Hacettepe University
Department of Computer Science

Report of Assignment 3
BBM104

19.05.2020
21985164
Tuana Çetinkaya

Hospital Management System

1.Problem Definition

We need a hospital management system which needs to be dynamic in registering and organizing patients. Hospital needs to remove add and bill the patients more efficiently.

2. Solution Model

This program provides a dynamic model called Decorator Design Pattern which allow us to build any combination of our examinations. Also hospital database is now an abstract structure that we could change by implementing our Data Access Object interfaces.

3.Steps of Algorithms

Program uses two DAO interfaces for patient database and admission database. That provides us to switch the source of both independent from each other. Hospital class manage the connection between commands taken by the input file and arrange the patient DAO and admission DAO according to those commands. Also admission interfaces uses the IExamination class which implements the Decorator Design Pattern to function. Then it creates an output file and adds all the tasks and requests as it does them so when everything is done system could update the database files and complete its task.

Hospital

-inputList: ArrayList<String>
-patientDAO: TexPatientDAO
-admissionDAO: TextAdmissionDAO
-outFile: PrintWriter

+performCommands(): void
+updateFiles():void
-addPatient(String patientID, String name,String surname,
String phone, String address): void
-removePatient(int patientID): void
-createAdmission(int admissionID, int patientID): void
-totalCost(int admissionID): void
-listPatients():void
-cropArray(String[] array, int startIndex, int endIndex): String[]

has

input.txt

Main

uses

creates/ stores

<<interface>>
IPatientDAO

+addPatient(Patient p): void
+deletePatient(int patientID): boolean
+updatePatient(Patient old, Patient new): void
+getPatientList(): ArrayList<Patient>

<<interface>>
IAdmissionDAO

+createAdmission(int admissionID, int patientID): Admission
+deleteAdmission(int admissionID): boolean

+addExamination(int admissionID, String type, String[] operation
+totalCost(int admissionID): String
+getAdmissions(): ArrayList<Admission>

creates/updates

implements

implements

creates/updates

Patient

- patientID: int
-surname: String
-name: String
-address: String
-phoneNumber: String

+getters and setters

TextPatientDAO

-initialPatientList: ArrayList<String>
-patients: ArrayList<Patient>

+addPatient(Patient p): void
+deletePatient(int patientID): boolean
+updatePatient(Patient old, Patient new): void
+getPatientList(): ArrayList<Patient>
-createPatients():ArrayList<Patient>

Admission

-admissionID: int
-patientID: int
-examinations: ArrayList<IExamination

+addExamination(String type, Sting[] allOperations): voi
+addExamination(String type, String operations): void
+toString(): String
+totalCost: String
+getters and setters

patient.txt

uses

ReadFiles

-listFormat:: Arraylist<String>
-fileName: String

+getListFormat(): ArrayList<String>
+toString(): String
-readFile(): ArrayList<String>

uses

TextAdmissionDAO

-initialAdmissionList: ArrayList<String>
-admissions: ArrayList<Admission>

+createAdmission(int admissionID,
    int patientID): Admission
+deleteAdmission(int admissionID): boolean
+addExamination(int admissionID,
    String type, String[] operations):void
+totalCost(int admissionID): String
+getAdmissions(): ArrayList<Admission>
+removedPatientUpdate(int patientID): void
-initializeAdmissions(): void
-findAdmission(int admissionID): Admission
-isNumeric(String value): boolean

admission.txt

reads

Text File