California State University Fullerton

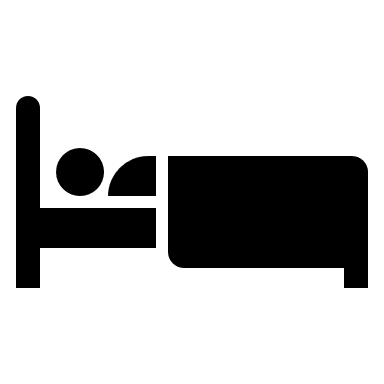
CPSC 462



Object Oriented Software Design

SW Architecture Document (SAD)

for the



Hotel Room Reservation System

|  |
| --- |
| **Allen Rivas** |
| Chief Technical Officer (CTO) |
| [allen.rrivas30@csu.fullerton.edu](mailto:allen.rrivas30@csu.fullerton.edu) |

Revision History:

| Version | Date | Summary of Changes | Author |
| --- | --- | --- | --- |
| 1.1 | April 27, 2021 | * Initial Release. Updated Controller GRAPS Decision. Updated Logical View, providing a Package Diagram and Interface Diagram for Domain Layer and Technical Services. | Allen Rivas |
| 1.2 | May 16, 2021 | * Add information into the Low Coupling/High Cohesion table and Design Patterns. Updated with change bars. | Allen Rivas |

Table of Contents

[1 Architectural Representation 1](#_Toc50289891)

[2 Architectural Decisions 2](#_Toc50289892)

[2.1 Low Coupling / High Cohesion GRASP Decision 2](#_Toc50289893)

[2.1.1 Decision to be made 2](#_Toc50289894)

[2.1.2 Options Considered 2](#_Toc50289895)

[2.1.3 Selection and Rationale 2](#_Toc50289896)

[2.2 Creator GRASP Decision 2](#_Toc50289897)

[2.2.1 Decision to be made 2](#_Toc50289898)

[2.2.2 Options Considered 2](#_Toc50289899)

[2.2.3 Selection and Rationale 3](#_Toc50289900)

[2.3 Information Expert GRASP Decision 3](#_Toc50289901)

[2.3.1 Decision to be made 3](#_Toc50289902)

[2.3.2 Options Considered 3](#_Toc50289903)

[2.3.3 Selection and Rationale 3](#_Toc50289904)

[2.4 Controller GRASP Decision 3](#_Toc50289905)

[2.4.1 Decision to be made 3](#_Toc50289906)

[2.4.2 Options Considered 4](#_Toc50289907)

[2.4.3 Selection and Rationale 4](#_Toc50289908)

[3 Logical View 5](#_Toc50289909)

[3.1 Package Diagrams 5](#_Toc50289910)

[3.1.1 Presentation (UI) Layer Components 5](#_Toc50289911)

[3.1.2 Domain (Application) Layer Components 5](#_Toc50289912)

[3.1.2.1 Profile Manager 5](#_Toc50289913)

[3.1.2.2 Reservation Manager 5](#_Toc50289914)

[3.1.2.3 Session Manager 5](#_Toc50289915)

[3.1.3 Technical Services Layer Components 5](#_Toc50289916)

[3.1.3.1 Logging 5](#_Toc50289917)

[3.1.3.2 Persistance 5](#_Toc50289918)

[3.2 Interface Diagrams 5](#_Toc50289920)

[3.2.1 Presentation (UI) Layer Interface Diagram 5](#_Toc50289921)

[3.2.2 Domain Layer Interface Diagram 5](#_Toc50289922)

[3.2.3 Technical Services Interface Diagram 5](#_Toc50289923)

[3.3 Design Patterns 6](#_Toc50289924)

[3.3.1 Polymorphism GRASP Pattern 6](#_Toc50289925)

[3.3.1.1 Generalization / Specialization Diagrams 6](#_Toc50289926)

[3.3.1.2 Factory Pattern Diagrams 6](#_Toc50289927)

[3.3.1.3 Source Code References 6](#_Toc50289928)

[3.3.2 Protected Variations GRASP Pattern 6](#_Toc50289929)

[3.3.2.1 Generalization / Specialization Diagrams 6](#_Toc50289930)

[3.3.2.2 Abstract Factory Pattern Diagrams 6](#_Toc50289931)

[3.3.2.3 Source Code References 6](#_Toc50289932)

# Architectural Representation

This document summarizes the architectural understanding of the hotel room reservation system. The key parts in this document are the architectural decisions, specifically the controller GRASP decision, and the logical view which includes a representation of the package diagram and interface diagrams. In the controller GRASP decision, we provide two alternatives where each has a class and sequence diagram, where then a selection is made and provided with rationale. Our main goal is to identify which controller sub pattern is being used between either façade controller or session controller. In the logical view, is provided a package diagram identifying the UI, Domain, and Technical Services layers. Within each of these layers are also identified the different sub packages, which also are included with descriptions of each of the components. There is also provided interfaces diagrams for the Domain Layer and Technical Services where is provides an understanding of what each of the sub packages represent.

# Architectural Decisions

## Low Coupling / High Cohesion GRASP Decision

### Decision to be made

<Describe the specific dilemma you’re facing, for example: What specific question are you trying to answer? What specific problem are you trying to solve? Include in your description the properties a good and poor decision will have>

### Options Considered

| Low Coupling / High Cohesion | Static View | Dynamic View |
| --- | --- | --- |
| Option 1  (Rejected) |  |  |
| Option 2  (Selected) |  |  |
| Design Model Reference | Design Model Static View, Page 1 on the Design Model documentation. | Design Model Dynamic View, under the Dynamic Folder in the VPP file. |

### Selection and Rationale

Option 1 has been discarded because the SessionManagerHandler and GuestSession need to share information with both the ProfileManagerHandler and ReservationManagerHandler. This results in high coupling.

Option 2 has been selected because we can create a Profile Object that is shared with just the ProfileManagerHandler, and not the other handler metioned above. This results in low coupling. The SessionManagerHandler and ReservationManagerHandler have information provided and has separate responsibilities so that results in high cohesion. In the end this represents low coupling and high cohesion.

## Creator GRASP Decision

### Decision to be made

<Describe the specific dilemma you’re facing, for example: What specific question are you trying to answer? What specific problem are you trying to solve? Include in your description the properties a good and poor decision will have>

### Options Considered

| Creator | Static View | Dynamic View |
| --- | --- | --- |
| Option 1  (Rejected) | <insert rejected class diagram SNIPPET here> | <insert rejected sequence diagram SNIPPET here> |
| Option 2  (Selected) | <insert Selected class diagram SNIPPET here> | <insert rejected sequence diagram SNIPPET here> |
| Design Model Reference | <point to paragraph, page number, and where on the page where the selected option snippet appears in your bigger, overall design’s Static View> | <point to paragraph, page number, and where on the page where the selected option snippet appears in your bigger, overall design’s Dynamic View> |

### Selection and Rationale

Option 1 has been discarded because <…>

Option 2 has been selected because <…>

## Information Expert GRASP Decision

### Decision to be made

<Describe the specific dilemma you’re facing, for example: What specific question are you trying to answer? What specific problem are you trying to solve? Include in your description the properties a good and poor decision will have>

### Options Considered

| Information Expert | Static View | Dynamic View |
| --- | --- | --- |
| Option 1  (Rejected) | <insert rejected class diagram SNIPPET here> | <insert rejected sequence diagram SNIPPET here> |
| Option 2  (Selected) | <insert Selected class diagram SNIPPET here> | <insert rejected sequence diagram SNIPPET here> |
| Design Model Reference | <point to paragraph, page number, and where on the page where the selected option snippet appears in your bigger, overall design’s Static View> | <point to paragraph, page number, and where on the page where the selected option snippet appears in your bigger, overall design’s Dynamic View> |

### Selection and Rationale

Option 1 has been discarded because <…>

Option 2 has been selected because <…>

## Controller GRASP Decision

### Decision to be made

Problem: What first object should handle functionality?

Solution 1: Façade controller

Represents the overall system, this is helpful when there are not too many system events.

Solution 2: Session controller

Represents a use case scenario within the system operation occurs.

The decision is that there will be using the use case scenario within the system operation.

### Options Considered

| Controller | Static View | Dynamic View |
| --- | --- | --- |
| Option 1  (Rejected) |  |  |
| Option 2  (Selected) |  |  |
| Design Model Reference | Found in the Static View Diagram, in Design Model File, on page 1 under the Hotel Reservation System. | Found in the Dynamic View, in Design Model File, on Page 2 under the Login Sequence of Execution. |

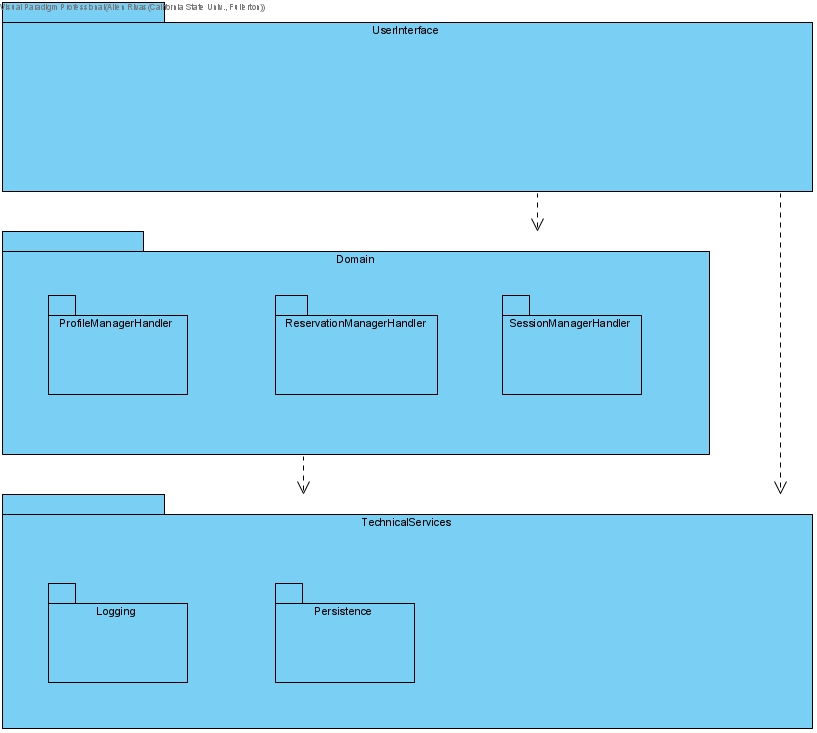
### Selection and Rationale

Option 1 has been discarded because having a primary object that has all the different user operations is risky and goes against the idea of having low coupling and high cohesion.

Option 2 has been selected because having a session handler is helpful by handling all the different user operations.

# Logical View

## Package Diagrams



### Presentation (UI) Layer Components

N/A

### Domain (Application) Layer Components

#### Profile Manager

Profile manager sub package contains all the CRUD operations that the user can do. The user can create, update, or delete the information that is within the user’s profile.

#### Reservation Manager

Reservation Manager sub package contains all the reservation operations that are available for the user. From getting the reservation, settling, and preparing the bill. To assigning the room and generating a key for the user to open their reserved room.

#### Session Manager

Session Manager sub package contains the authentication and termination of the user session. Essentially, when the user logs in and when the user terminates their login session.

### Technical Services Layer Components

#### Logging

Logging sub package essential is checking for logging in errors. Where the user is not able to get into their user profile.

#### Persistence

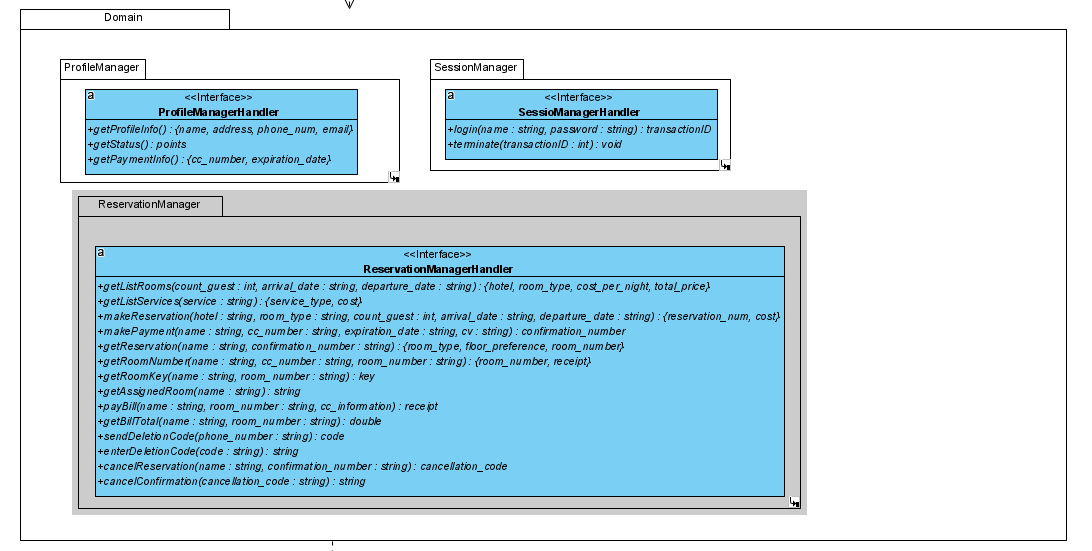
Persistence sub package contains all the information about the different users. Essentially, it is the database that contains all the users’ information from the hotel.

## Interface Diagrams

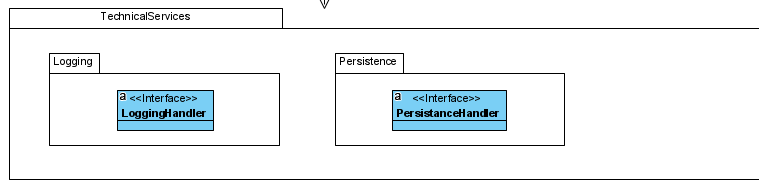
### Presentation (UI) Layer Interface Diagrams

N/A

### Domain Layer Interface Diagrams



### Technical Services Interface Diagrams



## Design Patterns

### Polymorphism GRASP Pattern

#### Generalization / Specialization Diagrams

| Static View |
| --- |
|  |
| The different payments (PayPal, Venmo, Debit and Credit card) are inheriting from the PaymentType providing the all the different user information. |

#### Factory Pattern Diagrams

| Static View | Dynamic View |
| --- | --- |
|  |  |
| PaymentType is inheriting from the ProfileMangerHandler so it can provide the information from the different ways of payment. | Returning the payment information from the user that is logged in. |

#### Source Code References

| Source code file name | Line number(s) |
| --- | --- |
| Objects.hpp | 35-68 |
|  |  |

### Protected Variations GRASP Pattern

#### Generalization / Specialization Diagrams

| Static View |
| --- |
|  |
| GuestSession is inherting from the SessionManagerHandler so it can let the session know that it is currently logged in as a guest. |

#### Abstract Factory Pattern Diagrams

| Static View | Dynamic View |
| --- | --- |
|  |  |
| GuestSession is inherting from the SessionManagerHandler so it can let the session know that it is currently logged in as a guest. | Shows that a session is being started by a guest login |

#### Source Code References

| Source code file name | Line number(s) |
| --- | --- |
| Objects.hpp | 55-66 |
|  |  |