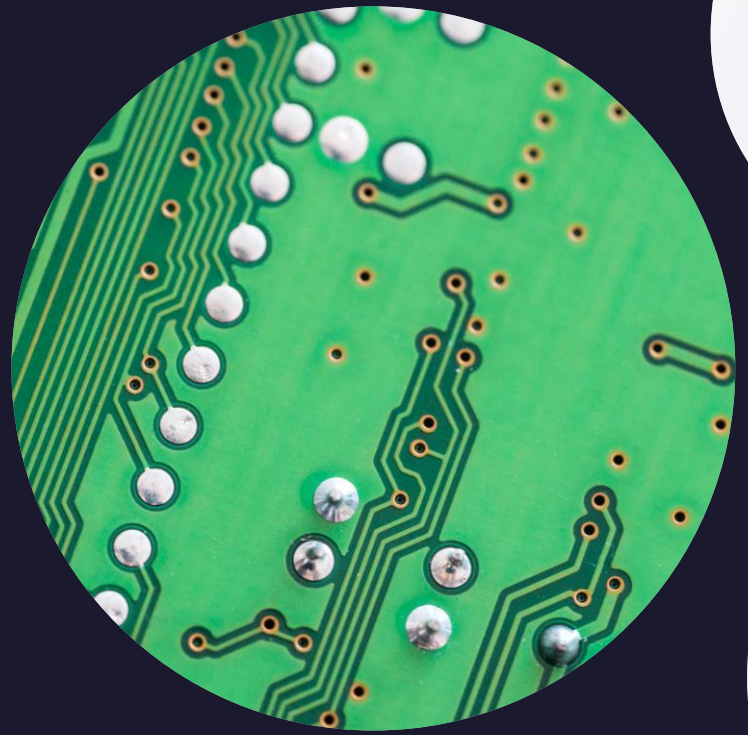
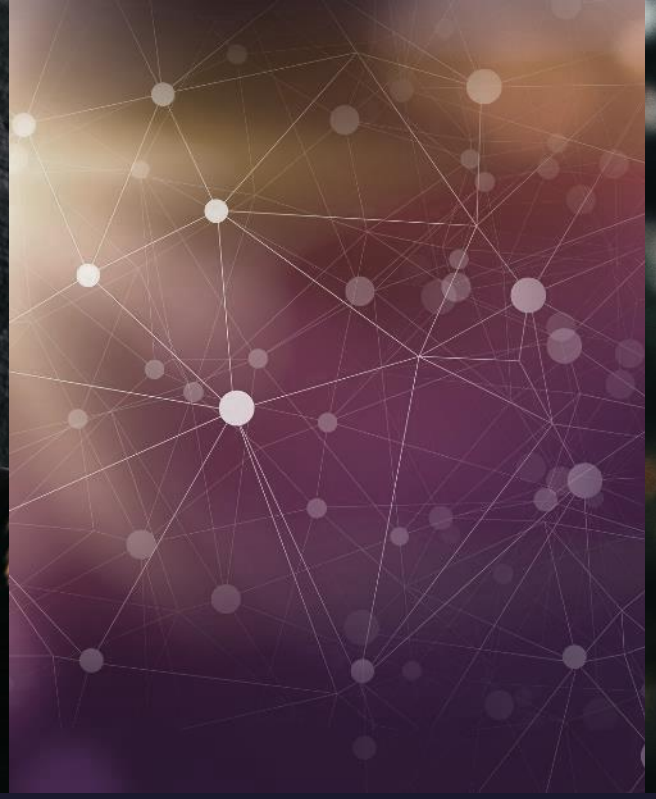
Drone Services App

Eugeniu Orlov

Agenda

- Descrierea aplicației
- Criterii de design
- Arhitectura generală a aplicației
- Backend
- Frontend
- Structura bazei de date
- Securitatea aplicației



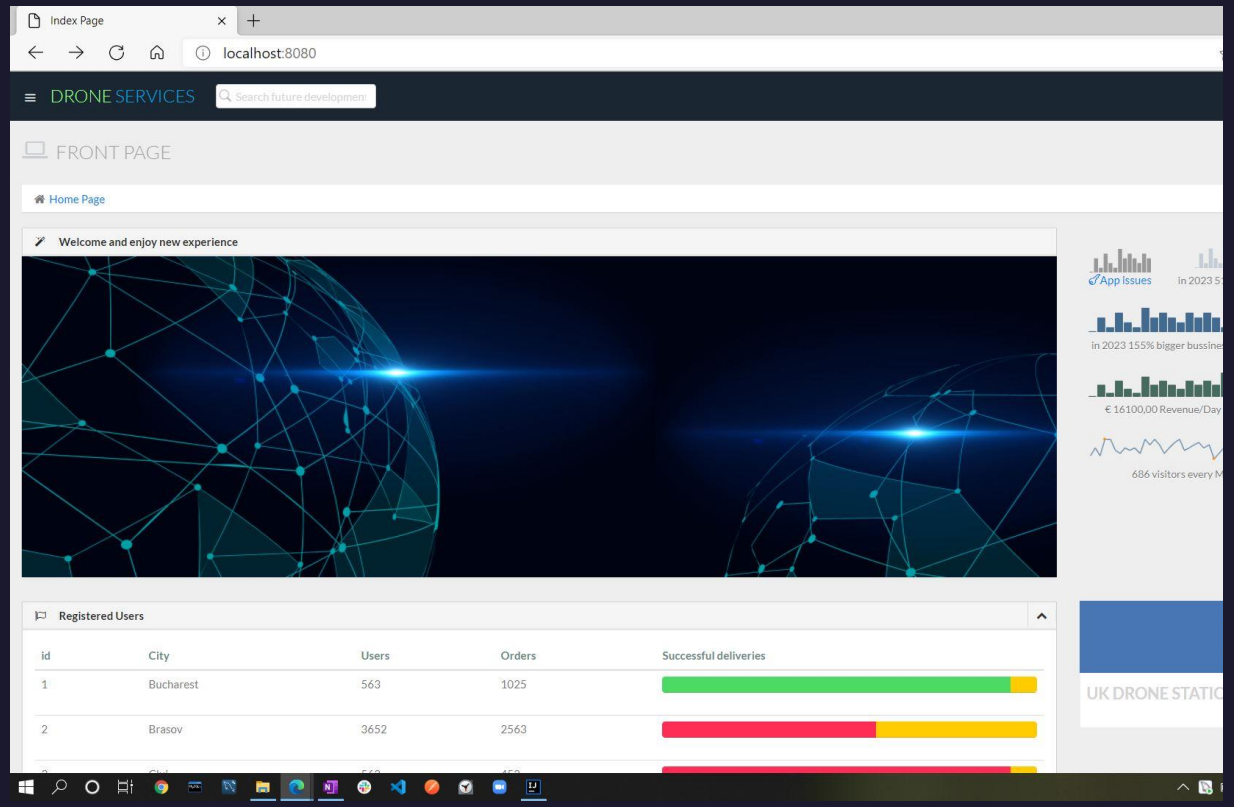



Introducere

- APLICAȚIE WEB PENTRU SERVICII CU DRONE
- Aplicația permite utilizatorilor să acceseze servicii precum livrarea de pachete mici sau servicii de securitate la distanță și transport persoane / animale, în zone greu accesibile, cu ajutorul dronelor

Descrierea aplicației

Descrierea aplicației



- Aplicația modulară integrează servicii cu drone, având capacitatea să calculeze distanțele, costurile și capacitatea de transport în funcție de greutate și de gradul de încărcare al bateriilor.
- Conceptul inițial se bazează pe livrări de pachete mici din punct fix în punct fix cu autonomie de zbor până la 5km în aceeași localitate, respectiv cu verificarea disponibilității serviciului în localitatea indicată de client.
- Aplicația generează statistici în caz de indisponibilitate și asigură analizarea creării unor noi centre geografice în funcție de cerere.

Descrierea aplicației

SISTEMELE PRINCIPALE DE FUNCȚIONARE

Localitatea
- Regiunea de acțiune
- Managementul listei de locații

Clienții
- Managementul Listei de clienți

Dronele
- Tipuri de drone
- Managementul Listei de drone

TEHNOLOGII UTILIZATE

- Java
- JavaScript
- Spring Security - user login
- Spring Data JPA
- Hibernate
- MySql
- Thymeleaf
- HTML
- CSS
- Bootstrap și Jquery
- GitHub

ENTITĂȚILE DE BAZĂ

(geografic) - Oras/localitate	*Nume, Descriere, Cod postal/cordonate geografice*
(users) - Client	*Nume, Adresa, Orasul/localitatea, E-mail, Telefon*
(services) - Drone	*Nume, disponibilitate*
(services) - Tipul dronei	*Descriere, Detalii*
(geografic) - Locatii disponibile pentru preluari/livrari indirecte	*Adresa, Descriere, Detalii*
(geografic) - Zone de actiune/regiuni	*Nume comun/cartier, Detalii, Coordonate/cod postal*

Aplicația este împărțită în:

BACKEND

- Partea funcțională a aplicației care face legătura cu baza de date și interacționează exclusiv cu programatorul.

FRONTEND

- Partea aplicației web vizibilă în interacțiunea cu utilizatorul, compusă din:
  - Zona dedicată administratorului bazei de date
  - Zona dedicată tuturor vizitatorilor site-ului

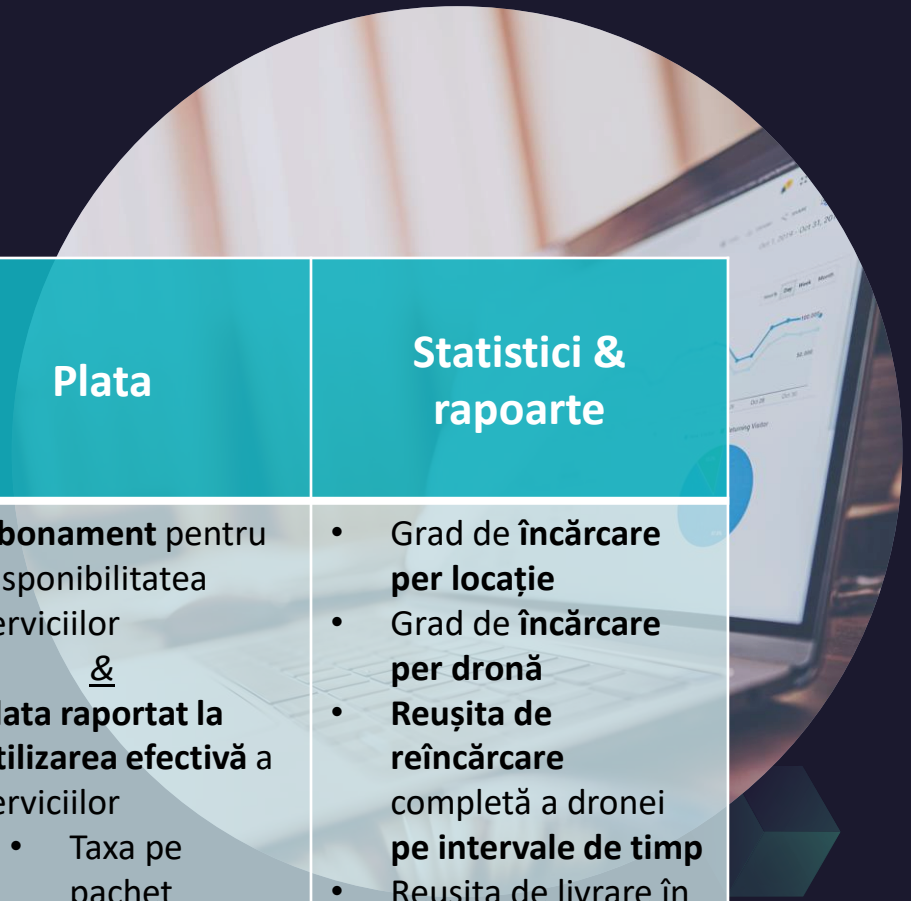
Criterii de design

Criterii de design

Sistemul creat este destinat companiilor de curierat, instituțiilor sau companiilor care transmit plicuri/pachete mici în mod curect între locații fixe.

Ca atare, aplicația trebuie să poată fi **personalizată ușor** și să ofere funcționalități de **management de bază** pentru fiecare entitate (*creare, editare, ștergere*).

Funcționalități de bază



Locațiile	Dronele	Clienții	Plata	Statistici & rapoarte
• **Editabile** în funcție de cerere • **Clasate** pe regiuni de acțiune • **Interconectate** cu cererea clienților	• **Alocate între 2 locații** din raza de acțiune / regiune • **Utilizabile** de mai mulți clienți • **Livrare imediată sau cu temporizator**	• **Predefiniți sau înrolați** extern	• **Abonament** pentru disponibilitatea serviciilor & • **Plata raportat la utilizarea efectivă** a serviciilor • Taxa pe pachet (*greutate, distanță*)	• Grad de **încărcare per locație** • Grad de **încărcare per dronă** • **Reușita de reîncărcare** completă a dronei **pe intervale de timp** • Reușita de livrare în funcție de **condițiile exterioare de zbor** • **Întârzieri** la livrare • **Eșuarea** livrării

Constrângeri de business

- Clarificarea cadrului legal aplicabil privind utilizarea spațiului aerian
- Identificarea clienților potențiali
- Identificarea și atragerea investitorilor
- Identificarea de parteneri și furnizori
  - Pentru echipamente UVA
  - Pentru stații de reîncărcare și soluții de automatizare
  - Pentru platformele de preluare și livrare a pachetelor

Arhitectura generală a aplicației

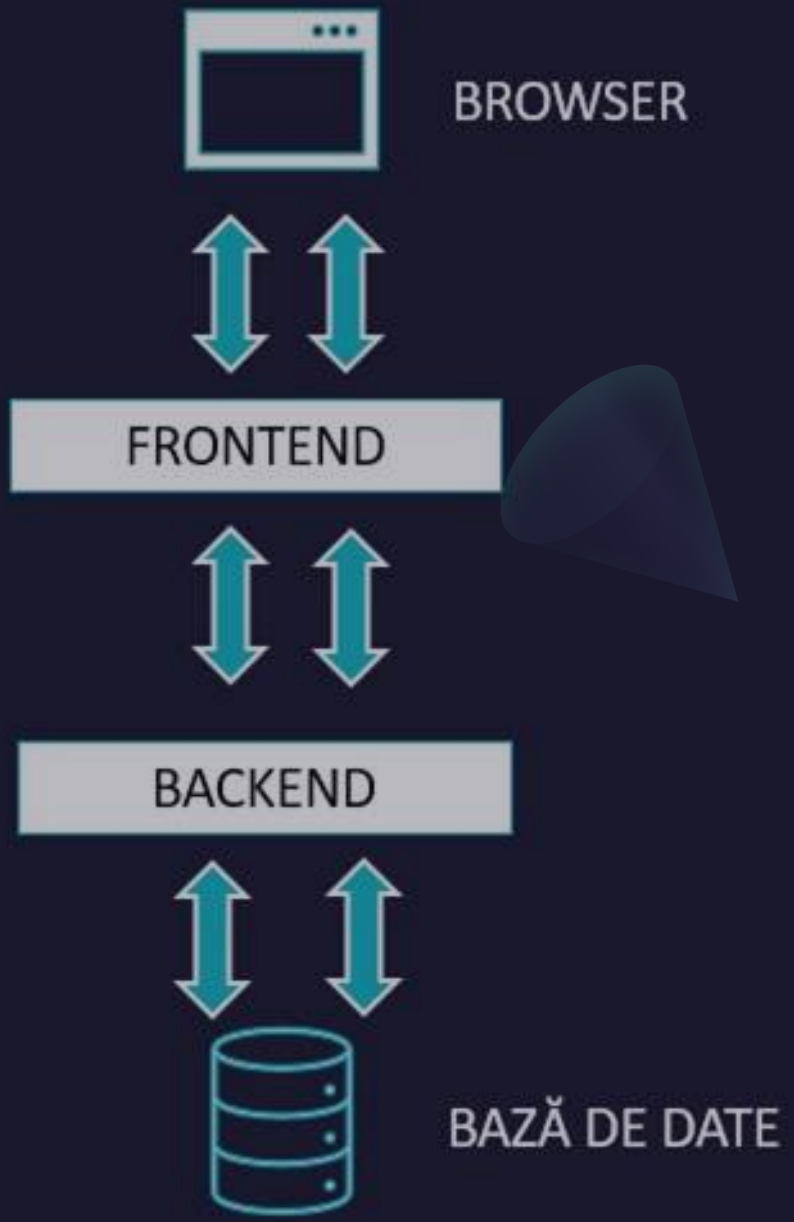
Arhitectura generală a aplicației

INSTRUMENTE FOLOSITE

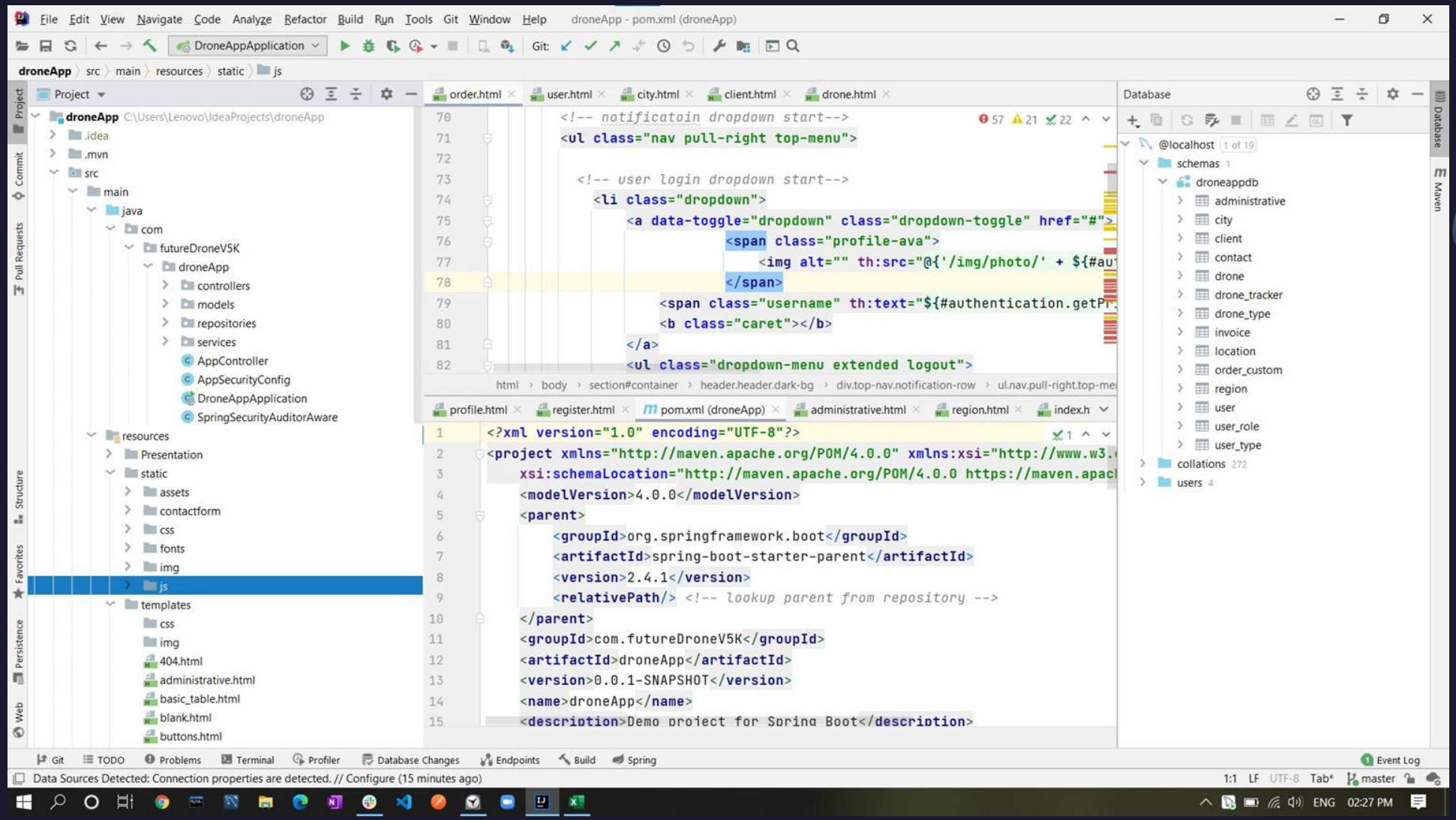
- **Frontend & Backend**
  - Intelij IDEA
- **Baza de date**
  - MySQL
- **Testare**
  - Browser

TEHNOLOGII / FRAMEWORK UTILIZAT

- **Frontend**
  - HTML, CSS, Thymeleaf,
  - JavaScript, Bootstrap, Jquery
- **Backend**
  - Java
  - Spring Suite (JPA,Data,Security)
- **Baza de date**
  - Hibernate, MySQL, Spring Datasource

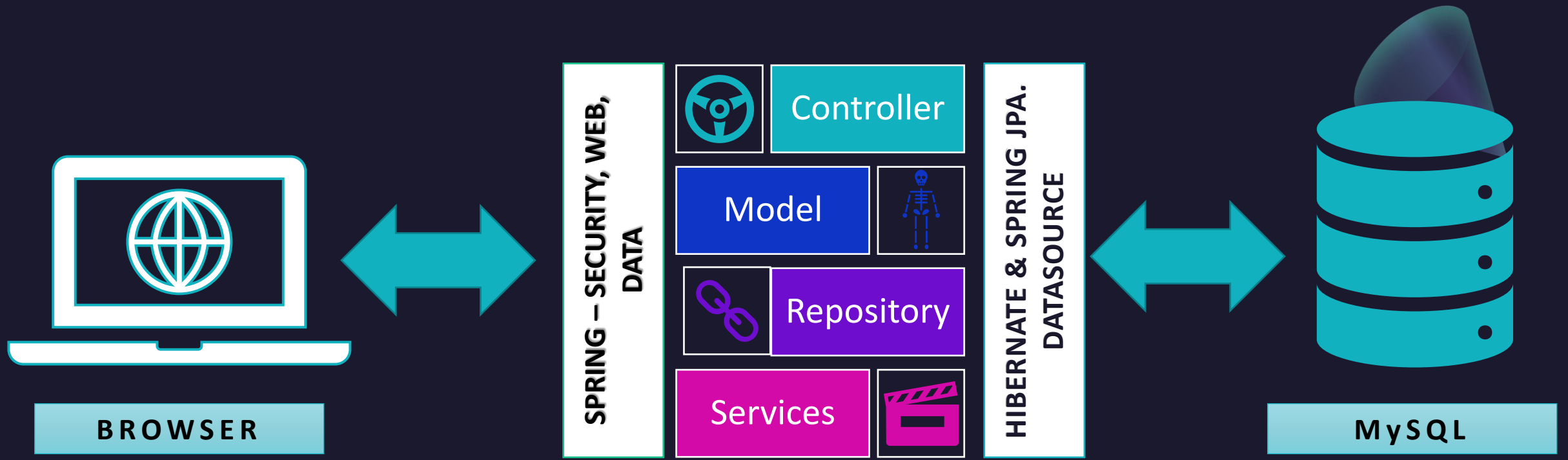


Structura proiectului

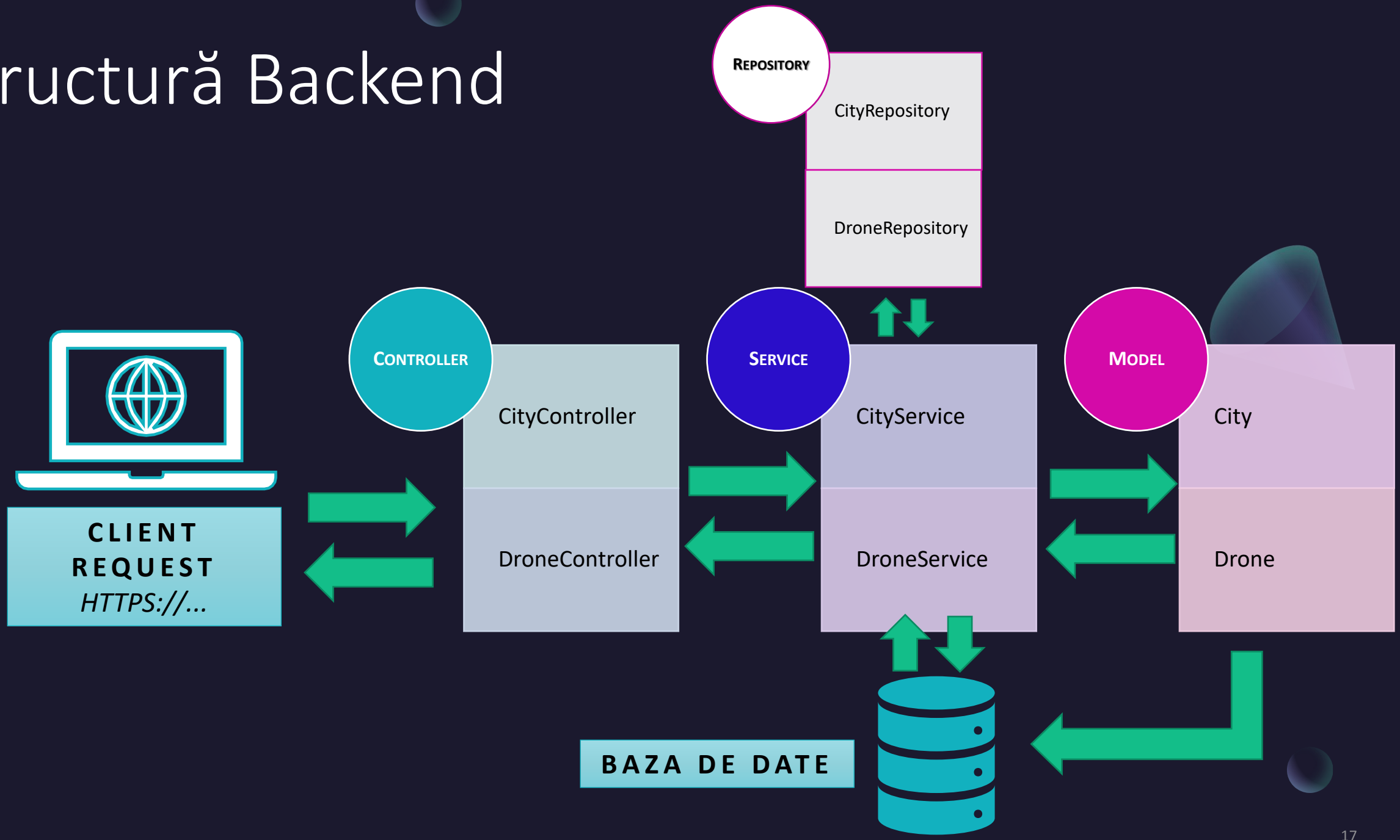


Backend

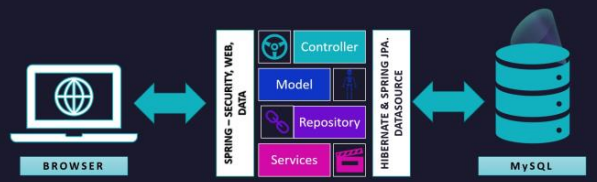
Arhitectură Backend



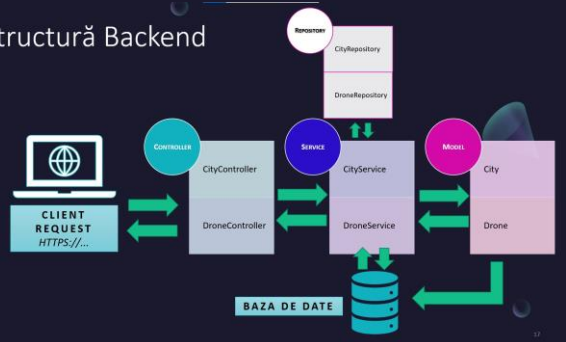
Structură Backend

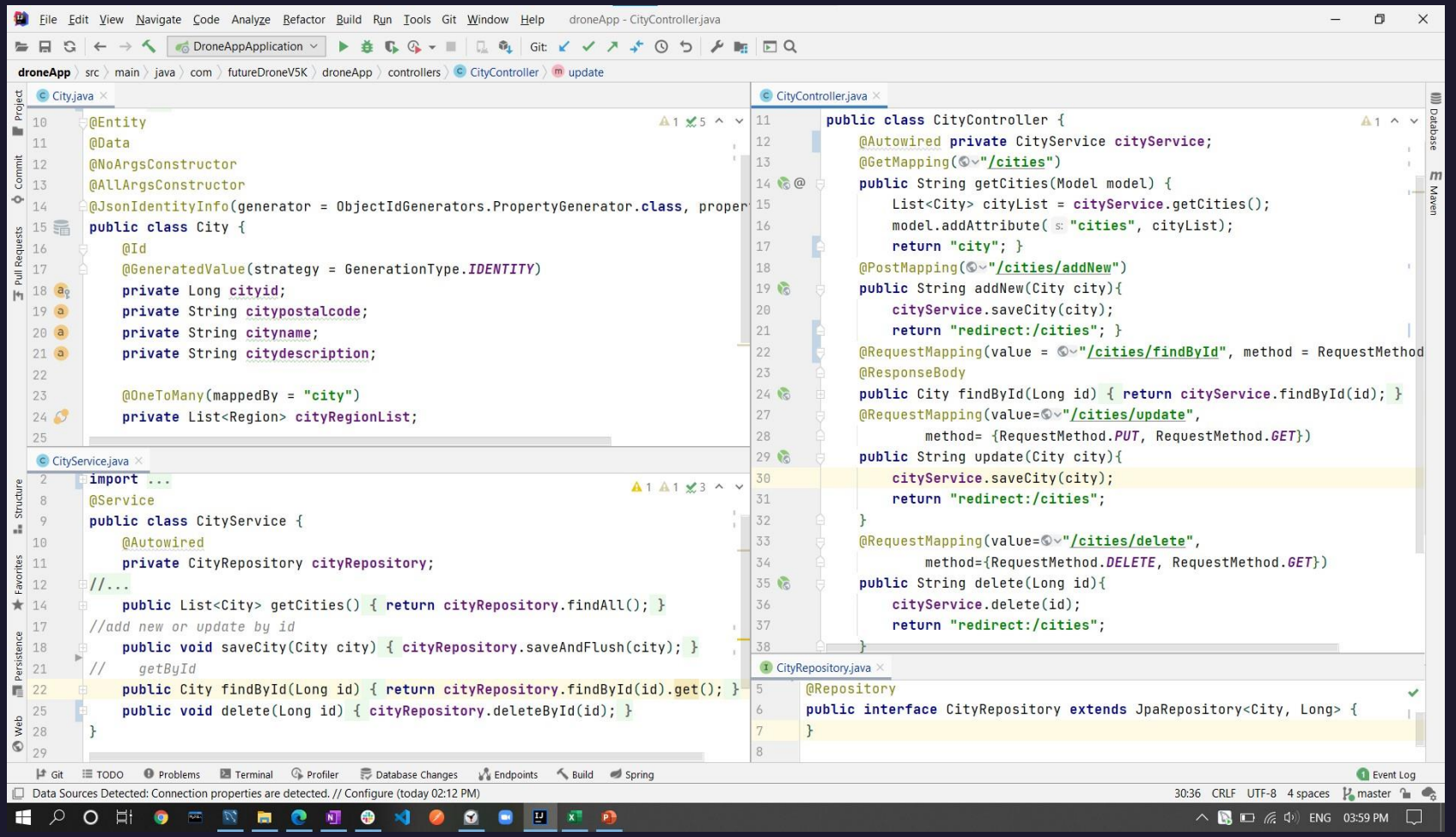


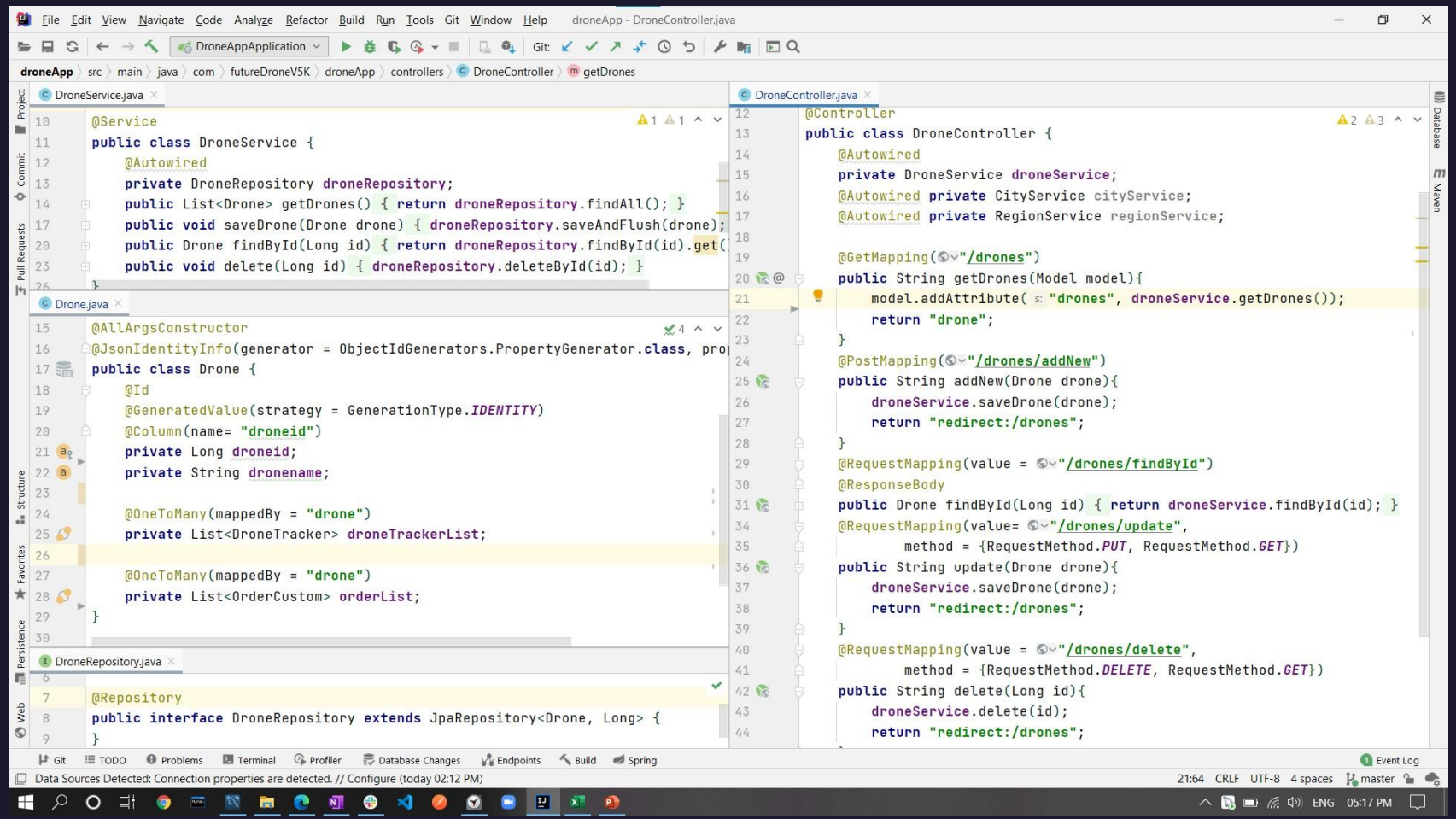
Arhitectură Backend



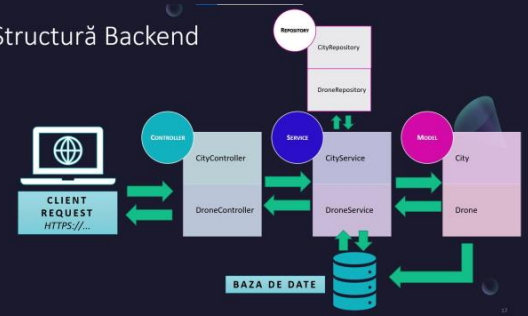
Structură Backend



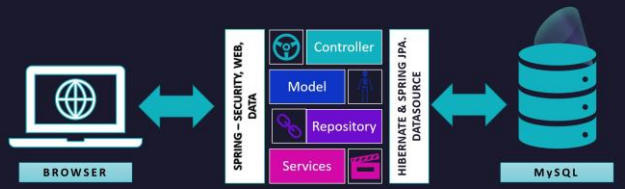




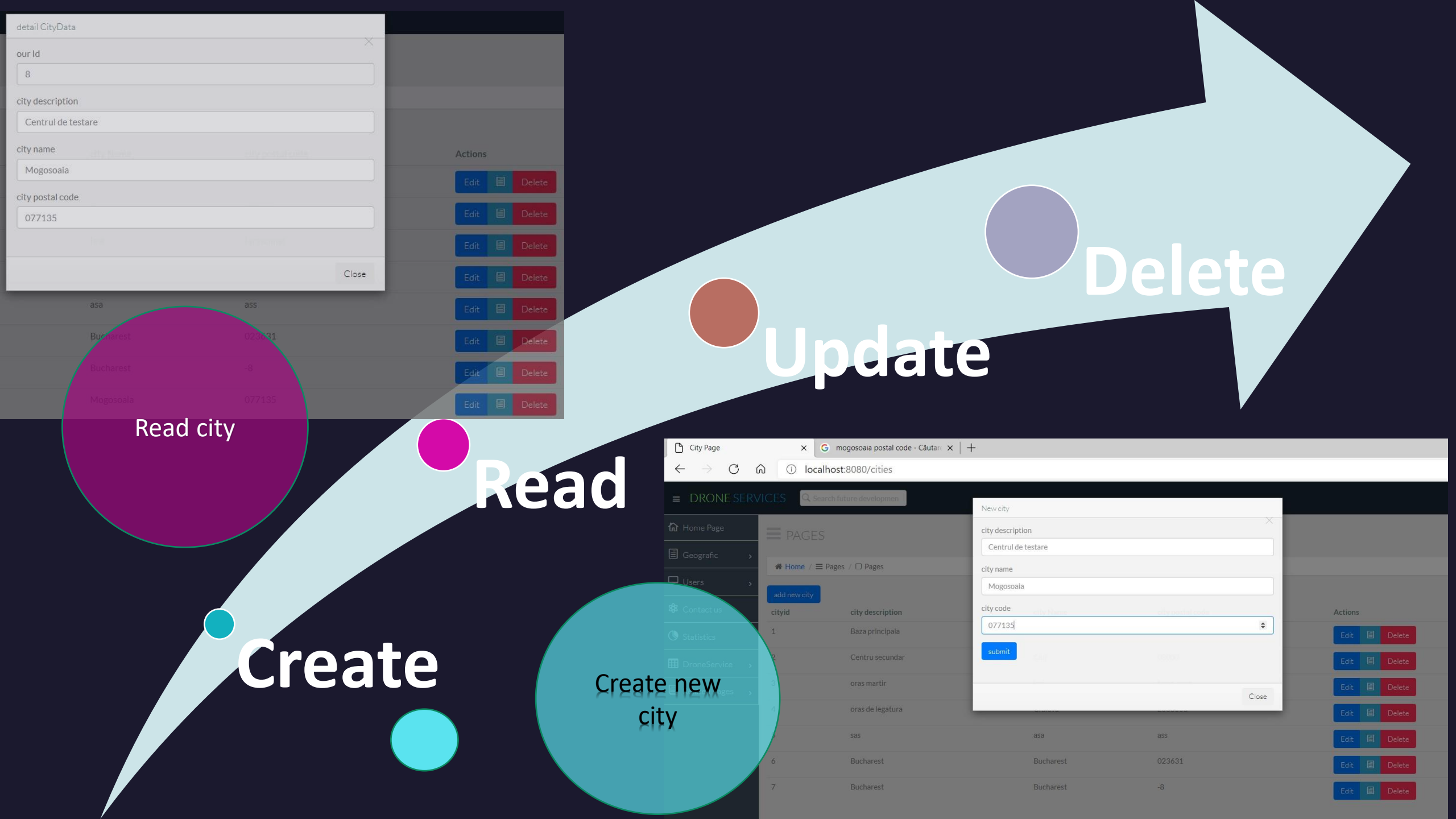
Structură Backend

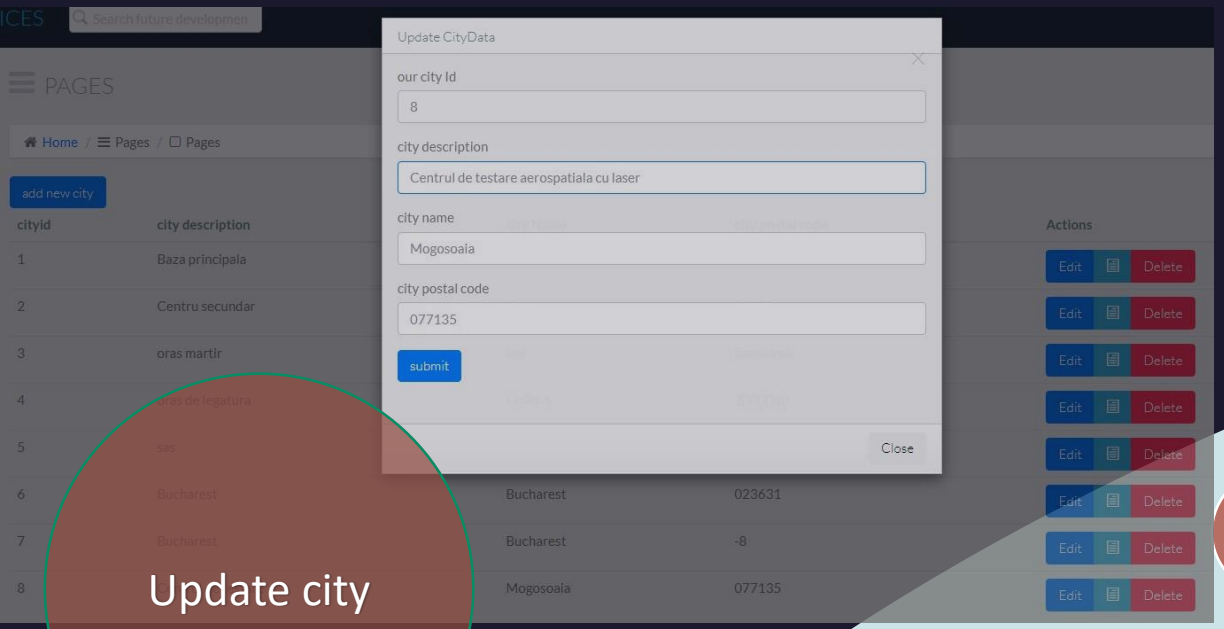


Arhitectură Backend



Frontend


detail CityData
our Id
8
city description
Centrul de testare
city name
Mogosoaia
city postal code
077135
Close
Actions
Edit
Delete
Read city
Read
Update
Delete
Create
Create new city
City Page
mogosoaia postal code - Căutare
localhost:8080/cities
DRONE SERVICES
Home Page
Geografic
Users
Contact us
Statistics
DroneService
PAGES
Home / Pages / Pages
add new city
cityid
city description
1
Baza principala
Centru secundar
oras martir
oras de legatura
sas
asa
ass
6
Bucharest
Bucharest
023631
7
Bucharest
Bucharest
-8
New city
city description
Centrul de testare
city name
Mogosoaia
city code
077135
submit
Close



Update city

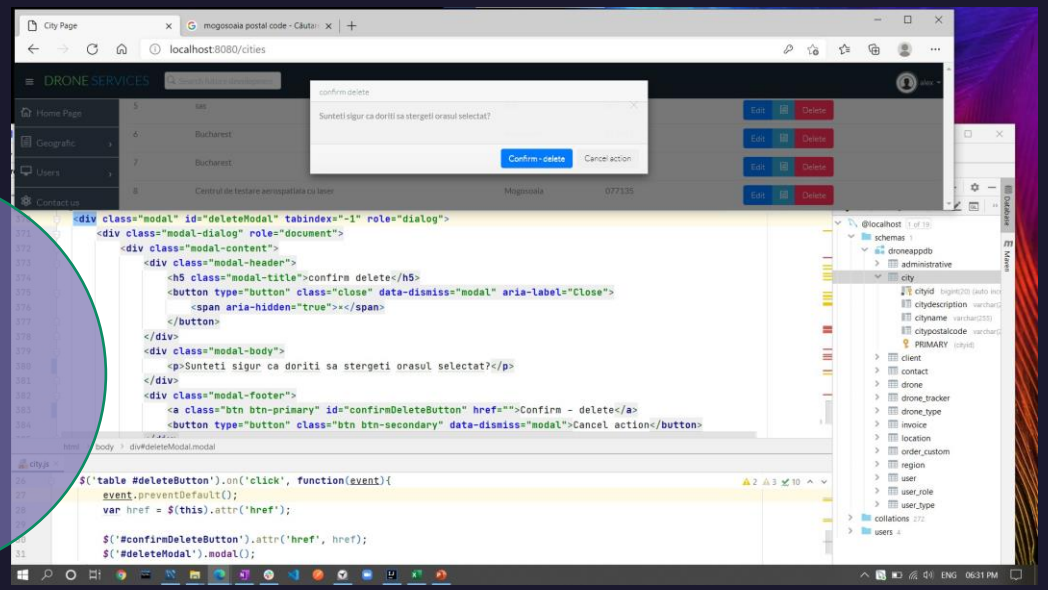
Read

Create

Delete

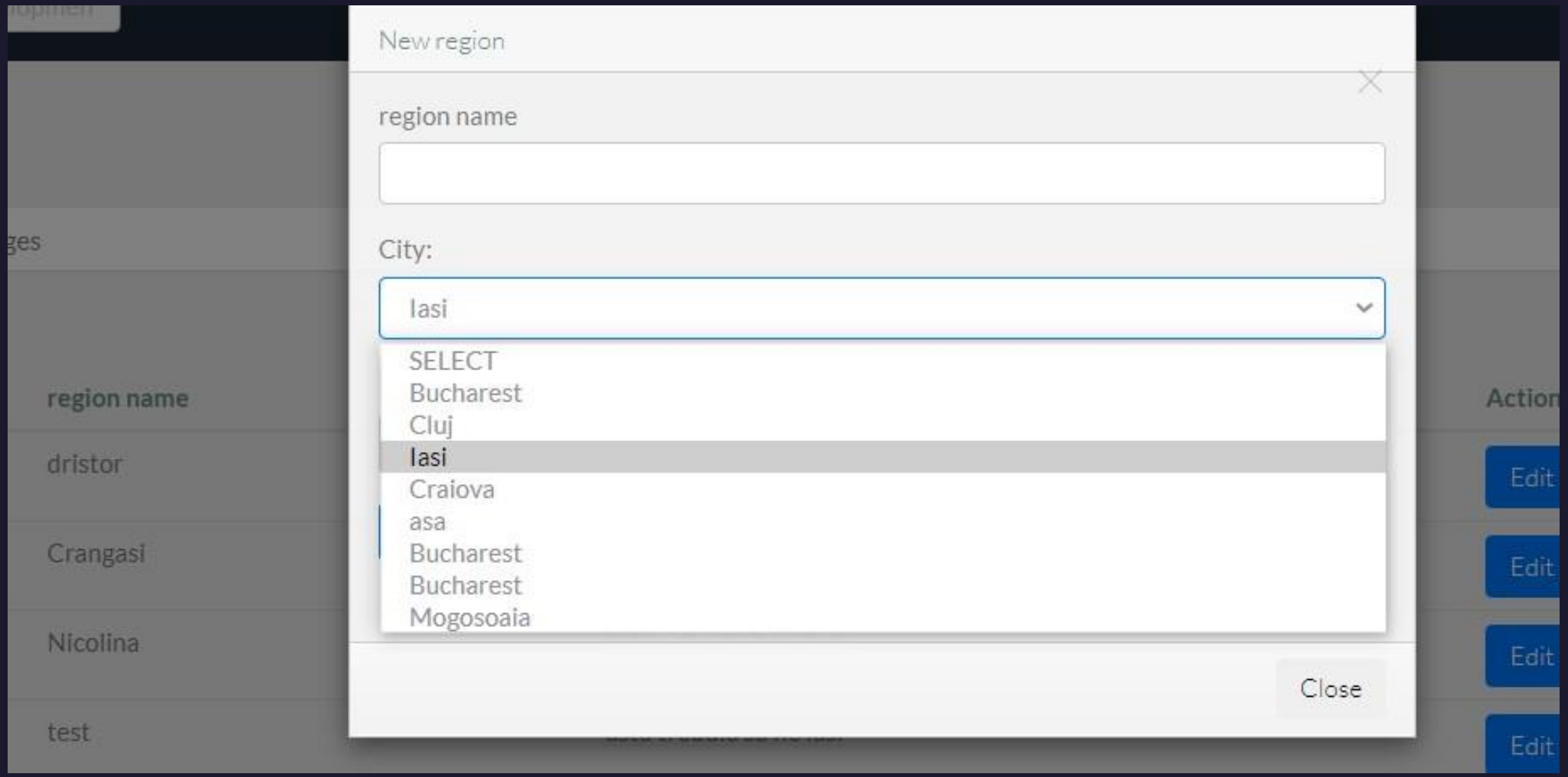
Update

Delete city



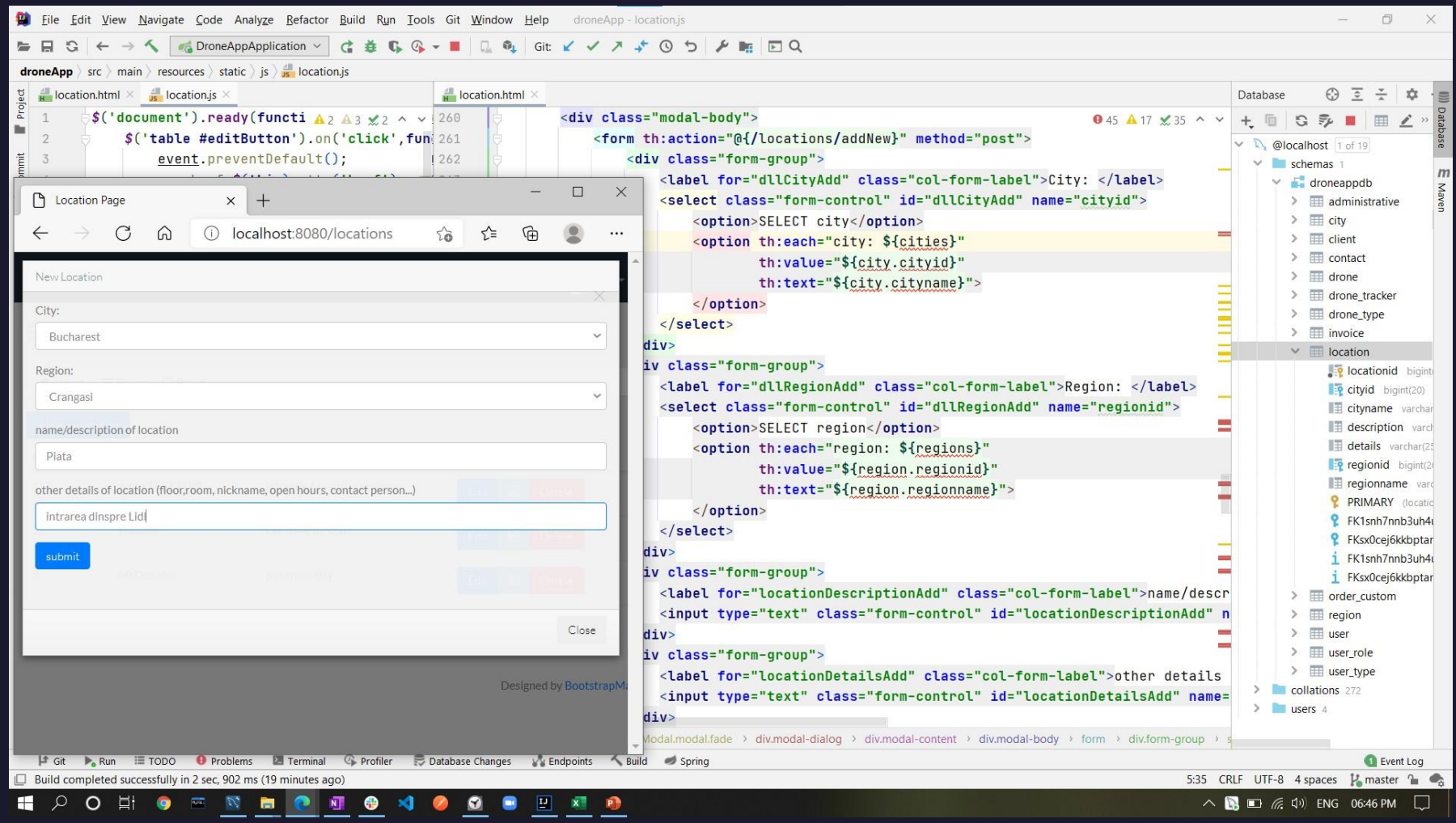
User Interface

PICK CITY FROM EXISTING LIST



User Interface

CREATE NEW LOCATION BY SELECTING EXISTING CITY & REGION



Baza de date

Baza de date MySQL

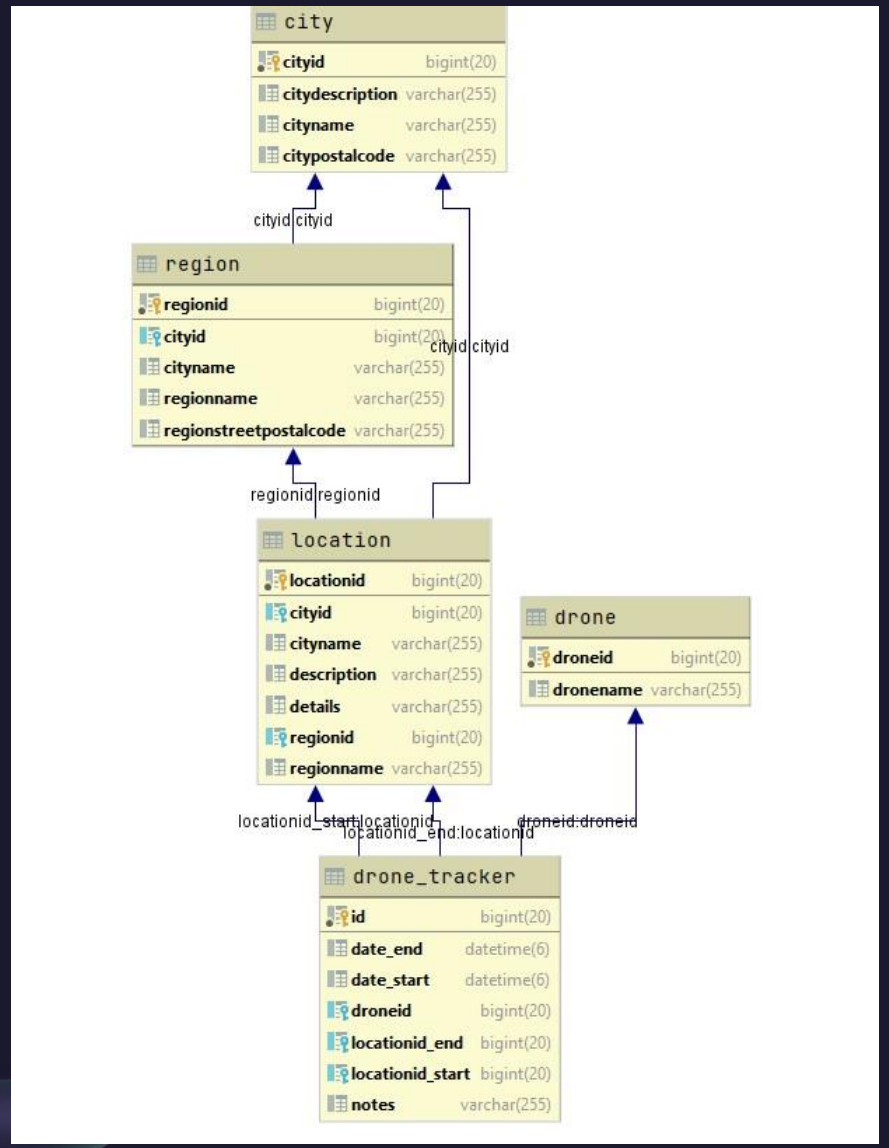


DIAGRAMA REGION/DRONE

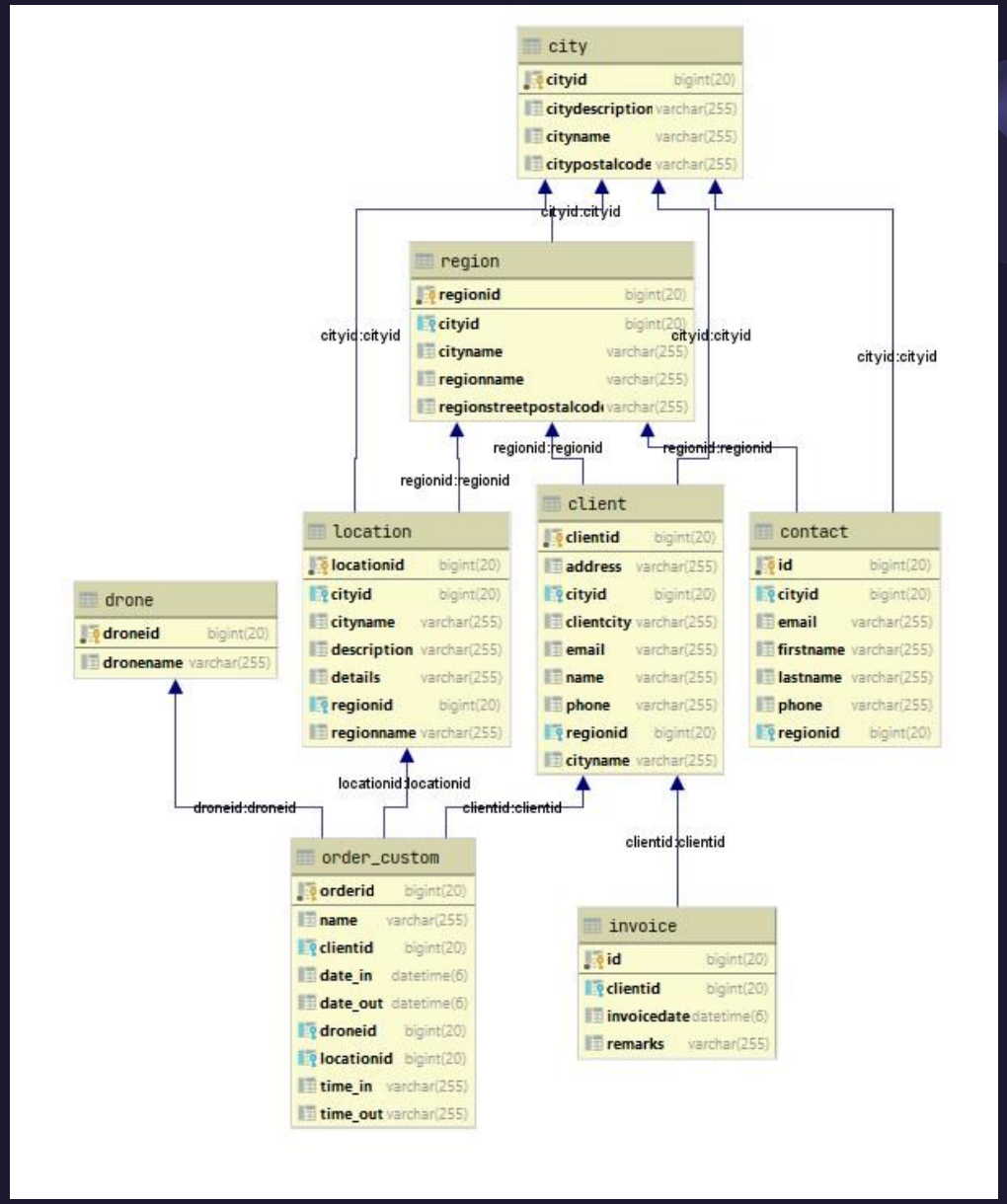
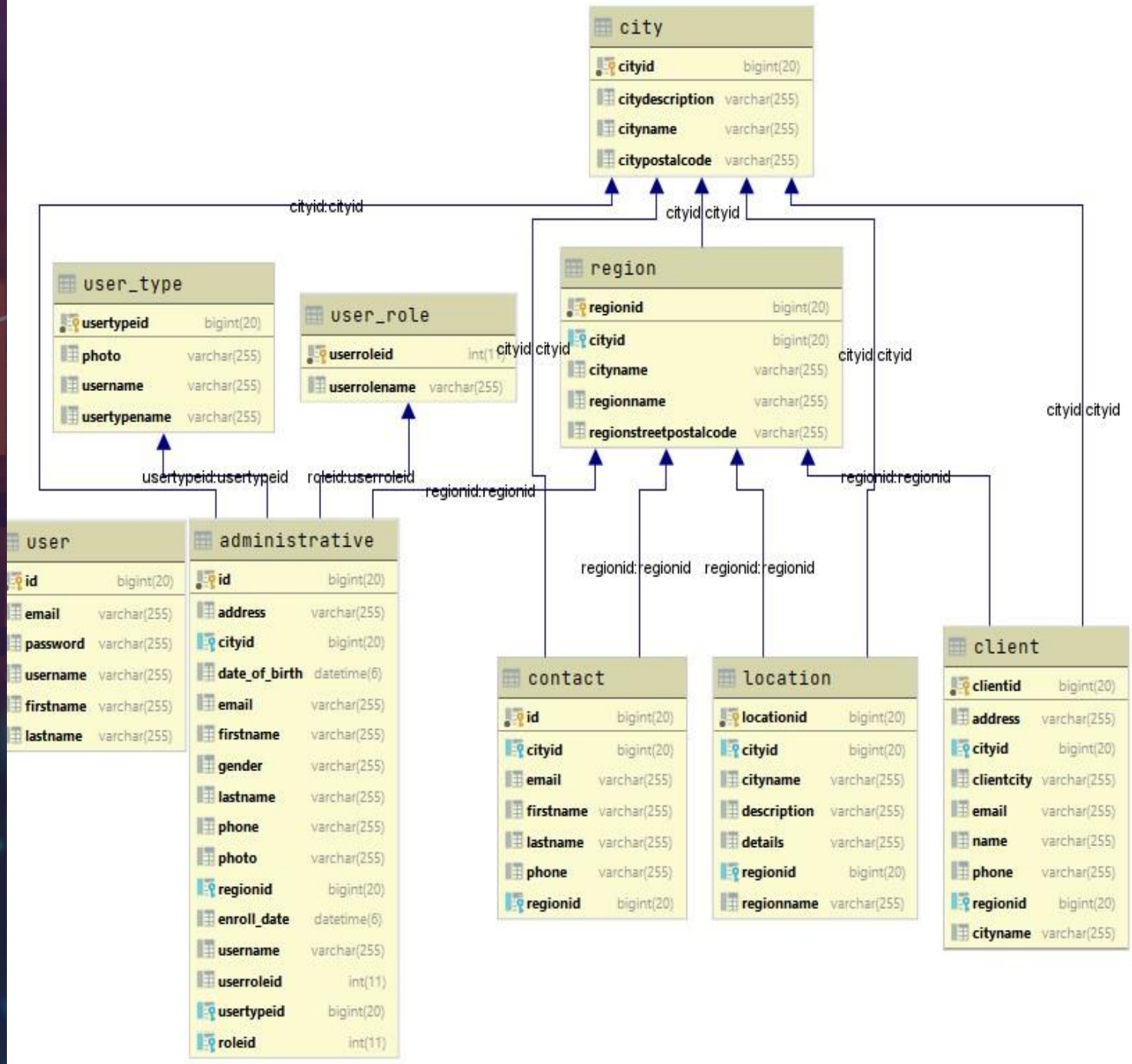


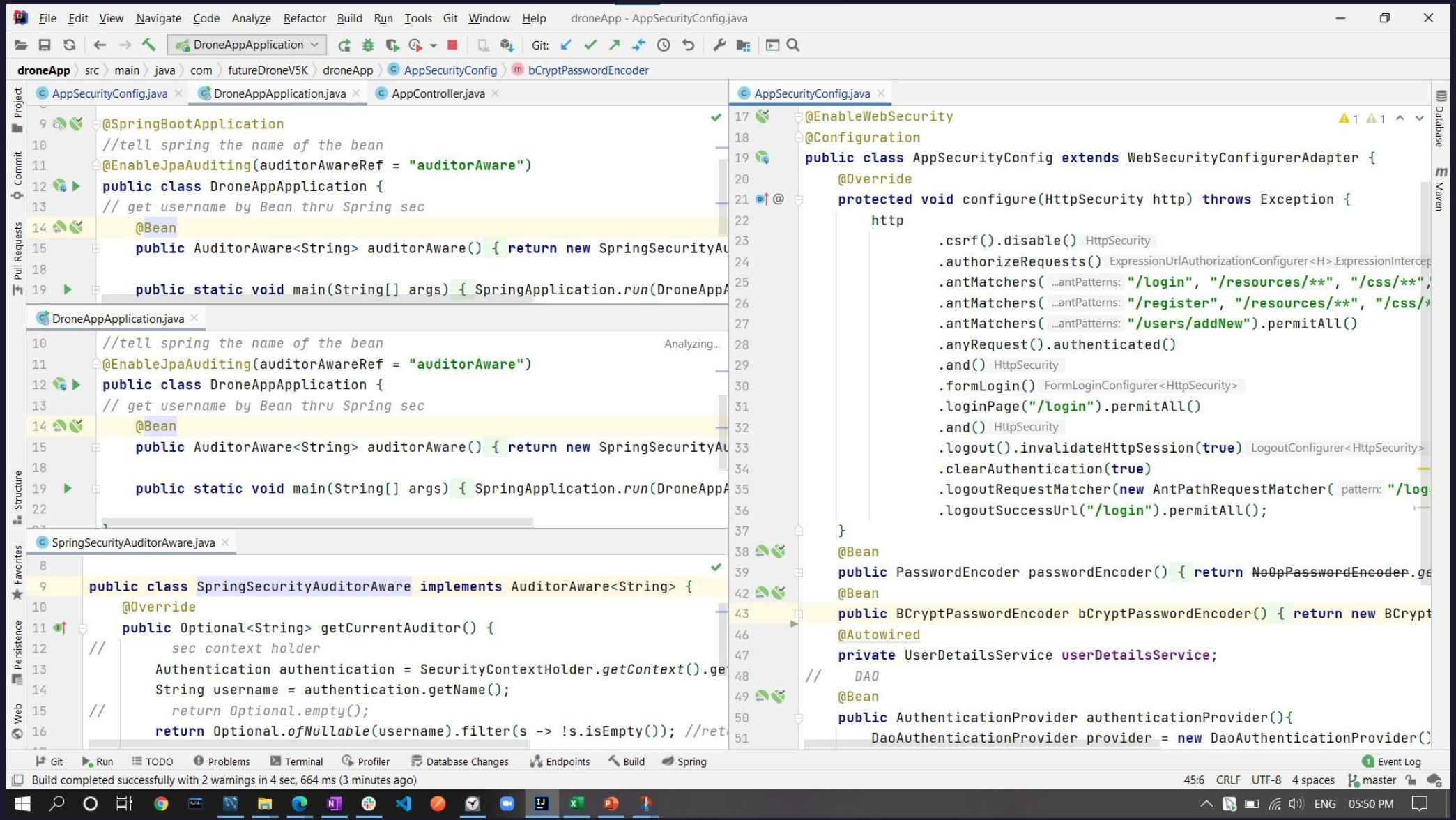
DIAGRAMA ORDER BY REGION

Baza de date MySQL

DIAGRAMA USER BY REGION



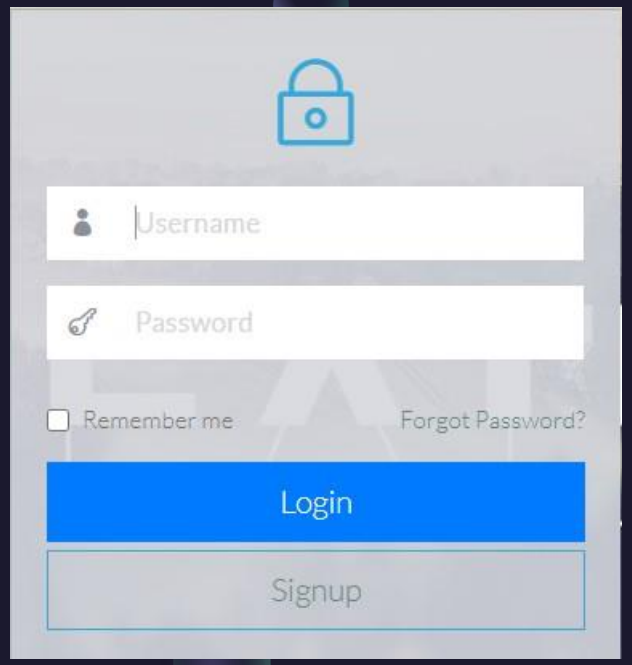
Elemente de securitate

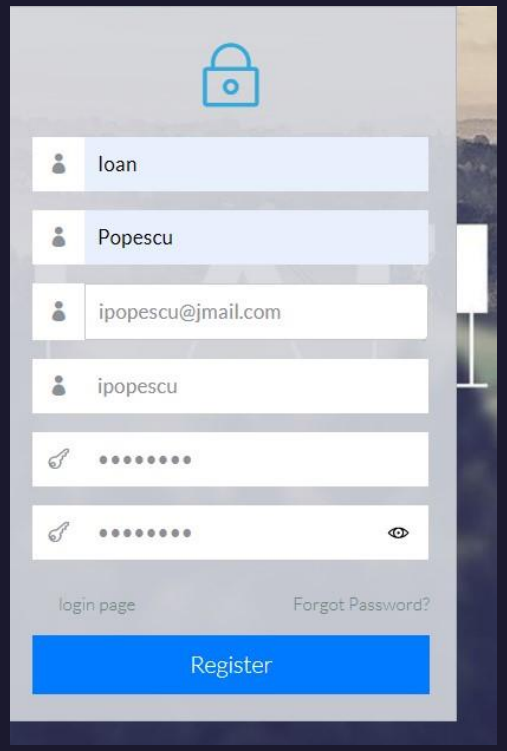

File Edit View Navigate Code Analyze Refactor Build Run Tools Git Window Help
droneApp - AppSecurityConfig.java
DroneAppApplication
droneApp src main java com futureDroneV5K droneApp AppSecurityConfig bCryptPasswordEncoder
AppSecurityConfig.java
DroneAppApplication.java
AppController.java
SpringSecurityAuditorAware.java
@SpringBootApplication
//tell spring the name of the bean
@EnableJpaAuditing(auditorAwareRef = "auditorAware")
public class DroneAppApplication {
// get username by Bean thru Spring sec
@Bean
public AuditorAware<String> auditorAware() { return new SpringSecurityAu
public static void main(String[] args) { SpringApplication.run(DroneAppA
Analyzing...
public class SpringSecurityAuditorAware implements AuditorAware<String> {
@Override
public Optional<String> getCurrentAuditor() {
sec context holder
Authentication authentication = SecurityContextHolder.getContext().ge
String username = authentication.getName();
return Optional.ofNullable(username).filter(s -> !s.isEmpty()); //ret
@EnableWebSecurity
@Configuration
public class AppSecurityConfig extends WebSecurityConfigurerAdapter {
@Override
protected void configure(HttpSecurity http) throws Exception {
http
.csrf().disable()
.authorizeRequests()
.antMatchers(...antPatterns: "/login", "/resources/**", "/css/**",
.antMatchers(...antPatterns: "/register", "/resources/**", "/css/*
.antMatchers(...antPatterns: "/users/addNew").permitAll()
.anyRequest().authenticated()
.and()
.formLogin()
.loginPage("/login").permitAll()
.and()
.logout().invalidateHttpSession(true)
.clearAuthentication(true)
.logoutRequestMatcher(new AntPathRequestMatcher(pattern: "/log
.logoutSuccessUrl("/login").permitAll();
@Bean
public PasswordEncoder passwordEncoder() { return NoOpPasswordEncoder.ge
@Bean
public BCryptPasswordEncoder bCryptPasswordEncoder() { return new BCrypt
@Autowired
private UserDetailsService userDetailsService;
DAO
@Bean
public AuthenticationProvider authenticationProvider(){
DaoAuthenticationProvider provider = new DaoAuthenticationProvider()
Git Run TODO Problems Terminal Profiler Database Changes Endpoints Build Spring
Event Log
Build completed successfully with 2 warnings in 4 sec, 664 ms (3 minutes ago)
45:6 CRLF UTF-8 4 spaces master
ENG 05:50 PM

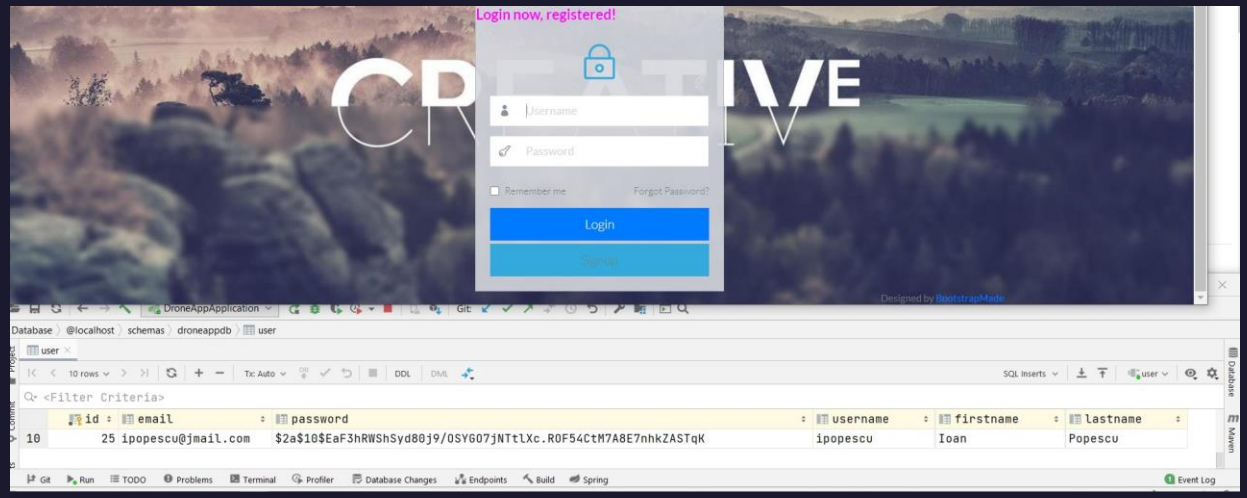
SIGN UP

REGISTER

REGISTRATION COMPLETE







Mulțumesc!

Eugeniu Orlov