# **Smart Shopping Cart System**

# **Project Plan**

**Sprint 0: High-Level Analysis and System Architecture (Sep 22 - Sep 28)**

* **Goal:** Perform a high-level analysis to define the architecture and key components of the Smart Shopping Cart System.
* **Key Activities:**
  + Requirements Gathering
  + Use Case Diagram
  + Component Identification
  + System Architecture
  + Technology Stack Selection
* **Outcomes:**
  + Use Case Diagram (overall system functionality)
  + Component Diagram (high-level structure of components)
  + Architecture Diagram (how components interact)
  + Preliminary Project Roadmap (breakdown of sprints)

**Sprint 1: Accounts Management and first seminar preparation (Sep 29 - Oct 12)**

**Objectives:**

* Enable the system admin to add store/mall owners and manage their accounts.
* Enable the store owners to set up their account password.
* Enable the store owners to login.
* Prepare a presentation for the first seminar.

**Tasks:**

1. **Requirement Gathering**:
2. **Analysis**:
   * Identify the main actors: Store Manager, and system admin.
   * Define functional requirements
3. **Design**:
   * **Use Case Diagram**
   * **Class Diagram**
   * **Sequence Diagram**
4. **Implementation**:
   * **System Admin Dashboard**:
     + Interface for adding and managing store owners.
     + Password setup interface.
     + Login interface.
   * **Testing**:
     + Ensure the admin can successfully add store owners.
     + Ensure store owners can successfully setup their account password.
     + Ensure store owners can successfully log into their dashboards.

**Deliverables:**

* Store manager’s interface for defining store zones and product locations.
* Working shopping list management (UI + backend).
* Diagrams and documentation.
* First Seminar Presentation

**Sprint 2: Store Setup and Shopping List Management (Oct 13 - Oct 24)**

**Objectives:**

* Enable the store/mall manager to add products, categories, etc.
* Enable the store/mall manager to define the store's layout (store zones, product locations).
* Allow users to create and manage their shopping lists.

**Tasks:**

1. **Requirement Gathering**:
   * Gather requirements for how the store's structure and product locations will be defined by the store manager.
   * Collect requirements for shopping list creation and management by users.
2. **Analysis**:
   * Identify the main actors: Store Manager, Customer(user), and system admin.
   * Define functional requirements
     + Store manager must be able to define the store layout (zones) and assign products to specific locations within the store.
     + Users must be able to create, update, and manage shopping lists.
3. **Design**:
   * **Use Case Diagram**
   * **Class Diagram**
   * **Sequence Diagram**
4. **Implementation**:
   * **Store Manager Dashboard**:
     + Interface for the manager to define store zones, set up product locations, add product and categories.
   * **Shopping List Management (User)**:
     + Frontend: UI for adding/removing items to/from the shopping list.
     + Backend: Store shopping lists in a database, define API endpoints for shopping list CRUD operations.
   * **Testing**:
     + Validate store zone definition.
     + Test shopping list creation and management functionality.

**Deliverables:**

* Store manager’s interface for defining store zones and product locations.
* Working shopping list management (UI + backend).
* Diagrams and documentation.

**Sprint 3: Product Search, Catalog, and Basic Navigation (Oct 25 - Nov 9)**

**Objectives:**

* Enable users to search for products from a catalog and display product details.
* Provide basic navigation to products using the store zones defined in Sprint 1.

**Tasks:**

1. **Requirement Gathering**:
   * Define the product search functionality (by name, category, price).
   * Gather requirements for basic navigation based on predefined store zones.
2. **Analysis**:
   * Define functional requirements:
     + Users can search for products by name, category, or other filters.
     + Products will be displayed with details (price, availability, location in the store).
     + Users will see basic navigation to product zones (no dynamic pathfinding).
3. **Design**:
   * **Use Case Diagram**: For product search and navigation.
   * **Class Diagram**:
   * **Activity Diagram**: For searching a product and navigating to its location.
4. **Implementation**:
   * **Frontend**:
     + Search bar with filtering options.
     + Product catalog page displaying product details and zone location.
   * **Backend**:
     + Set up APIs to handle product search queries and retrieve catalog information.
     + Integrate product locations with store zones for navigation.
   * **Testing**:
     + Test search accuracy, filtering options, and navigation to product zones.

**Deliverables:**

* Product search and catalog interface.
* Basic product navigation (using store zones).
* Diagrams and test results.

**Sprint 4: Store Map and Static Navigation (Nov 10 - Nov 25)**

**Objectives:**

* Implement a static store map with product location overlays based on store zones.

**Tasks:**

1. **Requirement Gathering**:
   * Define how the store map will be displayed and how product locations will be shown.
2. **Analysis**:
   * Functional requirements:
     + Users can view a store map with product locations marked.
     + Products from the search will be linked to the map.
3. **Design**:
   * **Use Case Diagram**: For interacting with the map and viewing product locations.
   * **Class Diagram**:
     + **Map**: Layout, zones, product overlays.
   * **Sequence Diagram**: For linking searched products to map locations.
4. **Implementation**:
   * **Frontend**:
     + Static store map with product locations shown as pins or markers.
   * **Backend**:
     + API to retrieve product locations and map them to the static store layout.
   * **Testing**:
     + Test the accuracy of product location markers on the map.

**Deliverables:**

* Working static store map with product locations.
* Diagrams and test results.

**Sprint 5: Pathfinding and Navigation (Nov 26 - Dec 11)**

**Objectives:**

* Enable pathfinding from the store entrance to the desired product location.

**Tasks:**

1. **Requirement Gathering**:
   * Define pathfinding requirements (shortest route to products).
2. **Analysis**:
   * Functional requirements:
     + System calculates and displays the shortest path from the entrance to the product.
3. **Design**:
   * **Use Case Diagram**: For navigation.
   * **Class Diagram**
   * **Sequence Diagram**: For calculating and displaying the path.
4. **Implementation**:
   * **Frontend**:
     + Visualize the path on the store map.
   * **Backend**:
     + Implement a pathfinding algorithm (e.g., Dijkstra's).
   * **Testing**:
     + Test pathfinding accuracy and user experience.

**Deliverables:**

* Functional dynamic navigation feature.
* Diagrams and test results.

**Sprint 6: Final Documentation and Presentation (Dec 12 - Dec 27)**

**Objective**: Complete project documentation and prepare for the final seminar presentation.

**Tasks:**

1. **Documentation**:
   * Write user guides, technical documentation, and a project report.
   * Prepare final system documentation including API endpoints, and database schema.
2. **Final Presentation Preparation**:
   * Prepare the final presentation slides and demo for the seminar.
   * Ensure the system is ready for live demonstration.

**Deliverables for Final Seminar**:

* Complete project documentation
* Final presentation slides and system demo