

Assignment 3 – A&E Department

Project Report

DT8265

Higher Diploma in Computing

**Bob Bobga Nti**

School of Computing

Dublin Institute of Technology

**10/12/2017**



Declaration

I hereby declare that the work described in this assignment is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

Bobganti

Bob Bobga Nti

10/12/2017

Contents

[1. Introduction 4](#_Toc501395145)

[2. Implementation 5](#_Toc501395146)

[3. How the application has followed the SOLID Principles 10](#_Toc501395147)

[4. Conclusion 10](#_Toc501395148)

# Introduction

* The **A&E** Application is a Desktop App developed to manage the A&E Department of any Hospital.
* The System has been developed following the **SOLID** Design Principles
  1. **Single Responsibility Principle (SRP)**

This states that a class should have one and only reason to change. This is further explained by the SoC (Separation of Concerns). A class should have only on task to perform.

* 1. **Open-Closed Principle (OCP)**

This states that Objects are open for extension, but closed for modification. This is closely related to the SRP in that when there is separation of concerns, the functionality of the program can be extended with minimum changes in the existing code. When the need for adding a new functionality arises, a new class is simply created to take care of that.

* 1. **Liskov Substitution Principle (LSP)**

This states that Subclasses should be substitutable for their base classes.

* 1. **Interface Segregation Principle (ISP)**

Instead of Inheriting one large Interface containing Methods not needed, split the “fat” Abstract class into groups of responsibility. In this way, a class can choose which Methods to sign-up to and drop the contracts it doesn't need.

* 1. **Dependency Inversion Principle (DIP)**

This states that an Object should not control the creation of its dependency. It should just advertise its needs and gets provided by its caller. This is described as Objects using Objects.

The A&E comprises 3 work sections: the Reception, Nurse, Doctor; completely independent of each other.

All Transaction is persisted in an SQLite Database called **DLList**.

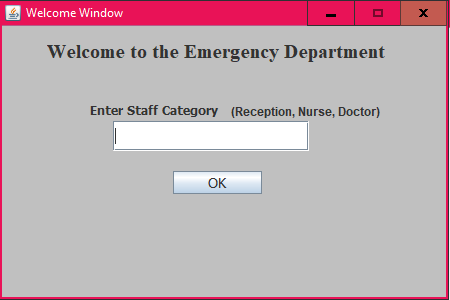
The DLList contains the “**Patients**” table for storing all the patients who visit the A&E. When the Doctor finishes with the patient, the patient is removed from the “Patients” table. But since the A&E needs to keep record of all visits, the history of the patient is then recorded in another table, “**Treated\_Patients**”

All three sections of the A&E have access to the DLList for resources.

# Implementation

* GUIs have been provided, using Swing, for the following classes:
  + WelcomeFra me
  + ReceptionistPanel
  + NursePanel2
  + Doctor
* In addition to the above classes, the system also include the following classes:
  + DoublyLinkedList
  + Node
  + DataBaseConnection
* The system implements an Interface (ISaver)
* **Validation:**
* All numeric fields (Age, Priority) accept only numbers. The System makes an Error sound when keys other than numeric are typed.
* The Priority number is checked for range between 1 and 10. Out of this range the System displays an Error message.

When run, the Application begins with the “WelcomeFrame”. This provide the GUI for the user to choose the section he/she has been assigned to work in.

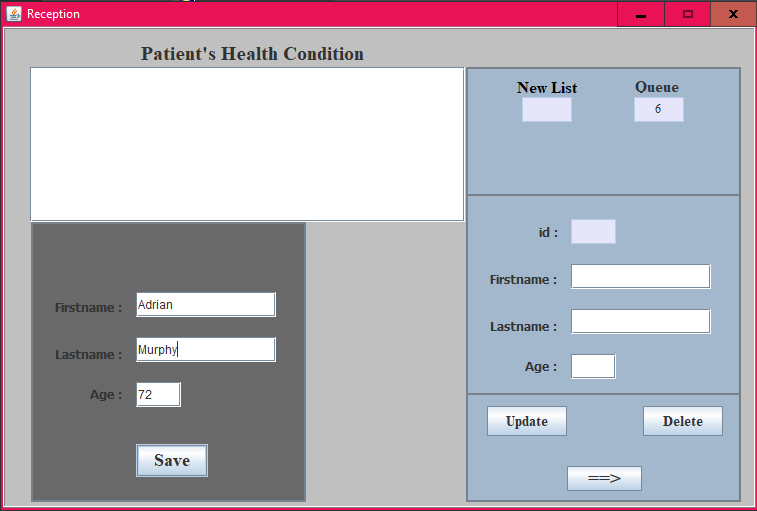


The input box is not case sensitive. The choices are:

* Reception or Receptionist
* Nurse
* Doctor

If the user is a Receptionist, the “ReceptionPanel” is called.

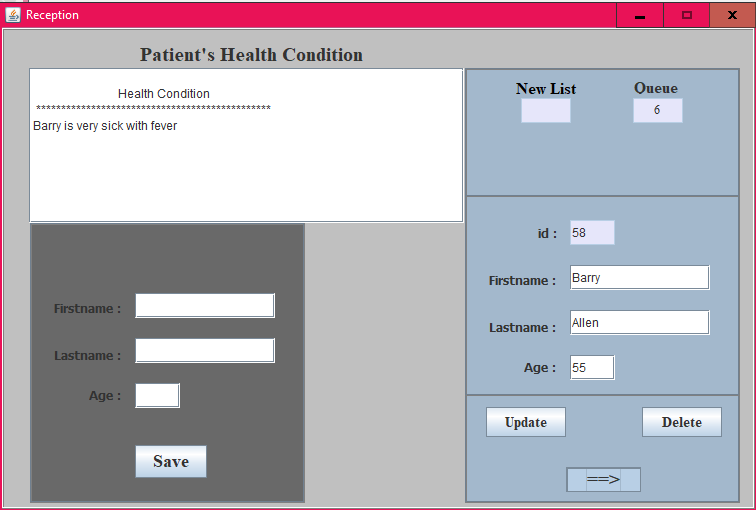
1. The Receptionist can add (Save) patients to the list which is subsequently saved to the database. The registration process involves obtaining Firstname, Lastname and Age from the patient.



Each registration session creates a “New List” of patients. The patient is simultaneously saved to the DLList as they are being added to the “New List”.

The “Queue” is the list of all the registered patients in all sessions (pulled from the DLList In this way, the other operators; Nurse, Doctor, are able to access and process the Queue of patients independently (doesn’t matter whether the Reception is up running or closed).

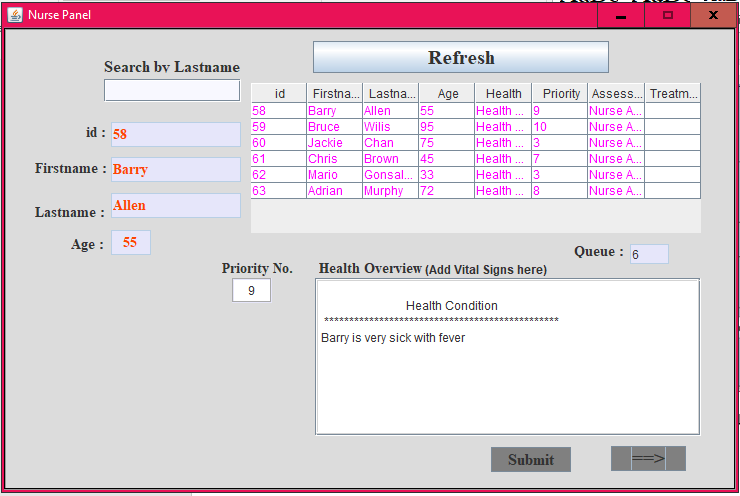
1. The Receptionist can Traverse the Linked list using the “==>” button to navigate. In this way the Receptionist can navigate to a patient for Update or Deletion. When navigating, the current patients details are displayed in the respective text boxes.



Barry Allen’s details can be Updated or Barry completely deleted from the System.

Only the Receptionist can Delete a patient from the System.

If the user is a Nurse, the “NursePanel2” is called.



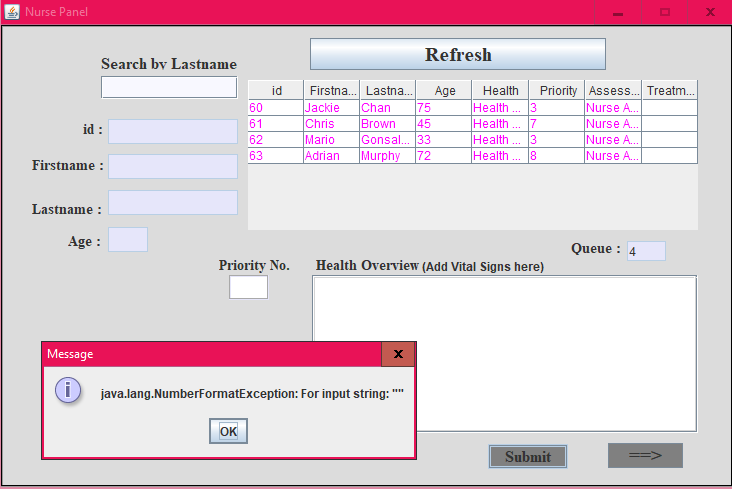
The NursePanel consist of a JTable which is populated with data from the database, at the same time, added to a Linked List.

The Nurse traverses the Linked List using the “==>” button to get the next patient on the list. The next patient is by “First In First Out” (FIFO) algorithm.

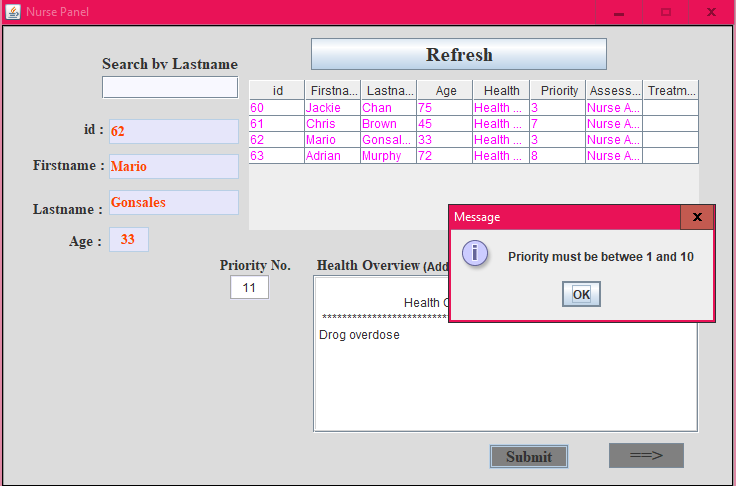
The nurse includes the Vital Signs and the Priority and then clicks on “Submit”.

The additional information is added to the patient’s details.

The System displays Error message if fields are blank:

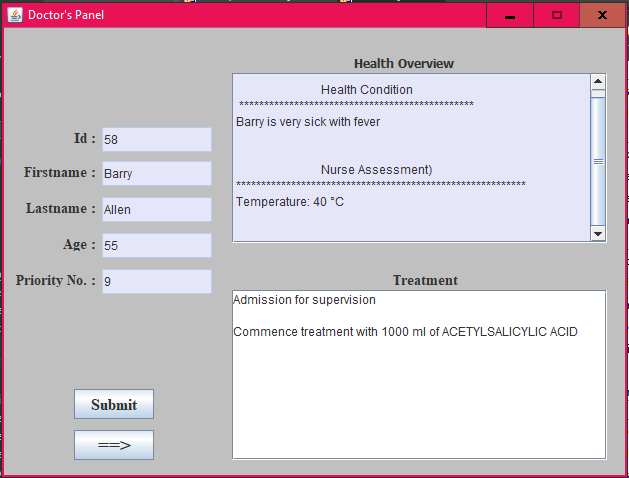


If the nurse enters a Priority number which is out of range, the System displays the Error below:



The nurse can jump to a particular patient by doing a “Search by Lastname”.

If the user is a Doctor, the ”Doctor” is called.



The “Doctor” panel pulls data from the database and adds to a Linked List each time the “==>” button is clicked. The next patient is the one with the Highest priority (10 => 1), 1 being the lowest priority. If a patient shares the same priority number with another, the system implements “FIFO” on the affected patients.

The “Health Overview” and personal details fields are all view only. The Doctor can comment and add treatment in the “Treatment” area only

When the Doctor clicks “Submit”, the Treatment is recorded against that patient the patient is removed from the Linked List, removed from the “Patients” table and stored in the “Treated\_Patients” table, with all the information spanning form Condition, Assessment and type of treatment. In this way, the Hospital is able to keep tract of the patient.

# How the application has followed the SOLID Principles

This Application “implements SOLID design”.

”

* SRP: Each class performs one duty only
  + The Patients class holds details pertaining to a patient
    - First name
    - Last name
    - Age
    - Health Condition
    - Nurse’s assessment
    - Priority number assigned
    - Treatment provided by the Doctor
* OCP: Record information to database and to Linked List is done have been encapsulated to DatabaseConnection and DoublyLinkedList. Access is by objects created from these classes.
* ISP: ISaver provides an Interface Type for the object that calls the Add and Update Methods for both List and database.

# Conclusion

The present Application is limited in scope (A&E Department). But it is designed in a way that functionalities could be added subsequently by including new classes. The Application could extend to include other departments of any Hospital.