CONCEPTION PHASE

The project aims to create a database for renting a place to stay like apartments or bedrooms. This database is the data persistence part of the platform users' input, which is used for the back-end. The platform aims to contract between host and guest booking; therefore, the database is based on the user role to keep transactions. Some restrictions need to have functioned with the front-end of the platform, for-example host is not paid until twenty-four hours after the guest is satisfied with what is described on a platform and what s/he sees on the facility. In another example of front-end restrictions functionality, the host has to provide at least one picture and phone, but the guest has to provide more information than that to be a verified user. This project will not focus on how to build front-end applications, but on how to make data persistent by using SQL (Structured data Query Language) based database for a platform that is part of the back-end. The problem description for the project can be summarized as, keeping records of the users. The host has the facility, for example, s/he wants to keep records of the facility locations, guest records, and their details such as invoice order. Also, guest can keep their order records. The advantages of this method is to have historical data of the facility's and historical data of guests'. Disadvantages of this method is that user need to spend some time to enter input and also facility administrator need to spend some time to enter input to the platform as well as this can be employed an extra employee for the facility. After the problem is defined well, there will be ERM (Entity Relationship Model) for the emphasized database. The figure will be placed on the last page of the conception phase. There will be twenty-one entities in the database model, each database has its structured data to increase efficiency so that the use case can be more scalability and can be run faster while querying through SQL. No-SQL based database systems will not be used on this project as SQL based solutions widely used on industry. The list of entities and a short description of tables are shown below:

No	Tables	Description
1	userRole	It defines user role e.g., Guest/Host
2	userTaxId	User Tax Id for invoices
3	bank	To keep record of bank accounts
4	carrier	To keep details of shipment
5	expectedIncome	Host may want to analyze expected and real income
6	facilityRate	To keep record of guest rate for the facility
7	facilityOrder	Facility orders
8	facilityLocation	Geo-location of facility
9	facilitySocial	Saves facility social media links
10	facilityArchitecture	Architectural properties
11	facilityPrice	Keep records of facility price per day
12	facilityInvoice	facility invoice belongs to user
13	facilityAvailability	keep availability dates
14	facility	place to be stay
15	facilityReview	Comments of guest to be saved
16	facilityTaxRate	tax rate for invoice
17	facilityOverDayStayed	guest can stay more than planned as days
18	facilityAdvantages	advantages of facility

19	accountPayable	payments details
20	accountReceivable	receive details
21	employee	employee details

Table 1 List of Entries

These tables are connected so that SQL queries can run via joining tables to each other. Therefore, the conception project is to build the database for CRUD (Create, Read, Update, Delete) operations. However, only Create and Read statements will be used in this project for simplicity and not to complicate the project. After the ERM (Entity Relationship Model) has been built, tables are also identified and designed with their column and data types. All primary keys are unique, not null, and auto increment in each table. Primary keys are mostly used to connect tables so that each row can be identified when SQL queries are running, there will be also foreign keys on the corresponding table while foreign keys are the primary keys of a connected table. MySQL database is used for the database, with MySQL workbench GUI (graphical user interface) for database modeling. The reason of choosing MySQL workbench GUI is to holistically have solution for the project. Int, varchar, timestamp, data types will be used for a variety of input. Relationships are mostly one-to-many but there are also many-to-many relationships that exist. On many-to-many relationships, has an extra table between two tables that will keep records of both primary keys to match records. However, these extra tables from many-to-many relationships are not included in the table list above. As Each table and its columns can be seen below, SQL statements;

userRole			**carrier**	
id_userRole	INT		id_carrier	INT
userRole	VARCHAR(45)		name	VARCHAR(50)
name	VARCHAR(45)		itemSent	VARCHAR(45)
surname	VARCHAR(45)		trackingNo	VARCHAR(25)
email	VARCHAR(45)		create_time	TIMESTAMP
telephone	VARCHAR(45)		update_time	TIMESTAMP
create_time	TIMESTAMP			
update_time	TIMESTAMP		**bank**	
			id_bank	INT
userTaxId			bankName	VARCHAR(250)
id_userTaxId	INT		iban	VARCHAR(50)
userTaxNo	VARCHAR(25)		accountHolderNameSurname	VARCHAR(50)
create_time	TIMESTAMP		create_time	TIMESTAMP
update_time	TIMESTAMP		update_time	TIMESTAMP
facilityRate			**expectedIncome**	
id_facilityRate	INT		id_expectedIncome	INT
rate	INT		expectedIncome	INT
create_time	TIMESTAMP		create_time	TIMESTAMP
update_time	TIMESTAMP		update_time	TIMESTAMP
facilityOrder			**facilityLocation**	
id_facilityOrder	INT		id_facilityLocation	INT
dayOrder	INT		latitude	VARCHAR(45)
id_userRole	INT		longitude	VARCHAR(45)

create_time	TIMESTAMP		cityName	VARCHAR(45)
update_time	TIMESTAMP		countryName	VARCHAR(45)
			create_time	TIMESTAMP
			update_time	TIMESTAMP
facilitySocial			**facilityArchitecture**	
id_facilitySocial	INT		id_facilityArchitecture	INT
facebookLink	VARCHAR(150)		hasRoom	INT
instagramLink	VARCHAR(150)		totalSq	INT
create_time	TIMESTAMP		hasBalcoon	TINYINT
update_time	TIMESTAMP		create_time	TIMESTAMP
			update_time	TIMESTAMP
facilityPrice			**facilityReview**	
id_facilityPrice	INT		id_facilityReview	INT
pricePerDay	INT		guestComment	VARCHAR(150)
create_time	TIMESTAMP		create_time	TIMESTAMP
update_time	TIMESTAMP		update_time	TIMESTAMP
facilityInvoice			**accountReceivable**	
id_facilityInvoice	INT		id_acctReceivable	INT
stayedDay	INT		amount	INT
dailyPrice	INT		orgName	VARCHAR(50)
create_time	TIMESTAMP		address	VARCHAR(50)
update_time	TIMESTAMP		create_time	TIMESTAMP
			update_time	TIMESTAMP
facility				
id_facility	INT		**facilityTaxRate**	
type	VARCHAR(45)		id_facilityTaxRate	INT
name	VARCHAR(45)		taxRate	INT
adress	VARCHAR(45)		create_time	TIMESTAMP
email	VARCHAR(45)		update_time	TIMESTAMP
telephone	VARCHAR(45)			
create_time	TIMESTAMP		**facilityOverDayStayed**	
update_time	TIMESTAMP		id_facilityOverDayStayed	INT
			overDays	INT
employee			create_time	TIMESTAMP
id_employee	INT		update_time	TIMESTAMP
name	VARCHAR(50)			
surname	VARCHAR(50)		**facilityAdvantages**	
telephone	VARCHAR(50)		id_facilityAdvantage	INT
create_time	TIMESTAMP		advantage	VARCHAR(45)
update_time	TIMESTAMP		create_time	TIMESTAMP
			update_time	TIMESTAMP
accountPayable				
id_acctPayable	INT		**facilityAvailability**	
amount	INT		id_facilityAvailability	INT
orgName	VARCHAR(50)		availableStartDate	TIMESTAMP
address	VARCHAR(50)		availableEndDate	TIMESTAMP
create_time	TIMESTAMP		create_time	TIMESTAMP
update_time	TIMESTAMP		update_time	TIMESTAMP

Table 2 list of attributes

At the below page it may be seen the Entity Relation Model.

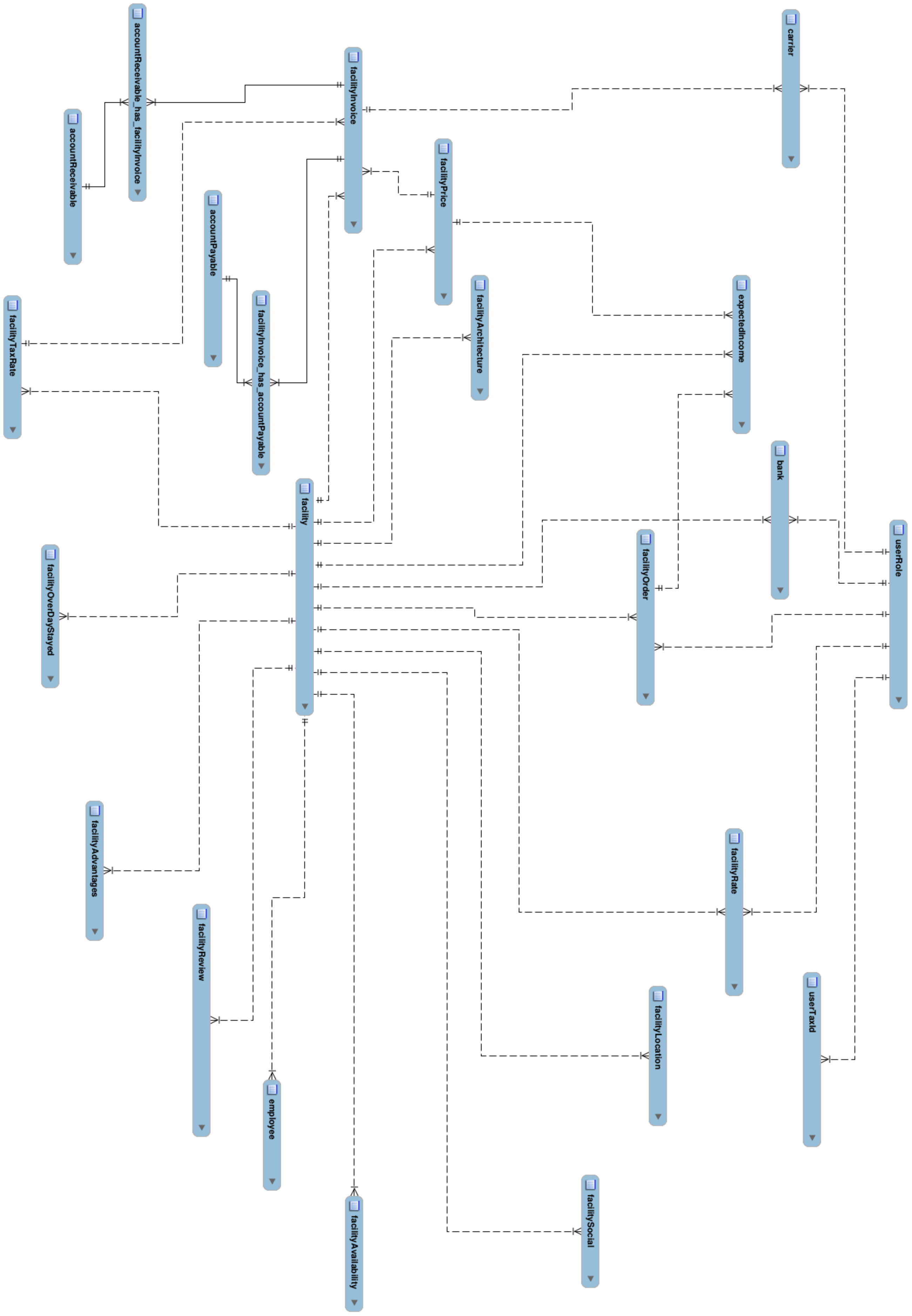

Figure 1 ERM of database