

A horizontal row of twelve empty square boxes, intended for children to draw or color in.

□□□XXX□□□□□□□□

□□□XXX□□ □□□XXX □□□XXXXXXXX

□□□□□XXX □□□XXX

□□□□□2026 □ X □ X □

1

“” Vue 3 + Vite + Element Plus Spring Boot MyBatis-Plus /

Spring Boot + Vue 3

Abstract

This thesis presents the design and implementation of a homestay recommendation system. The frontend is built with Vue 3, Vite, and Element Plus, while the backend leverages Spring Boot and MyBatis-Plus, combining collaborative filtering and content-based recommendation to provide personalized listings, online booking, and host property management. The work covers background, requirement analysis, system architecture, key technologies, database and process design, implementation, and testing.

Keywords: Homestay recommendation; Personalized recommendation; Spring Boot; Vue 3; Hybrid recommender

1

1. □□
 2. □□□□□□□□□□
 3. □□□□
 4. □□□□
 5. □□□□
 6. □□□□□□ER □□
 7. □□□□□□□□□□□□
 8. □□□□□□□□
 9. □□□□□
 10. □□□□□
 11. □□□□
 12. □□

1. 项目

1.1 项目概述

本项目是一个基于Spring Boot + Vue 3的全栈项目，主要功能包括用户管理、订单管理、商品管理等。项目采用前后端分离架构，前后端通过API进行交互。

后端主要使用Spring Boot框架，结合MyBatis-Plus进行数据库操作。前端使用Vue 3框架，结合Element Plus进行UI设计。项目还使用Axios进行HTTP请求，Lombok进行简化代码，Spring Validation进行表单校验。

项目“前后端分离”的优势在于前后端逻辑分离，易于维护和扩展。同时，通过API接口，可以轻松地将前端部署在不同的平台上。

项目借鉴了Airbnb的技术栈，使用JWT进行身份验证。同时，为了保证系统的高可用性和稳定性，项目还实现了QPS限流和Redis缓存。

项目的主要模块包括用户管理、商品管理、订单管理等。通过合理的权限控制，确保了系统的安全性和稳定性。

项目整体架构清晰，易于理解。通过前后端分离，提高了项目的可维护性和可扩展性。

1.2 项目需求

- 用户管理：包括用户注册、登录、密码找回等功能。
- 商品管理：包括商品添加、修改、删除、查询等功能。
- 订单管理：包括订单创建、修改、删除、查询等功能。

1.3 项目架构

项目整体架构清晰，易于理解。通过前后端分离，提高了项目的可维护性和可扩展性。

2. 技术栈

- 前后端分离：Vue 3 + Vite + Element Plus + Pinia + Vue Router + Axios
- 后端框架：Spring Boot + MyBatis-Plus + JWT + Spring Validation + Lombok
- 数据库：MySQL + Redis
- 缓存：Redis

3.

- **HTTP**/HTML/**JSON**/**XML**/**REST API**
 - **HTTP**/HTML/**JSON**/**XML**
 - **HTTP**/HTML/**JSON**
 - **HTTP**/**JWT** **REST API**

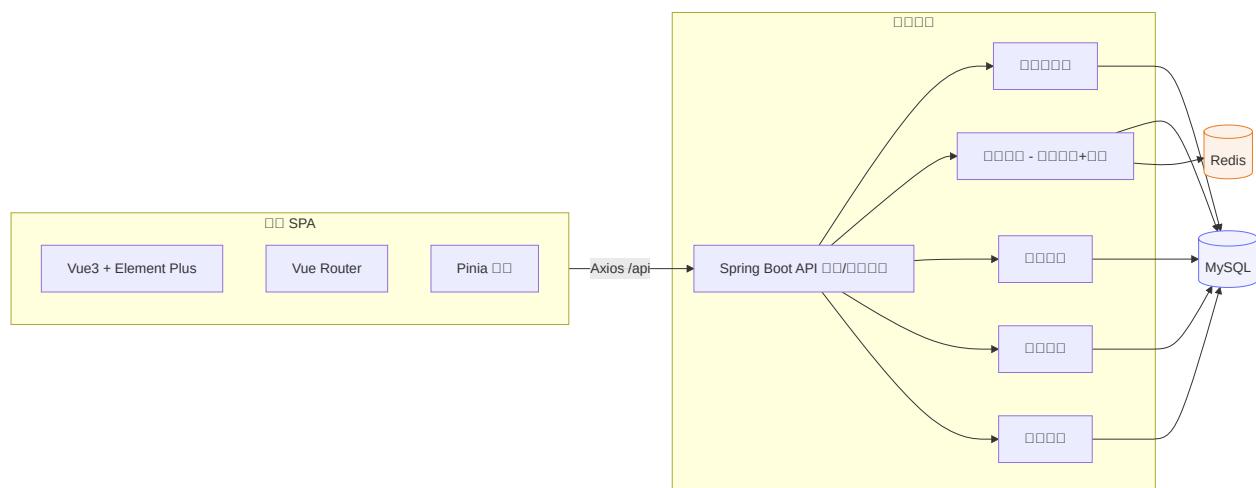
3.1 □□□□□□□

3.2 □□□□□

- JWT JSON Web Token
 - JSON Web Token / JSON Web Signature
 - JSON Web Key
 - JSON Web Encryption

3.3 □□□□□

4.



4.1 □□□□□

```

flowchart LR
    subgraph Client[SPA]
        UI[Vue3 + Element Plus]
        Router[Vue Router]
        Store[Pinia]
    end

    subgraph Backend[Backend]
        APIGW[Spring Boot API]
        Auth
        Rec
        Order
        Property
        UserSvc
    end

    DB[(MySQL)]:db
    Cache[(Redis)]:cache

    Client -->|Axios /api| APIGW
    APIGW --> Auth
    APIGW --> Rec
    APIGW --> Order
    APIGW --> Property
    APIGW --> UserSvc
    Auth --> DB
    Rec --> DB
    Order --> DB
    Property --> DB
    UserSvc --> DB
    Rec --> Cache
    classDef db fill:#f2f2ff,stroke:#6370f4;
    classDef cache fill:#fdf2e9,stroke:#e67e22;

```

4.2 前端

```

flowchart TB
    View[Vue3 + Element Plus] --> BFF[Axios + MyBatis-Plus]
    BFF --> Ctrl[Controller]
    Ctrl --> Service[Service]
    Service --> Mapper[MyBatis-Plus]
    Mapper --> DB[(MySQL)]
    Service --> RecCore[RecCore]
    RecCore --> Cache[(Redis)]

```

```
subgraph Infra[infra]
    Security[JWT + Spring Security]
    Validation[validation]
    Logging[logging]
end
Ctrl --> Infra
```

5. 安全性

- 使用JWT进行用户/房东/管理员身份验证
- 对所有敏感操作进行多因素认证/双因素认证
- 定期轮换密钥/证书/访问令牌
- 实施强密码策略
- 限制IP地址/地理位置访问
- 实施速率限制60% + 40% A/B测试/地理分发

安全性模块负责处理“用户/房东/管理员”角色的前端逻辑和安全措施。

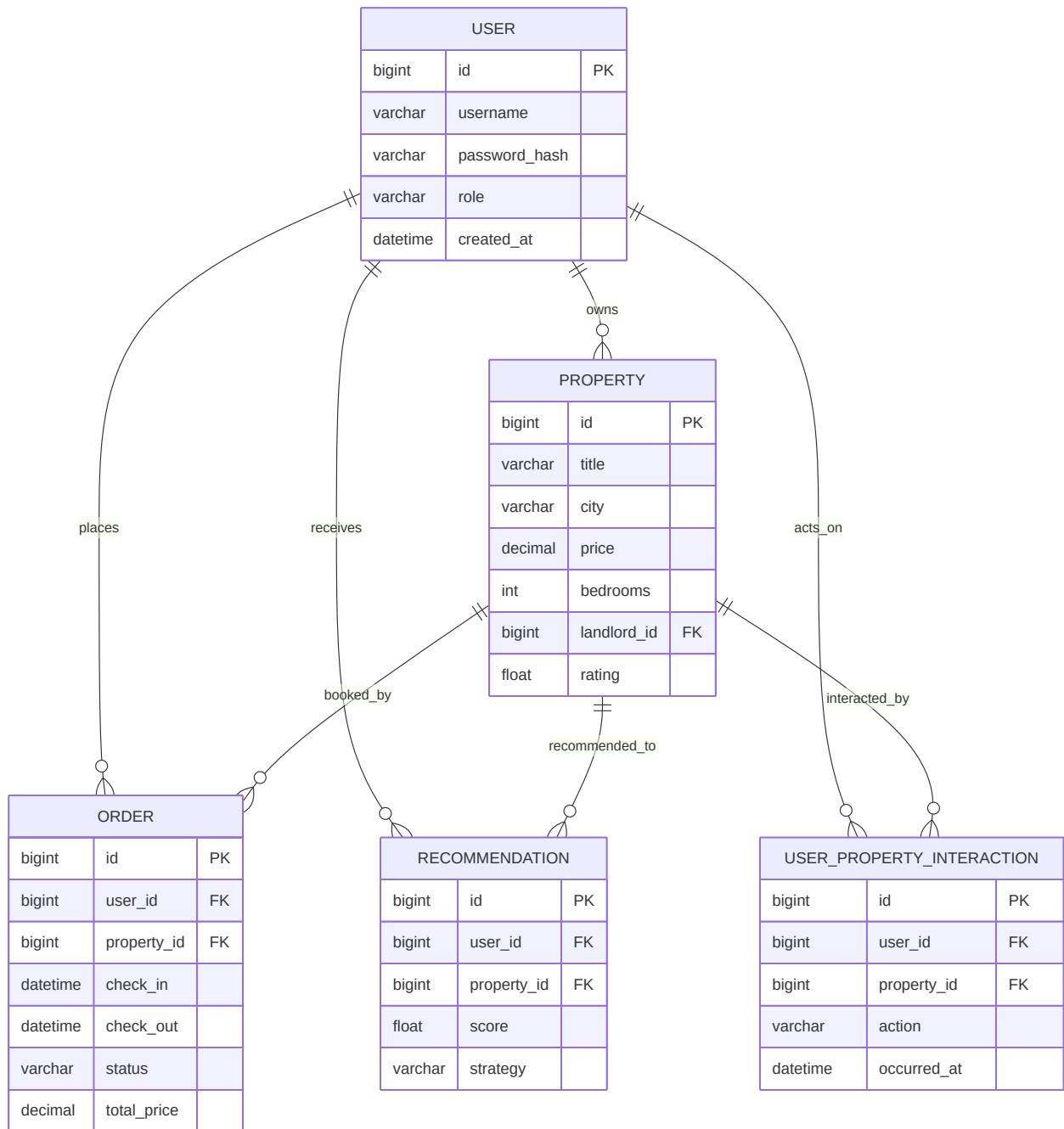
5.1 角色管理

- 管理USER/房东/管理员账户的创建、更新和删除
- 管理LANDLORD/管理员账户的创建、更新和删除
- 管理ADMIN/管理员账户的创建、更新和删除

5.2 策略管理

- 实施基于角色的访问控制-Top-N策略
 - 实施基于地理位置的访问控制
 - 实施0.6权重 + 0.4权重的A/B测试/地理位置
 - 实施速率限制/地理位置分发“XX/YY”
-

6. 数据模型ER图



erDiagram

```

USER {
    bigint id PK
    varchar username
    varchar password_hash
    varchar role
    datetime created_at
}

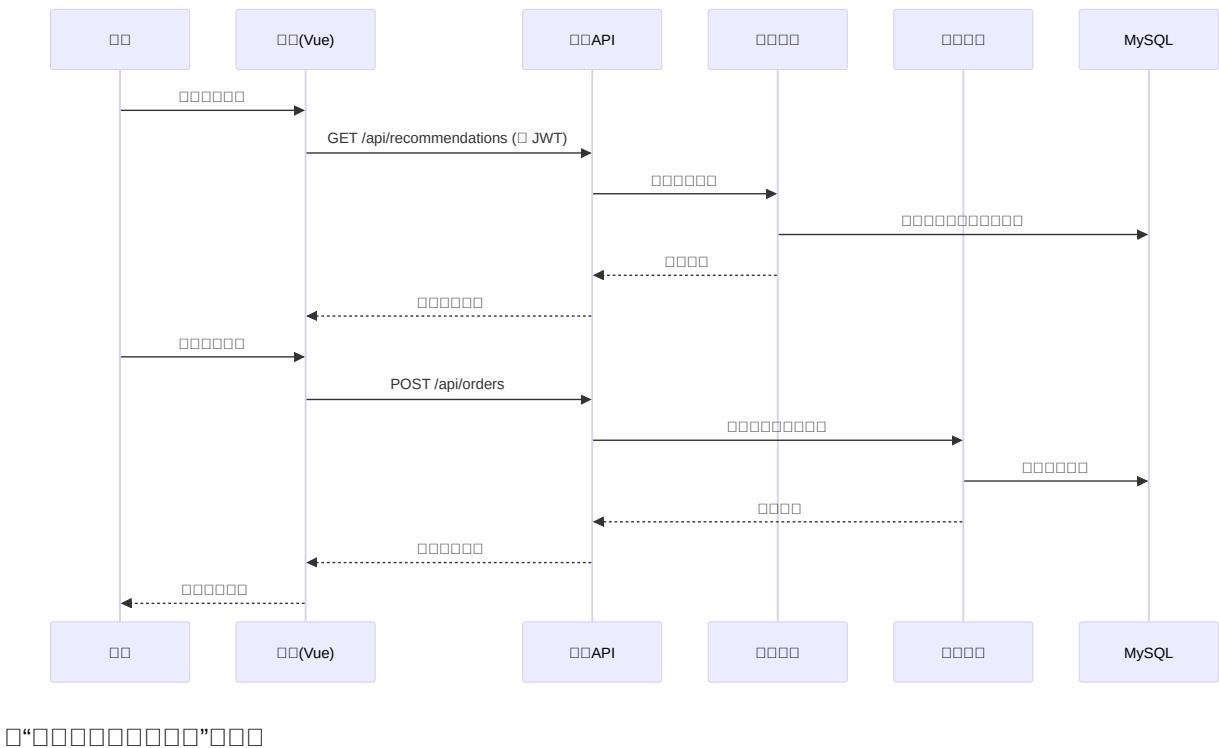
PROPERTY {
    bigint id PK
    varchar title
    varchar city
    decimal price
}
  
```

```

        int bedrooms
        bigint landlord_id FK
        float rating
    }
ORDER {
    bigint id PK
    bigint user_id FK
    bigint property_id FK
    datetime check_in
    datetime check_out
    varchar status
    decimal total_price
}
RECOMMENDATION {
    bigint id PK
    bigint user_id FK
    bigint property_id FK
    float score
    varchar strategy
}
USER_PROPERTY_INTERACTION {
    bigint id PK
    bigint user_id FK
    bigint property_id FK
    varchar action
    datetime occurred_at
}
USER ||--o{ ORDER : places
USER ||--o{ RECOMMENDATION : receives
USER ||--o{ USER_PROPERTY_INTERACTION : acts_on
PROPERTY ||--o{ ORDER : booked_by
PROPERTY ||--o{ RECOMMENDATION : recommended_to
PROPERTY ||--o{ USER_PROPERTY_INTERACTION : interacted_by
USER ||--o{ PROPERTY : owns

```

7. □□□□□□□□□□



8. フロントエンド

- cd frontend && npm install && npm run build

- mvn spring-boot:run
 - MySQL 8.x
 - application.yml

8.1

- **JWT + Axios** / **Token 401**
 - **JWT + Axios** / **Token 401**
 - **JWT + Axios** / **Token 401**
 - **JWT + Axios** / **Token 401**

8.2 □□□□□□□

- `user_id` `property_id` `city` `created_at`
 - `id`
 - `ID/UUID`
 - `url` / `path`

8.3 □□□□□□□

- **ELK**/MongoDB/Prometheus+Grafana/QPS/DB 监控
 - MySQL +Redis/CDN/CDN
 - HTTPS+JWT 安全性

9.

-                                                            <img alt="Icon representing a search or query" data-bbox="9

- `http://www.example.com/api/v1/tokens` Token `XXXXXXXXXX`
 - `http://www.example.com/api/v1/tokens` / `http://www.example.com/api/v1/tokens`
 - `http://www.example.com/api/v1/tokens` / `http://www.example.com/api/v1/tokens`
 - `http://www.example.com/api/v1/tokens` / `http://www.example.com/api/v1/tokens`
 - `http://www.example.com/api/v1/tokens` Token `XXXXXXXXXX` 401 `XXXXXXXXXX` 403 `XXXXXXXXXX`

9.2

- JUnit/MockMvc Vitest

- Postman/Newman □ Rest Client
 - JMeter/Locust □ 95/99
 -

10. □ □ □ □ □

1. □□□□□□□□□□□□□□□□□□
 2. □□□□□□□□□□□□□□□□□□□□□□
 3. □□ A/B □□□□□□□□□□□□
 4. □□□□□□□□□□□□□□□□□□□□

11.

- [1] Resnick P, Varian H R. Recommender systems. Communications of the ACM, 1997.
 - [2] He X, et al. Neural Collaborative Filtering. WWW, 2017.
 - [3] Sarwar B, et al. Item-based Collaborative Filtering Recommendation Algorithms. WWW, 2001.
 - [4] ○○○. ○○○○. ○○○○○○○, 2016.
 - [5] Kraska T. ML-based DBMS Design. SIGMOD, 2018.

12. □□