Implementation Plan

General Description

Our solution for this challenge will be a **Web application** where *Hotel Manager* will have the possibility to configure the system and manage the hotel resources and *Hotel Guest* will be able to view available rooms, place a reservation and manage it, perform a check–in/check-out, etc. For accessing the room we will create a **mobile application** as for *Hotel Guest* as for *Cleaning Staff*, where the guest will log-in and see the rooms for which he has already made the check-in (maybe he reserved 2 different rooms). The guest will be able to select the room and access it via phone NFC. Similar for cleaning staff: he/she will log-in and see all rooms + their status, will select one room and access it via mobile NFC.

For now, we will keep the project as simple as possible, but we will try to cover the most of client requests.

For Web pages we choose to use Nodejs for the back-end, because it is simple in use and we want to keep the client side (front-end) and the server side (back-end) in the same programmable language "JavaScript".

As we cannot perform the room access from the web page, we will create a simple mobile application which purpose is to see the room status and to open the door.

Architecture

Component Diagram

The diagram can be found here, also the screenshot can be found in drive.

Class Diagram

This class diagram doesn't cover all methods, maybe we will need to add an utility class, to split the big functions in the smallest one and add methods for validation, printing errors and so on.

The diagram can be found <u>here</u>, and the screenshot can be found in <u>drive</u>.

Sequence Diagram

This diagram details how operations are carried out (shows how these operations interact with each other).

The diagram can be found here, also the screenshot can be found in drive.

Deployment Diagram

This kind of diagram represents the communication between database system and web server (MVC).

The diagram can be found here and the screenshot can be found in drive.

Database Schema

Even if at first look the database schema seems to be small and simple, we manage to keep all necessary things in it and we don't supply it with data that we will not use now.

For some fields as *Reservation.Status*, *Room.Type*, *etc* we want to memorise data as enum, we know that we can simulate this by choosing varchar type and adding some constraints, but we prefer to keep those fields as TINYINT, and in our code we will have an associated enum for them (see the class diagram).

The diagram can be found <u>here</u>, and the screenshot can be found in the <u>drive</u>.

UI Wireframes

We've created very realistic wireframes for the most of pages and functionalities to make it easier to code the front-end.

For wireframing, designing was used Adobe XD and Adobe Illustrator for creating the background. All screenshots can be found in <u>drive</u>.

List of Technologies

Technology Name	Туре	Purpose
Bootstrap	JavaScript library	Designing the layout, and UI controls(forms,buttons)
React	JavaScript Framework	Develop UI
jQuery	JavaScript library	Easier writing of JavaScript
Node.js	Server environment	To run the server.
HTTP	Module	Used to transfer data.
Express.js	Web application framework	To manage the server side (root, handling requests and views).
Mocha	Testing framework for NodeJS	To create unit tests.
Mongoose	Web server	Hosting the database

Android Studio	IDE	Build NFC App using java language and nfc library, to read or detect NFC
		tags.

Work Breakdown

We plan to spend **66 hours** on this project as following:

- Project management 5 hours;
 - Create issues 2 hours;
 - Code review 3 hours:
- Create and Deploy Database 2 hours;
 - Create Database 1 hour;
 - Deploy and populate database with data 1 hour;
- Install server 1 hours;
 - Install server and all necessary modules 1 hour;
- Create Backend module 19 hours;
 - Develop project structure with Hello World 1 hour;
 - Develop server side 3 hours;
 - Develop Login 1.5 hours;
 - Develop room management by admin (add / update / delete) 4 hours;
 - Develop get available rooms for guests 2 hours;
 - Develop finding a room by some criterias 2 hours;
 - Develop add / cancel reservation 2 hours:
 - Develop checkin / checkout 1 hour;
 - Develop sending emails to guests 2.5 hours;
- Create UI module 14 hours;
 - Develop project structure with Hello World 1 hour;
 - Create general login page 30 min;
 - Create 2nd login page for security code 30 min;
 - Create guest interface 5 hours;
 - Create home page 2 hours;
 - Create calendar component 2 hours;
 - Create "my reservation" page 1 hour;
 - Create admin interface 6 hours;
 - Create room page 2 hours;
 - Create add room modal window 1 hour;
 - Create reservation page 2 hours;
 - Create send special offer page 1 hour;
- Integrate screen/feature with Backend 8 hours;
 - Login 1 hour;
 - Room management by admin 2 hours;
 - Get available rooms for guests 1 hour;

- Finding a room by some criterias 1 hour;
- Add / cancel reservation 1 hour;
- Checkin / checkout 1 hour;
- Sending emails 1 hour;
- Create test cases 7 hours;
 - Manual testing 2 hours;
 - Unit tests 3 hours;
 - Prepare the report 2 hours;
- Create Mobile App 10 hours;
 - o Create running application 2 hours;
 - o Develop Login 2 hours;
 - Get room status 2 hours;
 - o Implement NFC activity (guest, cleaning staff) 4 hours;