□□□□□□□□□□□□

□□□XXX□□□□□□□□
□□□XXX□□ □□□XXX □□□XXXXXXXX
□□□□□XXX □□□XXX
□□□□□2026 □ X □ X □

□□

□□□□□“□□□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□ 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. □□

1.1 □□□□

□□“□□+□□”□□□“□□—□□—□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□ ER □□□□□□□/□□□□□□□□□□□□□□□□□□□□□

□□“□□+□□□”□□□

1.2 □□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□
- □□ ER □□□□□□□□□□□□□□□□□□□□□□□□

1.3 □□□□

□□□

---

2. □□□□□□□□□

- □□□Vue 3□Vite□Element Plus□Pinia□Vue Router□Axios□
- □□□Spring Boot□MyBatis-Plus□JWT □□□Spring Validation□Lombok□
- □□□
- □□□□MySQL□□□□□□□□□□□ Redis □□□□/□□□□□

---

3. □□□□

- □□□□□□□/□□□□□/□□/□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□/□□/□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□
- □□□□□□□□□JWT □□

3.1 □□□□□□□

- □□/□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□

3.2 □□□□□

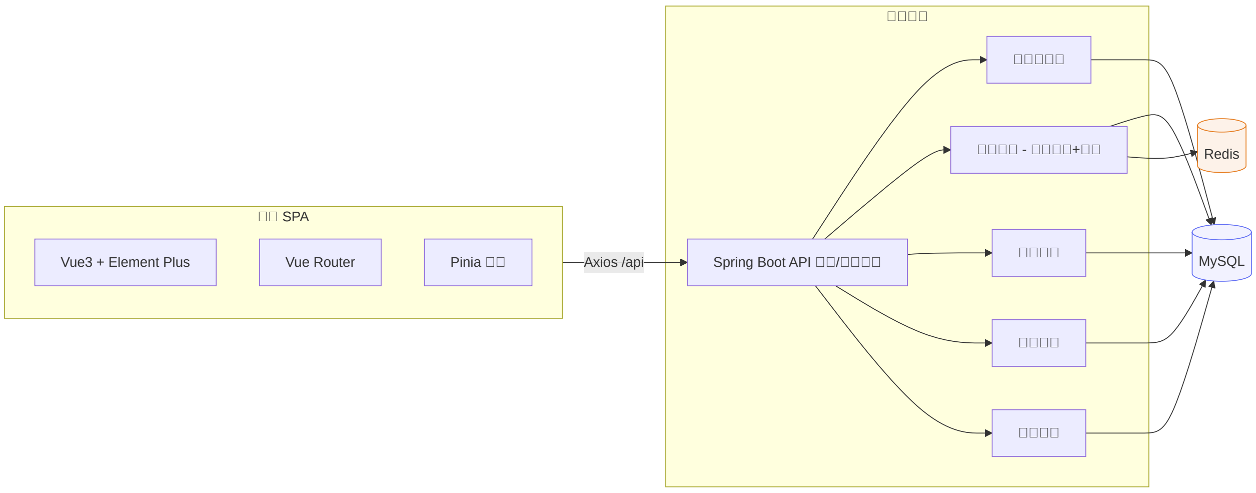
- JWT □□□□□□□□□□□□□□□ Token□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□□□□□□□

3.3 □□□□□

- □□□□□□□□□/□□
- □□□
- □□ token□
- □□□□□□□□□□□□□ HTTPS □□□Token □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

---

4. □□□□



4.1 □□□□□

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

    subgraph 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 □□□□□

```
flowchart TB
    View[□□□<br/>Vue3 + Element Plus] --> BFF[□□□<br/>Axios + □□□□]
    BFF --> Ctrl[□□□ Controller]
    Ctrl --> Service[□□□ Service]
    Service --> Mapper[□□□□□ MyBatis-Plus]
    Mapper --> DB[(MySQL)]
    Service --> RecCore[□□□□<br/>□□□□/□□□□□]
    RecCore --> Cache[(Redis/□□□□)]
    subgraph Infra[□□□□]
        Security[JWT + Spring Security]
        Validation[□□□□]
        Logging[□□□□□]
    end
    Ctrl --> Infra
```

---

5. □□□□

- □□□□□□□□□□□JWT □□□□□□USER/LANDLORD/ADMIN□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□/□□□/□□□
- □□□□□□□□□□□□□/□□/□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□60% □□□□ + 40% □□□□□□□□□□□□□□□□□□□□□□□□□□□

  □□□□□□□□□□□□□□□□□“□□□□/□□□□/□□□□”□□□□□ `frontend` □□□□□□□□□□□□□□□□□□□
  □□□□□□□

5.1 □□□□□□□□□

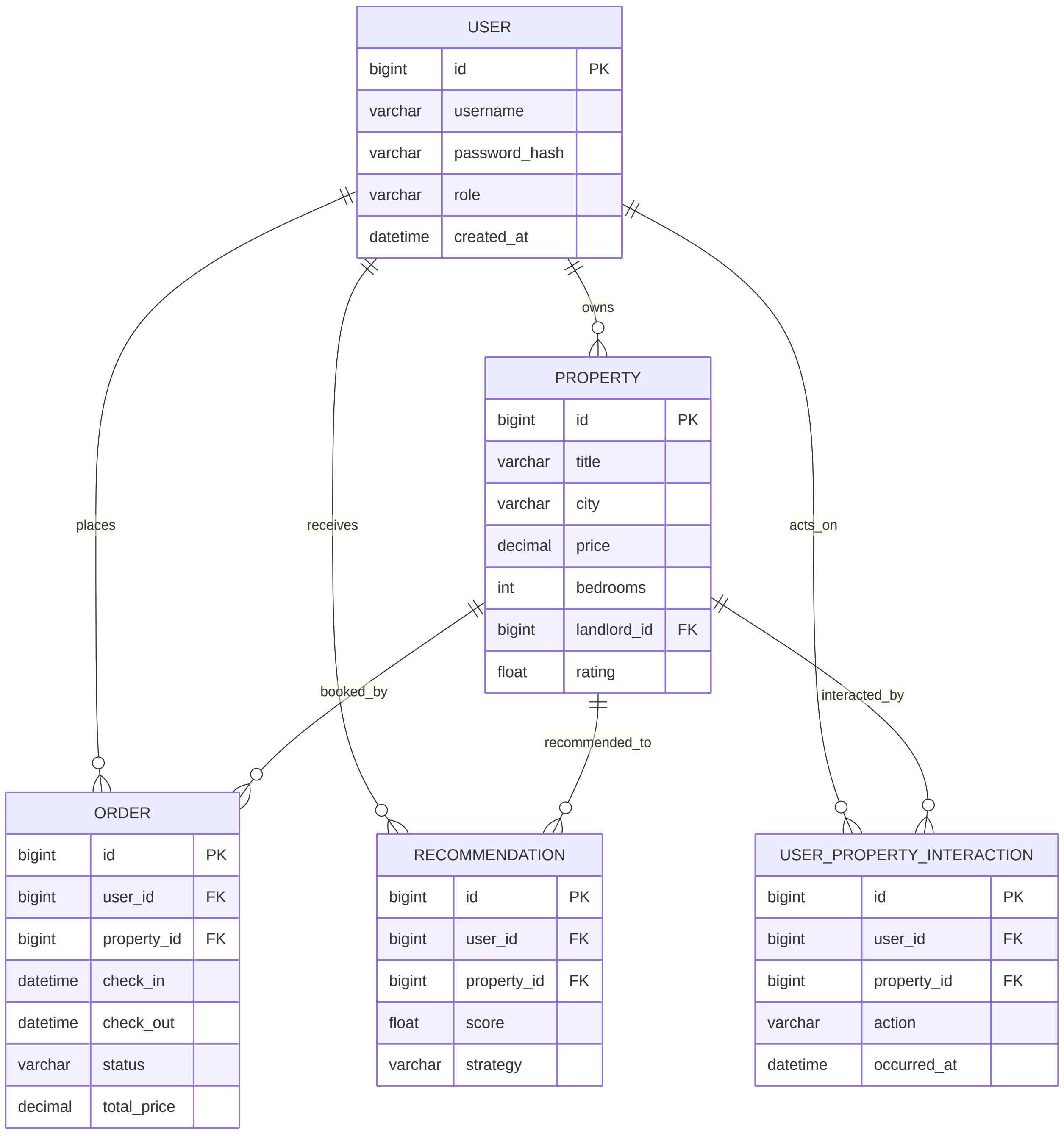
- □□□□□**USER**□□□□/□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□**LANDLORD**□□□□□□□□□□□□□□□□/□□/□□□□□□□□□□□□□
- □□□□**ADMIN**□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

5.2 □□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□-□□□□□□□□□□□□□□□ Top-N □□□
- □□□
- □□□□□□□ 0.6□□□□□□ + 0.4□□□□□□□ A/B □□□□□□□□□□□□□□□□/□□□□□□
- □□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□□□“□□□□□□ XX/□□ YY”□

---

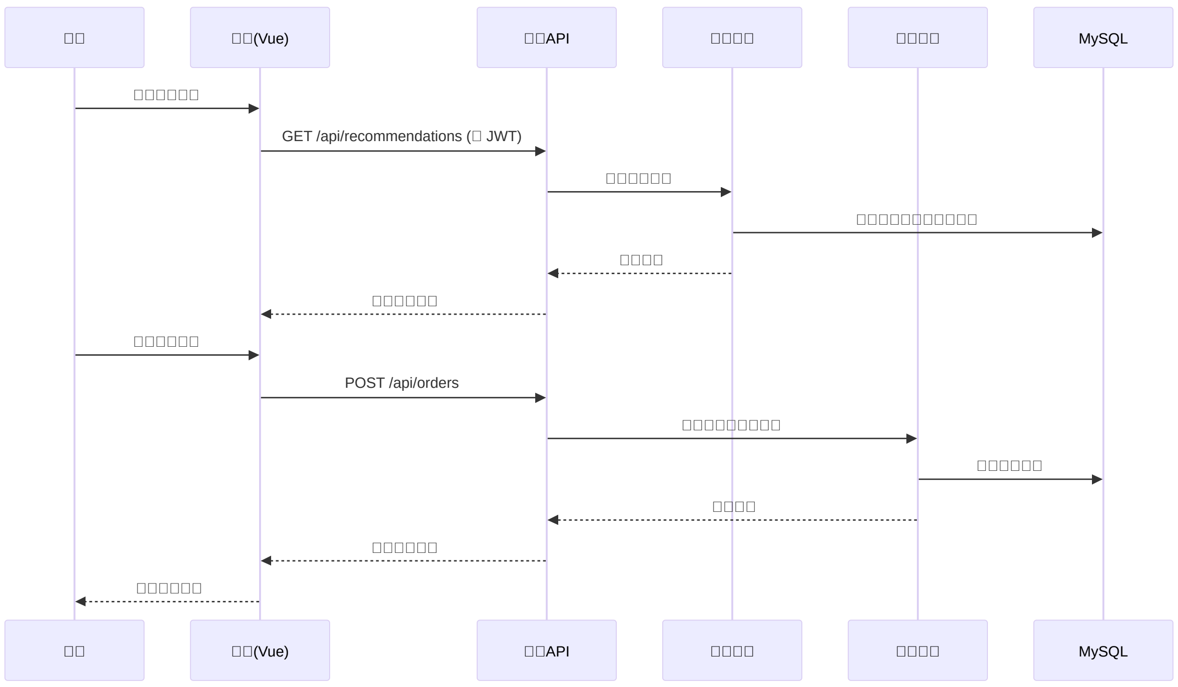
6. □□□□□□□ER □□



```
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. □□□□□□□□□□□



□“□□□□□□□□□□”□□□

```
sequenceDiagram
    participant U as □□
    participant FE as □□(Vue)
    participant API as □□API
    participant REC as □□□□
    participant ORD as □□□□
    participant DB as MySQL

    U->>FE: □□□□□□
    FE->>API: GET /api/recommendations (□ JWT)
    API->>REC: □□□□□□
    REC->>DB: □□□□□□□□□□□
    REC-->>API: □□□□
    API-->>FE: □□□□□□
    U->>FE: □□□□□□
    FE->>API: POST /api/orders
    API->>ORD: □□□□□□□□□
    ORD->>DB: □□□□□□
    ORD-->>API: □□□□
    API-->>FE: □□□□□□
    FE-->>U: □□□□□□
```

---

8. □□□□□□□□

- □□□`cd frontend && npm install && npm run build`□□□□□□ `frontend/dist`□

- □□□mvn spring-boot:run □□□□□□ java -jar target/*.jar□
- □□□□□MySQL 8.x□□□□□□□□□□□ sql/ □□□□□□□□□ Redis □□□□□□
- □□□□□□ application.yml □□□□□□JWT □□□□□□□□ frontend/vite.config.js □□ API □□□□□□□□

8.1 □□□□□□

- □□□□□□□□□□ JWT + □□□/□□□□□□□□□□□□□ Axios □□□□□□□□□ Token □□□ 401 □□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□/□□/□□/□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□
- □□□

8.2 □□□□□□□

- □□□user_id□property_id□city□created_at□□
- □□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□ ID/UUID□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□/□□□

8.3 □□□□□□□

- □□□□□□□□/□□/□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□ ELK□□□□□□□□□□ Prometheus+Grafana□□□□□□□□□□□QPS□□□□□□DB □□□□□□□□□□□□
- □□□□□□□□ MySQL □□□□+□□□□□□□□□□□□□□□□□□□□□□□□□□ CDN/□□□□
- □□□□□□ HTTPS□JWT □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

---

9. □□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□/□□□□□□□□□□□□
- □□□□□□□□ Postman/Rest Client □□□□ API□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□ JMeter □□

9.1 □□□□□□

- □□/□□□□□□□□□□□□□□□□□□□□□Token □□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□□□□□□□□
- □□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□□□□□□□□□□□□□□□
- □□□□□□ Token □□□□□□□□□□□□ 401□□□□□□□□□ 403□□□□□□□□□□□□□□□□□□□

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. □□

□□
□□□□□