Medical Stock Management System

ITMD 510

Project phase I

Table of contents

Description of Project\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2

Main menu for medical Stock Management System\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2

Class description\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

UML Diagrams\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

DDL statements\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5

Table Creation\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5

List of database entities\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7

Entity relationship diagram\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8

Project Team Members:

|  |  |  |
| --- | --- | --- |
| Name | CWID | Email\_id |
| Vighnesh Sawant | A20399823 | vsawant@hawk.iit.edu |
| Prateek Deshmukh | A20399898 | pdeshmukh@hawk.iit.edu |

DESCRIPTION OF THE PROJECT

The project is about Medical Stock Management System where we can add details of Medicines and Suppliers. This application can be accessed by two different kinds of users

* Administrator (Admin)
* Medical Store Manager

Administrator has a control over the different Medical store managers i.e. he/she can add/delete different Medical Store Managers. Also, the administrator can access or generate the daily purchase report of medicines of each and every medical store.

The medical Store Manager will add new medicine / delete an existing medicine / search for an existing medicine / update an existing medicine and also list or view the existing stock of medicine he / she has in his / her store and also can add, search, remove a supplier.

Main menu for Medical Stock Management System

The main menu for the medical stock management system consist of the following things

* Supplier
* Medicine
* Report
* About
* Exit

The Supplier consists of the following

* Add new Supplier
* Delete supplier
* Search supplier
* Update supplier
* List of suppliers

The Medicine consists of the following

* Add new medicine
* Delete medicine
* Update medicine
* Search medicine
* List of medicines

CLASS DESCRIPTION

Medicine and Suppliers will act as interfaces i.e. Interface Medicine and Interface Supplier

Medicine interface will have functions like Add New Medicine which will new medicine into the existing list of medicine, search medicine which will search for an existing medicine, delete medicine to remove an existing medicine, update medicine will update the details of the existing medicine and list of medicine which will be show the entire list of medicine in the stock.

Supplier interface will have functions like Add new supplier which will new supplier into the existing list of suppliers, search supplier which will search for an existing supplier, delete supplier to remove an existing supplier, update supplier will update the details of the existing supplier and list of suppliers which will be show the entire list of suppliers providing the medicines.

Class DailyPurchaseReport will implement medicine interface and class SupplierWiseMedicineReport will implement Supplier interface.

In this way, we will achieve multiple inheritance.

Also, class DailyPurchaseReport and class SupplierWiseMedicineReport will extend class Report

In the DailyPurchaseReport, we will perform analysis of the daily purchased medicines i.e. we will perform day to day analysis of the purchased medicine.

Also, in the SupplierWiseMedicineReport, we will perform supplier wise analysis of medicine where we are going to analyze the different medicines sold by different suppliers as per names

Class Medicine consists of all the basic details of medicines like M\_batchnumber, M\_MedicineName, M\_Company, M\_Quantity etc and implementation methods like addNewMedicine, searchMedicine etc

Class Medicine implements interface MedicineOperations and therefore, all the methods in this interface will be inherited by class Medicine.

Class MedicineDetails consists of count variable which will help in generating the daily purchase report of medicines from a particular Medical Store. Also, class Medicine Details is extending Class Medicine. In this way, we will achieve multi-level inheritance.

Class Supplier consists of all the basic details of supplier like S\_SupplierId, S\_SupplierName, S\_SupplierAddress, S\_SupplierPhoneNumber etc and implementation methods like addNewSupplier, searchSupplier etc

Class Supplier implements interface SupplierOperations and therefore, all the methods in this interface will be inherited by class Supplier.

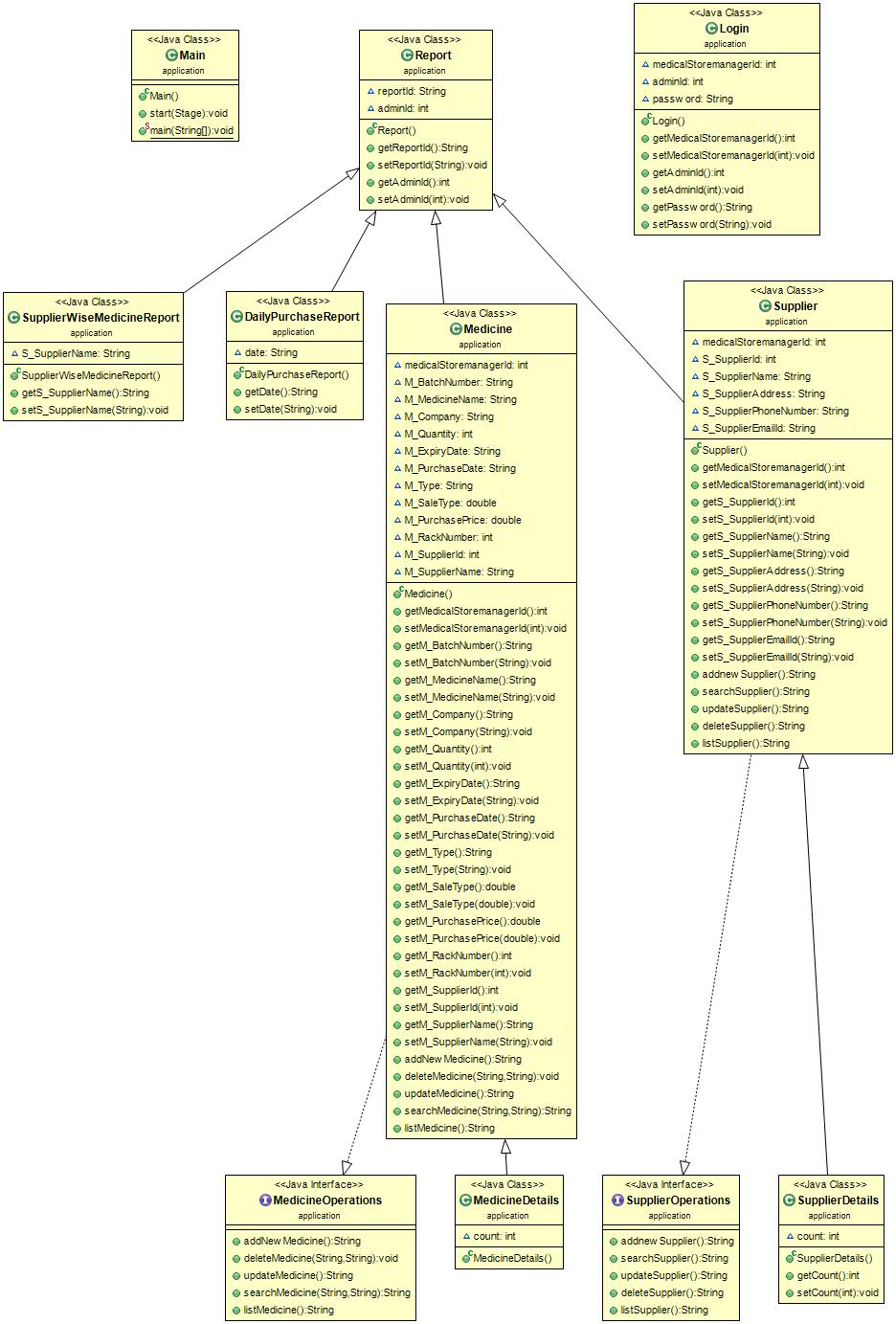
Class SupplierDetails consists of count variable which will help in generating the Supplier wise medicine report of medicines from a particular Supplier. Also, class SupplierDetails is extending Class Supplier. In this way, we will achieve multi-level inheritance

Class Report will be the super class containing variable reportId and adminId and implementing method like Report. All the sub classes like dailyPurchaseReport, supplierWiseMedicineReport, Supplier and Medicine will extend Report.

In this way, we will achieve hierarchical inheritance.

Login class is a class which is used by each different Medical store owners to login, manage and check for their inventory. Therefore, every different medical store owners will have their own different login. It will also have a admin login to generate Reports.

UML DIAGRAM:



DDL Statements

The DDL statements includes create, alter and delete statements

Create statement is used to create tables, schemas in the database

Alter statement is used to alter, modify tables in the database

Delete statement is used to delete the data, tuples from the table but does not delete the entire schema in the database.

Table Creation:

Creating table medicine:

Create table medicine (

typeid number(15),

mcompanyid varchar(30),

mbno varchar(30) not null,

mname varchar(30) default null,

mcompany varchar(30) default null,

mqty number(20) default null,

mexpdate varchar(30) default null,

mpurdate varchar(30) default null,

mtype varchar(30) default null,

mpurprice number(10,2) default null,

msaleprice number(10,2) default null,

mrackno varchar(30) default null,

sid varchar(30) default null,

sname varchar(30) default null,

constraint mbno\_pk primary key (mbno),

constraint typeid\_fk foreign key (typeid) references medicine (typeid),

constraint typeid\_fk foreign key (typeid) references supplier (typeid),

constraint mcompanyid\_fk foreign key (mcompanyid) references Companydetails (mcompanyid),

constraint mbno\_fk foreign key (mbno) references medicinedetails (mbno),

constraint sid\_fk foreign key (sid) references supplier (sid),

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

Creating table supplier:

Create table supplier (

sid varchar(30) not null auto\_increment,

typeid number(15) not null,

sname varchar(30) default null,

saddress varchar(30) default null,

sphoneno varchar(20) default null,

semailid varchar(30) default null,

constraint sid\_pk primary key (sid),

constraint typeid\_fk foreign key (typeid) references UserType (typeid)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=17 ;

Creating table Companydetails:

create table Companydetails (

mcompanyid varchar(30) NOT NULL,

mcompany varchar(30),

mcompanyno number(15),

mcompanyaddress varchar(30),

constraint mcompanyid\_pk primary key (mcompanyid));

Creating table medicinedetails:

Create table medicinedetails (

mbno varchar(30),

nname varchar(30),

mpurprice number(10,2)

msaleprice number(10,2),

constraint mbno\_pk primary key (mbno));

Creating table UserType:

Create table Usertype (

typeid number(15),

typename varchar(30),

typedescription varchar(30),

constraint typeid\_pk primary key (typeid));

Creating table Report:

Create table Report(

reportId number(15) NOT NULL,

adminId number(15),

constraint reportId\_pk primary key(reportId),

constraint adminId\_fk foreign key(adminId) references Login(adminId));

Creating table DailypurchaseReport:

Create table DailyPurchaseReport(

reportId number(15),

date varchar(30),

constraint reportId\_fk foreign key (reportId) references Report(reportId));

Creating table SupplierWiseMedicineReport:

Create table SupplierWiseMedicineReport(

reportId number(15),

supplierName varchar(30),

constraint reportId1\_fk foreign key (reportId) references Report(reportId));

Creating table Login:

Create table Login(

medicalStoremanagerId number(15) NOT NULL,

adminId number(15) NOT NULL,

typeId number(15),

password varchar(30),

constraint medicalStoremanagerId\_pk primary key (medicalStoremanagerId),

constraint typeId\_fk foreign key(typeId) references UserType(typeId),

constraint adminId1\_pk primary key(adminId));

List of Database Entities:

* Medicine
* Supplier
* Companydetails
* medicinedetails
* UserType
* Report
* SupplierWiseMedicineReport
* DailyPurchaseReport
* Login

Entity Relationship Diagram:

