Mayo Clinic Patient Care Manager (PCM)

Andrew Krause, Spring 2022

The final project document contains information pertaining to the database that was created for the CS 364 semester project. The document contains sections such as the synopsis, the database ER diagram with descriptions, functionalities, stakeholders, and technological requirements.

# Table of Contents

Section 1: Synopsis

Managing patient data has become an important aspect of health care in the modern world. Assessing patient related issues, diagnosing the problem, and applying a treatment is a fundamental process for health care providers such as Mayo Clinic Health Systems. The nonprofit organization is ranked the number one hospital in the United States (newsnetwork.mayoclinic.org). The three pillars of focus represented in the three shields of Mayo Clinic's logo stand for clinical practice, education, and research. The three areas of focus in the hospital enable patients to find hope and healing at a place that strives to deliver effective care for medical issues. Mayo Clinic sees over 1 million patients a year (www.mayoclinic.org). To continue providing the variety of services for patients seeking medical assistance, the hospital needs an efficient and effective system that can organize and display relevant patient data.

This project seeks to develop a database that organizes patient, provider, and treatment related data. The project will be a simple web application with a SQL database as a part of the backend framework along with Node.js for the server. The front end will help to organize and display the data in a clear, simple format. The front end will be created using HTML and CSS. There will be four primary views of the data: Patients, Providers, Treatments, and Departments. There will also be a view to display the most urgent patient cases currently being evaluated at the hospital. Users (assumed to be staff of Mayo Clinic) will have the ability to view all the data on display, make updates to certain features as needed, create new entries into different parts of the database, and remove data.

Upon completion, the web application will be hosted on a Heroku server (www.heroku.com), which is a service for publishing work on the world wide web. Although there will be no authentication or security in accessing this project, please note that this is a prototype with dummy data. Future releases of the project may include implemented security features. The primary goal of this project is to develop a system with a database that enables care providers to view organized patient data, make informed decisions, provide different services, and track treatment plans.

Section 2: Database

The Entity Relationship (ER) diagram for the Mayo Clinic Patient Care Manager (PCM) Database and descriptions are shown on the next pages:
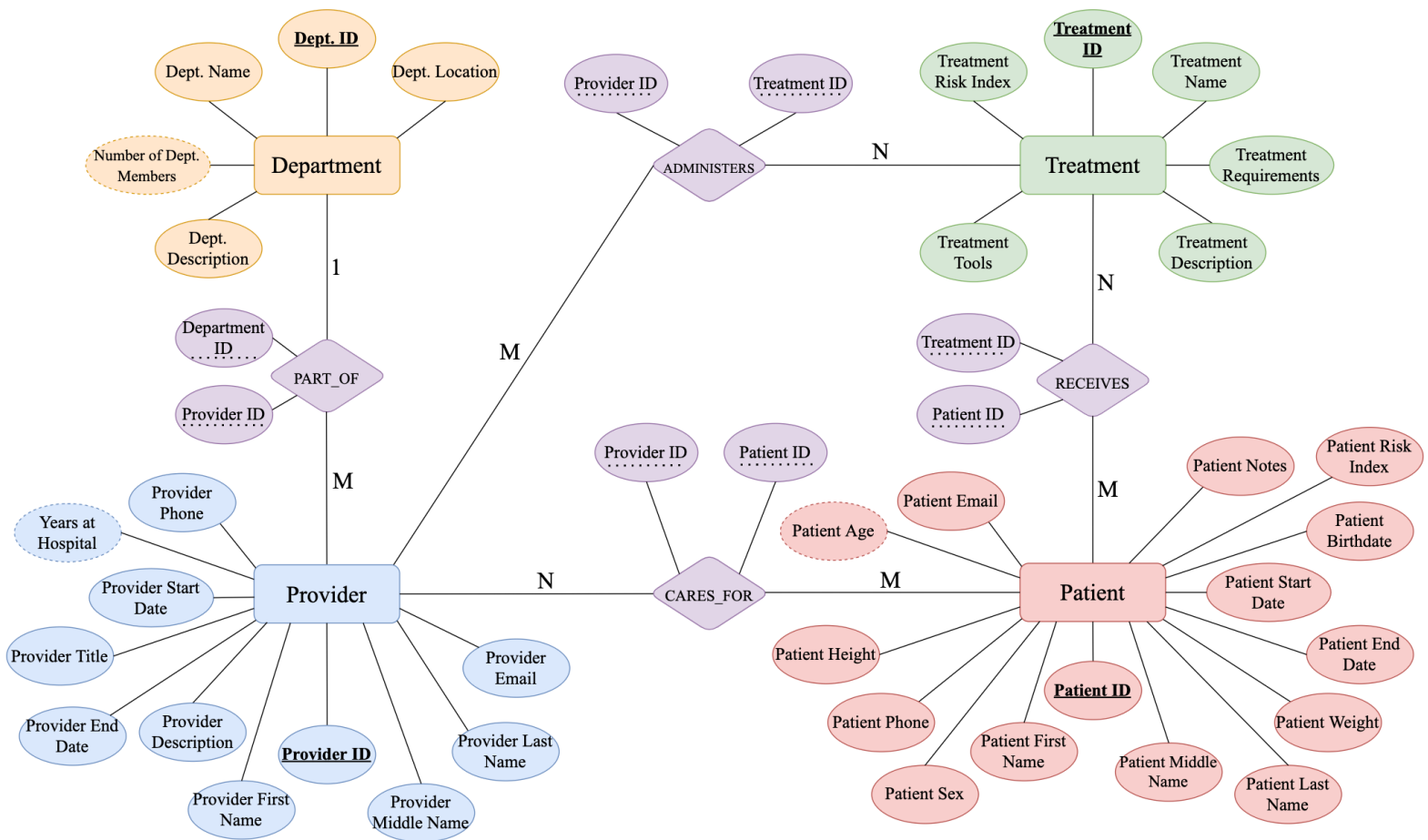
Treatment

- The treatment entity contains attributes related to the medicine or procedure that can be applied to a patient. A unique ID is assigned to each treatment.
- The attributes include the name, description of the treatment, requirements (such as authorization to practice the treatment), and the tools needed for the treatment.
- A risk index (risk ranges form 0-10) is included as an attribute to indicate how dangerous the treatment can be to patient health.

Patient

- The patient entity contains the largest number of attributes. Basic attributes include the patient's first name, last name, birthdate, weight, height, medical notes, unique identifier, and contact information.
- The date the patient was admitted as well as the date they were discharged (if applicable) is included.
- The patient's age is a derived attribute calculated from the patient birthdate and current year.
- The patient entity shares relationships with the treatment and provider attributes.

Provider

- Providers share relationships with the patient, treatment, and department entities. Providers the medical staff authorized to administer treatments.
- Attributes such as the provider name, contact information, title are listed.
- A derived attribute for the number of years the provider has been at the hospital, based on current year and start date, is included.
- Provider start date and end date (if applicable) are used to determine how long the provider has worked with the hospital.

Patient Receives Treatment

- In the relationship, the patient ID and the treatment ID are used as foreign keys.
- This relationship is important give that it allows the user to observe which patients are receiving which treatments at the hospital.
- This is an example of a many-to-many relationship.

Provider Cares for Patient

- In the relationship, the patient ID and the provider ID are used as foreign keys.
- This relationship is important given that it allows a user to observe which patients are in the care of certain providers in the hospital.
- This is an example of a many-to-many relationship.

Provider Part of Department

- The relationship shows that providers are grouped into departments that specialize in various treatments.

Department

- The department entity shares a one-to-many relationship with the provider entity (many providers belong to one department).
- Departments help group providers into teams that specialize in certain treatments or procedures.
- The department name, location, number of members, and description are included as attributes.
- Like the other entities in the database, each department entity contains a unique identifier.

- The department ID and provider ID are included as foreign keys.
- This is an example of a one-to-many relationship.

Provider Administers Treatment

- The provider entity and the treatment entity are joined in a relationship where the provider administers treatments.
- The provider ID and the treatment ID are used as foreign keys to join the two tables.
- This is an example of a many-to-many relationship.

Section 3: Functionalities

There will be five main uses of the Patient Care Management website:

1) **Patient View**

   When a user selects the "patient" view, they will be given a list containing all the patients that have been to Mayo Clinic in the past decade. The patients in the database include patients who are visiting for a simple appointment such as a checkup. The data will be ordered first by patients who are currently in treatment (i.e., have not yet been discharged from the hospital), and next alphabetically. In addition to the use of the SELECT-FROM-WHERE clauses, HAVING and ORDER BY clauses will also need to be used.

2) **Treatment View**

   The "treatment" view displays all the different medical treatments and services offered to patients at the hospital. The department in charge of administering each treatment will listed as an attribute along with a description, requirements, and patients currently receiving the care. This view will be joined with the Department View table and the Patient View table.

3) **Provider View**

   Upon selecting the "provider" view, the user will be shown a list of all the medical staff that are actively administering care to patients in the hospital. Information regarding years at the hospital, education, and department will also be listed for each provider. The providers will be ordered first by years at Mayo Clinic and next by first and last name. Like the Patient View, the ORDER BY clause will be utilized.

4) **Department View**

The "department" view shows the various divisions of health care service within Mayo Clinic. Attributes include a description of the department, members of the department from the Provider View table, and treatments administered by the department from the Treatments View table. A GROUP BY clause will be used to include the attributes related to the department from the other tables that will be joined with the Department View table in a relationship.

5) **Statistics View**

The final view that is included in the database, the "statistics" view, will list the results of the three complex queries that are executed in the database. For example, data resulting from a query that obtains patients who have a higher risk index would be displayed on this page. Any information that is listed on the statistics page is also existent within the database given that the information was pulled from one of the four other tables/views on the website.

Other Functions:

- All the tables in the database have CRUD operations (Create, Read, Update, Destroy). For each table, users can add new data, update current data, view data, and remove data.

- Users can interact with the database via a web application with a front end that is designed to be simple and easy to navigate.

- A large portion of the data is listed out in a clean, human-readable format. The data can be manipulated to reflect changes that are occurring in the organization using the database.

- An important function of the database is the ability to view the status of patients and make changes if necessary. The Statistics View table is an important feature implemented through SQL that filters the data to list out information such as the high-risk patients at the hospital.

Section 4: Stakeholders

Users of the Mayo Clinic Patient Care Management (PCM) database are authorized hospital staff and medical personnel. Security features will be added later to limit access to authorized users as well as ensure that personal information is encrypted. Medical users may utilize the data to track the number of patients admitted for various illnesses as well as observe the duration of time patients are in the hospital for a given treatment or procedure. By using the available data, hospital staff and researchers may be able to develop more effective treatment strategies over time.

Section 5: Technological Requirements

The platform for the database is a web application created using HTML, CSS, JavaScript, EJS, Node.js, and SQL. A Heroku server is used to publish the web app online along with JawsDB, an add-on for Heroku, as the SQL database. The website will use searches based on queries written to organize and display data in the appropriate tables for the database. Users will also be able to interact with the database by reading data, updating data, adding data, and removing data.

Section 6: Screenshots of Project

Patient Care Manager

| View → | About → |

Home Page of Patient Care Manager Website

PCM   Home   View   About

Select Category to View

Patients	Treatments
Providers	Departments
Statistics	Exit

View Page of Patient Care Manager Website

PCM   Home   View   About

Patients

| ← Back | ⊕ New |

💓 Anthony Marshal Jones	ⓘ ✏️ 🗑️
💓 Isaiah Steven Mathews	ⓘ ✏️ 🗑️
💓 Mary Ann Jensen	ⓘ ✏️ 🗑️
💓 Andrew Lanning Jenkins	ⓘ ✏️ 🗑️

Patients Page of Patient Care Manager Website

Section 7: Advanced Queries

Advanced Query 1: Top 5 High-Risk Patients *(Group 1)*

```
SELECT * FROM Patient
      WHERE Patient.PatientRiskIndex == ?
      ORDER BY Patient.PatientRiskIndex DESC
      LIMIT ?;
```

The first complex query selects all the patients that have a risk index (an attribute in the patient table) greater than or equal to the value of 6. The top 5 high-risk patients are displayed on the "Statistics" page of the web application. The user can also limit the number of results returned from the query. Note that "?" equates to user input.

Advanced Query 2: Providers Caring for 5 or More Patients *(Group 2)*

```
SELECT Provider.ProviderFirstName, Provider.ProviderLastName,
               Provider.ProviderTitle, count(*) AS NumPatients
      FROM Provider
      JOIN Cares_For
            ON Provider.ProviderID = Cares_For.Cares_ProviderID
      JOIN Patient
            ON Cares_For.Cares_PatientID = Patient.PatientID
      GROUP BY Provider.ProviderID
      HAVING count(*) >= ?
      LIMIT ?;
```

The second complex query selects all the providers who are caring for a number of patients that is greater than or equal to the input entered by the user. The providers who satisfy the user search input are displayed on the "Statistics" page of the web application. Also, the user can limit the number of results returned. Note that "?" equates to user input.

Advanced Query 3: Providers Authorized to Administer High-Risk Treatments *(Group 3)*

SELECT Provider.ProviderID, Provider.ProviderFirstName,
Provider.ProviderLastName, Provider.ProviderTitle,
Treatment.TreatmentName
FROM Provider
JOIN Administers_Treatment
ON Provider.ProviderID =
Adminsters_Treatment.Adminsters_ProivderID
JOIN Treatment
ON Administers_Treatment.Administers_TreatmentID =
Treatment.TreatmentID
WHERE Treatment.TreatmentName IN (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
LIMIT = ?;

The third and final complex query selects which providers are currently administering at least one of the treatments at the hospital that are entered by the user as input. The providers who are displaying at least one of the treatments are listed along with the treatment they are administering on the "Statistics" page of the website. The user can also limit the number of results returned from the query. Note that "?" equates to user input.

Section 8: Future Features

Features that may not be implemented or that will likely be implemented in the future are listed here:

- **Treatment completed indicator.** Feature that allows a user to "discharge" a patient from Mayo Clinic when their treatment is completed. Upon being discharged from the hospital, the amount of time that the patient received treatment from the hospital will be displayed within that patient's information.

- **Secure login and authentication functions.** Features that encrypt data and allow only authorized users to access the Mayo Clinic Patient Care Management database may be implemented in the future.

- **Responsive layout.** Update the styling and layout of the application to view the database on a variety of devices such as larger desktops and mobile devices. The transitions from different screen widths will be clean and simple.

- **Additional queries.** In the future, queries may be written to expand the number of ways users can view and analyze the data. With the creation of more complex queries, the users of the database will be able to make new informed decisions regarding patient health.