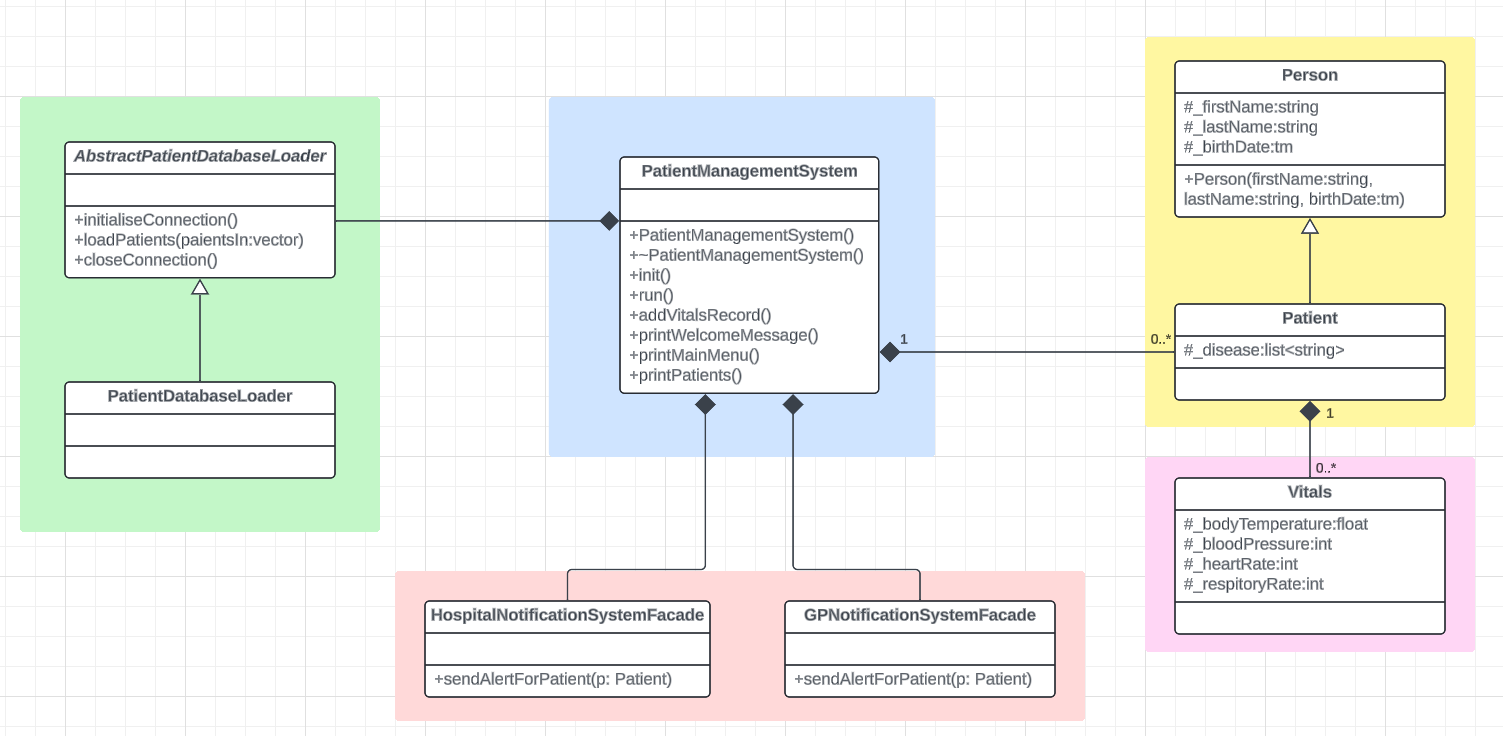
# **Patient Vitals Management System**

## System Design Documentation

1. **System Overview**

The Patient Vitals Management System is designed to manage patient information, vital recording, and notifications for healthcare providers with several core components.

[Figure 1.1 – Initial State]



[The table below only summarizes the above diagram following the partial UML diagram in the Assignment Spec document.]

|  |  |
| --- | --- |
| **Main Component** | **Description** |
| Main Driver | * PatientManagementSystem (A) controls the program. It starts and runs the system, adds vitals, shows menus, and tracks patients. |
| Patient Data Loading | * Patient data is loaded using an interface called AbstractPatientDatabaseLoader (B). * The concrete implementation PatientDatabaseLoader simulates database access with a mocked connection. |
| Patient Model | * The Patient class (C) inherits from the Person class, encapsulating personal information such as first name, last name, and birthday. * Each patient maintains a list of diseases (represented as strings) and can record multiple Vitals instances (D). * Vitals include measurements such as body temperature, blood pressure, heart rate, and respiratory rate. |
| Notification System | * The system provides two facade interfaces, HospitalNotificationSystemFacade and GPNotificationSystemFacade (E). * Designed to send alerts about patient conditions to hospitals and general practitioners, respectively. (Not yet integrated into the main system workflow) |

1. **Mapping Functional Requirements to Design Patterns**

|  |  |  |  |
| --- | --- | --- | --- |
| **Functional Requirements #** | **Description** | **Design Pattern** | **Progress** |
| FR1 | Load patients from a file. | Adapter Pattern |  |
| FR2 | Load patients from both the file and the database. | Composite Pattern |  |
| FR3 | Implement algorithms to identify patient alert levels when new vitals are recorded. | Strategy Pattern |  |
| FR4 | Alert hospitals and GPs when the patient alert level is Red. | Observer Pattern |  |

1. **Design and Implementation Details**