Group 5 A.L.L Project 2

Gym Tracker Customer Loyalty Program

Table of Contents

Business Expansion & Justification

The Current Software Solution & Business Model

Currently we have developed a program which takes user fitness data collected from a smartwatch and saves the data to a specific user record which the user may access by entering their ID and PIN. Based on the user's performance data the program will then generate specialized discounts and perks, also saving this information to the users record.

Our program adds value by helping the business identify engaged customers. By offering these engaged customers incentives in the form of discounts and other in-gym exclusives the gym may be more successful in retaining these high value customers and building customer loyalty. As a result, this could boost sales.

Why Expand

For our project scenario, our program has become a success and there is high demand for it. As such the software solution could be expanded to serve multiple gym branches as opposed to only serving a single branch. In our scenario the gym has run our system in one branch as a trial to test the feasibility of implementing such a system. The gym has concluded that the current trial is a success and that further developing the system would be beneficial and in the interest of the gym. The gym hopes to use our system to boost sales across all of its branches.

The Approach

The current expansion approach is to expand into more locations, target a wider customer base and aim to sell and market more products and services by using targeted discounts. These three main goals can be achieved by further developing and expanding the current software solution to serve all branch locations around the country. Currently the program collects the data then saves it to a single machine. For our expansion, this will not be suitable. A database will have to be used instead so that multiple branches will be able to share and sync customer data. The user must be able to access their records over several branches.

Define Deliverables

What is to be delivered by the end

-By the end of the planning and development phase our aim is to implement a database within our solution.
-Our python program needs to be able to save information to and pull information from the database.
-Our python program needs to present a report to the user showing them their fitness, personal, discount and payment information. The user must be able to filter the information
-An entity relationship diagram must be created to show how the database works and how the information is saved for documentation purposes.

How can this be achieved

In order to develop a database, we must identify then group information collected by the python program. This way we can then begin to structure the data and start to develop an entity relationship diagram. Once we have created and finalized the entity relationship diagram, we can then begin to build the database.

Once we have built the database, we then need to link this to our python program. To do this, research into connecting the two must be carried out. Once this research has been completed, we can then begin to put the two aspects of the software solution together. Furthermore, we can have the python program present the user with the report and filter information by using sql statements and querying the database, only showing relevant information.

Finally, developing a finalized entity relationship diagram is part of the database development process. The finalized entity relationship diagram can be taken and used as one of the deliverables for documentation purposes. The software solution also needs to be in a fully functioning state, ideally with no bugs. This can be achieved through using an iterative software development framework as well as constantly checking and testing the program.

Stakeholders

There are a number of key stakeholders which could be impacted by our service. These include

Business (Gym) : The business as a whole is the largest beneficiary of the entire system. The objective of implementing such a system is to boost sales by building customer loyalty and increasing customer retention.

Customer: Gym customers are one of the main stakeholders that would feel the direct impact of the system. The system is built to serve this group of stakeholders by showing them their performance metrics but also to offer them specialized discounts and promotions offered by the gym.

Management: With the data generated by the users, management can use this data to better inform their decision-making process. Managers can find out which activities, or fitness regimes are popular, and which are not.

Investors: If successful on a larger scale, the proposed system would boost sales. This would likely result in a stronger financial standing for the business. Investors would benefit as their share prices would likely increase.

Employees: Creating a system will make it easier for employees within the organization if they need to lookup customer details for any particular reason. Creating a database where customer information is stored means it's easier to bring up.

Service Infrastructure

The service infrastructure refers to the necessary technology and systems required to provide the service. Four key technologies have been identified. These technologies are vital in enabling us to provide the service. The four key technologies include:

Database management system (DBMS)

Our data will be stored on a database as it will allow our service to work as efficiently as possible. This is because a database management system allows us to store, organize and manage large amounts of data simultaneously. This is important for our application as a database management system provides a highly efficient method for handling different types of data. Without a database management tasks will have to be carried out manually which will take more time and therefore delay our service.

Local storage

For our service we intend to use a physical storage server. This is because we will have greater levels of control over the data and can build and tailor our storage system specifically to our capacity and security needs. Furthermore, a separate service level agreement will not be required between us and the 3rd party data storage provider. If we were to go with a cloud-based solution our service would be subject to the 3rd party's levels of availability and reliability. The upfront cost of a cloud solution is lower however we would be giving up complete control over our data.

Program

Our program has been designed to work in conjunction with a database. This is because our program allows businesses to input and manage the data which is stored onto the database. Our program also adds an extra layer of security to the main database, this is because passcodes have been added to restrict access to sensitive customer information. The program will be developed in python as our previous solution was developed in python. In addition, to this python can be run on several machines and is a versatile programming language.

Internet connection

An internet connection is one of the four key requirements. An internet connection allows for client systems to remotely contact the server where the database is being hosted. The internet is the backbone of the entire system, facilitating communication and allowing for remote access and connectivity. If the service grows significantly then increased levels of bandwidth may be required. As such a separate service level agreement will be needed between the service provider and the internet service provider (ISP)

Service Level Requirements

The software solution we will deliver must be able to perform these functions. These are requirements that must be met in order to fulfill the needs of the business we aim to serve (gym). The solution must be able to;

Save Customer Data

The software solution provided for the business must be able to save customer data. The type of customer data which we will be saving are first names, surnames, addresses, contact details etc. We have decided to save this customer data as we can create customer profiles and offer targeted discounts. Furthermore the way in which we hold customer data must comply with governmental standards (GDPR). We must not share or use the customer's data irresponsibly as we may be penalized.

24/7 Access

As we will be saving customer data and as customers may use the gym at any time their information needs to be accessible 24/7. Gym facilities are opened 24/7 meaning that at all times customer information should always be accessible to the customer if they need to make any changes to their information. The customers need to be able to see their information and the gym needs to be able to access customer records at any time.

Present customer data

Being able to present customer data is not only important but a minimum requirement. This is because customer's data needs to be accessible at any time. If customers have an inquiry about their information or need to update it such as changing their address, if we are not able to present their information, this would make our system look very limited to consumers. Customers need to be able to see all of their information in an organized manner. They must be able to filter and choose to see specific information. As such data needs to be grouped together into categories.

Organize customer records

As we will be holding customer data, we must also ensure that we have good organization of records. The customer records which are organized are as such, first name, last name, address, contact details etc. Customer data will be grouped into five categories. Personal details, invoices, discounts, sessions and performance. With having the organization of customer records this helps because it allows the information to be held to be found easier within our system. If the data isn't organized this could lead to misplacement of customers' information making it harder to find individual customer information and even a loss of records which will cause the service's reliability to decrease considerably with customers.

Present customer with personalized discount

As reporting personalized data to customers is part of our USP, this is also a minimum requirement. This is important because our service needs a feature that would make our service more diverse than services that could be similar to ours. Therefore, offering personalized discounts to customers is good because it gives us something unique about our service and our service is able to target customers which may be more likely to return and spend and the business we aim to support.

Security (Logins/ Passcode)

Another minimum that we must provide for individuals is the ability to have their own login/passcode. With the service that we are providing customers will get their own login/passcode to access their information and track their progress after their session. Each login code must be different and unique only tied to one specific user. Implementing passwords and logins are important as it prevents other users from gaining access to a specific customer's data.

Service Level Agreement

Availability

Availability Management focuses on the management of agreed-upon availability requirements. It aims to improve all aspects of the availability of IT services so that the systems can perform when they are required. This means making sure that all IT Infrastructure is necessary for availability targets. Availability management in our service is important because our IT services need to be efficient if we were to have a large number of customers. Our systems will be operational 24/7 for use by the business and their customers

Utility

Also called 'fitness for purpose', Utility regards to the level of performance of the services and the ability of the service to remove constraints from the client. Our service is able to offer utility as our system will be able to track customer data and generate highly engaged customers with a discount. The system will also keep a record of invoice information and personal details such as phone numbers, emails, names etc. Our service aims to greatly improve and streamline the record keeping, data storage and tracking process to enable the business (in our case, a gym) to focus on business activities that are more essential to their core business operations.

Warranty

Also referred to as 'fitness for use', Warranty refers to the level of reliability associated with the services and the ability of the services to operate reliably and for a long period. A one-off charge will be made to the business for providing them with this system. However, any continual improvements or system maintenance operations will be carried out and charged separately to the business. This way we are able to provide the business with a high-quality functioning software solution and any additional support in a financially sustainable manner.

Responsibilities

Service provider responsibility: Our responsibility will be to provide a working, fully functional system with the stated requirements to the business. The costing model will be a one of payment for a full and complete delivery of the software solution.

Any additional improvements, changes or maintenance operations carried out on the software solution will be charged and carried out separately from the main solution. These added levels aim to provide quality support in a financially viable and cost-effective manner.

Any activities carried out will be carried out in full or not at all.

Client responsibility: It is the responsibility of the business to check all requirements are met. Request for maintenance and pay the appropriate service fees before work is to be carried out.

In the case that the solution does not fully meet the specified requirements of the business or the software solution delivered has a critical bug or if maintenance operations have not been conducted correctly. The service provider provides support and work to fix the solution at no additional charge to the client.

Maintenance Schedule

For a small fee maintenance checks and operations will be carried out every month in order to ensure the software solution is in a healthy state. Each aspect of the solution will be checked. Data collection, displaying and storage will be checked to ensure smooth operation for the gym. By offering support in this manner we remove constraints from the gym allowing them to focus on essential business operations while we manage and maintain their data storage system.

Continual service improvement (CSI)

The software solution provided to the business can be improved by getting key stakeholder feedback. This could include but is not limited to interviewing customers, managers and frontline staff. The needs, requests and suggestions of these key stakeholders would be key in proposing and delivering improvements to the service as well as adding new and improved features to the software solution. An iterative software development methodology can be used to help implement the design and testing processes necessary for the implementation of these improvements

Value Additions

One possible value addition of our service is it **targets certain customers to improve customer retention**. Customer retention is the ability to keep customers for a period of time and in order for this to happen the service has to offer an incentive that keeps them returning to the service. The customers that this service targets are customers that want to exercise but lack the motivation to repeatedly attend sessions.
With our services, these customers should attend more regularly in order to benefit from the rewards for doing so.

Another possible value addition that our service provides is it **gives out specialized discounts in order to build customer loyalty**. In our python code, our service gives out specialized discounts based on the individual physical performance metrics and how often they attend. The more progress they make towards this, the better their personalized discount is.
By having this system in place, this motivates the customers to use the gym more often so not only will they get cheaper prices to access the gym, they will also see personal fitness improvements. The business benefits as well as they are able to sell their services on a more regular basis to their customer, this should boost sales.

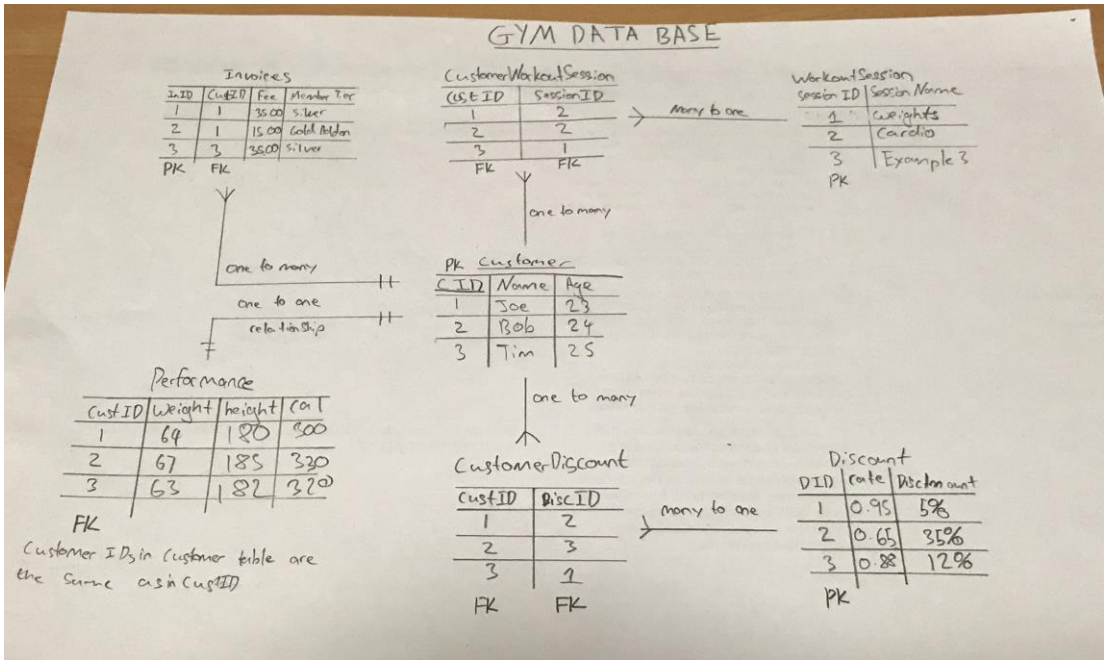
Furthermore, a third possible value addition in our service is it **improves customer experience by offering real time data and feedback about their performance**. Real time data is information that is immediately delivered after its collection. Real time data is useful because it means that its information is the most accurate and up to date. By offering real time data and feedback to customers about their performance, this allows them to access their performance results faster so that they can process the information more efficiently and possibly set targets for themselves quicker for their next session.

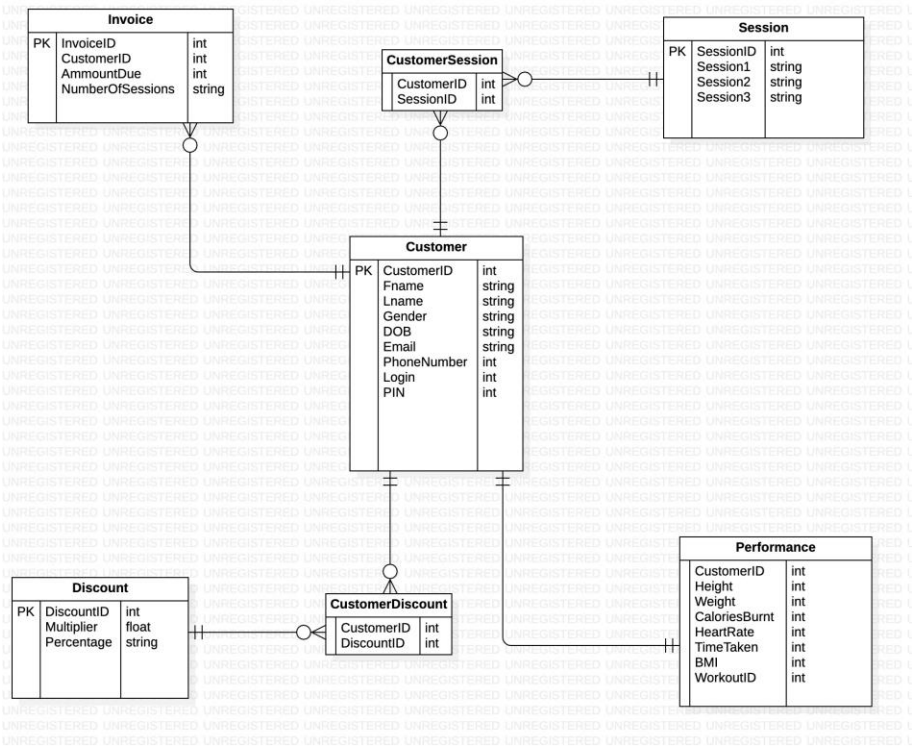
Preliminary research DBMS

A database allows data to be electronically structured by data-storage devices, which are used to find specific parts of information (Cox, 2014). According to Gorman (2014), this cannot be classified as a technology but rather a declaration of clarity and accuracy. However, a database management system is a software that enables businesses to analyze past records, increasing the likelihood of success through efficient planning. Dr E.F. Codd developed the relational model in 1970, which uses two-dimensional tables to show information (Narang, 2011). Therefore, a relational database is a collection of these types of tables. A conceptual model such as an Entity-Relationship (ER) diagram is utilized to depict data where tables are entities which have their own attributes (Bagiu & Earp, 2003). Normalization is a process where table structures are organized to reduce redundancies of data and to improve design (Coronel & Morris, 2014).

The database designed by Oracle was made for distributing resources across computer networks. This produces more servers and modular space to administer data and metadata (Oracle, 2020a). For example, SQL*Plus is a query tool based on characters and is used to run Data Definition Language (DDL) commands, which enables the database to be updated and kept in a suitable manner (Gennick, 2004). To run this command line interface for Oracle, Oracle Database must be installed. A login username and password are made before starting a preset database that a user can connect to. Once the login details are correct, a user can begin running statements (Oracle, 2020b).

Implementation of ERD





An entity relationship diagram is a type of diagram that is used to illustrate the relationship between tables in a database. There are three main types of relationships used in this diagram. One to one, many to many and one to many.

M:M

Where there is a many to many relationship(as can be seen with Customer and Discount) an intermediary table must be used. The intermediary table named CustomerDiscount holds CustomerID which is a forign key from the customer table and DiscountID which is a forign key from the Discount table. By using an intermediary table like this a many to many relationship can be defined between customer and discount. The same thing is also done with the Customer and Session tables. An intermediary table called CustomerSession is used and helps define the many to many relationship

1:M

The relationship between Customer and Invoice is a one to many relationship. This relationship is created by having the CustomerID value as a forignkey in the Invoices table. One customer can have many orders however each order is unique(an order cannot have multiple customers). There are two types of unique values in the Invoices table. InvoiceID which is the primary key of that table and CustomerID which is a forign key located in the Customer table. This type of relationship is also used for the intermediary tables.

1:1

This relationship is the simplest. All attributes(performance metrics) are tied to a forign key which in this case is CustomerID. One customer can only have one set of performance metrics and one set of performance metrics can only have one customer.

Entity Relationship Diagram & StarUML

The data has been grouped and sorted this way as it allows the program to easily store and retrieve information. We went through each entity and tried to identify the most logical relations between them, then set out to create a more solid and finalized ERD. Developing this finalized ERD took several iterations, some categories were separated into new ones whereas some were joined together. The rough ERD was done on paper using crow foot notation whereas the finalized ERD was made using StarUML. StarUML allowed for the modeling and defining of the database structure.

Once the ERD and schema had been conceptualized and developed. Turning the ERD into code was relatively easy. The correct SQL statements just needed to be typed and executed to create the database.

Design Relational Database & Collect Data

The database was created using a separate python file. The sqlite3 library was used to import the execute function which would allow sql statements to be executed. Sqlite3 is part of the standard library, as such no additional items need to be installed. The sql commands are encapsulated in the execute function. Once the statements for each table have been typed out. The python script is run which creates a .db file, in this case GymDatabase.db. The statements were run each time for each table to be created in the database. In total the script with the edited code is run six time to create each of the sex tables and define their relationships. The ERD played a very important role in this phase as the defined structure of the database was copied from the ERD. This made coding the database easy saving lots of time.

```
import sqlite3 as sql

conn = sql.connect("GymDataBase.db")

c = conn.cursor()

#create workout table
c.execute("""CREATE TABLE Discount(
        DiscountID integer PRIMARY KEY
        Multiplier real,
        Ammount text
    )""")

conn.commit()

conn.close()
```

```
import sqlite3 as sql

conn = sql.connect("GymDataBase.db")

c = conn.cursor()

#create workout table
c.execute("""CREATE TABLE Customer(
        CID integer PRIMARY KEY,
        Fname text,
        Lname text,
        Gender text,
        DOB text,
        Email blob,
        PhoneNumber integer,
        Login integer,
        PIN integer
    )""")

conn.commit()

conn.close()
```

```
1  import sqlite3 as sql
2
3  conn = sql.connect("GymDataBase.db")
4
5  c = conn.cursor()
6
7  #create workout table
8  c.execute("""CREATE TABLE CustomerDiscount(
9          cdCustID integer references Customer(CID) on update cascade on delete cascade,
10         DiscID integer references Discount(DiscountID) on update cascade on delete cascade,
11     )""")
12
13 conn.commit()
14
15 conn.close()
16
```

NOTE: Ignore line 7 “#create workout table”, we forgot to remove this line when coding the database tables. This line is a redundant comment. The same code was used six times, the only things that were changed were the SQL statements and as such we forgot to remove that line of redundant code. To see the name of the table being created look at the line with the c.execute function (line 8).

To create the tables simple SQL, create statements were used. In the first two examples primary keys are created by typing “PRIMARY KEY” after the datatype of the attribute. Creating a simple table as can be seen in the Discount table and Customer table examples are straight forward. Give the table a name, define the primary key then define other attributes required along with their data types. Once the

statement is complete ensure that the brackets are closed and that the .commit and .close functions are entered. The .commit function saves any changes made whereas the .close function closes the connection to the database.

```
import sqlite3 as sql

conn = sql.connect("GymDataBase.db")

c = conn.cursor()

#create workout table
c.execute("""CREATE TABLE CustomerSession(
       csCustID integer references Customer(CID) on update cascade on delete cascade,
       sID integer references Session(SessionID) on update cascade on delete cascade,
    )""")

conn.commit()

conn.close()
```

```
import sqlite3 as sql

conn = sql.connect("GymDataBase.db")

c = conn.cursor()

#create workout table
c.execute("""CREATE TABLE Invoices(
       InvoiceID integer PRIMARY KEY,
       iCustID integer references Customer(CID) on update cascade on delete cascade,
       Fee real,
       Comment text
    )""")

conn.commit()

conn.close()
```

```
import sqlite3 as sql

conn = sql.connect("GymDataBase.db")

c = conn.cursor()

#create workout table
c.execute("""CREATE TABLE Performance(
       CustID integer references Customer(CID) on update cascade on delete cascade,
       Weight real,
       Height real,
       CalsBunt real,
       HeartRate real,
       TimeTaken real,
       BMI real,
       Steps integer
    )""")

conn.commit()

conn.close()
```

```
import sqlite3 as sql

conn = sql.connect("GymDataBase.db")

c = conn.cursor()

c.execute("""CREATE TABLE Session( SessionID integer PRIMARY KEY, SessionName text )""")

conn.commit()

conn.close()
```

Primary keys were created by specifying the attribute's name followed by the datatype in this case, CID integer PRIMARY KEY, as can be seen in the first image.

Foreign keys are created by referencing primary keys of other tables. The command to create a forign key in my example is : CustID integer REFERENCES Customer(CID) ON UPDATE CASCADE ON DELETE CASCADE

What the cascade on deletion and on update command does is if the primary key is changed/deleted it will also change/delete in every table it is being referenced in. This ensures atomicity and consistency in the data. This prevents bad data meaning missing entries or other inconsistencies are prevented.

Querying and Dynamic Querying

```
conn = sql.connect("GymDataBase.db")
c = conn.cursor()
c.execute("SELECT Login FROM Customer")
EveryLogin = c.fetchall()
```

```
def personal_details(LOGIN,PIN):
    conn = sql.connect("GymDataBase.db")
    c = conn.cursor()
    c.execute("SELECT * FROM Customer WHERE Login = ? AND PIN = ?", (LOGIN, PIN))
    customerdata = c.fetchone()
    print(*customerdata, sep = "\n")
    conn.close()
    print ("\n"+"press enter to go back")
    i = input ()
    return output(LOGIN,PIN)
```

For querying and dynamic querying two examples will be used. However, both forms of querying are used constantly throughout the code to present the user with information, or in the case of dynamic querying, specific information only pertaining to that specific user.

In the first example the command SELECT Login FROM Customer selects the login attribute from table Customer and the fetchall function retrieves all customer login records and saves them to a sequence to be used by the register function.

In the second example, dynamic querying is used. The question marks and tuple (LOGIN, PIN) are used as place holders. By writing the dynamic query this way SQL injection is prevented. There is an alternative way, this involves using a tuple and dictionary. This is done for other dynamic query statements in the code. However, for this example the ? and tuples are used in “SELECT * FROM Customer WHERE Login = ? AND PIN = ?”, (LOGIN, PIN). It will select all attributes from the customer table where the record with matching login and pin values. The desired attributes must be entered into the tuple.

```
conn = sql.connect("GymDataBase.db")
c = conn.cursor()
c.execute("INSERT INTO Performance VALUES (:pCustID, :Weight, :Height, :CalsBurnt, :HeartRate, :TimeTaken, :BMI, :Steps)",
          {'pCustID': cid,'Weight': weight,'Height': height,'CalsBurnt': cals,'HeartRate': hrt,'TimeTaken': time,'BMI': bmi,'Steps': steps})
conn.commit()
conn.close()
```

The .connect() function is used to connect to the database file. The .cursor() creates a cursor that allows the script to interact with the database. The .execute() function allows the SQL statements to be executed.

In this example the second method for dynamic sql querying is used. This involves using a dictionary where the key is the attribute name and the value is the corresponding variable which takes the user input. This method like the ? method is better than using {} This is because it prevents SQL injection. This way the user isn't able to pass SQL commands to the input which is important as we dont want the user to be able to type commands and alter the database.

Finally it is important to save the new data written to the database by using the .commit() function and close the connection to the database by using the .close() function. If the connection is left open and the program crashes, this could corrupt the file leading to a loss of data.

```
def sessions(LOGIN,PIN):
    conn = sql.connect("GymDataBase.db")
    c = conn.cursor()
    c.execute("SELECT CID FROM Customer WHERE Login = ? AND PIN = ?", (LOGIN, PIN))
    cid = c.fetchone()

    c.execute("SELECT sID FROM CustomerSession WHERE csCustID = ?", (cid))
    sessionids = c.fetchall()
    for SID in sessionids:
        c.execute("SELECT SessionName FROM Session WHERE SessionID = ?", (SID))
        Sessions = c.fetchone()
        print ("Session \n", Sessions)

    conn.close()
    print ("\n"+"press enter to go back")
    i = input ()
    return output(LOGIN,PIN)
```

To show data from the Session table two tables must be accessed. As this table has a many to many relationship with the Customer table an intermediary table must be used. The first query gets the user's customer id from the customer table using two attributes, the user’s login code and pin. Once the customer id is retrieved, it is then passed through the next query which selects every session id from the CustomerSession intermediary table that shares the given customer id value. These values are then retrieved and saved to a sequence. The for-loop loops through the statement and passes the session IDs to the final query which prints the names of the sessions the specific user is attending.

Program demo

```
You are back to the begining of the program, press ENTER to continue...

Would you like to sign up ? y/n
n
Enter login code
9493
Enter PIN
109
```

The user is asked if they would like to sign up. If they are an existing user, they can sign in with their specific login code and pin. Once logged in the user can select five different options. They can see the personal details which includes name, dob, emails etc. Their performance and fitness information which includes calories burnt, time taken, weight, height etc. Their invoice info which includes how much they pay and for how many sessions. Their discount info which shows them what discounts they are entitled to. Finally, in the session selection they can see what sessions they are attending. Each selection can be selected by entering its respective number. The program will query the database using the entered login code and pin as points of reference. This can be seen for each of the five options

```
Press specified number to see related details

[1] FOR PERSONAL DETAILS
[2] FOR PERFORMANCE & FITNESS INFO
[3] FOR INVOICES
[4] FOR DISCOUNTS
[5] FOR SESSIONS
              OR
[0] TO EXIT TO HOME

1
2
Bob
Smith
M
1JAN2001
Bob.Smith@email.com
2345654
9493
109

press enter to go back
```

```
2
2
70.0
185.0
500.0
90.0
300.0
20.45
500

press any enter to go back

3
(45.0, 'fee for three sessions in GBP£')

press enter to go back
4
Discount/s
 ('5%',)

press enter to go back
5
Session
 ('Weights',)
Session
 ('Cardio',)
Session
 ('Flexibility',)

press enter to go back
```

The user can press the specified key to see information pertaining to that category. This way only relevant information to the user is presented. This acts a basic form of filtering the user can achieve without having to know or enter any sql commands. We decided to structure the program in this manner to make it as user friendly as it could possibly be within our capabilities.

These screenshots show the part of the program concerned with showing the user their personalized data

The screenshot below shows the part of the program responsible for creating a unique user profile which user info can be saved to. The below screenshots will go through the steps required to save user data.

```
You are back to the begining of the program, press ENTER to continue...

Would you like to sign up ? y/n
y
Welcome new user
We have generated your customer ID
3331
Please enter your first name
Bob
Please enter your last name
Black
Please enter your gender e.g M/F
M
Please enter date of birth in this format 1JUL1997
4MAY1995
Please enter an email address
Bob.Black@email.com
Please enter a phonenumber
2345678
Please enter a session one ID ([1] weights, [2] cardio, [3] flexibility)
3
Please enter a session two ID ([1] weights, [2] cardio, [3] flexibility)
If you aren't doing a second session enter [0]
0
Please enter a session three ID ([1] weights, [2] cardio, [3] flexibility)
If you aren't doing a third session enter [0]
0
We have generated your 3 digit PIN
Your 3 digit PIN is 841 Don't forget it
Details SAVED, please keep a note of your login info
LOGIN code : 3331
PIN : 841
```

In this part of the program, when the user answers with a “y” the program goes down a different branch. This section of the program collects information from the user, ties it to a unique customer id then writes it to a database. A unique logic code and pin is generated and given to the user. The user can use the login code and pin to access their personal info.

```
please enter number of steps
400
please enter calories burnt
500
please enter time taken in seconds
600
please enter heart rate
90
please enter weight
70
please enter height in cm
190
Calculating BMI...
BMI calculated 19.39
results save
You are back to the begining of the program, press ENTER to continue...
```

The program then goes on to ask the user for more information, specifically their performance metrics, which are then tied to the unique customer id and saved to the database. The results saved comment assures the user that their data has been saved and only displays once their data has been successfully written to the database. The user is able to get these performance metrics from the smart watch provided by the gym.

```
You are back to the begining of the program, press ENTER to continue...

Would you like to sign up ? y/n
n
Enter login code
3331
Enter PIN
841
Press specified number to see related details

[1] FOR PERSONAL DETAILS
[2] FOR PERFORMANCE & FITNESS INFO
[3] FOR INVOICES
[4] FOR DISCOUNTS
[5] FOR SESSIONS
              OR
[0] TO EXIT TO HOME
```

As can be seen in this screenshot. The user can in fact login with the details given to them and access their data as could be seen in the earlier example. When the user enters their specific login, only they can see their specific information and no one else's. An advantage of using a database is that the information is organized and can be read clearly as opposed to a flat text file. This can be seen in the example below.

Database Tables

Customer

	CID	Fname	Lname	Gender	DOB	Email	PhoneNumber	Login	
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Keegan	Pereira	M	25APR2000	keeganpereira2...	52664667	8625	130
2	2	Bob	Smith	M	1JAN2001	Bob.Smith@ema...	2345654	9493	109
3	3	Bob	Black	M	4MAY1995	Bob.Black@emai...	2345678	3331	841

CustomerDiscount

	cdCustID	DiscID
	Filter	Filter
1	1	5
2	2	5
3	3	1

CustomerSession

	csCustID	sID
	Filter	Filter
1	1	2
2	1	1
3	1	4
4	2	1
5	2	2
6	2	3
7	3	3
8	3	4
9	3	4

Discount

	DiscountID	Multiplier	Ammount
	Filter	Filter	Filter
1	1	0.9	10%
2	2	0.85	15%
3	3	0.8	20%
4	4	1.0	0%
5	5	0.95	5%

Invoices

	InvoiceID	iCustID	Fee	Comment
	Filter	Filter	Filter	Filter
1	1	1	30.0	fee for two sessi...
2	2	2	45.0	fee for three ses...
3	3	3	15.0	fee for one sessi...

Performance

	pCustID	Weight	Height	CalsBurnt	HeartRate	TimeTaken	BMI	Steps
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	58.0	174.0	400.0	90.0	330.0	19.16	300
2	2	70.0	185.0	500.0	90.0	300.0	20.45	500
3	3	70.0	190.0	500.0	90.0	600.0	19.39	400

Session

	SessionID	SessionName
	Filter	Filter
1	1	Weights
2	2	Cardio
3	3	Flexibility

As can be seen in these screenshots. The SQL statements earlier created the database tables. The dynamic SQL statements written have populated the tables with the correct data.

```
conn = sql.connect("GymDataBase.db")
c = conn.cursor()
c.execute("SELECT Login FROM Customer")
EveryLogin = c.fetchall()


LOGIN = (random.randint(1000,9999)) #Randomly generates login code
while LOGIN in EveryLogin: #Checks if unique login code already exitst
    LOGIN = (random.randint(1000,9999)) #If it does a login code is assigned
    continue #Keeps looping till assigned login code is not an login code
print ("We have generated your customer ID")
print (LOGIN)
conn.close()
```

This part of the code can be found at the beginning. It is important as it queries the database for all login codes and saves them to a sequence. Once in a sequence a random login code is generated. A while loop is then used to compare the newly generated login code to items in the sequence. If the randomly generated code matches an item in a sequence the while loop regenerates a login code. This is done to prevent the same login code from being assigned twice. In total there can be a thousand unique four-digit codes. The loop keeps running till the login code is not in the sequence and is unique.

These are just some few key examples of where SQL queries both static and dynamic are used. However, within our code similar SQL like the ones shown are used constantly throughout. These were just some key examples we picked out. This section would be very long if each and every instance was mentioned, explained and analyzed.

Legal, Ethical, Social and Professional Considerations

Legal

The legal principles which affect us directly include the General Data Protection Regulation and the Data Protection Act (2018) is an updated version of British protection laws. These laws work alongside the GDPR, which is an update to EU laws. This requires us to have legal liability for any breaches. For example, an employee could act unethically and illegally by holding customer information for criminalization. There are measures and consequences for the quality of data storage and how long certain types of information should be kept (Gov.uk, 2018). As a gym database system, we will need to set provisions for accuracy, subsequently planning ways of correcting or preventing accidental mistakes such as wrong addresses.

Ethical

The ethical principles of our professional setting are more focused on what we should do, and our legal principles are focused on what we must do. Alternatively, this can be perceived as what is right or wrong. These choices are based on morality which differs according to cultures and geographical locations. Our database management system will need to reflect fairness in a way that everyone is treated equally and appropriately. This includes customers, employees and other stakeholders who all need to interact in a professional manner to maintain integrity. For example, a customer may not trust the safety of their information or adjustable discount rules. Ethics and law overlap one another because regulations can be used to ensure that ethical practices are followed.

Social

Our service may be influenced by what is known as “Fitness culture” which refers to the culture surrounding exercise and physical fitness. These physical activities are usually carried out in gyms, wellness centers and health clubs. An international survey discovered that over 27% of the world's total adult population attends fitness centers, and that 61% of regular exercisers are currently involved in “gym-type” activities. (Les Mills Global Consumer Fitness Survey, 2013)

This may influence our service as we will focus our attention on the 39% of regular exercisers who have strayed from “gym-type” activities, by giving customers special discounts to entice them into investing in our service.

Analysts with American Sports Data state that 25% of the 41 million health-club members in the United States are over the age of 55 and claim that “this has been the fastest growing segment of health club membership” since their research in fitness gym trends had started in 1998. This trend is due to the vast number of aging Baby Boomers - people born between 1944 and 1964 - which is expected to reach 70 million by 2030 according to CDC and Merck Institute of Aging data.

A 2014 study insinuates that group fitness attracts more female clients, outnumbering males 5:1 in this category. IHRSA’s 2014 also suggested that males only make up 38% of club attendance figures.

Professionalism

Professionalism refers to the competence or skill expected of a professional. Using our idea that will be provided for gyms specifically the different gym facilities need to show their professionalism in different aspects of the business. There are several ways that they can maintain their professionalism, for example, codes and standards, computer laws, ethical decision making etc. We will be using a database management system and with this we will be gathering data from customers which are seen as being confidential. These include examples such as an address, credit card details, pins, passwords. The different gym facilities gathering the information need to follow computer laws such as using not abusing their authority as they have access. With all the access to this information, they shouldn't misuse this information by changing any information or using this information for personal gain as this is seen as breaking computer laws.

Why is Professionalism Important

Professionalism enhances growth and it adds value to the organization. The quality of service provided by the organization also increases and trust between the clients and the organization would be solidified. Companies that interact directly are obligated to provide the best service they can, and the organization should be presented in the best possible light(Scott, S, 2020). Professionally written organization reports, business plans or different correspondence assist businesses to remain accountable with their degree of service(Scott, S, 2020). The influence given in the paperwork submitted is vital in enforcing the proper impression of your business.

Summary

In summary the holistic process of IT service management has enabled us to conceptualize and prototype a service from start to finish. It has also helped us in identifying the impact of our service on relevant key stakeholders as well as consider the technologies, requirements, limitations, processes and real world considerations that need to be managed in order to deliver and provide a fully functioning service that is successful while being economical.

Bibliography

Bagui, S. & Earp, R. (2003) Database Design Using Entity-Relationship Diagrams. USA: CRC Press, pp.24-25

Burgess, Tim. (2013). "Fitness is the World's Biggest Sport". [online] Les Mills Global Consumer Fitness Survey. Les Mills Available from: <https://www.clubindustry.com/franchise-wise/defining-right-demographics-your-fitness-franchise> [Accessed 5 March 2020]

Chron (2019) "What is the target market fo a fitness gym". [online] Chron.com Available from" <https://smallbusiness.chron.com/target-market-fitness-gyms-3354.html>

Coronel, C. & Morris, S. (2014) Database Systems: Design, Implementation & Management. USA: Cengage Learning, p.191

Cox, S.A. (2014) Managing Information in Organizations: A Practical Guide to Implementing an Information Management Strategy. UK: Palgrave Macmillan, p.62

Gennick, J. (2005) Oracle SQL*Plus: The Definitive Guide, Second Edition. USA: O'Reilly Media, p.1

Gorman, M.M. (2014) Database Management Systems: Understanding and Applying Database Technology. Oxford: Butterworth-Heinemann Ltd, p.1

Gov.uk (2018) *Guide to the General Data Protection Regulation*, viewed 20th February 2020, https://www.gov.uk/government/publications/guide-to-the-general-data-protection-regulation

Narang, R. (2011) Database Management Systems, Second Edition. New Delhi: PHI Learning Private Limited, p.68

Oracle (2020a) Introduction to the Oracle Database, viewed 6 February 2020, https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm

Oracle (2020b) SQL*Plus Quick Start, viewed 6 February 2020, https://docs.oracle.com/cd/B12037_01/server.101/b12170/qstart.htm#i1055567

Scott, S., (2020). *The Importance Of Professionalism In Business*. [online] Smallbusiness.chron.com. Available at: <https://smallbusiness.chron.com/importance-professionalism-business-2905.html> [Accessed 16 March 2020]