



Bilkent University

Department of Computer Engineering

Senior Design Project

whotello: time to interact with your hotel

Final Report

Alina Zhumasheva, Bayram Muradov, Imran Hajiyeu

Supervisor: Uğur Doğrusöz

Jury Members: H. Altay Güvenir, İbrahim Körpeoğlu

Final Report
July 31, 2019

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Table of Contents

1. Introduction	4
2. Final Architecture and Design	5
2.1 Final Architecture	5
2.1.1 Mobile Application Architecture	5
2.1.1.1 Client-side Architecture	5
2.1.1.2 Server-side Architecture	5
2.1.2 Website Architecture	6
2.1.2.1 Client-side Architecture	6
2.1.2.2 Server-side Architecture	6
2.1.3 Database Architecture	6
2.1.4 API Architecture	7
2.1.4.1 Whotello Chatbot API	7
2.1.4.2 Whotello Statistics API	7
2.2. Design Trade-Offs	8
2.2.1 Functionality vs. Usability	8
2.2.2 Cost vs. Availability	8
3. Algorithms	9
4. Impact of Engineering Solutions	10
4.1 Global Impact	10
4.2 Social Impact	10
5. Tools and Technologies Used	10
5.1 Front-End	10
5.1.1 React JS	10
5.1.2 Flutter	11
5.1.4 Material-UI	11
5.2 Back-End	11
5.2.1 Firebase Authentication	11
5.2.2 Firebase RealTime Database	12
5.2.3 Firebase Storage	12
5.2.4 Flask Web Application Framework	12
5.2.5 Python SciKit Learn Toolkit	13
5.2.6 Heroku Cloud Application Platform	13
5.3 APIs	14
5.3.1 Google Maps	14
5.3.2 RESTful API	14
5.3.3 QR.Fluter	14
5.3.4 Barcode Scanner	14
5.3.5 Location	14

6. Contemporary Issues	15
6.1 Limited Model Access	15
6.2 Raspberry Pi	15
6.3 Hotel Negotiation	15
7. User's Manual	16
7.1 Mobile App	16
7.2 Website	30
8. Installation Manual	34
9. References	35

1. Introduction

The number of hotel users around the globe is quite large, surpassing the benchmark of 15.5 million rooms available for active usage scattered around the globe [1]. However, despite the fact that hotel service exists from 705, even nowadays using hotel services sometimes becomes challenging for both the guests and the administrators [2]. Reserving restaurants, chaise longues, requesting the cleaning or food, learning answers to frequently asked questions and even managing the room temperature for guests; managing the personnel, user preferences, food habits, overall hotel statistics for administrators may present itself a bit difficult due to the human factor in all of the processes and all of them may be optimized for the both groups benefits.

Our system is designed in a way to tackle and solve all of those problems, making the guest experience at the hotel as effortless and pleasant as possible and at the same time easing and simplifying the managing process for the administrators, giving the latter undesired benefits such as specified statistics and data for the further improvement of the hotel service - and all of this is automated, removing the human factor.

2. Final Architecture and Design

2.1 Final Architecture

2.1.1 Mobile Application Architecture

2.1.1.1 Client-side Architecture

The client side of the mobile application is implemented using Flutter which is the newest and fastest developing toolkit for creating mobile native apps. The application has native performance on both iOS and Android, which means it is really fast and reliable. Application consists of 3 main activities and a 5 sub-activities, which is a small number for such a big project. All the functionality is compact, simple, and intuitive, requiring user to use nothing more than a thumb. We have also integrated many external APIs to our user interface to improve our users experience. Some of these external services are Google Maps, QRCode Generator, Material Design and etc.

2.1.1.2 Server-side Architecture

Our mobile application server-side consists of Firebase Database and ChatBot Api that is deployed to both Heroku and Google Cloud. Application is connected and fetches/writes data to the database in real-time. All the changes are saved even if the application is in offline mode. As soon as it gets internet all the saved changes are pushed into the database. ChatBot API works using RESTful API, that is POST and GET operations, from either of the servers - Heroku or Google Cloud.

2.1.2 Website Architecture

2.1.2.1 Client-side Architecture

The client-side of the website was implemented and designed using React, since it is the JS library that makes it easy to create interactive UIs. Implementing small components to combine them in a well-functioning system was a good idea, since it makes it easy to compose, share and reuse these components and integrate them as you like. From our main site “whotello.com” hotel admin can go to their dashboard by signing in. We designed and implemented website in a way so that users can easily use it and manage their job.

2.1.2.2 Server-side Architecture

The server-side of the website consists of Firebase Database that is deployed to Google Cloud. Website is connected and writes data to the database in real-time, at the same time it fetches data from database that was written through mobile-app server-side. All the changes are saved even if the application is in offline mode. As soon as it gets internet all the saved changes are pushed into the database.

2.1.3 Database Architecture

The database we chose for this system is Google Firebase Realtime Database. We decided to use this database because we needed a No-SQL database, that will be functioning in real-time. Since we are running both a web application and a mobile application, all the communications between backend server and client-side application are based on JSON objects. Our external private custom ChatBot API is

also communicating using JSON objects. Therefore, we thought it is a good idea to store our objects in the database that uses JSON format. We use a single database for both mobile application and website, where information, inputted by the administrator to the website, has to be retrieved by mobile application and shown to the guests. Therefore all the updates to the database have to be visible the second they are made, which is why we are using a real-time database. Such interconnected information is usually the events the hotel is going to host, hotel facilities and room openings.

2.1.4 API Architecture

2.1.4.1 Whotello Chatbot API

Whotello Chatbot API gets POST requests with question string and name of the hotel attached to the JSON file. After getting the question string from the user it creates response and sends the response back to the client. Flask web-application framework has been used in order to create the RESTful API since its flexible and doesn't enforce any dependencies on developer. Heroku Cloud Application Platform has been used to deploy the API.

2.1.4.2 Whotello Statistics API

Whotello statistics API gets POST request from the user with hotel name string attached to the JSON file. Once the API gets hotel name it fetches data from database accordingly and processes them. After processing API encodes the output of process and sends it back to the client. Flask web-application framework has been

used in order to develop the API. Heroku Cloud Application Platform has been used to deploy the API.

2.2. Design Trade-Offs

2.2.1 Functionality vs. Usability

Functionality and usability are one of the most important parameters for the “whotello” project. Since the main purpose of the project whotello is to make the hotel industry as easy and simple to use as possible - helping both Guest and Administration side to do their main tasks in a simplified and comfortable way - only the core and must-have functionality is included. If the project will carry out a lot of functionality, then automatically it becomes much harder to use. Therefore, it is important to decide which features are decreasing the usability parameter and which ones we can implement while maintaining a good UX and simple UI.

2.2.2 Cost vs. Availability

Connecting to the room devices, such as TV and AC is done in either of two ways. One way is to connect to TV and AC using infrared signals, that will be emitted from the mobile phone itself and thus controlling the TVs and ACs using correct HEX codes. The cost of this implementation is the lowest possible, as no additional devices have to be bought and placed in each hotel room. However, that comes with a trade-off of availability. Some models, won't be able to be controlled by the Guest's phone, because they can't detect infrared signals and require either a Wi-Fi connection or radio frequency in order to be controlled. Other problem is that some

of the phone models cannot emit infrared signals. Therefore, for such scenarios, it is a must to buy and place Raspberry Pi, to be able to control them using a mobile phone. Server is going to be deployed on Raspberry Pi that controls the electronic devices in the room. However, that results in a sharp increase in the cost.

3. Algorithms

Our system makes use of different algorithms and mathematical methods in both of the APIs it makes use of. In this section implementation details inside of the APIs will be discussed.

Whotello Chatbot API makes use of TfidfVectorizer and cosine similarity method in order to sort and cluster the possible responses among the pool of predefined answers and choose the best one. These methods are already implemented and provided in Python SciKit-learn toolkit and Whotello chatbot makes use of them in order to give satisfactory response to the users.

Whotello Statistics API provides statistics to the admin console in sorted order. Data that has been fetched from the database is being sorted by the API by using counting sort which ensures linear time response in the worst case.

4. Impact of Engineering Solutions

4.1 Global Impact

Whotello system is aimed to make a difference in traditional hotel management systems. Only few applications are available currently in the market which provide the use cases that Whotello has. We believe that by following appropriate marketing strategies system can be expanded to be used by the hotels all around the world.

4.2 Social Impact

Whotello is designed as a system to tackle hotel-guest interaction problems. By using Whotello application users on both sides including admins and the guests will ease their workload and save time.

5. Tools and Technologies Used

5.1 Front-End

5.1.1 React JS

A framework to build encapsulated and written in JavaScript components that manage their own state; they can be composed to make complex UIs, since rich data can easily be passed through app [3].

5.1.2 Flutter

Flutter is Google's mobile app SDK for crafting high-quality native interfaces on iOS and Android in record time. It is free and open source and works with existing code, and used by developers and organizations around the world [4].

5.1.3 Material Design

Material is an adaptable system of guidelines, components, and tools that support the best practices of user interface design. Backed by open-source code, Material streamlines collaboration between designers and developers, and helps teams quickly build beautiful product [5].

5.1.4 Material-UI

React Components that are designed according to Material Design principles published by Google [6].

5.2 Back-End

5.2.1 Firebase Authentication

Firebase Authentication aims to make building secure authentication systems easy, while improving the sign-in and onboarding experience for end users. It provides an end-to-end identity solution, supporting email and password accounts, phone auth, and Google, Twitter, Facebook, and GitHub login, and more [7].

Built by the same team that developed Google Sign-in, Smart Lock and Chrome Password Manager, Firebase security applies Google's internal expertise of managing one of the largest account databases in the world.

5.2.2 Firebase RealTime Database

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real time. When your users go offline, the Realtime Database SDKs use local cache on the device to serve and store changes. When the device comes online, the local data is automatically synchronized. The Realtime Database integrates with Firebase Authentication to provide simple and intuitive authentication for developers. You can use our declarative security model to allow access based on user identity or with pattern matching on your data [8].

5.2.3 Firebase Storage

Cloud Storage is designed to help you quickly and easily store and serve user-generated content, such as photos and videos. Your users aren't always online, so we built the Firebase SDK for Cloud Storage with mobile connectivity in mind. It will automatically pause and resume your transfers as the app loses and regains mobile connectivity, saving your users time and bandwidth [9].

5.2.4 Flask Web Application Framework

Flask (source code) is a Python web framework built with a small core and easy-to-extend philosophy. Flask is considered more Pythonic than the Django web

framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running. [10]

5.2.5 Python SciKit Learn Toolkit

Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms. It's built upon some of the technology you might already be familiar with, like NumPy, pandas, and Matplotlib! [11]

5.2.6 Heroku Cloud Application Platform

Heroku is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. Our platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market.[12]

5.3 APIs

5.3.1 Google Maps

Google Maps is a service provided by Google that let us put the map in the background of our main page. We show the markers of the source and destinations on the map [13].

5.3.2 RESTful API

RESTful API is designed to take advantage of existing protocols, it usually takes advantage of HTTP when used for Web APIs. In order to take advantage of a REST API design developers do not need to install libraries or additional software [14].

5.3.3 QR.Fluter

QR.Fluter is a Flutter library for simple and fast QR code rendering via a Widget or custom painter [15].

5.3.4 Barcode Scanner

A flutter plugin for scanning 2D barcodes and QR codes [16].

This provides a simple wrapper for two commonly used iOS and Android libraries:

iOS: <https://github.com/mikebuss/MTBBarcodeScanner>

Android: <https://github.com/dm77/barcodescanner>

5.3.5 Location

This plugin for Flutter handles getting location on Android and iOS. It also provides callbacks when location is changed [17].

6. Contemporary Issues

6.1 Limited Model Access

Currently our application supports only one brand of the TVs and ACs, because only their HEX codes are inputted inside the application. In order to support more than one model, lots of brand and model specific HEX codes for specific commands have to be stored and retrieved at the right time, and application structure will need to be changed in order to support it and provide simple and automated user experience to guests.

6.2 Raspberry Pi

Currently IoT part of our application works only with infrared (IR) signals. However, in order to cover more scale, that is make the system as scalable as possible, Raspberry Pi is going to come in handy. Some rooms may need to install a Raspberry Pi, that will run as server and transmitter of signals in order to cope with problems of phone models that do not have IR transmitter and also if needed support TVs and ACs that use radio frequency (RF) signals.

6.3 Hotel Negotiation

In order to control the doors and lights, the application needs to be integrated into the hotel's lights and doors service, which can only be done by having a contract with a hotel.

7. User's Manual

7.1 Mobile App

2:00



DEBUG



E-mail



Password

CANCEL

NEXT

Do not have an account?

2:00



DEBUG



Sign Up



Full Name



E-mail



Password



Phone Number

SIGN UP

2:00



QR Code Scanner



START CAMERA SCAN

USE SHARED CODE



My Room



Door

Lock or unlock your hotel room.



Air Conditioner

Remote control Air Conditioner in your room.



Lights

Remote control lights in your room.

Turn ON/OFF



TV

Remote control TV in your room.

Turn ON/OFF

Volume UP

Volume DOWN

Channel UP

Channel DOWN





My Room



Welcome to Bilkent Hotel!



Events



Hotel Information



Map



Requests



Reservations



2:01



Profile Page



Imran Hajiyeu

Personal Information:



Imran Hajiyeu



imran@google.com

Room Preferences:

Preference Set #1

Description goes here.

Preference Set #2

Description goes here.

Preference Set #3

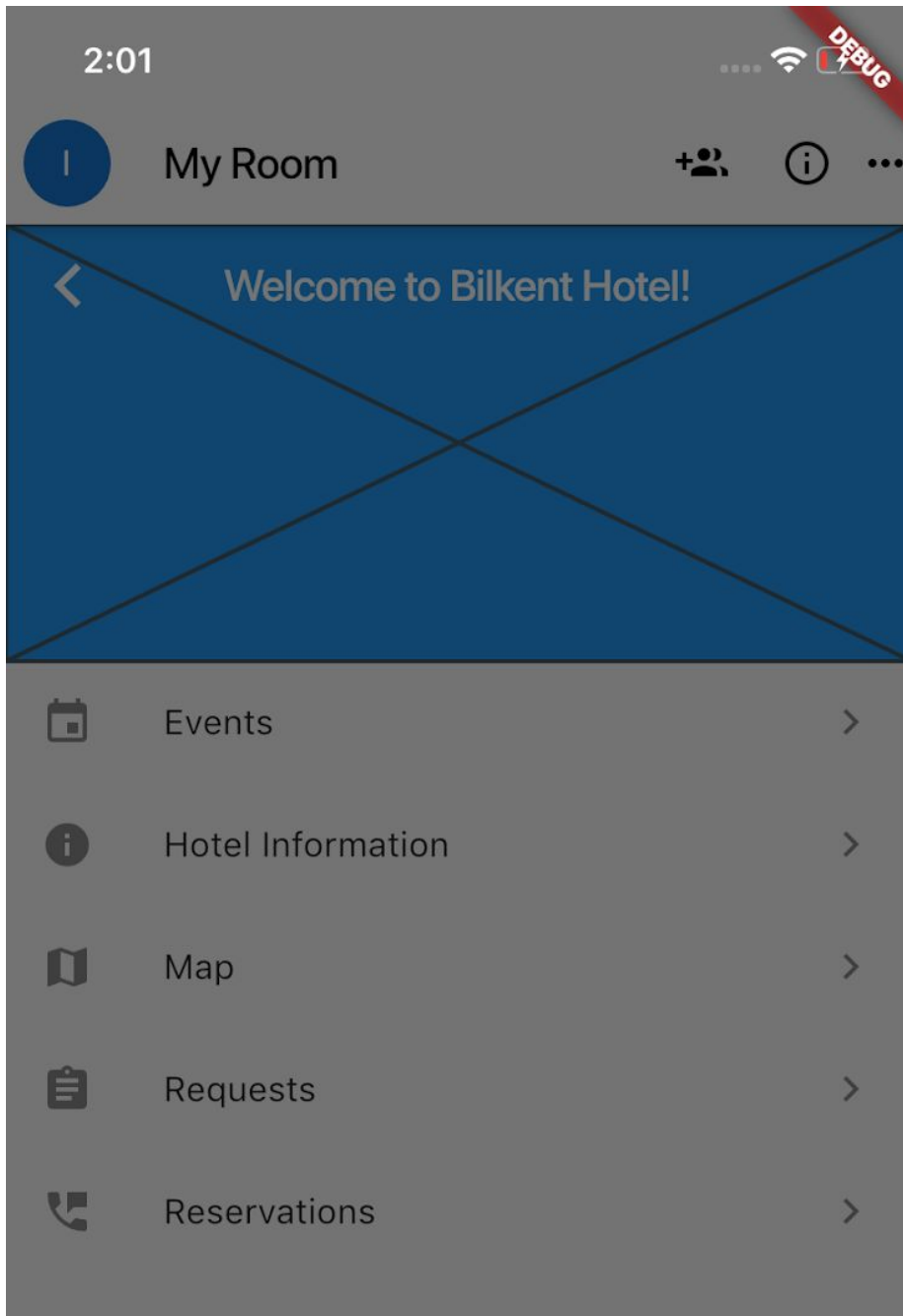
Description goes here.

Preference Set #4

Description goes here.

Preference Set #5

Description goes here.



Add people to your room:



2:01



DEBUG



My Room



Door

Lock or unlock your hotel room.



Air Conditioner

Remote control Air Conditioner in your room.



Contact Information:

+90 (536) 890-12-93

+90 (553) 938-40-12

Emergency Contacts:

+90 (555) 404-40-04

+90 (555) 101-40-04

Turn ON/OFF



Close

Turn ON/OFF

Volume UP

Volume DOWN

Channel UP

Channel DOWN



DEBUG



My Room



Imran

adsasd



Bot

I am sorry! I don't understand you



Imran

Where can I smoke cigarettes?



Bot

cigarettes can be smoked at the terrace.



Enter Message



2:03



DEBUG



Google Maps



2:03



Requests

Breakfast

Dinner

Complete Request Form

You are requesting *Supper*

Arrival Time:

Select Time

CANCEL

SEND REQUEST

Medicine

Doctor

Help

2:05 ↗



DEBUG



Reservations

The North Shield Restaurant



+ Add reservation

2:05



Reservation Form

1

Choose an activity

List of available activities is provided below

Alis Patisserie Restaurant



CONTINUE

CANCEL

2

Enter number of people

Provide number of people involved

3

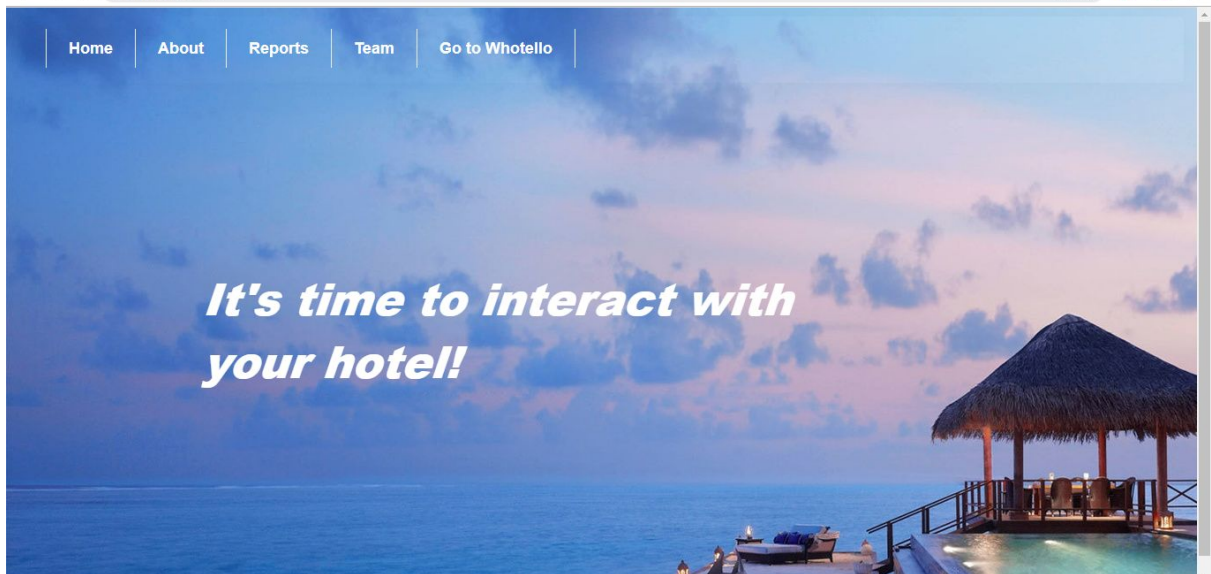
Pick a time slot

Pick an available date and time for you activity



Complete reservation

7.2 Website



Log in


e-mail

Password


Sign upLog in

☐ Remember

[Forget Password?](#)



[Settings](#)
[QRCode](#)
[Log out](#)



Demo User

📄 HOTEL INFO

★ HOTEL EVENTS

🍽️ RESTAURANT

📅 RESERVATIONS

📋 REQUESTS

🕒 STATISTICS

👍 GUEST REVIEWS

Hotel Information

Hotel name

Website

Address

City


Country

Postal Code


About Hotel

Here can be your description

Update Hotel Info



[Settings](#)
[QRCode](#)
[Log out](#)



Demo User

📄 HOTEL INFO

★ HOTEL EVENTS

🍽️ RESTAURANT

📅 RESERVATIONS

📋 REQUESTS

🕒 STATISTICS

👍 GUEST REVIEWS

23	24	25	26	27	28	29
30	01	02	03	04	05	06

Add new event

Event name

Start date


Start time

End date


End time

☐ All day
 ☐ All night
 ☐ Birthday

Add event



[Settings](#)
[QRCode](#)
[Log out](#)



Demo User

📄 HOTEL INFO

★ HOTEL EVENTS

🍽️ RESTAURANT

📅 RESERVATIONS

📋 REQUESTS

🕒 STATISTICS

👍 GUEST REVIEWS

Restaurant

Breakfast

Here can be your description

Update Breakfast


Lunch

Here can be your description

Update Lunch


Dinner

Here can be your description



SettingsQRCode

Log out



Demo User

HOTEL INFO

HOTEL EVENTS

RESTAURANT

RESERVATIONS


REQUESTS

STATISTICS

GUEST REVIEWS


List of Reservations

Name	Room	Reservation	Date	Time	Accept All
Anna	9	spa	10/04/2019/	15:00	Accept
Mehmet	23	chaise longue	10/04/2019/	17:00	Accept



SettingsQRCode

Log out



Demo User

HOTEL INFO

HOTEL EVENTS

RESTAURANT

RESERVATIONS

REQUESTS

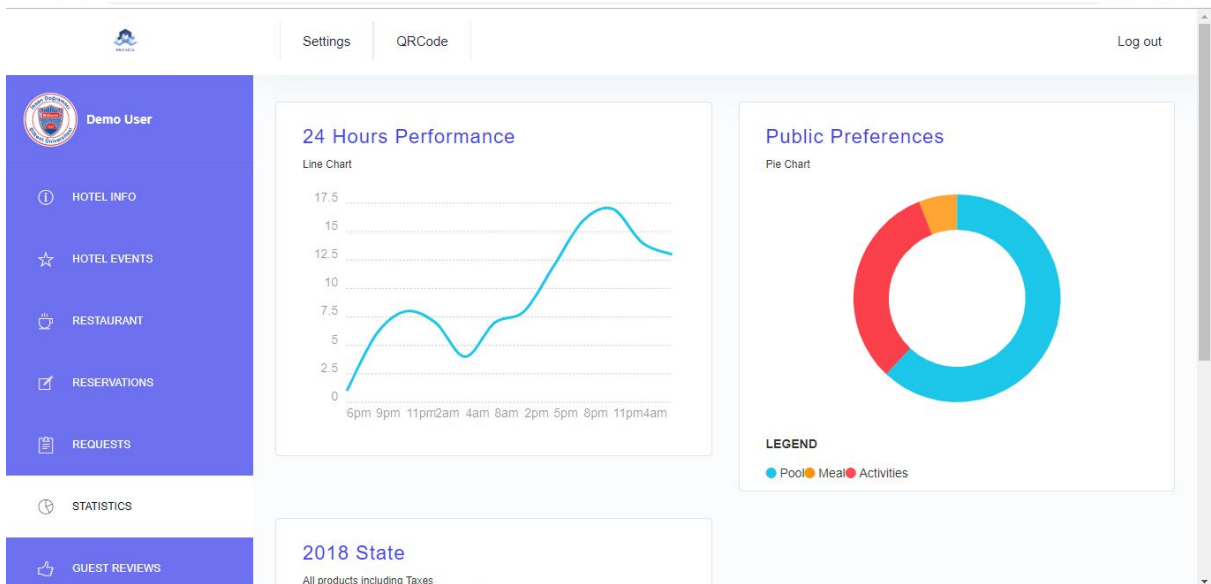
STATISTICS


GUEST REVIEWS

Current requests

Name	Room	Request	Accept All
Anna	9	cleaning	Accept
Mehmet	23	drink	Accept

localhost:3000/?#/requests






Settings

QRCode

Log out



Demo User

① HOTEL INFO

☆ HOTEL EVENTS

☕ RESTAURANT

☑ RESERVATIONS


📄 REQUESTS

🕒 STATISTICS

👍 GUEST REVIEWS


Reviews from users

Date	Grade	Comment
09/04/2019	8	I was glad to have a weekend here. Food was very delicious. But my room wasn't clean as I wish
10/04/2019	5	It wasn't good



[Settings](#)
[QRCode](#)

[Log out](#)



Demo User

HOTEL INFO

HOTEL EVENTS

RESTAURANT

RESERVATIONS

REQUESTS

STATISTICS

GUEST REVIEWS


Settings

Email address

Password


Confirm password

Change Settings



[Settings](#)
[QRCode](#)

[Log out](#)



Demo User

HOTEL INFO

HOTEL EVENTS

RESTAURANT

RESERVATIONS

REQUESTS

STATISTICS


GUEST REVIEWS

QR Code

Room	QR code	
1	<div>Show QR</div>	<div>Remove All</div> <div>Remove</div>
2	<div>Show QR</div>	<div>Remove</div>
3	<div>Show QR</div>	<div>Remove</div>
4	<div>Show QR</div>	<div>Remove</div>
5	<div>Show QR</div>	<div>Remove</div>

1

Generate QR



8. Installation Manual

Whotello is a system that is public and accessible from everywhere. In order to use it as a user you just need to go to whotello.com website if you are an administration of the hotel, or if you are an ordinary traveller/hotel user - download iOS version of whotello application from the AppStore or Android version from Google Store. However, if you want to run our project now as a developer, you need to access the GitHub repository and download the source

code from there. After that just use and follow the README file that is provided in the GitHub repository.

9. References

1. Rebecca Lake, "*Hotel Industry Statistics*", 26-Apr-2016. [Online]. Retrieved from: <https://www.creditdonkey.com/hotel-industry-statistics.html>
[Accessed: 24-Feb-2019].
2. Guinness World Records, "*Oldest Hotel*", 29-Oct-2011. [Online]. Retrieved from: <http://www.guinnessworldrecords.com/world-records/oldest-hotel/>
[Accessed: 24-Feb-2019].
3. React, "*JavaScript Library for building user interfaces*" [Online]. Retrieved from: <https://reactjs.org/>
[Accessed: 25-July-2019].
4. Flutter, "*Flutter makes it easy and fast to build beautiful mobile apps*" [Online]. Retrieved from: <https://github.com/flutter/flutter>
[Accessed: 25-July-2019].
5. Design, "*Material Design*" [Online]. Retrieved from: <https://material.io/design/>
[Accessed: 25-July-2019].
6. "The world's most popular React UI framework - Material-UI," Material. [Online]. Retrieved from: <https://material-ui.com/>
[Accessed: 26-July-2019].
7. Firebase, "*Simple, free, multi-platform sign-in*" [Online]. Retrieved from: <https://firebase.google.com/products/auth/>
[Accessed: 24-July-2019].

8. Firebase, "*Store and sync data in real-time*" [Online]. Retrieved from:
<https://firebase.google.com/products/realtime-database/>
[Accessed: 24-July-2019].
9. Firebase, "*Store and serve content with ease*" [Online]. Retrieved from:
<https://firebase.google.com/products/storage/>
[Accessed: 24-July-2019].
10. Full Stack Python, "*Flask*" [Online]. Retrieved from:
<https://www.fullstackpython.com/flask.html>
[Accessed: 24-July-2019].
11. Codecademy, "*What is SciKit Learn?*" [Online]. Retrieved from:
<https://www.codecademy.com/articles/scikit-learn>
[Accessed: 24-July-2019].
12. "*What is Heroku?*" [Online]. Retrieved from: <https://www.heroku.com/about>
[Accessed: 24-July-2019].
13. Google Street View API. [Online]. Available:
<https://developers.google.com/maps/documentation/streetview/intro>.
[Accessed: 09-May-2019].
14. MuleSoft, "*What is REST API?*" [Online]. Retrieved from:
<https://www.mulesoft.com/resources/api/what-is-rest-api-design>
[Accessed: 25-July-2019].
15. QR Flutter [Online]. Retrieved from: https://pub.dev/packages/qr_flutter
[Accessed: 25-July-2019].
16. Pub Dev, "*Barcode Scanner*" [Online]. Retrieved from

https://pub.dev/packages/barcode_scan

[Accessed: 25-July-2019].

17. Pub Dev, "*Flutter Location Plugin*" [Online]. Retrieved from

<https://pub.dev/packages/location>

[Accessed: 25-July-2019].