DESIGN AND IMPLEMENTATION OF A HOTEL RESERVATION FRONT-END WITH SOCIAL- MEDIA AND CUSTOMIZED WEB SERVICE

by

KRITIKAKIS EMMANOUIL

Mechanical Engineering B.A., Technical Educational Institute of Crete, 2020

A THESIS

Master Thesis submitted in partial fulfillment of the requirements for the degree
Master of SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SCHOOL OF ENGINEERING

HELLENIC MEDITERRANEAN UNIVERSITY

2023

Approved By:

Major Professor
Aivalis Konstantinos

Abstract

In the last decades, rapid technological development has caused radical changes in the hotel booking process. From the traditional method where the customer booked a hotel via their physical presence to the accommodation or a phone call, we have reached the point where room reservations are made up to the last minute.

Nowadays there are many different systems (Hotel Distribution Channels) which aim to manage or sell hotel rooms online, some of those are simple hotel websites, online travel agents (OTAS) e.g., booking.com, Expedia and even global distribution GBD systems. There are also programs that allow hotels to simultaneously manage several distribution channels (channel managers). In modern times you find it necessary for a hotel to promote its rooms through these channels so that it can be competitive with other hotels and increase its profits.

In this work an online booking platform was developed that has the role of an OTA and acts as an intermediary between hotels and customers with the aim of selling rooms. Also, we use the system as a marketing tool by advertising the hotels on a Facebook page through the Facebook graph API and the Spring social framework System also gives hotel owners the ability to manage some of the basic functions of their hotel through the website. Its innovation is the creation of a customized web service that allows real-time communication between the system and a RMS (Reservation Management System).

The application consists of two main components: the client-side application built using the AngularJS Framework and the server-side application built with the Spring boot Framework.

Περίληψη

Τις τελευταίες δεκαετίες η ραγδαία τεχνολογική ανάπτυξη έχει προκαλέσει ριζικές αλλαγές στην διαδικασία κλεισίματος δωματίου σε ένα ξενοδοχείο. Από την παραδοσιακή μέθοδο οπού ο πελάτης έκλεινε ξενοδοχείο με μια επίσκεψη στο κατάλυμα ή ένα τηλεφώνημα έχουμε φτάσει στο σημείο να γίνονται κρατήσεις δωματίων μέχρι και το τελευταίο λεπτό.

Στην σημερινή εποχή υπάρχουν πολλά διαφορετικά συστήματα (Hotel distribution channels) τα οποία έχουν σκοπό να διαχειριστούν ή πουλήσουν δωμάτια ξενοδοχείων στο διαδίκτυο, κάποια εκ των οποίων είναι απλές ιστοσελίδες ξενοδοχείων , online ταξιδιωτικοί πράκτορες (OTAS) πχ booking.com, Trivago, Expedia μέχρι και global distribution systems GBD. Υπάρχουν επίσης προγράμματα που δίνουν την δυνατότητα σε ένα ξενοδοχείο να διαχειριστεί ταυτόχρονα τέτοια κανάλια (channel managers).

Στην σύγχρονη εποχή θεωρείτε απαραίτητο ένα ξενοδοχείο να προωθεί τα δωμάτιά του μέσω τέτοιον καναλιών ούτως ώστε να μπορεί να είναι ανταγωνιστικό σε σχέση με τα άλλα ξενοδοχεία και να αυξήσει τα κέρδη του.

Σε αυτή την εργασία αναπτύχθηκε μια διαδικτυακή πλατφόρμα κρατήσεων που παίζει τον ρόλο ενός OTA και λειτουργεί ως μεσάζοντας μεταξύ ξενοδόχων και πελατών με στόχο να πουλήσει δωμάτια. Ταυτόχρονα χρησιμοποιούμε το σύστημα και σαν εργαλείο μάρκετινγκ διαφημίζοντας τα ξενοδοχεία σε μια σελίδα στο Facebook μέσω του Facebook graph API και του spring social framework.

Επίσης στους ιδιοκτήτες ξενοδοχείων δίνετε η δυνατότητα να διαχειρίζονται κάποιες από τις βασικές λειτουργίες του ξενοδοχείου τους μέσα από την ιστοσελίδα. Η καινοτομία της εφαρμογής είναι η δημιουργία ενός customized web service που επιτρέπει την επικοινωνία σε πραγματικό χρόνο μεταξύ του συστήματος και ενός RMS(reservation management system).Για την επικοινωνία αυτή δημιουργήθηκε ένα πρωτόκολλο που ανταλλάζει τις κρατήσεις και τα διαθέσιμα δωμάτια με το RMS.

Η εφαρμογή αποτελείτε από δύο βασικά στοιχεία: την εφαρμογή που βλέπει ο πελάτης η οποία δημιουργήθηκε με την χρήση τουAngularJS framework και το κομμάτι του εξυπηρετητή που φτιάχτηκε με το Spring boot Framework

Table of contents

List of figures

List of Tables

Acknowledgements

I would really like to thank my supervisor Dr Aivalis Konstantinos for his guidance and support through this difficult process. I also want to thank my colleague and friend Kyriakos Kalkanis for the collaboration.

Dedication

This project is dedicated to my sister Ioanna and to my beloved mother.

Chapter1-Introduction

Rapid technological development has completely changed the way people work in today's era compared to earlier times. New programming tools and techniques have been developed to help people work faster and more efficiently [17].

Since the early 1990s, the tourism sector has benefited from the use of such technologies [6]. The traditional method of booking rooms has been replaced [4] by fast remote online bookings. People no longer need to go to travel agents or make phone calls to make a hotel reservation. Instead, bookings are completed easily and quickly with a few clicks online [1].

This change in customer behavior has directly affected the hotel revenue, which has led to the need for them to be constantly connected to several channels (Hotel Distribution Channels) that manage, promote, or sell their available rooms to potential customers online. Some of the most important categories of such channels are online websites, online travel agents OTAS, and global distribution systems GDS. Research shows that most hotels are at least connected to three of these channels [2]. In fact, you consider the proper management of these channels very important because it is directly related to the hotels' revenue.

Today OTAS have managed to influence the tourism economy on a global [5] level, having a large share of the annual profits of both offline and especially online bookings. Also, by investing in mobile based websites and apps they have gained ground in the countries that use a lot of mobile and wearable devices. In this work [5] an increase in the use of OTAs by 12% annually is mentioned. Research [3] conducted in low budget hotels in China proved that OTAS directly affects a hotel's profit and how it was in third place just after offline methods for closing rooms.

What makes OTAS more than just an intermediary channel is the fact that they can also be used as a very efficient marketing tool. Hotels have the opportunity to advertise in social media and expand their range of customers e.g., approaching teenagers, backpack travelers, low budget travelers etc. [5]. In addition, customers through a visit to the internet can find what they need for their trip because OTAS offers a wide range of goods and services such as plane tickets, bus tickets, car rentals, etc. This is also the key to their proper use; the hotel should not see them as a simple channel but exploit the full range of possibilities and benefit as much as possible [5].

The emergence of the covid pandemic in 2019 created many challenges for the global economy. Companies around the world are trying to overcome obstacles and remain competitive. The online travel market was expected to grow by 18% in 2022 and gross bookings in 2023 to surpass pre-covid rates. Research in America showed that Expedia and Booking collectively accounted for about 93% of the leisure business and unmanaged travel OTA market in 2021[18]. Mobiles were the winners of the pandemic era, promoting safer and seamless travel, which is why OTAs are now paying more attention to their applications [18].

1.1-Motivation

OTAS are an integral part of the tourism and experience industry; they own a huge percentage of the annual world bookings leading to huge amounts of profit. That means that new tools and technologies develop every day to contribute to the tourist industry that itself was a motivation for conducting this work. Also Learning new programming tools and technologies contributed greatly to the selection of this topic. From my personal experience OTAS are my number one choice of choosing a hotel also the interaction I had with colleagues to complete the assignment gave me a great motivation to complete it.

1.3 Scope and Objectives

In this work, a website was designed and developed that has the role of an OTA, its job is to host hotels and sell their available rooms to customers online. In addition to the hotel registration process, the website offers a dashboard to manage the hotel's basic functions such as creating room availability, viewing reservations, editing the address, editing photos and its basic information. In addition to the functionality we give to hotel owners, the website shows the registered hotels to customers, giving them the option to choose a hotel that fits their criteria and book a room in it. PayPal's Sandbox project helped us to integrate online virtual payments in the system and thus gave a more realistic approach to the application. Also, the system has direct interaction with the Facebook Graph and automatically publishes ads (based on the availability of rooms and the current date) on a page that we manage through the application.

The frameworks used are Angular for the client part of the application and Spring boot combined with MySQL for the server part, more information about the technologies used in the application can be found in chapter 3.

1.4 Outline

Chapter 1: Introduction: In this chapter an introduction to the topic is given, we explain in more detail what exactly is required in this work, what was the motivation for conducting it and what was its scope. At the end of this chapter there is an outline that describes each chapter in detail.

Chapter 2: Background: This chapter explains in more detail the main terms we use in this report and presents the state of the online travel agents in the field.

Chapter 3: Requirements and Design: In this chapter you can see the steps we followed to develop the system. The chapter is divided into three major parts: System Analysis, System Design, and the technologies we used.

Chapter 4: Implementation: In the implementation chapter we presented our work through images and explained each step-in detail. To do that we followed the two basic Use Cases described in chapter 3.

Chapter 5: Customized web service support: A customized web service created for the communication of this system with a reservation management system is analyzed.

Chapter 6: Conclusions and future work: In this chapter a conclusion is made based on the experience of creating an OTA, also we propose some additional ideas that may improve the functionality of system.

Chapter 2-Background

In this chapter, an explanation of the terminology used for this work will be given and the state-of-the-art technologies in the field of OTAS will be presented.

2.1 Key Terms

2.1.1 Hotel Distribution Channels

By the term distribution channels we refer to all the ways in which a hotel can promote and consequently sell its rooms to potential buyers. This is an umbrella term that includes offline distribution channels e.g. (bookings in person, by phone or through a travel agency) and online distribution channels (e.g. hotel websites, mobile apps, online travel agents, global distribution systems, metasearch engines and but). Hotel Distribution Channels are an integral part of the booking process, and most hotels are constantly working with several of such channels to increase their profit [7],[8].

2.1.2 Online Travel Agents

This term refers to the online platforms that are usually managed by a travel company and act as an intermediary between the customer and the hotel for the purpose of selling goods and services. Through them, the customer can not only book a room, but also buy other things that will complete the travel experience, for example train tickets, bus tickets, etc. OTAS can be also used as an excellent marketing tool because they advertise the hotel on social media, provide buyers with links to visit hotel websites but also enable interaction with the hotel [9][10].

Some of the positives of using OTA include [11][12]:

- Their use as a marketing and management tool.
- Forecast and statistical analysis charts that OTAS produce.
- Interaction with clients via comment section.

Some of their negatives include:

- The commission they keep from each booking.
- Restrictive policies in terms of use they may have.
- Harmful consequences for hotel revenue if not used efficiently.

Briefly we mention some of these systems which are Booking.com, Expedia, Trip.com

2.1.3 Global Distribution Systems

GDB are networks which are mainly used by travel agents for the purpose of closing airline tickets, hotels, etc. [9][10].

GDBs keep a hotel's information up to date worldwide for potential travel agents and customers to view. Through them, agents see reviews and scores for hotels and book entire packages that meet the needs of their customers. Such systems are Travelport, Amadeus, and Sabre.

2.1.4 Hotel Revenue management

The term hotel revenue describes the profit a hotel has from its sales. But hotel revenue management are those methods that hotels use to have the maximum profit for the given moment by selling as much as possible the best quality products to their customers at the best price for them. This includes the use of statistics and performance data for hotels to predict the behavior of their customers. Since the early 1980s when revenue management techniques were being applied, they report an increase in hotel profits of up to 2-5%. For the success of revenue management, inventory allocation methods are often used which judge whether a reservation should be made at any given moment [2].

2.1.5 Channel Managers

Often the use of several channels at the same time can be difficult and demanding for a hotel, that is why software has been developed that takes over the management of the inventory of the channels on behalf of the hotel [9]. Chanel Manager allows the hotel to keep an up-to-date presence and synchronize operations through distribution channels.

2.2 OTA applications

Booking com

Top online travel agent in the world booking.com starting as a startup company in 1996 in Amsterdam, booking.com (a member of booking holdings) managed to grow into one of the largest e-commerce platforms in the world. By investing in technology that helps simplify travel, it provides millions of travelers with unforgettable experiences, a variety of transportation options, and incredible accommodations, from apartments to hotels and more. It has websites in 48 different languages and millions of registered accommodations for its customers [13].

Expedia.com

At the top of the list you can also find Expedia.com, a member of the Expedia group based in Seattle, America. Expedia.com offers millions of accommodations, airline tickets, car rentals, bus tickets, cruises and more through its website or mobile app. In fact, in 2014 the company started accepting bitcoins as a payment method [14].

Trip.com

Trip.com is also one of the famous online travel agents in the world. Trip.com is member of Trip.com Group based in China, rightfully considered one of the biggest travel agents as well with 400 million users all over the world.Trip.com is considered the parent of Skyscanner. Selling The site provides booking services for flights, hotels, trains, car rentals, airport transfers, tours, and attraction tickets, and claims to offer more than 1.2 million hotels in 200 countries and regions, as well as over 2 million flight routes connecting more than 5,000 cities. Train tickets for use in the UK, Germany, Japan, South Korea, and Mainland China are also available for purchase on the site.[15]. Its websites are translated in several languages including English, Greek, German, Italian, Malaysian and much more.

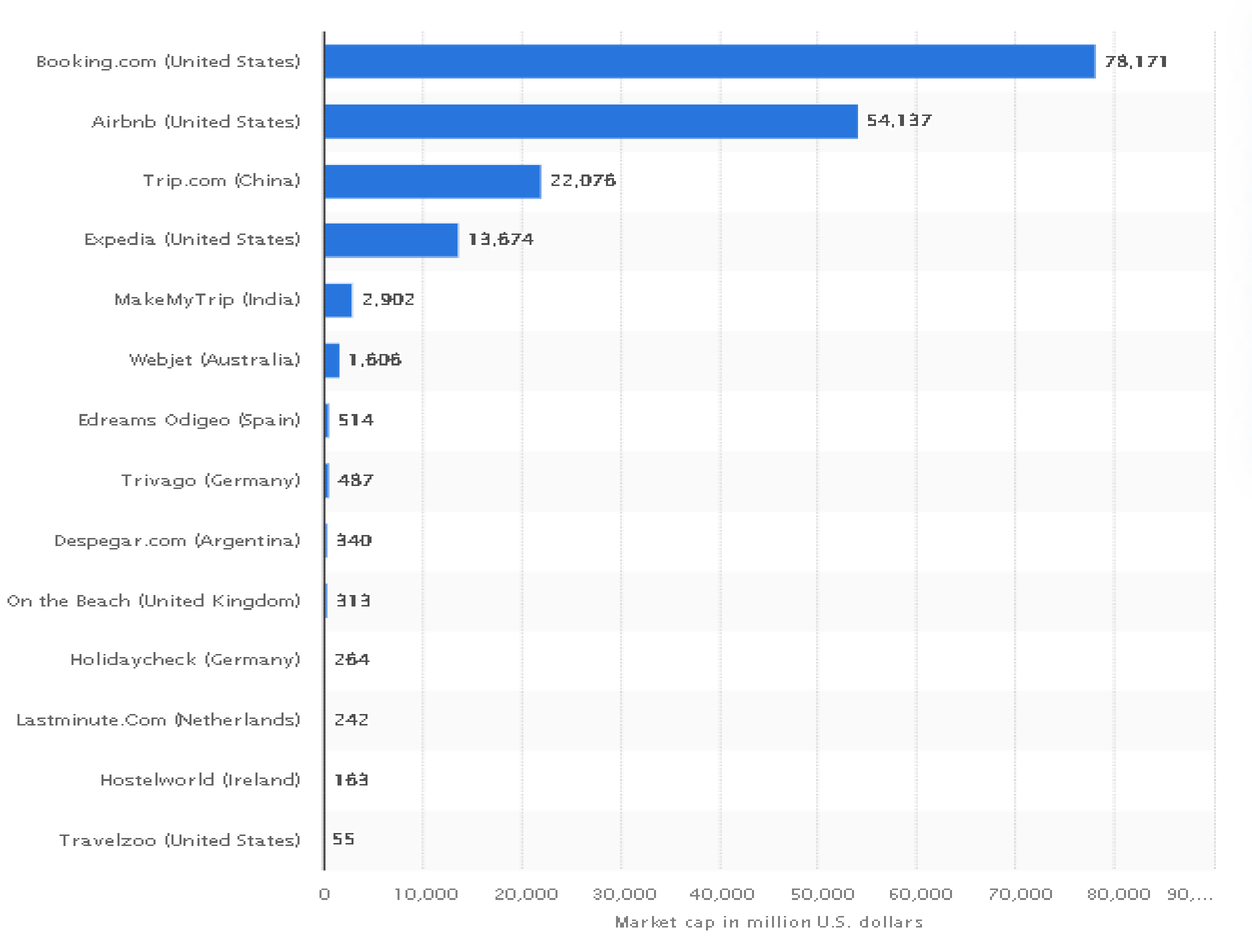


Figure 1 Top online travel agent applications based on annual profit.

The above figure is recently published by Statista Research Department showing the best online travel agents worth wide based on their market cup in million USA dollars. .

Chapter 3 – System Analysis and design

In this chapter our goal is to describe the steps we followed to determine the system requirements, to design it and finally to implement it. The chapter is divided into three main sections, the first is the analysis of the system or the second is its design and the third is the tools and technologies we used.

3.1- System Analysis

The system analysis contains the textual description of the system, a table with system requirements as they emerged from the description we made, as well as the Use Cases of the application.

3.1.1 – Textual Description

Our goal was to create a website that would allow all its users to see the available hotels and make reservations in any of them. Users will be given the option to search for a hotel based on some specific criteria e.g., date, city, number of people and the system will respond accordingly to their request by displaying the correct hotels. When users select a hotel, they will be able to see more information and its available rooms. After room selection, users proceed to the reservation stage where they will have to fill in the reservation form with their details and choose a payment method (e.g., payment at the hotel, pay by credit card, pay by PayPal account) and complete their reservation.

If they wish, they can add their hotel to the system through a registration form (this step includes entering information about the hotel and creating a user to associate with it).

Authorized users, i.e., those who have successfully completed the registration process, are given the opportunity to log in to the system and manage their hotel properties. Through a dashboard, users will be able to update hotel information, available rooms for customers, hotel services, hotel photos, reservations made by customers. The system will also provide the option to log out

The system will have a server that will communicate with the website and manage the database that will store the necessary system information. More specifically, in the database we want the available hotels to be stored, which implies the available users to know which user each hotel belongs to.

We also want the server to connect to the Facebook Graph and advertise the available hotels on a page that we will manage from it. We want these ads to have a photo of the accommodation, name and availability for each day.

3.1.2 – Requirements

Based on the above textual description we made for the system, the following table with the system requirements emerged.

Requirement	Title	Description
RQ1	Show available hotels	The website will show all available hotels to users.
RQ2	Search hotels by criteria	The system will allow users to search for a hotel based on certain criteria.
RQ3	Show hotel Availability	After choosing a hotel the user will be able to see the available rooms and choose the ones he wants.
RQ4	Reservation	The user will be able to make a reservation by filling in the reservation form with their details and choosing a payment method.
RQ5	Payment Options	The system at the reservation stage will support Payment at hotel ,payment by card and payment by PayPal

RQ6	Hotel Registration	The system will allow the user to register a hotel on the system.
RQ7	User Creation	During hotel registration system creates user account associated with hotel.
RQ8	User Login	Users who have completed their registration process system will give the option to log in.
RQ9	Hotel Dashboard	Logged in user can manage their hotel properties via dashboard
RQ10	Facebook advertisement	System will automatically advertise hotels to a facebook page

Table 1 Requirements Table

The following UML diagram shows the relationship between the system requirements as defined in the analysis phase.

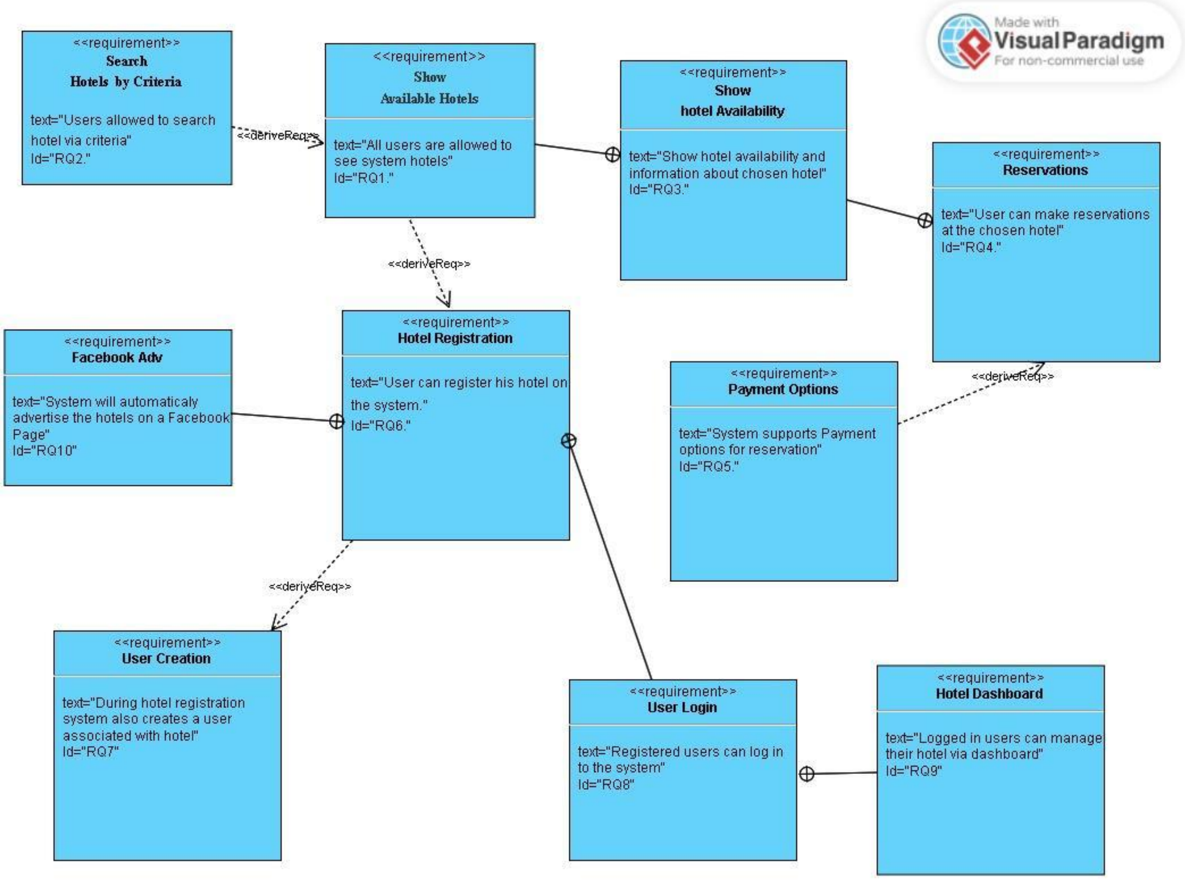

Made with Visual Paradigm For non-commercial use
<<requirement>>
Search Hotels by Criteria
text="Users allowed to search hotel via criteria"
Id="RQ2."
<<deriveReq>>
<<requirement>>
Show Available Hotels
text="All users are allowed to see system hotels"
Id="RQ1."
<<requirement>>
Show hotel Availability
text="Show hotel availability and information about chosen hotel"
Id="RQ3."
<<requirement>>
Reservations
text="User can make reservations at the chosen hotel"
Id="RQ4."
<<deriveReq>>
<<requirement>>
Hotel Registration
text="User can register his hotel on the system."
Id="RQ6."
<<requirement>>
Facebook Adv
text="System will automaticaly advertise the hotels on a Facebook Page"
Id="RQ10"
<<deriveReq>>
<<requirement>>
Payment Options
text="System supports Payment options for reservation"
Id="RQ5."
<<deriveReq>>
<<requirement>>
User Creation
text="During hotel registration system also creates a user associated with hotel"
Id="RQ7"
<<requirement>>
User Login
text="Registered users can log in to the system"
Id="RQ8"
<<requirement>>
Hotel Dashboard
text="Logged in users can manage their hotel via dashboard"
Id="RQ9"

Figure 2 UML Requirements Diagram

3.1.3 – Use Cases

We can say that the main Use Case scenarios of the system are two. In the first one an unauthorized user enters the system and in the second one an authorized user uses the system, i.e. a user who has completed the registration process and registered his hotel.

In the first case, the system allows the user to make hotel reservations and register in the system if he wishes. In the second case, the user can also complete reservations in the system and manage his hotel through the dashboard.

In the following diagram we can see the main use cases of the system as well as the main roles that the system will support.

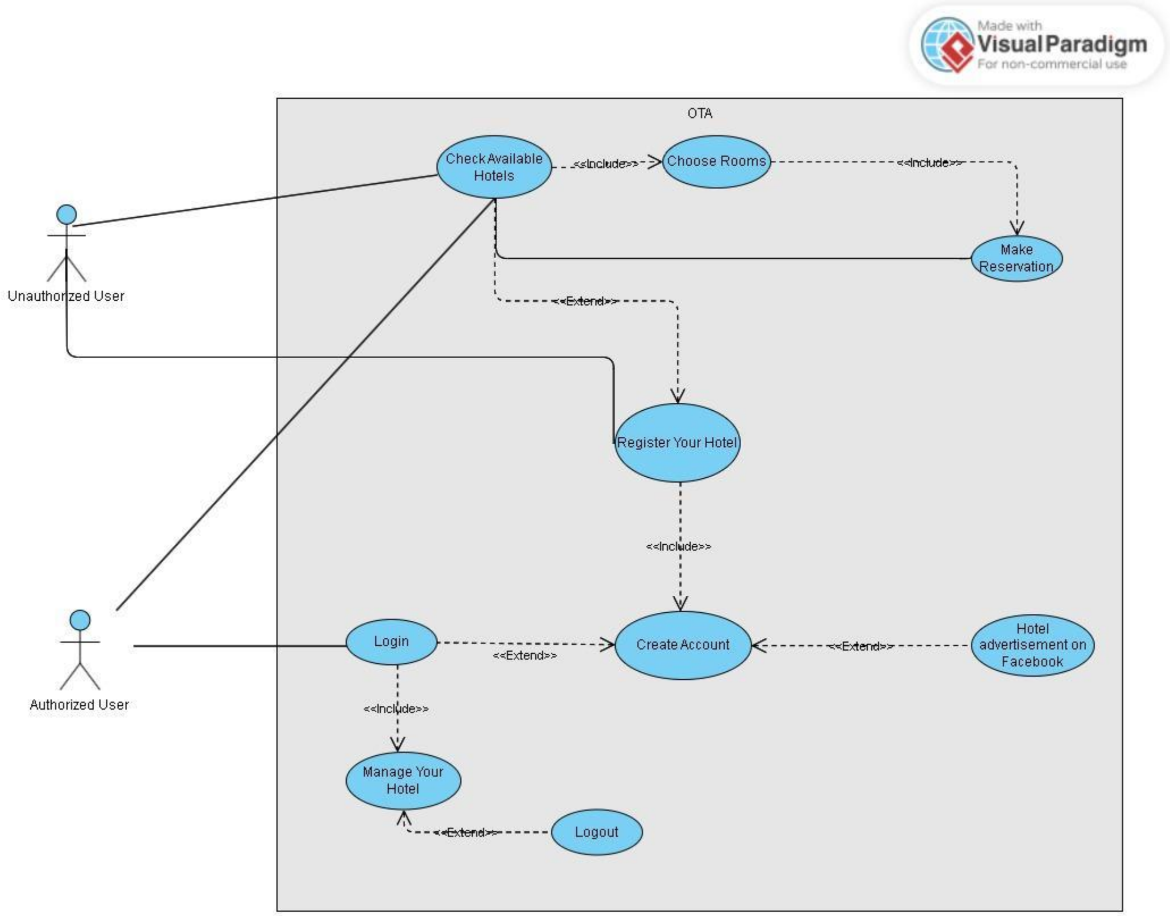


Figure 3 UML Use Case Diagram

3.2- System Design

In this section the architecture of the system is presented, UML diagrams that describe its basic functions are included as well as the ER diagram of the database.

3.2.1 – Architecture

For the development of our software, we adopt the Client-Server model and divide our software into two separate applications that interact with each other to show the website to the user. Using this model gives us the flexibility to manage the two pieces of our system separately and use tools appropriate for each piece.

The following diagram shows how our system is based on the client server model as well as the interaction with external components.

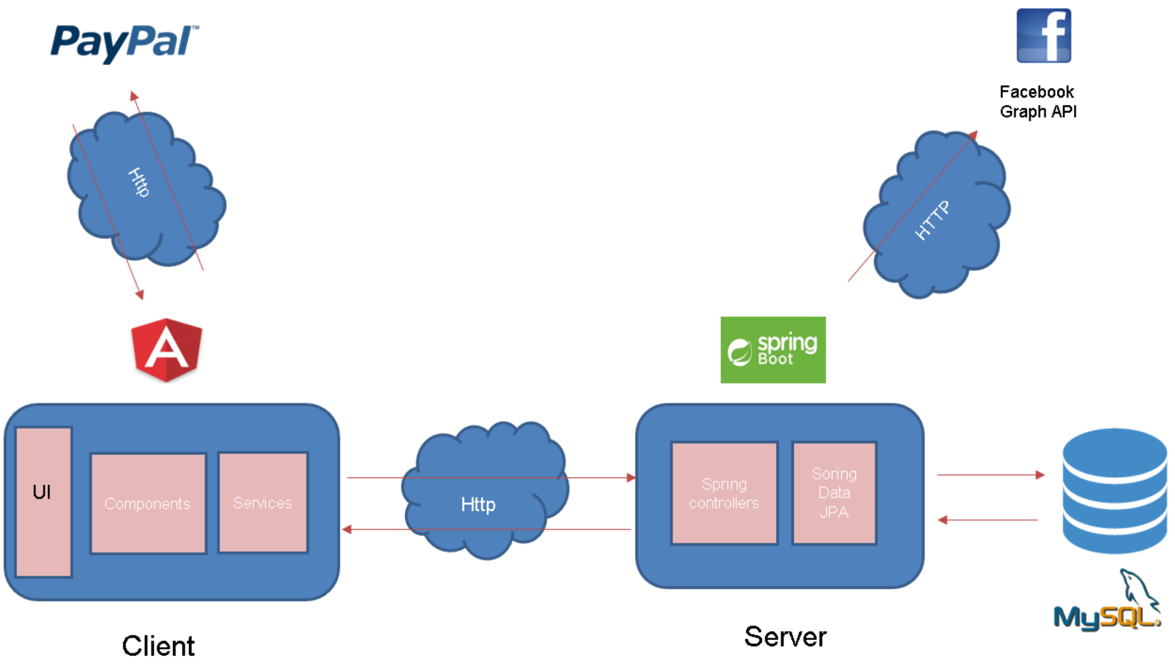


Figure 4 System Architecture

3.2.2 – Components Diagram

The diagram below illustrates the basic components of the application in a general form.

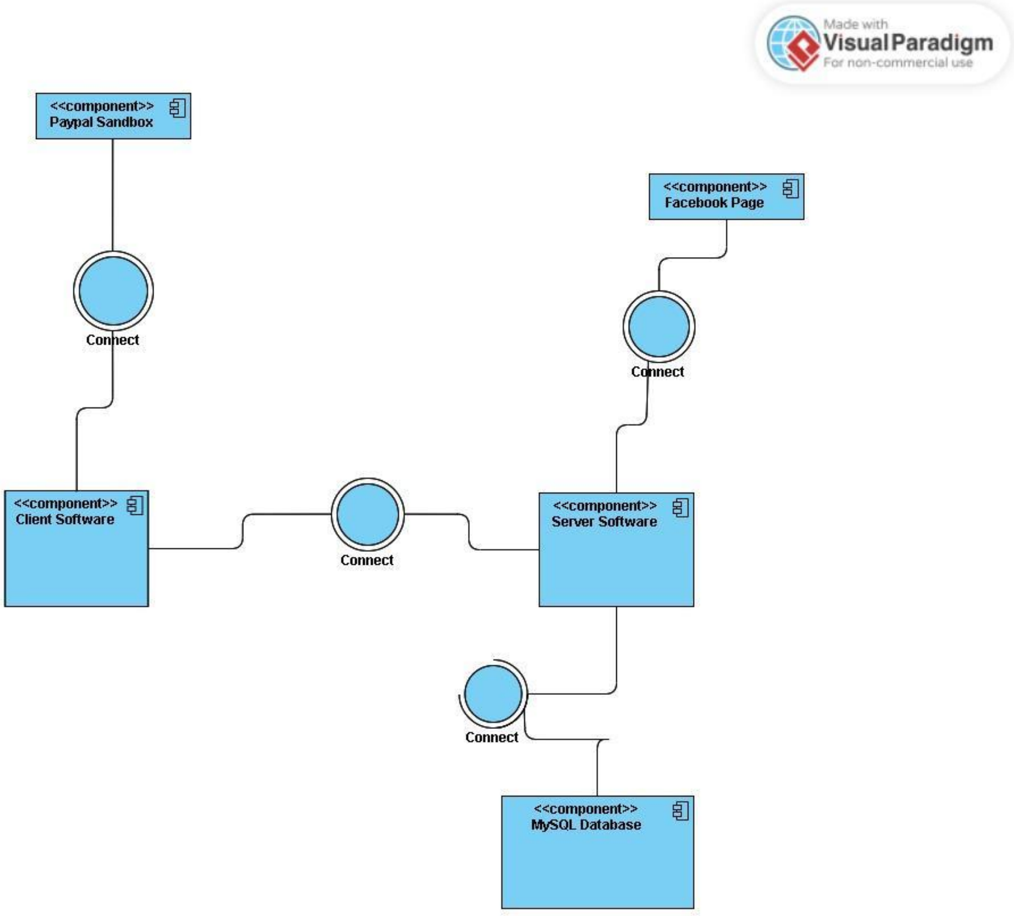

Made with Visual Paradigm For non-commercial use
<<component>> Paypal Sandbox
Connect
<<component>> Facebook Page
Connect
<<component>> Client Software
Connect
<<component>> Server Software
Connect
<<component>> MySQL Database

Figure 5 UML Component Diagram

3.2.3 – Sequence Diagram

In the sequence diagram we can see the sequential continuity in the processes performed between the main parts of the application. This diagram shows the three most basic functions of the system in my opinion.

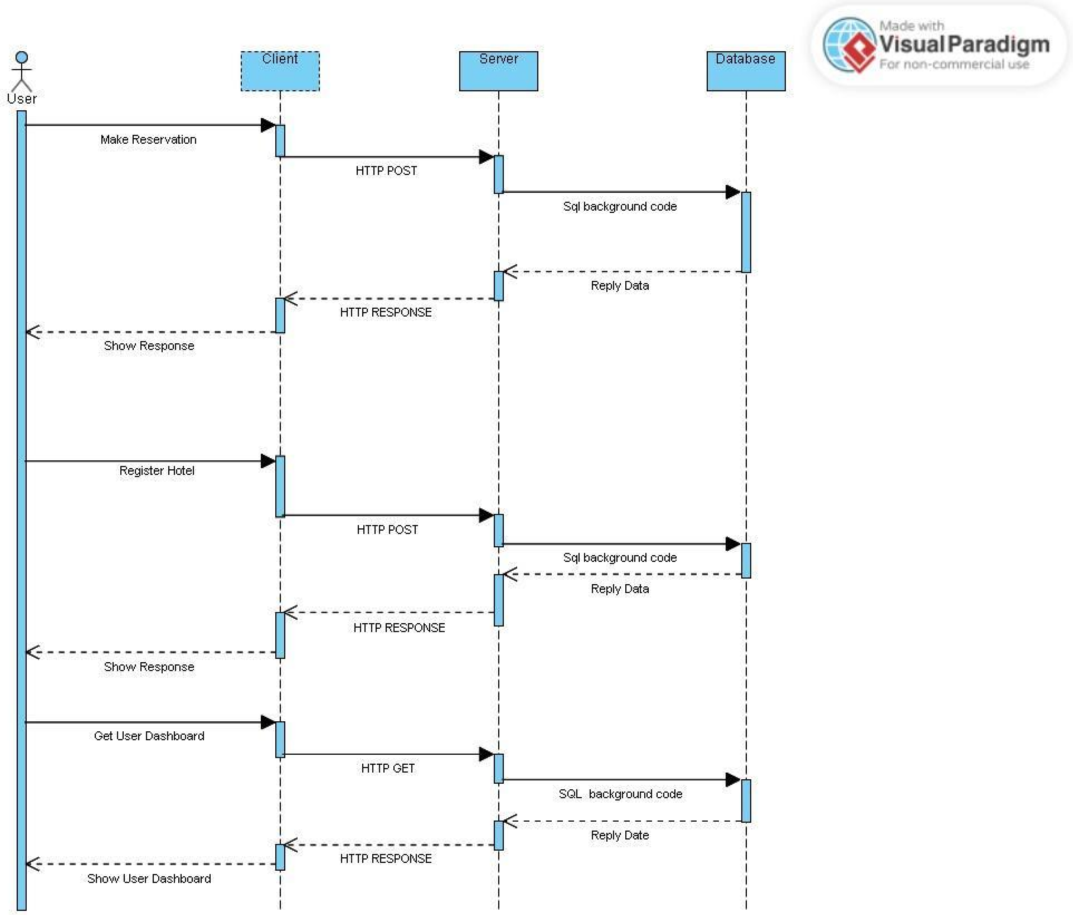


Figure 6 UML Sequence Diagram

3.2.4 – Database Schema

The figure below illustrates the tables of the databases as well as the relationships between them. It is obvious that the table that has the leading role is the one that holds the hotel information and most of the connections will be created on it.

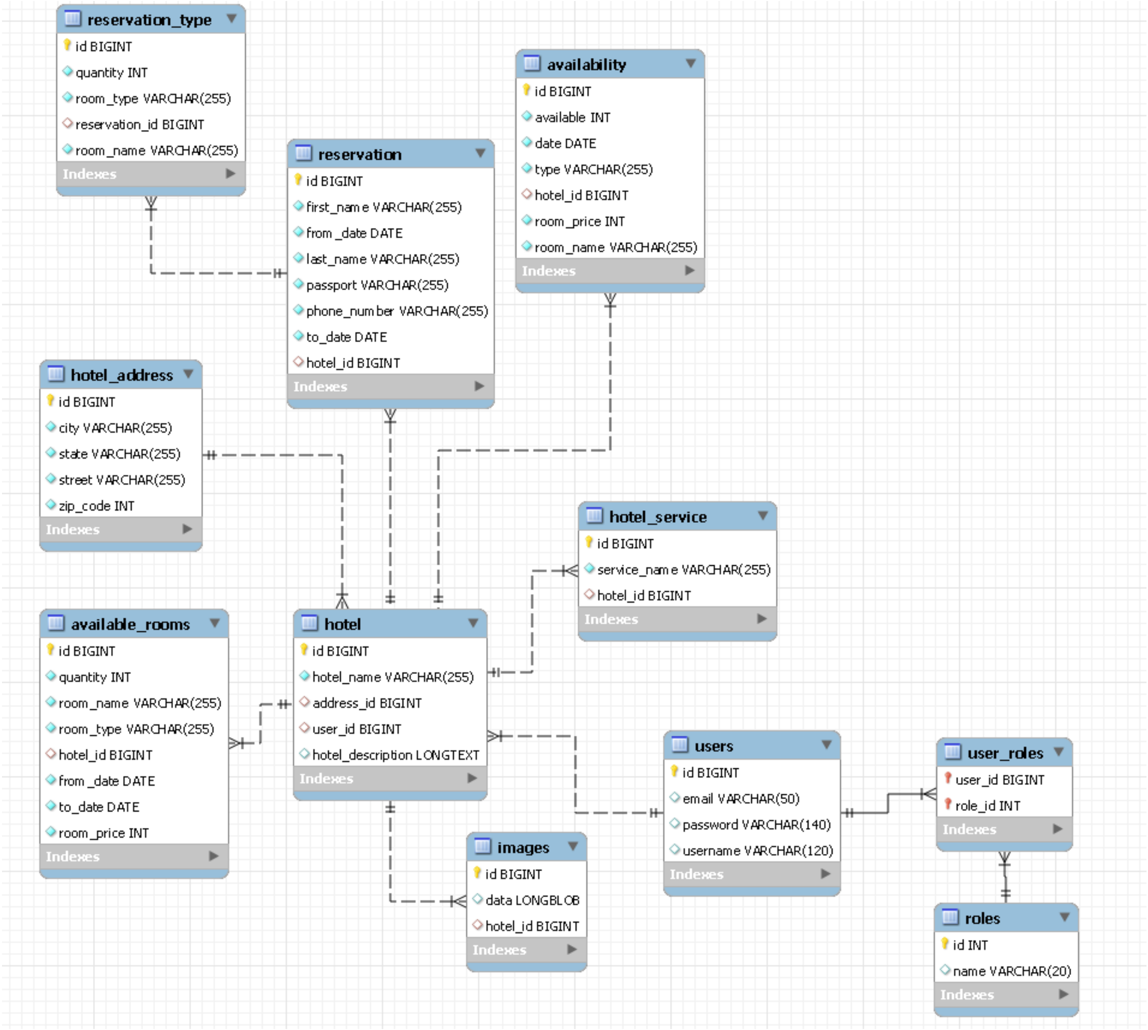


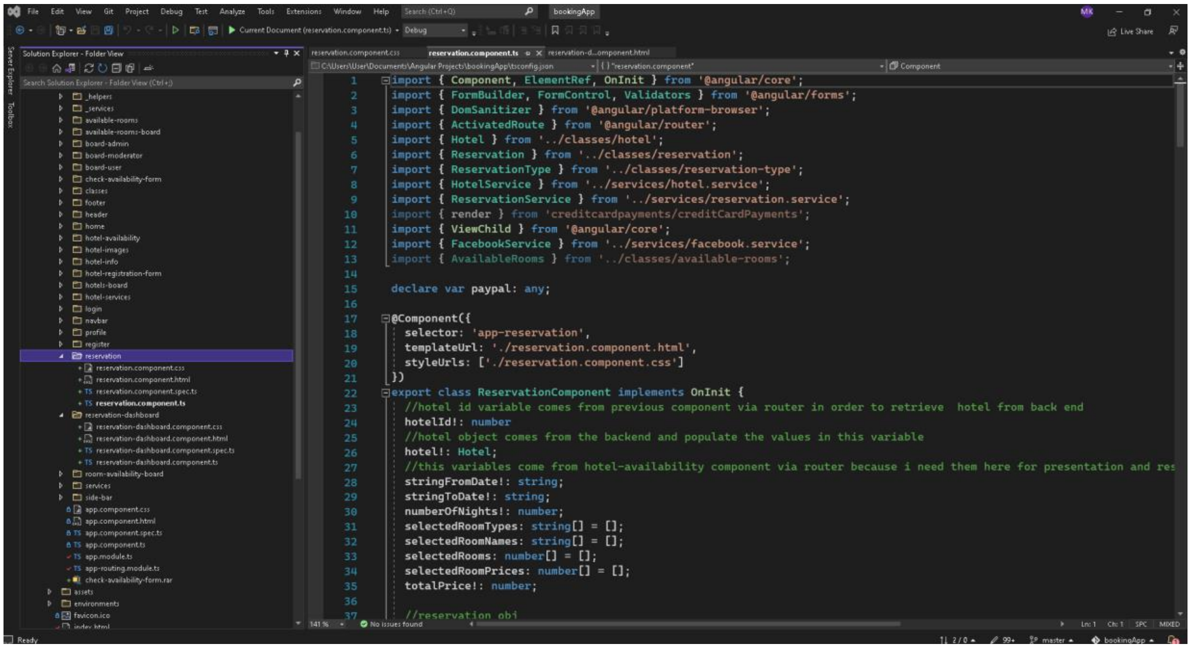
Figure 7 Database ER Model

3.3- Tools and Technologies

In this subsection we present some of the most important tools and technologies which we used in our application.

3.3.1 – Angular

To implement the application from the client side (client-side component) we use Angular. Angular is a platform for designing and developing single page applications based on the typescript programming language [19]. Angular contains a framework that helps us build web applications and a wide range of libraries that implement routing functions, form development and other. It also contains a suite of developer tools for developing, building, extending our code. Through its component-based architecture, angular helps us to have clean and organized code as it divides the code into components where each one is a package of html, CSS, typescript code.



```
import { Component, ElementRef, OnInit } from '@angular/core';
import { FormBuilder, FormControl, Validators } from '@angular/forms';
import { DomSanitizer } from '@angular/platform-browser';
import { ActivatedRoute } from '@angular/router';
import { Hotel } from '../classes/hotel';
import { Reservation } from '../classes/reservation';
import { ReservationType } from '../classes/reservation-type';
import { HotelService } from '../services/hotel.service';
import { ReservationService } from '../services/reservation.service';
import { render } from 'creditcardpayments/creditCardPayments';
import { ViewChild } from '@angular/core';
import { FacebookService } from '../services/facebook.service';
import { AvailableRooms } from '../classes/available-rooms';

declare var paypal: any;

@Component({
  selector: 'app-reservation',
  templateUrl: './reservation.component.html',
  styleUrls: ['./reservation.component.css']
})
export class ReservationComponent implements OnInit {
  //hotel id variable comes from previous component via router in order to retrieve  hotel from back end
  hotelId!: number
  //hotel object comes from the backend and populate the values in this variable
  hotel!: Hotel;
  //this variables come from hotel-availability component via router because i need them here for presentation and res
  stringFromDate!: string;
  stringToDate!: string;
  numberOfNights!: number;
  selectedRoomTypes: string[] = [];
  selectedRoomNames: string[] = [];
  selectedRooms: number[] = [];
  selectedRoomPrices: number[] = [];
  totalPrice!: number;

  //reservation obj
```

File Edit View Git Project Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
bookingApp
Current Document (reservation.component.ts)
Debug
Live Share
Solution Explorer - Folder View
reservation.component.css
reservation.component.ts
reservation-d...omponent.html
C:\Users\User\Documents\Angular Projects\bookingApp\tsconfig.json
"reservation.component"
Component
141 %
No issues found
Ln: 1
Ch: 1
SPC
MIXED
Ready
master
bookingApp

Figure 8 Angular Project

3.3.2 – Spring Framework

Spring is a free open-source framework for the development of a wide range of java-based applications. From standalone projects to cloud based applications spring gives the flexibility to create applications that lasts in time and supports changes in the architecture in any stage of the development. Spring refers to the basic project created in 2003 but usually people use the term spring framework to describe all the additional projects created after spring Framework. These projects is referring as spring modules and developers have the option to choose any of these modules that fits to their application needs, including spring core project itself. These projects comes on the top of the spring core but they have their own repository. These projects support the development of web applications for example spring web, others provide interconnection with database like spring Data JPA and much more. Some of the modules are spring security, spring cloud, etc.

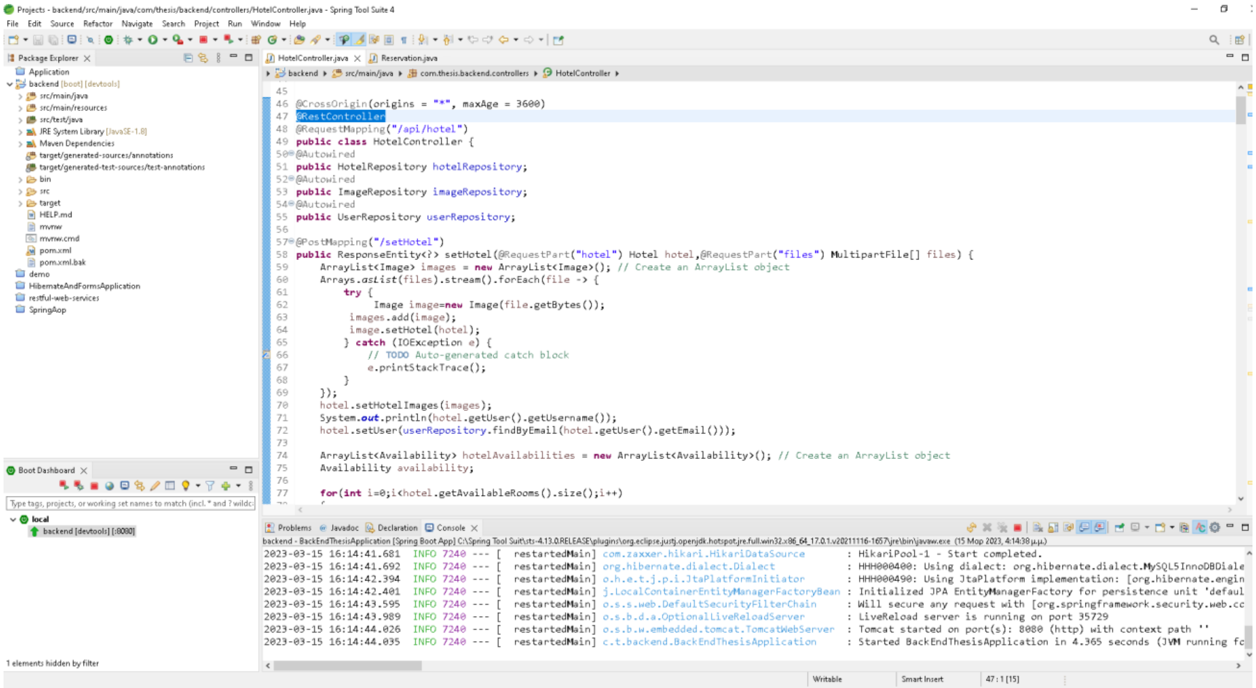
Historically speaking core project crated in 2003 responding to the difficulties of the early J2EE specifications. Despite the believe that wants spring and J2EE to be competing, spring created mote as a complementary framework to J2EE. Spring doesn't embrace the Jakarta specification platform but rather uses carefully selected parts of the EE umbrella. Spring has a very strong support community with its versions involving in time including many other projects and features such as hibernate ORM, compatibility with tomcat and many more spring architecture follows some of the above principals.

- Support changes in the architecture in any stage of the development process.
- Doesn’t have very strong opinions on how things should be done that gives the option to create many types of applications.
- Spring has strong backwards compatibility supporting the applications for many years.
- Cares about the API design.
- Set standards for good code quality.

Spring Framework is used by more than 50% of java developers as proven by a 2018 survey and that is why we chose it for the development of our application.

3.3.3 – Spring boot

For the application running on our server, we used Spring boot. Spring boot was created to eliminate the boilerplate code that existed with the use of spring [21]. Its primary objective is to develop fast ready to run applications on the go. Spring boot automatically implemented the dependencies between classes and made our code much more readable and efficient. It also helped us to easily create rest API controllers to share information with the client side. Spring boot also simplified the communication we had with the database using spring data JPA which implements hibernate object relational mapping between classes and the tables of the database. This means that the SQL code was automatically executed in the background by Spring, and we simply used the spring data JPA repository to manage the database.


Projects - backend/src/main/java/com/thesis/backend/controllers/HotelController.java - Spring Tool Suite 4
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Application
backend [boot] [devtools]
src/main/java
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
bin
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
pom.xml.bak
demo
HibernateAndFormsApplication
restful-web-services
SpringAop
Boot Dashboard
Type tags, projects, or working set names to match (incl. * and ? wildc
local
backend [devtools] [:8080]
1 elements hidden by filter
HotelController.java
Reservation.java
backend
src/main/java
com.thesis.backend.controllers
HotelController
Problems
Javadoc
Declaration
Console
Writable
Smart Insert
47 : 1 [15]

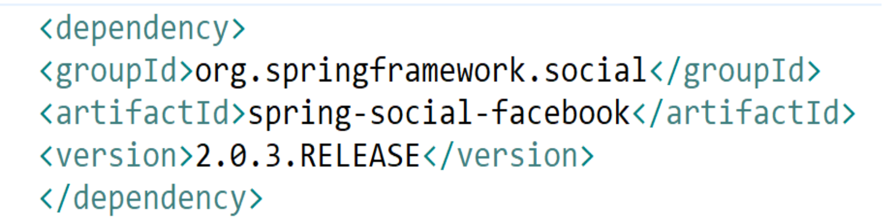
```
@CrossOrigin(origins = "*", maxAge = 3600)
@RestController
@RequestMapping("/api/hotel")
public class HotelController {
@Autowired
public HotelRepository hotelRepository;
@Autowired
public ImageRepository imageRepository;
@Autowired
public UserRepository userRepository;

@PostMapping("/setHotel")
public ResponseEntity<?> setHotel(@RequestPart("hotel") Hotel hotel,@RequestPart("files") MultipartFile[] files) {
    ArrayList<Image> images = new ArrayList<Image>(); // Create an ArrayList object
    Arrays.asList(files).stream().forEach(file -> {
        try {
             Image image=new Image(file.getBytes());
         images.add(image);
         image.setHotel(hotel);
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    });
    hotel.setHotelImages(images);
    System.out.println(hotel.getUser().getUsername());
    hotel.setUser(userRepository.findByEmail(hotel.getUser().getEmail()));

    ArrayList<Availability> hotelAvailabilities = new ArrayList<Availability>(); // Create an ArrayList object
    Availability availability;

    for(int i=0;i<hotel.getAvailableRooms().size();i++)
```

```
backend - BackEndThesisApplication [Spring Boot App] C:\Spring Tool Suit\sts-4.13.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe  (15 Μαρ 2023, 4:14:38 μ.μ.)
2023-03-15 16:14:41.681  INFO 7240 --- [  restartedMain] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2023-03-15 16:14:41.692  INFO 7240 --- [  restartedMain] org.hibernate.dialect.Dialect            : HHH000400: Using dialect: org.hibernate.dialect.MySQL5InnoDBDiale
2023-03-15 16:14:42.394  INFO 7240 --- [  restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000490: Using JtaPlatform implementation: [org.hibernate.engin
2023-03-15 16:14:42.401  INFO 7240 --- [  restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'defaul
2023-03-15 16:14:43.595  INFO 7240 --- [  restartedMain] o.s.s.web.DefaultSecurityFilterChain     : Will secure any request with [org.springframework.security.web.cc
2023-03-15 16:14:43.989  INFO 7240 --- [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on port 35729
2023-03-15 16:14:44.026  INFO 7240 --- [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
2023-03-15 16:14:44.035  INFO 7240 --- [  restartedMain] c.t.backend.BackEndThesisApplication     : Started BackEndThesisApplication in 4.365 seconds (JVM running fc
```

Figure 9 Spring book project

3.3.4 – Spring Social Facebook and Facebook Graph API

The main tool we used for our connection with Facebook was Spring social Facebook. Spring social Facebook is an implementation of spring social that is customized for connection with Facebook. It has a big collection of methods to easily connect with Facebook. The first step was to inject the dependencies for spring social Facebook in our spring boot application and establish a connection with a Facebook page.



```
<dependency>
<groupId>org.springframework.social</groupId>
<artifactId>spring-social-facebook</artifactId>
<version>2.0.3.RELEASE</version>
</dependency>
```

Figure 10 Spring Social pom file dependency

As most tools that interact with Facebook spring social also use Facebook graph API at a very low level. The Facebook Graph API is the main way for applications to interact with Facebook Platform [22], with the help of this API developers can get user or page information, post on pages feed, get data associated with users etc. Most of the tools that Facebook gives to developers use this API at some point. The data transfer conforms to the HTTPS transfer protocol so API calls can be made with the use of a language that supports HTTP communication such as curl or via browser. A typical API call made with curl is:

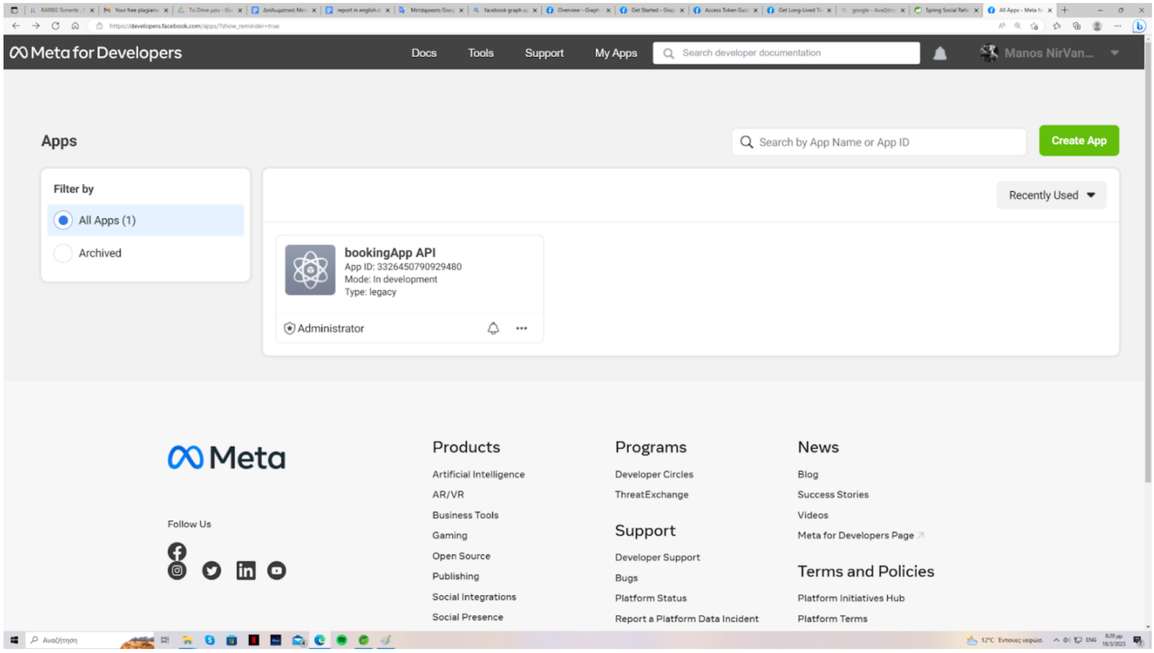
```
curl -i -X GET
"https://graph.facebook.com/facebook/picture?redirect=false"
```

Graph API is named after the idea of a social graph. All information in Facebook is represented in the form of nodes, fields, and edges. A Facebook node usually represents a user or Facebook page or a post. Nodes have a unique id and a predefined set of fields. For example, a user node has fields such as id, name, date of birth etc. Edges represent the connections between nodes. Developers can get, update, delete the information of this graph but they need an access token to escort their requests.

Access tokens are unique string values that identify the user, page, application that has access to the graph. There are four types of access tokens based on the nature of their owner: user access tokens usually given through a log in dialog, page access tokens require the creation of a Facebook page, client access tokens and app access tokens.

There are two major categories: short term tokens and long-term tokens. Usually, the access tokens that Facebook gives to clients are short-lived tokens and expire between a few hours but there is the option of transforming a short-lived access token to long-live. long live tokens usually are used for server-to-server communication.

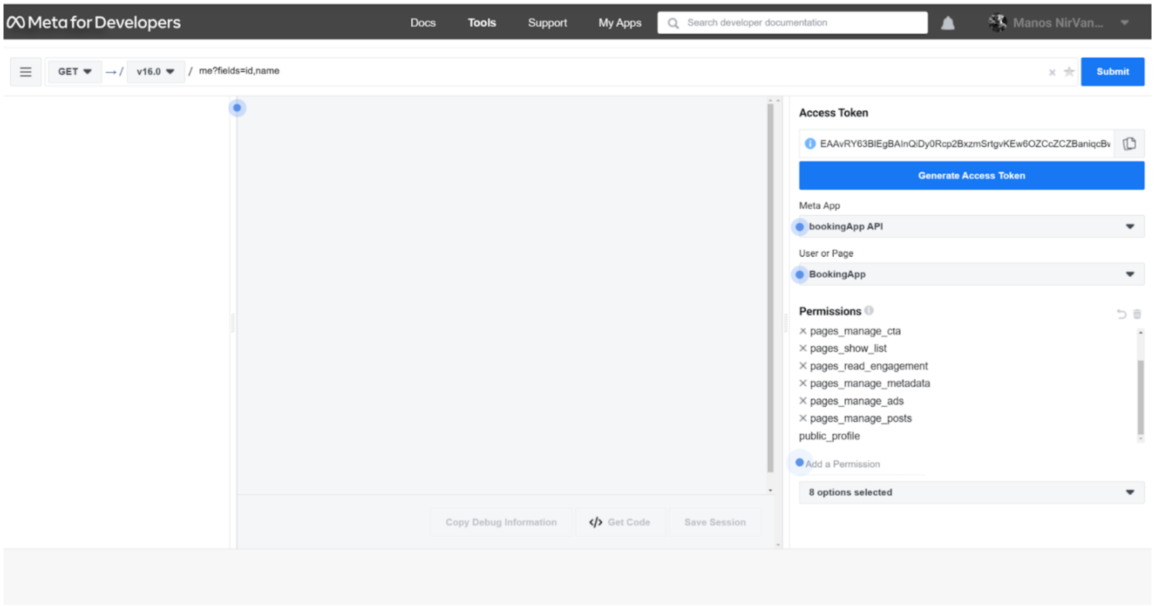
For this project, we created a Facebook page linked to a Facebook account. Then we registered on Facebook for developers and created an application linked with this page. Below figure shows this application we created in Facebook for developer's site.



bookingApp API
App ID: 3326450790929480
Mode: In development
Type: legacy

Figure 11 Facebook Application

Through graph API explorer we created a short-lived access token which we transform to a long-term token for our server to communicate to Facebook server. As you can see in the image below, we through graph API explorer we created the key with all the requirement properties for posting to a page.



GET → / v16.0 / me?fields=id,name

Access Token

EAAvRY63BlEgBAlnQiDy0Rcp2BxzmSrtgvKEw6OZCcZCZBaniqcBv

Meta App: bookingApp API

User or Page: BookingApp

Permissions

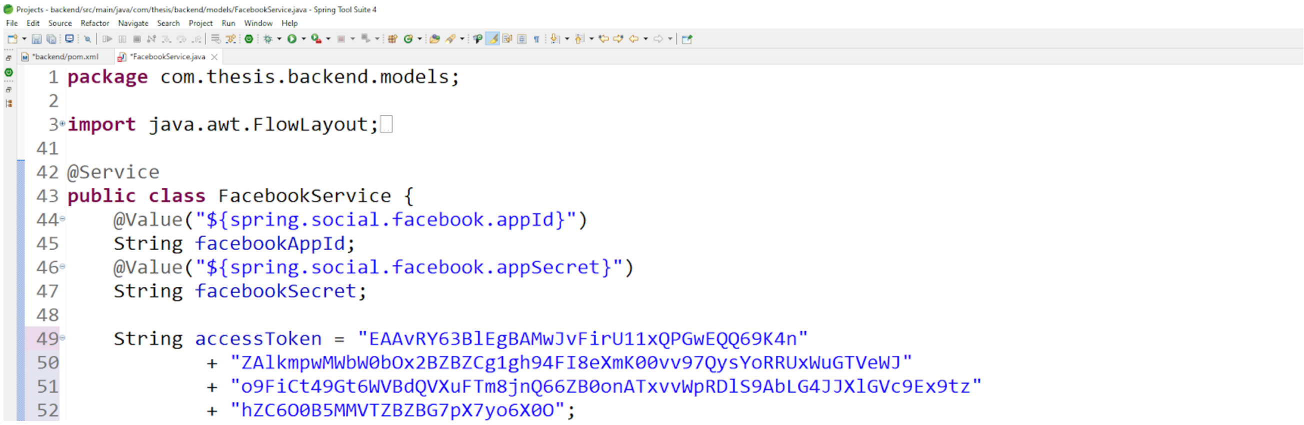
× pages_manage_cta
× pages_show_list
× pages_read_engagement
× pages_manage_metadata
× pages_manage_ads
× pages_manage_posts
public_profile

Add a Permission

8 options selected

Figure 12 Facebook graph explorer

We used this token in our server's application as a predefined variable and we pass this token as a parameter to every Facebook Graph API call, we make via Spring social. The following figures contain parts of our code with spring social,

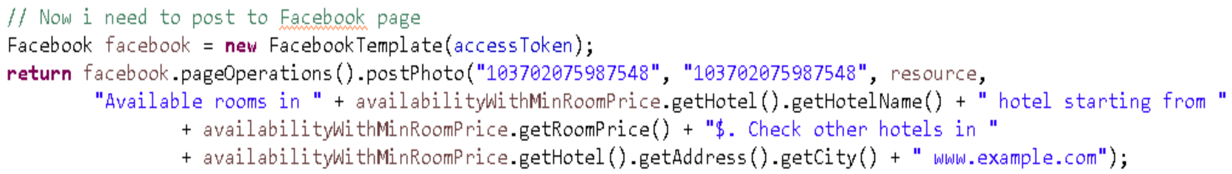

Projects - backend/src/main/java/com/thesis/backend/models/FacebookService.java - Spring Tool Suite 4
File
Edit
Source
Refactor
Navigate
Search
Project
Run
Window
Help
*backend/pom.xml
*FacebookService.java

```java
package com.thesis.backend.models;

import java.awt.FlowLayout;

@Service
public class FacebookService {
    @Value("${spring.social.facebook.appId}")
    String facebookAppId;
    @Value("${spring.social.facebook.appSecret}")
    String facebookSecret;

    String accessToken = "EAAvRY63BlEgBAMwJvFirU11xQPGwEQQ69K4n"
            + "ZAlkmpwMWbW0bOx2BZBZCg1gh94FI8eXmK00vv97QysYoRRUxWuGTVeWJ"
            + "o9FiCt49Gt6WVBdQVXuFTm8jnQ66ZB0onATxvvWpRDlS9AbLG4JJXlGVc9Ex9tz"
            + "hZC6O0B5MMVTZBZBG7pX7yo6X0O";
```

Figure 13 Facebook service

Figure 14 Spring Social code for connection with Facebook

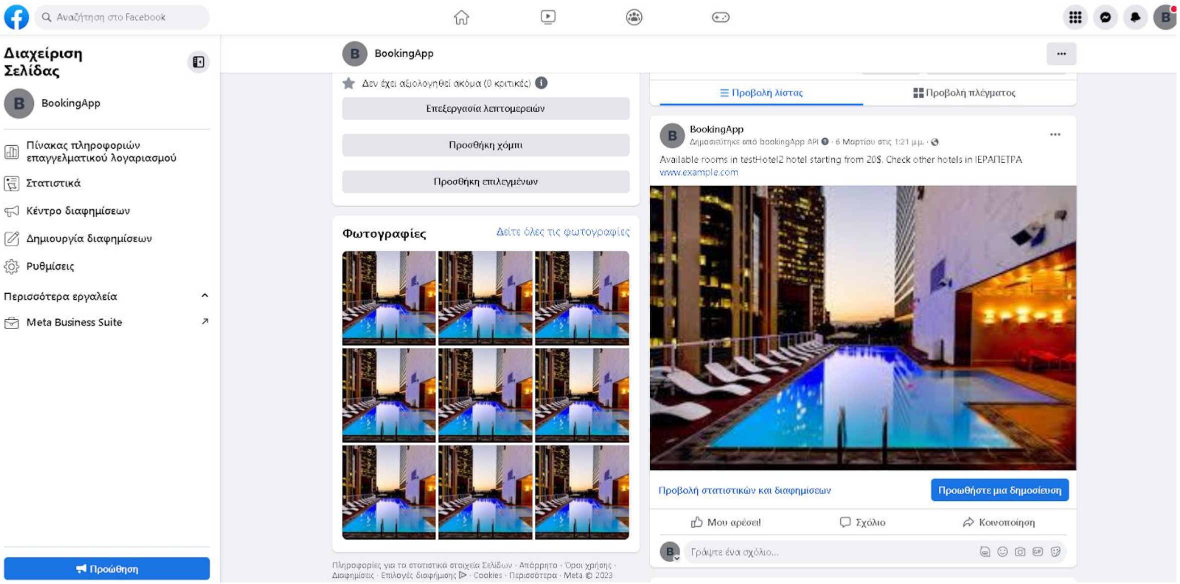
in the above figure you can see that spring social requires the application id and applications secrete id to establish the connection. Also, there is the long-term access token as a variable. Below figure contains the code with sending posts on Facebook page.



```
// Now i need to post to Facebook page
Facebook facebook = new FacebookTemplate(accessToken);
return facebook.pageOperations().postPhoto("103702075987548", "103702075987548", resource,
        "Available rooms in " + availabilityWithMinRoomPrice.getHotel().getHotelName() + " hotel starting from "
                + availabilityWithMinRoomPrice.getRoomPrice() + "$. Check other hotels in "
                + availabilityWithMinRoomPrice.getHotel().getAddress().getCity() + " www.example.com");
```

Figure 15 Spring social code upload on Facebook page

Eventually our server posts information about hotel rooms on the Facebook page daily.



BookingApp
Δημοσιεύτηκε από bookingApp API · 6 Μαρτίου στις 1:21 μ.μ.
Available rooms in testHotel2 hotel starting from 20$. Check other hotels in ΙΕΡΑΠΕΤΡΑ
www.example.com

Figure 16 Advertise hotel on Facebook page.

3.3.5 – PayPal Sandbox

PayPal has created this developer tool which allows its users to simulate online payments on their website [23]. The sandbox allows us to test our page by copying most of the properties of real transactions but without interacting with any of the real PayPal accounts. PayPal sandbox consists of two main parts:

Component	Access Through
Sandbox test site	https://www.sandbox.paypal.com/
Sandbox>Accounts Page	https://developer.paypal.com/developer/accounts

Table 2 Sandbox major parts

the testing site and the user accounts site. From the testing site we can manage the transactions that will be made on our page. From the accounts site we create and manage the sandbox accounts which we will use to make our transactions.

Sandbox accounts can be of two kinds: business accounts and personal accounts. The business accounts act as the merchant and personal accounts act as buyers in a transaction. As soon as the user creates an account in PayPal for developers, the sandbox automatically creates an account from each of them. For example, pp.merch01-facilitator@example.com A default personal account. For example, pp.merch01-buyer@example.com.

Some applications have the need for more accounts and that is why through the accounts website the user can create other accounts which are either personal or business. The user has the option to change details of the accounts, change password, to link any of the accounts to his papal for developers account, to clone or delete the accounts he wishes. Users are not allowed to delete the two original accounts created by the sandbox during registration.

The sandbox also enables the user to mass upload up to forty accounts to their system, thus covering a wide range of possible scenarios of customers and transactions that the site may have. This property of sandbox is referred as bulking upload. To do this the user must upload a file that will describe these accounts and link to their sandbox account. This file is in the form of an array with the heading having in each column types of values that describe the sandbox account, e.g., Country_code, Account_type, Password etc. and in the lines of the array are the virtual accounts he wants to create. Rows number can vary from one to forty and their values must correspond to the corresponding column heading type.

To check the quality of transactions executed by credit card, the sandbox gives the possibility of simulating successful transactions and transactions that have an error. To simulate successful card transactions, the user must make sure he is in sandbox mode and then make a transaction. The sandbox allows users to give any future expiration date and for the CVV card code they are allowed to give any three-digit number except for American Express cards which want a four digit combination. For the card number, the sandbox offers a ready-made table of numbers that can be used for transactions.

The sandbox enables us to trigger incorrect transactions by card by giving the correct details of the transaction except the name of the card owner to which we must pass one of the predefined rejection triggers sandbox gives us. The table below shows some of the rejection triggers.

Test Name	Trigger	Processor Response code
Fraudulent card	CCREJECT-SF	9500
Card is declined	CCREJECT-BANK_ERROR	5100

Table 3 Sandbox predefined rejection triggers

When the user creates the accounts he wants for testing purposes and establishes the communication with PayPal payments form to his website, he can start making transactions. Each account involved in these transactions will begin receiving notifications that the user can view from the sandbox accounts site. When the user tests his application, he uses the sandbox mode and when he is ready, he returns to the normal PayPal mode.

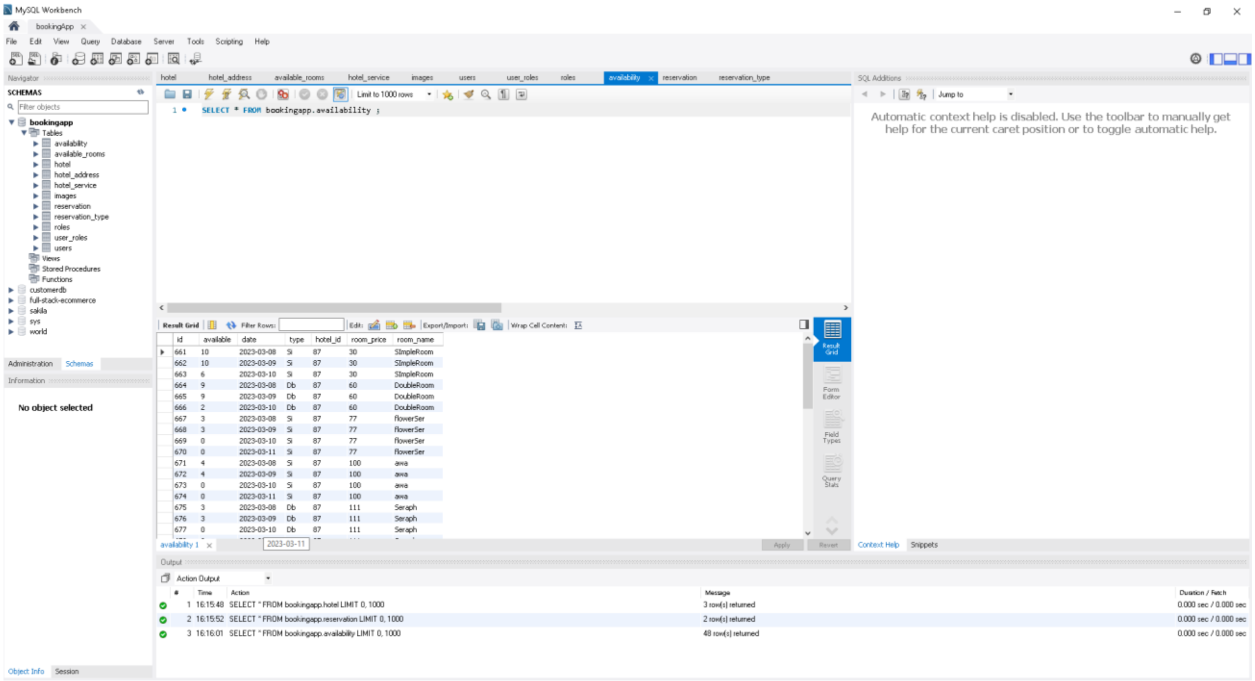
Sandbox is a positive environment for testing, which means that by its nature it is an environment that promotes the improvement of our application through the wrong scenarios.

In addition to errors that can be detected in card transactions, we can also trigger errors that have to do with our transaction in general. Through parameters that the user can pass to the API during the transaction, he is enabled to trigger errors that have to do with the amount part of the transaction and errors for the non-amount part of the transaction.

In our app we used the PayPal payment form in the reservation process where we gave the user the option to pay at the property or use the form to pay with a PayPal sandbox account or card.

3.3.6 – MySQL workbench

MySQL workbench is a tool used to create and manage relational databases with the SQL programming language [24]. It provides a large set of data management, design, and visualization tools. In our application, we use workbench extensively. With its help we created our database and managed our tables and created useful ER diagrams to visualize the relationships between tables.



```
SELECT * FROM bookingapp.availability ;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

id	available	date	type	hotel_id	room_price	room_name
661	10	2023-03-08	Si	87	30	SImpleRoom
662	10	2023-03-09	Si	87	30	SImpleRoom
663	6	2023-03-10	Si	87	30	SImpleRoom
664	9	2023-03-08	Db	87	60	DoubleRoom
665	9	2023-03-09	Db	87	60	DoubleRoom
666	2	2023-03-10	Db	87	60	DoubleRoom
667	3	2023-03-08	Si	87	77	flowerSer
668	3	2023-03-09	Si	87	77	flowerSer
669	0	2023-03-10	Si	87	77	flowerSer
670	0	2023-03-11	Si	87	77	flowerSer
671	4	2023-03-08	Si	87	100	awa
672	4	2023-03-09	Si	87	100	awa
673	0	2023-03-10	Si	87	100	awa
674	0	2023-03-11	Si	87	100	awa
675	3	2023-03-08	Db	87	111	Seraph
676	3	2023-03-09	Db	87	111	Seraph
677	0	2023-03-10	Db	87	111	Seraph

2023-03-11

```
#  Time      Action                                              Message             Duration / Fetch
1  16:15:48  SELECT * FROM bookingapp.hotel LIMIT 0, 1000        3 row(s) returned   0.000 sec / 0.000 sec
2  16:15:52  SELECT * FROM bookingapp.reservation LIMIT 0, 1000  2 row(s) returned   0.000 sec / 0.000 sec
3  16:16:01  SELECT * FROM bookingapp.availability LIMIT 0, 1000 48 row(s) returned  0.000 sec / 0.000 sec
```

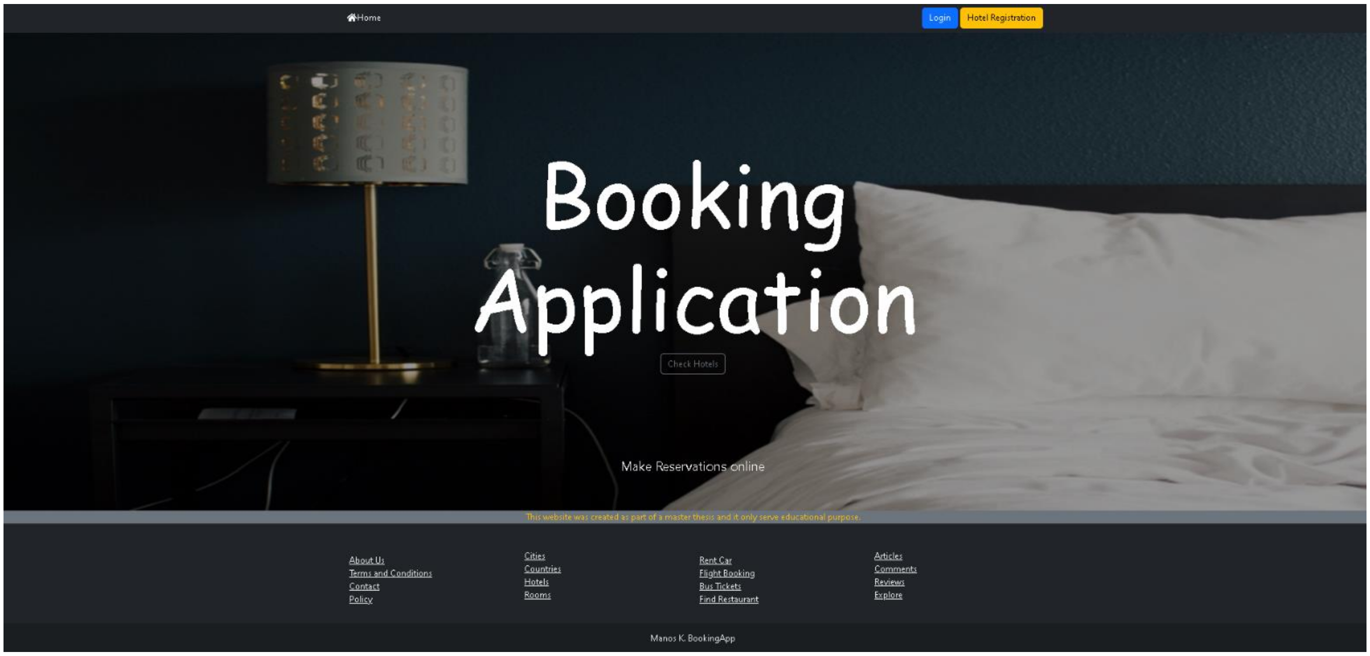
Figure 17 MYSQL Workbench

Chapter 4 – Implementation

In this chapter we will see the final application implemented and we will explain how it works through the two Use Cases we described in chapter 3.

4.1 – BookingApp website

The first thing our website users see in both cases is the home page.

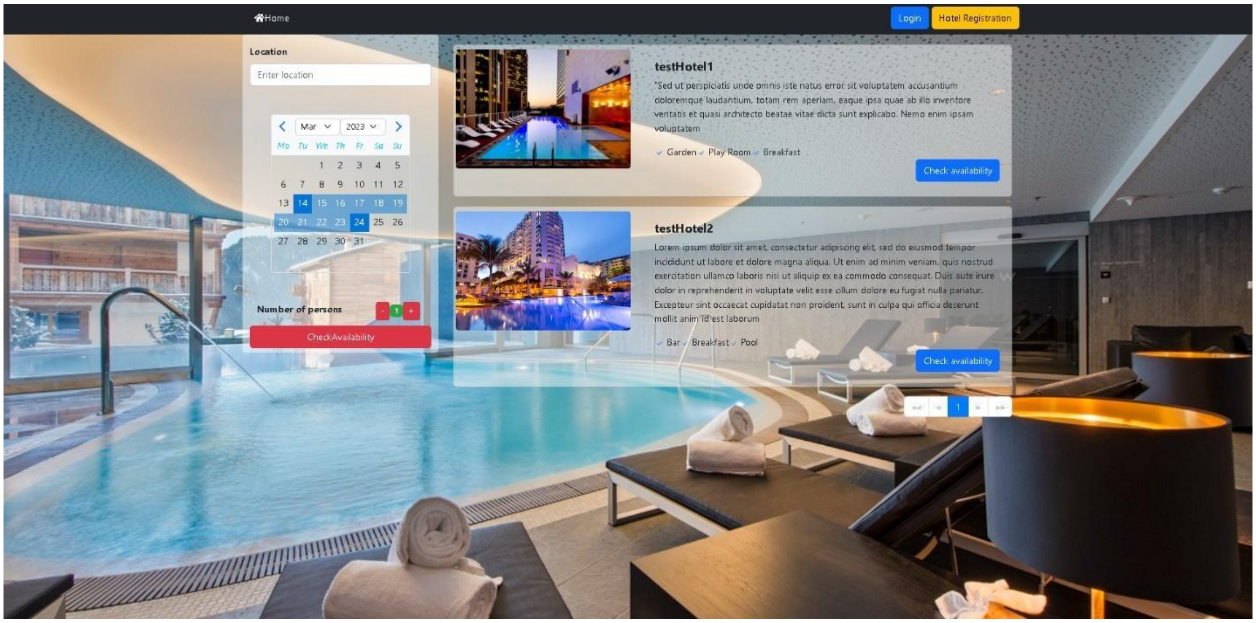


This website was created as part of a master thesis and it only serve educational purpose.

Figure 18 Homepage

From there the user has the option to see the hotels by pressing the Check Hotels button found in the middle of the page, he also has the option to register his hotel in the system via the Hotel Registration button and he has the option to log in to the system if he is an authorized user.

As you see in figure by pressing the Check hotels button the user can see all the available hotels in the system presented in the form of cards that contain a hotel image, services, description, and a button in case the user selects the hotel.



testHotel1

"Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem

✓ Garden ✓ Play Room ✓ Breakfast

testHotel2

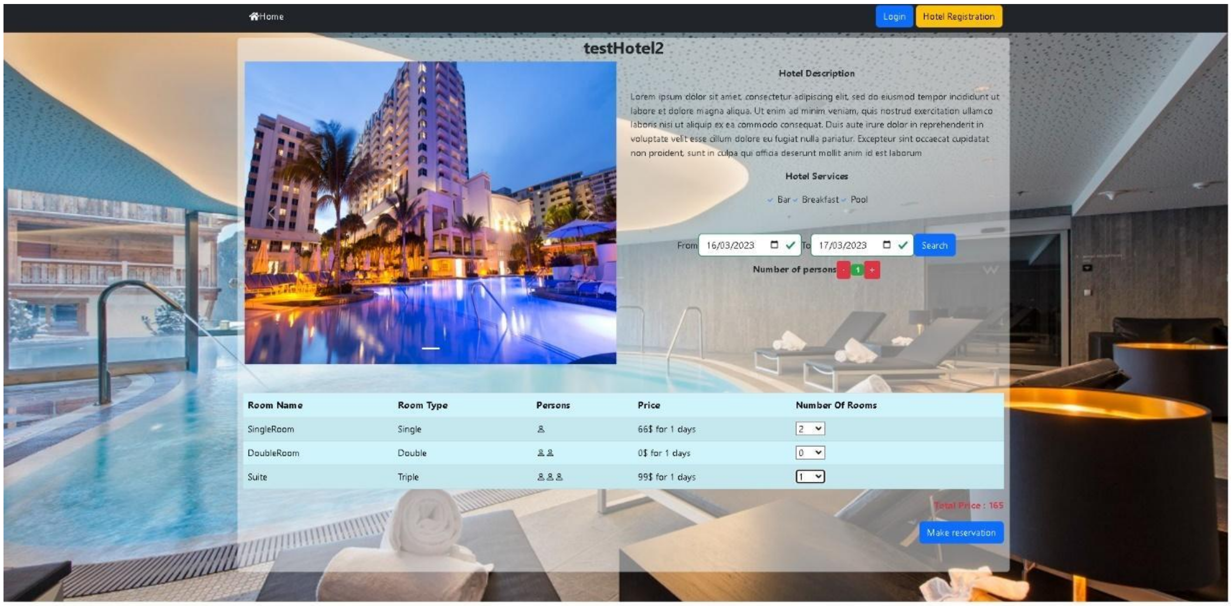
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

✓ Bar ✓ Breakfast ✓ Pool

Figure 19 Check hotels

From there the user can search for hotels based on some specific search criteria such as the city, the dates of arrival and departure as well as the number of people. By pressing the red check availability button, the grid refreshes with the hotels that match the search criteria.

In the case that the user selects one of the hotels, he must press the blue button found on the hotel's card. As can be seen in the image below, at this point the user can see more detailed information about the hotel he is interested in, search for available rooms and select rooms if he wishes.



testHotel2

Hotel Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Hotel Services

✓ Bar ✓ Breakfast ✓ Pool

From 16/03/2023 To 17/03/2023

Number of persons - 1 +

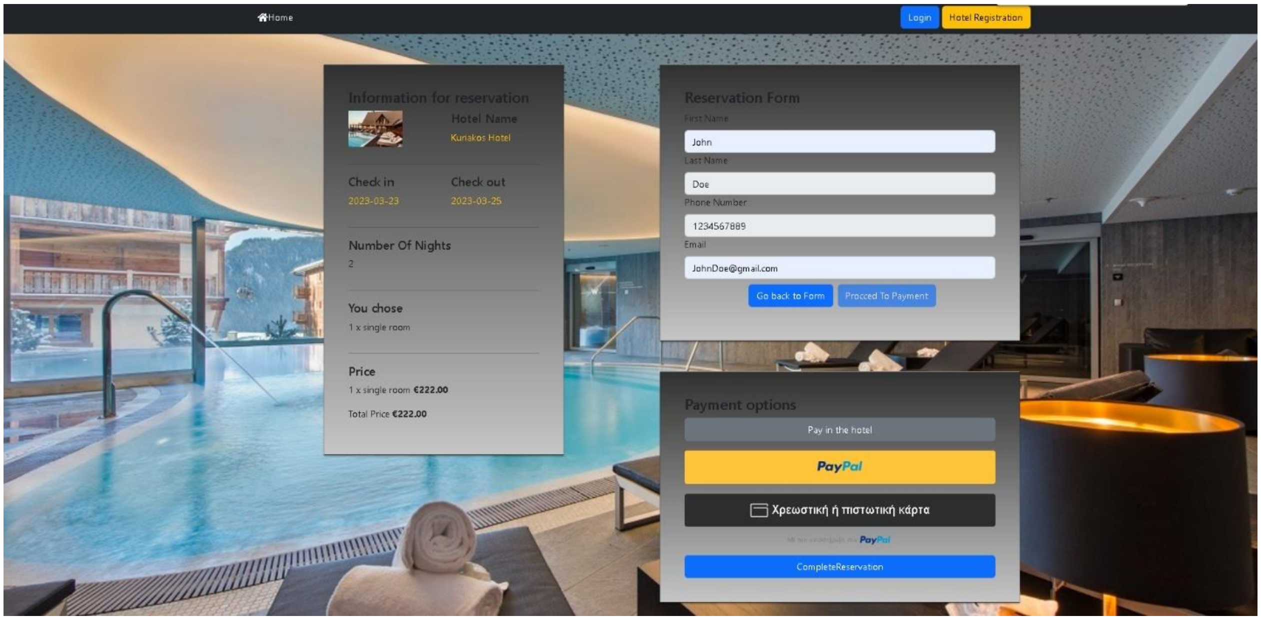
Room Name	Room Type	Persons	Price	Number Of Rooms
SingleRoom	Single	👤	66$ for 1 days	2
DoubleRoom	Double	👤👤	0$ for 1 days	0
Suite	Triple	👤👤👤	99$ for 1 days	1

Total Price : 165

Figure 20 Check room availability.

In case that user selects the rooms, he is interested in and proceeds with the reservation process, he must press the blue Make Reservation button at the bottom of the page. Here we must point out that the button becomes available for selection only when the user selects rooms.

When the user proceeds to make a reservation, the system redirects him to the following page as shown in the image below.



Information for reservation

Hotel Name
Kuriakos Hotel

Check in
2023-03-23

Check out
2023-03-25

Number Of Nights
2

You chose
1 x single room

Price
1 x single room **€222.00**
Total Price **€222.00**

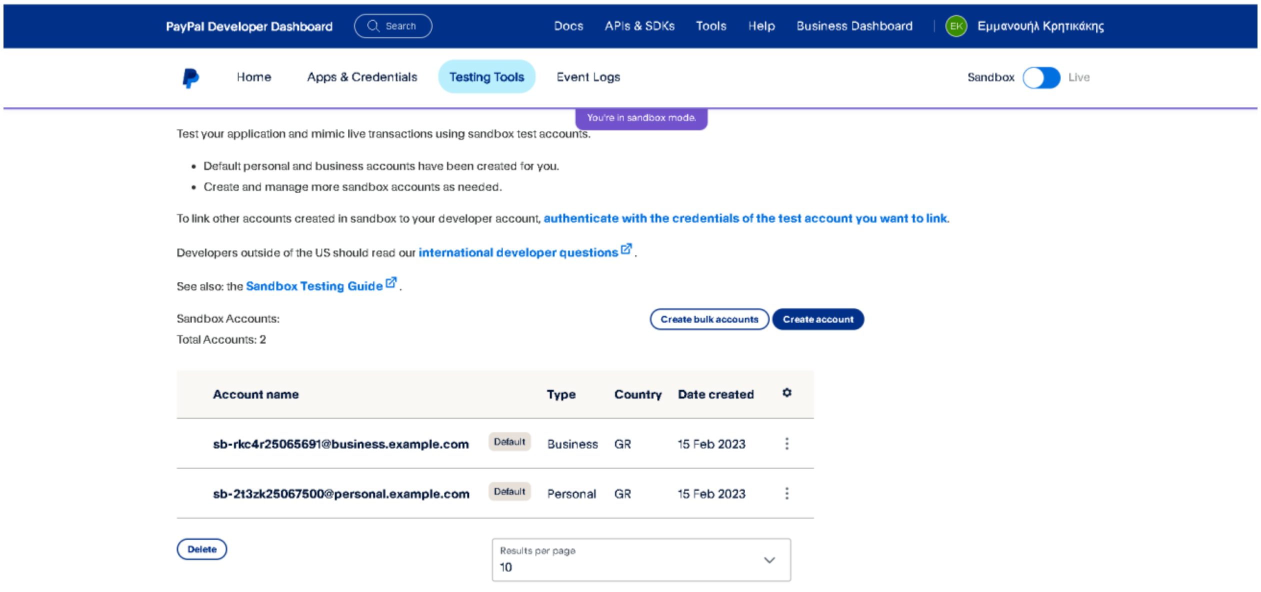
Reservation Form

First Name: John
Last Name: Doe
Phone Number: 1234567889
Email: JohnDoe@gmail.com

Payment options

Figure 21 Reservation

On the left side of the page the user sees information about his reservation and on the right side there is the form with the reservation details that the user must fill out. When the details are filled in, the user must choose a payment method to complete their reservation. If he chooses Pay in the Hotel, he must complete his reservation via the blue button at the bottom of the page. If the user chooses to pay by PayPal or card, the system proceeds to complete the booking automatically if the payment is successful. In the second case for testing purposes, we need to obtain the required credentials for the transaction from PayPal sandbox website. The figure above shows the available accounts sandbox created for us.



Test your application and mimic live transactions using sandbox test accounts.

- Default personal and business accounts have been created for you.
- Create and manage more sandbox accounts as needed.

To link other accounts created in sandbox to your developer account, **authenticate with the credentials of the test account you want to link.**

Developers outside of the US should read our **international developer questions**.

See also: the **Sandbox Testing Guide**.

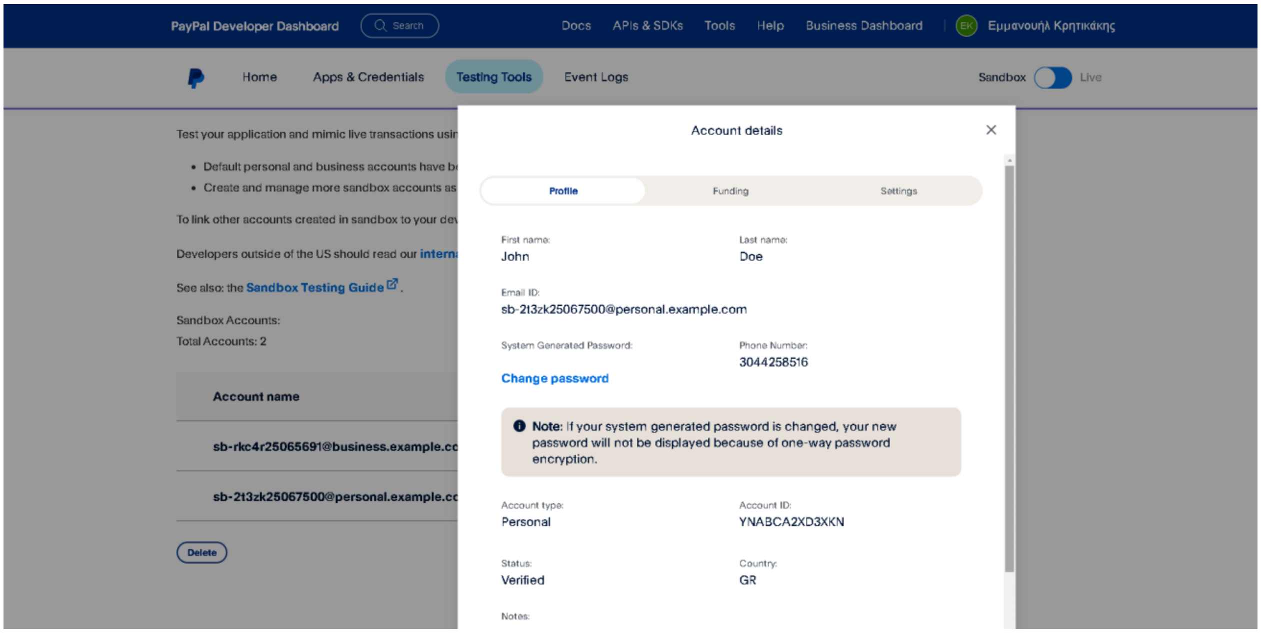
Sandbox Accounts:
Total Accounts: 2

Account name		Type	Country	Date created
sb-rkc4r25065691@business.example.com	Default	Business	GR	15 Feb 2023
sb-2t3zk25067500@personal.example.com	Default	Personal	GR	15 Feb 2023

Results per page
10

Figure 22 Sandbox accounts website

We use the credentials from personal account to demonstrate a PayPal payment in our website.



Account details

First name:
John

Last name:
Doe

Email ID:
sb-2t3zk25067500@personal.example.com

System Generated Password:

Phone Number:
3044258516

Change password

Note: If your system generated password is changed, your new password will not be displayed because of one-way password encryption.

Account type:
Personal

Account ID:
YNABCA2XD3XKN

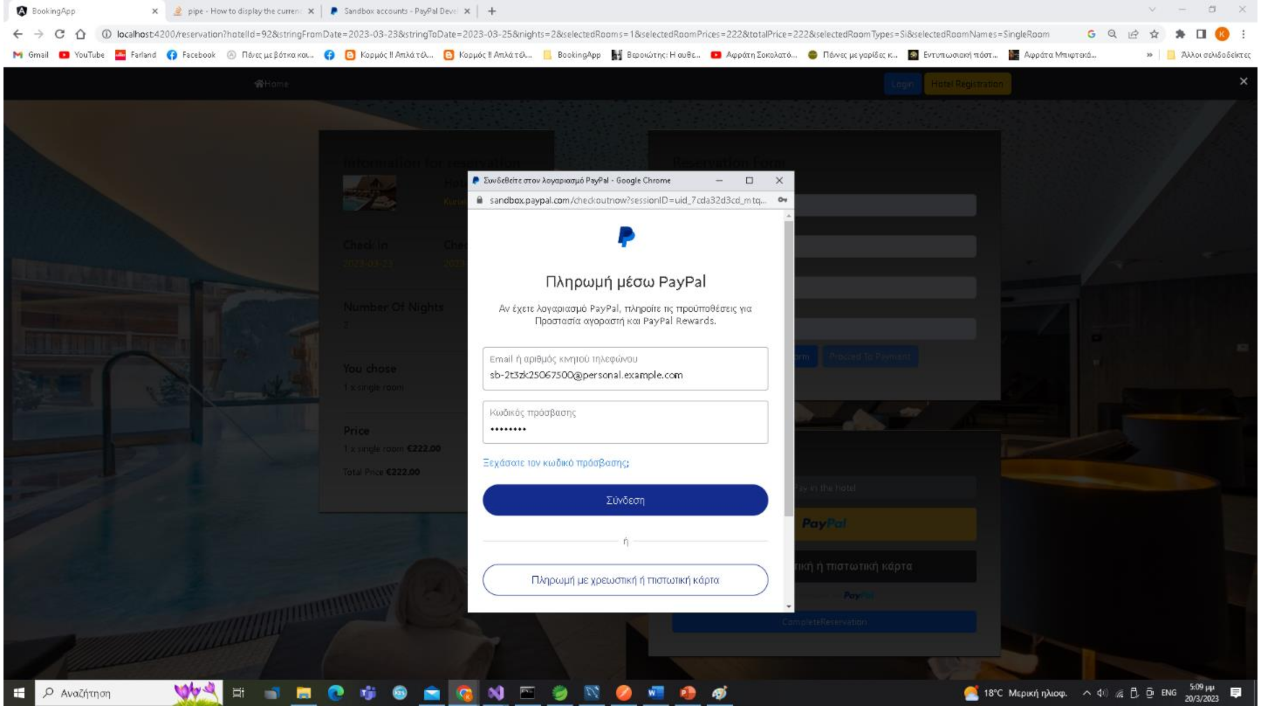
Status:
Verified

Country:
GR

Notes:

Figure 23 Sandbox personal account credentials

Using email and password from sandbox personal account we can proceed to payment by PayPal.



Συνδεθείτε στον λογαριασμό PayPal - Google Chrome

sandbox.paypal.com/checkoutnow?sessionID=uid_7cda32d3cd_mtq...

Πληρωμή μέσω PayPal

Αν έχετε λογαριασμό PayPal, πληροίτε τις προϋποθέσεις για Προστασία αγοραστή και PayPal Rewards.

Email ή αριθμός κινητού τηλεφώνου: sb-2t3zk25067500@personal.example.com

Κωδικός πρόσβασης: ••••••••

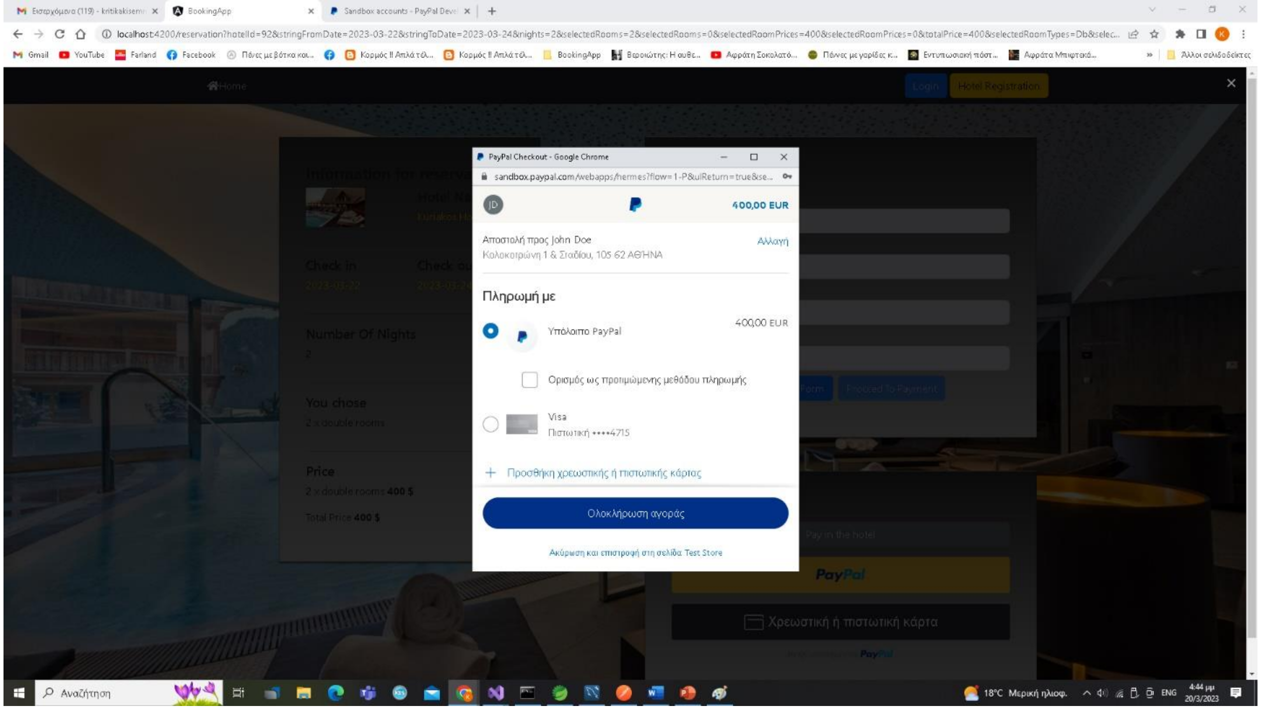
Ξεχάσατε τον κωδικό πρόσβασης;

Σύνδεση

ή

Πληρωμή με χρεωστική ή πιστωτική κάρτα

Figure 24 PayPal form login



PayPal Checkout - Google Chrome

sandbox.paypal.com/webapps/hermes?flow=1-P&ulReturn=true&se...

JD 400,00 EUR

Αποστολή προς John Doe — Αλλαγή

Κολοκοτρώνη 1 & Σταδίου, 105 62 ΑΘΗΝΑ

Πληρωμή με

● Υπόλοιπο PayPal 400,00 EUR

□ Ορισμός ως προτιμώμενης μεθόδου πληρωμής

○ Visa
Πιστωτική ••••4715

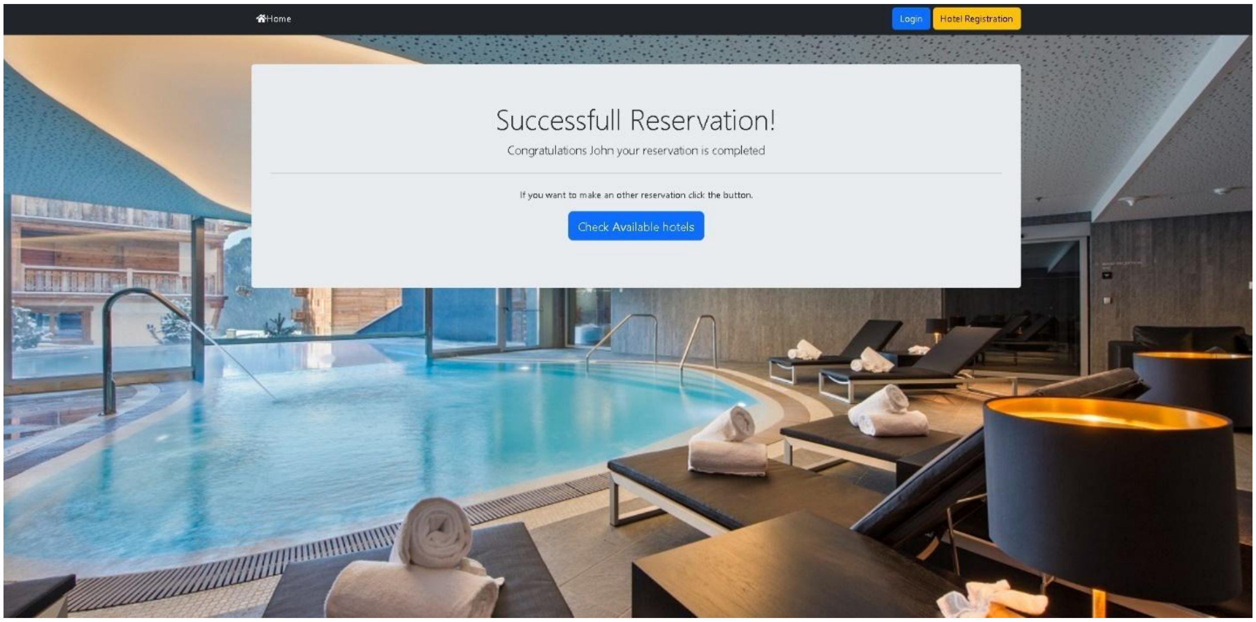
+ Προσθήκη χρεωστικής ή πιστωτικής κάρτας

Ολοκλήρωση αγοράς

Ακύρωση και επιστροφή στη σελίδα Test Store

Figure 25PayPal payment credits

In case of successful booking, the system displays the message shown in the image below.



Successfull Reservation!

Congratulations John your reservation is completed

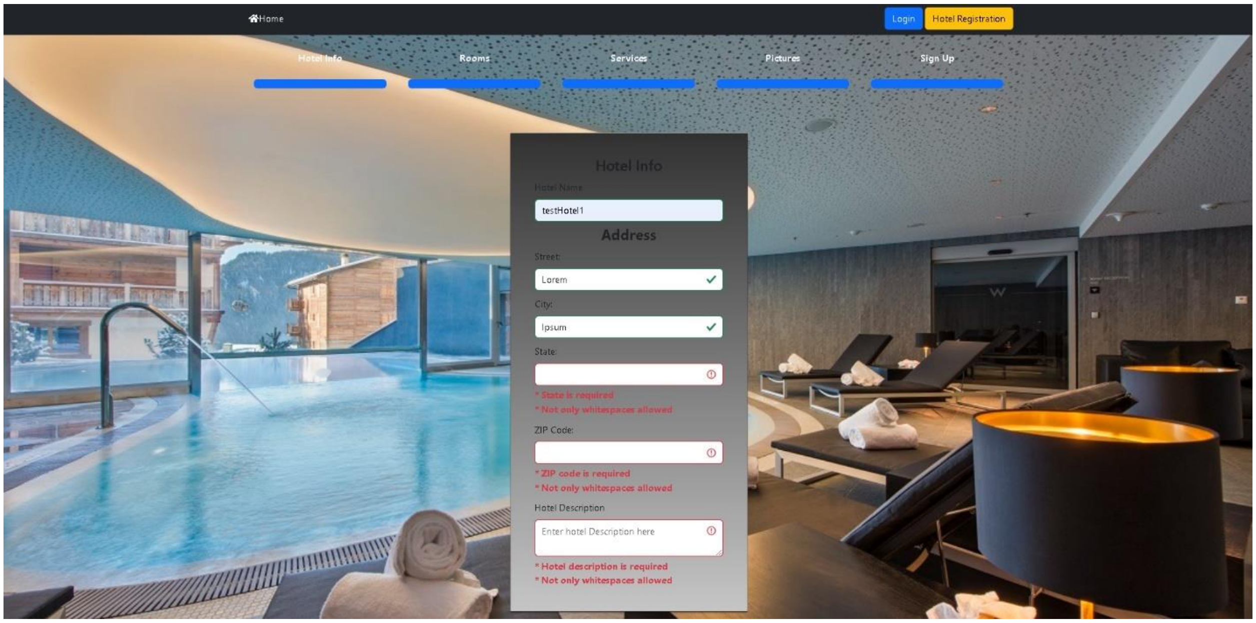
If you want to make an other reservation click the button.

Figure 26 Successful Reservation

There is a button there that redirects the user to the beginning of the process if desired.

If the user wants to register his hotel in the system, he must press the yellow button at the top of the page and follow the steps. The registration process contains 5 steps in which the user must provide the hotel information, provide available rooms, choose hotel services, upload images, create an account through which it will be managed by his hotel.

In the image below we see the first step of the process where the user must provide hotel name, address, and description.



Hotel Info

Hotel Name: testHotel1

Address

Street: Lorem ✓

City: Ipsum ✓

State:

* State is required
* Not only whitespaces allowed

ZIP Code:

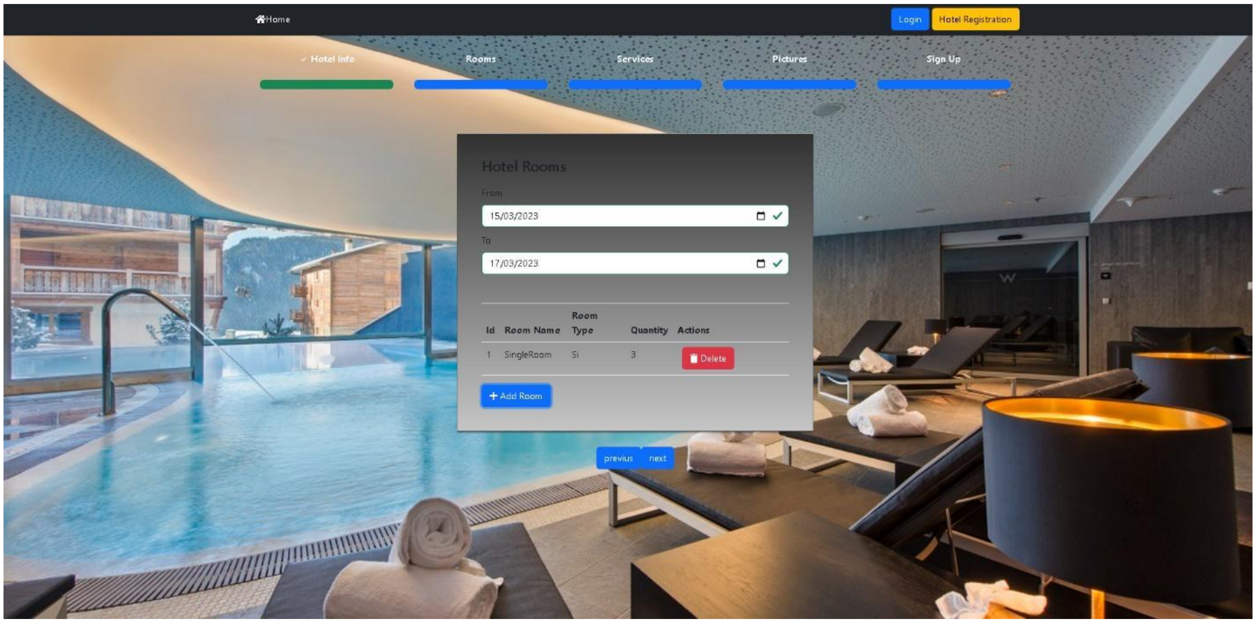
* ZIP code is required
* Not only whitespaces allowed

Hotel Description: Enter hotel Description here

* Hotel description is required
* Not only whitespaces allowed

Figure 27 Registration step 1

The second step follows where the user must state which rooms, he wants to make available and for specific date range. Date range is the first thing he needs to fill and then continues the room types through a modal that opens each time.



Hotel Rooms

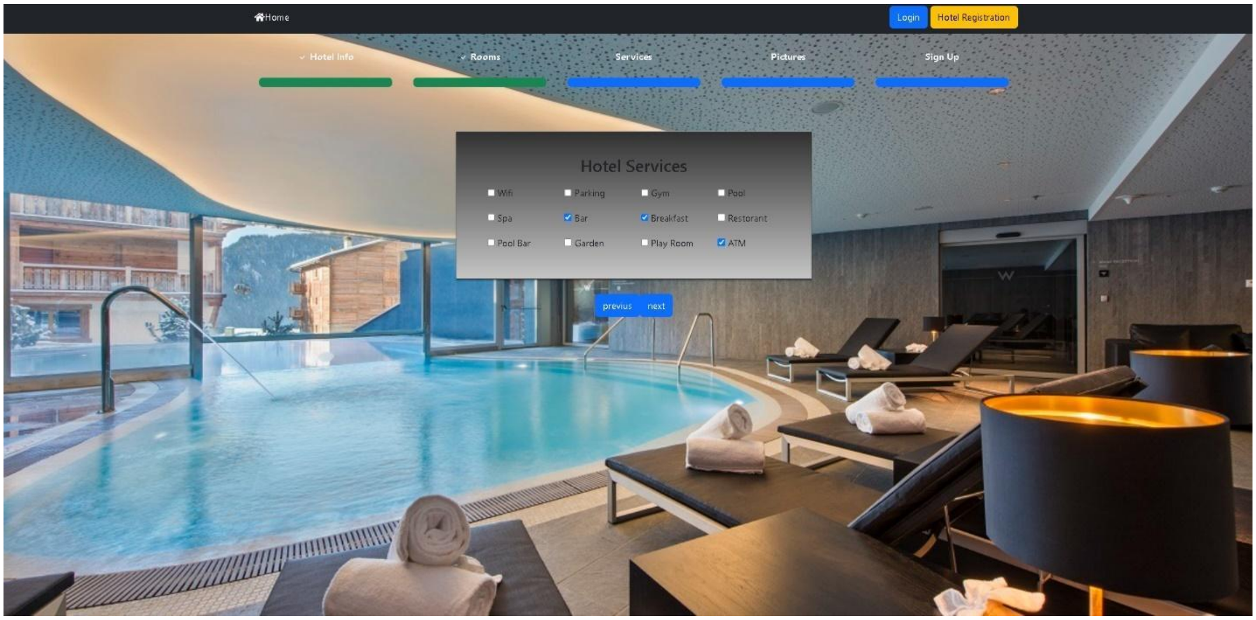
From: 15/03/2023 ✓

To: 17/03/2023 ✓

Id	Room Name	Room Type	Quantity	Actions
1	SingleRoom	Si	3	Delete

Figure 28 Registration step 2

The next step is optional and has to do with the services offered by the hotel. Here, if the user wants, he can choose the services and proceed. The process is shown in the image below.

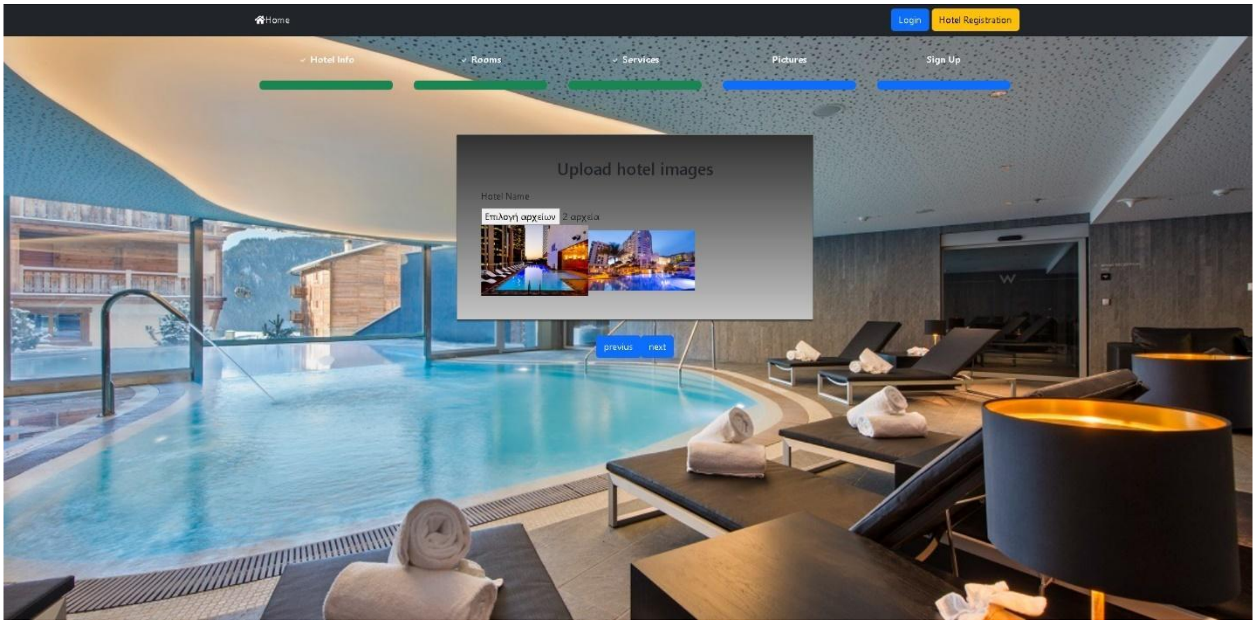


Hotel Services

□ Wifi	□ Parking	□ Gym	□ Pool
□ Spa	☑ Bar	☑ Breakfast	□ Restorant
□ Pool Bar	□ Garden	□ Play Room	☑ ATM

Figure 29 Registration step 3

Then the user must upload a photo of his hotel or more and press continue.



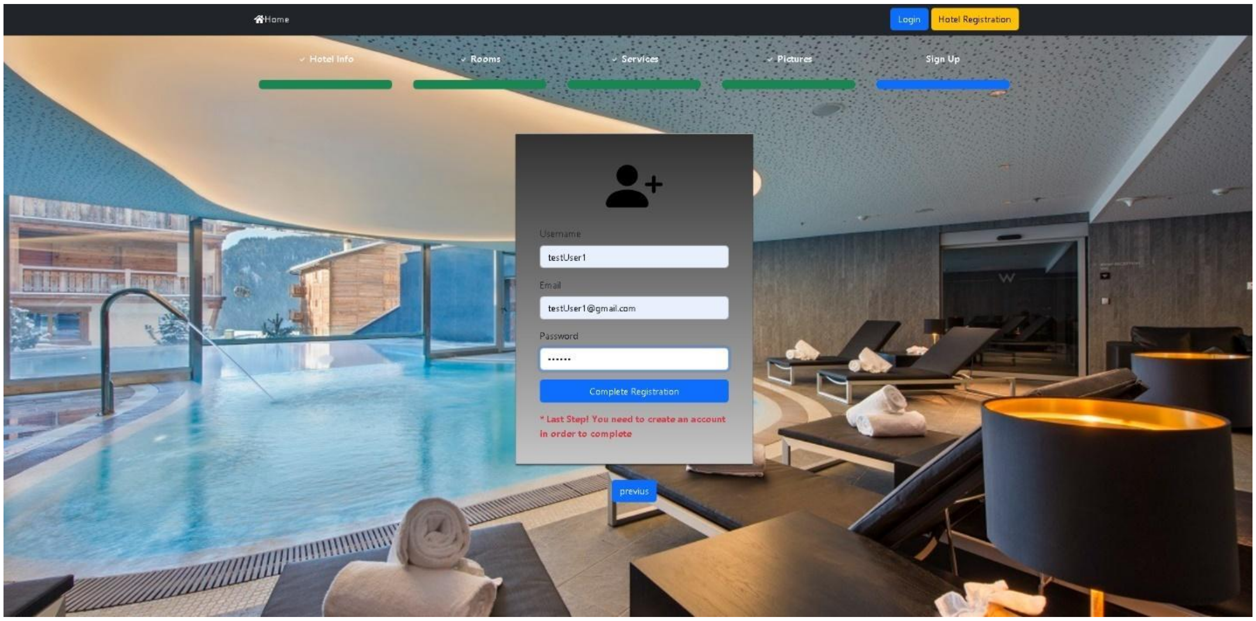
Upload hotel images

Hotel Name

Επιλογή αρχείων 2 αρχεία

Figure 30 Registration step 4

The last step has to do with creating an account that is linked to the hotel and has access to the system. There the user must provide the system with a unique username, unique email and password. The process is shown in the image below.



Username: testUser1

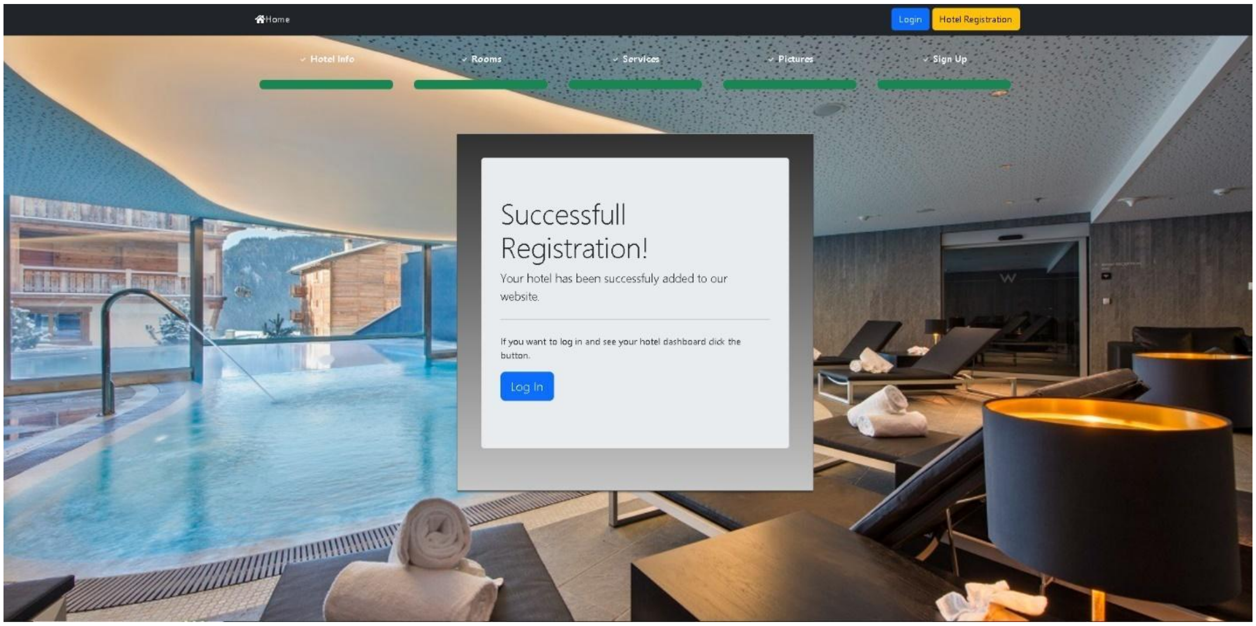
Email: testUser1@gmail.com

Password: ••••••

* Last Step! You need to create an account in order to complete

Figure 31 Registration step 5

Finally, by pressing the Complete Registration button, you complete the registration process and show the user the following message with the option to connect to the account he just created if he so wishes.



Successfull Registration!

Your hotel has been successfuly added to our website.

If you want to log in and see your hotel dashboard dick the button.

Figure 32 Successful Registration

The last Use Case scenario concerns users who have completed the registration process and want to manage their hotel. The system allows them to connect as shown in the image below.

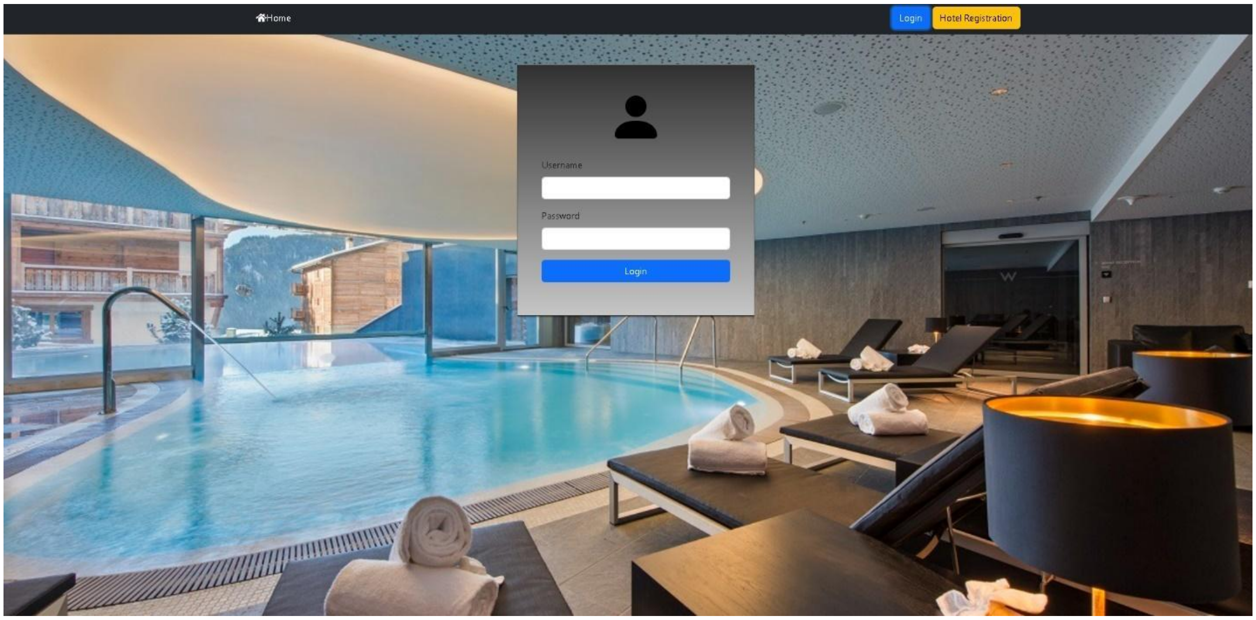
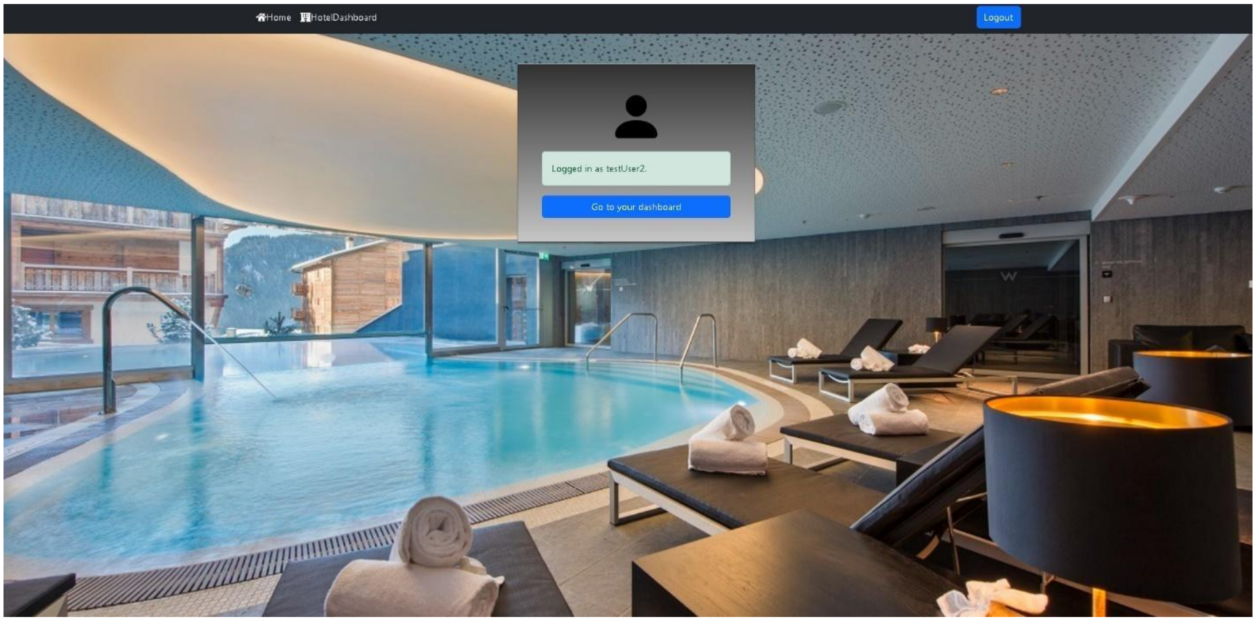


Figure 33 Login

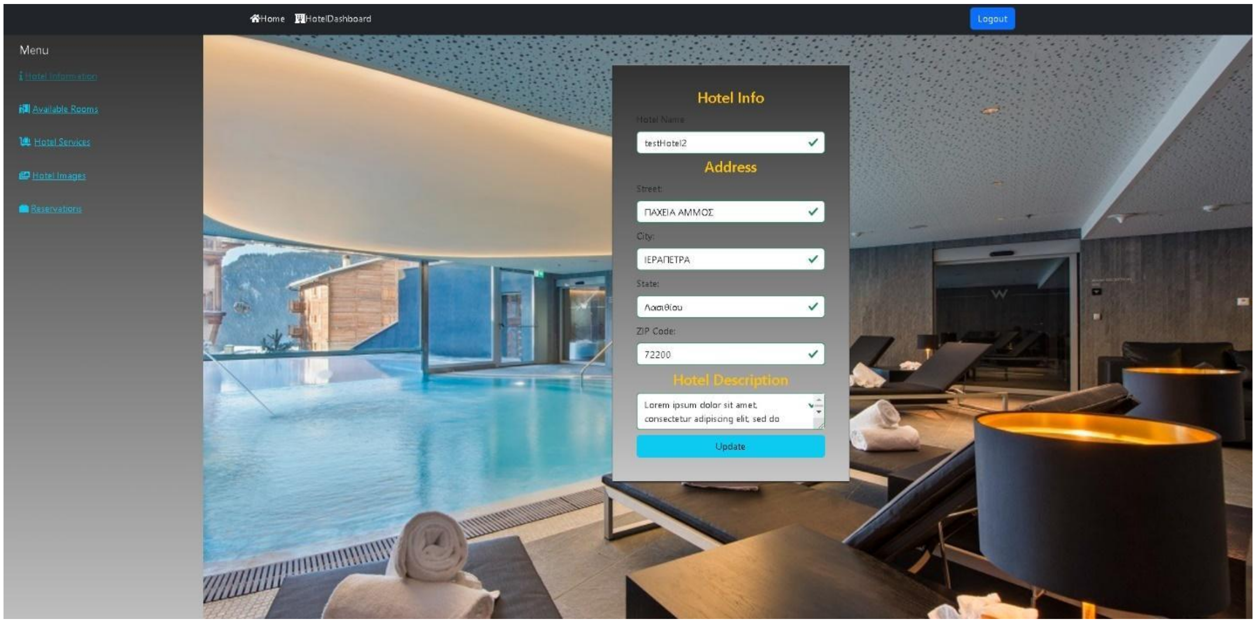
If login is successful, the system displays the hotel dashboard option in the top bar and gives them access to their hotel dashboard.



Logged in as testUser2.

Figure 34 Successful login

The dashboard contains five options for hotel management.



Hotel Info

Hotel Name: testHotel2

Address

Street: ΠΑΧΕΙΑ ΑΜΜΟΣ

City: ΙΕΡΑΠΕΤΡΑ

State: Λασιθίου

ZIP Code: 72200

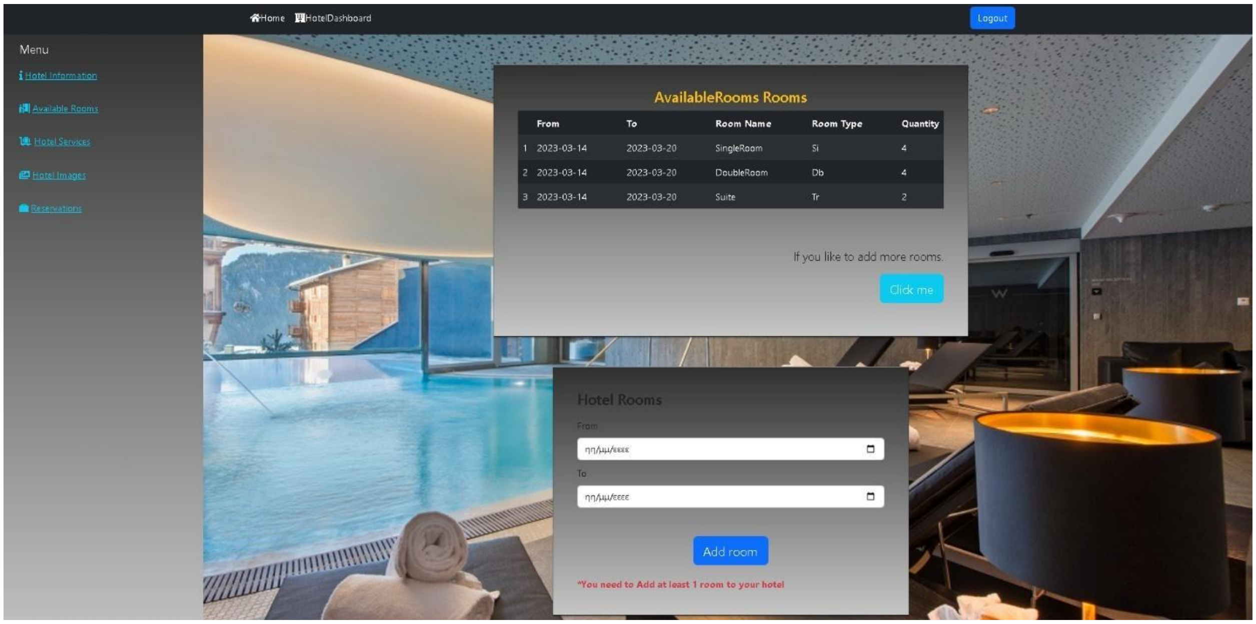
Hotel Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do

Figure 35 Hotel Dashboard hotel info

In the first case, as shown in the image above, the user can update hotel information and press the Update button

In the second tab of the dashboard, the user can see a table with the available rooms that he has declared and also add other available rooms if he wishes. The image below shows the second tab of the dashboard.



AvailableRooms Rooms

	From	To	Room Name	Room Type	Quantity
1	2023-03-14	2023-03-20	SingleRoom	Si	4
2	2023-03-14	2023-03-20	DoubleRoom	Db	4
3	2023-03-14	2023-03-20	Suite	Tr	2

If you like to add more rooms.

Hotel Rooms

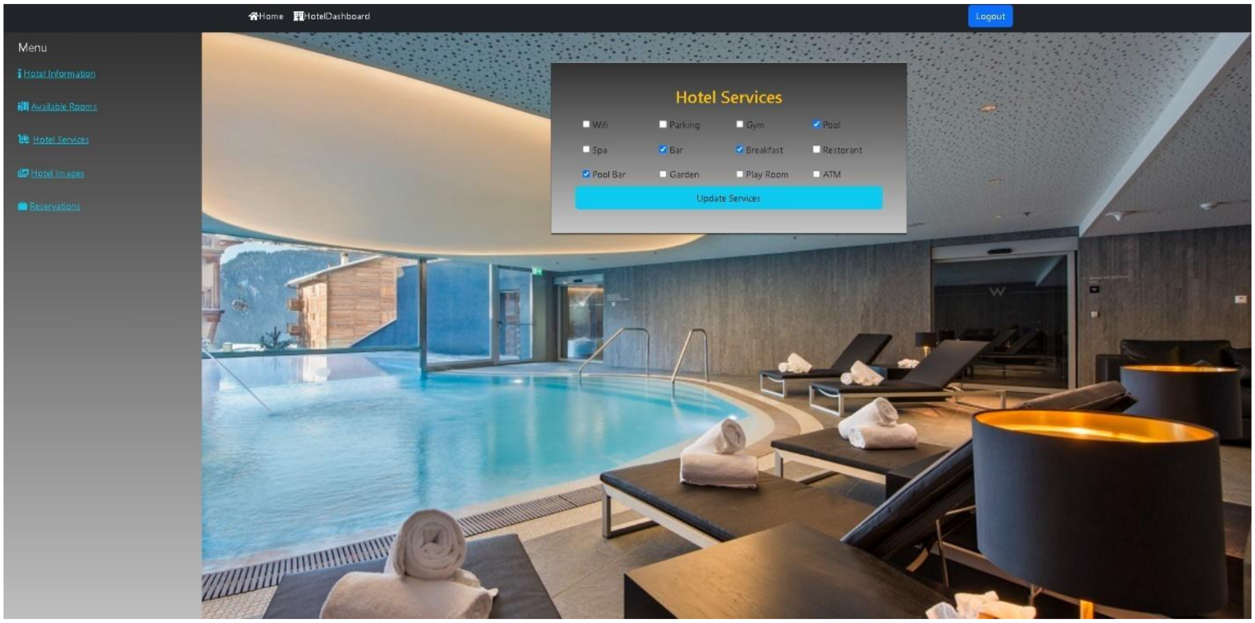
From: ηη/μμ/εεεε

To: ηη/μμ/εεεε

*You need to Add at least 1 room to your hotel

Figure 36Hotel Dashboard available rooms

In the third tab, the system loads the services dynamically and gives the user the possibility to update them through checkboxes, the relevant image follows.

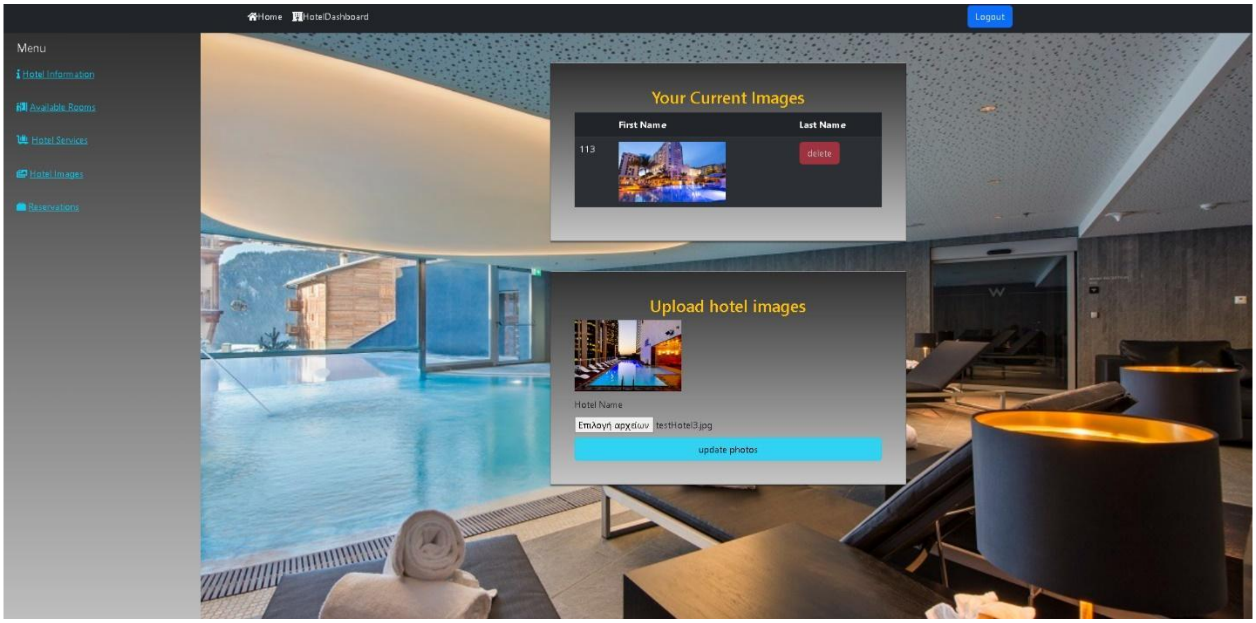


Hotel Services

□ Wifi □ Parking □ Gym ☑ Pool
□ Spa ☑ Bar ☑ Breakfast □ Restorant
☑ Pool Bar □ Garden □ Play Room □ ATM

Figure 37 Hotel Dashboard hotel services

Through the button entitled Update Services the user can apply his changes to the hotel. System also gives the user the option to add and delete images of his hotel, the only condition is that he is not allowed to have less than one image. The system takes care of this through the necessary code. The image below shows this process.



Your Current Images

	First Name	Last Name
113		delete

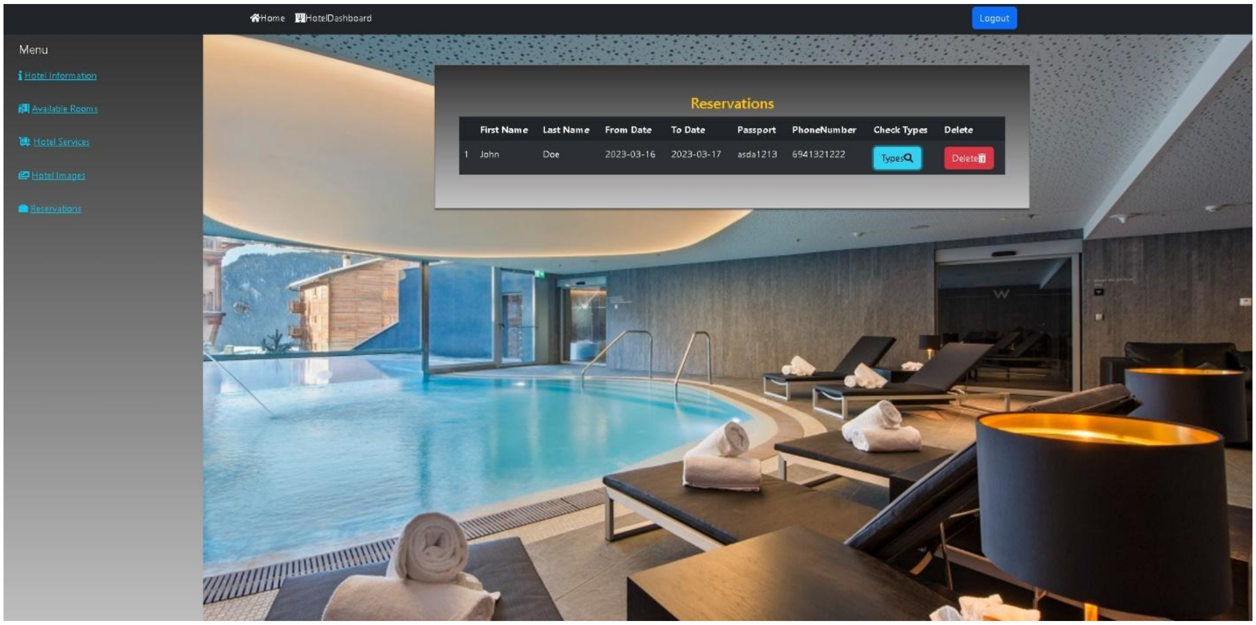
Upload hotel images

Hotel Name

Επιλογή αρχείων testHotel3.jpg

Figure 38 Hotel Dashboard hotel images

Finally, the user can see the reservations made by customers and the types of rooms they chose. User can delete a reservation if he wants.

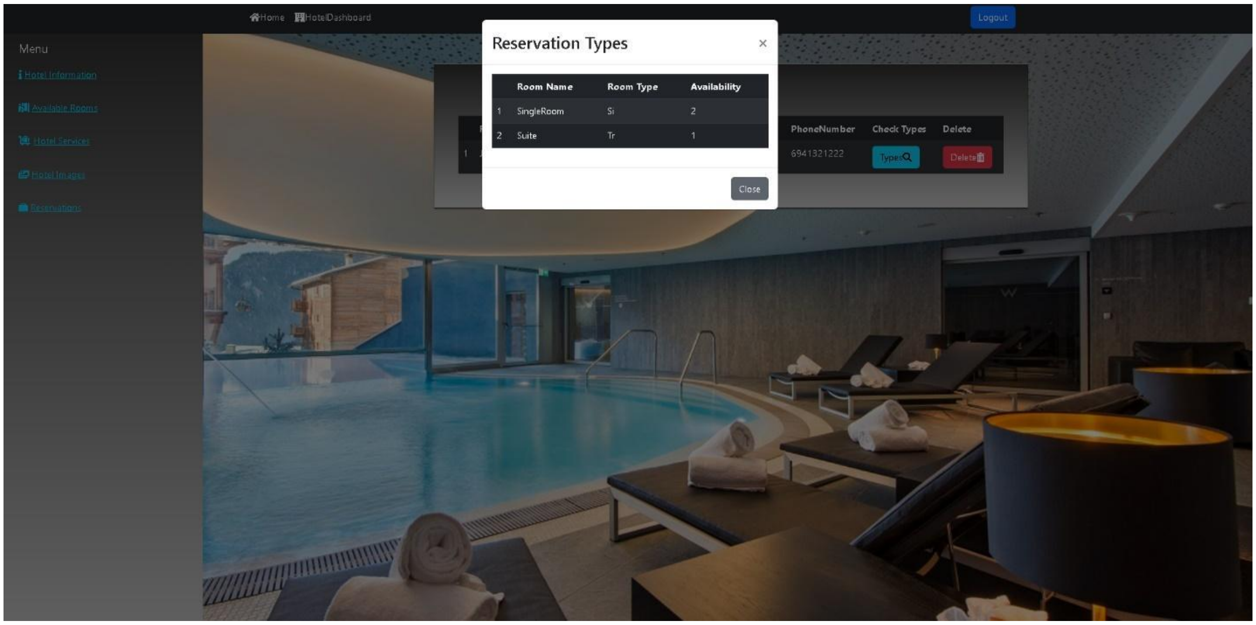


Reservations

	First Name	Last Name	From Date	To Date	Passport	PhoneNumber	Check Types	Delete
1	John	Doe	2023-03-16	2023-03-17	asda1213	6941321222	Types	Delete

Figure 39 Hotel Dashboard hotel reservations

With the blue button he can see the available types and with the red button he can delete the reservation.

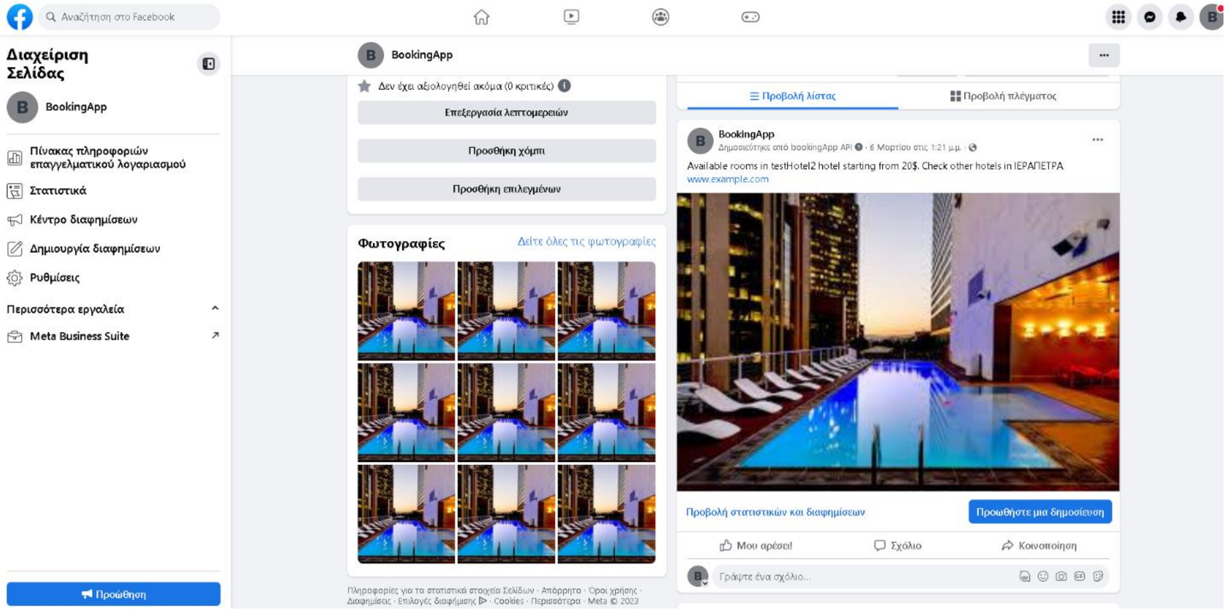


Reservation Types

	Room Name	Room Type	Availability
1	SingleRoom	Si	2
2	Suite	Tr	1

Figure 40 Hotel Dashboard hotel reservation types

In addition to these, the system automatically posts to a page on Facebook via server.



BookingApp
Δημοσιεύτηκε από bookingApp API · 6 Μαρτίου στις 1:21 μ.μ.

Available rooms in testHotel2 hotel starting from 20$. Check other hotels in ΙΕΡΑΠΕΤΡΑ
www.example.com

Figure 41 Facebook Ads

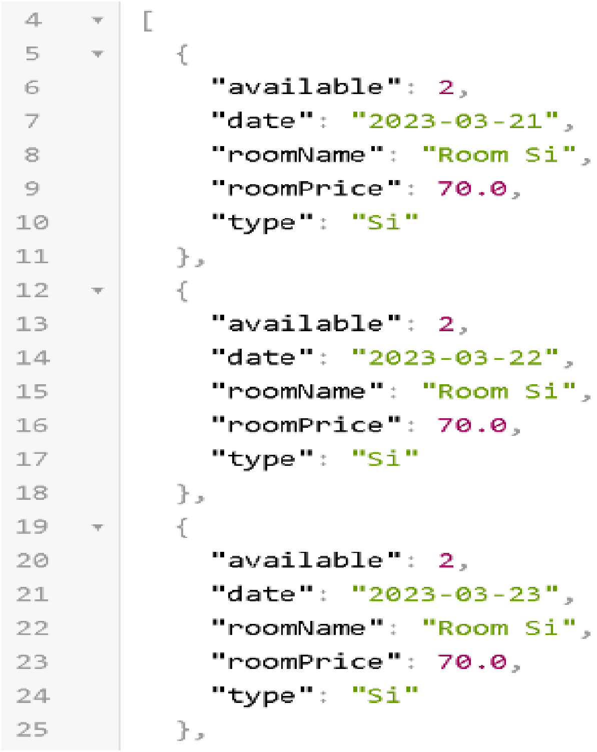
As shown in the image above, the system posts hotel name, minimum room rate and city in order to reach customers.

Chapter 5- Customized web service support

In this section, the customized web service support provided for the communication and sale of rooms will be analyzed. More specifically, a communication protocol was created for continuous communication between this system and a PMS application. The following endpoints are provided for communication:

1. bookingEngineReservations/getBookingAppAvailability
2. bookingEngineReservations/saveReservation

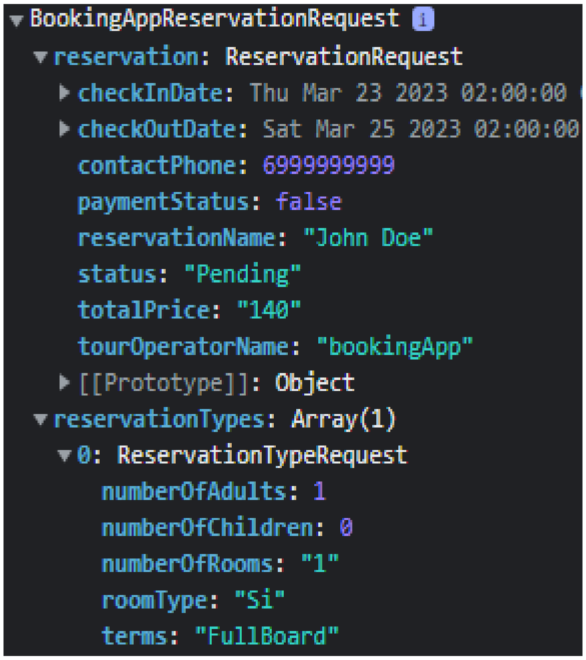
The first endpoint provides availability and the second accepts and stores the reservations that have been made through the booking application. The above endpoints are public so they can be used by external applications. Calling the first endpoint returns a JSON file of the structure shown in Figure 76 which includes the number of rooms according to their type and their availability per date.



```
[
  {
    "available": 2,
    "date": "2023-03-21",
    "roomName": "Room Si",
    "roomPrice": 70.0,
    "type": "Si"
  },
  {
    "available": 2,
    "date": "2023-03-22",
    "roomName": "Room Si",
    "roomPrice": 70.0,
    "type": "Si"
  },
  {
    "available": 2,
    "date": "2023-03-23",
    "roomName": "Room Si",
    "roomPrice": 70.0,
    "type": "Si"
  },
```

Figure 42 Get Availability Endpoint

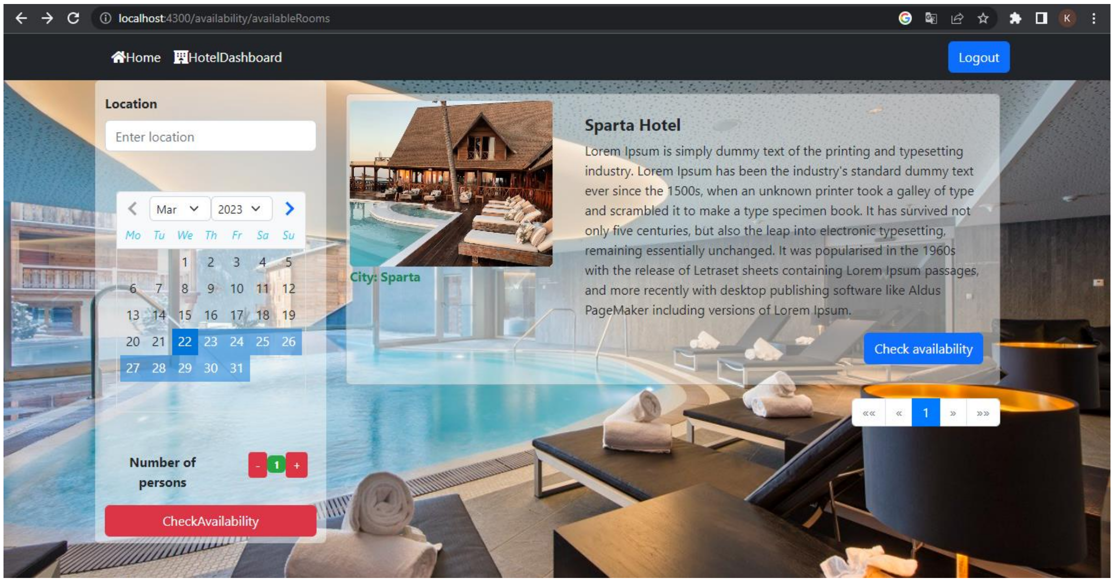
The information about room pricing per night is also provided. Calling the second endpoint returns a JSON file of the structure shown in Figure 77 which includes information about reservation such as check-in date, check-out date, reservation status, customer info, total price, payment status, and information about reservation types such as room type, number of rooms, number of children, number of adults and nutrition plan.



```
▼ BookingAppReservationRequest
  ▼ reservation: ReservationRequest
    ▶ checkInDate: Thu Mar 23 2023 02:00:00
    ▶ checkOutDate: Sat Mar 25 2023 02:00:00
      contactPhone: 6999999999
      paymentStatus: false
      reservationName: "John Doe"
      status: "Pending"
      totalPrice: "140"
      tourOperatorName: "bookingApp"
    ▶ [[Prototype]]: Object
  ▼ reservationTypes: Array(1)
    ▼ 0: ReservationTypeRequest
        numberOfAdults: 1
        numberOfChildren: 0
        numberOfRooms: "1"
        roomType: "Si"
        terms: "FullBoard"
```

Figure 43 Save Reservation Endpoint

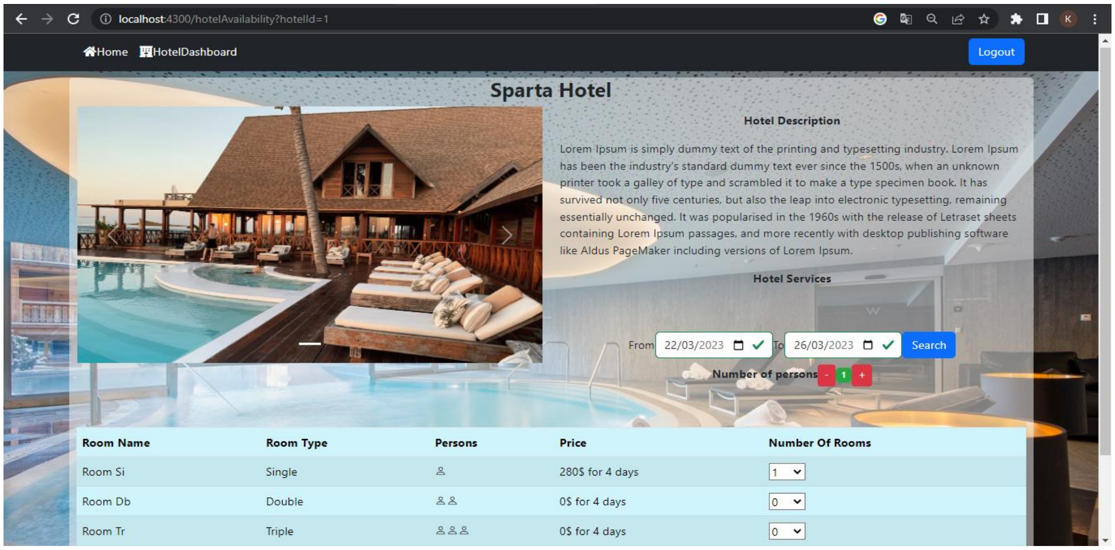
As shown in Figure 78, the available hotels of the booking application are shown, where the hotel that appears in the list is the hotel that uses the specific PMS. Then, as seen in Figure 79, the user selects a single room for the dates 2023-03-22 to 2023-03-26. All room availability data shown in the image are updated dynamically based on the customized web service.



Sparta Hotel

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Figure 44 Available Hotels Booking System



Sparta Hotel

Hotel Description

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

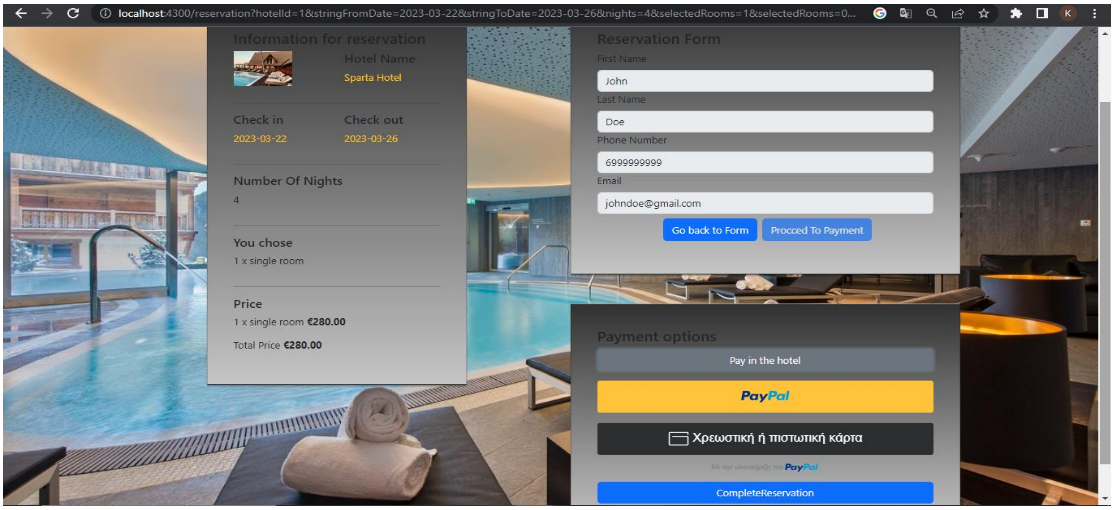
Hotel Services

From 22/03/2023 To 26/03/2023

Room Name	Room Type	Persons	Price	Number Of Rooms
Room Si	Single	👤	280$ for 4 days	1
Room Db	Double	👤👤	0$ for 4 days	0
Room Tr	Triple	👤👤👤	0$ for 4 days	0

Figure 45 Available Rooms Of Hotel Booking System

Then, in Figure 80 below, the form that the user must fill in with his personal details as well as the payment options can be seen. Finally, Figure 81 shows that the reservation from the side of the booking application was successful.



localhost:4300/reservation?hotelId=1&stringFromDate=2023-03-22&stringToDate=2023-03-26&nights=4&selectedRooms=1&selectedRooms=0...

Information for reservation

Hotel Name
Sparta Hotel

Check in
2023-03-22

Check out
2023-03-26

Number Of Nights
4

You chose
1 x single room

Price
1 x single room **€280.00**

Total Price **€280.00**

Reservation Form

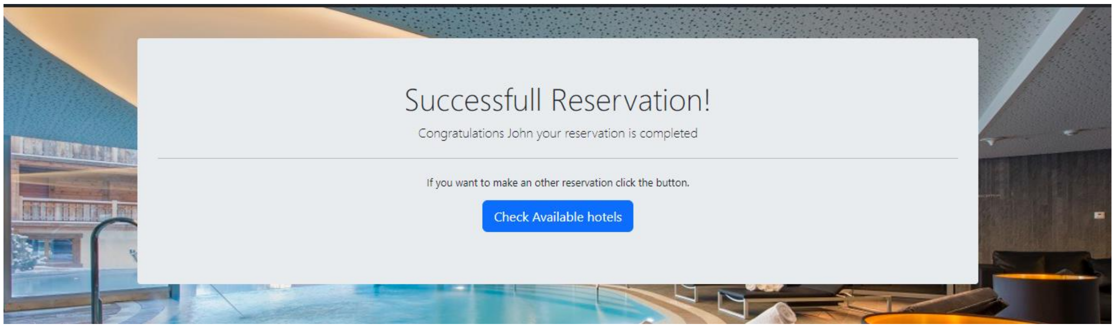
First Name: John

Last Name: Doe

Phone Number: 6999999999

Email: johndoe@gmail.com

Figure 46 Payment Booking System



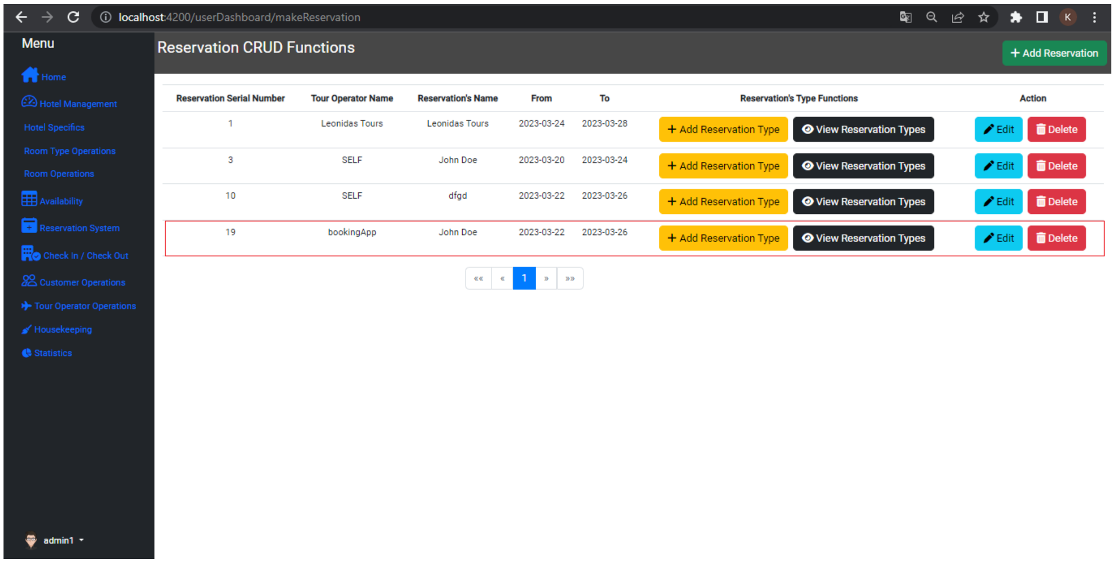
Successfull Reservation!

Congratulations John your reservation is completed

If you want to make an other reservation click the button.

Figure 47Successful Reservation Booking System

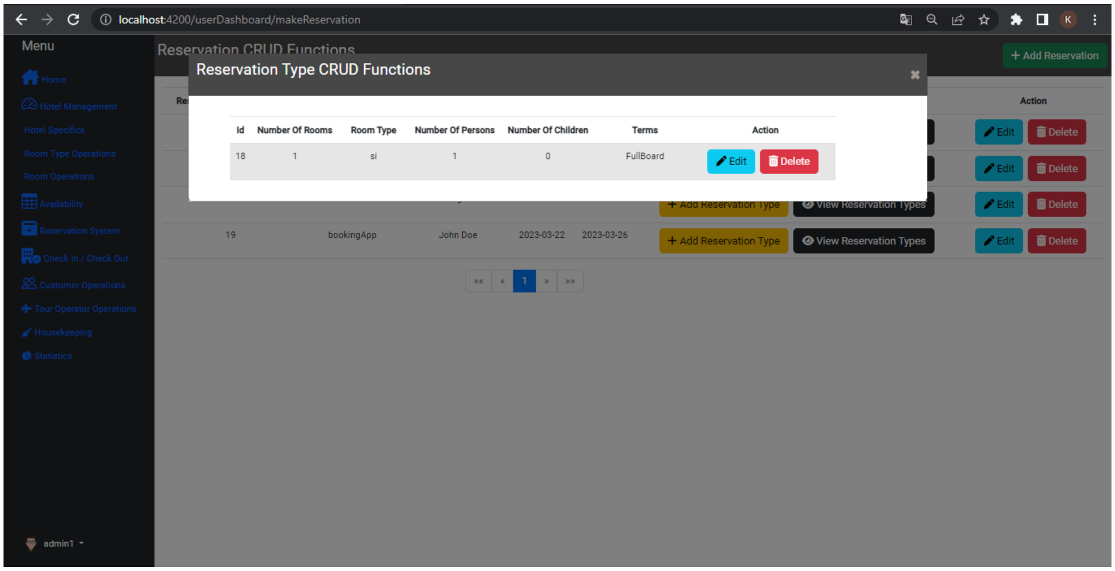
In Figure 81 below, the list of reservations can be seen, and it can be observed that the last reservation is the reservation received from the booking application. Finally, Figure 82 illustrates the reservation types of the received reservation.



Reservation CRUD Functions

Reservation Serial Number	Tour Operator Name	Reservation's Name	From	To	Reservation's Type Functions	Action
1	Leonidas Tours	Leonidas Tours	2023-03-24	2023-03-28	+ Add Reservation Type View Reservation Types	Edit Delete
3	SELF	John Doe	2023-03-20	2023-03-24	+ Add Reservation Type View Reservation Types	Edit Delete
10	SELF	dfgd	2023-03-22	2023-03-26	+ Add Reservation Type View Reservation Types	Edit Delete
19	bookingApp	John Doe	2023-03-22	2023-03-26	+ Add Reservation Type View Reservation Types	Edit Delete

Figure 48 List of Reservations



Reservation Type CRUD Functions

Id	Number Of Rooms	Room Type	Number Of Persons	Number Of Children	Terms	Action
18	1	si	1	0	FullBoard	Edit Delete

Figure 49 View Reservation Types

Figure 84 below illustrates the communication architecture between the PMS application and the Booking application. More specifically, through the communication protocol that was created, the Booking app can request availability from the PMS through an http GET request. Also, the Booking application makes an http POST request to the PMS application when a new booking is made. With the architecture described above, the communication between the PMS and the Booking App was ensured in real time.

Architecture with customized web support

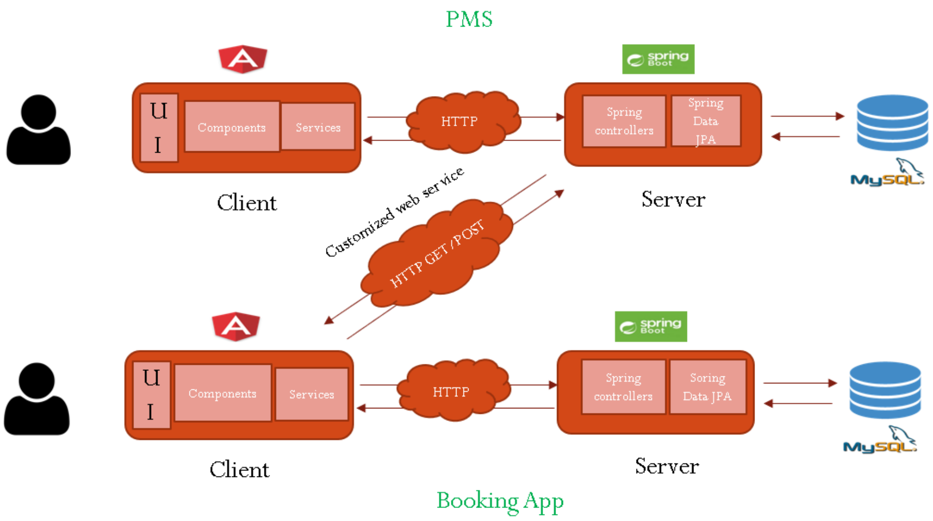


Figure 50 Architecture with customized web service support

Chapter 6- Conclusion and Future work

6.1 Conclusion

In conclusion, we could say that creating an online travel agent is a very interesting process. It is one of the first full stack applications that I designed and implemented, and it gave me the opportunity to apply techniques and technologies that I did not have to work with in the past. It's a hot topic in the ever-growing tourism industry and worth investing some time into. My goal was to create a system that will help hotels increase their profit and help customers find what they want.

6.2 Future Work

The development of such a system has many requirements and difficulties, certainly some parts of the work could be done differently, and more things could be added. Some of these ideas are shown below:

- Add more hotel search filters.
- Add more information about hotel rooms such as pictures and services.
- We could send the reservations via email to the clients.
- We could add other types of accommodation besides hotels e.g., Apartments, cabins etc.
- The system could also promote other services such as finding tickets or restaurants.
- The system could be redesigned to be fully responsive on all devices.
- Add other security mechanisms.
- Add more features to the dashboard.
- To redesign the system for Facebook ads so that it includes other parameters for choosing the hotel that advertises.
- Give users more option to register to the system.
- Add the location of the hotels on a map.

Certainly, many more ideas could improve and make the system more complete, nevertheless I hope I managed to create a worthwhile application.

References

[1] Timothy Webb, (2016), "Form travel agents to OTAs: How the evolution of consumer booking behavior has affected revenue management", Journal of Revenue and Pricing Management, pp. 276-282, DOI :10.1057/rpm.2016.16

[2] Sunmee Choi and Sheryl E. Kimes, (2002), "Electronic Distribution Channels' Effect on Hotel Revenue Management", Volume 43 Issue 3, June 2002 DOI: 10.1177/0010880402433002

[3] Soy Sut Ieng Lei, Juan L. Nicolau, Dan Wang, (2019), "The impact of distribution channels on budget hotel performance", pp. 141-149, DOI: 10.1016/j.ijhm.2019.03.005

[4] Dalia Zaki, (2016), "Internet VS. Travel Agencies: The Effect of Online Booking on Travelers Purchasing Behavior and the Future of Travel Agencies", Minia Journal of Tourism and Hospitality, Volume 1, no 1

[5] Bill Carroll and Lorraine Sileo (2014) "Online Travel Agencies: More Than a Distribution Channel" PhoCusWright White Paper

[6] Singh, Sujay & Ranjan, Rajeev, (2019), ONLINE TRAVEL PORTAL AND THEIR EFFECT ON TRAVEL AGENCY: A STUDY ON OUTBOUND VISITORS OF VARANASI, 6, pp. 387-393. 10.1729/Journal.21000.

[7] Laura Fredericks (2021) "The Complete Guide to Hotel Distribution Channels" https://www.cvent.com/en/blog/hospitality/hotel-distribution-channels

[8] Revenue hub "A Few Thoughts on Your Hotel's Distribution & Channel Strategy" https://revenue-hub.com/a-few-thoughts-on-your-hotels-distribution-channel-strategy/

[9] Expedia Group "What's an Online Travel Agency and Why Should I Care?" https://welcome.expediagroup.com/en/resources/hotel-distribution-strategy-resources-tips/otas-work-use-one

[10] Patrick Landman "OTA – Online Travel Agency" https://www.xotels.com/en/glossary/ota

[11] Rob Mathison (2022) “Advantages and Disadvantages of Online Travel Agencies (OTAs)” https://www.rezgo.com/blog/advantages-and-disadvantages-of-online-travel-agencies-otas/

[12] JungleBee (2022) “The Pros and Cons of OTA Partnerships for Tours & Activities”

[13] Booking.com “About us” https://www.booking.com/content/about.el.html?aid=356980&keep_landing=1&sid=0f2a04058e73ef17728928a5c3de3ab3&label=gog235jc-1DCBQoggJCBWFib3V0SDNYA2hciAEBmAEIuAEXyAEM2AED6AEBiAIBqAIDuALnoKGgBsACAdICJDM0ZThmMzE3LTAyNzgtNDdkZS04ZDI5LTg1Mjc1ZWEwOGE1YtgCBOACAQ

[14] Expedia.com https://www.expedia.com/

[15] Trip.com “About us” https://gr.trip.com/pages/about-us/

[16] Cloudbeds Article “Top Online Travel Agencies” https://www.cloudbeds.com/online-travel-agencies/best/

[17] Dimitra Psefti, Ioulia Poulaki, Alkistis Papaioannou & Vicky Katsoni (2021) “The Evolution of Online Travel Agencies in the Last Decade: E-Travel SA as an Exceptional Paradigm” Part of the Springer Proceedings in Business and Economics book series (SPBE), pp. 01-03 DOI: 10.1007/978-3-030-72469-6_40

[18] Phocuswright (2022) “The State of Online Travel Agencies” https://www.hospitalitynet.org/news/4109835.html

[19] Angular Documentation (2022) https://angular.io/guide/what-is-angular

[20] Spring Documentation (2023) https://docs.spring.io/spring-framework/docs/current/reference/html/

[21] Spring Boot Reference Documentation (2023) https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/

[22] Facebook Graph API (2023) “Overview” https://developers.facebook.com/docs/graph-api/overview

[23] PayPal sandbox Testing Guide (2023) https://developer.paypal.com/tools/sandbox/

[24] MySQL workbench https://www.mysql.com/products/workbench/