vnv

Software Requirements Specification For vnv

Version 1.0

Revision History

Date	Version	Description	Author
10/24/2016	1.0	initial version	Waldemar Merk, Patrick Treyer, Gabriel Zachmann

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the vnv project. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

1.2 Scope

This SRS applies to the entire vnv project. ‘vnv’ is a social platform to lend or comp goods or services. This project will be realized as an Web-Application. The overview of the features and subsystems are documented in the Use-Case model shown in Overall Description.

1.3 Definitions, Acronyms, and Abbreviations

SRS	Software Requirements Specification
vnv	verleihen, Dienstleistungen, verschenken
n/a	not applicable
tbd	to be determined
AAI	Authentication and Authorization Infrastructure

1.4 References

GitHub

Use case diagram

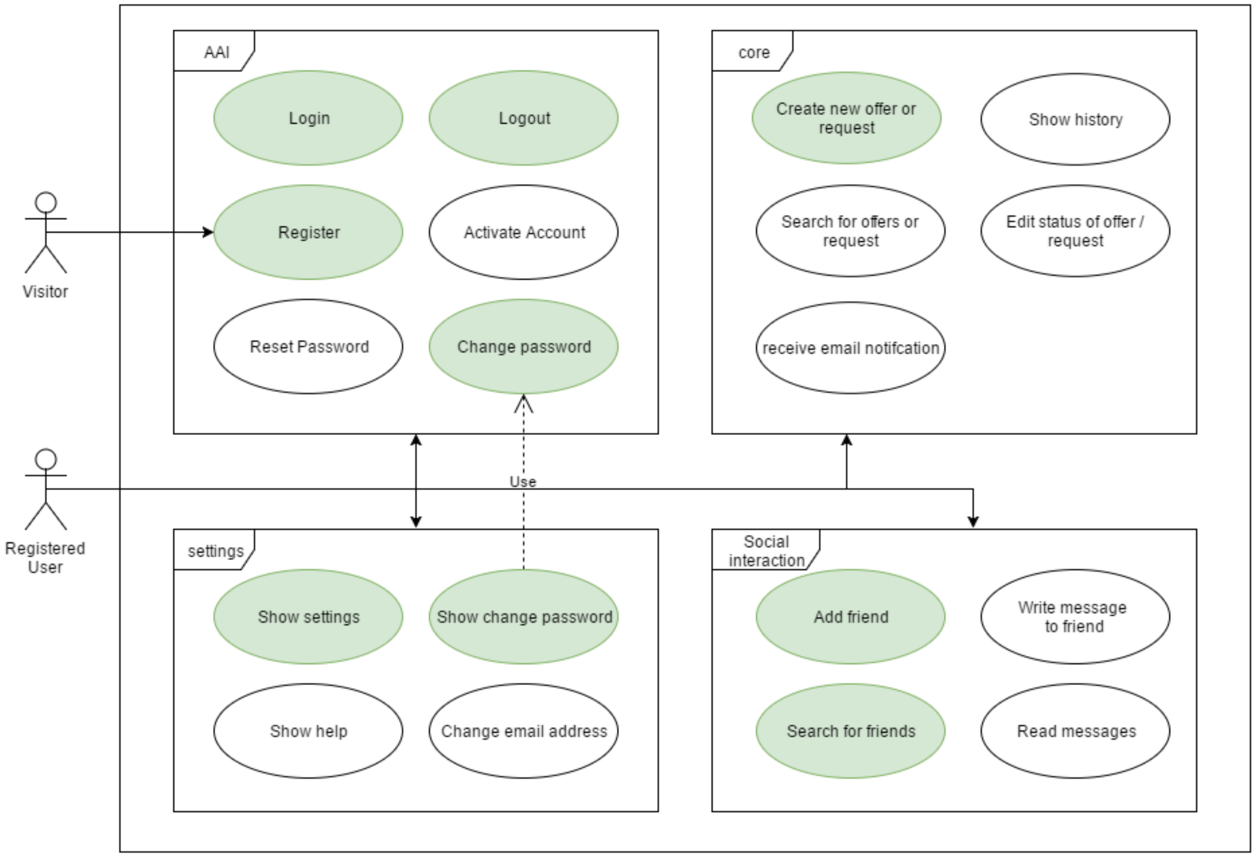
Blog

1.5 Overview

The remainder of this document is structured in the following way:
The next chapter, the Overall Description section, gives an overview of the functionality of the product and includes an use-case-diagram.
The third chapter, the Requirements Specification section, provides an more detailed description of the requirements.

2. Overall Description

vnv will be a web portal, where users can connect with their friends to borrow items from them, give things they don't use anymore to a friend looking for exact that one or users can check if one of their mates can help them with a problem (e.g. if one needs help with his network).
The Server Backend application will be written in Java using the Spring Framework. The Frontend will be using AngularJS.



3. Specific Requirements

3.1 Functionality

3.1.1 AAI

3.1.1.1 Register

The user can register by providing an username, password and email address. Optionally the possibility to register with Google or Facebook may be provided.

3.1.1.2 Activate Account

An email should be send to the users email address to confirm the address.

3.1.1.3 Login

The user can log in to the service by providing his username and password.

3.1.1.4 Logout

The user can log out.

3.1.1.5 Change password

When logged in the user can change his password by providing the old and new password.

3.1.1.6 Reset password

If the user forgot his password he can request to reset his password. An email should be send to his address.

3.1.2 Social Interaction

3.1.2.1 Search for friends

The user can search for his friends.

3.1.2.2 Add friend

The user can add another user as a friend.

3.1.2.3 Write message

The user can write messages to his friends to organise details.

3.1.2.4 Read message

The user can read the conversations with his friends.

3.1.3 vnv core

3.1.3.1 Create new offer or request

The user can create new offers and requests.

3.1.3.2 Search for offers / requests

The user can search for offers and requests.

3.1.3.3 Show history

Displays the history of the users requests and offers with their status (verliehen, zurückgegeben, verschenkt, …)

3.1.3.4 Edit status of offer / request

The user can change the status of an request / offer. E.g. from “verliehen” to “zurückgegeben”.

3.1.3.5 Email notification

An email notification could be possible. User could be informed about the status of his offers / requests or if someone requests something the user is capable of providing. Also notification via Telegram / Whatsapp could be possible.

3.1.4 settings

3.1.4.1 Change email address

The user can change his email address. the new address has to be confirmed again. (See Activate Account)

3.1.4.2 Show help

An help on how to use the service may be provided.

3.2 Usability

3.2.1 Training time

The normal user is able to use the application without a special training time. The insertion of an entry for other users is very easy. It takes the routined user less than 5 minutes. The GUI is designed to be very clear and visceral.

3.2.2 Language

The language will be German. The user is expected to to be capable of it. An English version is not planned yet.

3.3 Reliability

3.3.1 Availability

The service shall be available 90% of the time. Downtimes during night are tolerable. Availability beneath 90% is acceptable during development.

3.3.2 Mean Time Between Failures

Should be as high as possible. No estimation possible at this point.

3.3.3 Mean Time To Repair

Should be as low as possible. No exact estimation possible at this point. Current estimated MTTR is three working days.

3.4 Performance

3.4.1 Response time

Should be as low as possible. Maximum response time is 3 seconds. Average response time should be less than 1 second.

3.4.2 Capacity

the system should be capable to manage thousands of registered users and up to hundred users at the same time.

3.5 Supportability

3.5.1 Conventions

The code should follow the Java Naming Conventions and using speaking variable and function names

3.6 Design Constraints

3.6.1 Git

Git is used for version control.

3.6.2 YouTrack

YouTrack is used as issue tracker.

3.7 Online User Documentation and Help System Requirements

The design of the website should be intuitive. An online help may be provided

3.8 Purchased Components

n/a

3.9 Interfaces

3.9.1 User Interfaces

The user interface is the Web-portal. Functionality is described in Functionality.

3.9.2 Hardware Interfaces

n/a

3.9.3 Software Interfaces

n/a

3.9.4 Communications Interfaces

Communication to a Mail Server for sending mails to users should be possible.

3.10 Licensing Requirements

n/a

3.11 Legal, Copyright, and Other Notices

n/a

3.12 Applicable Standards

n/a

4. Supporting Information

n/a