

# Platforma do łączenia programistów w zespoły projektowe

- Daniel Chorągwicki - choragwickid@gmail.com
- Maciej Garncarz - maciek.g95@interia.pl

## Opis projektu

Główną funkcją aplikacji, będzie umożliwienie w łatwy i przejrzysty sposób tworzenie projektów i poszukiwanie osób zainteresowanych współpracą. Odbiorcami usługi zostaną osoby mocno powiązane z szeroko pojętym programowaniem m.in. aplikacji, gier, serwisów tj.: programiści, testerzy, graficy, project managerzy a także osoby, mające tylko pomysł.

Atutem aplikacji będzie rozbudowany system wyszukiwania interesującej treści:

- Dla osoby poszukującej: projektów i wolnych wakatów
- Dla osoby inicjującej: osób zainteresowanych uczestniczeniem w projekcie

Wyszukiwanie będzie odbywać się za pomocą filtrów, przykładowo: technologie, lata doświadczenia, poziom umiejętności, wiek, liczbę członków itd.

Każdy użytkownik, będzie miał możliwość stworzenia profilu (osoby poszukującej czy też projektu) w którym zawrze konkretne informacje dotyczące swoich umiejętności, doświadczenia i swoich oczekiwań.

Osoby połączone w jeden team, dostaną dostęp do panelu projektu w którym będą miały dostęp do różnych funkcji do zarządzania zespołem.

Dodatkowo, platforma będzie stopniowo rozbudowywana o funkcjonalności pozwalające użytkownikom (szczególnie inicjatorom) posiadanie większej kontroli nad projektem, a także integrujące zewnętrzne usługi.

# Model biznesowy

Platforma będzie skierowana między innymi do:

## Inicjatorów

- Studentów, szukających prostego narzędzia wyszukania partnerów do projektu uczelnianego oraz jego organizacji
- Startupów szukających specjalistów z różnych dziedzin programistycznych i około programistycznych
- Niekomercyjnych projektów np. open source
- Osób, które mają pomysł na własną aplikację, serwis, grę czy jakikolwiek inny projekt związany z programowaniem bez zasobów ludzkich
- Mikro przedsiębiorców poszukujących freelancerów i podwykonawców
- *Dużych korporacji mających problem z organizacją i doбором pracowników w swoich projektach oraz teamach\**

## Poszukujących

- Specjalistów z różnych dziedzin programowania
- Junior developerów oraz studentów szukających okazji do zdobycia pierwszego doświadczenia
- Osób powiązanych z projektami programistycznymi, np. graficy, project managerzy czy testerzy
- Freelancerów szukających możliwości zarobku

*Z platformy będzie można korzystać za darmo, lecz będą dostępne funkcje dostępne tylko dla użytkowników premium. (dodanie po skończeniu pierwszej wersji aplikacji)*

**Aspekt spieniężenia** planujemy podzielić na 3 elementy:

### **1. Konto premium dla użytkownika poszukującego**

*Tanie, mające wymiar bardziej wsparcia platformy. Funkcje:*

- *Możliwość aplikowania do projektu niemające otwartej rekrutacji – Wyróżniona pozycja w wyszukiwarce*
- *Nieograniczona liczba aplikacji do projektów jednocześnie*
- *Motywy premium profilu*
- *Zaawansowane filtry wyszukiwania*

### **2. Konto premium dla użytkownika inicjującego projekt**

*Droższe, mające przełożenie na efektywność prowadzenia projektu. Funkcje:*

- *Wyróżnione w sliderze na stronie głównej*

- Wyróżniona pozycja w wyszukiwarce projektów/wakatów – Nieograniczona liczba członków zespołu
- Nieograniczona liczba otwartych rekrutacji
- Możliwość stworzenia więcej niż jednego projektu
- Motywy premium dla profilu oraz panelu
- Zaawansowane filtry wyszukiwania
- Ukryta tożsamość członków
- Moduły premium w panelu projektu\*

### **3. Zupełnie osobna, zamknięta instancja aplikacji dostosowana pod konkretną firmę\***

*Aplikacja spersonalizowana pod daną firmę, jednak przy zachowaniu idei.*

*\*W celu zatrzymania użytkowników na dłuższy okres czasu, planujemy stopniowo rozszerzać aplikację o nowe funkcjonalności, dodające możliwości większej kontroli, zarządzania i organizacji projektem poprzez m.in. integrację z zewnętrznymi serwisami. Aplikacja byłaby swoistym „centrum dowodzenia”.*

## **Analiza rynku**

Na rynku nie ma portali specjalizujących się w dobieraniu programistów oraz łączenia ich w projekty ze względu na specjalizacje i umiejętności. Nie należy mylić portalu ze zwykłym portalem do szukania pracy oraz wystawiania ofert np. Pracuj.pl, jest jednak kilka portali które są podobne i należy na nie zwrócić uwagę.

### **1. Honeypot.io**

Platforma dzięki której możemy wystawić naszą firmę i szukać specjalistów, działa to również w drugą stronę, tak samo my możemy założyć profil i rozpocząć poszukiwanie pracy. Strona ma piękną oprawę graficzną oraz łatwość w użytkowaniu. Maskotka strony przykuwa uwagę oraz pozwala zapamiętać stronę na dłużej. Wadami portalu jest uciążliwy proces rekrutacji oraz rejestrowania się oraz zasięg portalu jest nastawiony wyłącznie na Europe.

### **2. zlott.io**

Platforma dzięki której w szybki sposób połączymy pracodawcę oraz potencjalnego pracownika. Strona ma nieprzeciętny design, jednak rejestracja wymaga czasu oraz jest bardzo wolna. Strona działa w obrębie stanów zjednoczonych oraz nie obsługuje freelancerów.

Przewagą naszego portalu będzie łatwa, szybka i przyjemna rejestracja. Łatwość nawiązywania kontaktu oraz wyszukiwania specjalistów lub ciekawych projektów. Piękny i przyjemny UI oraz najwyższej klasy UX.

## **Aplikacja i funkcjonalności**

- **Intuicyjny interfejs**
  - Wersja mobilna
- **Precyzyjny i przyjazny system rejestracji**
  - Użytkownik:
    - Dane podstawowe - imię i nazwisko, dane kontaktowe
    - Dane specyficzne - umiejętności, specyfikacja poszukiwań
    - Podsumowanie danych i akceptacja
  - Właściciel projektu:
    - Dane podstawowe - nazwa projektu, dane kontaktowe
    - Dane specyficzne - opis projektu/projektów, pożądane umiejętności w projekcie, wakaty
    - Podsumowanie danych i akceptacja
- **System powiadomień mailowych**
  - powiadamianie użytkownika w zależności od ich preferencji i ustawień:
    - przyjęcie do teamu/projektu
    - zaproszenie
    - rezygnacja
    - interesujące oferty
    - potencjalni użytkownicy spełniający wymagania
- **Rozbudowany system filtrów**
  - po wpisaniu nazwy
  - po wybraniu pożądanych umiejętności
  - po wpisaniu doświadczenia, wieku
- **System rekrutacji i system statusów**
  - wysyłanie aplikacji
  - akceptacja użytkownika do teamu
  - odrzucenie propozycji
  - oczekiwanie
- **System propozycji**
  - proponowanie użytkownikom projektów na podstawie umiejętności
  - proponowanie właścicielom projektów użytkowników na podstawie wymagań
- **Komunikator**
  - porozumiewanie się wewnątrz teamu/projektu

***Funkcjonalności mniejszego priorytetu, dodawane po skończeniu pierwszej wersji aplikacji:***

- *Konta premium - spieniężenie projektu*
- *System modułów wewnątrz panelu projektu/teamu – dzięki temu, właściciel projektu będzie w stanie dowolnie rozszerzyć panel o dodatkowe funkcje takie jak: kalendarz, listy to do, integracje z zewnętrznymi platformami. (przykład podobnego rozwiązania: Power-Up Trello).*

#### **Widoki w aplikacji:**

- Landing Page
  - Homepage
  - About
  - Contact
- Rejestracja użytkownika
  - Krok 1: dane podstawowe
  - Krok 2: dane specyficzne dotyczące użytkownika
  - Krok 3: podsumowanie
- Rejestracja teamu/projektu
  - Krok 1: dane podstawowe
  - Krok 2: dane specyficzne dotyczące teamu/projektu
  - Krok 3: podsumowanie
- Logowanie
- Filtrowana lista teamów/projektów
- Filtrowana lista użytkowników
- Filtrowana lista wolnych wakatów
- Publiczny profil użytkownika
- Publiczny profil teamu/projektu
- Dashboard teamu/projektu
- Panel zarządzania projektem/teamem dla administratora
  - Panel rekrutacji
- Panel edycji profilu użytkownika
- Panel ustawień aplikacji

#### **Komponenty layoutu aplikacji:**

- A. Loader
- B. Header
  - a. Navbar landing page
  - b. Navbar użytkownika niezalogowanego
  - c. Navbar użytkownika zalogowanego
- C. Footer
  - a. Lang Switcher (Polski/Angielski)
- D. Formularz
  - a. Rejestracyjny
  - b. Logowania
  - c. Edycji
  - d. Form slider
- E. Moduł filtrowania

- F. Moduł listy
  - a. Element
- G. Dashboard
- H. Content Text
- I. Moduł ustawień
- J. Moduł profilu
  - a. Elementy dla zalogowanych
  - b. Elementy dla niezalogowanych
  - c. Elementy dla zarządzających

## Epics

### **E1 – Layout**

All stories involving browsing, UX, and UI design.

### **E2 – Tech**

All stories involving dev environment.

### **E3 – Dev users.**

All stories involving dev users that want to participate in projects.

### **E4 – Project owner users.**

All stories involving project owners.

### **E5 – Filters.**

All stories involving usage of filters.

### **E6 – Communicator.**

All stories about communicator.

### **E7 – Notification system.**

All stories about notification system.

## User stories

### **[Layout] As\_A\_Developer\_I\_Want\_To\_Have\_Flexible\_Grid\_System**

Implementing grid system that is:

ACC:

1. Mobile friendly
2. Medium device's friendly.
3. Large device's friendly.

### **[Dev users] [Project owner users] As\_An\_User\_I\_Want\_To\_Register**

ACC:

1. Sign-up with google account.
2. Sign-up with facebook account.
3. Sign-up through sign-up page.

**[Dev users] As\_An\_User\_I\_Want\_To\_Search\_For\_Projects**

ACC:

1. User can type in search to find projects.
2. User can use filter system to find projects.

**[Filters] As\_An\_Application\_I\_Want\_To\_Have\_Filters\_System**

Filters that contains:

ACC:

1. Technologies
2. Activities
3. Place of project
4. Salary
5. Benefits
6. Working hours
7. Type of employment

**[Dev users] As\_An\_User\_I\_Want\_To\_Apply\_To\_Project**

ACC:

1. User can see detailed info about project.
2. User can apply to project through button click.

**[Dev users][ Notification system] As\_An\_User\_I\_Want\_To\_Get\_Notifications**

ACC:

1. User gets notifications about any change's made to project.
2. User can turn on/turn off notifications.
3. User can set preferences about notifications.
4. User gets notifications about interesting projects based on he's preferences.

**[Communicator system] As\_An\_Application\_I\_Want\_To\_Have\_Communicator**

Communicator that will allow to:

ACC:

1. Communicate between dev and project owner.
2. Communicate between project members.
3. Keep all messages in database.

**[Project owner users] As\_An\_User\_I\_Want\_To\_Create\_Project**

ACC:

1. Create project.
2. Specify project details
3. Activity's
4. Salary
5. Type of project
6. Type of employment
7. Name
8. Description

**[Project owner users] As\_An\_User\_I\_Want\_To\_Manage\_Devs\_In\_Project**

ACC:

1. Accept dev to project.
2. Refuse dev to project.
3. Kick out dev from project.

**[Tech] As\_An\_Developer\_I\_Want\_To\_Have\_System\_That\_Marks\_Projects\_Obsolete**

System will mark projects obsolete in case of:

ACC:

1. Project is old
2. No activity
3. No developer requests after some period of time.
4. Project is closed.

**[Tech] As\_An\_Developer\_I\_Want\_To\_Servers\_Setup**

ACC:

1. Set up Testing server.
2. Set up Production server.
3. Database backups.
4. Easy-to-deploy containers.
5. Tests being fired before every pull request.

**[Tech] As\_An\_Developer\_I\_Want\_To\_Have\_Branching\_Model\_Setup**

Set up branches:

ACC:

1. Production
2. Develop
3. Feature
4. Bugfix
5. Hotfix



## 6. Release candidates

### Back-end

- **ASP.NET Core** – podstawa architektury aplikacji w oparciu o architekturę mikroservisów przy użyciu docker'ów, logika biznesowa oraz wystawienie API.
- **ORM** – Entity framework do modelowania bazy przy podejściu code first oraz użyciu migracji, dodatkowo Dapper który za zadanie ma pomóc udźwignąć „cięższe” zapytania.
- **MSSQL** – system zarządzania bazą danych.
- **RabbitMQ** – System do kolejkowania zadań głównie wykorzystywany do powiadomień.

### Front-end

- **Javascript oraz biblioteka React** – podstawowy sposób tworzenia widoków i wyświetlania danych z back-endu. Utrzymanie aplikacji w rozwiązaniu single page application (SPA).
- **Redux** – wykorzystywany do zarządzania globalnym stanem aplikacji.
- **RxJS** – obsługa danych asynchronicznych.
- **JWT** – obsługa sesji użytkownika.
- **Webpack** – podstawowe narzędzie do pracy z modułami aplikacji.
- **Jest** – testowanie aplikacji po stronie front-endu.

### Wzorce projektowe

**Inversion of control and Dependency Injection** – zastosowanie kontenerów IoC oraz wstrzykiwanie zależności usprawni aplikacji oraz pomoże rozwiązać problem „mocnego powiązania”.

**Strategia** – Pozwala zdefiniować rodzinę klas oraz algorytmów i korzystać z nich wymiennie.

**Fabryka abstrakcji** – Tworzy klasy związane kontraktem bez podawania określonej klasy.

**Dekorator** – Pozwala rozszerzyć klasę bez bezpośredniej modyfikacji dzięki opakowywaniu klasy.