# Web based Estate Agent

Author:Paweł Otrębski
Supervisor: Dr Wojciech Macyna

Wroclaw University of Technology

# Introduction

**Aim**

# Introduction

**Aim**

- Advertise sundry living space

# Introduction

**Aim**

- Advertise sundry living space
- Expose offers to users

# Introduction

**Aim**

- Advertise sundry living space
- Expose offers to users
- Rudimentary management for clients and admins

# Introduction

**Aim**

- Advertise sundry living space
- Expose offers to users
- Rudimentary management for clients and admins
  - bookings
  - confirmations
  - cancellations

# Introduction

**Aim**

- Advertise sundry living space
- Expose offers to users
- Rudimentary management for clients and admins
  - bookings
  - confirmations
  - cancellations
- Rate experience minimize/mitigate spamming

# Introduction

**Aim**

- Advertise sundry living space
- Expose offers to users
- Rudimentary management for clients and admins
  - bookings
  - confirmations
  - cancellations
- Rate experience minimize/mitigate spamming
- API that is front end agnostic

# Motivation

**Key motivations**

- sorting by price
- avoid formal accommodation services
- build a personal relationship
- client feedback
- rate a client

# Users and Use cases

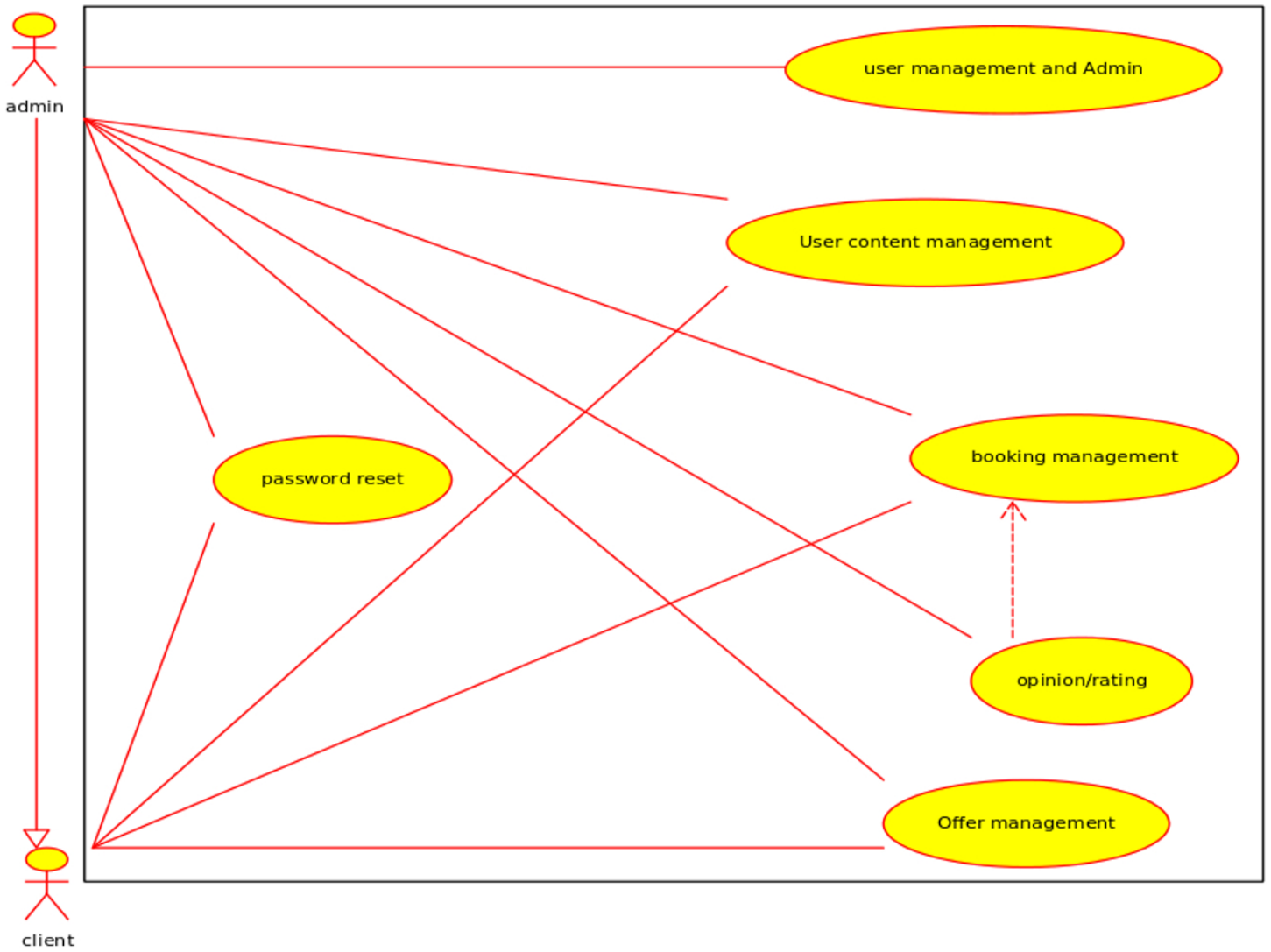
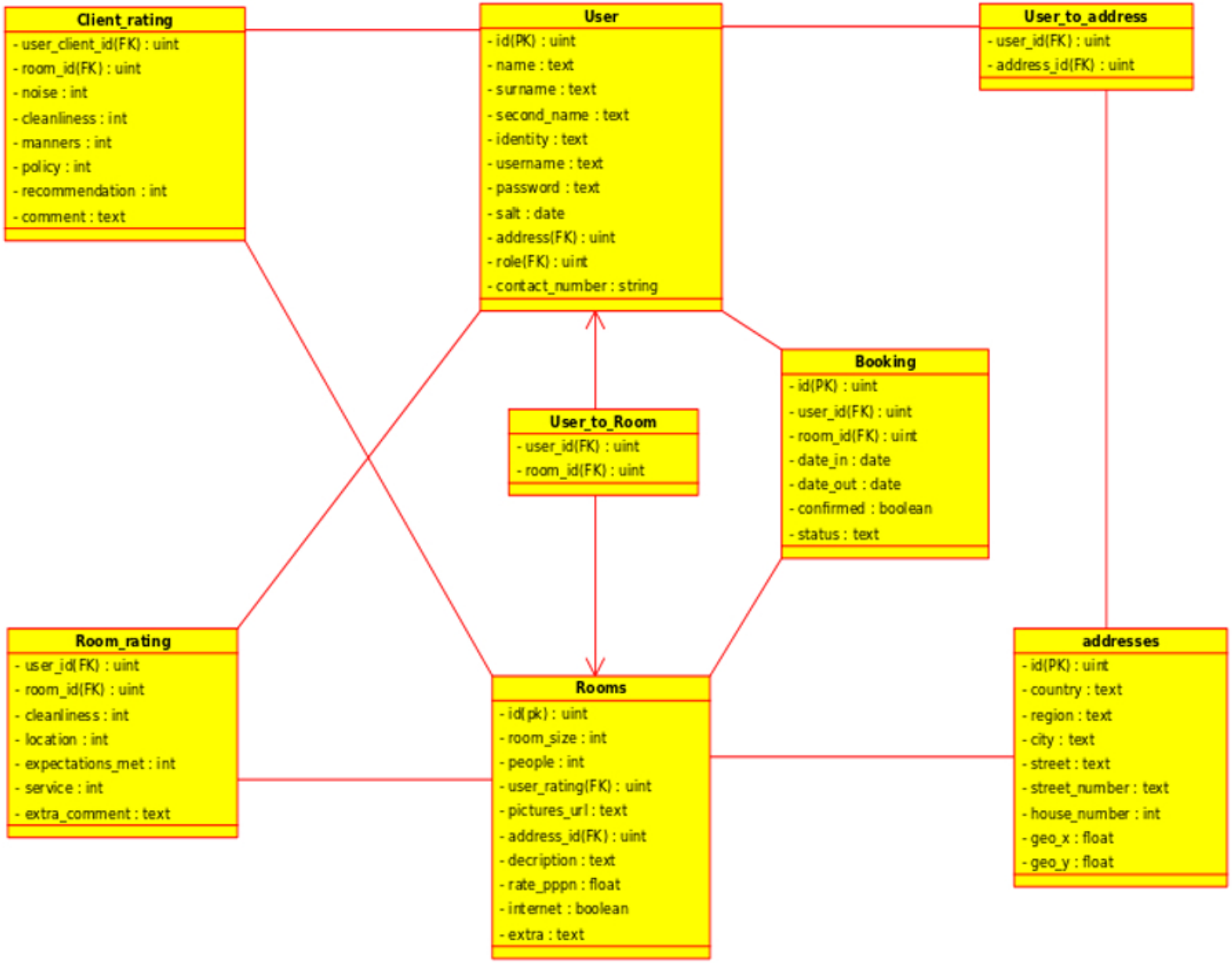



Figure : use case diagram in UML

# Database Design



# Architecture

**Two Tiered Architecture(hopefully three)**

# Architecture

**Two Tiered Architecture(hopefully three)**

- Database server

# Architecture

**Two Tiered Architecture(hopefully three)**

- Database server
- application server and WWW server

**Design Patterns**

# Architecture

**Two Tiered Architecture(hopefully three)**

- Database server
- application server and WWW server

**Design Patterns**

- Model-View-Controller(MVC)

# Architecture

**Two Tiered Architecture(hopefully three)**

- Database server
- application server and WWW server

**Design Patterns**

- Model-View-Controller(MVC)
- underlying design patterns thanks to framework

**Protocols and format**

# Architecture

**Two Tiered Architecture(hopefully three)**

- Database server
- application server and WWW server

**Design Patterns**

- Model-View-Controller(MVC)
- underlying design patterns thanks to framework

**Protocols and format**

- **Re**presentational **S**tate **T**ransfer