Guest Management System

SWE599 Project Report

Instructor

Prof. Dr. Taflan Gundem

Tugcan Olgun

2016719210

Intro

This is a ‘Guest management system’ project for the class SWE599 in Bogazici University. It is under supervision of Prof. Dr. Taflan Gundem. The aim of the project is to implement a ‘Guest management system’ also referred as a ‘Hotel management system’ which allows managing room creations, bookings and also supervising all the services connected with the proper management of a hotel. For simplified and easier usage I have created a website for all the management and functional processes as well as an app which would be used by room service staff to control all the room services such as cleaning, supplying, or even ordering food.

There are hundreds of hotel booking systems available. I wanted my system to support running a complicated business such a hotel. What I wanted to aim is to offer other innovative features which would distinguish my system from the existing one without losing the functionality of the hotel business.

In order to achieve the above I had to research what kind of procedures hotels implement and on the basis of that I create a system which would make all of those efficient. My hotel system has the capacity of handling unlimited number of rooms, provided the database is sufficient for the amount.

I called this project YAHOMAS. It stands for ‘Yet another hotel management system’. The web component can be seen at https://github.com/tugcanolgun/YAHOMAS the mobile component can be seen at https://github.com/tugcanolgun/YAHOMAS.App and the live website can be visited at https://yahomas.tugcan.net

Motivation

Hotel management systems are complex. There are many little details that needs to be taken cared of. If there are no guests assigned to a booking, that booking cannot be tracked to the owner, it needs to have at least one guests. But, a booking may have more than one guests assigned to it. As the number is basically `n`, the tracking needs to be elegant so the booking can be processed correctly. In a project such as hotel management system, these back-end details are hard to be shown in front-end to the end-user.

As front-end part is very important for such a project, I wanted to learn how to write such a user experience as I lack the information given that I only wrote back-end related things in my professional software career.

Requirements

Functional Requirements

1. The system should allow to create invoice for a particular guest or collection of guests
2. The system should allow to add room service details such as foods, beverages
3. The system should allow to add discount on rooms based on their initial costs
4. The system should allow to book/check availability of a room
5. The system should allow photo upload for the rooms
6. The system should allow three types of users with different privileges
7. The system should allow managers to add/edit/delete rooms in a hotel
8. The system should allow managers to add/delete users in a hotel
9. The system should provide a REST API for mobile interaction
10. The system should allow room service personnel to send room cleaning, item entry and removal requests as well as NFC and QR room unlocking options.

Non-functional Requirements

1. The system shall be designed in a way to fulfill the further requirements supporting other platforms like mobile phones.
2. The system shall be scalable, resource-efficient and high-available.
3. The system should support the modern web-browser engines and be accessible and easily usable from the variety of devices like mobile phone, tablets, and PCs.
4. The system shall be runnable on the various operating systems such as Windows and
5. Linux, in particular on the Linux-derivative operating systems.
6. The system shall be fast enough and has a high performance. System response time for the request will be at most 3 seconds.
7. The HTML codes of the website shall be compliant with the W3C standards.

System & Design

This chapter briefly describes the main System & Design implementations of the system. As there are two three different components for the system and their respective designs, this chapter will be divided into three; the web, the app and the rest api.

The Web Component

Dependencies of the Software

The web component is mainly written with python. There are also javascript and css elements along with html. I tried to avoid using ready libraries as much as possible as the main purpose of this project for personal gain was to learn front-end capabilities of the web. The system is built with MVC pattern using Django and connected to one database. The database is created by code-first approach.

For the template building part, Django template system is used and the reusable components are combined in base.html. The whole website is built with mobile-first method. The sidebar can be toggled by clicking on the top button.

The system used is 16.04 ubuntu server image. Using gunicorn for reverse proxt, nginx is set to use the site with HTTPS protocol (certbot). The live site can be reached at https://yahomas.tugcan.net

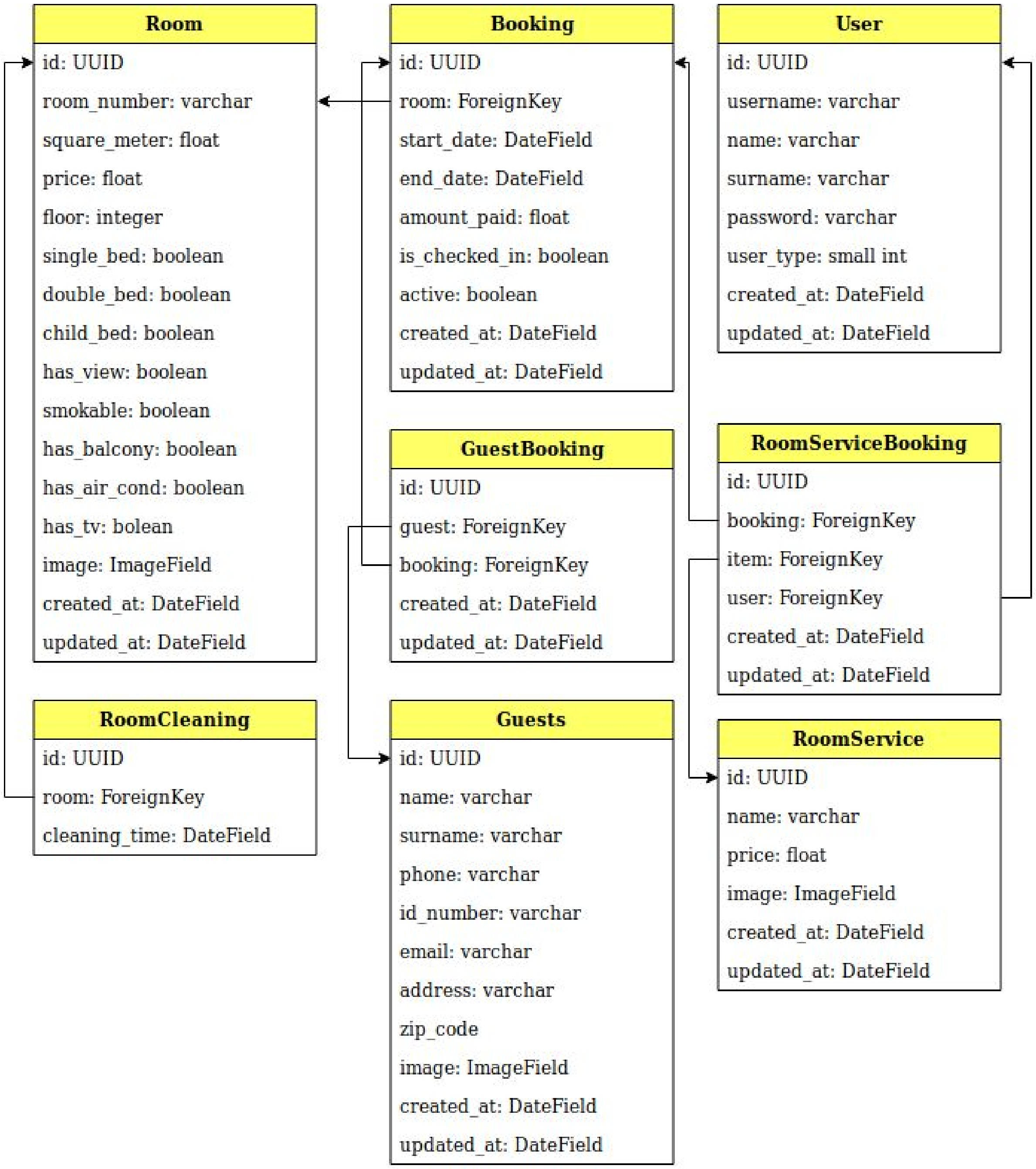
Build instructions for this system and its requirements are explained in the Github page of this project. Here is a brief overview,

- python 3.6+ (Type annotations are used with `mypy`)
- django (For the main MVC pattern)
- djangorestframework (For the REST API)
- django-rest-auth (For the REST API security)
- django-auto-complete-light (For jquery guest search)
- django-crispy-forms (For bootstrap template of django forms)
- django-secure (For securely create and store Django secret key and database information)
- gunicorn (To run wsgi on systemd, on linux)
- InvoiceGenerator (For creation of invoices)
- psycopg2-binary (For Postgresql connection)
- pycryptodome (For encryption of important information)
- qrcode (For creation of bank details in QR)

These dependencies are and their counterparts can be found in `requirements.txt` file in Github.

Database Schema

The following page contains database diagram. The project uses PostgreSQL database.



Room
id: UUID
room_number: varchar
square_meter: float
price: float
floor: integer
single_bed: boolean
double_bed: boolean
child_bed: boolean
has_view: boolean
smokable: boolean
has_balcony: boolean
has_air_cond: boolean
has_tv: bolean
image: ImageField
created_at: DateField
updated_at: DateField

Booking
id: UUID
room: ForeignKey
start_date: DateField
end_date: DateField
amount_paid: float
is_checked_in: boolean
active: boolean
created_at: DateField
updated_at: DateField

User
id: UUID
username: varchar
name: varchar
surname: varchar
password: varchar
user_type: small int
created_at: DateField
updated_at: DateField

GuestBooking
id: UUID
guest: ForeignKey
booking: ForeignKey
created_at: DateField
updated_at: DateField

RoomServiceBooking
id: UUID
booking: ForeignKey
item: ForeignKey
user: ForeignKey
created_at: DateField
updated_at: DateField

RoomCleaning
id: UUID
room: ForeignKey
cleaning_time: DateField

Guests
id: UUID
name: varchar
surname: varchar
phone: varchar
id_number: varchar
email: varchar
address: varchar
zip_code
image: ImageField
created_at: DateField
updated_at: DateField

RoomService
id: UUID
name: varchar
price: float
image: ImageField
created_at: DateField
updated_at: DateField

System Main Functions

Brief product functions are as follow:

- Make reservations
- Search available rooms with criteria
- Search bookings
- Add payment
- Issue bills
- Manage guests
- Manage room retails
- Set rates
- Manage users
- Add room items
- Manage room service details

Rooms

Room creation is a very important topic. There are details that can be very important for the customers to care about. I have research numerous room booking sites to the features that are presented there. The booking agent has to be able to see these details so the customer's questions and expectations are met.

While creating a room, following fields can be entered: room name, price, square meter, floor, image, single bed, double bed, child bed, has view, smokable, has balcony, has air cond, has tv. The only constraint on this creation for is that the room name has to be unique. There cannot be two room number 101. These details can later be tweaked.

A room's detail page consist of three tabs; Bookings, Information and Update.

Bookings tab shows the current and future bookings of the room. The details of the bookings can be reached by clicking on the booking.

Information tab shows the details of the room as well as its cleaning logs.

Update tab is a form to update any detail about the room.

Users

Users can be created and managed in `Users` link on the sidebar. There are three functionalities in this page. Users can be created with the following fields: username, first name, last name, user type, password, password confirmation. Passwords has to be at least 8 characters long and cannot be all numerals. The passwords of a user is stored with encryption so in case of a database breach, passwords cannot be fetched.

Room Service

Room service link on the sidebar shows the room related services. These can be related to food, beverages and as well as other entities such as massage, spa, pool services and etc. Upon clicking `Add new item` button, the following fields are shown: name, price, image. These fields are mandatory. The items added here can later be added to any booking that is currently active in the hotel via the app. When acquiring and printing the invoice, these items are listed.

Guests

Guest addition and selection screens can be seen in three places. First is on ‘Guests’ link on the sidebar where user creation with following fields; name, surname, phone, image, id number, email, address, zip code is possible. Also user deletion and update can be achieved via this page. Guest addition and autocomplete search is possible on booking add and booking details pages. Autocomplete searches the guests as the user starts typing. If the guest has their phone field filled, the results are shown with “name surname - phone number” schema so the guests with the same name and surname can be distinguished.

Booking

Bookings can be listed on ‘Booking’ link on the sidebar. Bookings can be deleted and updated on this page. Upon clicking on any booking the following tabs with following features are listed,

Details tab shows the booking and guest information. Adding/removing guests are possible in this tab.

Check out tab is used for listing the final invoice in both listing and creating pdf invoice. In order for this tab to be active, the booking has to be marked as checked in on Update tab. If the status is ‘checked in’, discount input field becomes active. If there is any discount on the final invoice, the receptionist or the manager can enter it here. The invoice consist the amount of days the user has spent in the hotel as well as the room service items. The discount does not affect the item prices, only the total price of the room.

Update tab is used to update on the booking. Changing rooms, dates, amount paid and checked in status is possible in this tab.

Booking search

The rooms can be searched via the calendar select on the index page. Upon clicking a calendar is presented. Beginning and an end date can be selected here along with single bed, double bed and child bed selections. If there are any available rooms, the following page shows the results with some details that may be want to present the customer with.

The REST API Component

The rest api is created with ‘djangorestframework’ library using MVC pattern with ‘django’. The endpoints are basically csrf disabled endpoints for CORS (Cors-origin resource sharing) availability.

The following endpoints are used in this component.

Items

The room service related items can be fetched, updated and deleted.

Rooms

Every room that is in the hotel and related informations can be fetched through this endpoint.

Bookings

The current day’s active bookings can be fetched along with its necessary information.

Room cleaning

If a room is not cleaned, a call to this endpoint can be made after cleaning the room.

The App Component

Dependencies of the Software

The App is written in React-Native, a framework produced by Facebook for writing mobile applications for both iOS and Android in javascript. All of the components that are used while writing this application are custom components that can be reused anywhere. Upon successful user login the credentials are saved in AsyncStorage that is provided in both iOS and Android platforms. These are not accessible outside of the parent App. Upon every relaunch of the app, the user’s active status is checked. If the user is deleted or being disabled by a manager, the user cannot use the further functionality of the app.

The following dependencies are used for creating this application:

- react-native (0.57.8)
- react-native-popup-dialog (For pop up dialogs in the app)
- react-native-router-flux (For easy navigation)
- react-native-vector-icons (For icons (community))
- react-native-qrcode (For producing QR codes)

Login screen

The login screen does not provide for user creation or forgot password functionalities as these are managed by the hotel manager. The login screen only provides means to log in to the app.

Main screen

Main screen is used for listing every room in the hotel with following functionalities.

The rooms that are booked and actively contain guests are shown as Booked and the empty rooms are tagged as Empty. There are four buttons for every room in this card view.

Room service button is only active if the room has guests inside. Via clicking this button the user can navigate to items screen and add/delete items.

Broom icon is only active if the room hasn’t been clean current day. If the button is active, the user can click the broom icon and upon clicking a dialog window will be shown asking if the room is cleaned. Clicking yes will send the request to the server, disabling the icon on the app and showing in the room details page on web component.

NFC and QR icons are experimental buttons for creating easy access to the room. Given that this is a generalized system, there is no used. But in a real hotel, these settings can be used for room service personnel to access the rooms via either NFC or QR functionalities resulting logging for individual room service personnel behavior to be tracked.

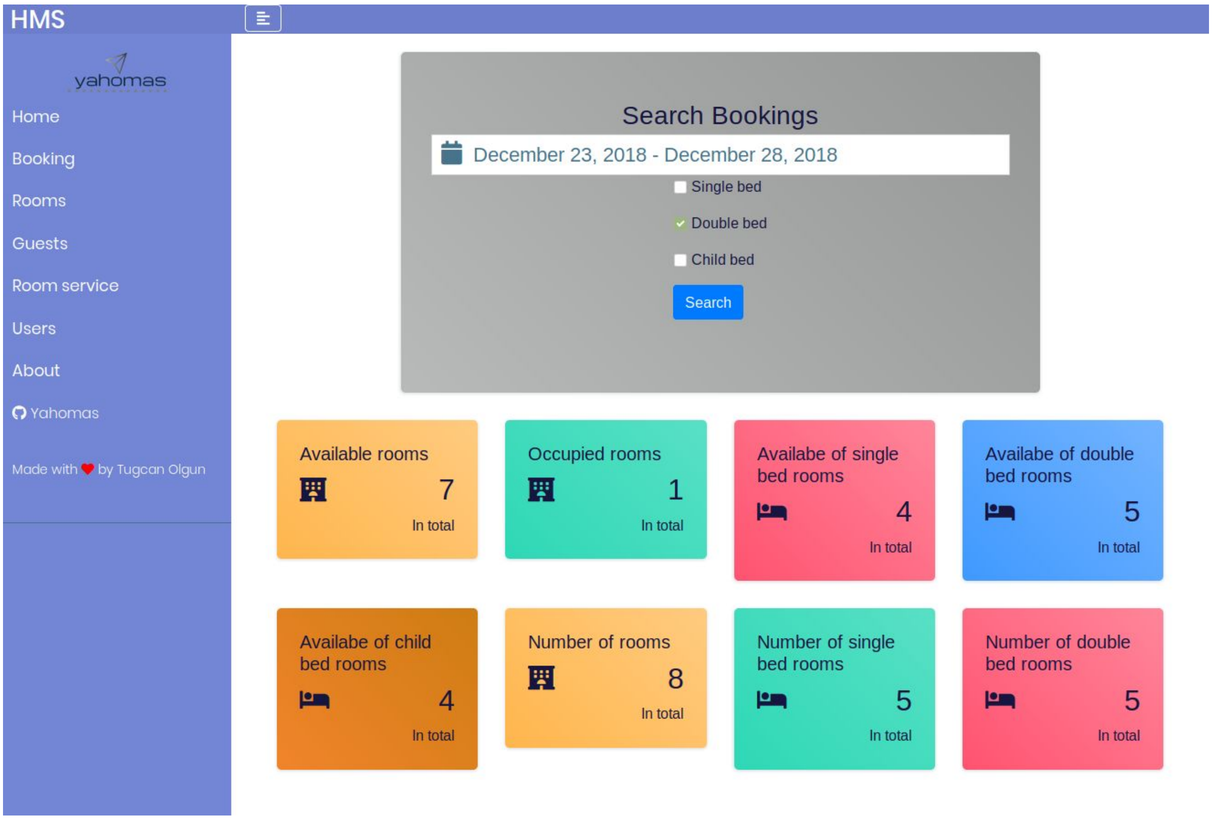
Item Screen

Item screen shows the available room service items that are created in web component. The room service employee can see how many of each item is already associated with this room and can add/delete items.

Demonstration

This chapter gives a very basic overview of the project. The live site can be seen at https://yahomas.tugcan.net This chapter includes screenshots of several pages:

Main page



Search Bookings

December 23, 2018 - December 28, 2018

□ Single bed

☑ Double bed

□ Child bed

Available rooms 7 In total

Occupied rooms 1 In total

Availabe of single bed rooms 4 In total

Availabe of double bed rooms 5 In total

Availabe of child bed rooms 4 In total

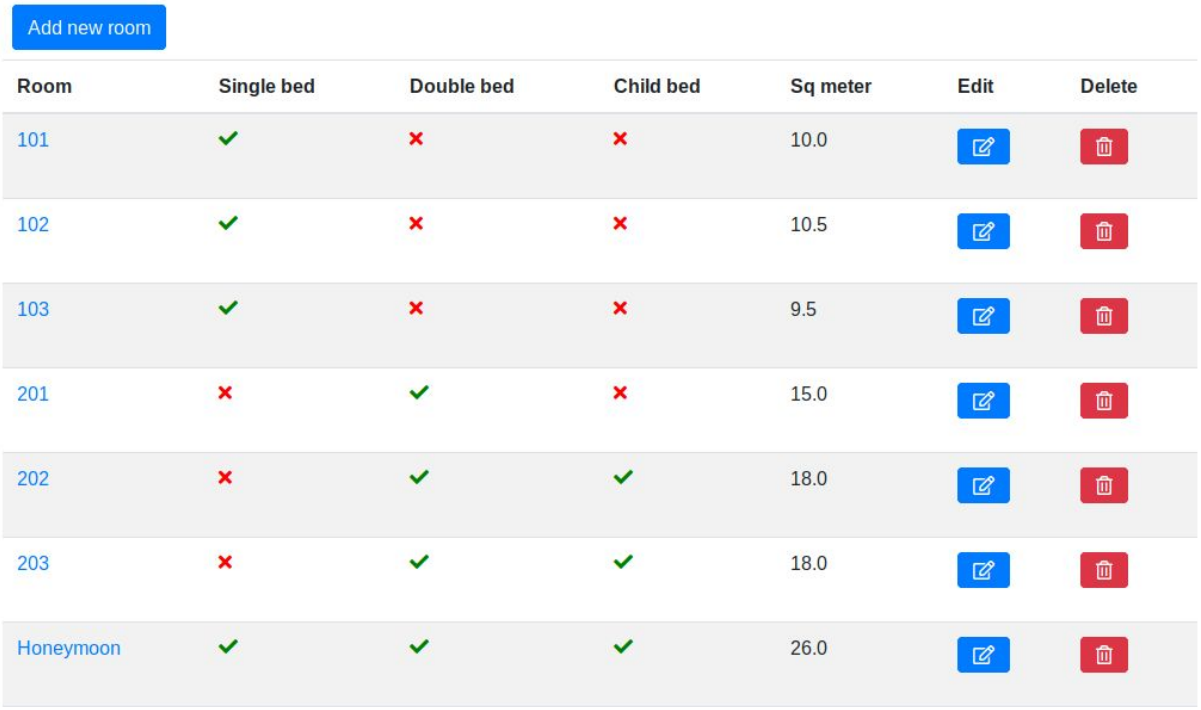
Number of rooms 8 In total

Number of single bed rooms 5 In total

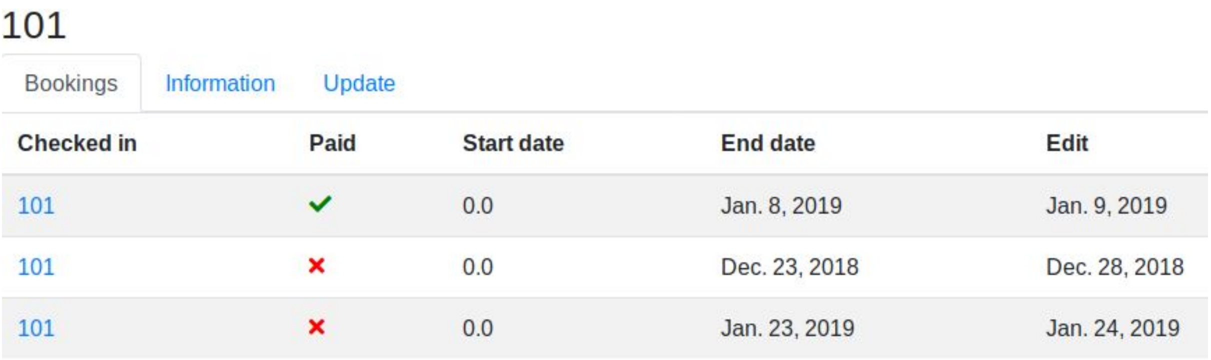
Number of double bed rooms 5 In total

The following screens will not contain the side bar for better readability.

Rooms

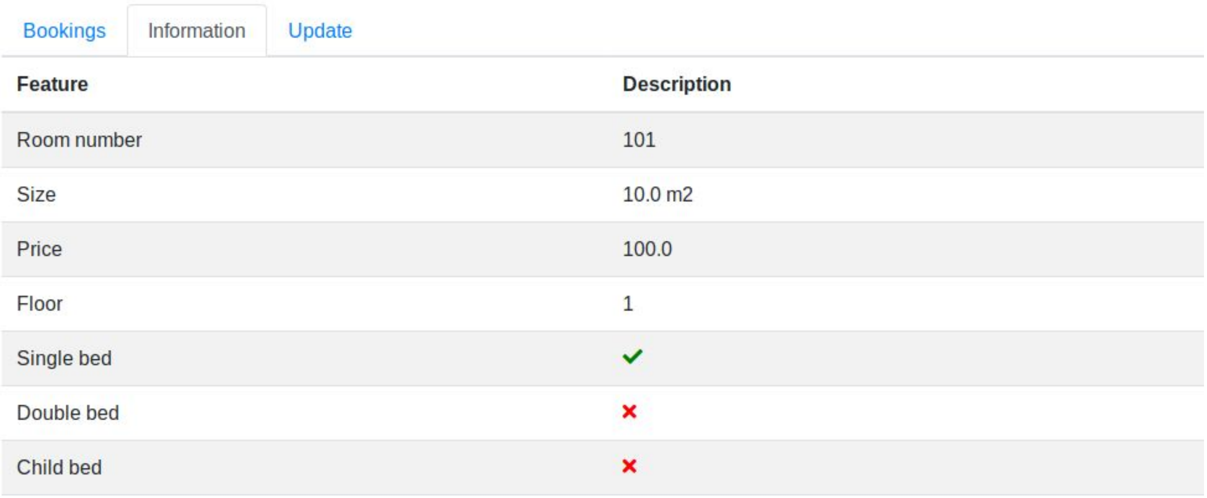


Room	Single bed	Double bed	Child bed	Sq meter	Edit	Delete
101	✓	✗	✗	10.0	✎	🗑
102	✓	✗	✗	10.5	✎	🗑
103	✓	✗	✗	9.5	✎	🗑
201	✗	✓	✗	15.0	✎	🗑
202	✗	✓	✓	18.0	✎	🗑
203	✗	✓	✓	18.0	✎	🗑
Honeymoon	✓	✓	✓	26.0	✎	🗑

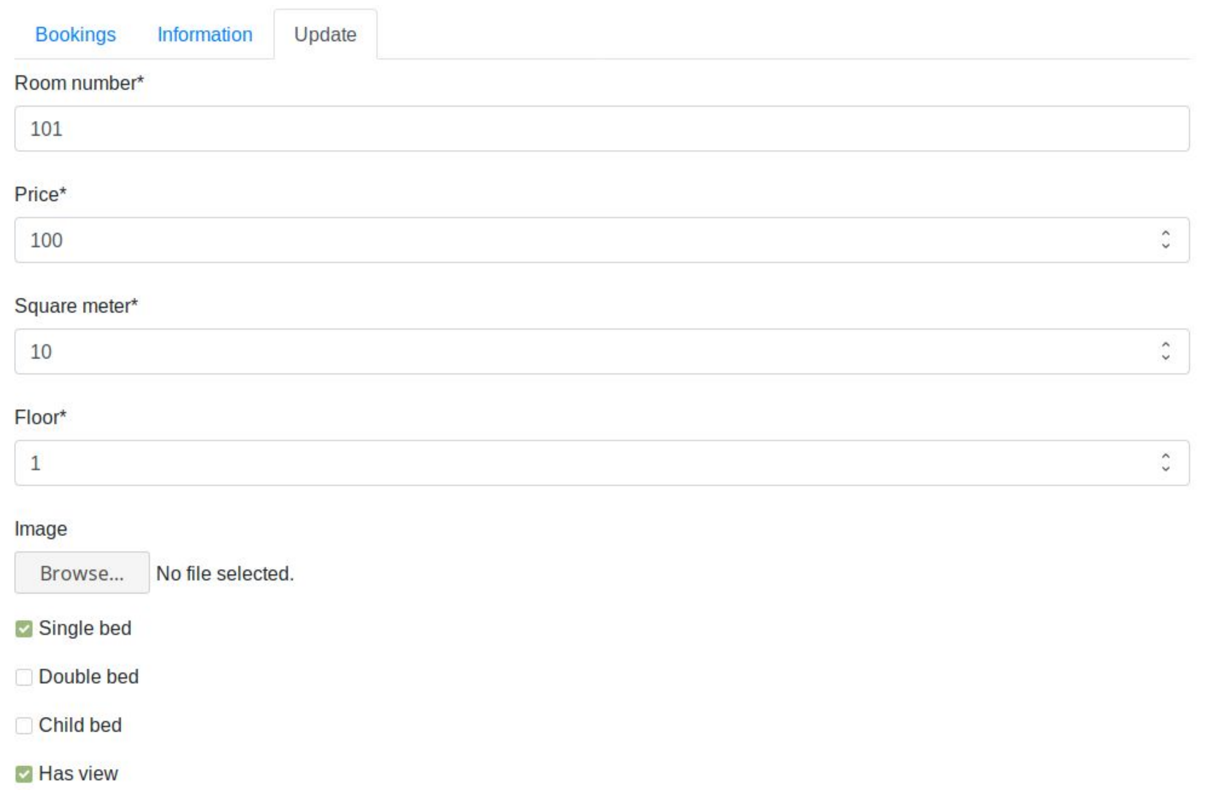


101

Checked in	Paid	Start date	End date	Edit
101	✓	0.0	Jan. 8, 2019	Jan. 9, 2019
101	✗	0.0	Dec. 23, 2018	Dec. 28, 2018
101	✗	0.0	Jan. 23, 2019	Jan. 24, 2019



Feature	Description
Room number	101
Size	10.0 m2
Price	100.0
Floor	1
Single bed	✓
Double bed	✗
Child bed	✗



Room number*

101

Price*

100

Square meter*

10

Floor*

1

Image

No file selected.

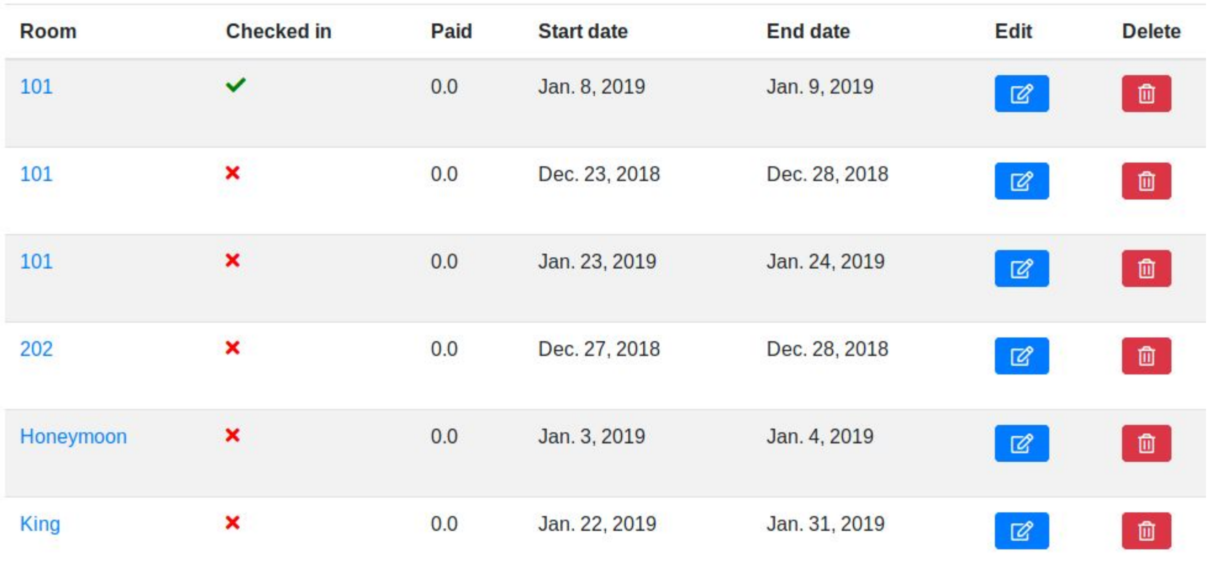
☑ Single bed

□ Double bed

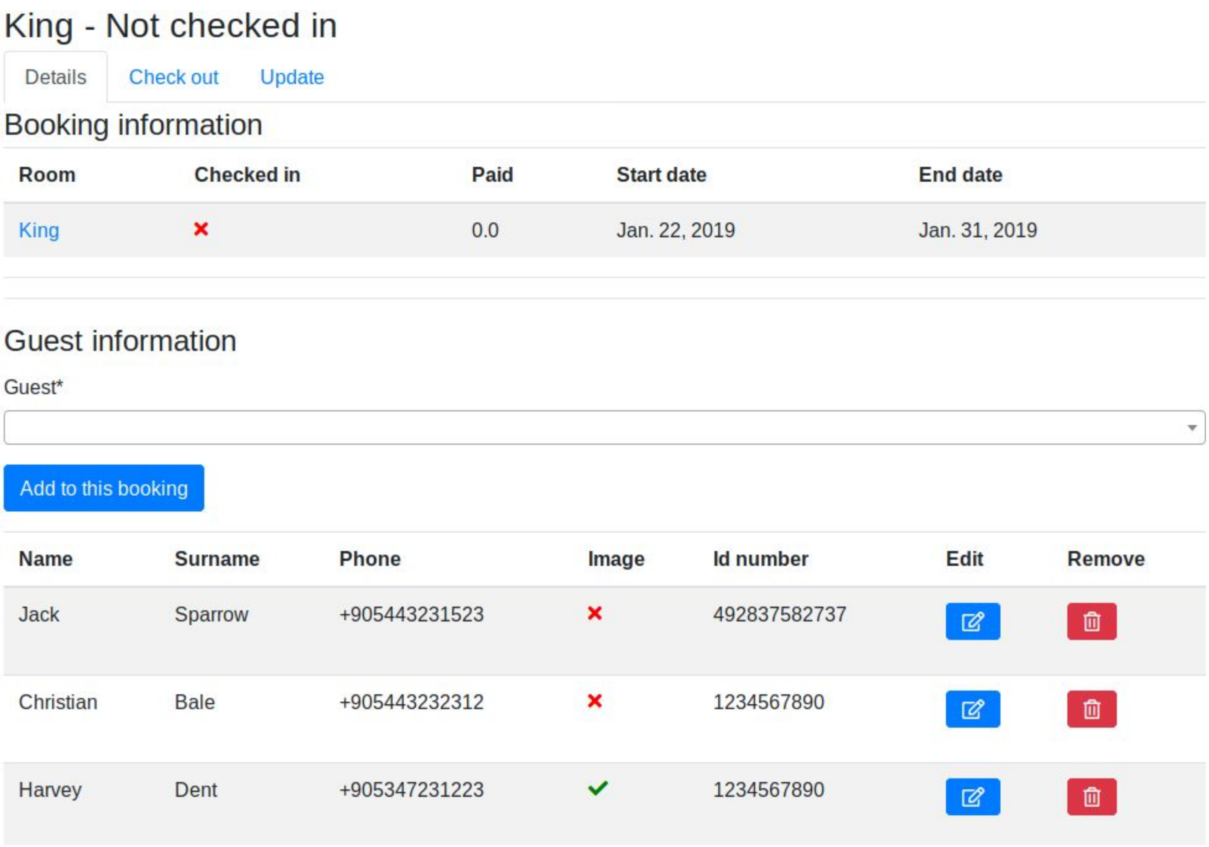
□ Child bed

☑ Has view

Bookings



Room	Checked in	Paid	Start date	End date	Edit	Delete
101	✓	0.0	Jan. 8, 2019	Jan. 9, 2019	✎	🗑
101	✗	0.0	Dec. 23, 2018	Dec. 28, 2018	✎	🗑
101	✗	0.0	Jan. 23, 2019	Jan. 24, 2019	✎	🗑
202	✗	0.0	Dec. 27, 2018	Dec. 28, 2018	✎	🗑
Honeymoon	✗	0.0	Jan. 3, 2019	Jan. 4, 2019	✎	🗑
King	✗	0.0	Jan. 22, 2019	Jan. 31, 2019	✎	🗑



King - Not checked in

Details | Check out | Update

Booking information

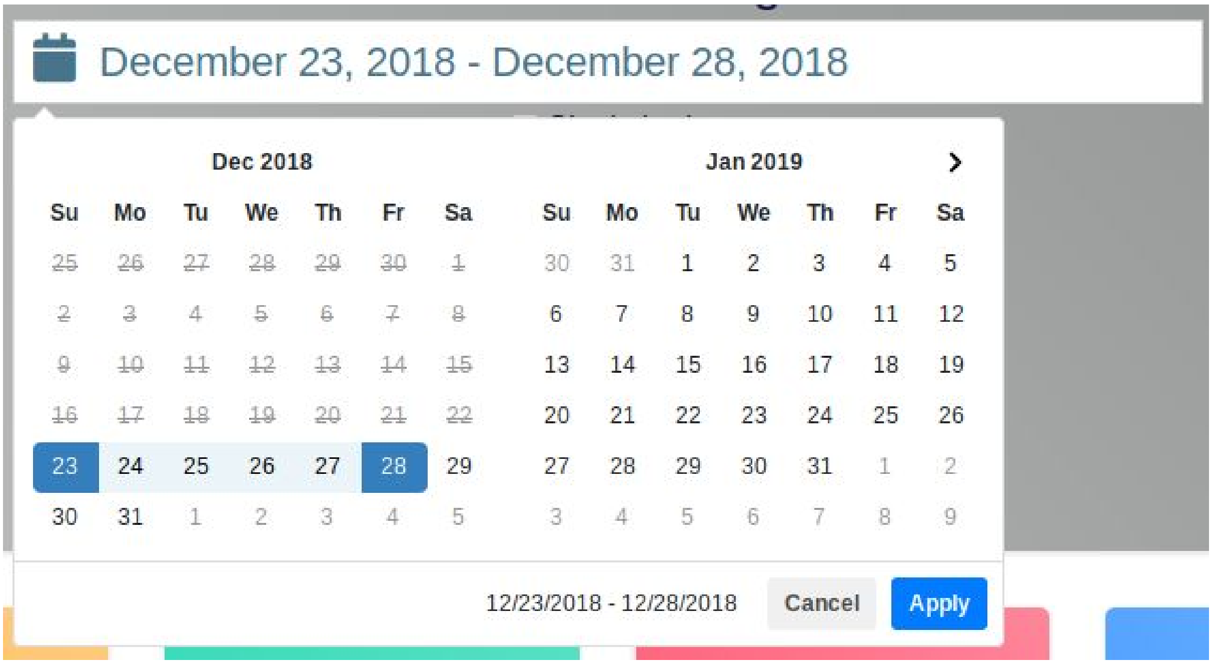
Room	Checked in	Paid	Start date	End date
King	✗	0.0	Jan. 22, 2019	Jan. 31, 2019

Guest information

Guest*

Add to this booking

Name	Surname	Phone	Image	Id number	Edit	Remove
Jack	Sparrow	+905443231523	✗	492837582737	✎	🗑
Christian	Bale	+905443232312	✗	1234567890	✎	🗑
Harvey	Dent	+905347231223	✓	1234567890	✎	🗑



December 23, 2018 - December 28, 2018

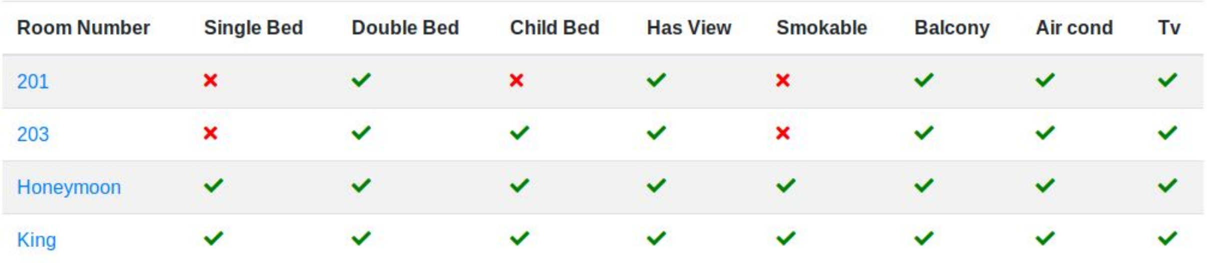
Dec 2018

Su	Mo	Tu	We	Th	Fr	Sa
~~25~~	~~26~~	~~27~~	~~28~~	~~29~~	~~30~~	~~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	31	1	2	3	4	5

Jan 2019

Su	Mo	Tu	We	Th	Fr	Sa
30	31	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	31	1	2
3	4	5	6	7	8	9

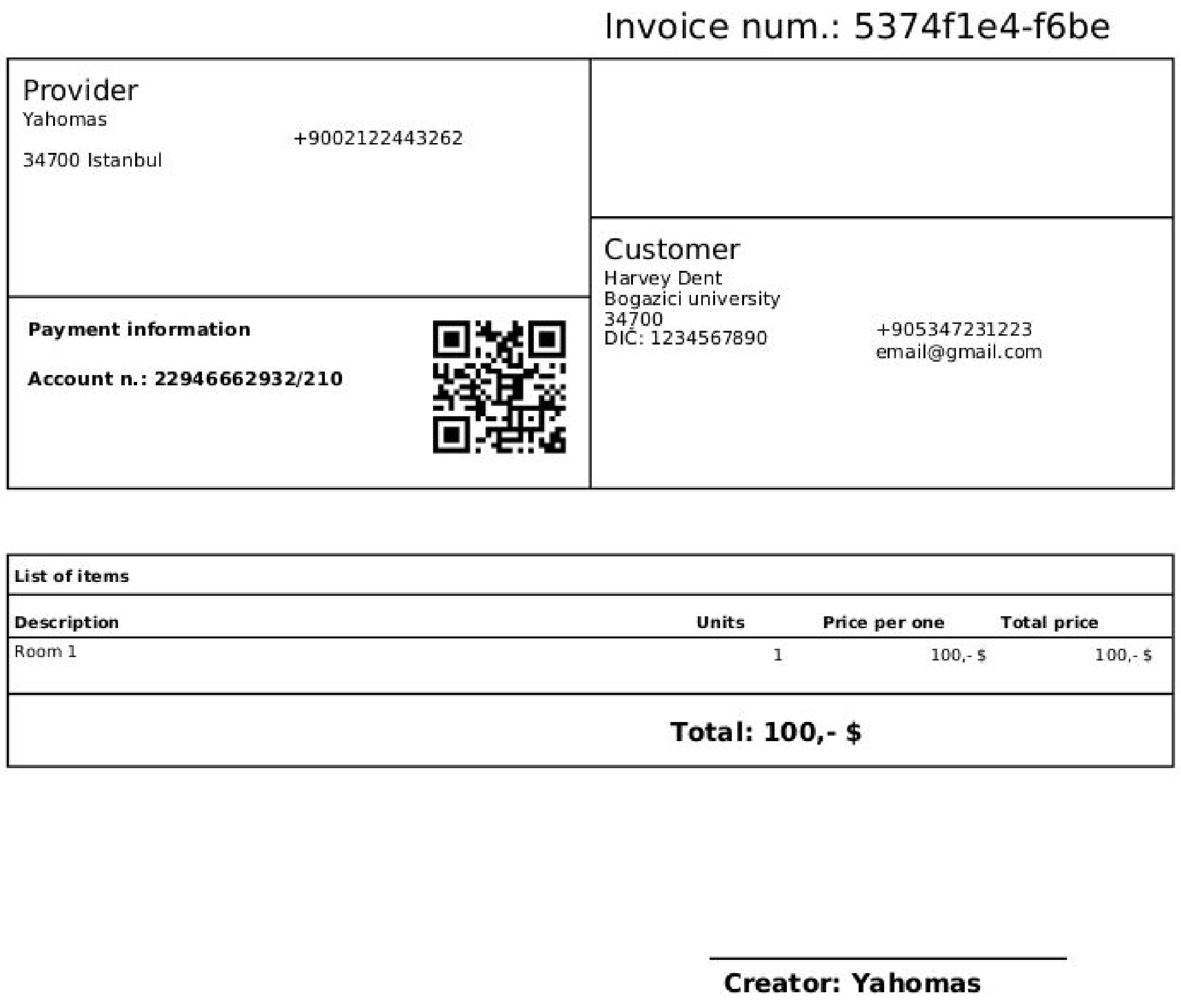
12/23/2018 - 12/28/2018 Cancel Apply



Room Number	Single Bed	Double Bed	Child Bed	Has View	Smokable	Balcony	Air cond	Tv
201	✗	✓	✗	✓	✗	✓	✓	✓
203	✗	✓	✓	✓	✗	✓	✓	✓
Honeymoon	✓	✓	✓	✓	✓	✓	✓	✓
King	✓	✓	✓	✓	✓	✓	✓	✓

Invoice

An invoice example created by YAHOMAS,



Invoice num.: 5374f1e4-f6be

Provider
Yahomas
+9002122443262
34700 Istanbul

Payment information

Account n.: 22946662932/210

Customer
Harvey Dent
Bogazici university
34700
DIČ: 1234567890
+905347231223
email@gmail.com

List of items

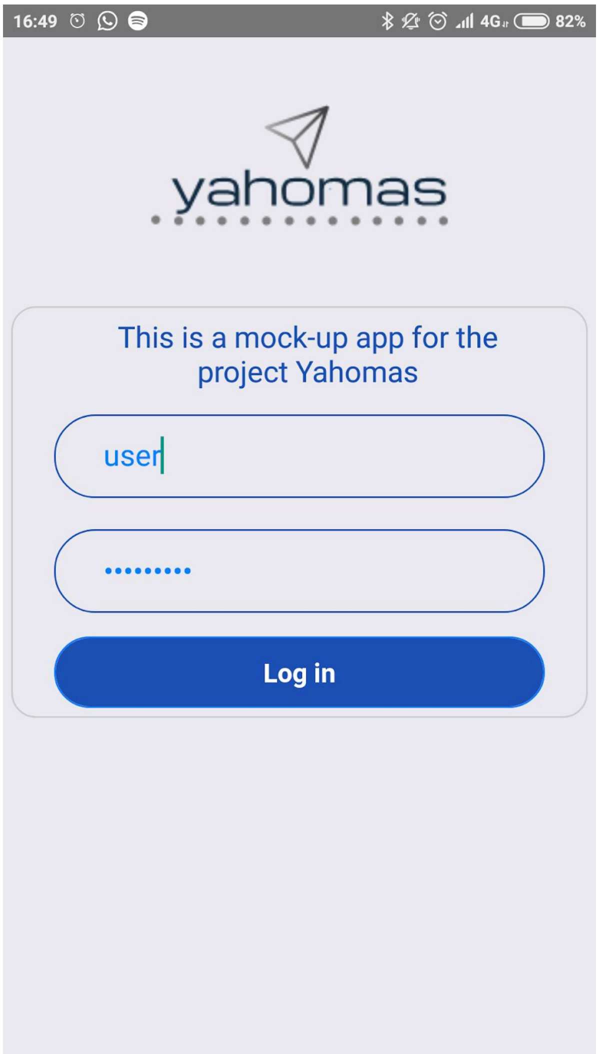
Description	**Units**	**Price per one**	**Total price**
Room 1	1	100,- $	100,- $

Total: 100,- $

Creator: Yahomas

The app

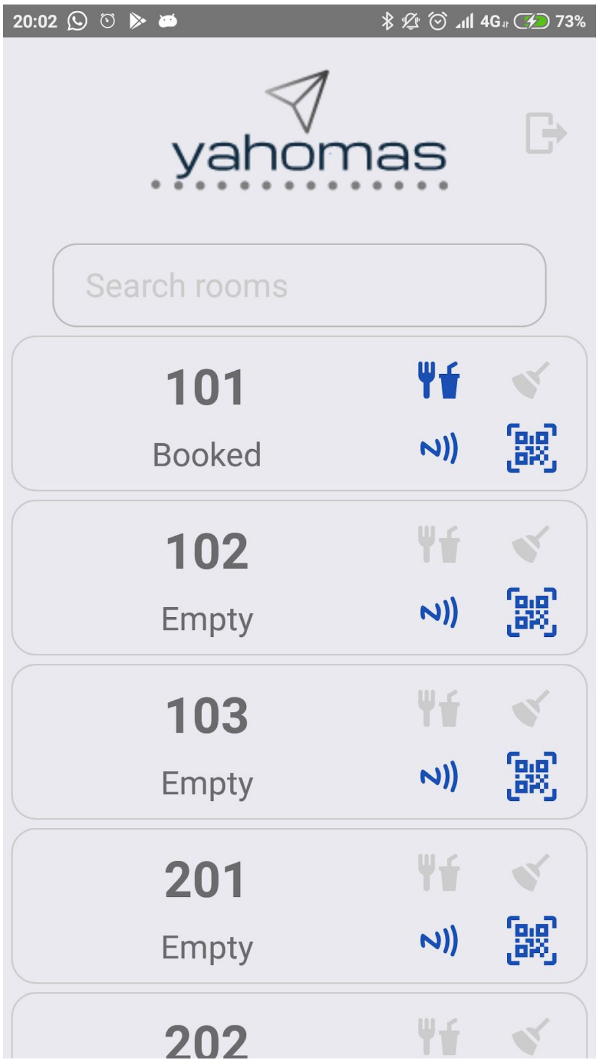
The app screens are as follows: Login screen, Room screen, QR code generator and Item screen.



This is a mock-up app for the project Yahomas

user

•••••••••



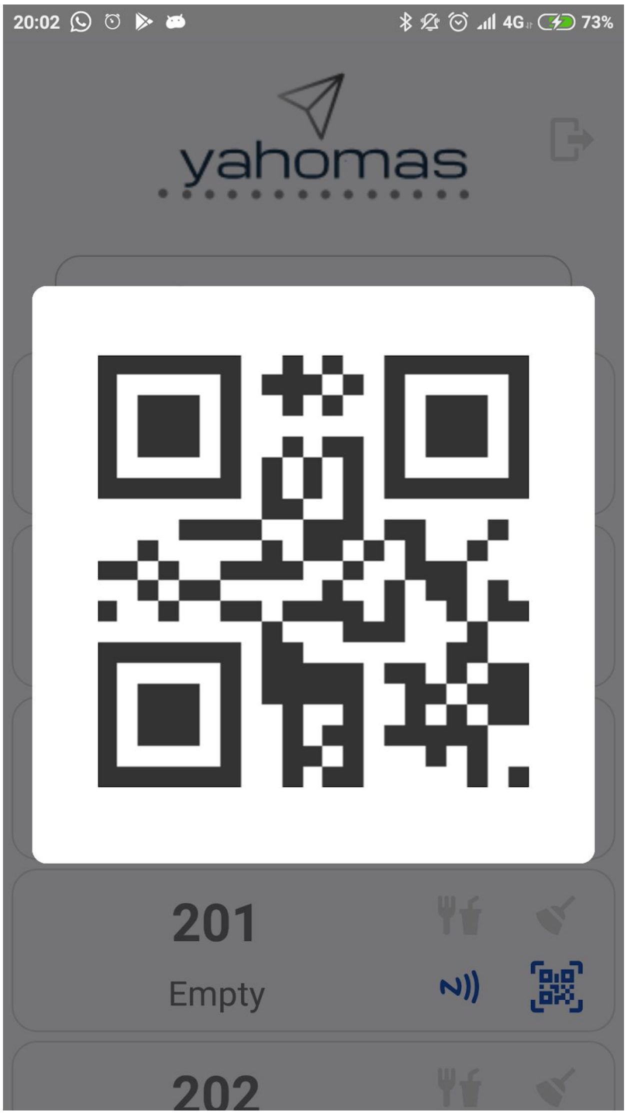
101
Booked

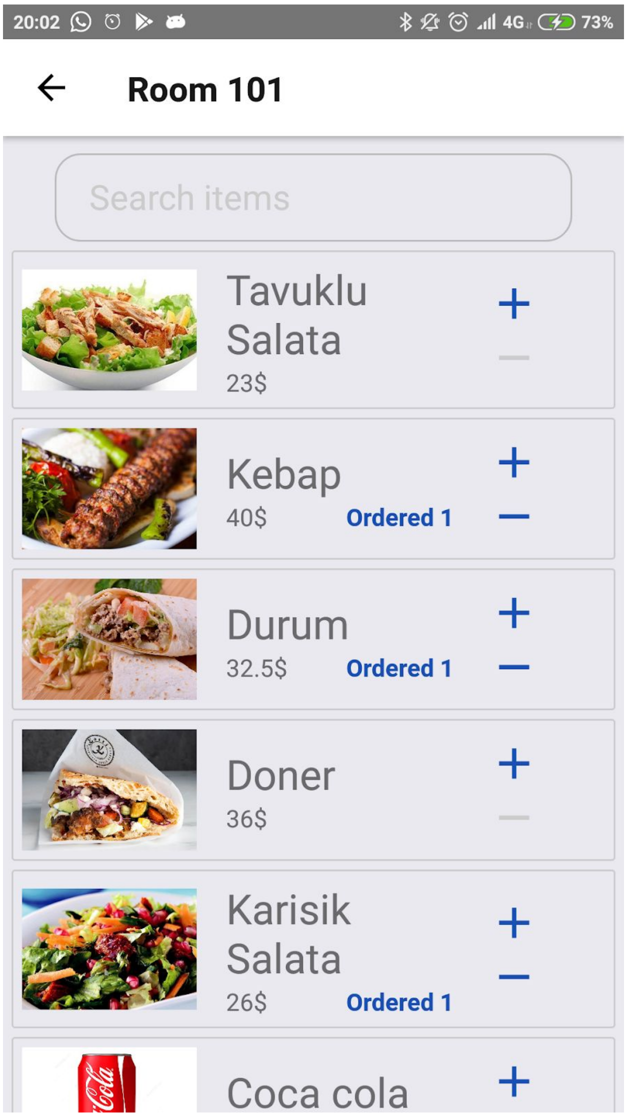
102
Empty

103
Empty

201
Empty

202





Room 101

Search items

Tavuklu Salata	23$	
Kebap	40$	Ordered 1
Durum	32.5$	Ordered 1
Doner	36$	
Karisik Salata	26$	Ordered 1
Coca cola		

Conclusion

In conclusion, I was able to create a website and an app that can talk to one another. Using django for web made things easier in terms of separation of concerns and code first approach. It is very fast and the components are easy to create for reuse. The final product is secure in terms of protocol (https) and with this project I managed to learn many things, front-end development (javascript, css and layout) as well as mobile component (react-native). The project have various aspect to be improved. But the current features are easy to use and sufficient for wide variety usage. I improved my abilities via this project and writing the code was very educational.

The performance of the webapp is very good. Django is a well known framework for fast, web app production. Several libraries that are mentioned in this report, as well as the Github page used for enhancing the experience of the user. It provides easy to use components and produces invoices. The mobile first approach made it possible that any functionality that is provided in the web can be easily viewed by a mobile phone.

The mobile app is written once in javascript using react-native framework and ready for production in both iOS and Android platforms.

References

1. Ogirima,, S. (2014). Online computerized Hotel Management System. *Scienceq*.
2. Hotel Management System, Jin Y, 2015 Governors State University
3. Hotel Management System, N Jacob, KR Hemant
4. (2015) Hotel Management System. *StudyM*
5. Noor-A-Rahim, el at. (March 2011) An Electronic Intelligent Hotel Management System
6. For International Marketplace. *IJACSA*
7. Django project https://www.djangoproject.com/
8. Rest framework https://www.django-rest-framework.org/
9. React native https://facebook.github.io/react-native/
10. Security crypto https://pycryptodome.readthedocs.io/en/latest/