

Bazy danych

Rowery miejskie - dokumentacja

Jarosław Dzikowski 273233

Wrocław, 10 czerwca 2016

1 Zawartość dokumentacji

Dokument składa się z trzech części:

1. Tej części ostatecznego modelu konceptualnego, która została zaimplementowana w aplikacji.
2. Instrukcja instalacji aplikacji.
3. Dokumentacja aplikacji.

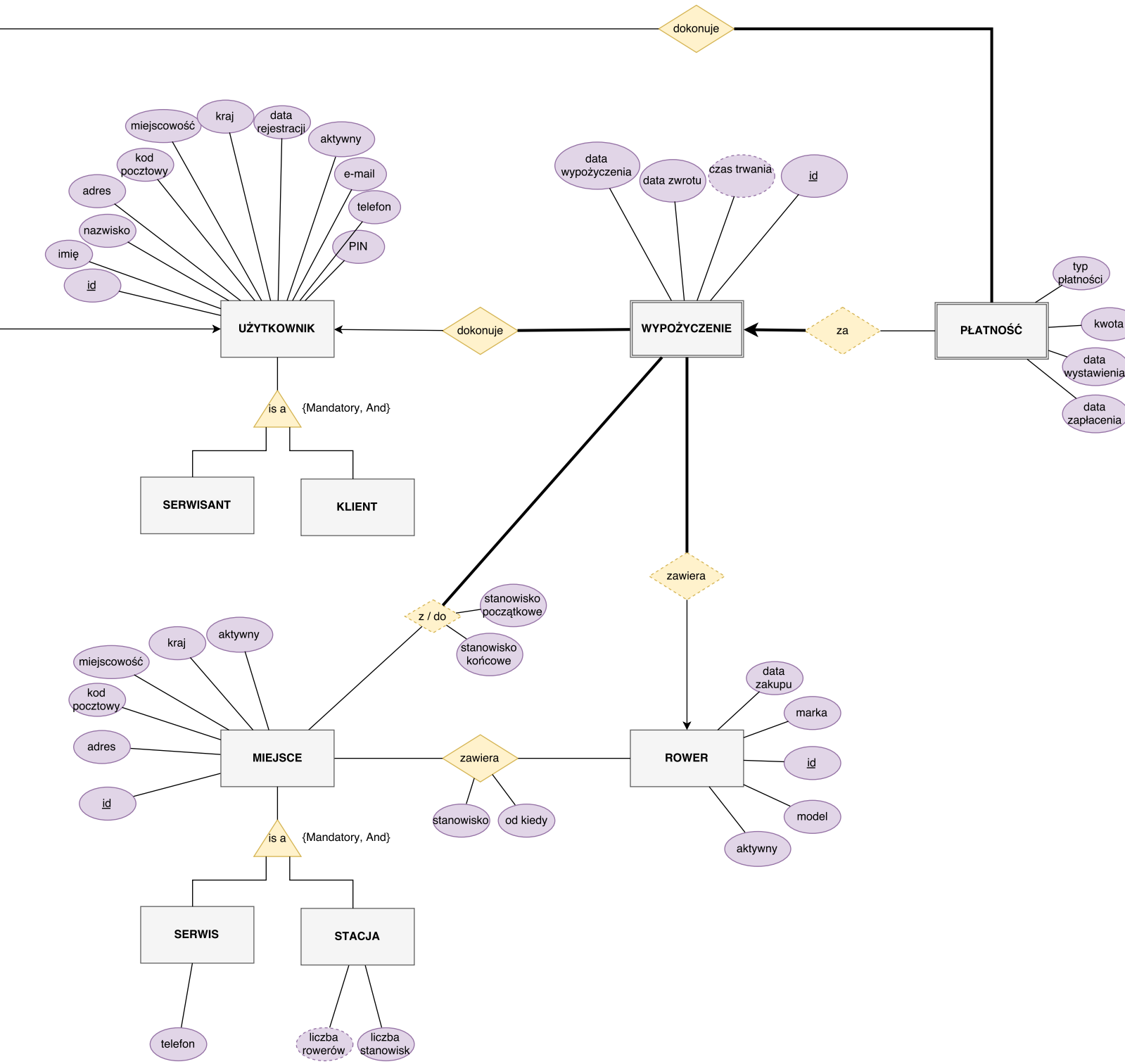
2 Ostateczny model konceptualny

Diagram E-R został zamieszczony na osobnej stronie ze względu na swoją wielkość.

2.1 Komentarz

Na diagramie widnieje dziewięć encji. Cztery z nich są oczywiste z punktu widzenia zagadnienia rowerów miejskich. Mamy użytkowników, rowery, miejsca oraz wypożyczenia. Użytkownicy mogą być jednym z dwóch typów: klientem lub serwisantem. Mimo że na diagramie użytkownik ma atrybut PIN, to tak naprawdę nie będziemy tego hasła pamiętać. Więcej o tym pod koniec tej sekcji. Tak samo miejsca mogą być jednym z dwóch typów: stacją lub serwisem. Oczywiście stacje i serwisy mają swoje własne charakterystyczne cechy takie jak liczba stanowisk na stacji bądź telefon do serwisu.

Użytkownicy dokonują wypożyczeń, które obejmują rowery (1 rower na wypożyczenie) oraz 2 miejsca (początkowe i końcowe) a także stanowiska początkowe oraz końcowe. Wypożyczenie obejmuje dwie sytuacje: wypożyczenie roweru przez klienta oraz przewiezienie roweru przez serwisanta. Miejsca docelowe i końcowe mogą być albo serwisem albo stacją, klient oczywiście będzie wypożyczał i zwracał rowery ze stacji, natomiast serwisant nie ma takiego ograniczenia. Gdy



jednym z miejsc wypożyczenia jest serwis, to stanowisko nie ma wówczas żadnego znaczenia (W serwisach nie ma stanowisk z tego, co wiem).

Rowery znajdują się albo na stacjach, albo są wypożyczone, albo znajdują się w serwisie (Na diagramie relacja Rower-Miejsce). Za wypożyczenia naliczane są płatności, które zostały przedstawione jako kolejna encja. Księgowy zajmujący się płatnościami może nakładać płatności związane z trzema źródłami: wypożyczeniem, rejestracją, karą. Wypożyczenie w płatności może być nullem, jeśli płatność dotyczy rejestracji.

Uwaga: W modelu fizycznym jest więcej tabel, widoków etc niż widać na tym diagramie. Nie zostały one umieszczone, ponieważ nie są one ważne z punktu widzenia umodelowania zagadnienia rowerów miejskich. Przykładem mógłby być sposób uwierzytelniania użytkowników. Loginem jest telefon, PIN jest hasłem, lecz w bazie danych pamiętamy tak naprawdę losowy salt oraz hash skonstruowanego salt'a z PINem podanym przez użytkownika.

Nieopisane więzy są całkiem oczywiste, nie znalazłem żadnych bardziej skomplikowanych więzów:

1. Liczba stanowisk na stacji musi być nieujemna.
2. Rower może być tylko na nieujemnym stanowisku (Na stacji).
3. Stanowiska z których wypożyczono oraz do których zwrócono rowery muszą być albo NULL albo nieujemne.
4. Czas zwrotu roweru musi być nie wcześniejsze niż czas wypożyczenia roweru.
5. Kwota naliczona w płatności musi być nieujemna.
6. Data pokrycia płatności musi być nie wcześniejsza niż data jej wystawienia.

2.2 Opis ról

2.2.1 Rola klienta

1. Może dodawać nowe wypożyczenia i aktualizować swoje wypożyczenie.
2. Prawo do wglądu do historii wypożyczeń.
3. Prawo do wglądu do historii płatności.
4. Może zobaczyć, gdzie są stacje.

2.2.2 Rola serwisanta

1. Może przewozić rowery (Między stacjami i serwisami).
2. Może sprawdzać, gdzie są rowery.
3. Może sprawdzać, ile rowerów jest na której stacji.
4. Może przeglądać katalog rowerów oraz dodawać nowe.
5. Może przeglądać katalog stacji oraz serwisów oraz dodawać nowe.

2.2.3 Rola księgowego

1. Ma prawo do modyfikacji oraz odczytu danych płatności.
2. Ma prawo do odczytu historii wypożyczeń.
3. Ma prawo do odczytu danych użytkowników i aktywowania/dezaktywowania użytkowników.
4. Nakłada płatności na użytkowników.
5. Znajduje wypożyczenia z brakiem zwrotu roweru i nakłada kary na użytkowników wypożyczających.

3 Instalacja aplikacji

Docelowo instalacja aplikacji jest przeznaczona na linuxy, konkretnie opisana zostanie procedura instalacji na Ubuntu. W przypadku dystrybucji nie będących pochodną Ubuntu proces instalacji zależności może wyglądać inaczej.

3.1 Wymagania

Do instalacji aplikacji potrzebne są następujące zależności:

1. openjdk-7-jdk
2. postgresql - przynajmniej wersja 9.3 (Najlepiej wersja 9.4)
3. postgresql-contrib (Zawiera moduł pgcrypto używany przy uwierzytelnianiu)
4. maven2 (Manager zależności projektu)
5. tomcat8 albo tomcat7 (Serwer)

3.2 Instalacja oraz uruchamianie

Poniższe polecenia uruchamiamy znajdując się w katalogu projektu ("/project"). Jeśli zainstalowano tomcat7, to tomcat8 należy zmienić na tomcat7.

```
$ sudo -u postgres psql -f src/main/resources/physicalmodel.sql
$ mvn package
# cp target/project.war /var/lib/tomcat8/webapps/
# service tomcat8 start
```

Wyjaśnienie: Pierwsza z komend tworzy bazę danych z podanego modelu fizycznego. Operacja ta wymaga zalogowania się w postgresie na konto superużytkownika, domyślnym superużytkownikiem jest „postgres”. W przypadku, gdy nie ma takiego superużytkownika, należy zalogować się na jakiegoś innego superużytkownika.

Drugie polecenia uruchamia maven, który pobiera zależności opisane w pliku pom.xml i buduje plik project.war.

Następnie wyprodukowany project.war kopiujemy do katalogu, w którym serwer tomcat przechowuje aplikacje webowe.

Na samym końcu uruchamiamy serwer. Aby połączyć się z projektem należy użyć przeglądarki internetowej i połączyć się z adresem *localhost:8080/project*.

4 Dokumentacja aplikacji

4.1 Technologia

Aplikacja jest oczywiście webowa. Została napisana w Javie z wykorzystaniem frameworków Spring MVC oraz Hibernate ORM. Wykorzystano właściwości mapowania obiektowo relacyjnego takie jak: leniwe wczytywanie, cache pierwszego poziomu, filtry, oraz dziedziczenie encji (Np. Hibernate pozwala nam utrzymywać klasę Miejsce oraz jej podział na podklasy - Stacja i Serwis według wartości kolumny „rodzaj” w tabeli miejsce).

4.2 Struktura projektu

Pliki w katalogu src/main/java zostały podzielone na paczki (ang. packages) odpowiednio według kategorii:

1. com.entities - Klasy reprezentujące encje na diagramie E-R implementowane za pomocą Hibernate.
2. com.dao - Zastosowanie wzorca projektowego Data Access Object - w tej paczce znajdują się interfejsy i implementacje.
3. com.controllers - Kontrolery do wzorca projektowego MVC oferowanego przez framework Spring. Odpowiedzialne za przetwarzanie żądań.

4. `com.services` - Usługi, z których mogą korzystać kontrolery. Używają m.in. instancji klas DAO do wykonywania poleceń kontrolerów. W odróżnieniu od klas DAO, serwisy posiadają w sobie logikę, klasy DAO po prostu wydobywają dane z bazy danych.
5. `com.businessLogic` - Pozostałe klasy używane przy przetwarzaniu żądań (Własne wyjątki, klasy wrappery na dane uzyskiwane z formularzy etc.).

Oprócz tego w katalogu `src/main/webapp/WEB-INF` znajdują się pliki konfiguracyjne aplikacji oraz katalog z widokami (do wzorca MVC).

4.3 Możliwości aplikacji

Strona główna udostępnia dwie możliwości: logowanie oraz rejestrację.

4.3.1 Użytkownik

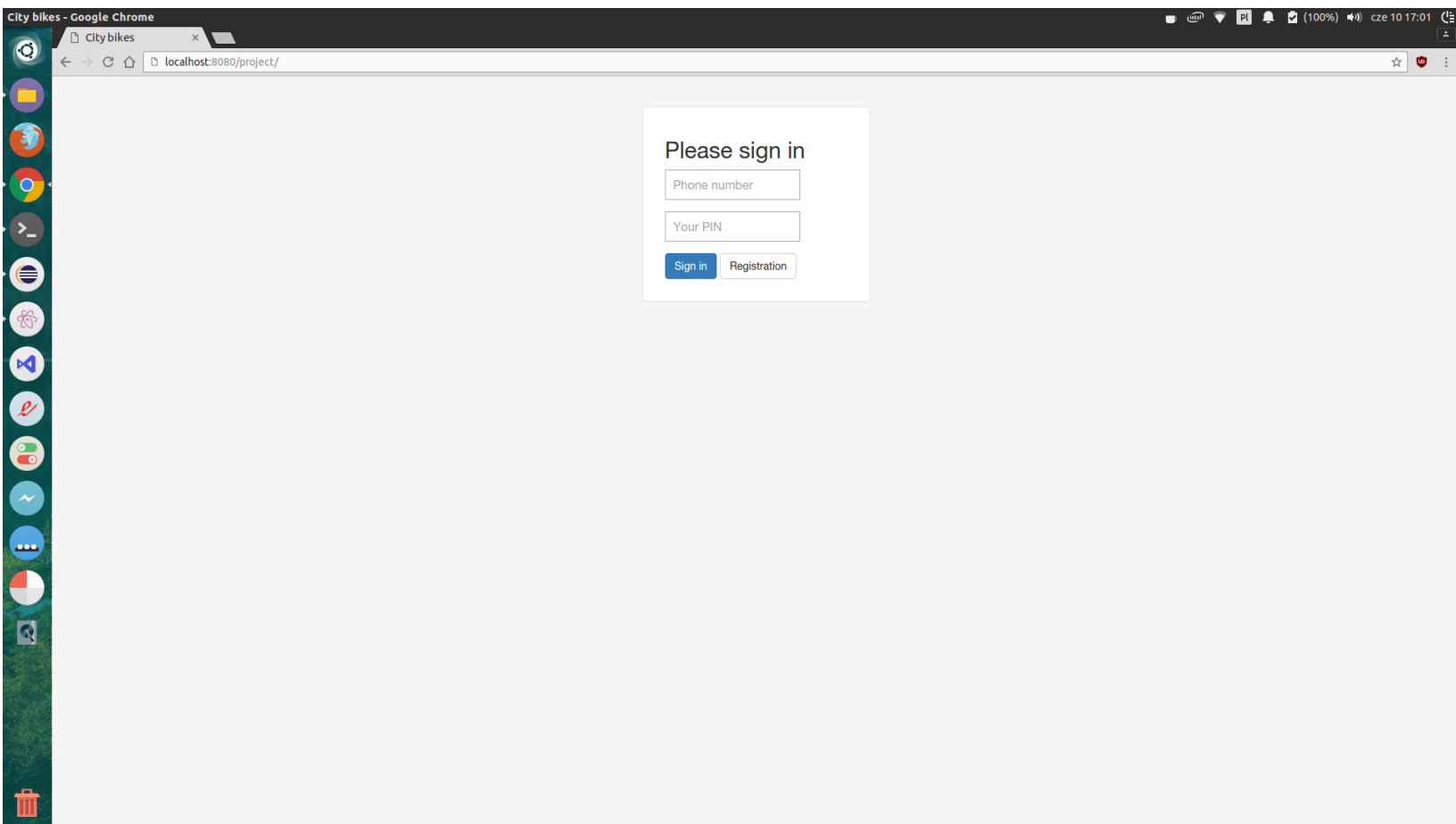
Po zalogowaniu użytkownika wita jego strona główna, na której wypisane są jego ostatnie wypożyczenia. Na stronie głównej użytkownik ma następujące możliwości:

1. Przeglądnięcie swojego konta + zmiana informacji osobowych.
2. Przeglądnięcie historii wypożyczeń oraz płatności.
3. Wypożyczenie roweru.
4. Wylogowanie się.

4.4 Administrator

Administrator ma więcej możliwości. Po zalogowaniu się widzi swoją stronę główną, gdzie może zobaczyć swoje aktywne przeniesienia rowerów oraz ma do wyboru następujące akcje:

1. Przejście do strony zarządzania użytkownikami (Zmiana typu [klient/serwisant], usuwanie).
2. Przejście do strony zarządzania rowerami (Dodawanie, edytowanie, usuwanie).
3. Przejście do strony zarządzania miejscami (Dodawanie, edytowanie, usuwanie).
4. Przejście do strony ze wszystkimi aktywnymi wypożyczeniami.
5. Wypożyczenie (Przeniesienie) roweru.
6. Wylogowanie.



Rysunek 1: Strona główna

City bikes - Registration - Google Chrome

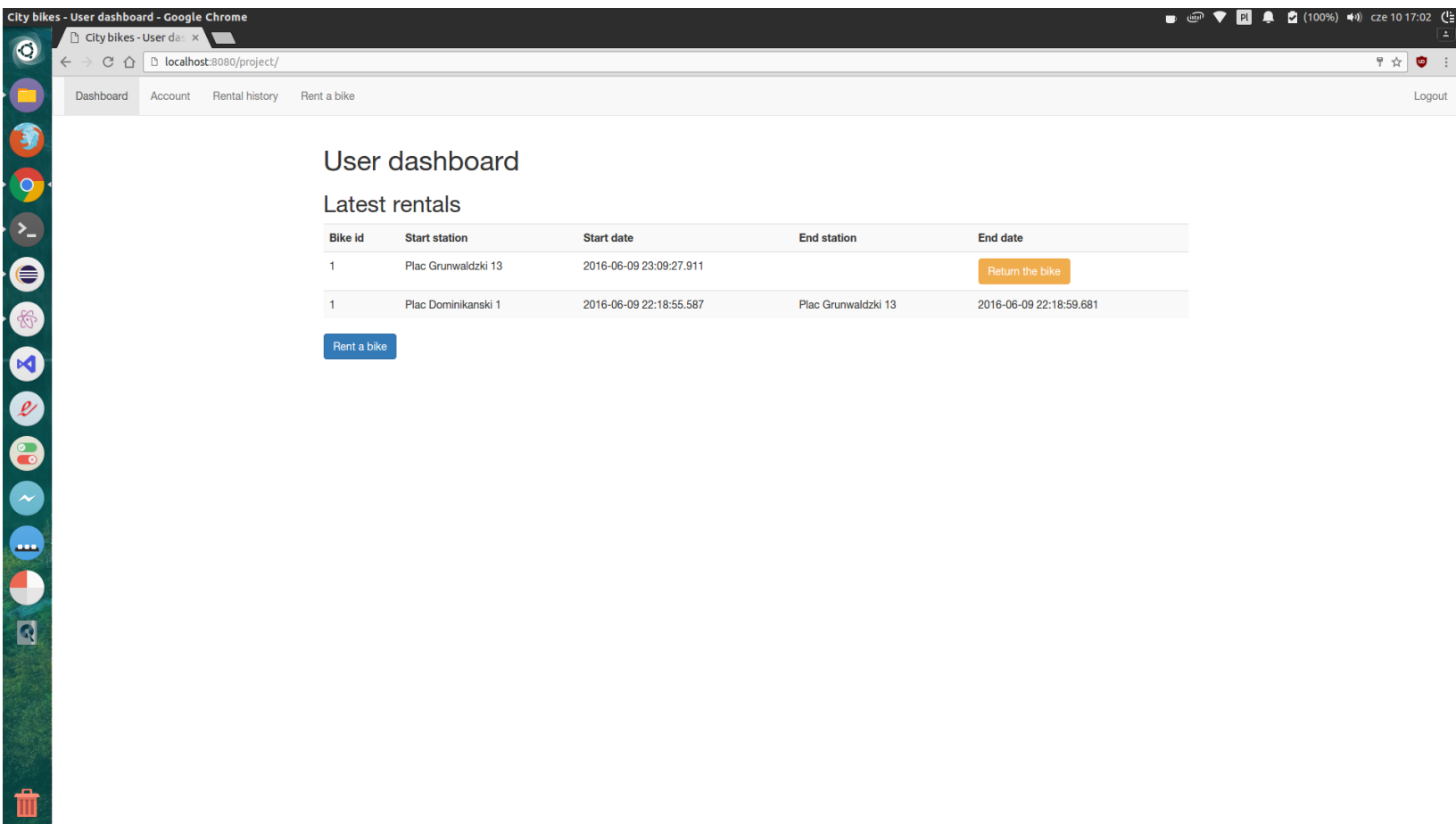
City bikes - Registro x

localhost:8080/project/register

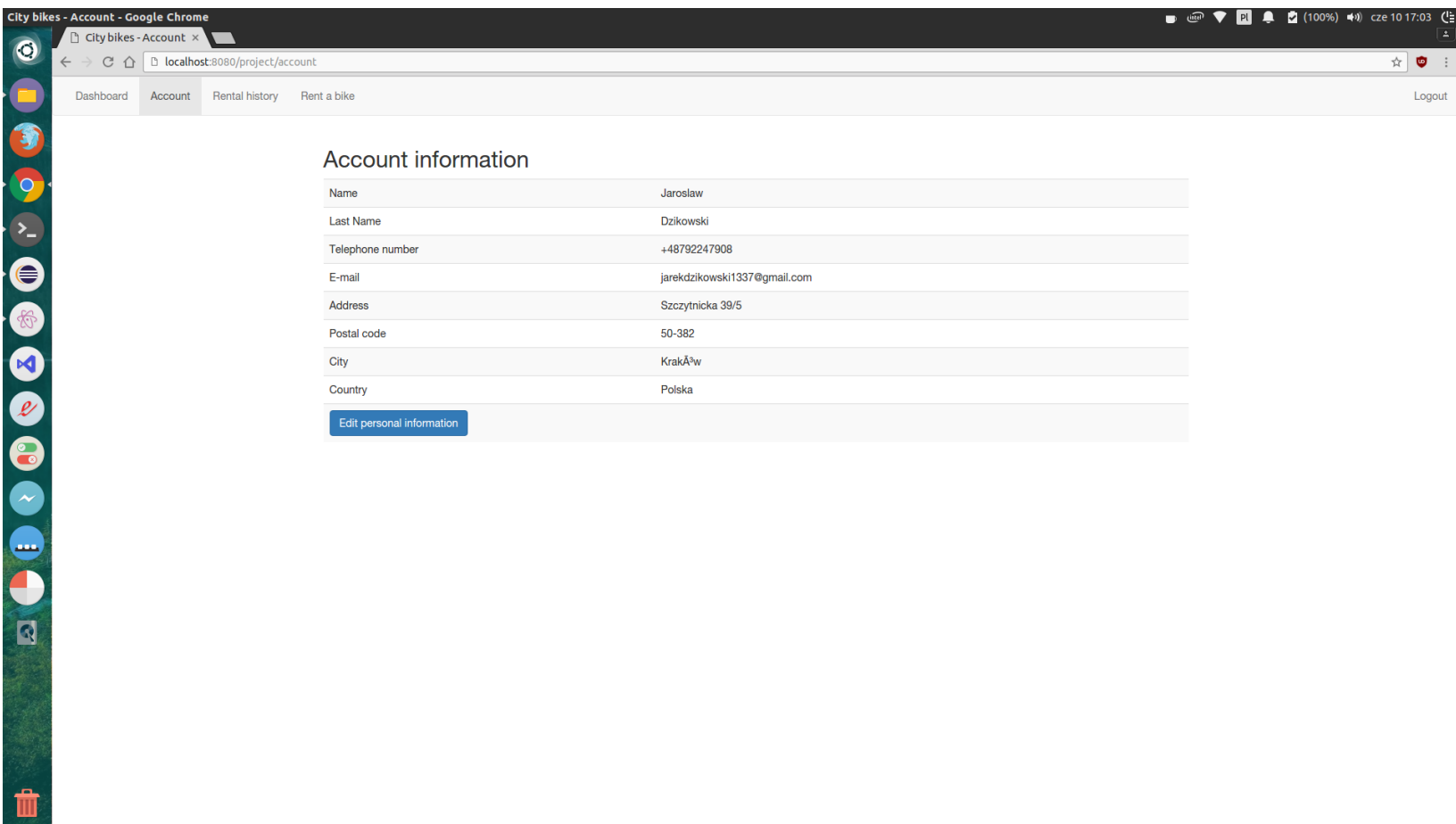
Registration

Telephone number	<input type="text"/>
PIN	<input type="text"/>
Repeat PIN	<input type="text"/>
E-mail	<input type="text"/>
Name	<input type="text"/>
Last name	<input type="text"/>
Address	<input type="text"/>
Postal code	<input type="text"/>
City	<input type="text"/>
Country	<input type="text"/>
<input type="submit" value="Submit"/>	

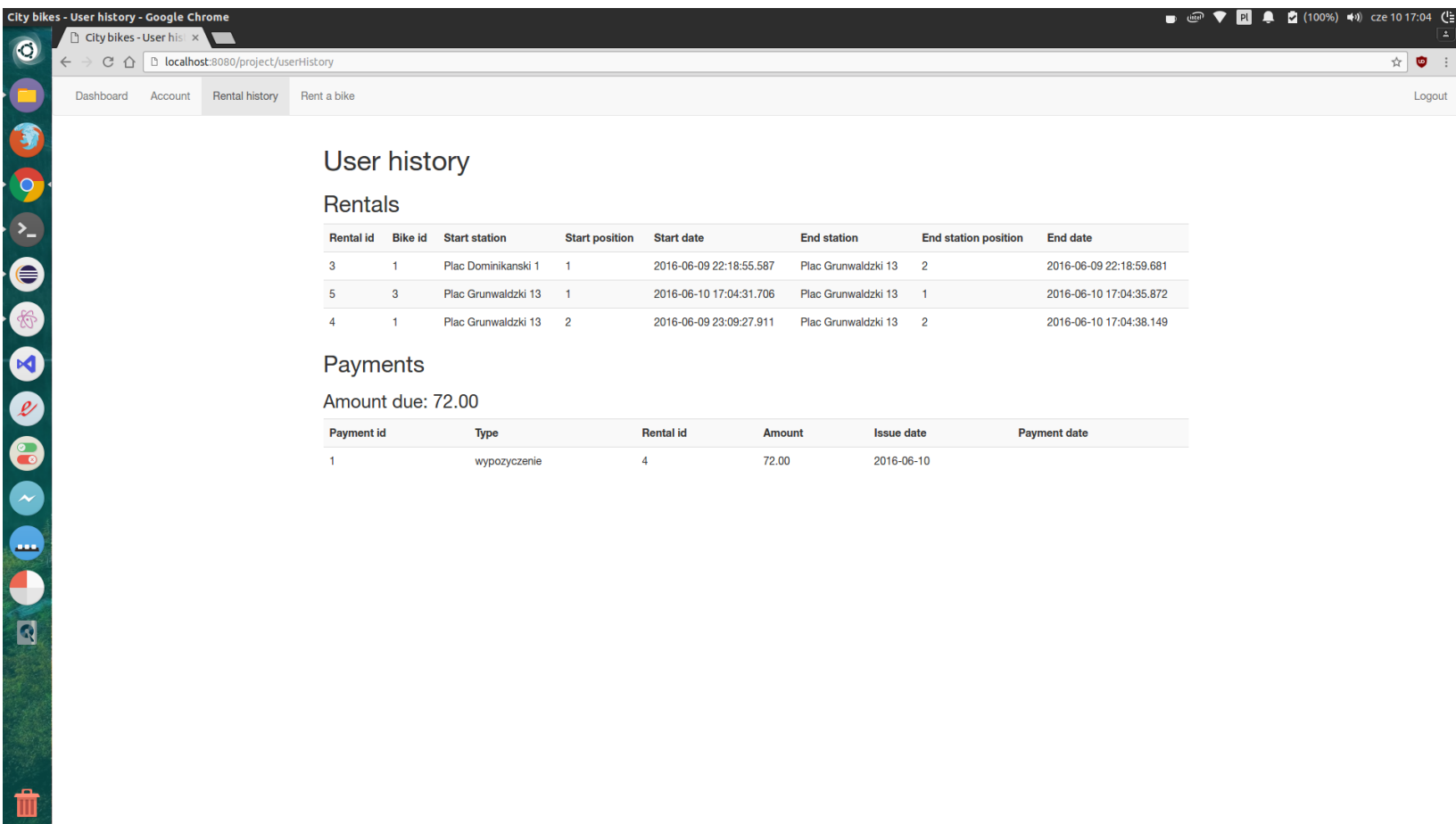
Rysunek 2: Rejestracja



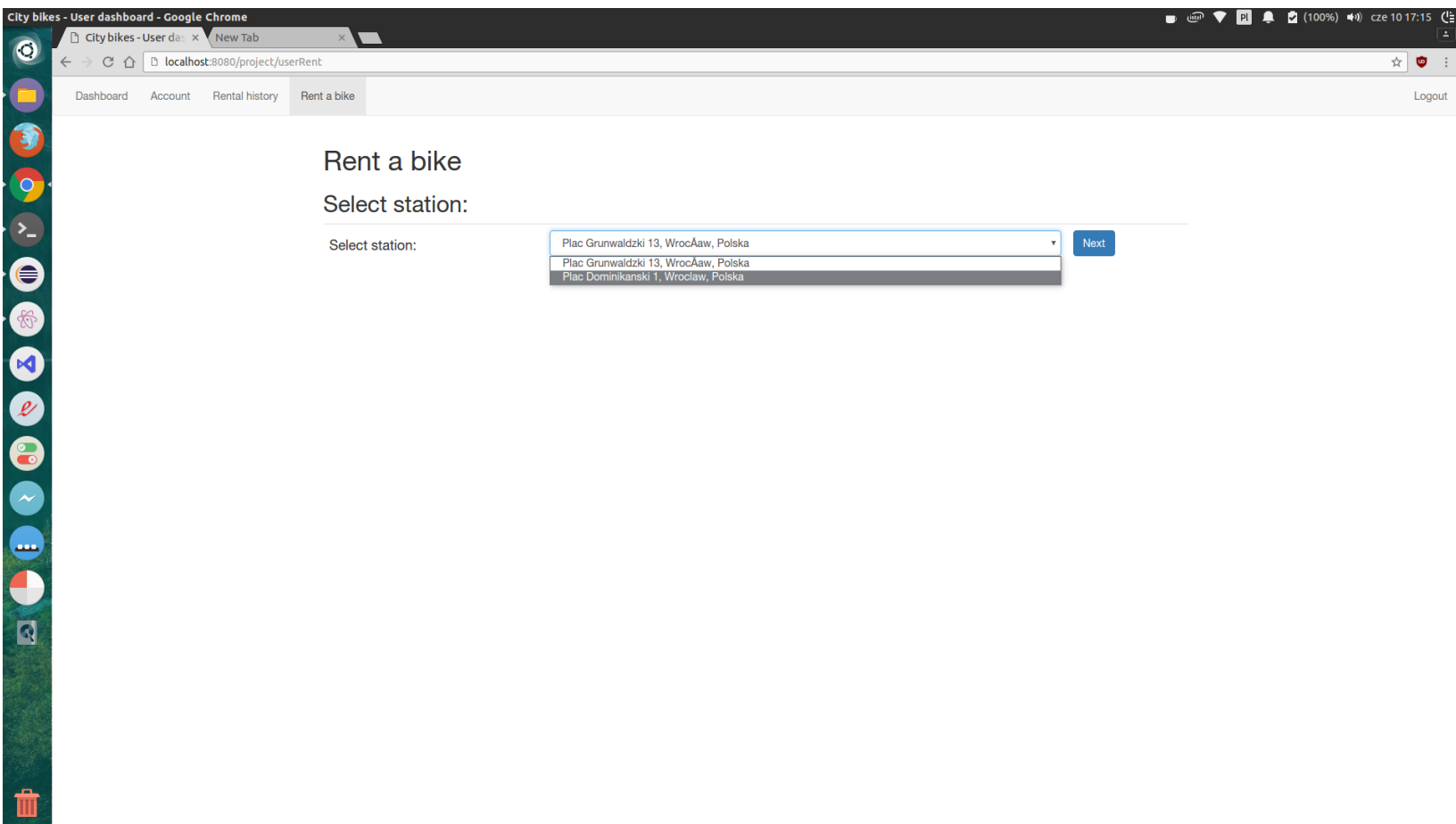
Rysunek 3: Strona główna użytkownika



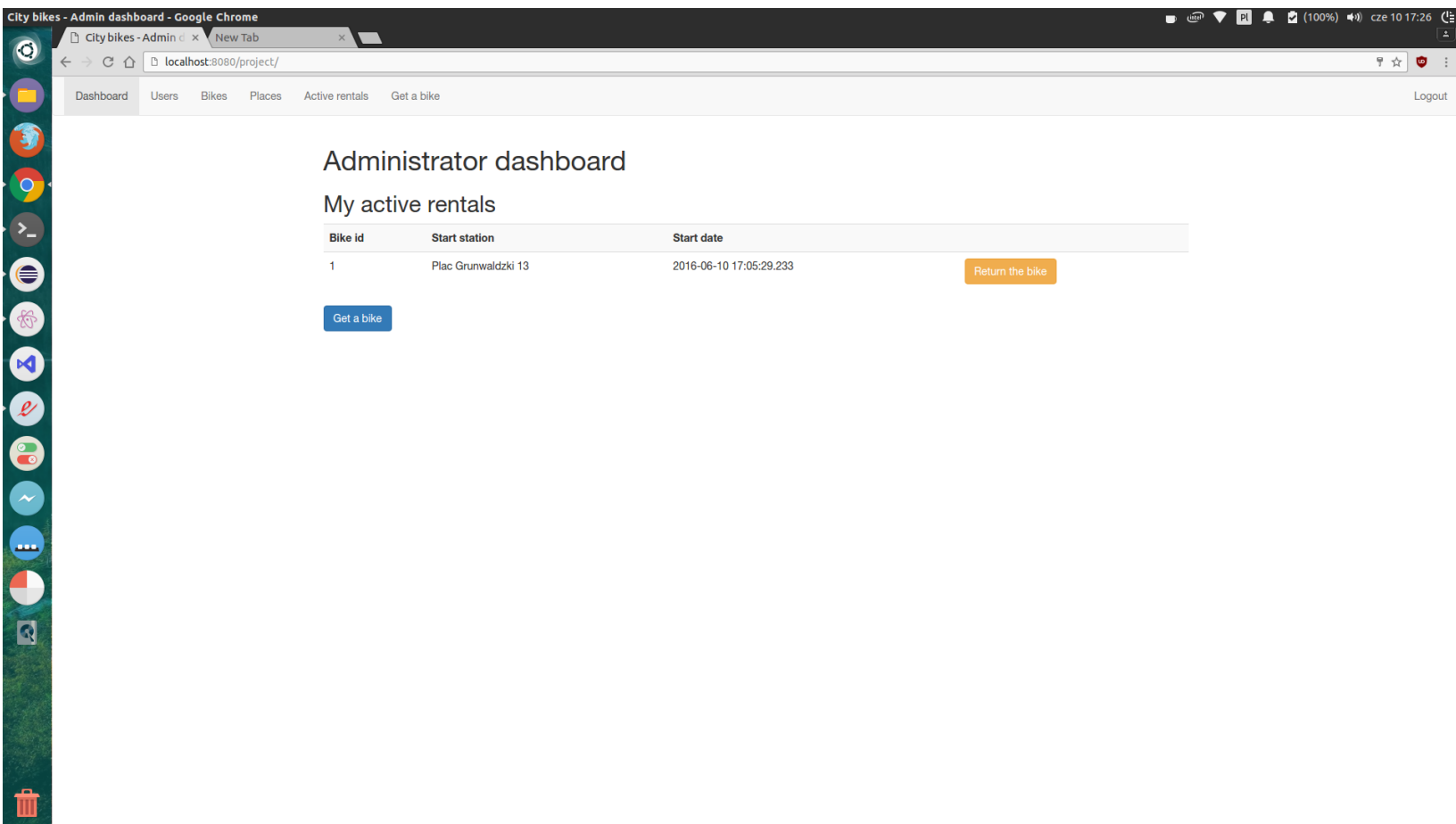
Rysunek 4: Konto użytkownika



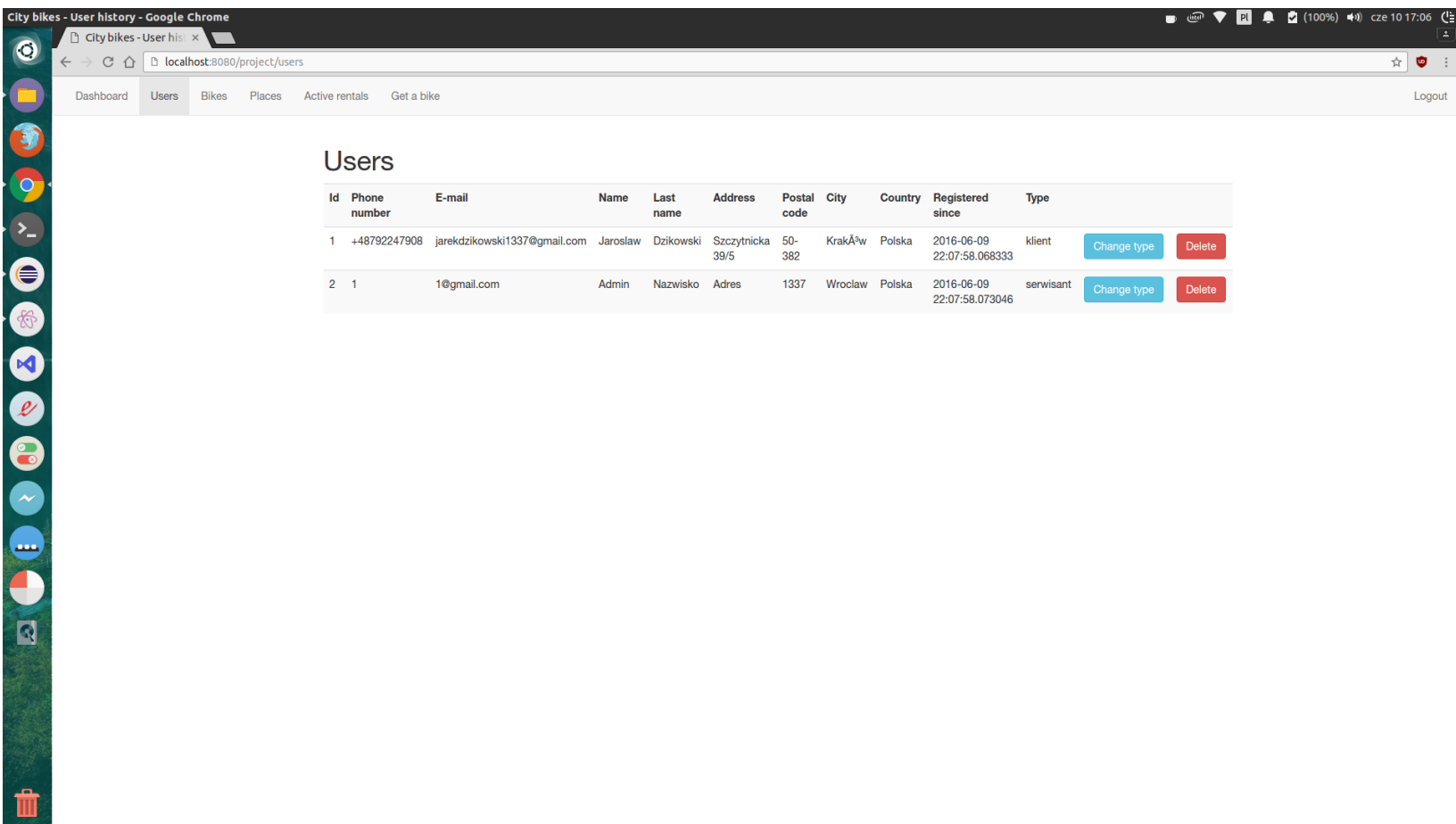
Rysunek 5: Historia użytkownika



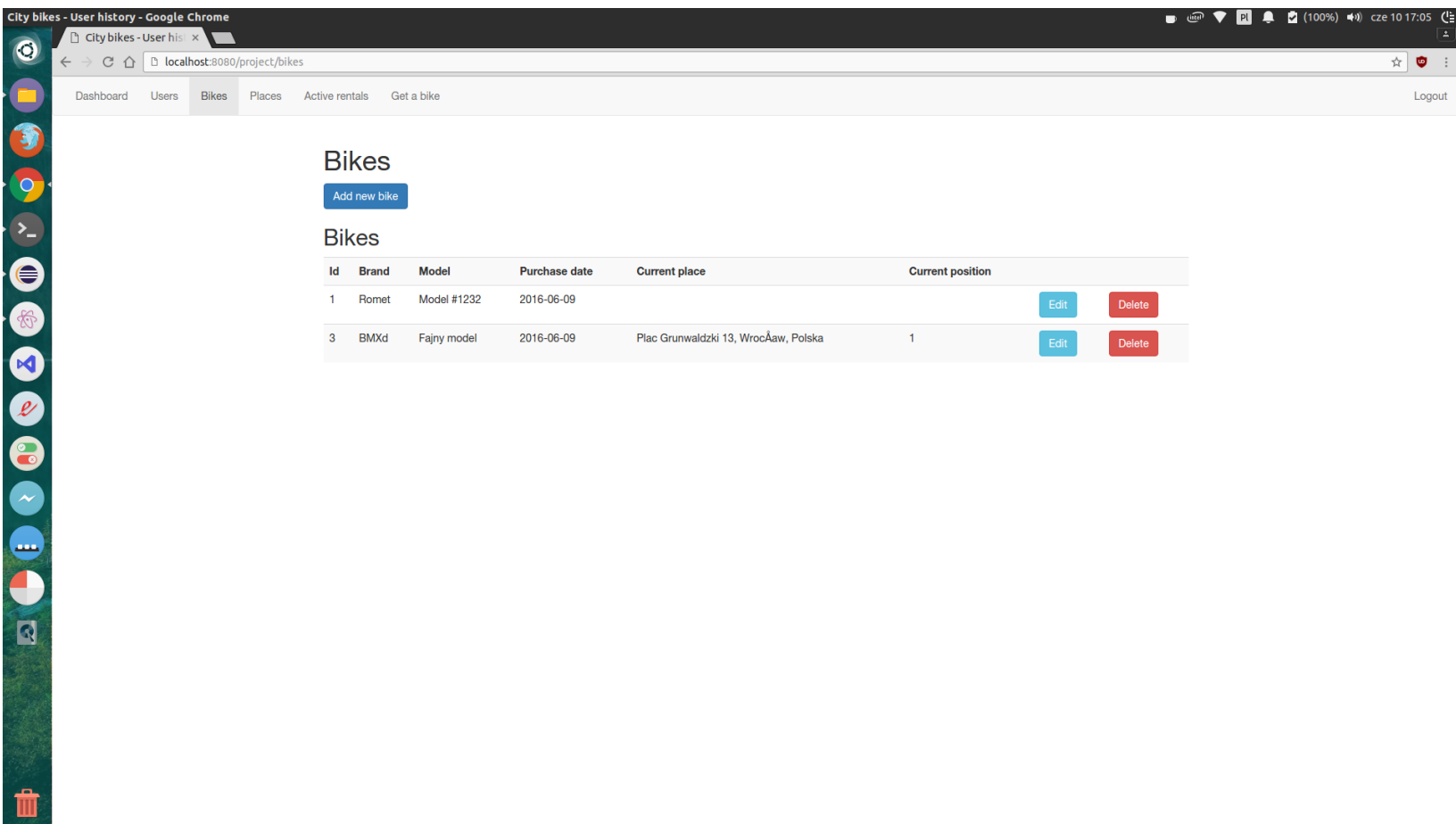
Rysunek 6: Wypożyczenie roweru



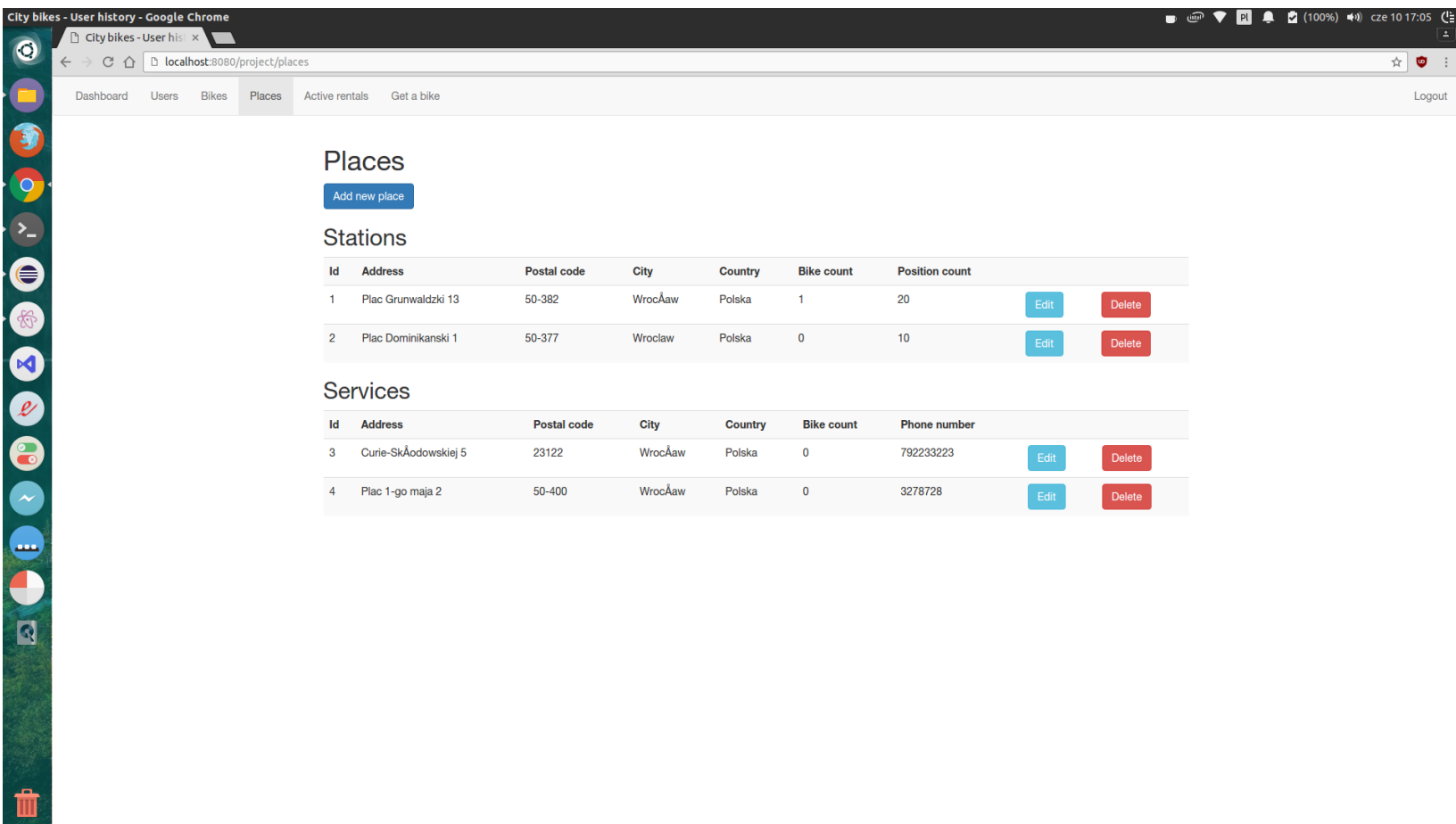
Rysunek 7: Strona główna administratora



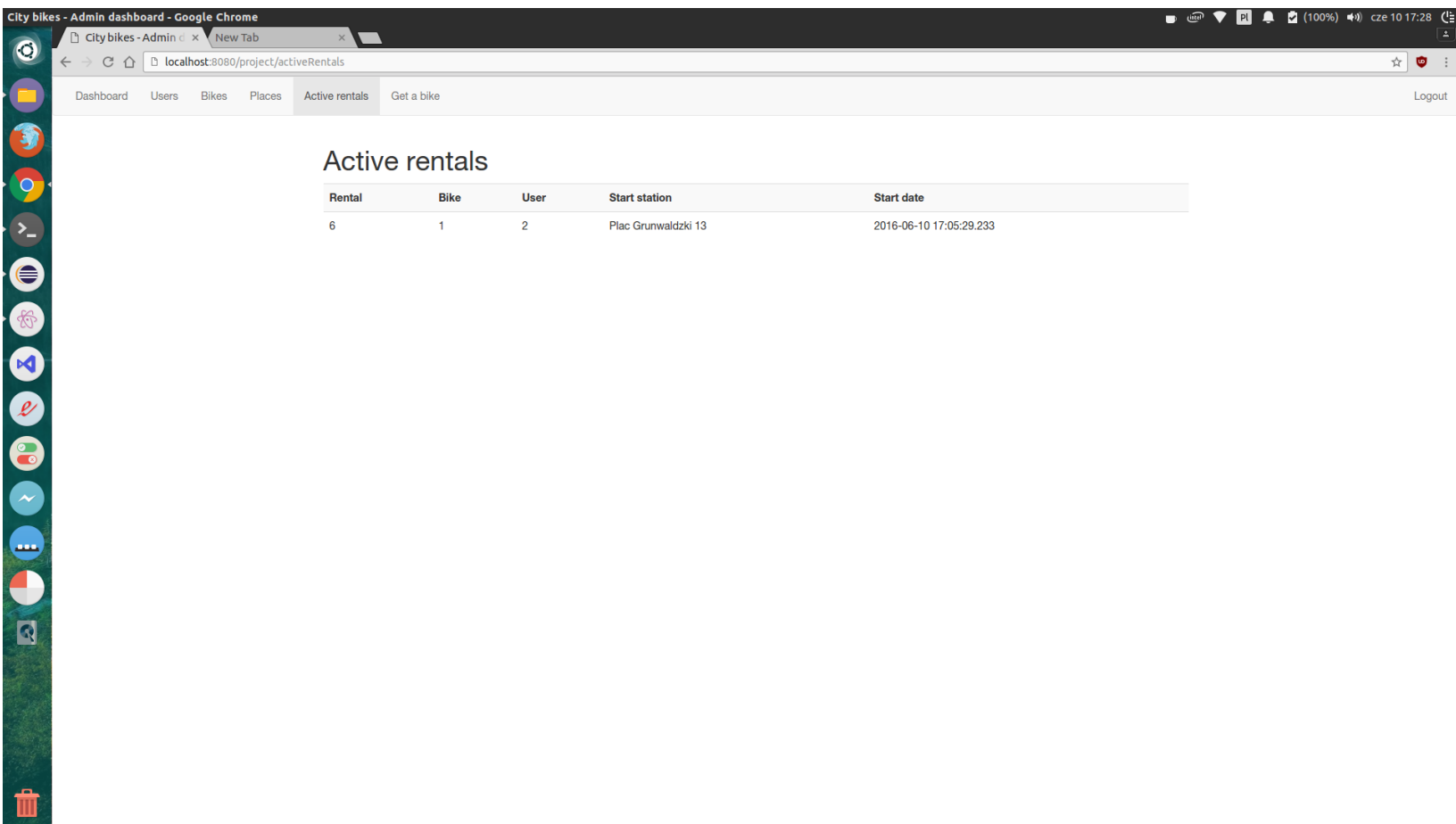
Rysunek 8: Zarządzanie użytkownikami



Rysunek 9: Zarządzanie rowerami



Rysunek 10: Zarządzanie miejscami



Rysunek 11: Aktywne wypożyczenia