



SmartCart Acceptance Test Plan

Version 1.1

Doc. No.:

SmartCart	Version: 1.1
Acceptance Test Plan	Date: 2011-12-15

Revision History

Date	Version	Description	Author
2011-11-03	0.1	Initial Draft for Prototype 1	Bin Wu
2011-12-07	0.2	Update the test cases according to the implementation	Bin Wu
2011-12-09	1.0	Review of the draft	Igor Czerwinski
2011-12-14	1.1	Review following supervisor's comments	Igor Czerwinski

SmartCart	Version: 1.1
Acceptance Test Plan	Date: 2011-12-15

Table of Contents

1. Introduction	5
1.1 Purpose of this document	5
1.2 Intended Audience	5
1.3 Definitions and acronyms	5
1.3.1 Definitions	5
1.3.2 Acronyms and abbreviations	5
1.4 References	5
2. Test items	5
3. Features to be tested	5
3.1 Level 1 features	6
3.2 Level 2 features	6
4. Features not to be tested	7
5. Approach	7
6. Item pass/fail criteria	7
7. Suspension criteria and resumption requirements	8
8. Environmental needs	8
8.1 Mobile application	8
8.2 Server application	8
9. Test procedure	8
9.1 Test case specifications	8
9.1.1 Mobile Application Test Case – MATC	8
9.1.1.1 <i>The user adds items to the shopping list – MATC-001</i>	8
9.1.1.2 <i>The user can add new products via touchscreen – MATC-002</i>	8
9.1.1.3 <i>The user adds new products via barcode – MATC-003</i>	9
9.1.1.4 <i>The user has access to an ‘auto-complete’ search feature – MATC-004</i>	9
9.1.1.5 <i>The user updates the quantity of a particular item in the shopping list – MATC-005</i>	10
9.1.1.6 <i>The user removes product from the shopping list – MATC-006</i>	10
9.1.1.7 <i>The user gets suggest stores locations ranked by lowest price – MATC-007</i>	10
9.1.1.8 <i>The user gets the suggest stores locations ranked by proximity – MATC-008</i>	11
9.1.1.9 <i>The user gets information about the total price of the shopping list, and the address of the store – MATC-009</i>	11
9.1.1.10 <i>The user views all the suggested store locations on the map – MATC-010</i>	12
9.1.1.11 <i>The user views the store's prices for each of the products in the shopping list – MATC-011</i>	12
9.1.1.12 <i>The user set the location radius for suggestion results – MATC-012</i>	12
9.1.1.13 <i>The user adds and edits the price about products – MATC-013</i>	13
9.1.1.14 <i>The user views all the nearby stores – MATC-014</i>	13
9.1.1.15 <i>The user views all the products available at a store – MATC-015</i>	13
9.2 Test plan	14
10. Responsibilities	14
10.1 User representative	14

SmartCart	Version: 1.1
Acceptance Test Plan	Date: 2011-12-15

10.2	Systems administrator	14
10.3	Tester	14
10.4	Project manager	14
11.	Risks and contingencies	14

SmartCart	Version: 1.1
Acceptance Test Plan	Date: 2011-12-15

1. Introduction

1.1 Purpose of this document

The objective of Acceptance Testing is to confirm that the developed application meets its requirements and to ensure that the system works correctly and is usable before it is formally “delivered” to the end user. This document is the Acceptance Test Plan for the SmartCart application. It establishes the acceptance test framework used by the acceptance test team to plan, execute, and document acceptance testing. It describes the scope of the work performed and the approach taken to execute the tests created to validate that the system performs as required. The details of the ATP are developed according to the requirements specifications, and show traceability back to those specifications.

1.2 Intended Audience

This document is intended for the:

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

1.3 Definitions and acronyms

1.3.1 Definitions

Keyword	Definitions
SmartCart	Project Name and Team Name
DSD	Course Name
Android	Mobile Operating System/Platform

1.3.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
WIFI	Mechanism for connecting electronic devices wirelessly
3G	3 rd generation standard for mobile telecommunication services
GPS	Global Positioning System
PMS	Project management system

1.4 References

- [SmartCart Project Proposal](#)
- [SmartCart Requirement Definition](#)
- [SmartCart Design Description](#)
- [SmartCart Project Plan](#)
- [SmartCart related Documents](#)

2. Test items

In the user acceptance testing phase we have identified only one test item: the SmartCart mobile application itself. Within the mobile application all features will be tested.

3. Features to be tested

All the requirements outlined in the requirements definition document will be tested in order of their priority. The tables in sections 4.1, 4.2 and 4.3 list the requirement and their priority.

SmartCart	Version: 1.1
Acceptance Test Plan	Date: 2011-12-15

3.1 Level 1 features

Requirement Identity	Description
Mobile application	
MA-1.1.	The user can add items to the shopping list.
MA-1.1.1.	The user can add new products via touchscreen.
MA-1.2.	The user can set the quantity of a particular item
MA-1.3.	The user can remove items from the shopping list.
MA-3.	The user can get the locations of all stores in his set radius that suits his shopping list.
MA-3.1.	The user can get these locations ranked by lowest price.
MA-3.2.	The user can get these locations ranked by proximity.
MA-3.4.	The user gets information about the total price of the shopping list, and the address of the store.
MA-3.4.2	The user can view all the suggested store locations on the map.
MA-3.4.3	The user can view the store's prices for each of the products in the shopping list.
MA-6.	The user can add or edit information about products.
MA-6.1.	The user can add this information via touchscreen.
MA-7.3.	The user can add a new product to the store's product list.
MA-7.4.	The user can view all the products available at the store.
MA-8.	The user can view all the store locations on the map.
Server	
SER-2.	The server should calculate optimal locations for shopping based on data available in the database and user input.
Non-functional requirements – mobile application	
NFR-1.	The product names and IDs should be stored locally to reduce server load and bandwidth usage.
NFR-1.1.	This data should be synchronized on startup.
NFR-3.	The mobile application will be made for the Android OS.

3.2 Level 2 features

Requirement Identity	Description
Mobile application	
MA-1.1.2.	The user can add new products via barcode.
MA-1.1.5.	When picking an item, the user will have access to an 'auto-complete' feature.
MA-3.4.4	The user can set the location radius in which he wants to get the store suggestion results
Server	
SER-1.	The server should sanitize all data inputs from users, to keep information easily readable and to prevent security exploits.
Database	
DB-1.	The database should have archives of all the previous prices with a time dimension indicating when these prices were active
Non-functional requirements – mobile application	
NFR-2.	Information on recently used products should be stored to facilitate an auto-complete feature.

SmartCart	Version: 1.1
Acceptance Test Plan	Date: 2011-12-15

4. Features not to be tested

The user acceptance testing phase will not test the server application directly. Instead, the server application will be tested indirectly – as a result of users' interaction with the mobile application.

Also, the reliability of the barcode scanner will not be extensively tested because that functionality is provided by a third party library.

5. Approach

As stated in the project plan document, we have decided to manage the development effort using the Rational Unified Process. This methodology stresses the use of iterations, each of which finishes with a testing phase. During this phase the application was presented to the customer to let them evaluate the progress and comment on the prototype. Therefore, the application has been developed with a close interaction with the customer ensuring that it fulfills their expectations. However, to provide confidence that the application meets the formal requirements stated in the Requirements Document we have created a traceability matrix which maps all the requirements to corresponding test scenarios which can be seen in Figure 1.

Test Id Req Id	MATC-001	MATC-002	MATC-003	MATC-004	MATC-005	MATC-006	MATC-007	MATC-008	MATC-009	MATC-010	MATC-011	MATC-012	MATC-013	MATC-014	MATC-015	Test cases
MA-1.1.	x		x		x	x	x	x	x		x		x			9
MA-1.1.		x														1
MA-1.1.2.			x													1
MA-1.1.5.				x												1
MA-1.2.					x	x										2
MA-1.3.						x										1
MA-3.							x	x								2
MA-3.1.							x		x	x						3
MA-3.2.								x	x							2
MA-3.4.									x							1
MA-3.4.2.										x						1
MA-3.4.3.											x					1
MA-3.4.4.												x				1
MA-6.													x			1
MA-6.1.													x			1
MA-7.3.													x			1
MA-7.4.														x	x	2
MA-8.															x	1
NFR-1.	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	15
NFR-1.1.	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	15
NFR-2.				x												1
NFR-3.	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	15
SER-1.	x	x	x													3
SER-2.							x	x					x			3
DB-1.													x			1

Figure 1: Traceability matrix

6. Item pass/fail criteria

If each tested item realizes the corresponding requirement, then the test item is considered a pass. Otherwise, if the test item does not fulfill the traced requirement, it will be considered a fail. Additionally, the user representative has the final word in asserting the correctness of the test item.

SmartCart	Version: 1.1
Acceptance Test Plan	Date: 2011-12-15

7. Suspension criteria and resumption requirements

The acceptance testing procedure can be suspended in the following situations:

1. Unavailability of the server application
2. When a defect has been introduced that cannot allow any further testing.
3. Critical path deadline is missed so that the customer will not accept delivery even if all testing is completed.
4. A specific holiday shuts down both development and testing.

The process can be resumed only under the following circumstances:

1. When the server application becomes available again.
2. When a fix is successfully implemented.
3. The contract is renegotiated with the customer to extend delivery.
4. The holiday period ends.

8. Environmental needs

8.1 Mobile application

The SmartCart mobile application requires a mobile device with a touchscreen, GPS, auto-focus camera and Internet connectivity. The device must be running the Google Android operating system in version 2.2 or later.

8.2 Server application

The SmartCart mobile application requires a Windows Server machine with IIS 6 or later and the Microsoft SQLServer 2008. Additionally, the application requires the .NET runtime environment to be installed with version 4.0 or newer.

9. Test procedure

9.1 Test case specifications

Each case is specified with one-line summary and detailed description covering test type, precondition, input and output.

9.1.1 Mobile Application Test Case – MATC

9.1.1.1 The user adds items to the shopping list – MATC-001

Description:

The user browses the “Products” and adds the items which he/she wants into the “Shopping list”.

Preconditions:

1. The items which the user wants to add to the Shopping list exist in the Products.

Input definition:

1. The user starts the SmartCart application, opens the applications menu and clicks “Products” button.
2. The user browses the products list and selects products from “Products”.
3. The user click on the “ = ” icon associated with the product.

Output definition:

1. The message “added to the shopping list. Product: name, quantity: 1” appears if the item didn’t exist in the shopping list before.
2. The message “Increased quantity. Product: name, quantity: n” appears if the item existed in the shopping list before.
3. When the user checks the shopping list, he/she can see the new added products.

Remarks:

Level 1

9.1.1.2 The user can add new products via touchscreen – MATC-002

SmartCart	Version: 1.1
Acceptance Test Plan	Date: 2011-12-15

Description:

The user browses the “Products” and adds new products to the “Products” via touchscreen.

Preconditions:

1. The mobile supports touch screen mode.

Input definition:

1. The user starts the SmartCart application, opens the applications menu and clicks “Products” button.
2. The user clicks on the “add new Product” icon on the lower left of display, and a window pops up.
3. The user fills in all the information about the new add product like Product, Category, Price, and Store.
4. The user clicks on the “Save product” button on the bottom of the display.

Output definition:

1. The message “added product successful” appears if the item didn’t exist in the store before.
2. When the user browses the Products, he/she can see the new added products.
3. The message “the product existed” appears if the item existed in the store before.

Remarks:

Level 1

9.1.1.3 The user adds new products via barcode – MATC-003

Description:

The user browses the “Products” and adds new products to the “Products” via barcode scanning.

Preconditions:

1. The mobile has camera to support barcode scanning.

Input definition:

1. The user starts the SmartCart application, opens the applications menu and clicks “Scan product” button.
2. The user clicks on the “Scan” button to scan the barcode of the product.
3. The user click on the “add new product” button.

Output definition:

1. The message “added product successful” appears if the item didn’t exist in the store before.
2. When the user browses the Products, he/she can see the new added products.
3. The message “the product is updated” appears if the item existed in the store before.

Remarks:

Level 2

9.1.1.4 The user has access to an ‘auto-complete’ search feature – MATC-004

Description:

The user searches the particular product with ‘auto-complete’ search feature.

Preconditions:

None

Input definition:

1. The user opens the SmartCart application, opens the applications menu and clicks “Products” button.
2. The user clicks the “search” button.
3. The user fills in part of product name.

Output definition:

1. The ‘auto-complete’ search feature will help to get the results in a generated list which contains the part of the product name.

SmartCart	Version: 1.1
Acceptance Test Plan	Date: 2011-12-15

Remarks:

Level 2

9.1.1.5 The user updates the quantity of a particular item in the shopping list – MATC-005

Description:

The user updates the quantity of the product in the shopping list.

Preconditions:

1. The product already exists in the user’s shopping list.

Input definition:

1. The user opens the SmartCart application, opens the applications menu and clicks “Shopping list” button.
2. The user browses the “Shopping list” and chooses the product which the user wants to update the quantity.
3. The user clicks a plus or minus button associated with the product to increase or decrease the quantity.

Output definition:

1. The quantity of the product is updated.

Remarks:

Level 1

9.1.1.6 *The user removes product from the shopping list – MATC-006*

Description:

The user removes the product which he doesn’t want to buy from the shopping list.

Preconditions:

1. The product already exists in the user’s shopping list.

Input definition:

1. The user opens the SmartCart application, opens the applications menu and clicks “Shopping list” button.
2. The user clicks the “Del” button left to the name of the product.

Output definition:

1. The product disappears in the shopping list..

Remarks:

Level 1

9.1.1.7 *The user gets suggest stores locations ranked by lowest price – MATC-007*

Description:

This feature help user to get the suggested stores locations, which have all or most of the items on the shopping and the total shopping price is ranked by lowest price by lowest.

Preconditions:

1. There is at least one product in the Shopping list.
2. The Google map is associated with the mobile application.

Input definition:

1. The user opens the SmartCart application, opens the applications menu and clicks “Settings” button.
2. The user clicks the “Method” button to choose “Price” option.

SmartCart	Version: 1.1
Acceptance Test Plan	Date: 2011-12-15

3. The user clicks “Radius” button to set the Radius in which you want to search.
4. The user back to the applications menu and click “Shopping list” button.
5. The user clicks “Suggest stores”.

Output definition:

1. A window named “Suggested stores in radius xxx m” is shown up, and the suggested stores which have the most articles are ranked by lowest price. And there are delimiters to divide the stores by available products.
2. Each suggested store includes the information about distance from your location, totally price, available products and store name.

Remarks:

Level 1

9.1.1.8 The user gets the suggest stores locations ranked by proximity – MATC-008

Description:

The user wants to get the suggest stores locations, which are nearest to the user and have all or most of the items on the user's shopping list.

Preconditions:

1. There is at least one product in the Shopping list.

Input definition:

1. The user opens the SmartCart application, opens the applications menu and clicks “Settings” button.
2. The user clicks the “Method” button to choose “Distance” option.
3. The user clicks “Radius” button to set the Radius in which you want to search.
4. The user back to the applications menu and click “Shopping list” button.
5. The user clicks “Suggest stores”.

Output definition:

1. A window named “Suggested stores in radius xxx m” is shown up, and the suggested stores which have the most articles are ranked by proximity. And there are delimiters to divide the stores by available products.
2. Each suggested store includes the information about distance from your location, totally price, available products and store name.

Remarks:

Level 1

9.1.1.9 The user gets information about the total price of the shopping list, and the address of the store – MATC-009

Description:

The user gets the *information about the total price of the shopping list, and the address of the suggested store.*

Preconditions:

1. *The suggested stores are shown in the Google map.*

Input definition:

1. The user clicks on the store sign.

Output definition:

1. The information about the store name, Total prices, Total products and address is shown.

Remarks:

Level 1

SmartCart	Version: 1.1
Acceptance Test Plan	Date: 2011-12-15

9.1.1.10 *The user views all the suggested store locations on the map – MATC-010*

Description:

The user can view *all the suggested store locations* by showing them on the Google map.

Preconditions:

1. The user has got the suggested stores by distance or price.

Input definition:

1. The user clicks “Show on Map” button.

Output definition:

1. All the suggested stores are shown in the Google map.

Remarks:

Level 1

9.1.1.11 *The user views the store's prices for each of the products in the shopping list – MATC-011*

Description:

The user can check the store's prices for each product in the shopping list in 3 suggested stores.

Preconditions:

1. There is at least one product in the Shopping list.

Input definition:

1. The user opens the SmartCart application, opens the applications menu and clicks “Shopping list” button.
2. The user opens the SmartCart application, opens the applications menu and clicks “Settings” button.
3. The user clicks the “Method” button to choose “Price” or “Distance” options.
4. The user back to the application menu and clicks “Shopping list” button.
5. The user clicks the “Suggest stores” button.
6. The user clicks the store name which he/she want to check the prices for each of the products in the shopping list.

Output definition:

1. The store's prices for each of the products in the shopping list are shown.

Remarks:

Level 1

9.1.1.12 *The user set the location radius for suggestion results – MATC-012*

Description:

The default value of the location radius is 5km, but the user can modify the value to y to get the store suggestion results as he like.

Preconditions:

None

Input definition:

1. The user opens the SmartCart application, opens the applications menu and clicks “Settings” button.
2. The user clicks the “Radius (meters)” button to set the value x.
3. The user clicks the “ok” button.
4. The user back to the application menu and clicks “Shopping list” button.
5. The user clicks the “Suggest stores” button.

Output definition:

SmartCart	Version: 1.1
Acceptance Test Plan	Date: 2011-12-15

1. A window named “Suggested stores in radius x m” pops up. And the distance from user current location to each of suggestion results is in radius x m.

Remarks:

Level 2

9.1.1.13 The user adds and edits the price about products – MATC-013

Description:

The user wants to edit price about the particular product via touchscreen.

Preconditions:

1. The mobile device is connected to the Internet

Input definition:

1. The user opens the SmartCart application, opens the applications menu and clicks “Products” button via touchscreen.
2. The user adds a new product to the product list (as in MATC-001)
3. The user clicks on the newly added product.
4. The user clicks “edit” button.
5. The user updates the some information.
6. The user clicks the “Save” button via touchscreen.

Output definition:

1. The user views the product to check that the information about the product has been updated successful.
2. When the 3G wireless network is disconnected, the application can not send the updated information to the server, the updated process failed, and an error message is displayed.
3. When the user clicks “cancel” before save, the update data will be lost.

Remarks:

Level 1

9.1.1.14 The user views all the nearby stores – MATC-014

Description:

The user wants to see all stores which are nearby.

Preconditions:

1. The mobile device is connected to the Internet.

Input definition:

1. The user opens the SmartCart application and goes to the “Shops” option.

Output definition:

1. A maps is displayed with all the shops nearby.

Remarks:

Level 1

9.1.1.15 The user views all the products available at a store – MATC-015

Description:

The user wants to see a list of products available at a store.

Preconditions:

1. The mobile device is connected to the Internet

SmartCart	Version: 1.1
Acceptance Test Plan	Date: 2011-12-15

Input definition:

1. The user opens the SmartCart application and goes to the “Shops” option.
2. The user taps the store they are interested in and tap “Show all products”

Output definition:

1. A list of all the products available at the chosen store is displayed.

Remarks:

Level 1

9.2 Test plan

All the test cases have been divided into three priority levels, each priority level match with the requirement level. The level 1 test cases are planned to be tested in the alpha prototype which is critical for the success of the project. And the level 2 test cases will be tested in the beta prototype. We also define the level 3 test cases which will be tested in case there is extra time left before the end of the project. So for the test plan, what we should do in which phase is very clear.

10. Responsibilities

10.1 User representative

- ⤴ responsible for conducting the User Acceptance Test

10.2 Systems administrator

- ⤴ responsible for installing the server application in a live environment and ensuring that there is no contention for the system resources and there are no scheduled IT maintenance tasks in that time

10.3 Tester

- ⤴ responsible for ensuring that the acceptance test procedure follows the methodology defined in the Acceptance Test Plan
- ⤴ responsible for liaising with the User Representative to determine which test cases will be rerun during the acceptance test and determining the need to design and implement test scenarios to execute specific aspects when requested by the User Representative
- ⤴ responsible for assisting the User Representative with the execution of the test scenarios as defined in the acceptance test plan and in interpreting the results of each test scenario
- ⤴ responsible for creating a summary of the acceptance tests detailing its outcomes

10.4 Project manager

- ⤴ responsible for liaising with the development team to determine the progress of the development and the likely dates for delivery of the application for the acceptance testing
- ⤴ responsible for informing the systems administrator of the date, time and duration of the acceptance tests

11. Risks and contingencies

1. Insufficient test time
 1. Risk: If the amount of time available is too short, the acceptance test team may not have enough time to complete acceptance testing or perform regression testing.
 2. Mitigation: Develop a critical path of tests, prioritized by importance.
2. Incomplete requirements
 1. Risk: May result in insufficient testing of the system.
 2. Mitigation: Use the traceability matrices to track the testing/requirements relationship.
3. A test environment that is not the same as the production environment -
 1. Risk: May prevent the detection of some defects and issues.
 2. Mitigation: Note the differences and work to have them as close as possible.