

Delivery app project

Задание

Разработать систему управления для службы доставки используя фреймворк Ruby on Rails на основе тестового задания.

User stories

US 1. As system I want to store information about Package in the database

Package should have following fields:

- UUID(primary key)
- created date
- updated date
- estimated delivery date
- tracking number(format: YA<8 digits sequence>AA)
- delivery status(new, processing, delivered, cancelled)

Tracking number should be generated automatically by system

Default delivery status of package should be new

US 2. As system I want to store information about DeliveryManager in the database

DeliveryManager should have following fields:

- UUID(primary key)
- created date
- updated date
- enabled

US 3. As system admin I want to be able to manage system resources using ActiveAdmin interface

- Install and configure Active Admin administration framework

US 4. As DeliveryManager I should be able to login to admin interface

- only active DeliveryManagers can login to system

US 5. As DeliveryManager I should have specific permissions to manage system resources

- Use Pundit library to configure system policies
- DeliveryManager should be able to manage Packages in system
- DeliveryManager should be able to manage Couriers in system

US 6. As system I should be able to save PackageAssignment

- PackageAssignment should be related to the Package and the Courier

- US 7. As DeliveryManager I should be able to assign Courier to Package
- Only packages in processing status can be assigned
 - Package should have status assigned on success
- US 8. As DeliveryManager I should be able to cancel package via Admin panel
- Packages with status new and processing can be cancelled
- US 9. As Courier app developer I should be able to access API documentation
- API endpoints should be documented by Swagger
 - Integration tests for application should use Swagger docs
- US 10. As Courier I should be able to login to system via API
- Courier can login using email and password
 - On success login user gets auth token for further actions
 - Auth token should expire in 24 hours
- US 11. As Courier I should be able authenticate using Auth token
- Handle auth token sent in Authorization header
- US 12. As Courier I should be able to see package details by API
- Package should have following fields:
- estimated delivery date
 - tracking number
 - delivery status
- US 13. As Courier I should be able to get list of assigned packages by API
- List of packages should contain the following information:
 - UUID
 - estimated delivery date
 - tracking number
- US 14. As Courier I should be able to pick-up assigned package by API
- Courier can pick-up packages with assigned status
 - On pick-up action status of package changes to pickuped
- US 15. As Courier I should be able to cancel assigned by API
- Courier can cancel packages with assigned status
 - On cancel action status of package changes to processing
- US 16. As Courier I should be able to generate delivery confirmation URL
- Confirmation url can be accessed by anyone
 - Confirmation url should be valid for 10 minutes
 - On confirm, Package status should be set on delivered

US 17. As Courier I should be able to upload my driver license for verification

- Courier can upload multipart image as his driver license
- Courier can upload image only once

US 18. As DeliveryManager I should be able to validate Courier

- DeliveryManager can check validity of Couriers driver licence in admin panel
- DeliveryManager can validate Courier using button
- DeliveryManager can not edit driver license

US 19. As non-valid Courier I should not be able to access application

- Non-valid couriers are not allowed to access application

US 20. As Courier app developer I should be able to deploy my program on Heroku using Gitlab pipeline

- Setup on demand deployment process to Heroku hosting using Gitlab CI
- Deployment should be available only when all other Pipeline tasks are completed successfully