Index

1. Software Requirement Specification..1

1.1. Introduction..1

1.1.1 Aim of the document..1

1.1.2 Overview of the defined system..1

1.1.3 Operational Settings..2

1.1.4 Related system, Pros and Cons..2

1.2 User Stories..2

1.3 Functional Requirements..3

1.4 Use Cases..4

1.5 Dictionary..6

1. Software Requirement Specification

1.1. Introduction

1.1.1 Aim of the document

The document covers a description dealing with general aspects of the system, differences with related systems, Functional Requirements, User Stories, Operational Settings, Use Cases Diagram.

1.1.2 Overview of the defined system

The system implements method for exchange favors between users. A user can request a favor to all users, posting an ad which details the favor, or request a favor from a specific user, answering to an ad where the publisher makes himself available for doing that favor.

When an ad is published, it becomes active until it's closed by the user or it's active time expires.

A user can find an ad by browsing a wall or using a pinpoint map, where active ads are displayed by their proximity to the user location.

A user can make his profile visible to Company Talent Scouts in order to allow them to contact him if they want to be hired by them.

1.1.3 Operational Settings

Desktop version of the System requires a JRE version 8 or above with JavaFX package included. Known suitable JREs are:

- jre_1.8.0_241 (JavaFX is included);
- JavaSE-1.8 + jfx8_2.3.0;
- openjdk version "1.8.0_252" + openjfx.

All other dependencies are integrated within the system, so no further operations are required.
It should run in end-users desktop machines which met above conditions. A Internet connection is required for some features, like ViewMap feature.

WebApp version requires a Web Server which can handle JSP language. Known working Web Servers are:

- Apache Tomcat/9.0.

Working directory of Web Server must be the same one the system is installed. Knowing browsers which render web pages correctly are:

- Mozilla Firefox 76.0.1;
- Version 81.0.4044.138 (Official Build) (64 bit).

1.1.4 Related system, Pros and Cons

Subito.it

The system is built following the model of trading site “Subito.it” where users can buy or sell an item using posts. A main difference between them is that the communication between users happens inside the system unlike “Subito.it” where it happens providing telephone number or sending emails.

Indeed.com

Indeed.com is a social platform where companies can publish an ad when they need to recruit. An user can apply to one of these ads, and they will be contacted by the company if it accepted their apply. Only companies can open an ad, while in the System both a Requester and a Offerer can open ads. Besides, users of Indeed.com cannot tell if a company decided to refuse their apply or it is still deciding, while users of the System can decline Answers to their ad. In thata case, a notification to the answerer would be sent.

1.2 User Stories

More User Stories can be found in the repository on GitHub, among tickets labeled “User Stories”.

User Stories written by Lorenzo Mei (lorenzomei):

- As a user, I want to know the location of the advertiser, so that I can know which ones are in my local area, 10 km at most (ticket #5);
- As an advertiser, I want to publish ads, so that I can inform the other users that I need or I can offer a favor (ticket #4);
- As a user, I want to filter the search of ads, so that in my search there aren't unwanted results. (ticket #2).

User Stories written by Mihai Jianu (mihaiJianu):

- As a new user, I want some free **token**, so I can request a **favor** without offering one before (ticket #18);
- As a user, I want to view the **profile** of other users, so I can know the person's characteristics (e.g. skills and personal information) before we get in touch (ticket #19);
- As a user, I want to see the reviews about the **advertisers**, so I can know their reliability reading other users opinions (ticket #20).

User Stories written by Daniele La Prova (Torkin1):

- As a beta tester, I want to use a feedback system, so that I can let developers know my opinion about the software (ticket #10);
- As a veteran user I want to have a favorite user list, so that I can remember who are the best users in my experience (ticket #11);
- As a new user, I want to have a tutorial, so that I may learn how to use the software (ticket #12);

1.3 Functional Requirements

All terms written in **bold** are defined in Dictionary section.

Functional Requirements written by Lorenzo Mei (lorenzomei):

- The system shall provide categories with which the ads can be distinguished (like accommodations, botany, hydraulics, and mechanics) (ticket #9);
- The system shall provide the possibility to post an ad with the following frame (ticket #8):
  • A title, which will summarize the content of the ad;
  • A tag, that will flag the ad as an offer or a request;
  • A text box, that will contain the detailed content of the ad;
  • A dedicated area, where the user can attach videos, documents and pictures;
  • A feature that allows the users to get in touch.
- The system shall provide a map that pinpoint the location of the advertiser that have an **active ad** at the moment (ticket #7);

Functional Requirements written by Mihai Jianu (mihaiJianu):

- The system shall provide, a 5-star rating method, (a 1-star rating represents the worst judgment, and a 5-star represents the best one) with a text space, in which the user can add further details about the motivation of the judgment (ticket #17);
- The system shall provide a Wall-like space, where a user can upload their “jobs done” pictures and curriculum-like space where a user can write their skills and personal information (ticket #16);
- The system shall send an e-mail to the first access users that contain a free **token** as a gift and a welcome message, which greets the user, and explains how to use the **token**.

Functional Requirements written by Daniele La Prova (Torkin1):

- The system shall provide a tutorial to first-access users, containing the following items (ticket #6):
  • An explanation about how the user is supposed to use tokens and how to obtain them;
  • An explanation about how the user is supposed to get in touch with other users in order to exchange favors;
  • An explanation about how to get in touch with the support team if further help is needed;
  • A visual demo of how to achieve the steps above.
- The system shall provide the possibility to mark a user as a “favorite” and a section where the user can see a list of users marked as favorite by them (ticket #3);
- The system shall provide an agent available for beta testers, with the following behavior when invoked:
  • It shall let the tester specify what software features the tester wants to write a report about;;
  • It shall open up a text box where the tester can write a report about how experienced a software feature;
  • It shall allow the tester to attach screenshots and videos to the text he wrote;
  • It shall send a report to the developers via e-mail.

1.4 Use Cases

Follows a picture showing an Use Case Diagram summing up what the system is meant to behave by an user perspective.

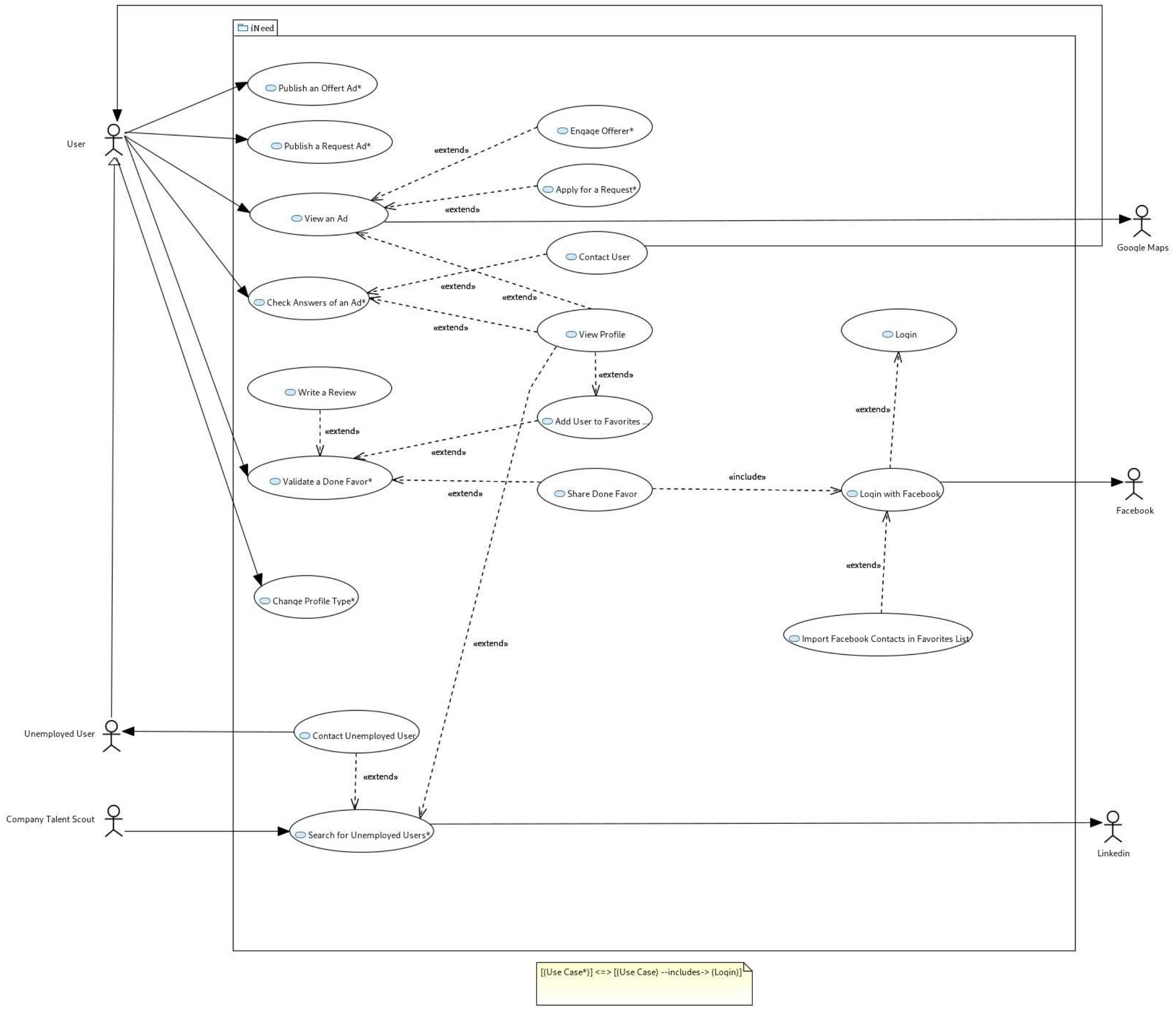

Figure 1: Use Case diagram

1.5 Dictionary

Validation	It's the procedure that will check the authenticity of a done favor and will grant a token to the user that did the favor.
Advertiser	It's the owner of an active post where they request or offer a **favor**.
Token	It’s the currency used to exchange **favors** between the users.
Favor	A service that users can request by or offer to other users.
Offerer	A User who is chosen by a Requester in order to fulfil the favor they requested, or who opened an Offer Ad.

Requester	A User which requested a favor by posting an ad or by Answering an Offer Ad.