

# Code Review 2

## Team Members:

- Alexander Greff (greffal1) [alex.greff@mail.utoronto.ca](mailto:alex.greff@mail.utoronto.ca)
- Saad Syed Ali (alisaad2) [saadsyed.ali@mail.utoronto.ca](mailto:saadsyed.ali@mail.utoronto.ca)
- Mohammed Osumah (osumahmo) [mohammed.osumah@mail.utoronto.ca](mailto:mohammed.osumah@mail.utoronto.ca)
- Gyeongwon Choi (choigyeo) [gyeongwon.choi@mail.utoronto.ca](mailto:gyeongwon.choi@mail.utoronto.ca)

## Table of Contents

- [Strategy](#)
- [Review Distribution](#)
- [Review Summary](#)
  - [Alexander Greff](#)
  - [Mohammed Osumah](#)
  - [Gyeongwon Choi](#)
  - [Saad Syed Ali](#)

## Strategy

- Look for how well SOLID design principals and the various design patterns are implemented. Is there anything that could be changed that would make it have a better design?
- Check to see that the code conforms to SRP principals.
- Look for how well documented the code is. Is the internal code commented well? Are there docstrings? Are the variables and methods named appropriately?
- Check to see if the code matched how they were presented in the tasks (verification).
- Check to see if the code helps meet the expectations of the clients (validation)
- How well make are the corresponding unit/integration tests for the code? Do they cover all edge-cases adequately?

## Review Distribution

Group Member	System	Package
Saad	Report System	com.team5.report
Won	Database Connection System	com.team5.database
Alex	GUI Interface	com.team5.gui
Mo	Parsing System	com.team5.parser

## Review Summary

- Alex:

### **GUIManager.java**

What was done well:

- Code is easy to read and is well documented.
- Follows SRP principals very well.

What to improve on:

- Hard-coded constants in the file (ADMIN, ICARE, REPORT). Put them in the configuration file and load them.
- Hard-coded values in method calls (line 69 with load("Admin")).
- **public void load()** has a misleading name. Should change to loadDefault()

### **ICarePanel.java, ReportPanel.java**

What was done well:

- Methods and variable names are named appropriately.

What to improve on:

- There needs to be more internal comments, especially in the constructor since there is a lot of code there whose functionality is not instantly apparent.
- Docstrings need to be added.

### **SuperPanel.java**

What was done well:

- Methods and variable names are named appropriately.
- Internal comments were adequate.

What to improve on:

- The name of the class is very vague. It should be renamed to something better.
- Docstrings need to be added.

### **EventHandler.java:**

What was done well:

- The code is pretty self explanatory and the internal comments explain any unclear parts.

What to improve on:

- SRP is violated and there are many if statements differentiating states.
  - To fix, a composition/decorator design pattern should be implemented.
- Hard-coded constants in the file. Put them in the configuration file and load them.
- Docstrings need to be added.

- Mo:

#### **TemplateParser.java**

What was done well:

- Code is documented well.
- Variables are named appropriately.
- Follows SRP well.

What to improve on:

- Name of variable inputobject should be inputObject to conform to google style guide (small fix).
- More commenting should be added to the code.
- Add @SupressWarnings annotation to suppress the warnings in the methods just to keep code error and warning free.
- Refactor demo code (Some of the file paths are incorrect).

#### **MissingFieldException.java**

What was done well:

- Code is easy to understand.
- Method names are consistent.

What to improve on:

- Remove all the unused imports because it causes the java compiler to mark the file as having warnings.
- Either add a default or generated serial version ID to remove warnings or add a @SuppressWarnings annotation.
- Add javadocs to the code.

- Won:

#### **MongoDriver.java**

What was done well:

- All variables and methods are named properly and have good descriptive names.
- The code follows SRP principles all methods have one purpose.
- Very well documented and very easy to understand.

What to improve on:

- A few typos in the comments.
- Some unnecessary comments describing very trivial things.

#### **DatabaseDriver.java**

What was done well:

- All variables and methods are named properly and have good descriptive names.
- The code follows SRP principles all methods have one purpose.
- Well documented.

What to improve on:

- Missing some docstring parameters in some of the functions.

- Saad:

#### **Overall:**

What was done well:

- Inheritance is a good structure to use for charts.
- Well commented.
- Responsibilities are grouped effectively, not only by package but by individual classes. Each chart has its own type and generator.
- Comments are thorough.

What to improve on:

- Could create some superclasses that define functions that are commonly repeated in children to reduce duplicated code.
- Could use some cleanup to remove unnecessarily commented code.