

**Software Engineering II Project**

**Cleaning Help**

Part task: Android

**Group 1**

Haoze Zhang

Zhaowen Gong

Zelin Wu

Xiaoqi Ma

**Content**

[1. Specification 4](#_Toc456527673)

[1.1. Description 4](#_Toc456527674)

[1.1.1. Product Purpose 4](#_Toc456527675)

[1.1.2. Product Scope 4](#_Toc456527676)

[1.1.3. Potential Customers 4](#_Toc456527677)

[1.2. Product functions 4](#_Toc456527678)

[1.3. User characteristics 5](#_Toc456527679)

[1.4. Functional requirements 5](#_Toc456527680)

[1.4.1. Presentation 5](#_Toc456527681)

[1.4.2. Content 6](#_Toc456527682)

[1.4.3. Add, modify and delete 7](#_Toc456527683)

[1.4.4. Search 8](#_Toc456527684)

[1.4.5. Language 9](#_Toc456527685)

[1.5. Non-functional requirements 9](#_Toc456527686)

[1.5.1. Portability 9](#_Toc456527687)

[1.5.2. Performance and Availability 10](#_Toc456527688)

[1.5.3. Security and Extensibility 10](#_Toc456527689)

[1.6. Development environment and framework 10](#_Toc456527690)

[1.6.1. Software 10](#_Toc456527691)

[1.6.2. Framework 10](#_Toc456527692)

[1.6.3. Reference 10](#_Toc456527693)

[2. UML Specification 12](#_Toc456527694)

[2.1. Use Cases 12](#_Toc456527695)

[2.2. Class Diagrams 12](#_Toc456527696)

[2.2.1. Class Diagram 12](#_Toc456527697)

[2.2.2. Entity Relation 13](#_Toc456527698)

[2.2.3. Preference Control 13](#_Toc456527699)

[3. GUI Design 14](#_Toc456527700)

[3.1. PC Version 14](#_Toc456527701)

[3.1.1. Structure 14](#_Toc456527702)

[3.1.2. Screenshots and description 15](#_Toc456527703)

[3.2. Android Version 23](#_Toc456527704)

[3.2.1. Structure 23](#_Toc456527705)

[3.2.2. Screenshots and description 24](#_Toc456527706)

[4. Test 30](#_Toc456527707)

[4.1. Usability Test 30](#_Toc456527708)

[4.1.1. User Group 30](#_Toc456527709)

[4.1.2. Test tasks 30](#_Toc456527710)

[4.2. Equivalent classes and boundary tests 31](#_Toc456527711)

[4.2.1. Test cases 31](#_Toc456527712)

[4.2.2. Special test cases 32](#_Toc456527713)

[5. Evaluation 33](#_Toc456527714)

[5.1. Group Work 33](#_Toc456527715)

[5.2. Task Responsibilities 33](#_Toc456527716)

# Specification

# Description

# Product Purpose

Our product is a cleaning agent searching software, featured with quick off-line searching, modifying and commenting about cleaning agent based on customers’ needs and favorites.

It is available on desktop computers and android mobile devices.

# Product Scope

Our product provides three different languages, including Chinese, English and German, containing most commonly used cleaning agent in China and Germany.

# Potential Customers

This product is specially designed for those who travel, work or study abroad, aiming to assist them to easily find the right cleaning agent in an unfamiliar cultural context.

# Product functions

* Search cleaning agents:
  + Search cleaning agents by tags.
  + Search cleaning agents by keywords.
  + Combined search with both tags and keyword. "And" logic is used.
* Add, modify and delete:
  + User can add, modify and delete a cleaning agent and a tag.
* Memo:
  + User can take a memo on a certain cleaning agent.
  + User can find all memos in the user center.
* Set preference:
  + The software serves an unique user on a same device.
  + User can log in with user name.
  + System would greet the user by the user name.
  + User can choose interface and content languages.
  + User can enable God mode and violate system protection.

# User characteristics

* Users are assumed to:
  + have basic computer operation knowledge, such as log in, GUI interaction and typing.
  + be uninfected with hearing, vision, touching and any other kinds of disability.
  + speak at least one of the following language: Chinese, English and German.
  + be oversea students, business or casual travelers in the following country: China and Germany.
  + have basic knowledge in choosing cleaning agent and cleaning.
  + have either a computer or an Android mobile phone.
  + have access to cleaning agents in nearby supermarket.
* Users need the software to:
  + find a cleaning agent by its purpose.
  + find a cleaning agent through its name, brand, description or other attributes.
  + get detail information of a cleaning agent
  + get multiple language contents of certain information.
  + add new cleaning agents and tags and take memos.

# Functional requirements

# Presentation

* Display the search result
  + Related cleaning agents shall be displayed in the form of a table with headers.
  + Each cleaning agent can be selected and the detail information of which can be displayed.
* Display the cleaning agent detail window
  + User cannot edit in detail window except taking memo.
  + There is an entry to modify the cleaning agent.
* Display tag modify window
  + Window shall highlight the tag or switch to a certain category tab based on user desire.

# Content

* Cleaning agent
  + Provide information including:  
    Cleaning agent name (in 3 languages);  
    Description (in 3 languages);  
    Instruction (in 3 languages);  
    Application time;  
    Frequency;  
    Rate;  
    Main language;  
    Image of the cleaning agent;
  + Special announcement:  
    Application time and frequency are only related to a certain cleaning agent. This information indicates the common use case of the cleaning agent. It is not instructional information of applying the cleaning agent in a specific scenario, but a general reference. Specific application time and frequency can be recored in "Instruction" field. We are not intending to teach user how to clean but assist the user to find the right cleaning agent.
  + Database arrangement:  
    Table CleaningAgents

|  |  |
| --- | --- |
| Column Name | SQL Type |
| cleaningAgentID | int |
| nameEn | string |
| nameCn | string |
| nameDe | string |
| descriptionEn | string |
| descriptionCn | string |
| descriptionDe | string |
| instructionEn | string |
| instructionCn | string |
| instructionDe | string |
| applicationTime | long |
| frequency | long |
| cleaningAgentType | boolean |
| rate | int |
| mainLanguage | int |
| image | image |
| memo | string |

* Tag
  + Provide information including:  
    Tag name (in 3 languages);  
    Tag type (room, items or others);
  + Database arrangement:  
    Table Tags:

|  |  |
| --- | --- |
| Column Name | SQL Type |
| tagID | int |
| nameEn | string |
| nameCn | string |
| nameDe | string |
| tagType | int |
| isSystem | boolean |

* Relation
  + Tag - cleaning agent relation  
    Stored in database.
  + Database arrangement:  
    Table TC:

|  |  |
| --- | --- |
| Column Name | SQL Type |
| cleaningAgentID | int |
| tagID | int |

* + Tag - tag relation  
    Generated according to the rule that two tags are related if they have at least one same related cleaning agent.

# Add, modify and delete

* Cleaning agents
  + New cleaning agent must have a name.
  + User cannot do anything to a system predefined cleaning agent unless God mode is enabled.
  + In God mode the user shall also be notified when editing a system predefined cleaning agent.
  + User has no restriction of manipulating a user defined cleaning agent.
  + User can either abort an edit by clicking "Cancel" or confirm an edit by clicking "Save".
  + User input shall be checked especially numbers.
* Tags
  + New tag must have a name.
  + User cannot do anything to a system predefined tag unless God mode is enabled.
  + In God mode the user shall also be notified when editing a system predefined tag.
  + User has no restriction of manipulating a user defined tag.
  + User will be warned when he or she try to add a tag which is already there.
* Memo
  + Memo and cleaning agent is one to one relation.
  + Memo is always and only related to a cleaning agent.
  + User can modify a memo in any case no matter whether the cleaning agent is system predefined or not.
  + All memos are listed in user center. The latest memo will be displayed in the first place.

# Search

* The keywords shall be:
  + Separated by common delimiters including but not limited to space, colon, semicolon and etc.
  + Language insensitive.
  + Case insensitive.
  + Partially compared.
  + Search will be cascaded from a cleaning agent to its elated tags and all other attributes.
  + Each pattern match of a cleaning agent contributes to its relevance index.
  + Full word match has higher weight of relevance.
  + Tag match has higher weight of relevance.
  + containing unrestricted number of words.
  + Clicking on "Clear" will remove keywords in input field.
* The tags shall be:
  + Separated in three categories which are "Rooms", "Items" and "Others".
  + selected only one in each category.
  + "And" logic is used between tags.
  + Clicking on "Clear" will remove tag selections.
* Search result shall be:
  + All cleaning agent if no tags or keywords are specified.
  + Combining tags and keywords result by "and" logic.
  + Ranked by relevance.
  + Shown with "origin", "name" and "tags" fields.
  + Presented in real-time during inserting keywords and selecting tags.
  + Limited to 100 in number.
  + Number of result will be displayed.

# Language

* The software GUI supports: Chinese, English and German.
* The content is prepared in: Chinese, English and German. Contents of different languages are provided in different tabs and can be switched during software run time.
* Changing language setting will result in software restart.
* Unspecified string of a certain language can and will always be replaced by a string of another language if there is one available.

# Non-functional requirements

# Portability

* The software has two distinctive sub versions which support desktop computers and Android devices.
  + Windows 10 and OS X El Capitan with Java SE Runtime Environment 8u91 are fully tested.
  + Android should run on 4.1–4.1.2 Jelly Bean (API level 16) or later.

# Performance and Availability

* The software shall start up and greet the user for 500ms.
* Search result shall be presented in less than 1 second.
* The software shall be always available during runtime.

# Security and Extensibility

* User data and system data are stored with database specific security.
* The software is not designed to extend.

# Development environment and framework

# Software

* Eclipse Java EE IDE for Web Developers
  + Version: Mars.2 Release (4.5.2)
  + Plugins: e(fx)clipse – IDE, version 2.3.0.201603030809
* JavaFX Scene Builder, version: 8.2.0
* Sqlite Expert Personal Edition, version: 4.1.2.651 (x64)
* GitHub Desktop, verision: 3.1.1.4

# Framework

* Java SE JDK, version: 1.8.0\_91
* JavaFX JDK, version: 8
* Sqlite JDBC Connector (.jar file), version: 3.8.11.2

# Reference

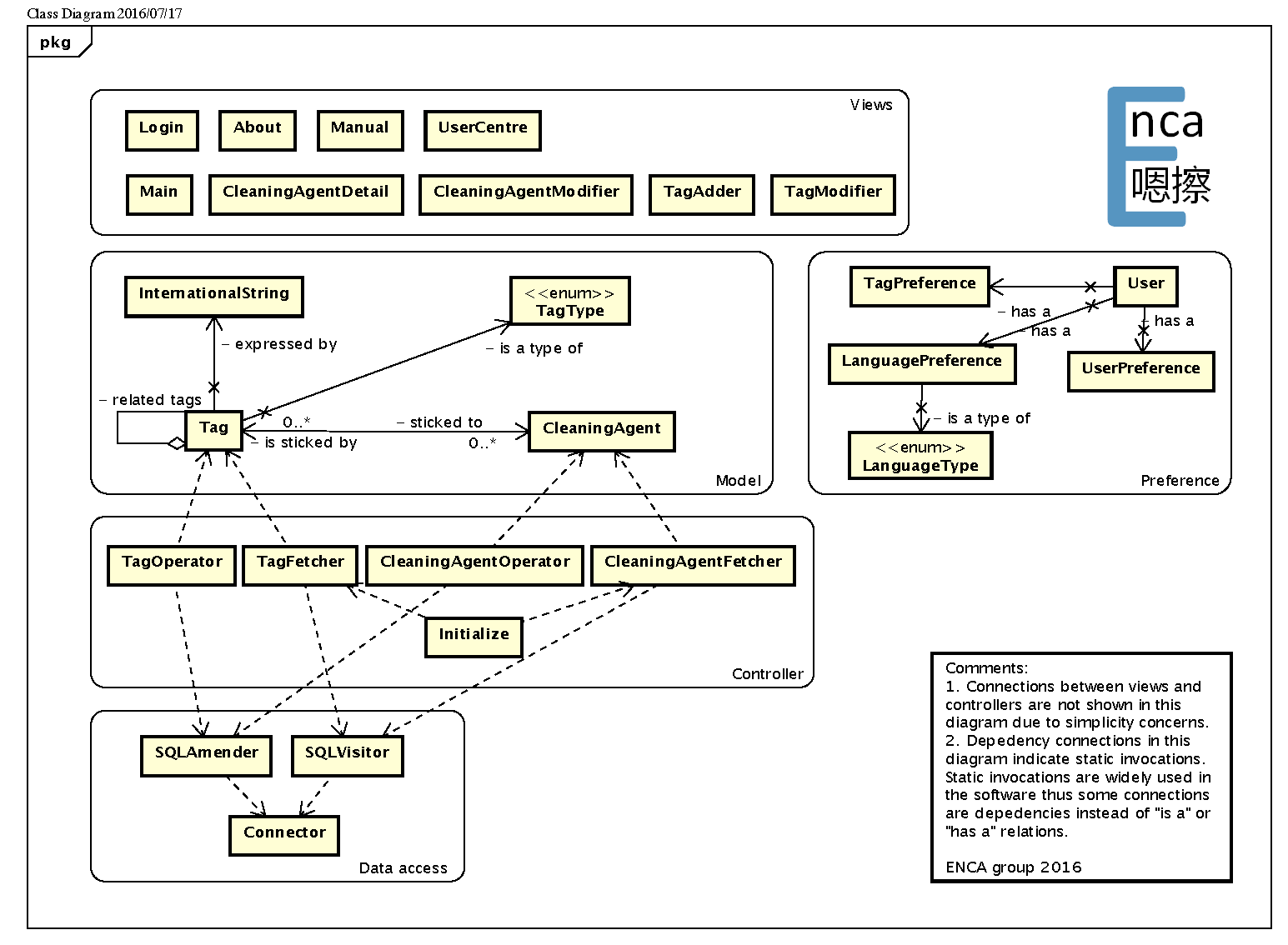
* Java™ Platform, Standard Edition 8 API Specification
  + <http://docs.oracle.com/javase/8/docs/api/>
* JavaFX 8 API Specification
  + <http://docs.oracle.com/javase/8/javafx/api/toc.htm>
* JavaFX CSS Reference Guide
  + <http://docs.oracle.com/javase/8/javafx/api/javafx/scene/doc-files/cssref.html>
* JavaFX 8 Tutorial
  + <http://code.makery.ch/library/javafx-8-tutorial/>

# UML Specification

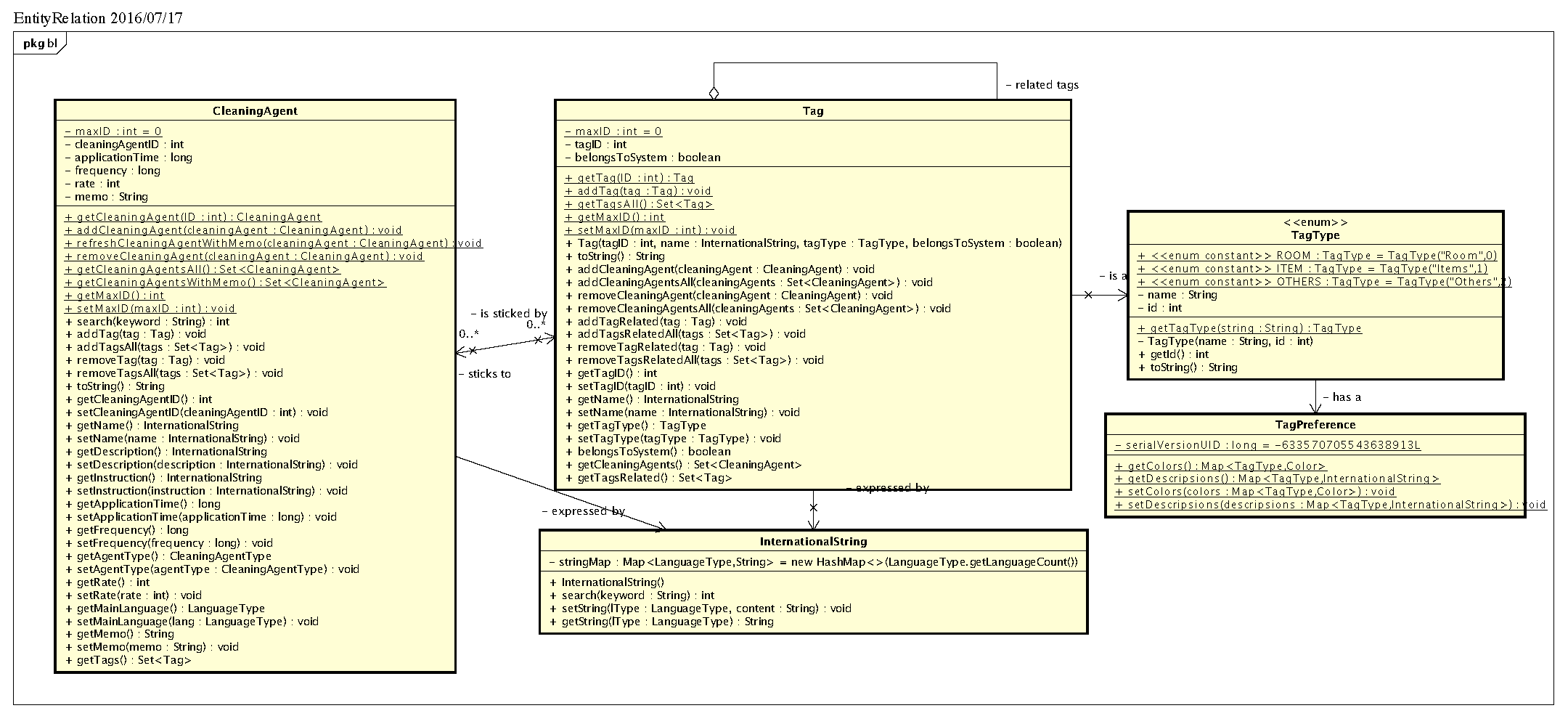
# Use Cases

# Class Diagrams

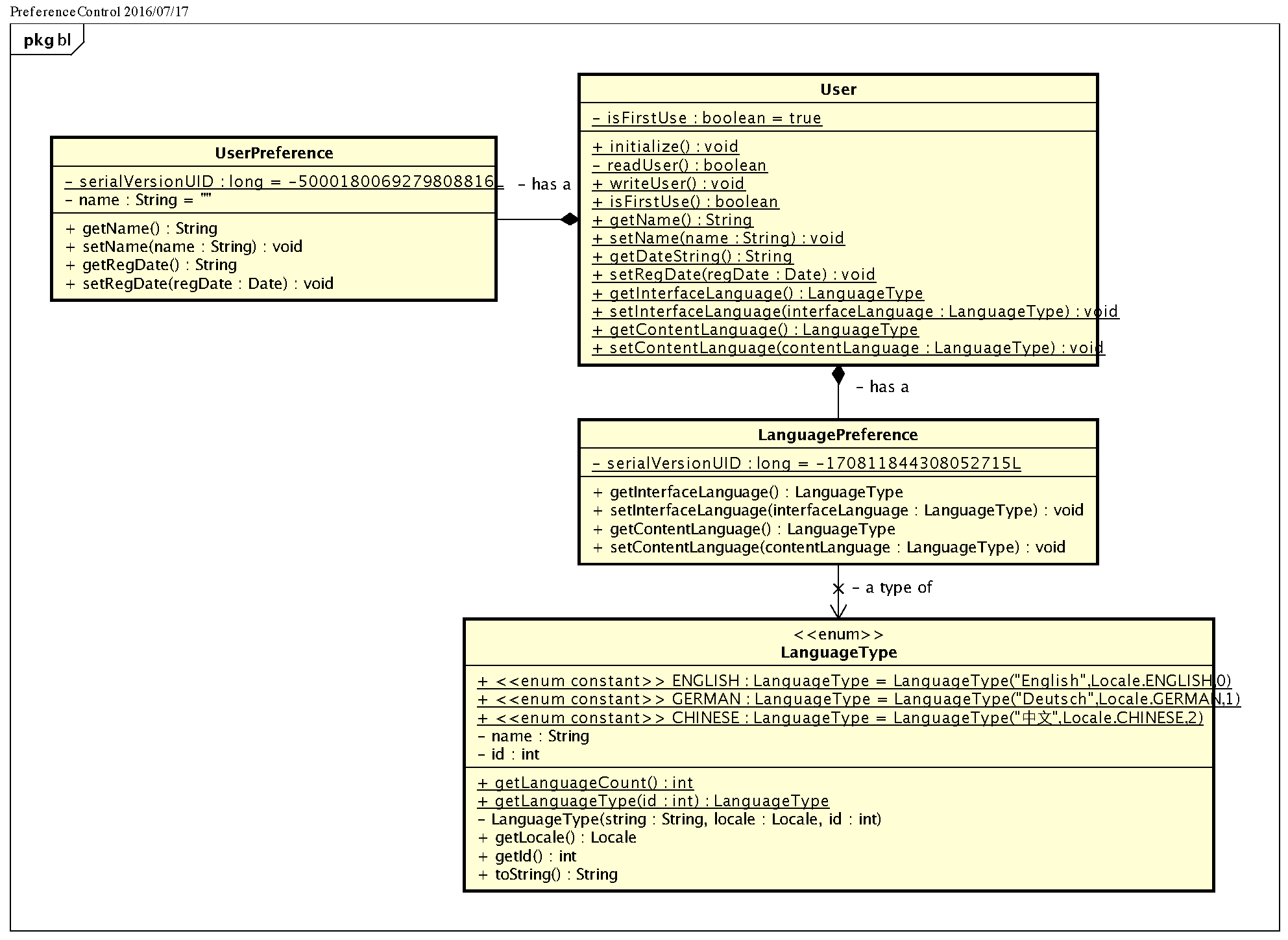
# Class Diagram



# Entity Relation



# Preference Control



# GUI Design

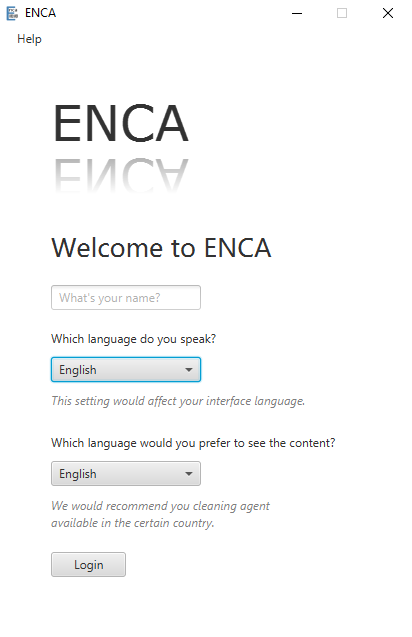
# PC Version

# Structure

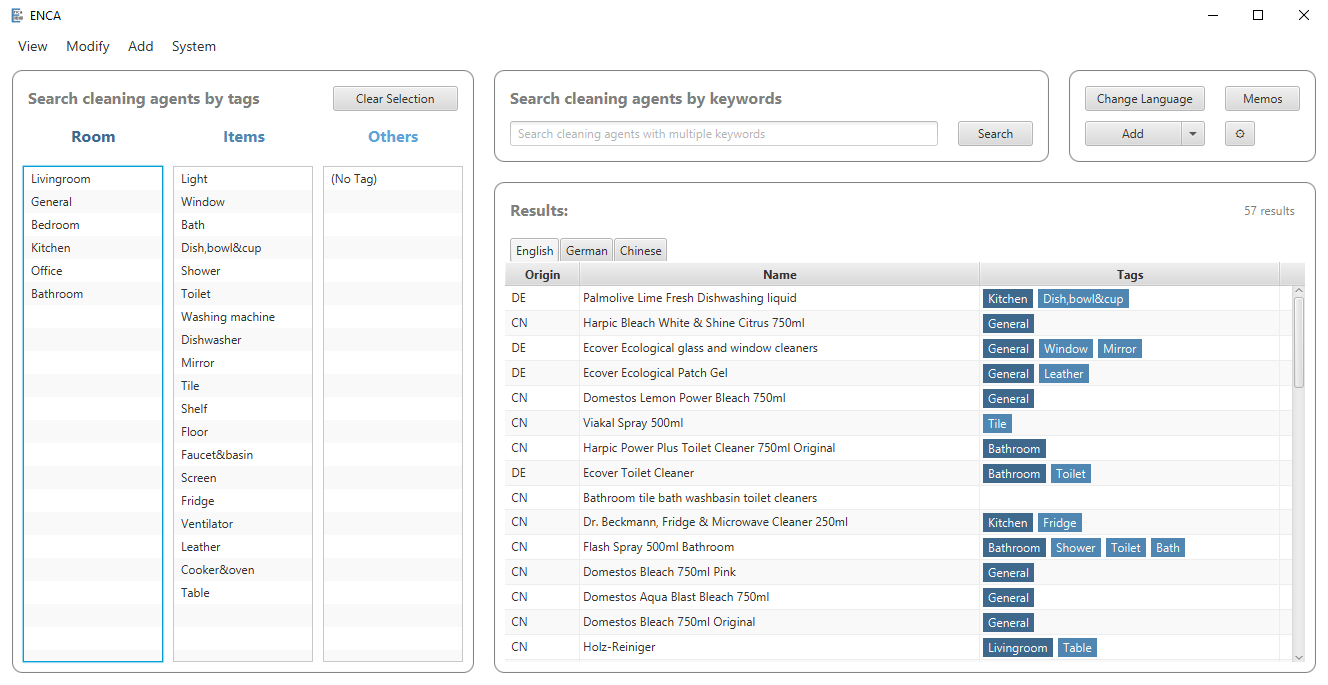


# Screenshots and description

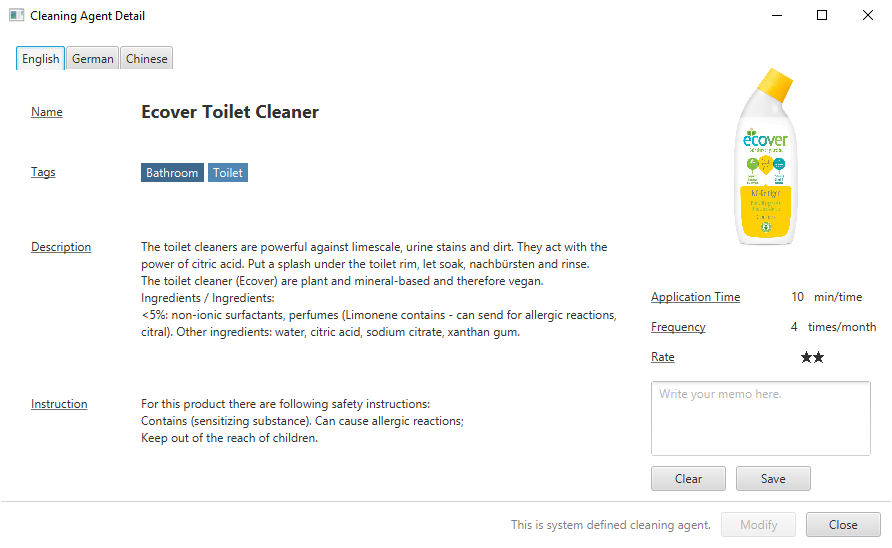
* Login
  + Screenshot:



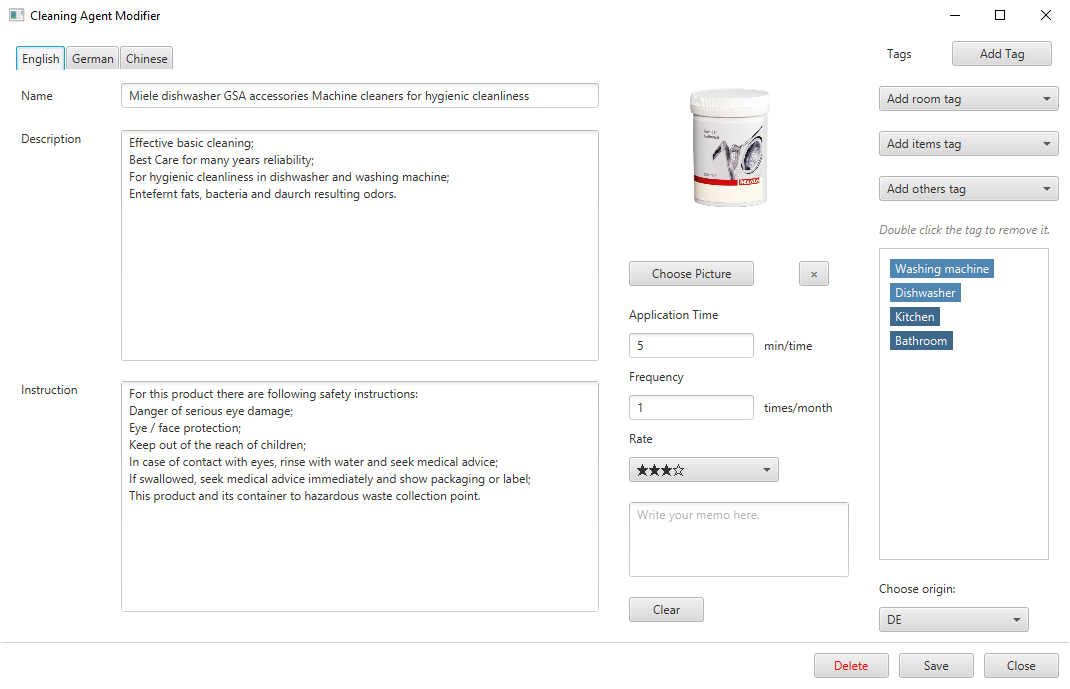
* + Description:  
    Interface for user to login.  
    Allow entering user name, choosing interface language and content language to meet user’s requirement. Meanwhile, manual and about interface are also provided through help menu, allowing user to grant basic information of the software before using it.
* Main
  + Screenshot:



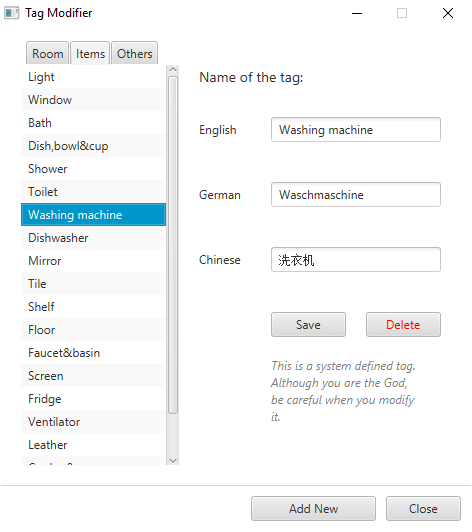
* + Description:   
    Main interface of the software.  
    Allow searching cleaning agents by tags or by keywords.  
    Information about origin, name and tags of cleaning agents as results will be shown in the tables.  
    When single tag is chosen, tags of other types will be filtered and only those related tags of the chosen tag will be shown.  
    Search result will be displayed in real time when user types in the keywords.  
    Multiple entries to other interfaces are implemented in buttons, context menus and menus in menu bar.  
    Size of this interface can be adjusted.
* Cleaning Agent Detail
  + Screenshot:



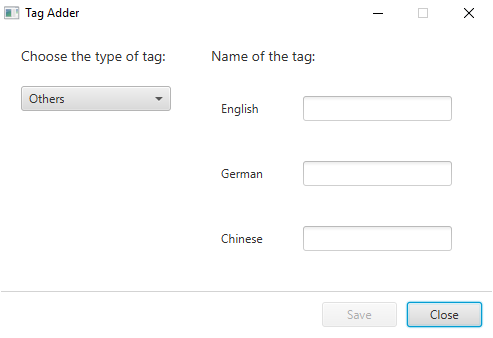
* + Description:   
    Interface showing detail of single cleaning agent.  
    Content of all three languages will be displayed simultaneously, as well as the image.  
    Function of writing memo is also provided in this interface and a dialog will be shown when the memo is saved.  
    Size of this interface can be adjusted.
* Cleaning Agent Modifier
  + Screenshot:



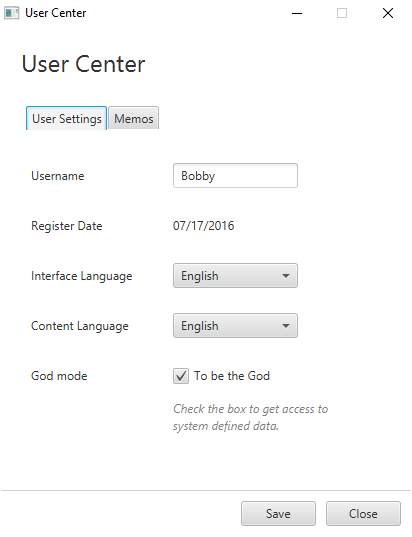
* + Description:   
    Interface allowing modifying, creating or deleting cleaning agent.  
    Content of all three languages can be modified simultaneously, as well as the image.  
    Relations between tags and cleaning agents can also be created and removed here.  
    Choose tag from three combo boxes to stick it to the current cleaning agent. Double click the tag to remove it.  
    Entry to tag adder is also provided allowing user to create new tag for the cleaning agent.  
    Before saving, boundary check including name, application time and frequency will be executed. Saving execution will be dropped if exception occurs during boundary check along with error message being shown.  
    After saving, this interface will be disposed and detail of this cleaning agent will be shown.
* Tag Modifier
  + Screenshot:



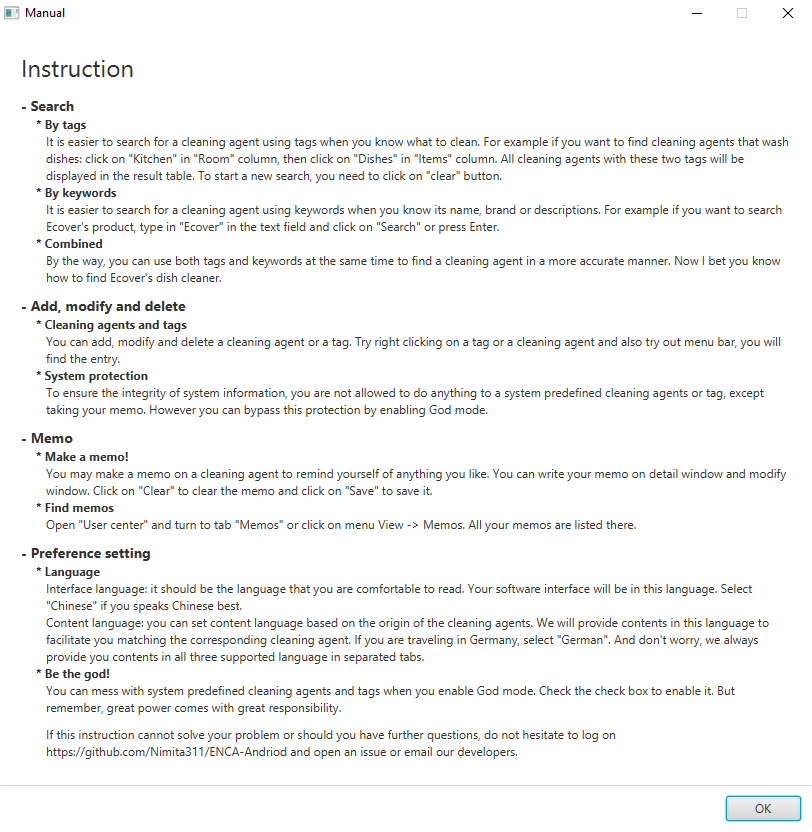
* + Description:   
    Interface allowing modifying and deleting tags.  
    The tag should be chosen in three lists on the left before modifying and deleting, then the names of the tags will be displayed on the right, allowing modifying.
* Tag Adder
  + Screenshot:



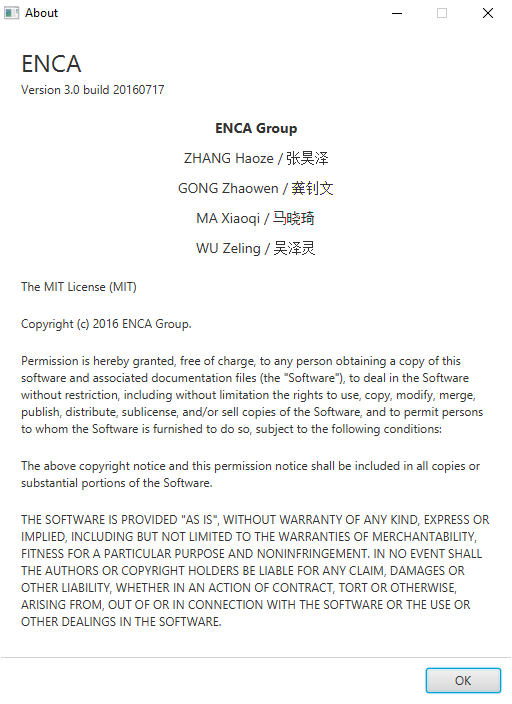
* + Description:   
    Interface allowing creating tags.  
    The type of the tag can be chosen on the left and the names of the tag can be assigned on the right.
* User Center
  + Screenshot:



* + Description:   
    Interface for settings and view memos.  
    Name, language prferences are shown and are allowed to modify in the user settings tab, along with the register date shown.  
    God mode can also be toggled here.  
    Cleaning agents with memos are shown by their images in memos tab. Those images are also entries to detail of those cleaning agents.
* Manual
  + Screenshot:



* + Description:   
    Interface showing instruction about how to use the software.  
    Will be shown automatically when user first uses the software.
* About
  + Screenshot:



* + Description:   
    Interface showing basic and legal information of the software.

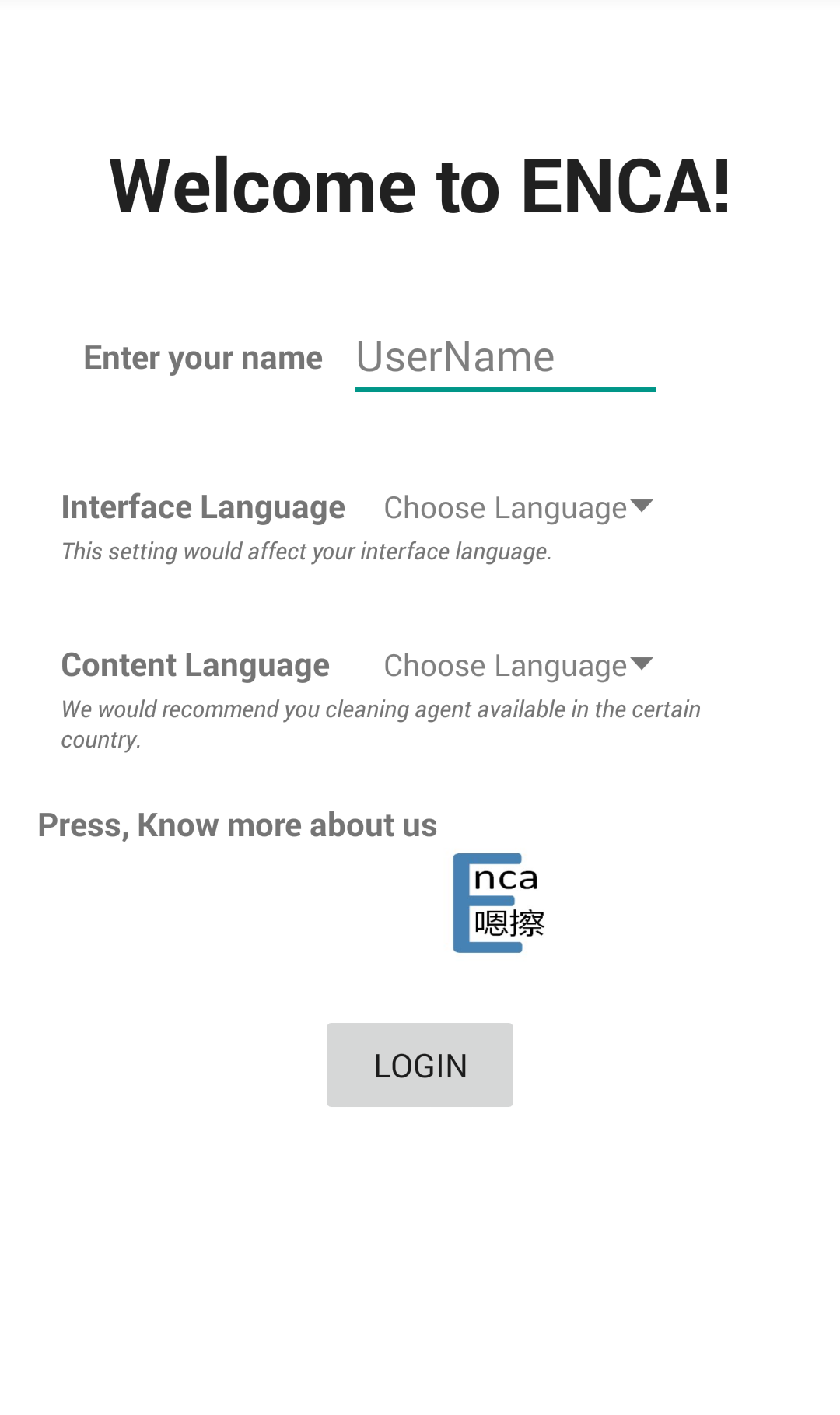
# Android Version

# Structure

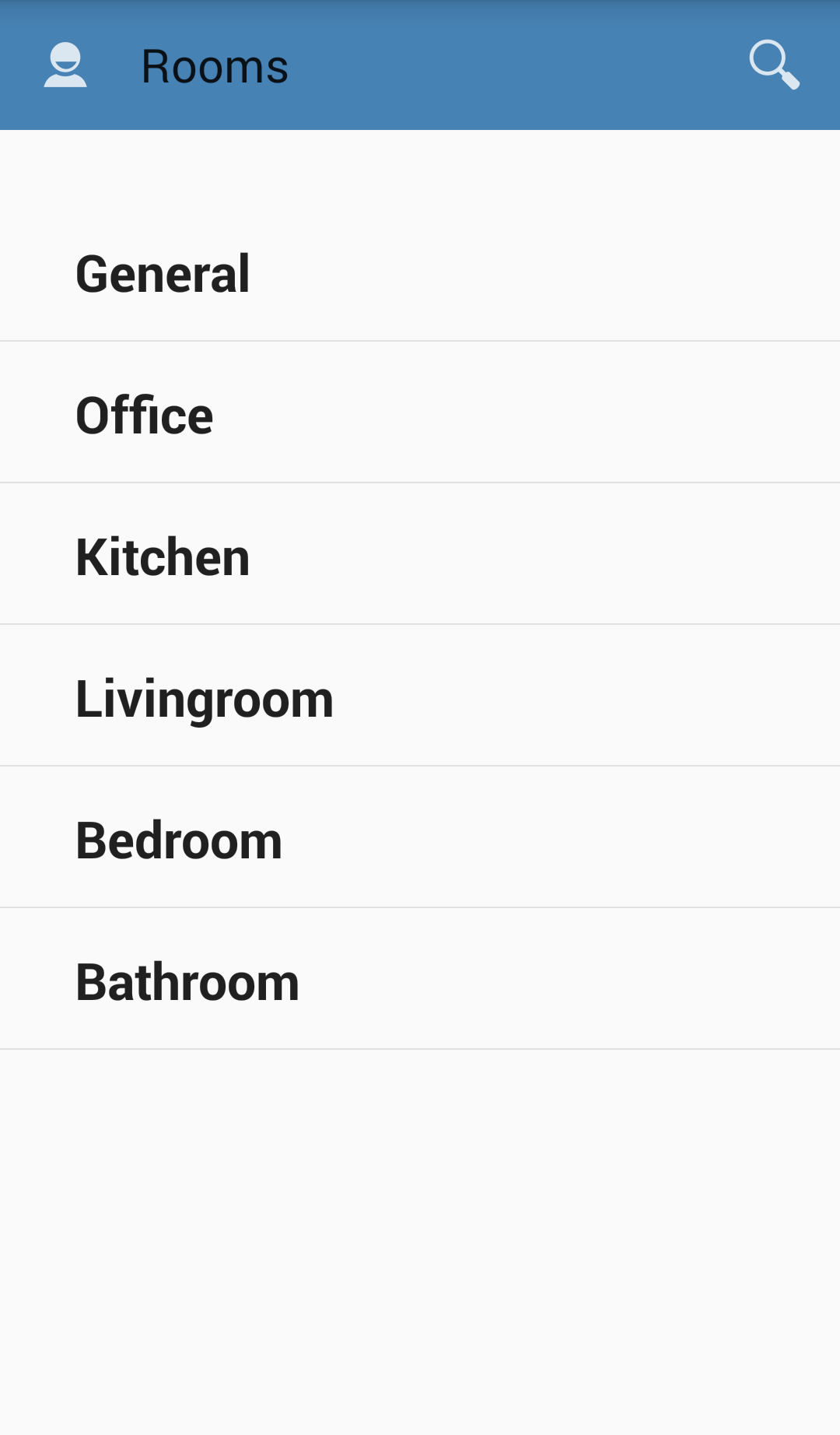


# Screenshots and description

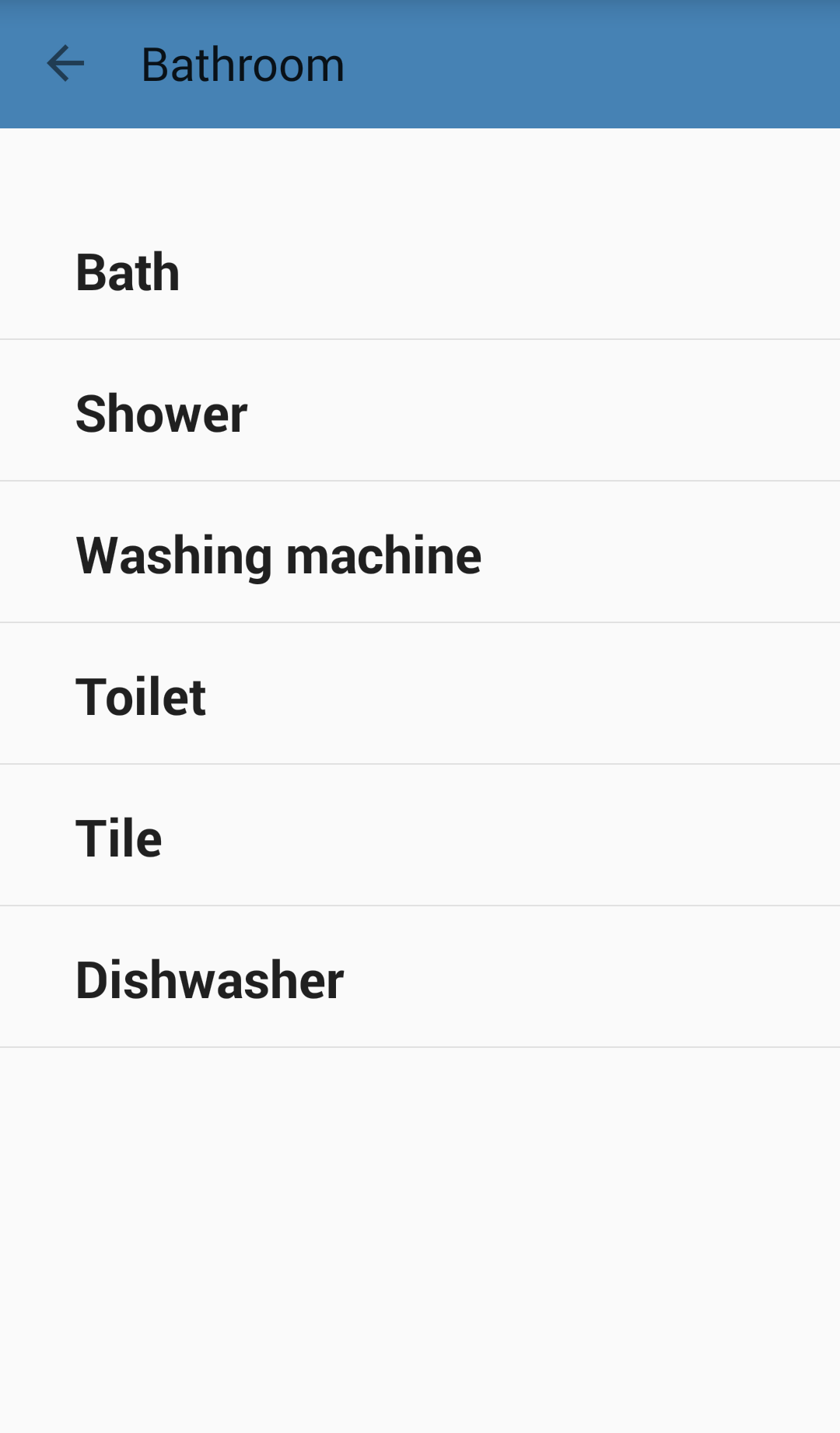
* Login
  + Screenshot:



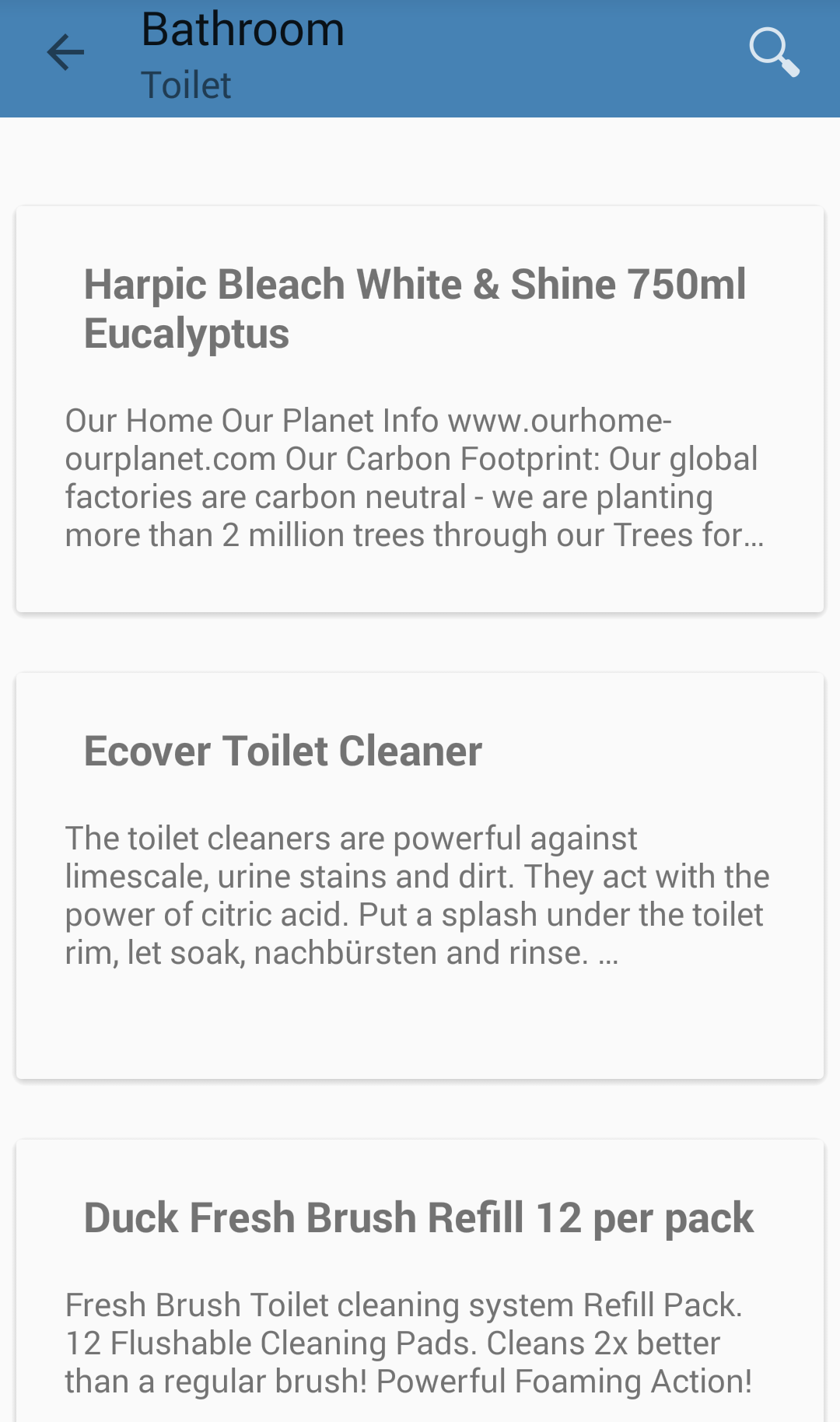
* + Description:  
    Enter user name, choose interface language and content language to meet user’s requirement. Meanwhile, press the logo and you will know more about our develop team.
* Room Selection
  + Screenshot:

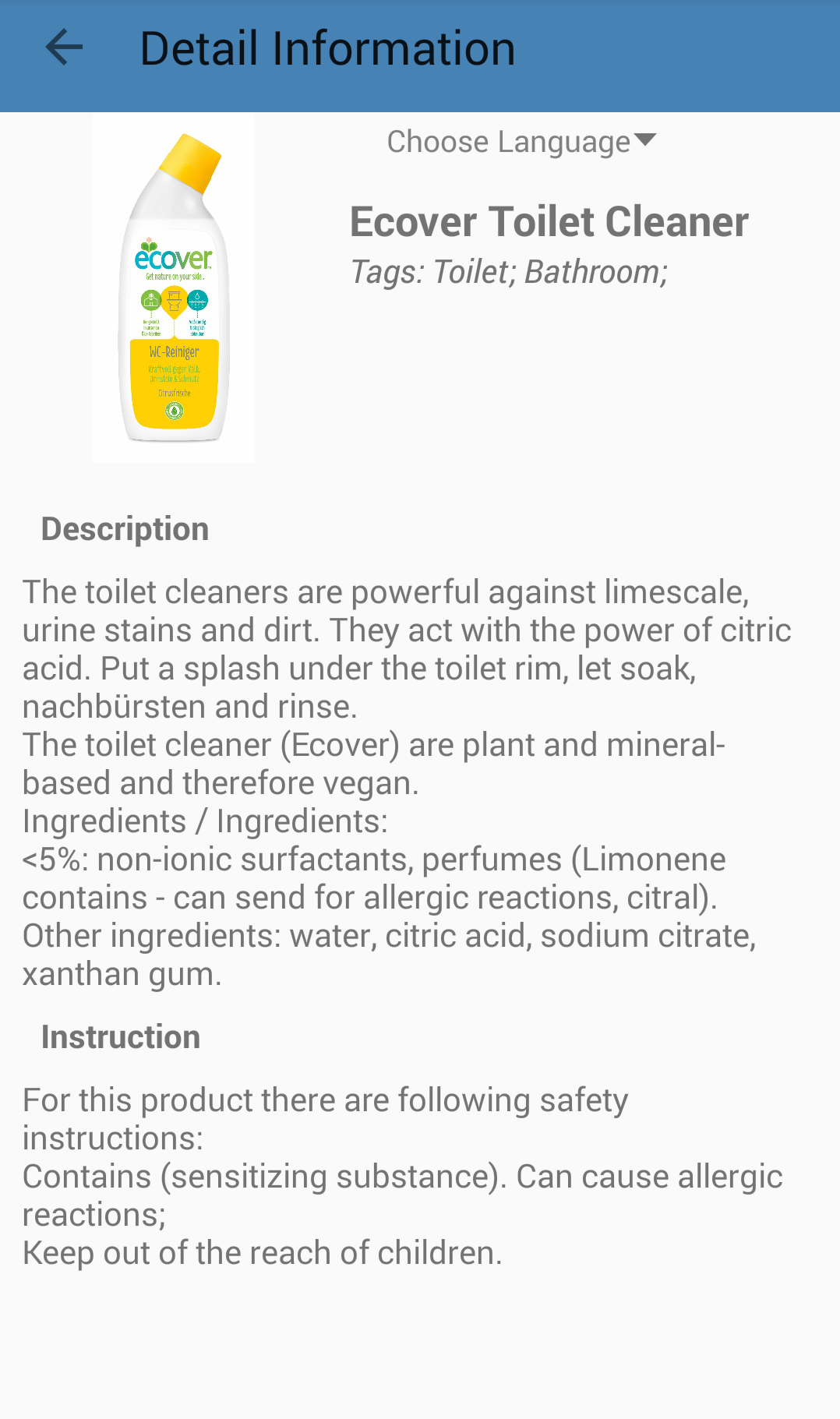


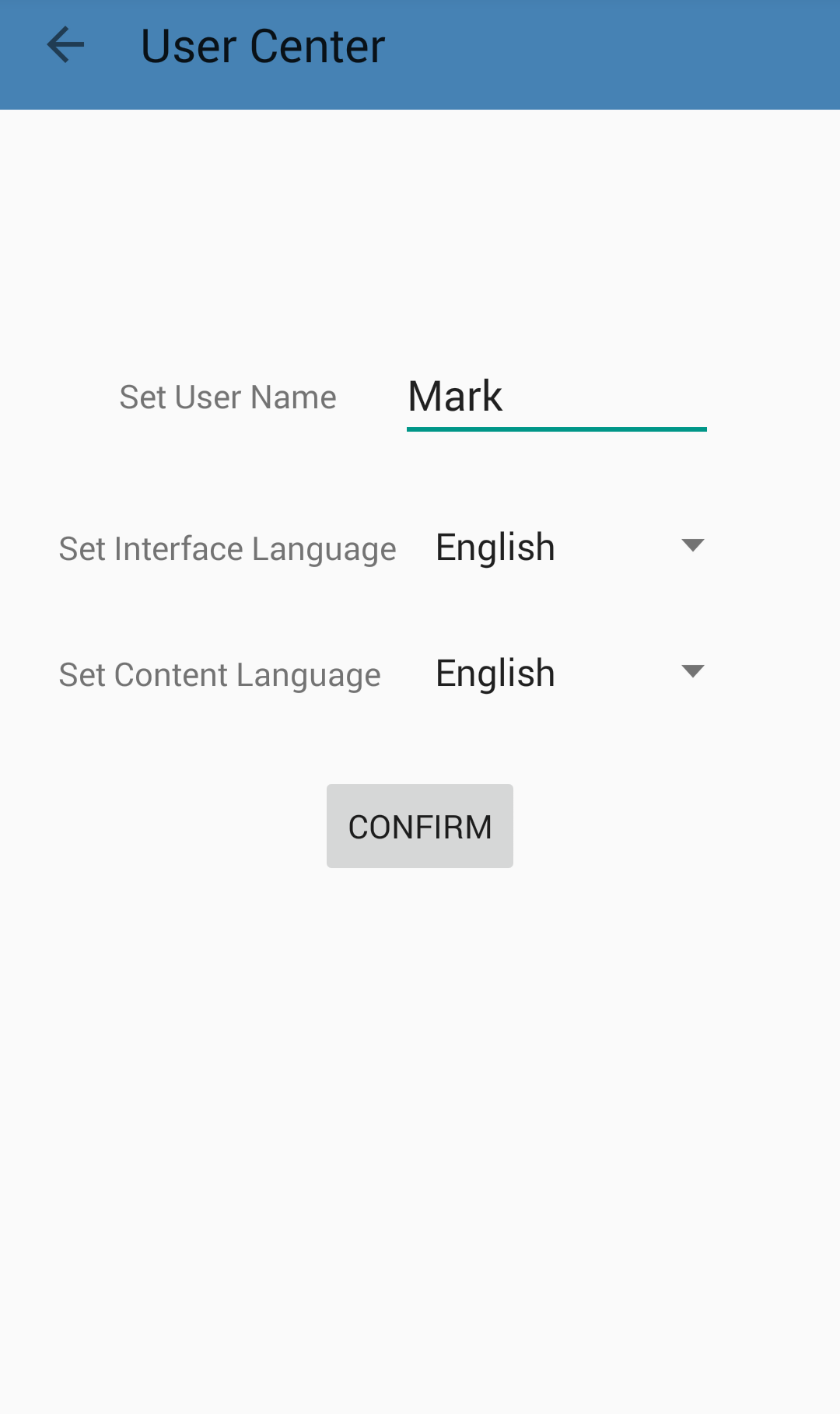
* + Description:  
    Choose one room you desire to clean. Meanwhile, if you click the “user icon”, you are allowed to enter User Center Activity to modify name and language; otherwise if you click search view, you are allowed to type anything concerns with the cleaning agent to facilitate to search for result.
* Item Selection
  + Screenshot:



* + Description:   
    Choose one item which belongs to the room you have already chosen.
* Cleaning Agent
  + Screenshot:



* + Description:  
    Presenting the result either by direct typing searching or by tags related searching.
* Cleaning Agent Detail
  + Screenshot:  
    
  + Description:  
    Present detail information about cleaning agent. In addition, you are allowed to change content language in this interface, in order to a better comparison. Present detail information about cleaning agent. In addition, you are allowed to change content language in this interface, in order to a better comparison.
* User Center
  + Screenshot:



* + Description:   
    Allows user to modify their name and choose interface language as well as content language to adapt to their requirement.

# Test

# Usability Test

# User Group

We choose Chinese international students in Germany as our test user group. Firstly they account for the majority of our target user group and moreover their knowledge of cleaning is somehow reflecting the average level.

# Test tasks

* Task 0:
  + Aim:  
    Test ‘first login’, ‘preference setting’ and ‘manual readability’.
  + Detail:  
    Brief the user of the software and show README to the user. Then user shall log in, fill out name and choose language preference. Additionally, read manual.
* Task 1:
  + Aim:  
    Test ‘searching by keywords’ and ‘memo taking’.
  + Detail:  
    User shall find a certain cleaning agent and make a memo as "to buy" on it.
  + Target CA:  
    [German name: "AJAX Frischeduft"] (Opening the detail window of this CA is considered to be task accomplished)
* Task 2:
  + Aim:  
    Test ‘searching by tag’ and ‘memo taking’.
  + Detail:  
    User shall be given a cleaning purpose and find a suitable CA and make a memo as "to buy".
  + Target CA:  
    [Purpose: Clean toilet] (Opening any detail window of CAs which have tags "Bathroom" and "Toilet" is considered to be task accomplished)
* Task 3:
  + Aim:  
    Test ‘adding CA’ and ‘adding tag’.
  + Detail:  
    create a CA with given information and stick a new tag "interested"
  + Target URL:   
    <https://www.amazon.de/Bissell-1137E-Remover-Enzyme-Cleaning/dp/B00DPVNTG8/ref=sr_1_4?ie=UTF8&qid=1467900739&sr=8-4&keywords=cleaning+agent>  
    (Hitting save button on the modify window is considered to be task accomplished)
* Task 4:
  + Aim:  
    Test ‘deleting CA’ and ‘deleting tag’.
  + Detail:  
    delete the CA and tag which have just been created.
* Task 5:
  + Aim:  
    Test ‘viewing memo’ or ‘user center’.
  + Detail:  
    Find the CAs with notes. (Opening the user center memo tab is considered to be task accomplished)

# Equivalent classes and boundary tests

# Test cases

* Application time and frequency are required when modifying and creating cleaning agents. Valid input shall be a positive number or left empty.
  + Valid EC:  
    V1 = {value | value is null}  
    V2 = {value | value = "5"}
  + Invalid EC:  
    V3 = {value | value = "abc"}  
    V4 = {value | value = "-5"}  
    V5 = {value | value = "0"}
* Name cannot be empty when creating a new cleaning agent. Valid input would be a not null cleaning agent name.
  + Valid EC:  
    V1 = {name filled, other fields not filled}
  + Invalid EC:  
    V2 = {name not filled, other fields filled}

# Special test cases

* Keywords are used in regular expression patterns, thus special charecters in the keywords such as "\*.()[]{}" shall be escaped. Keywords with such characters shall be input in a boundary value test.
  + Input: [Keywords: ".\*"]
* Any content that is eventually written to database may affect SQL query, thus charecters such as ' shall be escaped. Test of saving a field which contains ' into the database shall be performed.
  + Input: [Tag name: "Mary's favourite"]

# Evaluation

# Group Work

In ENCA group, every member shows great passion, expertise and profession in the project. Our members are truly original and critical thinkers and strict practitioners.

We strongly stick to the process of real-life software development, achieving phases such as modelling, testing, refactoring and etc as professional as possible. We witness our progress in every iteration. It is the most delightful thing to finally put our software for delivery.

We base our team collaboration on GitHub. Check it out: <https://github.com/Nimita311/ENCA-Andriod>

Our progress milestones and agenda are as followed.

|  |  |
| --- | --- |
| Date | Arrangement |
| 2016-05-09 | First get-together |
| 2016-05-14 | Spec discussion |
| 2016-05-16 | Structure design |
| 2016-05-17 | \*Spec deadline |
| 2016-05-21 | Supermarket trip, cleaning agents investigation;  Class design |
| 2016-05-28 | Database and tag spec meeting |
| 2016-05-31 | \*UML deadline |
| 2016-06-10 | GUI monk |
| 2016-06-14 | \*GUI deadline |
| 2016-06-18 | GUI theme issues |
| 2016-06-25 | Database ready |
| 2016-06-28 | Orientation issues |
| 2016-06-30 | Prototype ready |
| 2016-07-01 | Usability issue; logic finished |
| 2016-07-04 | JUnit finished |
| 2016-07-10 | Usability test |
| 2016-07-15 | Usability enhanced; usability test |
| 2016-07-17 | \*Project deadline |

# Task Responsibilities

We set coherent division of responsibilities, yet our collaboration flows just as smooth. Division of responsibilities is as followed.

* Haoze Zhang:
  + Requirement analysis;
  + System structure;
  + UI design;
  + Team collaboration.
* Zhaowen Gong:
  + System structure;
  + Business logic implementation;
  + PC UI design and implementation;
  + Code revision.
* Zeling Wu:
  + Data acquisition;
  + Database management;
  + Junit testing.
* Xiaoqi Ma:
  + Android application;
  + Android UI design and implementation.