



Object Oriented Programming

BSSE – 2nd Semester

Fall 2024

LAB PROJECT

Submitted To :

Urwa Farooq

Submitted By :

M.Ahazz, Hassan

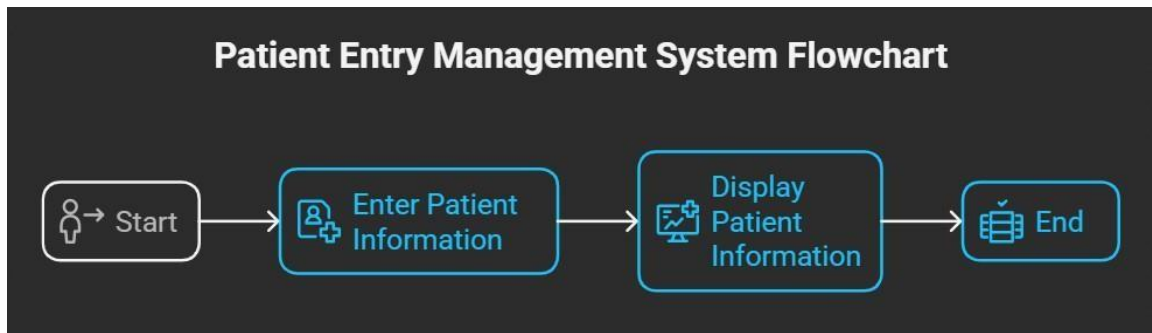
Ehtisham, Mirza Rehan

BS SOFTWARE ENGINEERING PROGRAM DEPARTMENT
OF COMPUTER SCIENCE HITECH UNIVERSITY TAXILA

Patient Entry Management System:

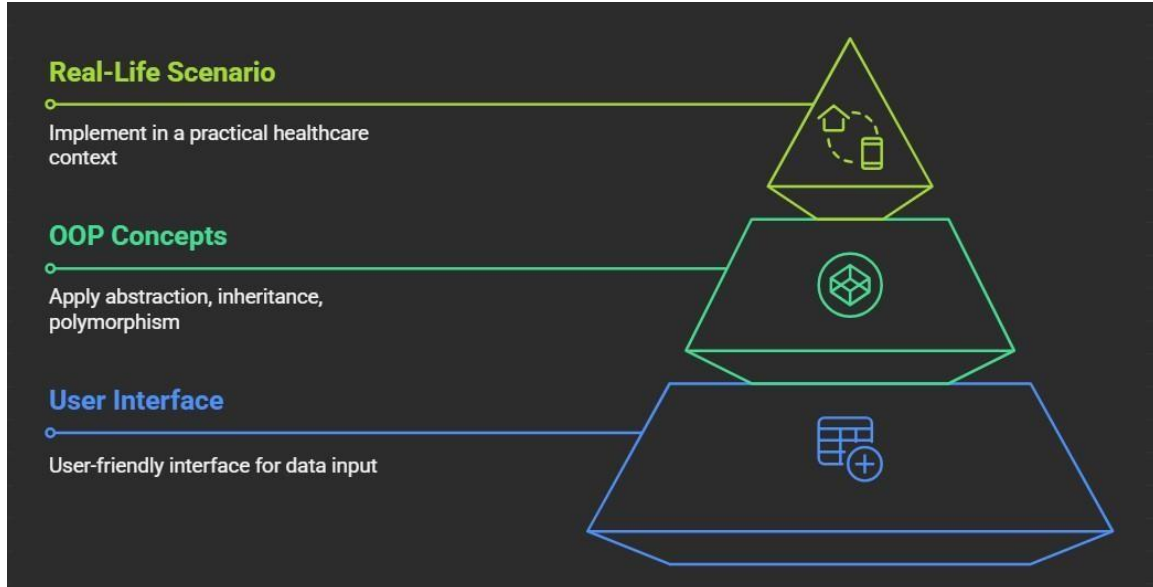
1. Introduction

This project is a simple Java application named 'Patient Entry Management System'. It allows users to enter and display patient information such as name, age, and patient type (Regular or Emergency) using a Graphical User Interface (GUI) built with Java Swing.



2. Objective

The main goal of this project is to provide a user-friendly interface to input patient data and practice object-oriented programming (OOP) concepts such as abstraction, inheritance, and polymorphism in a real-life scenario.



3. Tools and Technologies Used

- Java Programming Language
- Java Swing for GUI

- Object-Oriented Programming Concepts
- Arrays and Strings

4. Program Structure

4.1 Abstract Class (Abstraction) :

The abstract class `Person` includes common patient properties like name and age. It also defines an abstract method `getType()` and a concrete method `getBasicInfo()` to return the basic details of the patient.

4.2 Inheritance and Polymorphism:

Two classes, `RegularPatient` and `EmergencyPatient`, inherit from `Person`. These classes override the `getType()` method to return their specific type. This demonstrates the use of inheritance and polymorphism in Java.

4.3 GUI Components:

We used Java Swing components to build the form:

- `JLabel` and `JTextField` for Name and Age input
- `JComboBox` with a String array for patient type selection
- `JButton` to submit the form and display a message box with patient details
- Colors and layout to improve the visual appearance

5. Use of Arrays :

We used a simple String array `String[] types = {"Regular", "Emergency"};` to store patient type options for the dropdown (JComboBox).

6. Exception:

Save our system from crash which occur due to user not fully professional and known to code.

Try and Catch used in complete code:

1. **Parsing Age Input (Number Format Issue)**
2. **Login Validation**
3. **File Writing**

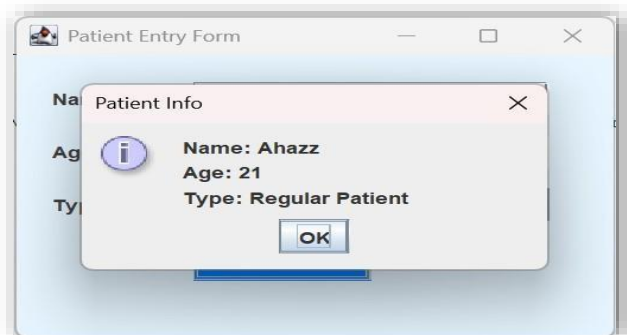


7. FILE HANDLING:

To save patient data in a file for further use we have use file handling (file writer is used).

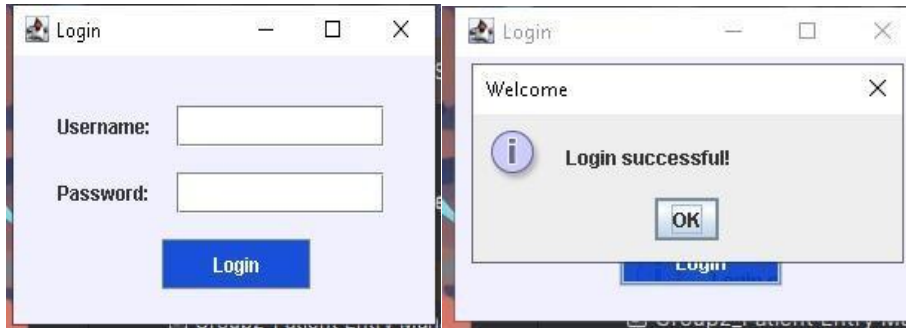
8. Key Features and Output

When the user fills the form and clicks the Submit button, the program creates an object of either RegularPatient or EmergencyPatient. Then it displays the entered details in a message dialog using the methods defined in the classes.



9. Login Page:

We have made a login page so before entering the system first user login in to system then enter patient data.



- 7. Lessons Learned**
1. Learned how to use basic OOP concepts in a real project.
 2. Gained hands-on experience with Java Swing for GUI design.
 3. Improved teamwork and problem-solving while debugging and designing the form.



Conclusion:

Through this project, we were able to learn how to construct a simple Java graphical user interface application and practically apply key programming concepts.

GUI Application Development

Conceptual Understanding

Theoretical programming knowledge base

Design GUI

Plan the user interface

Implement Application

Write code for functionality

Functional Application

Practical application of concepts

