

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

# SmartCart Requirements Definition

Version 1.4

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

## Revision History

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
2011-10-20.	0.1	Initial Draft	FG, IS, LB, IC
2011-10-21	0.2	Revised draft	BW
2011-10-22	1.0	Revised draft no. 2 – ready for submission	IC
2011-10-25	1.1	Corrected use case table	IC
2011-10-27	1.2	Revised requirements	IC, LB
2011-11-06	1.3	Revised requirements	LB
2012-01-09	1.4	Updated Use case diagram and use cases, updated requirements	LB

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

## Table of Contents

1.	Introduction	4
1.1	Purpose of this document	4
1.2	Intended Audience	4
1.3	Scope	4
1.4	Definitions and acronyms	4
1.4.1	Definitions	4
1.4.2	Acronyms and abbreviations	4
1.5	References	4
2.	Requirements Description	5
2.1	Introduction	5
2.2	General requirements	5
2.3	Functional requirements	5
2.4	Non-functional requirements	5
3.	Use Case Models	6
3.1	Use case model	6
3.1.1	P-1: Search for available products	7
3.1.2	P-2: Search products using search box	7
3.1.3	P-3: Search products using barcode scanner	7
3.1.4	P-4: Browse products by categories	8
3.1.5	P-5: See product details	8
3.1.6	P-6: Edit the product price in the particular store	8
3.1.7	P-7: Add new product to the product list	9
3.1.8	L-1: Browse the shopping list	9
3.1.9	L-2: Add product to the shopping list	9
3.1.10	L-3: Remove the product from the shopping list	10
3.1.11	L-4: Edit the quantity of the product in the shopping list	10
3.1.12	S-1: Browse stores on the map	11
3.1.13	X-1 Edit the settings	11
3.1.14	F-1: Get the store suggestion	11
3.1.15	F-2: Get the stores nearest to the user	12
3.1.16	F-3: Get the stores by the shopping list's total price	12
3.1.17	F-4: Browse shopping list details for the suggested store	12
3.1.18	F-5: Show suggested stores on the map	13
4.	Requirements Definition	14
4.1	Requirement Group Definitions	14
4.2	Requirement Sources	14
4.3	Requirements definitions	14
4.3.1	Change Log	16
5.	Future Development	17
5.1	General Overview	17

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

## 1. Introduction

Nowadays, there are many stores which offer the same or similar products, but the prices often vary from store to store. In today's times of the recession, people are trying to buy products where they are cheaper and save money. This turns out to be a challenging task because of the sheer number of stores and products that they offer.

SmartCart is imagined as a mobile application which should ease up the daily process of buying groceries for the end users. SmartCart will enable users to enter groceries they want to buy to a list, and then find out in which store they can buy the cheapest groceries and save money, or find the closest store with the requested products. Although, there are some existing web sites and applications with purpose of finding cheapest items across multiple stores, they do it on the basis of a single item. SmartCart on the other hand, will do it for the whole list of groceries.

### 1.1 Purpose of this document

The purpose of this document is to describe the requirements that the system should implement. Good definition of requirements is very important because it is the fundament for the next phase of building the system. Requirements will be defined and categorized in the manner of essential or optional requirements, and for each requirement a use-case scenario will be presented. That way this document will also serve as a reference on what is expected from this project and when are the functionalities expected to be implemented.

### 1.2 Intended Audience

Intended audiences of this document are all project stakeholders:

- Project customer,
- Project supervisor,
- Team members,
- All persons responsible for monitoring the project.

### 1.3 Scope

Scope of this document includes definition of the requirements, elaboration of the use-case models and notes regarding future work on the SmartCart project.

### 1.4 Definitions and acronyms

#### 1.4.1 Definitions

Keyword	Definitions
SmartCart	Project name

#### 1.4.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
SVN	Subversion - version control system
WP7	Windows Phone 7

### 1.5 References

- ❖ Project homepage
  - <http://www.fer.unizg.hr/rasip/dsd/projects/smartcart>

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

❖ Project plan

➤ [http://www.fer.unizg.hr/download/repository/Project\\_Plan\\_v1.0.pdf](http://www.fer.unizg.hr/download/repository/Project_Plan_v1.0.pdf)

## 2. Requirements Description

### 2.1 Introduction

The purpose of this project is to provide the client with a mobile application which assists the client in their grocery shopping by providing them with the information about nearby shops and their product prices. Special emphasis will be placed on the usability and relevancy of the data provided.

### 2.2 General requirements

SmartCart must make the grocery shopping easier for the user and provide the functionality of finding the cheapest / closest store which has the products the user wants. System must be constantly updated with up-to-date prices either by gathering information from stores' web pages or by manual input from users.

### 2.3 Functional requirements

User shall be able to:

- ◆ browse all products in the system,
- ◆ add, modify quantity and remove products from the list (shopping list),
- ◆ add / update information about a product, including its price and the store where the product is located,
- ◆ get suggestions of stores according to his shopping list and selected criteria (products' price or store distance)
- ◆ browse stores nearby on map

Mobile application shall be able to:

- ◆ store list of products and nearby stores in memory for fast access,
- ◆ update the products and stores nearby on startup or on user demand,
- ◆ store user shopping list with quantity of products,
- ◆ display a single product's details (price in different stores),
- ◆ display the cheapest stores or the closest stores containing products in the list.

Server shall be able to:

- ◆ automatically collect products and prices from available store's web site (web scraper),
- ◆ communicate with mobile applications across different platforms,
- ◆ automatically insert the stores and their locations into database (store loader).

### 2.4 Non-functional requirements

- ◆ Mobile application shall be developed for Android mobile platform.
- ◆ Mobile application could be developed for WP7 mobile platform.
- ◆ Mobile application interface should be simple and user friendly.
- ◆ The communication with the server on a 3G connection cannot exceed 4hours continuously.
- ◆ The results shown to user (stores, products) should contain only a subset of total results, according to user's location and the distance radius selected by the user.

### 3. Use Case Models

#### 3.1 Use case model

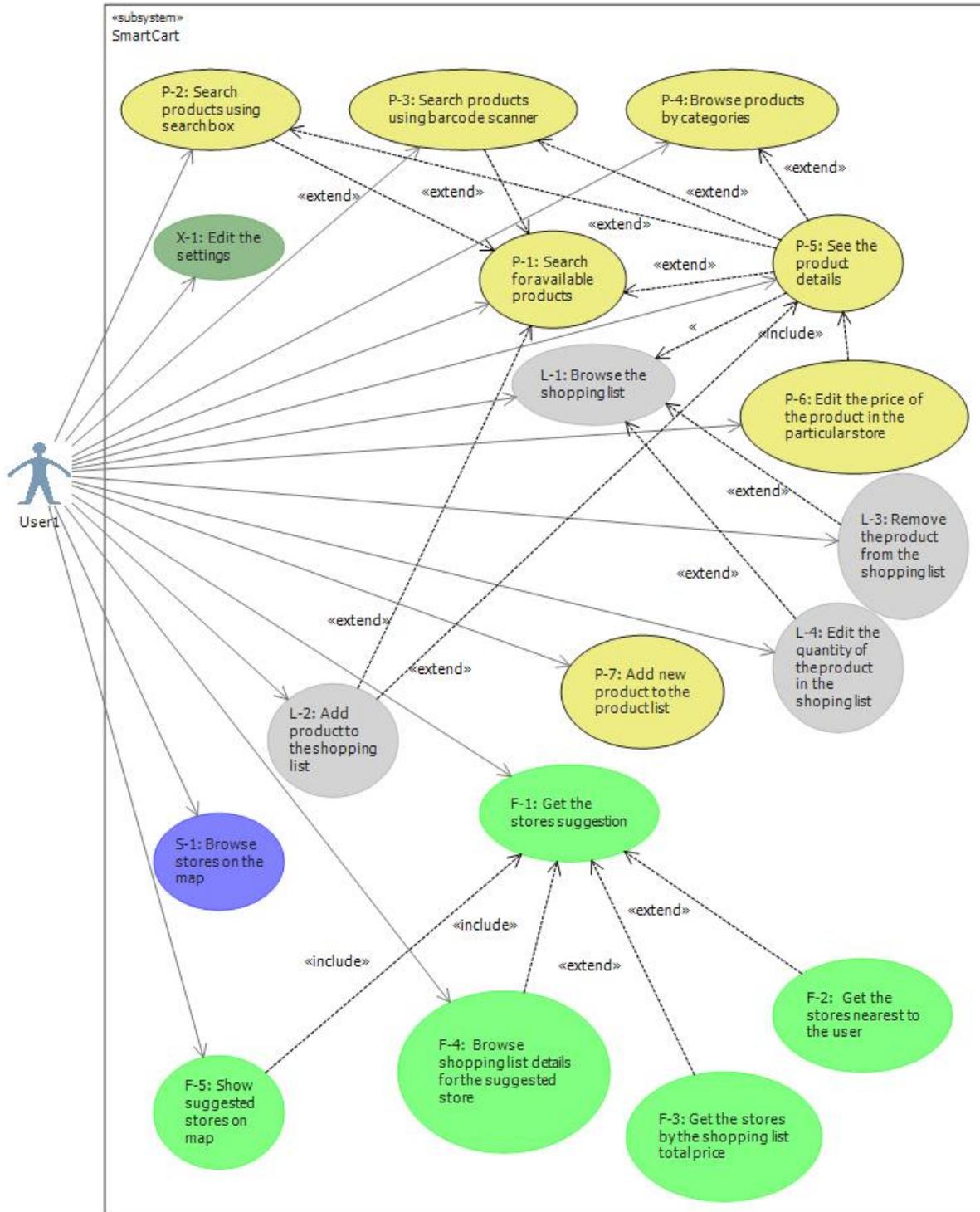


Figure 1 Use case model

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

### 3.1.1 P-1: Search for available products

**Initiator:**

*User*

**Goal:**

Display in the client application the product which user would like to add to his shopping list or view the details.

**Main Scenario:**

1. *The user clicks on the "Search" tab in the application*
2. *The application displays 2 search methods: by typing the products name (P-2) or by scanning the barcode (P-3).*
3. *The user chooses one of the methods*
4. *The searched product's details are shown*

### 3.1.2 P-2: Search products using search box

**Initiator:**

*User*

**Goal:**

Display in the client application the product which user would like to add to his shopping list or view the details.

**Main Scenario:**

1. *The user clicks on the "Search" tab in the application*
2. *The user starts typing the products name in the search box*
3. *The products whose names begin with the typed letters are displayed dynamically as the user types.*
4. *The user clicks on the product in the generated list*
5. *The searched product's details are shown*

### 3.1.3 P-3: Search products using barcode scanner

**Initiator:**

*User*

**Goal:**

Display in the client application the product which user would like to add to his shopping list or view the details.

**Main Scenario:**

1. *The user clicks on the "Search" tab in the application*
2. *The user clicks on the "Scan barcode" button*
3. *The user points the camera to the barcode on the product*
4. *The barcode is scanned*
5. *Barcode ID is sent to server*
6. *Server returns product details for the product with received ID*
7. *The searched product's details are shown*

**Extensions:**

- P-3a. The client application cannot connect to the webserver.*
1. *An error message is displayed explaining the internet connectivity problem.*

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

*P-3b. There is no product with scanned barcode ID in the database*

1. Server returns that there is no product with that barcode ID
2. App offers user to add the new product with scanned ID (P-7)

### 3.1.4 P-4: Browse products by categories

**Initiator:**

*User*

**Goal:**

See the list of available product sorted by categories

**Main Scenario:**

1. *The user clicks on the "Products" tab*
2. *The user chooses the category by which he'd like to filter the list of products from the drop down list*
3. *The application displays the list of products in selected category*

### 3.1.5 P-5: See product details

**Initiator:**

*User*

**Goal:**

View the detailed information about the product.

**Main Scenario:**

1. *The user is browsing the products list.*
2. *User clicks the product's name on the list.*
3. *The Application retrieves the product information from the server.*
4. *The application displays the product's details.*

**Extensions:**

*P-5a. The client application cannot connect to the webserver.*

1. *An error message is displayed explaining the internet connectivity problem.*

*P-5b. The user scanned the barcode instead of browsing the list of products*

1. *P-3 use case*

*P-5a. The user used the search box*

1. *User is positioned at in the Search screen*
2. *P-2 use case*

### 3.1.6 P-6: Edit the product price in the particular store

**Initiator:**

*User*

**Goal:**

*Update the product price in a particular store.*

**Main Scenario:**

1. *The user is viewing the product's details.*
2. *User clicks the "Update price" button.*
3. *User types in the price details.*

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

4. *User clicks the "Save" button.*
5. *The application sends the updated information to the server.*
6. *Server stores the new details.*
7. *The application displays the product details.*

**Extensions:**

- P-6a. The client application cannot connect to the webserver.*
1. *An error message is displayed explaining the internet connectivity problem.*

**3.1.7 P-7: Add new product to the product list**

**Initiator:**

*User*

**Goal:**

*Add the new product and the details about it to the product list and the system.*

**Main Scenario:**

1. *The user is browsing the products list.*
2. *User clicks the "Add New Product" button.*
3. *User types the details about the product.*
4. *User chooses the store where the product is.*
5. *User scans the barcode*
6. *User clicks the "Save product" button.*
7. *Application sends the new product and its details to the server.*
8. *The new product is stored in the phone memory.*
9. *The application displays the product's details.*

**Extensions:**

- P-7a. The client application cannot connect to the webserver.*
2. *An error message is displayed explaining the internet connectivity problem.*
- P-7b. The user is trying to add a product to a store which does not exist.*
1. *User adds the new store (Use Case S-1) where the product is.*
- P-7c. The user is trying to add a product with the price of 0.*
1. *The server will handle it as a malicious input, an error response is sent to User.*
  2. *The user re-enter the price (back to step3).*

**3.1.8 L-1: Browse the shopping list**

**Initiator:**

*User*

**Goal:**

*See the detailed information about the product.*

**Main Scenario:**

1. *The user opens the SmartCart application, opens the applications menu and clicks the "Shopping list" button.*
2. *The application displays the shopping list (list of selected products) stored in the phone memory.*

**Extensions:**

*None*

**3.1.9 L-2: Add product to the shopping list**

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

**Initiator:**

*User*

**Goal:**

*Adding the product from the product list to the user's shopping list.*

**Main Scenario:**

1. *The user is browsing the product list (Use case P-1).*
2. *The user clicks the "Add to shopping list" button.*
3. *The product is added to the shopping list.*
4. *The application displays the notification about adding the product to the list pad.*
5. *The application displays the list of products.*

**Extensions:**

*None*

*3.1.10 L-3: Remove the product from the shopping list*

**Initiator:**

*User*

**Goal:**

*Removing the product from the user's shopping list.*

**Main Scenario:**

1. *The user is browsing the shopping list (Use case L-1).*
2. *User clicks the "Remove" button next to the name of the product which he wants to remove from the shopping list.*
3. *The product is removed from the shopping list.*
4. *The application displays the shopping list (list of selected products) stored in the phone memory.*

**Extensions:**

*None*

*3.1.11 L-4: Edit the quantity of the product in the shopping list*

**Initiator:**

*User*

**Goal:**

*Change the quantity of a product which was added to the shopping list.*

**Main Scenario:**

1. *The user views the shopping list (see use case L-1 for details) and there is at least one product in the shopping list.*
2. *The user clicks a plus or minus button associated with the product to increase or decrease the quantity.*
3. *After each increase or decrease operation, the application saves the new value in the persistence storage.*

**Extensions:**

*None*

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

### 3.1.12 S-1: Browse stores on the map

**Initiator:**

*User*

**Goal:**

Display in the client application all stores which have been added to the system in the form of a list.

**Main Scenario:**

1. *The user opens the SmartCart application, opens the applications menu and clicks "Show all stores" button.*
2. *The application obtains the current location of the user.*
3. *The application displays all the stores stored in the database which are less than 5km away from the current location of the user.*
4. *A list of stores is displayed.*

**Extensions:**

*None*

### 3.1.13 X-1 Edit the settings

**Initiator:**

*User*

**Goal:**

Editing of the settings used to get suggestions and in other features

**Main Scenario:**

1. *The user clicks the menu button on his phone keyboard*
2. *The settings sign is shown at the bottom of the screen*
3. *User clicks the settings sign*
4. *The pop up screen with settings options is shown*
5. *Users edits the settings regarding the radius and method of suggesting stores*
6. *The settings are stored in the app*

### 3.1.14 F-1: Get the store suggestion

**Initiator:**

*User*

**Goal:**

Display a list of suggested stores for the shopping list.

**Main Scenario:**

1. *The user creates their shopping list and then clicks "suggest a store" button.*
2. *The application displays a popup dialog allowing the user to select the type of suggestion (see extensions F-2 and F-3 for possible suggestion types).*
3. *When the user clicks the suggestion type, a list of 5 stores matching the criterion is displayed.*

**Extensions:**

*None*

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

### 3.1.15 F-2: Get the stores nearest to the user

**Initiator:**

*User*

**Goal:**

Display a list of stores nearest to the user and which have all or most of the items on the user's shopping list.

**Main Scenario:**

1. *The use case starts when the user clicks the "show stores nearest to me" button.*
2. *The system finds all the shops within 5km of the user and checks which stores contain all the products from the shopping list.*
3. *The list of stores is displayed to the user in an ascending order based on the distance to the shop.*

**Extensions:**

*None*

### 3.1.16 F-3: Get the stores by the shopping list's total price

**Initiator:**

*User*

**Goal:**

Display a list of stores which have all or most of the items on the shopping list and whose total shopping price is the lowest.

**Main Scenario:**

1. *The user clicks the "show stores cheapest to me" button.*
2. *The system finds all the shops within 5km of the user and checks what is the total price of all the items in the shopping list and how many are available at the store.*
3. *The system selects from the list 5 stores which have the most articles and the lowest price.*
4. *The list of stores is displayed to the user in an ascending order based on the total price of all the items in the shopping list.*

**Extensions:**

*None*

### 3.1.17 F-4: Browse shopping list details for the suggested store

**Initiator:**

*User*

**Goal:**

Display the individual price for each item in the shopping list at the suggested store and also provide the details of the shop (e.g. address).

**Main Scenario:**

1. *The user selects one of the suggested stores.*
2. *The name of the store and it's address are displayed.*
3. *User's shopping list is displayed with the individual prices of each item at that store.*

**Extensions:**

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

*None*

*3.1.18 F-5: Show suggested stores on the map*

**Initiator:**

*User*

**Goal:**

Display the locations of the suggested stores on the map.

**Main Scenario:**

- 1. The user clicks the “show stores on the map” button.*
- 2. The system finds all the shops within 5km of the user and checks which stores contain all the products from the shopping list.*
- 3. The shops which have all or most of the products are displayed on a map.*

**Extensions:**

*None*

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

## 4. Requirements Definition

### 4.1 Requirement Group Definitions

Identification	Requirement Group	Rem.
MA	Mobile application	
SER	Server	
DB	Database	
NFR	Non-functional requirements	

### 4.2 Requirement Sources

Source	Description	Rem.
CTM	Customer.	
SYS	Required as a consequence of system design.	
DEV	Developer.	
ADM	Database administrator	

### 4.3 Requirements definitions

Identity	Status	Priority	Description	Source
			Mobile application	
MA-1.	D	III	The user can create a shopping list.	CTM
MA-1.1.	I	I	The user can add items to the shopping list.	CTM
MA-1.1.1.	I	I	The user can add new products via touchscreen.	CTM
MA-1.1.2.	I	II	The user can add new products via barcode.	CTM
MA-1.1.3.	D	III	The user can move items between shopping lists.	DEV
MA-1.1.4.	D	III	The user can set items to be added to a shopping list periodically (eg. a weekly magazine)	DEV
MA-1.1.5.	I	II	When picking an item, the user will have access to an 'auto-complete' feature.	DEV
MA-1.2.	I	I	The user can set the quantity of a particular item in the shopping list	CTM
MA-1.3.	I	I	The user can remove items from the shopping list.	CTM
MA-1.4.	D	III	The user can mark a shopping list as "bought".	DEV
MA-2.	D	III	The user can delete a shopping list.	CTM
MA-3.	C	I	The user can get the locations of all stores in his set radius that suits his shopping list.	CTM
MA-3.1.	I	I	The user can get these locations ranked by lowest price.	CTM
MA-3.2.	I	I	The user can get these locations ranked by proximity.	CTM
MA-3.3.	D	II	The user can get results for as many locations as he like.	CTM

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

MA-3.4.	I	I	The user gets information about the total price of the shopping list, and the address of the store.	CTM
MA-3.4.1	I	III	The user can ask the Google Maps/Navigation app to show the direction through clicking on the “Get direction” button.	CTM
MA-3.4.2	A	I	The user can view all the suggested store locations on the map.	CTM
MA-3.4.3	I	I	The user can view the store's prices for each of the products in the shopping list.	CTM
MA-3.4.4	A	II	The user can set the location radius in which he wants to get the store suggestion results	CTM
MA-4.	D	III	The user can have multiple shopping lists	CTM
MA-5.	D	III	The user can register an account.	DEV
MA-5.1.	D	III	The user can log in from a web site and retrieve their data (shopping lists, preferences, etc.)	DEV
MA-5.2.	D	III	The user can connect the application to the facebook.	DEV
MA-5.3.	D	III	The user can share shopping lists with their friends.	DEV
MA-5.3.1.	D	III	The user's friends can add items to the customer's shopping list (if given permission).	DEV
MA-6.	I	I	The user can add or edit information about products.	CTM
MA-6.1.	I	I	The user can add this information via touchscreen.	CTM
MA-6.2.	I	III	The user can add this information via barcode.	CTM
MA-7.	D	I	The user can add information about stores.	CTM
MA-7.1.	D	II	The user can mark their current location as a store.	CTM
MA-7.2.	D	II	The user can mark the location of a store via a map.	CTM
MA-7.3.	A	I	The user can add a new product to the store's product list.	CTM
MA-7.4.	D	I	The user can view all the products available at the store.	CTM
MA-8.	A	I	The user can view all the store locations on the map.	CTM
MA-9.	D	II	The user can edit information about stores.	CTM
Server				
SER-1.	I	II	The server should sanitize all data inputs from users, to keep information easily readable and to prevent security exploits.	ADM
SER-2.	I	I	The server should calculate optimal locations for shopping based on data available in the database and user input.	CTM
SER-3.	C	II	The server should contain a web-scraper that's able to get the product prices from the store and load the data to database	CTM
Database				
DB-1.	I	II	The database should have archives of all the previous prices with a time dimension indicating when these prices were active	ADM
Non-functional requirements – mobile application				
NFR-1.	I	I	The product names and IDs should be stored locally to reduce server load and bandwidth usage.	ADM
NFR-1.1.	I	I	This data should be synchronized on startup.	CTM
NFR-2.	I	II	Information on recently used products should be stored to facilitate an auto-complete feature.	CTM
NFR-3.	I	I	The mobile application will be made for the Android OS.	CTM

Requirement status:

- I = initial* (this requirement has been identified at the beginning of the project),
- D = dropped* (this requirement has been deleted from the requirement definitions),
- H = on hold* (decision to be implemented or dropped will be made later),
- A = additional* (this requirement was introduced during the project course).
- C = changed* (this requirement was modified as a result of customer feedback)

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

#### 4.3.1 Change Log

Identity	Action	Date	Comments
MA-1.1.3	D	2011-10-21	We need not implement this function.
MA-1.4	H	2011-10-21	Depends on the MA-5.x use cases.
MA-1.	M	2011-10-25	Changed the priority level to match the use case model.
MA-2.	M	2011-10-25	Changed the priority level to match the use case model.
MA-3.4.2	A	2011-10-25	Added a requirement to match the use case model.
MA-7.3.	A	2011-10-25	Added a requirement to match the use case model.
MA-7.4.	A	2011-10-25	Added a requirement to match the use case model.
MA-8.	A	2011-10-25	Added a requirement to match the use case model.
MA-9.	A	2011-10-25	Added a requirement to match the use case model.
MA-3.4.4	A	2011-10-25	Added on customer's request.
SER-3.	C	2011-10-25	Changed the priority level to match the use case model.
MA-1.	C	2011-12-06	Changed priority to III
MA-1.1.4.	C	2011-12-06	Changed priority to III
MA-2.	C	2011-12-06	Changed priority to III
MA-3.	C	2011-12-06	Rephrased
MA-3.3.	D	2011-12-06	Overlapping with MA-3.
MA-4.	C	2011-12-06	Changed priority to III
MA-7.	D	2011-12-06	Dropped because in a discussion with the customer it was concluded that this requirement isn't necessary.
MA-7.1.	D	2011-12-06	Related to MA-7.
MA-7.2.	D	2011-12-06	Related to MA-7.
MA-9.	D	2011-12-06	Related to MA-7.
SER-3.	D	2011-12-06	Dropped because the stores do not provide the information about the prices in a format suitable for web scraping technology.
SER-3.	C	2012-01-09	Changed description and status from dropped to changed.
MA-1.	D	2012-01-09	Dropped
MA-1.1.4.	D	2012-01-09	Dropped
MA-1.4.	D	2012-01-09	Dropped
MA-2.	D	2012-01-09	Dropped
MA-4.	D	2012-01-09	Dropped
MA-5.	D	2012-01-09	Dropped
MA-5.1.	D	2012-01-09	Dropped
MA-5.2.	D	2012-01-09	Dropped
MA-5.3.	D	2012-01-09	Dropped
MA-5.3.1.	D	2012-01-09	Dropped
MA-7.4.	D	2012-01-09	Dropped

#### Requirement status:

- D = dropped (this requirement has been deleted from the requirement definitions),
- H = on hold (decision to be implemented or dropped will be made later),
- A = added (this requirement was introduced during the project course),
- R = resurrected (dropped or on hold requirement was reactivated).
- C = changed (this requirement was modified as a result of customer feedback)

SmartCart	Version: 1.4
Requirements Definition	Date: 2012-01-09

## **5. Future Development**

### **5.1 General Overview**

Future development will not introduce any completely new features since the features currently defined cover the planned scope of the project. Instead, existing features will be refined, optimized, made more user friendly and might have certain additional options added. Particular attention will be paid to the integration of certain features via the Google API, especially features pertaining to maps and directions.

In case of an overabundance of time, work will also be directed towards the development of a Windows Phone application which replicates the functionality of the Android mobile application and utilizes the existing web services.