

**Software Requirement Document**

**Smoking Support Platform**

**1.1 Background**

Smoking is a serious global health issue. This platform aims to support users in their journey to quit smoking through data tracking, progress monitoring, community interaction, and professional coaching.

**1.2 Objectives**

- Help users plan and follow their smoking cessation journey.

- Track health and cost improvements.

- Provide professional consultation and motivational tools.

**1.3 Scope**

The system includes features for user account management, smoking status tracking, progress visualization, badge system, community sharing, online consultation, and admin control.

**1.4 Stakeholders**

- End Users (Guests, Members, Premium Members)  
- Coaches  
- System Administrators  
- Project Development Team

**1.5 Expected Outcome**

A functional web-based platform accessible via desktop and mobile, enabling users to quit smoking effectively with real-time support.

**1.6 Technologies**

- Frontend: ReactJS  
- Backend: NodeJS  
- Database: SQL Server  
- Authentication: JWT  
- Hosting: Cloud-based

**1.7 Timeline (Gantt Chart)**

Week 1-2: Requirement Gathering  
Week 3-4: Design  
Week 5-8: Development  
Week 9-10: Testing & Deployment

**1.8 Team Members**

- [Member A] – Frontend  
- [Member B] – Backend  
- [Member C] – Database  
- [Member D] – Documentation & Testing

**2.1 Functional Requirements**

- User registration, login, password reset

- Smoking status tracking

- Quit plan setup and customization

- Daily progress tracking

- Achievement badges and leaderboard

- Community blog and interaction

- Online consultation

- Admin management and analytics dashboard

**2.2 Non-Functional Requirements**

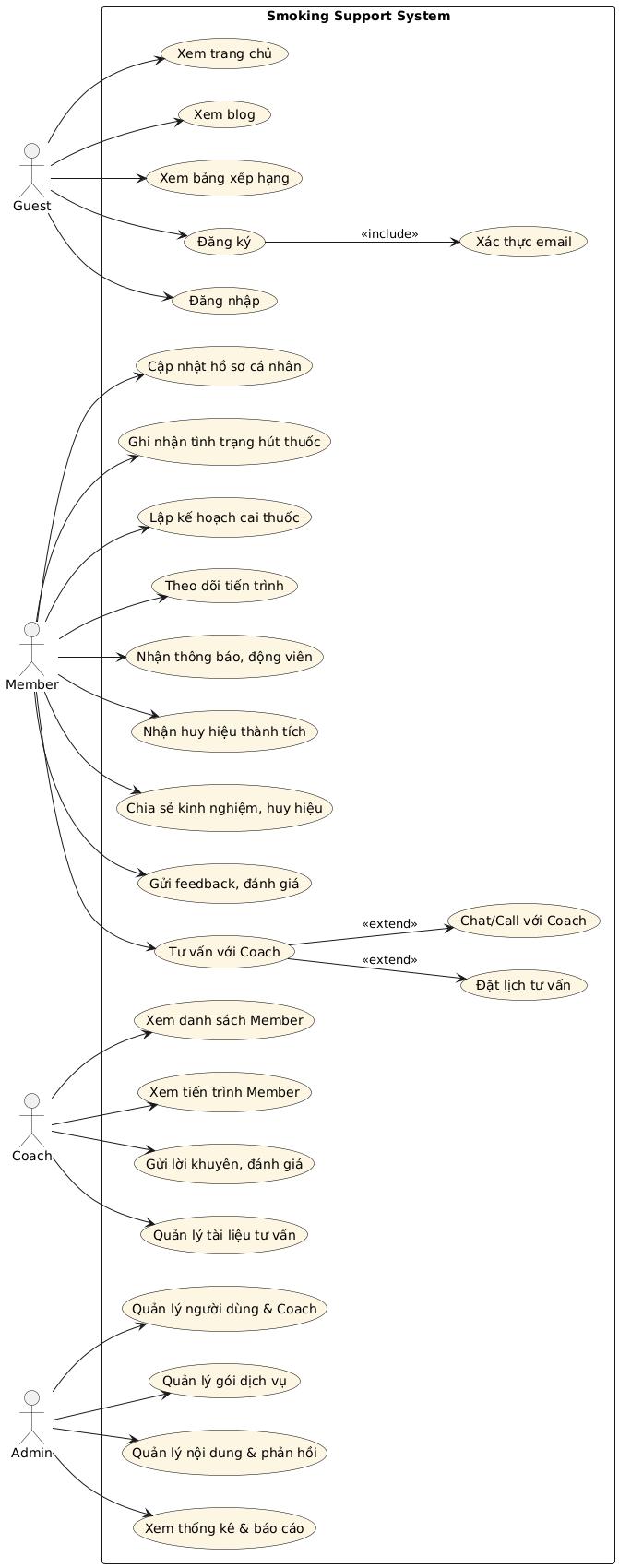
- Performance: <2s response, 5000 concurrent users

- Security: JWT auth, CORS, rate limiting

- Maintainability: Clean, documented code

- Compatibility: Works on Chrome, Firefox, Edge, mobile

**2.3 Use Case Diagram**



**2.4 Use Case Descriptions**

- Guest: View info, register, login

- Member: Track status, receive badges, interact in community

- Premium: Get consultation, access detailed reports

- Coach: Advise, monitor members, write guidance

- Admin: Manage users, content, system stats

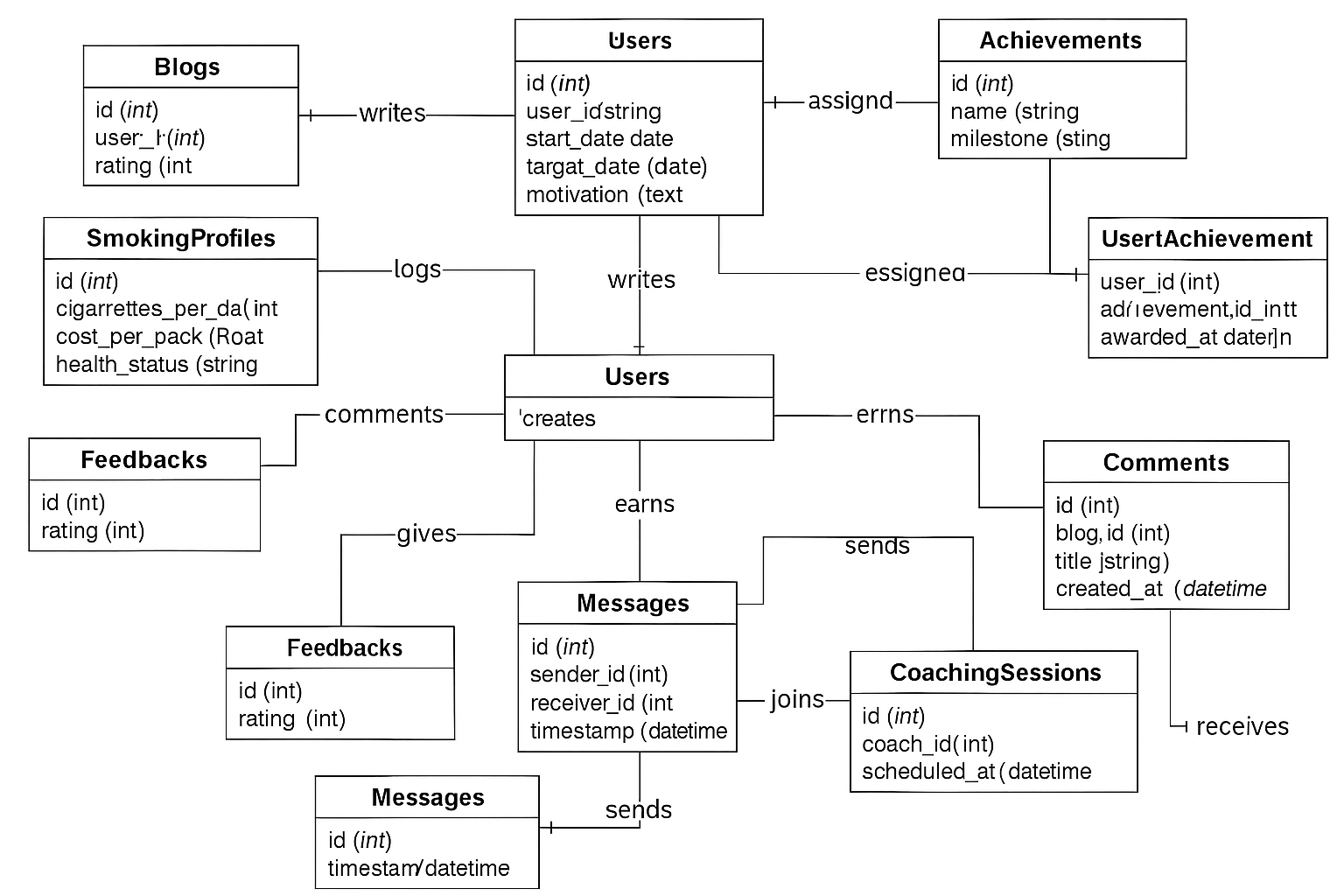
**2.5 User Roles**

- Guest, Member, Premium Member, Coach, Admin

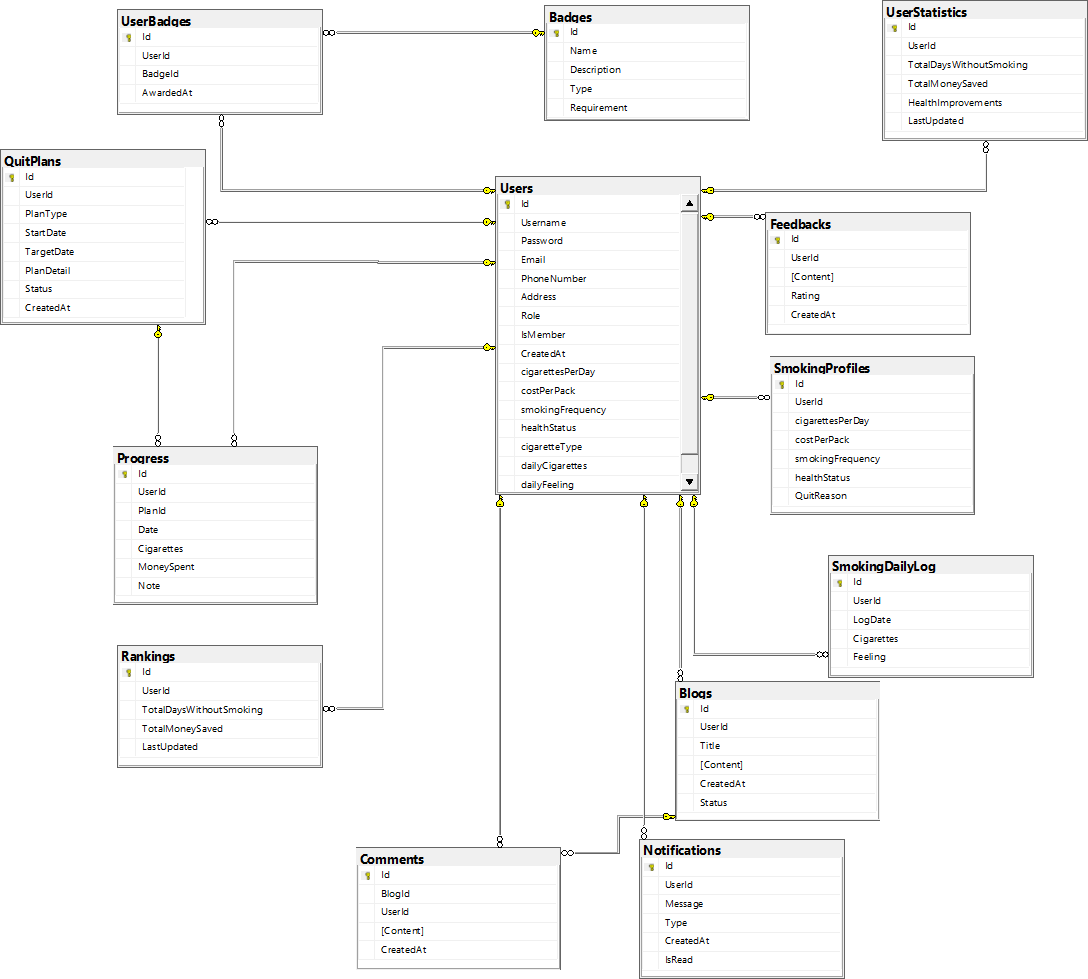
**3.1 Architecture Diagram**

Frontend <-> Backend API (Spring Boot) <-> MySQL DB  
+ JWT for auth

**3.2 Entity Relationship Diagram (ERD)**



**3.3 Database Schema**



**3.4 UI/UX Mockups**

(to be added based on implementation tools like Figma)

**3.5 Navigation Flow**

Guest > Register > Member Dashboard > Quit Plan > Progress Tracking > Community > Consultation > Reports

**4.1 Tools & Technologies**

- ReactJS, SQL SERVER, JWT, REST API, NODEJS

**4.2 Backend Implementation Overview**

- RESTful API  
- JWT for Auth  
- Controllers for Users, Plans, Progress, Badges, Admin

**4.3 Frontend Implementation Overview**

- React Pages: Login, Dashboard, Quit Plan, Community, Profile, SubscriptionPlans  
- Axios for API Calls

**4.4 Key Code Snippets**

(Add controller/service snippets in final version)

**4.5 Integration Plan**

- Backend API tested with Postman  
- Frontend integrated via Axios  
- Deployment to cloud (e.g., Firebase/Heroku/AWS)

**5.1 Test Strategy**

- Unit testing (JUnit, xUnit)  
- Integration testing  
- End-to-end scenarios

**5.2 Test Cases**

- Login success/failure  
- Plan creation  
- Badge award logic  
- Coach chat functionality

**5.3 Bug Reports**

- (To be added during implementation testing)

**5.4 Performance Evaluation**

- Expected response < 2s, Load tested to 5000 users

**5.5 Limitations & Future Improvements**

- Add AI chatbot in future  
- Mobile app version  
- Gamified challenges