Blooming Fitness Center

Group 5

By: Jacob Mikkola, Jordan Bailey, Katherine Ollearis, Kevin Ratsavong, Pritam Shyam

CONTENTS

BUSINESS CASE DESCRIPTION

Blooming Fitness Center has hired us to create a database management system to assist in running their business. After interviews with the owners we have developed the following list of attributes.

- BFC keeps track of past and present personnel. They record employee name, address, phone number, their hire date, position and their status of current or former. Each employee also has a four-digit employee ID. An employee can be either an instructor, clerk or administrative personnel. BFC would like to record an hourly rate for all clerks. For all fitness instructors BFC would like to know their qualifications and when the date they became qualified.
- All members of the BFC have a unique six-digit member ID. BFC also records name, address, their home and cell phone number, gender, age, DOB, and the date they joined BFC. Each member pays a one-time fee that is a set amount ($500, $300, $100, $50). This fee gives the members discounts based on the price of the fee. If a member paid $500, there discount will be 30% and if they paid $50, there will be no discount.
- BFC has over thirty fitness courses. Each course has a unique course number, course name, description, course difficulty, duration (number of weeks it runs), and a fee (the price of the course before discounts).
- Courses are offered as classes. Each class has a reference number, a unique course number and reference number. Also recorded for each class is the start date for the class, the days of the week it will be held, the time period and the instructor for the class.
- Each room has a unique room number, class capacity and room capacity (the max the room can accompany).
- Some rooms contain various equipment's. Each equipment has a serial number and BFC would like to record purchase date, the date of its last maintenance and its equipment type.
- BF has its own in center store and would like to keep track of the items sold in it. Each item has an item number, description, standard price (before discount), quantity on hand and reorder point. If an item falls below the reorder point it is marked for reorder from the vendor.

- After every purchase a 7-digit order number is generated, and the member making the purchase and date of transaction are recorded as well as the item purchased and quantity of each item with the selling price per unit. For each purchase the total is calculated.

BUSINESS RULES

Employee:

Blooming Fitness Center employs many different types of employees. All BFC employees are assigned a four-digit employee ID. BFC keeps track of current and former employees. Only Clerk employees have records of negotiated hourly rate. Employees are deemed qualified once approved by John or Joan Smith. An Instructor can teach none or many courses with or without qualifications.

Member:

All new members are assigned a Member ID that is associated with their membership type. BFC does not allow upgrades to higher level memberships. A member requesting to change their membership type must rejoin and be assigned a new Member ID. A member that changes membership type must pay the full joining fee. Members must choose one of the four membership types when joining. Each membership will last a lifetime. Each membership will have a discount rate percentage associated with their course and item purchases.

Course:

BFC keeps track of many personal exercise, fitness and nutritional courses. Each course must have an instructor assigned to it, but an instructor does not have to be qualified to teach the course. A member is allowed to enroll in any course. A course will be saved even if it is not being offered at this period in time.

Class and Room:

A class is a specific offering from a course. Each class will have exactly one instructor teaching the course. A schedule will contain an alternate instructor for each class in case of emergency situations. Each class must be assigned exactly one room. Class enrollment size must be less than room capacity.

Equipment:

Exercise equipment will not be moved and will remain in the room it is assigned. All Equipment must contain a Maintenance Log. Equipment will be stored as Equipment Type and categorized according to type.

Merchandise and Purchase:

BFC will maintain an inventory of items available for purchasing for employees and members. Every item purchase will be tracked and an inventory of remaining items will be stored. Every item will be assigned a reorder point and lead time to maintain adequate stock of each item. Only BFC members and employees can purchase items from the BFC store.

SUPPORTING DOCUMENTATION

Employee - Table has information about the employees, such as their employee id, name, address, phone number, hired date, position, status and their employee type. It contains:

- EmployeeID - Is the unique sequential four- digit number assigned to them when they are originally hired.
- EmployeeFirstName - First Name of the Employee
- EmployeeLastName - Last Name of the Employee
- EmployeeStreet - The employee's street address
- EmployeeState - The State the employee resides in
- EmployeeZip - The zip code of the employee
- EmployeeHomePhone - The employee's home phone number
- EmployeeCellPhone - The Employee's Cellphone number
- EmployeeHiredDate - Is the date the employee was hired
- EmployeeStatus – The Employee may include employees that may by past or present employees. Status is whether an employee is a current or former employee
- EmployeePositionType – Blooming Fitness Center employs a number of different types of employees:
    - Fitness Course Instructors – Qualifications and date qualified.
    - Clerks - Negotiated hourly rate

- Administrative Personnel - Admin certification

Member - table stores data specific to each member of the Blooming Fitness Center. This table has member's information such as; Member ID, name, address, phone number, gender, age, date of birth, date joined, and their Membership type:

- MemberID – Each member will have a unique six-digit identification number. Will be used to identify them.
- MemberFirstName- The first name of the member
- MemberLastName- The last name of the member
- MemberStreet - The member's street address
- MemberState - The State the member resides in
- MemberZip - The zip code of the employee
- MemberHomeNumber - The home number of the member
- MemberCellPhone - The Cell number of the member
- MemberGender - The gender of the member
- MemberAge - The age of the member
- MemberBirthDate - The date of birth of the member
- MemberJoinDate - The date which the specified member joined the BFC.
- MemberOneTimeFee - This is a one-time fee that established lifetime membership.
- MembershipDiscount - Each member has a discount based on their one-time fee.

Course - has information on the fitness courses that the Blooming Fitness center can provide. BFC offers over 30 different courses including yoga, tai-chi, weight training and aerobics and also other fitness-oriented topics such as nutrition and natural medicine. No prerequisites needed.

- CourseID - Is the unique course number assigned when a course is made.
- CourseName - Is the name of the Course, such as "weight training for beginners"
- CourseDescription - Is a brief description of the course.
- CourseDuration- Duration is how long the class will take place. The number of weeks that a course is designed to be run.
- CourseFee - The price of the course prior to discounts.

- CourseDifficulty - Category can fall under 3 categories; Novice, Intermediate, and Advanced

Class - is a specific offering of a course. Courses are usually offered periodically in the form of classes.

- ClassNumber - Each class has an associated reference number, which is not necessarily unique.
- ClassStartDate - The start date of the class
- ClassDays - is the days of the week the class will be held (e.g., M, M/W/F, Th/F)
- ClassTimePeriod - The time period the class will take place
- ClassCapacity - how many people are allowed into the class

Schedule - is an associate entity that connects instructors to classes.

- EmployeeID – Used to identify the instructor.
- ClassNumber – Used to obtain the class number for the above instructor.

Enrolled – is an associative entity that connects members to classes.

- MemberID – Used to identify the member.
- ClassNumber – Used to obtain class number for the above member.

Room - has a unique sequential room number and a class capacity associated with it. Some rooms can contain pieces of exercise equipment, which are not moved across the rooms.

- RoomNumber - The room number of the room which can be used to identify how much is the capacity of the room.
- RoomCapacity – The maximum room accompany number.

Equipment - contains information about all of the equipment within BFC including, the serial numbers for each piece of equipment, the model name of the equipment, the date it was purchased, the most recent maintenance performed on each piece of equipment, and the type category of each piece of equipment.

- EquipmentSerial - a unique identifier (provided by the manufacturer) that is used to track a piece of equipment in the fitness center.

- EquipmentName - specific name and model of the equipment.
- EquipmentPurchaseDate - the date the piece of equipment was purchased.
- EquipmentTypeMaintenanceDate - the most recent date that maintenance was performed on the equipment.
- EquipmentType - a category of equipment (treadmill, elliptical, etc.)

Merchandise - contains information about items that are for sale at BFC including, an item ID for each item that is for sale, the description of that item, the standard retail price of that item, the inventory levels of each item, and the reorder point for each item.

- ItemID - a unique number that identifies an item for sale.
- ItemDescription - description of the item for sale.
- ItemStandardPrice - retail price of the item for sale before discounts.
- ItemQuantity - the amount of a specific item in stock.
- ItemRe-OrderThreshold - the minimum quantity of an item that should be kept on hand. If an item's inventory level falls below this point, the item is noted for reorder from the vendor that supplies it.

Purchase - contains information about purchasing transactions between BFC and its members including, an order number for each order placed, the name of the member placing the order, the date the order was placed, the amount of each item sold, the selling price of each item sold, and the overall purchase total for all items sold.

- OrderNumber - a unique seven-digit number that designates an order has been placed by a BFC member.
- DateofTransaction - the date that the purchase was made.

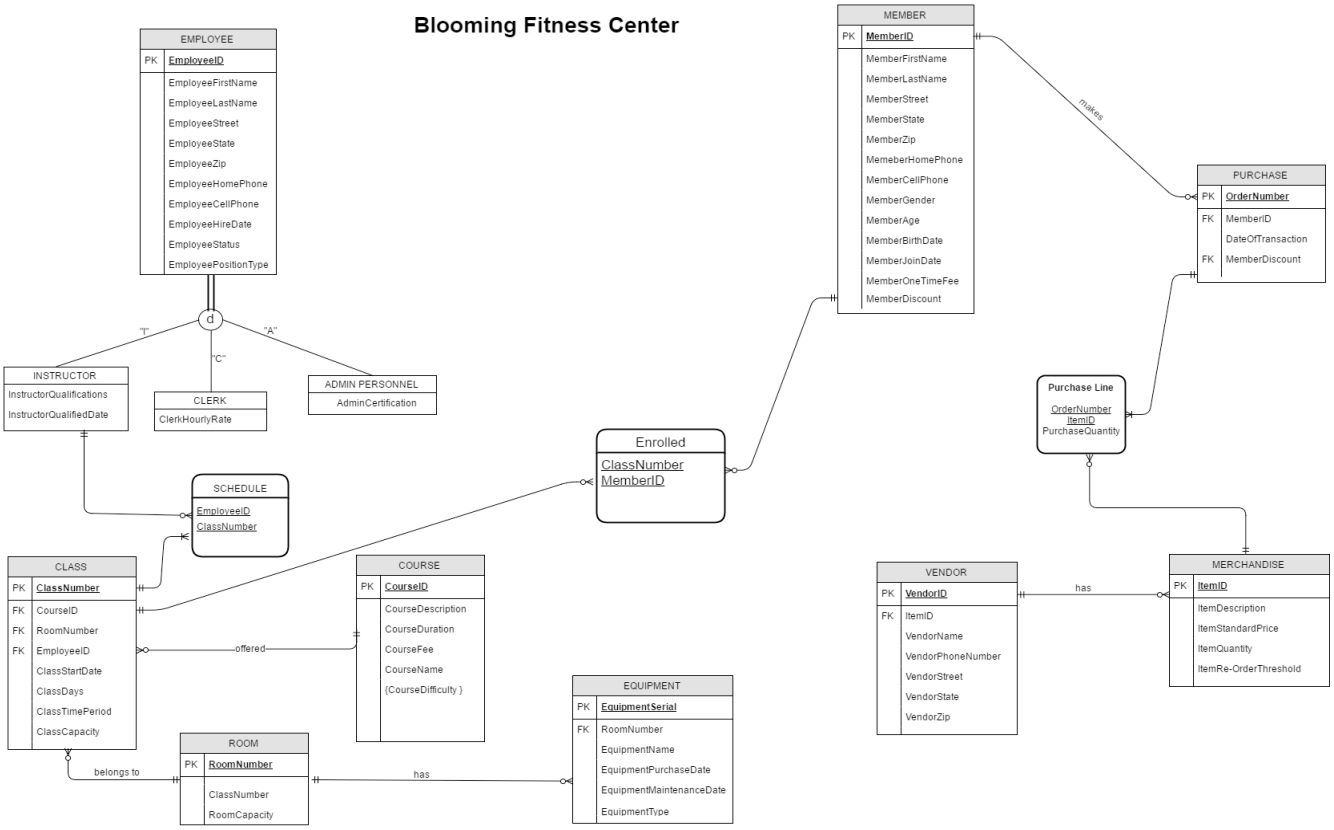
Purchase Line - is an associate entity that connects purchase to merchandise.

- OrderNumber – the order number for the purchase.
- ItemID – used to retrieve information about the item
- PurchaseQuantity – quantity of the item in purchase.

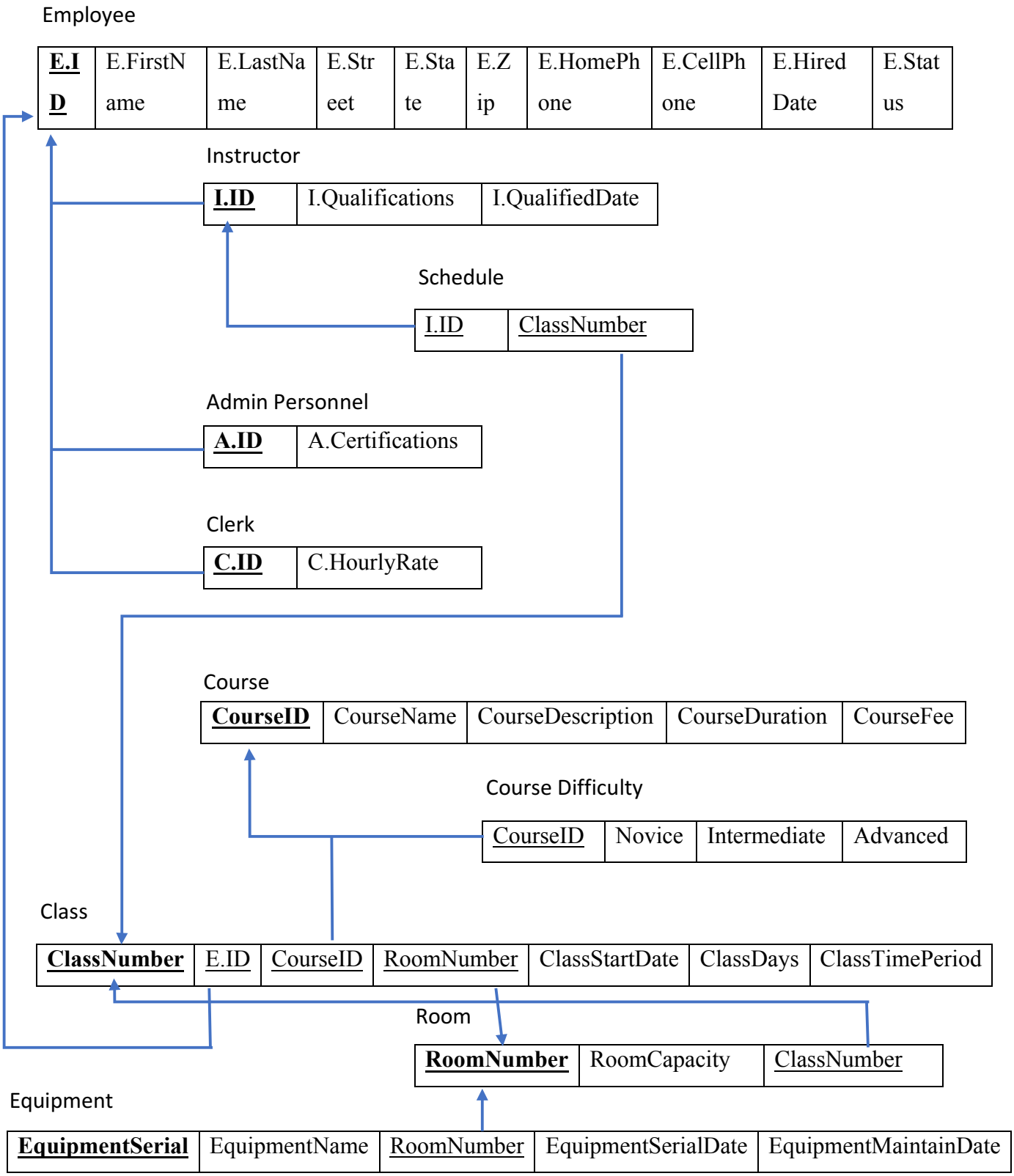
Vendor - This contains the vendor information required to contact them when the item from Merchandise is at re-order quantity.

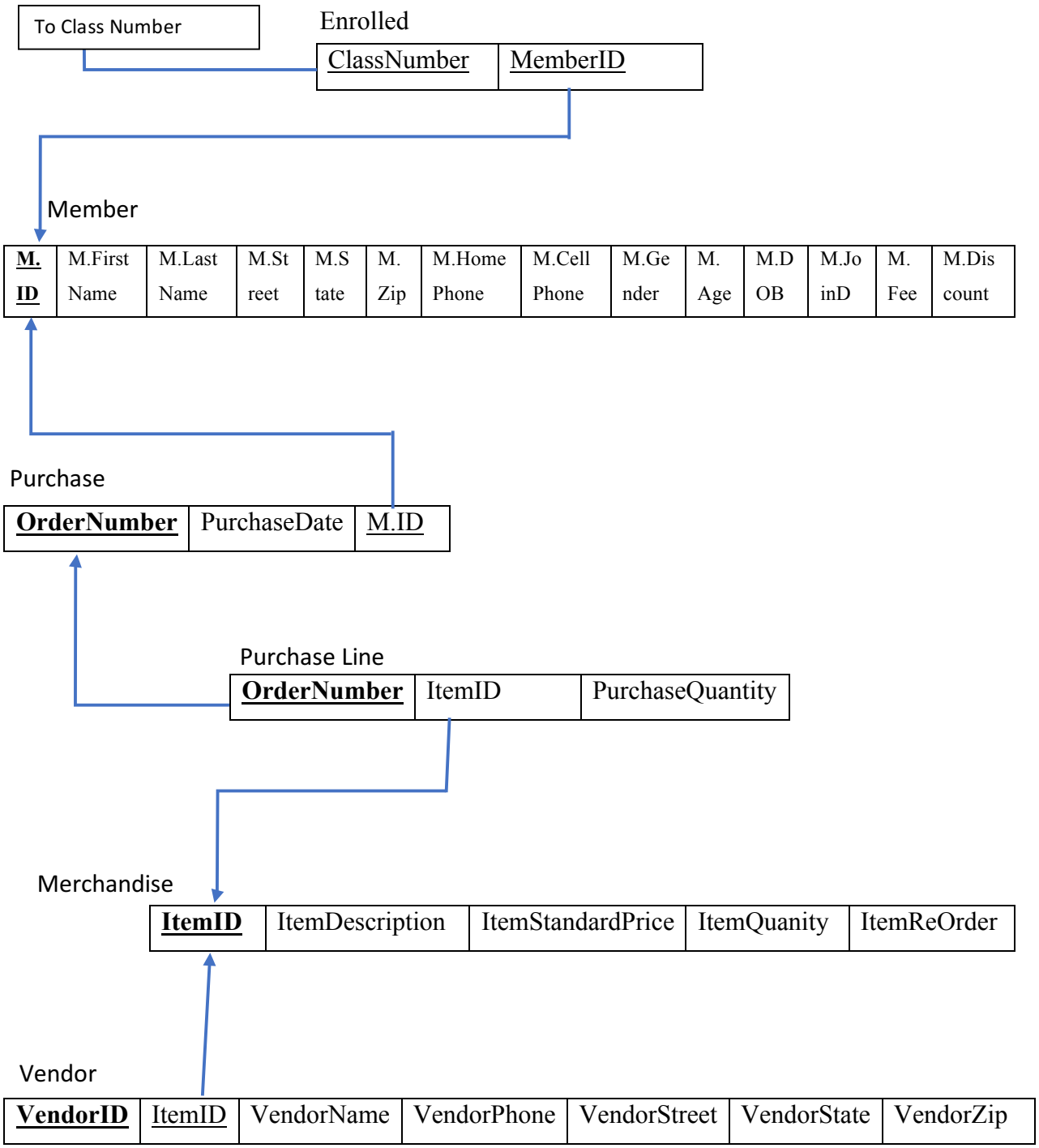
- VendorID - unique fix digit number identifying a supplier
- VendorName - given name of the supplier
- VendorPhoneNumber - contact number of supplier
- VendorStreet - street address of vendor
- VendorState - state of the vendor
- VendorZip - zip code for the vendor

ENTITY DIAGRAM



RELATIONAL SCHEMA





Underline = Foreign Key

Bold & Underline = Primary Key

DATA TYPES

Employee Table:

- EmployeeID – Numeric (4)
- EmployeeFirstName – VarChar2 (10)
- EmployeeLastName – VarChar2 (10)
- EmployeeStreet – VarChar2 (30)
- EmployeeState – Char (2)
- EmployeeZip – Char (5)
- EmployeeHomePhone – Char (10)
- EmployeeCellPhone – Char (10)
- EmployeeHiredDate - Date
- EmployeeStatus – VarChar2 (15)
- EmployeePositionType- VarChar2 (10)

Members Table:

- MemberID – Numeric (6)
- MemberFirstName- VarChar2 (10)
- MemberLastName- VarChar2 (10)
- MemberStreet - VarChar2 (30)
- MemberState - Char (2)
- MemberZip - Char (5)
- MemberHomeNumber – Char (10)
- MemberCellPhone – Char (10)
- MemberGender – Char (1)
- MemberAge – VarChar2 (3)
- MemberBirthDate - Date
- MemberJoin Date - Date
- MemberOneTimeFee – Numeric (3, 2)
- MembershipDiscount – Number (2, 2)

Course Table:

- CourseID – Numeric (3)
- CourseName – VarChar2 (10)
- CourseDescription – VarChar2 (40)
- CourseDuration- Time
- CourseFee – VarChar2 (3)
- CourseDifficulty – VarChar2 (15)

Class Table:

- ClassNumber – Numeric (4)
- ClassStartDate - Date
- ClassDays – VarChar2 (10)
- ClassTimePeriod - Time
- ClassCapacity – Numeric (3)

Room Table:

- RoomNumber – VarChar2 (6)
- RoomCapacity – Numeric (3)

Equipment Table:

- EquipmentSerial – Numeric (9)
- EquipmentName – VarChar2 (10)
- EquipmentPurchaseDate - Date
- EquipmentTypeMaintenanceDate - Date
- EquipmentType – VarChar2 (15)

Merchandise Table:

- ItemID – Numeric (6)
- ItemDescription – VarChar2 (30)
- ItemStandardPrice – Decimal (6, 2)
- ItemQuantity – Numeric (3, 0)
- ItemRe-OrderThreshold – Numeric (3, 0)

Purchase Table:

- OrderNumber – Numeric (7)
- DateofTransaction – Date

Purchase Line

- PurchaseQuantity – Numeric (4, 2)

Vendor Table:

- VendorID – Numeric (6)
- VendorName – VarChar2 (10)
- VendorPhoneNumber – VarChar2 (10)
- VendorStreet – VarChar2 (30)
- VendorState – Char (2)
- VendorZip – Char (5)

DATABASE IMPLEMENTATION

Tables:

```
CREATE TABLE Employee
(
        EmployeeID              NUMERIC (4) NOT NULL,
        EmployeeFirstName       VARCHAR2 (10) NOT NULL,
        EmployeeLastName        VARCHAR2 (10) NOT NULL,
        EmployeeStreet          VARCHAR2 (30),
        EmployeeState           CHAR (2),
        EmployeeZip             CHAR (5),
        EmployeeHomePhone       CHAR (10),
        EmployeeCellPhone       CHAR (10),
        EmployeeHiredDate       DATE DEFAULT SYSDATE,
        EmployeeStatus          VARCHAR2 (15),
        EmployeePositionType    VARCHAR2 (10),
        CONSTRAINT Employee_PK Primary Key (EmployeeID)
);

CREATE TABLE Instructor
(
        InstructorID            NUMERIC (4) NOT NULL,
        InstructorQualification VARCHAR2 (15),
        InstructorQualifiedDate DATE DEFAULT SYSDATE,
        CONSTRAINT Instructor_FK Foreign Key (InstructorID) REFERENCES
        Employee(EmployeeID)
);

CREATE TABLE Clerk
(
        ClerkID                 NUMERIC (4) NOT NULL,
        ClerkHourlyRate         NUMERIC (2, 0),
        CONSTRAINT Clerk_FK Foreign Key (ClerkID) REFERENCES
        Employee(EmployeeID)
);
```

```
CREATE TABLE Admin_Personnel
(
        AdminPersonnel          NUMERIC (4) NOT NULL,
        AdminCertification      VARCHAR2 (25),
                CHECK (AdminCertification IN ('Certified', 'Not Certified')),
        CONSTRAINT Admin_Personnel_FK Foreign Key (AdminPersonnel) REFERENCES
        Employee(EmployeeID)
);

CREATE TABLE Course
(
        CourseID                NUMERIC (3) NOT NULL,
        CourseName              VARCHAR2 (10) NOT NULL,
        CourseDescription       VARCHAR2 (40),
        CourseDuration          TIMESTAMP,
        CourseFee               VARCHAR2 (3),
        CourseDifficulty        VARCHAR2 (15)
                CHECK (CourseDifficulty IN ('Novice','Intermediate', 'Advanced')),
CONSTRAINT Course_PK Primary Key (CourseID)
);

CREATE TABLE Room
(
        RoomNumber              VARCHAR2 (6),
        RoomCapacity            NUMERIC (3),
        CONSTRAINT Room_PK Primary Key (RoomNumber)
);

CREATE TABLE Class
(
        ClassNumber             NUMERIC (4) NOT NULL,
        CourseID                NUMERIC NOT NULL,
        EmployeeID              NUMERIC (4) NOT NULL,
        RoomNumber              VARCHAR2 (6) NOT NULL,
        ClassStartDate          DATE DEFAULT SYSDATE,
        ClassDays               VARCHAR2 (10),
        ClassTimePeriod         TIMESTAMP,
        ClassCapacity           NUMERIC (3),
        CONSTRAINT Class_PK Primary Key (ClassNumber),
        CONSTRAINT Class_FK Foreign Key (CourseID) REFERENCES Course(CourseID),
```

```
        CONSTRAINT Class_FK1 Foreign Key (RoomNumber) REFERENCES Room
        (RoomNumber),
        CONSTRAINT Class_FK2 Foreign Key (EmployeeID) REFERENCES Employee
        (EmployeeID)
);

 CREATE TABLE Schedule
 (
        InstructorID            NUMERIC (4) NOT NULL,
        ClassNumber             NUMERIC (4) NOT NULL,
        CONSTRAINT Schedule_FK Foreign Key (InstructorID) REFERENCES
        Employee(EmployeeID),
        CONSTRAINT Schedule_FK1 Foreign Key (ClassNumber) REFERENCES
        Class(ClassNumber)
);

 CREATE TABLE Equipment
 (
        EquipmentSerial         NUMERIC (9)  NOT NULL,
        EquipmentName           VARCHAR2 (10) NOT NULL,
        RoomNumber              VARCHAR2 (6),
        EquipmentPurchaseDate   DATE DEFAULT SYSDATE,
        EquipmentTypeMaintenanceDate    DATE DEFAULT SYSDATE,
        EquipmentType           VARCHAR2 (15),
        CONSTRAINT Equipment_PK Primary Key (EquipmentSerial),
        CONSTRAINT Equipment_FK Foreign Key (RoomNumber) REFERENCES
        Room(RoomNumber)
);

 CREATE TABLE Member
 (
        MemberID                NUMERIC (6) NOT NULL,
        MemberFirstName         VARCHAR2 (10) NOT NULL,
        MemberLastName          VARCHAR2 (10) NOT NULL,
        MemberStreet            VARCHAR2 (30),
        MemberState             CHAR (2),
        MemberZip               CHAR (5),
        MemberHomeNumber        CHAR (10),
        MemberCellPhone         CHAR (10),
        MemberGender            CHAR (1)
```

```
            CHECK (MemberGender IN ('M','F')),
        MemberAge               VARCHAR2 (3),
        MemberBirthDate         DATE DEFAULT SYSDATE,
        MemberJoinDate          DATE DEFAULT SYSDATE,
        MemberOneTimeFee        NUMERIC (3,2),
        MembershipDiscount      NUMERIC (2,2),
        CONSTRAINT Member_PK Primary Key (MemberID)
);

 CREATE TABLE Purchase
 (
        OrderNumber             NUMERIC (7) NOT NULL,
        MemberID                NUMERIC (6),
        DateofTransaction       DATE DEFAULT SYSDATE,
        CONSTRAINT Purchase_PK Primary Key (OrderNumber),
        CONSTRAINT Purchase_FK Foreign Key (MemberID) REFERENCES
        Member(MemberID)
);

 CREATE TABLE Merchandise
 (
        ItemID                  NUMERIC (6)  NOT NULL,
        ItemDescription         VARCHAR2 (30) NOT NULL,
        ItemStandardPrice       DECIMAL(6,2),
        ItemQuantity            NUMERIC (3,0),
        ItemReOrderThreshold    NUMERIC (3,0),
        CONSTRAINT Merchandise_PK Primary Key (ItemID)
);

 CREATE TABLE Purchase_Line
 (
        OrderNumber             NUMERIC (7) NOT NULL,
        ItemID                  NUMERIC (6) NOT NULL,
        PurchaseQuantity        NUMERIC (4,2),
        CONSTRAINT PurchaseLine_FK Foreign Key (OrderNumber) REFERENCES
        Purchase(OrderNumber),
        CONSTRAINT PurchaseLine_FK1 Foreign Key (ItemID) REFERENCES
        Merchandise(ItemID)
 );
```

```
CREATE TABLE Vendor
(
        VendorID                NUMERIC (6) NOT NULL,
        VendorName              VARCHAR2 (10) NOT NULL,
        VendorPhoneNumber       VARCHAR2 (10),
        VendorStreet            VARCHAR2 (30),
        VendorState             CHAR (2),
        VendorZip               CHAR (5),
        CONSTRAINT Vendor_PK Primary Key (VendorID)
);
```

Function List:

Function 1: Search for employees by zip code.

```
Query: SELECT EmployeeFirstName, EmployeeLastName
       FROM Employee
       WHERE EmployeeZipcode IN 'xxxxx';
```

Function 2: Total Number of members enrolled in each class.

```
Query: SELECT CourseName, count(c.MemberID)
       FROM Class c, Enrolled e, Course co
       WHERE c.MemberId = e.MemberID AND co.CourseID = c.CourseID
       GROUP BY CourseName;
```

Function 3: Find total members in a course.

```
Query: SELECT MAX(COUNT(e.MemberID)), co.CourseID
       FROM Course co, Class cl, Enrolled e
       WHERE co.CourseID = cl.CourseID AND cl.ClassNumber = e.ClassNumber;
```

Function 4: Total Number of Members from each Membership type.

```
Query: SELECT COUNT(MemberID), MemberOneTimeFee
       FROM Member;
```

Function 5: The courses that each facility course instructor is qualified to teach.

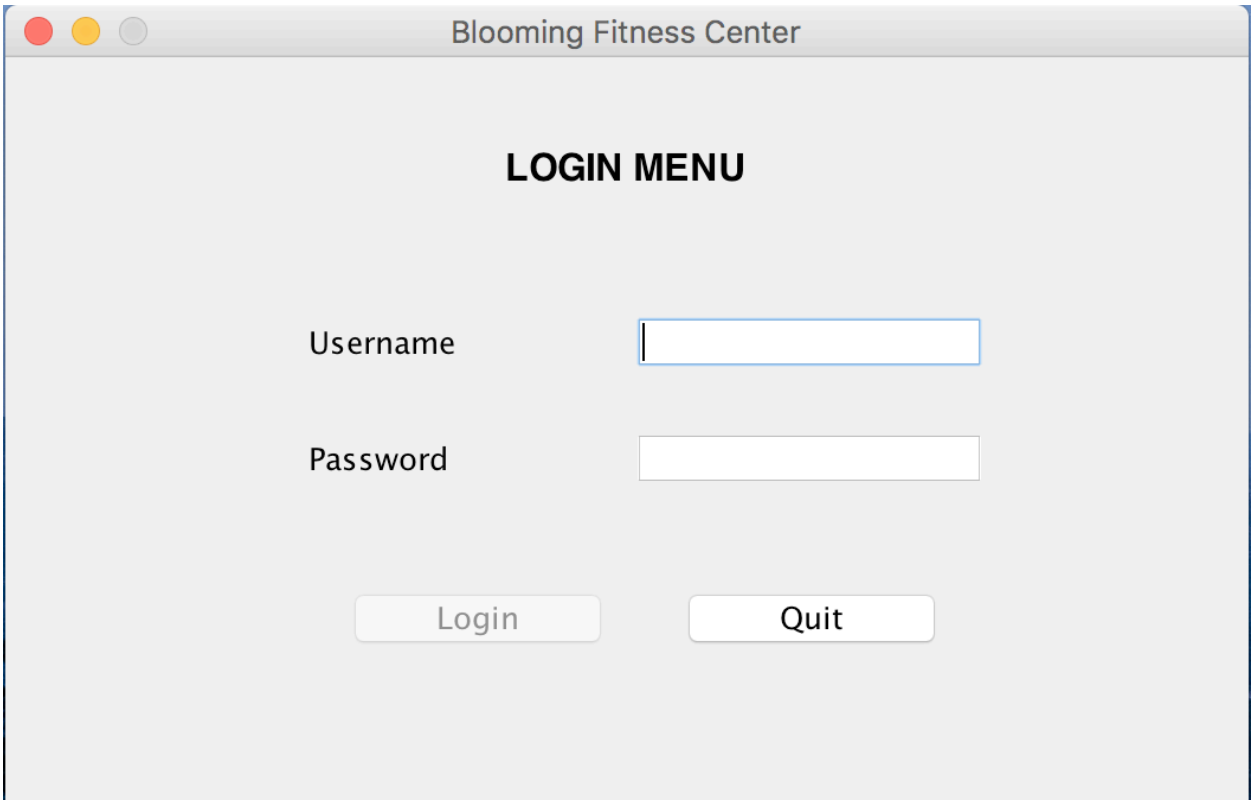
```
Query: SELECT i.EmployeeFirstName, i.EmployeeLastName, c.CourseName
       FROM Instructor i Schedule s, Class cl, Course c
       WHERE i.EmployeeID = c.EmployeeID
       AND cl.ClassNumber = s.ClassNumber
       AND cl.CourseID = c.CourseID;
```

APPLICATION DEVELOPMENT

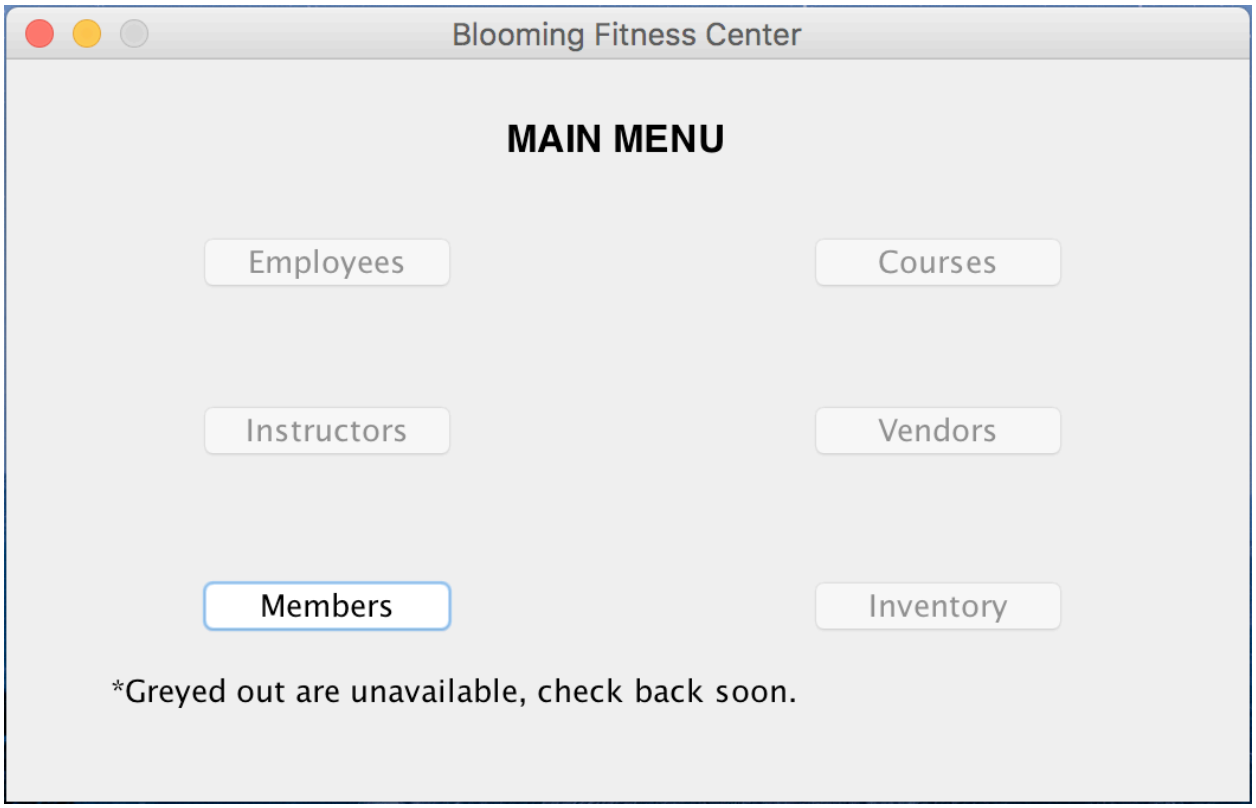
The user interface was implemented in Java with the use of Eclipse. We chose to use Java given the group member's previous experience with the language and familiarity with the IDE software. The entire implementation process was challenging and difficult but we were able to get a few functions performing correctly.

There are a few functions that can be accessed from the interface. We used the member table for the implementation process and the contents of the table can be manipulated directly through the interface. These functions include adding a new member, deleting an existing member, and browsing all the members in the table in a clean graphical interface. If a new member needs to be added, the user can click on the 'New' button and will wait for the user to input all the required fields. Once the user is satisfied with the input, proceed to hit 'Save' and that member will be added to the database. For the purpose of demonstration, we included a single table to manipulate and added two functions along with browse current database feature.

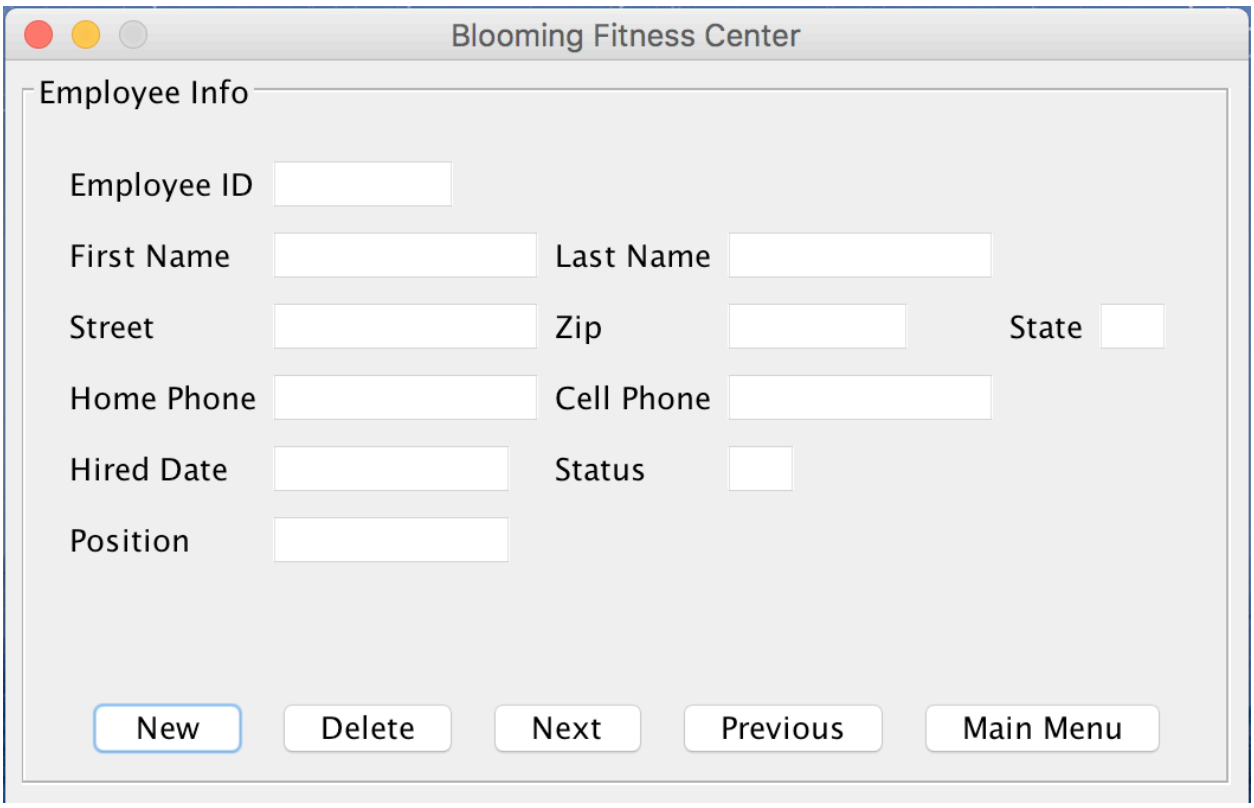
As mentioned earlier, the implementation was challenging and we learned a few things. The method we were shown in class involving Gradle would not work well with our interface because it was meant for a web page. We wanted to make our interface screenshots a working program. This made our implementation more challenging, given that the current resources do not support a graphical interface. We learned that the connection needs to be open every time a function is performed, in order for the query to complete. The issue we ran into was that the connection closed so we approached a different method using Java RowSet. This adds support for Java Database Connectivity, which retrieved content from the database and stored it as variables in Java. This made it easier to change data and use and Update method to send the changes to Oracle.



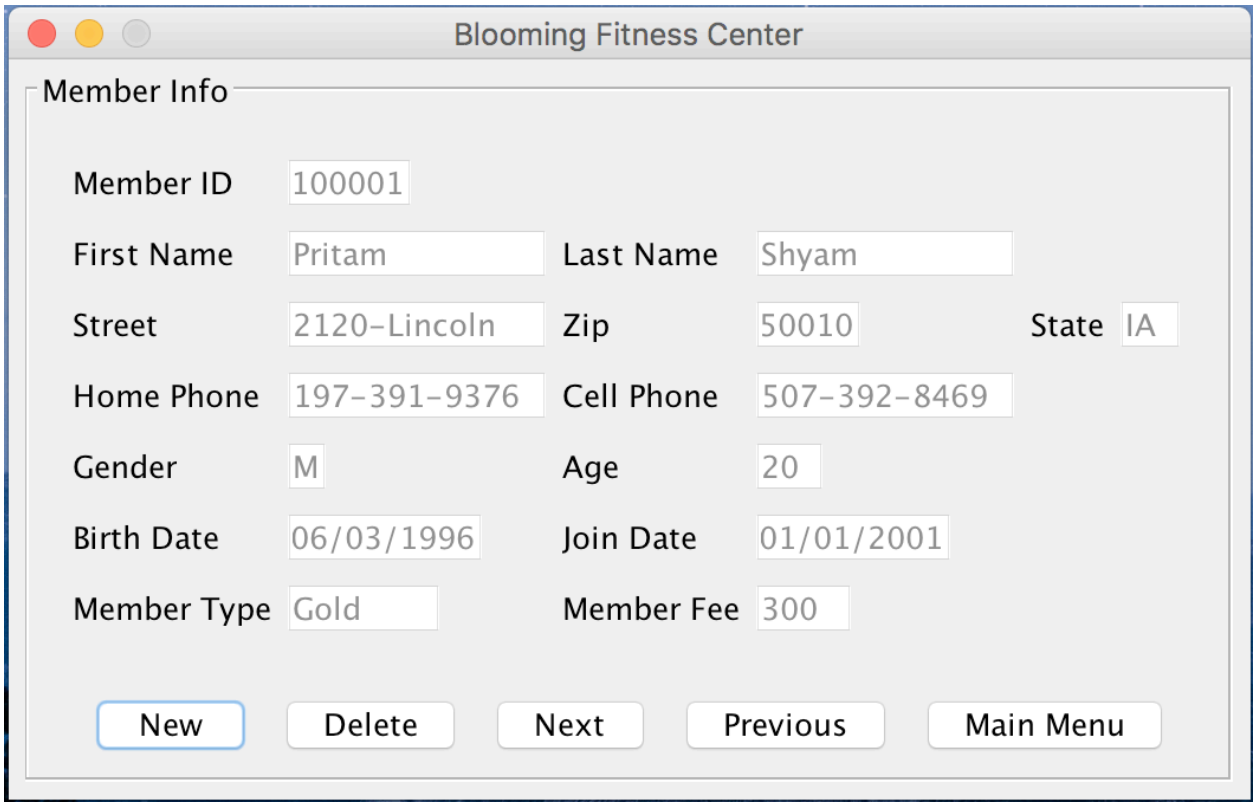
Login Screen: Requires authentication before allowing access, the authentication information will be stored locally within the application in an Array for added protection. The username and password are set on the first day for the employees.



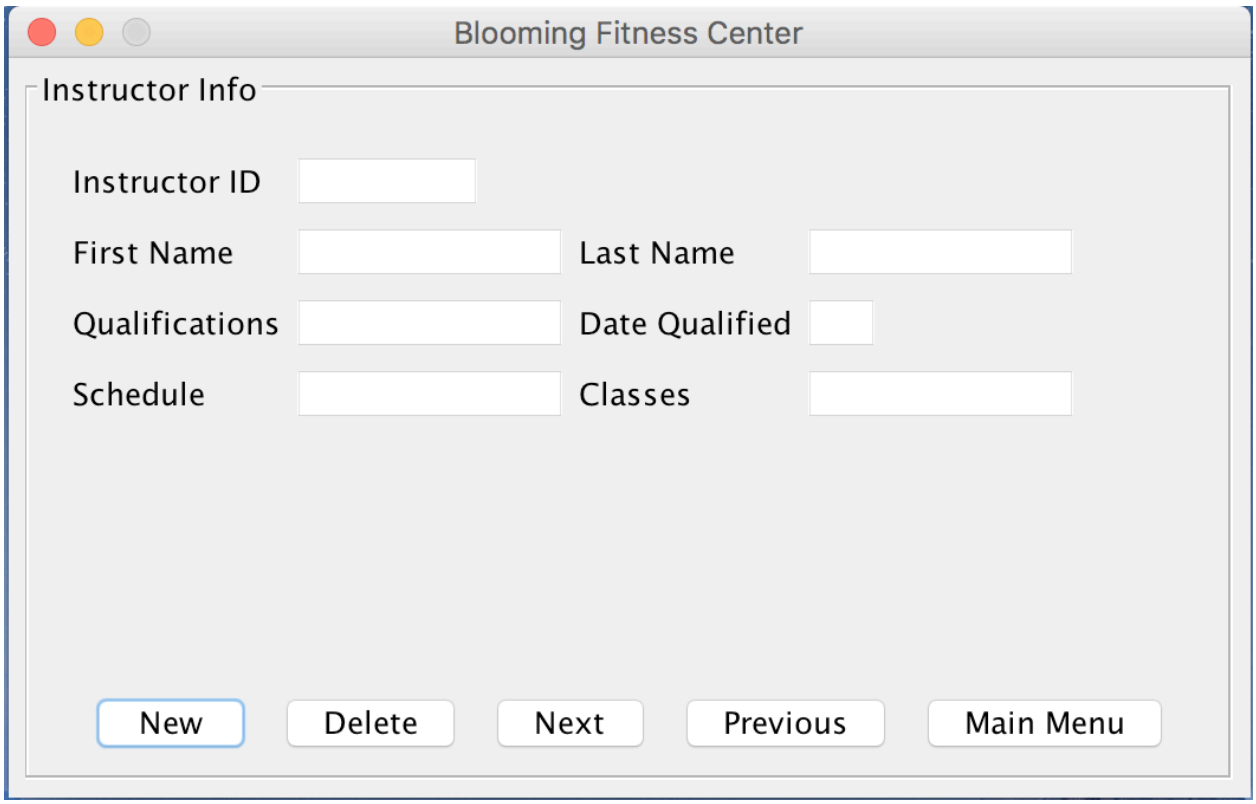
Main Menu: This will display all the options for the database, such as the list of employees, members etc. For demonstration purposes, only the members table is accessible, the rest will be added after our business approves our design.



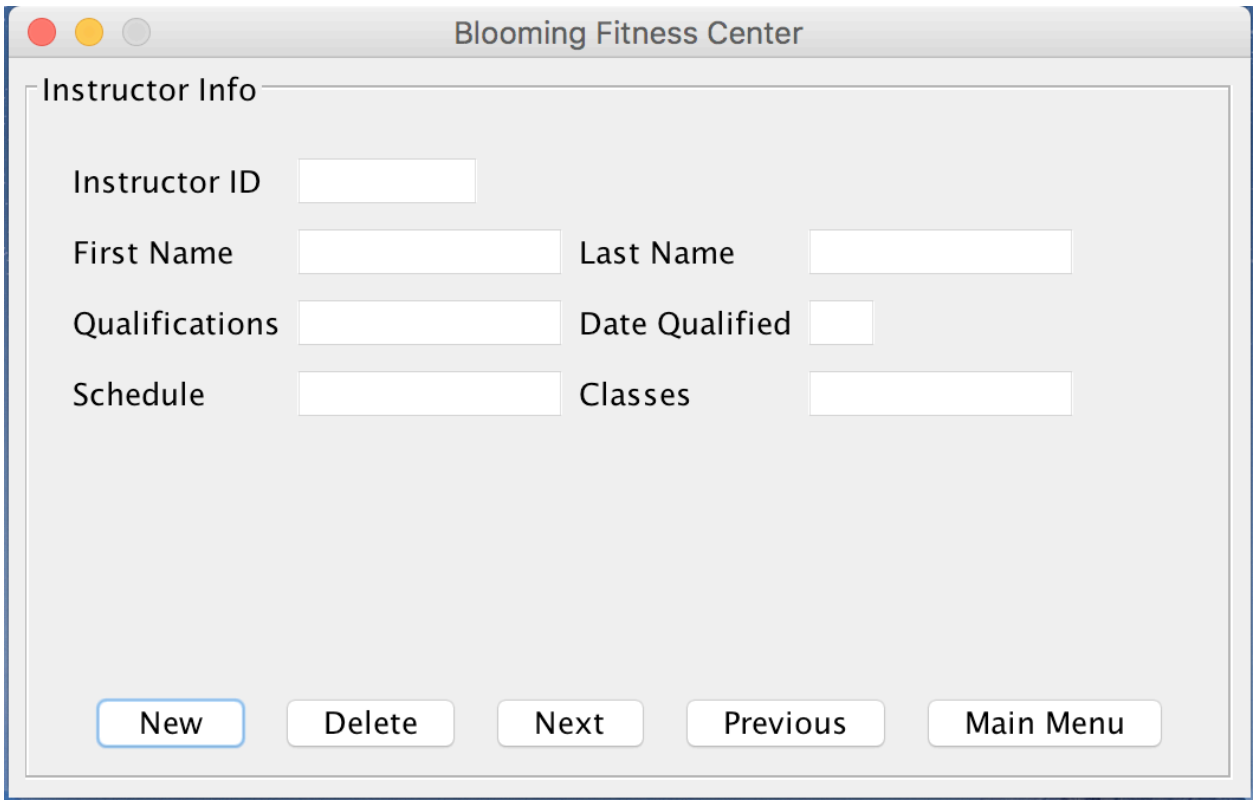
Employee List: This will display the details of the employee stored in the database. It will show all the employees and you can add/delete directly from the interface.



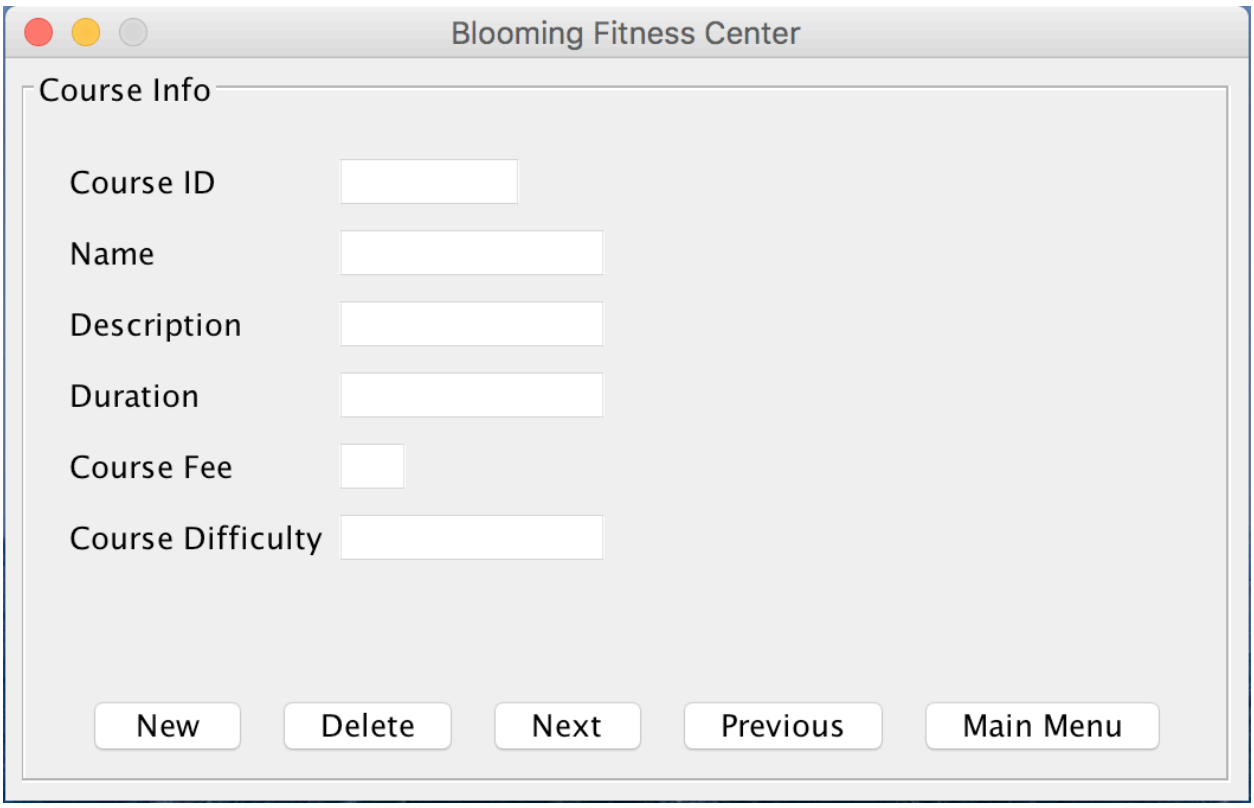
Member List: This will display the details of the members stored in the database. It will show all the members and you can add/delete directly from the interface.



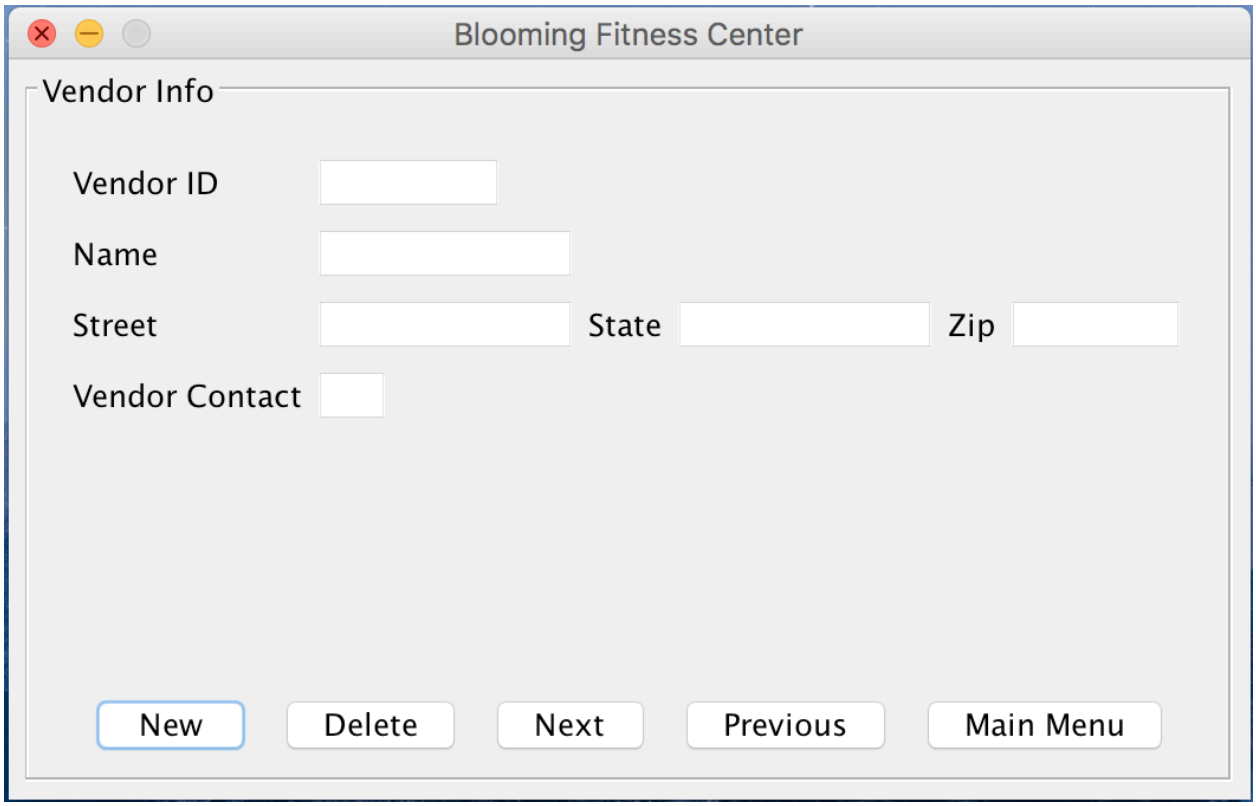
Instructor List: This will display the details of the instructors stored in the database. It will show all the instructors and you can add/delete directly from the interface.



Inventory List: This will display the details of the equipment's stored in the database. It will show all the items and you can add/delete directly from the interface.



Course List: This will display the details of the courses stored in the database. It will show all the courses and you can add/delete directly from the interface.



Vendor List: This will display the details of the vendors stored in the database. It will show all the vendor and you can add/delete directly from the interface.

CONCLUSION

Throughout the semester our team has ran into quite a few roadblocks as well as had some great successes. Initially our first struggle was developing a solid Business Case in which we could adequately create a database for. We chose to use Blooming Fitness as a Business case because it helped provide us with the structure and details needed to create a functional database.

In Part One of the project our team had great communication and was able to divide the work up and complete each section in a timely fashion. Some areas that were developing the business rules and creating entities. We were able create a solid foundation through part 1 that gave us a good foundation.

For Part Two, we again met as a team reviewed the assignment and divided up the project. This part of the project ran relatively smooth as a team developed our ERD and then created the logical schemas and normalizations individually. Members also reviewed each other's work to check for accuracy. One challenge was developing the graphic user interface. This was something no member had experience doing. However, Pritam took on the challenge of learning how to do so.

Part Three has provided the most challenges. In finalizing our project we have identified errors and also associative entities that were overlooked in the previous parts. Checking cardinality, relationships and design have not provided too many issues, minor tweaks were made to fix any problems we ran into. Implementing our SQL code and Java application has provided its own challenges. Understanding Primary Keys and Foreign and making sure they are applicable in iSQL has been difficult and also rewarding at the same time. Every member has looked at errors when creating our SQL Tables and helped problem solve the issue.

In conclusion we have been able to overcome many obstacles and provide a well-balanced database. As a team, we were able to play to each other’s strengths and weaknesses. Teamwork and communication was a strong point for this group project.

Team Member Name	Contribution Description	Contribution Percentage
Jacob Mikkola	Supporting Documents, Relational Schema (Class, Course, Equipment, Room), Business Riles, Implementaiton, Entity	20%
Jordan Bailey	Business Rules, Relational Schema (Employee, Instructor, Clerk, Admin Personnel, Schedule), Conclusion, Query Functions, Entity	20%
Katherine Ollearis	Case Description, Relational Schema (Member, Purchase), Entity	20%
Kevin Ratsavong	Supporting Documents, Relational Schema (Merchandise, Vendor), Implementation	20%
Pritam Shyam	Entity Diagram, User Interface Markups, Data Types, Application Devlopment, Impelementation	20%