

License Management System - Implementation Guide

Overview

Your driving school software now includes a complete **License Management System** that allows you to sell customizable feature packages, similar to Tesla's model. This enables you to offer a base product with optional premium features that customers can unlock through upgrades.

System Architecture

Database Models

1. Organization

Each driving school is an organization with:

- Basic information (name, contact details, branding)
- Subscription tier (BASE, PREMIUM, ENTERPRISE)
- Subscription status (ACTIVE, TRIAL, EXPIRED, etc.)

2. OrganizationFeature

Tracks which features are enabled for each organization:

- Feature key (e.g., "STUDENT_ACCESS", "VEHICLE_MANAGEMENT")
- Enabled/disabled status
- Audit trail (who enabled it, when)

3. LicenseKey

Pre-generated keys that unlock features:

- Unique key code
- List of features to unlock
- Expiration date (optional)
- Usage tracking

Premium Features Available

Currently Implemented Premium Features:

1.  **Student Access** (STUDENT_ACCESS)
 - Allows students to log in and access their portal
 - View lessons, progress, and schedules
 - By default: Only instructors and admins can log in

2.  **Vehicle Management** (VEHICLE_MANAGEMENT)
 - Full vehicle fleet management
 - Maintenance tracking
 - Vehicle assignments
 - Insurance and service schedules

3.  **Screenshot Protection** (SCREENSHOT_PROTECTION)
 - Prevent screenshots on sensitive pages
 - Protect proprietary content
4.  **Advanced Reporting** (ADVANCED_REPORTING)
 - Detailed analytics and insights
 - Custom report generation
5.  **SMS Notifications** (SMS_NOTIFICATIONS)
 - Automated SMS reminders
 - Real-time notifications
6.  **Mobile App Access** (MOBILE_APP)
 - Native iOS and Android apps
 - (Future implementation)
7.  **Payment Integration** (PAYMENT_INTEGRATION)
 - Integrated payment processing
 - Automated invoicing
8.  **Multi-Language Support** (MULTI_LANGUAGE)
 - Support for multiple languages
 - Localization features

Admin Interface

Accessing License Management

1. Log in as a **SUPER_ADMIN**
2. Click on “**License**” in the navigation menu
3. You’ll see the License Management dashboard

Dashboard Features

Organization Information

- View your organization name
- Current subscription tier
- Number of active premium features

License Key Activation

- Enter a license key to unlock features
- Instant activation
- Automatic feature enabling

Feature Management

- Toggle features on/off manually (admin-only)
- View feature descriptions
- See activation status with visual indicators
- Real-time updates

Feature Summary

- Visual grid of all features
- Quick overview of enabled/disabled features

- Color-coded status indicators

How to Use

For Initial Setup

1. Seed Script Already Run

- Organization “Elite Driving Academy” created
- All users linked to organization
- Sample license key generated: LIC-1760536155676-SAMPLE
- Features to unlock: STUDENT_ACCESS, VEHICLE_MANAGEMENT

2. Activate Sample License

```
bash
# Log in as admin
# Go to /admin/license
# Enter the license key: LIC-1760536155676-SAMPLE
# Click "Activate"
```

For Your Business Model

Scenario 1: Base Package

- Instructor management ✓
- Lesson scheduling ✓
- Basic reporting ✓
- Schedule map ✓
- No student login access
- No vehicle management

Scenario 2: Premium Package

- Everything in Base ✓
- Student login access ✓ (Enable STUDENT_ACCESS)
- Vehicle management ✓ (Enable VEHICLE_MANAGEMENT)
- Advanced reporting ✓
- SMS notifications ✓

Scenario 3: Custom Package

- Pick and choose features
- Create custom license keys
- Tailored pricing

Selling to Customers

Step 1: Initial Deployment

1. Deploy the base software to customer
2. Run the organization seed script:

```
bash
cd nextjs_space
yarn tsx scripts/seed-organization.ts
```

Step 2: Generate License Keys

- Use the admin panel or API to generate license keys

- Each key can unlock one or more features
- Set expiration dates (optional)

Step 3: Customer Activation

- Customer receives license key via email
- Customer enters key in License Management page
- Features activate instantly
- No downtime or redeployment needed

API Integration

Check Feature Status (Backend)

```
import { LicenseService } from '@/lib/services/license-service';

// Check if a feature is enabled
const isEnabled = await LicenseService.isFeatureEnabled(
  organizationId,
  'STUDENT_ACCESS'
);

if (isEnabled) {
  // Allow access
} else {
  // Show upgrade message
}
```

Feature Gates (Frontend)

```
import { FeatureGate } from '@/components/license/feature-gate';

<FeatureGate featureKey="VEHICLE_MANAGEMENT">
  <VehicleManagementComponent />
</FeatureGate>
```

API Route Protection

```
import { checkFeatureAccess } from '@/lib/middleware/feature-check';

export async function GET(request: NextRequest) {
  const featureCheck = await checkFeatureAccess('VEHICLE_MANAGEMENT');

  if (!featureCheck.allowed) {
    return NextResponse.json({
      error: 'Feature not enabled',
      requiresUpgrade: true
    }, { status: 403 });
  }

  // Continue with request
}
```

Example: Vehicle Management Feature

The Vehicle Management feature has been fully implemented with feature gates:

Frontend Protection

- `/admin/vehicles` page wrapped in `<FeatureGate>`
- Shows upgrade message if feature is disabled
- Seamless user experience

Backend Protection

- API routes check feature status
- Returns 403 with upgrade message if disabled
- Secure server-side enforcement

Usage Flow

1. Customer tries to access vehicle management
2. System checks if `VEHICLE_MANAGEMENT` is enabled
3. If disabled: Shows premium feature message
4. If enabled: Full access to vehicle management

Creating License Keys Programmatically

```
import { LicenseService } from '@/lib/services/license-service';

// Create a license key
const result = await LicenseService.createLicenseKey(
  organizationId,
  ['STUDENT_ACCESS', 'VEHICLE_MANAGEMENT'], // Features to unlock
  new Date('2026-12-31'), // Expiration date (optional)
  'Customer: ABC Driving School',
  adminUserId
);

if (result.success) {
  console.log('License Key:', result.key);
  // Send to customer via email
}
```

Migration Path to Python/Flutter

This license system is **database-driven** and **API-based**, making it easy to migrate to Python/Flutter later:

What Stays the Same

- Database schema (PostgreSQL)
- License keys and feature flags
- Business logic and rules

What Changes

- Backend: Next.js → Python (FastAPI/Django)

- Frontend: React → Flutter
- Database client: Prisma → SQLAlchemy/Django ORM

The **core concept and database structure remain identical**, so you can start with this implementation and migrate later without losing your license management infrastructure.

Business Benefits

For You (Software Vendor)

- ✓ Flexible pricing model
- ✓ Easy upselling
- ✓ Remote feature activation
- ✓ No redeployment needed
- ✓ Track feature usage
- ✓ Scalable business model

For Your Customers

- ✓ Pay for what they need
- ✓ Instant upgrades
- ✓ No service disruption
- ✓ Trial features before buying
- ✓ Flexible scaling

Testing the System

Test Scenario 1: Premium Feature Locked

1. Log in as admin
2. Go to `/admin/vehicles`
3. If `VEHICLE_MANAGEMENT` is disabled, you'll see:
 - “Premium Feature” alert
 - “This feature requires an upgrade” message
 - Option to contact administrator

Test Scenario 2: Activate License

1. Go to `/admin/license`
2. Enter license key: `LIC-1760536155676-SAMPLE`
3. Click “Activate”
4. Success message appears
5. Features are now unlocked
6. Refresh `/admin/vehicles` to see full access

Test Scenario 3: Manual Toggle

1. Go to `/admin/license`
2. Scroll to Premium Features section
3. Toggle `VEHICLE_MANAGEMENT` switch
4. Feature enables/disables instantly
5. Other pages reflect the change immediately

Next Steps

Immediate Actions

1. ✓ Organization setup complete
2. ✓ Sample license key available
3. Test the license activation
4. Explore the admin panel
5. Try toggling features on/off

Business Development

1. Define your pricing tiers
2. Decide which features are premium
3. Create license key generation workflow
4. Set up customer onboarding process
5. Plan sales and marketing strategy

Future Enhancements

1. Add more premium features
2. Implement usage analytics
3. Create customer portal for license management
4. Add subscription billing integration
5. Build license key marketplace

Support and Documentation

Files Created

- `/lib/config/license-features.ts` - Feature definitions
- `/lib/services/license-service.ts` - Core license logic
- `/lib/middleware/feature-check.ts` - API route protection
- `/components/license/feature-gate.tsx` - Frontend gate component
- `/hooks/use-license.ts` - React hook for license data
- `/app/admin/license/page.tsx` - Admin UI
- `/app/api/admin/license/*` - License API routes
- `/scripts/seed-organization.ts` - Organization setup script

Key Concepts

- **Organization:** A driving school customer
- **Feature:** A premium capability that can be enabled/disabled
- **License Key:** Pre-generated code that unlocks features
- **Feature Gate:** UI component that hides premium features
- **Subscription Tier:** BASE, PREMIUM, or ENTERPRISE

Summary

You now have a fully functional **Tesla-style feature licensing system** that allows you to:

- Sell different feature packages
- Remotely activate/deactivate features
- Offer flexible pricing tiers
- Scale your business model
- Upgrade customers without redeployment

The system is **production-ready** and **database-driven**, making it easy to migrate to Python/Flutter in the future while preserving all your business logic and customer data.

Your sample license key: LIC-1760536155676-SAMPLE

Unlocks: Student Access, Vehicle Management

Go to `/admin/license` to start exploring! 