



School of Electrical Engineering and  
Computer Science

# COS5019-B Enterprise Pro

## **Story Cards Document**

Project: My Supermarket Shop

Team 22

## Table of Contents

Introduction .....	3
Story Cards .....	3
Login .....	3
Admin Login .....	3
Logout .....	3
Register .....	4
View Products .....	4
View Supermarkets .....	4
Add to Basket .....	4
Order from Basket .....	5
View Basket .....	5
View Orders .....	5
Empty Basket .....	6
Admin View Orders .....	6
Admin Complete Order .....	6

## Introduction

The following document will address all the story cards regarding the project and their functional requirements.

## Story Cards

This section will consist of all the story cards.

### Login

Story: Login
Input: Username, Password
Background: The user navigates to login screen and inputted their username and password and press Login.
Output: User is redirected to the index page with their name showing and having the Order page available.
Prerequisites: <ul style="list-style-type: none"><li>• User should exist with inputted username and password.</li><li>• Username and password should be filled and not empty.</li></ul>
Implementation: User is logged in and their username is saved in a cookie.
Related Stories: Logout

### Admin Login

Story: Admin Login
Input: AdminUsername, Admin Password
Background: The admin navigates to login screen and inputted their username and password and press Login.
Output: Admin is redirected to a specific page where all the orders are available.
Prerequisites: <ul style="list-style-type: none"><li>• Admin username and password should be correct.</li></ul>
Implementation: Admin has the right to see all the order created on the platform. All the orders from each user is visible.
Related Stories: Login

### Logout

Story: Logout
Input: Press Button "Logout"
Background: User would like to disconnect from service.
Output: User is not signed in the platform any more.
Prerequisites: <ul style="list-style-type: none"><li>• User should be already signed in.</li></ul>
Implementation: The cookie is erased with the username of the logged in user.

Related Stories: Login
------------------------

#### Register

Story: Register
-----------------

Input: Username, password, email, and date of birth
-----------------------------------------------------

Background: Users wish to create an account on the platform.
--------------------------------------------------------------

Output: User is created.
--------------------------

Prerequisites:
----------------

- User with inputted username should not already exist.
- All fields should be filled.

Implementation: A new node is created under /Users/ in the database holding the user credentials. A check is done beforehand if the username already exists.
--------------------------------------------------------------------------------------------------------------------------------------------------------------

Related Stories: Login
------------------------

#### View Products

Story: View Products
----------------------

Input: User navigate to main page and sees the product list.
--------------------------------------------------------------

Background: User views available products to pick from in a list.
-------------------------------------------------------------------

Output: List is present with all available products.
------------------------------------------------------

Prerequisites: None
---------------------

Implementation: Products are drawn from the database and presented in the list.
---------------------------------------------------------------------------------

Related Stories: Search Supermarkets
--------------------------------------

#### View Supermarkets

Story: Search Supermarkets
----------------------------

Input: User has selected a product and views a list of supermarkets that offer it.
------------------------------------------------------------------------------------

Background: The user wants to view which supermarkets offer the selected product.
-----------------------------------------------------------------------------------

Output: Boxes with the product and the quantity and price of each in every supermarket that offers it.
--------------------------------------------------------------------------------------------------------

Prerequisites:
----------------

- User should have picked a product.

Implementation: The /Product/product node is taken and the supermarkets that offer it are inside and displayed with the name and price attribute.
---------------------------------------------------------------------------------------------------------------------------------------------------

Related Stories: View Products
--------------------------------

#### Add to Basket

Story: Add To Basket
----------------------

Input: User sets the desired quantity and presses "Add to Basket"
-------------------------------------------------------------------

Background: The user wants to add some items to the basket.
-------------------------------------------------------------

Output: The selected product, its quantity and price are added to the basket.
Prerequisites: <ul style="list-style-type: none"> <li>Quantity must be other than zero (0).</li> </ul>
Implementation: The “basket” cookie is created and add a string formatted in a way that saves all the necessary information, every time the string is updated to add more items.
Related Stories: View Supermarkets

#### Order from Basket

Story: Order from Basket
Input: User pressed “Order” in Basket page, filled in their personal information and ordered.
Background: User wishes to purchase the items in the basket.
Output: The order is submitted successfully.
Prerequisites: <ul style="list-style-type: none"> <li>User must have some items in the Basket.</li> <li>User must input and fill all the fields in the Order Details.</li> </ul>
Implementation: The /Orders/ node is updated where the “basket” cookie is pushed along with the details of the user’s sending location.
Related Stories: View Basket

#### View Basket

Story: View Basket
Input: User clicks on Basket link
Background: User wishes to see all available items they have added to basket.
Output: User view all items added to basket.
Prerequisites: <ul style="list-style-type: none"> <li>User should have added items to the Basket to view them.</li> </ul>
Implementation: The “cookie” is unfolded and present in a table format.
Related Stories: Add To Basket

#### View Orders

Story: View Orders
Input: User clicks on Orders link.
Background: The user wishes to see all pending orders.
Output: Users view all orders made along with Order ID.
Prerequisites: <ul style="list-style-type: none"> <li>Order should be made.</li> <li>User should be signed in.</li> </ul>

Implementation: the /Orders node draws all the order that have the “username” attribute same as the logged in user.
---------------------------------------------------------------------------------------------------------------------

Related Stories: Order From Basket
------------------------------------

#### Empty Basket

Story: Empty Basket
---------------------

Input: User clicks on “Empty Basket” in Basket page
-----------------------------------------------------

Background: User wishes to remove all items from the basket.
--------------------------------------------------------------

Output: No items in the user’s basket.
----------------------------------------

Prerequisites:
----------------

- Basket should have items to be emptied.

Implementation: The cookie “basket” is erased.
------------------------------------------------

Related Stories: View Basket, Add To Basket
---------------------------------------------

#### Admin View Orders

Story: Admin View Orders
--------------------------

Input: Admin logs in and presses Orders link
----------------------------------------------

Background: Admin wishes to see all orders made on the platform.
------------------------------------------------------------------

Output: Admin sees all orders.
--------------------------------

Prerequisites:
----------------

- Orders should be made by users.
- Admin should be signed in.

Implementation: All the items from the /Orders node from the database are drawn.
----------------------------------------------------------------------------------

Related Stories: View Orders
------------------------------

#### Admin Complete Order

Story: Complete Order
-----------------------

Input: Admin Presses “Complete Order” on Orders page.
-------------------------------------------------------

Background: Admin wishes to mark an order as completed.
---------------------------------------------------------

Output: order no longer exists.
---------------------------------

Prerequisites:
----------------

- Order should exist.
- Admin should be signed in.

Implementation: The node with the ORDER_ID will be removed from the /Orders node.
-----------------------------------------------------------------------------------

Related Stories: View Orders
------------------------------