



**Politechnika
Śląska**

**Wydział Automatyki, Elektroniki
i Informatyki**

Technologie aplikacji internetowych
Projekt
PairPay

Natalia Stręk,
Jakub Kula,
Paweł Wójtowicz

Gliwice 2024

1 Komponenty systemu

Przeanalizuj, jak zrealizować wymagania od strony technicznej. Jakie komponenty składają się na system? Czy będzie to jeden serwer czy kilka? Jakie technologie będą używane?

1.1 Backend - REST API

- Technologie: Python z frameworkiem Flask
- Uzasadnienie: Flask jest oparty o język Python, co umożliwia korzystanie z jego licznych zalet. To lekki framework, który umożliwia szybkie tworzenie aplikacji backendowych. Daje pełną elastyczność w projektowaniu różnych rozwiązań. Jest idealny do tworzenia REST API i umożliwia łatwą integrację między frontendem a backendem. Kolejnym ważnym argumentem jest jego wsparcie dla OAuth, co pozwala zapewnić bezpieczne logowanie i autoryzację użytkowników. Kolejnym ważnym aspektem jest to, że Flask świetnie współpracuje z takimi bibliotekami jak SQLAlchemy czy Tortoise ORM, co znacznie upraszcza operacje na bazie danych i tworzenie powiązań między tabelami. Ogromną zaletą Flaska jest również wsparcie społeczności i rozbudowana dokumentacja, która umożliwi szybkie znalezienie rozwiązań na różne problemy.

1.2 Frontend

- Technologie: TypeScript, React
- Uzasadnienie: React to popularna biblioteka do budowy interfejsów użytkownika, która pozwala na tworzenie dynamicznych i responsywnych aplikacji webowych. Dzięki zastosowaniu komponentów można łatwo zarządzać stanem aplikacji oraz ponownie wykorzystywać kod, co znacząco przyspiesza proces developmentu. React oferuje także wirtualny DOM, co przyczynia się do wydajności aplikacji, umożliwiając szybkie aktualizacje UI bez obciążania przeglądarki. Co więcej dostępne jest duża ilość bibliotek z gotowymi komponentami UI (NextUI, Shadcn, itp) pozwala to na dojść szybki i prosty sposób tworzenia elementów interfejsów użytkownika, dzięki czemu przyspieszy to proces developmentu.

1.3 Baza danych

- Technologie: PostgreSQL
- Uzasadnienie:

1.4 Development

- Technologie: Git (GitHub) oraz CI/CD (GitHub Actions) ??TODO docker-compose i zastanowić się czy to dać
- Uzasadnienie: Umożliwiają monitorowanie zmian w kodzie, co pozwala na łatwe śledzenie postępów w projekcie oraz ułatwia pracę zespołową poprzez wykorzystanie branchy do izolacji pracy nad różnymi funkcjonalnościami aplikacji, co zmniejsza ryzyko konfliktów w kodzie. GitHub Actions, jako narzędzie wbudowane w GitHub, automatyzuje procesy CI/CD bezpośrednio w repozytoriach i oferuje elastyczność w definiowaniu potoków CI/CD w plikach YAML.

1.5 Diagram komponentów

1.6 Diagram sekwencji

2 Użyte technologie w projekcie

1. Python, Flask - Flask jest jednym z czterech największych frameworków backendowych w Pythonie, wyróżniającym się szybkością wdrażania oraz skalowalnością. Jednym z istotnych atutów Flask jest jego rozbudowana i aktywna społeczność, co sprawia, że w razie potrzeby łatwo jest znaleźć pomoc i zasoby wspierające rozwój projektu. Dzięki temu framework ten idealnie nadaje się do tworzenia małych i średnich projektów, umożliwiając szybkie prototypowanie i iterację. Istotnym elementem jest także to, że Flask bazuje na Pythonie – języku programowania znanym ze swojej czytelności oraz obszernego ekosystemu bibliotek i narzędzi. Python, będący jednym z najłatwiejszych do nauki języków, oferuje jednocześnie szerokie możliwości przetwarzania danych, co stanowi istotną zaletę w kontekście projektów wymagających analiz i przetwarzania informacji.
2. BAZA danych
3. FRONT END

3 Bezpieczeństwo

3.1 Zabezpieczenia logowania i autoryzacji Zarządzanie sesjami

Mechanizmy uwierzytelniania: Rozważ użycie protokołów takich jak OAuth 2.0

3.2 Zabezpieczenia bazy danych

Szyfrowanie haseł: Używaj algorytmów takich jak bcrypt do szyfrowania haseł przed ich zapisaniem w bazie danych.

Unikanie SQL Injection: Używaj zapytań z parametryzowaniem lub ORM-ów, aby zapobiec wstrzykiwaniu kodu SQL.

3.3 Zarządzanie dostępem