WoGet

Nathanël B
Naomie Oyer
Jordan Perez

Instructor : Prof. Franck Assous

Software Requirements Specification

Document

Version: 1.0 **Date: 30/11/2021**

1.1 Purpose

The SRS document is intended to explain the product and its purpose to the customer.

The document is intended for the target audience of the (web) application, which are all types of workers from freelancers to students and others.

1.2 Scope

The system is divided in two interfaces:

(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 Definitions, Acronyms, and Abbreviations.

Nowadays, a lot of people need a space to work, alone or with a team, should they be students, freelancers, building a startup, etc. To complete this, working at home, at a coffeeshop or in a crowded library isn't always the best option.

To be able to focus and provide results, people should have a quiet available space, furnished as they need and in their area.

Our idea is to create a platform providing multiple options of spaces to rent near your location, for a few hours or more and even have the opportunity to get all the tools you need to accomplish your tasks in the best way.

Furthermore, we allow anybody that have a vacant space to put it for rent on our plateform and be contacted by people in need.

WoGet is the abbreviation for Get Work.

1.4 Overview

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

SECTION 2 is for the customer. It lists all the requirements the customer needs for the product, as well as the best use of the product.

SECTION 3 is for programmers. It lists all the functional and non-functional requirements of the product and explains the system setup.

2. The Overall Description

2.1 Product Perspective

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 pleasant workspace **with all the tools adapted for their needs** such as boards, desks, projectors, Wi-Fi, air-conditioning, multiple wires, coffee and more. 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.**
As students living far from the campus, we felt the lack of this kind of service: we could not find a proper private space to meet in and conveniently work together for a few hours, somewhere close to all of us with all the tools needed. And this is how WoGet was born. Pros of libraries without the cons : no more wasting time looking for a place in crowded libraries, and you get your own personal space with all the convenience that comes with it.

2.1.1 System Interfaces
User login will be done with data from Firebase; in addition, we will use the API of the hash key to encode the user interface.

2.1.2 Interfaces
There are two interfaces available via the WoGet App: one for the "*Workers*" and another for the "*Nudgers*".

2.1.3 Hardware Interfaces
No hardware interface

2.1.4 Software Interfaces
WoGet is a web app developed in React, the most popular JavaScript front-end framework in use today.

2.1.5 Communications Interfaces
The server-side application will be implemented in Node.JS for its scalability and its easy-to-understand code.

2.1.6 Memory Constraints
There are no memory constraints.

2.1.7 Operations
Workers can search for their workspace in a catalogue built by the Nudgers.

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.

2.1.8 Site Adaptation Requirements

Users must register to the site to use the system. From there, they can pick between Worker or Nudger.

2.2 Product Functions

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.
Dispute case to implement.

2.3 User Characteristics

No character is required to use the system, it is accessible to anyone with an Internet connection at home.

2.4 Constraints

- There has to be a balance between Workers and Nudgers. Too many Workers will result in a non-sufficient amount of available Nudges, and too many Nudgers will result in a non-sufficient amount of booked Nudges.
- Browser and Internet connection are required.
- The login will be done by using a username and password or a Google account.
- Both Workers and Nudgers should be able to open a dispute in case something went wrong (items stolen/broken , last-minute cancelation, etc) and we need to provide a fair system to deal with it.
- Payment methods need to be implemented and need to guarantee a refund system following some dispute cases
- Terms and Conditions in accordance of local laws

3. Specific Requirements

3.1 Functional Requirements

- 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 External Interfaces

Software:

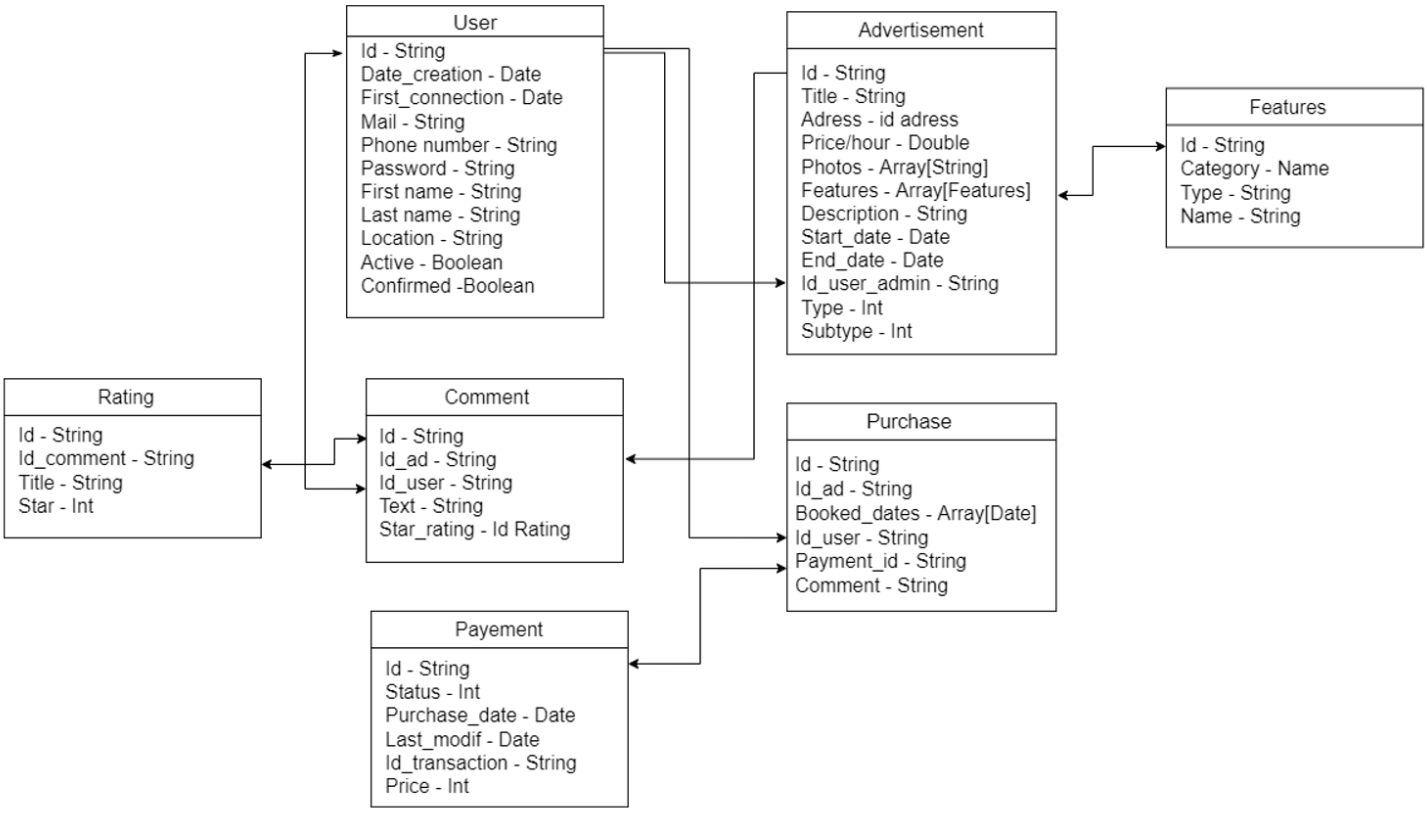
- The software is a web application programmed using a client-server structure:
    - Client side: will include JS, CSS, HTML.
    - Server side: will include node.js, react.js.
- The software will use the MongoDB database to store all the data:
    - Each user with its information, purchases, comments
    - Each ad of rent with its information, comments, grade
    - Each purchase
    - Each payment
- The software will provide multiple services such as:
    - Customer connection with mail/phone number verification
    - Payment - implement online payment gateway

- Dynamic search algorithm based on criteria and popularity.

3.3 Other Nonfunctional Requirements

- Performance: It is important that the software should work smoothly, without bugs and should display each element promptly. Clarity and a simple handling of the app are important requirements
- Safety: All users should be respected on the app, rudeness or hateful messages should be banned
- Security: All the private information of the customers (first name, name, address, password, mail…) should be protected and not be communicated as well as payment information. In addition, the users should have the ability to change only their data and not the data of other users in the system.
- Quality: The app should display multiple options, in a smart and understandable way and provide all the services in the best way

3.4 Logical Database Requirements



3.5 Design Constraints

- The app should be fully responsive, and the interface should be user friendly, clear and simple to use. The design should make it possible to anybody to use it.
- The app will include a lot of features such as filters, photos, fields for information, comments, price tag, pop up for payment.

4.Document Approvals

[YOUR SUPERVISOR]

Identify the approvers of the SRS document. Approver name, signature, and date should be used