

# Projekt resekollen



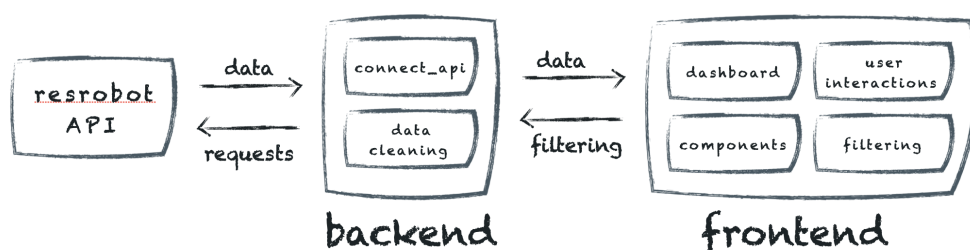
## Syfte

Projektet syftar till att implementera backend och frontend för att lösa ett verklighetsnära problem. Förutom att använda ett visst antal tekniker finns ett fokus kring att jobba tillsammans i ett mjukvaruteam. Detta innebär både agil utveckling och att kunna använda git och github i ett team.

## Bakgrund

Företaget Resekollen AB vill bygga en användarvänlig dashboard för att både visualisera statistik och för att planera resa med kollektivtrafik i Sverige. En ide med deras produkt reskolla är att man ska kunna planera en resa med kollektivtrafik oavsett stad man befinner sig i. På så sätt särskiljer sig reskolla från konkurrenter som vanligtvis visar upp lokal

Reskollen har haft en BI-utvecklare (BI) och en Pythonutvecklare (PY) som tillsammans med product owner (PO) byggt kommit fram till en övergripande arkitektur. PY och BI skulle egentligen velat göra färdigt fullstacklösningen med komplett dokumentation, tester med mera, men företagets VD ville snabbt se en prototyp för att kunna söka investering.



Dessvärre blev PY headhuntad till ett annat bolag och valde att säga upp sig. Innan han slutade implementerade han modellen till viss del och BI-utvecklaren lyckades bygga en initial dashboard. När PY slutade skickade företaget BI-utvecklaren på konsultuppdrag hos en kund.

VDn visade detta för sina investerare och fick in riskkapital från bland annat Yohanna. Yohanna som tidigare investerat i dataintensiva projekt förstår vikten av noggrannhet, då det oftast straffar sig av att ta genvägar. Därför har hon gett PO mer befogenheter att göra det här projektet noggrant. Dessutom är det viktigt med fler funktionaliteter i dashboarden som ger bättre kundupplevelse.

## Roller i organisationen

Person	Roll	Beskrivning
Yohanna	riskkapitalist	investerat i dataintensiva projekt, begränsad teknisk kompetens, bra businesskunskaper
Roager	VD	tuff förhandlare, duktig på business
Börje	product owner	begränsad teknisk kompetens, bra kontakt med både Yohanna och Jacobi
Feliz	frontendutvecklare	sitter för närvarande med annat projekt, men kan assistera med git och github
Yulia	BI-utvecklare	sitter ute hos kund

## Scenario

Yohanna har varit med i tidigare projekt som letts av studerande på yrkeshögskolan inom data engineering och varit mycket nöjd över resultaten. Därför bestämmer sig Yohanna tillsammans med Roager att ta in LIA till detta projekt också.

Teamet i Reskollen skissar hastigt en onboardingplan och en kravställning.

## Onboarding

Roager berättar bakgrund

Feliz går igenom hur ni ska jobba med git och github i det här projektet

- bjuda in teamet
- git branches
- pull requests
- github projects
- tilldela personer till tasks
- issues

Börje går igenom agil projektmetodik i arbetskollen

- jobba enligt kanban
- backlog refinement
- definition of done
- 1 task åt gången

## Uppgift 0 - uppvärmning

Det som beskrevs i bakgrunden av det som PY och Yulia satte upp hittar ni i föreläsningarna 11-12 i kursrepot. Föreläsning 13 behandlar git och github i team. Se till så att varje person självständigt går igenom dessa föreläsningar och lär sig innehållet.

Detta kan göras asynkront, dvs att andra uppgifter kan göras utan att ha gått igenom alla lektionerna.

## Uppgift 1 - setup

### Githubrepo

En person sätter upp ett publikt githubrepo och bjuder in övriga. Sätt även upp github projects för att ha en kanbanboard där ni kan se vilka tasks som behöver göras och vilka som jobbar med vilken task.

#### VIKTIGT

Kom ihåg att jobba i egna branches och kör pull från main branchen innan ni gör pull request till main. På så sätt löses eventuella merge conflicts i sin egna branch.

Gör många committs, vänta inte på att göra få stora committs. På så sätt versionshanteras koden bättre, ni får backup och kodbasen utvecklas över tid.

### Virtual environment

En person installerar de paket ni behöver och tar fram en requirementsfil som pushas till github. Övriga personer installerar de olika dependencies som finns i requirementsfilen.

### CI/CD

Sätt upp github actions för CI/CD för projektet, se lektion 14.

## Uppgift 2 - EDA över resrobot

Börje berättar att PY tidigare sagt att resrobot API har mycket att bjuda på som PY inte haft tid att undersöka. Gör explorativ undersökning av reseplanering, tidtabeller för olika hållplatser i olika städer och se vilken data ni kan plocka ut från APIet som eventuellt kan vara värt att visa upp på dashboarden.

## Uppgift 3 - tidtabellsmodul

PY har byggt trips.py i backend och BI har byggt frontenden, som ni märker är det byggt med OOP och ganska välstrukturerat. Bygg en

- tidtabellsmodul i backenden som exponerar relevant data som frontenden kan använda.
- tidtabellskomponent i frontenden som dashboarden kan använda sig av för att visa tidtabellen

Tidtabellen ska minst ha följande features

- visa avgående tåg/bussar/spårvagnar för specifik hållplats
- kan filtrera på hållplats
- visa tid kvar till tåg/bussar/spårvagnar i minuter
- visa för en timme framåt i tiden
- annat som ni bestämmer

## Uppgift 4 - reseplaneraren

PY och Yulia har byggt data och visualisering för en karta mellan Umeå och Malmö. Städerna var tyvärr hårdkodad för prototypen. Nu ska ni låta användaren kunna söka på två städer och den ska kunna visa upp:

- karta med punkter för varje station mellan ursprung och slutdestination
- antal stopp emellan
- antal byten
- total tid för resan
- annat som ni bestämmer

## Uppgift 5 - dashboarden

Snygg och användarvänlig är ledorden här. Yohanna och Roager ser framemot att bli positivt förvånad över era coola dashboards. Tänk ut features som ni tror användaren skulle tycka om med er dashboard, skriv ned dem i ert projektbacklogg och implementera dem.

## Presentera för stakeholders

Ni har 15 minuter per grupp att presentera ert projekt. Tänk på att er publik består av Börje, Feliz, Yohanna, Roager, Yulia och klassen så

- presentera dashboarden
- presentera hur ni jobbat agilt
- kort presentera er implementation
- hur ni jobbat med git och github i teamet

### VIKTIGT

ALLA ska presentera

## Bedömning

Projektet bedöms individuellt, ni kommer skicka in er gemensamma kodbas i github och kanbanboardet. Där går det att följa vad respektive person gjort under projektet.

## Godkänt

- gjort uppgifterna korrekt i grupp
- varit aktiv i projektet, utfört relevanta tasks
- gjort flera relevanta commits med pull requests mot main

## Väl godkänt

- projektet är utfört på tillräcklig hög nivå
- koden är strukturerad och OOP har använts på ett effektivt sätt