



Lunch Ordering SaaS

Each company that uses the services of delivery meals, sooner or later, comes to the great and original idea to develop their own service to order a lunch. Many start, but only few of them bring it to the end. The objective of this task is to develop very small MVP, which could be taken and used already in production.

We will try to simplify the problem as much as possible. Please find User Stories below. Let's assume that there is only one organization in the application, all users belong to it.

- As a guest I should be able to sign up with name, email and password.
- As a guest I should be able to sign in with email and password.
- As a first registered user in the system, I become a Lunches Admin.

- As a user I can edit my profile
- As a user I can see a weekdays on the dashboard page
- As a user, when I click on the weekday(today or days in the past), I can see menu - list of items with prices.
- As a user, when I choose items from menu, I can only choose one item from the first course, one item from the main course and one drink. Total 3 items.
- As a user I can press Submit button to process my order.

- As a Lunches Admin, I can browse registered users.
- As a Lunches Admin, I can browse days and see users' orders there.
- As a Lunches Admin, I can add items in menu only for today by adding a name and price.
- As a Lunches Admin, I can upload photo for each menu item
- As a Lunches Admin, on the date page I can see the list of orders and total lunch cost for today.

Let's assume that instead of calling to the Lunches Delivery Company, they have a smart ERP which can request the order for today through our API at a specific time.

- As a system, I should be able to provide list of the orders for today with details for each person through RESTful JSON API endpoint.
- As a system, I should have a secure API.

General Notes

- Keep your code clean.
- Thin controllers, fat models. Don't forget about validations
- Prefer quality to the speed of delivering this task
- The code should be covered with unit, functional and integration tests. (Please use RSpec+Capybara)
- Front-end can be pretty very simple.
- The source code of application should be uploaded on github.com and deployed to heroku in the end.
- Connect Travis-CI to your public repository

Recommended technologies

If you know your way, skip this step. Otherwise we are expecting from you to use these gems in project - Rails 4+, devise, twitter bootstrap, haml, simple_form, carrierwave, rspec, capybara



Next complexity level:

- Add organizations
- Add registration through Google Apps
- Auto Suggest for menu items
- Weekly Menus