Students Names & Numbers:

10581193 Sharon Farrell

10609974 Arivoli Ravichandran

10609647 Sugeeth Suresh Geeth

Course Title: **Advanced Database B9IS100**

Lecturer Name: Shazia A Afzal

Module/Subject Title: Advanced Database

Assignment Title: Online Bookings System Project

Table of Contents:

Figures

Background for the development of online system

We have been asked to design and implement an online booking system for a Family Entertainment Centre based in Dublin. The brief from the client is as follows:

Design an online booking system solution that allows potential guests to self-book and pay through the clients' website, giving the best tools to run and scale the operation, all in one place.

The goal of this project is to analyse, design, develop and implement an online reservation system for a Children's Family Entertainment Centre.

The purpose of an online booking system is to allow customers to self-book and pay through a website, securely store customer's data, manage staff and keep the business running long after we have gone home and closed our doors for the day. It is estimated that 56% of young adults with young children search and book activities for their children online in the evenings when their children are in bed.

The center is 7,000sq foot play structure, trampolines, inflatables, café and childcare programmes, a safe, secure, and fun environment for families to play and exercise while being challenged and stimulated, regardless of the Irish weather, with fantastic, fresh, healthy and locally sourced food offering.

But if the business has successfully gotten this far without an online booking system, we might think:

- Do we even need an online booking system?
- Are we getting enough reservations?
- Why bother taking on that extra expense?

Our aim as software developers is to show how an online booking system can benefit and grow the company to the next level.

Scope of Works

A booking system can help the client do way more than just passively accept bookings and payments online. Reservation technology (res-tech) has changed the way businesses conduct their business online and through this technology business can operate 24/7 365 days of the year. They can offer a more efficient experience for the user. [Kyla Steeves, October 1, 2020]

Specifically, this project aims to create a reservation system that offers the following:
Increase the number of party/general admission guests

Reduce the time consumed during reservation
Highly integrated data
Spend less time searching and retrieving files
Create monthly sales report
Keep up with competitors by catering for wider markets
Track growth with real-time reporting
Avoid double bookings with channel management
Assign inventory and staff to individual activities
Gather details about guests before they arrive
Break down day to day priorities
Staff management and cost savings

Business Requirements

Before adopting a new system, we should always consider if it's right for the business. We will show how and why it is of benefit to the client that they adopt a new system for online bookings.

1. Business open 24/7

One of the main advantages of an online booking system is that the center can stay open for business 24/7, no matter what the hours of operation are. Like how the website allows guests to view and find out about the business and its offerings, an online booking system makes it, so they don't have to wait until till the business is open to reserve an activity. They also have the comfort and convenience of booking in their own time and when it's right for them. In other words, an online booking system is always open for bookings.

2. Securing

Online booking systems are cloud based and so keep the customers' and client's information stored in the cloud. This ensures that the information is protected from in-house computer crashes, and as the information is not stored on the client's desktop computer. Information stored on cloud-based servers is more secure than information stored on desktops. A hard drive crash or computer virus can wipe the entire system in a matter of seconds, whereas online booking software relies on highly protected servers to safeguard these issues. This also means that all the information required by the client is accessible 24/7 through a secure online login system.

3. Cost savings for the workload

An on-Booking system can save the company time by automatic email confirmation, producing daily reports, balancing card payments, time spent with customers in house making bookings can be cut in half, answering calls regarding bookings as the customer can be directed to the online booking system. This can allow staff to spend more time on the customer experience

rather than being tied up on the phone.

4. Creates a better guest experience

Having an online booking system allows staff to spend more time on the customer experience while visiting the facility, it gives the user the time to book there at our convenience thus allowing the staff freer time while they visit. It elevates the stress for the staff knowing a customer is standing waiting to book in while they are busy taking phone calls and phone bookings. It also gives the customer a view of what is on offer and knows the procedure when they arrive. Customers know days expect everything to be available online and are surprised when a facility does not have the option of online booking.

5. Organised database

An online booking system is like a hub where it stores the companies database of information about the business including customer information, revenue, waivers, forms, invoices, inventory, staff, daily bookings, tracker details end-of-day reports and lots more.

6. Everything is all in one place.

All the clients will need to do is log into the system to view all the information. This means that the client does not need to be on site all the time to keep an eye on what is going in the day to day running of the business, they can log in and see everything including all the sales the number of customers, what activities they are participating in, how the daily sales are, what staff are in and working. This also provides an essential tool for future planning and marketing. It gives the client an insight into the popular days, age profile of the customer and the busiest times. With this information the client and plan and decide what direction the business should in. One advantage of the client and staff having customers information in one place is they can add notes on the bookings and see how many times a particular customer has been in the facility and reward them for their loyalty. This also keeps the staff aware of any issues there may have been in the past.

7. Prevents cancellations and no-shows

When a customer books online, they have to pay immediately and so there is a very small chance of them not showing up for the activity. All the terms and conditions of the bookings are clearly laid out prior to confirming the booking and each user has to tick a box saying they had read these terms and conditions. Online bookings set out a very clear and precise booking agreement. The client will have all the customer details and so they will not be at a loss of revenue due to no shows as all the bookings are paid influx at the time of booking. At the moment the only bookings that are paid for in full are the activities whereas the parties (which is the client's biggest revenue spin) are paid with a deposit rather than in full. We are advising the client all online bookings be paid in full at the time of booking. The online bookings will receive a confirmation email and follow up email reminding the user of the

booking they made and the date and time of the booking.

8. Business and customer information

With an online booking system, the clients are able to track and view all sales and customer information and so make sound business decisions based on the information gathered from the bookings. They can plan out where they might need extra staff, make marketing decisions for times that are quiet and give offers to customers to increase the footfall at different times, reward loyal customers and have all the reporting information necessary to run the business.

9. Online Booking systems can integrate software

Online bookings can integrate some marketing tools such as MailChimp where the customer data can be used for promoting the business (The customers would have to tick a box to say they are happy to receive marketing information). Send notifications to the customers making them aware of different promotions. Accounts software can be integrated with the sales reports. Relieving staff of manually inputting information. Track stock levels and automated reordering reports.

10. Complement the existing business

An online booking system can complement and make the existing business more efficient through staff management and bookings.

Business Rules

The system will have several capabilities and functions including the following.

11. Customers' ability to set up an account with a login and password, once the account is created, they have easy access to the system. Required details entered once.
12. Risk acknowledgements forms valid for at least 1 year once registered on the system (no waste of time entering details every time you log in) Party Bookings are easy to manage as risk acknowledgement forms are already stored on the system.
13. Children details are limited to Full Names Gender and DOB.
14. Access to all activities and extras (regular customers will be able to book their favorite activity in only one click as system will store their preferences) The system must be capable of cross-referencing names and addresses emails and telephone numbers to avoid duplication.
15. The payment details will be stored, and customers will only have to enter their CVC number and accept the payment.
16. Staff records are easily managed (working hours, shifts) once entered when starting the job.

17. Access for administrators to all reports (sale reports, staff hours, stock control, customers details, child information)
18. Once the booking is completed an email alert will be sent to the company email alerting them of the booking. Admin can check the details of the bookings to make sure that the booking is ok, and details are correct. The information in the email will be the sale and activity ID, the payment made, payment id, number of participants, date and time of the booking and activities. If a party is booked Terms and conditions will be sent to the user.

Rational Schema and Relationship Diagram

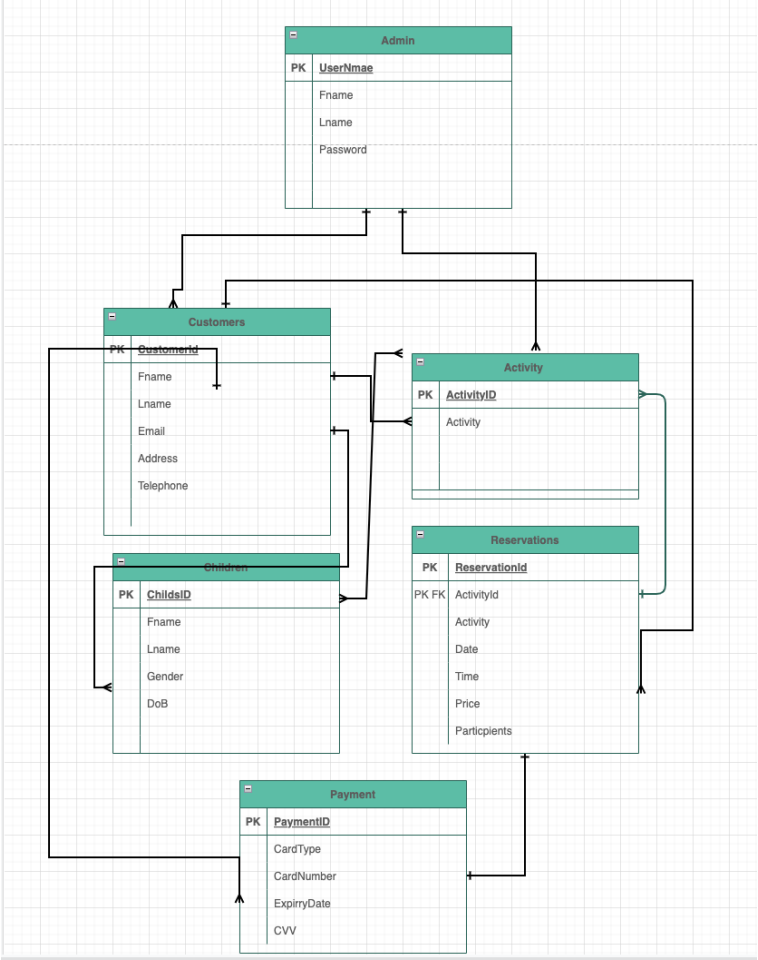


Figure 1 - Entity Relationship Diagram (ERD) 1NF

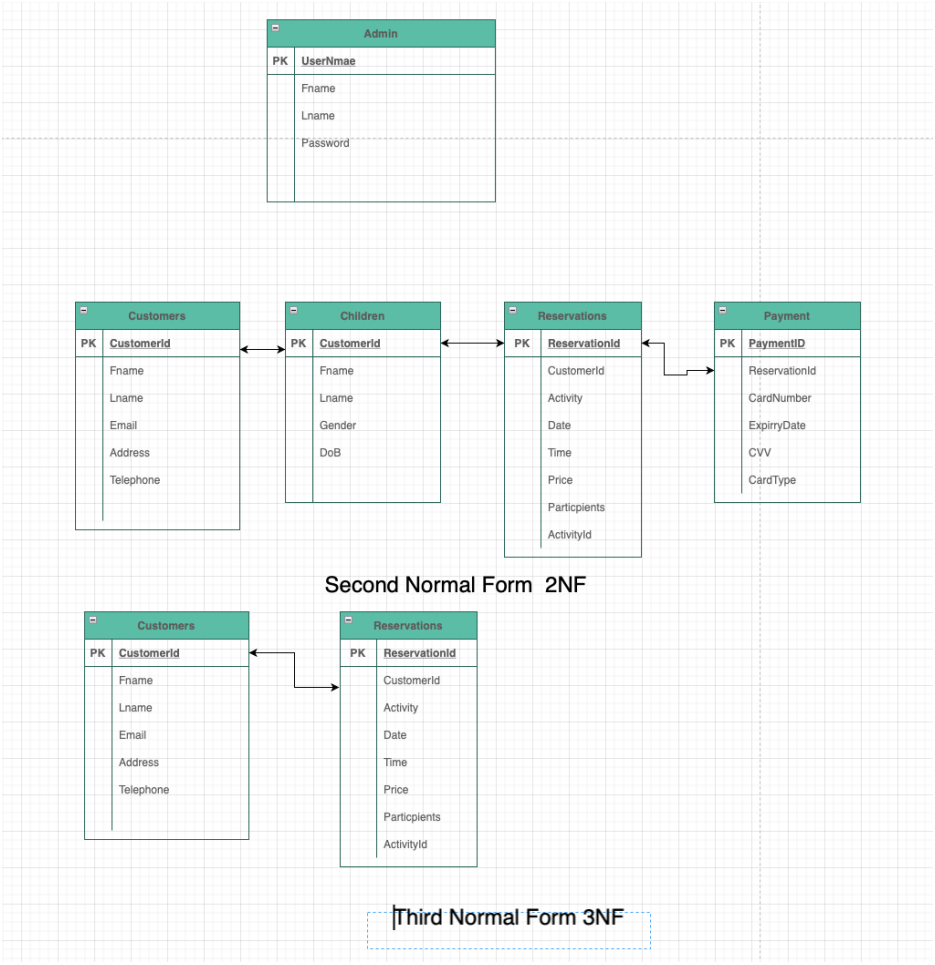


Figure 2 showing 2NF and 3NF

The diagram in Figure 1 shows the relationships between each entity in the proposed database as described below.

The relationships that exist between all the entities are as follows:

- Customers and Child - The user needs to set up an account to login and complete a risk acknowledgment form where they will add their child's or children's details including their full name, date of birth and gender. The reason we need the date of birth is that some of the activities are for children over the age of five years.

- Admin and Activities – The administrator can login onto the system with their unique login details and change the pricing, dates, time slots and duration of any of the activities.

- Customers and Activities – The user can view the Activities on the website and choose an activity and then click to book where they will be brought to a page that shows the dates and time available to book online.

- Customers, Reservations and Payment – the user can click to book online and will be brought to the online booking portal and given the option to choose dates, times, capacity and extras when booking on completion the User will then be brought to the payment portal where they will make a payment.

- Admin and Payments – the administrator will be able to access the payments reports from the merchant supplier and print daily sales.

- Admin and Reservations – Once a booking is made online the admin will receive an alert email showing the details of the booking and payment details. They will also be able to look at the updated tracker on the system to see what bookings have been made. This will give the admin a list of all online and in-house slots that are booked on a particular date. If a party is booked this is viewable in the calendar.

Database Creation (Arivoli Ravichandran)

The link to the XML and database is on a Repository we set up on GitHub

https://github.com/SharonF11/AdvancedDatabaseCA1.git

```
create database dbca;
```

CREATE TABLE ADMIN

```
CREATE TABLE Admin(
   AdminID varchar(10) not null PRIMARY KEY,
   Fname varchar(30) not null,
   Lname varchar(30) not null,
   UserName varchar(15) not null,
   Password varchar(10) not null
);
```

AdminID	Fname	Lname	UserName	Password
ad001	Maria	Aders	Maria01	Anders
NULL	NULL	NULL	NULL	NULL

CREATE TABLE CUSTOMERS

```
CREATE TABLE Customers(
   CustomerID varchar(10) not null PRIMARY KEY,
   Fname varchar(30) not null,
   Lname varchar(30) not null,
   Email varchar(15) not null,
   Address varchar(30) not null,
   Telephone varchar(10) not null
);
```

CustomerID	Fname	Lname	Email	Address	Telephone
cd001	Ana	Trujillo	ana@gmail	Obere St,berlin	9865747654
cd002	Antonio	Moreno	antonia@gmail	Constitución ,mexico	9876543210
cd003	Christina	Jack	Christina@gmail	Mataderos 2312,mexico	9867554321
cd004	Thomas	Hardy	Thomas@gmail	120 Hanover Sq.,london	9876554321
cd005	John	Petter	john@gmail	Thomas st, london	9068932145

CREATE TABLE ADMIN CUSTOMER RELATION

```
CREATE TABLE AdminCustomerRelation(
   CustomerID varchar(10) NOT NULL,
   AdminID varchar(10) NOT NULL,
   FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
   FOREIGN KEY (AdminID) REFERENCES Admin(AdminID),
   UNIQUE (CustomerID, AdminID)
);
```

CustomerID	AdminID
cd001	ad001
cd002	ad001
cd003	ad001
cd004	ad001
cd005	ad001

CREATE TABLE ACTIVITY

```
CREATE TABLE Activity(
    ActivityID varchar(10) not null PRIMARY KEY,
    Activity varchar(30) not null,
    CustomerID varchar(10) not null,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

ActivityID	Activity	CustomerID
ac001	WaterGame	cd003
ac002	Horror House	cd004
ac003	Snow world	cd002
ac004	Free fall tower	cd005
ac005	Roller Coaster	cd001
ac006	Kids Taxi	cd001

CREATE TABLE PAYMENT

```
CREATE TABLE Payment(
    PaymentID varchar(10) not null PRIMARY KEY,
    ccNo varchar(16),
    ExpiryDate varchar(20),
    CVVNO varchar(15)
);
```

PaymentID	ccNo	ExpiryDate	CVVNO
py001	23456789	09/2025	403
py002	43256789	07/2023	581
py003	654237890	09/2033	342
py004	54372896	10/2026	765
py005	65254896	12/2027	897
py006	6586439	02/2028	287

CREATE TABLE RESERVATION

```
CREATE TABLE Reservation(
   ReservationID varchar(10) not null PRIMARY KEY,
   Activity varchar(30) not null,
   Date varchar(30) not null,
   Time varchar(30) not null,
   Place varchar(15) not null,
   Participants varchar(30) not null,
   ActivityID varchar(10) unique not null,
   PaymentID varchar(10) unique not null,
   FOREIGN KEY (ActivityID) REFERENCES Activity(ActivityID),
   FOREIGN KEY (PaymentID) REFERENCES Payment(PaymentID)
);
```

ReservationID	Activity	Date	Time	Place	Participants	ActivityID	PaymentID
rs001	Kids Taxi	04/07/2022	10.15	Mexico	7	ac006	py001
rs002	snow world	04/07/2022	10.20	Mexico	6	ac003	py002
rs003	Roller Coaster	04/07/2022	11.00	London	5	ac005	py003
rs004	Horror House	04/07/2022	11.05	London	3	ac002	py004
rs005	Free fall Tower	04/07/2022	12.00	London	2	ac004	py005
rs006	WaterGame	05/07/2022	11.45	London	4	ac001	py006

CREATE TABLE CHILDREN

```
CREATE TABLE Children(
   ChildrenID varchar(10) not null PRIMARY KEY,
   Fname varchar(30) not null,
```

Lname varchar(30) not null,

DOB varchar(15) not null,

CustomerID varchar(10) not null,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

ChildrenID	Fname	Lname	DOB	CustomerID
ch001	kathrine	Hardy	12/07/2017	cd001
ch002	Michal	Karasick	28/04/2015	cd001
ch003	John	Smith	16/11/2018	cd002

JOIN

SELECT customers.CustomerID, Customers.FName, children.Fname

FROM children

	CustomerID	FName	Fname
▶	cd001	Ana	kathrine
	cd001	Ana	Michal
	cd002	Antonio	John

GROUPBY WITH HAVING

SELECT COUNT(ReservationID),Place

FROM reservation

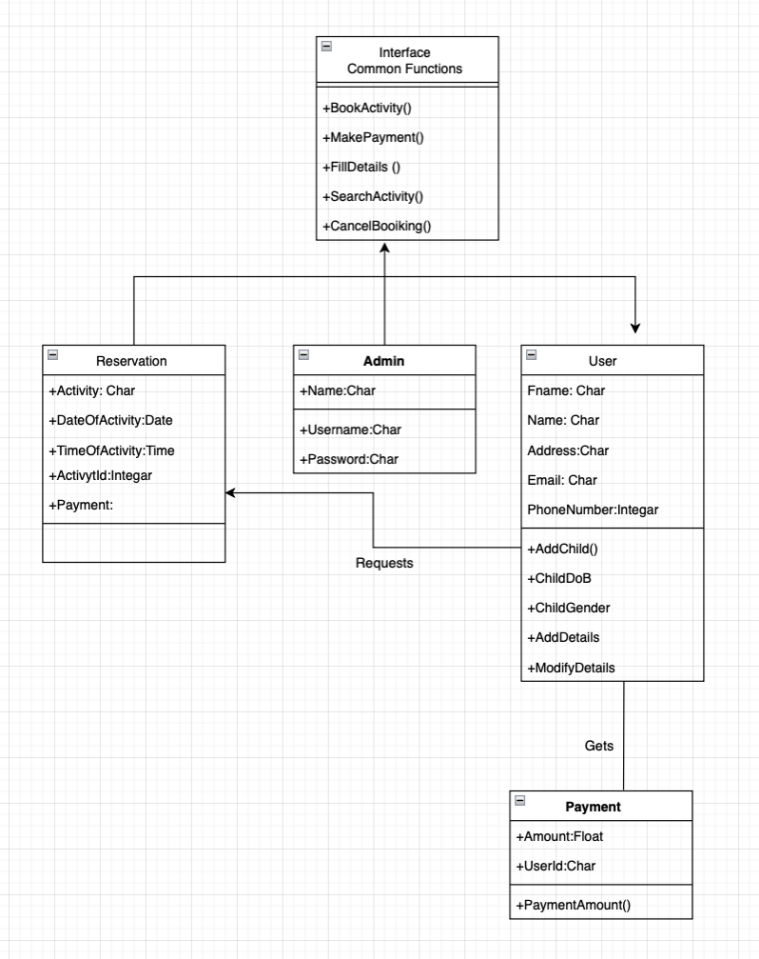
WHERE Participants>2

GROUP BY Place

HAVING COUNT(ReservationID)>1 ;

	COUNT(ReservationID)	Place
▶	2	Mexico
	3	London

Database (Sugeeth Suresh Geeta)



Tables

User table

user_id	f_name	name	address	email	phone_number
u1011	james	jamesbrown	newyork	jamesbrown7@gmail.com	9856724895
u1012	john	johnwhite	texas	johnwhite7@gmail.com	9889724895
u1013	luca	lucaluther	california	lucaluther7@gmail.com	9279724895
u1014	alex	alexantony	barcelona	alexantony7@gmail.com	9279574895
u1015	john	johnpeter	lebanon	johnpeter7@gmail.com	9276974895
NULL	NULL	NULL	NULL	NULL	NULL

Payment table

	id	amount	user_id
▶	p1011	1000	u1011
	p1012	2000	u1012
	p1013	3000	u1013
	p1014	5000	u1014
	p1015	4000	u1015
*	NULL	NULL	NULL

Reservation table

	activity_id	activity	date_of_activity	time_of_activity	payment	user_id
▶	ac1011	direct_booking	1999-10-10	01:05:20	y	u1011
	ac1012	indirect_booking	1999-10-11	01:08:20	n	u1012
	ac1013	direct_booking	1999-10-12	02:08:20	y	u1013
	ac1014	indirect_booking	1998-10-12	03:08:20	n	u1014
	ac1015	direct_booking	1999-10-12	02:09:50	y	u1013
*	NULL	NULL	NULL	NULL	NULL	NULL

Admin table

Create and insert queries

```
create table admin (
id varchar(20) unique not null,
name varchar(20) not null,
username varchar(20) not null,
password varchar(20) not null,
primary key (id)
);
```

```
create table user (
```

```
user_id varchar(20) unique not null,
f_name varchar(20) not null,
name varchar(20) not null,
address varchar(30) not null,
email varchar(30) not null,
phone_number varchar(11) not null,
primary key (user_id)
);

create table reservation (
activity_id varchar(20) unique not null,
activity varchar(20) not null,
date_of_activity date not null,
time_of_activity time not null,
payment varchar(2) not null,
primary key (activity_id),
user_id varchar(20) not null,
foreign key (user_id)
references user (user_id)
);

create table payment (
id varchar(20) unique not null,
amount integer not null,
user_id varchar(20) not null,
foreign key (user_id)
```

references user (user_id)

);

insert into admin values('ad1011', "raju", "raju7", "abcd"), ('ad1012', "jaggu", "jaggu7", "abcde"), ('ad1013', "bheem", "bheem7", "abcdef"), ('ad1014', "chutki", "chutki7", "ab"),

('ad1015', "balu", "balu7", "abs");

insert into user values('u1011', "james", "jamesbrown", "newyork", "jamesbrown7@gmail.com", "9856724895"), ('u1012', "john", "johnwhite", "texas", "johnwhite7@gmail.com", "9889724895"),

('u1013', "luca", "lucaluther", "california", "lucaluther7@gmail.com", "9279724895"), ('u1014', "alex", "alexantony", "barcelona", "alexantony7@gmail.com", "9279574895"),

('u1015', "john", "johnpeter", "lebanon", "johnpeter7@gmail.com", "9276974895");

insert into reservation values('ac1011', "direct_booking", "1999-10-10", "01:05:20", "y", "u1011"), ('ac1012', "indirect_booking", "1999-10-11", "01:08:20", "n", "u1012"),

('ac1013', "direct_booking", "1999-10-12", "02:08:20", "y", "u1013"), ('ac1014', "indirect_booking", "1998-10-12", "03:08:20", "n", "u1014"),

('ac1015', "direct_booking", "1999-10-12", "02:09:50", "y", "u1013");

insert into payment values('p1011', 1000, "u1011"), ('p1012', 2000, "u1012"), ('p1013', 3000, "u1013"), ('p1014', 5000, "u1014"), ('p1015', 4000, "u1015");

●Two triggers

Trigger to validate phone number:

DELIMITER //

Create Trigger before_insert_user_phone_number BEFORE INSERT ON user FOR EACH ROW

BEGIN

IF CHAR_LENGTH(NEW.phone_number) > 10 THEN SET NEW.phone_number = "invalid";

END IF;

END //

```
DELIMITER //
Create Trigger before_insert_customer_phone_number BEFORE INSERT ON user FOR EACH ROW
BEGIN
IF CHAR_LENGTH(NEW.phone_number) > 10 THEN SET NEW.phone_number = "invalid";
END IF;
END //
```

Output

Action Output

#	Time	Action	Message
1	21:36:08	Create Trigger before_insert_customer_phone_number BEFORE INSERT ON user FOR EACH ROW BEGIN IF CHAR_LENGTH(NEW.phone_n...	0 row(s) affected

insert into user(user_id, f_name, name, address, email, phone_number) values('u1016', "james", "jamesbrown", "newyork", "jamesbrown7@gmail.com", "invalid");

select * from user;

```
insert into user(user_id, f_name, name, address, email, phone_number) values('u1016', "james", "jamesbrown", "newyork", "jamesbrown7@gmail.com", "invalid");
select * from user;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

user_id	f_name	name	address	email	phone_number
u1011	james	jamesbrown	newyork	jamesbrown7@gmail.com	9856724895
u1012	john	johnwhite	texas	johnwhite7@gmail.com	9889724895
u1013	luca	lucaluther	california	lucaluther7@gmail.com	9279724895
u1014	alex	alexantony	barcelona	alexantony7@gmail.com	9279574895
u1015	john	johnpeter	lebanon	johnpeter7@gmail.com	9276974895
u1016	james	jamesbrown	newyork	jamesbrown7@gmail.com	invalid
NULL	NULL	NULL	NULL	NULL	NULL

Trigger to validate duplicate email

DELIMITER //

Create Trigger before_insert_user_phone_number BEFORE INSERT ON user FOR EACH ROW

BEGIN

IF EXISTS (select * from user where user.email=NEW.email) THEN SET NEW.email = "duplicate";

END IF;

END //

```
DELIMITER //
Create Trigger before_insert_user_phone_number BEFORE INSERT ON user FOR EACH ROW
BEGIN
IF EXISTS (select * from user where user.email=NEW.email) THEN SET NEW.email = "duplicate";
END IF;
END //
```

Output

Action Output

#	Time	Action	Message
1	22:20:12	Create Trigger before_insert_user_phone_number BEFORE INSERT ON user FOR EACH ROW BEGIN IF EXISTS (select * from user where user...	0 row(s) affected

insert into user(user_id, f_name, name, address, email, phone_number) values('u1017', "james", "jamesbrown", "newyork", "jamesbrown7@gmail.com", "invalid");

select * from user;

```
insert into user(user_id, f_name, name, address, email, phone_number) values('u1017', "james", "jamesbrown", "newyork", "jamesbrown7@gmail.com", "invalid");
select * from user;
```

user_id	f_name	name	address	email	phone_number
u1011	james	jamesbrown	newyork	jamesbrown7@gmail.com	9856724895
u1012	john	johnwhite	texas	johnwhite7@gmail.com	9889724895
u1013	luca	lucaluther	california	lucaluther7@gmail.com	9279724895
u1014	alex	alexantony	barcelona	alexantony7@gmail.com	9279574895
u1015	john	johnpeter	lebanon	johnpeter7@gmail.com	9276974895
u1017	james	jamesbrown	newyork	duplicate	invalid

#	Time	Action	Message
1	22:20:12	Create Trigger before_insert_user_phone_number BEFORE INSERT ON user FOR EACH ROW BEGIN IF EXISTS (select * from user where us...	0 row(s) affected
2	22:20:38	insert into user(user_id, f_name, name, address, email, phone_number) values(u1017', "james", "jamesbrown", "newyork", "jamesbrown7@gmai...	1 row(s) affected
3	22:20:38	select * from user LIMIT 0, 1000	6 row(s) returned

●<u>Two views</u>

View for showing users from a particular location

CREATE VIEW newyork_users AS

SELECT * FROM user WHERE address = 'newyork';

SELECT * FROM newyork_users;

```
CREATE VIEW newyork_users AS
SELECT * FROM user WHERE address = 'newyork';

SELECT * FROM newyork_users;
```

user_id	f_name	name	address	email	phone_number
u1011	james	jamesbrown	newyork	jamesbrown7@gmail.com	9856724895
u1017	james	jamesbrown	newyork	duplicate	invalid

#	Time	Action	Message
1	22:25:31	CREATE VIEW newyork_users AS SELECT * FROM user WHERE address = 'newyork'	0 row(s) affected
2	22:25:31	SELECT * FROM newyork_users LIMIT 0, 1000	2 row(s) returned

View to show customers with payment amount above a particular limit

CREATE VIEW cost_filter AS

SELECT u.f_name, u.name, u.address, u.phone_number, p.amount from user as u join payment as p on u.user_id=p.user_id where p.amount > 1000;

SELECT * FROM cost_filter;

```
1 •  CREATE VIEW cost_filter AS
2    SELECT u.f_name, u.name, u.address, u.phone_number, p.amount from user as u join payment as p on u.user_id=p.user_id where p.amount > 1000;
3
4 •  SELECT * FROM cost_filter;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

f_name	name	address	phone_number	amount
john	johnwhite	texas	9889724895	2000
luca	lucaluther	california	9279724895	3000
alex	alexantony	barcelona	9279574895	5000
john	johnpeter	lebanon	9276974895	4000

cost_filter 5

Output

Action Output

#	Time	Action	Message
1	22:37:11	CREATE VIEW cost_filter AS SELECT u.f_name, u.name, u.address, u.phone_number, p.amount from user as u join payment as p on u.user_id...	0 row(s) affected
2	22:37:11	SELECT * FROM cost_filter LIMIT 0, 1000	4 row(s) returned

Innovation

You usually come to a point when using software where you need assistance from the developers sooner or later. Nothing is more aggravating in those situations than not being able to acquire the assistance you expect or even assistance that was promised from the software house. Frequently, a thorough Help database is enough for you to immediately find the answer to issues with bookings, reservations, or capacity. But if that's not the case, you ought to be able to get in touch with the developer over the phone and online. One piece of advice: test the Support function as well when evaluating the booking software.

Conclusion (Sharon Farrell)

As we can see, there are many ways the client's business can benefit from investing in an online booking system. Not only will bespoke booking software increase their bookings (general admission slots and parties), but it will also help with scheduling, allowing for the better planning of resources, with reduced admin costs. The advancement in Reservation Technology is improving the system and makes booking easy and more attractive to the customer.

Using online booking software will also help the company to plan their resources more effectively. For example, for the coffee shop, the online booking software will be able to tell how many extras (food platers) they have and how many people they need to cater for it will show the number of parties on a given day and the guest attending. This will make scheduling staff easier, as they will be able to work out how many people they need on the play structure, kitchen, reception and parties. Better planning of resources will prevent time delays through staff shortages and enable them to keep their customers 100% satisfied and create a good relationship between the customers and staff

Investing in an online booking system will prove to offer an excellent return on investment, as not only will it help the business to increase the number of bookings they take, but also allow them to make a higher profit. The great thing about online booking systems and technology is that they can also be used to upsell their products or services, advertising things like add-ons and extras to customers. These may be things that their staff would forget to mention or not feel comfortable with, if they were taking a

booking over the phone during a busy period. Customers are also more likely to upgrade their booking and buy extras if they can do it online, as they will have more time to make their decision and the design of the system could make it very attractive to the customers.

References

https://www.checkfront.com/blog/what-is-an-online-booking-system/ author Kyla Steeves, October 1, 2020

Personal Experiences

Sharon Farrell Student No: 10581193

On researching this project and assignment I realised that there would be a lot of work and decided to set up a GitHub repository so that we could as a group contribute to the documents in the ReadMe file. This would allow us to follow work that each had done or completed. I was difficult to decide on who would do which part so that the marks given would be even throughout the assignment. We made a couple of zoom calls as we were not always in college together. We discussed the details of the project and what was involved and chose an element of the project we felt we could achieve.

I took the technical document where I described the project in detail and the relationships between each class and what it was, we were hoping to achieve. The project was based on an online booking system for a Family Entertainment Centre. I researched what other similar companies were doing in this industry and what functions they were providing for their customers. With this in mind I set out the entity relationships of a customer going to the website to make an online booking from the viewing of the website and the process that this customer would go through to choose a product make the reservation, make the payment and receive a confirmation email. I looked at each function individually and discussed all use cases with some colleagues of mine to see if they had any ideas or changes, they would make. I also discussed this with the other team members. The result was the entity diagrams and all the use cases that could come out of this.

I then set about pulling all the information together and structuring the report.

On reflection of both the assignment and the module I have found working in a team of 3 quite interesting, as the other members of the team did not come to classes personally and did a lot of their classes on teams, this meant that a lot of what would normally be discussed in person was left to the individual to work out, It was difficult to get a time that suited everyone to see over zoom or teams. If all teams' members attended class personally it would have been easy to discuss issues, we were having. I fully understand that due to Covid a lot of classes in previous semesters were online, but I feel that if you commit to doing a level 9 degree there should be a certain level of engagement.

While the lecturer did their best to have people engage online there was very little interaction even while on zoom members of the class did not turn on their cameras and engaged in no conversation. There is only so much one can do to help others but if there is no participation coming from certain members it is hard to help. Even with the report on this assignment it has been difficult to get the configuration of the document together.

Arivoli Ravichandran

We are a group of 3 people, Sharon Farrell, Sugeeth Suresh Geetha, and myself Arivoli Ravichandran. None of us know each other's before me and Sugeeth came to this country, Sugeeth and I are from India and Sharon is from Ireland, we 3 met at DBS and when everyone one was trying to form a group for all the subject's class assignments, we somehow formed a group.

We worked in the same group in two other subjects and the database is the reason for all three of us to group up because this is the first subject we formed into a team.

We sat and spoke about the assignments in our break times in and we also set up zoom meetings and discussed what we are going to do as a project. All of us had many different ideas and we were a bit confused for a few days on which topic to choose as our group project. And then we decided to do a database for a family environment company, and we started discussing who we are going to do this project with. Later as the semester went on we found it difficult to focus on the one subject project because the it felt like the time started moving faster and we had a lot of work to do in all the subjects, meanwhile I personally too got some issues like I got too depressed because this is the first time I've moved out of India and the first time being alone without my family in the 22 years of my life, this 3 to 4 months though me not just technical knowledge but also many life lessons. But I always had my team members by my side supporting me.

And as the time passed, we came back to work hard on our projects I heard all my classmates say they have not slept properly for more than a week because we all had our assignments submissions days coming closer for all the subjects together at the same time. So, we had to break our project work into different parts so all group members can contribute equally and also the work gets done faster.

To work in the database assessment, we discussed a lot about who's doing what work for the assignment and finally decided that Sharon will do the Technical Report, I will work on building the Database System and Sugeeth

Would do the part of triggers, Views to demonstrate the development of virtual tables and the Innovation.

We somehow managed to get the work done and have a proper assessment to submit and we have submitted the assessment.

It has been a wonderful and a thoughtful experience to be a part of this program of Information Systems at Dublin Business School.