Development Manual

Team CtrlAltDefeat - Brittany Nall & Melissa Smith

CS414 – Fall 2018

Table of Contents

[1 INTRODUCTION 3](#_Toc531032503)

[1.1 Purpose 3](#_Toc531032504)

[1.2 Tools Needed 3](#_Toc531032505)

[2 SETUP GUIDE 3](#_Toc531032506)

[2.1 Installing Eclipse 3](#_Toc531032507)

[2.2 Adding Required Plugins 3](#_Toc531032508)

[2.3 Clone the GitHub Repository 3](#_Toc531032509)

[2.4 Import Existing Project in Eclipse 3](#_Toc531032510)

[2.5 Add XStream to Eclipse Project 3](#_Toc531032511)

[2.6 Add GitHub Repository to Eclipse 3](#_Toc531032512)

[3 DEVELOPMENT GUIDE 3](#_Toc531032513)

[3.1 Using WindowBuilder to Modify the UI 3](#_Toc531032514)

[3.2 Running the Gym Management System 4](#_Toc531032515)

[3.3 Running the JUnit Tests 4](#_Toc531032516)

[4 PRODUCT DESCRIPTION 4](#_Toc531032517)

[4.1 Classes 4](#_Toc531032518)

[4.1.1 Address 4](#_Toc531032519)

[4.1.2 Customer 4](#_Toc531032520)

[4.1.3 Employee 4](#_Toc531032521)

[4.1.4 Equipment 4](#_Toc531032522)

[4.1.5 Exercise 4](#_Toc531032523)

[4.1.6 Manager 5](#_Toc531032524)

[4.1.7 MembershipStatus 5](#_Toc531032525)

[4.1.8 PersonalInformation 5](#_Toc531032526)

[4.1.9 Schedule 5](#_Toc531032527)

[4.1.10 Trainer 5](#_Toc531032528)

[4.1.11 User 5](#_Toc531032529)

[4.1.12 UserType 5](#_Toc531032530)

[4.1.13 Weekday 5](#_Toc531032531)

[4.1.14 WorkoutRoutine 5](#_Toc531032532)

[4.1.15 WorkTime 5](#_Toc531032533)

[4.2 JUnit Tests 6](#_Toc531032534)

[4.3 GUI 6](#_Toc531032535)

[4.3.1 GymSystemUI 6](#_Toc531032536)

[4.3.2 GymSystemCreator 6](#_Toc531032537)

[4.3.3 GymSystemController 6](#_Toc531032538)

[4.3.4 SystemDao 6](#_Toc531032539)

[**4.3.5** XStream 7](#_Toc531032540)

# INTRODUCTION

## Purpose

The purpose of this document is to provide instructions on how to get a development environment set up in order to make required changes and maintain the system.

## Tools Needed

1. Eclipse IDE
2. WindowBuilder Plugin
3. JUnit5

# SETUP GUIDE

## Installing Eclipse

Download and install the Eclipse IDE from the provided link and follow the installation instructions.

[*https://www.eclipse.org/downloads/packages/installer*](https://www.eclipse.org/downloads/packages/installer)

## Adding Required Plugins

Download and install WindowBuilder from the provided link and follow the installation instructions.

[*https://www.eclipse.org/windowbuilder/download.php*](https://www.eclipse.org/windowbuilder/download.php)

## Clone the GitHub Repository

1. Clone the GitHub repository using the following link:

[*https://github.com/britnall/cs414-f18-801-CtrlAltDefeat*](https://github.com/britnall/cs414-f18-801-CtrlAltDefeat)

1. Checkout the ‘master’ branch to begin development.

## Import Existing Project in Eclipse

1. Select File > Import > Gradle > Existing Gradle Project
2. The project root directory should be the path to the src directory in the cloned cs414-f18-801-CtrlAltDefeat GitHub Repository

## Add GitHub Repository to Eclipse

1. Select Window > Show View > Other > Git > Git Repositories
2. Select the cloned cs414-f18-801-CtrlAltDefeat GitHub Repository

# DEVELOPMENT GUIDE

## Using WindowBuilder to Modify the UI

1. Right Click on GymSystemUI.java (found in the edu.colostate.cs.cs414.ctrlaltdefeat.UI package) > Open With > WindowBuilderEditor
2. Select the Design tab at the bottom of the code block window to modify the user interface and generate the user interface code

## Running the Gym Management System

To run the Gym Management System, run GymSystemUI.java (found in the edu.colostate.cs.cs414.ctrlaltdefeat.UI package) as a Java Application from Eclipse.

## Running the JUnit Tests

To run all of the Gym Management System unit tests, run TestAll.java (found in the edu.colostate.cs.cs414.ctrlaltdefeat.Test package) as a JUnit Test from Eclipse.

# PRODUCT DESCRIPTION

## Classes

These classes are used to create the basic functionality of the gym system. Each consists of attributes and getters and setters. Below is described what information they are used to contain and their use in the gym system.

### Address

An address object consists of a street, city, state and zip code. Addresses are then assigned to the relevant personal information of a customer or an employee.

### Customer

The customer class contains the personal information of a customer, their membership status, and the workout routines assigned to the customer. When a customer is added to the system they are only required to have personal information and membership status is set to ACTIVE upon creation. Workout routines for a customer can be assigned or removed through the user interface but are not necessary for the creation of a customer.

### Employee

The class contains the information for an employee of the gym and consists of the Employee’s user information, personal information and user type. The class is extended by Manager and Trainer.

### Equipment

Equipment consists of the information relevant to a specific equipment in the gym’s inventory. Its fields are name, picture (a file), and quantity which indicates the number of this particular equipment that are part of the gym’s inventory. Equipment objects are typically used as part of the equipment inventory but can also be used if an exercise uses specific equipment.

### Exercise

The class has a name, number of sets, number of repetitions, and the equipment used for the exercise. Exercises are typically used as part of workout routines.

### Manager

The manager class extends the Employee class creating an employee of with the user type of Manger in the system.

### MembershipStatus

This is a class defining the enumeration values that are used to indicate a customer’s membership status in the system. A Customer can have a membership status of active or inactive within the system.

### PersonalInformation

The class contains the personal information of a Customer, Trainer, or Manager within the system. Such as first name, last name, email, phone, health insurance provider, and address.

### Schedule

Schedule is used to create the work schedules of trainers for the gym system. It consists of an array of WorkTime objects that make up the employee’s schedule.

### Trainer

Extends the Employee class allowing the creation of an employee with a user type of Trainer in the system.

### User

Contains the login information of a user on the system.

### UserType

A class defining the enumeration values for the user types allowed on the system, specifically Manager or Trainer.

### Weekday

A class defining the enumeration values to be used in the WorkTime class with each value representing a day of the week.

### WorkoutRoutine

A workout routine consists of a routine name and then the list of exercises that are to be performed as part of the routine.

### WorkTime

Each object created for this class creates a representation of a day of the week and the start and end times for a shift at the gym. This class is used in conjunction with Schedule to keep track of the trainer’s work schedule at the gym. It uses the enumeration values provided in Weekday to represent each day of the week.

## JUnit Tests

The JUnit test classes are used to test and verify the basic functionality of the gym system. Ensuring each class can perform the tasks it was designed for. They can be run individually or all together as part of a test suite using TestAll.java to verify any changes in the code.

Each test includes test cases to verify the corresponding class’s constructor(s), getters, setters and the overloaded hash code and equals functions.

* AddressTest
* CustomerTest
* EmployeeTest
* EquipmentTest
* ExerciseTest
* PersonalInformationTest
* ScheduleTest
* UserTest
* WorkoutRoutineTest
* WorkTimeTest

## GUI

To create a simple GUI for our gym system, we utilized a tool referred to as WindowBuilder. WindowBuilder allows for the simple creation of simple forms and complex windows with the framework of the Java code being generated automatically using the design view. The generated code doesn’t require any additional custom libraries to compile.

### GymSystemUI

Contains the code for the user interface functionality and consists of one JFrame with multiple JPanels to provide the different screens for the user to access the functionality of the system.

### GymSystemCreator

Contains the functions used create new objects in the system using the input from the user interface.

### GymSystemController

Contains the functions that interacts between the user interface and SystemDao to control the objects in the system and persist the data using XStream.

### SystemDao

Contains the data and functions to manipulate the data that is in the gym management system to be serialized using XStream.

### XStream

XStream is a simple Java-Based library to serialize Java objects to xml and back. It was chosen because it is easy to use, provides default mapping for most of the objects to be serialized, and other features.

For more documentation on XStream can be found at:

<https://www.tutorialspoint.com/xstream/index.htm>

More information on how to use it can be found at:

<http://x-stream.github.io/persistence-tutorial.html>