ORDRE DES INGÉNIEURS DU QUÉBEC

MAY 2023 SESSION

Open-book examination
Calculators : only authorized models
Duration : 3 hours

19-LO-A3 SOFTWARE DESIGN

**This text describes the main features of the IntelliCharge system**

The omnipresence of electronic devices in our daily lives leads to a proliferation of the number of devices using batteries. These batteries must be recharged regularly and possibly changed when they are no longer able to maintain a sufficient charge. To maximize their lifespan, battery manufacturers generally recommend keeping rechargeable batteries charged between 20% and 80% of their maximum capacity and not leaving batteries connected to a charger for long periods of time. Unfortunately, it is easy to forget our plugged-in devices and many of them do not have the necessary functionality to interrupt charging when a charge limit is reached. In order to facilitate the management of the batteries and to maximize their lifespan, we suggest that you develop a new type of intelligent charger, in the form of a programmable wall socket, connected to the internet. This hardware-software system, called IntelliCharge, can be programmed according to different operating modes depending on the type of device connected to it. Here are the main functionalities envisaged for the IntelliCharge system, which includes on the one hand the IntelliChargeApp application, which can run on a smartphone, tablet or computer, and on the other hand one or more IntelliCharger smart devices, which can be plugged into standard electrical outlets and into which the devices to be recharged will be plugged:

1. The IntelliChargeApp application must be able to simultaneously control several chargers of type IntelliCharger and allow the user to name each IntelliCharger by giving it a unique name. Configuration data related to PolyChargers, devices to be recharged, charging methods for each device and schedules may be stored on an external server in a user profile accessible through a website and through the IntelliChargeApp application.
2. Each IntelliCharger device will have a standard electrical outlet into which any electrical device can be plugged, and USB outlets to specifically charge phones and tablets.
3. The IntelliCharge application will allow to choose from three charging modes:
    a) **Smart charging mode**. In smart mode, the IntelliCharger will communicate with the IntelliChargeApp application, installed directly on the device to be recharged, in order to manage the battery charge level in real time. Depending on the type of connection used, the IntelliChargeApp application will be able to automatically recognize the IntelliCharger to be controlled (if it is connected via a USB port), or thanks to the unique name provided by the user. This mode will be useful, for example, to charge phones, tablets and laptops that do not have the capacity to limit the recharge of their battery. The application installed on the device can be configured by the user either to charge the battery up to a fixed level (for example 80%) then interrupt the power supply, or to maintain the charge within a determined range and thus allow the device connected to the charger to be used without risk to the battery.
    b) **Semi-automatic charging mode.** In semi-automatic mode, the charger will charge a device that does not itself have the capacity to run the IntelliChargeApp application. In this situation, the user will install the application on another device (phone, tablet, computer) and use the application to control the charging of the device by providing the application with the characteristics of the device to be charged and the characteristics of its battery. For example, to recharge a small hand-held vacuum cleaner, the user can indicate the brand of the vacuum cleaner

and the technical data of its battery. An external server will make available technical data for a wide variety of devices containing rechargeable batteries. Based on the characteristics of the battery and analyzing the electrical characteristics of the device during recharging, the IntelliChargeApp application will be able to roughly estimate the level of battery recharging and interrupt recharging when a level set by the user will be reached.

c) **Manual charging mode**. In manual mode, the charger will operate as a programmable timer. By using the IntelliChargeApp, the user can program a start time and a recharge duration in minutes. The charger will thus make it possible to recharge a device for a determined time and to interrupt the recharge once the time has elapsed. The recharge period programmed in this way can run once, or repeatedly, according to a schedule programmed using the IntelliChargeApp application.

4. For each device configured in the IntelliChargeApp, charging time and battery status data will be collected and made available for viewing by the user. Based on this data, the IntelliChargeApp application will be able to provide diagnostics and advice in order to extend the life of the batteries and possibly indicate to the user when the time has come to replace a battery. Statistics on electricity consumption and charging costs will also be kept and made available to the user.

**Question 1 - Design process and use case (24 points)**

Modern object-oriented software development processes rely on developing use cases of the software to be designed.

a) In the context of the use cases, define what is meant by an actor and a stakeholder, explain the difference between the two, and give an example of each, taken from the IntelliCharge system (4 points).

b) Draw the UML diagram of the primary use cases of the IntelliCharge software system (context diagram). (12 points)

c) For a single use case of your choice, among those identified in the use case diagram, write:

i. The high-level description of the use case (4 points)

ii. The description of the main scenario of the use case (4 points)

**Question 2 - Logical architecture (22 points)**

Modern software is developed according to logical architectures comprising several layers or levels (multi-tier architecture).

a) Indicate two important advantages of decomposing software into a classic multi-tier architecture (2 points),

b) Propose a multi-level architectural breakdown of the IntelliCharge software. Provide your answer in the form of a UML package diagram (10 points),

c) When designing a multi-tier architecture for an interactive application, a frequently used architectural pattern is the Model-View-Controller (MVC) pattern. Describe each component of this architectural pattern in terms of its responsibilities and link

each component to the appropriate package(s) of the architecture diagram you designed for the IntelliCharge software. (6 points),

d) An emerging approach to designing applications is to favor an architectural decomposition based on services (Service Oriented Architecture or SOA). Explain what a service is architecturally, specify the advantages of the SOA approach and give a possible example of using a service within the framework of the IntelliCharge system. (4 points).

**Question 3 – Object-oriented analysis (22 points)**

a) In the context of object-oriented design, two principles for assigning responsibilities in a system are particularly important, namely the principles of **low coupling** and **high cohesion**. Define in your own words each of these principles and justify the importance of each in an object-oriented design approach. (4 points)

b) Limiting yourself only to the functionalities and concepts described in the contextual text, provide a UML diagram of the domain concepts of the IntelliCharge system (18 points).

**Question 4 - Design Patterns (32 points)**

During the analysis and design phase of the IntelliCharge system, here are some problems that were identified and that you are asked to solve using design patterns:

a) During the charging process, in intelligent charging mode, the IntelliChargeApp application must continuously monitor the charge level of the battery and automatically notify the user when the charge level reaches the pre-set limit he has determined and interrupt the charging process. It is therefore necessary to set up a mechanism for automatically updating the charge level displayed in the application and notify the user once charging is completed.

b) While implementing the automatic update mechanism, you realize that several devices may need to update their status and also inform several entities in the IntelliCharge system. You want to avoid directly coupling devices with the entities that must be updated.

c) When configuring the charger, several charging modes are available, and the user needs to choose the appropriate charging mode for a specific device. The user should be allowed to dynamically select the appropriate charging mode for a given device, and it should be possible to add new charging modes without modifying the code that uses and controls the charging. It is therefore necessary to set up a mechanism that allows to dynamically switch between charging modes and support new ones.

d) To allow the IntelliChargeApp application to control each type of device that must be recharged, a specific object corresponding to each specific device needs to be instantiated. An instantiation mechanism must therefore be provided that allows creating concrete objects of different types that correspond to the characteristics of the device and its battery, without tying the IntelliChargeApp to each concrete class.

For each design problem, you are asked:

i) To identify the appropriate design pattern (from the book by Gamma et al.) that should be used to solve the problem and justify your choice (2 points),

ii) To apply the pattern by providing an annotated class diagram, where the roles identified in the pattern are explicitly associated with the appropriate classes of the IntelliCharge system (4 points).

iii) Discuss the advantages, disadvantages and consequences of using the chosen pattern to solve the design problem, in the specific context of the IntelliCharge system (2 points).