Software Engineering II Project

Cleaning Help: ENCA

Part task: Android

Group 1

Haoze Zhang

Zhaowen Gong

Zelin Wu

Xiaoqi Ma

Content

1. Specification

1.1. Description

1.1.1. 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.

1.1.2. Product Scope

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

1.1.3. 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.

1.2. 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 a 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.

1.3. 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 travellers 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.

1.4. Functional requirements

1.4.1. 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.

1.4.2. 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 recorded 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

  - A default tag called "(No Tag)" is always generated, and is connected to those cleaning agents which have no related tags.

- 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.

1.4.3.   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.

1.4.4.   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 ("or" logic is applied between each keyword).
- 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.

1.4.5. 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.

1.5. Non-functional requirements

1.5.1. 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.

1.5.2.   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.

- All of the data stored in database except image will be read into memory during the process of initialization. This policy may consume more memory when the software is running, but dramatically increase the search performance with the assistance of tag – cleaning agent relations and tag – tag relations. Otherwise, disk will be read more times when executing search and the overall performance of the software will drop.

1.5.3.   Security and Extensibility

- User data and system data are stored with database specific security.

- Assume that we want to exchange and extend our software:

  - Scenario 1: To add language we only need to:
    Add element to language type;
    Change international string;
    Add one column for each international string in database.
  - Scenario 2: To add attribute to tag or cleaning agent we only need to:
    Change database;
    Change class SQLvisitor;
    Change view.
  - Conclusion: Not much work is needed when extending the software. Extensibility is good.

- The software is not designed for third party to extend.

1.5.4.   Miscellaneous

- Preference file will be written to disk and be read during the start of the software. Preference including user settings, language choice and tag colors will be recorded and stored.

- Database located in .jar file will be copied to the disk when the software is run for the first time. In case the software is lost or deleted, a new default database will always be generated.

1.6.   Development environment and framework

1.6.1.   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)

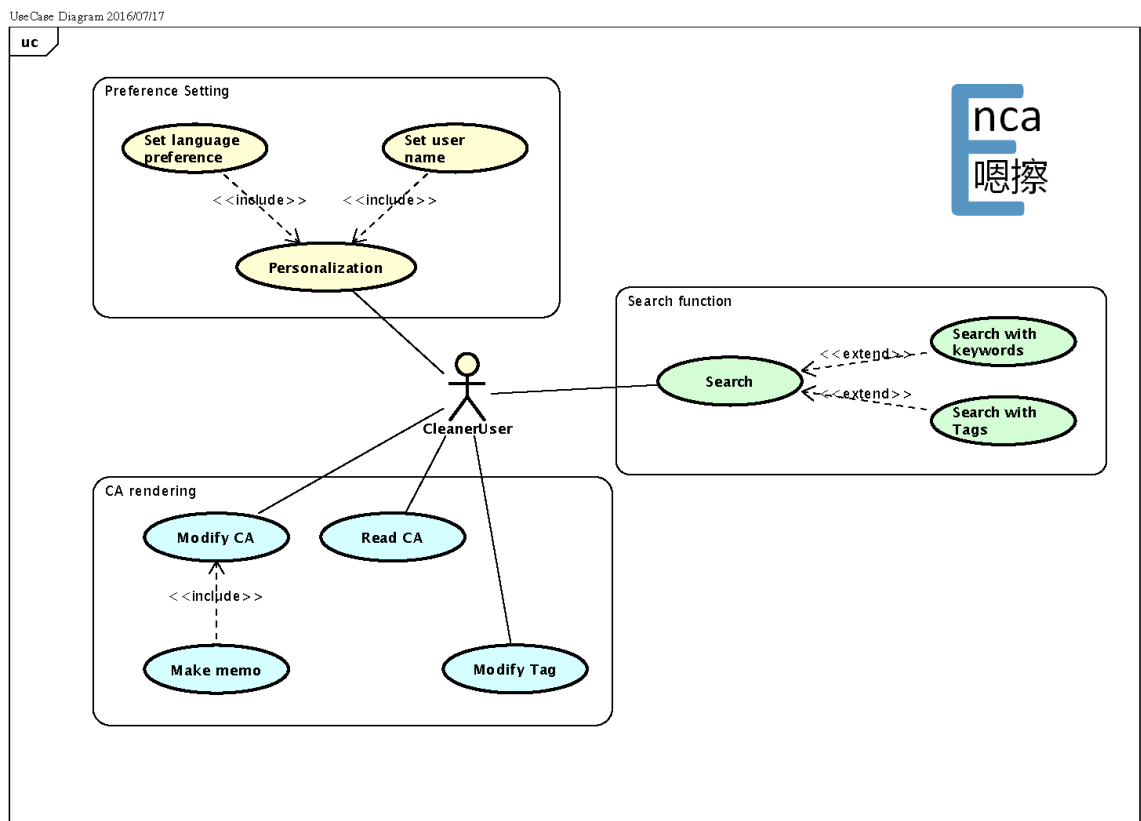- GitHub Desktop, verision: 3.1.1.4

1.6.2.   Framework

- Java SE JDK, version: 1.8.0_91

- JavaFX JDK, version: 8

- Sqlite JDBC Connector (.jar file), version: 3.8.11.2

1.6.3.   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/

2. UML Specification

2.1. Use Cases

2.2. Class Diagrams

2.2.1. Class Diagram
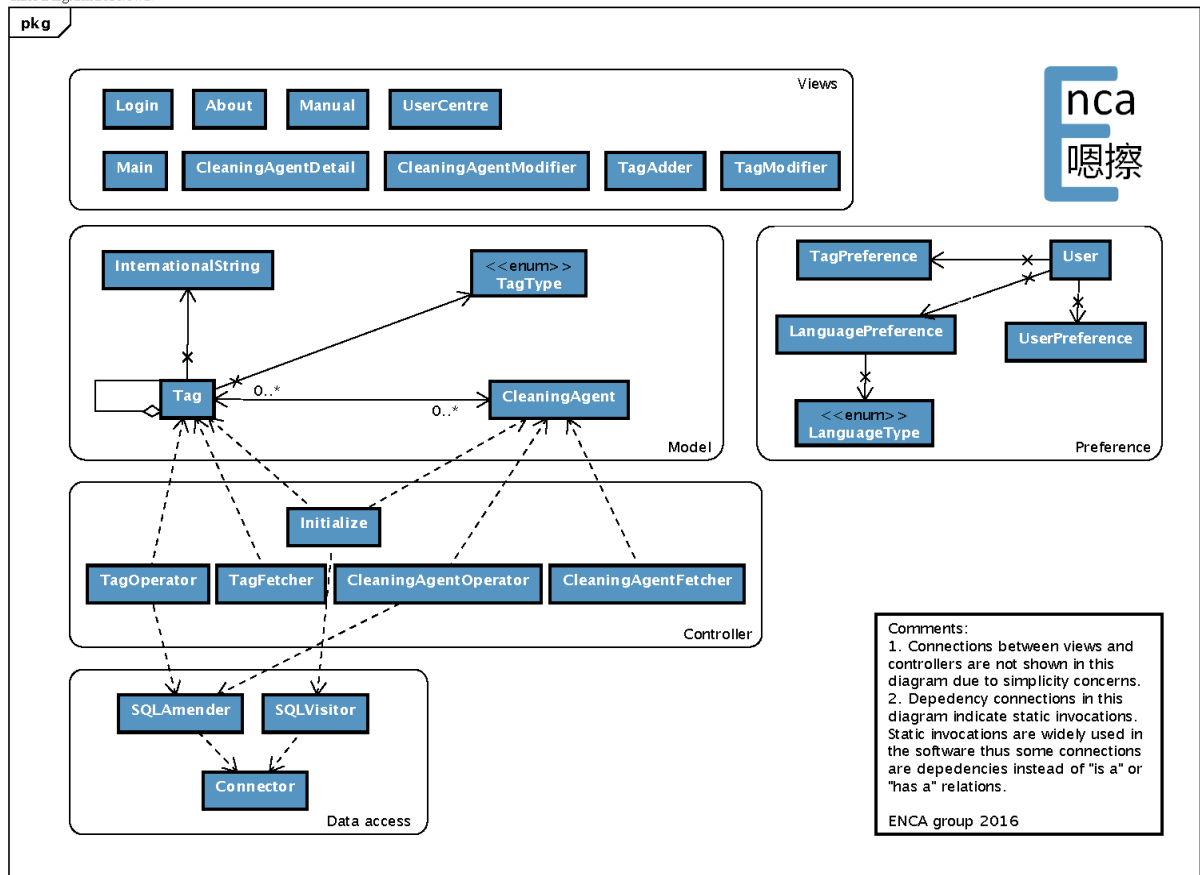
2.2.2. Entity Relation
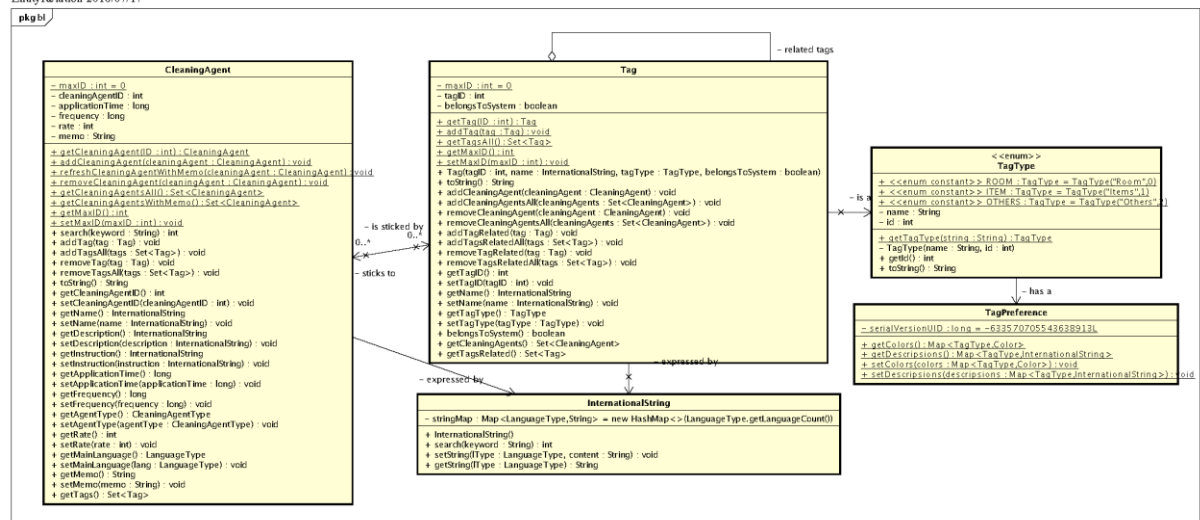
2.2.3. Preference Control

Preference Control 2016/07/17

pkg bl

UserPreference

– serialVersionUID : long = –5000180069279808816L
– name : String = ""

+ getName() : String
+ setName(name : String) : void
+ getRegDate() : String
+ setRegDate(regDate : Date) : void

– has a

User

– isFirstUse : boolean = true

+ initialize() : void
– readUser() : boolean
+ writeUser() : void
+ isFirstUse() : boolean
+ getName() : String
+ setName(name : String) : void
+ getDateString() : String
+ setRegDate(regDate : Date) : void
+ getInterfaceLanguage() : LanguageType
+ setInterfaceLanguage(interfaceLanguage : LanguageType) : void
+ getContentLanguage() : LanguageType
+ setContentLanguage(contentLanguage : LanguageType) : void

– has a

LanguagePreference

– serialVersionUID : long = –170811844308052715L

+ getInterfaceLanguage() : LanguageType
+ setInterfaceLanguage(interfaceLanguage : LanguageType) : void
+ getContentLanguage() : LanguageType
+ setContentLanguage(contentLanguage : LanguageType) : void

– a type of

<<enum>>
LanguageType

+ <<enum constant>> ENGLISH : LanguageType = LanguageType("English",Locale.ENGLISH,0)
+ <<enum constant>> GERMAN : LanguageType = LanguageType("Deutsch",Locale.GERMAN,1)
+ <<enum constant>> CHINESE : LanguageType = LanguageType("中文",Locale.CHINESE,2)
– name : String
– id : int

+ getLanguageCount() : int
+ getLanguageType(id : int) : LanguageType
– LanguageType(string : String, locale : Locale, id : int)
+ getLocale() : Locale
+ getId() : int
+ toString() : String

3. GUI Design

We apply flat design style through our software. The theme color of our software is blue. White background, blue labels representing tags and grey controls and borders are implemented across all the interfaces, in order to give user a simple look and guide user to focus on the soundness of the functions.

In terms of usability, redundant entries are implemented allowing user to user our software in their habits, and useful tips and dialogs are shown when needed. Besides, three of our main interfaces are able to be size-adjustable.

Consistency within the software and across the platform is deeply concerned when designing GUI. Every interface in a platform have the same design style and layout, and we try out best to maintain this consistency between the PC version and Android version.
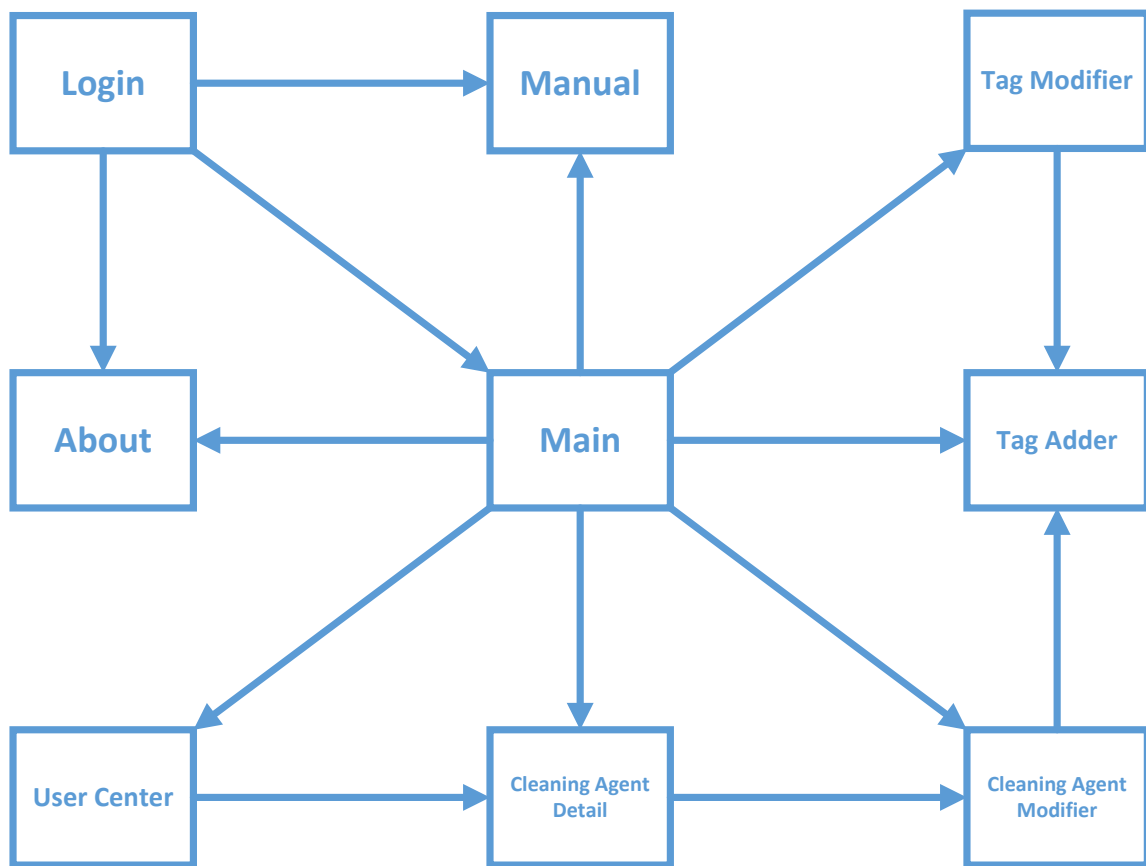
3.1. PC Version

The PC version is designed and implemented under JavaFX 8 framework.

JavaFX is intended to replace Swing as the standard GUI library, and now is implemented as a native Java library, included in Java SE runtime and development kit.

The default design of JavaFX is better than Swing, and JavaFX supports more controls and controls with more powerful functions. The layout of the JavaFX is also different from those in Swing, granting us more flexibility when designing the interface.

JavaFX distinguishes view, model and controller in the framework level. The view which contains information about static layout, controls and content on the interface, is implemented as a FXML file (.fxml), which can be designed using JavaFX Scene Builder. The controller which contains the dynamic actions of the controls is implemented as a Java class, using @FXML annotation to connect members and methods to the view automatically. The model is often implemented as a Java Bean, suitable for being displayed and transmitted on the interface. Reflection is usually used when getting the properties of the model.
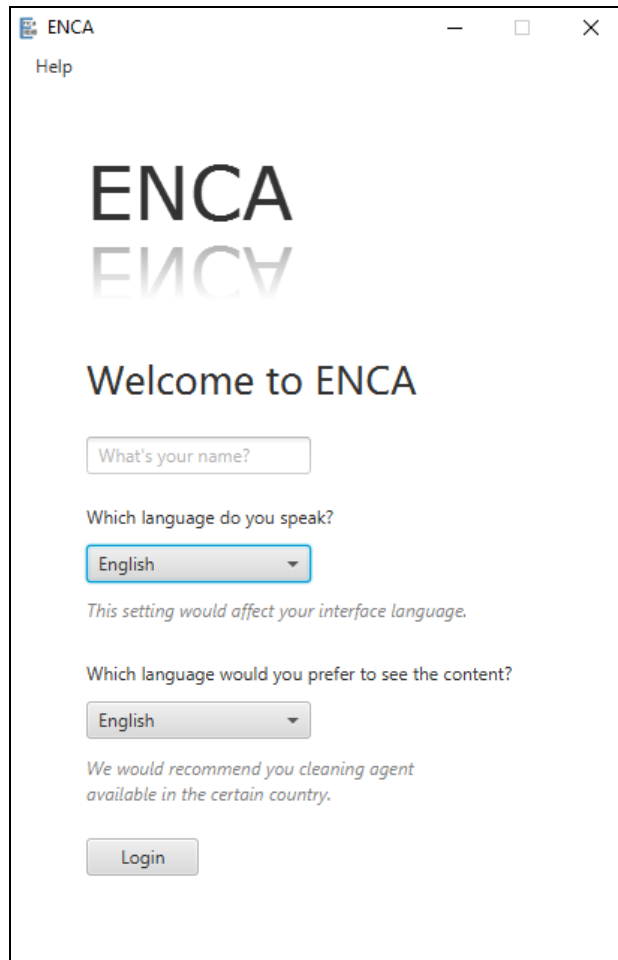
3.1.1.   Structure


```
┌─────────┐              ┌─────────┐              ┌──────────────┐
│  Login  │─────────────▶│ Manual  │              │ Tag Modifier │
└─────────┘              └─────────┘              └──────────────┘
     │                        ▲                          │
     │                        │                          │
     ▼                        │                          ▼
┌─────────┐              ┌─────────┐              ┌──────────────┐
│  About  │◀─────────────│  Main   │─────────────▶│  Tag Adder   │
└─────────┘              └─────────┘              └──────────────┘
                              │                          ▲
                              ▼                          │
┌─────────────┐  ┌──────────────────┐  ┌──────────────────────┐
│ User Center │─▶│ Cleaning Agent   │─▶│   Cleaning Agent     │
│             │  │     Detail       │  │      Modifier        │
└─────────────┘  └──────────────────┘  └──────────────────────┘
```

────────────▶ Indicates how user can jump from one interface to another

3.1.2.   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:
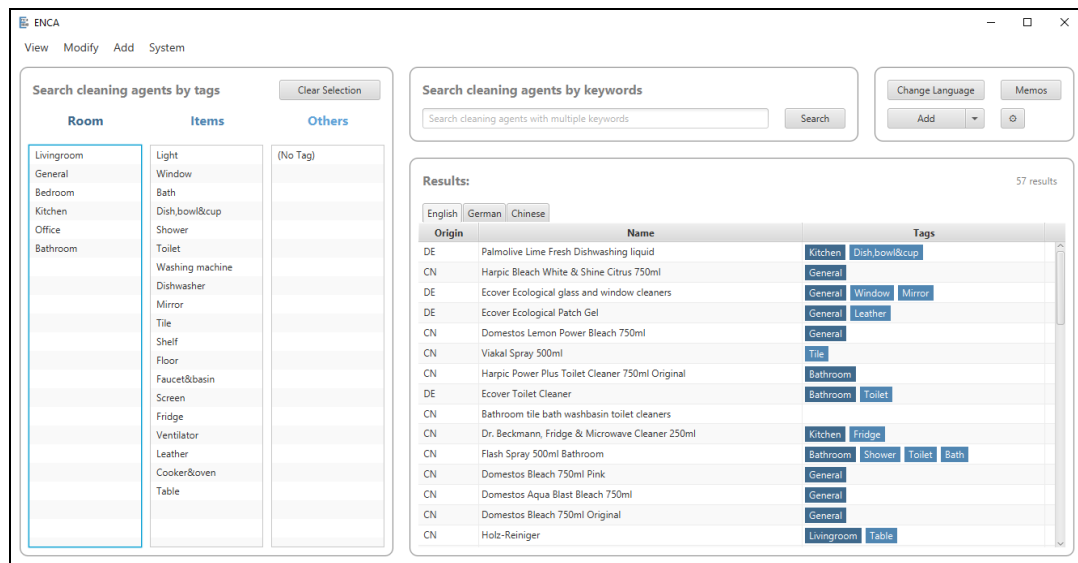
  - Description:

    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.
    Size of this interface can be adjusted.

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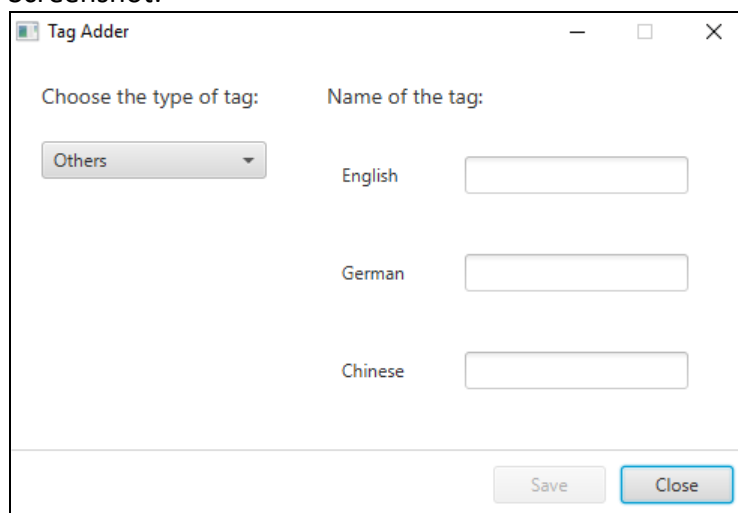
  - Screenshot:
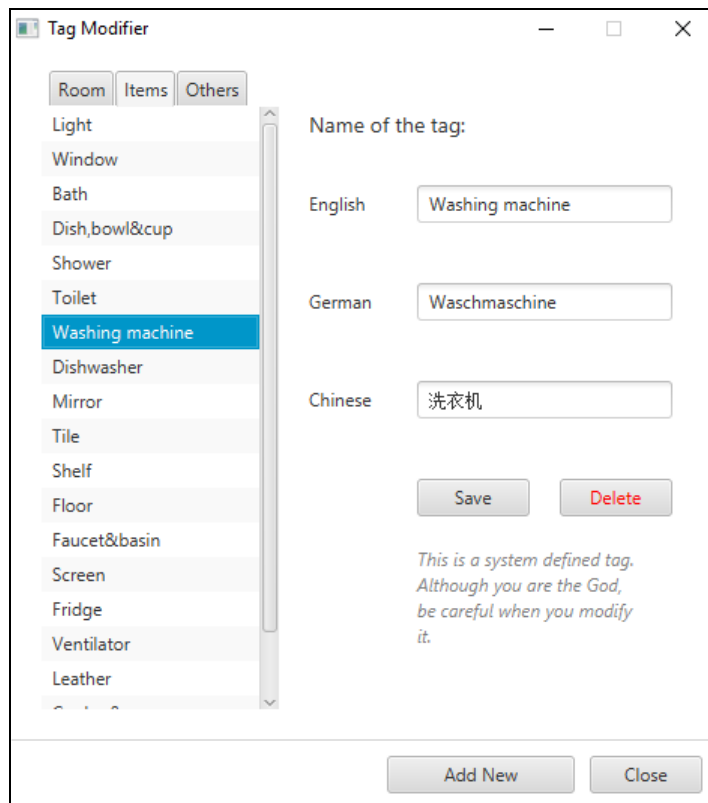
  - Description:
    Interface allowing modifying and deleting tags.
    The tag should be chosen in three lists on the left before modifying and deleting,
    then the names of the tags will be displayed on the right, allowing modifying.
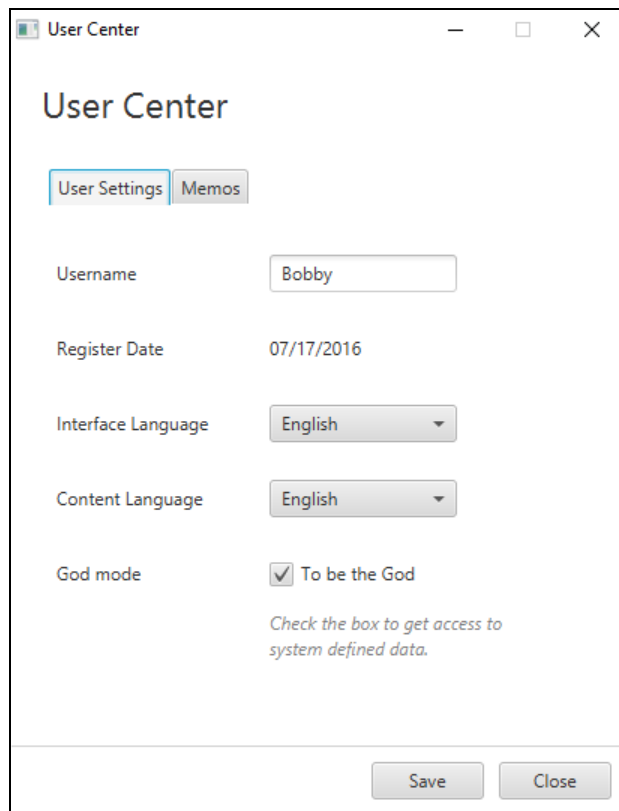
- Tag Adder

  - 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:
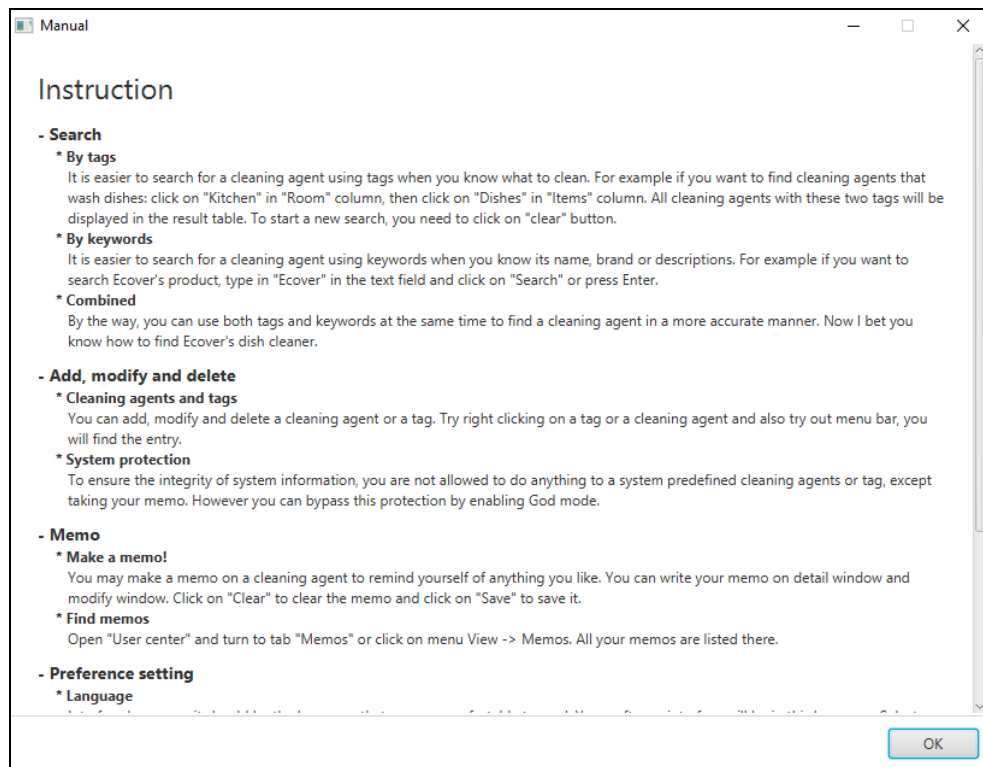   Interface for settings and view memos.
    Name, language preferences are shown and are allowed to modify in the user settings tab, along with the register date shown.
    God mode can also be toggled here.
    Cleaning agents with memos are shown by their images in memos tab. Those images are also entries 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.
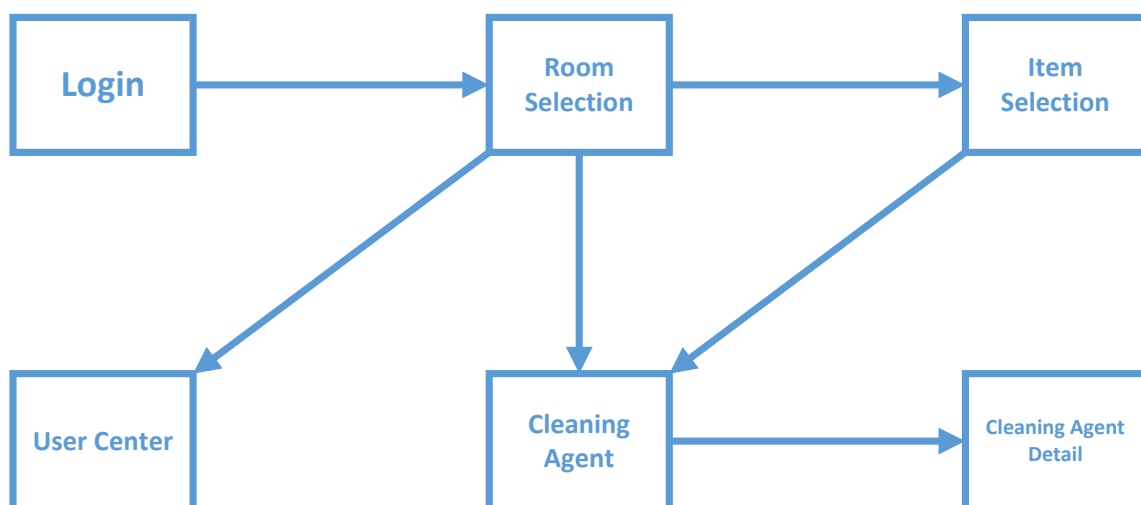
- About
  - Screenshot:

  - Description:
    Interface showing basic and legal information of the software.
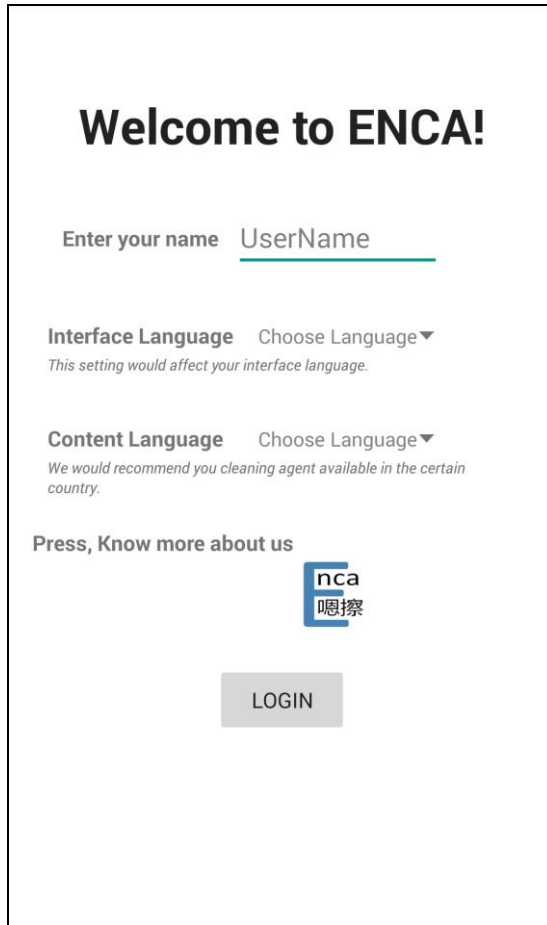
3.2. Android Version

3.2.1. Structure

Indicates how user can jump from one interface to another

3.2.2. 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:
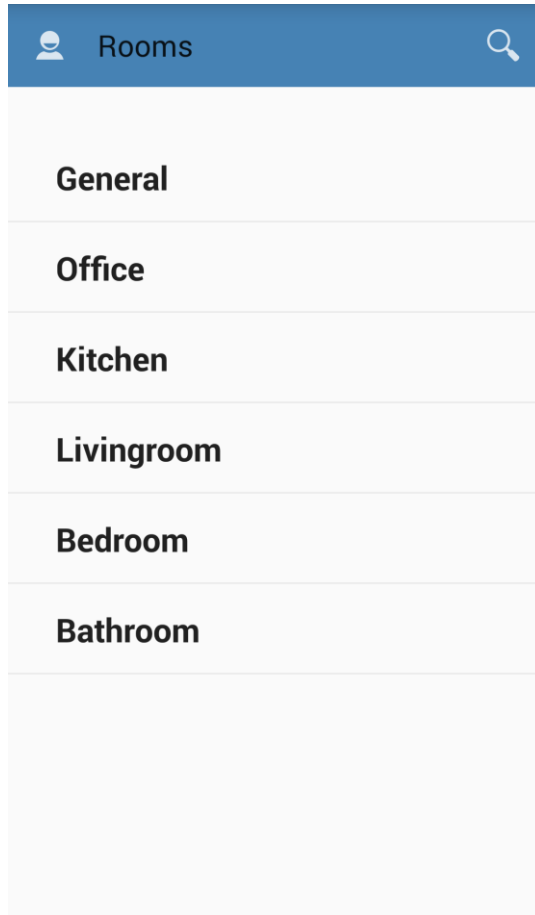
  - Description:
    Choose one room you desire to clean. Meanwhile, if you click the "user icon", you are allowed to enter User Center Activity to modify name and language; otherwise if you click search view, you are allowed to type anything concerns with the cleaning agent to facilitate to search for result.

- Item Selection
  - Screenshot:
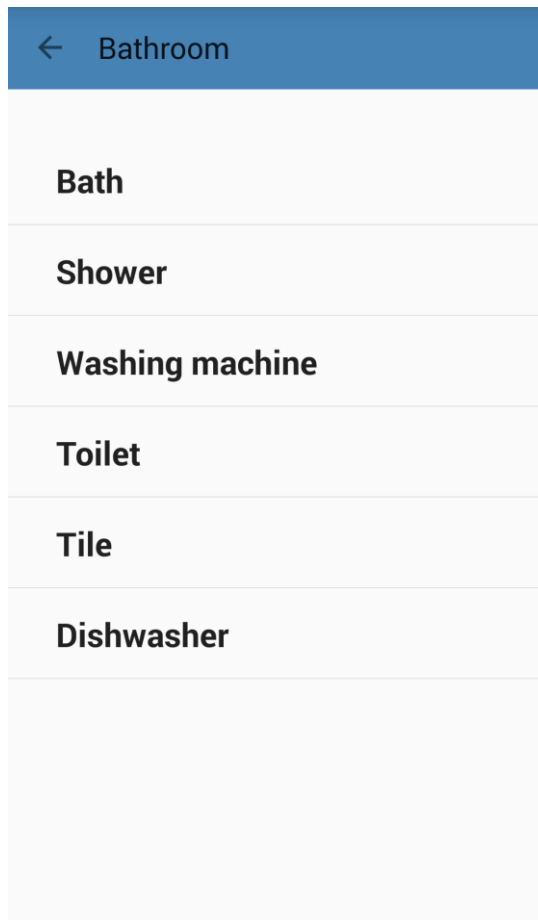
  - Description:
    Choose one item which belongs to the room you have already chosen.

- Cleaning Agent

    - Screenshot:
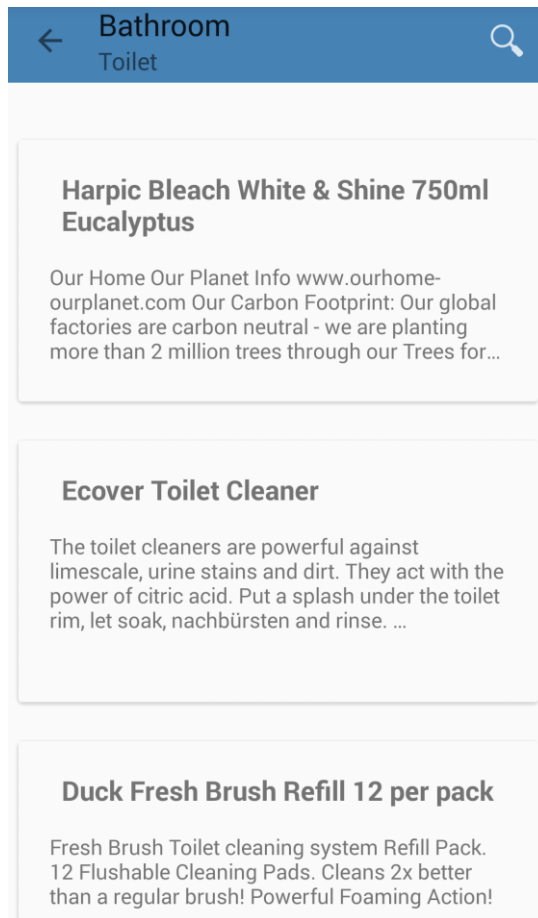
    - Description:
      Presenting the result either by direct typing searching or by tags related searching.

- Cleaning Agent Detail

  - Screenshot:
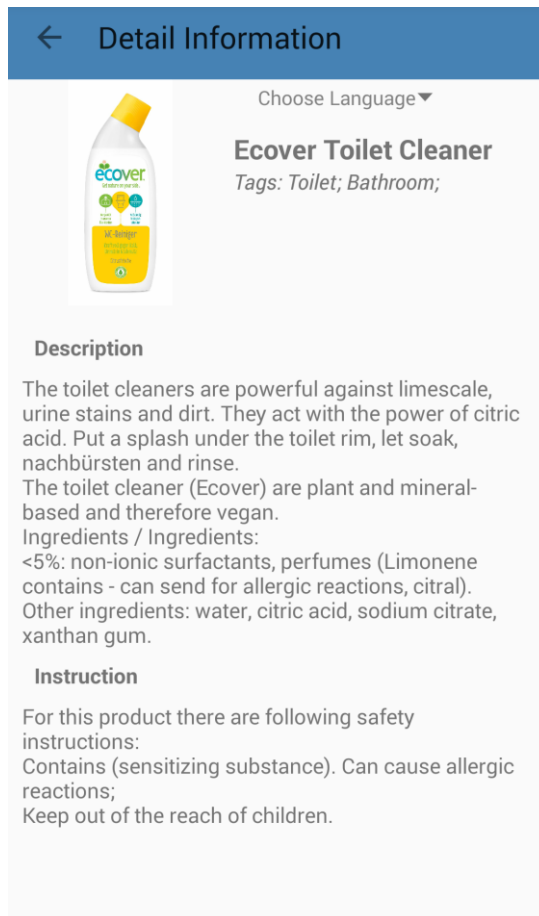
  - Description:
    Present detail information about cleaning agent. In addition, you are allowed to change content language in this interface, in order to a better comparison. Present detail information about cleaning agent. In addition, you are allowed to change content language in this interface, in 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.

4. Test

4.1. PC Usability Test

4.1.1. 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.

4.1.2. 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)
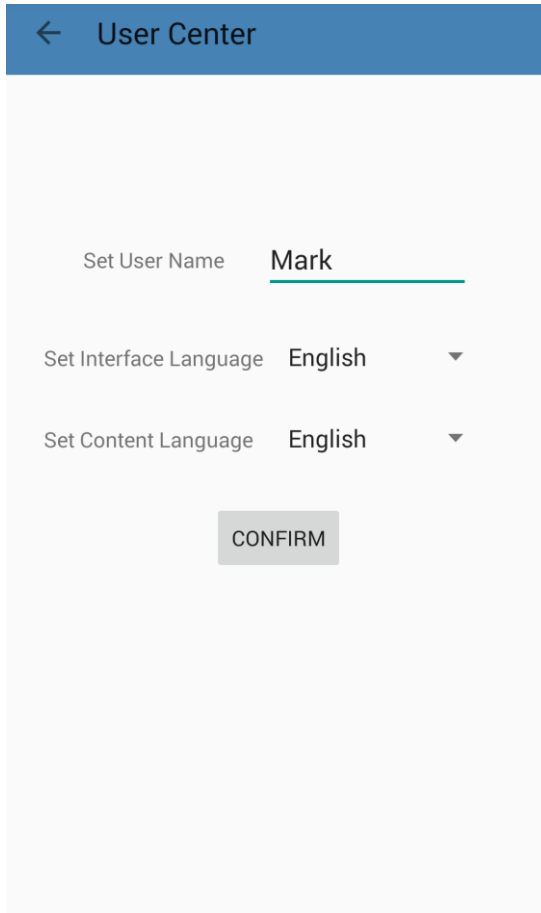
- Task 2:

  - 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)

- Record: (for each task)
  - time consumption
  - user activity, window entry
  - user complaint or feedback

4.1.3.   First round usability test

- Result: see appendix.

- Issues:
  - Bug: user cannot add the same tag again when he or she added it just now and delete it. The state of combo box which contains tags is not changing, so no activity is triggered.
  - Bug: in add CA window, tag lists are not refreshing after user add a new tag. User cannot find the tag which has just been added.
  - Chinese students tend to set their interface language to Chinese even though they can speak English and English is Enca's most supported language. Problems such as description mismatch occur during search.
  - When users set content language to Chinese, it is hard for them to search by German keywords. Because even though the result would always come up, the contents are all in Chinese and it contributes to the difficulty to identify the exact cleaning agent. We have already used parallel tabs for different languages, however users tend to ignore it.
  - Users tend to always use keyword search instead of tag search. Tags are somehow confusing the user. Though they can get the desired result, it is not the most optimal way.
  - Users may easily miss click on a tag and delete it when modifying a CA.
  - Users have no idea about what "memo" is.
  - Users have no idea of the existence of "user center" or what it is for.
  - Activities such as "Save" lack conformations.

- Improvement plan:
  - Fix tag adding bug by clearing combo box state after each deletion.
  - Fix tag refreshing bug.

- Main window visual separation. Highlight "tag" and "keywords" area with a non-transparent box. Clearly separate main window into three parts which are "search by tags", "search by keywords" and "result".
- Give conformation windows for multiple activities such as "save memo" and "clear memo".
- Change button text on some buttons. Such as close button in CA detail window is changed from "Cancel" to "Close".
- Double click to remove a tag instead of single click.
- Add "instruction" and prompt user to read it during first login.
- Make tab header more visible, change background color from white to grey.

4.1.4.   Second round usability test

- Result: see appendix.

- Issues:

    - Users cannot remove a picture in add CA window.
    - Users cannot easily find "Clear" button and reset search result.
    - "User center" is still confusing the user.
    - Users have trouble finding cleaning agents with short names.

- Improvement plan:

    - Add "remove picture" button on add CA window.
    - Move "clear" button to "search by tags" box and change name to "clear selection".
    - Add "tool box" with multiple setting entry.
    - Full word match gives more relevance.

4.2.   Android Usability Test

4.2.1.   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'.
    - Detail:
      User shall find a certain cleaning agent.
    - 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 by keyword'.
  - Detail:
    User shall be given a cleaning purpose and find a suitable CA.
  - Target CA:
    [Purpose: Clean toilet] (Opening any detail window of CAs which have tags "Bathroom" and "Toilet" is considered to be task accomplished)

4.2.2.   Usability test

- Result: see appendix.

- Issues:

  - Users are reluctant to use tag.
  - Users set language to Chinese and had trouble finding the German CA.

- Improvement plan:

  - Add "instruction".
  - Easier language switch.

4.3.   Equivalent classes and boundary tests

4.3.1.   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}

4.3.2.   Special test cases

- Keywords are used in regular expression patterns, thus special characters 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 characters such as ' shall be escaped. Test of saving a field which contains ' into the database shall be performed.
    - Input: [Tag name: "Mary's favourite"]

5. Evaluation

5.1. 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

5.2. 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.

Appendix I

Instruction:

- Search

    - By tags
      It is easier to search for a cleaning agent using tags when you know what to clean. For example, if you want to find cleaning agents that wash dishes: click on "Kitchen" in "Room" column, then click on "Dishes" in "Items" column. All cleaning agents with these two tags will be displayed in the result table. To start a new search, you need to click on "clear" button.
    - By keywords
      It is easier to search for a cleaning agent using keywords when you know its name, brand or descriptions. For example, if you want to search Ecover's product, type in "Ecover" in the text field and click on "Search" or press Enter.
    - Combined
      By the way, you can use both tags and keywords at the same time to find a cleaning agent in a more accurate manner. Now I bet you know how to find Ecover's dish cleaner.

- Add, modify and delete

    - Cleaning agents and tags
      You can add, modify and delete a cleaning agent or a tag. Try right clicking on a tag or a cleaning agent and also try out menu bar, you will find the entry.
    - System protection
      To ensure the integrity of system information, you are not allowed to do anything to a system predefined cleaning agents or tag, except taking your memo. However, you can bypass this protection by enabling God mode.

- Memo

    - Make a memo!
      You may make a memo on a cleaning agent to remind yourself of anything you like. You can write your memo on detail window and modify window. Click on "Clear" to clear the memo and click on "Save" to save it.
    - Find memos
      Open "User center" and turn to tab "Memos" or click on menu View -> Memos. All your memos are listed there.

- Preference setting

    - Language
      Interface language:
      it should be the language that you are comfortable to read. Your software interface will be in this language. Select "Chinese" if you speak Chinese best.
      Content language:
      you can set content language based on the origin of the cleaning agents. We will provide contents in this language to facilitate you matching the corresponding cleaning agent. If you are traveling in Germany, select "German". And don't worry, we always provide you contents in all three supported languages in separated tabs.

- Be the god!
  You can mess with system predefined cleaning agents and tags when you enable God mode. Check the check box to enable it. But remember, great power comes with great responsibility.

If this instruction cannot solve your problem or should you have further questions, do not hesitate to log on https://github.com/Nimita311/ENCA-Andriod and open an issue or email our developers.

Appendix II

Usability Test (User handout):

Thank you very much for your participation, your assistance will help us develop software that is easier to use.

Enca is a cleaning help software that helps you find the right cleaning agent in a foreign country. Our software supports three languages which are English, German and Chinese and we also prepared plenty of cleaning agents on Chinese and German market in our database for you to search. You may search for a certain cleaning agent with keywords when you know its name or description. You may also search for cleaning agents that suit your need with our predefined tags.

Now do you want to try out Enca? We have some simple tasks for you to fulfill. Please do the following instructions.

1. Please open Enca and log in. Read the manual if you would like to.

2. Your German friend recommends you "AJAX Frischeduft" but you cannot remember what kind of cleaning agent it is. Search for it in Enca and make memo on it as "to buy".

3. Your toilet cleaner is used up and you would like to buy some more. Use Enca to find out what options do you have and make a memo on one of them which you would like to buy as "to buy".

4. This cleaning agent online caught your eye and you would like to record it in Enca: 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. Add information of this cleaning agent in Enca and stick an additional "interested" tag in "others" category to it.

5. You bought the previously mentioned cleaning agent in Amazon and found it so rubbish that you would like to delete it from Enca. Delete the cleaning agent and the tag you have just created.

6. Now you would like to go to the supermarket and buy some cleaning agents. Remember that you have made some memos as "to buy"? Find those cleaning agent with memos.

That is the end of the test. Thank you again for your cooperation and have a nice day.

Appendix III

PC first round usability test result:

- Result 1

    - Time consumption:

Task 0	0:17
Task 1	1:14
Task 2	2:11
Task 3	2:49
Task 4	0:47
Task 5	0:20

    - User Activity:
        Language: CN, CN
        Search: keyword, keyword
        Add CA: add button
        Add tag: Cleaning agent modifier
        Remove tag: Menu bar -> Tag modifier

- Result 2

    - Time consumption:

Task 0	0:20
Task 1	0:56
Task 2	0:55
Task 3	2:49
Task 4	1:28
Task 5	0:33

    - User Activity:
        Language: CN, CN
        Search: keyword, keyword
        Add CA: add button
        Add tag: Cleaning agent modifier
        Remove tag: Menu bar -> Tag modifier

- Result 3

    - Time consumption:

Task 0	0:11
Task 1	0:25
Task 2	0:40
Task 3	4:30
Task 4	0:25
Task 5	0:30

    - User Activity:
        Language: CN, CN
        Search: keyword, keyword

       Add CA: add button
       Add tag: Cleaning agent modifier
       Remove tag: Menu bar -> Tag modifier

PC second round usability test result:

- Result 1

  - Time consumption:

Task 0	0:10
Task 1	1:11
Task 2	0:39
Task 3	2:04
Task 4	1:02
Task 5	0:24

  - User Activity:
    Language: EN, EN
    Search: keyword, tag
    Add CA: add button
    Add tag: Cleaning agent modifier
    Remove tag: Context menu -> Tag modifier

- Result 2

  - Time consumption:

Task 0	0:10
Task 1	1:11
Task 2	0:39
Task 3	2:04
Task 4	1:02
Task 5	0:24

  - User Activity:
    Language: CN, EN
    Search: keyword, tag
    Add CA: add button
    Add tag: Cleaning agent modifier
    Remove tag: Context menu -> Tag modifier

Android usability test result:

- Time consumption:

Task	Result 1	Result 2	Result 3	Result 4	Result 5
Task 0	0:20	0:35	1:20	0:40	0:30
Task 1	2:00	0:40	1:00	2:40	0:30
Task 2 (by tags)	0:10	0:10	0:08	0:08	0:15
Task 2 (by keyword)	0:15	0:25	0:15	0:10	0:25

- Time consumption: