Heavy equipment rental webpage

Analysis and Design Document

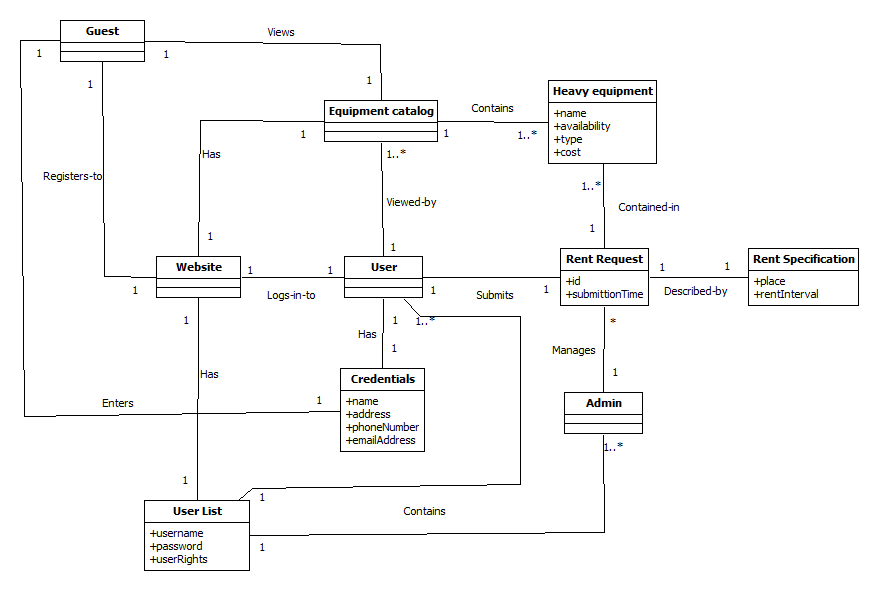
# Elaboration – Iteration 1.1

# Domain Model

List of conceptual class candidates:

*Guest – User – Admin –Website – Equipment Catalog – Heavy Equipment – Rent Request – Rent Specification – User Credentials – User List*

Partial Heavy Equipment Rental Website domain model:

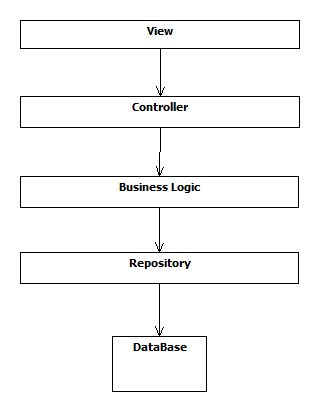


# Architectural Design

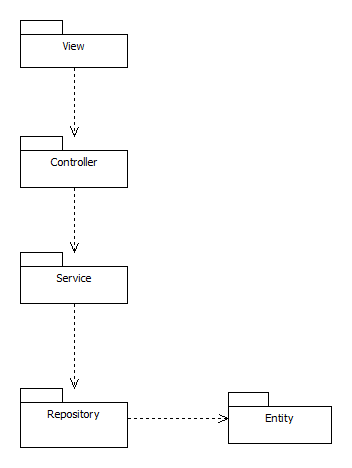
## Conceptual Architecture

Layered architecture provides maintainability. If its interface is unchanged, a new layer with extended functionality can replace an existing layer without changing other parts of the system.

The Business Logic layer contains the independent logic. High-level Presentation layer doesn’t have to interact directly with low-level Repository layer, it interacts through the services contained in the Business Logic layer.

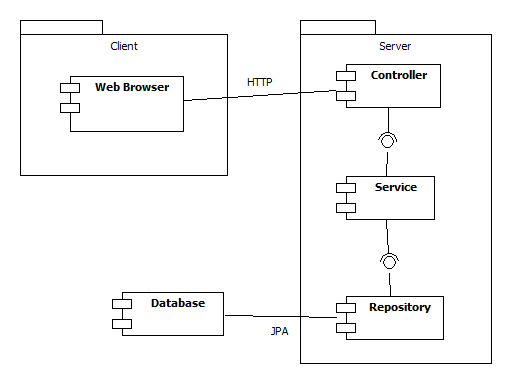


## Package Design

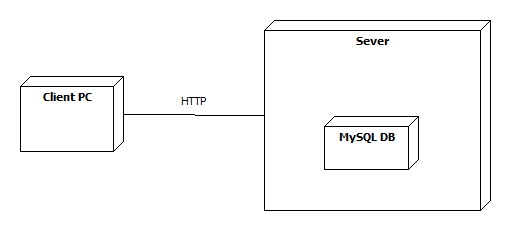
**

## Component and Deployment Diagrams

Component diagram:



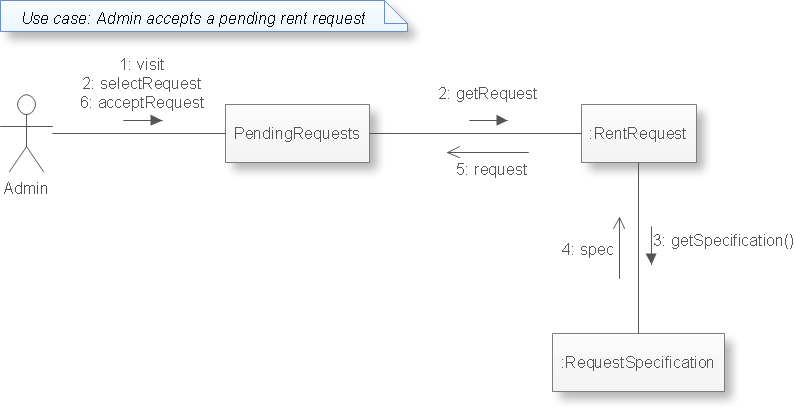
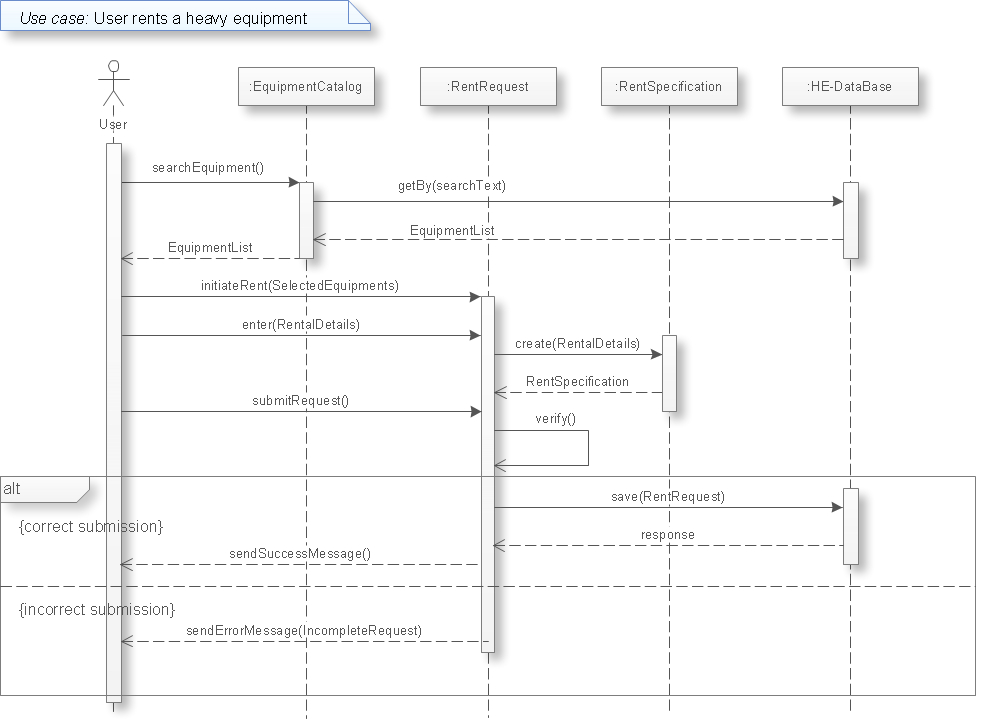
Deployment Diagram:



# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior



## Class Design

Use the Abstract Factory pattern when

• a system should be independent of how its products are created, composed,

and represented.

• a system should be configured with one of multiple families of products.

• a family of related product objects is designed to be used together, and you

need to enforce this constraint.

• you want to provide a class library of products, and you want to reveal just

their interfacneost, their implementations.

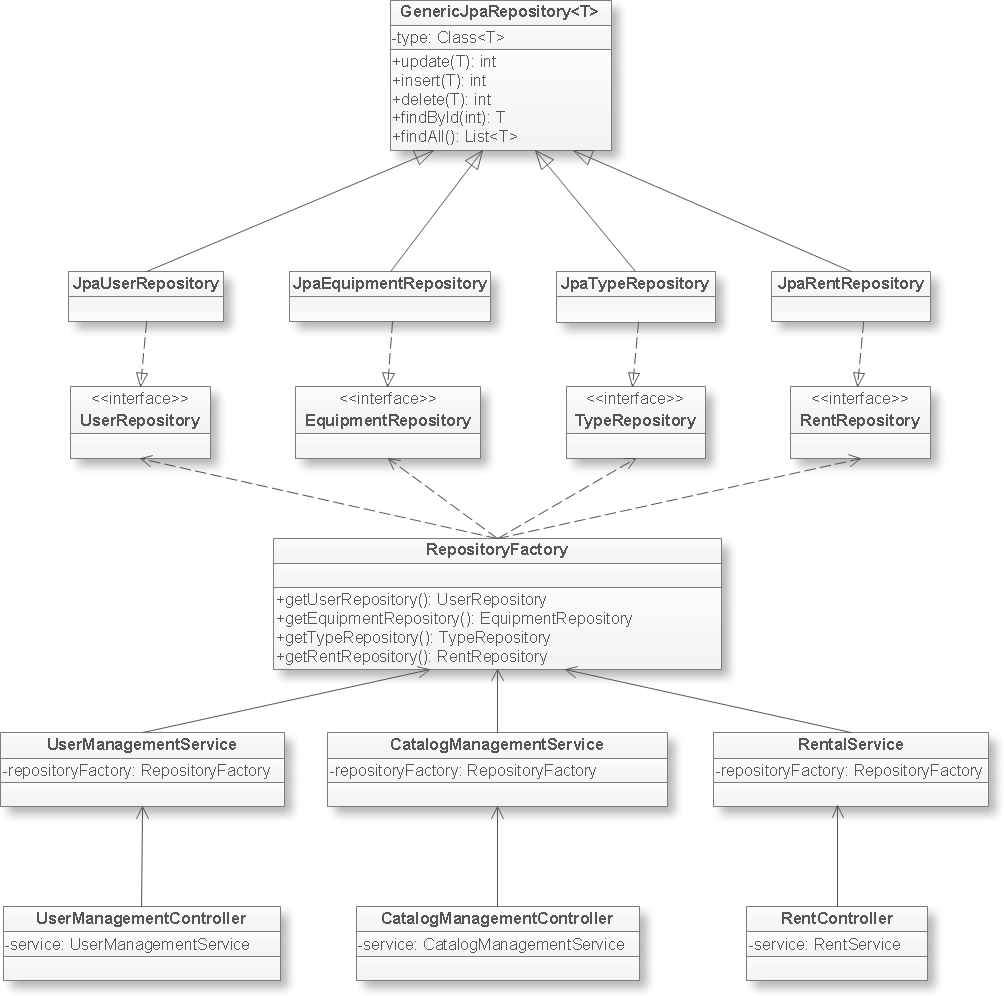
Use the Factory Method pattern when

• a class can't anticipate the class of objects it must create.

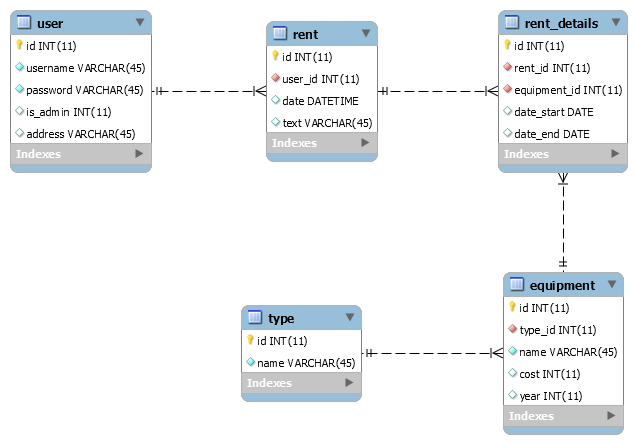
• a class wants its subclasses to specify the obje cts it creates.

• classes delegate responsibility to one of several helper subclasses, and you

want to localize the knowledge of which helper subclass is the delegate.



# Data Model



# Test Strategy

Unit testing and postman. Testing use case scenarios.

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

*[Present future improvements for the system]*