

**Web Applications A.Y. 2021-2022**  
**Homework 1 – Server-side Design and Development**

**Master Degree in Computer Engineering**  
**Master Degree in Cybersecurity**  
**Master Degree in ICT for Internet and Multimedia**

Deadline: 22 April, 2022

Group	Project	
MANTA	Hotel Management Web App	
Heldal	Julie Marie	206086
Hermansen	Kathrine	2060854
Kosiakov	Arina	1187327
Lotta	Tommaso	2054765
Manuilova	Nadezhda	2049581
Mohammadi	Mohammadreza	2005535
Roennestad	Andreas	2037142
Vignaga	Marco	2053043
Zuccolo	Giada	2055702

# 1 Objectives of the System

The objective of the project, in continuation with the one developed during the Foundation of Databases course, is to develop a web interface to manage data for the Hotel Booking System. The project focuses on an online booking system that allows guests to make secure online bookings through the hotel's website and helps hotels accept bookings and collect payments online.

## 2 Main Functionalities

This web application will be mainly used to allow registered users to book a room in the hotel (with different booking options based on the logged account) and to enroll for the events that are taken in the hotel.

The web interface allows also to modify user's data and add new users. It will also be possible for the hotel manager, and it's employees, to manage orders received from customers.

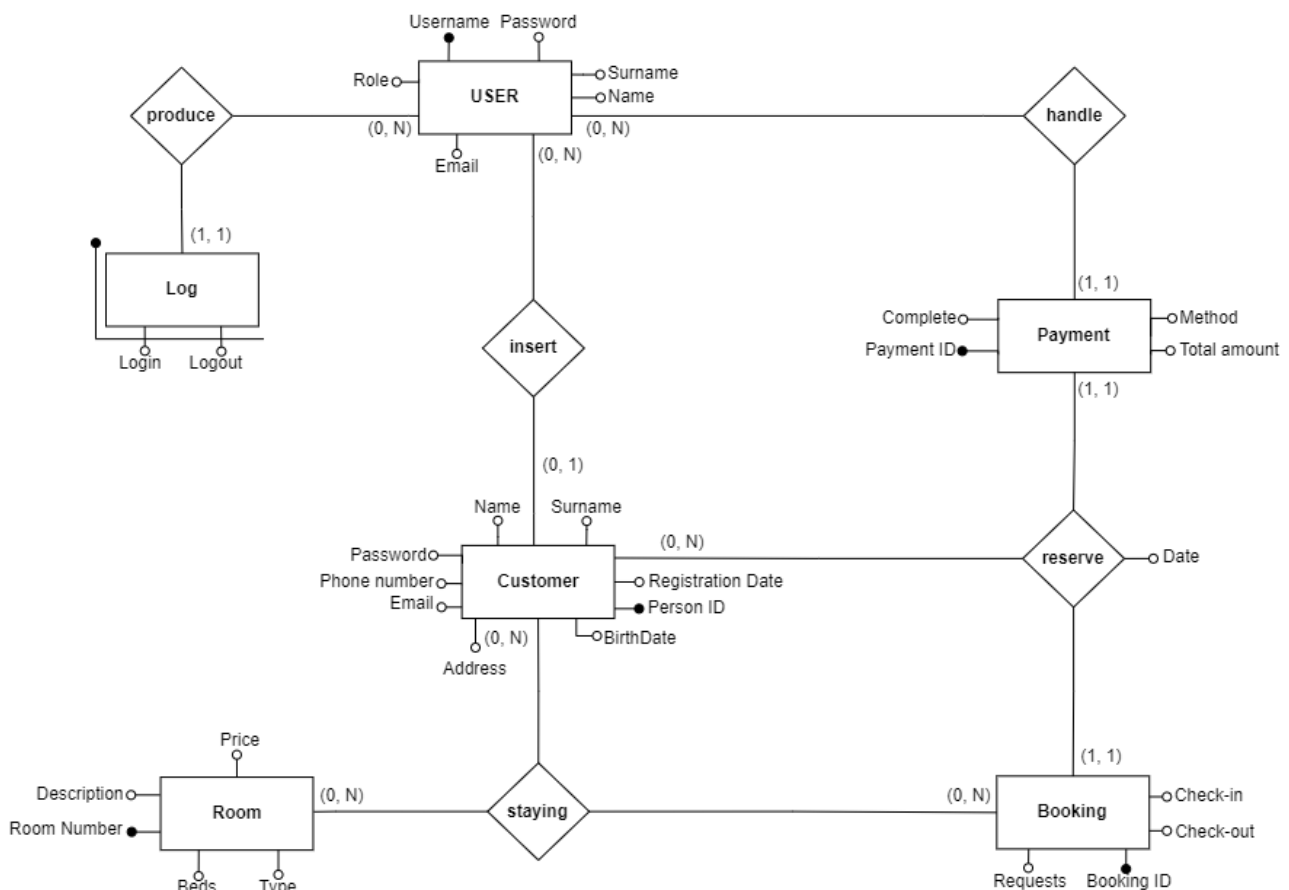
The website is divided in 5 main areas:

- Public area: this area corresponds to the homepage of the web application and it is accessible to both registered and unregistered users. In particular, in this area is possible to:
  - find basic information about the hotel and activities;
  - find contact information (email, telephone, address);
  - find available dates for booking.
- Customer area:
  - Booking:
    - show available rooms for required dates;
    - show details of the room;
    - send confirmation emails;
    - show payment methods;
    - show past bookings;
  - Profile:
    - check personal information;
    - check bookings;
    - change settings;
    - help desk;
    - choose payment methods.
- Login area:
  - registration for customers;
  - login for customers;
  - login for users.
- Administration area: the administration area is accessible only to users with the role "Hotel Manager". It contains functionalities to operate over the users in the system. In particular, in this area is possible to:
  - search users via username, email;
  - retrieve information on a specific user;

- check payments;
  - modify the user record by changing their first name, last name and role;
  - view all the information about the bookings;
  - add new users;
  - perform all the actions of the "Front Office" user;
- User area: the user area is accessible to users with the role "Front Office". It contains a series of functionalities in order to interact with customers. Some of this functionalities are the following:
    - accept payments;
    - retrieve inforcurrent bookings;
    - check the available rooms;
    - add, delete,update a booking;

### 3 Data Logic Layer

#### 3.1 Entity-Relationship Schema



The entity-relationship contains 6 main entities:

- User: contains the basic information about the users. Each user has, as primary key, their Username(CHAR). For each user we also record Surname(CHAR), Name(CHAR), email(CHAR), Role(ENUM) and password(CHAR). Note that the password is hashed through md5 before storing it.

- Customer: the main entity of the database. Contains other additional information about the user. Person ID is a primary key. Phone number (CHAR), Address(CHAR), Registration Data(DATE), BirthDate(DATE).
- Log: made by users. The Log entity represents data for logging purposes. The entity Log keeps the data of the login and the logout of each user accessing the database through the produce relationship.
- Payment: Payment ID (ENUM) is a primary key which identifies the number of the payment. For each payment we also know the Method(ENUM): Visa, MasterCard, Maestro, American Express, PayPal, Cash; Total amount(CHAR); Complete(BOOL).
- Booking: each has an ID as primary key, is associated with : the personID of the customer who made it; the checkin and checkout timestamps of the staying; the paymentID created as soon as the booking is made; the date in which the booking has been made; optional text type requests.
- Room: each is identified by its unique Room Number, which is of type text since it can contain also characters. In the entity are present also the attributes : number of beds, its price, the room type which is of category type plus a description of this type.

### 3.2 Other Information

For the creation of the database 3 custom enumerations have been created: roletype, associated to the roles of hotel users (Hotel Manager or Front Office); paymentmethod, which contains the possible ways to pay the staying (Visa, MasterCard, Cash ...); category, containing a value for each type of room (Single, Double ...). Customers are supposed to group people that make the bookings or people that stay in the hotel and also people that do both. This is the reason why some attributes can be null, for example the email is usually required for a booking but not for a staying and opposite for the document number. Log is the only entity with which it is not possible to interact since it is automatically filled.

## 4 Presentation Logic Layer

The site will be divided into two main areas, administration area and customer area, as follows:

- **Administration area:** allows the staff member (front office users and manager) to have an overview of the hotel information.
  - **Front office users:**
    - \* can add a new booking;
    - \* can see only currently booking and sooner booking, grouped by rooms;
  - **Manager:**
    - \* can add a new booking;
    - \* can see past booking (download a log), currently booking and sooner booking, grouped by rooms;
    - \* can view payment status of the booking (pending list and confirmed list) and the payments methods of customers;
    - \* can manage the front office users (create one, edit information, delete one, send reset password link);
    - \* can view the information about front office users (like the last access);
- **Customers area**
  - **Homepage:** contains general information about the hotel, pictures and short descriptions about the different types of rooms and activities. Including a link to the login/registration page;
  - **Log in page:** allows customers, manager and front office users, to login into the web application;
  - **Registration page:** allows registration of customers user;
  - **Personal profile:** allows the customer to check its personal information, manage its booking - with two different buttons, one for starting a new booking procedure, and the other one for deleting a previous booking -, and choose the payment methods;
  - **Booking page:** allows the customer to book a room for one or more guests, select the type of room and add some requests;

## 4.1 Administration Area

Front office users and manager have a common administration interface, with the difference that some links and some buttons can only be used by the manager and not by the front office users, who will not be allowed to access via a message "access denied" error.

## 4.2 Manager Pages

The first page that the manager can see, after logging in, is his homepage, where he sees an overview of the latest news, such as the number of new bookings or a list with the currently bookings and their relative details, and a list with the last accesses of the front office users.

In the booking history, the manager can navigate through an accordion list of all the rooms, and see, for each of them, some details about the bookings in that room. The manager can click on a room's name and all the details appeared: an overview of the room's booking is shown, and two tables that show the information about of currently and soon bookings. For each room, the manager can download a log file.

In the payments page, the manager check information about the payments. A first table shows a list of pending payments with relative details, and a second table shows a list of confirmed payments with relative details. For both of them a log file can be downloaded.

The users page is divided into two area. In the first one, the manager can compile a form and add a new front office user. In the second one, a table shows a list of all the front office users with relative details and relative buttons to manage them: there is a button that allows the manager to edit the details of the users (e.g., email), another button that allows to send reset password link, and in the end there is another button to delete the user from the system.

The last one page, called "add booking", shows the procedure to add a booking of a customer. In the case the customer does not book online but physically on the hotel, the manager can proceede with the booking from this interface.

**ADMINISTRATION AREA** LOG OUT

[HOME](#) [BOOKING HISTORY](#) [PAYMENTS](#) [USERS](#) ADD BOOKING

### BOOKINGS

There are n new prenotations!

< prev ROOMS next >

Room's name  
Some details

DETAILS

Booked today: n Num. ADULTS: n  
Pending Payment: n Num. CHILDREN: n

CHECK PAYMENTS

There are n booking currently

ID prenotation	Room	Check-in	Check-out
ID01	Room's name	2022-04-05	2022-04-15
ID02	Room's name	2022-04-05	2022-04-15
ID03	Room's name	2022-04-05	2022-04-15
ID04	Room's name	2022-04-05	2022-04-15
ID05	Room's name	2022-04-05	2022-04-15
ID06	Room's name	2022-04-05	2022-04-15
ID07	Room's name	2022-04-05	2022-04-15

DOWNLOAD COMPLETE LOG

### USERS LAST ACCESS

Username	Last access	Role
Username01	last access: 2022-04-05 23:36	Role
Username02	last access: 2022-04-05 22:52	Role
Username03	last access: 2022-04-05 21:17	Role
Username04	last access: 2022-04-05 21:06	Role
Username05	last access: 2022-04-05 20:43	Role
Username06	last access: 2022-04-05 20:11	Role
Username07	last access: 2022-04-05 19:33	Role
Username08	last access: 2022-04-05 18:45	Role
Username09	last access: 2022-04-05 20:43	Role
Username10	last access: 2022-04-05 20:11	Role
Username11	last access: 2022-04-05 19:33	Role
Username12	last access: 2022-04-05 18:45	Role

SHOW ALL DOWNLOAD LOG

Figure 1: Administration area: Manager homepage

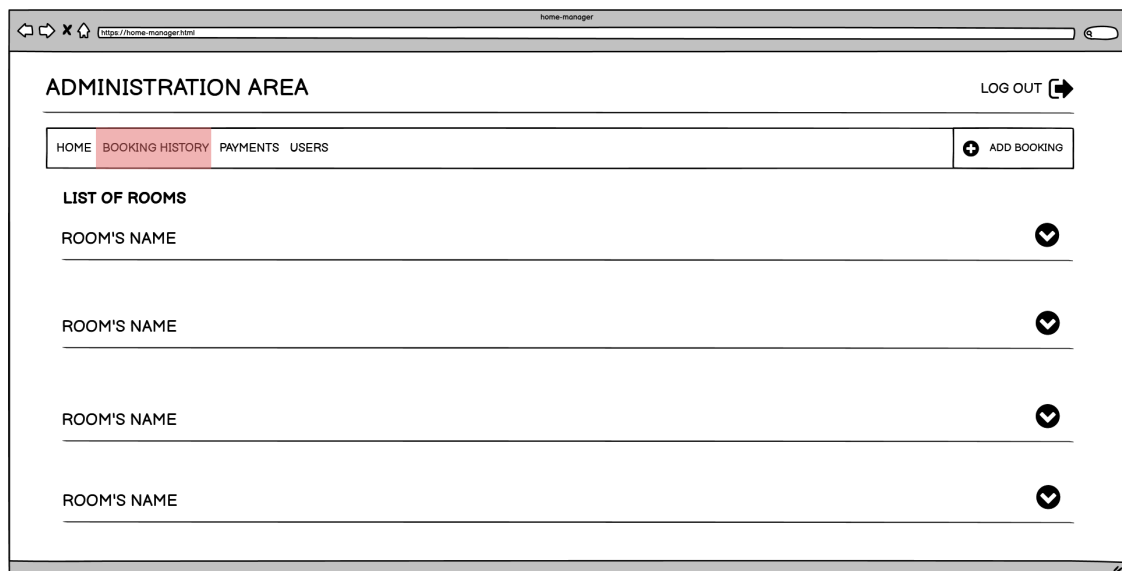


Figure 2: Administration area: Booking history page

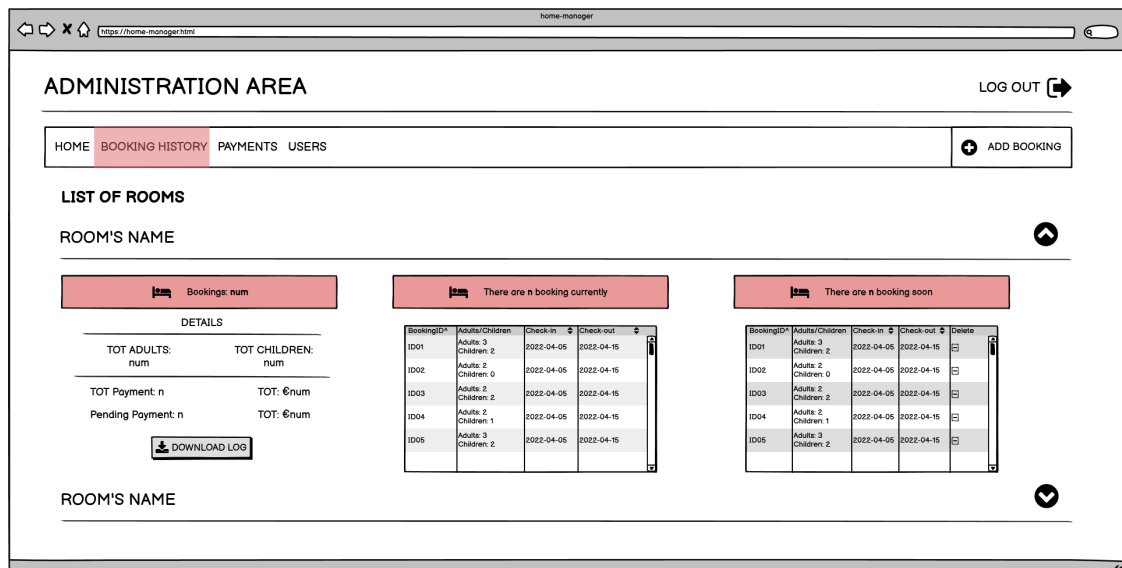


Figure 3: Booking history page: the room's details

home-manager

https://home-manager.html

ADMINISTRATION AREA

LOG OUT

HOME BOOKING HISTORY **PAYMENTS** USERS

ADD BOOKING

PENDING PAYMENTS

search

DOWNLOAD LOG

Payment ID	Booking ID	Personal ID	Method	Value	From Date
P03	Booking_ID_03	Personal_ID	Cash	€150	since 2022-04-03
P04	Booking_ID_04	Personal_ID	VISA	€150	since 2022-04-03
P05	Booking_ID_05	Personal_ID	VISA	€150	since 2022-04-03
P06	Booking_ID_06	Personal_ID	Cash	€150	since 2022-04-03
P04	Booking_ID_04	Personal_ID	VISA	€150	since 2022-04-03
P05	Booking_ID_05	Personal_ID	VISA	€150	since 2022-04-03
P06	Booking_ID_06	Personal_ID	Cash	€150	since 2022-04-03
P12	Booking_ID_12	Personal_ID	VISA	€150	since 2022-04-03
P12	Booking_ID_12	Personal_ID	VISA	€150	since 2022-04-03

CONFIRMED PAYMENTS

search

DOWNLOAD LOG

Payment ID	Booking ID	Personal ID	Method	Value	Date
P01	Booking_ID_01	Personal_ID	VISA	€150	2022-04-03
P02	Booking_ID_02	Personal_ID	Cash	€150	2022-04-06
P07	Booking_ID_07	Personal_ID	Cash	€150	2022-04-03
P07	Booking_ID_07	Personal_ID	Cash	€150	2022-04-03
P08	Booking_ID_08	Personal_ID	Cash	€150	2022-04-03
P09	Booking_ID_09	Personal_ID	Cash	€150	2022-04-03
P08	Booking_ID_08	Personal_ID	Cash	€150	2022-04-03
P09	Booking_ID_09	Personal_ID	Cash	€150	2022-04-03
P10	Booking_ID_10	Personal_ID	VISA	€150	2022-04-03

Figure 4: Manager area: Payments page

home-manager

https://home-manager.html

ADMINISTRATION AREA

LOG OUT

HOME BOOKING HISTORY **PAYMENTS** **USERS**

ADD BOOKING

REGISTRATION

Name

Surname

Email

Username

Password

Choose a role

SUBMIT

LIST OF USERS

search

DOWNLOAD LOG

User	Last access	Role	Edit info	Reset pwd	Delete
name surname email	last access: 2022-02-29 23:36	Role		(+)	
name surname email	last access: 2022-02-29 22:52	Role		(+)	
name surname email	last access: 2022-02-29 21:17	Role		(+)	
name surname email	last access: 2022-02-29 21:06	Role		(+)	
name surname email	last access: 2022-02-29 20:43	Role		(+)	
name surname email	last access: 2022-02-29 20:11	Role		(+)	
name surname email	last access: 2022-02-29 19:33	Role		(+)	
name surname email	last access: 2022-02-29 18:45	Role		(+)	
name surname email	last access: 2022-02-29 18:43	Role		(+)	
name surname email	last access: 2022-02-29 18:11	Role		(+)	
name surname email	last access: 2022-02-29 17:53	Role		(+)	
name surname email	last access: 2022-02-29 17:25	Role		(+)	

Figure 5: Manager area: Users management page



**ADMINISTRATION AREA** LOG OUT

HOME [BOOKING HISTORY](#) [PAYMENTS](#) [USERS](#) ADD BOOKING

**CHOOSE DATE**

Check in:   Check out:

Insert requests...

**SELECT ROOMS AND GUESTS**

Rooms:  Choose room:  +

Adults:

Children:

Insert requests...

**Guest information**

Name:

Surname:

Birthday:

Address:

Document number:

Phone number:

Email:

+

**SELECT PAYMENTS**

Method of Payments:

**CONFIRM**

If everything is done, click on the confirm button

CONFIRM

Figure 6: Administration area: Add bookings page

### 4.3 Front office

The administration interface of a front office user is basically the same as that of the manager, with the difference that some settings are not accessible. The most different page is the homepage. The front office user sees the same manager homepage, but he cannot see the list of other users with their last accesses and therefore his homepage is only divided into two macro areas. In the first one he can see an overview of the latest news, such as the number of new bookings with some details about that. In the second one a list with the currently bookings and their relative details is shown. Based on the pages that make up the manager interface, a front office user does not have the permissions to access to the payments page, users management page and every download log button present in all pages.

**ADMINISTRATION AREA** LOG OUT

HOME [BOOKING HISTORY](#) [PAYMENTS](#) [USERS](#) ADD BOOKING

**BOOKINGS**

! There are n new prenotational

< prev ROOMS next >

Room's name

Some details

**DETAILS**

Booked today: n Num. ADULTS: n

Pending Payment: n Num. CHILDREN: n

📅 There are n booking currently

ID prenotational	Room	Check in	Check-out
ID01	Room's name	2022-04-05	2022-04-15
ID02	Room's name	2022-04-05	2022-04-15
ID03	Room's name	2022-04-05	2022-04-15
ID04	Room's name	2022-04-05	2022-04-15
ID05	Room's name	2022-04-05	2022-04-15
ID06	Room's name	2022-04-05	2022-04-15
ID07	Room's name	2022-04-05	2022-04-15

Figure 7: Front office user homepage

## 4.4 Customers Area

The pages that make up the customer area are now described.

A guest user can do with the booking procedure, but to complete it, a registration is required.

## 4.5 Homepage

Different sections are created within the homepage to keep all the information easily available. One section is used for displaying information about the hotel itself. Another section summarises the types of rooms a potential customer can choose from, with different images and the last section contains the contact information of the hotel with a given email address and phone number. From this page it is possible to access to the page to login or register, if there is a new costumer.

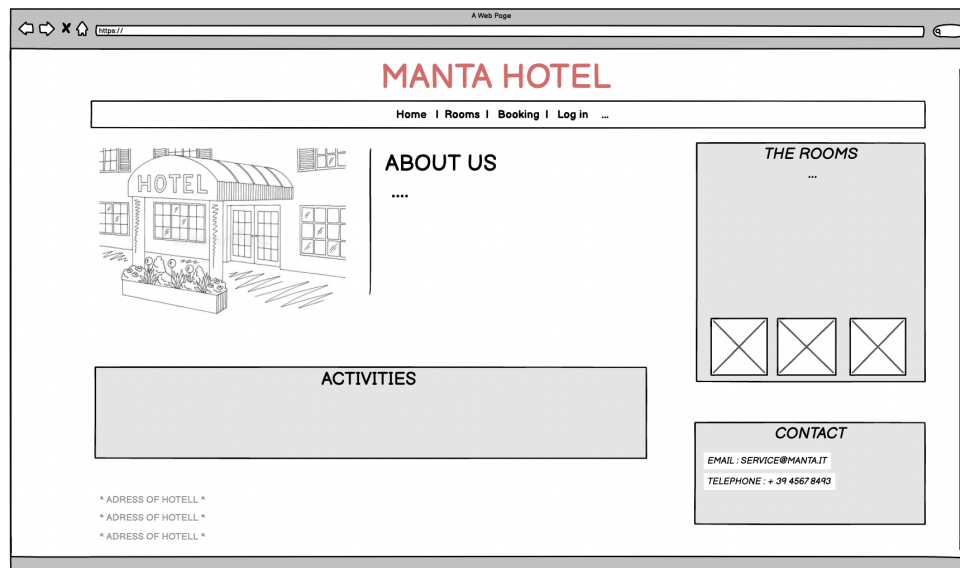


Figure 8: Customers area: Homepage

## 4.6 Booking page

The booking procedure is held through six steps divided into two pages.

In the first page there are three subsections: the first one is used to decide the date of arrival and departure with a possibility to add some requests in the specific window, the second one is used to select the number of rooms and guests, the third subsection is reserved to the selection of the room, presented as a carousel with the type and the description of each room, it is possible to use the "prev" and "next" button to navigate the carousel.

The second page displays three steps to complete the booking. The first section is dedicated to the insert guest information, some of the fields i.e. phone number and email are requested only for people over age of 18. It is possible to add different guests by using the plus button. The next section shows the summary of the booking with all the details and the last one directs to the payment.

Pages: 1

Home

Profile


Logout

Home > Bookings


<< Back to Hotel Home

Chose your date

Check in



Check out



2022

Fri, Apr 15

<

APRIL 2022

>

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

CANCEL OK



Check in time, after 2.00 p.m.

Check out time, before 10.00 a.m.



If any change in check in or check out time is needed insert your request below.

Select rooms and guests



Rooms



Adults



Children



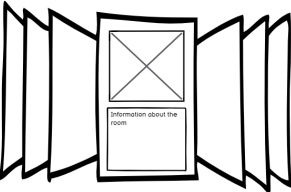
Ages (0-14)

Chose your room

< prev

AVAILABLE ROOMS

next >



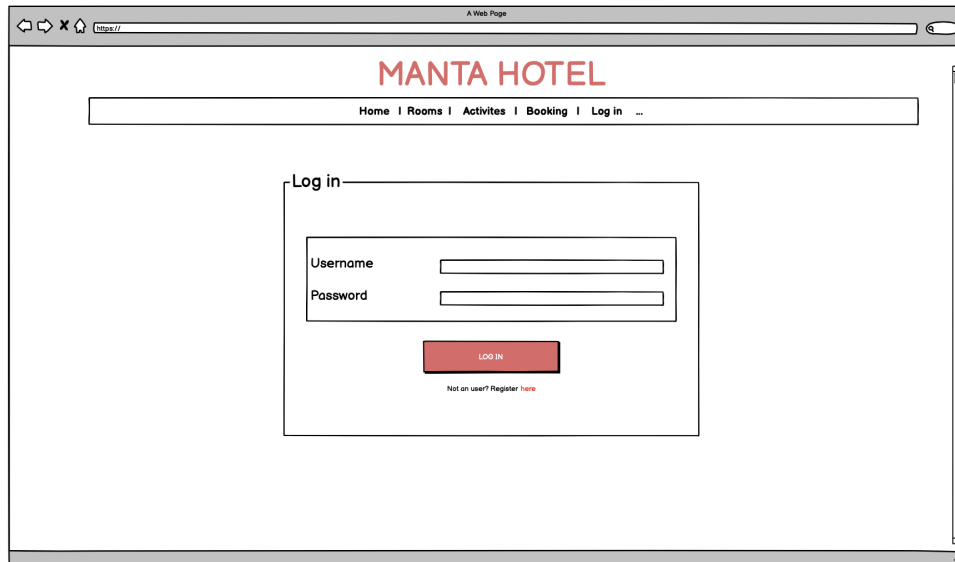
information about the room

Confirm

## 4.7 Log in and registration page

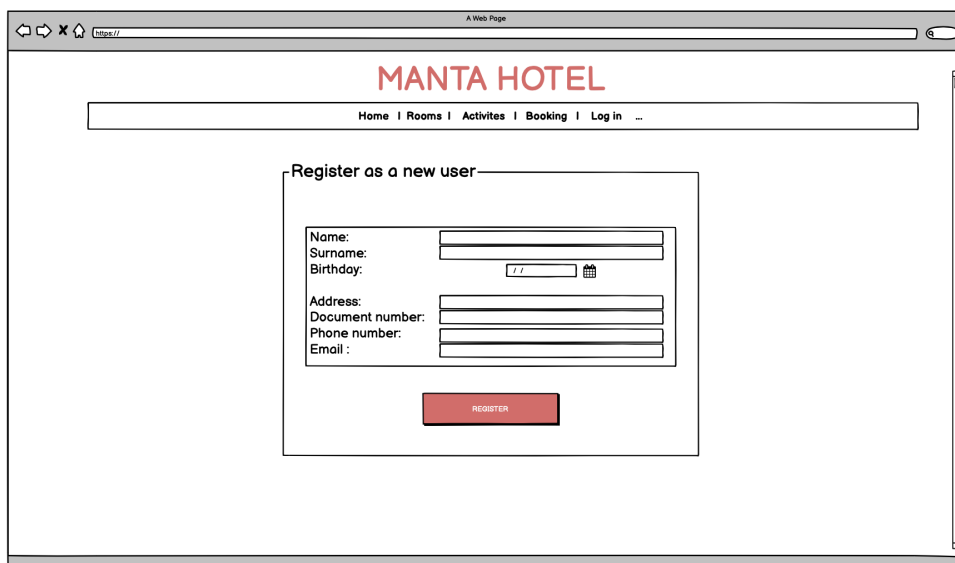
The log in page allows both customers and users (staff members in the reception) to log in and use the Manta Hotel web application, with only their username and password.

If one is not a user, one can register by clicking the registration link. This will send the soon to be user to a registration form where one has to fill out all the necessary information in order to register.



A screenshot of a web browser displaying the Manta Hotel login page. The browser's address bar shows "https://". The page has a header with the "MANTA HOTEL" logo in red and a navigation menu with links: "Home", "Rooms", "Activites", "Booking", "Log in", and "...". The main content area is titled "Log in" and contains a form with two input fields: "Username" and "Password". Below these fields is a red "LOG IN" button. At the bottom of the form, there is a link that says "Not an user? Register [here](#)".

Figure 11: Customers area: Login page



A screenshot of a web browser displaying the Manta Hotel registration page. The browser's address bar shows "https://". The page has a header with the "MANTA HOTEL" logo in red and a navigation menu with links: "Home", "Rooms", "Activites", "Booking", "Log in", and "...". The main content area is titled "Register as a new user" and contains a form with several input fields: "Name:", "Surname:", "Birthday:" (with a calendar icon), "Address:", "Document number:", "Phone number:", and "Email :". Below these fields is a red "REGISTER" button.

Figure 12: Customers area: Registration page

## 4.8 Personal page

In the personal area page the customer can navigate through personal information, past and current bookings (check symbol button), it is possible to book a new room (+ button) and chose different payment methods. The bookings can be sorted by date. There are two buttons to check the past bookings and start a new one. It is also possible to open a tend from the profile button in the right top corner in order to check the settings, help section or log out.

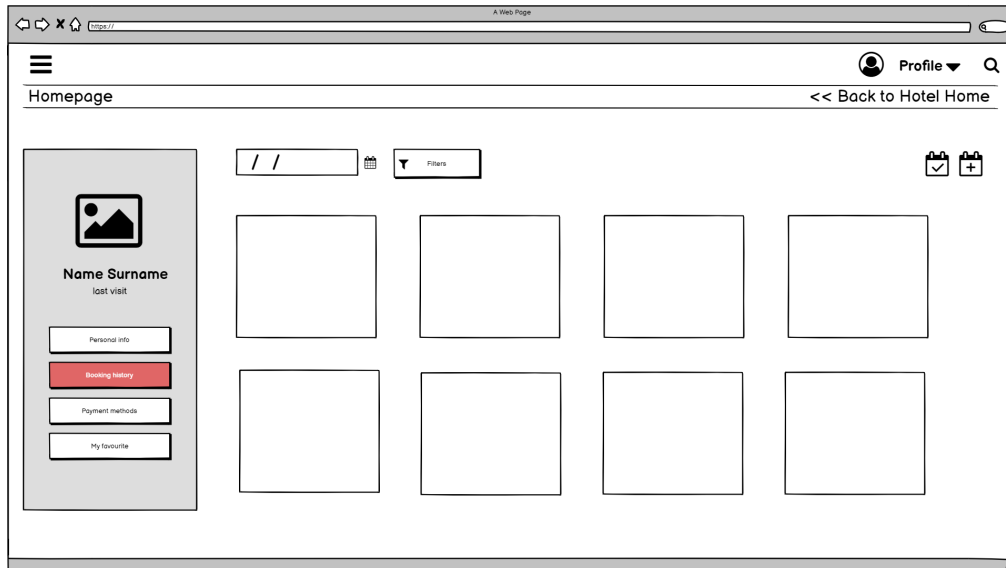


Figure 13: Customers page

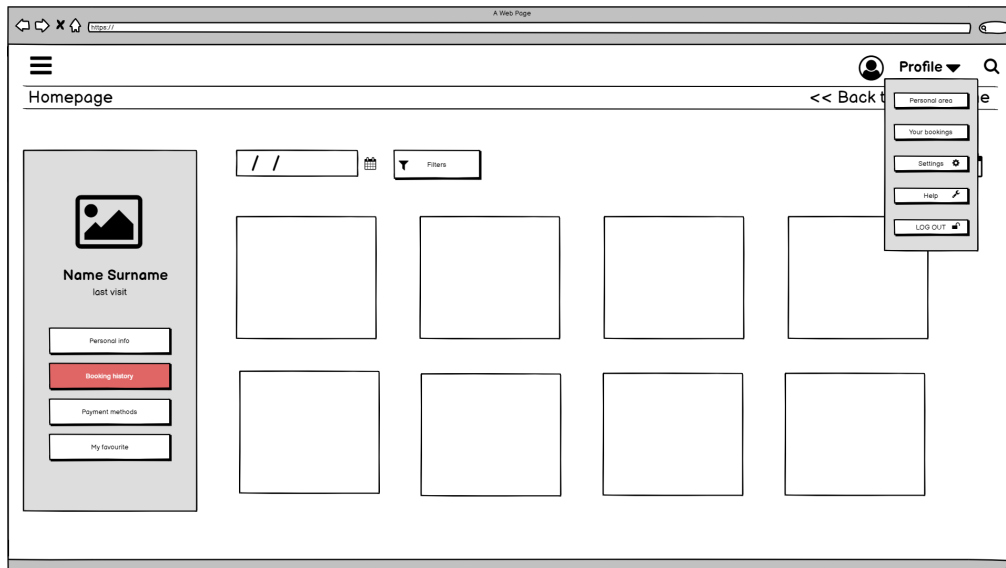
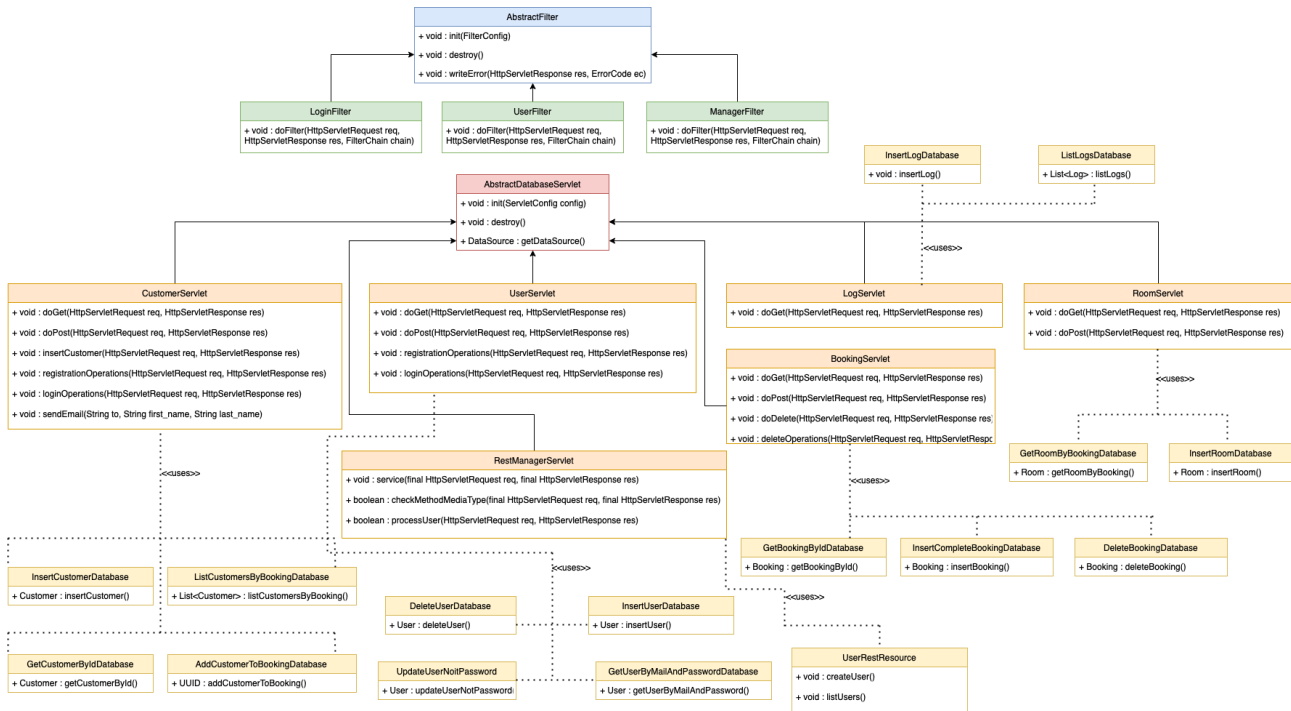


Figure 14: Customers page: tend menu

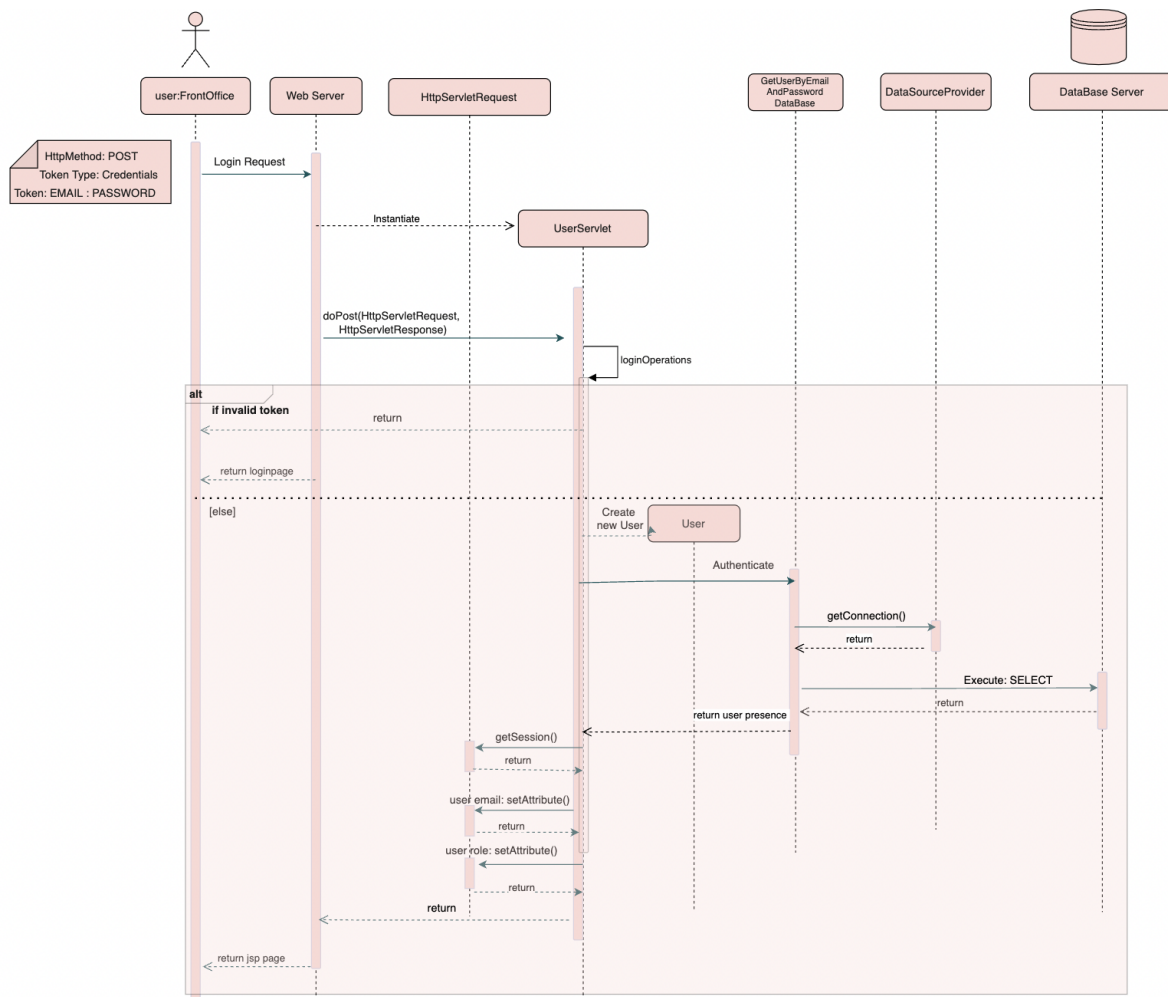
## 5 Business Logic Layer

### 5.1 Class Diagram



The class diagram contains most of the classes used to handle the six different resources; users, customers, bookings, rooms, logging, and manager. All of the resources have their own servlet class which is extended by the **AbstractServlet**. The **AbstractServlet** is a subclass of **HttpServlet** which offers different HTTP requests; **doGet**, **doPost**, **doPut** and **doDelete**. Whereas the first five resources mentioned have the traditional Java Servlet methods using the Http design, the Manager servlet is based on a traditional REST approach. This servlet mainly parses the URI and determines the type/id the manager wants to interact with. As the request has been processed, it forwards it to the **UserRestResource** class, which implements the methods to properly handle the resources. All of the servlets uses several Data Access Objects (DAO) which allows us to obtain different resources.

## 5.2 Sequence Diagram



The sequence diagram illustrates the login operations for the front office user in the application. A Login Request is sent from the user to the web server in by the HTTPMethod POST and simultaneously passing the applied email and password credentials. Further on, the web server instantiates the UserServlet and calls the doPost Method which passes the HttpServletRequest and HttpServletResponse. After recognizing the request as an attempt to login the UserServlet calls the method loginOperations to verify the given credentials. If credentials are invalid or missing, it returns the login jsp-file with the attached error message, and else a new user object is initialized with its data. The new user object data needs to be authenticated and is being passed on as an argument to the method responsible. The method in the DAO-class (GetUserByMailAndPasswordDatabase) requests a connection to the DataSourceProvider. This makes it possible for the DAO-class to retrieve information about whether the given user is current in the database or not from the database server. The information is then being returned to the UserServlet. After retrieving the session with the users email and role as parameters from HttpServletRequest, the user is being forwarded to the instantiated homepage.jsp with the user's data.

## 5.3 REST API Summary

In this first part of the project we implemented the rest related to booking get methods and user get and insert methods in the traditional way, with the information to be inserted in the URI. We then implemented the other endpoints with parameters to be inserted in the request (this was made also for booking and user). Part of the

URIs are filtered through different filters. M indicates that only users with the role of manager can access to the endpoint, U indicates that only users (manager and front office members) can access the endpoint.

URI	Method	Description	Filter
booking/customer-list/	GET	Returns the list of bookings made by the customer whom personId is retrieved from the Http session	
booking/insert/info/	POST	Allows to see all the available rooms in the requested time period and number of beds, and it passes the parameters needed for inserting the booking and its associated payment	
booking/insert/complete/	POST	Allows to insert, in order, a payment, a booking and a staying relation between the booking, the costumer who made it and the room selected	
booking/add-customer/	POST	Allows to add in the database the relationship between a booking and a guest (which is a costumer who didn't make the actual booking)	U
booking/search/	GET	Returns the booking which id is passed as data	U
booking/list/	GET	Returns the list of bookings made by a customer whom personId is passed as data	U
booking/list/date/	GET	Returns the list of bookings with checkin or checkout dates equal to date passed as data	U
user/logout/	GET	Allows to logout a user and at the same time make an insert on Log table	U
user/userList/	GET	Returns a list of all the hotel users	U
user/login/	POST	Allows to pass to the web server the credentials of a user and checks if the user is in the database and their mail and password correspond	U
user/register/	POST	Allows to register a new user	M
user/update/	POST	Allows to modify personal user parameters passed as data	U
customer/logout/	GET	Allows to logout a customer	
customer/login/	POST	Allows to pass to the web server the credentials of a user and checks if the user is in the database and their mail and password correspond	
customer/register/	POST	Allows to register a new customer	
logs/user/	GET	Returns the login and logout timestamps related to the user and date which are passed as data	M



logs/day/	GET	Returns all the login and logout times-tamps of all the users, made during the current day	M
payment/search/	GET	Returns the payment related to a booking which id is passed as data	U
payment/list/	GET	Returns the list of payments associated to a customer whom id is passed as data	U
payment/edit/info/	POST	Allows to see the information about a payment which id is passed as data, and it requests the parameters which have to be updated	U
payment/edit/	POST	Allows to update a payment with the parameters passed as data	U
rooms/	GET	Returns the list of all rooms	
room/insert/	POST	Allows to insert a new room with the parameters passed as data	U
rest/booking/bookingId	GET	Allows to get information on a booking	
rest/booking/customer/personId	GET	Allows to get the list of all bookings associated to the customer personId	
rest/booking/date/date	GET	Allows to get the list of all bookings with checkin or checkout dates equal to the date date	
rest/user/	GET	Returns the list of all hotel users	
rest/user/	POST	Allows to insert a new user	

## 5.4 REST Error Codes

The following list contains the different possible errors which can occur by using the application. In particular, application specific errors have the application error which follows a progressive numeration starting from -100. METHOD NOT ALLOWED errors are identified with the error code -500. Internal errors, which correspond to crashes, servlet exceptions, or problems with the input/output streams are identified with the Error Code -999.

Error Code	HTTP Status Code	Description
-100	BAD_REQUEST	Wrong Person id format.
-101	NOT_FOUND	The user is looking for a room which is not present.
-102	BAD_REQUEST	The user tried to register without specifying all the details. Used also with forms that need to be fully filled.
-103	BAD_REQUEST	The user tried to login without specifying the Email.
-104	BAD_REQUEST	The user tried to login without specifying the password.
-105	BAD_REQUEST	User submitted wrong credentials in the login form.
-106	NOT_FOUND	User to be deleted not found.
-107	BAD_REQUEST	Different passwords when repeating the password in the registration phase.
-108	CONFLICT	Mail already used for another registration.
-109	BAD_REQUEST	Unrecognized role.
-110	CONFLICT	User is already present.
-111	BAD_REQUEST	Invalid document number.

-112	BAD_REQUEST	Wrong rest request format.
-113	NOT_FOUND	Costumer not found.
-114	CONFLICT	Person ID requested and Person ID provided do not match.
-115	NOT_FOUND	User not found.
-116	BAD_REQUEST	Person ID is missing.
-117	NOT_FOUND	Booking not found.
-118	BAD_REQUEST	Date is missing.
-119	BAD_REQUEST	The input json is in the wrong format.
-200	BAD_REQUEST	Operation unknown.
-500	METHOD_NOT_ALLOWED	The method is not allowed.
-999	INTERNAL_SERVER_ERROR	There was an internal error that the server was not able to manage.

Table 3: Describe in this table your REST API

## 5.5 REST API Details

### User Resource

This endpoint allows to insert a User or get list of all users, in this case we present only the inserting operation.

- URL: `http://localhost:8080/MantaWA-HW1-1.0/rest/user/`
- Method: POST
- URL Parameters: None
- Data Parameters:
  - firstname: text
  - lastname: text
  - email: text
  - role: text
  - password: text
- Success Response:
  - { "message": { "message": "Success", "code": "200", "details": "User inserted successfully" } }
- Error Response:
  - { { "message": { "message": "Duplicate User", "code": "110", "details": "User already exist" } } }

## 6 Group Members Contribution

**Heldal Julie Marie** worked on homework report, in particular on the mockup for the homepage, PLL text file for homepage and sequence diagram BLL section.

**Hermansen Kathrine** worked on the homework report by making mockups for the login, registration and front office page, PLL text files (login and registration, front office) and the Class Diagram BLL section.

**Kosiakov Arina** worked on the homework report, in particular on the mockups for booking and personal page, PLL text files (Booking page, Personal page, Main Fncionalities), BLL section some contributions to the coding Log part

**Lotta Tommaso** worked on the coding part related to Booking (servlet, resource, dao and rest), Payment, Log and other general files needed for the coding part in java. Worked on the database sql code. Wrote the rest api summary and details.

**Manuilova Nadezhda** worked on the homework report: Objectives of the system, Main functionalities, Data logic layer (ER-Schema).

**Mohammadi Mohammadreza** Room Entity servlet, resource - General setting of POM, web.xml - User Rest API - Rest API report - REST API general setting - REST API message class

**Roennestad Andreas** worked on the implementation of the Customer-entity related part of the backend providing functionality for registering, updating, deleting Customers etc. This included the Customer servlet, registration jsp, and the related database operation classes.

**Vignaga Marco** User, Customer and Payments related jsp, dao and servlets.

**Zuccolo Giada** Mockups and report.