

Overview

This application is a membership-based portal that provides a comprehensive cannabis growing service. Members subscribe to have plants grown on their behalf, track growth via logs and images, manage payments through Payfast, and interact with an admin. The system is designed with strict compliance, privacy, and legal standards in mind, as reflected in the business model documentation (e.g. The Dank House EULA).

1. Core Functional Requirements

1.1 Authentication & User Roles

- **Authentication System:**
 - Utilize Laravel's built-in auth scaffolding or Laravel Breeze/Jetstream for secure login, registration, and password resets.
 - Implement email verification, two-factor authentication, and secure password storage.
- **User Roles:**
 - Define roles such as **Member** and **Admin**.
 - Admins have full control over user data, plant records, and subscriptions. Members can only view and update their own profile and plant data.
- **QR Code Generation:**
 - Automatically assign and generate a unique QR code per user upon registration.
 - Members can view/download their QR codes; Admins can scan codes to quickly retrieve member information.

1.2 Database Structure

- **Users Table:**
 - Standard fields (name, email, password, etc.) plus a field for a unique QR code.
- **User Profiles Table:**
 - Contains extended member data (ID number, date of birth, contact details, etc.).
 - Members can update their information; Admins have full CRUD permissions.

- **Plants Table:**
 - Fields: id, strain, planting_date, harvest_date, user_id (foreign key), status, etc.
 - Members can view their plants; Admins can create, update, or delete records.
- **Growth Logs Table:**
 - Fields: plant_id (foreign key), date, notes, height, phase, etc.
 - Serves as a timeline for plant development.
- **Images Table:**
 - Fields: plant_id (foreign key), image_path, upload_date, description.
 - Supports gallery functionality for plant images.
- **Subscriptions & Payments Table:**
 - Track membership fees (R40/month), payment status, transaction details, and integration with Payfast.

1.3 File Storage & Image Handling

- **File Storage:**
 - Configure Laravel's filesystem to support image uploads.
 - Evaluate storage options: Local disk for development and potentially S3/cloud storage for production.
- **Image Optimization:**
 - Implement automatic image optimization and thumbnail generation to improve performance and usability.

2. Member Dashboard Interface

The member dashboard is designed for ease of use and a clear overview of each member's plants and subscription status.

- **Overview:**
 - Display a summary of all plants belonging to the logged-in member.
- **Notifications:**
 - List and manage admin-sent notifications (updates, payment reminders, etc.).
- **Growth Progress & Analytics:**
 - Present growth logs and key metrics using interactive charts/graphs (e.g. using Chart.js).

- Include a timeline view of plant development.
- **Image Gallery:**
 - Allow members to view and manage an image gallery for each plant.
- **Subscription & Payment Details:**
 - Provide a section with current subscription status, payment history, and next billing date.
- **Messaging Interface:**
 - Enable in-app chat for direct communication with the admin (potentially leveraging Laravel Echo or websockets).

3. Admin Dashboard Interface

The admin dashboard provides comprehensive controls and oversight for managing the service.

3.1 User & Member Management

- **User Listing:**
 - Display all registered members along with their QR codes and detailed profiles.
- **CRUD Operations:**
 - Allow admins to create, update, or delete member profiles.
- **QR Code Scanner:**
 - Implement a feature to scan a user's QR code to quickly pull up their information.

3.2 Plant Management

- **Plant Records:**
 - List all plants with status indicators.
 - Allow admins to create, update, and delete plant records.
- **Growth Logs & Images:**
 - Provide access to all growth logs and images for each plant.
 - Support bulk actions (e.g., image uploads, log corrections).

3.3 Subscription & Payment Administration

- **Payment Tracking:**

- Display current subscription statuses, payment history, and pending transactions.
- **Integration with Payfast:**
 - Manage recurring payments (R40/month) through Payfast.
 - Process refunds, handle manual overrides, and adjust membership fees when needed.

3.4 Content & Notifications Management

- **Announcements:**
 - Manage site-wide notifications and announcements.
- **Legal & Compliance:**
 - Update and display legal documents (e.g. Terms of Service, EULA) that members must agree to.
- **Reporting & Analytics:**
 - Generate detailed reports on plant yields, subscription revenue, user activity, and overall system health.

4. Additional Business Model & Feature Integrations

4.1 QR Code & Membership Verification

- **QR Code Integration:**
 - Each member's unique QR code is central for identity verification and rapid member lookup by admins.

4.2 Legal Agreement & Compliance

- **Display Legal Documents:**
 - Ensure the EULA (and other relevant agreements) are accessible to users.
 - Require members to acknowledge and consent to the legal terms during registration.
- **Data Privacy & Security:**
 - Handle personal information in strict compliance with data protection laws.
 - Ensure all member data is encrypted and access-controlled.

4.3 Notifications & Real-time Messaging

- **In-app Notifications:**
 - Real-time notifications for new growth log entries, upcoming payments, or admin messages.
- **Messaging System:**
 - A chat system for members to contact the admin directly for support or queries.

4.4 Responsive & Accessible Design

- **User Experience:**
 - Design both the member and admin interfaces to be fully responsive and accessible across all devices using Tailwind CSS.
- **Performance:**
 - Optimize all pages for speed, especially where large image files or chart data are involved.

5. API Integration & External Services

- **Payment Gateway:**
 - Integrate with Payfast for secure processing of recurring membership fees.
- **Optional External APIs:**
 - If additional services (such as the Travelport SOAP API) are needed, document integration points and ensure robust error handling and secure API key management.

6. Testing, Deployment & Documentation

6.1 Testing

- **Unit & Feature Tests:**
 - Develop comprehensive tests using Laravel's testing framework to cover all functionalities.
- **User Acceptance Testing (UAT):**

- Plan and execute UAT sessions with both admin and member users to validate usability and functionality.

6.2 Deployment

- **CI/CD Pipeline:**
 - Set up continuous integration/continuous deployment pipelines.
- **Environment Management:**
 - Configure separate staging and production environments.
- **Monitoring & Logging:**
 - Implement robust logging and monitoring to ensure system reliability and to quickly address issues.

6.3 Documentation

- **Developer Documentation:**
 - Provide detailed documentation of the codebase, API endpoints, and architectural decisions.
- **User Guides:**
 - Create user manuals and FAQs for both members and admins.
- **API Documentation:**
 - If any external or internal APIs are exposed, include full API documentation for ease of integration.

Conclusion

This comprehensive development specification outlines the technical and business requirements for the cannabis growing membership platform. It details the authentication system, database structure, file storage, member and admin dashboards, subscription management, legal compliance, and future scalability considerations. By following this specification, the project can ensure a secure, user-friendly, and compliant system that meets both the business model and technical requirements.