MoovMe

Introducción a la Programación II
Trabajo Práctico Especial
2do Cuatrimestre 2019

Objetivo

Se desea implementar una aplicación para la gestión de servicios de movilidad, con modo de multi-usuario.

Requerimientos del TP

1. **Gestión de Usuarios:**
    a. Habrá 2 tipos de usuarios, los administradores de MoovMe y los clientes que pagan por sus servicios
    b. Se requiere que exista un **ABM de usuarios administradores**
    c. Los **clientes se deben registrar por sus propios medios**, se registran con un número de teléfono celular, por única vez.
    d. Se tiene que tener acceso para los administradores para poder **bloquear y desbloquear clientes**, de acuerdo a las situaciones que se enuncian más adelante en este enunciado.
2. **Persistencia**
    a. Toda la información tiene que estar **persistida** en archivos.
    b. Se requiere la implementación de acceso a datos con uso de **generics**.
    c. Se requiere que la **persistencia sea reemplazable** por implementaciones con Collections en memoria.
3. **Gestión de activos:**
    a. MoovMe dará de alta distintos **tipos de activo**, sean autos, motos, monopatines, bicicletas, etc., tienen que ser dinámicos.
    b. Por cada tipo de activo se tienen que configurar una **cantidad de puntos que se otorgan** al cliente por haber usado dicho activo.
    c. Además cada tipo de activo tiene un **tarifario por minuto de uso**, de acuerdo a la zona en donde se ha usado dicho activo. Un activo siempre se utiliza dentro de una misma zona.
    d. Se requiere implementar un caso de uso, que se va a hacer de manera manual aunque en la realidad esta información llegará de modo automático, que consiste en informar el evento de que un activo está siendo sacado de su zona. En este caso por cada **evento de amenaza de seguridad por tipo de activo se multa** con un valor determinado al cliente y se le bloquea la cuenta, para que no la pueda usar hasta que sea desbloqueada.
    e. Los administradores podrán **crear lotes de compras de activos**. Un lote va a contar con un código autogenerado incremental, y contará también con un tipo de activo, y una cantidad de activos comprados.

f. De esta manera, **se generarán los activos** per se, que tendrán también un código autogenerado, y conocerán su lote al que pertenecen.

4. **Scoring:**
    a. Los cliente podrán **beneficiarse del sistema de scoring**, cada vez que quieran alquilar un activo a través de descuentos en el precio.
        i. **Los descuentos tienen que ser configurables** por tipo de activo, y se configuran indicando tipo de activo, puntaje mínimo requerido, zona y porcentaje de descuento. Por ejemplo, para bicicletas en CABA canjenado 50 puntos se puede tener un descuento
    b. El sistema de scoring contará por zona con un **tablero de líderes mensuales**, que debe recordarse si se cierra y se vuelve a abrir el sistema.
        i. Se deben **ver los 10 mejores puntajes por zona** y debe mostrarse solamente el alias de los clientes.
        ii. Los datos a almacenar serán: Alias, puntaje.
        iii. A final de mes, se debe **premiar a los 3 primeros** de cada tablero con 50% de descuento en su próximo viaje.
5. **Pick up y delivery**
    a. Otra entidad que los administradores deben crear son las **terminales de estacionamiento** de los activos.
    b. Todo activo al ser alquilado por el cliente, debe ser **retirado de una terminal** y cuando lo entregue debe ser también depositado en otra terminal.
    c. Opcionalmente, el cliente, puede **elegir una hora de entrega** y si la cumple se le otorgan 20% más de puntos

Plagio

Se pueden consultar ideas y estrategias de implementación con otros grupos, pero no se puede copiar el código. El plagio entre grupos será penalizado con la desaprobación del trabajo práctico de cada grupo involucrado.

Entregables

El material a entregar se detalla a continuación:

- Informe que detalle el diseño elegido y su justificación.
- Versión digital del código fuente
- Script de compilación para generar el ejecutable (jar) y su ejecución

Plan de trabajo

Se harán 3 entregas a modo de demo, para validar desarrollo, concepto y evolución del sistema.
Los grupos deberán ser de 4 o 5 personas.
Cada entrega tendrá una calificación que será utilizada para la nota final del TP.

La nota del TP será de la siguiente manera: 20% entrega 1 + 30% entrega 2 + 50% entrega final.

La fecha de inicio es el 30/09/2019.
Conformación de los grupos, fecha límite, el 02/10/2019.
La entrega 1 será el 07/10/2019. Se podrán hacer ajustes hasta el 14/10/2019.
Debe existir el repositorio git con una propuesta inicial de solución.
La entrega 2 será el 28/10/2019.
La entrega 3 (final y defensa oral) será el 11/11/2019.

Consideraciones Técnicas

Deberá usarse un repositorio git propio de cada grupo y se deberá dar acceso a la cátedra.
Deberán realizarse tests con JUnit.