Development Manual

Team CtrlAltDefeat - Brittany Nall & Melissa Smith

CS414 – Fall 2018

TABLE OF CONTENTS

1. INTRODUCTION 3
   1. Purpose 3
   2. Tools 3
2. QUICK START GUIDE 3
   1. Getting Setup 3
   2. Getting Started 3
3. SETUP GUIDE 3
   1. Installing Eclipse 3
   2. Adding Required Plugins 3
   3. XStream 3
   4. Pulling the code from GitHub 4
4. PRODUCT DESCRIPTION 4
   1. Classes 4
   2. JUnit Tests 6
   3. GUI 7
   4. XStream 7
5. **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*

* + 1. Eclipse IDE
    2. WindowBuilder Plugin
    3. XStream

1. **QUICK START GUIDE**

# *Getting Setup*

* + 1. Install the Eclipse IDE
       1. <https://www.eclipse.org/downloads/packages/installer>
    2. Add in plugin for WindowBuilder GUI
       1. <https://www.eclipse.org/windowbuilder/download.php>
    3. Add XStream jar files
       1. <https://www.tutorialspoint.com/xstream/xstream_quick_guide.htm>

# *Getting Started*

* + 1. Pull the code from gitHub from the provided link into the eclipse IDE:
       1. <https://github.com/britnall/cs414-f18-801-CtrlAltDefeat>

1. **SETUP GUIDE**

# *Installing Eclipse*

* + 1. Download and install the Eclipse IDE from the provided link and follow the installation instructions.
       1. <https://www.eclipse.org/downloads/packages/installer>

# *Adding Required Plugins*

* + 1. Download and install the WindowBuilder from the provided link and follow the installation instructions.
       1. <https://www.eclipse.org/windowbuilder/download.php>

# *XStream*

XStream is a java library to serialize objects to XML (or JSON) and back in order to persist data across program executions.

* + 1. More information on XStream and how to use it can be found at: <http://x-stream.github.io/persistence-tutorial.html>
    2. Add the Jar files to the eclipse project.

# *Pulling the code from GitHub*

* + 1. The main code base is contained on the project teams gitHub repository at the following link: <https://github.com/britnall/cs414-f18-801-CtrlAltDefeat>
    2. Add the git repository to Eclipse
    3. Window > Show Views > Other > Git Repository
    4. Select “Clone the git repository” to begin the process and follow the prompts.
    5. Select “Clone the git repository” to begin the process and follow the prompts.

1. **PRODUCT DESCRIPTION**

# *Classes*

Theses 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.

* + 1. Address

An address object consists of a a street, city, state and zip code. Addresses are then assigned to the relevant personal information of a customer or an employee. An address is not required upon the creation of a customer or an employee but can be added then or at a later time when modifying personal information.

* + 1. 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 at anytime but are not necessary for the creation of a customer.

* + 1. 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.

* + 1. 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 are typically used as part of an exercise if a certain equipment type is needed for the exercise.

* + 1. Exercise

The class has name, number of sets, and a list of equipment that are used for the exercise. It is extended by the class ExerciseReps and the class ExerciseTime. Exercises are typically used as part of workout routines.

* + 1. Manager

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

* + 1. MembershipStatus

This its a class defining the enums 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.

* + 1. PersonalInformation

The class contains the personal information of a Customer, Trainer, or Manager within the system. Such as first name, last name, email, phone, healthInsuranceProvider, and the address. It also assigns an id number to each object allowing each to be unique.

* + 1. Schedule

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

* + 1. Trainer

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

* + 1. User

Contains the login information of a user on the system.

* + 1. UserType

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

* + 1. Weekday

A class defining the enums to be used in the WorkTime class with one representing each day of the week.

* + 1. WorkoutRoutine

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

* + 1. 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 employees work schedule at the gym. It uses the enums 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 is can perform the tasks it was designed for. They can be run individually or all together as part of a test suite to verify any changes in the code.

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

* + 1. AddressTest
    2. CustomerTest
    3. EmployeeTest
    4. EquipmentTest
    5. ExerciseTest
    6. ExerciseRepsTest
    7. ExerciseTimeTest
    8. GymSystemTest
    9. ManagerTest
    10. PersonalInformationTest
    11. ScheduleTest
    12. TrainerTest
    13. UserTest
    14. WorkoutRoutineTest
    15. 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.

* + 1. UI

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.

* + 1. GymSystemCreator

Contains the functions to create new data on the system.

* + 1. GymSystemController

Contains the functions that interacts between the UI and the backend to modify the backend data.

* + 1. SystemDao

GymSystemController

Contains the functions that interacts between the UI and the backend to modify the backend data.

# *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 the following link:

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