WoGet

Nathanël Benichou
Naomi Oyer
Jordan Perez

Instructor : Prof. Franck Assous

Software Design Document

Date: (16/12/2021)

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

This software design document describes the architecture and system design of the software WoGet to help people rent unused personal places and others to use these places for work.

1.2 Scope

The system is divided in two parts (both software):

(1) The customer part: Interface set up to provide users all types of workspaces.
(2) The part dedicated to the user wish to make their workspace available.

- The system works using a dynamic database and a pleasant and easy-to-use display interface.
- The interface offers different type of evolutive algorithms allowing continuous improvement of the service's quality.

1.3 Overview

The SRS document contains detailed information about the product and the system, both for the client and the programmer.

1.4 Reference Material

We use as base support two existing application: Airbnb and Uber

1.5 Definitions and Acronyms

WoGet aims to let "**Workers**" rent private workspace from "**Nudgers**":
Workers are people who want **to have their own working place** to study/work/meet in a wokspace.The Woget app puts the Workers in touch with *Nudgers*, people/businesses **who have a spare and unused local/garage/basement/room/ or any kind of place** –that we call a "***Nudge***" - they can furnish in accordance of working needs and **want to rent it for a few hours/days.**

2. SYSTEM OVERVIEW

Workers can search for their workspace in a catalogue built by the Nudgers and make contact with a Nudger to rent their workspace.
Workers have the ability to filter results by location and the radius around it, price, rating and tools at disposal.
Nudgers have the ability to post their Nudge online by uploading pictures, a description, and by precising the address, the maximum number of workers the Nudge is intended to and all the features/tools the Nudge has at its disposal.
After finishing using the services, a worker can leave a review on the Nudger's space.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

- The application will ask the user to log in or create an account.
- If the user already has an account, the application should ask for his mail and password. If the user is new the application should ask him to fill multiple fields: first name, name, mail, password, phone number, profession, city.
- After logging in, the user should choose what action he wants to complete:
    - Display profile with personal information
    - Display history of rent, communication with renter/customers
    - Search for rent
    - Purchase a rent (allow customer to make a payment)
    - Leave comment/review/grade on rent
    - Add a place to rent (should be able to make modifications/remove item)
    - Leave comment/grade on customers
- Search for rent: should display filters to optimize the search and make it accurate and quicker.
    - Filters: distance/location/area, type of location (room, flat, conference room, office…), services (Wi-Fi, air conditioning…), furniture (table, computer, board, minibar, projector…), rates (star reward principle)
- Add a place to rent and generate an ad – the renter should fill fields with all the information on its space (location, type, size, furniture, services provided, duration of rent, price…)
  The new item should be added to the database
- The customer should be able to leave a review for the rent and rate (on 5 stars) his experience. The grade system should later affect the ordering of the rents in the search page.

3.2 Decomposition Description

Data flow diagram:

3.3 Design Rationale

At first, we debated which language we should use to code our software and which database, at the end we decided to use React and NodeJS because Both Nodejs and React are javascript languages that can be executed both client and server-side and developers can execute the React code directly in the Nodejs environment and React has components specifically designed to work with Nodejs that reduce lines of code, making server-side rendering comparatively easy.

We debated which database use and at the end we chose MongoDB since we need to use a lot of different data.

We also debated if we wanted to make a web app or a mobile app only, at the end we chose the web app because it allows multiple users to access to the same version of an application and also because Web apps can be accessed through various platforms such as a desktop, laptop, or mobile.

4. DATA DESIGN

4.1 Data Description

First of all, the system needs to know the personal data (to establish a profile and to allow the customer to connect) of each customer (Nudger or Worker) which includes first name, name, mail, password, phone number, profession, city.
In the profile will be displayed history of rent and communication with renter/customers.

When a Nudger upload a place to rent he needs to provide distance/location/area, type of location (room, flat, conference room, office…), services (Wi-Fi, air conditioning…), furniture (table, computer, board, minibar, projector…), price, duration of rent and images (optional). These features will be added to the filter option for the Workers.

Reviews: reviews can be given on a Nudge (in shape of stars (up to 5) and comments (optional). The stars will be a possibility of filter in the search.

4.2 Data Dictionary

Worker: Show_profile(), Show_history(), Add_comment(), Delete_User()
Nudger:Show_profile(), Show_history(), Add_comment(), Delete_User()
Nudge:Add_Nudge(), Remove_Nudge(), Add_features(), Add_Payement(), Receive_Payement()
Server: Analyse_Data(), Upload(), Request(), Autorisation(), Send_Information().
Database: Show_Data(), Remove_Data()

5. COMPONENT DESIGN

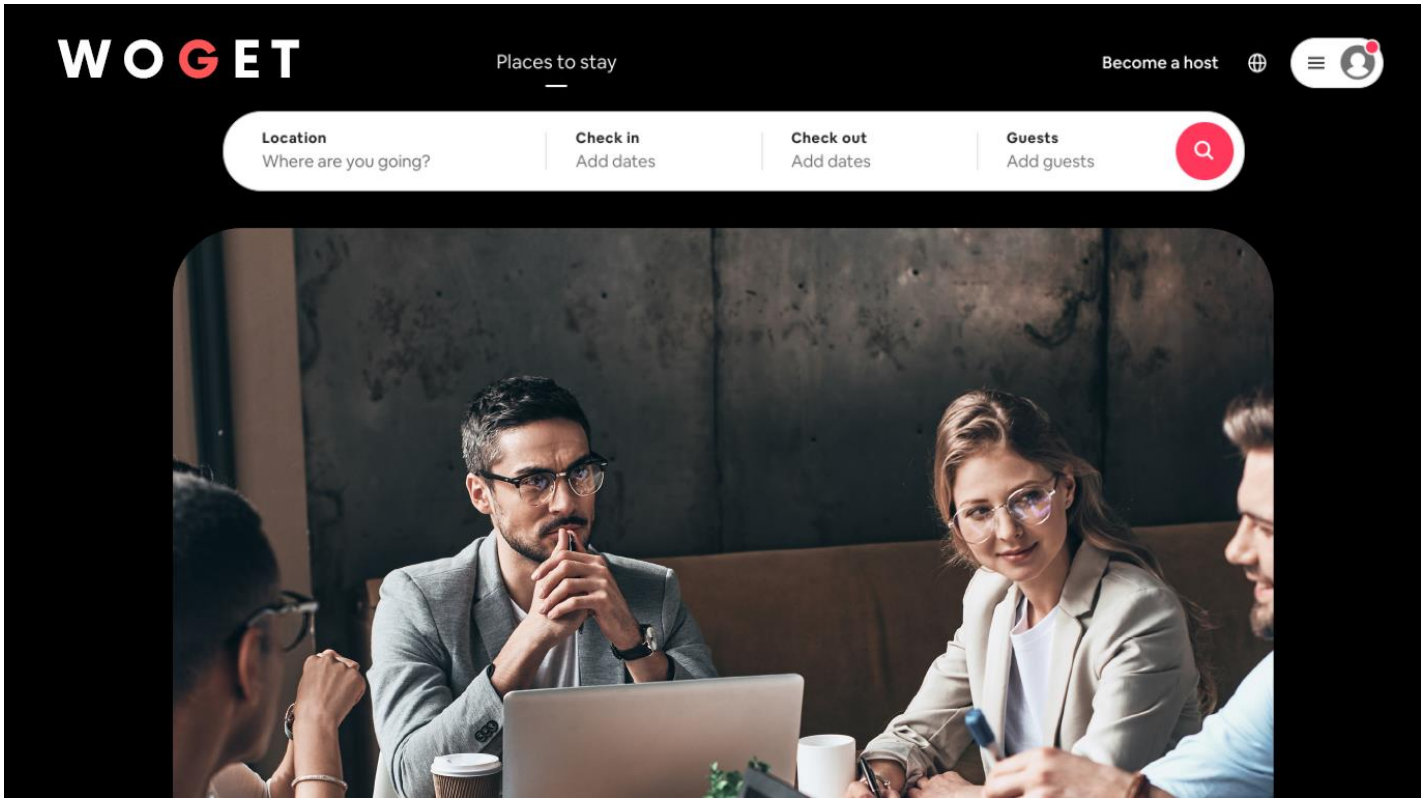
Algorithms:
A searching algorithm for Workers.
A posting interface for Nudgers.
Both Workers and Nudgers can get special rewards leading to discounts according to their rating and their fidelity. Such rewards are earned when a specific algorithm determines that the user deserves to be rewarded. This aims to increase the fidelity and the quality-service of the users.
Finally ,there is an algorithm that calculates the "estimated price" of a Nudge according to its parameters such as its rating, its location and its services.
Workers and Nudgers have the ability to chat together.
Payment methods to implement.

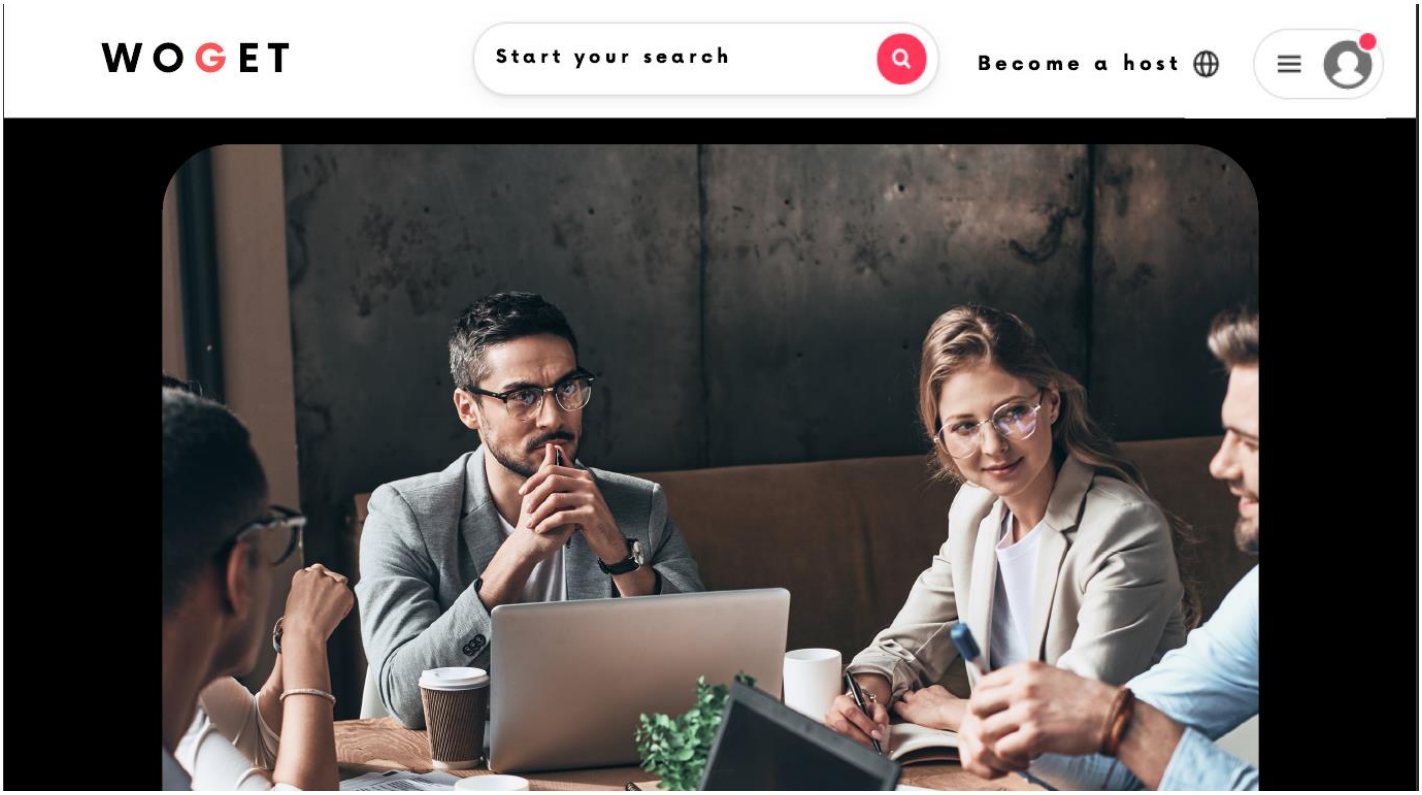
6. HUMAN INTERFACE DESIGN

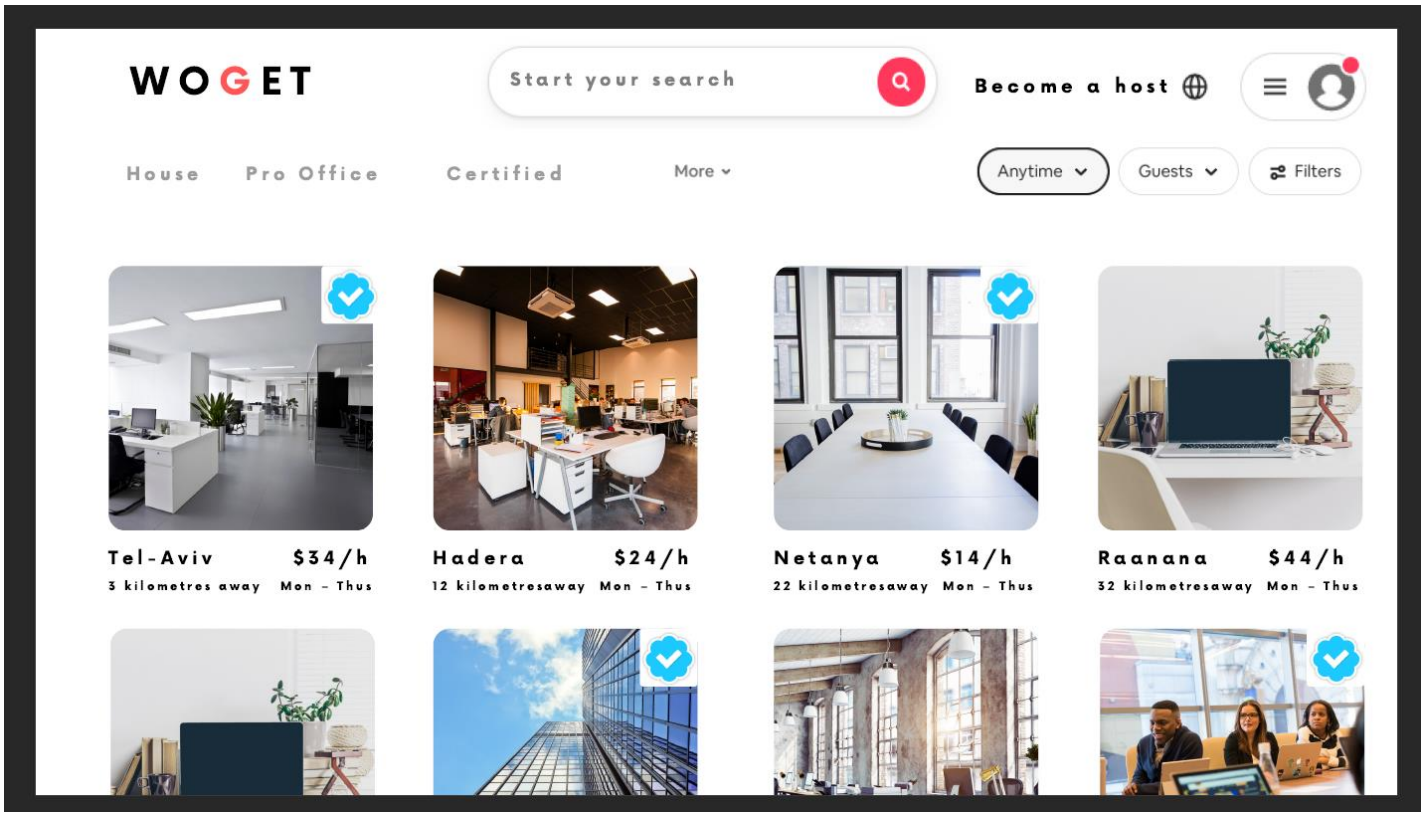
6.1 Overview of User Interface

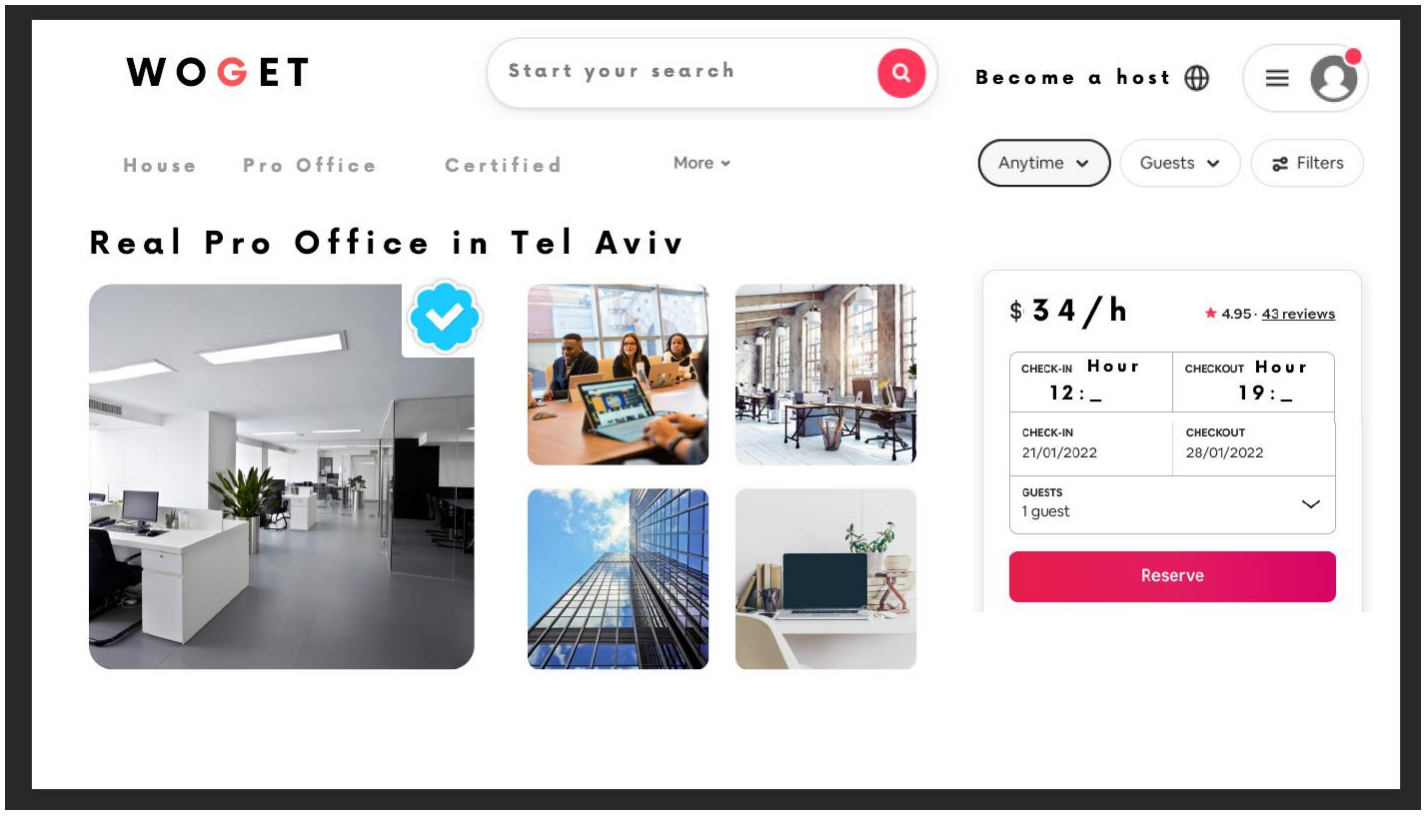
First the user needs to register – fill fields of information. This action will create a new user with log in information and profile.
After logging in, the user can do several actions:
Worker:
- Visualize and change his personal information.
- Search for a Nudge: in the search for a Nudge section, the user can choose filters to narrow the results and find a place that suits him best. If the user found an interesting proposition, he can select to book a place or decide to contact the Nudger (for further information).
- After using the Nudge services, the user can access in his history of rents to grade the space (select up to 5 stars) and leave a comment or not.
Nudger:
- Visualize and change his personal information
- Add a Nudge: the user needs to fill fields of information on the place h wants to rent. After uploading a Nudge, the user can make change on it in his history of ads (or remove it). The Nudge will be displayed in the search for Nudges section.
-At the end of the rent time, a Nudger can leave a comment on the customer on his personal page.
Obviously, a worker can be a Nudger and a Nudger can be a worker!

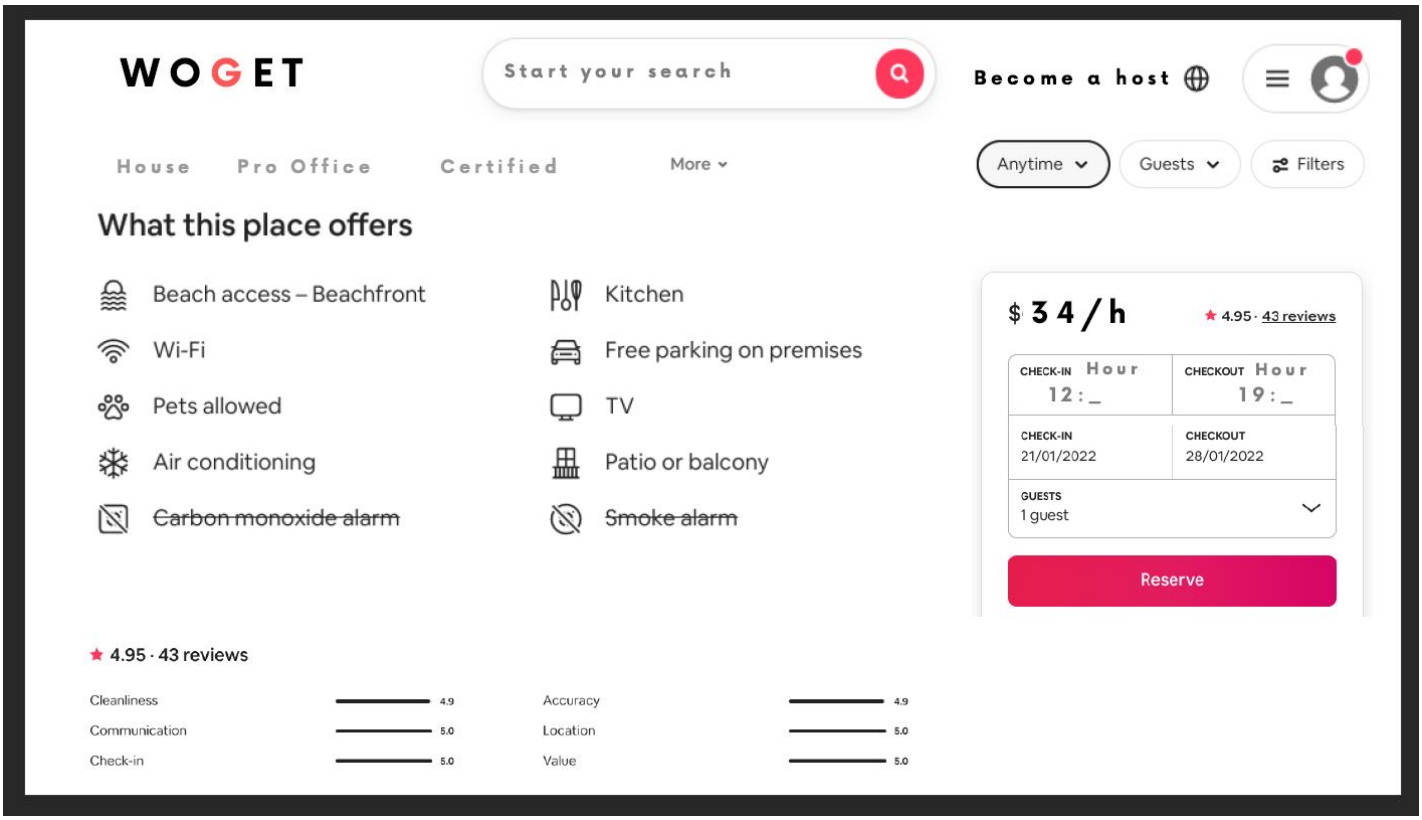
6.2 Screen Images






WOGET
Start your search
Become a host
House
Pro Office
Certified
More
Anytime
Guests
Filters
Tel-Aviv $34/h
3 kilometres away Mon – Thus
Hadera $24/h
12 kilometresaway Mon – Thus
Netanya $14/h
22 kilometresaway Mon – Thus
Raanana $44/h
32 kilometresaway Mon – Thus


WOGET
Start your search
Become a host
House
Pro Office
Certified
More
Anytime
Guests
Filters
Real Pro Office in Tel Aviv
$ 34/h
4.95 · 43 reviews
CHECK-IN Hour
12 : _
CHECKOUT Hour
19 : _
CHECK-IN
21/01/2022
CHECKOUT
28/01/2022
GUESTS
1 guest
Reserve



6.3 Screen Objects and Actions