

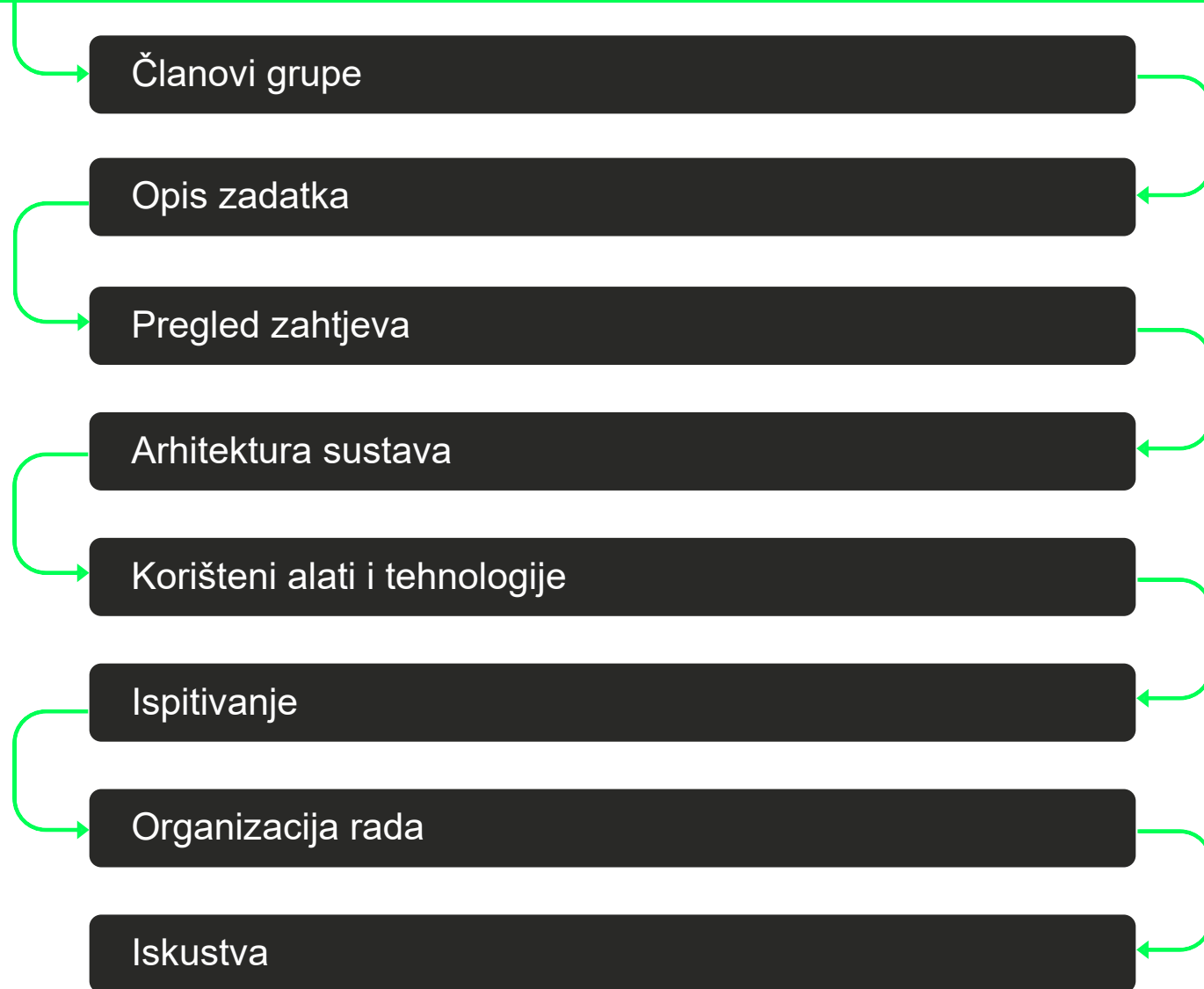
# GEAR SHARE

---

Tim: TG 02.2 Pokemoni

Ak. god. 2025./2026.

# SADRŽAJ



# ČLANOVI GRUPE

- Nikola Horvat → Voditelj, baza podataka, deployment
- Noa Bašić → Dokumentacija
- Igor Vuković → Dokumentacija
- Jakov Marković → Dokumentacija, backend
- Leon Katić → Backend
- Antonio Batarilović → Frontend
- Leonard Kovač → Frontend

# OPIS ZADATKA

## PROBLEM

Kupnja sezonske sportske opreme je skupa i rizična jer se koristi rijetko, dok trgovci imaju neiskorištene zalihe i neefikasan sustav rezervacija



## RJEŠENJE

Povezivanje kupaca sezonske sportske opreme i trgovaca u cilju jednostavnijeg, povoljnijeg i fleksibilnijeg najma opreme

# OPIS ZADATKA

USPOREDBA SA SLIČNIM RJEŠENJEM

## DECATHLON RENT

- Decathlon kao trgovac
- Lokacije ograničene na Decathlon dućane
- Decathlon brandovi
- Zatvoren na odluke Decathlona

## GEAR SHARE

- Bilo tko može biti trgovac
- Nema ograničenja u lokacijama
- Nema restrikcije na brandove
- Prilagođava se potrebama zajednice

# PREGLED ZAHTJEVA

## GLAVNI FUNKCIONALNI ZAHTJEVI

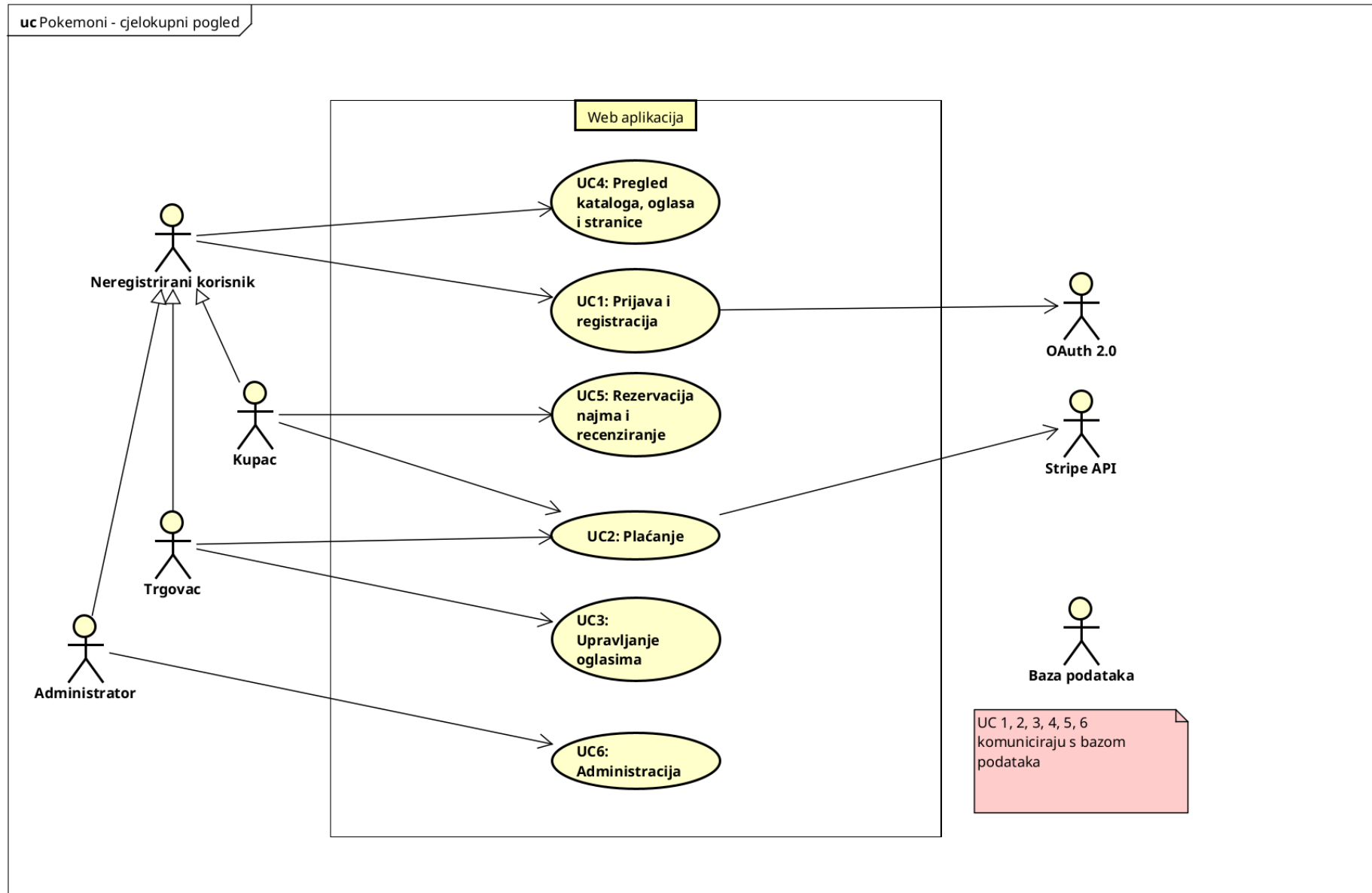
- Sustav omogućuje korisnicima registraciju i prijavu putem OAuth 2.0 autentifikacijskog sustava
- Sustav omogućuje javni pregled kataloga i detaljne stranice oglasa
- Korisnik može rezervirati period najma uz provjeru dostupnosti i privremeno zaključavanje termina do plaćanja.
- Trgovac može platiti i aktivirati članarinu potrebnu za objavu oglasa
- Administrator može pregledavati i upravljati korisnicima, moderirati oglase te obrađivati prijave nepravilnosti

# PREGLED ZAHTJEVA

## NEFUNKCIONALNI ZAHTJEVI I DOMENE PRIMJENE

- Vrijeme učitavanja stranica (initial load) ne prelazi 3 sekunde.
- Prosječno vrijeme odgovora API-ja  $\leq 500$  ms za tipične upite.
- Podaci korisnika šifrirani su u pohrani (enkripcija baze i objektne pohrane).
- Dostupnost sustava najmanje 99% mjesečno.
- Kod je modularan, lako proširiv i organiziran po slojevima.

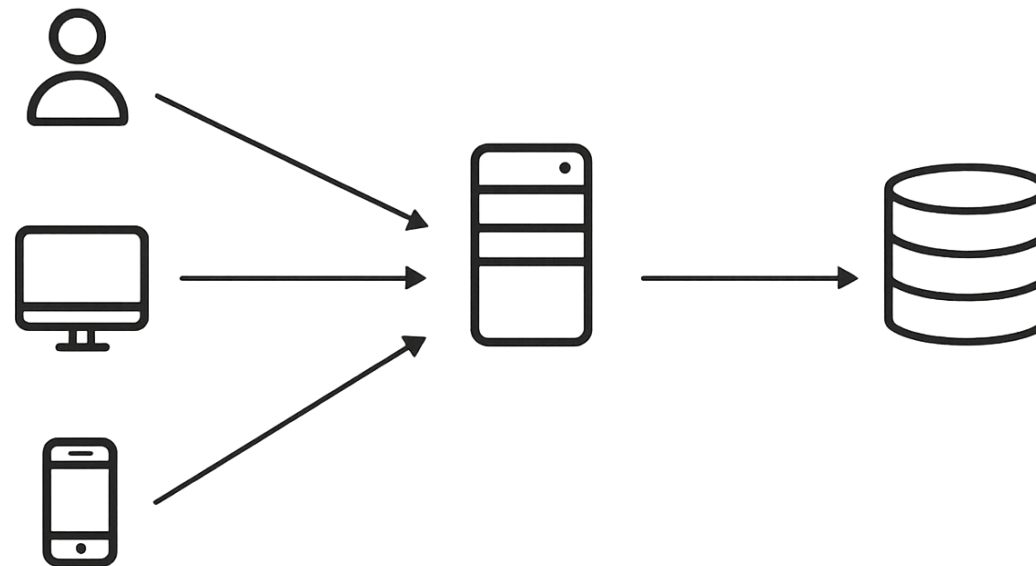
# UML DIJAGRAM OBRAZA UPORABE



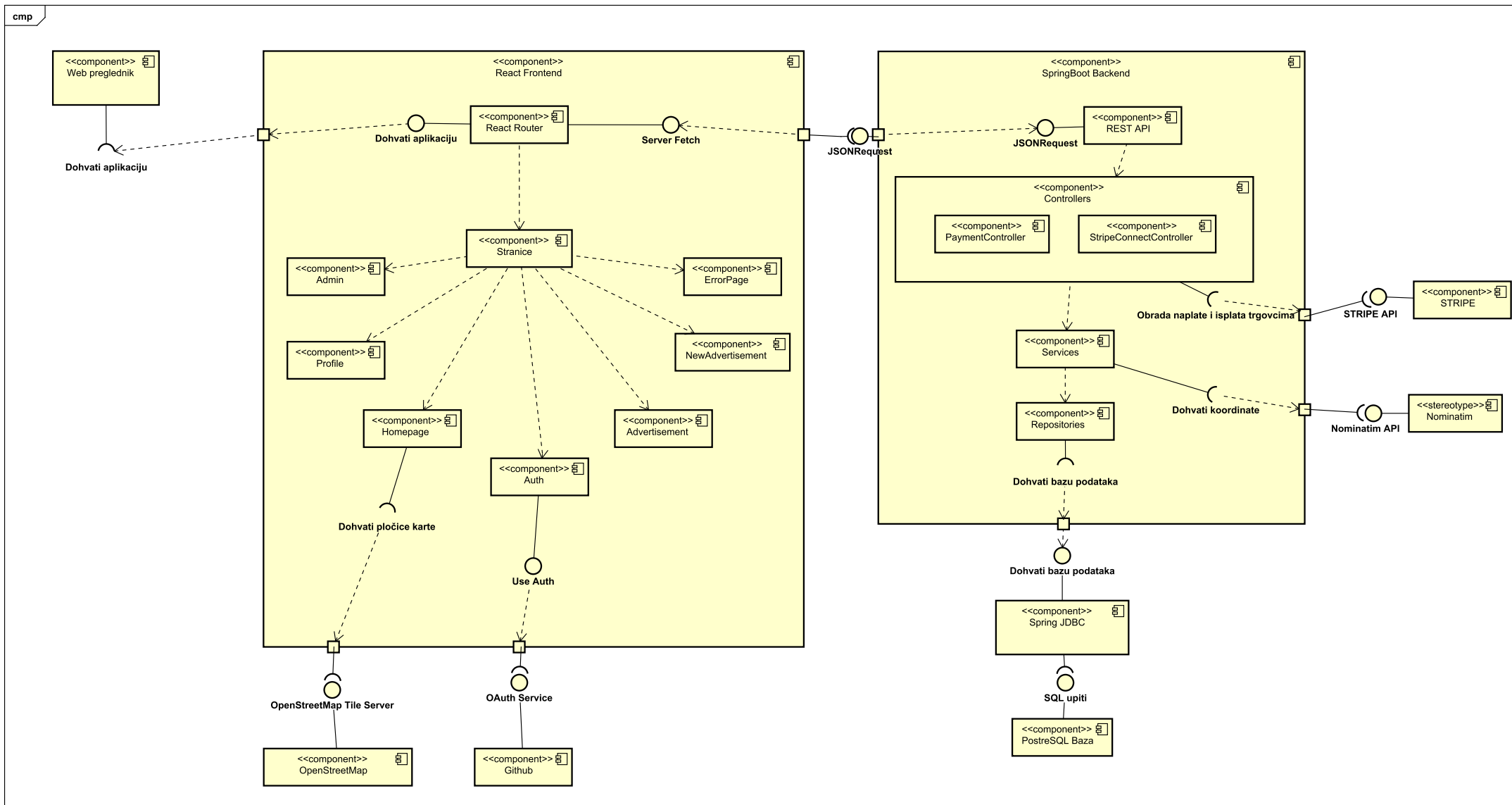


# ARHITEKTURA SUSTAVA

- klijent–poslužitelj arhitektura
- podsustavi
  - Frontent - korisničko sučelje
  - Backend – MCV arhitektura
  - Baza podataka
- Visoka kohezija, Slojevitost, horizontalno skaliranje, Sigurnost, Operativna jednostavnost



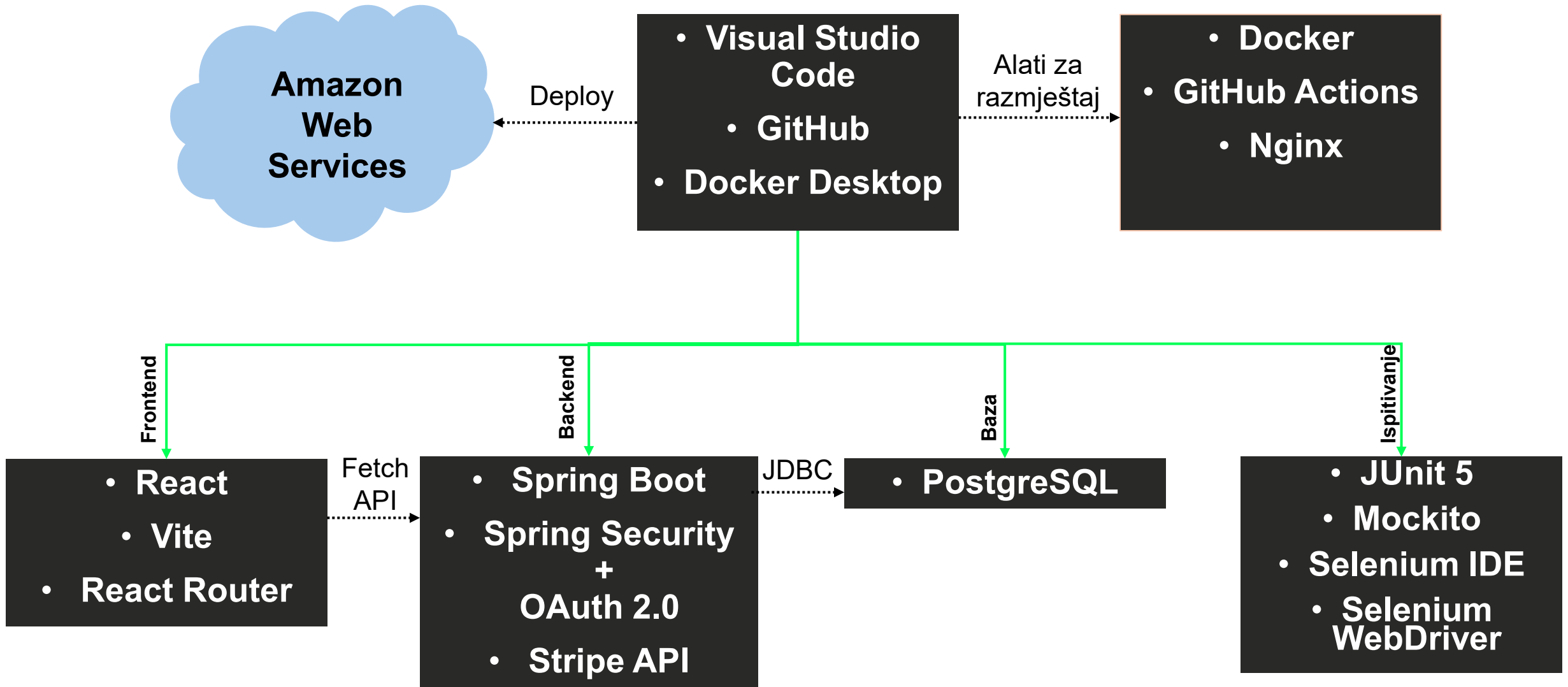
# ARHITEKTURA SUSTAVA



# ISPITIVANJE

- Backend ispitivanje:
  - Testirana poslovna logika aplikacije
  - Provjerena ispravnost rada servisa i API ruta
  - JUnit 5, Mockito i Spring MockMvc
- Frontend / sustavsko testiranje:
  - Testirano ponašanje aplikacije iz perspektive korisnika
  - Provjereno za normalne slučajeve, rubne situacije i greške
  - Selenium IDE i Selenium WebDriver

# KORIŠTENI ALATI I TEHNOLOGIJE



# ORGANIZACIJA RADA

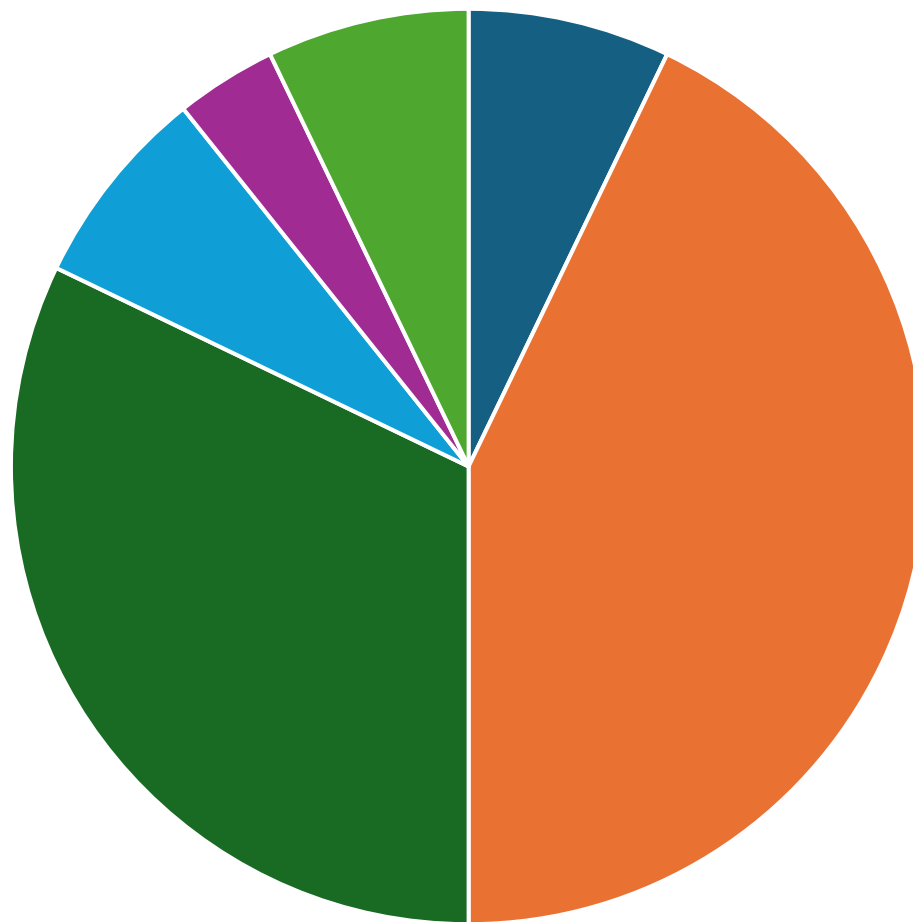
## ORGANIZACIJA KOMUNIKACIJE

- Discord
- Podkanali za pojedine timove
- Podkanali za komunikaciju između timova
- Izvještaj voditelja o napraavljenom i sljedećim koracima

## VODOPADNI MODEL

- Zašto:
  - Zahtjevi poznati prije oblikovanja
  - Zahtjevi se rijetko mijenjaju
- Tijek:
  1. Analiza zahtjeva
  2. Oblikovanje sustava
  3. Implementacija i ispitivanje
  4. Integracija i ispitivanje sustava
  5. Rad sustava i održavanje

# ORGANIZACIJA RADA



■ Specifikacija ■ Frontend ■ Backend ■ Baza podataka ■ Testiranje ■ Dokumentacija

# ISKUSTVA

- Bolja raspodjela zadataka u startu
- Bolja komunikacija
- Više sastanaka uživo
- Rad na vrijeme
- Preuzimanje tuđih poslova