### 5.6 Profile Feature

**Feature Goal**: Provide users with a personal profile to manage their account, view progress, and customize their experience.

#### System Architecture

**Frontend Components**:

* ProfileScreen.tsx: Main profile view
* SettingsScreen.tsx: User settings
* SubscriptionScreen.tsx: Subscription management
* AchievementsScreen.tsx: Badges and achievements

**Backend Components**:

* user\_service.py: User profile management
* subscription\_service.py: Subscription handling
* achievement\_service.py: Badge and achievement tracking

**Database Models**:

* users: User accounts

CREATE TABLE users (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

auth\_id VARCHAR(255) NOT NULL UNIQUE,

username VARCHAR(50) NOT NULL UNIQUE,

email VARCHAR(255) NOT NULL UNIQUE,

display\_name VARCHAR(100),

bio TEXT,

avatar\_url VARCHAR(255),

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

last\_login TIMESTAMP WITH TIME ZONE

);

* user\_settings: User preferences

CREATE TABLE user\_settings (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

user\_id UUID NOT NULL REFERENCES users(id) UNIQUE,

notification\_preferences JSONB NOT NULL DEFAULT '{}',

theme VARCHAR(20) DEFAULT 'auto',

language VARCHAR(10) DEFAULT 'en',

privacy\_settings JSONB NOT NULL DEFAULT '{}',

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

* user\_subscriptions: Subscription details

CREATE TABLE user\_subscriptions (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

user\_id UUID NOT NULL REFERENCES users(id) UNIQUE,

subscription\_tier VARCHAR(20) NOT NULL DEFAULT 'free',

stripe\_customer\_id VARCHAR(255),

stripe\_subscription\_id VARCHAR(255),

current\_period\_start TIMESTAMP WITH TIME ZONE,

current\_period\_end TIMESTAMP WITH TIME ZONE,

status VARCHAR(20) NOT NULL DEFAULT 'active',

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

* user\_badges: User achievements

CREATE TABLE user\_badges (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

user\_id UUID NOT NULL REFERENCES users(id),

badge\_id VARCHAR(50) NOT NULL,

earned\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

metadata JSONB,

UNIQUE(user\_id, badge\_id)

);

#### API Endpoints

* GET /api/v1/users/me: Get current user profile
  + Response:

{

"id": "350e8400-e29b-41d4-a716-446655440000",

"username": "philosophyFan",

"display\_name": "Philosophy Enthusiast",

"email": "user@example.com",

"bio": "Passionate about exploring philosophical ideas...",

"avatar\_url": "https://example.com/avatars/user.jpg",

"created\_at": "2025-01-15T10:00:00Z",

"xp": {

"total": 1250,

"level": 12,

"next\_level\_xp": 1500,

"progress\_percentage": 83

},

"subscription": {

"tier": "premium",

"status": "active",

"current\_period\_end": "2025-06-15T10:00:00Z"

},

"stats": {

"questions\_asked": 42,

"quests\_completed": 15,

"concepts\_explored": 78,

"journal\_entries": 23,

"forum\_posts": 12,

"streak\_days": 7

}

}

* PUT /api/v1/users/me: Update user profile
  + Request:

{

"display\_name": "Philosophy Enthusiast",

"bio": "Passionate about exploring philosophical ideas...",

"avatar\_url": "https://example.com/avatars/user.jpg"

}

* + Response:

{

"id": "350e8400-e29b-41d4-a716-446655440000",

"username": "philosophyFan",

"display\_name": "Philosophy Enthusiast",

"bio": "Passionate about exploring philosophical ideas...",

"avatar\_url": "https://example.com/avatars/user.jpg",

"updated\_at": "2025-05-28T17:30:00Z"

}

* GET /api/v1/users/me/settings: Get user settings
  + Response:

{

"notification\_preferences": {

"email\_digests": true,

"push\_notifications": {

"new\_responses": true,

"quest\_reminders": true,

"forum\_replies": true

}

},

"theme": "dark",

"language": "en",

"privacy\_settings": {

"show\_activity": true,

"public\_profile": true

}

}

* PUT /api/v1/users/me/settings: Update user settings
  + Request:

{

"notification\_preferences": {

"email\_digests": false,

"push\_notifications": {

"new\_responses": true,

"quest\_reminders": false,

"forum\_replies": true

}

},

"theme": "dark",

"language": "en"

}

* + Response:

{

"notification\_preferences": {

"email\_digests": false,

"push\_notifications": {

"new\_responses": true,

"quest\_reminders": false,

"forum\_replies": true

}

},

"theme": "dark",

"language": "en",

"privacy\_settings": {

"show\_activity": true,

"public\_profile": true

},

"updated\_at": "2025-05-28T17:35:00Z"

}

* GET /api/v1/users/me/badges: Get user badges
  + Response:

{

"badges": [

{

"id": "quest\_master",

"name": "Quest Master",

"description": "Complete 10 quests",

"icon\_url": "https://example.com/badges/quest\_master.png",

"earned\_at": "2025-04-10T14:20:00Z"

},

...

],

"total": 12

}

#### User Experience Flow

1. User accesses their profile via the ProfileScreen component
2. User can view their XP progress, level, and stats
3. User can update profile information (display name, bio, avatar)
4. User can manage notification preferences and app settings
5. User can view and manage their subscription
6. User can browse their earned badges and achievements

#### XP Integration

* Profile displays current XP, level, and progress to next level
* Shows badges and achievements earned through the XP system
* Displays streak information and current multipliers

#### Implementation Considerations

* **Avatar Management**: Support for uploading custom avatars or selecting from presets
* **Subscription Handling**: Integration with Stripe for subscription management
* **Privacy Controls**: Options for controlling profile visibility and activity sharing
* **Notification Preferences**: Granular control over notification types and frequency
* **Account Security**: Options for password change, two-factor authentication (future)
* **Data Export**: Ability to export user data (GDPR compliance)