Smart Home App

Biyun Huang, Han Lin, Jue Hou

Project report
UNIVERSITY OF HELSINKI
Department of Computer Science

Helsinki, May 16, 2017

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty Faculty of Science	Laitos — Institution — Department Department of Computer Science

Tekijä — Författare — Author

Biyun Huang, Han Lin, Jue Hou

Työn nimi — Arbetets titel — Title

Smart Home App

Oppiaine — Läroämne — Subject

Computer Science

Työn laji — Arbetets art — Level Project report	Aika — Datum — Month and year May 16, 2017	Sivumäärä — Sidoantal — Number of pages 11

Tiivistelmä — Referat — Abstract

People can face many troubles in their daily life. One common pain point often happens during cooking, which is that people often get their hands busy with processing foods or even fully covered with flour paste when they also have to interact with some smart devices for receipts. This is exactly what our project tries to tackle with: to help people interact with other appliances when they are cooking. We present a smart phone app implementing with an alternative interacting approach with voice. By using voice command, our implementation should be easily able to help people to literately "ask for" next instruction. To make our project more powerful, we embedded our app with WeMo smart switch so that people can use voice command to turn on or off electrical appliances not only in the kitchen but potentially all over the house. We choose to use Android as the platform to implement and Chrome embedded voice recognition. So, most of our project is implemented with JavaScript and Java. food2fork API is also used to get recipes.

Avainsanat — Nyckelord — Keywords

Smart Home, Cook, Recipe, Speech Recognition

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1 Introduction

With the continuous growth of population in using smart mobile devices, demands for applications on smart phones in people's daily life also increase. Smart home or home automation has became a popular and promising field. Applications in smart home can not only provide people a more comfortable life but also give disabled people additional help.

Nowadays, various apps exist for searching recipes for example allrecipes. People can find ingredients and instructions by using those apps but they have to input some keywords for searching. In a cooking scenario, people's hands can be busy or their hands can be wet or oily and at the same time, they need to find a recipe. Then it is not convenient for people to type some texts on the touch screen of a smart phone since they have to clean and dry their hands. In order to solve this kind of problems, we propose a smart home app. In this app, speech recognition is combined with searching recipes. Thus people can search recipes by talking with a smart phone to free people's hand when they are cooking. In addition, In order to give additional help to disabled people, a smart switch also embedded in this project for controlling lights remotely.

With the development of cloud based speech recognition, it is possible to apply speech recognition on Android platform. In addition, there are lots of API available for searching recipe. Therefore, in this project, Google Cloud Speech API is utilized and it is an Android app. The main tasks of this smart home app is: (1) searching for a recipe by speech recognition, (2) reading aloud the recipe, (3) stopping reading the recipe when saying "stop",(4) controlling lights remotely.

There are lots of related work. One of the earliest study about a speech enabled cooking assistant was in [5]. They made a survey of this kind of smart home systems, 80% of testers thought the system was useful but 50% of them thought it was hard to understand the the system. That study was conducted in 2001. One of reasons why so many testers had difficulties to understand the system is that the technology of speech recognition is not well developed at that time. In [1], a smart home system was designed. In their work, light switches, power plugs, current sensors, temperature sensors and other sensor were integrated and voice activation was designed for switching lights. A smart phone app was created for controlling electrical appliances in the system. In [2], a real time recipe recommendation system was designed. In their work, machine learning algorithms were utilized for object recognition. In their system, by pointing a camera on the smart phone to food, a list of recommended recipe would be shown in real time. Thus, they combine object recognition with searching recipe. In [3], a robust speech enabled cooking assistant was implemented. Their app is similar as ours. However, in their work, natural language processing was utilized so their app can also communicate with users. For example, the app can answer questions when people are cooking.

The rest of the report is arranged as: section 2 illustrates the need findings of our app. Section 3 illustrates the technique details about how to implement this app. Section 4 illuminates the evaluation of our project in aspects of recipe searching and speech recognition. Finally it is the conclusion.

2 Needfinding

The goal of the project is to develop a smart home app that benefits the daily life of people. And we narrow down the range into intelligent cook assistant. To find needs of people to the app, we first identified the stakeholders of our product. Then we conducted intercept interviews on 5 people in the campus.

Stakeholders Surrounding this topic, we discussed and identified the people we should engage throughout our project due to their stake in the outcome. Figure 1 illustrates the stakeholders. From the users' point of view, there are families or solitary, neighbors, special group that needs to be taken care of, such as babies, elders and disabled people, and cooking community including newbies and masters. Disabled people are subdivided as low vision, lame and deafness. From the view of the service providers, advertising sponsors also need to be considered. The cook himself or herself and the cooking community should be the most related to the app. The special group such as babies, elders and disabled people also take an important part. Relatively, neighbors and sponsors are not that significant.

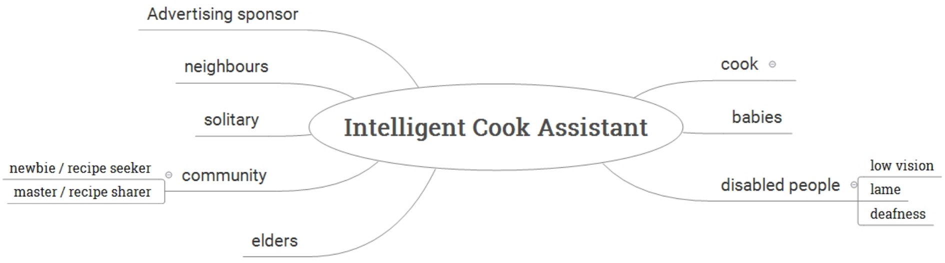


Figure 1: Stakeholders map

Interview We listed several questions through brainstorm and picked 3 questions as a questionnaire. And then we conducted intercept interviews to five people. After that, we summarized the interviews and extracted the needs as the objectives to be meet in our app.

Interview Questions After discussion, we made a questionnaire that can help to identify the needs of people when they cook.

1. What are the activities you do before you cook and while you are cooking?

2. Do you use any intelligent assistant (i.e. auxiliary tools)? If yes, what do you value about it and why? If no, what do you think it may help?

3. What problems have you come across while you are cooking?

Interviews Conduction and Summarization We conducted the interviews in an intercept manner to five people with each no more than 5 minutes in the campus. The interviewees include people who live with families and solitary who are students living in a student apartment here.

We summarized the response to the questions.

To the first question, the newbie cooks would search for recipes first. They may search possible recipes with the ingredients they have as keywords or they may be interested in some dishes, and search for the specific recipes before preparing the ingredients, while the skillful cooks have their own or marked recipes. That is, before cooking, people need recipes and gradients. When people are cooking, people are not that familiar with the recipes need to follow the instructions line by line, while they need to due with the materials, such as cleaning the food or tools, controlling the electrical appliances in the kitchen and cutting the meat or vegetables. Also they need to care about the time elapsed in each appliances, such as microwave oven and toaster. Meanwhile, they may also care about their babies in the baby room.

To the second question, as to the auxiliary tools people use when cooking, all of them use or have used Internet to find recipes, because it's convenient and they need the directions of the recipes before they cook. The recipes are from books, magazines, computers or mobiles. Some of them proposed that if there is a tool to help them read the recipes or control the devices, it would benefit.

To the last question, when people read the recipes, they may need to print the recipes, look up the instructions every step or even turn the page, while their hands are busy with all kinds of stuff dealing with the ingredients. It's troubling especially when their hands are wet. In the meantime, they may be disturbed buy something happening in next door. For example, they may need to open the door for a guest. It may be done by remote control. Furthermore, they may need to schedule turning on or off a device without a timer.

3 System Design and Implementation

In this section, technique details about the implementation of the system are illuminated. The system consists of four modules: search recipes, speech recognition, user interface and a smart switch. The flow chart of the speech enabled recipe search system is shown in Figure 2

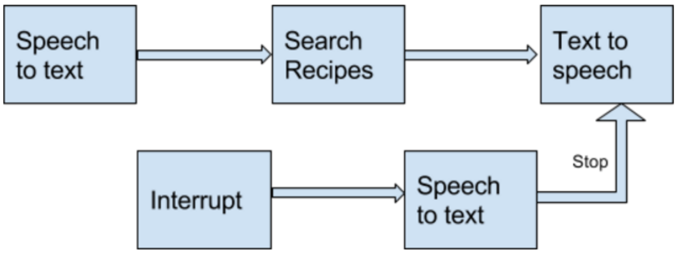


Figure 2: Flow chart

3.1 Search Recipes

There are all kinds of API for searching recipes. Food2fork API was chosen because it is light weighted and it searches recipes from 200000 recipes. This API belongs to REST API and it supports all kinds of languages. In this project, java was utilized for searching recipes. In general, two kinds of requests can be made. They are "search" and "get all recipes". In this project, requests of search was made to this API since it searches recipes based on some keywords like "chicken".

For each request, this API gives maximum 30 recipes. If the next 30 response recipes needs to be obtained, another request has to be made with the parameter of "page" and the value of page is 2. In other words, pages of response can be defined in the search request. In addition, this API also provides two kinds of sorting for searching queries. They are sorting by rate and sorting by trend. Sorting by rate means the response is given as the highest rate recipe shown on social network is the first one, namely all the response recipes are given in the decreasing order based on the value of the rate given by people on social networks. Sorting by trend means all the response recipes are given in the decreasing order based on the number of viewers on a recipe. In this project, sorting by rate is utilized since the higher a rate is, the more reliable a recipe is. Response of recipes is encoded in the format of JSON. Following is an example of my code about how to achieve a searching request to the API:

```
private static String run(String url) \
throws IOException, JSONException {
     StringBuilder result = new StringBuilder();
     URL uurl = new URL(url);
     HttpURLConnection conn = \
     (HttpURLConnection) uurl.openConnection();
     conn.setRequestMethod("GET");
     BufferedReader rd = new BufferedReader(new \
     InputStreamReader(conn.getInputStream()));
     String line;
     while ((line = rd.readLine()) != null) {
        result.append(line);
     }
     rd.close();
     return result.toString();
}
public static String search(String query)\
throws IOException, JSONException {
     final String url = \
     API_URL_BASE + "/search?key=" + API_KEY + \
     "&q=" + URLEncoder.encode(query, "UTF–8");
     return run(url);
     }
```

The response contains lots of information. They are original URLs where the response recipes locate, image URLs, titles of dishes, ingredients, social rates, publishers of recipes, descriptions of dishes and instructions. Some informations of the response are discarded like those URLs, social rates and publishers. Finally the response recipe are stored as objects and send the response as input as text to the speech module.

3.2 Speech Recognition

In this section, we will discuss the module of speech recognition. As one essential part of our implementation, speech recognition module is mainly responsible for accepting voice command and execute command accordingly. We will next introduce two important parts of this module: our implementation on command list and speech recognition library.

Command List The motivation for the command list is to simplify the

Table 1: Voice commands list

Voice Commands	Description
Start	Start to read the instruction of the selected recipe.
Stop	Stop playing any sound and deactivate voice recognition.
Again	Read the current instruction of the selected recipe again.
Next	Read the next instruction of the selected recipe.
Previous	Read the previous instruction of the selected recipe.
(Number)	Read the instruction that user specifically choose.

commanding system so that both user and we can benefit from the list. Users can easily follow the illustration to activate the right command. For us, we are free to implement some advanced machine learning such as Naive Bayes Classifier to identify all the instruction from users.Commands currently has already been implemented are listed as Table 1

Speech Recognition Library Our speech recognition is mainly based on Google's platform. The reason for this choice is that Google's product on speech recognition is of both high usability and accuracy. Our main goal is reached with the SpeechRecognizer which is embedded in the Android system and partially supported Google Search app.

```
Intent intent = new Intent(
        RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
// Getting an instance of PackageManager
PackageManager pm = getPackageManager();
// Querying Package Manager
List<ResolveInfo> activities =
        pm.queryIntentActivities(intent, 0);
if(activities.size()<=0){
        //Fail to find the activity
        return;
}
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
intent.putExtra(
        RecognizerIntent.EXTRA_PROMPT, "Title");
startActivityForResult(intent, SPEECHTOTEXT);
```

As what is showed above, we simply need to call an "intent" to activate speech recognizer, which means its functions are very easy to implement. Also, detailed documentation and other related implementation examples are easy to obtain from the Internet.

In addition, there two alternative solutions for speech recognition that we initially studied. One is the speech recognition API from native Google Cloud Platform. The documentation shows that these APIs are easy to call, but it requires for registration and authentication on app and developers, which we may not have much time to test on such trivial matter. Moreover, Google has set limitations on its free-of-charging usage. Free embedded APIs are more suitable for our project. Another choice is the embedded APIs in the chrome-based browser. The example and documentation show it as a practical solution and can be purely called with JavaScript code, however such APIs are developed as a beta test and is vendor prefixed. After further study, we find it very likely to have requirements on the version of WebView component or sometimes not even work on Android platform. This may be problematic when we start to integrate. The native supporting APIs for Android platform are obviously a better choice.

3.3 User Interface

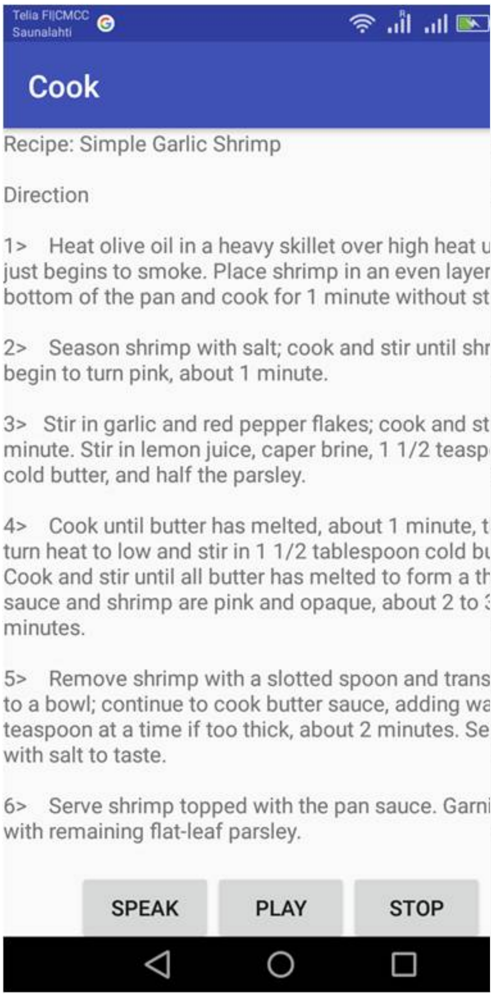
We chose Android to implement our UI, because Android is one of the most popular mobile operating systems. Also the speech recognition API and Android are both provided by Google, so that they can be integrated seamlessly. Moreover, Android is open source, which benefits the developers to an extent.

Figure 3 shows the icon of the application.

Figure 3: Icon

Figure 4 shows the main layout of the application. The layout is a scroll view consisting of two parts. The center is a text view showing the direction of a recipe. There are three buttons on the bottom. When users click the Play button, the application will read the instructions line by line. When users click the Stop button, the reading will be stopped. When users click the Speak button, the Google speech recognition UI will be shown. Users can speak to the phone and the voice input will be recognized as commands. For example, if users say "Play", then the play action will be triggered. Similarly, the stop action can be triggered by telling the application "Stop".

Users need to click the "Speak" button once to trigger the voice input. But after that, the listener will keep on working until users exit.



Recipe: Simple Garlic Shrimp

Direction

1> Heat olive oil in a heavy skillet over high heat u
just begins to smoke. Place shrimp in an even layer
bottom of the pan and cook for 1 minute without st

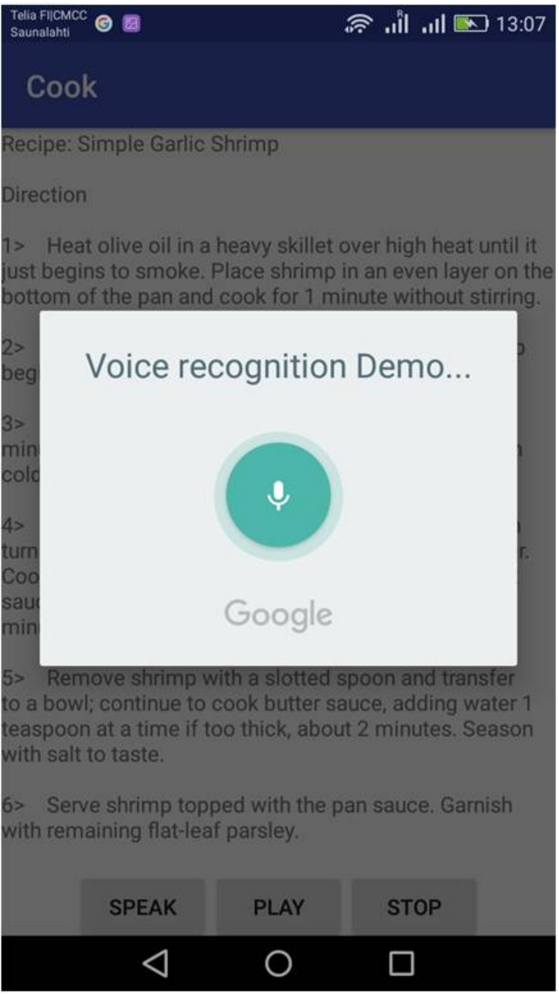
2> Season shrimp with salt; cook and stir until shr
begin to turn pink, about 1 minute.

3> Stir in garlic and red pepper flakes; cook and st
minute. Stir in lemon juice, caper brine, 1 1/2 teasp
cold butter, and half the parsley.

4> Cook until butter has melted, about 1 minute, t
turn heat to low and stir in 1 1/2 tablespoon cold bu
Cook and stir until all butter has melted to form a th
sauce and shrimp are pink and opaque, about 2 to 3
minutes.

5> Remove shrimp with a slotted spoon and trans
to a bowl; continue to cook butter sauce, adding wa
teaspoon at a time if too thick, about 2 minutes. Se
with salt to taste.

6> Serve shrimp topped with the pan sauce. Garni
with remaining flat-leaf parsley.



Recipe: Simple Garlic Shrimp

Direction

1> Heat olive oil in a heavy skillet over high heat until it just begins to smoke. Place shrimp in an even layer on the bottom of the pan and cook for 1 minute without stirring.

Voice recognition Demo...

5> Remove shrimp with a slotted spoon and transfer to a bowl; continue to cook butter sauce, adding water 1 teaspoon at a time if too thick, about 2 minutes. Season with salt to taste.

6> Serve shrimp topped with the pan sauce. Garnish with remaining flat-leaf parsley.

Figure 4: Main layout

Figure 5: Google speech interaction

3.4 Smart Switch

To support appliance controlling and monitoring, we applied smart switch in our project. We use WeMo insight switch as the firmware. It enables any appliance to be controlled through Wifi. Furthermore, with the functionality expansion provided by IFTTT, users or developers like us are allowed to design customized rule (recipe) for the switch.

The recipe for the smart switch basically follows the style of trigger-action. For example, if one's GPS coordinate of his cell phone shows that he is away from some place, then turn on the switch or other around. It is a simple programming style. And Study has shown that most smart home requests can be programmed as trigger-action [4]. Although we can now merely control its on or off, more action can be expected in the future as more smart appliances can get involved in the future. As for trigger part, it is already very well-developed in the IFTTT. Thousands of triggers are supported to meet different requests. More detailed introduction and discussion can be found in both [4] and the website of ITFFF.

For our project, we mainly implement two controlling links based on the network to control the switches or appliances, which is also one choice of the condition supported by IFTTT. Our implementation will only need to GET or POST to these links so that the switches or appliances can be turned on or off. Some other condition such as time control or geo-position control may also be useful in the context of our project.

4 Evaluation

4.1 Search Recipes

In order to evaluate the effectiveness of the function of searching recipes, three people are invited for using it. The evaluation criteria is whether the function of searching recipes is really helpful. The results is if a user searches recipes in using some simple words like "fried chicken", the response is given correctly. But if a user want to find instructions of Chinese dishes, no response was given. During the process of evaluation, searching criteria was changed to searching based on sorting of trend and two people thought the response recipes were not good as the response of searching based on sorting of rate. In addition, all of them thought searching recipes is helpful to their daily life although they thought this function is kind of simple since they want to obtain more informations about recipes such as pictures and videos. Therefore, images and videos can also be shown in app for our future work.

4.2 Speech Recognition

The goal of evaluation for speech recognition to test the accuracy of the recognition and other related factors. The evaluation is conducted in a way that ten people were invited to install our app and give voice commands

according to the command list. And then answer one questionnaire and give a short comment on their experience.
The rating of the accuracy and response shows that cloud-based speech recognition as Google is a good choice for its high response and accuracy. However, there are also some practical problems. One user complained that he will have to shout if he was not close to the cell phone so that he could get some response. This is indeed a problem since users will not always holding the cell phone or wearing headphone during cooking constantly. We haven't come up a solution for such a problem yet. But, this could be one of our future work. Another user complained about unable of interrupting instruction reading. Even though we designed some pulses between each sentence, commanding during the reading is still not supported. Solution can be a more fine-grained pulse setting or to implement some noise removing function.

5 Conclusion

In this report, we propose a speech enabled cook assistant app on Android platform. People can search recipes by talking with a smart phone using some keywords. Also, a smart switch is embedded for controlling lights. In addition, the technique details about the project are illustrated in the report. Google cloud speech API was utilized for speech recognition, food2fork API was used for searching recipes and the project was completed in java. For future work, natural language processing can be embedded in our project so when people search recipes, a smart phone can answer questions. In addition, people can search recipes by saying sentences instead of keywords to make the app smarter.

References

[1] Kumar, Shiu: *Ubiquitous smart home system using android application.* arXiv preprint arXiv:1402.2114, 2014.

[2] Maruyama, Takuma, Kawano, Yoshiyuki, and Yanai, Keiji: *Real-time mobile recipe recommendation system using food ingredient recognition.* In *Proceedings of the 2nd ACM international workshop on Interactive multimedia on mobile and portable devices*, pages 27–34. ACM, 2012.

[3] Schäfer, Ulrich, Arnold, Frederik, Ostermann, Simon, and Reifers, Saskia: *Ingredients and recipe for a robust mobile speech-enabled cooking assistant for german.* In *Annual Conference on Artificial Intelligence*, pages 212–223. Springer, 2013.

[4] Ur, Blase, McManus, Elyse, Pak Yong Ho, Melwyn, and Littman, Michael L: *Practical trigger-action programming in the smart home.* In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 803–812. ACM, 2014.

[5] Wasinger, Rainer M: *Dialog based user interfaces featuring a home cooking assistant.* University of Sydney, Australia, 2001.