System Design

for

Gym Reservation System

Prepared by:

Daler Karimov, Guillaume Reveillere, Hangil Kim,

Rashad Aliyev, Viktor Shin, Youngju Kim

CS408B Computer Science Project

KAIST

Contents

[List Of Figures 3](#_Toc257811204)

[1. GUI Prototype 4](#_Toc257811205)

[1.1 User Interface 4](#_Toc257811206)

[1.1.1 Reservation Page 4](#_Toc257811207)

[1.1.2 Pop-up Window 4](#_Toc257811208)

[1.2 Manager Interface 5](#_Toc257811209)

[1.2.1 Reservation Page 5](#_Toc257811210)

[1.2.2 Pop-up Window 5](#_Toc257811211)

[2. Overall Class Diagram 6](#_Toc257811212)

[2.1 Class Diagram 6](#_Toc257811213)

[4.2 Class Description 6](#_Toc257811214)

[4.2.1 Client Class 6](#_Toc257811215)

[4.2.2 Reservation Class 6](#_Toc257811216)

[4.2.3 Court Class 7](#_Toc257811217)

[4.2.4 Gym Class 7](#_Toc257811218)

[3. Sequence Diagram 8](#_Toc257811219)

[3.1 Use Case 1. User Places a reservation 8](#_Toc257811220)

[3.2 Use Case 2. User changes or cancels reservation 9](#_Toc257811221)

[3.3 Use Case 3. User views others users’ reservation 9](#_Toc257811222)

[3.4 Use Case 4. Manager updates/deletes/adds reservations 10](#_Toc257811223)

[4. Deployment Diagram 11](#_Toc257811224)

# List Of Figures

[Figure 1. Reservation Page for User 3](#_Toc257811172)

[Figure 2. Pop-window for filling information (User Interface) 4](#_Toc257811173)

[Figure 3. Reservation Page for Manager 4](#_Toc257811174)

[Figure 4. Pop-window for modifying. (Manager interface) 5](#_Toc257811175)

[Figure 5. Overall Class Diagram 6](#_Toc257811176)

[Figure 6. Sequence Diagram for Use Case 1. 8](#_Toc257811177)

[Figure 7. Sequence Diagram for Use Case 2. 9](#_Toc257811178)

[Figure 8. Sequence Diagram for Use Case 3. 9](#_Toc257811179)

[Figure 9. Sequence Diagram for Use Case 4. 10](#_Toc257811180)

[Figure 9. Deployment Diagram. 11](#_Toc257811181)

# GUI Prototype

## 1.1 User Interface

### 1.1.1 Reservation Page

Figure 1. Reservation Page for User

### 1.1.2 Pop-up Window

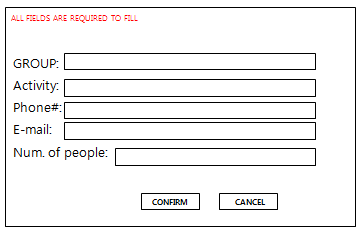


Figure 2. Pop-window for filling information (User Interface)

## 1.2 Manager Interface

### 1.2.1 Reservation Page

Figure 3. Reservation Page for Manager

### 1.2.2 Pop-up Window

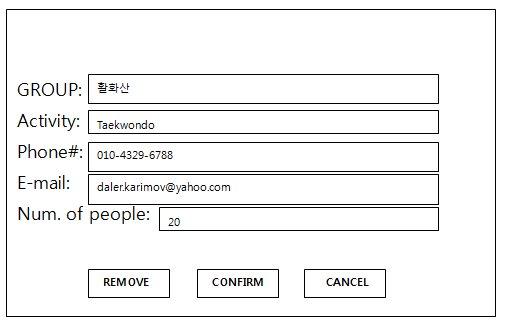


Figure 4. Pop-window for modifying. (Manager interface)

# CLASS diagram.bmpOverall Class Diagram

## 2.1 Class Diagram

Figure 5. Overall Class Diagram

## 2.2 Class Description

### 2.2.1 Client Class

“Client” class has information about people who are using our system. They are: students (called “user”), gym manager, administrator; all 3 types are defined as sub-classes of Client. Class’ attributes are as shown on the Figure 5. Class “GymManager” is communicating with “Gym” class, i.e. every gym manager is able to edit information about his own gym.

### 2.2.2 Reservation Class

“Reservation” class is responsible for handling processes of making or modifying reservations by clients. Reservations are done by client for court, therefore “Reservation” class has GymID and CourtID as attributes,

### 2.2.3 Court Class

“Court” class is composed class of “Gym”, because it has strong whole-part relation with it. This class attributes are CourtID, GymID and location, last attribute stands for location within the gym, so that there is no ambiguous point about reservations.

### 2.2.4 Gym Class

“Gym” class has information about courts that it has, manager who is in charge of it and location in campus where it is located.

# Sequence Diagram

## 3.1 Use Case 1. User Places a reservation

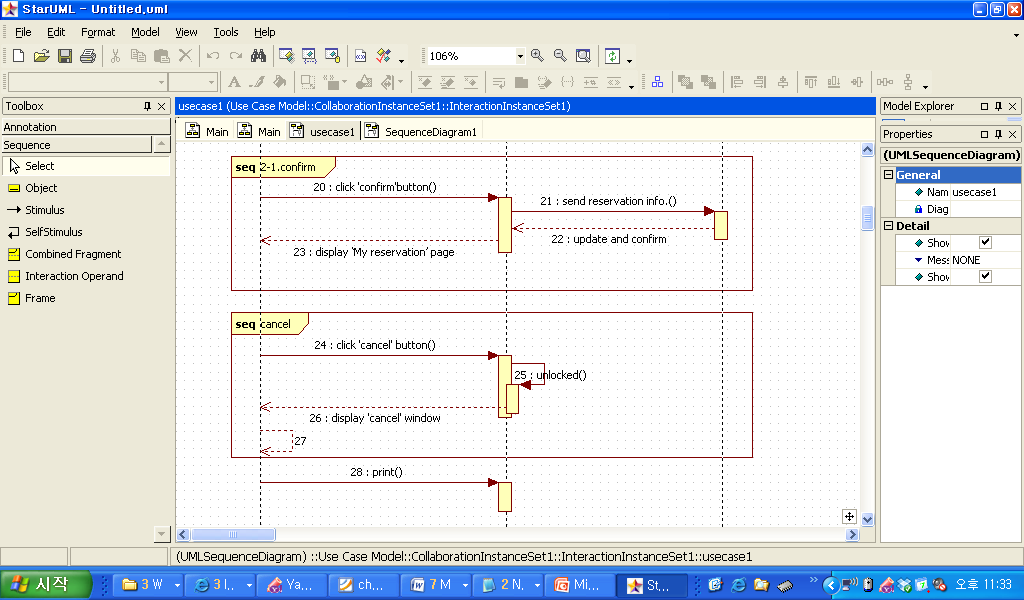
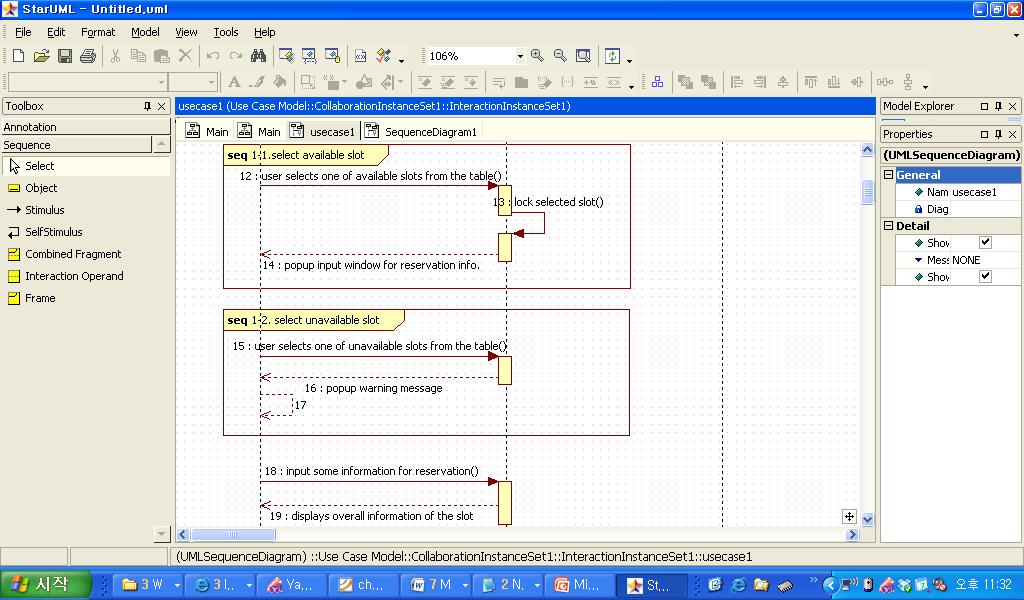
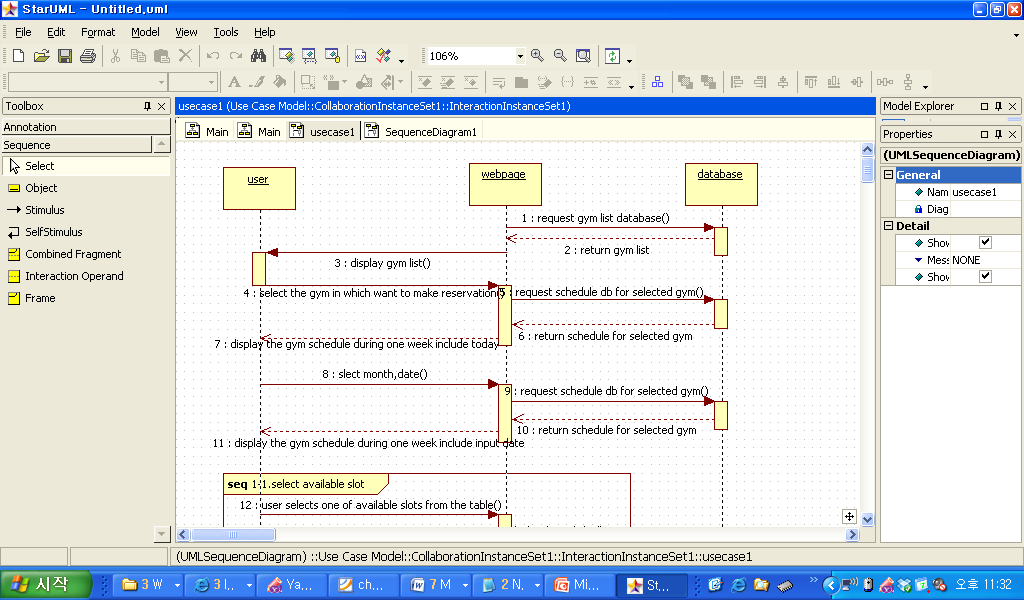


Figure 6. Sequence Diagram for Use Case 1.

## 3.2 Use Case 2. User changes or cancels reservation



Figure 7. Sequence Diagram for Use Case 2.

## 3.3 Use Case 3. User views others users’ reservation



Figure 8. Sequence Diagram for Use Case 3.

## 3.4 Use Case 4. Manager updates/deletes/adds reservations



Figure 9. Sequence Diagram for Use Case 4.

# deployment_final.jpgDeployment Diagram

Figure 9. Deployment Diagram.

LAMP (WAMP) architecture was chosen, because of its advantage for development and managing. HTTP and Database server were separated, so that administrator of the system has right to decide whether he wants to keep them on same machine or on different machines.