**CST 391: Milestone 1**

Michael Melichar

CST 391: JavaScript Web Application Development

Dinesh Gudibandi

5/14/23

**Introduction:**

The Product Management System is a web application designed to support the management of various products. The application allows users to perform CRUD operations on products, which are represented as objects with multiple properties. The system will consist of back-end services developed using Express and Node.js, a front-end web application built with Angular, and a second front-end application developed using React.

**Functionality Requirements:**

•As a user, I want to be able to view a list of products, so that I can see what products are available.

•As a user, I want to be able to create a new product, so that I can add new items to the inventory.

•As a user, I want to be able to edit an existing product, so that I can update its information if needed.

•As a user, I want to be able to delete a product, so that I can remove items that are no longer needed.

•As a user, I want to be able to view a list of categories, so that I can see how products are organized.

•As a user, I want to be able to create a new category, so that I can group products based on their characteristics.

•As a user, I want to be able to edit an existing category, so that I can modify its properties if necessary.

•As a user, I want to be able to delete a category, so that I can remove unnecessary groupings.

•As a user, I want to be able to view the details of a product, including its properties and associated category.

•As a user, I want to be able to view the details of a category, including its properties and the list of associated products.

**Initial Database Design:**

A picture containing text, screenshot, line, diagram

Description automatically generated

**Initial UI Sitemap:**

A screenshot of a computer

Description automatically generated with medium confidence

*In this example, almost every button the user could possibly interact with will send them to the login screen except the check boxes, which should remember their state after a user logs in. Once the user is logged in, they will instead see their username on the top right where the log in button is currently located.*

**Initial UML Classes:**

* Product:

Properties: id, name, price, description, category, quantity

Methods: getters and setters for properties

* Category:

Properties: id, name

Methods: getters and setters for properties

* DatabaseConnection:

Methods: connect(), disconnect(), executeQuery(), executeUpdate()

* ProductService:

Methods: getAllProducts(), getProductById(id), createProduct(product), updateProduct(product), deleteProduct(id)

* CategoryService:

Methods: getAllCategories(), getCategoryById(id), createCategory(category), updateCategory(category), deleteCategory(id)

**Risks:**

The project will identify and track unknowns and risks that may arise during development, ensuring appropriate mitigation strategies are in place to address them effectively.

The design report will document all technical decisions and designs, including UML diagrams, ER diagrams, UI designs, and other artifacts necessary to support the end-to-end solution and application design.

**API Endpoints**

* GET /api/users: Retrieves a list of all users in the system.
* POST /api/users: Creates a new user based on the provided data in the request body.
* PUT /api/users/:id: Updates the user with the specified ID using the data provided in the request body.
* DELETE /api/users/:id: Deletes the user with the specified ID from the system.