 **MSA University**

**Faculty of Computer Science**

**Date: 10-05-2018**

**Dentist Clinic Desktop SYSTEM**

This project built to fulfill the requirements of the following courses:

* CS214: System Analysis and Design
* CS215: Fundamentals of Database Systems
* CS314: Object oriented software engineering

**Chapter One: Introduction**

# Background

Our system will facilitate and computerize dealing with patients’ data including their personal data like his name, address, mobile number, career and age to his medical data like patient history, health, prescription, treatment, diagnosis and analysis.

# Problem Statement

We want to solve the problems about patients’ data being lost, not well organized and registration that takes too long. Also solving the check accounting disability and expiry of materials such as (Rubber base spatula, Temporary crowns, Gloves, Mouse wash, and Cotton buds) without alerts.

# Objective

Our system will solve the problem of registration, lost data and accounting system of materials, its requirements and organizes the patient's most important details.

# System Definition

The system will manage and organize the patient's data including their treatment plan, diagnosis, tests, prescriptions, arrival of materials. Also, the system will take care of the employees arrival time and leave time , their incomes, their Sanctions.

# System goals

• Save information about patients.

• Save information about employees.

• Save information about materials such as (Rubber base spatula, Temporary crowns, Gloves, Mouse wash, and Cotton buds).

• Calculate information about the payment statement of each patient according to materials that he has been used.

• Save information about lab statement.

# Scope

Organize the flow of data by saving it and managing it to be used any time and to help employees in their work and make it easier to access patients' data and faster.

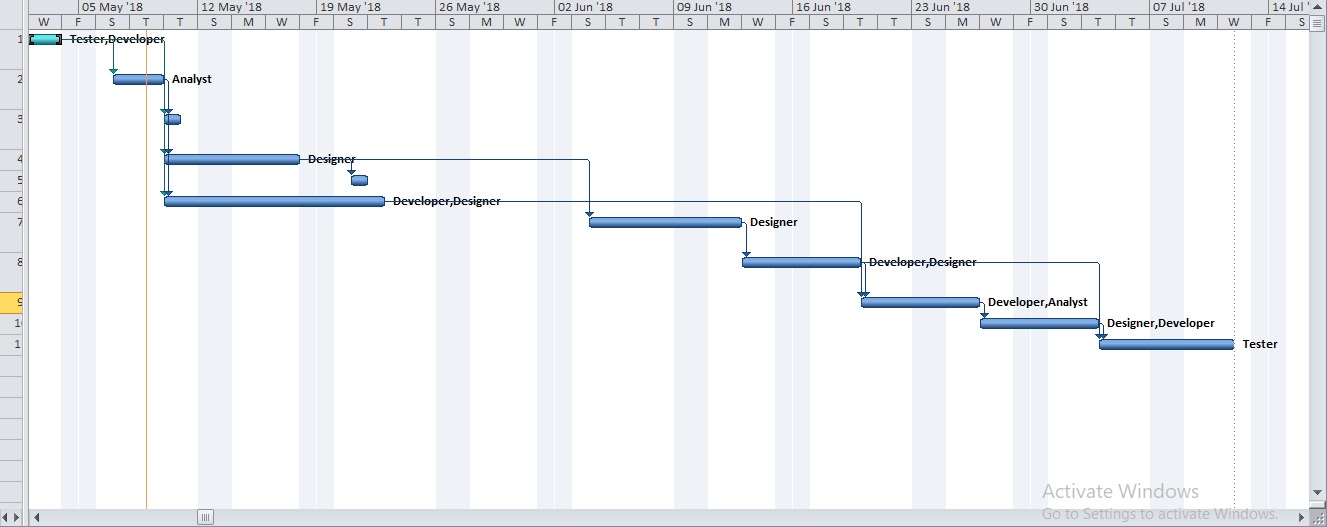
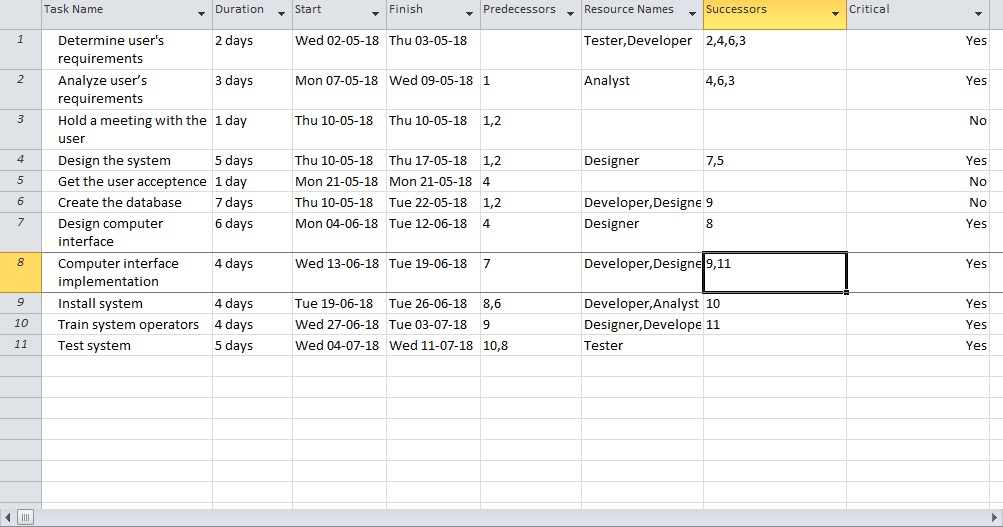
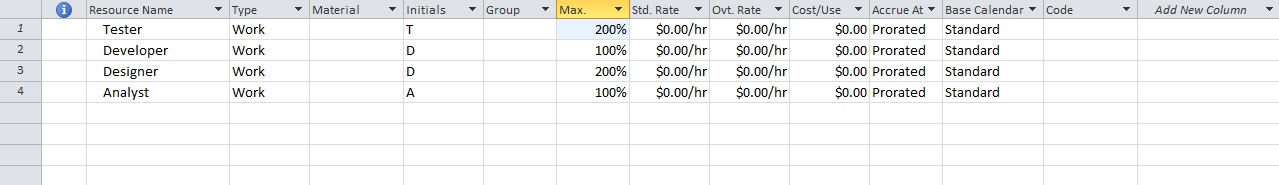
# Life Cycle Model

* Agile methodology will be used because the requirements will be edited according to the client’s needs. In addition to that all of functionalities will not be completed at the beginning of implementation because of the lack of time to start development and implementing the system. So, functionalities of the system will be evolving through the cycle of creating the system.
* Within the arrival of the patient to the clinic, the data about the patient is entered into the system which includes all the needed data about the patient’s personal information and health. During his appointment, the doctor is capable to update his records and he is also capable to write a medical prescription on the system, and it is made available for all the staff on the system to access. Throughout the year the system records all the financial information and all patients’ medical history and so it helps employees by providing them with the valuable alerts for the current patients' data and important materials that is needed for the current day or in the future.

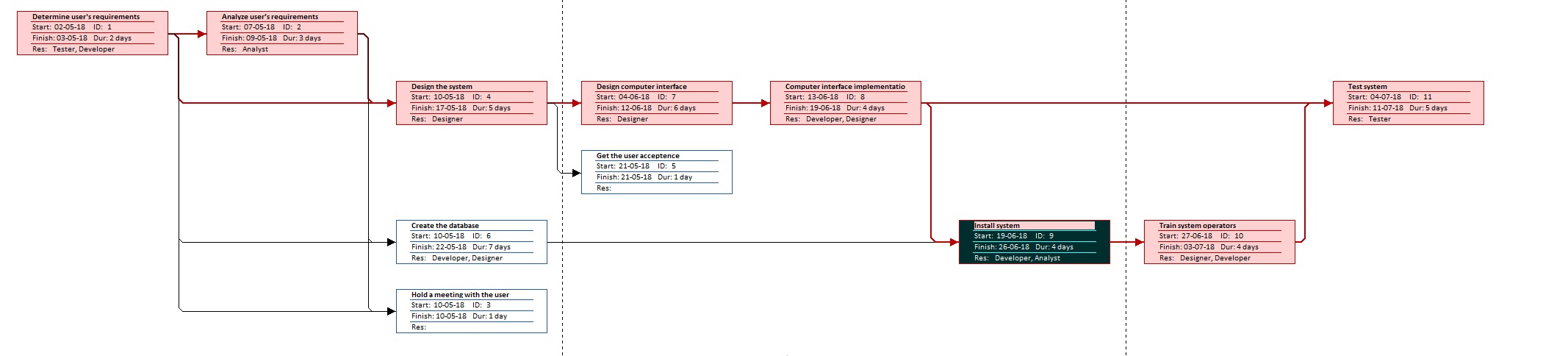
# Project Plan

* Hold a meeting with the client to make an initial agreement about developing a computerized system for the clinic.
* Interview with the client to gather his requirement and the details of the system.
* Analyze the client’s requirements and arrange the functionalities implementation according to its priority for the client.
* Take the client’s acceptance for the analysis and the requirements that gathered.
* Starting development of the screens and the database of the system.
* Testing the developed system and fixing any bugs and errors.
* Implementing the system.
* Maintenance the system.
* Close down the project.

# Project Gantt Chart



# Project Network Diagram



**Chapter Two: Requirements Extraction and Analysis**

* 1. **User Requirements**

Statements in a natural language, of what services the system is expected to provide

* + 1. ***Functional Requirements***
* Save patient’s information

\*When the patient enters the system if he/she is new, his history will be empty and anew page will be created. If the patient is existing on system, his history will be shown and his payment statement and his medical statement and his appointments and all the data necessary.

* Set information about employees

\*All the employees have a data base that have all data necessary about them like their jobs and salaries and vacations and work shifts and work times and weekends and rewards and salary deduction

* Set information about materials

\*The materials consist of two types: fixes materials and this type need maintenance every period of time. Disposables materials in this type we do not need maintenance but we need to order it every while.

* Set information about accounting statement

\*In this part of system, we focus on expenses and revenues and the profit

* Add and Update Patients:

\*Whenever a patient makes reservation, if it’s a new patient it creates a new ID for him/her and if it’s a current patient it updates its database and reservations.

* Add and Update Employees

\*Create new employees with new IDs and set the salary for them also update the salary for each current employee every once in a while.

* View attended Employees

\*For each day check attended employees and the time of their arrival and dismiss.

* Update Money entering from patients to the safe

\*Update the money entering from patients whether they have medical insurance or not.

* View Expenses

\*Calculate money which is going to be paid for the clinic’s employees, materials, location and taxes.

* Create Disposable or Fixed Materials
* Create Orders

\*Create a new order whether it’s for a disposable or fixed material

* Add Orders

\*Check for fixed and disposable materials for order dates.

* Update Materials

\*Update materials and set new order for almost out of stock materials or damaged materials.

* Create contracts with Suppliers
* Print Expenses, Profit, Total.
  + 1. ***Non-Functional Requirements***

1- Availability

The system is always available to use at any time, due to the running of the server all the time.

2- Security

Every access is different from priority of the account to the other.

Connected to the network with encrypted connection.

3- Friendly User Interface

The interface is easy to use with client end users, and easy to resemble.

4- Performance

The system uses high level fast programming language which makes it fast to response, due to its bandwidth which is 30GHz.

* + 1. ***Domain Requirements***1. The project will be done according to Egyptian Accounting regulations.
  1. **Key Assumptions**

1. Server will be provided and ready to operate and store the project’s data.
2. Client has the proper resources to run the software.
3. Tools and IDE is selected and provided.
4. There will be three people working on this project.
   1. **Project Constraints**
5. Two months to complete and finish the project.
6. Budget should not exceed L.E. 130,000.
   1. **Roles & Responsibilities**

|  |  |  |
| --- | --- | --- |
| **Client Side** | | |
| **Name** | **Role** | **Responsibilities** |
| Khaled El-Sawy | Client Project Manager | Sign-off the requirements |
| Ahmed Khaled | Business Consultant | Provide business requirements |
|  |  |  |
| **MSA Group No.** | | |
| **Name** | **Role** | **Responsibilities** |
| Asmaa El-Sayed | PM | * Project planning & Execution * Sign-off the requirements |
| Hedaya Ahmed | Business Analyst | Responsible for gathering Requirements |
| Ahmed Fayez | Developers | System development |
| Asmaa El-Sayed  Ahmed Fayez  Hedaya Ahmed | Tester | Testing the system |

* 1. **Requirement Validation**

1. Business and user specification document to be sent to PM for validation and signed-off.
2. UAT: to be send before project development phase.
3. Screenshots to be send and signed-off by the client.

**Chapter Three: System Analysis**

# Modeling System Requirements

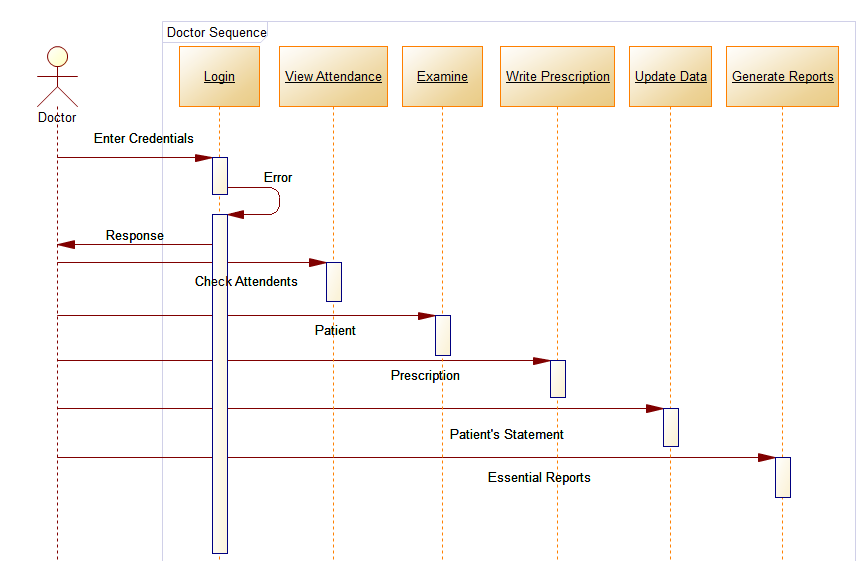
Include visual depictions of the functions and Users using the standard notation given in all the following diagrams.

### Use Case Diagram

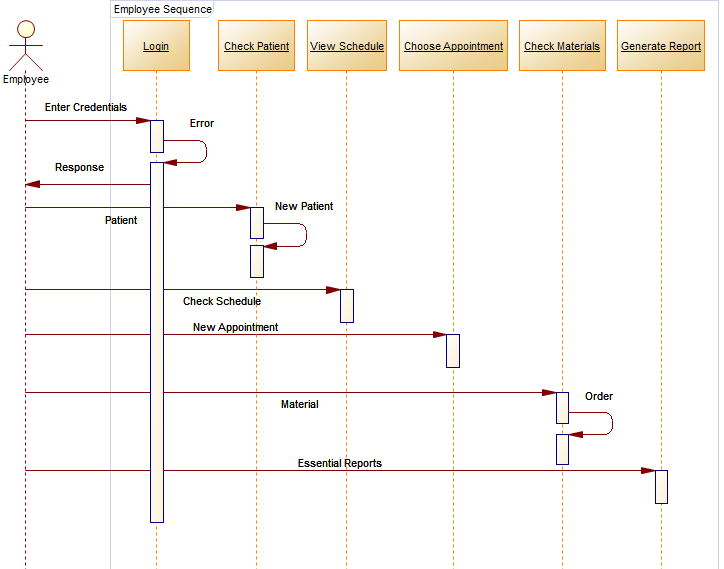


### Sequence Diagram

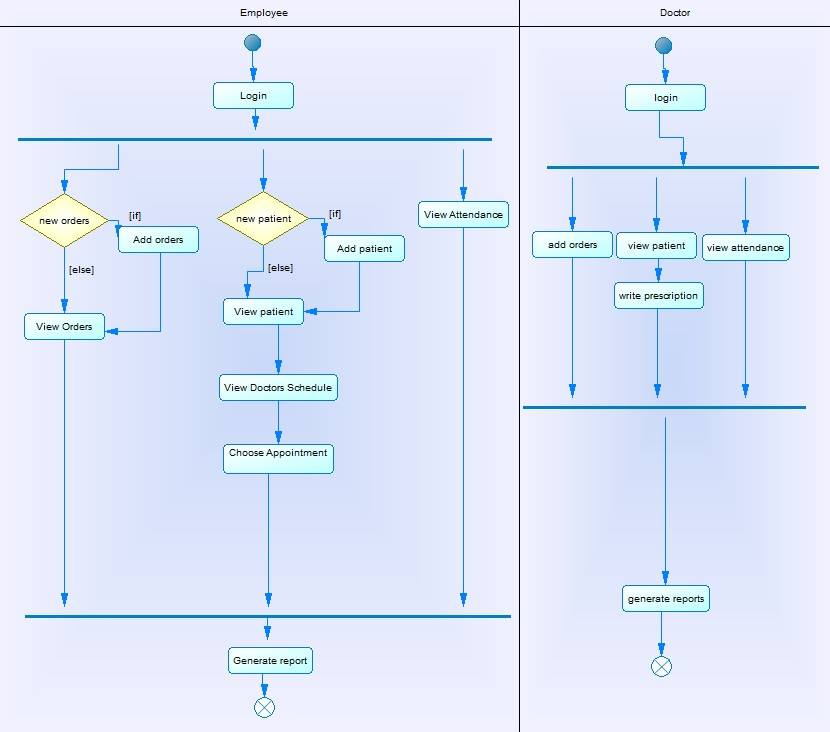
Doctor Sequence:



Employee Sequence:



### Activity Diagram



### Communication Diagram:

### 

### State Diagram

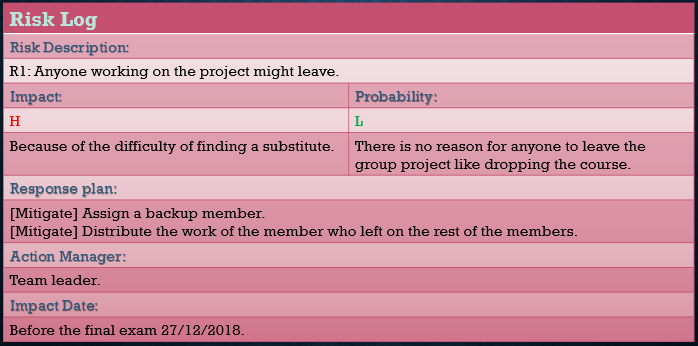
# A close up of text on a white background Description automatically generated

A close up of text on a black background

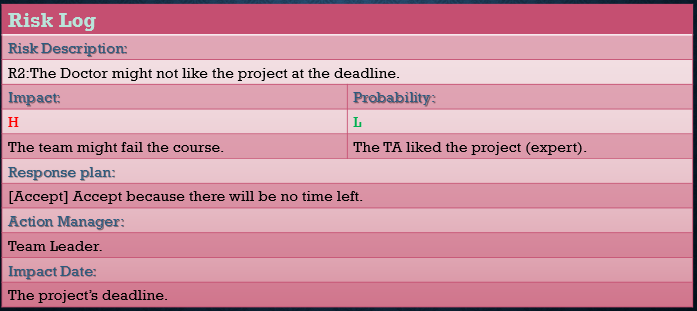
Description automatically generated

# Risk Management

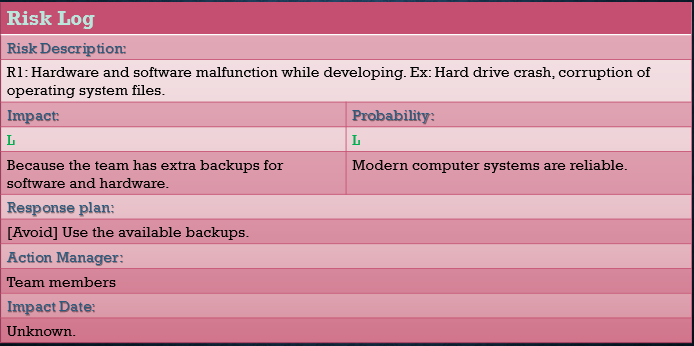
# 3.1 Risk Log



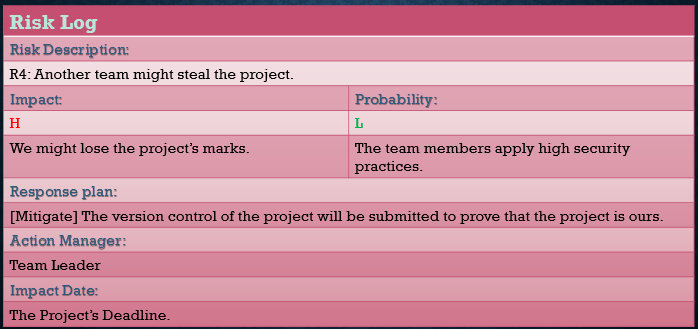
# 3.2 Risk Log



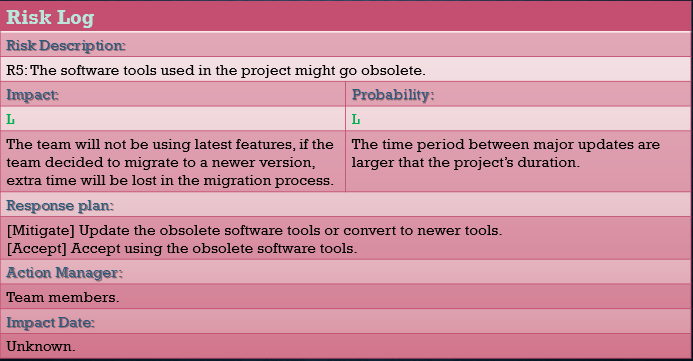
# 3.3 Risk Log



# 3.4 Risk Log



# Risk Log



**Chapter Four: System Design**

## ER Diagram & Conversion

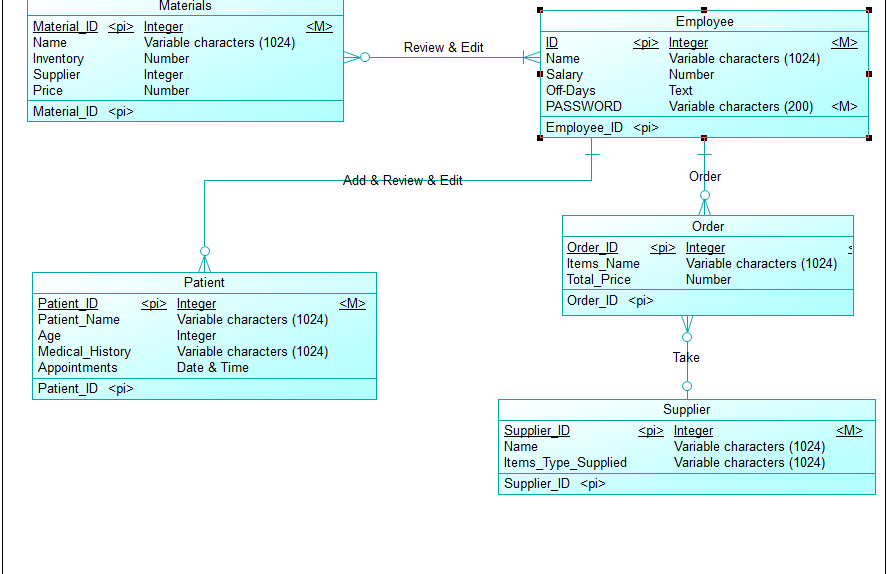
Use any powerful tool which allows entering Entity Relationship (ER) diagrams in a graphical form and produce physical database designs for popular relational database management systems such as Oracle.

### ERD

Use the notations given in the lectures or any other notations (ORACLE, ERWIN,VISIO, SmartDraw, PowerDesigner, DeZign, …)

Plot screenshots of the ERD Tables and the relationships between the tables.

### Conversion



## Data Definition & Manipulation Language (DDL & DML)

-- Database: `Clinic`

--

-- --------------------------------------------------------

--

-- Table structure for table `employee`

--

CREATE TABLE `employee` (

`ID` int(**11**) NOT NULL,

`NAME` varchar(**1024**) DEFAULT NULL,

`SALARY` decimal(**8**,**0**) DEFAULT NULL,

`OFF\_DAYS` text

) ENGINE=InnoDB DEFAULT CHARSET=utf32;

-- --------------------------------------------------------

--

-- Table structure for table `materials`

--

CREATE TABLE `materials` (

`MATERIAL\_ID` int(**11**) NOT NULL,

`NAME` varchar(**1024**) DEFAULT NULL,

`INVENTORY` decimal(**8**,**0**) DEFAULT NULL,

`SUPPLIER` int(**11**) DEFAULT NULL,

`PRICE` decimal(**8**,**0**) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf32;

-- --------------------------------------------------------

--

-- Table structure for table `order`

--

CREATE TABLE `order` (

`ORDER\_ID` int(**11**) NOT NULL,

`ID` int(**11**) NOT NULL,

`SUPPLIER\_ID` int(**11**) DEFAULT NULL,

`ITEMS\_NAME` varchar(**1024**) DEFAULT NULL,

`TOTAL\_PRICE` decimal(**8**,**0**) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf32;

-- --------------------------------------------------------

-- Table structure for table `patient`

CREATE TABLE `patient` (

`PATIENT\_ID` int(**11**) NOT NULL,

`ID` int(**11**) NOT NULL,

`PATIENT\_NAME` varchar(**1024**) DEFAULT NULL,

`AGE` int(**11**) DEFAULT NULL,

`MEDICAL\_HISTORY` varchar(**1024**) DEFAULT NULL,

`APPOINTMENTS` datetime DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf32;

-- --------------------------------------------------------

-- Table structure for table `review\_\_\_edit`

--

CREATE TABLE `review\_\_\_edit` (

`ID` int(**11**) NOT NULL,

`MATERIAL\_ID` int(**11**) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf32;

-- --------------------------------------------------------

--

-- Table structure for table `supplier`

--

CREATE TABLE `supplier` (

`SUPPLIER\_ID` int(**11**) NOT NULL,

`NAME` varchar(**1024**) DEFAULT NULL,

`ITEMS\_TYPE\_SUPPLIED` varchar(**1024**) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf32;

--

-- Indexes for dumped tables

--

--

-- Indexes for table `employee`

--

ALTER TABLE `employee`

ADD PRIMARY KEY (`ID`);

--

-- Indexes for table `materials`

--

ALTER TABLE `materials`

ADD PRIMARY KEY (`MATERIAL\_ID`);

--

-- Indexes for table `order`

--

ALTER TABLE `order`

ADD PRIMARY KEY (`ORDER\_ID`),

ADD KEY `FK\_ORDER` (`ID`),

ADD KEY `FK\_TAKE` (`SUPPLIER\_ID`);

-- Indexes for table `patient`

--

ALTER TABLE `patient`

ADD PRIMARY KEY (`PATIENT\_ID`),

ADD KEY `FK\_ADD\_\_\_REVIEW\_\_\_EDIT` (`ID`);

--

-- Indexes for table `review\_\_\_edit`

--

ALTER TABLE `review\_\_\_edit`

ADD PRIMARY KEY (`ID`,`MATERIAL\_ID`),

ADD KEY `FK\_REVIEW\_\_\_EDIT2` (`MATERIAL\_ID`);

--

-- Indexes for table `supplier`

--

ALTER TABLE `supplier`

ADD PRIMARY KEY (`SUPPLIER\_ID`);

--

-- Constraints for dumped tables

--

--

-- Constraints for table `order`

--

ALTER TABLE `order`

ADD CONSTRAINT `FK\_ORDER` FOREIGN KEY (`ID`) REFERENCES `employee` (`ID`),

ADD CONSTRAINT `FK\_TAKE` FOREIGN KEY (`SUPPLIER\_ID`) REFERENCES `supplier` (`SUPPLIER\_ID`);

--

-- Constraints for table `patient`

--

ALTER TABLE `patient`

ADD CONSTRAINT `FK\_ADD\_\_\_REVIEW\_\_\_EDIT` FOREIGN KEY (`ID`) REFERENCES `employee` (`ID`);

--

-- Constraints for table `review\_\_\_edit`

--

ALTER TABLE `review\_\_\_edit`

ADD CONSTRAINT `FK\_REVIEW\_\_\_EDIT` FOREIGN KEY (`ID`) REFERENCES `employee` (`ID`),

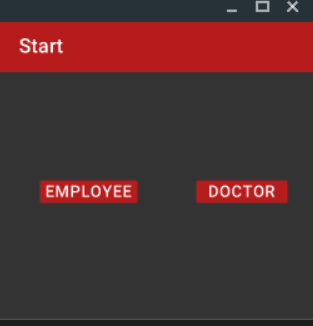
ADD CONSTRAINT `FK\_REVIEW\_\_\_EDIT2` FOREIGN KEY (`MATERIAL\_ID`) REFERENCES `materials` (`MATERIAL\_ID`);

COMMIT;

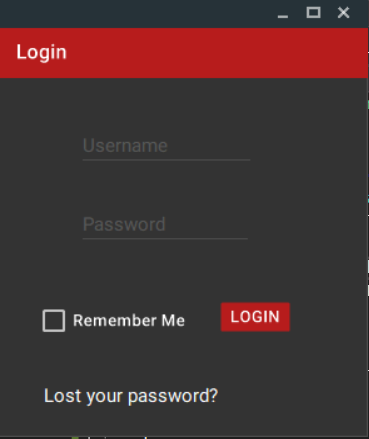
**Chapter Five: Implementation**

## Graphical User Interface

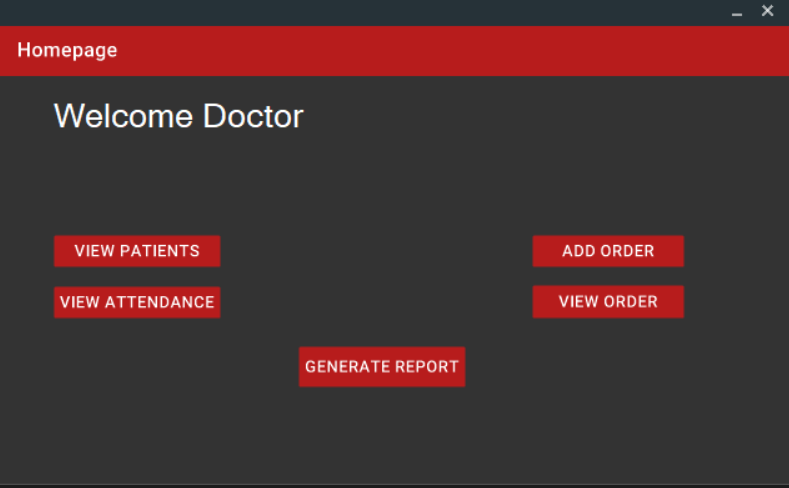
**Start:**

****

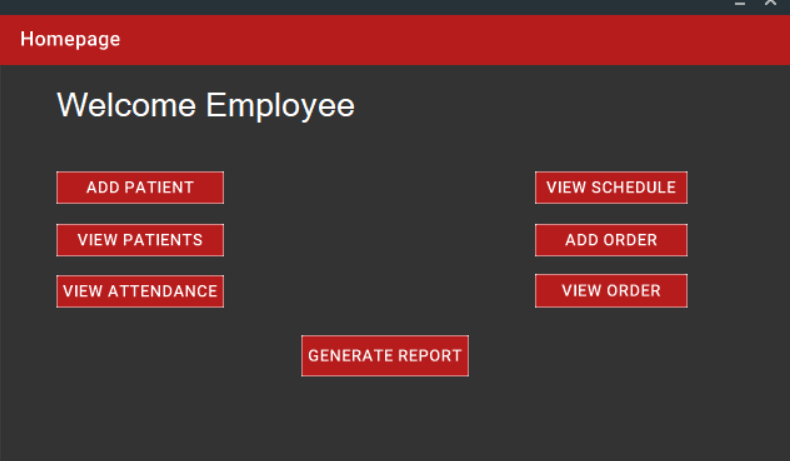
**Login:**

****

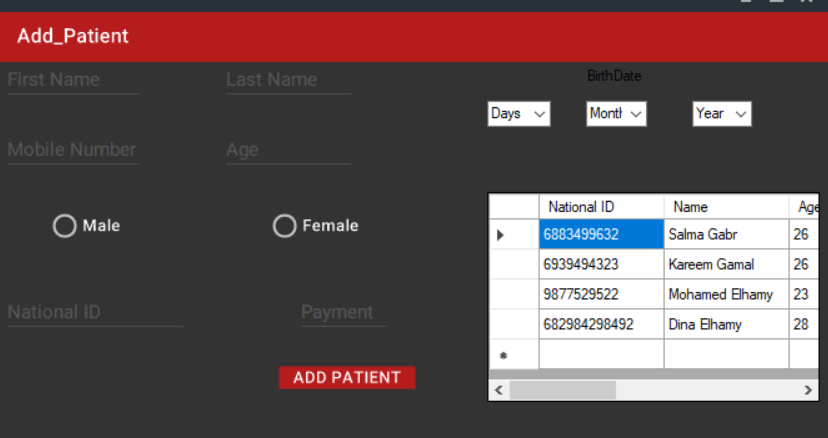
**Home Page Doctor:**

****

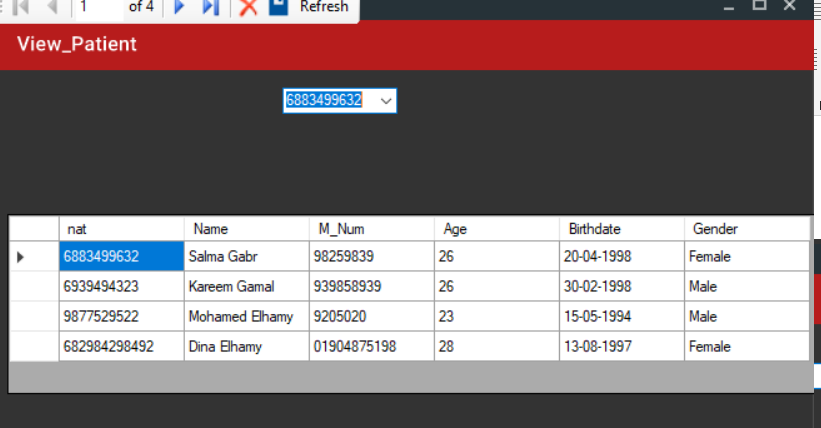
**Home Page Employee:**

****

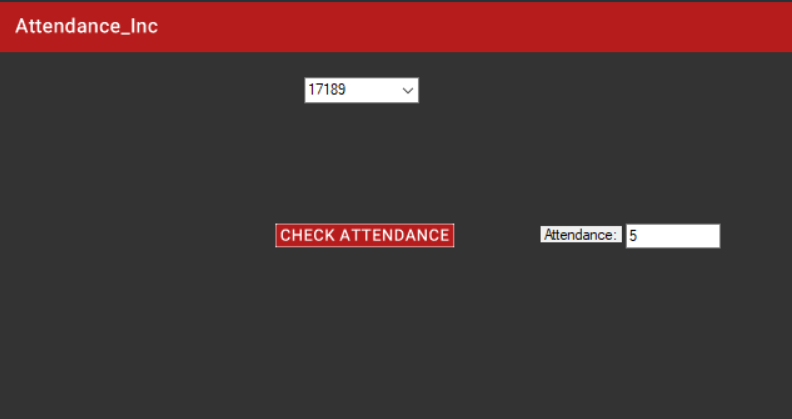
**Add Patient:**

****

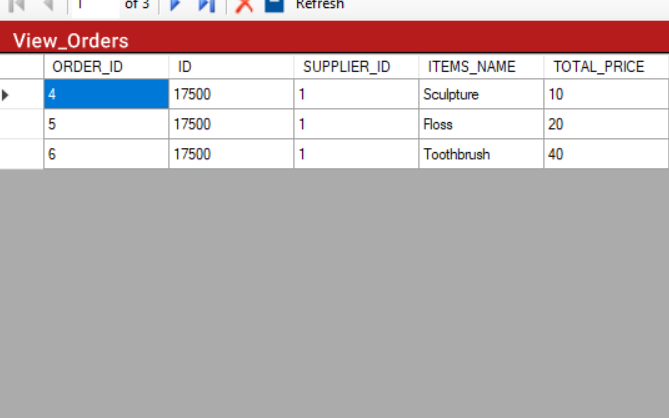
**View Patient:**

****

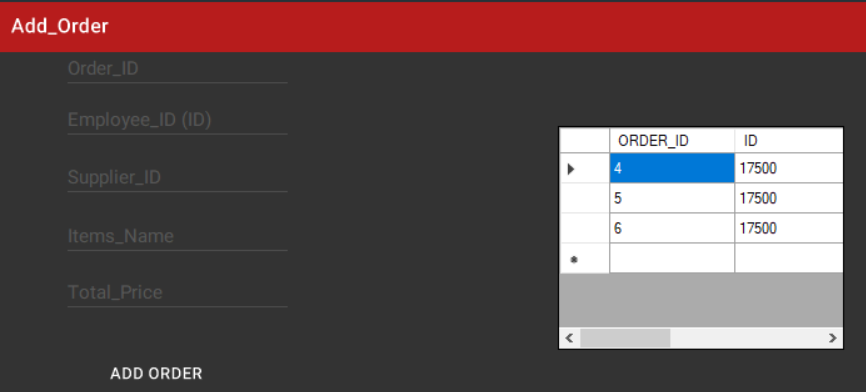
**View Attendance:**

****

**View Order:**

****

**AddOrder :**

****