

# SE 3313 Software Design and Architecture

## MVC Architecture Pattern

### Lab 11

## 1 Context

The MVC architecture pattern turns complex application development into a more manageable process. It allows several developers to work on the application simultaneously. MVC stands for model-view-controller. Here's what each of those components means:

- Model: The backend that contains all the data logic
- View: The frontend or graphical user interface (GUI)
- Controller: The brains of the application that controls how data is displayed

## 2 Problem Description

In this lab exercise, you will design and implement a simple To-Do List application using the Model-View-Controller (MVC) architectural pattern. The application should allow users to add tasks, view their list of tasks, and mark tasks as completed. You will need to organize the code following the MVC pattern to separate concerns effectively.

- Model Implementation
  - Create a Java class named Task to represent individual tasks.
  - Implement a Java class named TaskModel to handle data and business logic for tasks.
- View Implementation
  - Develop a Java class named TaskView responsible for displaying tasks to the user.
  - Create a simple text-based user interface for listing tasks.
- Controller Implementation
  - Build a Java class named TaskController to manage user interactions.
  - Implement methods for adding tasks, viewing tasks, and marking tasks as completed.

Ensure that the controller handles user input and updates the model and view accordingly.

Design a menu and test the your application after you completed: Add tasks with descriptions. View the list of tasks. Mark tasks as completed.

## 3 Measure of Success

Implementation of the application should add tasks, view tasks, mark tasks as completed, and exit the application. The tasks should be displayed with their completion status, and the application should follow the MVC pattern to separate concerns.

## 4 Sample Output

```
Options:
1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Exit
Enter your choice: 1
Enter task description: Complete lab exercise
```

```
Options:
1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Exit
Enter your choice: 1
Enter task description: Submit assignment
```

```
Options:
1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Exit
Enter your choice: 2
To-Do List:
1. [Waiting] Complete lab exercise
2. [Waiting] Submit assignment
```

```
Options:
1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Exit
Enter your choice: 3
Enter task index to mark as completed: 1
```

```
Options:
1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Exit
Enter your choice: 2
To-Do List:
1. [Done] Complete lab exercise
2. [Waiting] Submit assignment
```

```
Options:
1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Exit
Enter your choice: 4
Exiting...
```

Figure 1: