledávání frekvencí slov z nějakého textu. Vyzkoušet na web scraperu.

* Frekvence slov
* Hledání, zda obsahuje klíčová slova

import nltk  
from nltk.corpus import stopwords  
from nltk.tokenize import word\_tokenize  
  
nltk.download('punkt')  
nltk.download('stopwords')  
  
text = "Toto je ukázkový text, který obsahuje několik slov. Některá slova se vyskytují vícekrát, jiná jen jednou. Text obsahuje také předložky, zájmena a další slova, která nejsou klíčová."  
  
# Tokenizace textu  
tokens = word\_tokenize(text)  
  
# Odstranění stop slov  
stop\_words = set(stopwords.words('czech'))  
filtered\_tokens = [word for word in tokens if word.lower() not in stop\_words]  
  
# Odstranění interpunkce a číslic  
filtered\_tokens = [word for word in filtered\_tokens if word.isalpha()]  
  
# Odstranění koncovek  
stemmer = nltk.stem.SnowballStemmer('czech')  
filtered\_tokens = [stemmer.stem(word) for word in filtered\_tokens]  
  
# Výpočet frekvence slov  
freq\_dist = nltk.FreqDist(filtered\_tokens)  
  
# Vypsání nejčastějších slov  
print(freq\_dist.most\_common(10))

* Nejsnazší asi fakt bude udělat webovku
* Uživatel a komunikace s databází
* <https://flet.dev/docs/> zkusit flet? (běží na Flutter pro tvorbu multiplatform aplikací)
* Web scraper mít napsaný specificky pro každý web - hlavně pro konkrétní weby firem, než jobs etc.
* Interview jako sales funnel, calculate the conversion (a konverzi k nabídce od pohovoru)
  + Nabídek/rozeslaných poptávek \* 100 = %
* Zvážit WSL(2) a zjistit o co přesně jde a jak se to používá
* Jak zajistit bezpečnost? Stačí offline use?

## Postup

* “Skica”:
  + DB design
  + Flowchart
    - Vstup, úložiště, zpracování dat
    - Výstup, jak uživatel může mít přístup k datům
  + Data pipeline
  + GUI design – jen wireframe (Figma může pomoct, Sketch, Adobe XD, InVision)
    - Pár malých grafů (matplotlib?)

## Readme

This project is made to make my life easier and save myself a lot of tedious manual work when looking for a job and to have better feedback and understanding of my applying strategy and companie’s hiring strategy.

I do like databases and for some time it was enough to just write couple lines of where and when did I apply for a job. Later on I needed more. The frontend part of this project was never my goal, but it was necessary. I have never built an app before, so this posed a whole new set of challenges.

My attitude towards this project was based on how it could look like in an ideal world where I knew everything about making an app. Then I have took the core part which I needed most and started there with what I already knew or at least had an idea how it may be done.

* Jak nainstalovat a rozběhnout na vlastním stroji (detailní) - ideálně odkaz na web
* Co udělat pro to, aby mohli rozšiřovat a pracovat na tom taky (pull, fork něco takového)
* That there was an idea for adding a user profile but that would mean a ton of things I do not know yet and don’t want to learn at this moment, so I focused on my priority GUI communicating with a database.