FYT Final Project Report

Multi-stage Human-computer Interaction for Command Refining on an Intelligent Personal Assistant

by

Ivan Gondoprastowo

RO3

Advised by

Dr. David Rossiter

Submitted in partial fulfillment

of the requirement for COMP 4982H

in the

Department of Computer Science

The Hong Kong University of Science and Technology

2013-2014

Date of Submission: April 21st, 2014

Table of Contents

this page is intentionally left blank

1. Introduction

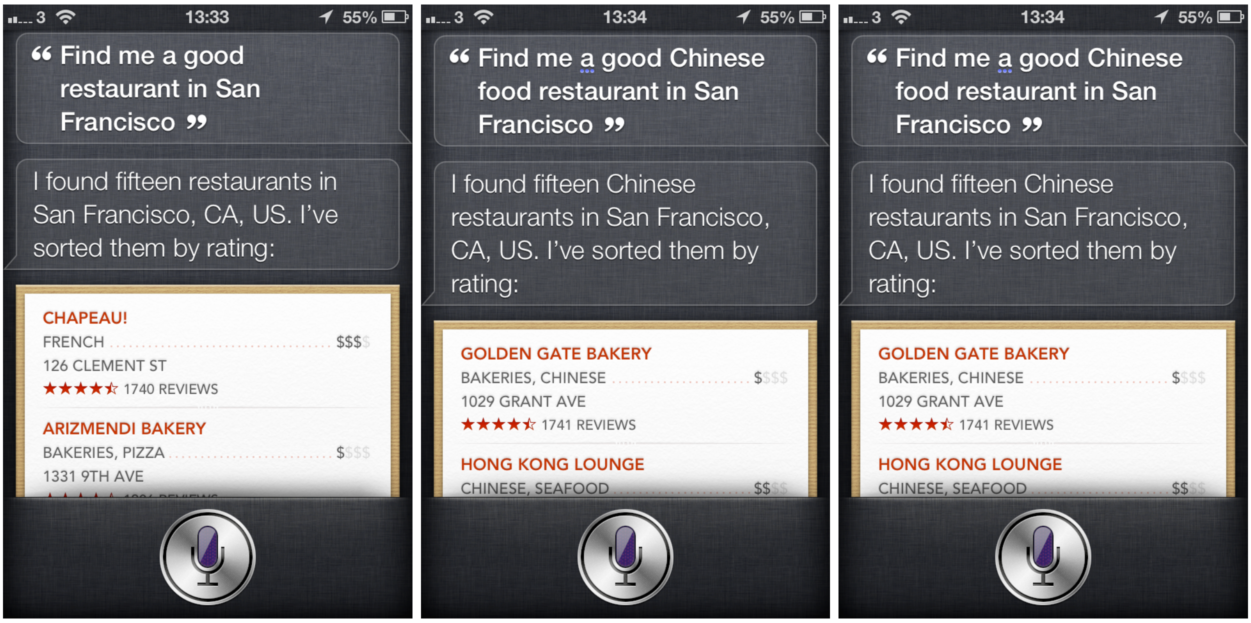
1.1 Overview

This project addressed the current implementation of digital voice assistants, such as Apple Siri and Google Now.

According to Wikipedia [1], a digital voice assistant is often described as an intelligent personal assistant and knowledge navigator. Implementation of this kind of assistant uses a natural language user interface to answer questions, give recommendations or perform actions by delegating requests to a set of Web Services. Some implementations of digital voice assistants also adapt to users' individual preferences over time and personalize the results, hence these assistants are often referred to as intelligent personal assistants.

Although there are a lot of implementations of this concept, two of the well-known ones are Apple Siri and Google Now. According to Wikipedia [2],[3], Apple Siri was first introduced on October 4th, 2011, and brought to the public with the release of iPhone 4S. Siri itself was a spin-off from the SRI International Artificial Intelligence Center and a DARPA-funded project. Google Now was first introduced on July 9, 2012, and it was first included in the Android operating system, version 4.1.

These digital voice assistants combine speech recognition, machine learning and natural language processing techniques to try to understand verbal commands given by the users and then perform specific tasks according to those commands. This results in specific system behavior when a verbal command is received; that is, the system will try to automatically understand the context and the semantics, predict what the user wants it to perform and then perform that action based on the single verbal command.



“Find me a good restaurant in San Francisco”

I found fifteen restaurants in San Francisco, CA, US. I’ve sorted them by rating:

CHAPEAU!
FRENCH .... $$$$
126 CLEMENT ST
★★★★☆ 1740 REVIEWS

ARIZMENDI BAKERY
BAKERIES, PIZZA .... $$$$
1331 9TH AVE

“Find me a good Chinese food restaurant in San Francisco”

I found fifteen Chinese restaurants in San Francisco, CA, US. I’ve sorted them by rating:

GOLDEN GATE BAKERY
BAKERIES, CHINESE .... $$$$
1029 GRANT AVE
★★★★☆ 1741 REVIEWS

HONG KONG LOUNGE
CHINESE, SEAFOOD .... $$$$

“Find me a good Chinese food restaurant in San Francisco”

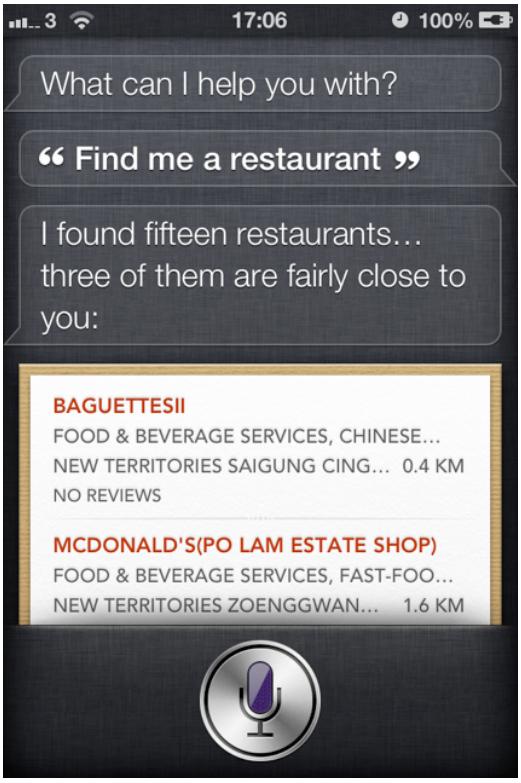
I found fifteen Chinese restaurants in San Francisco, CA, US. I’ve sorted them by rating:

GOLDEN GATE BAKERY
BAKERIES, CHINESE .... $$$$
1029 GRANT AVE
★★★★☆ 1741 REVIEWS

HONG KONG LOUNGE
CHINESE, SEAFOOD .... $$$$

Figure 1. Intelligent Personal Voice Assistant Implementation: Siri

While the process seems natural, the best current implementations often perform poorly and return unpredictable or unwanted results or actions. A study conducted by Piper Jaffray in 2012 [4], a Minneapolis-based investment bank, graded the speech technology for resolving requests with the letter “D” for accuracy. This is far from ideal, as the voice based input method in mobile operating system adoption is growing. Furthermore, Starner [5] predicts huge potential for voice based input method adoption for the next generation of wearable computers, such as Google Glass and smart watches.



What can I help you with?

“Find me a restaurant”

I found fifteen restaurants… three of them are fairly close to you:

BAGUETTESII
FOOD & BEVERAGE SERVICES, CHINESE…
NEW TERRITORIES SAIGUNG CING… 0.4 KM
NO REVIEWS

MCDONALD'S(PO LAM ESTATE SHOP)
FOOD & BEVERAGE SERVICES, FAST-FOO…
NEW TERRITORIES ZOENGGWAN… 1.6 KM

Figure 2. Finding a restaurant with Siri: a general command

Figure 1 shows Siri using its sophisticated ability to try to give some restaurant recommendations. The problem with this approach is that Siri does not really know exactly what the user wants to eat. (See Figure 2.) Siri does not know what type of cuisine the user is interested in, how far the user would be willing to travel or what the acceptable price range is. Although Siri may be able to accurately understand the user’s verbal command to find him/her a restaurant to eat at, further clarification from the user would yield much better suggestions and hence enhance the user experience.

The thesis of this project is that multi-stage human-computer interaction to refine verbal commands requires little user effort yet can greatly improve the quality and utility of the results. It can also avoid the necessity of users having to refine their commands by restarting the whole process, which is an intuitive reaction to poor search results. Furthermore, having the system ask for more details creates more natural human-computer interaction. Ultimately, higher quality results and more natural interaction will create a better user experience.

This thesis project is highly relevant to the design and development of applications and services for mobile devices and wearable computers, such as Google Glass. The project compared the user experience with the two approaches, single-stage verbal commands and multi-stage interaction for assistance. We measured the user experience by having volunteer testers try both methods. The user experience subject study included several measurements for user experience and usability, such as task completion rate, user experience, satisfaction of results and perceived system performance.

1.2 Objectives

The goals of this project were to study the effects of multi-stage human-computer interaction on user experience when using digital voice assistants and to compare this user experience with that of the current single-stage verbal command based approach.

In order to perform the study, we first built a small scale digital voice assistant that performs multi-stage interaction with users. This voice assistant implementation is also equipped with a single-stage verbal command system implementation. The implementation of the digital voice assistant handles a specific task: finding a suitable restaurant. Then, we conducted a formal user experience study to examine the effects of the proposed idea.

Our project mainly focused on the following objectives:

1. Develop a smart voice assistant system for the user experience study, utilizing the A/B Testing approach [6], such that:
    - A performs single-stage human computer interaction without command refining support to find a restaurant (similar to the way Siri and Google Now work)
    - B performs multi-stage human computer interaction for command refining support to find a restaurant
    - The system randomly switches back and forth
    - Both a web server and clients are implemented
    - The web server interfaces with Yelp! for finding restaurants in the United States of America.
2. Develop a plan to study the user experience based on A/B Testing and some specific user experience measurements.
3. Conduct relevant smaller studies to determine and support the system design decisions.
4. Conduct the user experience study on a voice assistant implemented with the proposed approach, focusing on generally accepted user experience measurements, e.g., task completion time, system response time, quality of results and overall user satisfaction.
5. Analyze the results of the study and derive a conclusion about the proposed approach.

To achieve the first goal, we utilized existing technologies and built necessary modules and technologies on the top of them. For the client side, we utilized HTML5 and related speech recognition techniques [7] to capture the speech and send it to the server. On the server side, we created a REST Web Service [8] that receives requests sent from the client, refines them and analyzes them. For the implementation of the A testing, the system analyzes the content of the input and identifies any missing information, which it then requests from the users. This is implemented as voice output, leveraging some existing text-to-speech technologies [9]. Based on users' responses, the system fills in the information until the query is complete. The implementations of both A and B testing send requests to the responsible web service to retrieve the results. Then the system displays the results on the screen, accompanied by some voice comments from the digital assistant.

To achieve the second goal, we reviewed existing literature on human-computer interaction using voice commands and planned a strategy that suited the narrow study we were conducting. This study uses the A/B testing method, and it measures user experience based on commonly accepted evaluation criteria, such as task completion rate, user experience, satisfaction of result and perceived system performance [10].

To achieve the third goal, we identified the heuristics that we were applying and decided to conduct smaller studies to analyze these heuristics.

To achieve the fourth goal, we recruited 27 volunteer testers to conduct the user experience study based on the devised strategy. Then, we recorded the data and feedback for further analysis.

To achieve the fifth goal, we analyzed the results of the study and looked for patterns and abnormal behavior, trying to understand the users' logic and considering the entire study in relation to a set of common applications of Siri and Google Now.

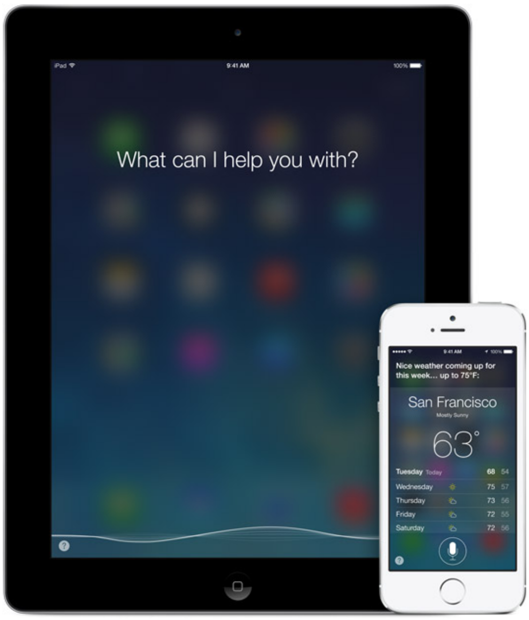
The biggest challenge we faced was implementing the voice assistant system for experimentation. Although we limited the application scope, we tried to make the system technologically advanced enough to represent current popular implementations of voice personal assistants. It was also a challenge to interface with data providers such as OpenRice. Another problem was dealing with certain local words, since current speech recognizers like Siri and Google Now usually don't support Cantonese words such as "Mong Kok" or "Dim Sum". In addition, Agushinta et al. [11] note that intelligent personal assistants such as Siri often have difficulties understanding English with accents other than Midwestern American English. Furthermore, designing and conducting a study that thoroughly captured and compared the user experience of both approaches (i.e. single-stage commands and multi-stage interaction) was a hefty challenge. For these reasons, we decided to limit the scope of our voice assistant implementation to finding places to eat at in the United States of America by interfacing to Yelp! as our information provider.

1.3 Literature Survey

We found the following systems and academic literature related to our project.

1.3.1 Siri

According to the Apple company website [12], Siri is an intelligent personal assistant that helps people get things done just by asking. It allows people to use their voice to send messages, schedule meetings, place phone calls and more. But Siri is not like traditional voice recognition software that requires memorization of keywords and specific voice commands. Siri understands natural speech, and it asks questions if it needs more information to complete a task. Siri is available for iPhone 5, iPhone 4S, iPad with Retina display, iPad mini and iPod touch (5th generation). Siri provided a good model for our system implementation.



What can I help you with?

Nice weather coming up for this week... up to 75°F:

San Francisco

Mostly Sunny

63°

Tuesday Today		68	54
Wednesday	☀	75	57
Thursday	⛅	73	56
Friday	⛅	72	55
Saturday	⛅	72	56

Figure 3. Siri user interface on iOS 7.

1.3.2 Google Now

According to Wikipedia [3], Google Now [13] is an intelligent personal assistant developed by Google that is available within the Google Search mobile application for the Android and iOS operating systems. Google Now uses a natural language user interface to answer questions, make recommendations and perform actions by delegating requests to a set of web services. Along with answering user-initiated queries, Google Now passively delivers information to the user that it predicts they will want, based on their search habits. It was first included in Android 4.1 (“Jelly Bean”), which was launched on July 9, 2012, and it was first supported on the Galaxy Nexus smartphone. Google Now provides an alternative example of a popular voice assistant system implementation, and it may become just as popular as Siri.

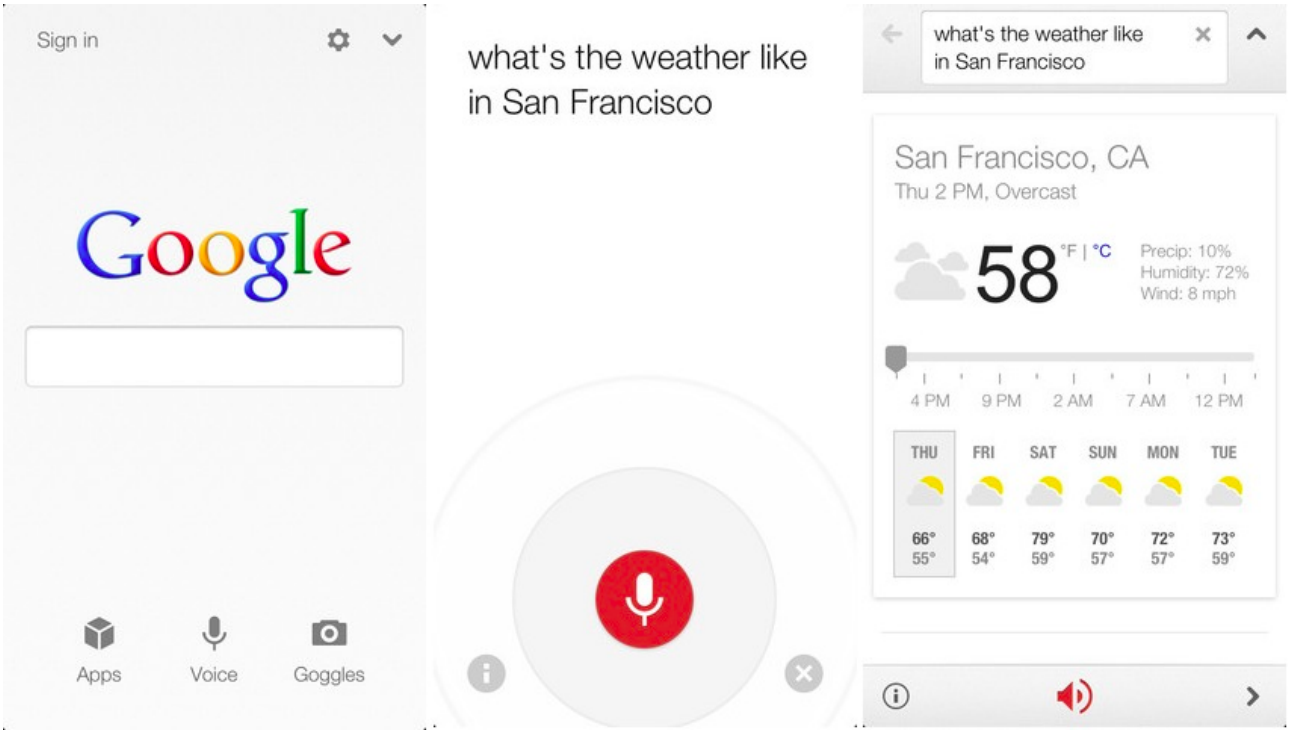


Figure 4. Google Now

1.3.3 OpenRice

OpenRice [14] is a food and restaurant guide website which is very popular among Hong Kong people. It stores detailed information about restaurants in Hong Kong and allows users to perform keyword searches and leave comments and ratings. Due to the change in design decision mentioned at the end of section 1.2, we chose not to link to this service.

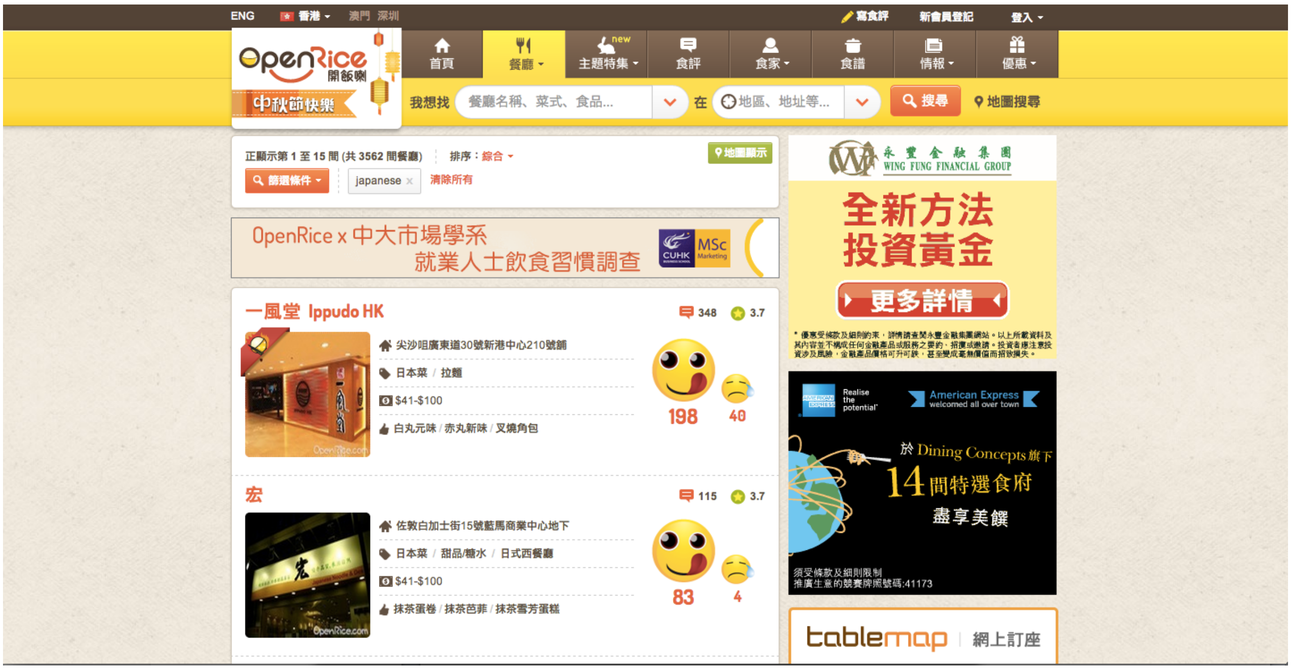


Figure 5. OpenRice Search Page, Desktop version

1.3.4 Yelp!

According to Wikipedia [15], Yelp! [16] initially operated a local search website. The website was relaunched in 2005 with a focus on user reviews for local businesses. It now includes restaurants and some other businesses. The service is widely used by US citizens to find reviews and get recommendations as well as information about places to eat. The Yelp! service provides developers with a publicly available API. Due to reasons mentioned at the end of section 1.2, we decided to choose Yelp! as the primary restaurant information provider for our system.

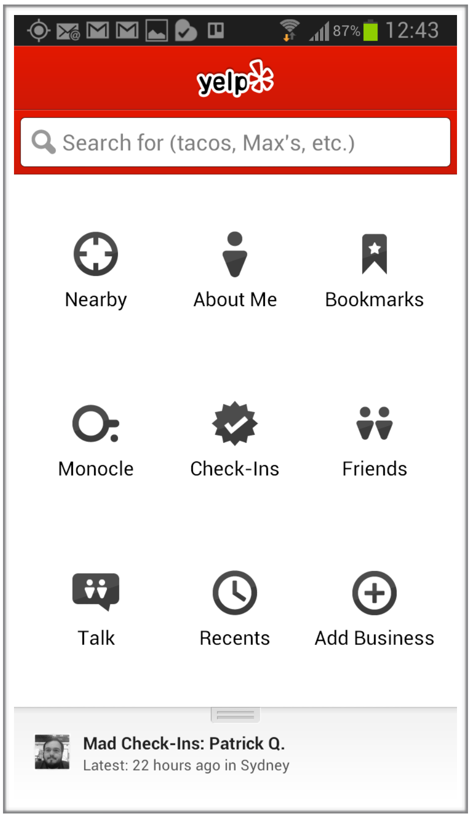

87%
12:43
yelp
Search for (tacos, Max's, etc.)
Nearby
About Me
Bookmarks
Monocle
Check-Ins
Friends
Talk
Recents
Add Business
Mad Check-Ins: Patrick Q.
Latest: 22 hours ago in Sydney

Figure 6. Yelp! Mobile User Interface

1.3.5. Webkit

According to Wikipedia [17], WebKit [18] is a rendering engine software component created to allow web browsers to render web pages. It is being used by Google Chrome and Apple Safari web browser applications. WebKit's WebCore and JavaScriptCore components are available under the GNU Lesser General Public License, and the rest of WebKit requires a license. Currently, WebKit is a trademark of Apple, registered with the U.S. Patent and Trademark Office. We used a WebKit based web-browser, Google Chrome as the target of deployment of our application.

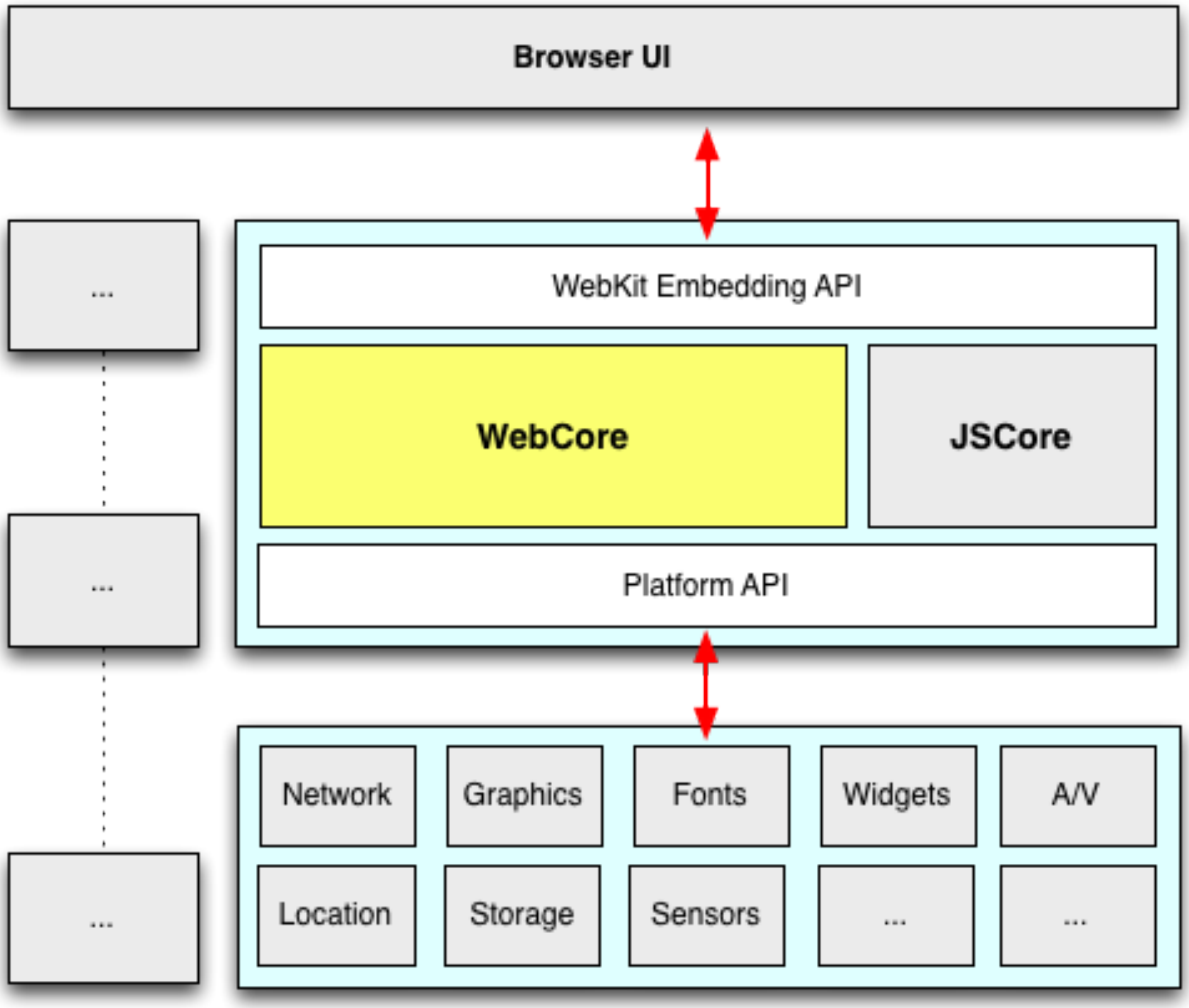


Figure 7. WebKit Based Browser System Diagram [19]

1.3.6 HTML Web Speech API

According to its specifications [9], the Web Speech API aims to enable web developers to provide speech input and text-to-speech output features that are typically not available when using standard speech recognition or screen reader software. These features are made available through a web browser. The API itself is not dependant upon the underlying speech recognition and synthesis implementation and can support both server- based and client-based/embedded recognition and synthesis. The API is designed to enable both brief (one shot) speech input and continuous speech input. Speech recognition results are provided to the web page as a list of hypotheses, along with other relevant information for each hypothesis. Our system uses API to both recognize speech input and perform text-to-speech output. The Web Speech API is only supported by Google Chrome 25 or above [20], which includes all versions since March 2013.

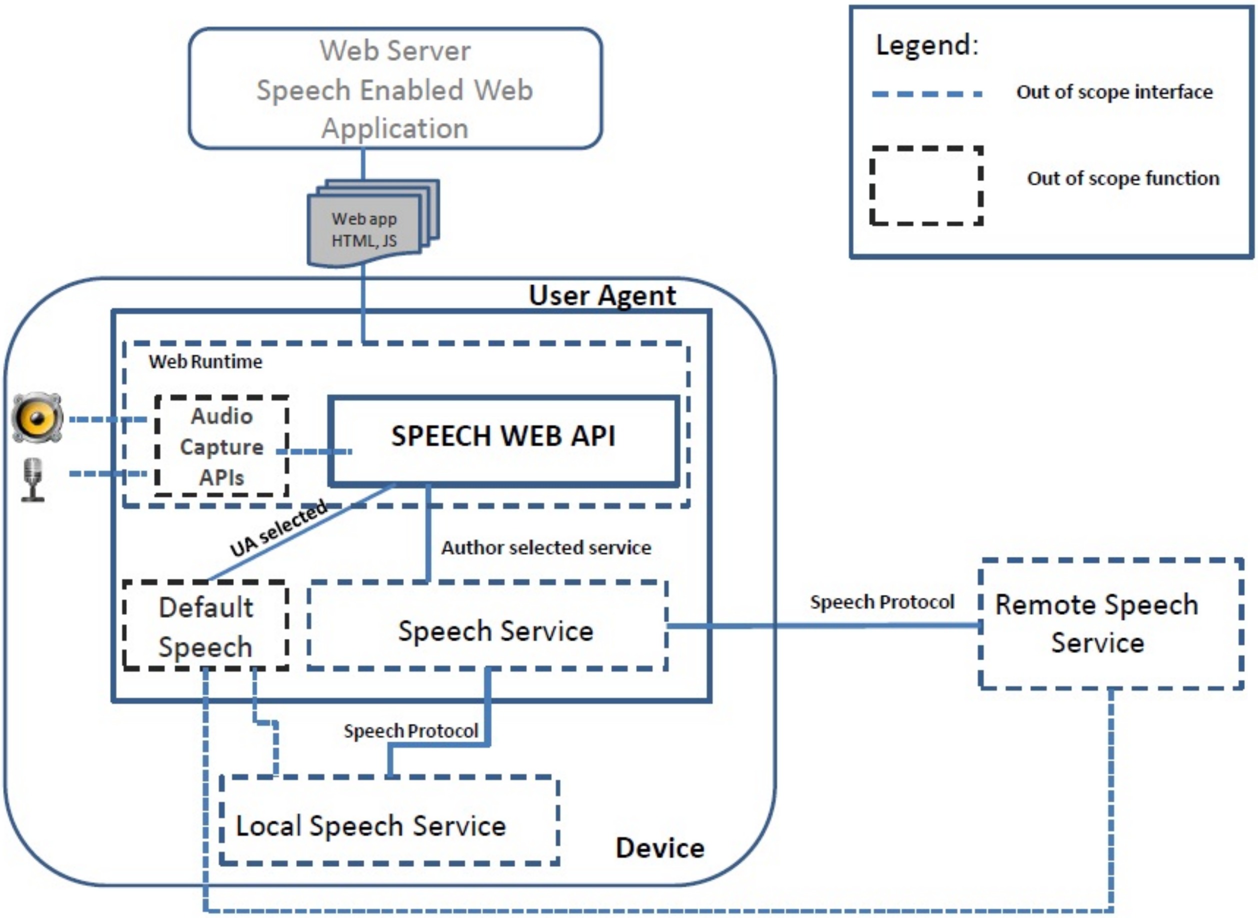


Figure 8. Overview of the HTML Web Speech API [9]

1.3.7 Google Chrome

Google Chrome [21] is a web browser developed by Google which is distributed as freeware. Google Chrome used the HTML WebKit layout engine until version 27, and WebKit Blink from version 28 and beyond. Its implementation on iOS, however, doesn't use the same engine as its build on other platforms.

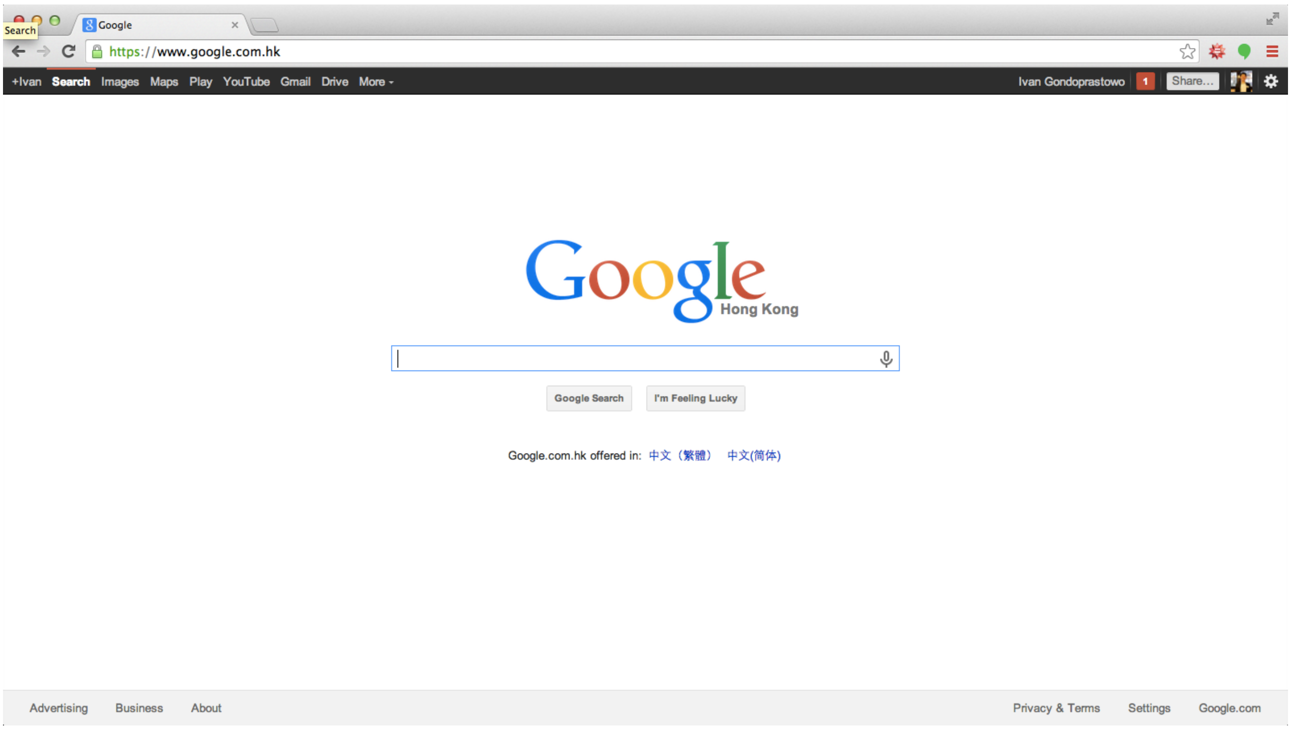


Figure 9. Google Chrome browser UI

1.3.8 Secure Sockets Layer (SSL)

According to the Apache Tomcat 6.0 SSL Configuration Page [22], SSL, which stands for Secure Socket Layer, is a technology which allows web browsers and servers to communicate over a secure connection. The data being sent is encrypted by one side, transmitted, and then decrypted by the other side before processing. As this is a multi-stage process, both the server and the browser encrypt all traffic before sending out data, and decrypt it when receiving data. Our system uses SSL because encryption is required by Google Chrome when a web application requires accessing a device’s mic or camera as a default.

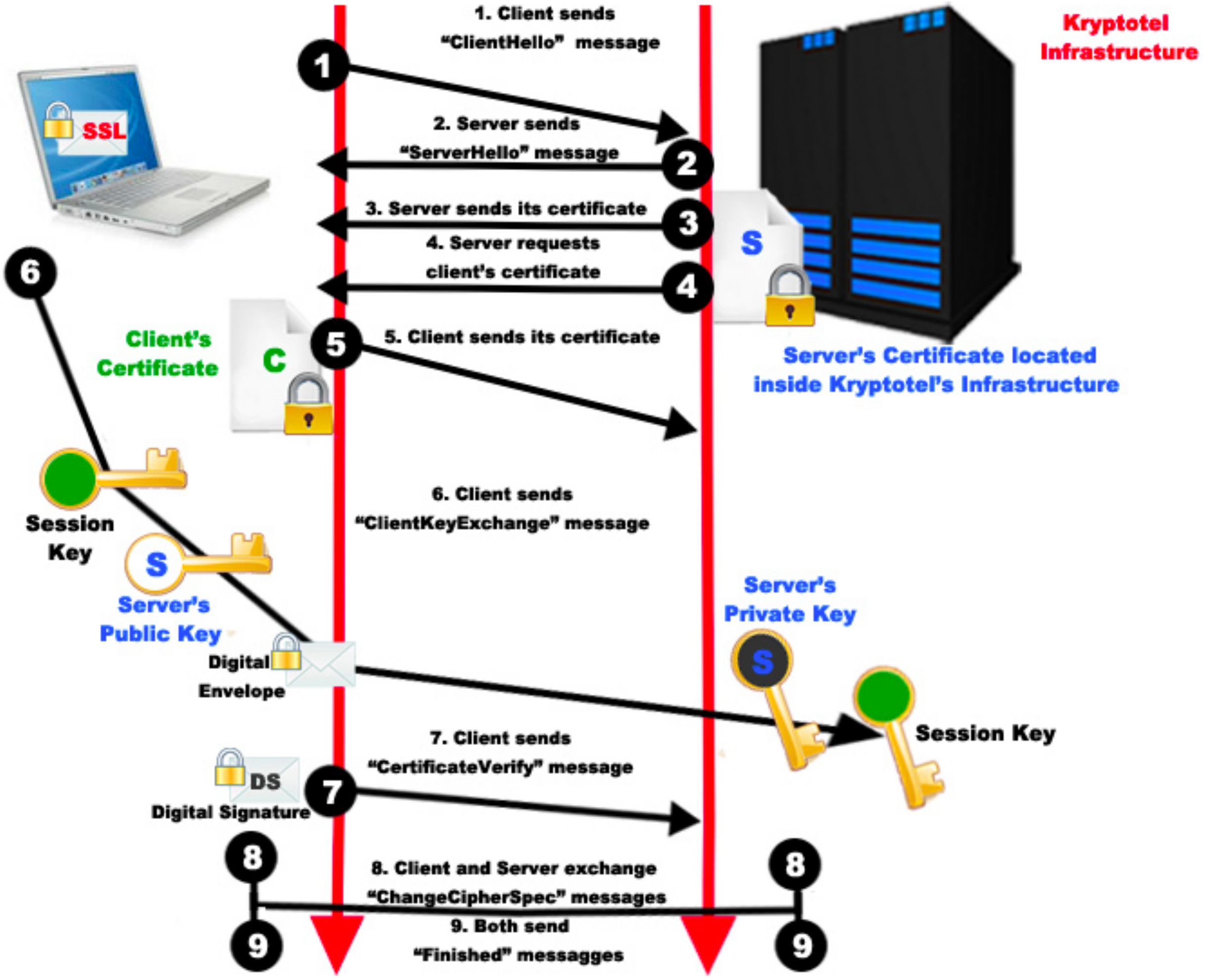

1. Client sends "ClientHello" message
Kryptotel Infrastructure
SSL
2. Server sends "ServerHello" message
3. Server sends its certificate
4. Server requests client's certificate
S
Client's Certificate
C
5. Client sends its certificate
Server's Certificate located inside Kryptotel's Infrastructure
6. Client sends "ClientKeyExchange" message
Session Key
S
Server's Public Key
Server's Private Key
S
Digital Envelope
Session Key
7. Client sends "CertificateVerify" message
DS
Digital Signature
8. Client and Server exchange "ChangeCipherSpec" messages
9. Both send "Finished" messagges
1
2
3
4
5
6
7
8
8
9
9

Figure 10. Illustration of how SSL works [23]

1.3.9 Named-Entity Recognizer

According to Nadeau and Sekine [24], Named Entity Recognition is one of the most important sub-tasks of information retrieval. The idea is that it is essential to recognize information units, like names, such as person, organization and location names, or numeric expressions, such as date, time or nominal values. This "Named Entity Recognition and Classification" was first coined by Grishman and Sundheim [25]. A Named Entity Recognizer is a natural language processing tool that locates and classifies atomic elements in a string into predefined classes, such as the names of places, persons, organizations, expressions of time, quantities and other classifications. An example of Named-Entity Recognizer output would be:

Input: Jim bought a burger from McDonalds

Output: Jim (name) bought a burger (food) from McDonalds (place)

Our system looks for user output for each required class and requests missing class data if necessary.

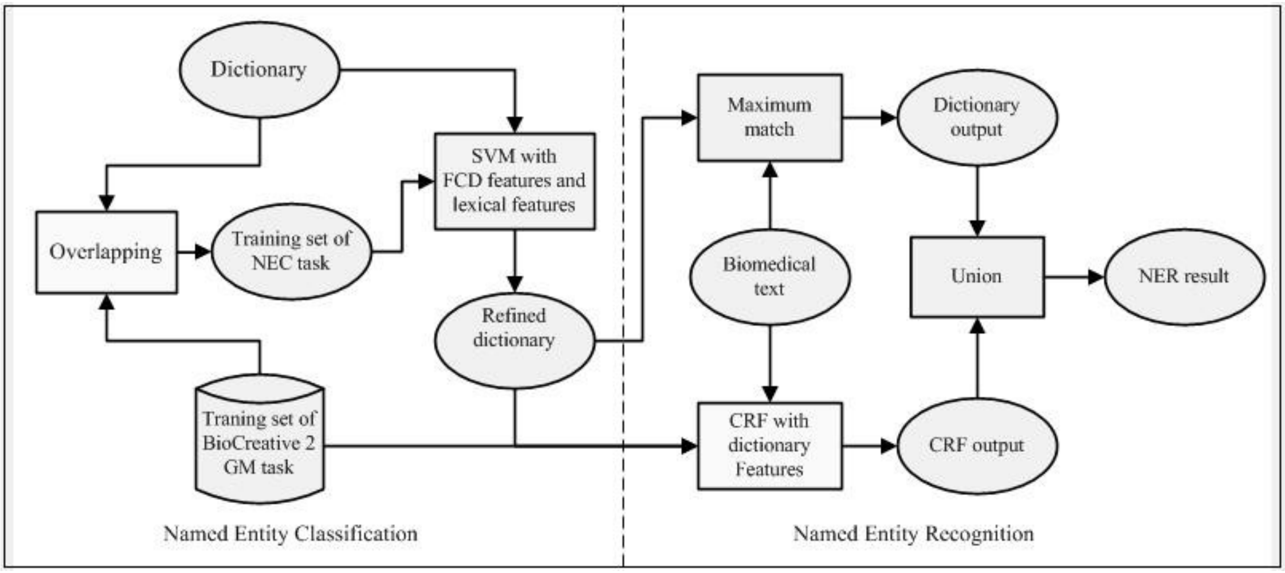


Figure 11. Overview of the Named Entity Recognition Process [26]

1.3.10 Part of Speech Tagger

According to Toutanova and Manning [27], Part of Speech (POS) Tagging is the automatic assignment of parts of speech in a sentence. This is usually done by employing machine learning methods. POS Tagger is a natural language processing tool that identifies the part-of-speech that a word or phrase belongs to. This is particularly useful when creating a system that needs to analyze natural language queries. We employ POS Tagger in our system to validate if input speech is grammatically correct and if it is a valid English sentence or not.

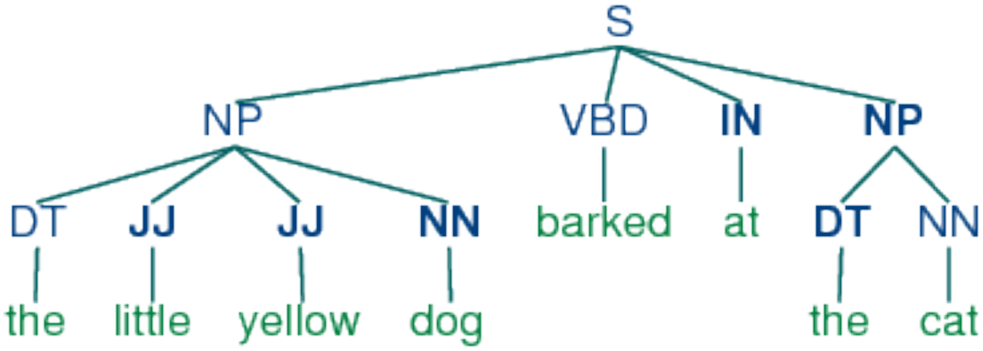


Figure 12. Example of Part of Speech Tagging Diagram [28]

1.3.11 Maven

Apache Maven [29] is a tool for software project management and comprehension. Its concept is based around the project object model (POM). Maven manages a project's build, reporting and documentation from a central piece of information. Maven allows developers to define and retrieve libraries from a centralized repository, hence reducing the redundancy and difficulties in dealing with different versions of library.

1.3.12 Academic Literature and Seminars

We also found some academic literature and seminars that seemed like they might be relevant to the project, either for implementation of our voice assistant or for user experience measurement.

Mishra and Bangalore [30] discuss question-answering by utilizing speech input for query retrieval. This is particularly relevant to our project, as our project and other voice assistant implementations that we address utilize the speech input method for query retrieval.

Vertanen and Kristensson [31] discuss recognition and correction of voice web search queries. While we are not very interested in web search queries, this paper contains methods to recognize and correct queries based on a speech input method.

We also considered tasks to be completed in our user experience study, since this is an essential part of our project that could determine its success. Crystal and Ellington [32] present practical ways to do task analysis for human-computer interaction problems. We relied heavily on their methodology to decide what tasks should be included in our experiment.

In determining user experience metrics that should be included in the study, we found that Tullis and Albert's suggestions [10] are particularly useful. They claim that there are several measurements that we can use. For the purpose of our study, we are interested in understanding task completion, system performance and accuracy, which affect user satisfaction regarding the two approaches.

Furthermore, as mentioned above, accents and other language proficiency issues are possible obstacles. A seminar by Ma [33] attempted to address these problems. Ma presented ways to utilize a crowdsourcing method to bridge communication barriers. It was interesting, but it was not be directly applicable to this study.

2. Design

We broke this project down into two big chunks, system implementation and the user experience study. A simple intelligent voice assistant system was designed to support and facilitate the user experience study. Ideally, implementation of the system would resemble the current widely-used intelligent voice assistant systems, but for practical purposes, we narrowed down the supported tasks to simply finding suitable places to eat and drink. The following sections describe our design procedure.

2.1 Analyze current voice assistant implementations

We thoroughly analyzed two current intelligent voice assistant systems, Siri and Google Now, to examine how they work and to decide which features to implement in our experimental voice assistant model.

2.2 Analyze the search function of the Yelp! API

Yelp! provides an API that is publicly available for registered developers. We carefully analyzed the Yelp! search implementation and used the Yelp! API to interface with their server. We designed a data parser for search result parsing, as the search results are available in JSON format through Yelp!'s REST API.

2.3 Design the voice assistant system for experimentation

2.3.1 Design the user experience and usage flow

Firstly, we designed the overall user experience and usage flow for the voice assistant. This includes the user experience for both flows, i.e., the flow of usage from the launch of the application or when the user activates the query recording mode, to the end of usage or when the user receives the result. (See Figures 13 and 14.)

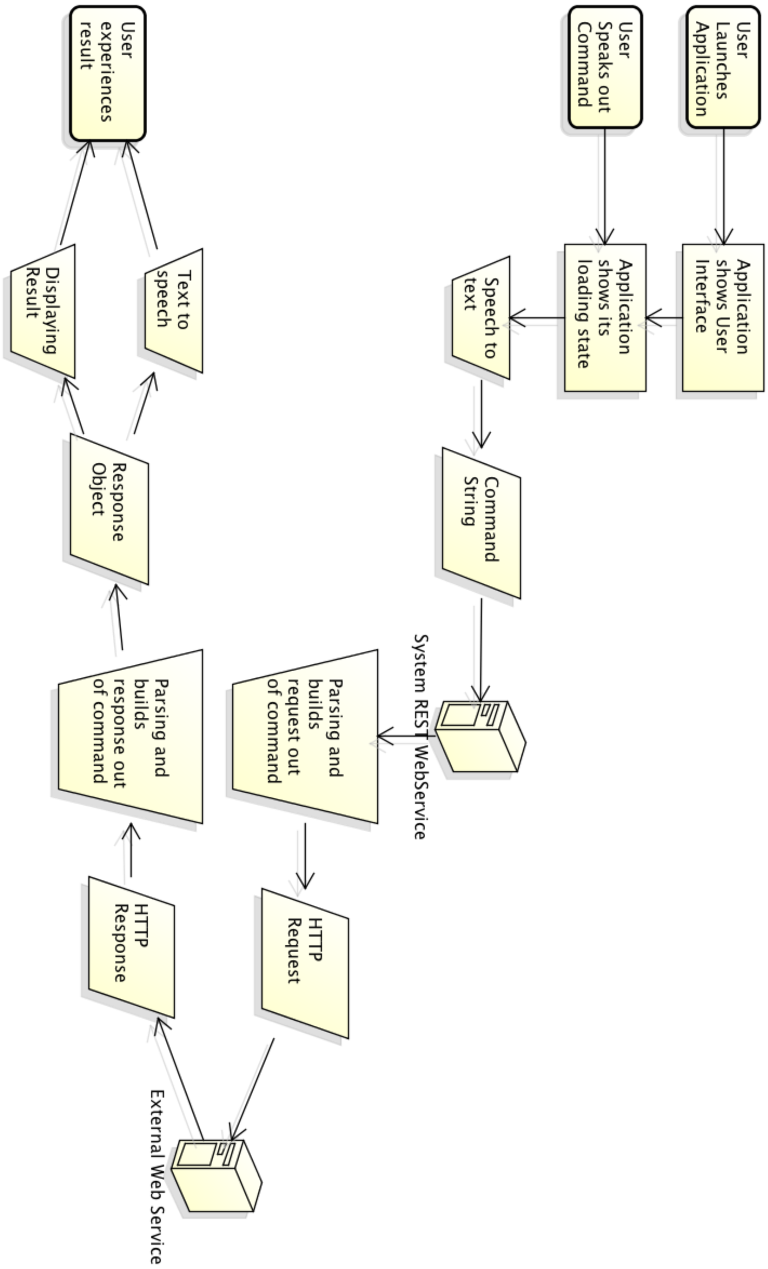


Figure 13. User Experience Design and Usage Flow for implementation A (without command refining)

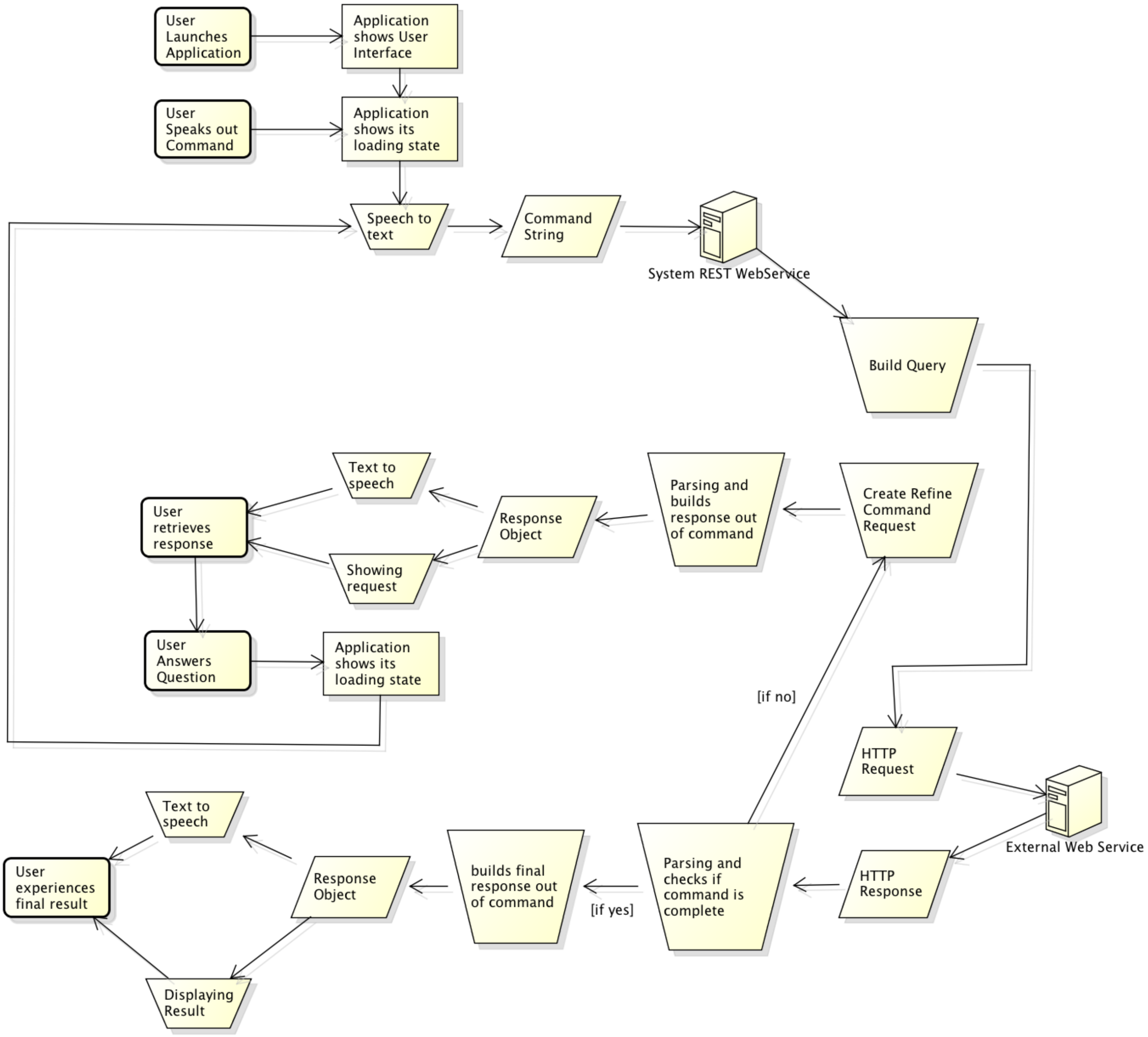

User Launches Application
Application shows User Interface
User Speaks out Command
Application shows its loading state
Speech to text
Command String
System REST WebService
Build Query
Text to speech
User retrieves response
Response Object
Parsing and builds response out of command
Create Refine Command Request
Showing request
User Answers Question
Application shows its loading state
[if no]
HTTP Request
External Web Service
Text to speech
User experiences final result
Response Object
builds final response out of command
[if yes]
Parsing and checks if command is complete
HTTP Response
Displaying Result

Figure 14. User Experience Design and Usage Flow for implementation B (with command refining)

2.3.2 Design the service implementation

Secondly, we designed a method to implement the service. This includes the flow from when the client sends a query or a request to when the service locally processes it and sends the processed request to the external web service, retrieves and formulates the response and then sends it back to the client side. It also includes deciding which platform to use to implement the intermediary web service (the server side). (See Figure 15.)

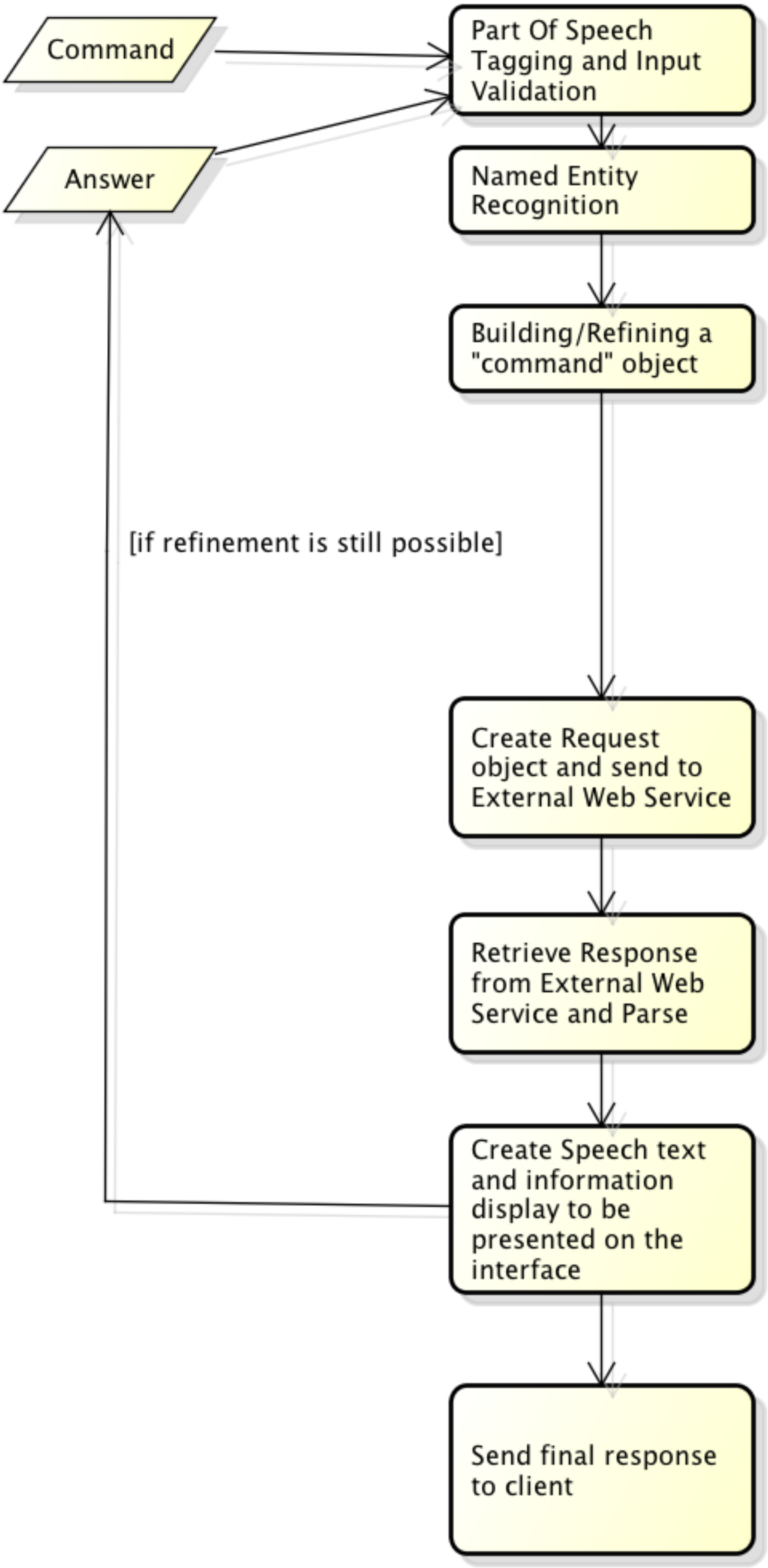

Command
Answer
Part Of Speech Tagging and Input Validation
Named Entity Recognition
Building/Refining a "command" object
[if refinement is still possible]
Create Request object and send to External Web Service
Retrieve Response from External Web Service and Parse
Create Speech text and information display to be presented on the interface
Send final response to client

Figure 15. Service Implementation Flow Diagram

2.3.3 Design the client-server interaction

Thirdly, we designed the client-server communication configuration, i.e., the method we use to implement communication from the client to the server (request) and from the server back to the client (responses). This includes the messaging format and the size limit as well as the request and response formats. (See Figure 16.)

2.3.4 Design the user interface

Finally, for the service implementation part, we designed the user interface to replicate the implementation of the Siri voice assistant user interface design. (See Figures 1 and 2 in section 1.1)

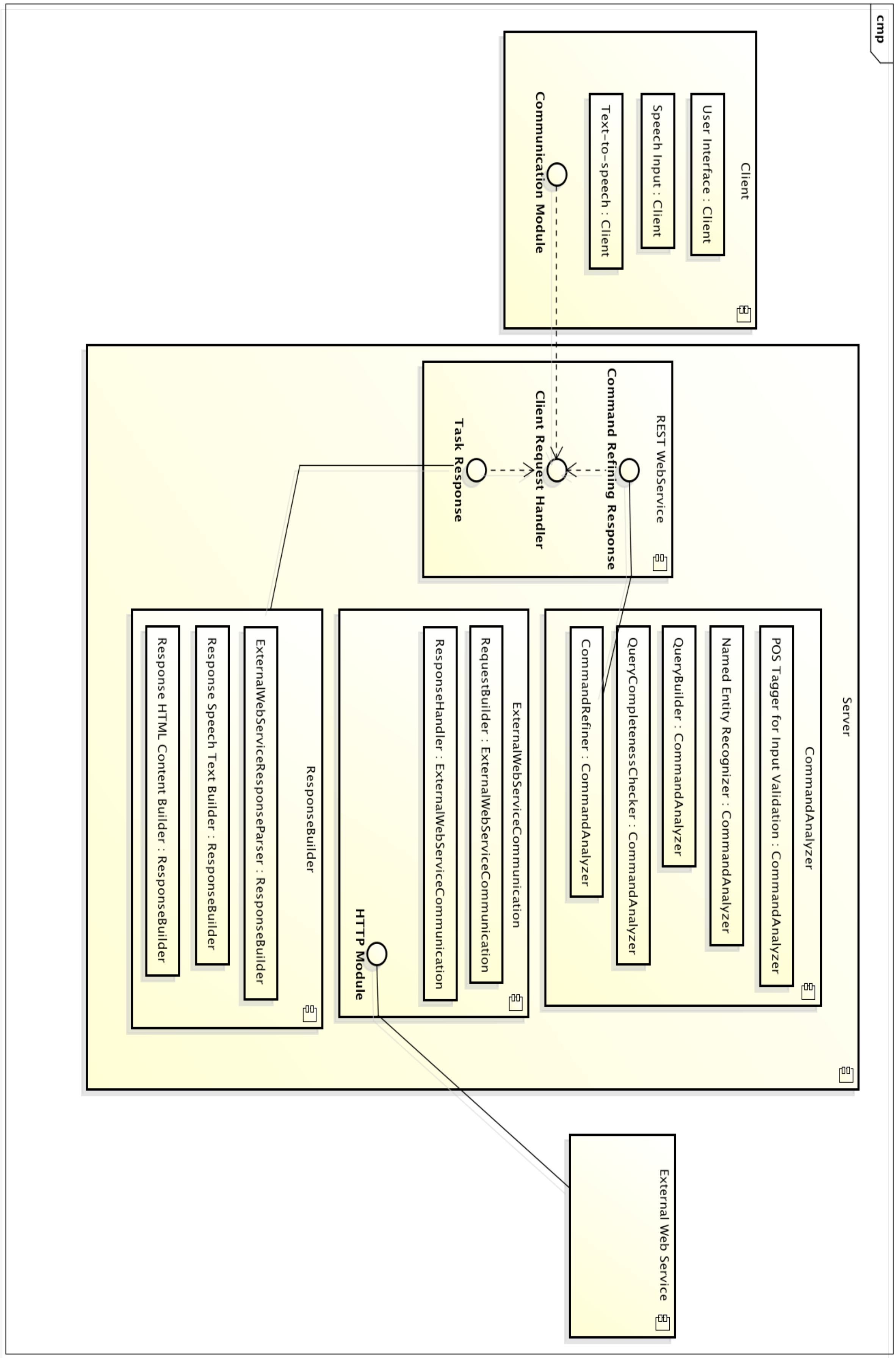

cmp
Client
User Interface : Client
Speech Input : Client
Text-to-speech : Client
Communication Module
Server
REST WebService
Command Refining Response
Client Request Handler
Task Response
CommandAnalyzer
POS Tagger for Input Validation : CommandAnalyzer
Named Entity Recognizer : CommandAnalyzer
QueryBuilder : CommandAnalyzer
QueryCompletenessChecker : CommandAnalyzer
CommandRefiner : CommandAnalyzer
ExternalWebServiceCommunication
RequestBuilder : ExternalWebServiceCommunication
ResponseHandler : ExternalWebServiceCommunication
HTTP Module
ResponseBuilder
ExternalWebServiceResponseParser : ResponseBuilder
Response Speech Text Builder : ResponseBuilder
Response HTML Content Builder : ResponseBuilder
External Web Service

Figure 16. Component Diagram of the Overall System Implementation

2.4 Design the user experience study

2.4.1 Plan the study subjects

Firstly, we decided to find suitable study subjects, around 30 students who had some experience with Siri or Google Now and those who did not have. They can be further divided into groups of people from different ages and socio-cultural backgrounds.

2.4.2 Design the questions

Secondly, we decided on measurements to be used to examine the user experience and we came up with a set of questions that captures the differences between the current and proposed approaches for implementation of human-computer interaction using an intelligent personal assistant.

2.4.3 Design the overall study flow

Thirdly, we planned the overall study flow, which meant what was going to happen when subjects enter the room, what screen would be shown or how we were going to let him or her use the application, when to present the questions, what things we were going to tell them and how we would end the study. We must also decide how many people to invite to each study session.

3. Implementation

The implementation of this project was divided into two phases. The first phase entailed implementing the system for experimentation, and the second phase involved data collection through the user experience study. We also analyzed the gathered data and derived conclusions from it.

3.1 Voice Assistant System Implementation

The voice assistant system is comprised of two parts. The first part follows the current implementation of widely used voice assistant systems using single-stage human-computer commands, and the second part follows the interactive approach. Both implementations were needed to support the user experience study, and their implementation followed the design described in section 2.

This voice assistant system implementation is comprised of front and back ends, following a client-server design pattern. The client includes the user interface, speech recognizers, a command validator and other necessary features. The server includes command analysis, query feature completion, communication with an external web service and retrieval of results from an external web service. Furthermore, we also implemented the designed communication method between the client and the server, which resembles requests and responses according to a client-server design pattern.

3.1.1 Back-end Implementation

The back-end implementation as mentioned above represents the server in the client-server design pattern. The server is responsible for the following duties:

- Providing an interface for clients to access the services offered by the server.

- Creating a command object which represents a unique voice assistant task requested by the user
- Randomly choosing either the A or B implementation for a particular command object
- Analyzing entities in each command
- Asking questions to request that users provide more information so as to refine the command
- Interfacing with the Yelp! API
- Sending a search query to the Yelp! API
- Parsing the response of the search query from Yelp! API
- Building necessary html responses
- Sending html responses back to where html requests are being sent from
- Logging user actions and processes

The following sections will describe how the parts above were implemented.

3.1.1.1. Providing an Interface for Client Access

One of the most important back-end functionalities of the server is to let the clients access the services it offers. This was done by implementing the representational state transfer (REST) architectural style. Implementation wise, this is simply another way of setting up a Java based web server. However, it allows clients to access the server through an interface that is defined slightly differently from basic Java servlet-based web server.

To implement a REST web service on a Java web server, in our case, Tomcat 7.0, we used Jersey, a reference implementation of JAX-RS, which is a Java API for RESTful Web Services [34]. This allows us to use annotations defined in JSR311, which made it easy for us to build a REST based web services by using the Java programming language.

In our codebase, the class that implements REST based web services is the `HelloWorld` class (`givan.ug.hkust.fyt.rest.HelloWorld`). Besides the higher level details in setting up the web server and the REST web service, the class contains two main functions, one helper function, and one function for system integration and flow testing purposes. The definitions for these functions are:

- `public String analyzeCommand(@FormParam("initialCommand") String command) throws ParseException`
- `public String refineCommand(@FormParam("originalCommand") String ori_command,@FormParam("refiningAnswer") String refine_ans, @FormParam("refiningType") String refine_type, @FormParam("refineLog") String refine_log) throws ParseException`
- `public String analyzeCommandWithoutSpeech() throws ParseException`
- `protected String searchYelpByCommand(Command c, HTMLBuilder htmlBuilder) throws ParseException`

The function `analyzeCommand` is responsible for handling a new instance of a command or a unique task being requested by the users. This function creates a command object based on the command the user inputs through the speech recognition function. The function then analyzes the command and extracts entities from the natural language sentence given. Finally, it randomly decides whether implementation A or B shall be used for this particular task.

In the event of implementation A (single-stage) being chosen, it checks if a location entity exists in the command, if not, it appends the default location to the command. The default location is set to San Francisco. It then requests the `CommandHandler` class to handle the request, i.e., formulate the query, send the query to the Yelp! server and parse the response back. The implementation A process ends with building an HTML response to the command and sending the response to the user.

In the event of implementation B (multi-stage) being chosen, it checks if the command has the necessary entities. These are defined to be location, price and restaurant type. If either of this is missing, the user is prompted with questions to refine his or her command. The system generates an HTML response which asks the user to provide more information, and directs him or her to `refineCommand` instead of `analyzeCommand`. Once the command has both entities needed, the service asks the `CommandHandler` to send a query to the Yelp! API, parse the response from Yelp!, and builds an HTML response to be sent back to the client.

Following is the pseudocode which implements the REST web service that acts as an interface between the back-end and the front-end part of the system.

```
Method: Voice Assistant Web Service
Return: HTTP response, String with HTML format
If the request is analyze command request
 Create command object based on input parameter (recognized speech)
 Generate random boolean number
 If implementation A is chosen
     Check if the location exists, if not, append the default location
     Set the command to the CommandHandler
     Search Yelp! given the command, and build the HTML response
 Else if implementation B is chosen
     Check if one or more required entities are still missing
         Build the HTML response to ask for refinement
     Else
         Search Yelp! given the command, and build the HTML response
```

```
Else if request is refinement command
 Create a refinement command, refined based on the recent answer and refine log
 Check if one or more required entities are still missing
         Append the recent answer to refinement log
         Build the HTML response to ask for refinement
 Else
         Search Yelp! given the command, and build an HTML response
Return the HTML to the clinet
End
```

3.1.1.2 Creation of Command object

As mentioned in the previous section, each unique task given by the user is treated as a new instance of a Command object. This object is to be refined throughout the whole process until the user receives the final response for the task. The object encapsulates several other String and Boolean objects. Following is the definition of `Command` (`givan.hkust.ug.fyt.communication`) object:

- location: String, stores the location entity detected from named entity recognition.
- object : String, stores the object entity detected by running named entity recognition and `KeywordList`
- price: String, stores the requested price range by running `KeywordList`
- original_command: String, stores the original command recognized by the speech recognizer on the front-end
- refine_log: String, stores the log of refinement of commands
- object_adj: boolean, checks if an object has an adjective
- do_search: boolean, checks if we have to ask the user to do a Google search instead

Upon creation of an instance of a `Command` object by the `analyzeCommand` function, the program stores and analyzes the entities in the original command and stores them in the variables described above. The process is slightly different when the object is instantiated by the `refineCommand` function. After the named entity recognition process is done, the `Command` object reads the log of command refinement which is being passed as a parameter for the constructor of this object.

The `Command` object implementation also includes a getter and a setter for all variables, and several helper functions, including implementation of refinement log reader, the named entity recognition result parser and the default location appender.

3.1.1.3 Named Entity Recognition

As mentioned in the previous section, named entity recoginition happens during the process of instantiating a `Command` object. For simplicity, we are using the classifier and library provided by Stanford University, which is often referred to as a CRFClassifier [35]. The classifier we use is english.conll.4class.distsim.crf.ser.gz, which is capable of recognizing 4 classes of entities:

- Location
- Person
- Organization
- Misc

This model uses distributional similarity features which, according to its website, provide some performance gain at the cost of increasing their size and runtime.

The process itself is being encapsulated as instantiation of the `EntityRecognizer` object. Upon the first call of this constructor, the classifier is loaded to the main memory, and being used for the purpose of recognizing the named entity. As the size of classifier is huge, we force the program to only call the constructor once, which forces at any point of time, there is at most one classifier loaded in the memory.

The entity recognition process is done whenever the program calls the `recognizeToXMLString` function. As the result returned is in XML format, the recognition process includes XML parsing, which extracts the entities and stores them in the variables in the `Command` object.

3.1.1.4 The Keyword List

The `KeywordList` object encapsulates vectors of keywords for different categories that might not be identified by the named entity recognizer as an entity of significance. The object contains a function that scans a sentence or part of a sentence for words that might be meaningful, such as price, restaurant type, and locations.

The keywords were sourced from the Yelp! API categories developer guide, or common knowledge. The list of restaurant category keywords which are supported by Yelp! API as search parameters are listed in Appendix B.

3.1.1.5 Command Handling

As mentioned in section 3.1.1.1, one of the processes being done is to send a query to the Yelp! API, retrieve the response, and parse it. These steps are being done by the `CommandHandler` object. The object encapsulates all the processes and object needed. Due to the size of the object, it only needs to be instantiated once. Later, we will describe what functions are in this object, that make it reusable.

The following are the attributes of the object:

- c: Command, the command to be handled
- yelpHandler: Yelp, the object that encapsulates generation of search queries, and interfacing with the Yelp! API, as well as retrieving the response back.
- yelpParser: YelpResultHandler, the object that is responsible for implementation of the Yelp! response in JSON format that parses into a `YEntity` object, which is an object that encapsulates information of a particular restaurant.

As mentioned above, each restaurant in the search results is represented as a `YEntity` object. The following are the attributes of the `YEntity` object:

- entityName: String
- entityAddress: String
- entityType: String, example: restaurant, bars
- entityPrice: int, represented as $ sign, 1 $ sign as the cheapest, 5$ as the most expensive
- entityRating: double, range between 0-5, with 0.5 steps
- entityReviewCount: long
- entityLink: String
- entityRatingImageLink: String

Our implementation of command handling is comprised of two parts, setting the command in the commandHandler through the `setCommand(Command c)` method, and getting a vector of the `YEntity` by sending a search query to Yelp!

and parsing it. In particular, the following the pseudocode of the Yelp! interfacing method is as follows:

```
Method: searchYelpUsingCommand

Returns: Vector of the YEntity

Start

 If(command is not ready, i.e. some entities are missing)

        Set the command to do a Google search instead

        Return an empty vector

 Search Yelp! using the parameter

 Parse the search results into a vector of YEntity

 Return the vector

End
```

It is worth noting that in sending our search query to the Yelp! interface, we are using OAuthRequest as specified for the Yelp! API 2.0. We add the necessary parameters and sign the request using our access token before sending the request out. Upon successful request, a response in JSON format is going to be retrieved. Below is an example of a JSON response from the Yelp! API:

```
{
  "businesses": [
    {
      "categories": [
        [
          "Local Flavor",
          "localflavor"
        ],
        [
          "Mass Media",
          "massmedia"
        ]
      ],
      "display_phone": "+1-415-908-3801",
      "id": "yelp-san-francisco",
      "is_claimed": true,
      "is_closed": false,
      "image_url": "http://s3-media2.ak.yelpcdn.com/bphoto/7DIHu8a0AHhw-
BffrDIxPA/ms.jpg",
      "location": {
        "address": [
          "140 New Montgomery St"
        ],
        "city": "San Francisco",
        "country_code": "US",
        "cross_streets": "3rd St & Opera Aly",
        "display_address": [
          "140 New Montgomery St",
          "(b/t Natoma St & Minna St)",
          "SOMA",
          "San Francisco, CA 94105"
        ],
        "neighborhoods": [
          "SOMA"
        ],
        "postal_code": "94105",
        "state_code": "CA"
      },
      "mobile_url": "http://m.yelp.com/biz/4kMBvIEWPxWkWKFN__8SxQ",
      "name": "Yelp",
      "phone": "4159083801",
      "rating_img_url": "http://media1.ak.yelpcdn.com/static/
201012161694360749/img/ico/stars/stars_3.png",
      "rating_img_url_large": "http://media3.ak.yelpcdn.com/static/
201012161053250406/img/ico/stars/stars_large_3.png",
      "rating_img_url_small": "http://media1.ak.yelpcdn.com/static/
201012162337205794/img/ico/stars/stars_small_3.png",
      "review_count": 3347,
      "snippet_image_url": "http://s3-media2.ak.yelpcdn.com/photo/
LjzacUeK_71tm2zPALcj1Q/ms.jpg",
      "snippet_text": "Sometimes we ask questions without reading an email
thoroughly as many of us did for the last event.  In honor of Yelp, the
many questions they kindly...",
      "url": "http://www.yelp.com/biz/yelp-san-francisco",
      "menu_provider": "yelp",
      "menu_date_updated": 1317414369
    }
  ],
  "region": {
    "center": {
      "latitude": 37.786138600000001,
      "longitude": -122.40262130000001
    },
    "span": {
      "latitude_delta": 0.0,
      "longitude_delta": 0.0
    }
  },
  "total": 10651
}
```

Upon retrieval of such a response, we parse the JSON file and store necessary information in the `YEntity` object.

3.1.1.6 HTML Response Building

HTML response building is a part of the process that is integral to the overall flow of the system. Initially, users access the static html through their web browsers. Upon the request however, users are to be returned responses that are dynamically created. These responses are being sent by the server as a result of the requests. `HTMLBuilder` is the object which is responsible for producing HTML output. This is done by re-using pre-defined templates and filling in data that is produced as a result of the process. The general pseudocode for HTML response methods is as follows:

```
Method: Build an HTML response
Return: String, Hypertext markup
Start
Allocate string space for the htmlResponse
Append a header template to the htmlResponse
If(command needs to be refined)
 Append the text response
 Append the voice response
 Append the text area form
Else
 Append the text response
 Append the voice response
 For each (YEntity in Vector of YEntity)
         Display the restaurant list
Append a footer template to the htmlResponse
```

```
Eeturn
End
```

It is worthwhile to note that the response actually takes the form of a String. However, for `analyzeCommand` and `refineCommand`, note that the response is in HTML has been made by using JAX-RS annotation. Therefore, the web browser handles the response as an HTML formatted page.

3.1.1.7 Logging

As this project put emphasis on the user experience study, logging or recording processes and user actions was a crucial part of the project. Log allows us to trace back the human-computer interaction and the communication happening between the user and our implementation of voice assistant. This is helpful for analyzing the result of corresponding communication happening during user's interaction with the system.

In our project, the logging implementation was done with the aid of log4j, a well-known Java library for logging. The log is stored in a text file. Each line of log is prefixed with the name of the class where the process happened and the time. For each command, the implementation that is chosen is also recorded, together with the user's initial command, entities detected in the command and refinement of the initial command.

Below is an example of what is recorded in the log:

```
[05 Feb 2014 16:10:23] 16:10:23,682 INFO  [HelloWorld] REFINE request is
received at Wed Feb 05 16:09:42 HKT 2014
[05 Feb 2014 16:10:23] 16:10:23,682 INFO  [HelloWorld] REFINE LOG:
```

```
[05 Feb 2014 16:10:23] 16:10:23,682 INFO  [HelloWorld] REFINE TYPE:
LOCATION
[05 Feb 2014 16:10:23] 16:10:23,682 INFO  [HelloWorld] REFINE ANSWER: San
Francisco
[05 Feb 2014 16:10:23] 16:10:23,682 INFO  [Command] instantiating command
object from REFINE request
[05 Feb 2014 16:10:23] 16:10:23,682 INFO  [Command] recognizing and
building command from string
[05 Feb 2014 16:10:23] 16:10:23,682 INFO  [EntityRecognizer] Recognizing
entities in given command
[05 Feb 2014 16:10:23] 16:10:23,685 INFO  [Command] refining command based
on answer and type
[05 Feb 2014 16:10:23] 16:10:23,685 INFO  [EntityRecognizer] Recognizing
entities in given command
[05 Feb 2014 16:10:23] 16:10:23,686 INFO  [HelloWorld] command is created
successfully, analysis of command:
[05 Feb 2014 16:10:23] 16:10:23,687 INFO  [HelloWorld]
COMMAND:find me some Japanese restaurant
OBJECT:Japanese restaurant
LOCATION:San Francisco
PRICE:
[05 Feb 2014 16:10:23] 16:10:23,687 INFO  [HelloWorld] command is now
complete, searching yelp and building html result
[05 Feb 2014 16:10:23] 16:10:23,687 INFO  [CommandHandler] Searching Yelp
using given command
[05 Feb 2014 16:10:25] 16:10:25,119 INFO  [YelpResultHandler] Parsing
Yelp! search result
[05 Feb 2014 16:10:25] 16:10:25,123 INFO  [HTMLBuilder] building HTML out
of search result
```

This log is stored under the /log folder in a textfile named thesis.log. The properties of log4j are described in an XML formatted file. (See Appendix D.)

3.1.1.8 Maven Project Management

Our project implementation uses Maven as its build manager. This was helpful in avoiding difficulties in dealing with storing libraries, referring to libraries, and managing different versions of libraries. A full list of dependencies of the project can be seen in the Maven project object model xml. (See Appendix E.)

It is worthwhile to note that the Stanford NER is not available on centralized Maven libraries. For this reason, the Stanford NER library is still manually appended to the project build.

3.1.1.9 Dynamic Hostname Handling

As user will be using the system on a tablet, it is necessary for us to create a service that is accessible through network. In doing this, we are using the Mac “File Sharing” feature to open the server to connection from outside. An important issue encountered was the fact that the network is assigning different hostname for each connection session to the server.

This creates a need for a module in the web service that allows for dynamic hostname to be handled by the server. This is being done by encoding the hostname for each request sent to the server. This hostname will be passed back and forth until the process finishes.

3.1.2 Front-end Implementation

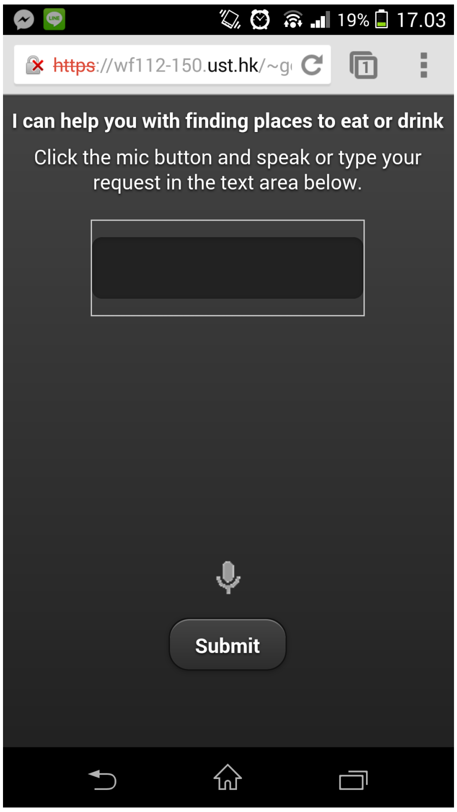
The front-end part of the system is comprised of a user interface, speech input module and text-to-speech output module. The UI and the user experience (UE) were modelled after the UI and UE of Siri on iOS 6. The front-end part of the system is implemented as an HTML document, styled with CSS and aided with JavaScript for implementation of speech input and text-to-speech output. The following sections describe how the components of the front-end part of the system are being implemented.

3.1.2.1 User Interface

In order to implement a design that is platform-agnostic, we are using the jQuery Mobile library. This library allows us to implement a consistent design that can be viewed from multiple platforms and various devices.

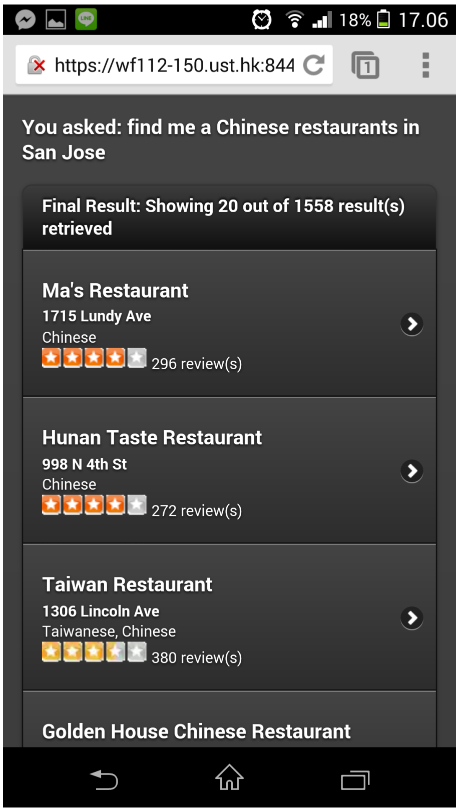
The user interface is comprised of several elements. The main elements are texts that are being used to display the original command, the response from the server, and also the resulting restaurants. The user interface also contains a text area to store the command recognized by the speech input module, and a mic button which acts as a trigger for speech recognition. Users are able to activate the speech recognition module by clicking on the mic button. In the event that the the speech recognition fails to recognize the end of the signal, user can tap the submit button to proceed. When users are not in the initial page, a button that redirects them to the start page will be shown.

Following are the screenshots of the user interface on different parts of the usage flow.



I can help you with finding places to eat or drink

Click the mic button and speak or type your request in the text area below.



You asked: find me a Chinese restaurants in San Jose

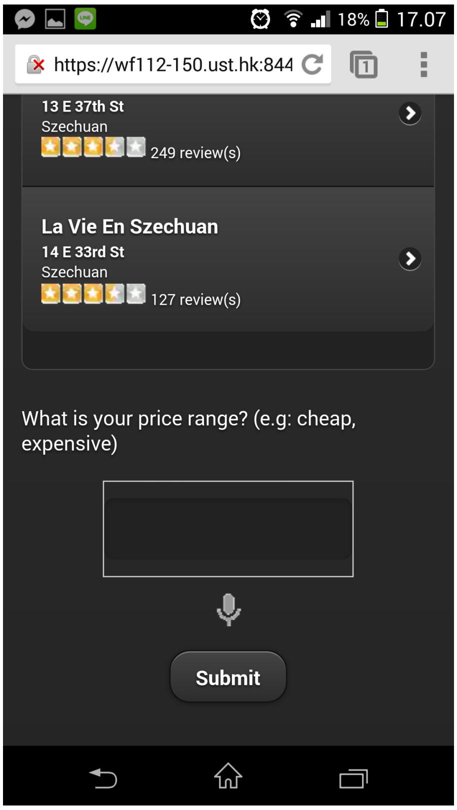
Final Result: Showing 20 out of 1558 result(s) retrieved

Ma's Restaurant
1715 Lundy Ave
Chinese
296 review(s)

Hunan Taste Restaurant
998 N 4th St
Chinese
272 review(s)

Taiwan Restaurant
1306 Lincoln Ave
Taiwanese, Chinese
380 review(s)

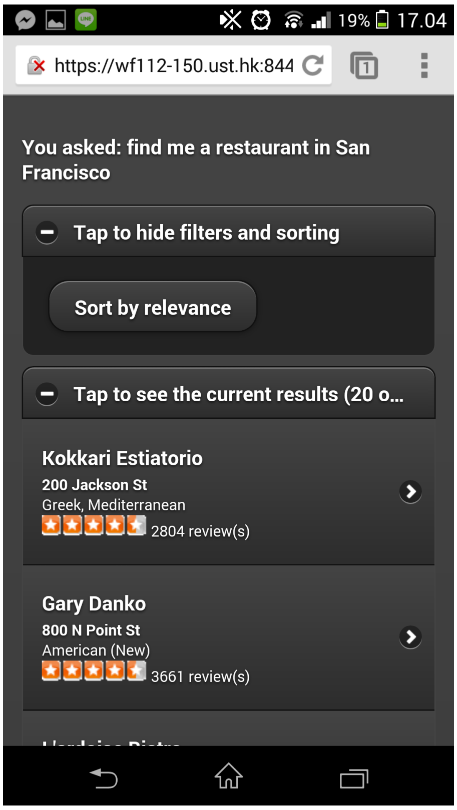
Golden House Chinese Restaurant



13 E 37th St
Szechuan
249 review(s)

La Vie En Szechuan
14 E 33rd St
Szechuan
127 review(s)

What is your price range? (e.g: cheap, expensive)


19% 17.04
https://wf112-150.ust.hk:844
Tap to hide filters and sorting
Sort by relevance
Tap to see the current results (20 o...

You asked: find me a restaurant in San Francisco

Kokkari Estiatorio
200 Jackson St
Greek, Mediterranean
2804 review(s)

Gary Danko
800 N Point St
American (New)
3661 review(s)

Figure 17. Home screen interface design

Figure 18. Interface for showing results from implementation A

Figure 19. Command refining interface

Figure 20. Interface for showing results from implementation B

3.1.2.2 Speech Input

In order to create a Siri-like user experience, users must be allowed to give commands to their intelligent personal assistant by using a voice input method. This can be achieved by using the HTML Speech Input API, a browser speech recognition module that is available on Google Chrome 25 or above. This module is also used on common voice-based application, such as Google Voice Search [36].

Apart from in-browser implementation, a web page can call for this voice input module by using JavaScript. In our implementation, we embed the JavaScript in the text area so that recognized words will be displayed on the text area at real time. This module is triggered by the mic button. When the user clicks on the mic button, the module is activated and actively appends the recognized words until a pause is detected. Upon detection of a pause, it stops recognizing the speech input and sends the command to the server as an HTTP request.

Following is the pseudocode for the general speech input method that is being implemented in the system:

```
Method: Speech Recognition
Upon a click on the mic button, start recognition
    Set the cursor to the next available place in the text area
    Append recognized words starting from the cursor
    Repeat if words can still be recognized
    Else
        Indicate that the recognizer has finished recognizing
        Change the mic icon into “deactivated” mode
        Store the text area content to a variable
        Stop recognition
        Send the variable as a parameter to HTTP request
End
```

It is worthwhile to note that during the implementation of voice input on the web page, a problem regarding double printing of a full command when speech recognition was stopped was encountered. In order to solve this, we store the final text area content in a different variable that is different from the one that is being used as temporary space.

3.1.2.3 Text-to-speech

The implementation of the front-end part of the system also includes a text-to-speech module. Implementation of voice assistants usually implements text-to-speech to create the interaction more natural. The spoken response creates a sense of having a more natural communication to the users.

The implementation of our text-to-speech module is done by utilizing a service of Google Translate [37]. Google Translate allows us to generate an mp3 sound file based on a given sentence. This mp3 file is then played by using the <audio> HTML tag. The service is available through HTTP GET(REST) request accessible through the following link:

```
http://translate.google.com/translate_tts?tl=en&q=text
```

It is worthwhile to note that Google Translate does not allow direct text-to-speech generation from an HTML file. One of the service restrictions is that it does not allow a request without referred that is different from Google Translate itself. For this reason, we are utilizing a php file that acts as a gateway to retrieve this MP3 file before playing the sound file.

3.1.2.4 User Interface Logging

In order to better provide us with insights when performing the analysis, we implemented user interface logging feature by using JavaScript to record user interaction with the system. This includes taps, scrolls, clicks, on canvas or object. The log includes the time and division or class each action was being done. The following is an example of how the log is recorded with italicized words serving as description of log:

```
[0ms]Page loaded
[3211ms]Results collapsible expanded (user taps on result collapsible,
leaving filter collapsible expanded)
[4006ms]Scroll starts (scrolling through the results)
[4526ms]Scroll ends
[4591ms]Scroll starts
[5030ms]Scroll ends
[5110ms]Scroll starts
[5729ms]Scroll ends
[5755ms]Scroll starts
[6692ms]Scroll ends
[9371ms]Click detected at speech-page-content (clicking the text area :
filling in by keyboard)
[11584ms]Click detected at submit-button (clicking the submit button, not
automatically sent via recognizer)
```

Each line of log is recorded in a hidden text area which is encoded in the request to the REST web service. The REST web service will store the user interface log as a part of system log by using the logging module.

3.1.3 Overall System Implementation

The server is an Apache Tomcat 7.0 web server, and it hosts Java modules which are available through the REST interface provided by using the JAX-RS implementation, Jersey.

The server communicates with the client over the secured hypertext transfer protocol (HTTPS). Although we disregard the importance of security issues in this project, HTTPS is required to automatically grant permission to use the mobile device's microphone for speech input. While HTTP allows a device's microphone to be used as well, the user would need to notify the browser that he/she allows the script to use the microphone each time he/she wanted to use voice input, which would not be good for the user experience.

3.1.4 Implementation of SSL

As mentioned in section 1.3.8 and 3.1.3, the SSL needs to be activated because Chrome requires that accesses requiring a microphone or a camera be encrypted. Thus, we implemented SSL on the server state. For this project, we used Apache Tomcat 7.0 and an Apache Server.

In order to do this, we had to add the following connector to the server configuration on Eclipse. A keystore file is also generated to create a private key for encryption. The following lines of code wwere added to the server configuration to enable the SSL on Apache Tomcat 7.0 through port 8443.

```
<Connector port="8443" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="/Users/gondoprastowo/Documents/
workspace/.metadata/.plugins/org.eclipse.wst.server.core/tmp0/conf/
keystore.key"
keystorePass="mypassword" />
```

The only modification needed on the Apache server is modification of the httpd.conf file. This is done by simply enabling the SSL when running in administrator mode.

3.1.5 Completed System Modification

3.1.5.1 System Adjustment after Preliminary Study

As mentioned above, in order to get to know the proper depth of filtering, we conducted a preliminary user experience study. Following the result of the preliminary study, the system was adjusted to fit the findings of the preliminary study.

The modifications made were:

- to show the immediate results upon every command and refinement made
- system will ask at most three questions
- location, type, and price being identified as important factors to be considered when asking for refinement.

More detailed result of the preliminary study will be discussed in the later section.

3.2 User Experience Study

The user experience study was conducted in two stages. Participants from the first stage is not allowed to join the main stage. People with knowledge of the project are also not allowed to participate in either stage of study.

3.2.1 Preliminary Study

The first preliminary stage was developed to determine the proper depth of refinement. In other words, the preliminary study aimed to understand the number of questions that yields the optimal user experience and satisfaction of results. The preliminary study included an observation of the relevant filters that user prefers for the use case selected for this project.

Subjects in the preliminary study are provided with set of four tasks, with a set of follow up questions to be answered based on their experience of using the system for completing the task. The four questions assessed users' preferences in different simulated situations. The preliminary study case and questionnaire form is available on Appendix E.

Upon completion of the preliminary study, we modified the system to adjust for the result of the preliminary study. The details of system adjustment are described in the previous section.

3.2.2 Main Study

The main stage of study was aimed to assess the user experience, satisfaction of result, and task completion rate when being applied the single-stage implementation and multi-stage interaction approach. In order to compare between the two approaches, we used A/B testing method and have the subjects use our implementation to accomplish several tasks of similar difficulty and complexity level, and had them compare the four metrics described above between each task and the task immediately preceding it.

The main study comprised of a case with 6 tasks that needed to be completed. These tasks were related to finding places to eat or drink at some given location. Subjects were given a role of recent graduate who is working in a city in the U.S. In the experiment, subject commuted to work, school event, and visited his/her friend in another location in the U.S. Upon completion of each task, subject was requested to answer some questions with regard to the user experience, result satisfaction, and overall system performance. The main study case and questionnaire form is available on Appendix F.

3.3 Study Analysis

Following the study, the results were analyzed, and a conclusion was derived for the proposed approach and propose some suggestions for the current implementation of a voice assistant. The result obtained from both studies mentioned in the previous section and the analysis of the result are presented in the later section.

4. Testing

We divide this section into two parts as our project comprises of two components. Firstly, we discuss the result of the testing that was done to the system. Following that, we discuss the result of both the preliminary study and the main study of this project.

4.1 System Testing

During the development of the voice assistant system, unit testing was done to ensure that all modules were implemented correctly. System integration testing was conducted after components were completed and assembled into the system. For this purpose, we broke down testing into several tasks.

4.1.1 Testing query processing (server)

Firstly, we tested the server's capability of handling queries. This includes queries given as strings or as speech. Each query was analyzed using natural language processing modules. The results of the analysis was stored as query objects to be processed. In the processing phase, an external web service request was made with data extracted from the speech or the input string. Finally, the response from the external web service is displayed on the user interface. All different parts of the query processing implementation were tested, and the system successfully passed the query processing test.

4.1.2 Testing connectivity and message passing (client-server)

Secondly, we tested the communication between the server and the client, focusing on the server-client design pattern implementation. The client was able to successfully send a request to the server and the server is able to send a response back to the client.

4.1.3 Testing the user interface (client)

Thirdly, we tested the usability of our user interface. We first tested if all parts of the user interface were functional. Following that, we tested if the user interface could handle unwanted exceptions or breaks of usage flow well. The implemented parts of the user interface are functional and the user interface is able to handle unwanted exceptions or breaks of usage flow well. We did not measure the usability and user experience of the user interface, since this was done by our volunteer testers.

4.1.4 Testing the overall usage flow

Finally, we tested the overall usage flow from the moment when a user starts the application to when he or she receives results. We ensured that the system handles exceptional cases and work smoothly from the beginning until the end of the usage flow.

4.2 Experiment Results

In this section, we present the result of both preliminary and main study for this project. First, we present the information about the preliminary study and the result and findings of the preliminary study. The preliminary study was conducted to find out the appropriate level of depth for filtering and the number of questions that users believe will give the optimal user experience and result. The available filters were location, type of restaurant, price, distance, and whether or not to show only businesses with Yelp! deals.

4.2.1 Preliminary Study Results

The subjects were seven students with age ranging from 16-22 years old. Subjects comprised of Hong Kong nationality (29%), Mainland Chinese (28%), and other countries (43%). The first task simulated a condition where subject was required to find a place to eat, subject was allowed to choose any location for dinner, without time pressure or economical condition restriction. For this task, 85% of the subjects successfully accomplished the task, giving the system 7.14 out of 10 for its performance. 57% of subjects stopped working on the task due to satisfying results, 29% of subjects got annoyed by the series of questions, and the rest (14%) stopped because they have reached the final result page (i.e. no more filter can be applied). On average, subjects felt that the number of questions that would provide them the best experience was 3 and the number of questions that was necessary to give a satisfying results was 2. Subjects believed that location, type of restaurants and price would be the appropriate filters with priority according to the order.

The second task required subject to find a place to eat near a specified location, under time pressure, but no economical restriction. Simulating the time pressure, subject got to try only once. For this task, 85% of the subjects successfully accomplished the task, giving the system 8 out of 10 for its performance. 72% of subjects stopped working on the task due to satisfying results, 14% of subjects got annoyed by the series of questions, and the rest (14%) stopped because they have reached the final result page. On average, subjects felt that the number of questions that would provide them the best experience was 2 and the number of questions that was necessary to give a satisfying results was 2. Subjects believed that location, distance and type of restaurants or price would be the appropriate filters with priority according to the order.

The third task requested subject to find a place to eat without any specified condition, but under time pressure and economic condition (i.e. limited budget). For this task, 85% of the subjects successfully accomplished the task, giving the system 6.85 out of 10 for its performance. 43% of subjects stopped working on the task due to satisfying results, 29% of subjects got annoyed by the series of questions, and the rest (28%) stopped because they have reached the final result page. On average, subjects felt that the number of questions that would provide them the best experience was 3 and the number of questions that was necessary to give a satisfying results was 3. Subjects believed that price, type of restaurant and distance would be the appropriate filters with priority according to the order.

The last task of the preliminary study was designed to discover the appropriate filters and number of questions by eliminating the extraneous variable that might have come from different ways of formulating the initial command. For instance, some subjects might have asked “Where can I find cheap restaurants in San Francisco”, while some others might have said “Find me a cheap restaurant”. The two examples show the different in initial depth of the command, with the former has higher depth as it incorporates the location while the latter does not. For this task, subjects were required to find a place to eat by saying “Find me a restaurant”. The task yielded 85% accomplishment rate with averaged overall performance of 7.71 out of 10, with 85% of termination resulted of user satisfaction of the result. On average, subjects felt that the number of questions that would provide them the best experience was 3 and the number of questions that was necessary to give a satisfying results was 3. Subjects believed that location, type of restaurants and price would be the appropriate filters with priority according to the order.

At the end of the experiment, subjects were asked about the performance of the system and number of questions to give them optimal user experience and results. The following table summarizes the finding from the preliminary study.

	Average across all tasks	**End-of-experiment Average**
Performance	7.43 / 10	7.57 / 10
No. of questions yielding best result	2.64	3.42
No. of questions yielding best user experience	3.32	2.86
Filters	Location, Type, Price	Location, Type, Price

Figure 21. Preliminary Study Result Summary

From the preliminary study, it was concluded that the number of questions that would optimize the result and user experience was 3. As shown on the table above, the filters that were important and useful were location, type, and price respectively. This was applied to the system by setting the maximum number of questions to 3 with filters available for specifying location, type, and price.

The preliminary study also yielded some interesting results that eventually was incorporated to the design of the system for the main study. Firstly, it was found that subjects prefer that the intermediate results were shown even when the system was still asking for further filtering, as 57% of subjects stated that it was better to have it shown rather than collapsed as implemented in the preliminary study. Secondly, it was discovered that subjects tend to accept a result set when they see businesses that they have been to, as 57% of subjects found a restaurant that they were familiar with in the result set and all of them accept the result set. Another interesting discovery was the fact that subjects generally want to be able to see the business page or get more detailed information about the business while still having the result page available.

4.2.2 Main Study Results

The main study compared the user satisfaction of the result, user experience, task accomplishment rate and perceived system performance when the different flows are being applied. The first flow which was the basis for comparison was the single-stage approach implemented on most available intelligent personal assistant. The second flow was the proposed multi-stage human-computer interaction that allowed for command refining. Note that the base flow and the proposed flow were being applied randomly and users did not get to know which flow they are currently in.

The subjects were twenty students (n=20) with age ranging from 17-27 years old. Subjects represented different nationalities such as Hong Kong (35%), Mainland China (20%), and other countries, including Indonesia, Czech Republic, Swiss, and India. Subject pool was comprised of 60% male and 40% female, with 60-40 proportion for their exposure to intelligent personal assistant, and 35% of the pool have visited the U.S.

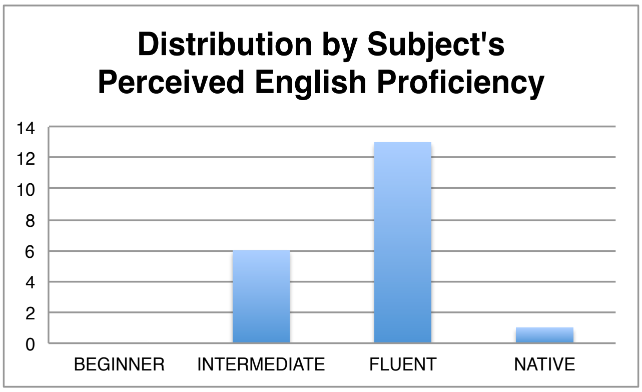


Figure 22. Main study distribution of English proficiency

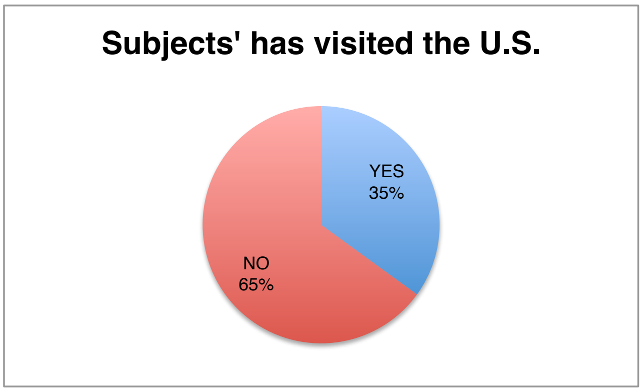


Figure 23. Main study distribution of subjects who have visited the U.S.

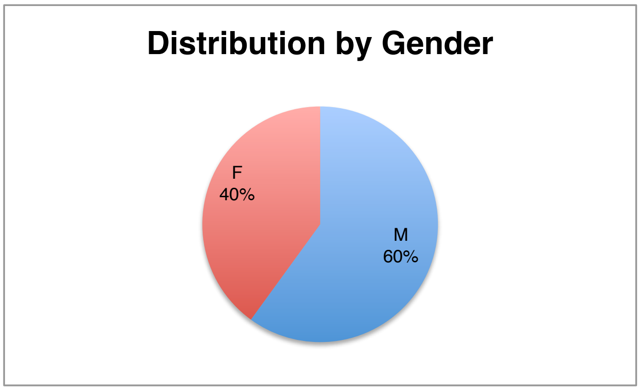


Figure 24. Main study distribution of participants' gender

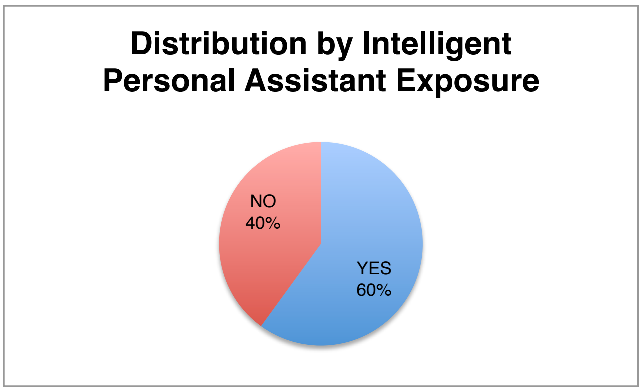


Figure 25. Main study distribution by participants' exposure to intelligent personal assistant system

Out of 120 tasks accomplished throughout the study, the single-stage implementation was being applied 44% of the time, and the multi-stage was randomly chosen 56% of the time.

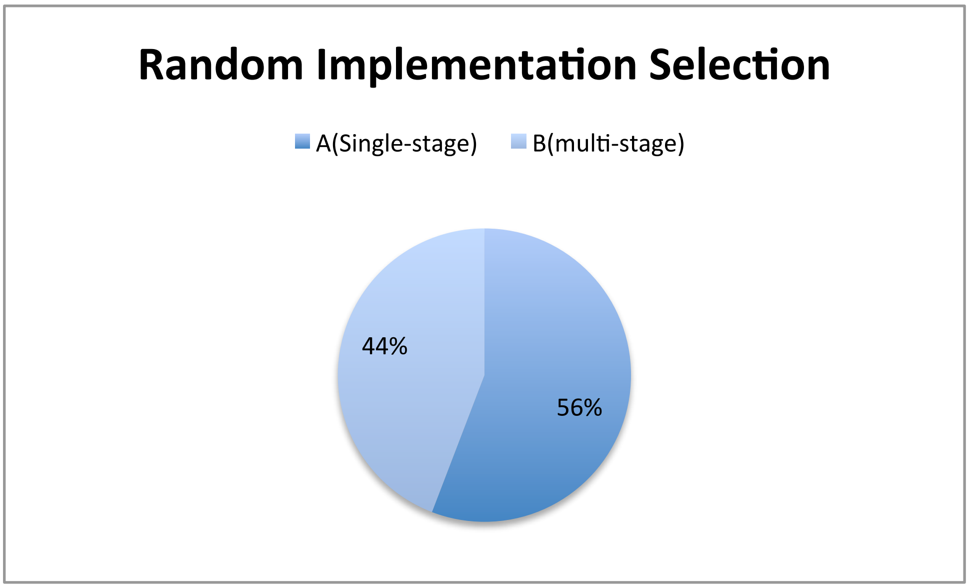


Figure 26. Main study implementation/approach selection result

We defined task accomplishment as whether or not subject was able to find the specific kind of place to eat at a particular location that complies with several filters/specification. The first task about finding a coffee shop in a city called Cupertino that is close to subject's office yields 25% task accomplishment rate. For the second task, subjects were being asked to find a Chinese food restaurant in a city called San Jose. 60% of subjects were able to accomplish the task. The third task was accomplished by 35% of subjects. The task was to find a specified mexican restaurant in a city called Merced. 60% of subjects were able to accomplish the fourth task. This rather simple task required subject to find a place to have a meal near the Empire State Building in New York. The fifth task was successfully accomplished by 55% of the pool. The task was to find a place to eat near the Rockefeller Center. Finally, the last task was accomplished by 80% of participants. The last task asked users to find a nice restaurant for dinner in Philadelphia.

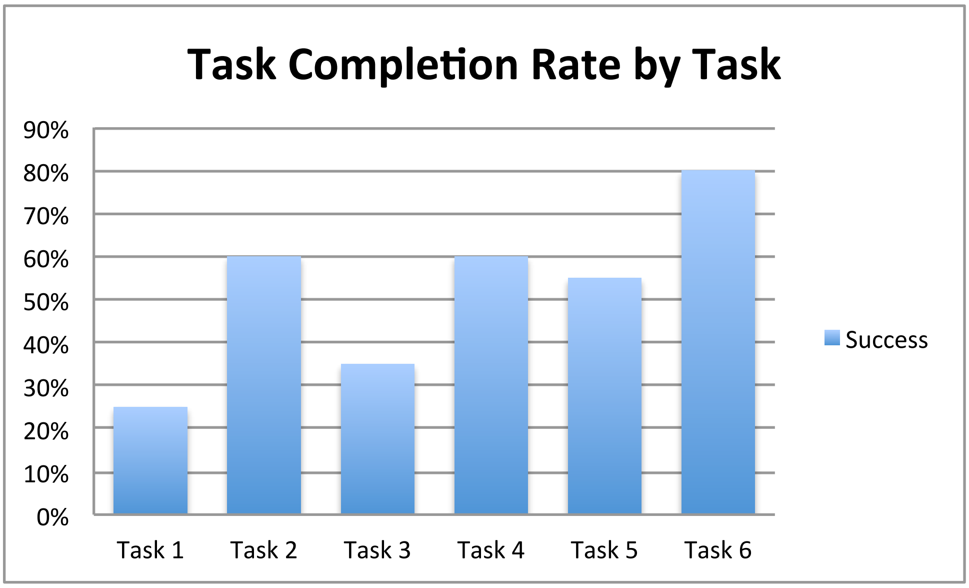


Figure 27. Main study task completion rate by task

Ultimately, with regard to the task accomplishment, it was discovered that the base implementation (single-stage human-computer interaction) yielded 37.31% task accomplishment rate. This was outperformed by the multi-stage human-computer interaction proposed which yielded 73.58% task accomplishment rate (36.27% increase in task accomplishment, 97.21% improvement compared to the base implementation)

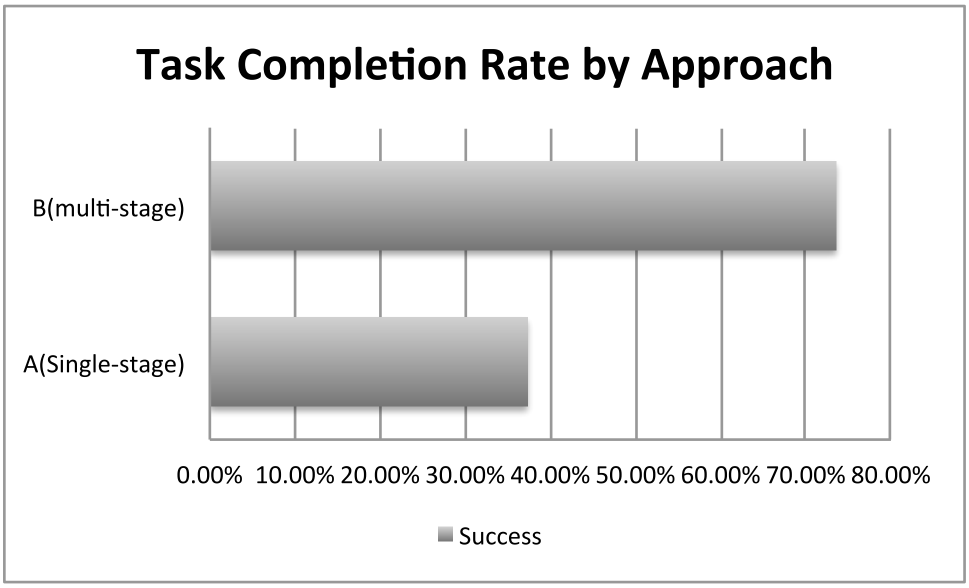


Figure 28. Main study task completion rate by selected implementation

The user satisfaction was defined as the degree of satisfaction of the result set presented the users get by using the system. The study showed that 53% of the time, subjects felt that they got a better satisfaction for the result set presented when they faced the multi-stage human-computer interaction immediately preceding a single-stage interaction approach or they got a worse satisfaction for the result set presented when they faced the single-stage human computer interaction immediately preceding a multi-stage interaction approach. For this 25% of the time, they felt that it gave them a worse satisfaction rate and the rest (22%) stated indifference between both approach.

The user experience was defined as the degree of good experience the users get from using our system. The study discovered that 51% of the time, subjects felt that they enjoy a better user experience for overall flow of the process when they faced the multi-stage human-computer interaction immediately preceding a single-stage interaction approach or they got a worse user experience for overall flow of the process when they faced the single-stage human computer interaction immediately preceding a multi-stage interaction approach. For this, 25% of the time, they felt that it gave them worse experience and the rest (24%) stated indifference between both approach.

The perceived system performance measured how well does the system perform according to the users. The study yielded that 44% of the time, subjects noticed that the performance of the system improved when they faced the multi-stage human-computer interaction immediately preceding a single-stage interaction approach or they noticed that the performance of the system gets worse when they faced the single-stage interaction immediately preceding a multi-stage interaction approach. For this 24% of the time, they feel that it gave them worse experience and the rest (33%) stated indifference between both approach.

From the end of study survey, 80% of subjects indicated that multi-stage interaction will increase their user satisfaction and 75% of them claimed that the interaction will help them to get a better result. Moreover, 85% of participants perceived that by having the intelligent personal assistant to ask some relevant questions, subjects will get a better user experience. Ultimately, 80% of subjects believed that the proposed approach yield a better overall system performance. The system obtained an average score of 7.9 out of 10 for system performance.

Moreover, end-of survey comments provided by the participants can be found in Appendix G.

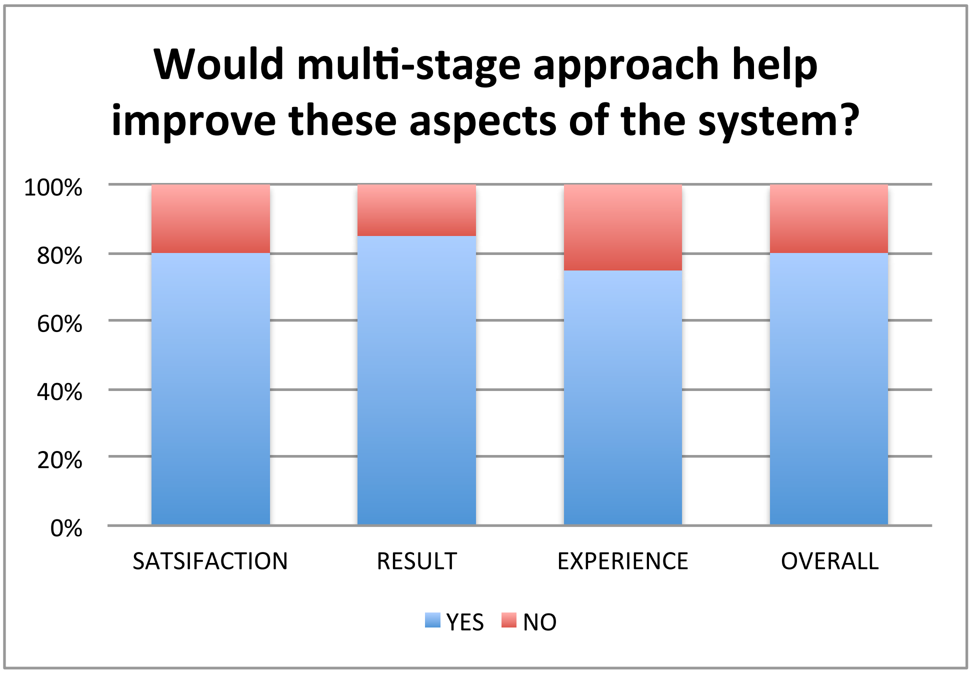


Figure 29. End-of-survey result summary

5. Evaluation

Evaluation was done not based on our testing, but through the user experience measurement/study. For all aspects listed below, random A/B testing was conducted. Further details on how the evaluation was conducted are described in the user experience study design (See Section 2.4).

5.1 Task Completion Rate

Task accomplishment/completion rate is whether or not subject was successfully able to find the specific kind of place to eat at a particular location that complies with several filters/specification. This aspect is very crucial as the main purpose of users in general when they are using an intelligent personal assistant is to be able to successfully complete the task without having to tap on multiple times and use different apps to do that. The main study shows a significant improvement of almost 100% increase in task completion rate compared to the base implementation (i.e, single-state interaction implementation). This is a very promising result as the adoption of voice-input and intelligent applications is growing very rapidly due to the shift in the computing devices trend from mobile to wearable.

5.2 System Response Time

Initially, we planned to record the system response time to measure the system performance. However, we figured out that this cannot be done properly as different subjects have different style of using an intelligent personal assistant. Specifically for multi-stage apporach, subjects were observed to have two ways of dealing with questions, the first way is to read the results first and answer the question, the second way is directly answer the question if perceived as something which is going to be useful.

5.3 User Satisfaction

For each task completed, user satisfaction was observed. Subjects were requested to rate their user satisfaction and answer a yes/no question regarding the user satisfaction. While the user satisfaction for each task does not give us any insights for comparing between the single-stage and multi-stage apporach, the combination of information and result about two tasks which adjacent to each other does. This combination provides us with implementation/approach chosen for a task and another task immediately preceding it. We were able to assess whether user prefers multi-stage over single-stage by analyzing these combinations.

Effectively, we obtained that 53% prefers multi-stage interaction for bringing them a better satisfaction of the result.

5.4 User Experience

For each task completed, users are requested to fill in a questionnaire with regard to user experience. The result was then processed in similar way with what is described in section 5.2. For this 51% of the time, subjects feel that multi-stage interaction approach gives them a better experience in using the system.

5.5 Perceived System Performance

For each task completed, users are required to provide is with an estimate of how they feel about the performance of the system. The result was then processed in similar way with what is described in section 5.2. For this 44% of the time, subjects perceived a better performance yielded when multi-stage approach is being selected/applied.

5.6 Overall System Performance

Upon the completion of six tasks, users are requested to indicate a score of the overall system performance from the range of 1 (very bad) to 10 (very good). The scores collected was averaged across all subjects. This yields a score of 7.9 out of 10 for overall system performance.

6. Discussion

In this section, we discuss some of the important results obtained from both the preliminary study and the main study part of this project. Moreover, we also discuss some interesting findings from both study and its implication for the design of the intelligent personal assistant human-computer interaction. Finally, we will provide some suggestions that could be useful for future improvement of intelligent personal assistant user experience design.

6.1 Preliminary Study Result

The preliminary study was conducted to find out the proper depth of filtering and the number of questions that would yield the optimal user experience and satisfaction of retrieved results. While this can be considered as an insignificant part of the main study itself, the preliminary study provided an isolated experiment that helped us get some information to support our main study. While the main study focused on comparing the user experience created by multi-stage human-computer interaction to the single-stage interaction used by most implementation of personal assistant available, it was crucial for us to understand how many questions that would yield the optimal user satisfaction of the experience in order to capture the optimal satisfaction yield by having multi-stage human-computer interaction approach. Failure to do so might bring unwanted and invalid results that were resulted from unoptimized implementation of the multi-stage human-computer interaction.

The preliminary study gave us a guidance for the design of multi-stage human-computer interaction. Moreover, this results yielded were relevant for the design of current intelligent personal assistant human-computer interaction. The result implied four important and useful suggestion for improvement:

- the multi-stage human-computer interaction was preferred when the questions were relevant.
- the intermediate result should be shown, as this gives user a choice whether to proceed with further refining or to stop at a particular filter.
- providing users with capability to look for more information about the restaurant while still giving them an option to go back to the result page might yield a better user experience.
- relevancy and ordering of filter options to the situations should also be considered. For example, during working hours, it might be better to have less number of questions and have filters for sorting by distance option asked first.

6.2 Main Study Result

The main study was conducted to compare the single-stage human-computer interaction on an intelligent personal assistant, the approach that was being used on most available intelligent personal assistant system, which also serves as our basis for comparison, to the proposed multi-stage human-computer interaction on an intelligent personal assistant to provide users with capability of refining their initial command with available filter options. We predicted that this approach will give users better user experience and satisfaction of result.

The study result showed that the proposed approach significantly increased the task accomplishment rate. The approach also yielded better user experience, satisfaction of result, and user perception of system performance compared to single-stage human-computer interaction. This implies that the proposed approach can be a good option when designing for intelligent personal assistant user experience. Having the personal assistant asks you questions for command refinement helps users accomplish the task. Moreover, the approach also created a better user experience if the number of questions and the questions asked were relevant. While further refinement might be necessary to bring an even better user experience and satisfaction of the result when multi-stage human-computer interaction was adopted, the result helped shed the light to a new direction or user experience design option.

With regard to the main study, apart from the results related to the defined evaluation metrics obtained, there are several other interesting insights that was gained from the experiment

6.2.1 Cold start

As mentioned in the previous section, the two tasks with the least task completion rate obtained was the first task and the third task. The low completion rate for the first task can possibly be explained by this cold start concept. The first task requires subjects to find a coffee shop near his/her office in Cupertino, California. For the first task, 71% of those who were able to complete the task have used an intelligent personal assistant implementation previously. This number represents 42% of subjects who have used an intelligent personal assistant. This is indeed much higher than the success rate when subject has never used an intelligent personal assistant previously, with only 25% of them was able to complete the task. Therefore, it is possible that the rest of participants who were not able to complete the task successfully because they have never used an intelligent personal assistant previously. A possible modification for the study will be proposed in the later section.

6.2.2 Location pronunciation and familiarity

An interesting finding gained from the study is when subjects are required to find a place to eat in a place which is relatively harder to pronounce for people who have never been to the U.S. Subject tends to mispronounce the location name, which leads to a failure in task completion. This happened to task 2 and 3. Task 2 requires subjects to find a place to eat Chinese food in San Jose. Task 3 asks subjects to find a Mexican restaurant called “J&R Tacos” in Merced.

For task 2, 100% of subjects who have been to the U.S. managed to successfully complete the task. In contrast, only 38% of subjects who have never been to the U.S managed to complete the task. This is consistent with the result for the other task mentioned as 43% of subjects who have ever been to the U.S. were also able to successfully complete task 3 and 30% of those who have not were able to complete the task successfully.

The task which yields lowest completion rate is also associated with the fact that subjects who have been to the U.S might not be familiar with a small town in Central Valley, California, called Merced.

In contrast, “Philadelphia”, which is a city mentioned in the last task is relatively more famous as it is the capital of the state of Pennsylvania. For this task, 71% of those who have been to the U.S were able to successfully complete the task. This number is relatively close the fact that 62% of those who have not been to the U.S were able to also successfully complete the task.

This phenomenon could best explain why the two tasks yield a low completion rate. A possible modification for the study will be proposed in the later section.

6.3 Limitations

In this section, we would like to present several limitations that we have identified throughout the period when we conducted the experiment. Recommendations for future improvements to address these limitations will be presented in the later section

6.3.1 Experiment context

As the experiment is being conducted in Hong Kong. It might be more appropriate for the study to be designed to fit tasks that can be done in Hong Kong location contexts. For example, finding a place to eat in Hong Kong rather than in the U.S.

However, the unavailability of data of businesses in Hong Kong that was open for public access forced us to use Yelp! API. Yelp! API provides a very robust developer services for finding places to eat. However, as mentioned above, only 30% of the subject pool has been to the U.S. Therefore, subjects might not know much about places to eat in the U.S. It also makes it harder for subjects to compare between results, as some subjects were not familiar with any places in the result set.

We have identified several ways to deal with this limitation, the recommendation is provided in the later section (section 7.2.1, 7.2.9 and 7.2.10)

6.3.2 Diversity of subject pool

In both studies that were conducted, the subject pool was not diverse enough in terms of their age and occupation. The subjects of the main study were all students, with age ranges from 16-27. This lack of variety of samples might affect the external validity of the experiment. It was possible that people from different age or occupation groups have different preference with regard to human-computer interaction design on an intelligent personal assistant.

6.3.3 Implementation of system for experimentation

The system implemented was rather simple compared to the implementation of Siri or other intelligent personal assistant available in the market. This was reasonable as those are artificial intelligence based system that would learn as it is being used by people. Our implementation did not have such capability. Moreover, compared to the language processing technology of the system, Siri outperformed our implementation of system used for the experiment. The effect of having this difference in performance was mitigated by implementing both approaches on our intelligent personal assistant instead of using Siri implementation as the basis for comparison.

6.3.4 Devices used for experimentation

Due to unavailability of Android-based phone or table in the beginning of main study , we had to conduct the study on a macbook air machine instead of an android devices. This could bring up an internal validity issue as the experience of using the system includes the device being used for accomplishing the task. Most intelligent personal assistant implementations are available on mobile devices. Our implementation was also designed to fit the mobile devices experience, but the experiment was conducted on a laptop machine for the first few participants.

However, this might not be significant as result shows that the studies that were being done on a laptop machine yields 58.33% task completion rate on average, whereas those that were conducted on a table yields 48.33% task completion rate on average.

6.3.5 Number of subjects

The number of subjects that we have conducted the experiments with was not big due to the time constrain. This possibly would have affected the averaged result of the studies as the result might have been prone to residual errors.

The next section will present the conclusion of the project and recommendations for future works to address the limitations of the project.

7. Conclusions and Recommendations for Future Work

7.1 Conclusions

The project proposed and investigated a novel approach to deliver a better result and user experience of intelligent personal assistant by applying multi-stage human-computer interaction that allows for command refining on an intelligent personal assistant. The project compared between the single-stage human-computer interaction which is adopted by most available implementation of an intelligent personal assistant to the proposed multi-stage human-computer interaction approach.

In order to conduct the experiment, a simple implementation of intelligent personal assistant was developed. The system incorporated A/B testing capability that would randomly choose between applying single-stage human-computer interaction or multi-stage human-computer interaction proposed when receiving a new command. This implementation utilized a set of logic and natural language processing tools to extract the important attributes of the command, which eventually were formulated to a specified format to be sent as a request to the external service provider, which in our case, was Yelp! API. Apart from that, the system also incorporated text-to-speech capability and voice input API provided by Google Chrome. The system user interface was built by using jQuery Mobile, to get the design optimized for mobile. All parts of the system were equipped with logging modules to record the user activity and information about processes being done for data analysis.

Throughout the project, we conducted two user experience studies. The preliminary study aimed to to find out the proper depth of filtering and the number of questions that would yield the optimal user experience and satisfaction of retrieved results. The main study compared the user satisfaction of the result, user experience, task accomplishment rate and perceived system performance yielded by applying either the single-stage human-computer interaction approach or the proposed multi-stage human-computer interaction approach. The preliminary study yielded three as the number of questions to be asked to yield the optimal user experience and satisfaction of the results, with location, type of restaurants, and price being the filters favored by subjects. The main study discovered 36.27% increase in task accomplishment rate, which is equivalent to 97.21% improvement when the multi-stage human-computer interaction was applied compared to single-stage. The study also found that the proposed approach yielded better user experience, satisfaction of result, and perceived performance of the system.

In the latter part of the report, we presented the limitation of the project as well as the plan for further work that can be done. It can be concluded that the proposed multi-stage human-computer interaction promises a better overall user experience and satisfaction to the user. It also helps users accomplish their task in a more natural way. Incorporation of the proposed approach could definitely be a choice that intelligent personal assistant user experience designers could consider to improve the user experience. Ultimately, further exploration needs to be done for understanding the advantages of using the proposed approach for voice-input based applications, especially the ones implemented on mobile and wearable devices.

7.2 Recommendations for Future Work

We understand that there exist some limitations in our project. We have listed them down in the previous section. On the top of that, there is also always a room for improvements. For further improvement on the project, we can aim to eliminate the limitations described above. Moreover, there have been some recommendations for improvements suggested by those who are interested in the project that could be useful when incorporated into the future works.

7.2.1 Addressing experiment context limitation

There are two approaches of tackling the problem. The first option is to conduct the experiment in the U.S. The second option is to get businesses data from Hong Kong based restaurant review companies. We predict that it is possible to partner up with Hong Kong based restaurant review providers such as OpenRice, or businesses and location data providers such as Kites. By slightly modifying the module of the system that interacts with the external service, we can set the system to send queries to their interface and parse the responses retrieved.

7.2.2 Dealing with external validity

Our experiment might carry some external validity issue. Further experiment should be conducted with subjects having different occupations representing different age groups. It is important to make the subject pool as diverse as possible to eliminate the external validity issue that might arise from the experiment.

7.2.3 System improvement

While it is not possible to fully replicate Siri capability, improvement on the system used for the experiment might be able to eliminate the problem. It is also preferable to incorporate artificial learning capability and scalable computational resources to speed up the process of accomplishing the task.

7.2.4 Devices for experimentation

It is preferable to conduct the experiment on the mobile device. This can be done by acquiring an android based mobile device for further experiments.

7.2.5 Sample size

The limitation of number of subjects can be solved by more participants for the study. This solution can be combined with the one which addresses the second limitation. In other words, it is favorable to have bigger subject pool with more diverse background and better distribution for each attribute of the subject.

7.2.6 Addressing the “Cold start” problem

We understand that not everyone has ever used intelligent personal assistant. It is preferable to still include representatives from that group in the experiment. In order to address the problem whereby subjects are not doing really well for the first task, experiment can include two or three free trials before subject starts to work on the tasks.

7.2.7 Location familiarity problem

As described in the previous section, the result for some tasks are affected by the name of the place and how familiar the subjects are to the location. It is possible but not preferable to have only samples that have been to the U.S. For solving this issue, it could be preferable to have locations which are more well-known to people. For example, using the capital city of well-known states or using the city with a lot of tourist attractions might be a better idea than using a small town.

7.2.8 Collaborative filtering

During the project exhibition at the President's Cup 2014 at HKUST, Dr. Kenneth Leung, one of the selection committee proposed a possible improvement using collaborative filtering, similar to information retrieval concept.

For this project, collaborative filtering can be applied to deciding the number of relevant questions to be asked. In this study, we conducted an experiment to determine how many questions and what are the relevant questions to be asked. Collaborative filtering approach would take this fact and analyzes users' usage pattern. If most users stop at the first or second question, then for the new request, we present only one or two questions instead of keeping three questions to be asked based on the preliminary study. This approach also works for extending the number of questions. If users always answer all the questions, then it could be possible that we need to increase the number of questions.

Another possible application of this concept for the project is collaborative filtering on the result, which is a more natural concept, understanding the fact that essentially what the nature of the task is, which is searching, an area of information retrieval. It could be possible that users can choose several restaurants for each stage, and the record will be stored as our internal ranking array. By doing this, we can provide a better order of showing the restaurants, based on the ones that most users select, rather than the ranking given by Yelp! Search API.

7.2.9 Using businesses in Hong Kong

During the system implementation, we described our difficulties of finding businesses in Hong Kong, as OpenRice [14] does not provide a publicly available API that we can access. Recently, it is discovered that there is a company which was just starting up. Kites [38] provides information about businesses in Hong Kong. They provide a publicly available API similar to Yelp! API but for restaurants and other places to eat or drink in Hong Kong. These businesses are categorized by type and district. By partnering up with this firm, we can use their information to provide a better location context if the study is to be conducted in Hong Kong. However, in order to support this, the system will need a speech recognizer module and a named entity recognizer that at least understand English and Romanized Cantonese (for example: Tsim Sha Tsui).

7.2.10 Multi-lingual speech recognition and named entity recognizer

In rhyme with the approach to deal with experimental context limitation, and supporting what is described in section 7.2.1 and 7.2.9. It could be beneficial for us to have a multi-lingual speech recognition and named entity recognizer modules. The reason for this is English speech recognition will fail to recognize some local romanized spelling of Cantonese words. Named entity recognizer module that supports this will also be needed as system will need to recognize Cantonese words as location (for example: Choi Hung) or type of restaurant (for example: Cha Can Teng).

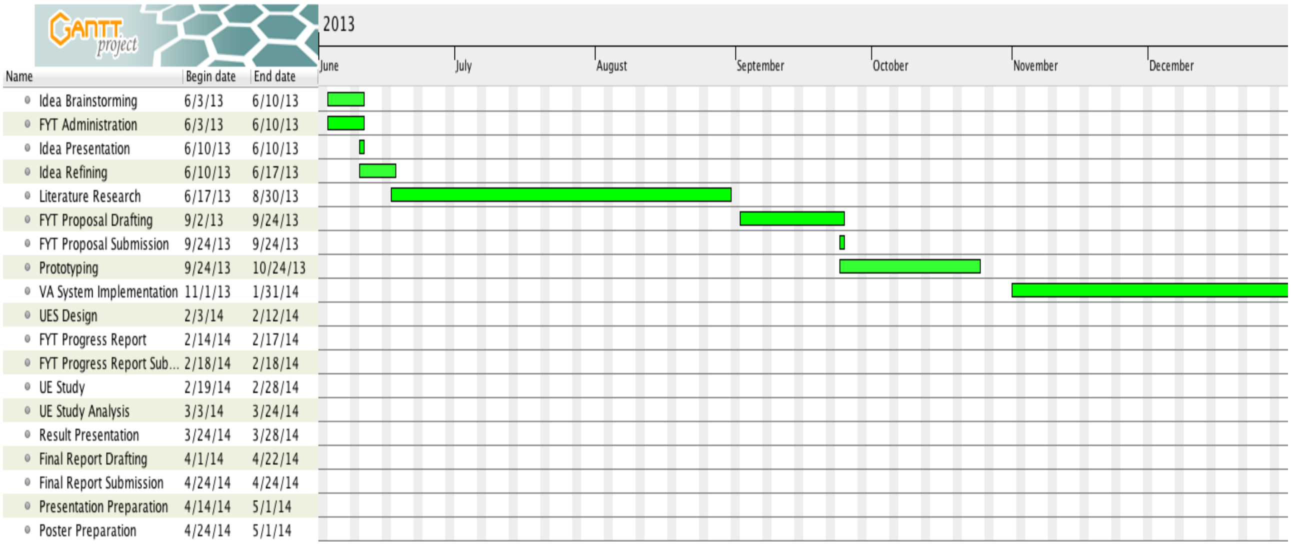
A more general suggestion that was suggested by one of the subjects (Qianlong) was to adjust the system to be available in Cantonese and Mandarin Chinese.

This is reasonable as subjects comprised of rather big population of local students (35%), which first language is Cantonese.

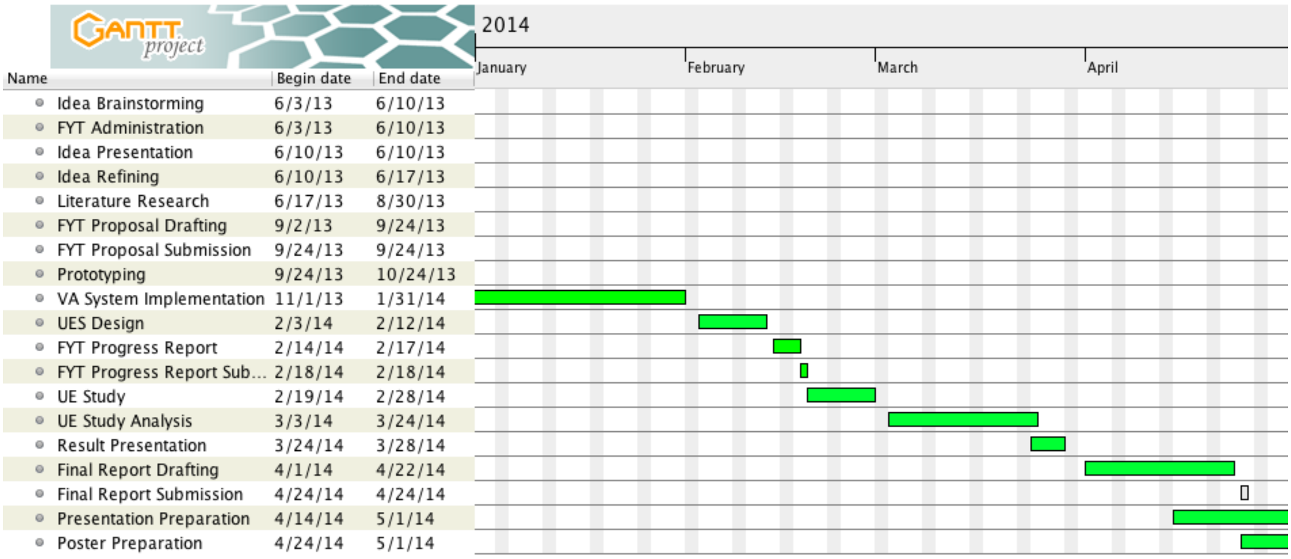
8. Project Planning

The plan for the of the project can be seen on the following page, which contains a GANTT chart showing the timeline of the project and tasks to be completed for each period of time.

With regard to resource allocation, since this project was an individual thesis project, all work was done by the author. However, feedback and support were obtained from the project supervisor, Dr. David Rossiter.



Name	Begin date	End date
Idea Brainstorming	6/3/13	6/10/13
FYT Administration	6/3/13	6/10/13
Idea Presentation	6/10/13	6/10/13
Idea Refining	6/10/13	6/17/13
Literature Research	6/17/13	8/30/13
FYT Proposal Drafting	9/2/13	9/24/13
FYT Proposal Submission	9/24/13	9/24/13
Prototyping	9/24/13	10/24/13
VA System Implementation	11/1/13	1/31/14
UES Design	2/3/14	2/12/14
FYT Progress Report	2/14/14	2/17/14
FYT Progress Report Sub…	2/18/14	2/18/14
UE Study	2/19/14	2/28/14
UE Study Analysis	3/3/14	3/24/14
Result Presentation	3/24/14	3/28/14
Final Report Drafting	4/1/14	4/22/14
Final Report Submission	4/24/14	4/24/14
Presentation Preparation	4/14/14	5/1/14
Poster Preparation	4/24/14	5/1/14



Name	Begin date	End date
Idea Brainstorming	6/3/13	6/10/13
FYT Administration	6/3/13	6/10/13
Idea Presentation	6/10/13	6/10/13
Idea Refining	6/10/13	6/17/13
Literature Research	6/17/13	8/30/13
FYT Proposal Drafting	9/2/13	9/24/13
FYT Proposal Submission	9/24/13	9/24/13
Prototyping	9/24/13	10/24/13
VA System Implementation	11/1/13	1/31/14
UES Design	2/3/14	2/12/14
FYT Progress Report	2/14/14	2/17/14
FYT Progress Report Sub…	2/18/14	2/18/14
UE Study	2/19/14	2/28/14
UE Study Analysis	3/3/14	3/24/14
Result Presentation	3/24/14	3/28/14
Final Report Drafting	4/1/14	4/22/14
Final Report Submission	4/24/14	4/24/14
Presentation Preparation	4/14/14	5/1/14
Poster Preparation	4/24/14	5/1/14

Figure 21. Gantt Chart of the Project

9. Required Hardware and Software

Development was done on a Macbook Air with a 1.8 GHz Intel Core i5 processor and 8 GB of 1600 MHz DDR3 memory. This machine also included a 128GB Solid State Drive running Mac OS X version 10.8.4. The server was hosted on the same Macbook Air machine. The server software used is mentioned in the chart below.

We used a mobile device to conduct the user experience experiments. The mobile devices used was an Android mobile phone and tablet running Android 4.0. This was to make sure that Google Chrome for Android is supported. This mobile device was equipped with GPS and a microphone (for voice input) as well as a built-in speaker (for speech output).

For project development and the user experience study, we used the following software:

Software Name	Purpose	Remarks
XCode	Prototyping	Free with Mac OSX Installation
Mac OS X	Operating System	Comes with Mac OSX
iOS 7	Operating System	Free, comes with iPhone 4S
Android 4.0	Operating System	Free, comes with an Android phone
Google Chrome for Android	WebKit based Browser	Free
Google Chrome	WebKit based Browser	Free
Apple Safari	WebKit based Browser	Free

Apple Safari for Mobile	WebKit based Browser for the Client Part	Comes with the iOS 6 installation
Java SDK	Voice Assistant System Development	Free with Mac OSX Installation
Eclipse IDE with Java	IDE, Text Editor	Open Source
Spring Framework with JAX-RS and/or Jersey	REST Web Service development	Open Source
Apache Tomcat Server	Web server	Open Source
Apache Web Server	Web Server	Open Source
Stanford NER	Named Entity Recognition	Open Source
Stanford POS Tagger	Part of Speech Tagging	Open Source
JQuery/JQuery Mobile	User Interface Design	Free
Adobe Photoshop	User Interface Design	Purchased
Astah Community	Project Documentation	Free with Student License
GanttProject	Project Documentation	Open Source
HTML Web Speech API	User Interface	Free with Google Chrome Installation
Maven	Dependencies Management	Open Source

This page is intentionally left blank

10. References

[1] Wikipedia. (2013). *Intelligent Personal Assistant.* [Online]. Available: http://en.wikipedia.org/wiki/Intelligent_personal_assistant.

[2] Wikipedia. (2013). *Siri.* [Online]. Available: http://en.wikipedia.org/wiki/Siri.

[3] Wikipedia. (2013). *Google Now.* [Online]. Available: http://en.wikipedia.org/wiki/Google_Now.

[4] Bloomberg. (2012). *Apple's Siri Gets Below-Average Grade From Piper Jaffray.* [Online]. Available: http://www.bloomberg.com/news/2012-06-29/apple-s-siri-gets-below-average-grade-from-piper-jaffray.html.

[5] T. Starner. "The Role of Speech Input in Wearable Computing". *IEEE Pervasive Computing*, vol. 1(3), pp. 89-93, July 2002.

[6] R. Kohavi, R. M. Henne, and D. Sommereld. "Practical Guide to Controlled Experiments on The Web: Listen to Your customers not to the Hippo," In *KDD '07: Proceedings of the 13th ACM International Conference on Knowledge Discovery and Data Mining,* New York, NY, USA, 2007, pp. 959-967.

[7] S. Sampath and B. Bringert. (2010). *Speech Input API Specification* [Online]. Available: http://lists.w3.org/Archives/Public/public-xg-htmlspeech/2011Feb/att-0020/api-draft.html.

[8] R. T. Fielding. "Architectural styles and the design of network-based software architectures." *PhD Thesis*, University of California, Irvine, 2000.

[9] G. Shires and H. Wennborg. (2010). *Web Speech API Specification* [Online]. Available: https://dvcs.w3.org/hg/speech-api/raw-file/tip/speechapi.html.

[10] T. Tullis and W. Albert. *Measuring The User Experience*. Burlington, MA: Morgan Kaufmann. 2008.

[11] D. Agushinta, B. Lolita, D. A. Setianti, H. Wahyudi, and I. P. Partadiyasa. "Comparative Study of Intelligent Personal Assistant". *International Journal of Engineering and Innovative Technology,* vol. 1, pp. 154-160, Jun. 2012.

[12] Apple. (2013). *Siri.* [Online]. Available: http://www.apple.com/ios/siri/.

[13] Google. (2013). *Google Now. The right information at just the right time.* [Online]. Available: http://www.google.com/landing/now.

[14] OpenRice. (2013). *About Us.* [Online]. Available: http://www.openrice.com/info/corporate/eng/about-us.html.

[15] Wikipedia. (2013). *Yelp, Inc..* [Online]. Available: http://en.wikipedia.org/wiki/Yelp,_Inc.

[16] Yelp. (2013). *Yelp.* [Online]. Available: http://www.yelp.com.

[17] Wikipedia. (2013). *WebKit.* [Online]. Available: http://en.wikipedia.org/wiki/WebKit.

[18] WebKit. (2013). *The WebKit Open Source Project.* [Online]. Available: http://www.webkit.org.

[19] Paul Irish. (2013). *WebKit for Developers*. [Online]. Available: http://www.paulirish.com/2013/webkit-for-developers/

[20] Google. (2013). *Web Speech API Demonstration.* [Online]. Available: https://www.google.com/intl/en/chrome/demos/speech.html

[21] Google. (2013). *Chrome Browser - Google.* [Online]. Available: https://www.google.com/intl/en/chrome/browser/

[22] Apache Tomcat. (2014). *Introduction to SSL.* [Online]. Available: http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html#Introduction_to_SSL

[23] Kryptotel. (2010). *SSL Protocol.* [Online]. Available: http://en.kryptotel.net/ssl.html

[24] D. Nadeau and S. Sekine. “A survey of named entity recognition and classification”. *Linguisticae Investigationes*, vol. 30, pp. 3-26, 2007.

[25] R. Grishman and B.Sundheim, “Message Understanding Conference - 6: A Brief History,” *In Proceedings of International Conference on Computational Linguistic*, 1996.

[26] Y. Li, H. Lin, and Z. Yang, “Incorporating Rich Background Knowledge for Gene Named Entity Classification and Recognition,” *BMC Bioinformatics*, vol. 10, pp. 223, 2009.

[27] K. Toutanova and C. D. Manning. "Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger," *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000),* 2000*,* pp. 63-70.

[28] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, O’Reilly Media, 2009.

[29] Apache Maven. (2014). *Welcome to Apache Maven.* [Online]. Available: http://maven.apache.org/

[30] T. Mishra and S. Bangalore, "Speech-driven query retrieval for question-answering, " *In Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 14-19 March 2010, pp.5318-5321.

[31] K. Vertanen and P. O. Kristensson, “Recognition and Correction of Voice Web Search Queries,” *Interspeech ’09: Proceedings of the International Conference on Spoken Language Processing*, 2009, pp. 1863-1866.

[32] A. Crystal and B. Ellington, “Task analysis and human-computer interaction: approaches, techniques, and levels of analysis,” *In Proceedings of the Tenth Americas Conference on Information Systems*, 2004.

[33] X. Ma. CSE Seminar. Topic: “Leveraging Crowds to Bridge Communication Barriers.” HKUST, September 30th, 2013.

[34] Wikipedia. (2013). *Representational state transfer.* [Online]. Available: http://en.wikipedia.org/wiki/Representational_state_transfer

[35] J. R. Finkel, T. Grenager, and C.Manning, “Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling,” In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 363-370, 2005.

[36] Google. (2013). *Inside Search.* [Online]. Available: http://www.google.com/insidesearch/features/voicesearch/index-chrome.html

[37] Google. (2013). *Google Translate.* [Online]. Available: http://translate.google.com/

[38] Kites. (2014). *Kites.* [Online]. Available: http://www.kites.hk.

11. Appendix A: Meeting Minutes

11.1 Minutes of the 1st Project Meeting

Date	: June 10th, 2013
Time	: 12:00 pm
Place	: Dr. David Rossiter's office
Present	: Ivan Gondoprastowo, Dr. David Rossiter
Absent	: None
Recorder	: Ivan

1. Approval of minutes

This was the first formal group meeting, so there were no minutes to approve.

2. Report on Progress

2.1 Ivan presented the proposed idea verbally

2.2 The initial idea has been sent as a document write-up

3. Discussion Items

3.1 The initial idea was too simple, so Dr. Rossiter suggested that we conduct a user experience study on something applicable to next generation computers: wearable computers.

3.2 We will try to discover other possibilities, while keeping the initial idea as fallback plan if the wearable computers user experience study does not work.

3.3 Ivan to refine the idea, but keep the idea of user experience study, and send before Sunday the 16th.

4. Goals for the next meeting

4.1 Come up with a refined idea that implementation wise is more challenging.

4.2 Come up with an idea that could be used for wearable computers user experience design decision.

4.3 Do literature review on ideas.

4.4 Learn about user experience and user interface design through Coursera.

5. Meeting adjournment and next meeting

The meeting was adjourned at 1pm.

The next meeting will be scheduled by email correspondence. Updates shall be sent through email.

11.2 Minutes of the 2nd Project Meeting

Date	: September 5th, 2013
Time	: 12:30 pm
Place	: Dr. David Rossiter's office
Present	: Ivan Gondoprastowo, Dr. David Rossiter
Absent	: None
Recorder	: Ivan

1. Approval of minutes

Minutes of first meeting were approved

2. Report on Progress

2.1 Ivan verbally presented the new proposed idea on “two-way interaction for voice assistants”

2.2 The idea had been sent as a document write-up, and had been approved by supervisor.

2.3 Ivan has done a literature survey on the topic.

2.4 Ivan has done the human-interaction course on Coursera.

3. Discussion Items

3.1 Limiting to merely find restaurant information will be necessary, since a full-fledged voice assistant implementation cannot possibly be developed in one year by one person.

3.2 Implementation of the system and the flow of the system.

3.3 When conducting a user experience study, certain questions need to be asked.

4. Goals for the next meeting

4.1 Come up with a draft of FYT proposal

4.2 If possible, build a small prototype of the idea using HTML and JavaScript

5. Meeting adjournment and next meeting

The meeting was adjourned at 1pm.

The following meetings will be on every Friday at 3pm.

11.3 Minutes of the 3rd Project Meeting

Date : September 13th, 2013

Time : 12:00 pm

Place : Dr. David Rossiter's office

Present : Ivan Gondoprastowo, Dr. David Rossiter

Absent : None

Recorder : Ivan

1. Approval of minutes

Minutes of the second meeting were approved

2. Report on Progress

2.1 Ivan has sent the proposal draft by email.

2.2 The prototype has not been completed yet.

3. Discussion Items

3.1 Regarding the layout, the figures should not be too big a caption should be included

3.2 Some parts need to be amended

4. Goals for the next meeting

4.1 Get a communication tutor to review the proposal

4.2 Submit the proposal on time

4.3 Build a small proof-of-concept using HTML and JavaScript

5. Meeting adjournment and next meeting

The meeting was adjourned at 12.30pm.

The next meeting is scheduled on September 27th. Discussion of the proposal draft will be done via email.

11.4 Minutes of the 4th Project Meeting

Date : October 18th, 2013

Time : 12:00 pm

Place : Dr. David Rossiter's office

Present : Ivan Gondoprastowo, Dr. David Rossiter

Absent : None

Recorder : Ivan

1. Approval of minutes

Minutes of the third meeting were approved

2. Report on Progress

2.1 Ivan has completed the proposal.

2.2 No prototype will be built, instead, start working on the actual system

2.3 UI has been implemented with voice recognition

2.4 REST Jersey has been implemented

2.5 Integrating NER tool with the server

3. Discussion Items

3.1 Progress is satisfying

3.2 Further discussion is to be done after the system implementation is completed

4. Goals for the next meeting

4.1 To complete the back-end system implementation

5. Meeting adjournment and next meeting

The meeting was adjourned at 12.30pm.

The next meeting is to be scheduled via email.

11.5 Minutes of the 5th Project Meeting

Date	: November 29th, 2013
Time	: 12:00 pm
Place	: Dr. David Rossiter's office
Present	: Ivan Gondoprastowo, Dr. David Rossiter
Absent	: None
Recorder	: Ivan

1. Approval of minutes

Minutes of the fourth meeting were approved

2. Report on Progress

2.1 The overall front-to-back flow has been completed

2.2 Functionalities are yet to be implemeted

2.3 Server-client communication has been established

2.4 Yelp! API search has been implemented

2.5 Front-end part has been implemented

2.6 Named Entity Recognition module has been implemented

2.5 Implementation without command refining has been completed

3. Discussion Items

3.1 Rearrangement of project timeline due to final exam

4. Goals for the next meeting

4.1 Complete the system implementation (A/B implementation)

5. Meeting adjournment and next meeting

The meeting was adjourned at 12.30pm.

The next meeting will be scheduled via email, during winter break.

11.6 Minutes of the 6th Project Meeting

Date	: January 16th, 2014
Time	: 3:00 pm
Place	: Dr. David Rossiter's office
Present	: Ivan Gondoprastowo, Dr. David Rossiter
Absent	: None
Recorder	: Ivan

1. Approval of minutes

Minutes of the fifth meeting were approved

2. Report on Progress

2.1 Implementation A and B completed

2.2 UI redesigned to fit mobile devices as well as laptop

2.3 Still need to re-allow program to use microphone

3. Discussion Items

3.1 Possibility of using google glass: make a google glass like device

3.2 Heuristics employed

3.3 A little study to figure out the proper depth of filter

3.4 Possibility of using Hong Kong restaurants to make the context easier

3.5 Design of user experience study

4. Goals for the next meeting

4.1 Complete the smaller study to figure out how deep the filtering should be

4.2 Start working on user experience study

5. Meeting adjournment and next meeting

The meeting was adjourned at 12.30pm.

The next meeting is to be scheduled via email.

11.7 Minutes of the 7th Project Meeting

Date	: February 26th, 2014
Time	: 2 pm
Place	: Rm. 3530
Present	: Ivan Gondoprastowo, Dr. David Rossiter
Absent	: None
Recorder	: Ivan

1. Approval of minutes

Minutes of the sixth meeting were approved

2. Report on Progress

2.1 Demo of system to be used for preliminary study

3. Discussion Items

3.1 System is not working well

3.2 Fail safe method needs to be implemented (adding extra button)

3.3 Some literals in the user interface need to be adjusted to create a more natural interface

3.4 Filters to be hidden and be displayed in a nicer way

3.5 Ordering of the elements in UI to be adjusted

4. Goals for the next meeting

4.1 Problem with the systems to be fixed

4.2 Changes to be applied

5. Meeting adjournment and next meeting

The meeting was adjourned at 3pm.

The next meeting is to be scheduled via email.

11.8 Minutes of the 8th Project Meeting

Date : February 28th, 2014

Time : 2 pm

Place : Rm. 3530

Present : Ivan Gondoprastowo, Dr. David Rossiter

Absent : None

Recorder : Ivan

1. Approval of minutes

Minutes of the seventh meeting were approved

2. Report on Progress

2.1 Demo of system to be used for preliminary study

2.2 Changes requested during the last meeting

3. Discussion Items

3.1 System is ready

3.2 Necessity of user interface logging

3.3 Discussion on preliminary study design

4. Goals for the next meeting

4.1 Preliminary study tasks needs to be adjusted

(this includes: gender bias, change in command, additional country (UK))

4.2 Logging on user interface should have been implemented

5. Meeting adjournment and next meeting

The meeting was adjourned at 2.30 pm.

The next meeting is to be scheduled via email.

11.9 Minutes of the 9th Project Meeting

Date	: March 12nd, 2014
Time	: 2.30 pm
Place	: Rm. 3530
Present	: Ivan Gondoprastowo, Dr. David Rossiter
Absent	: None
Recorder	: Ivan

1. Approval of minutes

Minutes of the eighth meeting were approved

2. Report on Progress

2.1 Demo of adjustments made for the preliminary study

2.2 Demo of logging

3. Discussion Items

3.1 Overall system for preliminary experiment is ready

4. Goals for the next meeting

4.1 Finished preliminary experiment

5. Meeting adjournment and next meeting

The meeting was adjourned at 3 pm.

The next meeting is to be scheduled via email.

11.10 Minutes of the 10th Project Meeting

Date	: March 19nd, 2014
Time	: 2.00 pm
Place	: Dr. Rossiter's office
Present	: Ivan Gondoprastowo, Dr. David Rossiter
Absent	: None
Recorder	: Ivan

1. Approval of minutes

Minutes of the ninth meeting were approved

2. Report on Progress

2.1 Discussion on preliminary study progress

2.2 Discussion on main study design

2.3 Demo of adjustments made for main study

3. Discussion Items

3.1 Overall system is ready

3.2 Minor adjustments need to be made in the questionnaire

4. Goals for the next meeting

4.1 Finished some rounds of main study experiment

5. Meeting adjournment and next meeting

The meeting was adjourned at 3 pm. The next meeting is to be scheduled via email.

11.11 Minutes of the 11th Project Meeting

Date	: March 24th, 2014
Time	: 1.30 pm
Place	: Dr. Rossiter's office
Present	: Ivan Gondoprastowo, Dr. David Rossiter
Absent	: None
Recorder	: Ivan

1. Approval of minutes

Minutes of the tenth meeting were approved

2. Report on Progress

2.1 2014 President's Cup Final Report Submission

2.2 Intermediate result of the main study

3. Discussion Items

3.1 Discussion on 2014 President's Cup Final Report Submission

3.2 Discussion on intermediate result of the main study

4. Goals for the next meeting

4.1 Get 20 participants for the main study

4.2 Get the result of the main study ready

5. Meeting adjournment and next meeting

The meeting was adjourned at 3 pm. The next meeting is to be scheduled via email.

12. Appendix B: Yelp! Search API Supported Categories

Type			Filter Keyword
Restaurants			restaurants
	Afghan		afghani
	African		african
		Senegalese	senegalese
		South African	southafrican
	American		New
	American		Traditional
	Arabian		arabian
	Argentine		argentine
	Armenian		armenian
	Asian Fusion		asianfusion
	Australian		australian
	Austrian		austrian
	Bangladeshi		bangladeshi
	Barbeque		bbq
	Basque		basque
	Belgian		belgian
	Brasseries		brasseries
	Brazilian		brazilian
	Breakfast & Brunch		breakfast_brunch
	British		british
	Buffets		buffets
	Burgers		burgers
	Burmese		burmese
	Cafes		cafes
	Cafeteria		cafeteria
	Cajun/Creole		cajun
	Cambodian		cambodian
	Caribbean		caribbean
		Dominican	dominican
		Haitian	haitian

Type			Filter Keyword
		Puerto Rican	puertorican
		Trinidadian	trinidadian
	Catalan		catalan
	Cheesesteaks		cheesesteaks
	Chicken Wings		chicken_wings
	Chinese		chinese
		Cantonese	cantonese
		Dim Sum	dimsum
		Shanghainese	shanghainese
		Szechuan	szechuan
	Comfort Food		comfortfood
	Creperies		creperies
	Cuban		cuban
	Czech		czech
	Delis		delis
	Diners		diners
	Ethiopian		ethiopian
	Fast Food		hotdogs
	Filipino		filipino
	Fish & Chips		fishnchips
	Fondue		fondue
	Food Court		food_court
	Food Stands		foodstands
	French		french
	Gastropubs		gastropubs
	German		german
	Gluten-Free		gluten_free
	Greek		greek
	Halal		halal
	Hawaiian		hawaiian
	Himalayan/Nepalese		himalayan
	Hot Dogs		hotdog
	Hot Pot		hotpot

Type			Filter Keyword
	Hungarian		hungarian
	Iberian		iberian
	Indian		indpak
	Indonesian		indonesian
	Irish		irish
	Italian		italian
	Japanese		japanese
	Korean		korean
	Kosher		kosher
	Laotian		laotian
	Latin American		latin
		Colombian	colombian
		Salvadoran	salvadoran
		Venezuelan	venezuelan
	Live/Raw Food		raw_food
	Malaysian		malaysian
	Mediterranean		mediterranean
		Falafel	falafel
	Mexican		mexican
	Middle Eastern		mideastern
		Egyptian	egyptian
		Lebanese	lebanese
	Modern European		modern_european
	Mongolian		mongolian
	Moroccan		moroccan
	Pakistani		pakistani
	Persian/Iranian		persian
	Peruvian		peruvian
	Pizza		pizza
	Polish		polish
	Portuguese		portuguese
	Russian		russian
	Salad		salad

Type			**Filter Keyword**
	Sandwiches		sandwiches
	Scandinavian		scandinavian
	Scottish		scottish
	Seafood		seafood
	Singaporean		singaporean
	Slovakian		slovakian
	Soul Food		soulfood
	Soup		soup
	Southern		southern
	Spanish		spanish
	Steakhouses		steak
	Sushi Bars		sushi
	Taiwanese		taiwanese
	Tapas Bars		tapas
	Tapas/Small Plates		tapasmallplates
	Tex-Mex		tex-mex
	Thai		thai
	Turkish		turkish
	Ukrainian		ukrainian
	Vegan		vegan
	Vegetarian		vegetarian
	Vietnamese		vietnamese

13. Appendix C: Logging Properties (Log4J XML)

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

  <appender name="console" class="org.apache.log4j.ConsoleAppender">
    <param name="Target" value="System.out"/>
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{[dd MMM yyyy HH:mm:ss]}
%-5p %c{1} - %m%n"/>
    </layout>
  </appender>

  <appender name="fileAppender"
class="org.apache.log4j.RollingFileAppender">
      <param name="append" value="false"/>
      <param name="file" value="/Users/gondoprastowo/Documents/workspace/
givan_ug_fyt/log/thesis.log"/>
      <param name="threshold" value="info" />
      <layout class="org.apache.log4j.PatternLayout">
         <param name="ConversionPattern" value="%d{[dd MMM yyyy HH:mm:ss]}
%d{ABSOLUTE} %-5p [%c{1}] %m%n"/>
      </layout>
   </appender>

  <root>
    <priority value ="debug" />
    <appender-ref ref="console" />
    <appender-ref ref="fileAppender"/>
  </root>

</log4j:configuration>
```

14. Appendix D: Maven Project Build Dependencies

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://
maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>hkust.</groupId>
  <artifactId>givan_ug_fyt</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>givan_ug_fyt Maven Webapp</name>
  <url>http://maven.apache.org</url>

  <repositories>
 <repository>
  <id>maven2-repository.java.net</id>
  <name>Java.net Repository for Maven</name>
  <url>http://download.java.net/maven/2/</url>
  <layout>default</layout>
 </repository>
</repositories>

  <dependencies>

   <dependency>
   <groupId>com.sun.jersey</groupId>
   <artifactId>jersey-server</artifactId>
   <version>1.8</version>
    </dependency>

    <dependency>
<groupId>com.googlecode.json-simple</groupId>
<artifactId>json-simple</artifactId>
<version>1.1</version>
</dependency>

    <dependency>
<groupId>org.scribe</groupId>
<artifactId>scribe</artifactId>
<version>1.1.0</version>
</dependency>

<dependency>
<groupId>log4j</groupId>
<artifactId>log4j</artifactId>
<version>1.2.15</version>
<exclusions>
            <exclusion>
                <groupId>com.sun.jmx</groupId>
                <artifactId>jmxri</artifactId>
            </exclusion>
            <exclusion>
                <groupId>com.sun.jdmk</groupId>
                <artifactId>jmxtools</artifactId>
            </exclusion>
            <exclusion>
                <groupId>javax.jms</groupId>
                <artifactId>jms</artifactId>
            </exclusion>
        </exclusions>
</dependency>

<dependency>
<groupId>dom4j</groupId>
<artifactId>dom4j</artifactId>
<version>1.6.1</version>
</dependency>

    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <finalName>givan_ug_fyt</finalName>
  </build>
</project>
```

15. Appendix E: Preliminary Study Form

RO3 - User Experience Study A Form for Subject

Name: ____________________________

Age: ___________

Nationality: __________________

Occupation: _____________________

Gender: M / F

Native Language: __________________

English Proficiency: Beginner / Intermediate / Fluent / Native

Have you ever used an intelligent personal assistant? YES / NO

Have you ever been to the US / UK? YES / NO

Thank you for participating in this user experience study. This study will assess your user experience and satisfaction when using a modified version of common intelligent personal assistant. Prior experience in using similar product is not needed. This study is going to take 5-10 mins.

How to work on each task:

1. Start with clicking the icon on the bookmark bar or button to go to start page.
2. Press the mic button, and wait until the animated mic icon is displayed.
3. Start giving your command verbally.
4. When you are done with the command, pause and wait for it to process, if it does not proceed to the next page, click Submit button.
5. Stop whenever you arrived at the final result page or satisfied with the result or annoyed answering the questions asked.
6. Write down your answer at the end-of-task question.
7. If you need any help or have any question, ask the experimenter.
8. You are allowed to restart the task at any point of time, unless specified, simply click the icon on the bookmark bar
9. Note: each task will take 1-2 mins to complete.

Case 1:

You are living in a big city in the US/UK. It is Saturday morning and you are relaxing at your home . You want to find a restaurant to go to, to have dinner with your partner tonight.

Use your intelligent personal assistant to help you find a place to eat. When you have finished, please answer the following questions.

Reason for termination	Final Result / Satisfied / Annoyed
Are you satisfied with the results?	
How many questions do you think is enough to get you to the set of results that satisfies you for this task?	

How many questions do you think the assistant should ask to give you a best experience in accomplishing this task?	
These are the features the assistant had: 1. Filter by Location 2. Filter by Type of restaurant 3. Filter by Price (Sorting by price) 4. Sorting by distance 5. Filter by showing businesses with deals only	What were the features you felt you need when using the assistant to accomplish this task? Ranks these features according to your preference, write only the features that you need: 1. 2. 3. 4. 5.
Do you feel that you have accomplished this task?	YES / NO
Do you feel that the system helps you in accomplishing this task?	YES / NO
Rate the performance of the system in accomplishing this task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good

Case 2: *(restart is not allowed)*

You are a trader (person who trades stock, generally earns a lot) at a huge investment bank. The market is extremely volatile today that you cannot afford to leave your desk for that long. Unfortunately, you just relocated to New York and you do not know any restaurant/place to eat around. Your office is located near the Empire State Building, and you decided that you are going to grab a sandwich at a coffee shop.

Use your intelligent personal assistant to help you find a place to eat. When you have finished, please answer the following questions.

Reason for termination	Final Result / Satisfied / Annoyed
Are you satisfied with the results?	
How many questions do you think is enough to get you to the set of results that satisfies you for this task?	
How many questions do you think the assistant should ask to give you a best experience in accomplishing this task?	
These are the features the assistant had: 1. Filter by Location 2. Filter by Type of restaurant 3. Filter by Price (Sorting by price) 4. Sorting by distance 5. Filter by showing businesses with deals only	What were the features you felt you need when using the assistant to accomplish this task? Ranks these features according to your preference, write only the features that you need: 1. 2. 3. 4. 5.
Do you feel that you have accomplished this task?	YES / NO

Do you feel that the system helps you in accomplishing this task?	YES / NO
Rate the performance of the system in accomplishing this task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good

Case 3:

You are living in a big city in the US/UK, and you are broke. Unfortunately, you have a girlfriend and you need to treat her to a dinner tonight because today is your 100th day relationship anniversary. You are in a hurry as you are on your way to your part-time job.

Use your intelligent personal assistant to help you find a place to eat. When you have finished, please answer the following questions.

Reason for termination	Final Result / Satisfied / Annoyed
Are you satisfied with the results?	
How many questions do you think is enough to get you to the set of results that satisfies you for this task?	
How many questions do you think the assistant should ask to give you a best experience in accomplishing this task?	
These are the features the assistant had: 1. Filter by Location 2. Filter by Type of restaurant 3. Filter by Price (Sorting by price) 4. Sorting by distance 5. Filter by showing businesses with deals only	What were the features you felt you need when using the assistant to accomplish this task? Ranks these features according to your preference, write only the features that you need: 1. 2. 3. 4. 5.
Do you feel that you have accomplished this task?	YES / NO
Do you feel that the system helps you in accomplishing this task?	YES / NO
Rate the performance of the system in accomplishing this task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good

Case 4:

You are living in a city in the US/UK. You chose to use your intelligent personal assistant to help you. You are to find a restaurant to have lunch this afternoon.

Use your intelligent personal assistant to help you find a place to eat. When you have finished, please answer the following questions.
For this task, please start with this command: "Find me a restaurant"

Reason for termination	Final Result / Satisfied / Annoyed
Are you satisfied with the results?	

How many questions do you think is enough to get you to the set of results that satisfies you for this task?	
How many questions do you think the assistant should ask to give you a best experience in accomplishing this task?	
These are the features the assistant had: 1. Filter by Location 2. Filter by Type of restaurant 3. Filter by Price (Sorting by price) 4. Sorting by distance 5. Filter by showing businesses with deals only	What were the features you felt you need when using the assistant to accomplish this task? Ranks these features according to your preference, write only the features that you need: 1. 2. 3. 4. 5.
Do you feel that you have accomplished this task?	YES / NO
Do you feel that the system helps you in accomplishing this task?	YES / NO
Rate the performance of the system in accomplishing this task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good

This is the end of the study, thank you for your participation and please fill in the following before handing this form to the experimenter.

How many questions do you think the assistant should ask to give you best possible result set when using it?	
How many questions do you think the assistant should ask to give you a best experience in using it?	
These are the features the assistant had: 1. Filter by Location 2. Filter by Type of restaurant 3. Filter by Price (Sorting by price) 4. Sorting by distance 5. Filter by showing businesses with deals only	Rank these features according to your preference, the first one being the highest, write only features that you feel you need: 1. 2. 3. 4. 5.
Rate the overall performance of the system	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good

Do you have any comments/remarks/ suggestions?	

RO3 - User Experience Study A Form for Experimenter

Subject Name:

Date:

Time:

Location:

English Proficiency: Beginner / Intermediate / Fluent / Native

Accent: YES / NO / Specify: _________

Case 1: (unlimited time)

Questions	**Loc**	**Tar**	**Pri**	**Dis**	**Dea**	**Fin**
What are the elements users already used in their initial command?				NA	NA	NA
Did the user tap on the filter list? (N for No, Y for Yes)						
Did the user tap to see the result? (N for No, Y for Yes)						NA
Did the user read the result? (N for No, Y for Yes)						
Did the user speak or type the command? (S for Speak, T for Type)						NA
Did the user stop at a question? (N for No, Y for Yes)						
Was the user given a help at any point? (N for No, Y for Yes)						
Did the user ask for help at any point? (N for No, Y for Yes)						
Did the user need to restart? If yes, where is the fail point?						

Case 2 (unlimited budget, limited time, location is specific, restart is not allowed)

Questions	**Loc**	**Tar**	**Pri**	**Dis**	**Dea**	**Fin**
What are the elements users already used in their initial command?				NA	NA	NA
Did the user tap on the filter list? (N for No, Y for Yes)						
Did the user tap to see the result? (N for No, Y for Yes)						NA
Did the user read the result? (N for No, Y for Yes)						
Did the user speak or type the command? (S for Speak, T for Type)						NA
Did the user stop at a question? (N for No, Y for Yes)						
Was the user given a help at any point? (N for No, Y for Yes)						
Did the user ask for help at any point? (N for No, Y for Yes)						
Did the user need to restart? If yes, where is the fail point?						

Case 3: (limited budget, limited time)

Questions	**Loc**	**Tar**	**Pri**	**Dis**	**Dea**	**Fin**
What are the elements users already used in their initial command?				NA	NA	NA
Did the user tap on the filter list? (N for No, Y for Yes)						
Did the user tap to see the result? (N for No, Y for Yes)						NA
Did the user read the result? (N for No, Y for Yes)						
Did the user speak or type the command? (S for Speak, T for Type)						NA
Did the user stop at a question? (N for No, Y for Yes)						
Was the user given a help at any point? (N for No, Y for Yes)						
Did the user ask for help at any point? (N for No, Y for Yes)						
Did the user need to restart? If yes, where is the fail point?						

Case 4: free form

Questions	**Loc**	**Tar**	**Pri**	**Dis**	**Dea**	**Fin**
What are the elements users already used in their initial command?				NA	NA	NA
Did the user tap on the filter list? (N for No, Y for Yes)						
Did the user tap to see the result? (N for No, Y for Yes)						NA
Did the user read the result? (N for No, Y for Yes)						
Did the user speak or type the command? (S for Speak, T for Type)						NA
Did the user stop at a question? (N for No, Y for Yes)						
Was the user given a help at any point? (N for No, Y for Yes)						
Did the user ask for help at any point? (N for No, Y for Yes)						
Did the user need to restart? If yes, where is the fail point?						

16. Appendix F: Main Study Form

RO3 - User Experience Main Study Form for Subject

Name: ______________________________

Age: ____________

Nationality: ___________________

Occupation: ______________________

Gender: M / F

Native Language: ___________________

English Proficiency: Beginner / Intermediate / Fluent / Native

Have you ever used an intelligent personal assistant? YES / NO

Have you ever been to the US? YES / NO

Thank you for participating in this user experience study. This study will assess your user experience and satisfaction when using an intelligent personal assistant. Prior experience in using similar product is not needed. This study is going to take 5-10 mins.

How to work on each task:

1. Start with clicking the icon on the bookmark bar or button to go to start page.
2. Press the mic button, and wait until the animated mic icon is displayed.
3. Start giving your command verbally.
4. When you are done with the command, pause and wait for it to process, if it does not proceed to the next page, click Submit button.
5. When being asked question, you can choose either to answer it or not, if you are already satisfied with the intermediate result, you can notify your experimenter and finish with the task.
6. Write down your answer at the end-of-task question.
7. If you need any help or have any question, ask the experimenter.
8. You are allowed to restart the task at any point of time, unless specified, simply click the icon on the bookmark bar

Notes:

1. Each task will take 1-2 minutes to be completed.
2. You are not allowed to read the next task before you finish your current task.
3. The “previous task” means the *most* recent task you finished prior to your current task

Case Background:

You are a recent graduate, currently living in San Francisco, US. You are working at a startup company and earn decent monthly pay. You are an alumni of University of California, Berkeley. You are a highly sociable person and you have friends from different student groups and age groups, including some of your juniors who are still studying in Berkeley. Apart from that, you just got to know that few of your friends are moving to other states to pursue further studies or work. For instance, a month ago, you got to know that Ben, your best friend is moving to New York to work at Goldman Sachs (an investment bank)

As a technology savvy, you use “the intelligent personal assistant” to help you find places to eat or drink.

Task 1:

It is Thursday, early in the morning and you are going to the office. Like other people, you drink coffee every morning and you are planning to grab a cup of coffee from a place near your office before you come in for work. Your office is located in a city called Cupertino. Since the city is really small, most places in the city are close to your office. You just left your home, and currently still in San Francisco.

Use your intelligent personal assistant to help you find a place to eat. When you have finished, please answer the following questions.

Did the assistant ask you question(s)?	YES / NO
Are you satisfied with the results?	
Rate your satisfaction	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
Rate your experience when performing the task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
Do you feel that you have accomplished this task?	YES / NO
Do you feel that the system helps you in accomplishing this task?	YES / NO
Rate the performance of the system in accomplishing this task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good

Task 2:

You have arrived at your office, and managed to grab your morning coffee. You get straight to work, and before you know it, it is already one hour away from your lunch break. You checked your calendar and found out that you have made an appointment with Amy. Amy also works at a startup company in a city called San Jose. Like Cupertino, San Jose is relatively small, hence most places are close to each other. As Cupertino and San Jose is close to each other, you decided to work for another 15 minutes before going for lunch. However, you need to tell Amy where should you two meet. You know Amy likes Chinese food.

Use your intelligent personal assistant to help you find a place to eat. When you have finished, please answer the following questions.

Did the assistant ask you question(s)?	YES / NO
Are you satisfied with the results?	
Rate your satisfaction	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the satisfaction you get from the results for this task with the previous one, which one is better?	This / Previous / Indifferent
Rate your experience when performing the task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the experience you get from performing this task with the previous one, which one is better?	This / Previous / Indifferent
Do you feel that you have accomplished this task?	YES / NO

Do you feel that the system helps you in accomplishing this task?	YES / NO
Rate the performance of the system in accomplishing this task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the performance of the system when performing this task with the previous one, which one is better?	This / Previous / Indifferent

Task 3:

You managed to satisfy Amy with your choice. After you said bye to Amy, you got back in your car, and check your schedule for the rest of the day. Unfortunately, you just got to know that you have a sharing to do at the University of California, Merced, for a class about internet marketing. The university is located in a city called Merced. The sharing you are going to speak on is scheduled at 8pm, so you are planning to have a dinner first before going to the sharing. Merced is 2 hours away from Cupertino. For the dinner tonight, you do not have any preference, however, you heard from one of your colleagues that there is a good mexican restaurant called "J&R Tacos" there. You are wondering where you should go for dinner tonight.

Use your intelligent personal assistant to help you find a place to eat. When you have finished, please answer the following questions.

Did the assistant ask you question(s)?	YES / NO
Are you satisfied with the results?	
Rate your satisfaction	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the satisfaction you get from the results for this task with the previous one, which one is better?	This / Previous / Indifferent
Rate your experience when performing the task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the experience you get from performing this task with the previous one, which one is better?	This / Previous / Indifferent
Do you feel that you have accomplished this task?	YES / NO
Do you feel that the system helps you in accomplishing this task?	YES / NO
Rate the performance of the system in accomplishing this task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the performance of the system when performing this task with the previous one, which one is better?	This / Previous / Indifferent

Task 4:

You are done for the day. It has been a long drive home from Merced to San Francisco. It is now 11 pm and you are about to go to bed. You realized that you are going to meet your buddy, Ben, this Saturday. Yes, you are going to fly to New York to spend a long weekend with Ben, and you are going to travel to cities nearby as well. You are meeting Ben for dinner on Satuday night in

New York as he is busy and will come in on Saturday morning to finish his work, and you know that Ben is working near the Empire State Building.

You want to treat him a nice dinner as you promised him to do so before you got a job.

Use your intelligent personal assistant to help you find a place to eat. When you have finished, please answer the following questions.

Did the assistant ask you question(s)?	YES / NO
Are you satisfied with the results?	
Rate your satisfaction	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the satisfaction you get from the results for this task with the previous one, which one is better?	This / Previous / Indifferent
Rate your experience when performing the task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the experience you get from performing this task with the previous one, which one is better?	This / Previous / Indifferent
Do you feel that you have accomplished this task?	YES / NO
Do you feel that the system helps you in accomplishing this task?	YES / NO
Rate the performance of the system in accomplishing this task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the performance of the system when performing this task with the previous one, which one is better?	This / Previous / Indifferent

Task 5:

It is Saturday late morning and you just arrived in New York. You checked in your hotel which is located near the Rockfeller Center. You are tired so you are planning to grab something to eat for lunch near your hotel.

Use your intelligent personal assistant to help you find a place to eat. When you have finished, please answer the following questions.

Did the assistant ask you question(s)?	YES / NO
Are you satisfied with the results?	
Rate your satisfaction	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the satisfaction you get from the results for this task with the previous one, which one is better?	This / Previous / Indifferent
Rate your experience when performing the task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the experience you get from performing this task with the previous one, which one is better?	This / Previous / Indifferent

Do you feel that you have accomplished this task?	YES / NO
Do you feel that the system helps you in accomplishing this task?	YES / NO
Rate the performance of the system in accomplishing this task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the performance of the system when performing this task with the previous one, which one is better?	This / Previous / Indifferent

Task 6:

One of the reason you go to the East Coast is that your partner is studying at a prestigious university called the University of Pennsylvania. The university is located in a city called Philadelphia. You just got a call from the office that your boss granted your request for work leave for 3 days, so you decided to take the next bus to Philadelphia to see your partner. It is 5pm and you just arrived in Philadelphia. You are planning to stay at your partner's place for the remaining nights. You want to surprise your partner by asking her to go to a nice restaurant in the city and meet him/her for dinner.

Use your intelligent personal assistant to help you find a place to eat. When you have finished, please answer the following questions.

Did the assistant ask you question(s)?	YES / NO
Are you satisfied with the results?	
Rate your satisfaction	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the satisfaction you get from the results for this task with the previous one, which one is better?	This / Previous / Indifferent
Rate your experience when performing the task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the experience you get from performing this task with the previous one, which one is better?	This / Previous / Indifferent
Do you feel that you have accomplished this task?	YES / NO
Do you feel that the system helps you in accomplishing this task?	YES / NO
Rate the performance of the system in accomplishing this task	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
How do you compare the performance of the system when performing this task with the previous one, which one is better?	This / Previous / Indifferent

That was a fantastic weekend, you are back in San Francisco to follow your daily routine. You just realized that your intelligent personal assistant is not working like it used to be (i.e. better or worse) and you want to send a feedback to the company who developed it. You got this feedback form to fill in. (please turn to the next page)

Do you feel more satisfied by having the assistant to ask you some questions?	YES / NO
Do you think that on the occasions where the assistant asked you questions, you get a better set of result?	YES / NO
Do you think that on the occasions where the assistant asked you questions, you get a better experience?	YES / NO
Do you think it is generally better to have the intelligent personal assistant to ask you questions?	YES / NO
Rate the overall performance of the system	Very Bad 1 2 3 4 5 6 7 8 9 10 Very Good
Do you have anything interesting to share about this experiment/comments/remarks/suggestions?	

This is the end of the study, thank you for your participation and hope you had fun.

RO3 - User Experience Main Study Form for Experimenter

Subject Name:

Date:

Time:

Location:

English Proficiency: Beginner / Intermediate / Fluent / Native

Accent: YES / NO / Specify: _________

Task 1	
Which implementation was applied?	A / B / Don't know
Task Completion time (in s)	
Was the subject able to complete the task?	
Was the subject stop the flow (for impl B), at which point?	
Any interesting comment from the subject? Any interesting phenomenon observed?	

Task 2	
Which implementation was applied?	A / B / Don't know
Task Completion time (in s)	
Was the subject able to complete the task?	
Was the subject stop the flow (for impl B), at which point?	
Any interesting comment from the subject? Any interesting phenomenon observed?	

Task 3	
Which implementation was applied?	A / B / Don't know
Task Completion time (in s)	
Was the subject able to complete the task?	
Was the subject stop the flow (for impl B), at which point?	

Any interesting comment from the subject? Any interesting phenomenon observed?	

Task 4	
Which implementation was applied?	A / B / Don’t know
Task Completion time (in s)	
Was the subject able to complete the task?	
Was the subject stop the flow (for impl B), at which point?	
Any interesting comment from the subject? Any interesting phenomenon observed?	

Task 5	
Which implementation was applied?	A / B / Don’t know
Task Completion time (in s)	
Was the subject able to complete the task?	
Was the subject stop the flow (for impl B), at which point?	
Any interesting comment from the subject? Any interesting phenomenon observed?	

17. Appendix G: Main Study Participants End-of-survey Comment

We included a non compulsory open-ended question at the end of our survey. The question was :

"Do you have anything interesting to share about this experiment / comments / remarks / suggestions?"

And the results we obtained were the following (random order):

"Sort by rating? add some type i.e. "cafe", "drink coffee"."

"It can ask even more follow up questions to drill down the results, or to sort them (by price, popularity, …)"

"The system tries to help me in accomplishing the tasks. When the task is less specific (I don't have a clear idea of what I want). It performs quite well. Once I have something in mind, it seems to give irrelevant results. But most of the time, if I don't have any specific thing in mind, I won't need an assistant as I will just use normal searching app and find something randomly"

"The system does not understand what I spoke so I prefer typing to speaking"

"When I type it fails, Apart from that, I enjoyed the overall experience and results generated were good"

"Maybe more language support"

"Design: can look better"

"Ask questions other than price range"

"The system cannot correctly detect what I was saying sometimes. Could add hotels or shopping malls to the system as well"

"Even though the assistant often asks similar or the same questions, I think I have a better experience after she asked me questions"

“Using place name gives all restaurants in the area. Good voice recognition”

“Cannot go back”