

# Online Food Ordering System

## 1. Introduction

An online food ordering system. Every restaurant, diner etc. can use this system. It's build to manage products and orders being placed. Customers can use it to find restaurants, diners etc. to order food. The company will have a nice overview of all the orders that are placed and they can view and edit all the products they offer. The customers can view these companies, order food and place them as favorites.

### **Expected List of Features**

As a company I should be able to...

- create an account, so that I can login.
- add and edit my accounts information.
- insert products in my product catalog.
- view, edit and delete my products.
- place a product out of stock.
- view all of my orders..

As a customer I should be able to...

- create an account, so that I can login.
- browse restaurants.
- view the profile/menu of a restaurant.
- place restaurants as my favorite.
- place products in my shopping cart.
- view my shopping cart and checkout.

As an admin I should be able to...

- login.
- view every restaurant and customer.
- add a restaurant to the list of featured restaurants

## 2. Design and Implementation

### 2.1 The REST API Specification

Routes	HTTP Methods
/companies	get - post
/companies/user/:userId	get
/companies/:companyId	get - put - delete
/companies/:companyId/comments	get - post
/companies/:companyId/comments/:commentId	get - put - delete
/companies/:companyId/dish-categories	get - post
/companies/:companyId/dish-categories/:dishCategoryId	get - put - delete
/companies/:companyId/dishes	get - post
/companies/:companyId/dishes/:dishcategoryId	get
/companies/:companyId/orders	get
/company-categories	get - post
/company-categories/:categoryId	get - put - delete
/dishes	get - post
/dishes/:dishId	get - put - delete
/dishes/company/:companyId	get - post
/orders	get - post
/orders/:orderId	get
/users	get
/users/:userId	get - put - delete
/users/:userId/favorites	get - post
/users/:userId/favorites/:favoriteId	delete
/users/:userId/orders	get
/users/register	post
/users/login	post
/users/logout	get
/users/facebook	get
/users/facebook/callback	get

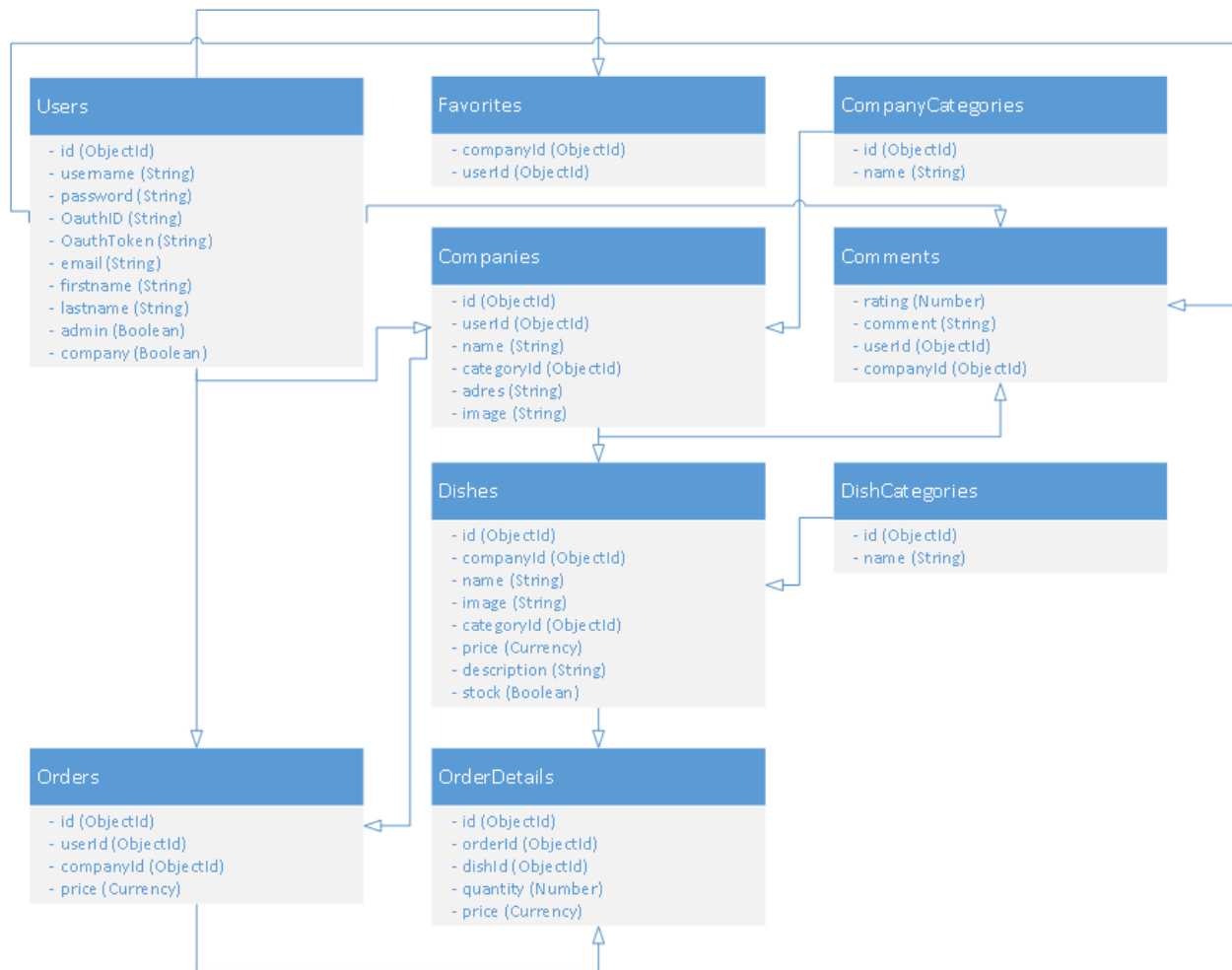
**Justification:** In my code I've placed comments for every route so it's clear what each route does. While looking at it you'll notice that I've done most of my querying in the backend. Some things I could've done in the frontend but I chose to do it on the server. I did this to make the processing in the frontend less. The processing in the backend will be more but by optimizing the server it'll take care of it.

## 2.2 Front-end Architecture Design

URL	TemplateUrl	Controller
/	views/front/home.html	-- NOT DONE --
/aboutus	views/front/aboutus.html	-- NOT DONE --
/contactus	views/front/contactus.html	-- NOT DONE --
/faq	views/front/faq.html	-- NOT DONE --
/favorites	views/front/favorites.html	FavoritesCtrl
/profile	views/front/profile.html	ProfileCtrl
/profile-edit	views/front/profile-edit.html	ProfileEditCtrl
/companies	views/front/companies.html	CompaniesCtrl
/companies/:id	views/front/company-menu.html	CompaniesMenuCtrl
/company-profile/:id	views/front/company-profile.html	CompaniesMenuCtrl
/orders	views/front/orders.html	OrdersCtrl
/order-checkout	views/front/order-checkout.html	-- NOT DONE --
/order-finish	views/front/order-finish.html	-- NOT DONE --
/admin/	views/admin/dashboard.html	LoginCtrl
/admin/category	views/admin/category.html	CategoryCtrl
/admin/category/add	views/admin/category-add.html	CategoryAddCtrl
/admin/category/:id	views/admin/category-details.html	CategoryDetailsCtrl
/admin/dishcategory	views/admin/dishcategory.html	DishcategoryCtrl
/admin/dishcategory/add	views/admin/dishcategory-add.html	DishcategoryAddCtrl
/admin/dishcategory/:id	views/admin/dishcategory-details.html	DishcategoryDetailsCtrl
/admin/products	views/admin/products.html	AProductsCtrl
/admin/product/add	views/admin/product-add.html	ProductAddCtrl
/admin/products/:id	views/admin/product-details.html	ProductDetailsCtrl
/admin/profile	views/admin/profile.html	AProfileCtrl
/admin/orders	views/admin/orders.html	AOrdersCtrl
/admin/users	views/admin/users.html	AUsersCtrl
/admin/companies	views/admin/companies.html	ACompaniesCtrl

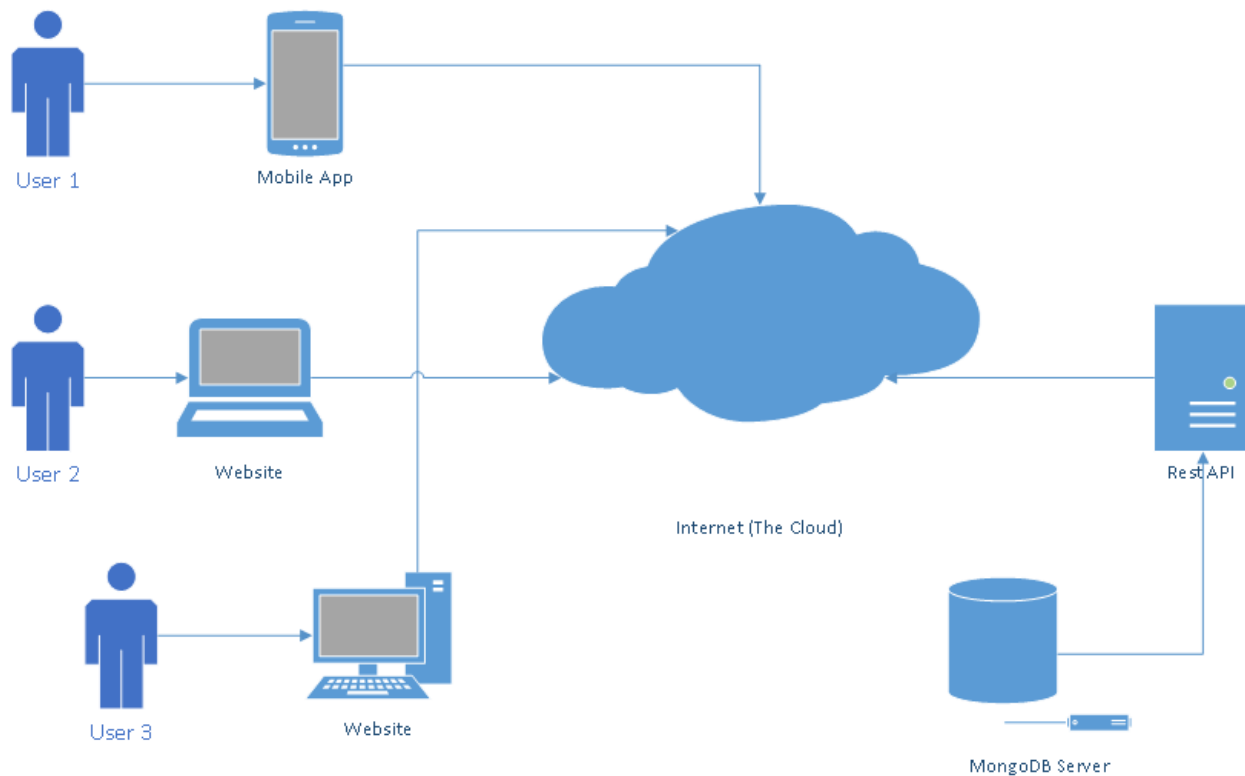
**Justification:** I deliberately chose to separate my views into two folders (front & admin). This makes it much easier to organize the views. I also chose to make a separate folder for my controllers and a separate file for every controller within that folder. It would be very unorganized if I would've left all the controllers in one folder.

## 2.3 Database Schemas, Design and Structure



**Justification:** If you view the database schemas you'll notice that there are a lot of embedded documents. I make use of embedded documents so the data is easier to query and easier to track. There are also a lot less routes because I'm using the embedded documents.

## 2.3 Communication



**Justification:** I used a separate MongoDB server which I use to host my MongoDB database. It was very straight forward to implement the database into the application thanks to hosting it separate.

### 3. Conclusions

By developing this project I learned a lot. Not only did I use the knowledge received in the five courses but I went deeper and tried to develop an app that's more complicated than the app developed through the courses. It was supposed to be a full fledged Content Management System connected to a website and a mobile app.

The current project isn't done and I would call it a draft but it's much better than I expected it to be. I'm also impressed with how I structured everything and how far I came with the project because it's a complicated and big project.

#### **Features not developed**

As a company I should be able to...

- edit my accounts information (upload a picture).
- place a product out of stock.

As a customer I should be able to...

- place products in my shopping cart.
- view my shopping cart and checkout.

As an admin I should be able to...

- add a restaurant to the list of featured restaurants

**Website:** <http://dsm-fullstack.herokuapp.com/app/#/>

**Admin panel:** <http://dsm-fullstack.herokuapp.com/app/#/admin/>

#### **Some login data**

- (Customer) username: marlene | password: password
- (Company) username: jmjake | password: password
- (Admin) username: stanley | password: password

### 4. References

- Code Hosting: <https://bitbucket.org/>
- Cloud Hosting: <https://heroku.com/>
- MongoDB Hosting: <https://mlab.com/>