

Vanier College  
Faculty of Science and Technology  
System Development  
420-436-VA

## **Deliverable 06**

Green Team  
Andy-Tason Nguyen-Chao  
Damiano Visalli  
Md Saqliyan Islam  
Uraib Ahmed Lakhani  
Yassine El Yamani

In collaboration with Muhammad Zahid Hussain

**We, the Green Team, certify that this assignment is our own work**

I, Andy-Tason Nguyen-Chao, student ID# 2071047, certify that I have contributed to this deliverable, A.T.N.C.

I, Md Saqliyan Islam, student ID# 2062427, certify that I have contributed to this deliverable, M.S.I.

I, Uraib Ahmed Lakhani, student ID# 2049924, certify that I have contributed to this deliverable, U.A.L

I, Damiano Visalli, student ID# 2062581, certify that I have contributed to this deliverable, D.V

I, Yassine El Yamani, student ID# 2059045, certify that I have contributed to this deliverable, Y.E

# Table Of Contents

---

<b>Table Of Contents</b>	<b>2</b>
<b>Executive Overview</b>	<b>3</b>
<b>Business Problem</b>	<b>4</b>
<b>Narrative Description of the Database Design</b>	<b>5</b>
<b>Appendix 01</b>	<b>6</b>
Client Table	6
Admin table	7
Message Table	8
Booking Info Table	9
Type Table	10
Destination Table	10
<b>Appendix 02</b>	<b>11</b>
Diagrams	11
Similarities and Changes	12
<b>Appendix 03</b>	<b>13</b>
<b>Appendix 04</b>	<b>14</b>
<b>Works Cited</b>	<b>17</b>

# Executive Overview

---

This deliverable's document will go over the business problem that remains the same as the previous deliverables, the narrative description of the database design of our project and the general information of the database system imposed.

As explained, the business problem remains unchanged as of now with the same problems listed as before and the documentation of the present system.

The narrative description of the database design will go through the interactions of all actors using the web application with the database. These would be the admin's, customer's and user's journeys with the database of the project's prototype, including CRUD operations.

The appendices listed below are the documentations of the database system which goes as is: Appendix 01 will be the data dictionary with its entities, their relations and their attribute types; Appendix 02 will showcase both an ER diagram of the database supporting the information system with its respective cardinalities and the description of the similarities and the changes of the class diagram of the Deliverable 03 (Appendix 05); Appendix 03 will be describe and explain the query optimization of the product's design; Appendix 04 goes through the explanation of the access speed required for the database and how it will harmonize with the design.

# Business Problem

---

The business problems remain the same since the last deliverable (05):

- It is to replace the current uncompleted website of the agency
- There is no function/process to collect customers' information/request to make a travel quote
- There is no function for the agency to return responses to customers, whether it is a quote or response from "Contact Us"
- There is no ability to gather leads for marketing with proper consent
  - An example of this may be requesting an email for promotions at checkout.
  - In need of a search engine optimization (*SEO*)

Our client's business service goes as is: Customers request quotes with their provided information (Number of People, Place of Travel, Hotel Requirement); The Agency collects said information; The Agency processes the information through the *Sabre* Software to make a quote; The Agency sends the created quote to the customer; The customer pays the received quote.

# Narrative Description of the Database Design

---

In the current system, there are a total of three actors: Admin, Customer and User. Both actors have their respective tasks and importances in the system. Admin will have access to view, edit and delete all entries of all the tables, which includes customers' profiles, booking requests made by those customers and customer service messages. Whereas the customer will only have access to view and edit their own profile and make requests.

First, the Admin will have to do little insertion to the database, but will rather view the entries and delete them in order to aid their process in supplying their service. Admin will be able to view customers' profiles information to be able to contact them. Viewing the booking requests made by customers will help in the process of inputting information in the *Saber/Asmodeus* software. Accessing the messages from the customer service will aid in answering the messages left by customers and general users about various topics.

Second, the Customer will be able to register an account to the database and edit said account, they have to enter their username, full name, email, phone number, and a password. They will only have the permission to edit their email and phone number. The main functionality of the customer is to create a booking request that will be inserted to its respective table. To do that, they will have to enter the places from departure and destination, the set dates, the number of each type of passenger, the type of trip, whether it is a single trip or trip over multiple destinations.

Last will be the general user which will only have the ability, that is also shared with the customer, to send messages. These messages will mostly entail questions. This will require the input of the user's name, email, phone, the subject and content to be sent. The difference with the customer will be the auto-input of the personal information.

## Appendix 01

---

### Client Table

Field name	Data type	Field size	Constraints	Description	Example
client_id	int	5	Primary Key	A unique ID for each client	00001
username	varchar	50	Not null	The username which a client will use to login to the web application	alex2089
password_hash	varchar	60	Not null	A hash generated to avoid storing password as plain text	5994471abb01112afcc18159f6cc74b4f511b99806da59b3caf
fName	varchar	50	Not null	The first name of the client	Andy
lName	varchar	50	Not null	The last name of the client	Chao
email	varchar	50	Not null	The email address of a client	johnsmith@gmail.com
phone_num	varchar	20	Not null	The phone number of the client which will be used by admin to contact the client	4383787895

## Admin table

Field name	Data Type	Field Size	Constraints	Description	Example
Admin_id	int	5	Primary Key	A unique Id for each admin	00001
username	varchar	30	Not null	The username which an admin will use to login to the website	ZawssAdmin
password_hash	varchar	60	Not null	A hash generated to avoid storing password as plain text	5994471abb01112afcc18159f6cc74b4f511b99806da59b3caf



## Message Table

Field name	Data Type	Field Size	Constraints	Description	Example
message_id	int	5	Primary Key	A unique Id for each admin	00001
fName	varchar	50	Not null	The first name of the client	Andy
lName	varchar	50	Not null	The last name of the client	Chao
email	varchar	50	Not Null	The email address of the user	johnsmith@gmail.com
phone_num	int	20	Not Null	The phone number of the user which will be used by admin for contacting	4383787895
content	TEXT	1000	Not Null	The message the user wants to send to the admin	Hi, how are you
dateSent	date	10	Not Null	The date when the user sent the message	13/04/2010

## Booking Info Table

Field name	Data Type	Field Size	Constraints	Description	Example
book_id	int	5	Primary Key	A unique id used to identify a booking	00001
client_id	int	5	Foreign Key	A unique id used to identify the client	00001
destination_id	int	5	Foreign Key	A unique id used to identify a destination	00001
flight_date	date	12	Not null	The date on which the client wants the flight	14/03/2020
return_date	date	12	Not null	The date on which the client is expected to return	23/05/2022
nbAdults	int	2	Not null	The number of adult passengers	03
nbChildren	int	2	Not Null	The number of children passengers	02
nbInfants	int	2	Not null	The number of infants passengers	01
type_id	int	5	Foreign Key	The type of trip that the client wants	00001

## Type Table

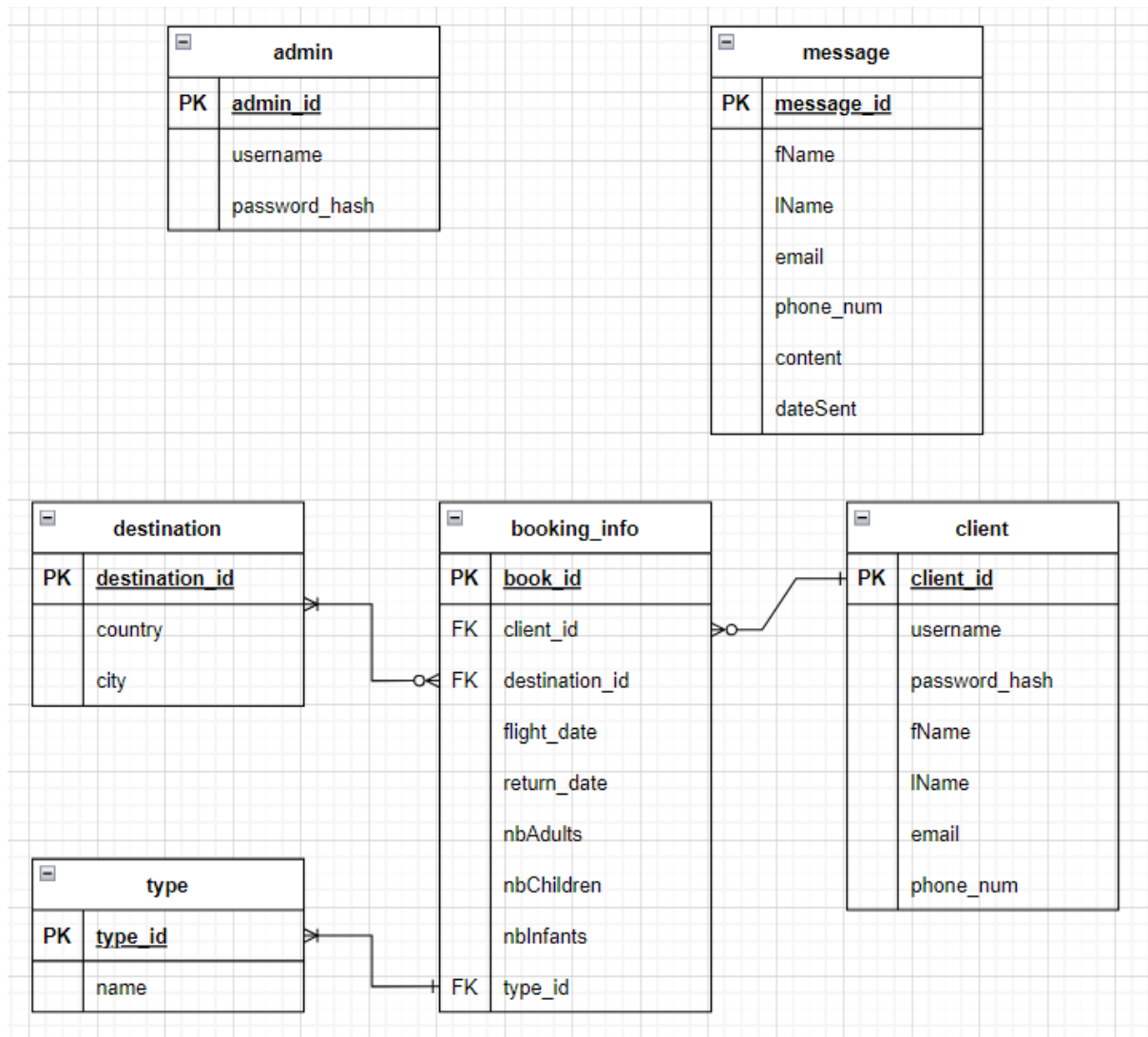
Field name	Data Type	Field Size	Constraints	Description	Example
type_id	int	5	Primary Key	ID to identify type of trip	00001
name	varchar	25	Not null	The name of the type of trip	Student

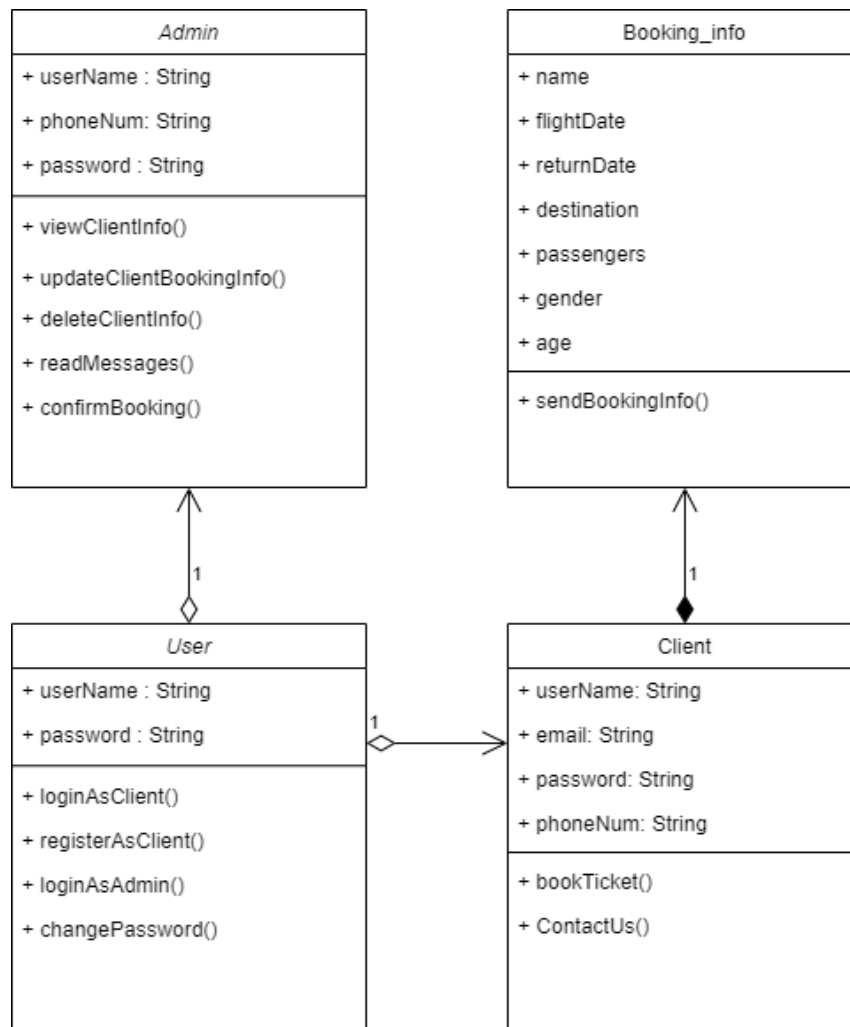
## Destination Table

Field name	Data Type	Field Size	Constraints	Description	Example
destination_id	int	5	Primary Key	A unique Id for each destinations	00001
country	varchar	30	Not null	The country of the destination	Italy
city	varchar	30	Not null	The city of the destinationt	Rome

## Appendix 02

### Diagrams





## Similarities and Changes

The diagrams shown above are an updated ERD and the class diagram from Deliverable 03 (Appendix 05).

There are a lot of differences between both as the structure of the database has mostly changed due functionalities being more specific. The *User* parent class has been removed as there would be little admin to be added to the system. The *Client* table was more defined with the name of the client. *Booking\_info* table was also changed to specify the details that are going to be used in the calculation of the quote given. Except being both fundamentally different types of diagram showing different details, they remain somewhat similar. There are now have 2 additional tables to our ERD: the *Destination* table and *Type* table. The *Destination* table allows the customer to choose destinations from a predetermined list and also allows for the admin to add new destinations. This brings about a change in the *Booking\_info* table where the destination field is now *destination\_id*. The *type* table allows the customer to chose what category (adult, student, elderly, etc) they fit into in order to possibly get discounts. This table brings a change to the booking table by adding the *type\_id* foreign key field, it also removes the need for the age field in the table.

## Appendix 03

---

Fixing and preventing performance problems is critical to the success of any application. We will use a variety of tools and best practices to provide a set of techniques that can be used to analyze and speed up any performance problem.

To optimize our data, we'll use **indexes**:

- On the booking table, we will use the first index, called (booking\_idx). This index will make the bookings appear faster in the admin panel.
- We'll use the second index on the client table. Our search algorithm will be sped up by this index, so the results will appear more quickly.

### Query optimization

We will have many queries in our design, we will do the following steps to optimize our queries:

- The first optimization we will use is to limit the number of bytes it needs to scan, so that way the data is processed faster than usual. An example of that is selecting only the needed columns instead of selecting every column in the table. Instead of doing this: `SELECT * FROM tablename`, we'll do this: `SELECT (specific COLUMN_NAME) FROM tablename`.
- To have as little impact as possible on the database performance, we will use `INNER JOIN` rather than `WHERE`.

### Normalization

After reviewing the ERD table, we normalized the diagram first to 1NF by making sure the columns in the tables have unique names, the tables have single valued attributes/columns and there are no repeating groups. There were two duplicate fields in our table. First we were storing the password along with the password hash which we removed. After the table was in 1NF we normalized it to 2NF by making sure the table does not contain partial dependency, by removing duplicate and redundant data. In our admin table we were storing the phone number of admin which is unnecessary and redundant data therefore we removed it. Finally we normalized our table to 3NF by making sure there are no transitive dependencies.

## Appendix 04

---

### Speed required:

We have no way of determining traffic, and the best way to manage it is to limit the number of users who can access the website at the same time. If Zahid's web application is representative of a business, this shouldn't be a problem for him.

As a company becomes more well-known, various precautions will need to be taken, such as using a cloud-based server to guarantee that all users can access the website without interruption.

The tables are normalized for best performance and to minimize querying interference. The statistics that the website and database are expected to have were also taken into account when building the database.

### Database traffic:

<b>Admin table</b>	
<b>Row name</b>	<b>Size</b>
admin_id	5
username	30
password_hash	72
<b>Total size:</b>	107 bytes

<b>Client table</b>	
<b>Row name</b>	<b>Size</b>
client_id	5
username	30
email	50
password_hash	72
phone_num	11
first_name	25
last_name	25
<b>Total size:</b>	<b>218 bytes</b>

<b>Booking_info table</b>	
<b>Row name</b>	<b>Size</b>
book_id	5
client_id	5
first_name	25
last_name	25
flight_date	3
return_date	3
destination	50
nb_adults	11
nb_children	11
nb_infants	11
type	20
<b>Total size:</b>	<b>218 bytes</b>

<b>Message table</b>	
<b>Row name</b>	<b>Size</b>
message_id	5
first_name	50
last_name	50
email	50
phone_num	20
content	1000
dateSent	3
<b>Total size:</b>	<b>1178 bytes</b>



<b>Type table</b>	
<b>Row name</b>	<b>Size</b>
type_id	5
name	25
<b>Total size:</b>	<b>30 bytes</b>

<b>Destination table</b>	
<b>Row name</b>	<b>Size</b>
destination_id	5
country	30
city	30
<b>Total size:</b>	<b>65 bytes</b>

## Works Cited

---

<https://www.lifewire.com/database-normalization-basics-1019735>

<https://www.techopedia.com/definition/22561/third-normal-form-3nf>