

# 1. Introduction

## 1.1. Purpose

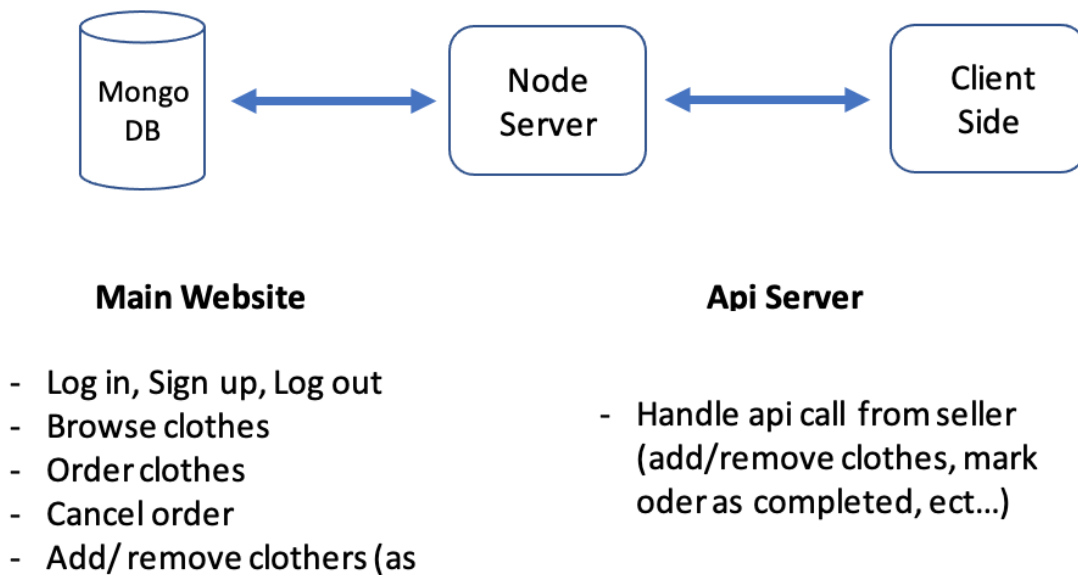
This is a document describing business needs and technicality for development of an e-commerce web app.

## 1.2. Proposal

We run a business of selling clothes. We want to create a merchandising web app that takes our business online to expand the market. The app is an MVP and needs to be done within 3 weeks so that we can test the market with it. We want to take advantage of the latest technology such as Node, ES6 and React.

## 2. Technical details

The app consists of 2 parts: a website and an api server. The website is a simple e-commerce site that lists clothings that we add. The api server handles requests made by the seller such as adding or removing listings. We want our api server to be developed using TDD and is 100% covered with unit tests. We also want gitflow workflow to be applied and ES6 to be used in the code.



## 3. User Roles

### 3.1. Customer

- Log in
- Log out
- Sign up
- Browse clothes

- Order clothes
- Cancel order

### 3.2. Seller

- There is only 1 seller, created beforehand (like an admin)
- Log in
- Log out
- Add clothes
- Remove clothes
- Edit clothes details
- Complete order

## 4. User Stories

*Note: User refers to both Customer and Seller*

### 4.1. Mandatory

ID	As a(n)	I want	Criteria
1	Customer	To sign up with email	
2	User	To login with email	
3	User	To log out	
4	Customer	To browse clothes	<ul style="list-style-type: none"> <li>• Display 20 listings per page</li> <li>• Pagination must be handled on the server</li> </ul>
5	Customer	To see what sizes are available	<ul style="list-style-type: none"> <li>• There are only 3 sizes: S, M, L</li> <li>• Each size has different number of items in stock</li> </ul>
6	Customer	To see the number of items in stock of a listings	
7	Customer	To add to cart	

8	Customer	To remove items from my cart	
9	Customer	To place order	<ul style="list-style-type: none"> <li>The ordered items must be taken into account when showing the number of items in stock</li> </ul>
10	Customer	To receive a confirmation email when I order	
11	Customer	To cancel my order	
12	Seller	To add a new listing	
13	Seller	To remove a listing	
14	Seller	To update number of items in stock of a listing	
15	Seller	To receive an email when an order is placed	
16	Seller	To receive an email when an order is cancelled	
17	Seller	To view pending orders	
18	Seller	To mark pending order as completed	

#### 4.2. Optional (Nice to have)

ID	As a(n)	I want	Criteria
19	User	To change my password	
20	Customer	To filter by size	<ul style="list-style-type: none"> <li>Filtering must be handled on the server</li> </ul>
21	Customer	To filter by availability (not out of stock)	<ul style="list-style-type: none"> <li>Filtering must be handled on the server</li> </ul>
22	Seller	To tag listing with a category when adding a new listing	

23	Seller	To update category of an existing listing	
24	Customer	To filter by category	<ul style="list-style-type: none"> <li>Filtering must be handled on the server</li> </ul>
25	Customer	To rate a listing that I bought	<ul style="list-style-type: none"> <li>The user has order this listing and the order contains this listing must be completed</li> <li>Use the standard 5-star system</li> </ul>

## 5. Constraints

- All pages must load under 2 seconds
- There must be no errors in the web browser's console unless introduced by third-party libraries
- All critical or sensitive interaction must be accompanied by a confirmation pop up (Are you sure?)