



MoovMe es una aplicación para la gestión de servicios de movilidad, con modo de multi- usuario.

A través de ella, los usuarios clientes podrán alquilar activos por cierto tiempo, acumular puntos y ganar descuentos para sus futuros viajes; por otro lado se encuentran los usuarios administradores, que llevan un registro de los viajes realizados y verifican el buen uso de los activos por parte de los clientes.

Luego de reflexionar sobre la situación presentada, decidimos dividir el diseño de la aplicación en cuatro partes: Visual, Model, Service y DAO.

Visual es el boceto de la User Interface, una interfaz simple e intuitiva, donde las distintas pantallas muestran los comandos disponibles, a través de los cuales los usuarios pueden acceder a lo que la aplicación ofrece.

Representa la interacción entre el usuario y la aplicación por consola, ya sean clientes o administradores.

Partiendo de las cosas que van a poder realizar los usuarios, creamos el Service, que se encarga de ejecutar los comandos del Visual, esta característica permitirá rastrear a los usuarios y obtener la información completa sobre su comportamiento, así como también los puntos que se les otorgaran a los clientes dependiendo de los activos y los viajes efectuados.

De esta forma el Service contiene el código que ejecuta todas las condiciones necesarias para el buen funcionamiento de nuestra aplicación.

Todos estos puntos se realizan sosteniéndose sobre el Model, un modelo de clases e interfaces que interactúan mientras se ejecuta la aplicación, estas son la base interna que ayuda al buen desarrollo de MoovMe, e ilustra los actores fundamentales que interaccionan en conjunto, a través de los Service, para llegar a la solución óptima de las situaciones presentadas por la aplicación.

Por último se encuentra el DAO, encargado de almacenar toda la información necesaria para que la aplicación tenga un respaldo y pueda brindar una mejor experiencia a los usuarios que la utilicen, acopiando todos los datos que se necesiten usar más adelante, ya sea en futuros viajes o a la hora de querer canjear puntos, por ejemplo.

Ayudará así al correcto funcionamiento de ciertos comportamientos, tales como clientes bloqueados que en algún futuro se quieran desbloquear. A estos datos se pueden acceder mediante archivos y listas.

Una vez que el diseño de la aplicación esté bien definido, recién comenzaremos con el desarrollo del código, teniendo en cuenta futuras modificaciones que, posiblemente, debamos incluir para facilitar el desarrollo de la solución.

Decidimos organizarnos de esta forma para evitar futuras trabas con cuestiones que no estén definidas de antemano, ya que estas podrían ocasionar muchos cambios en el código que tengamos escrito hasta entonces, retrasando el flujo de escritura de la solución.