



# Deliverable 6

## TEAM RED

Theodore Tsimiklis (2065694)

Jiahui Xia (2044092)

George Athanasatos (1951431)

Andrei Marinescu (1972210)

Client: Samer Assi

2022/04/06

## Table of Contents

<b>EXECUTIVE OVERVIEW .....</b>	<b>1</b>
<b>SUMMARY OF THE CLIENT .....</b>	<b>1</b>
<b>NARRATIVE DESCRIPTION OF THE DATABASE .....</b>	<b>2</b>
<b>BUSINESS PROBLEM .....</b>	<b>4</b>
Our way to resolve these problems: .....	4
<b>APPENDIX 1: DATA DICTIONARY .....</b>	<b>5</b>
<b>APPENDIX 2: ER DIAGRAM .....</b>	<b>8</b>
<b>APPENDIX 3: DESCRIPTIONS AND EXPLANATIONS OF DATABASE.....</b>	<b>11</b>
<b>APPENDIX 4: SIZE OF THE DATABASE.....</b>	<b>12</b>
<b>APPENDIX 5: THE ACCESS SPEED REQUIRED .....</b>	<b>13</b>
<b>CONFIRMATION OF EACH TEAM MEMBER .....</b>	<b>14</b>
<b>REFERENCES.....</b>	<b>15</b>

# EXECUTIVE OVERVIEW

In this delivery we develop and showcase our database. This also demonstrates the flow of our database and how it will operate.

After the completion of the database, we contacted our client to see how pleased or displeased he would be with it. He gave us some good and detailed comments as we will show soon and have/will change it to his liking. We also asked our client how many people he is planning to have on the database within the next 5 years.

## SUMMARY OF THE CLIENT

Since our last deliverable not much has changed about our client Sam. As mentioned, he is still well equipped with computer knowledge and progressing smoothly with his business. One thing our client started to do is go door to door to find clients for his business. Right now, he has signed 3 contracts and is looking to find more. Our website web application will facilitate

# NARRATIVE DESCRIPTION OF THE DATABASE

There is no change about system actors and actions in our system.

We have identified three primary roles relevant to the system. First, there is the user as a generic role; it represents anybody using the system, including anybody attempting to use it without prior authentication as well as anybody trying to perform a non-privilege-based privilege such as login, sign up or logging out. Then there are the administrators who can be thought of as the owners of the website; they have control of what is shown on the website, they can view critical information such as client profile (client information), bookings, services... Finally, the client is a role with limits who possess accounts on the website and who can set up bookings. The system will purposely maintain a separation between roles and permissions so that there are no roles and potentially hinder the website's data.

The actions an unprivileged user can take will be minimal. A viewer can view the website through clicking the links in navigation bar and choose the language which will display on the website. However, a viewer cannot make a booking or have a profile. A viewer be able to try sign up or login, it's depended on if he/she has an account.

When a viewer signup, the viewer needs to input a unique username. Otherwise, the viewer cannot complete sign up. At the same time, the viewer needs to input the password twice and, and the password need to be the same.

When a viewer provides the matching identifier and password, the viewer become a client. A client is still an unprivileged user but he/she can have more actions than a viewer. A client can view all the information about the website. A client is able to create a profile page and edit his/her

**TEAM RED**

profile. A client is able to make a booking but need to choose a service, enter contact information and pick up the time and date. A client also can add addition information(note) during making a booking. Finally, a client needs to confirm his/her booking to finish booking. A viewer also can ask question in contact us page. A client needs to input the contact information and input his/her question. Finally, a client needs to confirm all the information is correct, then the question will be sent to database and admin can see the question.

Once authenticated, if a user turns out to belong to the administrator role, he/she officially becomes an administrator. An administrator has create, update and delete permissions for services. This is means that he/she can add, remove or change a service. An administrator also has read permission for reading all the client's profile(information). One key ability for the administrator is to quickly see all the bookings or specific booking; to that end, the system will provide a number of facilities. The administrator will be able to sort bookings by alphabetical client's name as well as date. The administrator also can edit the store location, store contact phone and social media buttons.

Another key ability for the administrator is to quickly see all questions from clients. The administrator will be able to sort questions by alphabetical client's name as well as date. The administrator also can see the status of questions. He/ she can see which of client's questions have been answered and which have not been answered.

## BUSINESS PROBLEM

The painting business that we are working for has a couple of problems regarding the promotional aspect, customer experience, as well as the lack of demand for his services. There are multiple ways to resolve these issues and the application that we are meant to design for our client is designed to resolve at least the named ones.

### Our way to resolve these problems:

Since the customers are not aware of what our client has to offer to the fullest extent, he could just direct them to the site to provide better customer service. By doing so, if the customer is impressed by the end product, they can refer him through his site and acquire more customers which will raise the demand for his skills and creativity.

TEAM RED

## APPENDIX 1: DATA DICTIONARY

Database Name: painting

Data Entity: admin\_account

Field Name	Data Type	Data Format	Field Size	Description	Example
admin_account_id (PK)	Integer		11	Unique number ID for all the admins	12345
username	Varchar		255	Username for the admin which works as identification when signing in	username11
pass_hash	Varchar		255	Login password for the admin	a1s2d3

Data Entity: booking

Field Name	Data Type	Data Format	Field Size	Description	Example
booking_id (PK)	Integer		11	Unique number ID for all bookings	12345
service_id (FK)	Integer		11	This value will come from the service data entity (service_id)	12345
client_profile_id (FK)	Integer		11	This value will come from the client_profile data entity (client_profile_id)	12345
date	Date	DD/MM/YYYY	10	The booking date that the client wants to set as	09/04/2022
booking_description	Varchar		255	Here is where the client can add details on the booking	The Staining method...

## TEAM RED

**Data Entity: client\_account**

Field Name	Data Type	Data Format	Field Size	Description	Example
Email (PK)	Varchar		255	Email for the client which works as identification when signing in	username@gmail.com
pass_hash	Varchar		255	Login password for the client	a1s2d3

**Data Entity: client\_profile**

Field Name	Data Type	Data Format	Field Size	Description	Example
client_profile_id (PK)	Integer		11	Unique number ID for the client profile	12345
full_name	Varchar		255	Full name of the client to display on the profile page	Conor McGregor
Email (FK)	Varchar		255	This value will come from the client_account data entity (email)	username@gmail.com
phone	Varchar	(###)###-####	11	Phone number of the client to display on the profile page	(999)999-9999
image	Varchar		255	The image that will be displayed on the page	url

**Data Entity: question**

Field Name	Data Type	Data Format	Field Size	Description	Example
question_id (PK)	Integer		11	Unique number ID for the question that the client will ask admins	12345
client_profile_id (FK)	Integer		11	This value will come from the client_profile data entity (client_profile_id)	12345
status	Tinyint		1	This status will be set as 0 or 1 to display whether a message has been seen or not by the admin. (set as 0 originally)	0/1
question_description	Varchar		255	Here will be the question that the client will pose to the admins	Is Tim going to be there?



## TEAM RED

**Data Entity: reply**

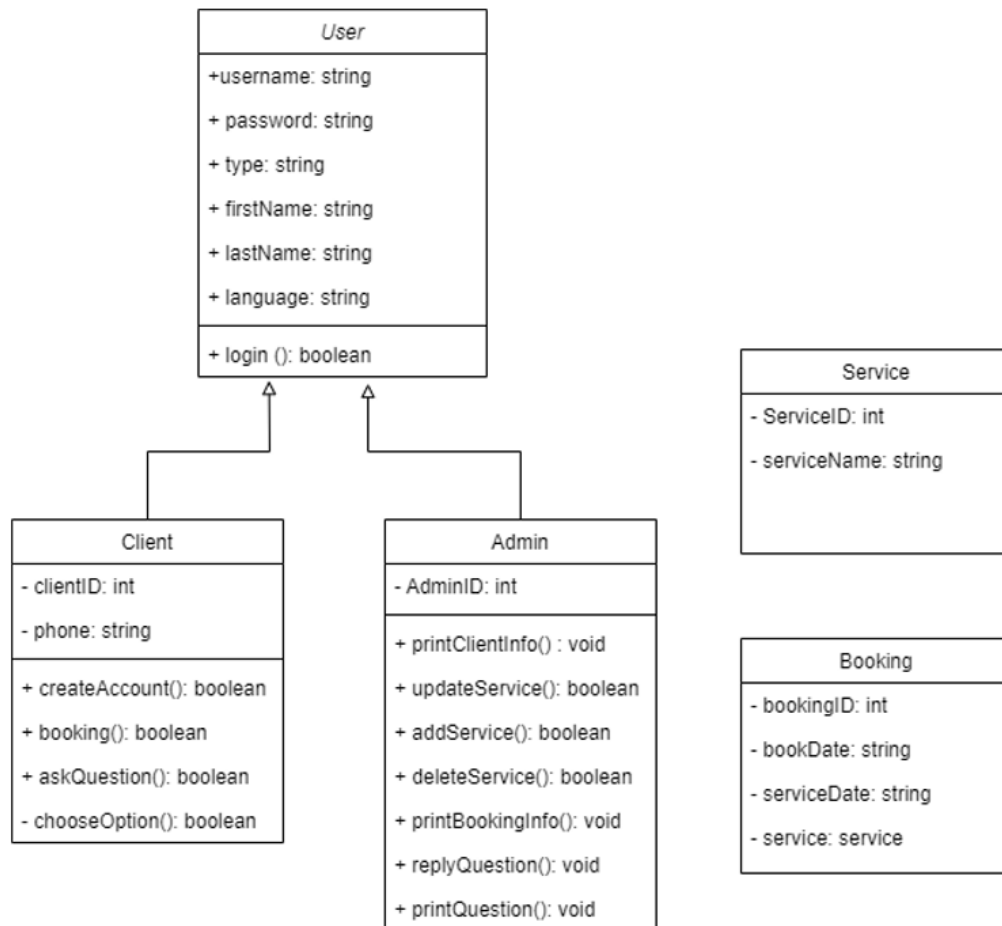
Field Name	Data Type	Data Format	Field Size	Description	Example
reply_id (PK)	Integer		11	Unique number ID for the admin's response to the client	12345
question_id (FK)	Integer		11	This question sent from the client will be retrieved from the question data entry (question_description)	12345
answer	Varchar		255	The response that the admin will submit to the client that has sent the question	Yes, Tim will be there.
admin_account_id (FK)	Integer		11	This ID will be retrieved from the admin_account data entry (admin_account_id)	12345

**Data Entity: service**

Field Name	Data Type	Data Format	Field Size	Description	Example
service_id (PK)	Integer		11	Unique number ID to distinguish which service it is	12345
service_name	Varchar		255	The name that the admin has submitted when creating a service	Staining
service_description	Varchar		255	The description that the admin will insert when creating a service	Staining is a form of...
service_image	Varchar		255	The image that the admin will insert (by url) to display the service	url

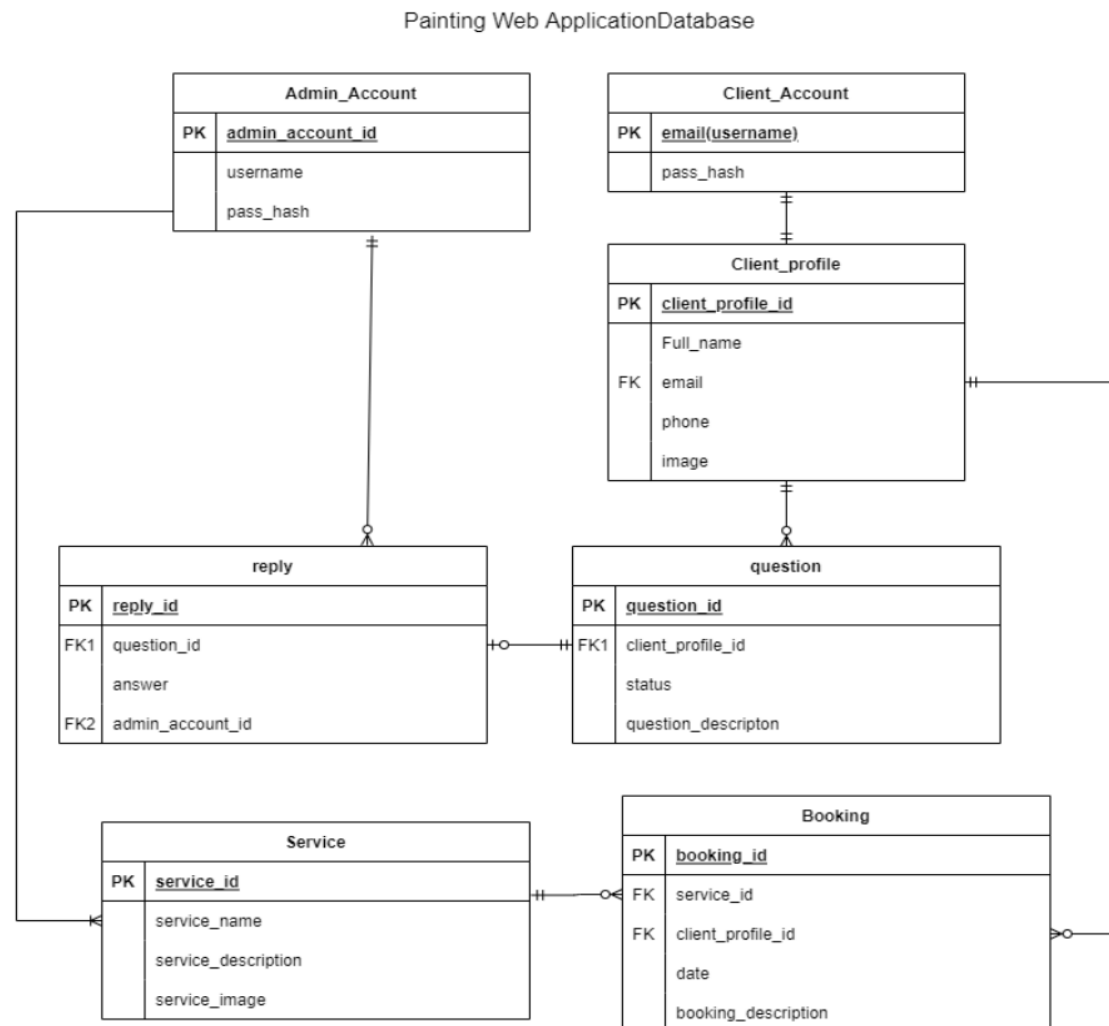
## APPENDIX 2: ER DIAGRAM

The original database from the third deliverable:  
class diagram:



## TEAM RED

Now here is the one that we have created for the implementation:



Since we developed the original database so early on, there were a lot of uncertainties in regards to the database entities as well as the contents within since we did not have a fully developed plan in mind yet. Once we had completed the previous deliverable (the mockup of the web application) we understood what needed to be completed for the current database.

---

**TEAM RED**

- **Similarities:**

- We kept the entities client and admin since those are necessary for our web application.
- The service and booking entities are quite similar to one another, but we have added more information and made the connections from one database to another more apparent.
- The logic is quite the same since it has the general aspects that we wanted and still want to develop.

- **Differences:**

- We have taken off the user entity and replaced it in a way to become the profile entity. This was also done since we wanted to distinguish one from another by making the client login with an email while the admin with a username.
- We have created the profile page so that the user can have an identity on the application.
- We have also included both question and reply entities since it would be necessary for communication between the client and admin.

## APPENDIX 3: DESCRIPTIONS AND EXPLANATIONS OF DATABASE

In the **booking** table:

Index	From	Description
service_id	service	The booking table should have a connection with the service since when booking we need a direct link to which service the client has chosen.
client_profile_id	client_profile	The client profile should be directly connected to know which client has chosen the service to be easier for the admin to know who wants what.

In the **client\_profile** table:

Index	From	Description
email	client_account	The profile should only be accessible to the client account that has been signed in. So, this profile relies on the client's account.

In the **question** table:

Index	From	Description
client_profile_id	client_profile	The question has to link to the client's profile to know who is the one that has asked the question rather than it being anonymous.

In the **reply** table:

Index	From	Description
question_id	question	To receive the question, we need to call this id so that there is a link between the 2 and so that later on the admin can view and reply to it.
admin_account_id	admin_account	The question will be answered directly from the specific admin account

At the moment, we have spent quite a bit of time on this database, optimizing as much as we can to simplify the database while keeping the logic intact, and have come to the point where we don't need optimization. As we develop further, we will see if there is any need for more optimization otherwise, we will keep it as it is right now.

## APPENDIX 4: SIZE OF THE DATABASE

We asked our client (Sam) how many clients he believes his database would need to hold throughout the next 5 years. He believes that the first few years will be kind of slow, since they are a new business and there is competition in this area of work. Then he believes his business will pick up by year 3, because this is where he will start to make connections

$$\text{Null\_Bitmap} = 2 + ((5+7) / 8) = 3.5$$

$$\text{Variable data size} = 2 + (1 \times 2) + 255 + 2 + (2 \times 2) = 409$$

$$\text{Row\_Size} = 33 + 409 + 3.5 + 4 = 449.5$$

$$\text{Rows\_per\_page} = 8096 / (449.5 + 2) = 17$$

$$\text{Num\_Pages} = 0/20$$

$$\text{Heap size} = 208 / 0 = 0 \text{ bytes}$$

Total size of current DB: 208 kilobytes

1<sup>st</sup> year: 20 people

2<sup>nd</sup> year: 40 people

3<sup>rd</sup> year: 100 people

4<sup>th</sup> year: 150 people

5<sup>th</sup> year: 250 people

## APPENDIX 5: THE ACCESS SPEED REQUIRED

The access speed would have to be relatively fast because, if a customer had submitted a question, they would want it to be answered pretty quickly. Around 200ms fast. We designed our database in way that when it is accessed by Sam, he will be able to see a table of what database he would want to check for example if he wanted to see the questions he would click in the table (on the right page) to view the questions. In Sam's case, he would need to access the database frequently to view all the necessary information. A lot of data would most likely be because most of the things the customers are going to do are going to go directly to the database and will be stored there. The response time all depends on Sam and his responsiveness to the web application.

TEAM RED

## CONFIRMATION OF EACH TEAM MEMBER



Signature:

Printed Name: Theodore Tsimiklis

Date: 2022-04-06



Signature:

Printed Name: Jiahui Xia

Date: 2022-04-06



Signature:

Printed Name: George Athanasatos

Date: 2022-04-06



Signature:

Printed Name: Andrei Marinescu

Date: 2022-04-06



## References

*30 examples of Business Problems*. Simplifiable. (n.d.). Retrieved February 17, 2022.

<https://simplifiable.com/new/business-problems>