Shopping List Management on Mobile Phones

This is an individual programming exercise.

The aim of the exercise is to develop a program for managing shopping lists on mobile phones.

The program will be developed in a number of stages, starting with a basic configuration which is enhanced by the introduction of additional features in a series of stages. In developing your program it is essential that you first ensure that the basic configuration is fully functional before you move on to the introduction of the additional features.

The exercise will also be developed in two phases with the first phase providing some feedback for the early stages of the basic functionality.

Phase 1 - Due Tuesday 18 February at 15:00.

For the first phase you need to complete the two stages below, and bring your implementation to the lab on Tuesday 18 February, where each of you will have to demonstrate their implementation to a group of your colleagues that will try to identify user interaction issues with it. **Each of you is responsible to note down a list of identified issues, signed by the other members of your group, and will then submit on myPlace the signed list together with a plan of how you intend to address them by Tuesday 25 February at 17:00**.

Failure to provide an implementation that can be meaningfully demonstrated will result in capping your mark for the two stages to 15 marks and to block you from being able to gain any marks for extra functionality.

Basic configuration

The basic configuration manages a single shopping list. The program allows users to create a new shopping list by typing in a set of shopping list items. The shopping list may contain any number of items, while for each item it contains a name and a quantity to be purchased. The program persistently stores the created shopping list. The program also allows users to load the created shopping list when out shopping, and to mark each product they buy as purchased in the list.

Tracking weekly shopping

This configuration manages a number of shopping lists, one for each shopping trip. The program allows users to create a new shopping list for a shopping trip, either from scratch by typing in the set of items, as in the basic configuration, or loading and configuring any of the previously created shopping lists. In the latter case the user can configure a shopping list by adding new items to it, removing some of the existing items, or editing the quantity of a particular item to be purchased. Every created shopping list is persistently stored and associated with a name and the date and time it was created.

Phase 2 - Due Monday 24 March at 17:00

Tracking shopping cost

This configuration adds to the above configuration the ability to record prices for purchased products and to track the current total of all purchased products. For this purpose the program should allow users to load a previously created shopping list and while shopping enter the price and quantity of each purchased item. As items are purchased the application calculates and constantly displays the total price of all items purchased so

far. The program should allow users to purchase a different quantity of a particular item than that listed on the list. The total price should correctly reflect that. The application also persistently stores all items purchased, their prices, and the total price. Note that this is separate from storing the original shopping lists, think of it as storing a receipt from a particular shopping trip. Note also that as the creation of a shopping list is separate separate from its use, it is unreasonable to expect users to provide prices at the list creation stage.

Price comparison

While shopping, whenever the user enters the price of a purchased item the program checks the earlier saved lists of purchases and if the same item is included it compares the current price to the latest one and indicates whether it is higher or lower. Note that the user must be given some time to notice this information before it is removed from the screen. The user should be able to enable/disable this feature.

Auto-removal of purchased items

This feature supports the user when shopping by automatically removing from the shopping list any item that has been purchased. As a result, the list is gradually reduced in size until it potentially becomes empty. The auto-removal of purchased items should not interfere with the persistent storage of information, i.e. is only limited to the interface of the application. The user should be able to enable/disable this feature.

Product memory and auto-completion

As the user is creating/editing shopping lists the program keeps track of the various products that the user has purchased and offers to auto-complete products when the user is typing a list based on those previously encountered.

Managing user preferences

The program should allow the user to set preferences regarding all the optional features (i.e. those that can be enable/disabled or switched on/off) and behave in accordance to those preferences. You can also introduce additional preferences that allow the user to configure the look of the application (e.g. colour schemes, backgrounds, etc).

Extra functionality

You should only attempt these after you have attempted all the above stages, and provided that they were not excluded after phase 1, otherwise you are not going get any marks for them.

Open Glasgow Integration

As part of the Future City Demonstrator the city of Glasgow has createdOpen Glasgow, as site for sharing and collecting city data. This feature is open in its specification. The idea is to specify and implement extra functionality for your app that either utilizes in a meaningful way data from Open Glasgow or generates data that could be generate a new interesting data set for it. As this feature is open in specification you will need to discuss with the class lecturer your feature specification to ensure an appropriate scope and the exact amount of marks it will carry. **You need to do so by Tuesday 4 March at 17:00**.

Social Shopping

This feature is open in its specification. The idea is to specify and implement extra functionality for your app that will allow it to meaningfully use features of social networking sites like Facebook, Google+ and Twitter. As this feature is open in specification you will need to discuss with the class lecturer your feature specification to ensure an appropriate scope and the exact amount of marks it will carry. **You need to do so by Tuesday 4**

March at 17:00.

Product list sharing

This feature allows the application to exchange product lists with other devices over Bluetooth. The use should be able to share any of the existing shopping lists. Any list received over Bluetooth should be visible in the list of shopping list and be able to be used (either configured or as is) for shopping. The list should be exchanged using an interoperable format, like XML or JSON. You should pair up with someone in order to allow testing of this feature between devices.

Tesco product price import

This feature allows the application to use the Tesco API to retrieve products and their prices. The idea is to use the Tesco products while a user is building a new shopping list, i.e. help auto-complete while the user is entering a product by presenting them with a list of matching products to select from. In addition to this, the feature downloads the price and computes an estimated cost for purchasing the items listed. Note that you should wait for the user to enter a few characters before searching for matching products. The user should be able to enable/disable this feature.;

Marking scheme

In order to get any marks for the exercise you need to demonstrate your code in the lab at the end of the exercise. We will use the labs of the last three week of the class for the demonstrations. A schedule will be published closer to the date.

You also need to submit a zip file with all the code for your application on myplace plus a document (no more that a couple of pages) outline key design and implementation choices.

The exercise is marked based on the completeness, robustness and quality of the supported functionality as judged by both the demonstration and the submitted documentation. Note that quality involves both user interaction quality and appeal, including taking into consideration the feedback after phase 1, as well as code organization and appropriate utilization of platform features (e.g. use of features like SQLite database for data storage, preferences screens, swipe gestures, contextual menus, etc).

The exercise is marked out of 100 with ability to get more than that. Each stage can gain the following marks:

- Basic Configuration - up to 20 marks
- Tracking Weekly Shopping - up to 20 marks
- Tracking Shopping Cost - up to 15 marks
- Price comparison - up to 10 marks
- Auto-removal of purchased items - up to 5 marks
- Product memory and auto-completion - up to 10 marks
- Managing user preferences - up to 5 marks
- Open Glasgow integration - up to 20 marks
- Social shopping - up to 20 marks
- Product list sharing - up to 15 marks
- Tesco product price import - up to 15 marks

An additional 5 marks for a native app demonstrated working on a real Android or Java-enabled device. An additional 15 marks for a native app demonstrated working on a real iPhone, iPad or Window Mobile device, only 10 marks if demonstrated on an emulator.