

Design Patterns

1. **Abstract Factory** – Different user types (Administrator, Representative, User) are implemented via factory classes. The factory classes implement the AbstractFactory interface.
2. **Singleton** - The factories are implemented with single instances (static variable of the instance)
3. **Adapter** – Database adapter interacts with the back end data in the files. Also handles requests.
4. **Flyweight** – Used in our factories. Uses a hash map that stores references to users that have already been created. Returns the reference if the user exists, otherwise it makes a new user.
5. **MVC (Model View Controller)** - Our model represents the user objects that hold information about the various users. Our view is represented through swing windows displaying the visualization for each of the models of the application. Our controllers interact with both the view and model. The controller listens for button clicks and updates the view accordingly.
6. **Command** - GUI Toolkits (Java Swing – interface Action, method actionPerformed (~Execute))
7. **State** – Each car object has various states as active, reserved, rented. If the car is active, it will be displayed in the main menu view, otherwise if it is reserved/rented, it will not be available to view.