

User Manual
Project:
Cafeteria Management System:
Reslove

T-RISE

Rendani Dau (13381467)

Elana Kuun (12029522)

Semaka Malapane (13081129)

Antonia Michael (13014171)

Isabel Nel (13070305)

July 9, 2015

Contents

1	System Overview	3
2	System Configuration	3
3	Installation	3
3.1	Prerequisites	3
3.2	Setting up CMS	4
4	Getting Started	5
5	Using The System	5

1 System Overview

The Cafeteria Management System is a system designed to assist users with efficiently ordering food online as well to assist cafeteria staff with dealing with orders in real time as well as managing inventory. The system is intended to be used in a corporate environment whereby users have the option to charge their cafeteria expenses to their salary or immediately pay for orders. In addition, the system will allow management to view expense reports of different users.

2 System Configuration

The system requires a Windows/Unix based host to run the server. This host must have NodeJS and MongoDB installed. In addition, the host must be connected to the internet in order to allow any required dependencies to be installed and set up for the operating system environment. The configuration of the server requires an active E-mail account to facilitate communication between the system and end users. Interaction with this host will be achieved using a standard mouse and keyboard as well as a monitor.

End users will only require a PC equipped with a web browser and an active internet connection.

3 Installation

3.1 Prerequisites

The Cafeteria Management System requires NodeJS and MongoDB to run. These are free and open source software and can be obtained from the following sites:

<https://nodejs.org/download/>

<https://www.mongodb.org/downloads>

The applications are available for both Windows and Unix environments and include setup guides on their respective sites.

Once installed, NodeJS includes a package manager called NPM. This package manager will be available from the terminal and will be used to

install all dependencies. The following dependencies have to be installed first:

```
$npm install -g bower
$npm install -g grunt-cli
```

3.2 Setting up CMS

Before starting the system, an email account has to be set up to facilitate communication between the system and end users. The details of this account can be configured in the following config file:

```
~/Cafeteria Management System/config/env/production.js
```

Under the section 'Mailer', the following fields should be specified:

- MAIELR_FROM: A name indicating the sender of mail.
- MAILER_SERVICE_PROVIDER: The service provider of the email account
- MAILER_EMAIL_ID: The email ID of the account set up for CMS
- MAILER_PASSWORD: The password of the account set up for CMS

In a terminal/command prompt, navigate to the CMS directory and execute the 'npm install' command. This will install all the packages required to run the system:

```
~/Cafeteria Management System$ npm install
```

If all dependencies were installed successfully, then MongoDB can be started with the following command:

```
~$mongod --dbpath "directory"
```

Where directory is a path to the folder which Mongo will use as a working directory.

Note that this command has to be executed in a separate terminal.

Once mongo has been started, then the CMS server can be started with the following command:

```
~/Cafeteria Management System$ grunt
```

4 Getting Started

Access to the Cafeteria Management System is through a standard web browser. Different types of users have access to different facets of the system. The system has a default Superuser account which can assign different roles (cafeteria manager, cashier, etc.) to the users. These users can then sign in to access the facet of the system they are authorised to.

Once logged in, the superuser can change their password from the profile management page which is available directly from the home page. They can also set additional fields such as an E-mail address which will assist in password recovery in the event of a forgotten password.

walkthrough from initiation to exit*

To terminate the sever, the user can enter the Ctrl+C command in the terminal. The user can also terminate the MongoDB service by executing the same command (Ctrl+C) in the MongoDB terminal.

5 Using The System