

Object Oriented Software Development Project Report

Ayano Yamamoto

19th December 2021

Table of Contents

Introduction	3
Assumptions	3
Feature 1 - Console Menu User Interface	4
1) Admission.....	4
New patient.....	4
Existing patient.....	4
2) Triage	5
3) Treatment	5
4) Search treatment records.....	6
Patient Name / Admission Date / Treatment Doctor	6
5) Update patient information	6
6) Delete patient record	6
Feature 2 – Functionality	7
Required Functionalities	7
General	7
Singly Linked List	7
Database.....	7
Additional Functionalities	7
Feature 3 – Data Persistence.....	8
Patients	8
Treatments	8
Feature 4 – Data Structure and Algorithm (DSA).	9
Singly Linked List.....	9
Priority Values	9
Conclusion	9

Introduction

The system supports the accident and emergency department's operation by storing patients' information and ensuring that the patient with the most serious condition gets seen by the doctor first.

When the patient is admitted to the department, the admission information recorded by the receptionist are stored in a Singly Linked List. The system then informs the triage nurse of the next patient to see in a First-In-First-Out order. The triage nurse records the patient's vital signs and assigns the priority value between 1 to 10, 1 being the least serious and 10 being the most serious conditions. The system subsequently informs the doctor of the next patient to see based on the priority values assigned, so that the patients with the highest priority values are seen first. After the treatment information are recorded, the patient is removed from the Singly Linked List and their record is stored in the database.

Since unexpected errors can occur while interacting with the database, the system tries to facilitate ease of use by:

- Error checking (try/catch)
- Input validation
- Using datetime to save admission_time, triage_time, and treatment_time
- Replacing apostrophes in string inputs before writing to database

Assumptions

- System requires one database
- Login functionality is not required
- PPS numbers are unique identifiers, but the department will also treat patients without PPS numbers (e.g., foreign citizens visiting Ireland)
- All patients will go through the three steps provided by the Chief Medical Officer. Patients will not be discharged after admission or triage steps.
- Treatment records are to be saved in the database after the patients are discharged.
- Treatment records are not to be edited after triage or treatment.

Feature 1 - Console Menu User Interface

The main console menu offers 6 options below.

1) Admission

Admission menu offers 2 options; new patient and existing patient, to avoid creating duplicate patient records. New patient option asks the user to input the patient's personal details, of which the PPS number must be unique to be saved in the database. Existing patient option asks the user to specify a patient from the database by their patient ID.

New patient

1. User inputs the new patient's personal details. PPS number can be empty for patients without one, but other fields must be filled in.
2. System creates a Patient object, inserts its information into the database, and returns the newly created patient ID.
3. If the new patient was inserted successfully, user inputs the admission summary.
4. System creates a Treatment object with the newly created patient ID, datetime of the admission, and the admission summary, and stores it in the Singly Linked List with the default priority_value of 0.

Existing patient

1. User inputs the patient ID of the existing patient.
2. System searches for the matching patient ID in the database. It displays a message if there are no matching patient.
3. System checks if the matching patient ID exists in the Singly Linked List. It displays a message if the patient is already admitted.
4. If the matching patient ID exists in the database, and the patient is not already admitted, user inputs the admission summary.
5. System creates a Treatment object with the patient ID, datetime of the admission, and the admission summary, and stores it in the Singly Linked List with the default priority_value of 0.

2) Triage

Triage option updates the Treatment object in the Singly Linked List with the triage information.

1. System goes through the Singly Linked List and identifies the first patient in with priority_value of 0 as the next patient awaiting triage. It displays a message if there are no patients awaiting triage.
2. System displays the patient's information from admission.
3. User inputs the triage nurse's name, patient's vital signs, triage summary, and a priority value of 1 to 10.
4. System updates the Treatment object with user's input.

3) Treatment

Treatment option updates the Treatment object in the Singly Linked List with the treatment information.

1. System goes through the Singly Linked List and identifies the patient with the highest priority_value as the next patient awaiting treatment. If two patients have the same priority_value, the one with a smaller node position will be identified for treatment first. It displays a message if there are no patients awaiting treatment.
2. System displays the patient's information from admission and triage.
3. User inputs the doctor's name and treatment summary.
4. System updates the Treatment object with user's input and replaces all single apostrophes with double apostrophes to avoid errors when writing to database.
5. System inserts the Treatment object information into the database, and if successful removes the patient from Singly Linked List.

4) Search treatment records

Search menu offers 3 options; to search by patient's name, admission date, and treatment doctor's name.

Patient Name / Admission Date / Treatment Doctor

1. User inputs the patient's name / admission date / doctor's name to search.
2. System searches for the matching records in the database.
3. System displays all treatment records from the search. It displays a message if there are no matching records.

5) Update patient information

Update patient information option updates a Patients table row in the database with user inputs.

1. User inputs patient ID to update.
2. System searches for the matching patient ID in the database. It displays a message if there are no matching patient.
3. If the matching patient ID exists, user inputs the new patient information.
4. System updates the patient information in the database. If the update is successful, the system displays a confirmation.

6) Delete patient record

Delete patient record option asks the user to specify a patient by their patient ID, and deletes corresponding rows from both Patients and Treatments tables in the database.

1. User inputs patient ID of the patient to delete.
2. System searches for the matching patient ID in the database. It displays a message if there are no matching patient.
3. If the matching patient ID exists, the system deletes the patient from Patients table in the database. Their treatment records from Treatments table will also be deleted. If the delete is successful, the system displays a confirmation.

Feature 2 – Functionality

Required Functionalities

General

- A text-based console user interface
- Load the default database file (ae.db)
- Error checking (try/catch)

Singly Linked List

- Create a Singly Linked List to store patient information
- First-In-First-Out queue for triage
- Create and update Treatment objects in Singly Linked List
- Read and display information from Treatment objects in Singly Linked List
- Treatment queue prioritisation based on priority values
- Delete a Treatment object from the Singly Linked List after treatment

Database

- Create new patient records in the database (Patients table) by user input
- Create new treatment records in the database (Treatments table) by inserting information from a Treatment object in Singly Linked List
- SQLite data persistence for patient information and their treatment records

Additional Functionalities

- Display numbers of patients awaiting triage and treatment in the menu title
- Input validations for:
 - numbers (integer or float)
 - positive integers
 - priority values (integers between 1 and 10)
 - strings (to validate if there is an input)
- Replace single apostrophes with double apostrophes before writing to database
- Read and display information from database (Search treatment records by patient's name / admission date / doctor's name)
- Update patient records in the database (Patients table)
- Delete patient and their treatment records from the database (Patients and Treatment table)

Feature 3 – Data Persistence

Patient and treatment records are stored in the database file ae.db so they are persistent after the system is closed. It contains two tables below.

Patients

Table to store patient's personal details. Rows can be created, read, updated, and deleted from functionalities offered in the system. Columns are:

- **patient_id**: primary key, integer, autoincrement, unique, not null
- **firstname**: text, not null
- **lastname**: text, not null
- **dob**: text, not null
- **address**: text, not null
- **phone**: text, not null
- **pspn**: text, unique, this field can be null

Treatments

Table to store treatment details of each admission. Rows can be created and read, but cannot be updated from the system since treatment records are assumed not to be edited after treatments. Deleting a row in Patients table will cascade to this table and delete rows with corresponding patient ID. Columns are:

- **admission_id**: primary key, integer, autoincrement, unique, not null
- **patient_id**: foreign key (references patient_id from Patients table) on delete cascade, integer, not null
- **admission_time**: text, not null
- **admission_summary**: text, not null
- **triage_time**: text, not null
- **triage_nurse**: text, not null
- **body_temp**: numeric, not null
- **pulse_rate**: integer, not null
- **respiration_rate**: integer, not null
- **bp_systolic**: integer, not null
- **bp_diastolic**: integer, not null
- **triage_summary**: text, not null
- **priority_value**: integer, not null
- **treatment_time**: text, not null
- **treatment_doctor**: text, not null
- **treatment_summary**: text, not null

Feature 4 – Data Structure and Algorithm (DSA).

Singly Linked List

Patients who have been admitted into the system are stored in a Singly Linked List so they can efficiently be identified for triage in a First-In-First-Out order.

Priority Values

After the patients are assigned priority values in triage, the system goes through the Singly Linked List to identify a patient with the highest priority value for treatment. The First-In-First-Out order also applies here, so if two patients with the same Priority Value are in the treatment queue, the one who was admitted first will be prioritised.

Conclusion

The system provides support for the receptionists, nurses, and doctors in the accident and emergency department to liaise with each other by storing and sharing information. Once patients are admitted, it uses Singly Linked List to identify the next patient for triage in First-In-First-Out queue, and combines it with assigned priority values to ensure that the patient with the most serious condition is prioritised for treatment. Once the patients are discharged after treatment, their treatment records are saved in SQLite database for data persistence. Information in the database can be searched, updated, and deleted.

To minimise input errors from the text-based console user interface, menu options and inputs have validations and try/catch to guide users.

While researching ways to validate user inputs, I have come across existing modules such as PyInputPlus that can be imported to handle different types of input validations. While I decided to write simple input validation functions in this project, I would like to learn more about input validation libraries and how to use them in the future.