**Cloud 9 Pharmacy**

Jordan Falcon, Ben Hapip, & Buster Schrader

[**http://68.69.250.133/c9/finalprojectsrc/**](http://68.69.250.133/c9/finalprojectsrc/)

**Overview of Goals**

* Create a client friendly web interface to allow pharmacists to confirm and complete prescriptions for customers.
* Give doctors an easy to use interface that allows them to set up prescriptions for their patients.
* Create a database that stores customer, doctor, employee, medication, and insurance information for pharmacy use.
* Grant an Administrator privilege to view all employees and also add or delete employee login privileges.
* Allow doctors to view all unfulfilled prescriptions and fulfilled prescriptions.

**Entity Relationship Diagrams(Below):**

*High Level Diagram*

*Physical Level Diagram*

|  |  |  |
| --- | --- | --- |
| **Employees** | | |
| PK | empId | INT(10) |
|  | firstName | VARCHAR(255) |
|  | lastName | VARCHAR(255) |
|  | phone | INT(10) |
|  | email | VARCHAR(255) |
|  | address | VARCHAR(255) |
|  | city | VARCHAR(255) |
|  | state | VARCHAR(255) |
|  | zipCode | INT(5) |

|  |  |  |
| --- | --- | --- |
| **Customers** | | |
| PK | custId | INT(10) |
|  | firstName | VARCHAR(255) |
|  | lastName | VARCHAR(255) |
|  | phone | VARCHAR(255) |
|  | email | VARCHAR(255) |
|  | address | VARCHAR(255) |
|  | city | VARCHAR(255) |
|  | state | VARCHAR(255) |
|  | zipCode | VARCHAR(255) |

|  |  |  |
| --- | --- | --- |
| **Prescriptions** | | |
| PK | prescripId | INT(10) |
| FK | doctId | INT(10) |
| FK | custId | INT(10) |
| FK | empId | INT(10) |
| FK | medId | INT(10) |
|  | refills | INT(10) |
|  | pillCount | INT(10) |
|  | instructions | VARCHAR(255) |

|  |  |  |
| --- | --- | --- |
| **Insurance** | | |
| PK | insId | INT(10) |
|  | name | VARCHAR(255) |
|  | address | VARCHAR(255) |
|  | city | VARCHAR(255) |
|  | state | VARCHAR(255) |
|  | zipCode | INT(5) |
|  | phone | INT(10) |

|  |  |  |
| --- | --- | --- |
| **Medications** | | |
| PK | medId | INT(10) |
|  | name | VARCHAR(255) |
|  | supplyQuantity | INT(10) |

|  |  |  |
| --- | --- | --- |
| **Medication\_Relation** | | |
| FK | medId | INT(10) |
| FK | medTypeId | INT(10) |

|  |  |  |
| --- | --- | --- |
| **Medication\_Type** | | |
| PK | medTypeId | INT(10) |
|  | typeName | VARCHAR(255) |

|  |  |  |
| --- | --- | --- |
| **Insurance\_Relation** | | |
| FK | insId | INT(10) |
| FK | custId | INT(10) |

**Functional Specifications**

If this particular database were implemented in real life, doctors and pharmacists would be the primary clients. This is because the doctors would need to be able to add patients and also prescriptions to the database to be used by the pharmacy. The information they add would then be used by the pharmacists to fulfill prescriptions. There would also be a team that would perform maintenance on the database and also expand it if the pharmacy administer needed it.

The tasks needed to be carried out are specified in the following sentences. Doctors will need to be able to add prescriptions to existing patients that will be accessed by the pharmacy staff. Doctors will also need to be able to see all prescriptions and whether or not they have been fulfilled. Pharmacy employees will need to be able to search for unfulfilled prescriptions and complete them. Administrators will be able to view all employees and add or delete them accordingly. They’ll be able to take away that privilege but will also keep the records of the employees so that prescription table entries will not be misconstrued.

**Prototype Walkthrough**

*Forms/Reports*

* Login Form (login.php)
  + Three different types of logins: Doctor, Employee, and Admin.
    - Doctor Username: jfalcon
    - Admin Username: admin
    - Employee Username: rhillman
    - Password for all test logins: cloud69
  + Each login displays different tabs and gives different access to certain parts of the application.
* Prescription Form (newPrescription.php)
  + Allows doctors to enter prescription information for patients i.e. medication type, amount of pills, and instructions.
* Prescription Report (showAllPrescriptions.php)
  + Allows doctors to view all prescriptions and groups them by fulfilled and unfulfilled.
* Employee Form (fulfillPrescription.php)
  + Allows employees to fulfill prescriptions by searching for prescriptions using a unique prescription ID.
* Employee Report (pharmacists.php)
  + Displays all the unfulfilled and fulfilled prescriptions for the current employee logged in.
* Admin Report (showCustomers.php)
  + Shows all customers in the system.
* Admin Staff Form (createStaff.php & deleteStaff.php)
  + Allow administrator to grant or remove employee privileges. This form creates or deletes logins but maintains the employee in the database to keep consistency of prescriptions.
* Admin Customer Form (createCustomer.php)
  + Allows the administrator to create new customers at the pharmacy.

**Queries**

*Display all Unfulfilled and Fulfilled Prescriptions for Pharmacists to see (showAllPrescriptions.php)*

This query is used on the doctor’s report page. Sorts unfulfilled prescriptions into a section and fulfilled prescriptions into another.

SELECT CONCAT(Customers.firstName,' ',Customers.lastName) AS Customer, Medications.name AS medicine, typeName AS type, refills, pillCount, instructions

FROM Prescriptions AS p

INNER JOIN Customers ON Customers.custId = p.custId

INNER JOIN Medications ON Medications.medId = p.medId

INNER JOIN Medication\_relation AS mr ON mr.medId = p.medId

INNER JOIN Medication\_type AS mt ON mt.medTypeId = mr.medTypeId

WHERE empId IS ";

*Fulfill Prescriptions Search & Update Queries (fulfillPrescription.php)*

Query that allows employees to search for a prescription .

SELECT Prescriptions.prescripId AS pid, CONCAT(Doctors.firstName,' ',Doctors.lastName,' ',licenseNo) AS Doctor, CONCAT(Customers.firstName,' ',Customers.lastName) AS Customer, Customers.phone AS phone, refills, pillCount,

instructions, clinic, Medications.name AS medicine, supplyQuantity

FROM Prescriptions

INNER JOIN Doctors ON Doctors.doctId = Prescriptions.doctId

INNER JOIN Cstomers ON Customers.custId = Prescriptions.custId

INNER JOIN Medications ON Medications.medId = Prescriptions.medId

WHERE Prescriptions.empId IS NULL AND Prescriptions.custId = '" . $\_POST["Customerid"] . "'";

Query that updates the prescription.

UPDATE Prescriptions SET empId = '" . $\_SESSION["logpersonid"] . "' WHERE prescripId = '" . $\_SESSION["prescript"] . "'";

*Create a Prescription (newPrescription.php)*

Query that inserts a new prescription into the prescription table.

$customer\_id = $\_POST["customerid"];

$medication\_id = $\_POST["medicationid"];

$pill\_count = $\_POST["pillcount"];

$refill = $\_POST["refill"];

$instructions = trim($\_POST["instructions"]);

$doctor = $\_SESSION["logpersonid"];

$stmt = $mysqli->prepare("INSERT INTO Prescriptions VALUES (NULL, ?, ?, NULL, ?, ?, ?, ?)");

$stmt->bind\_param("iiiiis", $doctor, $customer\_id, $medication\_id, $refill, $pill\_count, $instructions);

$res = $stmt->execute();

*Display all fulfilled and unfulfilled Prescriptions (showCurrent.php)*

Query that displays all fulfilled and unfulfilled prescriptions and sorts them by last name.

SELECT CONCAT(Customers.firstName,' ',Customers.lastName) AS Customer, Medications.name AS medicine, GROUP\_CONCAT(' ',typeName) AS type, refills, pillCount, instructions

FROM Prescriptions AS p

INNER JOIN Customers ON Customers.custId = p.custId

INNER JOIN Medications ON Medications.medId = p.medId

INNER JOIN Medication\_relation AS mr ON mr.medId = p.medId

INNER JOIN Medication\_type AS mt ON mt.medTypeId = mr.medTypeId

WHERE p.doctId = '" . $\_SESSION["logpersonid"] . "' AND empId IS ";

Sorts the above query. On the php page, these two queries are separate.

GROUP BY p.prescripId ORDER BY Customers.lastName";

*Create a new Customer (createCustomer.php)*

INSERT INTO Customers VALUES (NULL, '"

. $\_POST["firstname"] . "','"

. $\_POST["lastname"] . "','"

. $\_POST["phone"] . "','"

. $\_POST["email"] . "','"

. $\_POST["address"] . "','"

. $\_POST["city"] . "','"

. $\_POST["state"] . "','"

. $\_POST["zip"] . "')";

*Create a new Doctor (createStaff.php)*

Doctors VALUES (NULL,'" . $\_POST["firstname"] . "','"

. $\_POST["lastname"] . "','"

. $\_POST["license"] . "','"

. $\_POST["clinic"] . "')";

*Create a new pharmacist (createStaff.php)*

Employees VALUES (NULL,'" . $\_POST["firstname"] . "','"

. $\_POST["lastname"] . "','"

. $\_POST["phone"] . "','"

. $\_POST["email"] . "','"

. $\_POST["address"] . "','"

. $\_POST["city"] . "','"

. $\_POST["state"] . "','"

. $\_POST["zip"] . "')";

*Delete an Employee’s login information (deleteStaff.php)*

DELETE FROM Login WHERE job = '0' AND personId = '" . $\_SESSION["goodbye"] . "'";

*Delete an Employee’s information from the Employee Table (deleteStaff.php)*

DELETE FROM Employees WHERE empId = '" . $\_SESSION["goodbye"] . "'";

*Display all Customers in the database (showCustomers.php)*

SELECT \* FROM Customers;

**Future Improvements**

Further implementation of the Insurance Table is something that could be included in future updates. Being able to calculate a deductible for customers based on their insurance premiums is the next step. Also, being able to automatically update the number of Medications on hand when prescriptions are fulfilled would be handy. Including a way to branch out the pharmacy from only including one physical pharmacy into a branch of pharmacies that were all able to access the same prescription list would help to expand the company further.

Database design could also be improved to include more information regarding medications and their typing. Expanding the total number of medications offered is something minor that could easily be expanded for the future.