*Specyfikacja funkcjonalna z projektu “****SmartMarketplace****”*

*Przedmiot: projekt zespołowy*

*Koordynator przedmiotu: prof. dr hab. inż. Andrzej Dzieliński*

*Promotor: Bartosz Chaber*

*Uczestnicy zespołu: Filip Olejniczak, Volha Karnialiuk, Yevhenii Vasylyshyn, Eduard Ryzheuski, Vladyslav Podlesnyi, Ilya Kharashun, Aliaksei Selivonets*

**Spis treści:**

1. Opis ogólny.  
   1.1 Nazwa programu  
   1.2 Poruszany problem  
   1.3 Użytkownik docelowy  
   1.4 Lokalizacja  
   1.5 Niezbędne definicje
2. Opis funkcjonalności.  
   2.1 Jak korzystać z aplikacji?  
   2.2 Potwierdzenie konta.  
   2.3 Funkcjonalne ryzyka.
3. Główne technologie.  
   3.1 Truffle Framework  
   3.2 Solidity  
   3.3 “Web3.js” Ethereum JavaScript API  
   3.4 React
4. Interfejs aplikacji.  
   4.1 Strona główna  
   4.2 Strona rejestracji  
   4.3 Strona użytkownika  
   4.4 Schemat aplikacji
5. **Opis ogólny**

**1.1 Nazwa programu**

SmartMarketplace to internetowa platforma handlowa, która umożliwia zawieranie transakcji. Umożliwia wystawianie na sprzedaż przedmiotów należących do użytkowników. Można wystawić przedmiot na sprzedaż po z góry określonej cenie.

**1.2 Poruszany problem**

Celem programu jest stworzenie aplikacji internetowej, która pozwala stwarzać smart kontrakty, który pozwalają użytkownikom zautomatyzować ich działania na platformie blockchain. Biorąc pod uwagę znaczenie smart kontraktów we wspieraniu ważnych działań w różnych sektorach przemysłu, łańcuch dostaw, finansów, usług prawnych i medycznych, istnieje duże zapotrzebowanie na techniki weryfikacji i walidacji. Aktywy mogą być materialne (dom, samochód, pieniądze, ziemia) i niematerialne (własność intelektualna, patenty, prawa autorskie, marka).

Wady i zalety korzystania z smart kontraktów:

|  |  |
| --- | --- |
| Zalety | Wady |
| Szybsze transakcje i dostęp 24/7. | Dane nie są całkowicie niewrażliwe na ataki. |
| Wysoce bezpieczny, ponieważ jest to rozproszona księga, która tworzy wiele kopii bazy danych w sieci. | Zużycie energii elektrycznej może być szkodliwe dla środowiska. |
| Brak osób trzecich lub pośredników. | Brak centralnego pośrednika uniemożliwia odzyskanie konta w przypadku utraty klucza prywatnego. |

Istnieje wiele korzyści z korzystania ze smart kontraktów. Jego wydajność, bezpieczeństwo i brak pośredników mogą sprawić, że będzie to idealna opcja dla tych, którzy chcą bezpiecznie przechowywać szereg danych.

Ponieważ jednak nie ma scentralizowanej strony do zarządzania bazą danych, nigdy nie można odzyskać żadnych zasobów, jeśli jest utracony klucz prywatny. Ważne jest, aby o tym pamiętać podczas korzystania z transakcji opartych na blockchain.

**1.3 Użytkownik docelowy**

Aplikacja jest przeznaczona dla użytkowników, które są podzieleni na dwie grupy:

1) sprzedający - użytkownik podejmujący działania zmierzające do zbycia lub zbywający Towary w ramach oferty.

2) kupujący - użytkownik podejmujący działania zmierzające do nabycia lub nabywający Towary w ramach SmartMarketplace.

**1.4 Lokalizacja**

Serwis będzie dostępny dla każdego użytkownika z połączeniem Internet z dowolnej części świata.

**1.5 Niezbędne definicje**

|  |  |
| --- | --- |
| Pojęcie | Definicja |
| Rejestracja | procedura zakładania konta. |
| Konto | prowadzony dla użytkownika przez SmartMarketplace pod unikalną nazwą zbiór zasobów, w którym gromadzone są dane użytkownika oraz informacje o jego działaniach w ramach SmartMarketplace. |
| Towar | rzecz, usługa lub prawo będące przedmiotem oferty. |
| Oferta | propozycja zawarcia umowy sprzedaży towaru na warunkach określonych przez sprzedającego, w szczególności zawierających cenę i opis oferowanego towaru, przy wykorzystaniu i za pomocą funkcjonalności dostępnych w ramach SmartMarketplace. |
| Transakcja | procedury zawierania i wykonywania umów sprzedaży towarów pomiędzy użytkownikami w ramach SmartMarketplace. |
| Dyskusja | toczony między sprzedającym a kupującym spór dotyczący prawidłowości przeprowadzenia transakcji - odbywający się w ramach udostępnionych przez SmartMarketplace narzędzi wsparcia pozakupowego. |

1. **Opis funkcjonalności**

**2.1 Jak korzystać z aplikacji?**

Po wejściu na stronę internetową użytkownik może zapoznać się z pełną ofertą dostępną w SmartMarketplace. Nowi użytkownicy mogą się również zarejestrować i zalogować na konto (jeżeli użytkownik już posiada konto).

Podczas rejestracji użytkownik musi wprowadzić następujące pola obowiązkowe:

1) Imię i nazwisko;

2) Unikalny login;

3) Poczta elektroniczna.

4) Hasło (entropia hasła będzie brana pod uwagę w celu sprawdzenia jego siły i zabezpieczenia przed intruzami).

Użytkownik musi także określić, kim chce być: sprzedającym lub kupującym.

Jeśli użytkownik chce zalogować się na swoje konto, musi wprowadzić swój unikalny login i aktualne hasło.

Po zalogowaniu się na konto kupujący może dodać produkty od sprzedawców do koszyka, wybierając rozmiar/kolor/itd (jeśli są dostępne u sprzedawcy) i zapłacić za swoje zakupy.

**2.2 Potwierdzenie konta.**

Od razu po wprowadzeniu danych i pierwotnej akceptacji, na podany przez użytkownika adres e-mail zostanie wysłany link od oficjalnego konta “SmartMarketplace” link-podtwierdzenie.

**2.3 Funkcjonalne ryzyka.**

1. Użytkownik zapomniał hasła lub zmienił numer telefonu: użytkownik musi skontaktować się z administratorem;

2. Zalogowany Użytkownik będzie mógł uzyskać dostęp do zastrzeżonych części witryny, zmieniając fundamentalne zmiany witryny lub wprowadzać złośliwe oprogramowanie, co prowadzi do wprowadzenia w błąd.

3. Odwiedzający witrynę (niezalogowany użytkownik), będzie mógł uzyskać dostęp do sekcji witryny zarezerwowanych dla zalogowanych użytkowników, zmniejszając wiarygodność serwisu.

1. **Główne technologie.**

Dla stworzenia naszego produktu korzystamy z następnych narzędzi: “Truffle Framework”, “Solidity”, “React”, “Web3.js”.

**3.1 Truffle Framework.**

Truffle jest środowiskiem programistycznym i platformą testową dla Etherium. Bardzo pasuje dla naszego celu, ponieważ ma następne korzyści:

* Wbudowana inteligentna kompilacja kontraktów, łączenie, wdrażanie i zarządzanie binarne.
* Zautomatyzowane testy kontraktowe z Mocha i Chai.
* Konfigurowalny potok kompilacji z obsługą niestandardowych procesów kompilacji.
* Skryptowalna struktura wdrażania i migracji.
* Zarządzanie siecią do wdrażania w wielu sieciach publicznych i prywatnych.
* Interaktywna konsola do bezpośredniej komunikacji kontraktowej.
* Błyskawiczna odbudowa aktywów w trakcie rozwoju.
* Zewnętrzny program uruchamiający skrypty, który wykonuje skrypty w środowisku Truffle.

**3.2 Solidity.**

Solidity bardzo pasuje do naszych potrzeb, ponieważ jest zorientowanym obiektowo językiem programowania, stworzonym do wdrażania smart kontraktów, tak samo przeznaczony do obsługi wirtualnej maszyny Ethereum (EVM).

# **3.3 “Web3.js” Ethereum JavaScript API**

Web3.js jest zbiorem bibliotek, które umożliwiają interakcję z lokalnym lub zdalnym węzłem Ethereum przy użyciu połączenia HTTP lub IPC.

**3.4 React.**

React jest najpopularniejszą front-endową biblioteką JavaScript, która służy do tworzenia interfejsów użytkownika specjalnie dla aplikacji jednostronicowych. React pozwala nam również na tworzenie komponentów UI wielokrotnego użytku. Główną korzyścią jest to, że dzięki React’u, na naszej stronie mogą zmieniać się dane bez przeładowywania strony.

1. **Interfejs aplikacji**

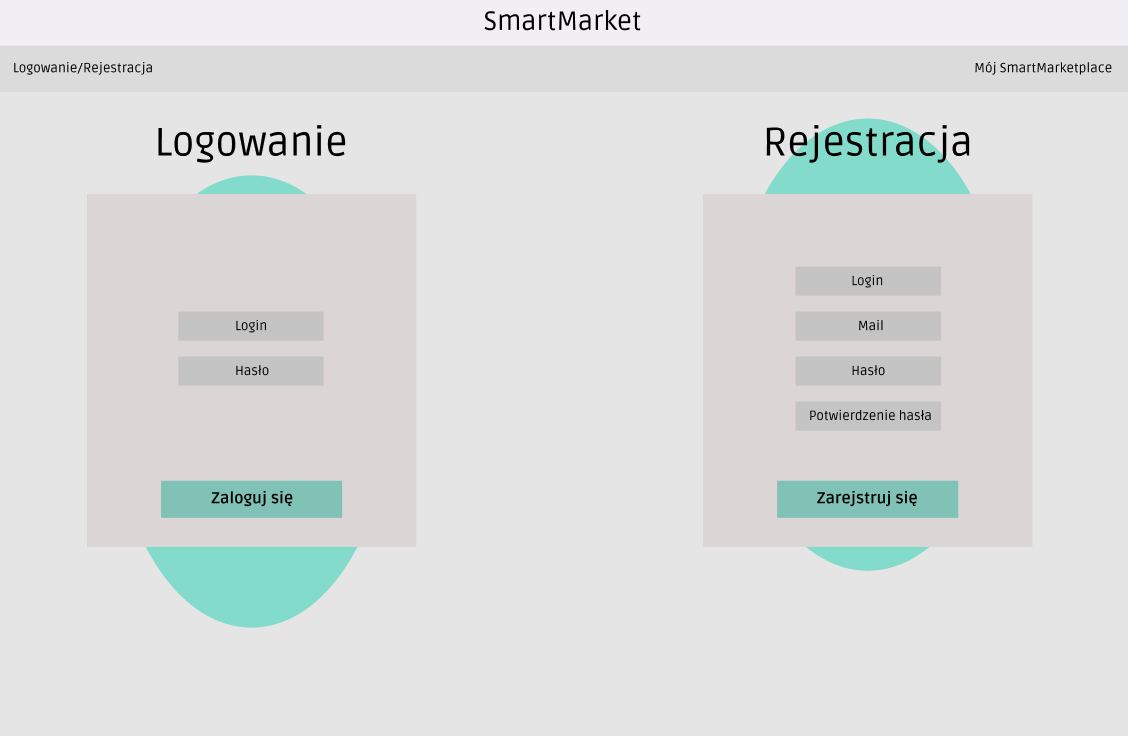
**4.1 Strona główna**

Strona główna zawiera nazwę aplikacji, przycisk do logowania albo rejestracji użytkownika, przycisk do przejścia na stronę użytkownika, kategorie główne, możliwość szukania, przyciski z kontem Instagram, Twitter i Facebook.



**4.2 Strona rejestracji**

Strona rejestracji zawiera nazwę aplikacji, przycisk do logowania albo rejestracji użytkownika, przycisk do przejścia na stronę użytkownika, pole do logowania użytkownika do aplikacji, pole do rejestracji użytkownika do aplikacji.



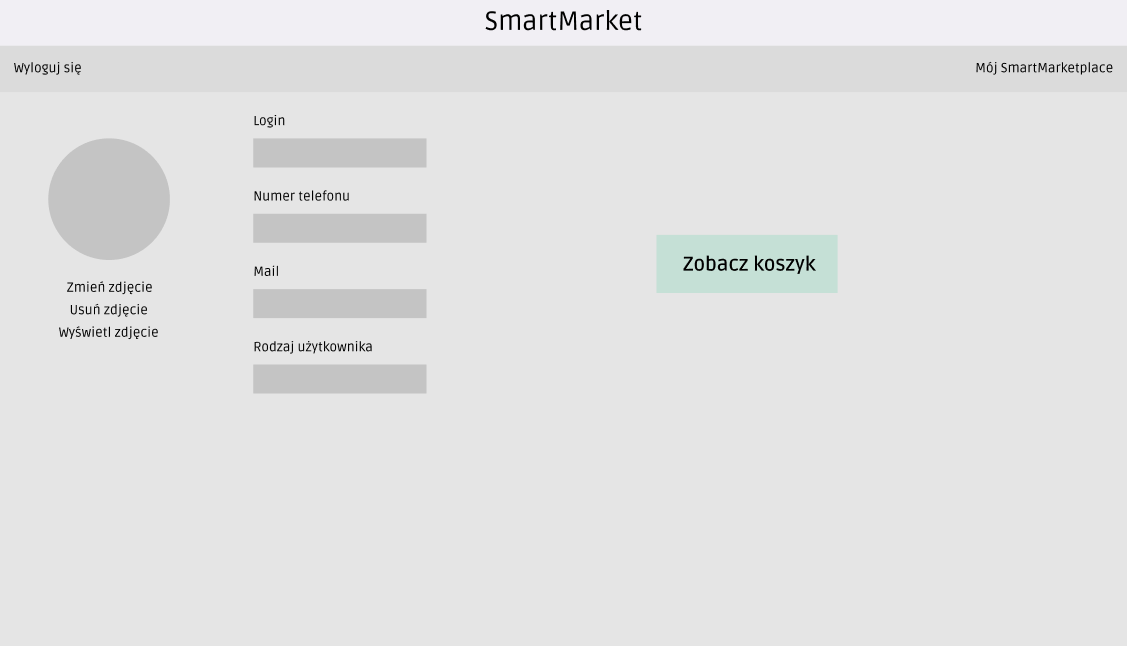
**4.3 Strona użytkownika**

Strona użytkownika zawiera nazwę aplikacji, przycisk wylogowania użytkownika, przycisk do przejścia na stronę użytkownika, aktualne login użytkownika, numer telefonu, mail, rodzaj użytkownika, możliwość sprawdzenia aktualnego koszyka użytkownika z zakupami, zdjęcie wraz z przyciskami, za pomocą których można:

1) zmienić zdjęcie;

2) usunąć zdjęcie;

3) wyświetlić zdjęcie.



**4.4 Schemat aplikacji.**

