FAZA III – KOMUNIKACIJA

1. Cilj faze

Cilj treće faze razvoja bio je uvođenje **asinhronog modela komunikacije** između komponenti sistema. Umesto oslanjanja isključivo na direktne (sinhrone) pozive, sistem je nadograđen tako da koristi **Message Broker** (RabbitMQ) za razmenu događaja u realnom vremenu.

2. Tehnološki stek za komunikaciju

- **Message Broker:** RabbitMQ (korišćen za upravljanje redovima poruka).
- **Protokol:** AMQP (Advanced Message Queuing Protocol).
- **Framework:** NestJS Microservices modul.
- **Transportni nivo:** `Transport.RMQ`.

3. Realizovani scenariji asinhronog rada

U tabeli su prikazani ključni događaji koji se emituju kroz sistem i njihova svrha:

Događaj (Event Pattern)	**Izvor (Service)**	**Svrha i Logički ishod**
`user_created`	`UsersService`	Potvrda uspešne registracije; osnova za slanje email obaveštenja.
`trip_plan_created`	`TripPlansService`	Logovanje kreiranja novog plana i obaveštavanje sistema o novoj destinaciji.
`member_added`	`TripMembersService`	Signalizacija da je novi korisnik dodat u grupu za putovanje.
`trip_locked`	`TripLocksService`	**Real-time lock:** Obaveštava sistem da korisnik trenutno menja plan (sprečavanje konflikata).
`trip_unlocked`	`TripLocksService`	**Real-time unlock:** Signalizira da je plan ponovo slobodan za editovanje drugima.

4. Komponenta za komunikaciju (`CommunicationController`)

Centralno mesto za obradu svih pristiglih poruka je `CommunicationController`. On koristi `@EventPattern` dekoratore za hvatanje poruka iz RabbitMQ-a. Trenutna implementacija simulira realni sistem notifikacija ispisivanjem detaljnih logova u terminalu, čime se dokazuje uspešan protok podataka kroz Message Broker.

5. Zaključak

Implementacijom Faze III, aplikacija je prešla sa monolitnog razmišljanja na **distribuirani model**. Uspešno je demonstrirano da promene na bazi podataka automatski trigger-uju asinhroni odgovor u komunikacionom modulu.