Bilkent University

Department of Computer Engineering

Senior Design Project

whotello: time to interact with your hotel

Analysis Report

Alina Zhumasheva, Bayram Muradov, Imran Hajiyev

Supervisor: Uğur Doğrusöz
Jury Members: H. Altay Güvenir, İbrahim Körpeoğlu

Analysis Report

March 25, 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

Contents

Contents **2**

1. Introduction **3**

2. Current System **3**

2.1. ALICE Platform 4

3. Proposed System **4**

3.1. Overview 4

3.2. Functional Requirements 5

3.3. Non-functional Requirements 6

3.4. Pseudo Requirements 7

3.4.1 Technical Constraints 7

3.4.2 Social Constraints 7

3.4.3 Implementation Constraints 7

3.4.4 Sustainability Constraints 7

3.4.5 Economic Constraints 8

3.4.6 Security Constraints 8

3.4.7 Language Constraints 8

3.5. System Model 8

3.5.1 Scenarios 8

3.5.1.1 Guest side 8

3.5.1.2 Administration Side 12

3.5.2 Use case model 15

3.5.2.1.Use Case Diagram #1: Guest 15

3.5.2.2 Use Case Diagram #2: Administration 16

3.5.3 Object and class model 16

3.5.4 Dynamic models 19

3.5.4.1 Sequence Diagrams 19

3.5.4.2 Activity Diagrams 25

3.5.4.2.1 Guest Activity Diagram 25

3.5.4.2.2 Administration Activity Diagram 26

3.5.5 User interface - navigational paths and screen mock-ups 26

4. Glossary **53**

5. References **53**

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. Current System

Nowadays there are a lot of mobile applications that help guests with the hotel of their choice. Those applications serve almost any simple goal a guest would need in their hotel - from ordering any kind of service and FAQ to opening a room lock with their phone. However, most numbers of such applications are hotel itself sponsored applications, that earn their active users some points that later can be redeemed for sales: *Marriott Bonvoy* of Marriott International [3]*, Hyatt Hotels* of Hyatt Corporation [4]*, Hilton Honors* of Hilton [5] - to name but a few. There are also some services for the administration part of the hotel as well, such as *eZee Absolute* [6], which can auto-generate your possible income, detailed business insights and lots more.

However, there is only one software currently that includes both sides of the hotel business, that is both the guest and administration sides, and it is the *ALICE* software. This system is similar to our proposed system in a number of aspects, however, does not tackle the problems that ours does. Here are the short description and features of the *ALICE* software.

2.1. ALICE Platform

Alice platform is a complex system that aims at two main aspects: guest satisfaction and operational excellence. This serves the reason for the following features/products of the platform:

- *Concierge* - provides guests with reservation details, tips, and makes the reservation for them across all the hotel services. Gives guests quick information on places to visit, nearby shops, restaurants, bus and taxi stops outside the hotel.
- *Guest Messaging* - helps with FAQ and other questions or requests that guests may have. Instead of talking to the personnel in real life, messages them.
- *Guest App & Website* - includes both of the features inside a mobile app, which makes it easier to interact with the hotel, without any additional technology to carry around.
- *Logbooks* - tracks every information about the guest's items during his hotel stay: personal items, souvenirs, shopping bags, laptops and etc. Package management and lost and found functionality included.
- *Service Delivery* - optimizes the work for the personnel to minimize the time for completion of their tasks, in order to serve as many guests as possible, so that no guest stays unserviced.
- Preventative Maintenance - maintain and track information on all the equipment inside the hotel, like statistics on what is used more frequently.

3. Proposed System

3.1. Overview

Whotello system consists of two main parts - hotel guest application and administration website. Both parts are connected to each other, even though it is invisible to their related users.

The guest application is just an ordinary cross-platform application that the users will be able to download from Google/App Store and use once they are assigned a certain room at the hotel. As the guests get their room cards, hotel staff is also gonna provide the user with a scannable QR code, which is connected to a specified room. If the group of guests is going to stay in one room, the person who got connected to the room first can share newly generated codes to the room with the rest of the group, in order for them all to be connected. Once setup is ready, the users can experience their application to the fullest, as all the functionality related to the room and hotel is now enabled in the application. Guests can both control their room through the app and also manage the hotel facilities. Control of the room

includes managing the air conditioners, television, lights and also the heating system if the latter is available. They can be controlled both from the inside and outside the room. These features will enable guests to change the room temperature before entering the room if they want to, turn off the lights/TV/AC if they forgot to and many more, depending on the guest preferences: full control of the equipment is at user's disposal. Also, guests can request the cleaning of the room, food being brought to the room and medical personnel in case of an emergency. Hotel facilities, on the other hand, is a functionality that allows guests to interact with their "hotel experience": guests can view all the events, brochures online; view the food menu for every restaurant for any time of day; reserve seats at the available restaurants; reserve places at the beach - chaise longues, to name but a few; get 24/7 help from the chatbot, which will be able to answer most of the guest questions and provide the guests with all the needed number and information about the hotel they are staying in. Chatbot that is going interact with users will also provide answers to most frequently asked questions by guests by combining keyword analysis of a pool of previous conversations and machine learning technique of clustering which will alleviate the workload of hotel administration. In case of unsatisfactory response, guests will be able to communicate with the administration directly.

The administration website is responsible for managing the all user's requests coming from the application, that are specified above, as well as collecting the statistics on the guests' choices, suggesting the administration what is most used/popular among guests and what is, on the contrary, can be removed or updated. Also, the administration account corresponding to the specified hotel is responsible for creating the QR codes that are going to be provided to the hotel guests. In order to create an administrator account corresponding to the hotel, the hotel must send the request to the website, and if the hotel is legal and is suitable for hosting guests, the hotel administration will receive an encrypted key that will allow them to go to the registration page and create their account. Administration can also choose which functionality does their hotel provide and which is not provided.

Together, the two parts - website and the application are merged into a full working system that is going to bring comfort and coziness into the lives of the hotel administration and hotel guests.

3.2. Functional Requirements

There will be two types of users: Guests and Administration.

- Guests will be able to use an app by reading the QR Code given to them by the administration.
- Guests will be able to specify other guests by sharing special code so that people with whom he/she stays in the same room will be able to use an app.

- Guests will be able to manage air conditioners, television, lights in their rooms.
- Guests will be able to view the hotel brochure.
- Guests will be able to view the menus of the restaurants in the hotel.
- Guests will be able to view information about upcoming events and activities in the hotel.
- Guests will be able to reserve a table at a restaurant, chaise longues at the beach.
- Guests will be able to request room cleaning, food, and drink to their room.
- Guests will be able to book slots for facilities of registered property.
- Guests will be able to request medical personnel to the room.
- Guests will be able to get 24/7 help from the chatbot.
- Guests will be able to post comments/feedbacks.
- The administration should be able to provide Guests with the QR Code generated using the website developed for Administration.
- The administration will be able to view requests done by guests.
- The administration will be able to manage requests done by guests.
- The administration will be able to supply menu information.
- The administration will be able to supply activities and events information.
- The administration will be able to view all the reservations done by Guests.
- The administration will be able to view the statistics on the Guests' choices.
- The administration will be able to view Guests' reviews.

3.3. Non-functional Requirements

Usability:

- The system will be implemented in English.
- Both website and application will have a user-friendly interface, pleasing to the eye and easy to use.
- To maximize user domain, IOS and Android versions of the application will be developed.

Privacy:

- To protect privacy, the system will not store user data.

Response Time/Performance:

- The system will have to provide fast and accurate interactions between guests, bots and facility staff.
- Managing TV/AC/Lights should not take more than 2 seconds.
- The system should be able to handle requests from several users at the same time.

- The system should be well functioning 24/7.

Maintainability:

- The system will be developed in a way to be flexible to changes in the future.

3.4. Pseudo Requirements

3.4.1 Technical Constraints

- Depending on user type our system will have mobile and web-based implementations. While administrators will use the web-based system, guests will be provided with the mobile environment.

3.4.2 Social Constraints

- Service will provide social interactions between guests, bots and facility staff.
- Guests will be able to send anonymous feedback to improve the service of the facilities registered in the system.
- Credentials of the guests will not be stored.
- A minimal amount of facility data will be stored in order to provide customer support.

3.4.3 Implementation Constraints

- React framework will be used to implement the frontend of a web application for administrators of the registered facilities [7].
- React Native will be used to develop cross-platform mobile applications for IOS and Android devices [8].
- Git and GitHub will be used for version control and collaborative work respectively.
- MySQL database will be used to store facility data.
- Firebase Real-time Database will be used to provide interaction between guests and facility staff.
- Python SciPy toolkit will be used in order to implement the main features related to chatbot [9].
- IBM Cloudant will be used to store real-time IoT data [10].
- RapidAPI will be used in order to implement QR code-related features.

3.4.4 Sustainability Constraints

- The system will be constantly updated to reflect user feedbacks and

requested services by guests.
- API integration will be adapted to changes in used APIs.

3.4.5 Economic Constraints

- Amazon Web Services will be used for hosting the server of the project. Prices will be taken into account [11].
- Since the Firebase Database provides a limited amount of storage for free, prices will need to be taken into consideration as the size of the data to be stored increases.[12]
- Pricing for IBM Cloudant and RapidAPI will need to be taken into consideration.[13,14]
- Use of open source frameworks and libraries will be free of charge.

3.4.6 Security Constraints

- The system will not store user data to protect privacy.
- Shareable string codes and QR codes will be globally unique to the whole system. Each QR code is going to be scanned only once.
- All kinds of codes in the system will be deactivated after the expiration date set by the facility administrator.
- In case of failed attempt to gain access to IoT devices of the particular facility, the device that had been used to gain access will be locked out and facility staff will be notified.
- Connections to the website will be protected with TLS certificate.

3.4.7 Language Constraints

- The user interface for the product will be in English since it is considered to be a global language.
- Depending on the size and locational majority of the user base new languages may be added in the feature.

3.5. System Model

3.5.1 Scenarios

3.5.1.1 Guest side

Sign Up

Use Case: Guest Sign Up

Primary Actor: Guest

Entry Conditions:

- User should be on the launched activity of mobile application.

Exit Conditions:

- User should return to the home screen of the mobile phone.
- User proceeds to the main activity of the mobile application.

Main Flow of Events:

1. User should fill in the *username*/*email* and *password* fields.
2. User confirms his/her email address outside of the application.
3. System adds the user to the database if user confirms email.

Alternative Flow of Events:

1. User decides to return to the home screen of the mobile phone:
    a. User presses “Return” button or exits the application.
    b. Application is terminated.
2. User cannot proceed to the main activity of the mobile application:
    a. User did not confirm their email address.

Sign In

Use Case: Guest Sign In

Primary Actor: Guest

Entry Conditions:

- User should be on the launched activity of mobile application, when signing in for the first time.
- User will be signed in automatically unless they were not already signed in.

Exit Conditions:

- User should return to the home screen of the mobile phone.
- User proceeds to the main activity of the mobile application, when signing in for the first time.

Main Flow of Events:

1. User should fill in the *username*/*email* and *password* fields.
2. System checks the user if they exist in the database:
    a. User exists in the database and system lets user to go to the main activity of the mobile application.
    b. User does not exist in the database. Sign Up required.

Alternative Flow of Events:

1. User decides to return to the home screen of the mobile phone:
    a. User presses “Return” button or exits the application.

b. Application is terminated.

2. User cannot proceed to the main activity of the mobile application:

a. User did not create any account and sign up is required.

Scan QR code

Use Case: Scan QR code

Primary Actor: Guest

Entry Conditions:

- User should be on the main activity of mobile application.

Exit Conditions:

- User should return to the main activity of the mobile application.
- User proceeds to the main activity of the mobile application, with the hotel room linked to the guest.

Main Flow of Events:

1. User should scan the QR code in order to link the application to his current hotel room if exists.
2. System checks the QR code:

a. QR code confirmed and system lets user to go to the room activity of the mobile application.

b. QR code rejected.

Alternative Flow of Events:

1. User decides to return to the main activity of the mobile phone:

a. User presses “Return” button.

Read shared code

Use Case: Scan QR Code

Primary Actor: Guest

Entry Conditions:

- User should be on the main activity of mobile application.

Exit Conditions:

- User should return to the main activity of the mobile application.
- User proceeds to the main activity of the mobile application, with the hotel room linked to the guest.

Main Flow of Events:

1. User should fill in the *shared code* field. Code is shared by the guest's room mate, who scanned the QR code first.
2. System checks if the shared code valid:
    a. Shared code is valid and guest get linked with the same room as their room mate/s.
    b. Shared code is rejected.

Alternative Flow of Events:
1. User decides to return to the main activity of the mobile phone:
    a. User presses "Return" button.

Hotel Activity

Use Case: Hotel Activity

Primary Actor: Guest

Entry Conditions:
- User should be on the hotel room activity of the mobile application.

Exit Conditions:
- User should return to the main screen of the mobile phone.

Main Flow of Events:
1. User should navigate through the hotel activity menu.
2. User should be able to control their room when pressed my room button:
    a. User should be able to turn on/off lights, TV, AC.
    b. User should be able to control TV and AC - channels, volume and temperature, fan speed, respectively.
3. User should be able to view the overall information about hotel by pressing hotel info button.
4. User should be able to view all the events that are happening in the hotel by pressing the events button.
5. User should be able to view information about the restaurant of their liking by pressing restaurants button:
    a. User should be able to view information about the restaurant.
    b. User should be able to see the online menu of the specified restaurant.
6. User should be able to book places at the different facilities of their liking by pressing reservations button:
    a. User should be able to book the available places at the restaurant.
        i. User should be able to specify the order for the booked place and time if desired.
    b. User should be able to book the available chaise-lounges on the beach.

c. User should be able to book the available time-slots for such hotel facilities as spa, gym, sport courts.

7. User should be able to chat with the chatbot to ask all the questions including FAQ, or information about facilities inside the hotel as well as information about shops outside the hotel - 24/7.
8. User should be able to request service into the room, including room cleaning service, medical service or food service by clicking the request button.
9. User should be able to view the map of the hotel and its surroundings by pressing the map button.
10. User should be to call the hotel help desk at anytime by pressing the contact us button.

Alternative Flow of Events:

1. User decides to return to the main activity of the mobile phone:
    a. User presses “Return” button.

3.5.1.2 Administration Side

<u>Sign Up</u>

Use Case: Administration Sign Up

Primary Actor: Administration

Entry Conditions:

- User should be on the registration page of the website.

Exit Conditions:

- User proceeds to the main page of the website.
- User closes the website.

Main Flow of Events:

1. User should fill in the *business email* field.
2. System gets a request from a potential hotel to register.
3. System verifies and validates the request from the hotel:
    a. If the hotel that submitted the form exists and legal, system sends encrypted key to the representative of the hotel. Hotel administration can register using that key only.
    b. If the hotel that submitted the form does not exist or is not legal, system ignores the form.

Alternative Flow of Events:

1. User decides to close the website.
2. User cannot proceed to the main activity of the mobile application:
    a. User was not approved by the system yet.

Sign In

Use Case: Administration Sign In

Primary Actor: Administration

Entry Conditions:

- User should be on the registration page of the website.
- User will be signed in automatically unless they were not already signed in.

Exit Conditions:

- User proceeds to the main page of the website.
- User closes the website.

Main Flow of Events:

1. User should fill in the *business email* and *password* fields. Password was sent by the system, and is encrypted.
2. System checks the user if they exist in the database:
    a. User exists in the database and system lets user to go to the main page of the website.
    b. User does not exist in the database. Sign Up required.

Alternative Flow of Events:

1. User decides to close the website.
2. User cannot proceed to the main activity of the mobile application:
    a. User did not create any account for the hotel and sign up is required.

Hotel Page

Use Case: Hotel Page

Primary Actor: Administration

Entry Conditions:

- User should be on the main page of the website.

Exit Conditions:

- User closes the website.

Main Flow of Events:

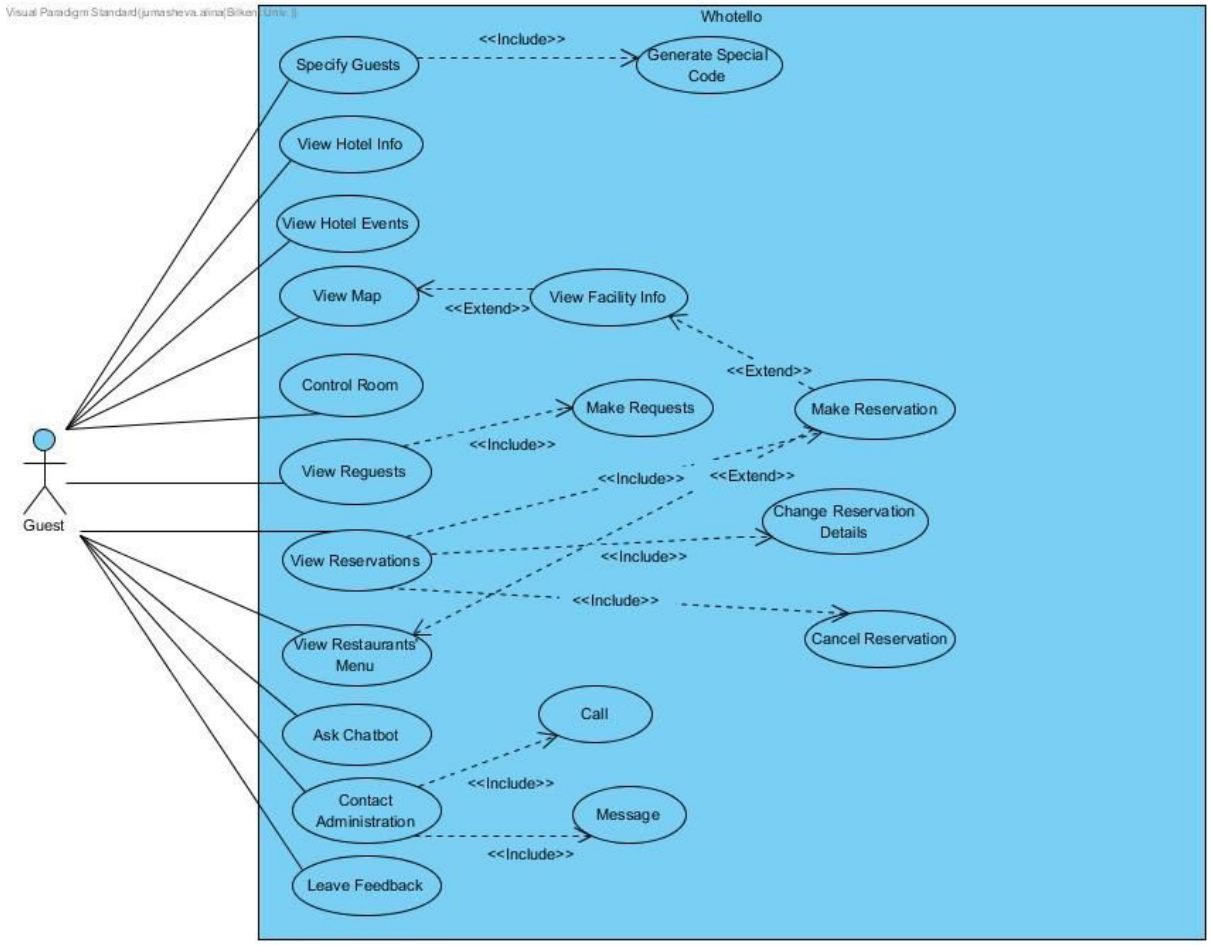
1. User should navigate through the hotel page menu.

2. User should be able to write, change and update the overall information about hotel by pressing hotel info button.
3. User should be able to create/post all the events that are happening in the hotel by pressing the events button.
4. User should be able to write, change or update information about the restaurants, add or delete restaurants by pressing restaurants button. The menu is provided in the information. System parses the menu information and adds it to the database.
5. User should be able to see the reservations made by their hotel guests at different facilities by pressing reservations button:
    a. User should be able to manage the available and reserved places at the restaurant.
    b. User should be able to manage the available and reserved chaise-lounges on the beach.
    c. User should be able to manage reserved and available time-slots for the hotel facilities as spa, gym, sport courts.
        i. User should be able to add or delete different facilities in the hotel.
6. User should be able to view, maintain and manage the request services into the room, including room cleaning service, medical service or food service, by clicking the request button.
7. User should be able to view the statistics of the hotel by pressing the statistics button. Statistics are given for all the facilities inside the hotel, as well as using the machine learning, hints and advices are generated that suggest the administration what to add/delete from the hotel facilities in order to maximize the hotel revenue and guest satisfaction.
8. User should be to view the guest reviews that are available online on hotel's social media/hotel website.

Alternative Flow of Events:
1. User decides to close the website.

3.5.2 Use case model

3.5.2.1.Use Case Diagram #1: Guest



Specify Guests: Guest can specify others guests of the same room in order to share the control of the room and use the app. In order to share room with other guests he/she will **Generate Special Code** for them.

View Hotel Info: Guest can view hotel information on certain page.

View Map: Guest can view hotel map, and by clicking on certain facilities on the territory of the hotel he/she can **View Facility Info** and **Make Reservation** to that facility.

Control Room: Guest can manage air conditioners, television, lights in their rooms and other available available equipment.

View Requests: Guest can view requests he/she done, and **Make Requests** on specific page**.**

View Reservations: Guest can view requests he/she done, and **Make Reservation, Change Reservation Detail or Cancel Reservation** on specific page**.**

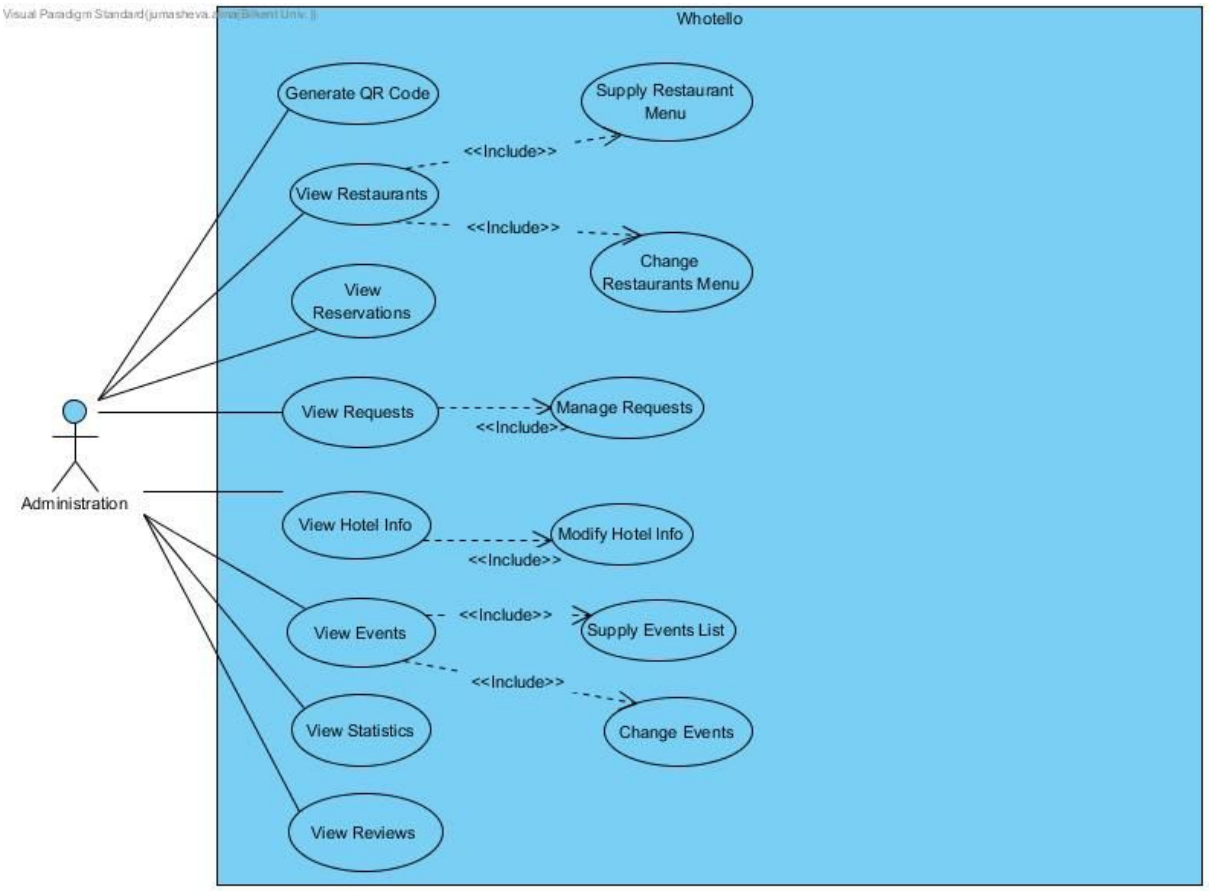
View Restaurant Menu: Guest can view restaurant menu and if he/she wishes he/she can **Make Reservation.**

Ask Chatbot: Guest can get 24/7 help from the chatbot.

Contact Administration: Guest can contact Administration by **Calling or Messaging.**

Leave Feedback: Guest can post comments/feedbacks on the specific page.

3.5.2.2 Use Case Diagram #2: Administration



Generate QR Code: Administration can generate QR Code for guests in order to provide access to their rooms for their app.

View Restaurants: Administration can view Restaurants and **Supply Restaurant Menu** or **Change Restaurant Menu** on a specific page.

View Reservations: Administration can view reservations done by the guests.

View Requests: Administration can view requests done by guests and **Manage Requests.**

View Hotel Info: Administration can view hotel information shown to the guests and can **Modify Hotel Info.**

View Events: Administration can **Supply Events List** for the guests to know which events and activities will be at their hotel and they can **Change Events** to provide correct information**.**

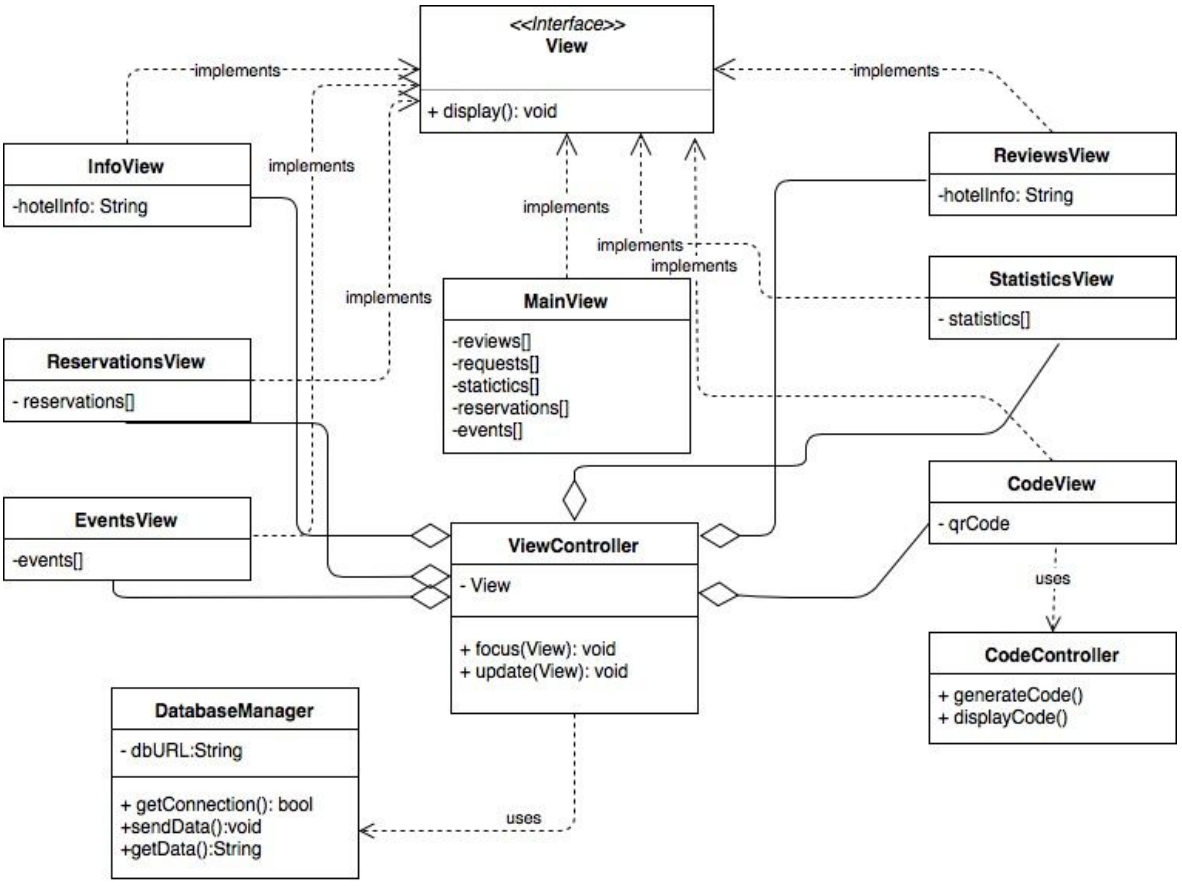
View Statistics: Administration will be able to view the statistics on the Guests' choices on the specific page.

Vlew Reviews: Administration will be able to view Guests' reviews on the specific page.

3.5.3 Object and class model

- Object and Class model for Mobile Application

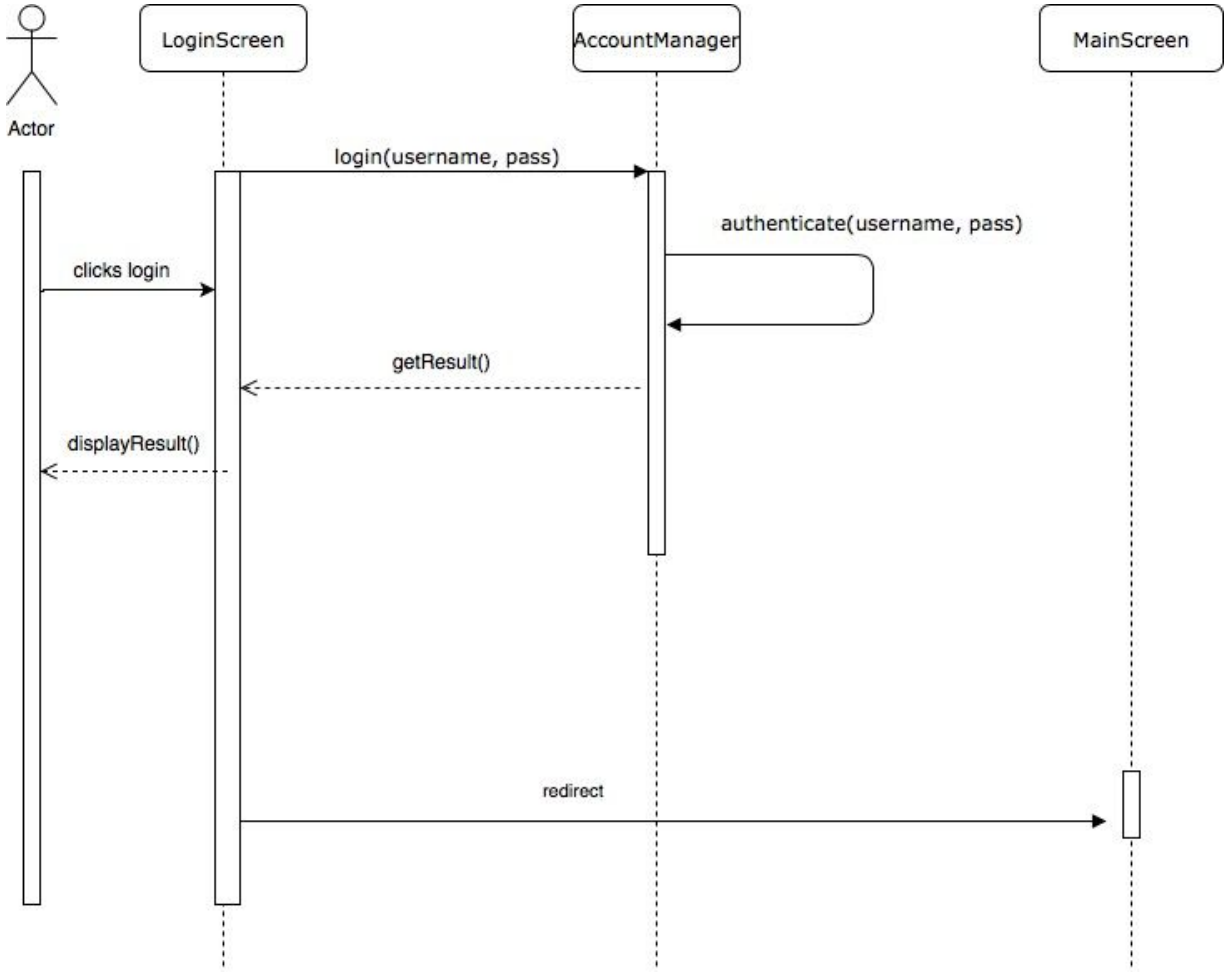
- Object and Class model for Web Application


<<Interface>>
View
+ display(): void
implements
implements
InfoView
-hotelInfo: String
implements
implements
implements
implements
ReviewsView
-hotelInfo: String
StatisticsView
- statistics[]
MainView
-reviews[]
-requests[]
-statictics[]
-reservations[]
-events[]
ReservationsView
- reservations[]
EventsView
-events[]
ViewController
- View
+ focus(View): void
+ update(View): void
CodeView
- qrCode
uses
CodeController
+ generateCode()
+ displayCode()
DatabaseManager
- dbURL:String
+ getConnection(): bool
+sendData():void
+getData():String
uses

3.5.4 Dynamic models

3.5.4.1 Sequence Diagrams

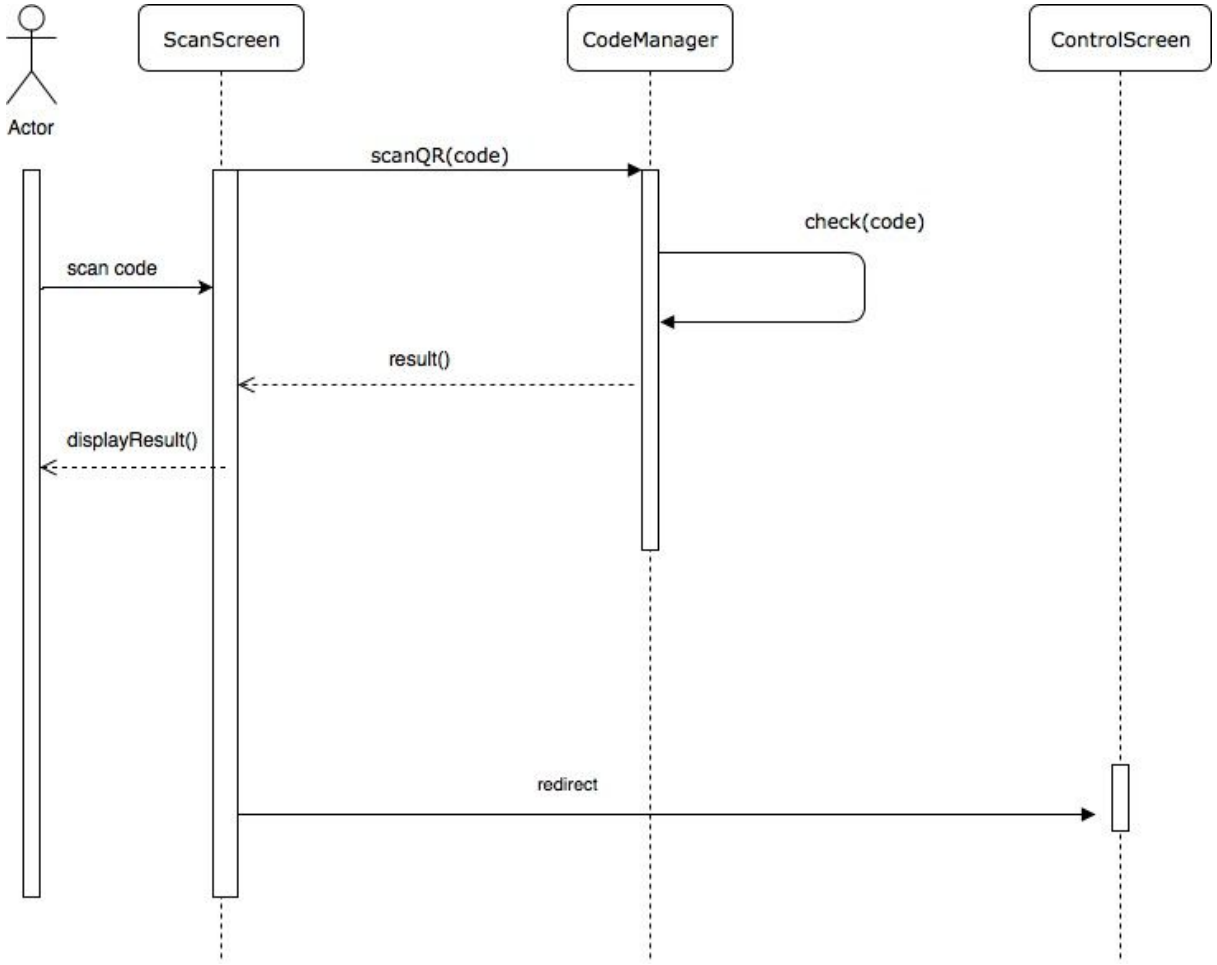
- Login Diagram



Description:

In this diagram Actor clicks login button after entering their credentials. AccountManager checks credentials and returns boolean value. User is either displayed error message or redirects to MainScreen according to authentication result.

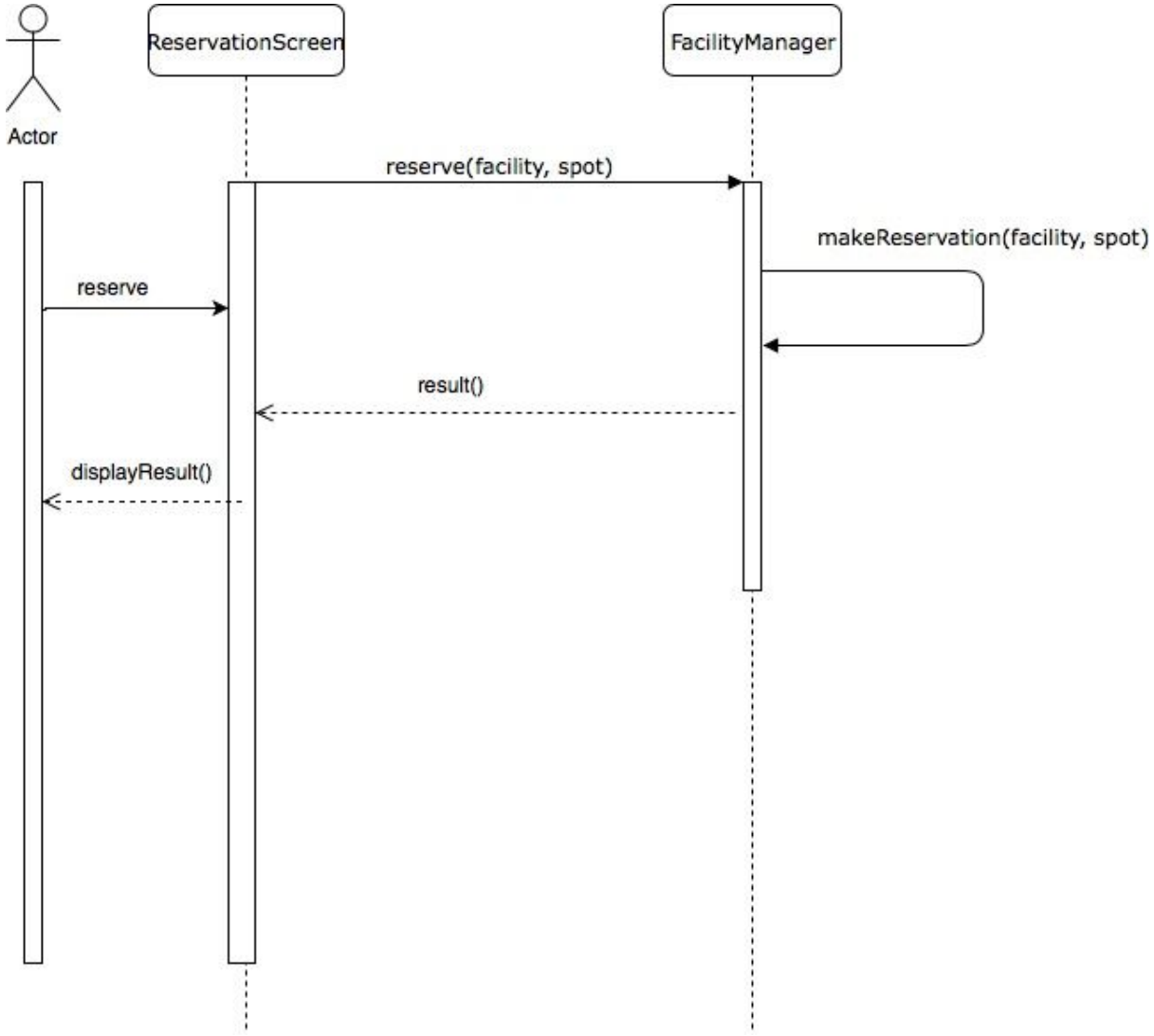
- Scan QR Code Diagram



Description:

In this diagram CodeManager checks QR code after user scans one. Actor is redirected to ControlScreen according to authentication result.

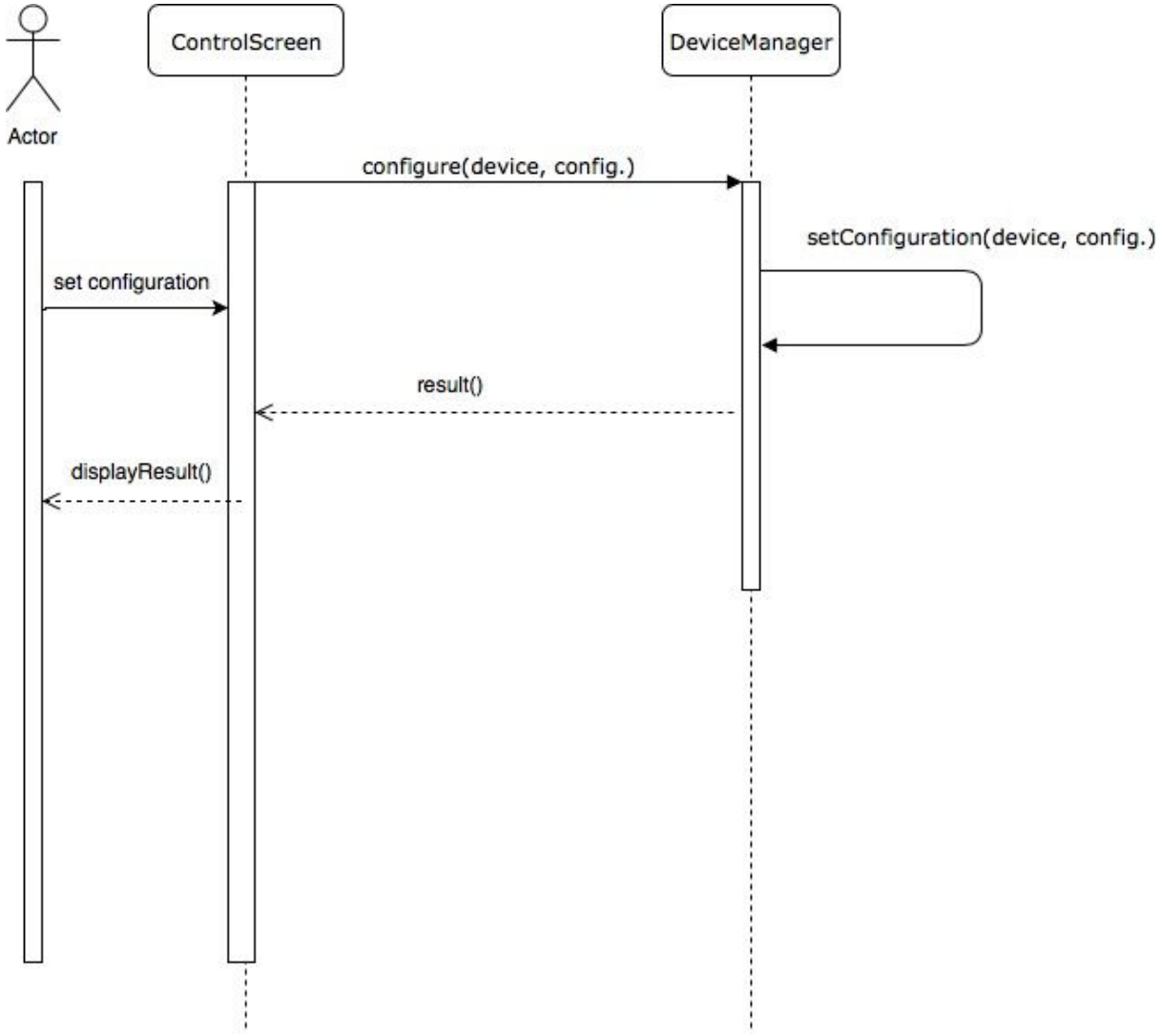
- Reserve Facility Diagram



Description:

In this Diagram Actor attempts to reserve a spot for a particular facility. FacilityManager handles the request and returns result. Success or failure is displayed according to the returned result.

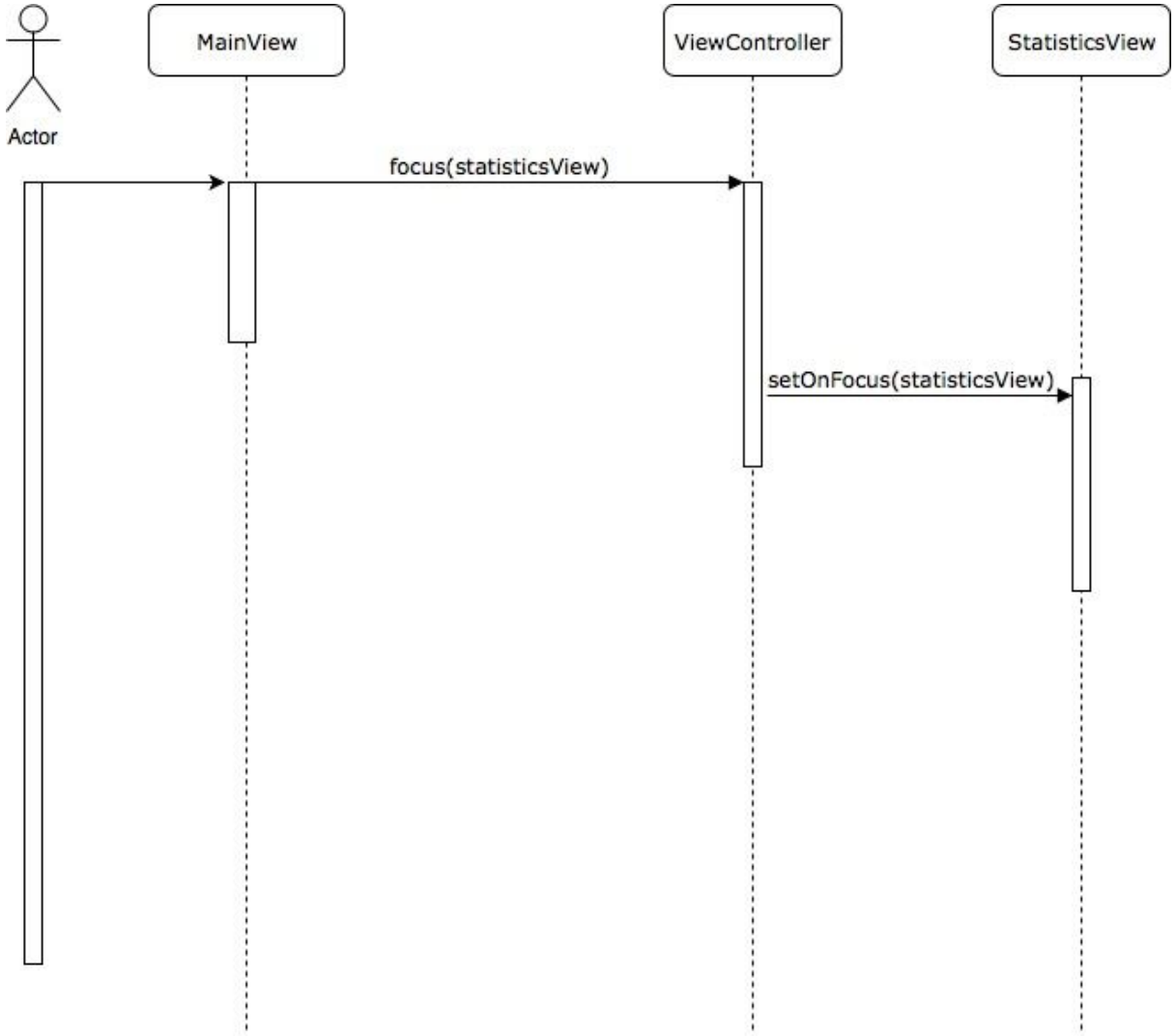
- Set Device Configuration Diagram



Description:

In this diagram, Actor attempts to change configurations of a particular device in their room. DeviceManager handles the request and ControlScreen is updated according to diplayed result.

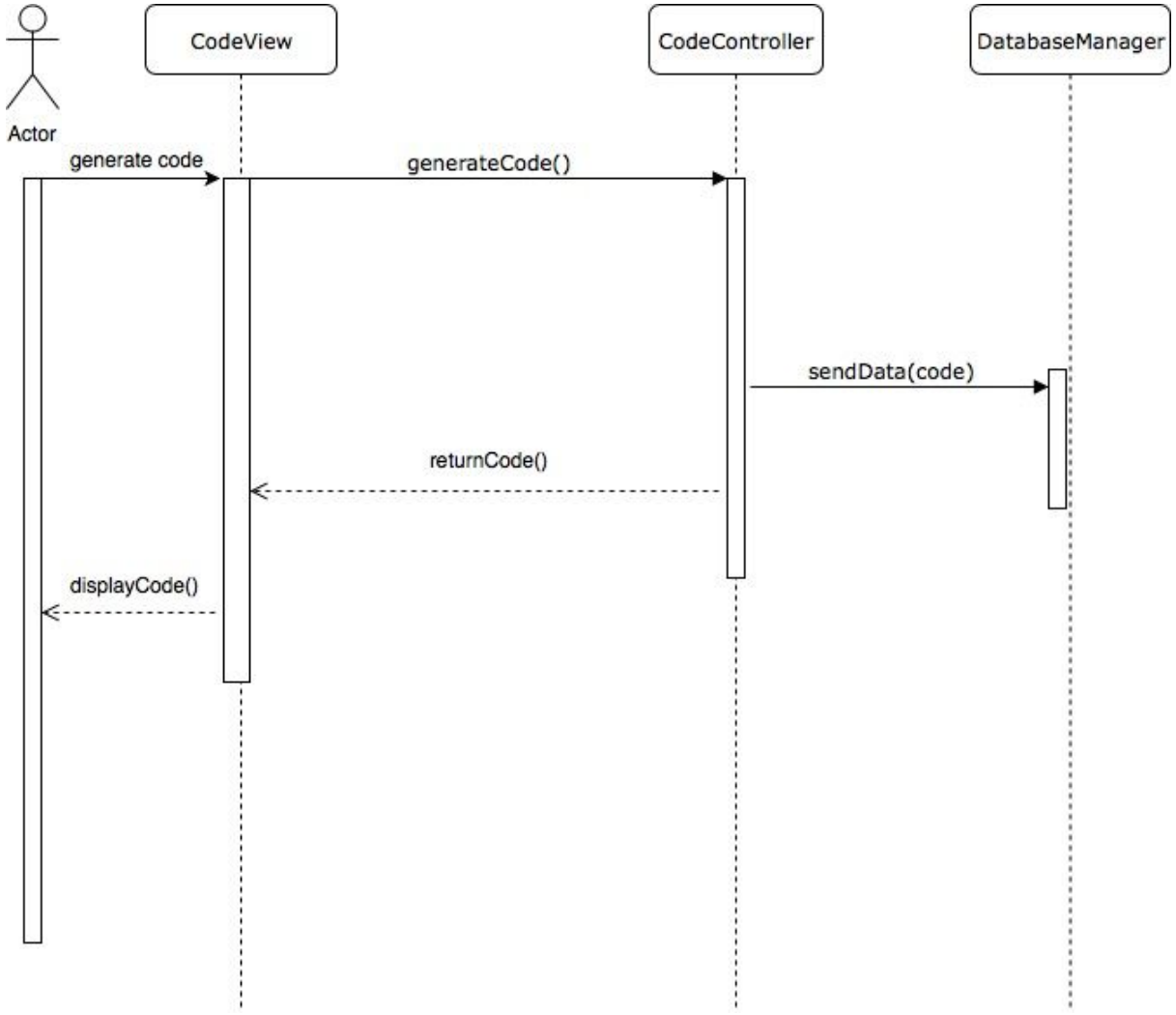
- View Statistics Diagram



Description:

In this diagram ViewController redirects user to StatisticsView once s/he clicks on statistics from the MainView.

- Generate QR Code Diagram

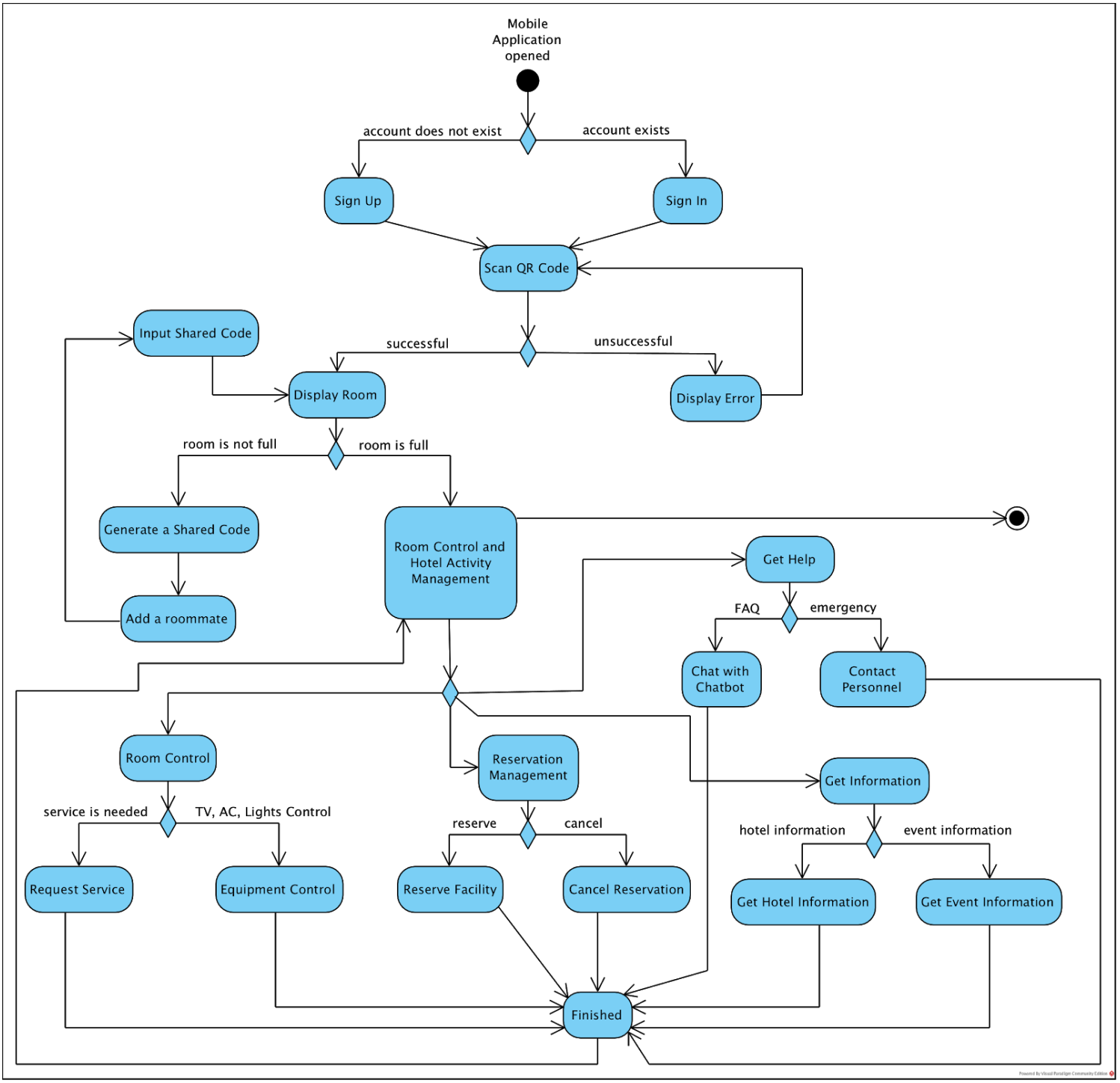


Description:

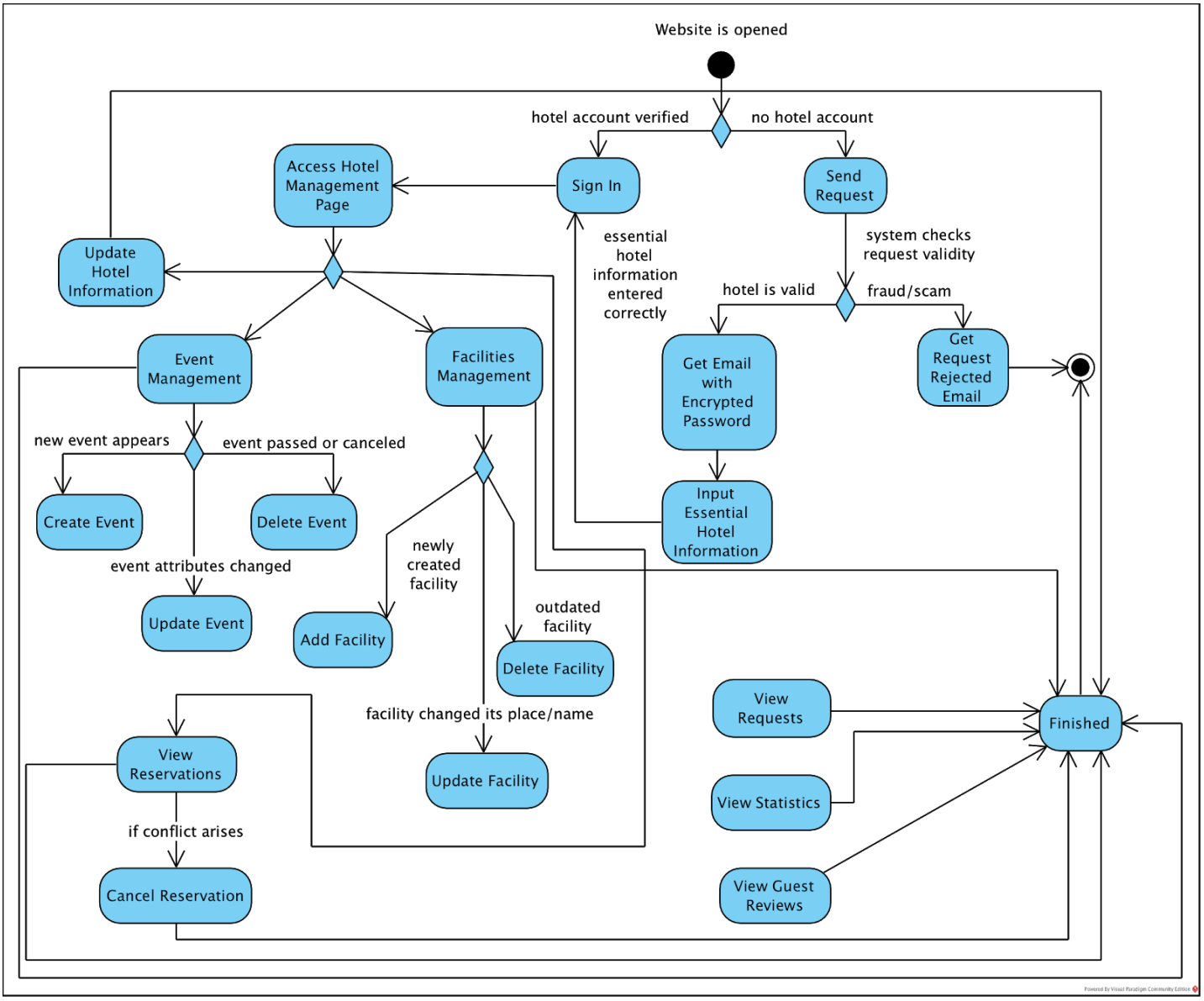
In this diagram CodeController generates QR code and sends it to Database after Actor requests code generation. Generated code is displayed to the user once CodeController returns it.

3.5.4.2 Activity Diagrams

3.5.4.2.1 Guest Activity Diagram



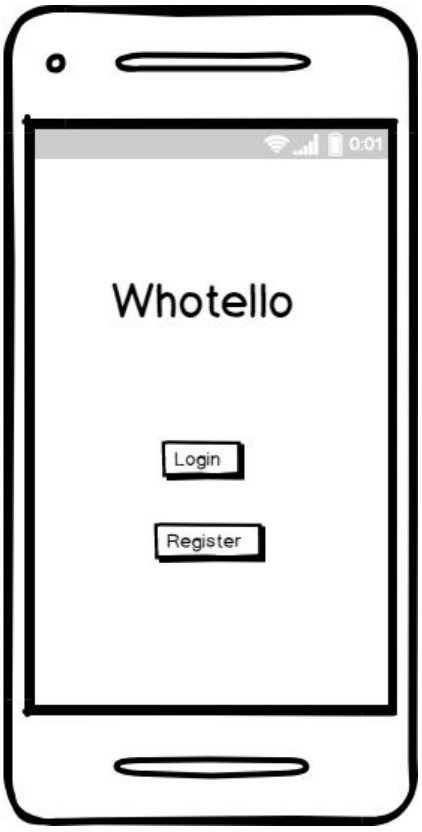
3.5.4.2.2 Administration Activity Diagram



3.5.5 User interface - navigational paths and screen mock-ups

Guest side

App Welcome screen



This page is ‘Welcome Screen’ which user sees when he/she launches the application for the first time. Here guest can choose to sign up for the first time to use this app at the hotels, or to sign in with registered credentials so that app can use his/her preferences used before in some hotel for the new hotel.

Register screen

This page where guest can register to the system by simply typing in his/her name, existing email address and choose a password for security purposes and further usage of the application.

Code Confirmation

On Code Confirmation Page User needs to enter confirmation code sent to his/her email address in order to use the application.

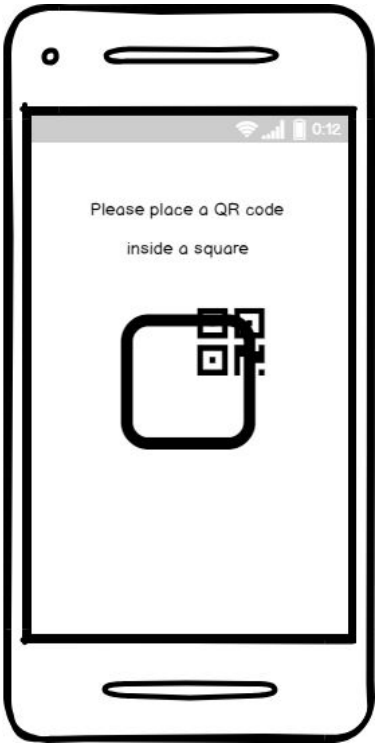
Login screen

For existing users, they can simply log in with their registered email addresses and corresponding passwords. The log-in screen won't be asked for each time the users open the application.

Synchronization with hotel screen

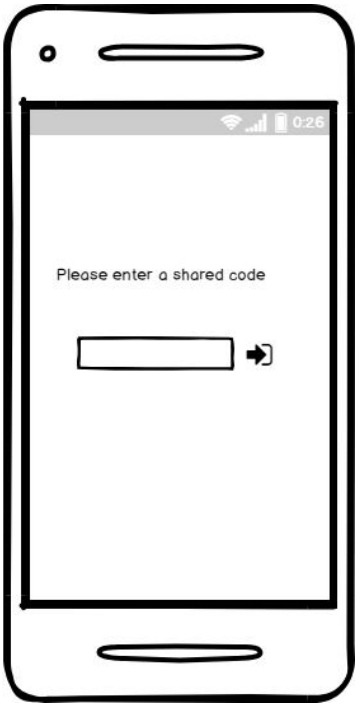
Welcome screen after signing in. In order to use an app he/she needs to use QR Code given by administration on check-in or use a shared code shared by the user who used QR Code to use an app for specific room.

Log in with QR Code



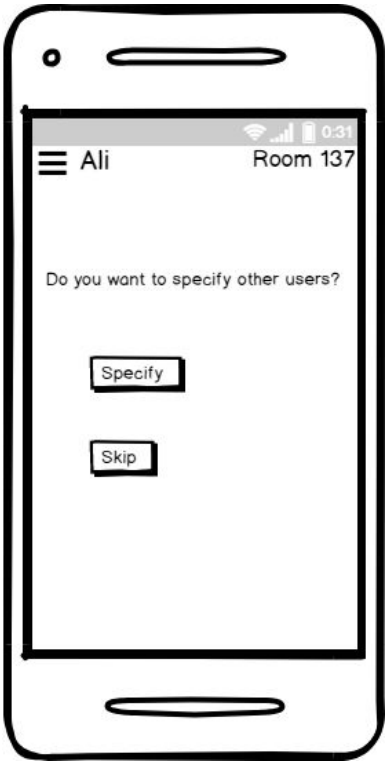
Guests should use the QR Code given by the administration while check-in in order to use an app at that hotel.

Log in with Shared Code



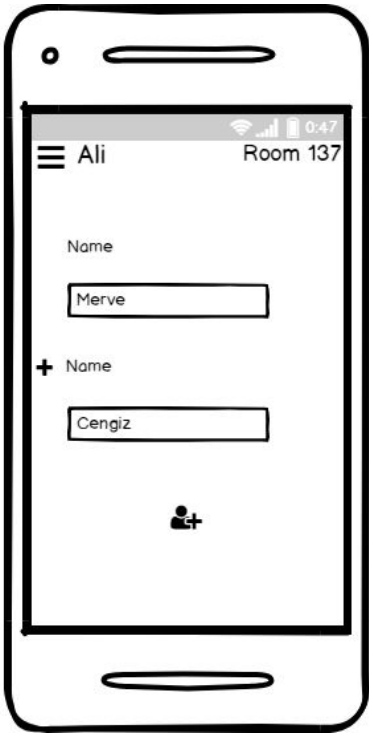
Guests should use code shared by the guest of that room who firstly synchronized his/her app to that room using QR Code.

Screen after synchronization with hotel



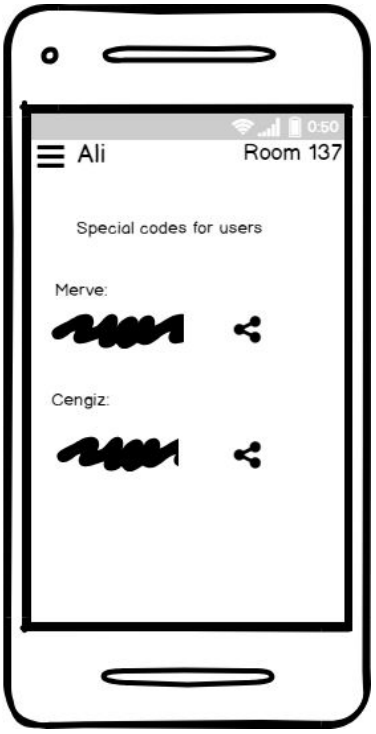
This screen shown only once to the user after he/she logged in for the first time. This screen asks whether guest wants to specify other users. He/She can specify users later from menu clicking on the top-left bar icon.

Screen to specify users



At this page, guest can specify other users of the same room.

Special codes

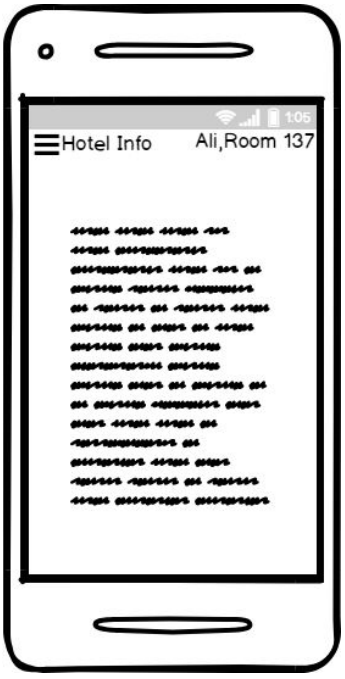


Special codes generated for other users. Merve and Cengiz will need these codes to use an app and to use it under their names.

Home Page

Home Page showing main functions that app can provide. More facilities such as, specify users, leave feedback, etc. can be found by clicking on tab on the top-left of screen.

Hotel Info page

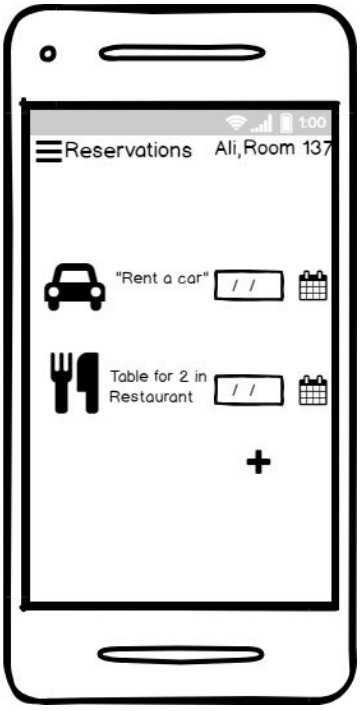


Here is the information that administration decides to show on this page about hotel.

Hotel map

On this page, user can see the map of the hotel with all the facilities on the territory of the hotel. By clicking on the facility he/she can see information about that facility or make a reservation.

Reservations

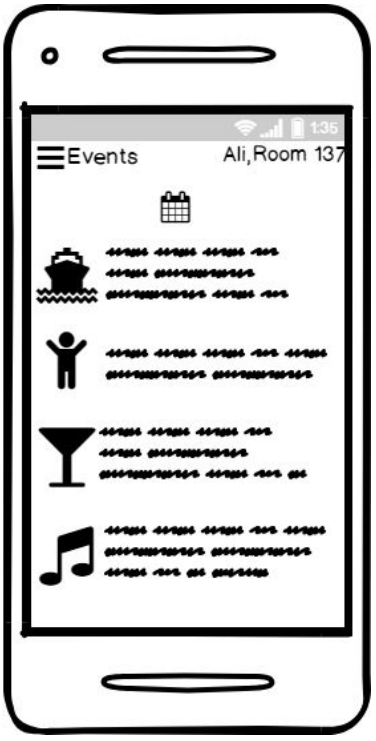


This is the page of reservations made by the user. Here he/she can change reservation details, cancel or make more reservations.

Requests

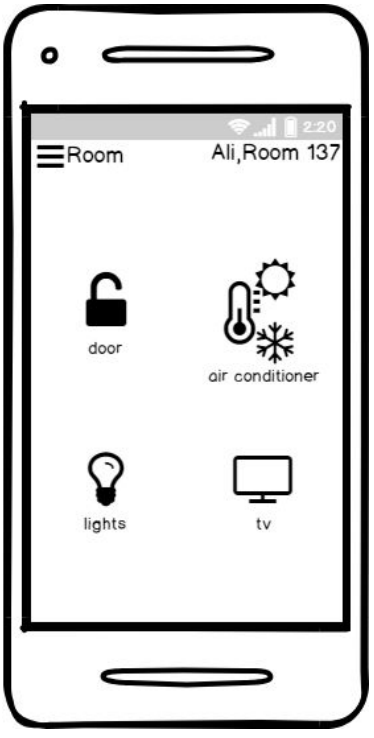
On this screen guest can see requests he done.Here he/she can make any other request such as the cleaning of the room, food being brought to the room, medical personnel, etc.

Events



On this page guest can view information about upcoming events and activities in the hotel.

Room control



This is ‘Room Control’ page.Control of the room includes managing the air conditioners, television, lights and also the heating system if the latter is available.

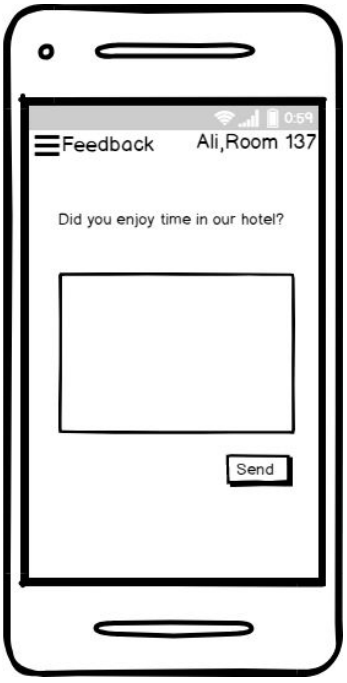
Contact Hotel

On this page user can call the administration or leave a message to them.

Chatbot

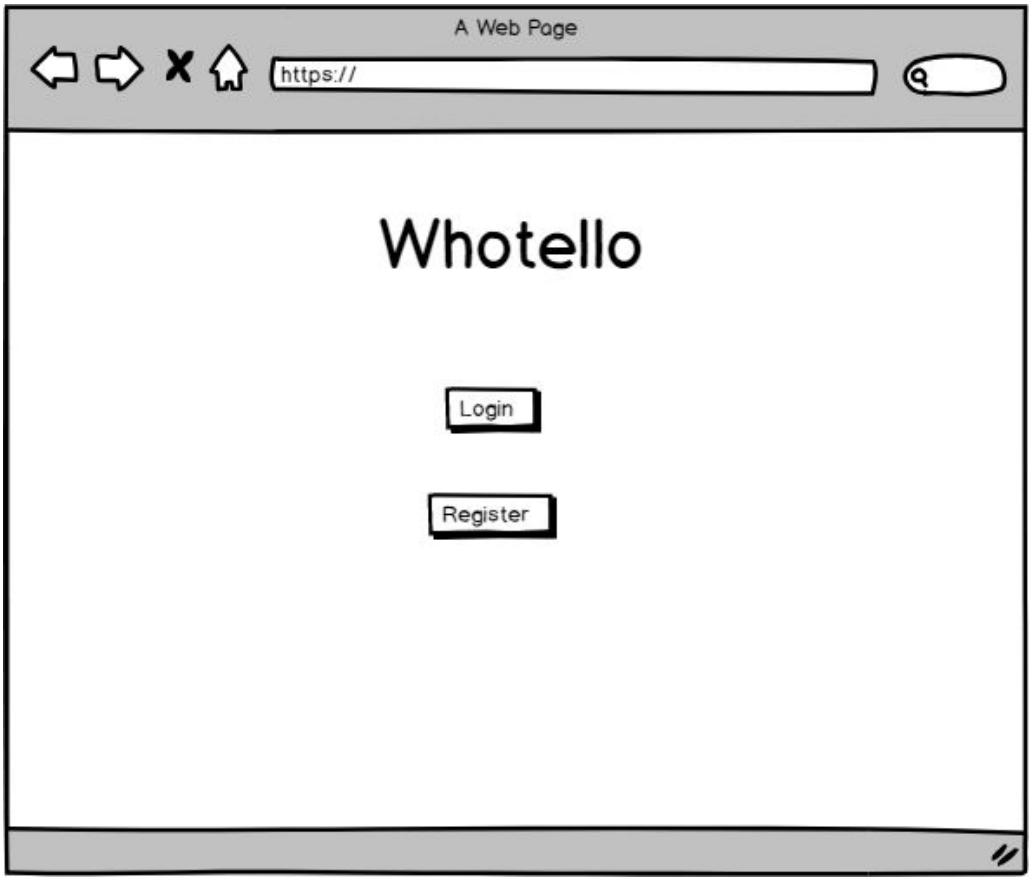
On this page guest can get 24/7 help from a chatbot which will be able to answer most of the guest questions and provide the guests with all the needed number and information about the hotel they are staying in.

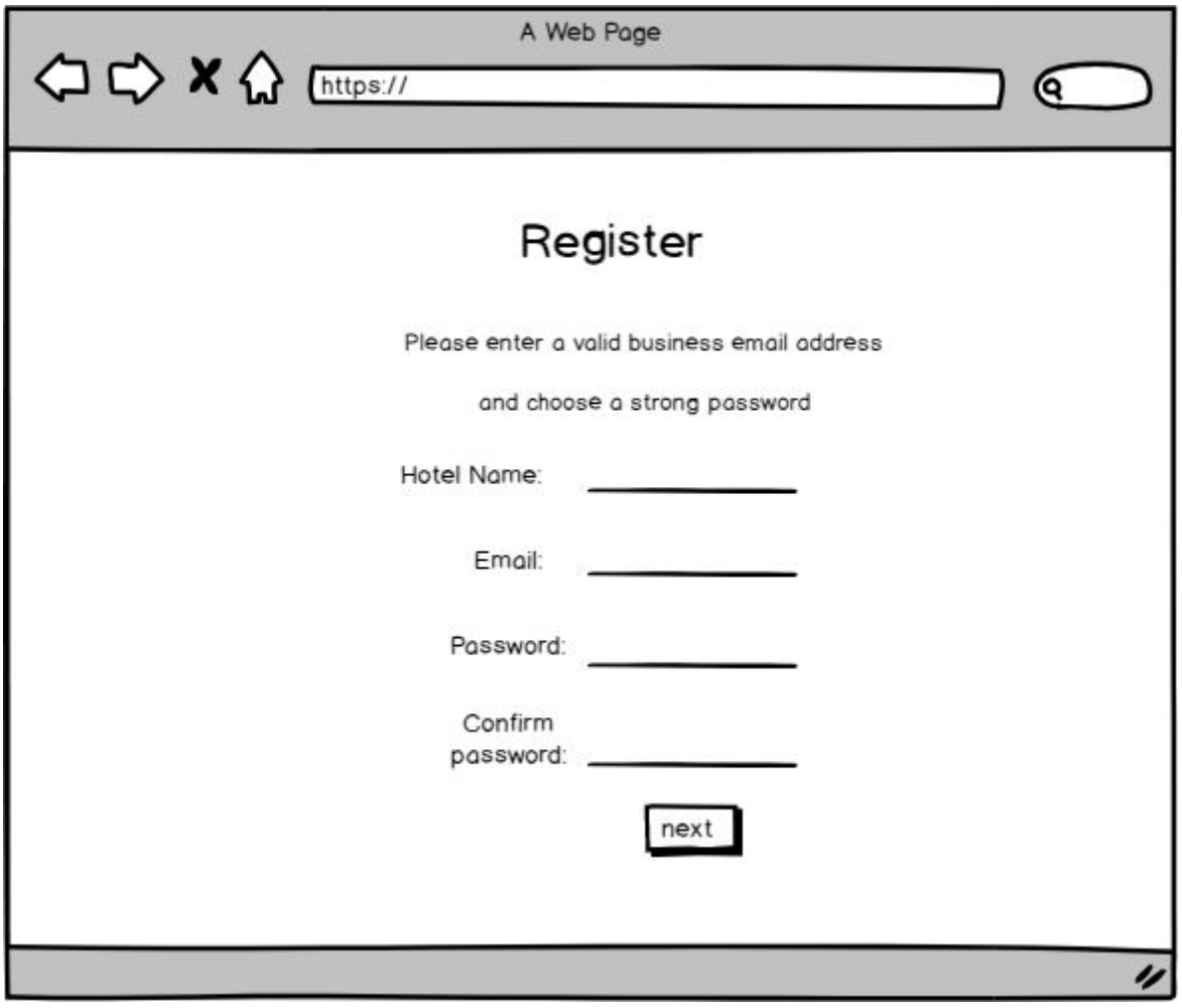
Feedback from a guest

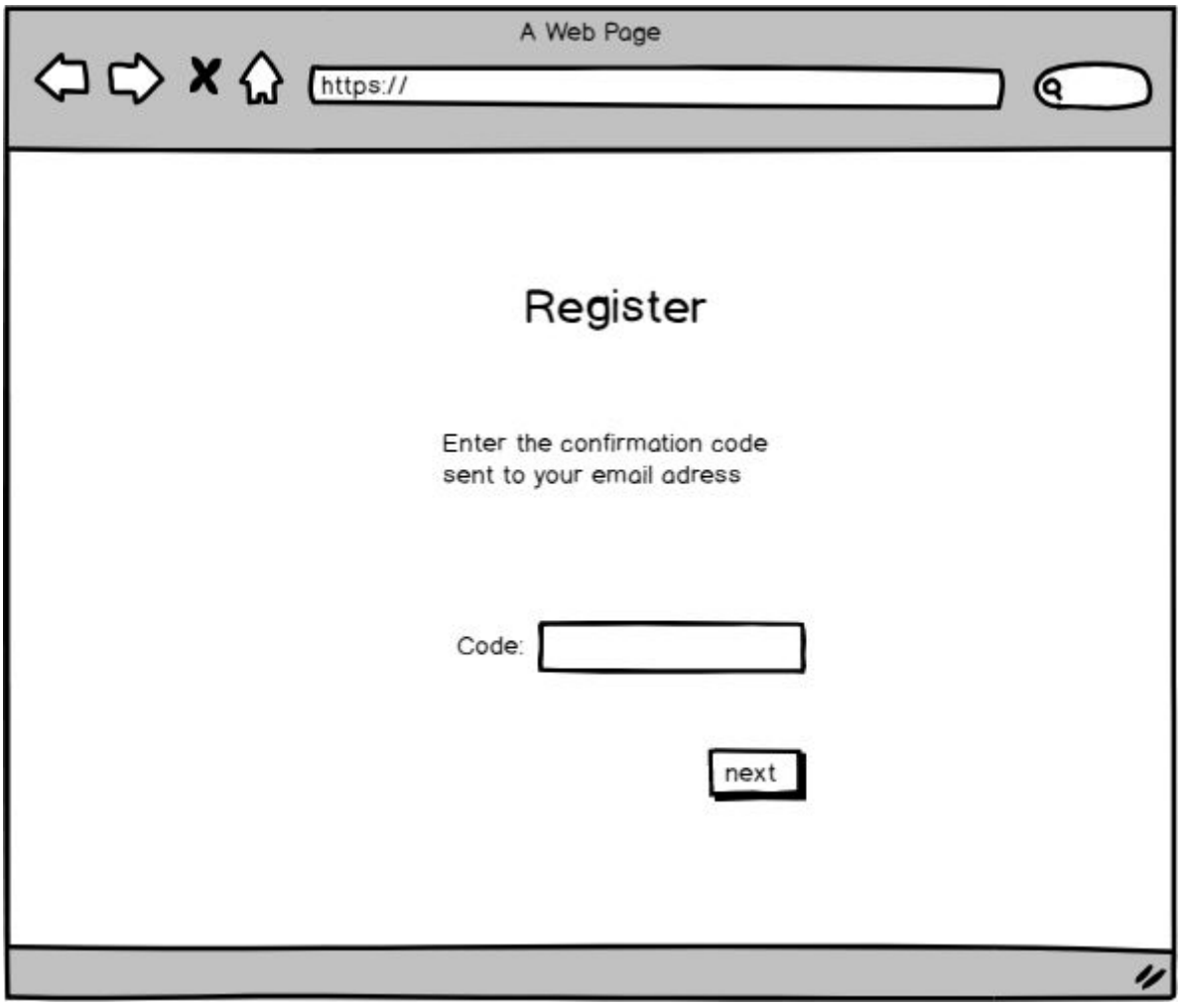


Guests will be able to send feedback to improve the service of the facilities registered in the system.

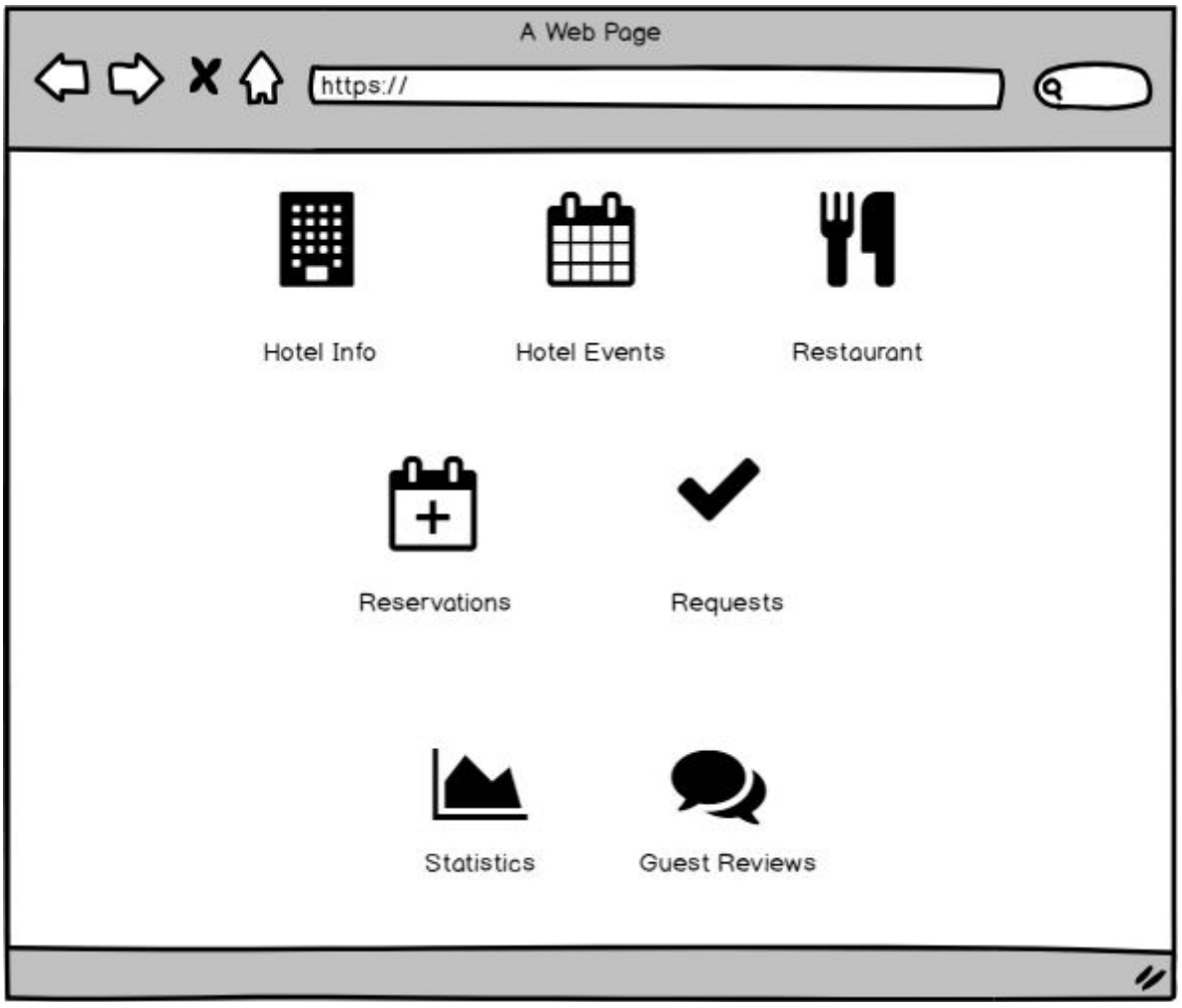
Administration side

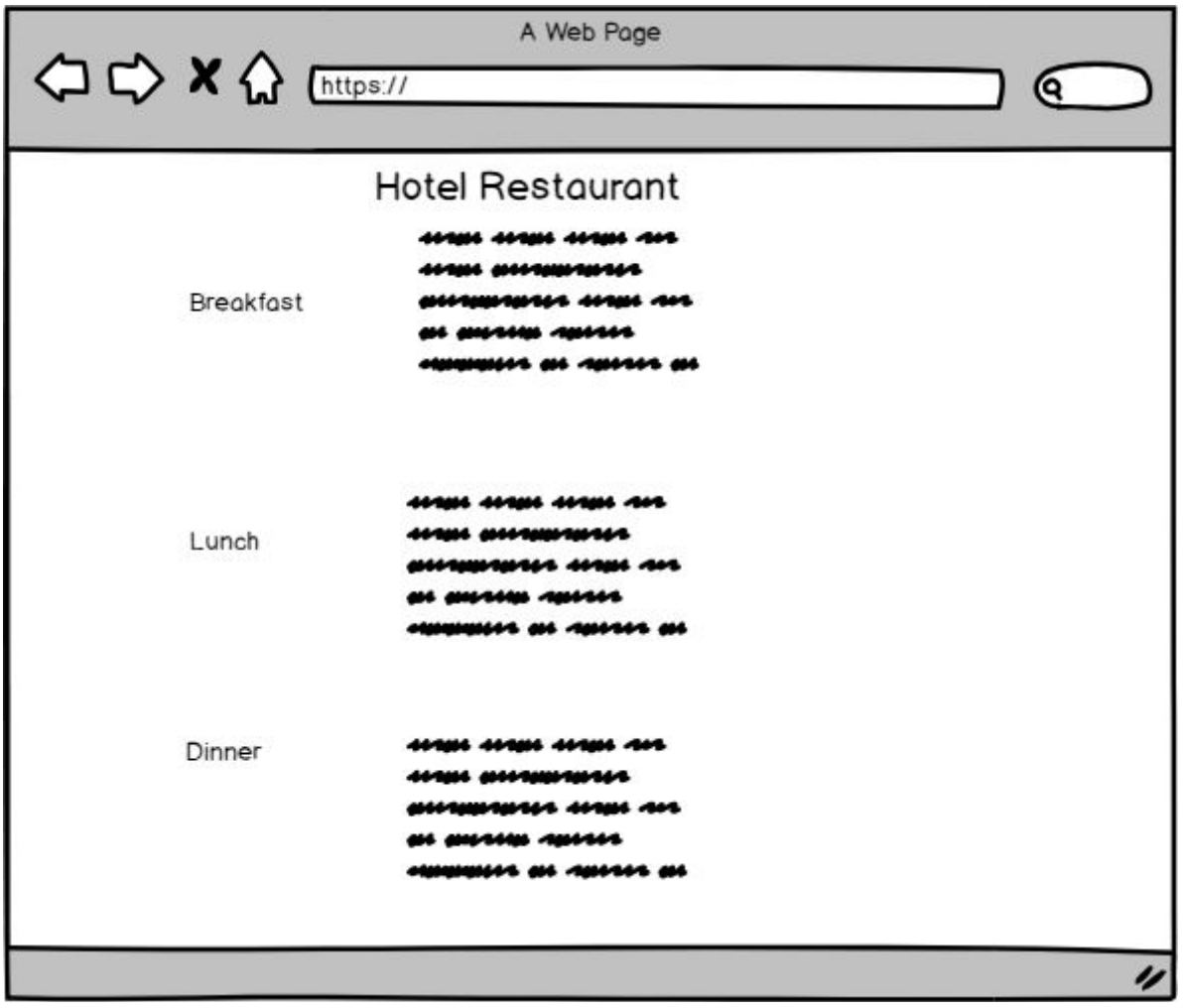


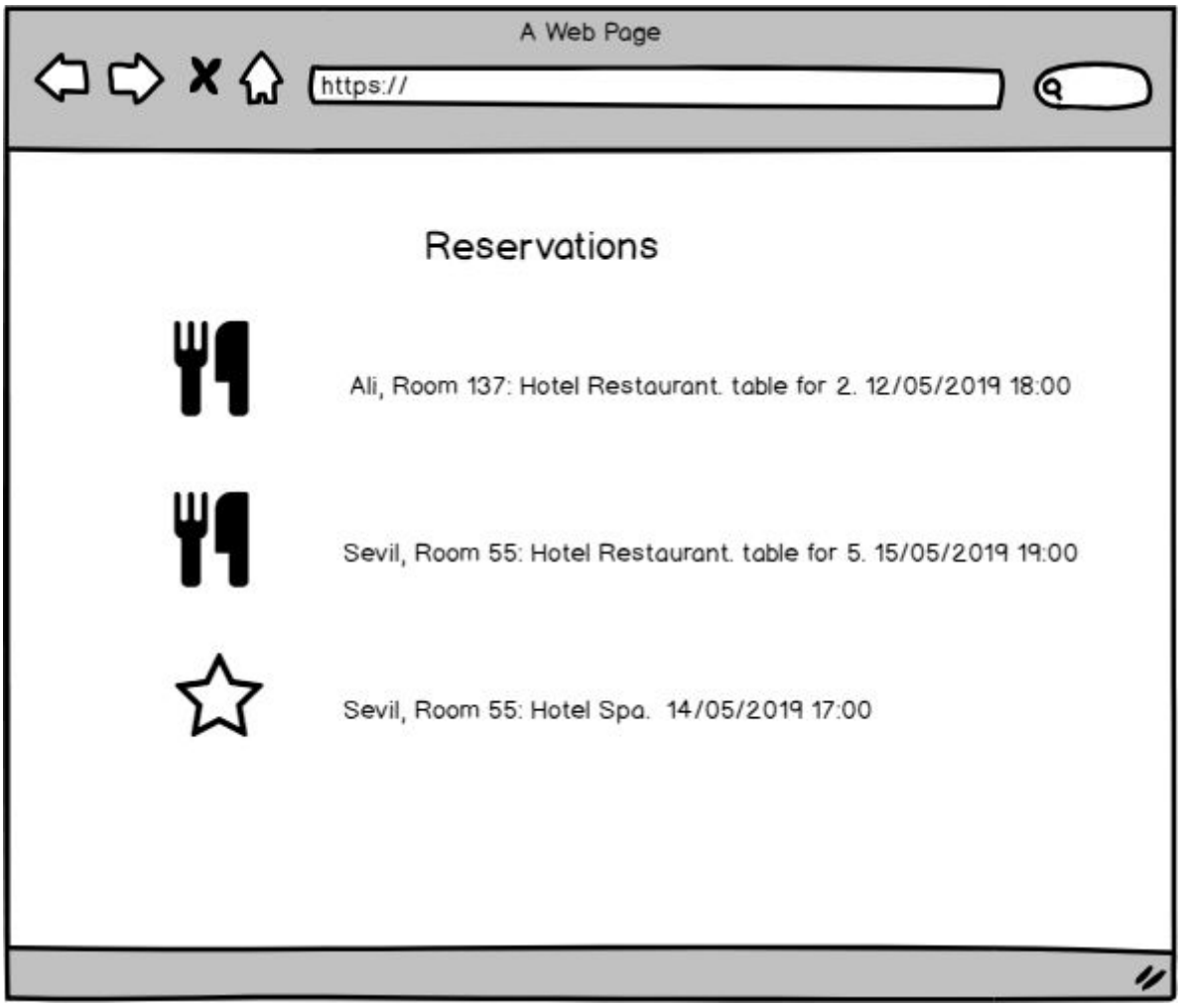

A Web Page
https://
Register
Please enter a valid business email address
and choose a strong password
Hotel Name:
Email:
Password:
Confirm password:
next

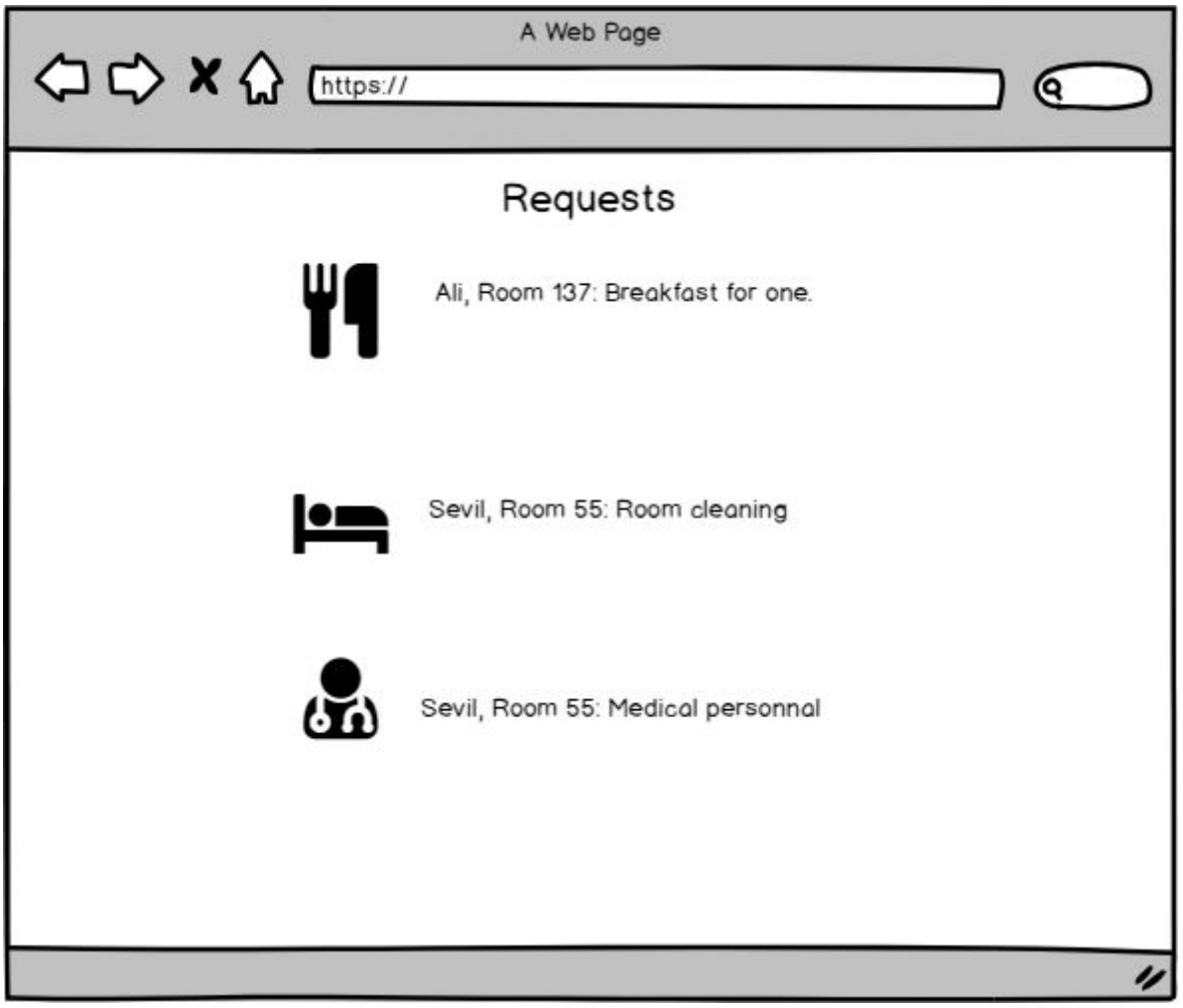

A Web Page
https://
Register
Enter the confirmation code
sent to your email adress
Code:
next

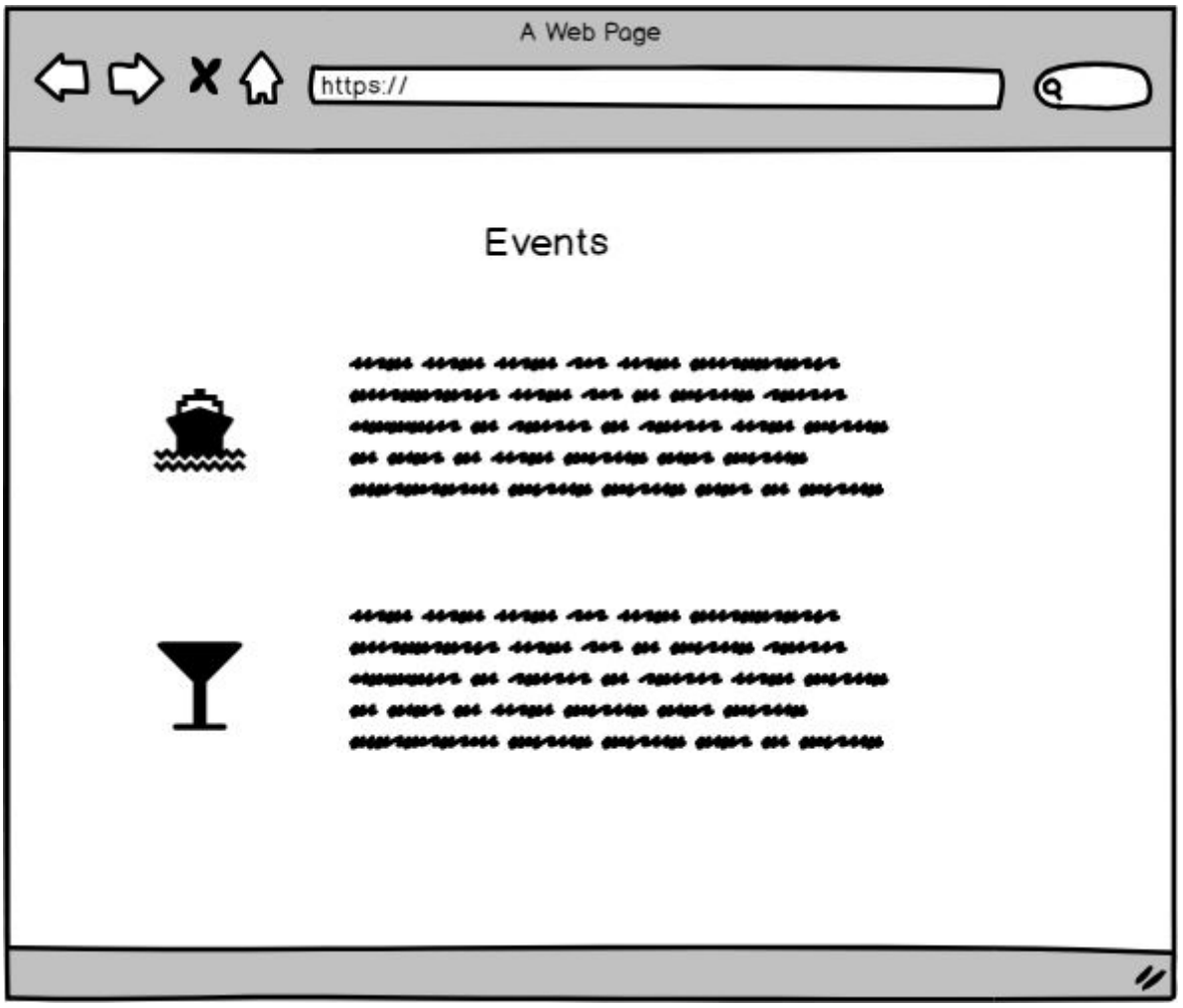

A Web Page
https://
Login
Please enter your hotel email and password
Email:
Password:
Forgot password?

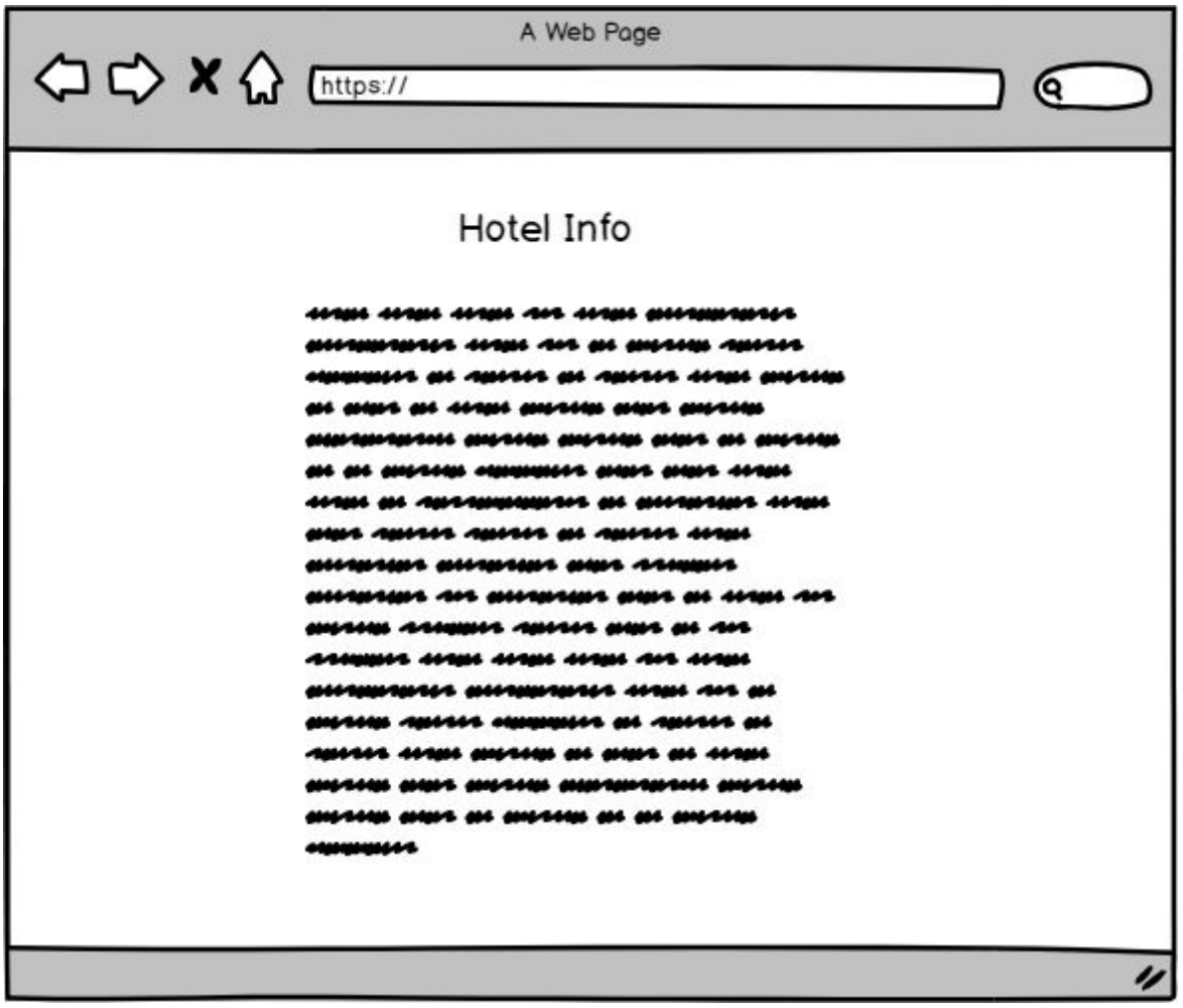

A Web Page
https://
Hotel Info
Hotel Events
Restaurant
Reservations
Requests
Statistics
Guest Reviews

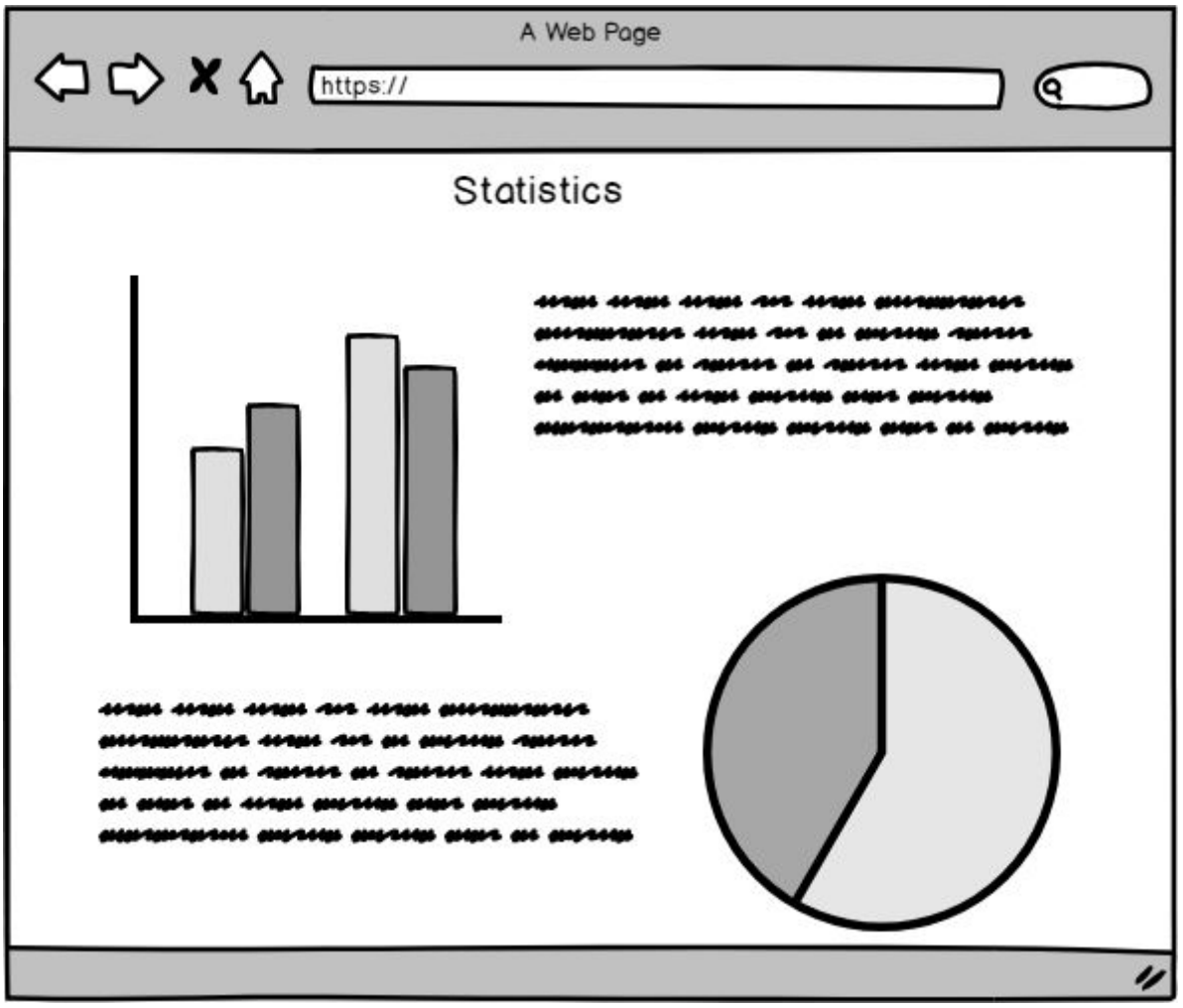

A Web Page
https://
Hotel Restaurant
Breakfast
Lunch
Dinner

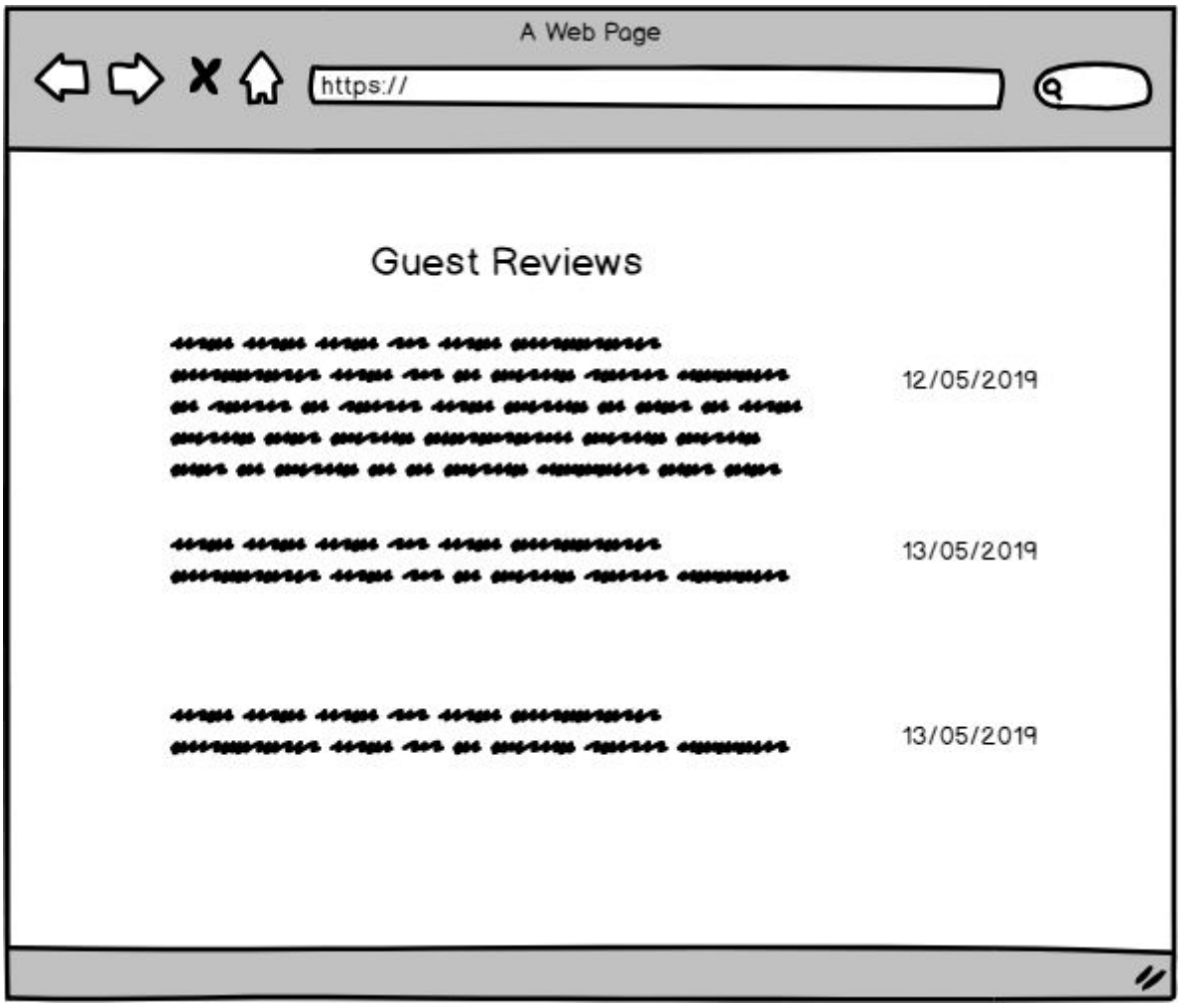

A Web Page
https://
Reservations
Ali, Room 137: Hotel Restaurant. table for 2. 12/05/2019 18:00
Sevil, Room 55: Hotel Restaurant. table for 5. 15/05/2019 19:00
Sevil, Room 55: Hotel Spa. 14/05/2019 17:00


A Web Page
https://
Requests
Ali, Room 137: Breakfast for one.
Sevil, Room 55: Room cleaning
Sevil, Room 55: Medical personnal


A Web Page
https://
Events


A Web Page
https://
Hotel Info


A Web Page
https://
Statistics



4. Glossary

Github is a repository hosting service that uses Git as a source control management tool.
React is a Javascript framework.

5. 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. Marriott International, "*Marriott Bonvoy*" [Online] Retrieved from: https://play.google.com/store/apps/details?id=com.marriott.mrt [Accessed: 24-March-2019].

4. Hyatt Corporation, "*Hyatt Hotels*" [Online] Retrieved from: https://play.google.com/store/apps/details?id=com.Hyatt.hyt [Accessed: 24-March-2019].
5. Hilton, "*Hilton Honors*" [Online] Retrieved from: https://play.google.com/store/apps/details?id=com.hilton.android.hhonors [Accessed: 24-March-2019].
6. eZee Absolute, "*eZee Absolute*" [Online] Retrieved from: https://www.ezeeabsolute.com/ [Accessed: 24-March-2019].
7. A. Gupta and J. Edovegi, "A list of most known JavaScript Frameworks," *Apiumhub* 21-Dec-2017. [Online]. Available: https://apiumhub.com/tech-blog-barcelona/top-javascript-frameworks [Accessed: 24-Feb-2019].
8. "React Native for Cross-Platform Application Development," *The Technology Stack Behind The Grandeur Success Of Uber*, 31-Dec-2018. [Online]. Available: https://www.bacancytechnology.com/blog/react-native-cross-platform-application [Accessed: 24-Feb-2019].
9. J. Huneycutt, "An Introduction to Clustering Algorithms in Python – Towards Data Science," *Towards Data Science*, 29-May-2018. [Online]. Available: https://towardsdatascience.com/an-introduction-to-clustering-algorithms-in-python-123438574097 [Accessed: 24-Feb-2019].
10. "Gather, Visualize, Analyze IoT data" *IBM Watson*. [Online]. Available: https://console.bluemix.net/docs/tutorials/gather-visualize-analyze-iot-data.html#gather-visualize-and-analyze-iot-data [Accessed: 24-Feb-2019].
11. "Amazon EC2 Pricing – AWS," *Amazon* . [Online]. Available: "Amazon EC2 Pricing – AWS," *Amazon* . [Online]. Available: https://aws.amazon.com/ec2/pricing/ [Accessed: 24-Feb-2019].
12. "Firebase" *Google*. [Online]. Available: https://firebase.google.com/pricing/ [Accessed: 24-Feb-2019].
13. "Cloudant - Pricing," *The Analytics Maturity Model (IT Best Kept Secret Is Optimization)*. [Online]. Available: https://www.ibm.com/cloud/cloudant/pricing [Accessed: 24-Feb-2019].
14. "Custom QR Code API Pricing," *Crunchbase API Documentation (crunchbase) | RapidAPI*. [Online]. Available: https://rapidapi.com/qrcode-monkey/api/custom-qr-code-with-logo/pricing [Accessed: 24-Feb-2019].
15. "The Code affirms an obligation of computing professionals to use their skills

for the benefit of society.,” *Code of Ethics.* [Online]. Available: https://www.acm.org/code-of-ethics
[Accessed: 24-Feb-2019].