Project Design Document

for

Hospital Management System

Prepared by
Group 19
Onur Altuğ Akça
Onuray Toroslu
Gökmen Anıl Dağ
Mehmet Ege Bozdemir
Berfin Yucak

7.11.2022

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Project Design Document	07.11.2022	Project Design Document is created.	1.0

1. Introduction

1.1 Definition

This Project Design Document for Hospital Management System provides detailed information on mainly the DBMS of a hospital. The acronyms and abbreviations used in this document are listed in the following subsection. The users that have access to the system, their needs, and corresponding solutions to that needs are discussed in subsection 1.3, Users. The rest of the first section is dedicated to explaining the assumptions and business rules. The HMS is a student project conducted by Group 19 of the CENG 315 Fall semester class of IZTECH and supervised by Associate Professor Belgin Ergenç Bostanoğlu and teaching assistants Büşra Güvenoğlu & Leyla Tekin. Hence, all the assumptions and business rules are based only on case studies.

The second section of the document focuses on providing a clear understanding of the entities and their relations that composes the DBMS via various texts, tables, and schemas.

1.2 Acronyms and Abbreviations

DBMS	Database Management System
HMS	Hospital Management System
CENG	Computer Engineering
IZTECH	İzmir Institute of Technology
MP	Medical Personnel
ER	Entity Relation
ERD	Entity Relation Diagram
RS	Relational Schema

1.3 Users

The HMS is planned to serve three types of users: Medical Personnels, Patients and Administrators. All three are discussed separately below.

1.3.1 Medical Personnels

The MPs refer to the doctors and the nurses that are working at the hospital. Their main needs are foreseen as managing the patient records and the appointments. To be more specific; to view, approve/cancel and add appointments, view the previous diagnoses and medicines that were given to the patient, and add new ones if necessary.

1.3.2 Patients

The patients need to manage the same data as the MPs, but on a lower authorization level. A patient can view only their own medical records: Diagnoses, medicines and appointments. The patient can add a new appointment or cancel an upcoming one. But cannot manipulate medicine or diagnoses for security concerns.

1.3.3 Administrators

The administrators are the most authorized group in all three types of users. They can manage the personnel or patient data, add/delete departments, remove/add medicines, cancel/approve/transfer appointments. For example, if a personnel is fired or on vacation, a medicine is pulled out of the market, a nurse changes departments etc. administrators are the ones to manipulate the regarding data accordingly.

1.4 Assumptions

- The physical infrastructure is provided by the employer.
- All users have sufficient and stable devices and internet connections to access the system.

1.5 Business Rules

- Each MP is assigned to a department.
- Each visit is made with an appointment. Appointments are created by the patients. A patient chooses a department to get an appointment from and chooses an available doctor, date, and time. Unless the appointment is canceled by the doctor or the administrator, it is automatically approved. The patient has the option to cancel it as well.
- For the patient records, the date admitted, ID, the date discharged are entered manually by the personnel at the front desk. The medicine and the diagnosis is entered by the MP. A diagnosis can be done only by a doctor.
- The insurance and other payment details are handled by a third party due to security concerns. In the payment table, only the transaction ID and payment status are held to be able to discharge the patient, or present to authorities if needed.

2. Design

2.1 Entity Sets & Attributes

The entity sets and their attributes are listed in separate tables below. The explanatory texts (if needed) follows the concerning table.

Personnel
personnelID
personnelName
personnelFirstName
personnelLastName
personnelPhonoNumber
personnelAddress
personnelEmail
personnelPassword

- ➔ All the personnel working at the hospital, such as doctors, nurses, secretaries, officers, administrators, generators, caregivers belong to this entity. They are grouped and distinguished by their ID’s. Only some of them which are defined in section 1.3 Users are authorized to login to the system and manipulate data.

Patient
patientID
patientName
patientFirstName
patientLastName
patientPhoneNumber
patientAddress
patientEmail
patientPassword

- ➔ All patients are identified with their 11-digits TR ID number.

Records
recordID
dateAdmitted
dateDischarged
diagnosisID
appointmentID

- ➔ For each visit, another record is created.

Department
departmentID departmentName building room

Diagnosis
diagnosisID testResult

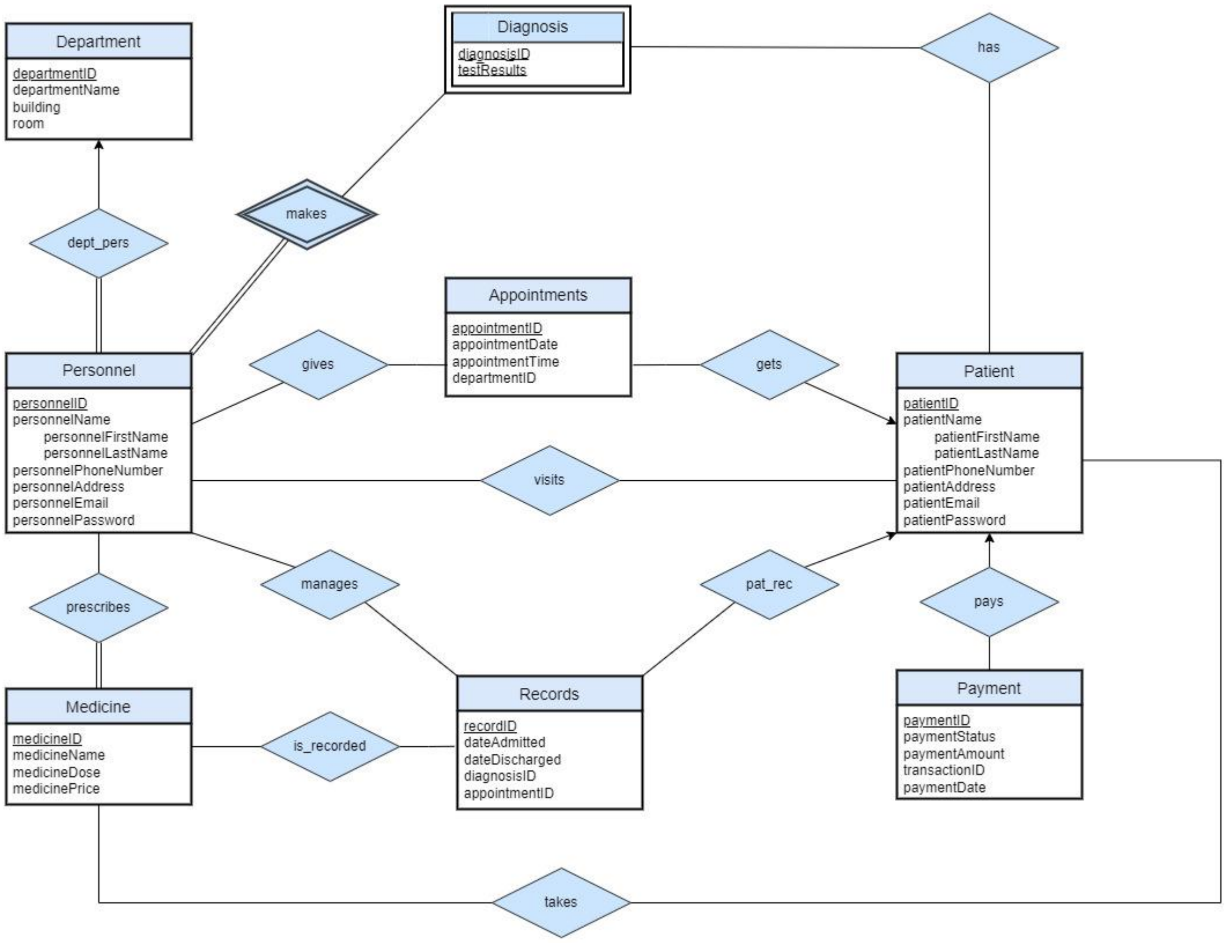
- ➔ Diagnosis is a weak entity since it can only be made by a doctor and does not exist otherwise.

Payment
paymentID paymentStatus paymentAmount transactionID paymentDate

Medicine
medicineID medicineName medicineDose medicinePrice

Appointments
appointmentID appointmentDate appointmentTime departmentID

2.2 ER Diagram



2.3 Relation Sets & Attributes

To create the relational schema, the necessary steps for converting the ERD to RS by reduction are listed and explained below.

2.3.1 Step 1: Listing the Entities

Department (departmentID, departmentName, building, room)

Personnel (personnelID,
personnelFirstName, personnelLastName,
personnelAddress,
personnelEmail,
personnelPassword)

Medicine (medicineID, name, dose, price)

Records (recordID, dateAdmitted, dateDischarged, diagnosisID, appointmentID)

Payment (paymentID, paymentStatus, paymentAmount, transactionID, paymentDate)

Patient (patientID,
patientFirstName, patientlLastName,
patientAddress,
patientEmail,
patientPassword)

Appointments (appointmentID, appointmentDate, appointmentTime, departmentID)

Diagnosis (personnelID, diagnosisID, testResults)

- Since Diagnosis is a weak entity, the primary key of the identifying strong entity set Personnel is included.

2.3.2 Step 2: Listing the Relations

- Between the entities Personnel & Department:

dept_pers = (personnelID, departmentID)

- Between the entities Records & Patient:

pat_rec = (recordID, patientID)

- Between the entities Personnel & Medicine:

prescribes = (personnelID, medicineID)

- Between the entities Personnel & Appointments:

gives = (personnelID, appointmentID)

- Between the entities Personnel & Patient:

visits = (personnelID, patientID)

- Between the entities Personnel & Records:

manages = (personnelID, recordID)

- Between the entities Patient & Diagnosis:

has = (patientID, diagnosisID)

- Between the entities Medicine & Records:

is_recorded = (medicineID, recordID)

- Between the entities Patient & Payment:

pays = (patientID, paymentID)

- Between the entities Patient & Appointments:

gets = (patientID, appointmentID)

- Between the entities Patient & Medicine:

takes = (patientID, medicineID)

Since the diagnose relation links the weak entity set Diagnosis to its identifying strong entity set Personnel and the Diagnosis schema already contains the attributes that would appear in the diagnose schema, the relationship set between the entities Diagnosis and Personnel is not included to this list.

2.3.3 Step 3: Redundancy of Schema

Department (departmentID, departmentName, building, room, personnelID)

Personnel (personnelID,
personnelFirstName, personnelLastName,
personnelAddress,
personnelEmail,
personnelPassword,
departmentID)

- departmentID referencing the relation between the entities Department and Personnel.

Medicine (medicineID, name, dose, price)

Records (recordID, dateAdmitted, dateDischarged, diagnosisID, appointmentID, **patientID**)

- patientID referencing the relation between the entities Patient and Records.

Payment (paymentID, paymentStatus, paymentAmount, transactionID, paymentDate, **patientID**)

- patientID referencing the relation between the entities Patient and Payment.

Patient (patientID,
patientFirstName, patientlLastName,
patientAddress,
patientEmail,
patientPassword)

Appointments (appointmentID, appointmentDate, appointmentTime, departmentID, **patientID**)

- patientID referencing the relation between the entities Patient and Appointments.

Diagnosis (personnelID, diagnosisID, testResults)

~~dept_pers =(personnelID, departmentID)~~

~~pat_rec =(recordID, patientID)~~

prescribes = (personnelID, medicineID)

- Since prescribes is a many-to-many relationship set, it is represented as a schema with attributes for the primary keys of the two participating entity sets: Personnel and Medicine.

gives = (personnelID, appointmentID)

- Since gives is a many-to-many relationship set, it is represented as a schema with attributes for the primary keys of the two participating entity sets: Personnel and Appointments.

visits = (personnelID, patientID)

- Since visits is a many-to-many relationship set, it is represented as a schema with attributes for the primary keys of the two participating entity sets: Personnel and Patient.

manages = (personnelID, recordID)

- Since manages is a many-to-many relationship set, it is represented as a schema with attributes for the primary keys of the two participating entity sets: Personnel and Records.

has = (patientID, diagnosisID)

- Since has is a many-to-many relationship set, it is represented as a schema with attributes for the primary keys of the two participating entity sets: Patient and Diagnosis.

is_recorded = (medicineID, recordID)

- Since is_recorded is a many-to-many relationship set, it is represented as a schema with attributes for the primary keys of the two participating entity sets: Medicine and Record.

~~pays =(patientID, paymentID)~~

~~gets =(patientID, appointmentID)~~

takes = (patientID, medicineID)

- Since takes is a many-to-many relationship set, it is represented as a schema with attributes for the primary keys of the two participating entity sets: Patient and Medicine.

2.4 Relational Schema

Department (departmentID, departmentName, building, room, personnelID)

Personnel (personnelID,
personnelFirstName, personnelLastName,
personnelAddress,
personnelEmail,
personnelPassword,
departmentID)

Medicine (medicineID, name, dose, price)

Records (recordID, dateAdmitted, dateDischarged, diagnosisID, appointmentID, patientID)

Payment (paymentID, paymentStatus, paymentAmount, transactionID, paymentDate, patientID)

Patient (patientID,
patientFirstName, patientlLastName,
patientAddress,
patientEmail,
patientPassword)

Appointments (appointmentID, appointmentDate, appointmentTime, departmentID, patientID)

Diagnosis (personnelID, diagnosisID, testResults)

prescribes = (personnelID, medicineID)

gives = (personnelID, appointmentID)

visits = (personnelID, patientID)

manages = (personnelID, recordID)

has = (patientID, diagnosisID)

is_recorded = (medicineID, recordID)

takes = (patientID, medicineID)