Linnæus University

Object Oriented Analysis and Design with UML 1DV407

Assignment 2

(Peer Review)

Students:

Shahriar Derhami

Ali Khalili

Pouya Khast

October 2014

Abstract

The following document is the peer review of the second work shop of the course object-oriented analysis and design using UML, presented in Linnaeus University. The work shop has been done by Jakob Wångö and his other teammates. During this document we will try to write down the strong points and weaknesses of the code and diagrams using “*Peer Review Instruction Workshop 2 Design*” document that our contact person has received.

Table of Contents

Introduction

This assignment was about implementing a simple CRUD system for yacht club and the goal of the assignment was to come up with a well-documented design which follows GRASP and of course there were some requirements that needed to be met.

In this document we are going to first have a general review of the both code and diagrams and at the end we will try to answer the main four questions that we have been asked to answer in the “*Peer Review Instruction Workshop 2 Design*” document which our contact person has received.

Aim

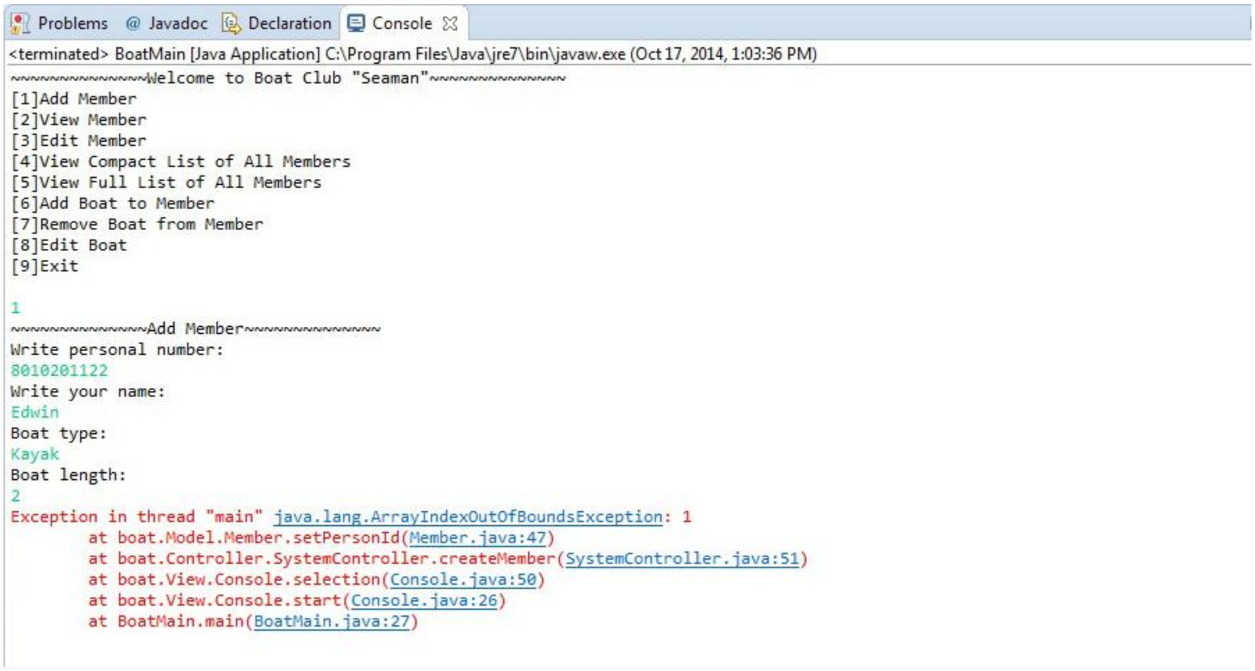
As we mentioned earlier the aim of this document is to do a peer review on the project that we have received form Jakob Wångö and his teammate/s based on the “*Peer Review Instruction Workshop 2 Design*”. First in the General section we are going to talk about the strong points and weaknesses/missing part of the project and then later at the end of the report we are going to answer the main four questions that we have been asked to answer about this project.

General Review

To begin with, we did not receive either an executable version of code or any instruction for compiling of the code. We had some difficulties for compiling of the code since there was a “jdom-2” file which needed to be imported again as an external jar file again in the project.

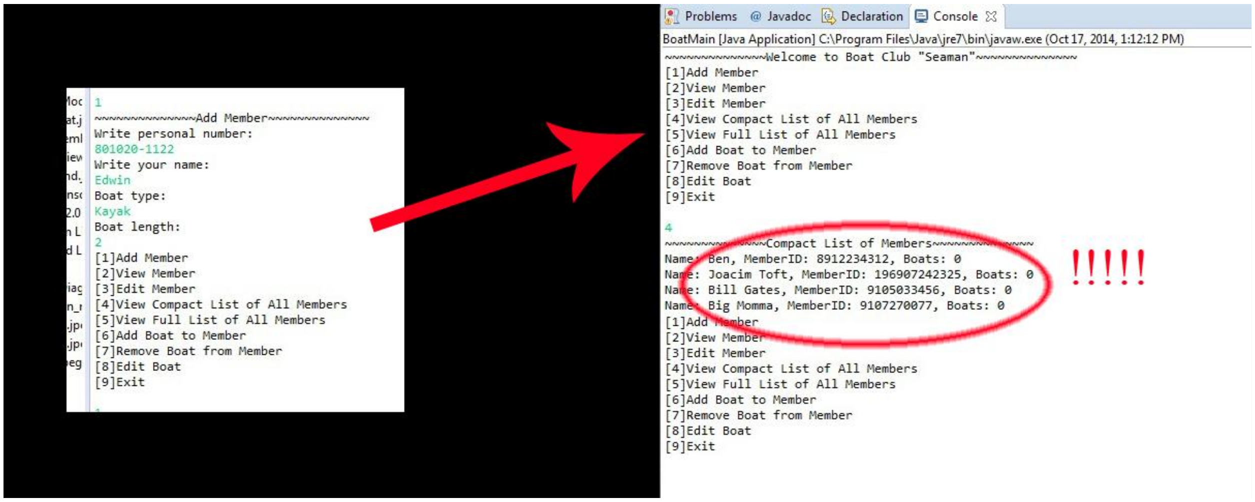
From where we stand, the code had lot of problems which are mentioned below:

1- As it’s shown in the screen shot below, the application has no instruction of how the type of inputs should be. Also instead of having error after the first wrong input it gives error after the whole adding operation was done.



```
<terminated> BoatMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Oct 17, 2014, 1:03:36 PM)
~~~~~~~~~~~~~~Welcome to Boat Club "Seaman"~~~~~~~~~~~~~~
[1]Add Member
[2]View Member
[3]Edit Member
[4]View Compact List of All Members
[5]View Full List of All Members
[6]Add Boat to Member
[7]Remove Boat from Member
[8]Edit Boat
[9]Exit

1
~~~~~~~~~~~~~~Add Member~~~~~~~~~~~~~~
Write personal number:
8010201122
Write your name:
Edwin
Boat type:
Kayak
Boat length:
2
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 1
        at boat.Model.Member.setPersonId(Member.java:47)
        at boat.Controller.SystemController.createMember(SystemController.java:51)
        at boat.View.Console.selection(Console.java:50)
        at boat.View.Console.start(Console.java:26)
        at BoatMain.main(BoatMain.java:27)
```

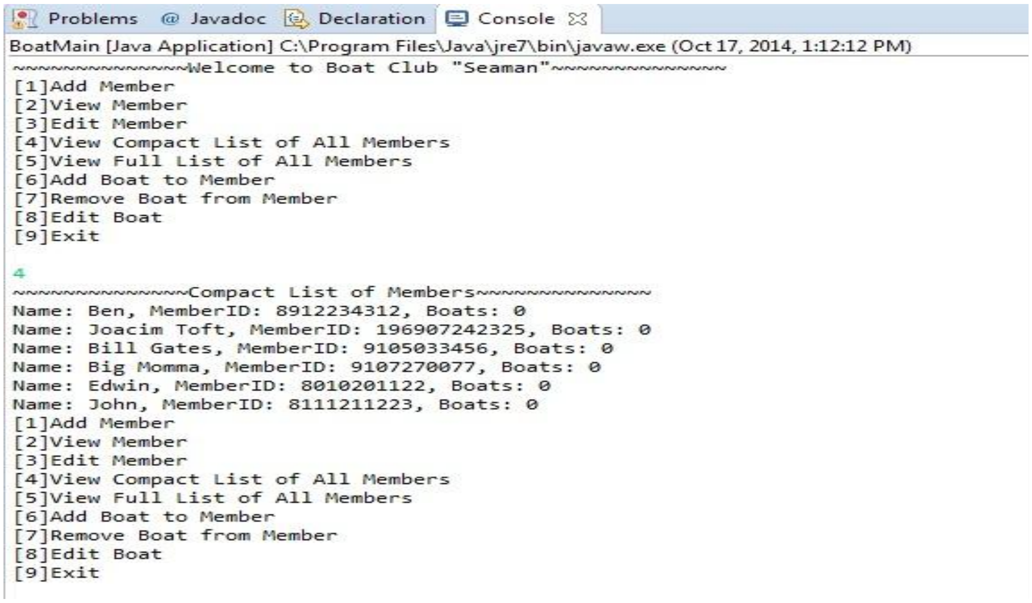
2- We did a simple adding of two different members in the code and then used the function number ‘4’ to view the compact list of all members and instead of seeing the name of members we saw something totally different which apparently been added before in the list.



```
1
~~~~~~~~~~~~~~Add Member~~~~~~~~~~~~~~
Write personal number:
801020-1122
Write your name:
Edwin
Boat type:
Kayak
Boat length:
2
[1]Add Member
[2]View Member
[3]Edit Member
[4]View Compact List of All Members
[5]View Full List of All Members
[6]Add Boat to Member
[7]Remove Boat from Member
[8]Edit Boat
[9]Exit
```

```
BoatMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Oct 17, 2014, 1:12:12 PM)
~~~~~~~~~~~~~~Welcome to Boat Club "Seaman"~~~~~~~~~~~~~~
[1]Add Member
[2]View Member
[3]Edit Member
[4]View Compact List of All Members
[5]View Full List of All Members
[6]Add Boat to Member
[7]Remove Boat from Member
[8]Edit Boat
[9]Exit

4
~~~~~~~~~~~~~~Compact List of Members~~~~~~~~~~~~~~
Name: Ben, MemberID: 8912234312, Boats: 0
Name: Joacim Toft, MemberID: 196907242325, Boats: 0
Name: Bill Gates, MemberID: 9105033456, Boats: 0
Name: Big Momma, MemberID: 9107270077, Boats: 0
[1]Add Member
[2]View Member
[3]Edit Member
[4]View Compact List of All Members
[5]View Full List of All Members
[6]Add Boat to Member
[7]Remove Boat from Member
[8]Edit Boat
[9]Exit
```

We stopped the program and ran it again and then tried to view members again and here is what we saw



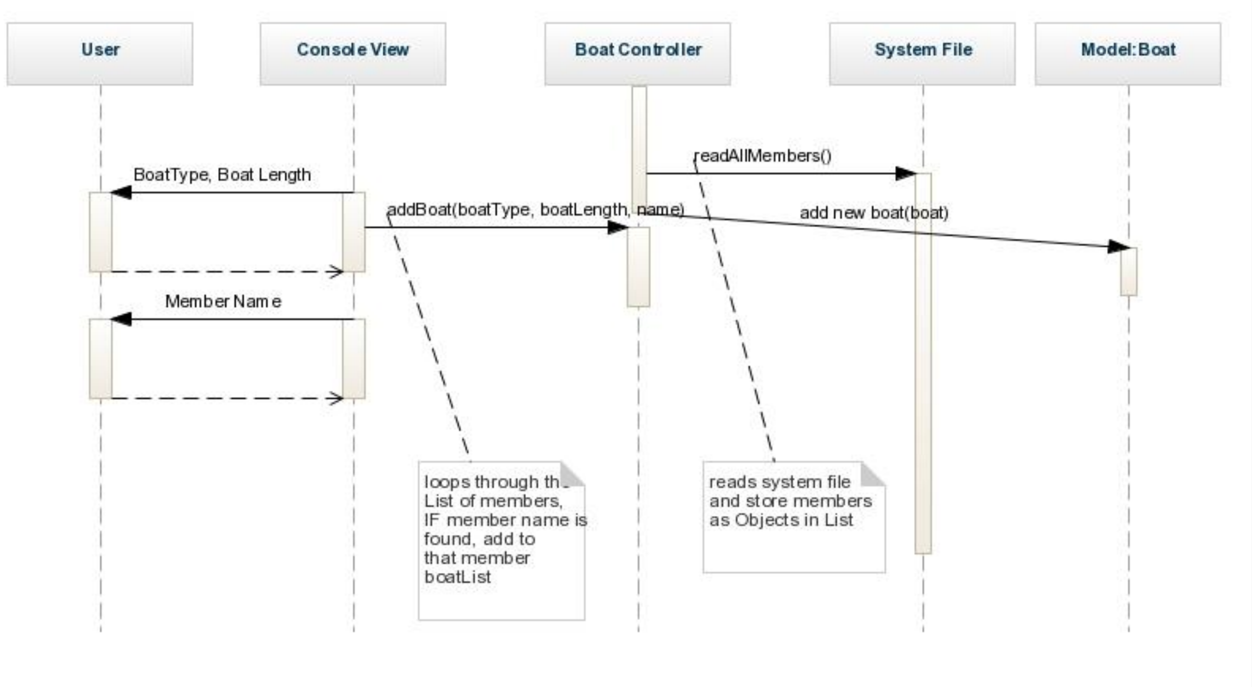
```
BoatMain [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Oct 17, 2014, 1:12:12 PM)
~~~~~~~~~~~~~~Welcome to Boat Club "Seaman"~~~~~~~~~~~~~~
[1]Add Member
[2]View Member
[3]Edit Member
[4]View Compact List of All Members
[5]View Full List of All Members
[6]Add Boat to Member
[7]Remove Boat from Member
[8]Edit Boat
[9]Exit

4
~~~~~~~~~~~~~~Compact List of Members~~~~~~~~~~~~~~
Name: Ben, MemberID: 8912234312, Boats: 0
Name: Joacim Toft, MemberID: 196907242325, Boats: 0
Name: Bill Gates, MemberID: 9105033456, Boats: 0
Name: Big Momma, MemberID: 9107270077, Boats: 0
Name: Edwin, MemberID: 8010201122, Boats: 0
Name: John, MemberID: 8111211223, Boats: 0
[1]Add Member
[2]View Member
[3]Edit Member
[4]View Compact List of All Members
[5]View Full List of All Members
[6]Add Boat to Member
[7]Remove Boat from Member
[8]Edit Boat
[9]Exit
```

3- Very poor exception handling, the code crashes in many cases.
4- The boats will not be added in the database as you can see even in the above screenshots.
5- The functions 3, 6, 7, 8 don’t even work and it’s not really clear what is the purpose of function 9.
6- There is no persistence function.
7- There is no Verbose List function.

About the class diagram we also have some concerns that are mentioned below:

1- We think it could be more appropriate if the classes were put in packages.
2- Wrong format of writing the variables and their types.
3- Wrong format of writing the methods.
4- Member controller class is missing in the class diagram

About the sequence diagrams also we have so points which are worth mentioning for that we are going to use below sequence diagram as an example:



1- Wrong way of naming except for the “Model:Boat”.
2- There is no class boat in the package “boat controller” (which package boat controller is referring to?)
3- For the second action “Member Name” use is only in touch with “Console view” which in fact should be in touch with “Member Controller”.
4- It is not really clear which operation these sequence diagrams are referring to.

Question 1

No, unfortunately as developers these models don’t really help us since the class diagram has lot of missing functions/methods and attributes and it’s really clear that which operations the sequence diagrams are referring to and in general the model (design) and the code don’t conform.

Question 2

We believe that the only strong point of this design could be its primary architecture design (packaging, classing) which prevent complexity.

Question 3

We already answered this question in General Review section.

Question 4

According to what we previously mentioned in General Review section, unfortunately we don’t believe that this design/implementation has passed the grade 3 criteria.