Information Systems and Data Modeling – IT1090

Assignment

Title: Online Fitness Trainer	
Batch Number: Y1.S2.WD.IT.01	Group Number: MLB_01.01_06

Declaration:

We hold a copy of this assignment that we can produce if the original is lost or damaged.

We hereby certify that no part of this assignment has been copied from any other group's work or from any other source. No part of this assignment has been written / produced for our group by another person except where such collaboration has been authorized by the subject lecturer/tutor concerned.

Group Members:

IT21808548	Gimhan T.M. S	shivanka
IT21808852	Peiris W.A.D	Achala
IT21829710	P.A.E.U.K Perera	epuni
IT21828966	Liyanage C.H	chamodi
IT21828720	Kularathna E.M.S.D	sanduni

Submitted on: 30/10/2022

Table Of Contents

1. Hypothetical Scenario
2. Requirements Analysis Document
3. Data Requirements
4. Entity Relationship(ER) Diagram
5. Relational Schema
6. Database
7. Special Performance Considerations
8. Special Security Requirements

1.Hypothetical Scenario

An online fitness trainer has customers. A customer has a first name, last name, gender, date of birth, address, email, password, age, and Registration Number. The registration number is unique for each customer and the age is derived from the date of birth. The system offers multiple programs. A program contains a program code which is unique, the program name, price, and duration. A customer can follow multiple programs. Programs are delivered by trainers. A trainer has an email, password, address, contact number, first name, last name, and a unique trainer id. The trainers have a manager who manage them. A trainer can deliver multiple programs. The customer can make payments. Each payment contains the date, amount and a unique payment id.

2.Requirement Analysis Document

Functional Requirements

Customer:

- A customer who uses the system for the first time must fill the registration form with the details: name, date of birth, email, password, gender and address. An account will be created upon submission.
- Login to system using email and password.
- Guest users who are not logged in can only view the program list page, contact page, about page and the program content will not be displayed.
- Logged in customer can view the program content in the fitness trainer.
- Logged in customer can buy and add new programs to follow.
- All customers can use the contact page to communicate with the team
- Logged in customer can change settings of the account: update any of the data taken at registration.
- Logged in customer can log out when needed.

Trainer:

- A Trainer who uses the system for the first time must fill the registration form with the details: name, email, password, contact Number and address. An account will be created upon submission.
- Trainer can login to system using email and password.
- Trainer can view and change the program content in the fitness trainer.
- Trainer can add new programs to the fitness trainer
- Trainer can change settings of the account: update any of the data taken at registration.
- Trainer can view the customers' messages sent from contact page and reply to it.
- Logged in trainer can log out when needed.

Non-Functional Requirements

- Availability-System must be available at any given time.
- Performance- System must have a fast response time not exceeding more than 20 seconds.
- Speed- Each page should take no longer than 7 seconds to load.
- Security- System should store data in a protective method.
- Usability- Navigation menus should be simple to understand.
- Scalability- System should be able to handle 5000 users at a time.

3.Data Requirements

Customer(Registered)

- Registration number
- Name (first and last)
- Email
- Password
- Address
- Date of Birth
- Gender
- Age

Trainer

- Trainer Id
- Name(First and Last)
- Email
- Password
- Contact Number
- Address
- Manager Id

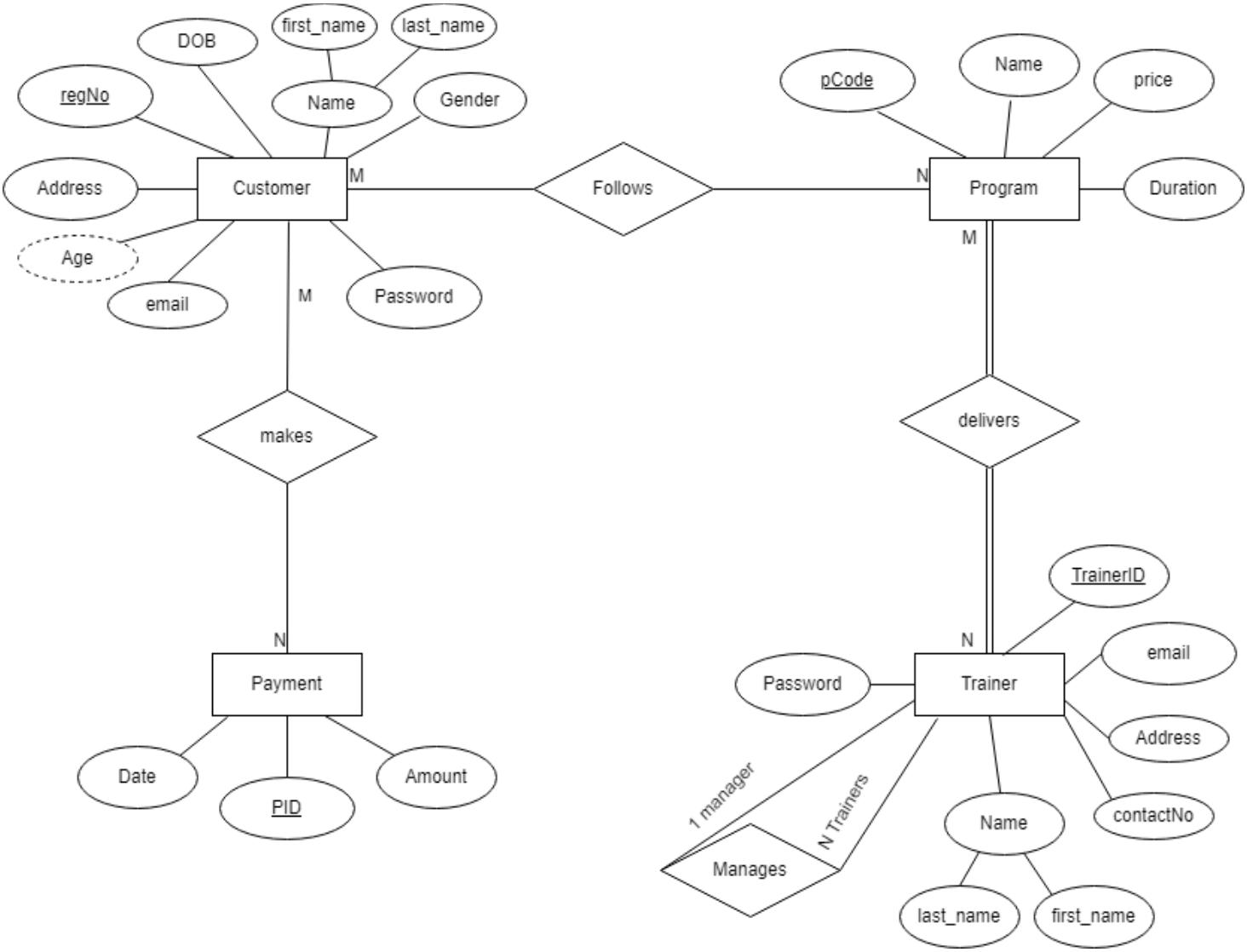
Payment

- Payment Id
- Date
- Amount

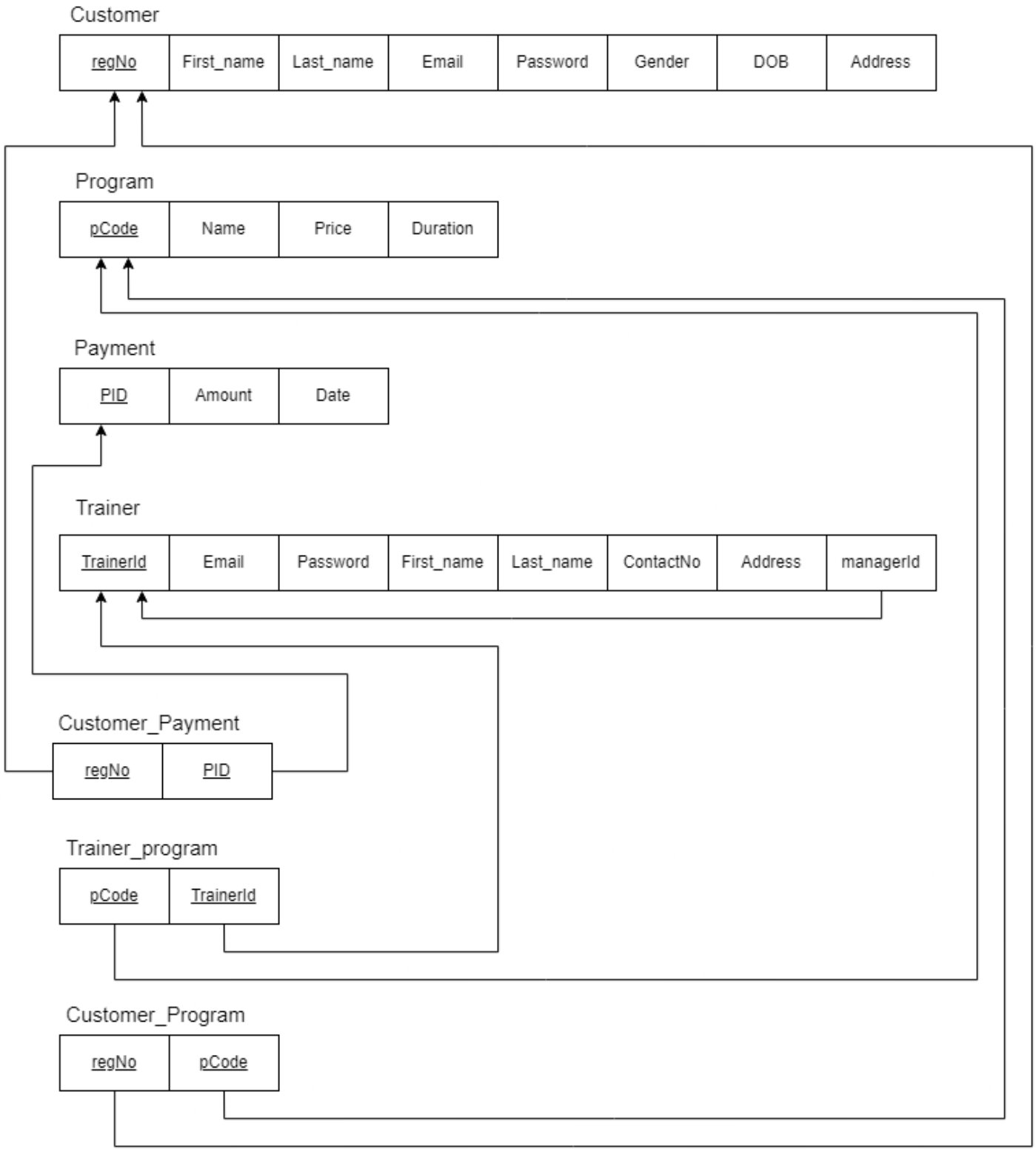
Program

- Program Code
- Name
- Price
- Duration

4. Entity Relation Diagram



5.Relational Schema



6. Database

```
CREATE TABLE Customer
(
  regNo INT,
  First_name VARCHAR(50),
  Last_name VARCHAR(50),
  email VARCHAR(100),
  Password VARCHAR(50),
  gender VARCHAR(10),
  DOB DATE,
  address VARCHAR(200),
  constraint Customer_PK PRIMARY KEY (regNo)
);

CREATE TABLE Program
(
  Pcode INT,
  Name varchar(50),
  Price INT,
  Duration INT,
  constraint Program_PK PRIMARY KEY (Pcode)
);

CREATE TABLE Payment
(
  PID INT,
  Amount INT,
  Date DATE,
  constraint Payment_PK PRIMARY KEY (PID)
);

CREATE TABLE Trainer
(
  TrainerId INT,
  email VARCHAR(100),
  Password VARCHAR(50),
  first_name VARCHAR(50),
  last_name VARCHAR(50),
  contactNo INT,
  address VARCHAR(200),
  managerId INT,
  constraint Trainer_PK PRIMARY KEY (TrainerId),
  constraint manager_FK FOREIGN KEY (managerId) REFERENCES Trainer(TrainerId)
);

CREATE TABLE Customer_program
(
  Pcode INT ,
  regNo INT ,
  constraint cusPro_PK PRIMARY KEY (Pcode, regNo),
  constraint cusPro_FK1 FOREIGN KEY (Pcode) REFERENCES Program(Pcode),
  constraint cusPro_FK2 FOREIGN KEY (regNo) REFERENCES Customer(regNo)
);

CREATE TABLE Trainer_program
(
  TrainerId INT,
  Pcode INT,
  constraint trainPro_PK PRIMARY KEY (TrainerId, Pcode),
  constraint trainPro_FK1 FOREIGN KEY (TrainerId) REFERENCES Trainer(TrainerId),
  constraint trainPro_FK2 FOREIGN KEY (Pcode) REFERENCES Program(Pcode)
);
```

```
CREATE TABLE Customer_payment
(
  regNo INT,
  PID INT,
  constraint cusPay_PK PRIMARY KEY (regNo, PID),
  constraint cusPay_FK1 FOREIGN KEY (regNo) REFERENCES Customer(regNo),
  constraint cusPay_FK2 FOREIGN KEY (PID) REFERENCES Payment(PID)
);

insert into Customer values(1,'jack','daniels','jack@gmail.com','jack@123','Male','12/03/2003','123
road 1');
insert into Customer values(2,'Kamal','perera','kamal1@gmail.com','kamal@123','Male','20030723','78
road 16');
insert into Customer values(3,'devi','silva','devi@gmail.com','devi@123','Female','20000809','147,main
street');
insert into Customer
values(4,'Saman','Mendis','Saman1@gmail.com','Samanrocks@123','Male','20020718','16,2nd street');
insert into Customer
values(5,'Alice','fernando','alice24@gmail.com','alice@123','Female','20040513','78 road 16');

insert into Program values(1,'Heavy',3000,1);
insert into Program values(2,'Light',2000,1);
insert into Program values(3,'Diet',2500,2);
insert into Program values(4,'beginner',5000,3);
insert into Program values(5,'legs',1000,1);


insert into Trainer values(1,'jacob@gmail.com','jacob123','jacob','lee','771234567','133,chilaw rd',1);
insert into Trainer values(2,'kamala@gmail.com','kamala123@456','kamala','perera','612345643','136,4th
Street',1);
insert into Trainer values(3,'jonas@gmail.com','jonas123','jonas','peiris','76123894','166/41,kandy
rd',1);
insert into Trainer values(4,'Khalid@gmail.com','khalid123','khalid','hassim','736942012','445, main
street',1);
insert into Trainer values(5,'Raj@gmail.com','Raj123','Raj','Thomas','78340960','87,Negombo rd',1);

insert into Payment values(1,1500,'20221012');
insert into Payment values(2,11500,'20221012');
insert into Payment values(3,3000,'20221014');
insert into Payment values(4,5000,'20221015');
insert into Payment values(5,7000,'20221016');



insert into Customer_program values(1,2);
insert into Customer_program values(2,1);
insert into Customer_program values(3,4);
insert into Customer_program values(4,2);
insert into Customer_program values(5,5);

insert into Customer_payment values(1,4);
insert into Customer_payment values(1,5);
insert into Customer_payment values(2,3);
insert into Customer_payment values(3,1);
insert into Customer_payment values(4,2);


insert into Trainer_program values(1,5);
insert into Trainer_program values(2,3);
insert into Trainer_program values(3,2);
insert into Trainer_program values(4,1);
insert into Trainer_program values(5,4);
```

7. Special Performance Considerations

- System must have enough resources, hardware, and software wise, to process 5000 user requests at once.

8.Special Security Requirements

Access Levels

- Customer- Read / Update (limited)
- Trainer- Read/Update

General

- Database should be backed up twice per day and backup should be in 2 different places.
- Customer data should be stored with highest security.
- Should not share customer data with third parties.