CSE116A,B Introduction to Computer Science II Fall 2000

Bina Ramamurthy

Project 1: Hand-held Point of Sale System

September 5, 2000

1 Objective

Major objective of the project is to familiarize the students to

- Analysis, conceptual design and implementation phases of software development cycle,
- Designing and developing the solution for a problem based on user requirements,
- Object-oriented design: classes and class diagrams,
- Separation of interface and implementation,
- Applying simple patterns, and
- Reusing the various graphical components specified in the Java API.

2 Problem Statement

You are required to point of sale system similar to the case Study in Chapter 4 of "Applying UML and Patterns" text. Instead of any random point of sale system we will consider a specific system for a fast food business McBurger Queen Inc. This business wants to try out a new concept in drive through orders. Instead of cars driving up to a window and ordering, car customers park in designated parking lots. These lots have a wireless device attached to a pole at each parking spot. This pole is somewhat similar to parking meters. Unlike parking meters these devices are intelligent, electronic and are capable of transmitting the information instantaneously to an order taker (/server in technical terminology). A customer will place the order using this wireless device (which is similar to PalmPilot or RIMM in appearance). We will refer to this device as Wireless Device CSE116 (WD116). It displays a menu of items from which the customer can choose to make an order. An order has many items, a customer may choose to remove an item just ordered, cancel the entire order and or place the order. Assume that the money transaction is automatically carried out and that it is beyond the scope of this project. When the order is ready it will be delivered to the car. McBurger Queen Inc. has hired you write this "mobile application" in Java. They have designed the user interface of the hand-held device according the specifications given by an ergonomist. Task assigned to each of you is to come up with the program that makes this interface work.

3 Design Details

Customer can

1. select burger type
2. select drink type and size
3. choose side order
4. add item to order
5. cancel last item
6. cancel entire order
7. place the order
8. review amount to paid

If many items are needed they will be ordered one by one even if they are the same. We want to keep the interface simple. You will be writing a program (i.e. logic behind the interface) that will

1. initialize the system with prices of items in an array,
2. display an user friendly interface,
3. initialize any data associated with an order,
4. be *listening* to the activation of the various controls on the device interface,
5. respond to the signal,
6. update data associated with the order,
7. display needed information (items ordered, price) on the interface.

4 Analysis

Read through Chapters 4,5, and 6 (II). make a list of all functional requirements in the format shown in Section 5.6. Make a list of all use cases. Identify the roles and concepts (Chapters 7-9 (II)). Identify the classes and draw the class diagram that includes associations. This will be the UML diagram for the design of YOUR solution to the problem.

5 Implementation Details

1. For each class(es) in the class diagram decide on a pattern that can used to implement it.
2. Define the behaviors (methods) for the classes and them the attributes.
3. Test the classes individually before integrating them into a complete application.

4. Each device that needs response needs to be listened to and when it is activated appropriate action needs to be performed. Listeners and action performed interfaces will be explained to you in lecture and in recitation. There are labs covering these details.

5. Do incremental development. Add the code in steps, and test it before you add more code.

6. Write an application that provides the interface and carries out the operation. Name it WD116.java

7. Document using javadoc conventions. The details of the expected documentation will be provided soon.

6 Materials to Submit & Tentative Grading Policy

A detailed grading policy will be given to you later.

1. (15 points) Requirement analysis submitted in the form of requirement list and use case list : requirements1, usecases1

2. (15 points) Class diagram explaining your design, presented in UMLDesign1.

3. (5 points) A README that gives the list of files in your submission, the purpose of each file.

4. (10 points) Javadoc comments as preamble for a file, class and each method. Javadoc generated first level content in prj1.html.

5. (50 points) Source file: WD116.java and other class files in a java archived WD116.jar file. Archiving details will be given to you.

6. (5 points) A user manual explaining how to install and use your software WD116.

Other items to check: Points that will be taken off if any of these items are not present.

1. (5 points) Internal documentation. Comments.

2. (5 points) Program appearance: indentation, spacing, readability.

3. (5 points) There should be a name box that gives your name, file name and description of the contents of the file.

7 Due Date

`submit_cs116b` filenames separated by space for bsection
`submit_cs116a` filenames separated by space for asection
BEFORE 12 midnight on 9/29/2000.