



**POLITECHNIKA
RZESZOWSKA**
im. IGNACEGO ŁUKASIEWICZA



Nowoczesne Technologie Programistyczne

Projekt

Aplikacja webowa wykorzystująca Rzeczywistość Rozszerzoną

Krystian Głowiak
Dominik Filip
L02

1. Opis aplikacji

Aplikacja służy do pokazania możliwości Rzeczywistości Rozszerzonej w przeglądarkach internetowych wraz z wykorzystaniem kamery urządzenia. Jest ona kompatybilna z urządzeniami z systemami operacyjnymi Windows, oraz Android. Dzięki niej możemy zobaczyć przez kamerę obiekty 3D na znacznikach zwanych Barcodami 2D 3x3.



Część klienta została stworzona na frameworku Angular 4, z wykorzystaniem AR.JS, a wygląd aplikacji został oparty na Angular Material.

Część serwera to Spring Boot z wykorzystaniem Hibernate i JsonWebTokens, na którym zostało postawione REST API. Zawarto DAO pattern do dostępu do bazy danych.

2. Opis technologii

2.1. Front-end

Angular – Jest to open sourcowy framework bazujący na TypeScript, który służy do tworzenia aplikacji webowych. Tworzony jest przez Google, oraz społeczność.

Node.js – to środowisko open sourcowe JavaScript zaprojektowane do tworzenia wysoce skalowanych aplikacji, np. serwerów napisanych w JS. Umożliwia tworzenie aplikacji sterowanych zdarzeniami wykorzystujących asynchroniczny system wejścia-wyjścia.

NPM – Node Package Manager jest to system paczek dla środowiska Node. Można dzięki niemu łatwo instalować, aktualizować i usuwać paczki używane w projekcie.

AR.JS – jest to efektywna biblioteka open sourcowa do tworzenia aplikacji wykorzystujących rzeczywistość rozszerzoną, oraz wirtualną rzeczywistość

A-FRAME – framework, który jest używany przy tworzeniu aplikacji wykorzystujących rzeczywistość rozszerzoną i wyświetlanie obiektów w 3D.

Angular Material – framework, dzięki któremu możemy tworzyć layout aplikacji w stylu Google Material Design. Dostarcza wiele elementów, z których można budować aplikację.

2.2. Back-end

SpringBoot – jest to framework oraz kontener IoC do tworzenia aplikacji w języku Java dla platformy Java Platform i Enterprise Edition. Dzięki Spring Boot możemy tworzyć aplikacje z prekonfigurowanymi ustawieniami gotowymi do wdrożenia.

Maven - narzędzie automatyzujące budowę oprogramowania na platformę [Java](#). Poszczególne funkcje Mavena realizowane są poprzez wtyczki, które są automatycznie pobierane przy ich pierwszym wykorzystaniu.

Hibernate - framework do realizacji warstwy dostępu do danych. Zapewnia on przede wszystkim translację danych pomiędzy relacyjną bazą danych a światem obiektywnym

Spring Security – moduł do Springa, dzięki któremu możemy zapewnić autentykację oraz autoryzację w tworzonych aplikacjach Java i JavaEE.

JsonWebToken – jest to standard, który definiuje bezpieczne transmitowanie danych między stroną klienta, a stroną serwera.

3. Prezentacja aplikacji

3.1. Panel administratora




















Admin page

Items

Users





Items list

Name	Marker	Model	Edit	Delete
przedmio	 0	../././assets/models/1.obj		
prz1	 1	../././assets/models/2.obj		
elem2	 2	../././assets/models/3.obj		
3	 3	../././assets/models/4.obj		
4	 4	../././assets/models/5.obj		
5	 5	../././assets/models/6.obj		

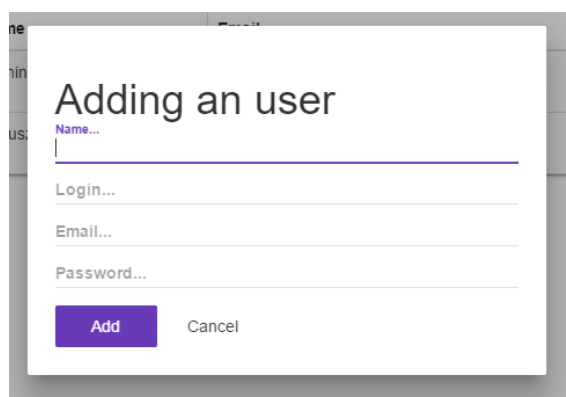
Items

Users

Users list

Login	Name	Email	More	Delete
Admin	Admin	admin@admin.com		
Janusz123	Janusz123	janusz@op.pl		

Dodawanie elementów, oraz przedmiotów do wyświetlania w kamerze.



Adding an user

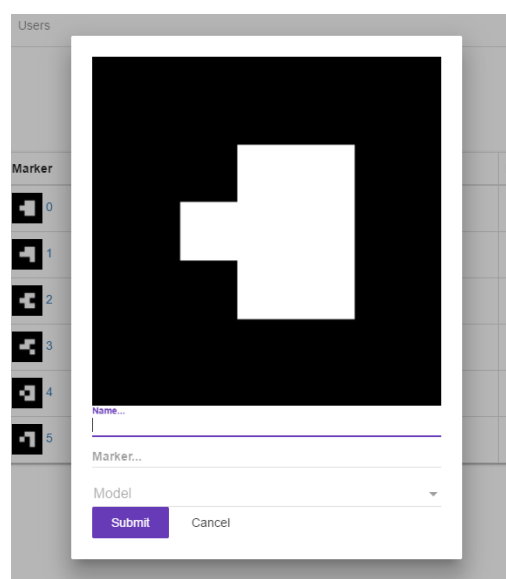
Name...

Login...

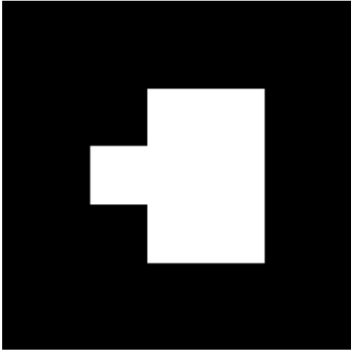
Email...

Password...

Add Cancel



Adding an item



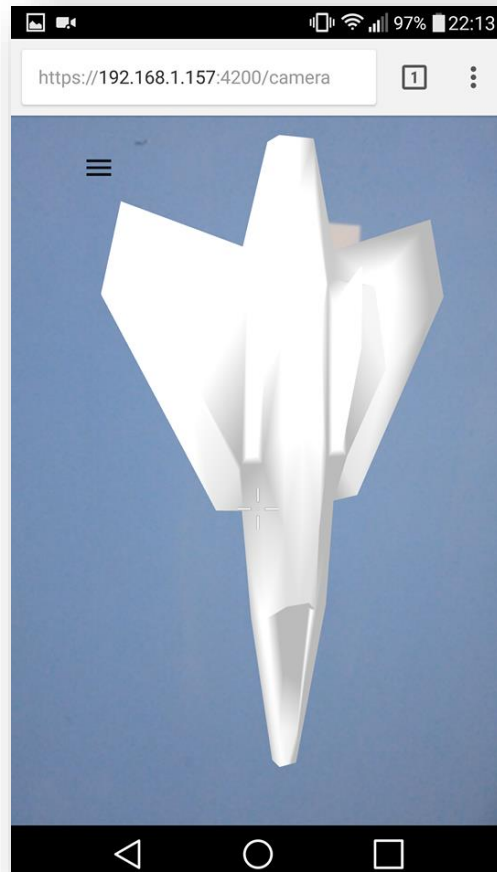
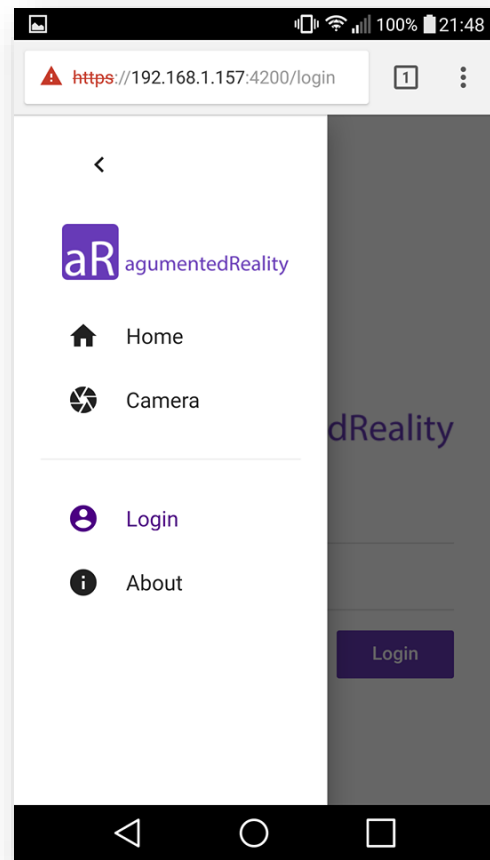
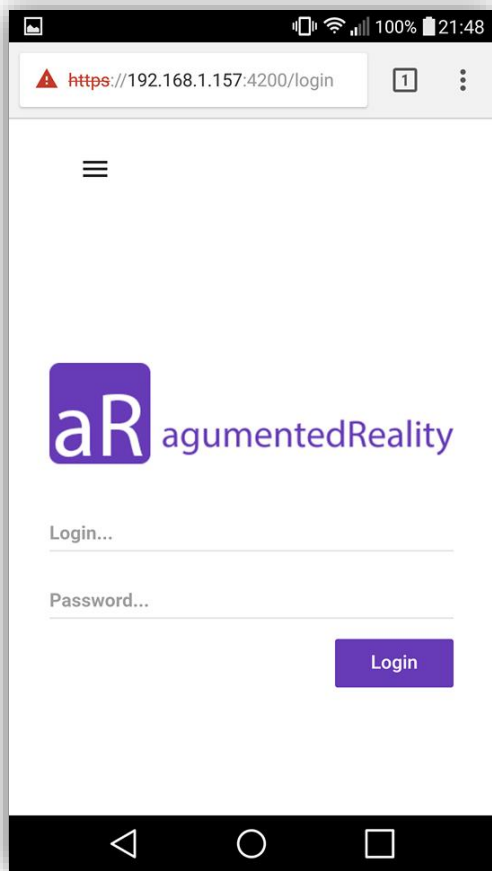
Name...

Marker...

Model

Submit Cancel

3.2. Strona klienta



4. Uruchomienie projektu

4.1. Backend

A) Aby uruchomić Backend lokalnie należy wykonać następujące czynności:

1. Zmienić parametry serwera bazy danych w pliku application.properties
 1. Adres serwera
`spring.datasource.url = jdbc:mysql://sql11.freemysqlhosting.net:3306/sql11181256`
 2. Username
`spring.datasource.username = sql11181256`
 3. Hasło
`spring.datasource.password = WgUheTFypm`
2. Zmodyfikować plik "Procfile".
 1. `$JAVA_OPTS => Xmx350m`
 2. `$PORT => 8080`
3. Spakować aplikację wpisując w terminalu:
`mvn package`
4. Uruchomić spring-boot komendą
`mvn spring-boot:run`

B) Aby uruchomić Backend na zdalnym serwerze heroku należy wykonać następujące czynności:

1. Zmienić parametry serwera bazy danych w pliku application.properties
 1. Adres serwera
`spring.datasource.url = jdbc:mysql://sql11.freemysqlhosting.net:3306/sql11181256`
 2. Username
`spring.datasource.username = sql11181256`
 3. Hasło
`spring.datasource.password = WgUheTFypm`
2. Zalogować się do swojego konta heroku
`cmd: heroku login`
3. Utworzyć nowy projekt
`cmd: heroku create`
4. Wysłać aplikację na serwer heroku
`cmd: git add.`
`cmd: git commit -am "initial commit"`
`cmd: git push heroku master`

4.2. Front-end

Aby zainicjować i uruchomić project należy pobrać node.js w najnowszej wersji z oficjalnej strony <https://nodejs.org/en/> a następnie go zainstalować. Po instalacji do zmiennych środowiskowych zostanie dodane npm (Node Package Manager), dzięki któremu jesteśmy w stanie instalować paczki udostępniane w <https://www.npmjs.com/>.

Następnie uruchamiamy okno linii poleceń i wpisujemy komendę:

```
npm install -g @angular/cli
```

Gdy proces się zakończy należy przejść do katalogu, gdzie znajduje się aplikacja i uruchomić w nim okno poleceń (w przypadku windowsa SHIFT+PPM w folderze, a następnie „Uruchom okno poleceń tutaj”). Gdy uruchomimy okno poleceń należy upewnić się czy lokacja jest prawidłowa i wpisać:

```
npm install
```

Proces ten to pobieranie wszystkich potrzebnych paczek i zależności, które są potrzebne, aby uruchomić projekt. Potrwa on kilka minut i jest jednorazowy.

Po zakończeniu możemy uruchomić projekt poprzez komendę:

```
npm start
```

Gdy zobaczymy poniższy komunikat aplikacja będzie gotowa do uruchomienia

```
** No live development server is running on http://localhost:4200 **  
Hash: 218d13bad6666ed7a7a0  
Time: 23373ms  
chunk {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 157 kB {5} [initial] [rendered]  
chunk {1} main.bundle.js, main.bundle.js.map (main) 74.8 kB {4} [initial] [rendered]  
chunk {2} styles.bundle.js, styles.bundle.js.map (styles) 194 kB {5} [initial] [rendered]  
chunk {3} scripts.bundle.js, scripts.bundle.js.map (scripts) 162 kB {5} [initial] [rendered]  
chunk {4} vendor.bundle.js, vendor.bundle.js.map (vendor) 5.79 MB [initial] [rendered]  
chunk {5} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [rendered]  
webpack: Compiled successfully.
```

Abu uruchomić aplikację wchodzimy w przeglądarce internetowej na stronę <http://localhost:4200/>.