

---

# RawMatch - Приложение за осиновяване на домашни любимци

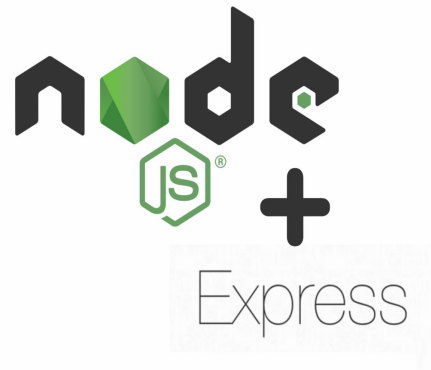
Технически спецификации и преглед на технологиите,  
които ще се използват за един проект с кауза

---

---

# Backend технологии

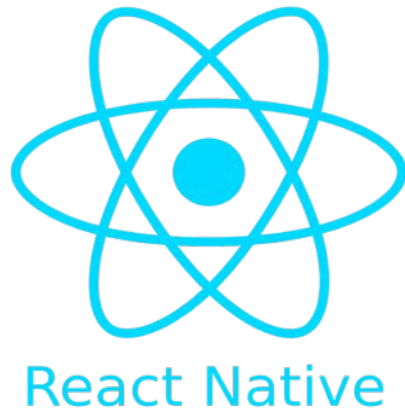
- Express.js — минималистичен и лесен за контрол framework, с който бързо и лесно се изгражда REST API.
- TypeScript — дава типизация и по-малко runtime грешки, което улеснява поддръжката и осигурява стабилност.
- REST API — стандартизиран и широко поддържан подход, който прави интеграцията с React Native предвидима и лесна.
- JWT Access Token — позволява stateless удостоверяване без server-side session, което улеснява scale-a на API-то.
- Refresh Token — осигурява по-добра сигурност и UX (по-кратък access token и възможност за подновяване без повторен login).
- Socket.IO (real-time чат) — дава най-лесна интеграция с Express и React Native и решава типични проблеми като reconnect и event-based комуникация.
- ORM (Prisma или TypeORM) — бърза и лесна работата с PostgreSQL чрез модели и миграции.



---

# Frontend технологии

- React Native — позволява разработка на едно мобилно приложение за Android и iOS с общ codebase и по-бърза итерация в сравнение с native поотделно.
- TypeScript — дава типизация на компонентите/данните от API и намалява грешките при работа.
- React Navigation — стандартното решение за навигация в React Native, което поддържа stack/tab навигации и deep linking при нотификации.
- react-native-maps — предоставя готова интеграция с Google/Apple maps..
- HTTP (Axios / Fetch) — реализира комуникацията с REST API-то (login, feed, заявки, репорти) с ясни интерсептори за токени и обработка на грешки.
- Push нотификации (Twilio) — позволява изпращане на известия за нов match, ново съобщение или update по сигнал без приложението да е отворено.
- State management (Redux) — централизира данните и кеша (feed, профил, репорти) и намалява сложността при синхронизация с бекенда.
- WebSocket — осигурява real-time чат и обновяване на съобщенията без постоянно polling-ване към API.

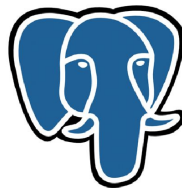


---

# Database + Deployment

## Database

- PostgreSQL – релационна база данни с възможност за голям scale и open-source

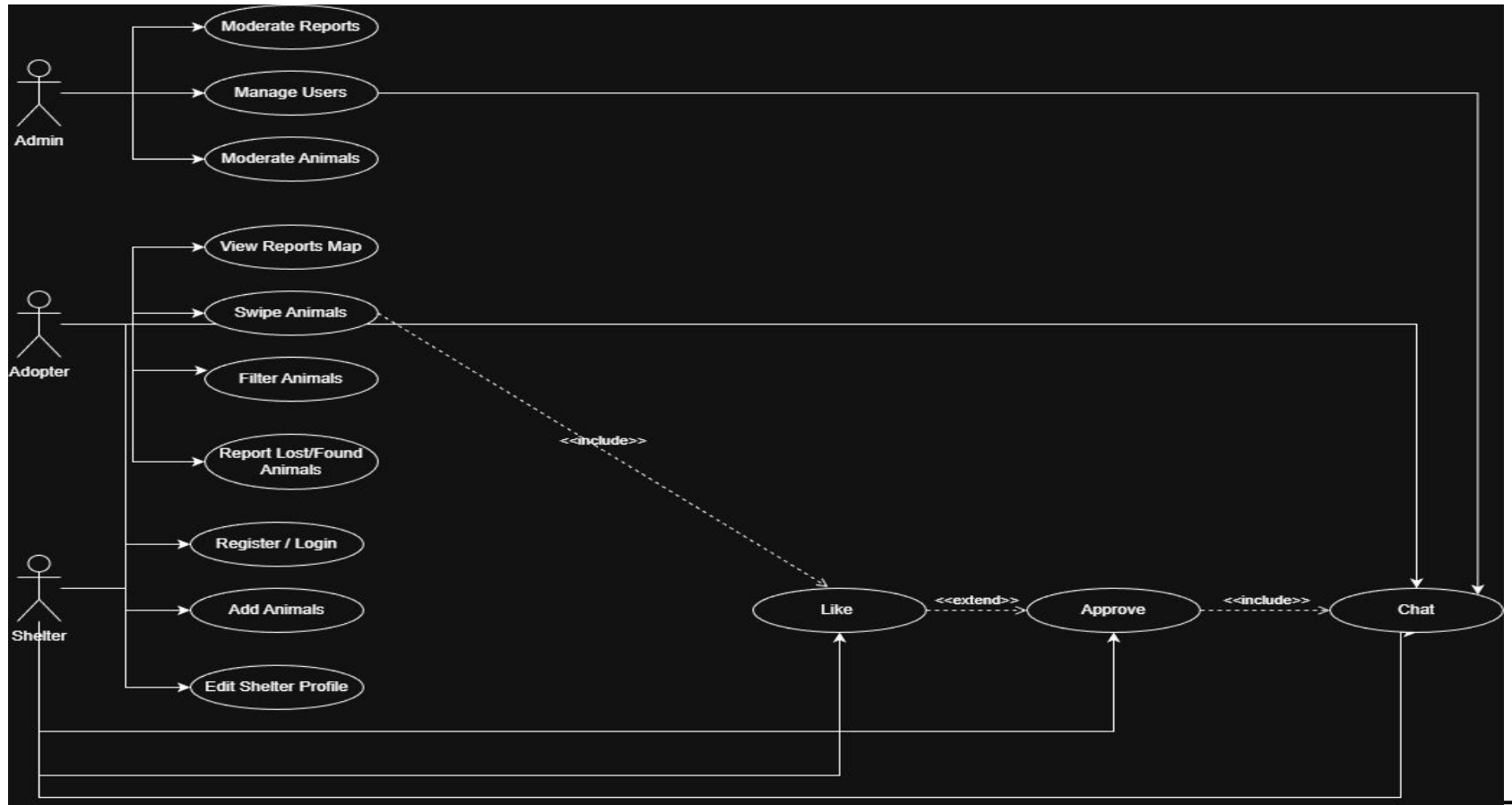


PostgreSQL

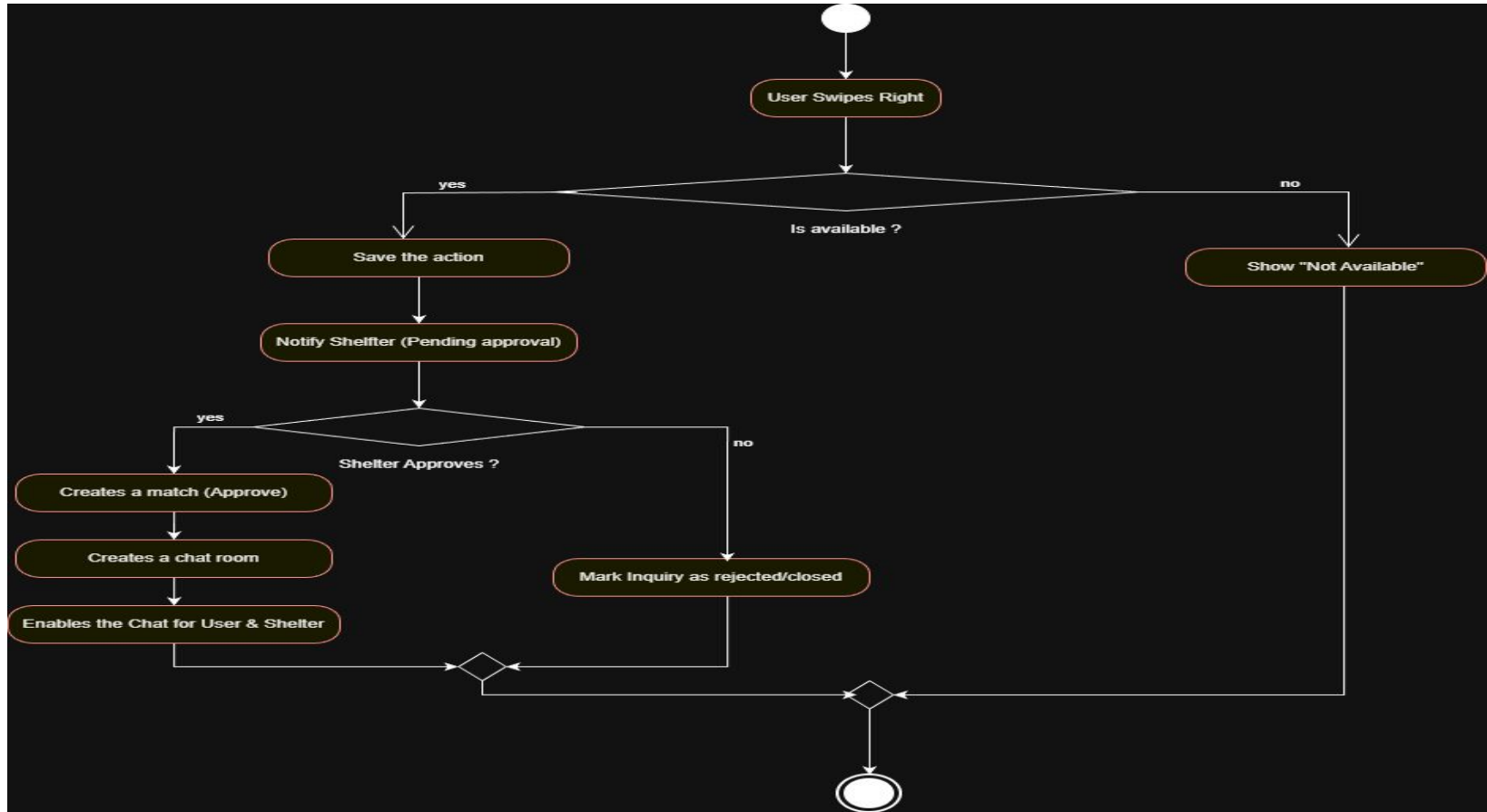
## Hosting / Deploy

- Backend: AWS Elastic Beanstalk (Node.js) – лесен deploy + работи със WebSockets/Socket.IO зад Load Balancer
  - Database: AWS RDS for PostgreSQL – managed PostgreSQL (backups и scaling)
  - Images/Files: AWS S3 – снимки на животни/сигнали
-

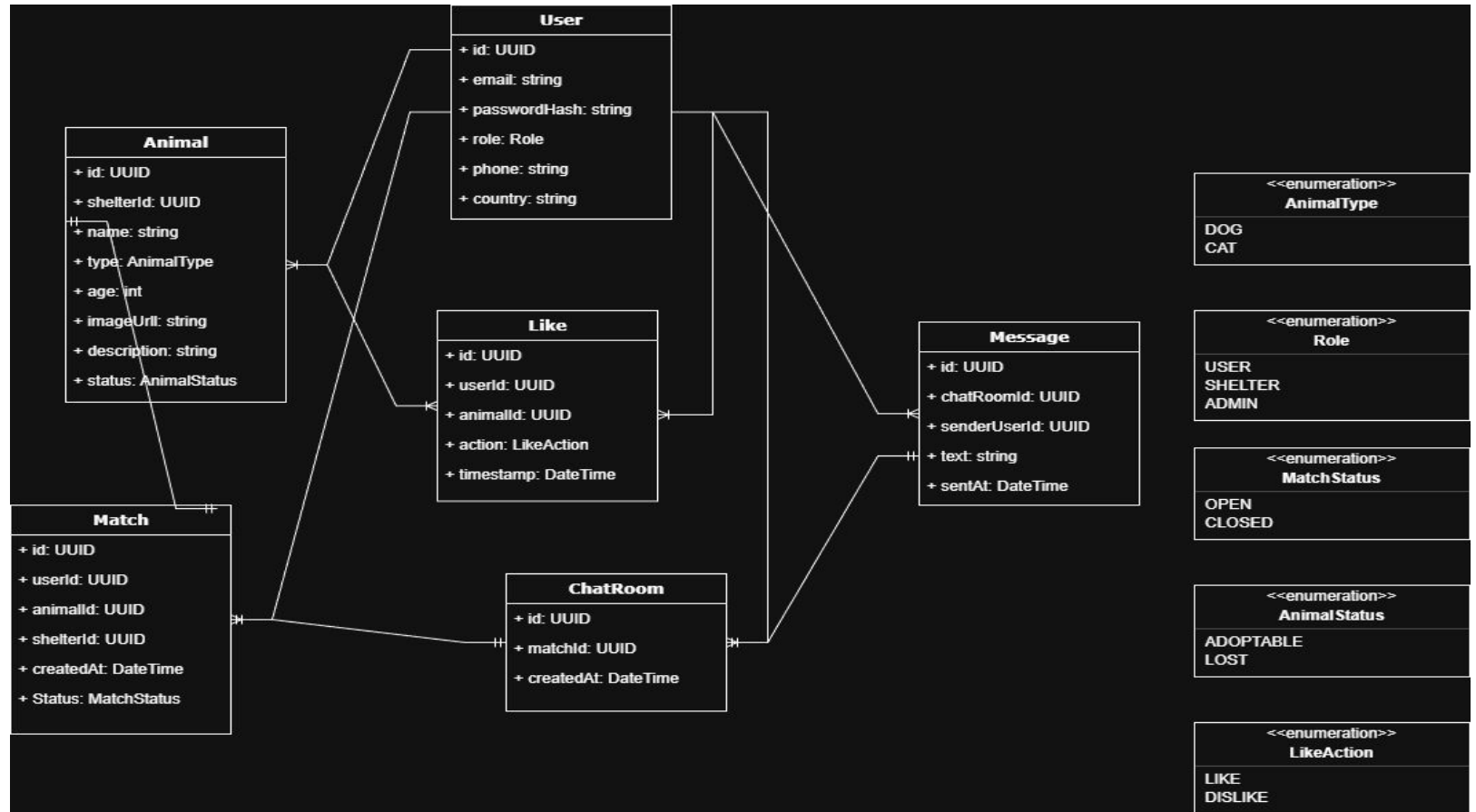
# Use case диаграма



# Activity диаграма



# Class диаграма



# Database cxema

