DESIGN PRINCIPLES

# Single Responsibility Principle

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
|  |  |  |  |

# Open/Closed Principle

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
| 2.1. | SessionScreenController and InvoiceScreenController | There’s only 1 way to calculate renting fees and final rent fees. Adding new formula requires changing current classes. | Using Strategy design pattern. |

# Liskov Substitution Principle

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
|  |  |  |  |

# Interface Segregation Principle

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
|  |  |  |  |

# Dependency Inversion Principle

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
| 5.1. | model PaymentTransaction and model CreditCard | PaymentTransaction is coupled with CreditCard (concrete class). Adding new type of cards requires changing PaymentTransaction | Use a parent class over all type of Cards. Then let PaymentTransaction to use this parent class only |