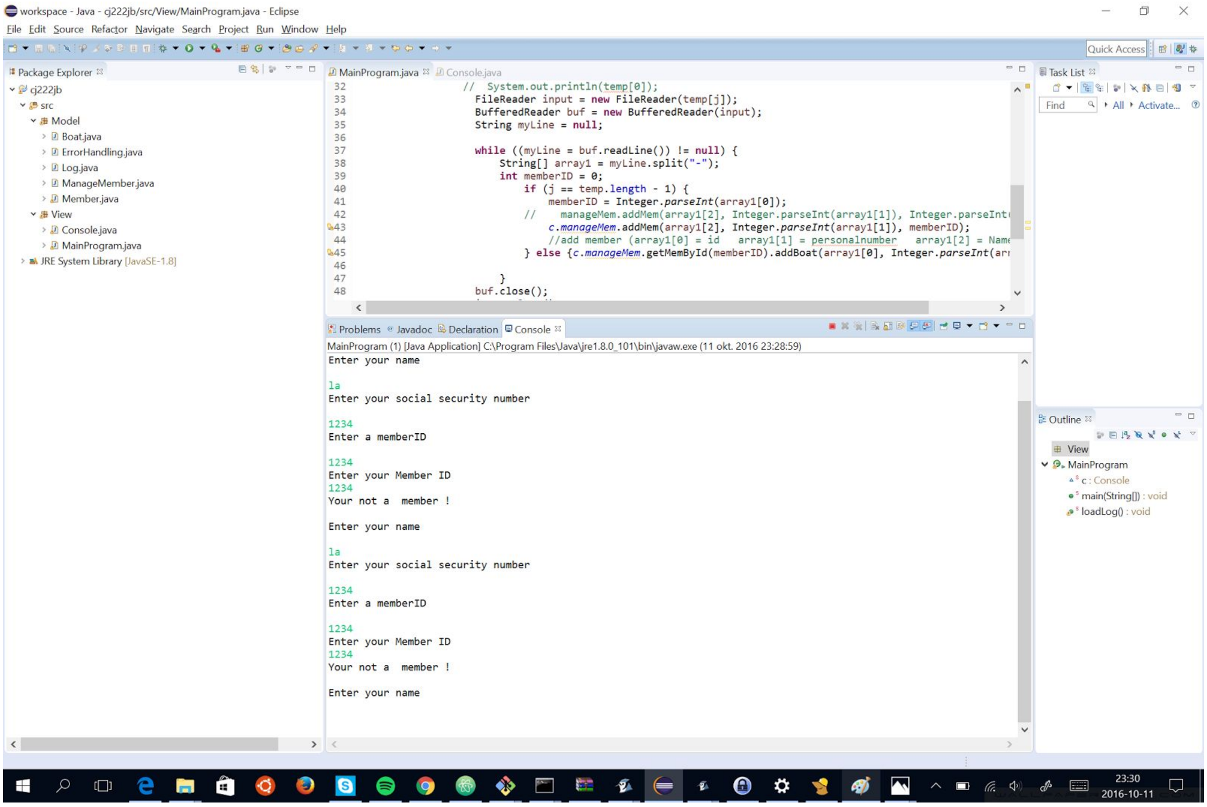
Can you get it up and running? Is anything problematic? Are there steps missing or assumptions made?

There are no instructions on how to run the application. No exe/jar file. I had to figure out how to run it. After the code is compiled and I run the main method, I can’t get passed this screen (see picture). No instructions about any info that I should enter, so there will be no feedback on how the applications runs. Also no diagrams are provided so can’t give feedback on that.



```
            //  System.out.println(temp[0]);
                FileReader input = new FileReader(temp[j]);
                BufferedReader buf = new BufferedReader(input);
                String myLine = null;

                while ((myLine = buf.readLine()) != null) {
                    String[] array1 = myLine.split("-");
                    int memberID = 0;
                    if (j == temp.length - 1) {
                        memberID = Integer.parseInt(array1[0]);
                    //    manageMem.addMem(array1[2], Integer.parseInt(array1[1]), Integer.parseInt(
                        c.manageMem.addMem(array1[2], Integer.parseInt(array1[1]), memberID);
                        //add member (array1[0] = id   array1[1] = personalnumber   array1[2] = Name
                    } else {c.manageMem.getMemById(memberID).addBoat(array1[0], Integer.parseInt(arr

                }
                buf.close();
```

MainProgram (1) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (11 okt. 2016 23:28:59)

```
Enter your name

la
Enter your social security number

1234
Enter a memberID

1234
Enter your Member ID
1234
Your not a  member !

Enter your name

la
Enter your social security number

1234
Enter a memberID

1234
Enter your Member ID
1234
Your not a  member !

Enter your name
```

Quality of source code

is there a model view separation?
No there is not a model view separation. The model has dependencies from the view. According to Larman on page 329, model objects should not have direct knowledge about the view.

Code standards
Good naming for variables and methods. Could not spot any dead code or duplications.

What is the quality of the design? Is it Object Oriented?

Objects are connected using associations and not with keys/ids.
Objects are connected by using associations but the model should not be associated with the view. In the member class it has a view component.

Is GRASP used correctly?
No, GRASP is not used correctly. Classes have low cohesion, high coupling, static variables, not encapsulated.

Classes have high cohesion and are not too large or have too much responsibility.
Several classes suffers from low cohesion, meaning that they are doing more than they are supposed to do. Example in the Member and ManageMember class. They print out details in the console, model should not be responsible for that, that should be done by the view. Larman states on page 461 a class with low cohesion does many unrelated things or to much work.

Classes have low coupling and are not too connected to other entities.
No. The model is bound to the view so it's not possible to change the view without changing the model. Larman states on page 444 a class with high coupling relies on many other classes. This can make the class harder to reuse.

Avoid the use of static variables or operations as well as global variables.
Static variables are used in some classes.

Avoid hidden dependencies.
It's very hard to tell since there is no model view separation.

Information should be encapsulated.
Information is not encapsulated. Fields are not declared as private. Some fields are public, some are left out which mean they get the default modifier. Accessible in class, and packages. The general rule is that that a field should be private and methods should be public. In other words the code is not protected so it can be accessible by other classes.

Larman states on page 918 that public are applied to some methods but not attributes since it violate encapsulation.

As a developer would the diagrams help you and why/why not?
I don't know. No diagrams were provided.

What are the strong points of the design/implementation, what do you think is really good and why? The good parts are naming of fields, methods and classes. Easy to understand what they they do. Model view design pattern is implemented (but did not fulfill all the rules.)

What are the weaknesses of the design/implementation, what do you think should be changed and why? The main weakness is the lack of model view separation. The model should not be aware of the view. The GRASP pattern is not implemented. Example for breaking the GRASP pattern, low cohesion, high coupling, use of static variables. The MainProgram class should not be in the view package, it's not part of the view so it's just confusing.

Do you think the design/implementation has passed the grade 2 criteria?
From what I can see the implementation has not passed the grade for criteria 2. It's not possible to run the application. No diagrams are provided with the solution. Model view separation is not achieved. Does not implement the guidelines for GRASP. I also spotted in the code that the user has to enter a member id, according to the instructions the system should generate a unique id for the user.

Reference Section

1. Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 9780131489066 (https://aanimesh.files.wordpress.com/2013/09/applying-uml-and-patterns-3rd.pdf)