# Streszczenie:

Przedmiotem niniejszej pracy inżynierskiej jest aplikacja mająca na celu ułatwienie wynajmowania wspólnych pokojów w domu studenta, oraz możliwość umieszczania ogłoszeń dla innych studentów mieszkającym w akademiku.

Zakres pracy dyplomowej zawiera część teoretyczną zawierającą opis poszczególnych narzędzi oraz technologii użytych przy pisaniu aplikacji oraz tą, w której jest opisana implementacja aplikacji, która się dzieli na część kliencką oraz serwerową. Część kliencka jest napisana na platformę Android, natomiast część serwerowa jest napisana w języku Java używając popularnych praktyk stosowanych w dużych projektach informatycznych.

Słowa kluczowe: aplikacja mobilna, rezerwacja, akademik, Java, Android

# Abstract:

The subject of this thesis is a platform aimed at facilitating the renting of shared rooms of dormitory, and possibility of placing advertisements for other students living in dormitory.

The scope of this thesis includes theoretical part containing description of the tools and technologies used while writing the application and one containing implementation part, which splits into client part and server one. The client part is written for the Android platform, while the server part is written in Java using popular practices used in large IT projects.

Keywords: mobile application, booking, dormitory, Java, Android

# **Spis treści:**

1. **Wstęp**
   1. Cele projektu
2. **Narzędzia i technologie**
   1. Java
   2. Android
   3. IntelliJ
   4. Android Studio
   5. Apache Tomcat
   6. Apache Maven I Gradle
   7. MySQL
   8. Lombok
   9. Ngrok
   10. Git VCS
   11. REST
   12. Spring
       1. Spring Boot
       2. Spring Security
3. **Implementacja**
   1. Część serwerowa
   2. Część kliencka

# Wstęp:

Kiedyś trzeba było osobiście wybrać się do banku, by wykonać przelew. Później można było zostać w domu i zalogować się na stronę banku by to zrobić. Dziś wystarczy zainstalować aplikację swojego banku na telefonie i możemy zrealizować przelew z dowolnego miejsca, czy to z autobusu jadąc do pracy, czy na plaży będąc na urlopie. Dwóch na trzech ludzi na świecie posiada smartfona, a w krajach rozwiniętych właścicielami smartfonów jest nawet 90 procent społeczeństwa. Z tych wszystkich telefonów, ponad 80 procent posiada system Android a pozostałą większość to telefony z systemem iOS. Biorąc te fakty pod uwagę coraz więcej firm produkuje aplikacje na telefon. Jest to ogromny rynek i ludzie chętnie korzystają z nowych sposobów na załatwianie spraw zdalnie.

Dom Studencki, potocznie akademik, posiada różne wspólne salki, jak salka telewizyjna, bilard, ping-pong, siłownia czy pralnia. Klucze do tych salek są w różnych pokojach, a czasem nawet w kilku. By wynająć salkę trzeba najpierw sprawdzić czy jest wolna, potem iść na parter zobaczyć gdzie jest klucz, iść po klucz, i z powrotem wrócić do salki. Stąd wziąłem inspirację na moją aplikację, bo mieszkając w akademiku już męczyło mnie bieganie po piętrach. Teraz można sprawdzić wszystko w aplikacji, a nawet zarezerwować wstępnie salkę na siebie. Oprócz tego aplikacja posiada tablicę ogłoszeń, jako formę komunikacji z pozostałymi mieszkańcami domu studenta.

# Cele pracy:

Głównym celem pracy dyplomowej było napisanie aplikacji mobilnej zatytułowanej Mobilny Informator Akademicki, która wykorzystując napisany przeze mnie również serwer, ułatwiałaby wymianę informacji pomiędzy studentami zamieszkującymi dom studenta w postaci ogłoszeń, jak i również łatwy dostęp do informacji o tym, czy ogólnodostępne salki w domu studenta są aktualnie zajęte czy nie, oraz gdzie się znajdują, i gdzie szukać klucza. W tym przypadku zamiarem było również zapoznanie się z tworzeniem rozwiązań dla aplikacji mobilnych pod systemem Android.

Ważnym celem tej pracy było również, zapoznanie się z dobrymi praktykami stosowanymi w dużych projektach IT, między innymi sposobem rozwijania oprogramowania używając Spring Framework w celu udostępniania sieciowego serwisu REST.

# Narzędzia i technologie

## Java

Java jest to zorientowany obiektowo język programowania początkowo wypuszczony przez amerykańską firmę Sun Microsystems w 1995 roku, później kupioną przez Oracle Corporation. Pisząc aplikację, kod nie jest kompilowany wprost do kodu maszynowego, lecz do tzw. kodu bajtowego Java inaczej nazwany kodem pośrednim, który jest interpretowany i wykonywany przez Java Virtual Machine (JVM). Wirtualna maszyna Java jest uniwersalna dzięki niezależności od budowy procesora, na którym działa i od systemu operacyjnego. Tłumaczy ona program napisany w Javie na kod maszynowy.

## Android

Android jest to system operacyjny początkowo rozwinięty przez firmę Android, później przez Google, rozprowadzany na licencji GNU. Jako otwarte oprogramowanie oparte na zmodyfikowanej wersji jądra systemu Linux oraz wirtualnej maszynie Javy jest przeznaczone na dotykowe urządzenia mobilne tj. smartfony lub tablety. Warto wspomnieć, że Google rozwinęło również wersje Androida dla telewizorów – Android TV, samochodów – Android Auto, oraz smartwatchy – Wear OS.

## IntelliJ

IntelliJ jest to zintegrowane środowisko programistyczne, inaczej IDE (od ang. integrated development environment) rozwijane przez firmę JetBrains. Narzędzie to jest nastawione na rozwijanie oprogramowania w języku Java, a co za tym idzie zapewnia świetną integrację z wieloma innymi narzędziami pomagającymi przy rozwijaniu oprogramowania np. Maven, Gradle, Lombok, Git.

## Android Studio

Drugim narzędziem programistycznym, użytym przy tworzeniu aplikacji jest Android Studio, które jest zbudowane na podstawie IntelliJ od JetBrains. Jest zaprojektowane specjalnie na potrzeby rozwoju aplikacji mobilnych na platformę Android. Android Studio posiada wiele udogodnień takich jak edytor interfejsu graficznego typu „drag and drop”, integracja z narzędziem Gradle będącym domyślnym sposobem budowania projektu, wbudowane szablony, komponenty oraz emulatory telefonów lub tabletów.

## Apache Tomcat

Apache Tomcat to kontener aplikacji webowych o otwartym kodzie źródłowym, stworzony przez Apache Software Foundation. Oferuje najbardziej podstawowe funkcjonalności, ale wystarczające do uruchomienia serwera. Szybko się ładuje i ponownie uruchamia, a generacja pliku \*.war i ładowanie go na serwer jest automatyczne, dzięki czemu osiągamy szybki cykl deweloperski.

## Apache Maven i Gradle

Oba te narzędzia służą automatyzacji budowy projektów na platformę Java. Są łatwe w obsłudze i bardzo przyspieszają pracę nad projektami.

* Gradle jest narzędziem zbudowanym na koncepcie Apache Ant i Apache Maven, lecz nie używa języka XML, ale języka opartego o Groovy. Jest standardowym narzędziem automatyzacji budowy oprogramowania na urządzenia mobilne na platformę Android.
* Maven jest narzędziem używającym pliku POM (*ang. Project Object Model*), który jest napisany w języku XML i który opisuje to jak projekt ma być budowany oraz to, jakie zależności są potrzebne do poprawnego zbudowania projektu. Zależności te są pobierane z repozytorium mavena przy każdym uruchomieniu budowania projektu.

## MySQL

MySQL jest relacyjnym systemem zarządzania bazą danych o otwartym kodzie źródłowym stworzony przez MySQL AB, które było kupione przez Sun Microsystems w 2008 roku. Aktualnie jest rozwijanie przez Oracle Corporation, które kupiło Sun Microsystems w 2010.

## Lombok

Lombok jest biblioteką Javy, która automatycznie może generować tzw. „boilerplate code”, czyli kod, który nie realizuje żadnych funkcji biznesowych i często jest generowany przez środowisko lub inne narzędzia. w aplikacjach najczęściej takim kodem są metody hashCode, equals, toString oraz gettery i settery. Zamiast zmniejszać czytelność naszej aplikacji takim kodem, możemy użyć odpowiednich adnotacji takich jak @Getter, @Setter, czy @Data by wygenerować potencjalnie potrzebne przez nas metody w momencie kompilacji.

## Ngrok

Ngrok jest narzędziem, które wystawia lokalne serwery do publicznego Internetu przez bezpieczne tunele. Dzięki temu, nie ma potrzeby hostować serwera na publicznej domenie, a zamiast tego aplikacja mobilna może się połączyć przez Internet do localhosta.

* 1. REST

REST (*ang. Representional State Transfer*) jest wzorcem architektury danych dla rozproszonych internetowych aplikacji wprowadzony i zdefiniowany w 2000r. przez Roya Fieldlinga. Jest popularny ze względu na jego prostotę i faktu, że jest zbudowany na istniejących już właściwościach protokołu HTTP. Charakteryzuje się tym, że dane są przedstawione w formie tekstowej, a używanie tego wzorca jest niezależne od użytego języka programowania.

Wyróżniamy pięć możliwych operacji przy użyciu protokołu HTTP:

* GET – pobieranie danych wskazanych przez adres z bazy danych
* PUT – przyjmowanie danych od klienta i ich aktualizacja w bazie danych, Operacja idempotentna.
* PATCH – przyjmowanie danych od klienta i aktualizowanie jednego z pól encji w bazie danych. Operacja ta nie jest idempotentna.
* POST – przyjmowanie danych od klienta i ich zapisywanie w bazie danych. Operacja ta nie jest idempotentna.
* DELETE – żądanie usunięcia konkretnych danych

Po wykonywaniu operacji stany kodu HTTP wskazują na wynik żądania:

* 1XX – informacja
* 2XX – sukces
* 3XX – przekierowanie
* 4XX – błąd klienta
* 5XX błąd serwera
  1. Spring

Spring to wielowarstwowy szkielet tworzenia aplikacji o otwartym kodzie źródłowym w języku Java za pomocą którego możemy rozwijać aplikacje używając odwróconego sterowania znanego również jako wstrzykiwanie zależności. Swoje aspoczątki szkielet ten zawdzięcza człowiekowi Rod Johnson, który w swojej książce zatytułowanej „Expert One-on-One J2EE Design and Development” z października 2002 roku omawia, jak można usprawnić rozwój oprogramowania na platformę Java, Enterprise Edition. Szkielet ten składa się z wielu modułów, które dostarczają szeroki wachlarz usług, z których najważniejsze moim zdaniem to:

* Kontener główny

Jest to centralny moduł szkieletu, którego główną zaletą jest *BeanFactory*, który kontroluje sposób tworzenia komponentów oraz *ApplicationContext,* który pomaga w łatwiejszej integracji z Spring’ową funkcjonalnością programowania aspektowego, zarządzania zasobami wiadomości oraz publikacją wydarzeń.

* Programowanie aspektowe

Moduł ten bazuje na integracji z językiem AspectJ, który pozwala na implementację zagadnień przekrojowych. Dzięki niemu możemy dodatkowo myśleć o strukturze programu w sposób aspektowy a nie obiektowy. Aspekt to zagadnienie, które przecina w poprzek klasy obiektowe i aplikuje swoją funkcjonalność. Dobrym przykładem zagadnienia może być transakcyjność albo logowanie.

* Dostęp do danych/Integracja

Moduł ten bardzo upraszcza dostęp do danych w bazie danych, dzięki integracji z wieloma technologiami takimi jak Hibernate, Java Persistance API albo Java DataBase Connectivity. Pozwala na pominięcie pisania kodu, który jest odpowiedzialny za logikę dostępu do danych oraz metod mapowania obiektów z bazy danych na model w naszej aplikacji.

Moduł ten zapewnia również abstrakcyjność transakcji oraz je koordynuje przekrojowo przez całą aplikację.

* Spring MVC

Szkielet Spring Model-Widok-Kontroler wprowadza rozdzielenie aplikacji na 3 różne części.

Model – reprezentacja danych w postaci klas obiektów naszej aplikacji

Widok – klasy odpowiedzialne za wyświetlanie treści interfejsu użytkownika

Kontroler – klasy odpowiedzialne za logikę aplikacji, przyjmują dane, przetwarzają i wysyłają do widoku

W przypadku

* + 1. Spring Boot

Spring Boot jest sposobem na tworzenie autonomicznych webowych aplikacji z możliwie minimalną lub zerową konfiguracją. Jest zbudowany na szkielecie Spring Zapewnia ustawienia domyślne dla konfiguracji kodu oraz zestaw plików budujących pom lub gradle, a także wbudowany serwer Tomcat. Dzięki niemu można wygenerować projekt, który ma już wszystkie potrzebne zależności by po prostu go uruchomić.

# Implementacja

* 1. Część serwera

Ta aplikacja została stworzona przy użyciu Spring Boot 2.1, a jej celem jest odpytywanie bazy danych w celu zwrócenia informacji do aplikacji mobilnej przez protokół HTTP.