



# CIS 404 JAVA PROJECT

---

Online Medical Management System by Brian Lee

# Agenda

---

- 
- Project Outline 3
  - Design Algorithm 5
  - SDLC Implementation 6
  - Software Development Life Cycle 7
  - JIRA 8



## Project Outline

### 1. Import necessary classes

- Import java.util.ArrayList;
- Import java.util.List;
- Import java.util.Scanner;

### 2. Define the Doctor class:

- Private fields: 'name' and 'specialization'
- Constructor: initializes 'name' and 'getSpecialization'

### 3. Define Patient class:

- Private fields: 'name', 'age', and 'assignedDoctor'
- Constructor: initializes 'name' and 'age'
- Getter and setter methods: 'getName()', 'getAge()', 'getAssignedDoctor()', and 'setAssignedDoctor()'



#### **4. Define the `MedicalManagementSystem` class:**

- Private fields: 'doctor' (lists of doctors) and 'patients' (list of patients)
- Constructor: initializes the 'doctors' and 'patients' lists
- Methods:
  - `'addDoctor()'`: Adds a doctor to the system
  - `'addPatient()'`: Adds a patient to the system
  - `'displayDoctors()'`: Displays the list of doctors with their name and specialization
  - `'displayPatients()'`: Displays the list of patients with their name, age, and assigned doctor

#### **5. Define the `OnlineMedicalManagementSystem` class (main class):**

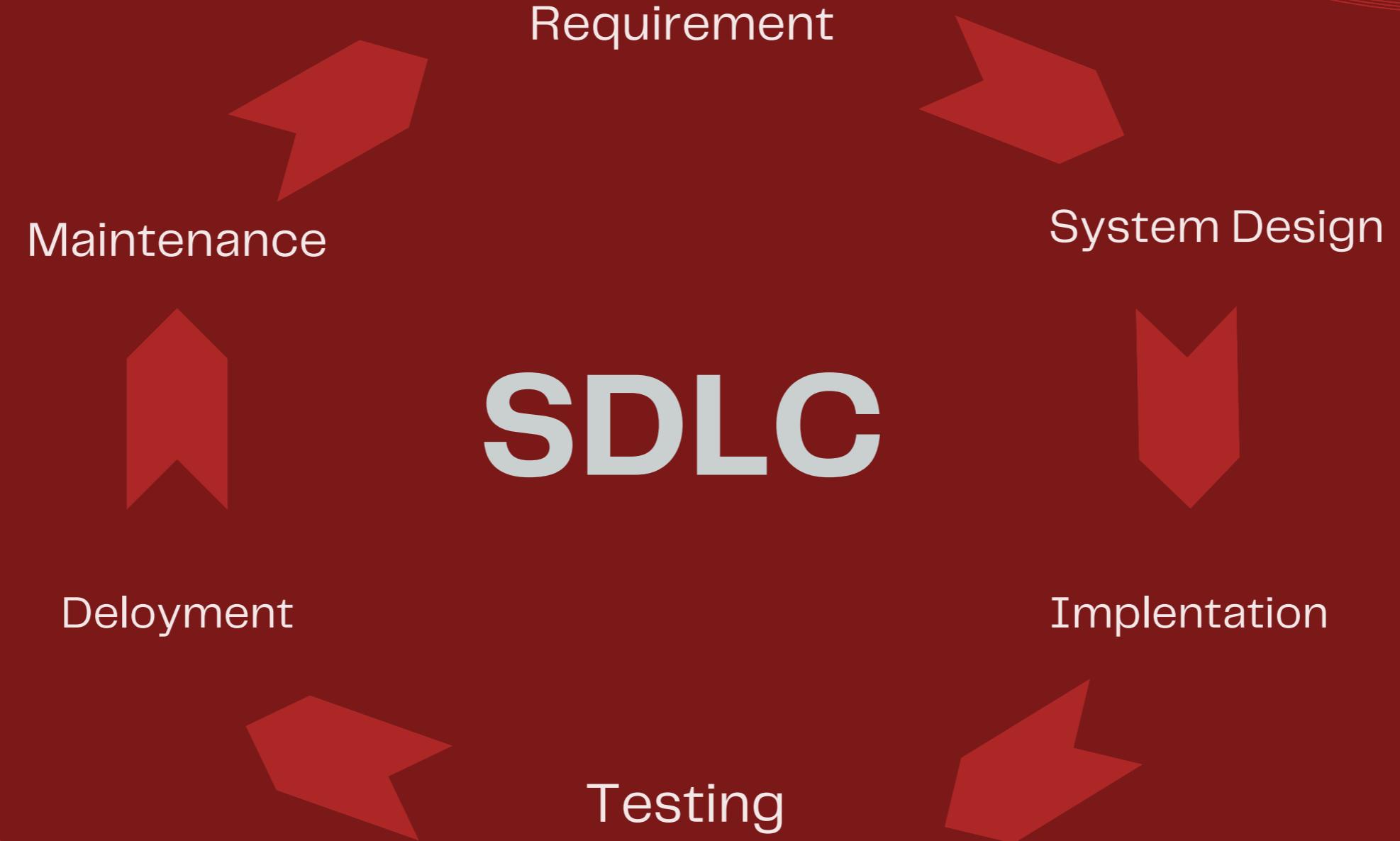
- Create an instance of `MedicalManagementSystem`
- Create doctors (`Doctor` objects) and patients (`Patient` objects)
- Add doctors and patients to the system using the appropriate methods
- Assign doctors to patients
- Use a `Scanner` to provide a command-line menu for the user:
  - Display doctors
  - Display patients
  - Exit the program



# Design Algorithm

```
Start
Create MedicalManagementSystem object
Call constructor
Create Doctor objects
Create Doctor object (Dr.Kim)
Create Doctor object (Dr.Blue)
Add doctors to the system
Create Patient objects
Create Patient object (Michael and Brian)
Assign doctors to patients
Display menu and get user's choice
If choice is 1 → Call displayDoctors() method
If choice is 2 → Call displayPatients() method
If choice is 0 → Exit the program
Repeat the loop until choice is 0
End
```

# SDLC



# JIRA

The image shows a Jira Kanban board with three columns: TO DO 2 ISSUES, IN PROGRESS 2 ISSUES, and DONE 3 ISSUES.

**TO DO 2 ISSUES**

- repeat loop
  - KAN-7
  - KAN-8

**IN PROGRESS 2 ISSUES**

- defining classes
  - KAN-5
- display menu
  - KAN-6

**DONE 3 ISSUES**

- Import
  - KAN-2
- create classes
  - KAN-1
- use scanner
  - KAN-4

+ Create issue