# Design document

FR3: Adapter pattern

PatientManagermentSystem now not only need to load data from database, and also need to try to load patient information through txt file, so I use adapter pattern to try to use PatientFileLoaderAdapter to adapt to the PatientFileLoader. As you can see in Figure 1, not only is PatientDatabaseLoader a subclass of AbstractPatientDatabaseLoader, but PatientFileLoaderAdapter is also a subclass of AbstractPatientDatabaseLoader. It is used to transform the PatientFileLoader and load data from it.

How it works:

1. The constructor of PatientFileLoaderAdapter accepts a parameter file\_loader which is a pointerof PatientFileLoader of type PatientFileLoader. PatientFileLoaderAdapter itself includes a private variable file\_loader Its type is pointer of PatientFileLoader.

2. The parameter file\_loader is given to the variable file\_loader through the constructor to complete the constructor, and there is also a destructor to free the memory of the file\_loader pointer.

3.PatientFileLoaderAdapter will override AbstractPatientDatabaseLoader's method loadPatients(std::vector<Patient\*>& patientIn), and in this method will call variable file\_loader's loadPatientFile() method, this method will read infomation from current txt file and then returns a list of pointers of type Patient and gives it to loadPatients' parameter patientIn, thus completing the conversion of the PatientFileLoader into the PatientDatabaseLoader.

4.Modify the following code in PatientManagementSystem: \_patientDatabaseLoader(std::make\_unique<PatientDatabaseLoade>(),

to:

\_patientDatabaseLoader(std::make\_unique<PatientFileLoaderAdapter>(new PatientFileLoader(‘patients.txt’))),

We replace <PatientDatabaseLoader> with <PatientFileLoaderAdapter> and add the parameter PatientFileLoader(‘patients.txt’) to initialise a pointer to the PatientFileLoader class and try to open the file ‘patients.txt’, and finally calling loadPatients mrthod to load the data.

Git commits:

* I first create the PatientFileLoaderAdapter in commit 27ee5a.
* I finished PatientFileLoader in commit d67ae4 af3099.
* Other commits include: 7990eb, 3cbf65.

FR4: Observer pattern

Whenever the patient's alert level changes, the system will try to notify the hospital (alert level = red) or the gp (alert level >= orange) so here you can design it using observer pattern. The observation object is patient and the observer is HospitalNotificationSystemFacade and GPNotificationSystemFacade. as shown in the figure, I created a new class called PatientObserver now HospitalNotificationSystemFacade and GPNotificationSystemFacade are subclasses of this class to make it easier to call their methods sendAlertForPatient and sendGPNotificationForPatient consistently.

How it works:

1. Now the Paitent class has an additional public variable subscribers of type PatientObserver, which is used to store each patient's observer. And the Patient class has an additional method addSubscribers to add observers to variable subscribers.

2. The init method in PatientManagementSystem iterates through each patient and adds instances of HospitalNotificationSystemFacade and GPNotificationSystemFacade to each patient.

3. Iterate through the list of \_subscribers for the current patient in the addVitalsRecord method of the PatientManagementSystem and for each observer in the subscribers call sendAlertForPatient and sendGPNotificationForPatient. Therefore each time record the vital or the alert level change immediately, the notification will send when equal or greater than orange(GP) or red(hospital).

Git commits:

* I first create the PatientObserver in commit 326e66.
* I finished logic in PatientManagementSystem in commit 486477.
* Other commits include: 4aa834, 96f51f, ad1d80.