Universität Bern
Introduction to Software Engineering

ShoppyApp

Software Requirements Specification Document

v2.0

11.12.2013

Customer

Mircea Lungu

Technical Tutor

Andrei Chis

Authors

Sammer Puran / Lukas Galliker / Marc Schneiter / Sébastien Broggi

Table of Contents

Version History

Version	Date	Revision Description
0.1	02.10.2013	First Version of the Document
0.2	08.10.2013	Document reviewed, changed Use Cases
0.3	25.10.2013	Added our latest progress, changed the Use Cases accordingly, updated the text
1.0	13.11.2013	Version 1 release! Finally implemented Sharing
2.0	11.12.2013	Version 2 release!

Introduction

Purpose

Shoppy is an Android App which makes it simple to organise your shopping lists on your phone and tablet, rather than on paper. It is a simple helper for managing your items in the shopping list and share it with your friends.

The goal of this application is to be a time-saver and cooperative tool to organise shopping with your friends and family.

- Android
- iOS
- Windows Phone
- Java ME
- BlackBerry
- Other

29%

54%

1%

7%

4% 5%
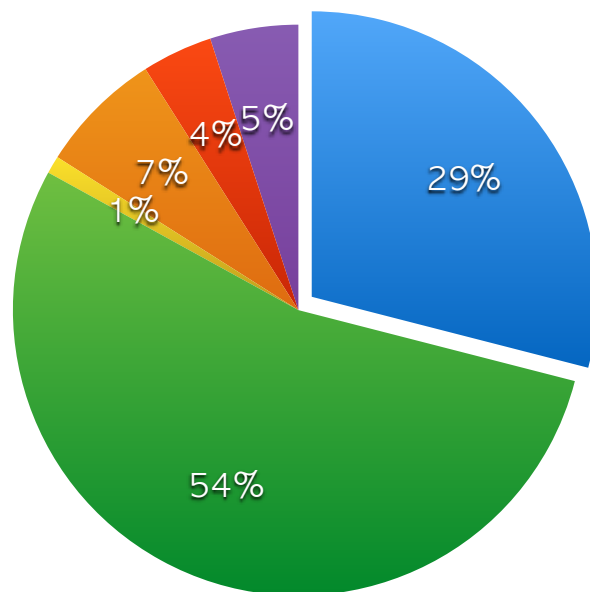
Mobile/Tablet Operating System Market Share as of September, 2013
Source: http://www.netmarketshare.com/

These numbers are difficult to evaluate and differ significantly depending on what source is used, but they can give an impression of the mobile market.

Stakeholders

The application is of use for anybody who likes to create a Shopping list before going to shop because he or she doesn't want to forget something. In that sense most people are potential customers of Shoppy. As we plan our initial release of the App only for the Android platform, just owners of an Android OS driven phones can use our application.

To make the potential amount of users a bit more visible, we include some statistics for mobile platform market shares.

System Overview

The name of the application is 'Shoppy', which is a short and memorable name and has its likeable and cute sub tone.

The user can create multiple shopping lists for different purposes, for example organising the items to buy for a party and can share them with his friends. All users of a shopping list can add, remove and edit items from shopping lists, mark them as bought and organise them in categories. All of these actions will be synchronised with our server and can be shared with other people. A shared list can be created by selecting people from your contacts in your list sharing option.

If the user starts typing in a new item for a list, the application suggests items that were already added to a list in the past
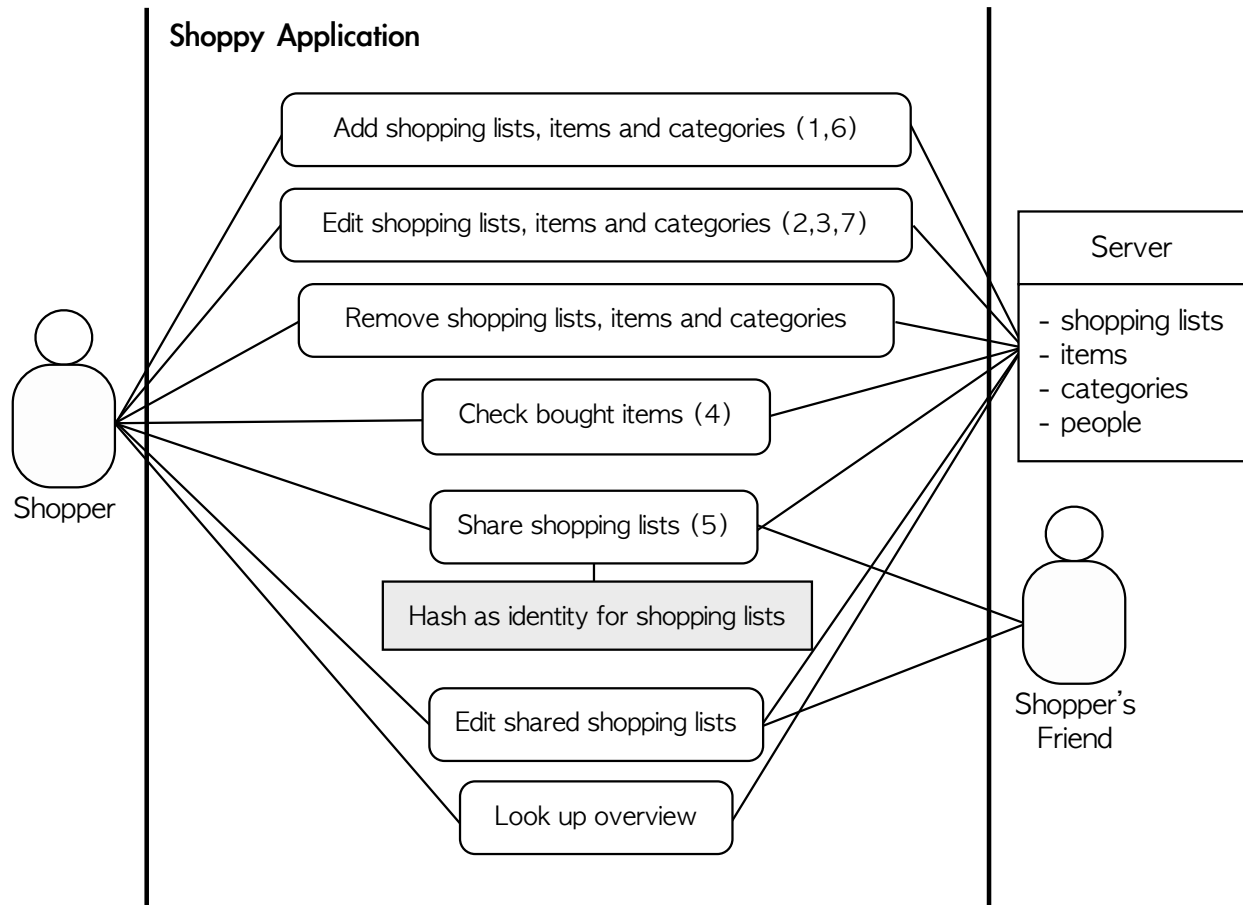
When you open the app it returns to the last opened list and presents you the items you have to buy.

Our Job

We made it our job to create a shopping application which has none of the downsides other applications in app stores have. We analysed the most required features in the comment sections of other application and tried to make a great mix out of them. We hope we'll succeed.

Overall Description

Use Case Diagram

Shoppy Application

Add shopping lists, items and categories (1,6)

Edit shopping lists, items and categories (2,3,7)

Remove shopping lists, items and categories

Check bought items (4)

Share shopping lists (5)

Hash as identity for shopping lists

Edit shared shopping lists

Look up overview

Shopper

Server

- shopping lists
- items
- categories
- people

Shopper's Friend

Use Case 1: Create Shopping List

Create Shopping List:

a. Actors
   i. Customer

b. Description
   i. As a customer I want to add a shopping list, name it and want to see it.

c. Trigger
   i. Press the "+" on the top right corner of the home-screen.

d. Pre-conditions
   i. The customer has to have bought the app.

e. Post-conditions
   i. A new shopping list has been added to the intern database and is displayed in the shopping lists tab.

f. Main Scenario
   i. Customer opens the app.
   ii. Customer presses the "+" button.
   iii. Customer names the list.
   iv. Customer optionally specifies who to share the list with. (no-person option)
   v. Customer saves the list.
      1. Friends receive invitations to the list.
      2. User receives notification of friends joining his list.
   vi. Customer starts editing new list. ("list" view)

g. Alternative Scenarios
   i. Add by importing a list from a friend (sharing):
      1. Customer opens the app.
      2. Customer receives invitation in the notification tab.
      3. Customer accepts or declines the invitation.
      4. List is retrieved from the server.
      5. Customer starts editing newly downloaded list.

h. Special Requirements - none

i. Notes
   i. The Customer cannot name a list the same as an existing list. It gives an error and the user is prompted to give it another name.

Use Case 2: Edit Shopping List

Edit Shopping List:

a. Actors
   i. Customer

b. Description
   i. As a customer I want to edit an already existing list, i.e. change it's name and properties.

c. Trigger
   i. Hold down your finger for 1 second on the list you want to edit, select "edit" option in the pop-up menu.

d. Pre-conditions
   i. There already exists a shopping list to edit.

e. Post-conditions
   i. The shopping list is successfully adapted to the customers needs (name, category, people to share).

f. Main Scenario
   i. Customer opens the app.
   ii. Customer selects a list from the list screen.
   iii. Customer holds down the finger on the list he wants to edit and selects the "edit" option in the pop-up menu.
   iv. Customer enters a new screen with the option he has to modify the selected list, i.e. rename the list, add categories, add/remove persons with whom he wishes (or not) to have the list shared with.
      1. Rename the list
         a. Customer types another or over the existing name to change the name
         b. The Customer cannot name a list the same as an existing list. It gives an error and the user is prompted to give it another name.
      2. Add Categories
         a. Customer adds one or more categories from an existing categories-library to the list.
      3. Share the list
         a. UC 5, Page 10
   v. Customer saves the changes by selecting the "save" option, which exits the edit-list view.

g. Alternative Scenarios - none

h. Special Requirements - none

i. Notes
   i. The Customer cannot name a list the same as an existing list. It gives an error and the user is prompted to give it another name.

Use Case 3: Add/Remove Items to/from the Shopping List

Add/Remove Items to/from the Shopping List:

- a. Actors
    - i. Customer
- b. Description
    - i. As a customer I want to add items to an already existing Shopping List.
- c. Trigger
    - i. Press the "+" on the top right corner in the list-view. (or hold your finger down on the item and select "remove" on the pop-up menu)
- d. Pre-conditions
    - i. There already exists a shopping list to add items to.
- e. Post-conditions
    - i. The item/-s is/are added/removed to the Shopping List.
- f. Main Scenario
    - i. Adding items:
        1. Customer opens the app.
        2. Customer adds a list (UC 1, Page 6) and is now in the list-view.
        3. Customer presses the "+" on the top right corner.
        4. Customer specifies the item (name, category, quantity).
        5. Customer saves the changes.
    - ii. Removing items:
        1. Customer opens the app
        2. Customer selects a list.
        3. Customer selects an item.
        4. Customer holds down the finger on the item and selects "remove" on the pop-up menu
- g. Alternative Scenarios
    - i. Add by manually add items to an already existing list:
        1. Customer opens the app.
        2. Customer enters an already existing list by clicking on it.
        3. Customer presses the "+" on the top right corner.
        4. Customer specifies the item (name, category, quantity).
        5. Customer saves the changes.
- h. Special Requirements - none
- i. Notes
    - i. The Customer cannot name an item the same as an existing item. If he does (and the same named items are in the same category), the quantities of both the items are added together.

Use Case 4: Use Shopping List

Use Shopping List:

a. Actors
  i. Customer

b. Description
  i. As a Customer I want an option to check/uncheck items in a shopping list to mark them as bought/not bought.

c. Trigger
  i. Customer presses the box on the left side of an item.

d. Pre-conditions
  i. There already exists a shopping list and an item in that shopping list the Customer can use.

e. Post-conditions
  i. All the checked items are marked as bought, the order of the items in that list is changed to put the items that are not bought yet on the top.

f. Main Scenario
  i. Customer starts the app.
  ii. Customer chooses a shopping list he wants to buy items from.
  iii. Customer presses the box on the left side of an item she/he just bought.

g. Alternative Scenarios - none

h. Special Requirements - none

i. Notes
  i. Main feature - Definitely solid implemented

Use Case 5: Share A Shopping List

Share Shopping List:

a. Actors
   i. Customer
   ii. Friend of the Customer (who also owns the app)

b. Description
   i. As a Customer I sometimes want to share one of my shopping lists with my friends, so they can see what I have to buy and buy it for me. The shared list is always up-to-date to simplify things.

c. Trigger
   i. Customer holds down his finger on the list he wants to share, selects 'share' in the option menu.

d. Pre-conditions
   i. There has to be an existing list the Customer can share.
   ii. The Friend the Customer wants to share the list with has to own the app too.

e. Post-conditions
   i. The Friend has to press refresh in his app, the new list appears on his list overview.

f. Main Scenario
   i. Customer starts the app.
   ii. Customer selects the list he wants to share.
   iii. Customer holds down his finger on the desired list until options menu pops up, then selects 'share'.
   iv. Customer adds friend he wants to share the list with (by email)
   v. Customer clicks share.
   vi. They enjoy the sharing option.

g. Alternative Scenarios - none
h. Special Requirements - none