Campus Eats – Design Document

Group: TarlBreuJacoBaraKnor
Date: 2025.10.23
Version: 1.0

Summary

Campus Eats is like Foodora for the campus. Any student on the campus can register as a user, view a menu of items that are available to order, pay for the items and have them brought to the location they specify.

This design document is meant to give a brief overview of our implementation of the Campus Eats software project, and won't get into great depth about the implementation, but should give the reader an idea of the direction we are taking to achieve the goals set by the requirements.

Universitetet
i Stavanger

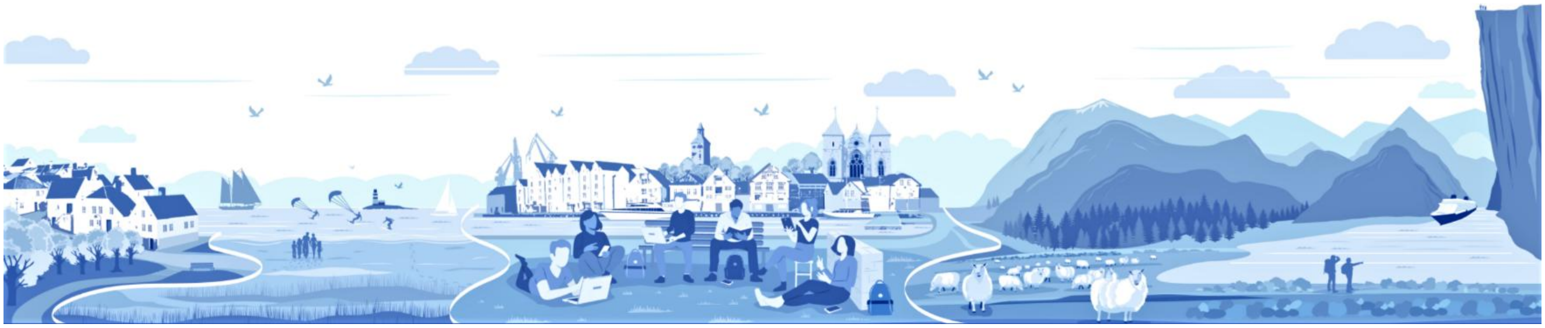

Table of Contents

Overview

Campus Eats is a food delivery service. Any student on the campus can register as a user, view a menu of items that are available to order, pay for the items and have them brought to the location they specify.

The web application has three main actors with associated workflows: Customer, Courier and Admin.

The definition of the previously state actors is as follows:

- A **customer** is a person that orders food. We'll need basic contact information like name, phone and email.
- A **courier** is person that responds to order requests, collects food and delivers it.
- An **administrator** is person that manages the site and its content.

Technologies

Category	Technology	Why It's used
Programming language	C#	We've been using C# for all the assignments and everybody in the group can develop with.
Main Framework	ASP.NET Core 8.0	Cross-platform, high-performance framework for building secure web applications and APIs.
Frontend	Razor Pages & Bootstrap	Having a hard look at the requirements, we have shifted our priorities to creating a good, functional webpage. Since the design requirements are not as rigorous, we have decided to use Bootstrap.
Database	PostgreSQL	Powerful, open-source relational database with strong ACID compliance, advanced indexing, JSON support, and better scalability than SQLite. Ideal for production and concurrent users.
Containerization	Docker	Packages all services (web, database) for consistent local and deployment environments.
Version Control	GitHub	Enables collaborative work, version history and issue tracking
Configuration Management	Environment Variables and appsettings.json	Provides flexible configuration.
Group Organization	Discord	Discord is the social network that everybody in the group has. We created a server for the project.
Task Management	Trello	Project management tool that permits to make tasks and make agile methodology.

Endpoints

Two design types were available to use for the endpoint design: either the resource-based, or the action-based URL design. We have decided to use a combination of the two, so we can leverage the advantages of both design types. Clarity in our endpoint naming is key for future maintainability, improvements, and developer experience. The endpoints are subject to change seeing as the requirements could change over the course of the project.

User specific endpoints

REST Method(s)	Endpoint	Description
GET	/	Gets the landing page of the web application
GET, POST	/register	Gets the registration page that lets a customer register as a new user
GET, POST	/login	Gets the login page where users/couriers can login
GET	/orders	Gets the order overview of a specific user with buttons
GET	/orders/{id}	Gets the details of a specific order
POST	/orders/ cancel/{id}	Cancels a specific order
POST	/orders/pay/{id}	Pays for a specific order
POST	/orders/tip/{id}	Tips the courier for the specific order
GET	/cart	Gets the contents of a specific user's shopping cart
GET, POST	/checkout/{id}	Takes a customer to the checkout page based on their cart id and lets them fill out a form to place their order
GET	/profile/{id}	Lets a user see their current profile information
GET, POST	/profile/edit/{id}	Lets a user edit their profile information

Courier specific endpoints

REST Method(s)	Endpoint	Description
GET	/dasboard/{id}	Gets the dashboard of a courier
GET	/orders/avaliable	Lets couriers see avaliable orders
POST	/orders/accept/{id}	Lets a courier accept a specific order
GET	/orders/history	Lets a courier see their order history
GET	/earnings/{id}	Lets a courier see their total earnings

Admin specific endpoints

REST Method(s)	Endpoint	Description
GET	/products	Gets the product page where admins can edit food item info
POST	/products/create	Creates a new food item
GET, DELETE	/products/{id}	Gets info about a specific food item with edit forms. (GET) Removes a food item based on id. (DELETE)
GET	/couriers	Gets an overview of the currently registered couriers and their status
POST	/couriers/approve/{id}	Approves or declines a courier's courier registration
GET	/admin/dashboard	Gets an overview of orders and their statuses
POST	/admin/login	Separate login for admins
POST	/admin/invite/{id}	Let's an admin invite a user to be an admin
GET	/revenue	Gets an overview of total revenue and how it has been split with the couriers

Authors

Name	Role
Lilian Breux	Chief Visionary Officer
Daniel Gaudland Jacobsen	Senior Lead Full Stack Engineer
Emma Knorová	Lead Software Architect
Théo Barat	French DevOps Lead
András Tarlós	Food Delivery Manager

Revision History

Version	Date	Changes
1.0	2025.10.23	Created initial design document.