

# Applied Research

Individual Track Assignment



16/05/2022 Eindhoven  
Version: 1.0

Esther Wolfs: 3329984

Tutor:  
Tim Kurvers  
Márcio Paixão Dantas

## Version history

Version	Date	Author(s)	Changes	State
0.1	16/05/2022	Esther Wolfs	Move everything research related from design document to this document	Finished

Contents:

<b>Version history</b>	<b>2</b>
<b>1. Introduction</b>	<b>4</b>
<b>2. How should the data for reservations be stored?</b>	<b>4</b>
2.1 What database to use	4
2.2 What information is relevant	5
2.3 What user data should be stored and how should it be stored	5
2.4 How to store the availability of a room	6
2.5 How to connect a room to the reservation	6
<b>3. Conclusion</b>	<b>6</b>
<b>Works Cited</b>	<b>7</b>

# 1. Introduction

This document will be about applied research. In the first paragraph of chapter 2 the main question will be explained. The main question will be divided into smaller sub questions, that together will answer the main question. To answer these questions I will be using certain research questions that will be specified in each sub-question paragraph. The last chapter will be the conclusion of my research, here you will find the answer to the main question.

## 2. How should the data for reservations be stored?

Reservations are a big part of this project. Users should be able to make reservations for rooms they select, after applying search filters. These reservations should then be stored in a database. These questions will give answers on how to store this information, so that it can be accessed by the relevant users.

### 2.1 What database to use

To start off this research it is important to figure out what kind of database to use, as there are many different kinds, all of which have different use cases. The relevant research methods for this question that will be used are literature study.

Relational databases store data using multiple related tables, the data is stored in rows and columns. SQL (Structured Query Language) is the most common language for performing CRUD operations. Relational databases work well with structured data and are very reliable. Some examples are Microsoft SQL Server, Oracle Database and MySQL (Matillion, 2020).

Non-relational databases are all databases that don't use SQL as its primary language. Data is a NoSQL database that does not have to conform to a predefined schema. It works well for organisations that want to store semi-structured or unstructured data. One advantage is that you can make changes to the database without it affecting the application. For example Apache Cassandra, MongoDB or CouchDB (Matillion, 2020)

A cloud database is any database that is designed to run in the cloud. It is low maintenance and flexible. (Matillion, 2020).

The best kind of database for this project is a relational database, because the data is structured and stored in different tables that relate to each other. A local MySQL database will be used.

### 2.2 What information is relevant

To find out what information is relevant, the problem needs to be stated. Problem analysis is a good research method for this. Why does the application need to store information, when does it need to retrieve this data.

For every reservation, there is a lot of necessary information. These include the date the reservation is made, the check in date, the check out date, what kind of room(s) the guest

wants to book. With this information we can determine the total price of the reservation. All of this needs to be linked to a guest. The status of the reservation is also important to store, so the availability of the rooms can be updated and the guest can get a refund if the reservation gets cancelled.

Every time a guest makes a search for a room the application needs to check for available rooms for the specified dates. Whenever a guest or employee wants to manage a booking the details of this specific reservation need to be retrieved.

## 2.3 What user data should be stored and how should it be stored

Stating the problem is a good method to find out what data should be stored related to the users of the application, to find out how to securely store account information literature study is used.

This application has two different kinds of users, both of which have different functionalities. The employee and the guest. The basic information about users that needs to be stored are their name and an email address. For employees the date of birth is also stored. Guests have a list of their reservations, this is done by a one to many relationship with the reservation table, because one guest can have multiple reservations.

To use the main functionalities of the application, the user needs to have an account. Employees need this to manage bookings to perform their daily work tasks. Guests are asked to log in whenever they want to make a reservation. When a new employee is hired and they are added to the system, a user account is automatically created for them that is linked to their data. A guest can sign up for an account.

To log in to the account, the software needs to authenticate the users. This process verifies that the person is whoever they say they are. Authorisation verifies that a user has the right permission to perform a specific task or access specific data. Authorisation is done after the authentication process. (Fernigrini, 2022)

Passwords are the most common methods to confirm the identity of a user. Storing a password in plain text is not a good idea, because everyone that has access to the database can see the password of the users. The data that needs to be stored for the users consists of the following:

- Account ID
- Username
- Encrypted password
- Email
- First Name
- Last Name

(Fernigrini, 2022)

In this application the information will be split into two separate tables: one that has the personal information like the email and the name, and table that stores the username and

encrypted password. These tables are connected with a one-to-one relationship; one user account is linked to one person.

To differentiate between the users, a list of roles will be given to each user. These roles are stored in another table in the database, with a many to one relationship with the user account table. There are three role types in this application: a customer that can perform CRUD operations on their own reservation and user account, an employee that can perform CRUD operations on all reservations and their own user account and an admin user that can perform CRUD operations on all user accounts. An employee can have both the employee role, as well as an admin role. This is why a many to one relationship is necessary.

## 2.4 How to store the availability of a room

available product analysis, best and good practices, problem analysis.

The hotel only has a certain number of rooms available. Whenever a guest makes a reservation, they have to select the dates and a list of available rooms is shown. The guest then selects a room and completes their reservation. The availability of the rooms need to be stored in the database, so the application knows which rooms to show.

We need to know the check in and check out date the guest selects.

## 2.5 How to connect a room to the reservation

The two most important parts that are needed for this application are the availability of the rooms and the reservations.

# 3. Conclusion

Conclusion, answer to main question [updated later]

## Works Cited

Fernigrini, Lisandro. "How to Store Login Data in a Database." *Vertabelo*, 3 February 2022,  
<https://vertabelo.com/blog/authentication-data-storage/>. Accessed 16 May 2022.

Polito, Julie. "The Types of Databases (with Examples)." *Matillion*, 18 November 2020,  
<https://www.matillion.com/resources/blog/the-types-of-databases-with-examples>.  
Accessed 16 May 2022.