

DBMS Assignment 5

Ophthalmic Software

Group Members: -

Mohit Apte

Daksha Bhargava

Ayush Ambhorkar

Problem Statement: -

The MYSQL-Based Ophthalmic Software project aims to provide a comprehensive solution for ophthalmologists to manage patient data, write glasses prescriptions, and generate medicine prescriptions for both in-patient (IPD) and out-patient (OPD) settings. This software allows ophthalmologists to record and store patient information securely in the MYSQL Database. It offers a user-friendly interface to input and manage patient data, including medical history, previous prescriptions, and examination results. With this information readily available, ophthalmologists can efficiently generate accurate glasses prescriptions tailored to each patient's needs.

Objectives: -

1. **Comprehensive Data Management:** The software serves as a holistic solution for ophthalmologists to oversee patient data efficiently.
2. **Secure Data Storage:** To record and safely store patient information in the MYSQL Database, ensuring data integrity and protection from unauthorized access.
3. **Prescription Generation:** Allows ophthalmologists to swiftly generate glasses prescriptions, ensuring that each prescription is tailored accurately according to the patient's needs.
4. **Medicine Prescription:** Facilitate the creation of medicine prescriptions for both in-patient (IPD) and out-patient (OPD) settings, allowing for better medication management.
5. **User-friendly Interface:** The software provides an intuitive user interface, making it simpler for ophthalmologists to input, access, and manage patient data.

6. **Complete Patient History:** Enables ophthalmologists to maintain a comprehensive record of a patient's medical history, prior prescriptions, examination outcomes, and more, fostering a holistic view of the patient's health.
7. **Efficiency:** Streamlines the workflow for ophthalmologists, reducing the time spent on manual paperwork and ensuring faster and more accurate service delivery to patients.
8. **Accessible Data Retrieval:** Ophthalmologists can quickly retrieve patient data when needed, facilitating informed decisions during patient consultations and treatments.
9. **Customizable:** Being database-driven, the software can be customized or scaled based on individual clinic or hospital needs, offering flexibility in data management and features.

Functional Requirements:

1. User Authentication:

- The software should provide a secure login mechanism for ophthalmologists and other authorized staff.
- The system should support password reset and recovery mechanisms.
- Role-based access controls to limit access to sensitive patient data.

2. Patient Registration and Management:

- Ability to register new patients with fields for personal details, contact information, and medical history.
- Edit or update existing patient information.
- Search for patients based on multiple criteria like name, patient ID, or date of birth.
- Delete or archive old patient records.

3. Prescription Management:

- Generate, edit, and save glasses prescriptions based on patient examination data.
- Generate medicine prescriptions, with dosage, frequency, and duration for IPD and OPD settings.
- Store and retrieve previous prescriptions for reference.
- Print and/or email prescriptions to patients or pharmacies.

4. Medical Examination Record:

- Input fields to record results from various eye examinations.
- Storage for image or document-based results like retina scans or corneal maps.
- Option to compare current examination results with previous ones.

5. Data Security:

- Encryption of sensitive patient data.
- Regular automated backups of the database.
- Secure connection to the MYSQL database.

6. User Interface:

- Intuitive dashboard showing pending tasks, recent patients, and other important information.
- Clearly labeled data entry forms and fields.
- Option for customizing the interface based on the user's preference.

7. Reporting:

- Generate daily, weekly, or monthly reports on the number of patients seen, prescriptions issued, etc.
- Export reports in various formats like PDF, Excel, etc.

Functional Dependencies:

1. **patient table:**

id → MRD, first_name, middle_name, last_name, age, sex, address,
mobile_no, land_no, misc

2. **prescription table:**

prescription_id → patient_id, rds, rdc, rda, rdv, rcs, rcc, rca, rcv, rns, rnc,
rna, rnv, lds, ldc, lda, ldv, lcs, lcc, lca, lcv, lns, lnc, lna, lnv, ipd, entry1,
entry2, entry3, entry4
patient_id → prescription_id

3. **OPD table:**

opd_id → patient_id, complaints, examination, diagnosis, medicine,
history, advised, date
patient_id → opd_id

4. **IPD table:**

ipd_id → patient_id, doatxt, t1txt, dodtxt, t2txt, cftxt, opnotestxt,
investigationtxt, postmedicinetxt, surgeryadvisingtxt,
adviseondischarge.txt, date
patient_id → ipd_id

5. **medicine table:**

medicine_id → patient_id, x, medname, medtype, medadvice, days,
dwm, qty
patient_id → medicine_id

6. **image table:**

image_id → patient_id, img_data
patient_id → image_id

Code Screenshots

```
from PIL import Image
import io
from docx.shared import Inches
from tkinter import messagebox
import pymysql
import datetime
import pymysql.cursors

def validateLogin(username, password):
    if username.get() == 'DR' and password.get()=="DR":
        main_page()

x = []
medname = []
medtype = []
medadvice = []
days = []
dwm = []
qty = []
img_data = ""

from reportlab.pdfgen import canvas

def draw_multiline_text(canvas, text, x, y, width, height, font_size):
    lines = []
    current_line = ""
    words = text.split()
    max_line_height = 0

    for word in words:
        if canvas.stringwidth(current_line + " " + word, "Helvetica", font_size) < width:
            current_line += " " + word
        else:
            lines.append(current_line)
            current_line = word

    if current_line:
        lines.append(current_line)

    OPD.history,
    OPD.advised,
    medicine.x,
    medicine.medname,
    medicine.medtype,
    medicine.medadvice,
    medicine.days,
    medicine.dwm,
    medicine.qty,
    IPD.doatxt,
    IPD.t1txt,
    IPD.dodtxt,
    IPD.t2txt,
    IPD.cftxt,
    IPD.opnotestxt,
    IPD.investigationtxt,
    IPD.postmedicinetxt,
    IPD.surgeryadvisingtxt,
    IPD.adviseondischarge.txt

FROM patient
LEFT JOIN prescription ON patient.id = prescription.patient_id
LEFT JOIN OPD ON patient.id = OPD.patient_id
LEFT JOIN IPD ON patient.id = IPD.patient_id
LEFT JOIN medicine ON patient.id = medicine.patient_id
WHERE
    patient.MRD LIKE %s AND
    patient.first_name LIKE %s AND
    patient.middle_name LIKE %s AND
    patient.last_name LIKE %s AND
    patient.age LIKE %s AND
    patient.sex LIKE %s AND
    patient.address LIKE %s AND
    patient.mobile_no LIKE %s AND
    patient.land_no LIKE %s AND
    patient.misc LIKE %s
"""

cursor.execute(search_query, (mrd.get(), fn.get(), mn.get(), ln.get(), age.get(), sex.get(), address.get(), mob
```

Project Screenshots

Out Patient Department

In Patient Department

New Patient

Old Patient

Patient Information

Name: John Manthan SmithAge: 52Sex: MaleMob: 9112009923Date: 24/11/2023

Patient Details

COMPLAINTS

EXAMINATION

DIAGNOSIS

MEDICINE

Complaints

Examination

Diagnosis

Medicine

HISTORY

DIAGRAM

ADVISED

PRESCRIPTION

History

Diagram

Advised

Prescription:

Save

Complete History

Ophthalmic Software

Out Patient Department

In Patient Department

Patient Information

Name: John Manthan SmithAge: 52Sex: MaleMob: 9112009923Date: 24/11/2023

Discharge Summary

Date of Admission

Time

Date of Discharge

Time

Patient Details IPD

Clinical Findings

Invesitgation

Surgery Advising

Clinical Findings

Invesitgation

Surgery Advising

Operation Notes

Post Operative Medicines

Advice on Discharge

Operation Notes

Post Operative Medicines

Advice on Discharge

Save

IPD History