

Documento de requerimientos

Un banco local pretende instalar una nueva máquina de cajero automático (ATM), para permitir a los usuarios (es decir, los clientes del banco) realizar transacciones financieras básicas.

Cada usuario sólo puede tener una cuenta en el banco. Los usuarios del ATM deben poder ver el saldo de su cuenta, retirar efectivo (es decir, sacar dinero de una cuenta) y depositar fondos (es decir, agregar dinero en una cuenta).

La interfaz de usuario del cajero automático contiene los siguientes componentes:

- una pantalla que muestra mensajes al usuario
- un teclado que recibe datos numéricos de entrada del usuario
- un dispensador de efectivo que dispensa efectivo al usuario, y
- una ranura de depósito que recibe sobres para depósitos del usuario.

El dispensador de efectivo comienza cada día cargado con 500 billetes de \$20.

El banco desea que usted desarrolle software para realizar las transacciones financieras que inicien los clientes a través del ATM. Posteriormente, el banco integrará el software con el hardware del ATM. El software debe encapsular la funcionalidad de los dispositivos de hardware (por ejemplo: dispensador de efectivo, ranura para depósito) dentro de los componentes de software, pero no necesita estar involucrado en la manera en que estos dispositivos ejecutan su tarea. El hardware del ATM no se ha desarrollado aún, en vez de que usted escriba un software para ejecutarse en el ATM, deberá desarrollar una primera versión del software para que se ejecute en una computadora personal. Esta versión debe utilizar el monitor de la computadora para simular la pantalla del ATM y el teclado de la computadora para simular el teclado numérico del ATM.

Una sesión con el ATM consiste en la autenticación de un usuario (es decir, proporcionar la identidad del usuario) con base en un número de cuenta y un número de identificación personal (NIP), seguida de la creación y la ejecución de transacciones financieras. Para autenticar un usuario y realizar transacciones, el ATM debe interactuar con la base de datos de información sobre las cuentas del banco (es decir, una colección organizada de datos almacenados en una computadora). Para cada cuenta de banco, la base de datos almacena un número de cuenta, un NIP y un saldo que indica la cantidad de dinero en la cuenta. [Nota: asumiremos que el banco

Programación Avanzada

planea construir sólo un ATM, por lo que no necesitamos preocuparnos para que varios ATMs puedan acceder a esta base de datos al mismo tiempo. Lo que es más, supongamos que el banco no realizará modificaciones en la información que hay en la base de datos mientras un usuario accede al ATM. Además, cualquier sistema comercial como un ATM se topa con cuestiones de seguridad con una complejidad razonable que no contemplaremos en esta instancia. No obstante, para simplificar nuestro ejemplo supondremos que el banco confía en el ATM para que acceda a la información en la base de datos y la manipule sin necesidad de medidas de seguridad considerables].

Al acercarse al ATM (suponiendo que nadie lo está utilizando), el usuario deberá experimentar la siguiente secuencia de eventos:

1. La pantalla muestra un mensaje de bienvenida y pide al usuario que introduzca un número de cuenta.
2. El usuario introduce un número de cuenta de cinco dígitos, mediante el uso del teclado.
3. En la pantalla aparece un mensaje, en el que se pide al usuario que introduzca su NIP (número de identificación personal) asociado con el número de cuenta especificado.
4. El usuario introduce un NIP de cinco dígitos mediante el teclado numérico.
5. Si el usuario introduce un número de cuenta válido y el NIP correcto para esa cuenta, la pantalla muestra el menú principal. Si el usuario introduce un número de cuenta inválido o un NIP incorrecto, la pantalla muestra un mensaje apropiado y después el ATM regresa al paso 1 para reiniciar el proceso de autenticación.

Una vez que el ATM autentica al usuario, el menú principal debe contener una opción numerada para cada uno de los tres tipos de transacciones: solicitud de saldo (opción 1), retiro (opción 2) y depósito (opción 3). El menú principal también debe contener una opción para que el usuario pueda salir del sistema (opción 4). Después el usuario elegirá si desea realizar una transacción (oprimiendo 1, 2 o 3) o salir del sistema (oprimiendo 4).

Si el usuario oprime 1 para solicitar su saldo, la pantalla mostrará el saldo de esa cuenta bancaria. Para ello, el ATM deberá obtener el saldo de la base de datos del banco (que en nuestra aplicación la representaremos mediante una estructura de datos como por ejemplo un Array).

Una vez que el sistema ejecuta una transacción en forma exitosa, debe volver a mostrar el menú principal para que el usuario pueda realizar transacciones

Programación Avanzada

adicionales. Si el usuario elije salir del sistema, la pantalla debe mostrar un mensaje de agradecimiento y después el mensaje de bienvenida para el siguiente usuario.

Actividades:

1. Analizar la documentación de requerimientos a fin de determinar el alcance del problema, objetos, clases y relaciones que permitan resolver el problema de diseño.
2. Realizar el diagrama de casos de uso para modelar las interacciones entre los clientes de un sistema (en este ejemplo práctico, los clientes del banco) y sus casos de uso. El objetivo es mostrar los tipos de interacciones que tienen los usuarios con un sistema sin proveer los detalles.
3. Realizar un diagrama de clases que permita modelar las clases en el sistema que representa el funcionamiento de un cajero automático y sus interrelaciones.
4. Mapear el diagrama de clases a código Java.
5. Se deberá hacer una investigación en la documentación del API de Java de la clase Scanner para la lectura de datos a través de la entrada por teclado. Esto nos permitirá que el usuario pueda interactuar con nuestra aplicación, por ejemplo, eligiendo las opciones para depositar o retirar dinero.