

# FPT UNIVERSITY

## Capstone Project Document

---

Anki Pan

AKP Team		
Project team	Bui Hong Nhung	SE02082
	Vu Hong Thai	SE02654
	Le Thanh Binh	SE02917
	Nguyen Duc Huy Hoang	SE02651
	Pham Trong Nghia	SE02781
	Ngo Thi Mai	SE02218
Supervisor	Mr. Tran Binh Duong	
Project code	AKP	

Ha Noi 14<sup>th</sup> Jul, 2015

## Acknowledge

*We specially thank to Mr. Tran Binh Duong, our supervisor, for his support through out this project. His instruction and motivation has been lessons for us not only about technique, but also about how to work as a team in a real project.*

*We also thank to our families and friends. Their encouragement has helped us to overcome many pressure to complete our product.*

## TABLE OF CONTENTS

I. INTRODUCTION .....	1
1. PROJECT INFORMATION .....	1
2. PROJECT TEAM .....	1
3. INITIAL IDEAL.....	2
3.1. Background .....	2
3.2. Existing Products.....	3
3.2.1. Google translate.....	3
3.2.2. CamDictionary.....	6
3.2.3. JDict.....	9
3.2. Our Proposal .....	11
II. PROJECT MANAGEMENT PLAN .....	13
1. PROJECT OVERVIEW .....	13
1.1. Project description.....	13
1.2. Scope.....	13
1.3. Standard Objectives.....	13
1.4. Milestones and Deliverables.....	14
1.4.1. Milestones .....	14
1.4.2. Deliverables .....	15
2. PROJECT ORGANIZATION.....	17
2.1 Software Process Model .....	17
2.2 Project Lifecycle .....	17
2.3 Roles and Responsibilities .....	18
3. TOOLS AND INFRASTRUCTURES .....	19
3.1. Hardware .....	19
3.2. Software .....	20
4. SCHEDULES.....	21
4.1. Detail Schedules.....	21
5. COMMUNICATION MANAGEMENT .....	25
5.1. Stakeholders and Contacts .....	25
5.2. Communication Management Approach .....	26
5.3. Communication Requirements .....	26
5.4. Communication Method and Technologies.....	26
5.5 Communication Matrix.....	27
6. RISK MANAGEMENT <sup>[1]</sup> .....	29
6.1. Risk Management Approach.....	29
6.2. Risk Identification.....	29
6.3. Risk Qualification and Prioritization .....	29
6.4. Risk Monitoring .....	29
6.5. Risk Register .....	30
6.5.1. Risk description .....	30
6.5.2. Probability – Impact matrix .....	31
6.5.3. Risk response plan & Risk status.....	32

7. QUALITY MANAGEMENT .....	34
7.1. Quality Management Overview.....	34
7.1.1. Organization, Responsibilities, and Interfaces .....	34
7.1.2. Tools, Environment, and Interfaces .....	34
7.2. Quality Planning.....	34
7.2.1. Define Project Quality .....	34
7.2.2. Measure Project Quality.....	35
7.3. Quality Assurance.....	36
7.3.1. Analyze Project Quality .....	36
7.3.2. Improve Project Quality.....	37
7.4. Quality Control .....	37
7.5. Action Plan.....	38
III. SOFTWARE REQUIREMENT SPECIFICATION .....	40
1. INTRODUCTION .....	40
1.1 Purpose.....	40
1.2. Scope.....	40
1.3. Overview.....	40
2. OVERALL DESCRIPTION .....	40
2.1. Scope of work.....	40
2.2. Assumptions.....	41
3. FUNCTIONAL REQUIREMENTS .....	42
3.1. Android application.....	42
3.1.1. User Case Diagram.....	42
3.1.2. Functions.....	43
3.2. Server .....	76
3.2.1 User case diagram .....	76
3.2.2. Functions.....	77
4. NON – FUNCTIONAL REQUIREMENTS .....	102
4.1. Usability .....	102
4.2. Availability.....	102
4.3. Performance .....	102
4.4. Interface .....	102
4.4.1. User interface.....	102
4.4.2. Hardware interface.....	102
4.4.3. Software interface.....	102
4.5. Security .....	102
IV. SOFTWARE DETAIL DESIGN .....	103
1. ARCHITECTURE DESIGN .....	103
1.1. Introduction.....	103
1.1.1. Purpose.....	103
1.1.2. Scope .....	103
1.1.3. Overview .....	103
1.2. Description of Architecture Design.....	104
1.3. Architecture Goals & Constraints.....	104
1.3.1. Technical Platform.....	105
1.3.2. Communication .....	105
1.3.3. Security .....	105
1.3.4. Persistence .....	105
1.3.5. Reliability/Availability.....	105

1.3.6. Performance .....	105
1.4. User Case View .....	105
1.5. Deployment View .....	107
1.6. Implementation View.....	108
1.7. Detail Design of Application.....	108
1.7.1. Overview .....	108
1.7.2. Design.....	109
1.8. Detail design of Server.....	162
1.8.1. Overview .....	162
8.1.2. Detail design .....	163
1.9. AKP image processing library .....	187
1.9.1. Purpose.....	187
1.9.2. Tesseract' shortcoming .....	188
1.9.3. Tesseract's recognition of Japanese.....	190
1.9.4. AKP Image Processing Library .....	191
1.9.5. AKP image processing library's limitation.....	201
1.10. Size And Performance.....	201
1.11. Quality.....	201
1.11.1. Scalability:.....	201
1.11.2. Reliability / Availability:.....	201
1.11.3. Portability: .....	201
1.11.4. Security: .....	201
<b>2. DATABASE DESIGN .....</b>	<b>202</b>
2.1. Introduction.....	202
2.1.1. Purpose.....	202
2.1.2. Overview .....	202
2.2. Database.....	203
2.2.1. Application database .....	203
2.2.2. Server database .....	206
2.3. Code Design.....	208
2.3.1. Japanese_dict.content.....	208
2.3.2. Kanji.content.....	208
2.3.3. User.password.....	208
<b>3. INTERFACE DESIGN .....</b>	<b>209</b>
3.1. Introduction.....	209
3.2. Interface Design of Application .....	209
3.2.1. Structure and Layout.....	209
3.2.2. Screen Description .....	211
3.3. Interface Design of Server .....	221
3.3.1. Structure and layout.....	221
3.3.2. Page content description.....	223
<b>V. EVALUATION.....</b>	<b>235</b>
<b>1. TEST PLAN .....</b>	<b>235</b>
1.1. Introduction.....	235
1.1.1. Purpose.....	235
1.1.2. Background information.....	235
1.1.3. Scope of testing .....	236
1.1.4. Constraints.....	237
1.1.5. Risk List.....	237
1.1.6. Training Needs .....	237
1.2. Requirement For Test.....	238

1.2.1. Test items.....	238
1.2.2. Acceptance test criteria .....	239
1.3. Test strategy .....	239
1.3.1. Test type.....	239
1.3.2 Test stages .....	241
1.4. Resource .....	242
1.4.1 Human resource.....	242
1.4.2 Environment .....	243
1.5. Test Milestones .....	243
1.6. Deliverables.....	244
2. TEST REPORT .....	245
2.1. Integration test .....	245
2.2. System test.....	245
3. CHECK LIST .....	246
VI. IMPLEMENTATION.....	248
1. DEVELOPMENT .....	248
2. PROGRESS REPORT.....	248
VII. INSTALLATION GUIDE AND USER MANUAL .....	251
1. INSTALLATION GUIDE .....	251
1.1. English Version.....	251
1.1.1. Prerequisites.....	251
1.1.2. Installation procedures.....	251
1.1.3. Uninstall procedures.....	251
1.2. Japanese Version .....	251
1. 1. 1. 前提条件 .....	251
1. 1. 2. インストール手順.....	251
1. 1. 3. アンインストール手順.....	252
2. USER MANUAL .....	252
2.1. Select a screen.....	252
2.2. Scan document .....	253
2.3. Select text on scanned document.....	253
2.4. Search word in Japanese dictionary.....	254
2.5. Search kanji in Kanji dictionary.....	255
2.6. View history .....	256
2.7. View all favorite words .....	256
2.8. Clear history.....	257
2.10. Select Dictionary package.....	258
2.11. Delete Dictionary package .....	259
2.12. Download Dictionary package .....	259
VII. CONCLUSION .....	261
1. LESSONS LEARN.....	261
2. FUTURE WORK.....	261
APPENDICES .....	262
1. SURVEY SUMMARY .....	262
2. MEETING MINUTES .....	264

3. CONFIGURATION MANAGEMENT.....	267
3.1. CI Identification and Naming Convention .....	267
3.2. Version Numbering Rules <sup>[6]</sup> .....	269
3.3. Directory Structure .....	270
4. CODING CONVENTION .....	271
4.1. Coding Convention.....	271
4.2. Comment Convention .....	272
REFERENCES:.....	273

## ACRONYMS & ABBREVIATIONS

Abbreviations	Description	Note
AKP	Anki Pan	
CI	Configuration Items	
OCR	Optical Character Recognition	
MM	Meeting Minute	
PM	Project Manager	
PMP	Project Management Plan	
SDD	Software Detail Design	
SRS	Software Requirement Specification	
SUM	Software User Manual	
W#	Week#	
WR	Weekly Report	
SAD	Software Architecture Design	
FRS	Functional Requirement Specification	
QA	Quality Assurance	
PM	Project Manager	
SRS	Software Requirement Specification	
TM	Team Member	
De	Designer	
DB	Database	
.db	SQLite Database file extension	
.ico	Icon file extension	
PK	Primary Key	
FK	Foreign Key	

## I. INTRODUCTION

### 1. PROJECT INFORMATION

- ❖ **Project name:** Anki Pan
- ❖ **Project code:** AKP
- ❖ **Project type:** Android application
- ❖ **Project category:** New development
- ❖ **Business domain:** Education
- ❖ **Project manager:** Bui Hong Nhung
- ❖ **Time line:** May 4<sup>th</sup> 2015 to August 15<sup>th</sup> 2015

### 2. PROJECT TEAM

- ❖ **Supervisor**

Name	Email	Title
<b>Tran Binh Duong</b>	duongtb@fpt.edu.vn	Supervisor

- ❖ **Team member**

Name	Student Roll	Email	Role
<b>Bui Hong Nhung</b>	SE02082	NhungbhSE02082@fpt.edu.vn	Leader
<b>Vu Hong Thai</b>	SE02654	thaivhSE02654@fpt.edu.vn	Member
<b>Le Thanh Binh</b>	SE02917	binhltSE02917@fpt.edu.vn	Member
<b>Nguyen Duc Huy Hoang</b>	SE02651	HoangndhSE02651@fpt.edu.vn	Member
<b>Pham Trong Nghia</b>	SE02781	NghiaptSE02781@fpt.edu.vn	Member
<b>Ngo Thi Mai</b>	SE02218	MaintSE02218@fpt.edu.vn	Member

### 3. INITIAL IDEAL

#### 3.1. Background

Japan is one of the most advanced countries in the world. In addition, Japanese culture is spreading very quickly to other countries through manga, anime,... Therefore, more and more people study Japanese. A research of JPF (Japan Foundation for Culture Exchange) shows that until 2012, there are 3,985,669 people studying Japanese, 63,805 Japanese teachers and 16,046 Japanese educational institutions. Those numbers are still constantly growing bigger and bigger.

表1-1-1 機関数・教師数・学習者数

	2012年	2009年
機関(機関)	16,046	14,925
教師(人)	63,805	49,803
学習者(人)	3,985,669	3,651,232

(機関 : Organization 教師 : Teacher 学習者 : Student)

Figure 1: JPF research in 2013

Most of people who study Japanese find it difficult to read Japanese document. Japanese includes a lot of Chinese-like characters called Kanji. Without knowing how to read the Kanji, it will be difficult to type those Kanji to search for their meaning.

We conducted a research among students who are studying Japanese. Those students often read books, comic, newspapers written in Japanese (see Chart 1). The result shows that 88.2% out of 78 students doing the survey find it difficult to read Japanese document (see Chart 2).

Do you often read Japanese document?

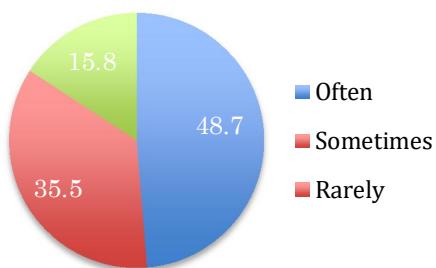


Chart 1

Is it hard to read Japanese?

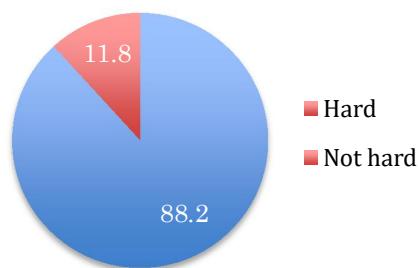


Chart 2

Almost of the students want an application that can scan Japanese in documents. More than 60% of them need help in translate the document, and search for the meaning of words.

(See AK\_SurveySummary in Appendix for detail about this survey)

## 3.2. Existing Products

### 3.2.1. Google translate<sup>[2]</sup>

Google translate is considered as the best application in language processing. It provides user a completed services including character recognition, speech to text and translate.

#### Functions:

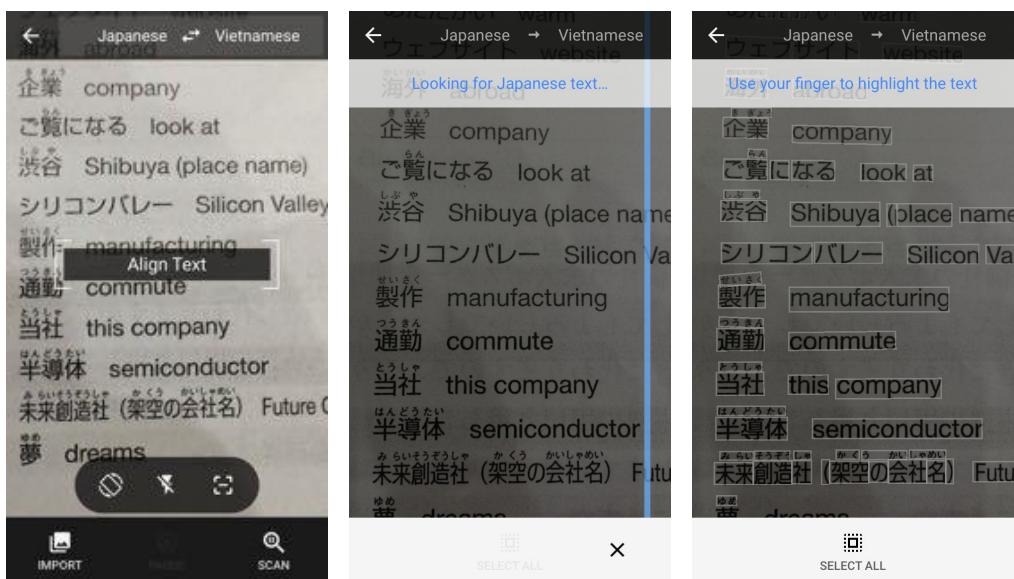
- Scan printed document and detect characters of many language, include Japanese
- Allow user to import image to detect characters
- Translate to many other language
- Convert speech to text and translate
- Save translated words and documents
- Bookmark words / documents
- Copy words / documents
- Turn on flash to take image

#### Some screenshots of Google translate:

Main screen



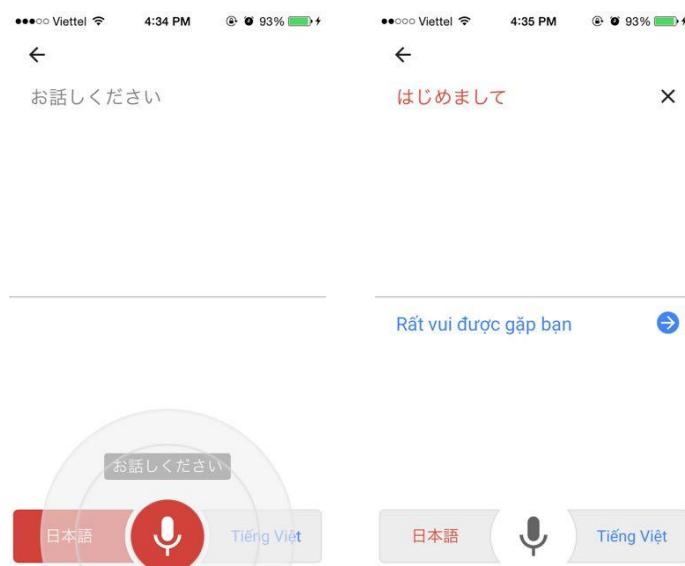
## Scan and recognize text:



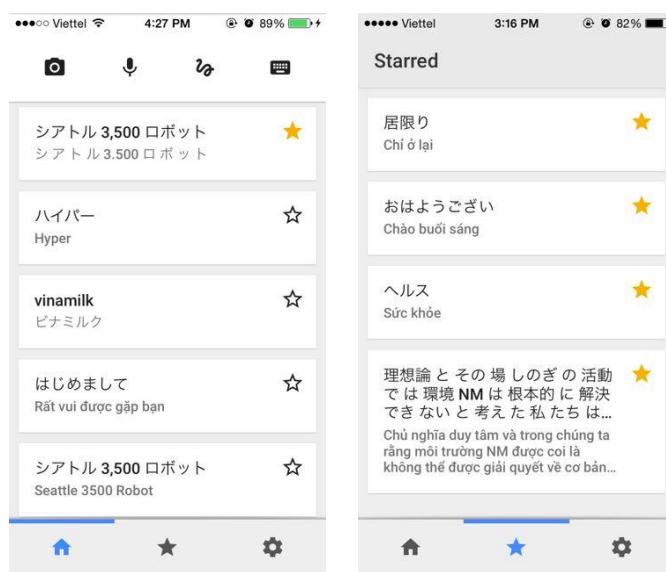
## Translate to other languages:



### Speech to text



### Save to history and bookmark

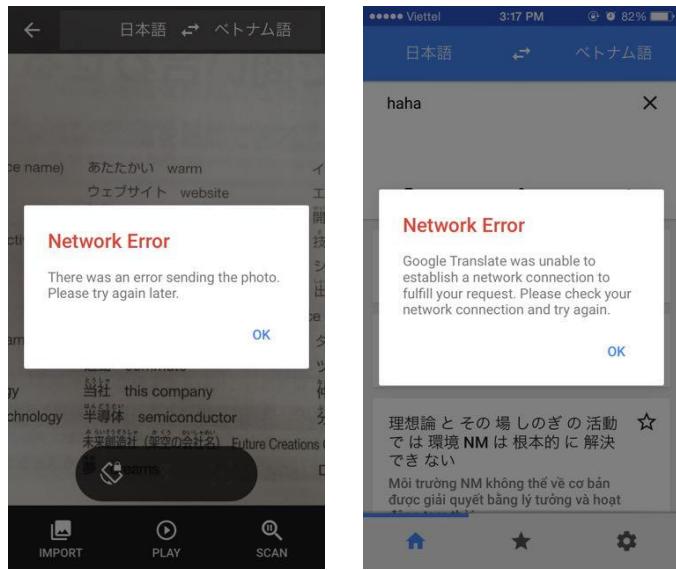


#### Pros:

- Interface design is simple and easy to use
- Document scanning is fast
- Character recognition is in high correction rate
- Recognize many language
- Translate from / to many language

#### Cons:

- Require Internet access for character recognition and translation



- Translation may return wrong meaning of words
- Does not provide other information about words (word-class, examples,...). Only the meaning of words may not satisfies people who study Japanese and want to understand the words deeply.

### **3.2.2. CamDictionary<sup>[3]</sup>**

CamDictionary is another good application in character recognition

#### **Functions:**

- Scan and recognize printed characters of many languages in three ways:
  - + User touch somewhere in the screen to show application the area to scan
  - + User move the application eye to the text area that they want to scan
  - + User capture to document, and then choose a part of the document to scan
- Allow user to import image to detect characters
- Turn on flash to take image
- Translate from / to many other languages
- Save translated words to History or My Vocabulary
- Download dictionary and search words in dictionary
- Send words and words' meaning through message or email

#### **Some screenshots of CamDictionary:**

## Scan and recognize text:



## Translate:



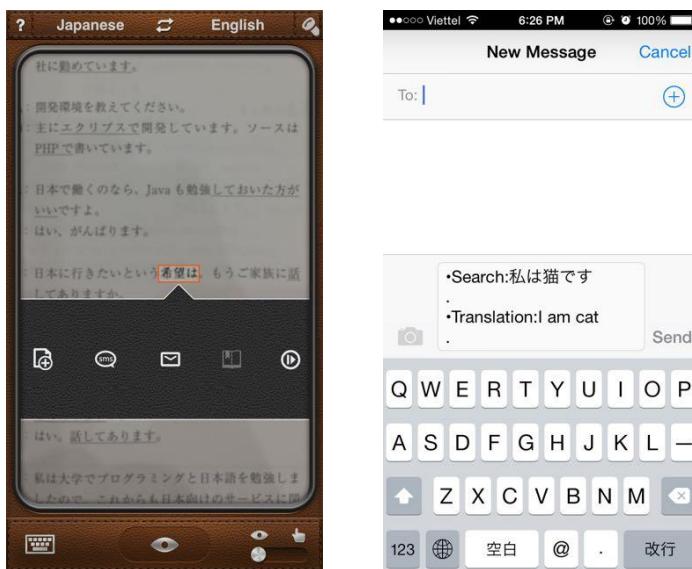
## Save to history or my vocabulary



## Download dictionary:



Send through email or message:



### Pros:

- The application is interactive. User can choose to scan document in 3 ways. The design for scanning screen is interesting.
- Text scan and character recognition is highly correct.
- The scanning function works offline
- User can purchase dictionary to search words, which give a more correct meaning and other information like word-class, example,...

### Cons:

- Require Internet access for translation
- User has to pay for the dictionaries

### 3.2.3. JDict<sup>[4]</sup>

Jdict is one of the most popular Japanese – Vietnamese dictionary application

#### Functions:

- Look up words in Japanese – Vietnamese dictionary, Vietnamese – Japanese dictionary and Kanji dictionary
- Save words to history
- Bookmark words

#### Some screenshots of JDdict

## Search words in dictionary



## Save to history and bookmark



### Pros:

- Simple and easy-to-understand interface design
- Give detail meanings, examples, word-class, pronunciation of words
- Can search for about 700.000 Japanese words, and almost Kanji
- Explain words' meaning in Vietnamese, which is helpful for Vietnamese people.

### Cons:

- User has to input words to search for the meaning. If they do not know how to read Kanji, they can not type the Kanji to search. It will take time to draw the Kanji from keyboard.

### 3.2. Our Proposal

We will develop a single application that support people who studying Japanese everything they need for reading printed Japanese document. The application will have the below main functions:

- Scan text in Japanese document
- Recognize Japanese characters
- Allow user to download free dictionaries
- Search for meaning of words, Kanji in dictionary
- Save words to history for user's later review
- Add words to favorite list

The application will be interactive, easy to use and work offline.



*Figure 2: Doraemon's copying toast*

We name the application Anki Pan, which means copying toast in English. Anki Pan is a Doraemon's gadget which allows user to memorize any contents 'printed' on it. To use the copying toast, one should press one side of the toast on a book page. The text on the book then will be printed on the toast. Eating the toast later, eater can remember everything that has been printed on the toast. Just like how to use the copying toast, user using our application will take their phone above the book page and press the capture button. The book page with text on it then will be saved (printed) to the application. The application will extract text from the book page so that user can require application to translate or to look up meaning of words. By doing this, user can understand the content of the book, remember words and their meaning, just like swallowing the content of book by eating a toast.

Our application in compare to Google Translate, CamDictionary and JDICT:

Function	Google Translate	CamDictionary	JDict	Anki Pan
Scan and recognize text	Many language	Many language	✗	Japanese only
Translate	✓	✓	✗	✗
Search words in dictionary	✗	✓	✓	✓
Free Dictionary	✗	✗	✓	✓
Save words to history	✓	✓	✓	✓
Bookmark words	✓	✓	✓	✓
Work offline	✗	✗	✓	✓

## II. PROJECT MANAGEMENT PLAN

### 1. PROJECT OVERVIEW

#### 1.1. Project description

In this project, we will develop an Android application that support Japanese learner in reading Japanese book. User can use this application to scan Japanese in a book page and look up for the meaning of each words. This will save user's time for searching words in the book that they do not understand, because they do not have to type every single words. The Application will be design so that user have to do least but receive best result. Beside, we will also develop a business service for admins to manage dictionary database.

#### 1.2. Scope

This project covers all processes, from planning, requirement specification, design, development, to testing.

Project team will develop an Android Application for user to scan Japanese document and search words' meaning, and a server to manage dictionary database and provide dictionary download service for the Application.

##### ❖ Functional requirements:

The Application allows user to scan text in Japanese documents and search for words' meaning in dictionaries. All searched words will be saved to history. User can also add a word to their favorite list. Application provides dictionary packages for user to download and use for searching word.

The server is developed for Admins and Roots Admins of Anki pan system. Admin / Root Admin can login, logout, manage their profile, manage dictionary packages (upload, edit, delete). Root Admin can add / edit / delete Admins' accounts.

##### ❖ Non-functional requirements:

The Application must be design for user's convenience. It should require no internet access to run, except for downloading dictionary package. Interface design should be good looking and easy to understand. Time delay for image processing and text recognition must be optimized.

#### 1.3. Standard Objectives

- This project must be finished no later than 15/08/2015
- The final Application covers 100% of requirements
- The 6 team members give best effort to complete the project

## 1.4. Milestones and Deliverables

### 1.4.1. Milestones

No.	Task	Start date	End date
1	Approve Project's initial idea	11/05/2015	11/05/2015
2	Create and submit Report 1: Introduction	18/05/2015	22/05/2015
3	Create and submit Report 2: Project Management Plan ver1.0	28/05/2015	1/06/2015
4	Create and submit Report 3: Software Requirement Specification ver1.0	2/06/2015	5/06/2015
5	Design interface ver1.0	3/06/2015	5/06/2015
6	Create Report 4: Software Architecture design ver1.0	5/06/2015	5/06/2015
7	Create and submit Test plan ver1.0	8/06/2015	10/06/2015
8	Evaluate Increment 1	12/06/2015	12/06/2015
9	Design database ver1.0	17/06/2015	22/06/2015
10	Report result of testing Tesseract library	23/06/2015	23/06/2015
11	Complete Scan screen	9/07/2015	9/07/2015
12	Evaluate Increment 2	13/07/2015	13/07/2015
13	Complete server	22/07/2015	22/07/2015
14	Create & execute & report ST of server	20/07/2015	24/07/2015
15	Complete Dictionary screen	23/07/2015	23/07/2015
16	Complete Result screen	23/07/2015	23/07/2015
17	Complete image processing library	29/07/2015	29/07/2015
18	Complete History screen	30/07/2015	30/07/2015
19	Complete Setting screen	31/07/2015	31/07/2015
20	Create & execute & report IT	30/07/2015	2/08/2015
21	Evaluate Increment 3	3/08/2015	3/08/2015
22	Create and submit Report 6: User manual	5/08/2015	6/08/2015
23	Execute & report ST of Application	5/08/2015	7/08/2015
24	Submit final report of Testing	7/08/2015	7/08/2015
25	Submit final report	14/08/2015	14/08/2015

### 1.4.2. Deliverables

No.	Deliverables	Format	Delivery date	Verified by	Deliver Medium
<b>❖ Initial</b>					
1	Introduction ver1.0	.pdf	22/05/2015	Supervisor	Email
	MM_15052015	.pdf	15/06/2015	Supervisor	Email
	MM_22052015	.pdf	22/05/2015	Supervisor	Email
	WR_W1_22052015	.pdf	22/05/2015	Supervisor	Email
<b>❖ Increment 1</b>					
	Communication matrix ver1.0	.xls	1/06/2015	Supervisor	Email
	Risk register ver1.0	.xls	1/06/2015	Supervisor	Email
	Work schedule	.mpp	1/06/2015	Supervisor	Email
	Project management plan ver1.0	.pdf	1/06/2015	Supervisor	Email
	Software requirement specification ver1.0	.pdf	5/06/2015	Supervisor	Email
	Component design	.asta	5/06/2015	Supervisor	Email
	Test plan ver1.0	.pdf	11/06/2015	Supervisor	Email
	MM_01062015	.pdf	1/06/2015	Supervisor	Email
	MM_12062015	.pdf	12/06/2015	Supervisor	Email
	WR_W3_29052015		29/05/2015	Supervisor	Email
	WR_W4_05062015		05/06/2015	Supervisor	Email
	WR_W5_12062015		12/06/2015	Supervisor	Email
<b>❖ Increment2</b>					
	Design of database	.jpg	22/06/2015	Supervisor	Email
	Result of testing Tesseract	.xls	23/06/2015	Supervisor	Email
	MM_15062015	.pdf	15/06/2015	Supervisor	Email
	MM_29062015	.pdf	29/06/2015	Supervisor	Email
	WR_W6_19062015		19/06/2015	Supervisor	Email
	WR_W7_27062015		27/06/2015	Supervisor	Email
	WR_W8_12072015		12/07/2015	Supervisor	Email

<b>❖ Increment 3</b>					
	Software requirement specification ver2.10	.pdf	14/07/2015	Supervisor	Email
	Work schedule ver2.0	.mpp	14/07/2015	Supervisor	Email
	Project management plan ver2.0	.pdf	14/07/2015	Supervisor	Email
	Introduction ver2.0	.pdf	16/07/2015	Supervisor	Email
	Screen design ver1.0	.pdf	29/07/2015	Supervisor	Email
	MM_20072015	.pdf	20/07/2015	Supervisor	Email
	MM_28072015	.pdf	28/07/2015	Supervisor	Email
	WR_W10_19072015	.pdf	19/07/2015	Supervisor	Email
	WR_W11_26072015	.pdf	26/07/2015	Supervisor	Email
	WR_W12_02082015	.pdf	02/08/2015	Supervisor	Email
<b>❖ Increment 4</b>					
	User manual ver1.0	.pdf	6/08/2015	Supervisor	Email
	Database design ver1.1	.pdf	6/08/2015	Supervisor	Email
	Software detail design ver2.0	.pdf	7/08/2015	Supervisor	Email
	Final report of Testing	.xls	7/08/2015	Supervisor	Email
	Final report	.pdf	13/07/2015	Supervisor	Email
		hard copy	14/07/2017	FPT University	Direct
	WR_W13_09082015	.pdf	09/08/2015	Supervisor	Email
	WR_W14_16082015	.pdf	16/08/2015	Supervisor	Email

## 2. PROJECT ORGANIZATION

### 2.1 Software Process Model

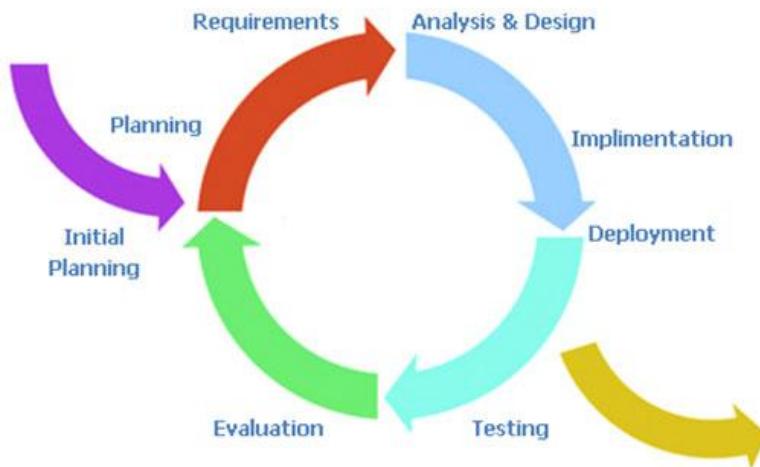


Figure 3: Incremental and Iterative Model

In this project, we apply Incremental and Iterative model as development process model.

### 2.2 Project Lifecycle

This project is divided into 6 phases: Initiation, Increment 1, Increment 2, Increment 3, Increment 4, Closing. Each Increment goes through planning, requirement specification, analysis and design, development, testing and evaluation. In each Incremental we bases on result of the previous Increment to plan and specify requirement and develop additional function for the system. Project manager monitors and controls the progress of project team through out the project.

## 2.3 Roles and Responsibilities

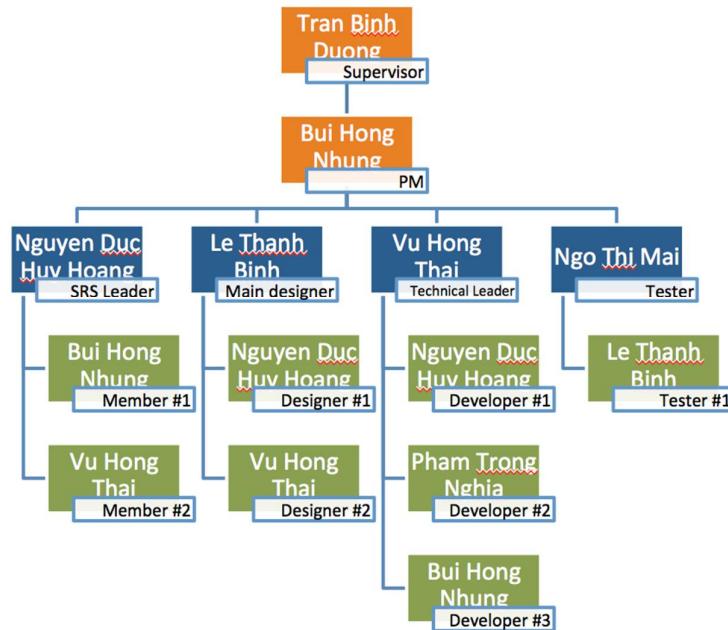


Figure 4: Organization structure

Role	Full name	Responsibility
Instructor	Tran Binh Duong	<ul style="list-style-type: none"> <li>- Give instruction for project team</li> <li>- Verify deliverables</li> <li>- Supervise project team's status</li> </ul>
PM	Bui Hong Nhung	<ul style="list-style-type: none"> <li>Have overall responsibility of the project</li> <li>- Create project plan</li> <li>- Assign task to team members</li> <li>- Tracking team member's work</li> <li>- Report work status to the instructor</li> </ul>
❖ Requirement analysis team		
SRS leader	Nguyen Duc Huy Hoang	Complete SRS document and submit to PM
Member #1	Bui Hong Nhung	Create user case diagram
Member #2	Vu Hong Thai	Specify non-functional requirement
❖ Design team		
Main designer	Le Thanh Binh	Design interface for Application and server

Designer #1	Vu Hong Thai	Support in design for Application
Designer #2	Nguyen Duc Huy Hoang	Support in design for server
❖ Development team		
Technical leader	Vu Hong Thai	<ul style="list-style-type: none"> <li>- Train other member of Android development</li> <li>- Create coding convention</li> <li>- Decide technique and tools to be used</li> <li>- Develop android Application</li> <li>- Create SDD</li> </ul>
Developer #1	Nguyen Duc Huy Hoang	<ul style="list-style-type: none"> <li>- Code server</li> </ul>
Developer #2	Pham Trong Nghia	<ul style="list-style-type: none"> <li>- Design database</li> <li>- Code for database related functions of android Application</li> </ul>
Developer #3	Bui Hong Nhung	<ul style="list-style-type: none"> <li>- Develop image processing library</li> </ul>
❖ Test team		
Test leader	Ngo Thi Mai	<ul style="list-style-type: none"> <li>- Create test plan</li> <li>- Create test cases</li> <li>- Execute test cases</li> <li>- Report test result</li> </ul>
Tester #1	Le Thanh Binh	<ul style="list-style-type: none"> <li>- Create test cases</li> <li>- Execute test case</li> </ul>

### 3. TOOLS AND INFRASTRUCTURES

#### 3.1. Hardware

- Personal computer for coding and testing with minimum configuration of 2GB RAM, 80GB of hard disk, Intel core 2 Duo
- Smart phone run on Android operating system
- Internet network connection

### 3.2. Software

Category	Software name	Version
Operating system	Microsoft Windows 7	Professional
Office tools	Microsoft Word	2010
	Microsoft Excel	2010
	Microsoft Power point	2010
Management tool	Microsoft Project	2010
	Trello	
	Google Driver	
Design tool	Astah	6.8.0
	<a href="http://www.draw.io">www.draw.io</a>	
	Adobe Photoshop	CS6
Development tool	Eclipse	
Database tool	SQLite manager	0.8.1
	MySQL	
Source code management tool	Github.com	

## 4. SCHEDULES

### 4.1. Detail Schedules

ID	Task Name	Duration	Start	Finish	Predecessors	Resource Names	04 May
1	<b>Pre-initiate</b>	<b>6 days</b>	<b>Mon 04/05/15</b>	<b>Sun 10/05/15</b>			
2	Training on Android development	5 days	Mon 04/05/15	Fri 08/05/15		NghiaPT,BinhLT	
3	Research on Japanese detection library	5 days	Mon 04/05/15	Fri 08/05/15		HoangNDH,NhungBH,ThaiVH	
4	<b>Initiate</b>	<b>9.5 days</b>	<b>Mon 11/05/15</b>	<b>Fri 22/05/15</b>			
5	<b>Kick off meeting</b>	<b>0.5 days</b>	<b>Mon 11/05/15</b>	<b>Mon 11/05/15</b>		BinhLT, HoangNDH, MaiNT	
6	Approve Project's initial idea	0.25 days	Mon 11/05/15	Mon 11/05/15			
7	Define member's responsibilities	0.25 days	Mon 11/05/15	Mon 11/05/15			
8	Set up tools	1 day	Tue 12/05/15	Tue 12/05/15			
9	<b>Training</b>	<b>7 days</b>	<b>Wed 13/05/15</b>	<b>Thu 21/05/15</b>			
10	Training on Android development	7 days	Wed 13/05/15	Thu 21/05/15		NghiaPT	
11	Study OpenCV library & image processing steps	7 days	Wed 13/05/15	Thu 21/05/15		HoangNDH, NhungBH	
12	<b>Meeting with instructor</b>	<b>0.5 days</b>	<b>Fri 15/05/15</b>	<b>Fri 15/05/15</b>		BinhLT, HoangNDH, Instructor	
13	Review project's idea	0.5 days	Fri 15/05/15	Fri 15/05/15			
14	Discuss on technique to be used	0.5 days	Fri 15/05/15	Fri 15/05/15			
15	Note for project's deliverables	0.5 days	Fri 15/05/15	Fri 15/05/15			
16	<b>Create project introduction document</b>	<b>4 days</b>	<b>Mon 18/05/15</b>	<b>Fri 22/05/15</b>			
17	Research on Japanese study in Vietnam	1 day	Mon 18/05/15	Mon 18/05/15		MaiNT, NghiaPT	
18	Do survey on Japanese study in Vietnam	3 days	Mon 18/05/15	Wed 20/05/15		MaiNT, NghiaPT, NhungBH	
19	Research on existing product	1 day	Mon 18/05/15	Mon 18/05/15		ThaiVH	
20	Create project introduction document	1 day	Thu 21/05/15	Thu 21/05/15	17,18,19	NhungBH	
21	Submit Introduction document	0 days	Fri 22/05/15	Fri 22/05/15	20	NhungBH	
22	<b>Meeting with instructor</b>	<b>0.5 days</b>	<b>Fri 22/05/15</b>	<b>Fri 22/05/15</b>		BinhLT, HoangNDH, Instructor	
23	Comment on Introduction document	0.5 days	Fri 22/05/15	Fri 22/05/15			
24	Discuss on Keypoint detection of OpenCV	0.5 days	Fri 22/05/15	Fri 22/05/15			

Figure 5: Schedule for Pre-initiate and Initiate phase

ID	Task Name	Duration	Start	Finish	Predecessors	Resource Names	04 M M
25	<b>Increment 1: Understand team's ability</b>	<b>14.5 days</b>	<b>Mon 25/05/15</b>	<b>Fri 12/06/15</b>			
26	Planning	3 days	Thu 28/05/15	Mon 01/06/15			
27	Create PMP	3 days	Thu 28/05/15	Mon 01/06/15			
28	<b>Choose process model</b>	0.25 days	Thu 28/05/15	Thu 28/05/15		NhungBH,ThaiVH	
29	<b>Create work schedule</b>	0.75 days	Thu 28/05/15	Thu 28/05/15		NhungBH	
30	Create risk management plan	0.5 days	Fri 29/05/15	Fri 29/05/15		BinhLT,NhungBH	
31	Create communication management plan	0.5 days	Fri 29/05/15	Fri 29/05/15		HoangNDH,NhungBH	
32	Complete project plan	0.5 days	Mon 01/06/15	Mon 01/06/15		NhungBH	
33	<b>Submit PMP to Supervisor</b>	0.5 days	Mon 01/06/15	Mon 01/06/15	32FF	NhungBH	
34	Requirement	3 days	Tue 02/06/15	Fri 05/06/15			
35	Create SRS	3 days	Tue 02/06/15	Thu 04/06/15		NhungBH,HoangNDH	
36	<b>Submit SRS to supervisor</b>	0 days	Fri 05/06/15	Fri 05/06/15	35	NhungBH	
37	Analysis and design	2 days	Thu 04/06/15	Fri 05/06/15			
38	Interface design	2 days	Wed 03/06/15	Fri 05/06/15			
39	Design dictionary screen	2 days	Wed 03/06/15	Thu 04/06/15	35FS-2 days	BinhLT,NghiaPT,ThaiVH	
40	<b>Submit design to group</b>	0 days	Fri 05/06/15	Fri 05/06/15	39	BinhLT	
41	Architecture design	1 day	Fri 05/06/15	Fri 05/06/15			
42	Design components structure for application	1 day	Fri 05/06/15	Fri 05/06/15	35	ThaiVH	
43	Implementation	14 days	Mon 25/05/15	Thu 11/06/15			
44	Study	5 days	Mon 25/05/15	Fri 29/05/15			
45	Study SQLite	5 days	Mon 25/05/15	Fri 29/05/15		NghiaPT	
46	<b>Study test process &amp; test related term</b>	5 days	Mon 25/05/15	Fri 29/05/15		MaiNT	
47	Study more about photoshop	4 days	Mon 25/05/15	Thu 28/05/15		BinhLT	
48	Coding	4 days	Mon 08/06/15	Thu 11/06/15	39,47	ThaiVH	
49	<b>Demo of dictionary functions</b>	3 days	Mon 08/06/15	Wed 10/06/15		NghiaPT	
50	Image processing library ver1.0	3 days	Mon 08/06/15	Wed 10/06/15		NhungBH	
51	Demo application	1 day	Thu 11/06/15	Thu 11/06/15	49,50	ThaiVH	
52	<b>Testing</b>	4 days	Mon 08/06/15	Thu 11/06/15			
53	<b>Create test plan</b>	3 days	Mon 08/06/15	Wed 10/06/15	35FS+1 day	MaiNT	
54	Submit test plan to PM	0 days	Thu 11/06/15	Thu 11/06/15	53	MaiNT	
55	Review test plan	1 day	Thu 11/06/15	Thu 11/06/15	53	NhungBH	
56	Evaluation	0.5 days	Fri 12/06/15	Fri 12/06/15	48	BinhLT,HoangNDH,Instruc	
57	<b>Evaluate demo application</b>	0.5 days	Fri 12/06/15	Fri 12/06/15			
58	Set goals for next increment	0.5 days	Fri 12/06/15	Fri 12/06/15			

Figure 6: Schedule for Increment 1

ID	Task Name	Duration	Start	Finish	Predecessors	Resource Names	04 M M
59	Increment 2: Develop main functions	<b>20.5 days</b>	<b>Mon 15/06/15</b>	<b>Mon 13/07/15</b>			
60	<b>Planning</b>	<b>1 day</b>	<b>Mon 15/06/15</b>	<b>Mon 15/06/15</b>			
61	Review and update PMP	1 day	Mon 15/06/15	Mon 15/06/15		NhungBH	
62	Requirement	<b>2 days</b>	<b>Mon 15/06/15</b>	<b>Tue 16/06/15</b>		<b>HoangNDH</b>	
63	Add more user cases	2 days	Mon 15/06/15	Tue 16/06/15		HoangNDH	
64	<b>Analysis and design</b>	<b>11 days</b>	<b>Wed 17/06/15</b>	<b>Wed 01/07/15</b>	<b>62</b>		
65	Database design	<b>4 days</b>	<b>Wed 17/06/15</b>	<b>Mon 22/06/15</b>			
66	Design Database for web service	1 day	Wed 17/06/15	Wed 17/06/15		NghiaPT, HoangNDH	
67	Design database for dictionary	2 days	Thu 18/06/15	Fri 19/06/15		NghiaPT, ThaiVH	
68	Review and Re-design database for dictionary	1 day	Mon 22/06/15	Mon 22/06/15		Instructor[10%], NghiaPT, T	
69	Approve on database design	0 days	Mon 22/06/15	Mon 22/06/15	<b>68FF</b>	NhungBH, ThaiVH, Instructor	
70	<b>Architecture design</b>	<b>2 days</b>	<b>Tue 23/06/15</b>	<b>Wed 24/06/15</b>			
71	Design for web service	1 day	Tue 23/06/15	Tue 23/06/15		ThaiVH, HoangNDH[50%]	
72	Design for application	1 day	Wed 24/06/15	Wed 24/06/15	<b>63,69,71</b>	NghiaPT, ThaiVH	
73	<b>Interface design</b>	<b>7 days</b>	<b>Tue 23/06/15</b>	<b>Wed 01/07/15</b>			
74	Design Dictionary, History screen	3 days	Tue 23/06/15	Thu 25/06/15		BinhLT	
75	Re-design application screen	3 days	Mon 29/06/15	Wed 01/07/15		BinhLT	
76	<b>Implementation</b>	<b>19 days</b>	<b>Tue 16/06/15</b>	<b>Fri 10/07/15</b>			
77	<b>Test Tesseract library</b>	<b>6 days</b>	<b>Tue 16/06/15</b>	<b>Tue 23/06/15</b>			
78	Develop tool for creating text image	3 days	Tue 16/06/15	Thu 18/06/15		NhungBH	
79	Create test data	3 days	Tue 16/06/15	Thu 18/06/15		MaiNT	
80	Test Tesseract with created image set	1 day	Fri 19/06/15	Fri 19/06/15	<b>78,79</b>	NhungBH	
81	<b>Summarize test result</b>	1 day	Mon 22/06/15	Mon 22/06/15	<b>80</b>	MaiNT, NhungBH	
82	Report test result	0 days	Tue 23/06/15	Tue 23/06/15	<b>81</b>		
83	Develop image processing library	<b>8 days</b>	<b>Mon 29/06/15</b>	<b>Wed 08/07/15</b>			
84	<b>Improve functions to extract text and text line</b>	5 days	Mon 29/06/15	Fri 03/07/15	<b>82</b>	NhungBH	
85	Convert image to threshold image	3 days	Mon 06/07/15	Wed 08/07/15			
86	Develop functions of web service	12 days	Wed 24/06/15	Thu 09/07/15	<b>66,71</b>		
87	Develop application	<b>12 days</b>	<b>Thu 25/06/15</b>	<b>Fri 10/07/15</b>			
88	<b>Code Dictionary screen</b>	12 days	Thu 25/06/15	Fri 10/07/15	<b>69,72</b>	NghiaPT	
89	Code Scan and Result screen	8 days	Tue 30/06/15	Thu 09/07/15	<b>69,83FF+1 day</b>	ThaiVH	
90	Testing	<b>13 days</b>	<b>Wed 24/06/15</b>	<b>Sat 11/07/15</b>			
91	Create ST test case for dictionary	2 days	Wed 24/06/15	Thu 25/06/15	<b>63</b>	MaiNT	
92	Create ST test case for history screen	2 days	Fri 26/06/15	Mon 29/06/15		MaiNT	
93	Create & execute UT test case for image processing library	1 day	Thu 09/07/15	Thu 09/07/15	<b>83</b>	NhungBH	
94	<b>Create &amp; execute UT test case for web service</b>	1 day	Fri 10/07/15	Fri 10/07/15	<b>86</b>	HoangNDH	
95	Create & execute UT test case for Scan function	1 day	Fri 10/07/15	Fri 10/07/15	<b>89</b>	ThaiVH	
96	Create & execute UT test case for Dictionary functions	1 day	Sat 11/07/15	Sat 11/07/15	<b>88</b>	NghiaPT	
97	<b>Report UT result</b>	0 days	Sat 11/07/15	Sat 11/07/15	<b>94,93,96,95</b>	ThaiVH	
98	Evaluation	<b>0.5 days</b>	<b>Mon 13/07/15</b>	<b>Mon 13/07/15</b>	<b>76</b>	BinhLT, HoangNDH, Instructor	
99	Report result of API_ImageProcessor	0.5 days	Mon 13/07/15	Mon 13/07/15		NhungBH	
100	<b>Evaluate application</b>	0.5 days	Mon 13/07/15	Mon 13/07/15			
101	<b>Evaluate web service</b>	0.5 days	Mon 13/07/15	Mon 13/07/15			

Figure 7: Schedule for Increment 2

ID	Task Name	Duration	Start	Finish	Predecessors	Resource Names	04 M.
102	Incremental 3: Complete all functions	14.5 days	Tue 14/07/15	Mon 08/08/15	59		
103	Planning	1 day	Tue 14/07/15	Tue 14/07/15			
104	Review and update PMP	1 day	Tue 14/07/15	Tue 14/07/15		NhungBH	
105	Requirement	3 days	Tue 14/07/15	Thu 16/07/15			
106	Review and update SRS	1 day	Tue 14/07/15	Tue 14/07/15		HoangNDH	
107	Update Introduction document	2 days	Wed 15/07/15	Thu 16/07/15		NhungBH	
108	Design	12 days	Tue 14/07/15	Wed 29/07/15			
109	Interface design	12 days	Tue 14/07/15	Wed 29/07/15			
110	Re-design interface for application	2 days	Tue 14/07/15	Wed 15/07/15		BinhLT	
111	Design interface for web service	2 days	Wed 22/07/15	Thu 23/07/15		BinhLT	
112	Create screen design document	3 days	Mon 27/07/15	Wed 29/07/15		BinhLT	
113	Submit screen design document to PM	0 days	Wed 29/07/15	Wed 29/07/15	112FF	BinhLT	
114	Implementation	13 days	Wed 15/07/15	Fri 31/07/15			
115	Web service	10 days	Wed 15/07/15	Tue 28/07/15			
116	Fix error in changing password	3 days	Wed 15/07/15	Fri 17/07/15		HoangNDH	
117	Complete web service' functions	3 days	Mon 20/07/15	Wed 22/07/15		HoangNDH	
118	Implement interface for web service	3 days	Fri 24/07/15	Tue 28/07/15	111,117	HoangNDH	
119	Application	12 days	Thu 16/07/15	Fri 31/07/15			
120	Complete Dictionary screen	6 days	Thu 16/07/15	Thu 23/07/15	110	NghiaPT	
121	Complete History screen	5 days	Fri 24/07/15	Thu 30/07/15	120	NghiaPT	
122	Complete result screen	6 days	Thu 16/07/15	Thu 23/07/15		ThaiVH	
123	Code setting screen	5 days	Mon 27/07/15	Fri 31/07/15		ThaiVH	
124	Add functions to crop selected text area	2 days	Tue 28/07/15	Wed 29/07/15		NhungBH	
125	Testing	14 days	Tue 14/07/15	Sun 02/08/15			
126	Review and update test plan	1 day	Tue 14/07/15	Tue 14/07/15		MaiNT	
127	Submit test plan	0 days	Tue 14/07/15	Tue 14/07/15	126FF	MaiNT	
128	Add ST test case for application	3 days	Wed 15/07/15	Fri 17/07/15		MaiNT	
129	Create ST test case for web service	3 days	Mon 20/07/15	Wed 22/07/15	128	MaiNT	
130	ST web service	2 days	Thu 23/07/15	Fri 24/07/15	129	MaiNT	
131	Report web service test result	0 days	Fri 24/07/15	Fri 24/07/15	130FF	MaiNT	
132	Create & execute UT Result screen's functions	0.5 days	Fri 24/07/15	Fri 24/07/15	122	ThaiVH	
133	Create & execute UT History functions	0.5 days	Fri 31/07/15	Fri 31/07/15	121	NghiaPT	
134	Create & execute UT Dictionary functions	0.5 days	Fri 31/07/15	Fri 31/07/15	120	NghiaPT	
135	Create & execute UT functions of cropping text area	0.5 days	Thu 30/07/15	Thu 30/07/15	124	NhungBH	
136	Create & execute UT Setting functions	0.5 days	Sat 01/08/15	Sat 01/08/15	123	ThaiVH	
137	Create IT test case	2 days	Thu 30/07/15	Fri 31/07/15		BinhLT	
138	Execute IT test cases	1 day	Sat 01/08/15	Sat 01/08/15	123	MaiNT, BinhLT	
139	Report UT result	0.5 days	Sun 02/08/15	Sun 02/08/15		NghiaPT, ThaiVH	
140	Report IT result	0 days	Sun 02/08/15	Sun 02/08/15	138	MaiNT	
141	Evaluation	0.5 days	Mon 03/08/15	Mon 03/08/15		BinhLT, HoangNDH, Instruc	
142	Evaluate web service	0.5 days	Mon 03/08/15	Mon 03/08/15			
143	Evaluate application	0.5 days	Mon 03/08/15	Mon 03/08/15			

Figure 8: Schedule for Increment 3

ID	Task Name	Duration	Start	Finish	Predecessors	Resource Names	04 M M
144	Increment 4: Refine	10 days	Sat 01/08/15	Sat 15/08/15			
145	<b>Planning</b>	1 day	Tue 04/08/15	Tue 04/08/15			
146	Review and Update plan for Increment 4	1 day	Tue 04/08/15	Tue 04/08/15		NhungBH,BinhLT,HbangND	
147	Implementation	10 days	Sat 01/08/15	Fri 14/08/15			
148	<b>Complete all document</b>	3 days	Wed 05/08/15	Fri 07/08/15			
149	<b>Create user manual</b>	2 days	Wed 05/08/15	Thu 06/08/15		NhungBH	
150	Review and Complete database design document	2 days	Wed 05/08/15	Thu 06/08/15		NghiaPT	
151	Review and Complete SDD	3 days	Wed 05/08/15	Fri 07/08/15		ThaiVH	
152	Refine application	10 days	Sat 01/08/15	Fri 14/08/15			
153	<b>Update result screen</b>	2 days	Sat 01/08/15	Sun 02/08/15		ThaiVH	
154	Fix bugs detected by testing	5 days	Mon 10/08/15	Fri 14/08/15		NghiaPT,ThaiVH	
155	Testing	3 days	Wed 05/08/15	Fri 07/08/15			
156	<b>System test: Test application</b>	3 days	Wed 05/08/15	Fri 07/08/15		BinhLT,MaiNT	
157	Report application test result	0 days	Fri 07/08/15	Fri 07/08/15	156FF	MaiNT	
158	Submit final report of testing	0 days	Fri 07/08/15	Fri 07/08/15		NhungBH	
159	Create final report	4 days	Wed 12/08/15	Sat 15/08/15			
160	<b>Create final report</b>	2 days	Mon 10/08/15	Tue 11/08/15		NhungBH	
161	Get Instructor's approve on final report	0.5 days	Thu 13/08/15	Thu 13/08/15	148,157	Instructor[10%]	
162	Submit final report to FPT university	0 days	Fri 14/08/15	Fri 14/08/15	161	NhungBH	
163	Closing	1 day	Mon 17/08/15	Mon 17/08/15			
164	Presentaion					BinhLT,HoangNDH,MaiNT	

Figure 9: Schedule for Increment 4 and Closing phase

## 5. COMMUNICATION MANAGEMENT

### 5.1. Stakeholders and Contacts

Name	Title	Role	Contacts
Tran Binh Duong	Mr.	Supervisor	<a href="mailto:duongtb@fpt.edu.vn">(84) 936-168-165</a>
Bui Hong Nhung	Mrs.	PM	<a href="mailto:Nhungbhse02082@fpt.edu.vn">(84) 166-7762-099</a>
Vu Hong Thai	Mr.	Technical Leader	<a href="mailto:Thaivhse02654@fpt.edu.vn">(84) 169-820-4900</a>
Le Thanh Binh	Mr.	Main Designer	<a href="mailto:Binhlnse02917@fpt.edu.vn">(84) 914-465-993</a>
Nguyen Duc Huy Hoang	Mr.	Developer	<a href="mailto:Hoangndhse02651@fpt.edu.vn">(84) 965-957-811</a>
Pham Trong Nghia	Mr.	Developer	<a href="mailto:Nghiaaptse02781@fpt.edu.vn">(84) 982-822-412</a>
Ngo Thi Mai	Mrs.	Test Leader	<a href="mailto:Maintse02218@fpt.edu.vn">(84) 966-656-692</a>

## 5.2. Communication Management Approach

Project team communicate frequently to ensure the progress of each member's work. Project manager report to the Supervisor frequently and honestly so that the Supervisor can track the team's work and give support/advice as need.

All request for change or proposal of new ideal must be discussed in team. If project team agree to change, project team must then discuss with the Supervisor. Once the change is approved, project manager will update the plan and to project team and Supervisor.

The communications requirements are documented in the Communications Matrix of this document. The Communications matrix will be used as guide for what/when/how/who/whom to communicate through out the project.

## 5.3. Communication Requirements

The project manager will communicate with the Supervisor in order to determine his preferred frequency and time of communication.

As all project team members still take part in classes while doing project, project manager should communicate to understand their schedule, and therefore specify appropriate communication plan for the team.

## 5.4. Communication Method and Technologies

- Create a Facebook private group for team member to discuss and share informal information and activity. This would be a place for member to communicate freely, and therefore would help in strengthen relationship between members.
- Use Google docs to share documents
- Public weekly reports of team members in Google docs to keep them writing report. This will help project manager in tracking work of team members, and also help team member understand progress of the others.
- Update Work schedule before weekly meeting with Supervisor

## 5.5 Communication Matrix

Communication Type	Objective	Medium	Frequency	Audience	Owner	Deliverable	Format
Kickoff Meeting	Discuss and agree on project objective, and scope	- Face to face	Once (At the start of the project)	- Instructor - Project Team	Project Manager	- Meeting Minutes	- Soft copies on Microsoft Word
Project Team Meeting	Review status of the project. Discuss solutions for any raised issues. Assign tasks for team members	- Face to face - Email - Conversation (Skype)	3 times per week, base on personal schedule.	- Project Team	Project Manager	- Project schedule - Assigned tasks	- Soft copies on Microsoft Word - Assigned tasks on Trello
Project Plan Meeting	Discuss and planning project process	- Face to face - Email - Conversation (Skype)	As needed	- Project Team	Project Manager	- Meeting Minutes - Project Management Plan	- Soft copies on Microsoft Word - WBS on Microsoft Project
Architecture Design Meeting	Review prototype Discuss, contribute and review System Architecture Design	- Face to face - Email	As needed	- Project Team	Technical Leader	- Meeting Minutes - System Architecture Design Document	- Soft copies on Microsoft Word - System Architecture Design on Astah
GUI Design Meeting	Discuss, contribute and review GUI Design	- Face to face - Email	As needed	- Project Team	Technical Leader	- Meeting Minutes - GUI Design - Document	- Soft copies on Microsoft Word - GUI Design on Photoshop
Database Design Meeting	Discuss, contribute and review Database Design	- Face to face	As needed	- Project Team	Technical Leader	- Meeting Minutes - Database Design	- Soft copies on Microsoft Word

Test Plan Meeting	Discuss, and review Test Plan	- Face to face	As needed	- Project Team	Technical Leader	- Meeting Minutes - Test Plan	- Soft copies on Microsoft Word
Bug Report	Report found bugs to team members and assign fix tasks	- Face to face - Email	As needed	- Project Team	Test Leader	- Bug Report	- Soft copies on Microsoft Excel
Personal Weekly Report	Report task status (what is done, what will be done next week, any issue) of personal work	- Email - Google driver	Every Friday	- Project team	Project Manager	- Personal Weekly Report	- Soft copies on Microsoft Excel
Project team weekly report	Report task status (what is done, what will be done next week, any issue)	- Email	Every Saturday	- Instructor	Project manager	- Weekly report	- Soft copies on Microsoft word
Meeting with Instructor	Report project status of team's work to instructor Get advices for project from instructor	- Face to face - Email	Every Monday	- Instructor - Project Team	Project Manager	- Meeting Minutes - Team Weekly Report	- Soft copies on Microsoft Word
Unexpected Issue	Find a solution for any unexpected raised issues.	- Face to face - Conversation (Skype) - Facebook private group	As needed	- Instructor - Project Team	Project Manager	- Meeting Minutes	- Soft copies on Microsoft Word

## 6. RISK MANAGEMENT<sup>[5]</sup>

### 6.1. Risk Management Approach

The approach to manage risks for this project is the process by which project team identifies and ranks the various risks. The most likely and high impact risks will be added to the risk register and will be delivered to all team members, to ensure that every member perceive of these potential risks. Project manager must pay attention to all items in risk register during project and take appropriate action when a risk triggered. Upon the completion of the project, project manager will analyze each risk as well as the management process. Base on this analysis, project manager will identify any improvements that can be made to risk management process, and capture these improvements as a part of lessons learned.

### 6.2. Risk Identification

Risk identification will be conducted in the first project team meeting. The method used by project team to identify risk will be brainstorming. Every team member raise their idea about any risk that might happen. PM will note down team member's idea. There must be no judgment or bias during brainstorming. When there is no other idea, project team will review all risks that have been noted down. Risks that are almost unlikely to happen will be erased. Risks with high impact or most likely to happen will be kept. There should be no more than 10 risks to be kept.

Besides, project team will review other capstone project in order to determine the most common risks and the strategies used to mitigate those risks.

### 6.3. Risk Qualification and Prioritization

In order to determine the severity of the risks identified by the project team, a probability and impact factor will be assigned to each risk. PM then will prioritize risks based on their probability and impact. Finally, PM will create a probability - impact matrix.

### 6.4. Risk Monitoring

Risk monitoring will be a continuous process throughout the project. Avoidance plan should be taken carefully from start of the project. In case a risk is about to happen, PM will apply contingency plan to prevent risk. If risk is already happening, PM will apply fall back plan to minimize impact.

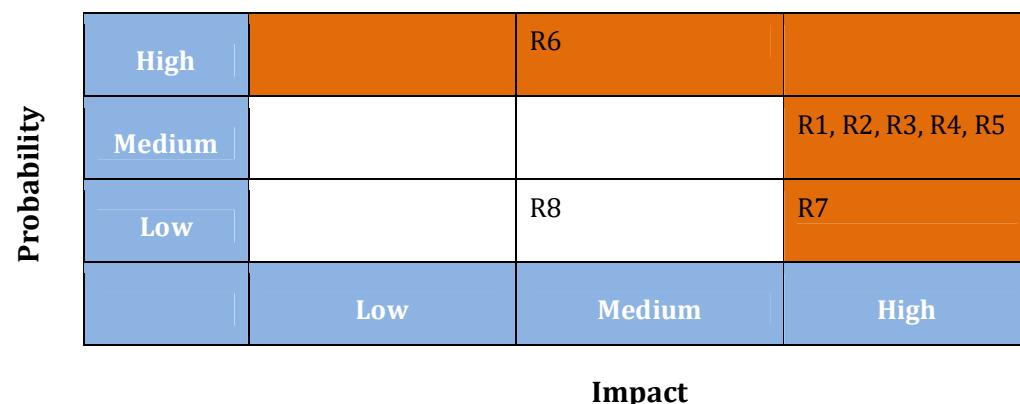
## 6.5. Risk Register

### 6.5.1. Risk description

No.	Risk Name	Description	Category	Route cause	Probability	Impact	Trigger
R1	Lack of skills and ability for a specified work	Team members are unable to complete work because they do not know how to do	Human	Team members have not studied or experienced similar work	Medium	High	Team members take a lot of time to do their task / Team member informed that they can not do the task
R2	Team member conflict	Team member disagree with each others and refuse to work	Human	Unclear requirement specification. Team members do not unify on solutions when problem occurs.	Medium	High	When somebody does not work because of disagreement
R3	Requirement changes	The scope may change, unplanned requirements may get added	Requirement	Request from Supervisor or team member for new function, or to change requirement	Medium	High	When Supervisor or team member request to change requirement
R4	User reject the product	Unfriendly GUI Large for downloading Application is not suitable with market trend	Requirement	Lack of graphic design skill Does not optimize code	Medium	High	User complaint about the application
R5	Team member distraction	Team members do not pay enough time for the project	Human	Team members are busy with studying while doing project	Medium	High	Team members does not spend time for work and tasks are not completed on time

R6	Schedule flaws	Time for one task may too long or too short Trouble may occurs during process that makes task longer than expected	Estimates	The project is planned assuming that all will take the most optimistic path possible Poor plan analysis Poor tem member assessment	High	Medium	Somebody have much time to free and some did not finish their task on time
R7	Unrealistic expectation	Project team expects more than what they can do	Requirement	Team members' abilities are overestimated	Low	High	When nobody known how to reach the goal
R8	Lack of Supervisor support	Supervisor may have not enough time to support project team	Human	Communicate with Supervisor inefficiently Supervisor is busy personal business, and support many team at the same of time	Low	Medium	It takes 2 day or more for mail response. Meeting with Supervisor is hard to arranged.

### 6.5.2. Probability – Impact matrix



### 6.5.3. Risk response plan & Risk status

No.	Risk name	Risk response			Status
		Avoidance plan	Contingency plan	Fall back plan	
R1	Lack of skills and ability for a specified work	Do well at training phase. PM should understand ability of each team member and assign right work for right person.	Assign more team members to support or to do the task.	Ask for support of external interfaces.	Activated / Solved
R2	Team member conflict	Clear requirement. Pay attention on team building activities.	Negotiate to deal with conflict.	Report to Supervisor	Not yet active
R3	Requirement changes	Specify software requirement between Supervisor and project team before start of development. Give special consideration in requirements.	Discuss with Supervisor to decide if project teams can follow old requirement.	Change requirement. Develop for new required functions.	Activated/ Solved
R4	User reject the product	Conduct a market survey Follow coding construction	Listen to users' complaint and find solution	Change requirement	Not yet active
R5	Team member distraction	Understand team members' schedule and assign suitable tasks and set appropriate deadlines. Require team member to set high priority for the project.	Arrange team work frequently. Sketch deadline for the assigned tasks.	Assign task to other member.	Activated / Solved
R6	Schedule flaws	Get to known team member skill Use Microsoft Project tool	Member who has free time involve in and support to finish	Create new time schedule	Activated / Solved

		Add slack time for importance task	task If there are some task that has more time than usual, member who take responsibilities should help each other		
R7	Unrealistic expectation	Understand members' ability. Set appropriate goal and all team member should agree on common goal	Remove unnecessary functions, and focus on key functions	Change project	Activated / Solved
R8	Lack of Supervisor support	Define a meeting schedule with Supervisor Use communication medium effectively	Reference from other source.	Ask for support of other interfaces.	Not yet active

## 7. QUALITY MANAGEMENT

### 7.1. Quality Management Overview

#### 7.1.1. Organization, Responsibilities, and Interfaces

Name	Role	Responsibilities
Tran Binh Duong	Supervisor	<ul style="list-style-type: none"> <li>- Helps define product quality expectations.</li> <li>- Determines final acceptance of product's quality.</li> </ul>
Bui Hong Nhung	PM	<ul style="list-style-type: none"> <li>- Create quality plan</li> <li>- Facilitate resolution of quality issues, escalating as needed</li> </ul>
Ngo Thi Mai	Test leader	Provide test and test management
Team members: - Vu Hong Thai - Pham Trong Nghia - Nguyen Duc Huy Hoang - Bui Hong Nhung	Developer	<ul style="list-style-type: none"> <li>- Provide feedback on quality plan, help determine metrics and criteria for this project</li> <li>- Be a part of quality reviews and provide feedback on deliverables</li> </ul>

#### 7.1.2. Tools, Environment, and Interfaces

Tool	Description
Cause-and-effect diagram	Used to find the root cause problem when there is a complaint about quality problem.
Flowchart	Used to illustrate a solution model for a problem.
Control chart	Used to detect logical error of functions. Apply Seven Run Rule.

### 7.2. Quality Planning

#### 7.2.1. Define Project Quality

- **System output:**
  - + A server to manage dictionary package and provide download service for the application
  - + An Android application supports end-users in scanning Japanese document and searching for words' meaning.

- ***Functionality:***
  - + Server allow admin / root admin to login, upload / edit / delete dictionary package. It also provide service to download dictionary package.
  - + Android application has main functions: scans and recognize Japanese characters, translates, searches words in dictionary, and downloads dictionary packages from server.
- ***Performance:***
  - + Time delay for image processing is less than 10s
  - + Other functions of server perform well while dictionary package is downloading to the application
- ***Reliability:***
  - + The application is available 24/7
  - + Server can handle at least 1000 clients concurrently
  - + The accuracy of Japanese detection is at least 80%
- ***Maintainability:***
  - + Android application is easily to be updated without any crashes. Source code is readability, complies with coding convention.
  - + System has to be designed to be easy to extend.
- ***Security***
  - + Information of admins / root admins on server is secured

### 7.2.2. Measure Project Quality

Metric	Goal
Accurate	<ul style="list-style-type: none"> <li>- The accuracy of text line detection is at least 80%</li> <li>- The accuracy of text recognition is at least equal Tesseract accuracy for image with out pre-processed.</li> </ul>
Response of mobile application	Time delay for image processing <= 10s
Bugs/Lines of Code	UT: 8 – 9 bugs / KLOC ST: 2 – 4 bugs / KLOC <i>(based on Fsoft norms)</i>
Maximum depth of loops	<= 4
Android Program Size	<= 100 MB
Algorithm complexity	<= O(n^2)

Android version support	Support Android version 4.4.2 Kitkat to 5.0.1 Lollipop
Android screen support	Multiscreen Must be tested on 320x480, 768x1024, 768x1336, 1080x1920 screens
Website support browser	Support Chrome version 41.0.xxx, Firefox version 36.0 or later.

## 7.3. Quality Assurance

### 7.3.1. Analyze Project Quality

Milestone	Deliverables	Goal	Review and Approved
5/06/2015	Interface design ver1.0	- Good looking & easy-to-use - Cover all functions specified in SRS	HoangNDH MaiNT
5/06/2015	Software architecture design ver1.0	Design to be easy to extend	Supervisor
23/06/2015	Result of testing Tesseract library	- Tested with Hiragana, Furigana, Kanji - Tested by single word and sentence - Tested with different text size	ThaiVH Supervisor
22/07/2015	Server	- Reveal API for use. - Information of admins / root admins on server is secured	MaiNT ThaiVH
23/07/2015	Dictionary screen	Time delay for searching a word <= 1s	MaiNT
23/07/2015	Result screen	The accuracy of text recognition is a least equal Tesseract accuracy for image with out pre-processed.	MaiNT
29/07/2015	Image processing library	- The accuracy of text line detection is at least 80% for image with text only - Text line extraction is <= 10s	ThaiVH MaiNT
2/08/2015	Integration test report	30 – 34 test cases / KLOC 2 – 4 bugs / KLOC	NhungBH
7/08/2015	System test report	30 – 34 test cases / KLOC 2 – 4 bugs / KLOC	NhungBH

### 7.3.2. Improve Project Quality

Issue	Action
Difficult to track project's progress	<ul style="list-style-type: none"> <li>- Weekly report, team work 3 days / week</li> <li>- Using Trello to track team members' work</li> </ul>
Maintainability	<ul style="list-style-type: none"> <li>- Specify coding conventions document</li> <li>- Concentrate on architecture design</li> </ul>
Low quality code	<ul style="list-style-type: none"> <li>- Unit Test</li> <li>- Peer review, peer coding among developers</li> <li>- Using Open Source and Framework: OpenCV, Tesseract, Spring framework</li> </ul>
Hard to gain 80% accuracy of text line extraction	Early study algorithms and combines some algorithms together to get higher precision
Reward and discipline	<ul style="list-style-type: none"> <li>- Teambuilding to increase communication ability between project's members</li> <li>- Have punishment rules when: <ul style="list-style-type: none"> <li>+ Submit terrible code (which causes to re-coding more than 10%)</li> <li>+ Miss deadline</li> </ul> </li> </ul>
Acceptance of users	Do survey to discovery what features user want from this projects. Do it before design progress.

### 7.4. Quality Control

Deliverables	Goal	Quality control activity	Frequency / Interval
Interface design	<ul style="list-style-type: none"> <li>- Good looking &amp; easy-to-use</li> <li>- Cover all functions specified in SRS</li> </ul>	<p>Designer has to:</p> <ul style="list-style-type: none"> <li>- Ask for advice of some other designer</li> <li>- Take comment from friends on completed work</li> <li>- Have approve from SRS leader</li> </ul>	Each time design a new screen
Software architecture design	Design to be easy to extend	Have review and judgment from Supervisor	On completion
Result of testing	<ul style="list-style-type: none"> <li>- Tested with Hiragana,</li> </ul>	Have approve from	On completion

Tesseract library	Furigana, Kanji - Tested by single word and sentence - Tested with different text size	Test leader and Technical leader	
Server	- Reveal API for use - Information of admins / root admins on server is secured	Testers execute security test	On completion
Dictionary screen	Time delay for searching a word <= 1s	PM requires developer to report to technical leader about method to be used	Report before and after coding
Result screen	The accuracy of text recognition is at least equal Tesseract accuracy for image with out pre-processed.	Technical leader , test leader review and evaluate image processing algorithm	Every new version of image processing library
Image processing library	- The accuracy of text line detection is at least 80% for image with text only - Text line extraction is <= 5s	Technical leader , test leader review and evaluate image processing algorithm	Every new version of image processing library
Integration test report	30 – 34 test cases / KLOC 2 – 4 bugs / KLOC	PM requires testers to report on work	Weekly
System test report	30 – 34 test cases / KLOC 2 – 4 bugs / KLOC	PM requires testers to report on work	Weekly

## 7.5. Action Plan

What	Who	When	How	Output
Define coding convention	NhungBH	28/05/2015	Read standard coding convention of Oracle Extract and modify to reuse it	Coding Convention ( <i>10.1 in this document</i> )

Training Trello	All team	13/05/2015	ThaiVH guides team members how to use Trello	
Market research	NhungBH NghiaPT MaiNT	18/05/2015 to 22/05/2015	Do survey to find out what functions users need	Survey summary
Traning OpenCV	NhungBH ThaiVH HoangND	13/05/2015 to 21/05/2015	<ul style="list-style-type: none"> <li>- Read OpenCV documents</li> <li>- Find out how to set up and complete 'Hello World' tutorial.</li> <li>- Try some core class and functions for image processing.</li> </ul>	<ul style="list-style-type: none"> <li>- OpenCV Training Report</li> <li>- HelloWorld Source Code Example</li> </ul>
Research for image processing algorithms	NhungBH	8/06/2015 to 8/07/2015	<ul style="list-style-type: none"> <li>- Research many algorithms includes:           <ul style="list-style-type: none"> <li>+ Detect contours</li> <li>+ Remove noise</li> <li>+ Convert image to threshold</li> </ul> </li> <li>- Actively discover and evaluate each algorithm and decide what to use</li> <li>- Get Advice from Supervisor and Experts</li> </ul>	Report on image processing library (steps, algorithm)
Training Unit Test	ThaiVH NhungBH HoangND NghiaPT	13/05/2015	Training how to create and perform Unit Test	Training Report

## III. SOFTWARE REQUIREMENT SPECIFICATION

### 1. INTRODUCTION

#### 1.1 Purpose

This document provides developers, testers, QAs a complete and comprehensive description of both the functional requirement and non-functional requirements of the Anki Pan System. Developers base on this document to develop the system, and testers base on this to assure the quality of the output system. Project manager base on this document to create schedule and assign task to team members.

#### 1.2. Scope

This document define all functional and non functional requirements for Anki Pan system.

#### 1.3. Overview

Section 2 of this document provides overall information of the project, while excluding the specific requirements. Instead, it provides the background for those requirements, which are defined in Section 3

Section 3 describes all the requirements in detail, including functional requirements and non-functional requirements. Each function is described based on the following format:

- Use Case: the detailed description of all use cases
- Screen: The mockup/prototype of the screen which describes the workflow of the Use Case

## 2. OVERALL DESCRIPTION

### 2.1. Scope of work

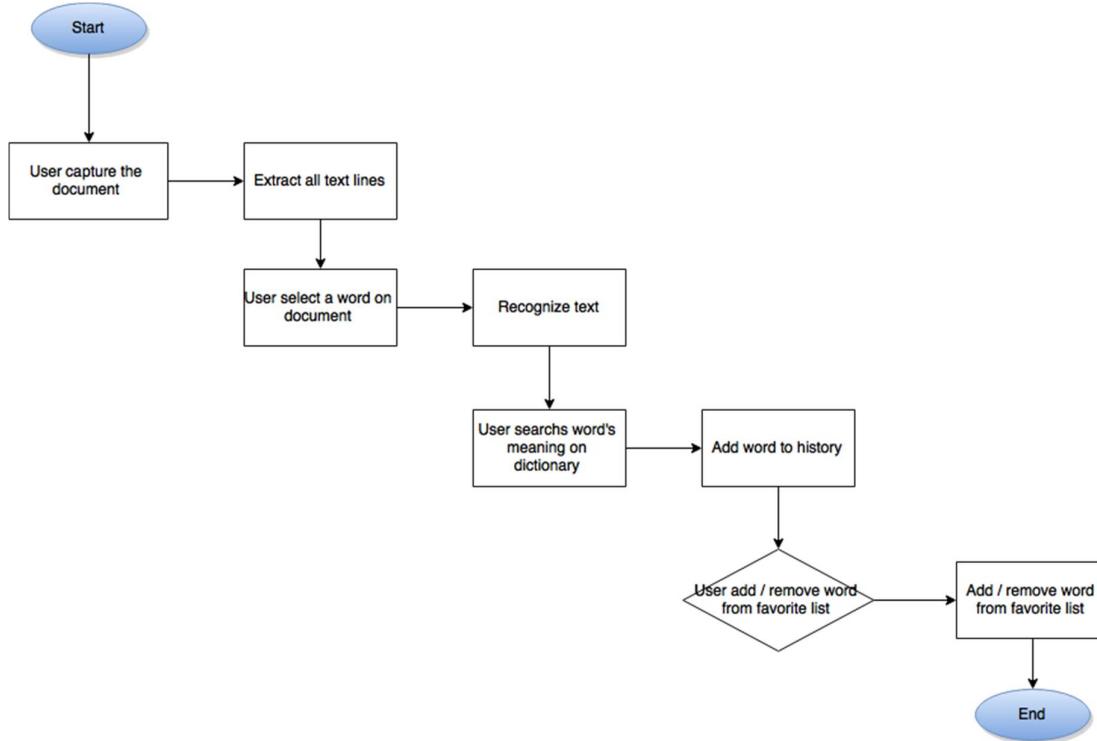
Project team will develop an Android Application for user to scan Japanese document and search words' meaning, and a server to manage dictionary database and provide dictionary download service for the Application.

The Application includes below main functions:

- Capture image
- Detect Japanese characters in image captured by users ↴
- Select text in captured image
- Search for meaning of Japanese words and Kanji in dictionaries ↴

- Download dictionary: allow user to download dictionary package, each package include a Japanese dictionary and a Kanji dictionary.
- Save words to history
- Bookmark words

The diagram below shows the flow of interact between user and application from capturing document to search word in dictionary.



*Figure 10: Flow of scan image and searching word*

The server is developed for admins of the system, with below main functions:

- Login as admin or root admin
- Logout
- Manage user' profile (add, delete, edit information)
- Upload dictionary packages, each package includes 2 kind of dictionaries: Japanese dictionary and Kanji dictionary
- Manage dictionary packages (edit, delete)

Project team use open source for Japanese recognition. Dictionary database will be download from Internet.

## 2.2. Assumptions

N/A

### 3. FUNCTIONAL REQUIREMENTS

#### 3.1. Android application

##### 3.1.1. User Case Diagram

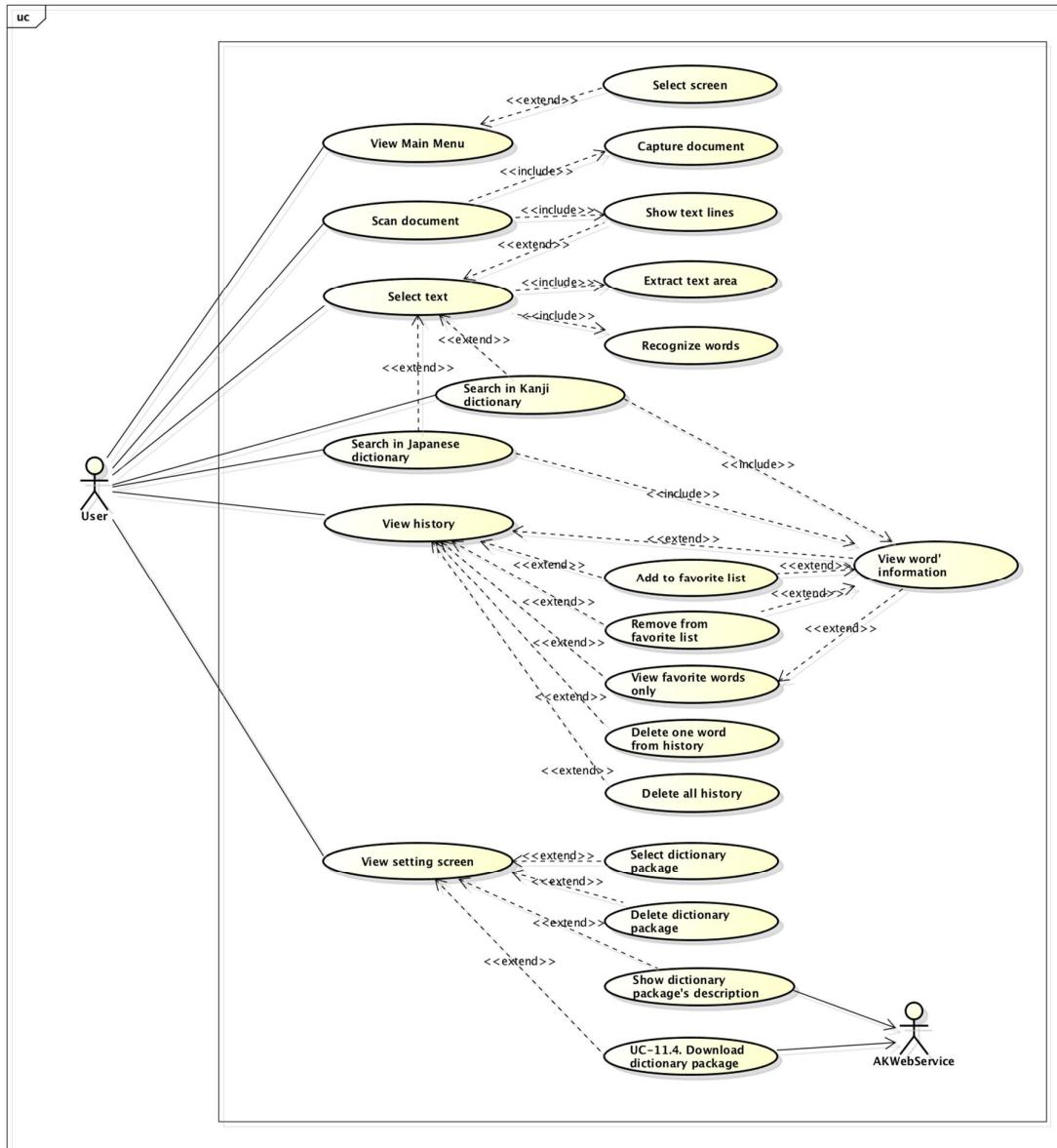


Figure 11: User case diagram of Application

### 3.1.2. Functions

#### 3.1.2.1. Main menu

##### 3.1.2.1.1. User cases

UC - 1. View Main Menu

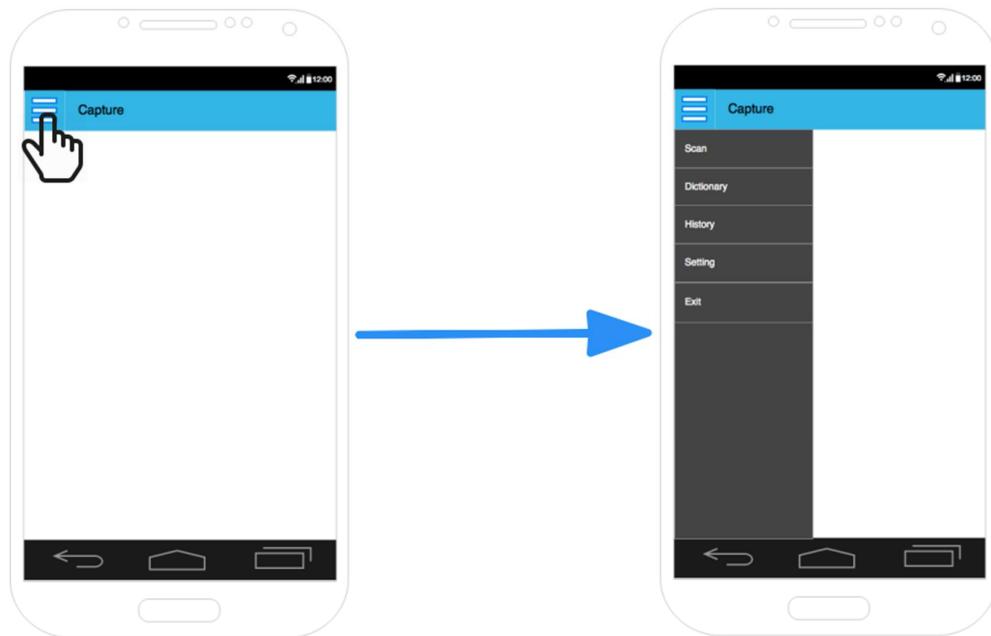


Figure 12: Screens of UC - 1

Use Case ID	UC-01	
Use Name	View main menu	
Actor	User	
Description	User views list of Screens in Anki Pan application	
Precondition	N/A	
Trigger	N/A	
Post-Condition	View list of Screens in Anki Pan application	
Main flow		
Step	Actor	Action
1	User	Tap the Menu icon on the top left corner of the screen
2	System	Show list of Screens include: 1. Scan 2. Dictionary 3. History 4. Setting 5. Exit Each item go with its own icon

Alternative flow	
Exceptions	N/A

UC – 1.1 Select screen

Use Case ID		UC - 1.1
Use Name		Select screen
Actor		User
Description		User selects a screen
Precondition		N/A
Trigger		N/A
Post-Condition		View screen that user selected
Main flow		
Step	Actor	Action
1	User	Tap on the entry of the screen in the Menu list that he want to open
2	System	If user tap on Scan, direct to scan screen If user tap on Dictionary, direct to tab Result of Dictionary screen If user tap on History, direct to tab Favorite of History screen, and show all history If user tap on Setting, direct to tab Dictionary Setting of Setting screen
Alternative flow		
Exceptions		N/A

**3.1.2.1.2. Screen description**

Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
Header	Include Main menu icon and the title of the application, as describe below.	-	-		-	-
Title bar	Show application name: Anki Pan	-	-	Label	-	-
Menu button	The main menu button, tapping on this button will extend the menu list from the left of the screen, tapping on it again will hide the menu list	-	-	Button	-	-

Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
Menu items list	The list of the main menu	-	-	List	-	-

### 3.1.2.2. Scan

#### 3.1.2.2.1. User cases

UC - 2. Scan document

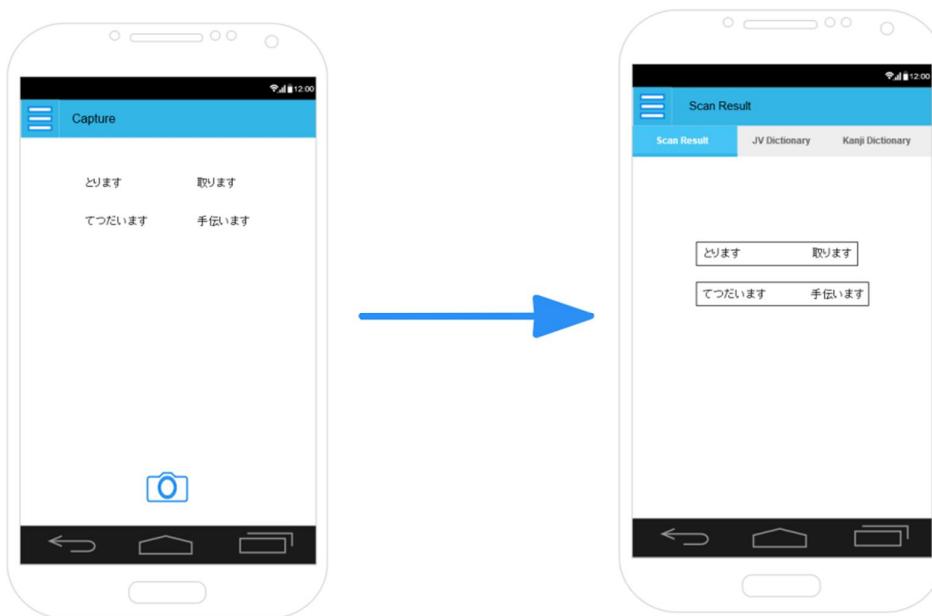


Figure 13: Screens of UC - 2

Use Case ID	UC - 2	
Use Name	Scan document	
Actor	User	
Description	User scans a Japanese document	
Precondition	User select Scan screen in UC - 1.1	
Trigger	N/A	
Post-Condition	View document with all text lines have been surrounded by rectangles	
Main flow		
Step	Actor	Action
1	System	Show scan screen with a line in the middle and Camera button
2	User	User bases on the line shown on the screen to adjust

		he smart phone so that the document is not so skew User presses the Camera button
3	System	Direct to Result tab of the Dictionary screen Show the document with all text lines have been surrounded in rectangles
Alternative flow		
Precondition		
1	User	In step 2 of main flow, user taps the Menu icon and selects another screen
2	System	Direct to the screen that user selected If user selects Dictionary screen and he had scan a document before this, view the Result tab with the last status
Exceptions		N/A

### 3.1.2.2.2. Screen description

Description of Scan screen:

Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
Header	Include Menu icon and the title of the application	-	-		-	-
Camera button	When tapping on this button, the system will capture image and perform image processing	-	-	Button	-	-
Tab bar	The tab bar of the dictionary feature, contain 3 items: Scan Result, JV Dictionary, Kanji Dictionary	-	-	Button	-	-

### 3.1.2.3 Dictionary

#### 3.1.2.3.1 User cases

##### UC – 3 Select text

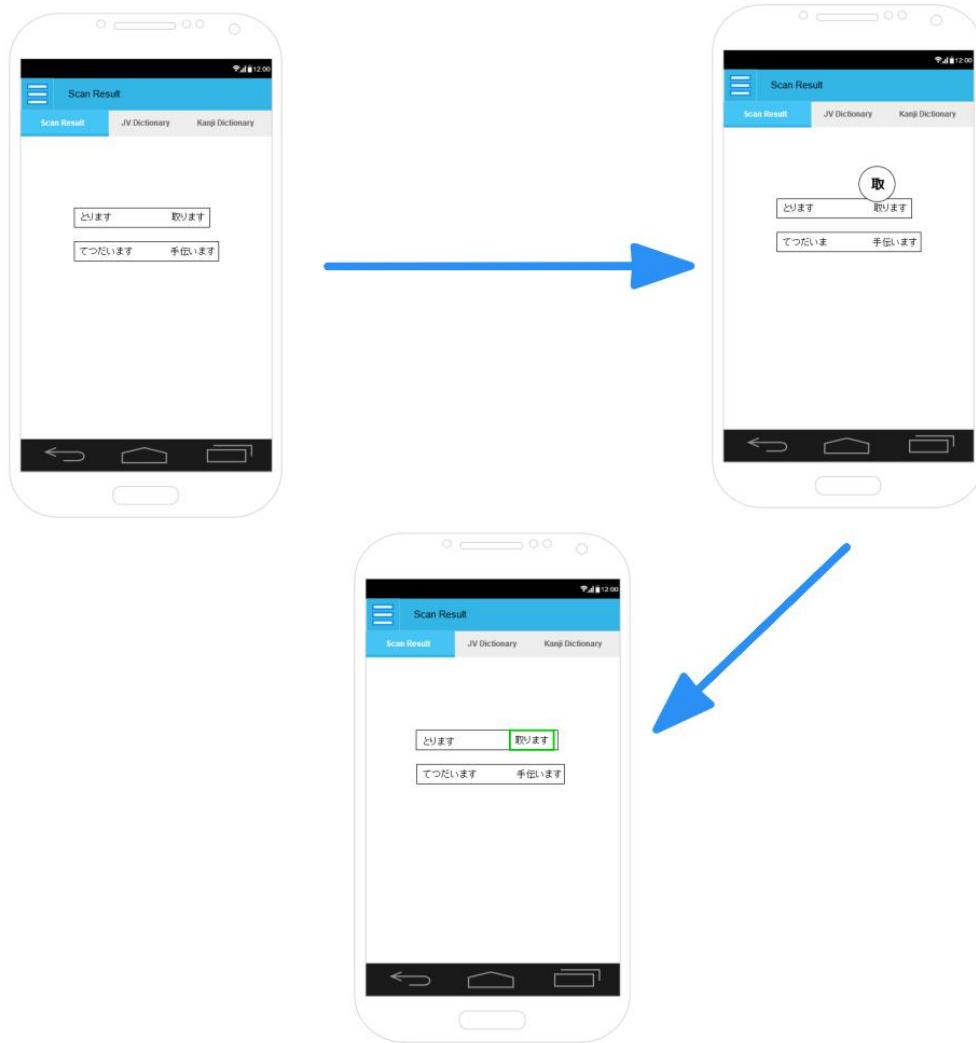


Figure 14: Screens of UC – 3

Use Case ID	UC – 3	
Use Name	Select text	
Actor	User	
Description	User selects text on scanned image	
Precondition	UC – 2	
Trigger	N/A	
Post-Condition	Selected text area is surrounded by a rectangle and selected text is written on the text bar on top of screen	
Main flow		
Step	Actor	Action

1	User	Tab finger on the first point of the text area that he want to select and move the finger until the last point of the text area.
2	System	View a Manifier with the zoomed out image of the area that user's finger is on while his finger is moving on the screen so that user can see the text clear in case the text on the document is small.
3	System	When user lifted up his finger, extract the exact text area that user want to choose. User may missed some strokes of the first or last characters in the selected area. System check to extract the area that include all characters that user may intended to choose.
4	System	Surround the exact text area by a rectangle
5	System	Recognize text in the exact selected area and write to the text field in top of the screen.
<b>Alternative flow</b>		
Precondition		UC-02
1	User	Just makes one tap on the screen
2	System	Do nothing. Keep the screen on current status
Exceptions		N/A

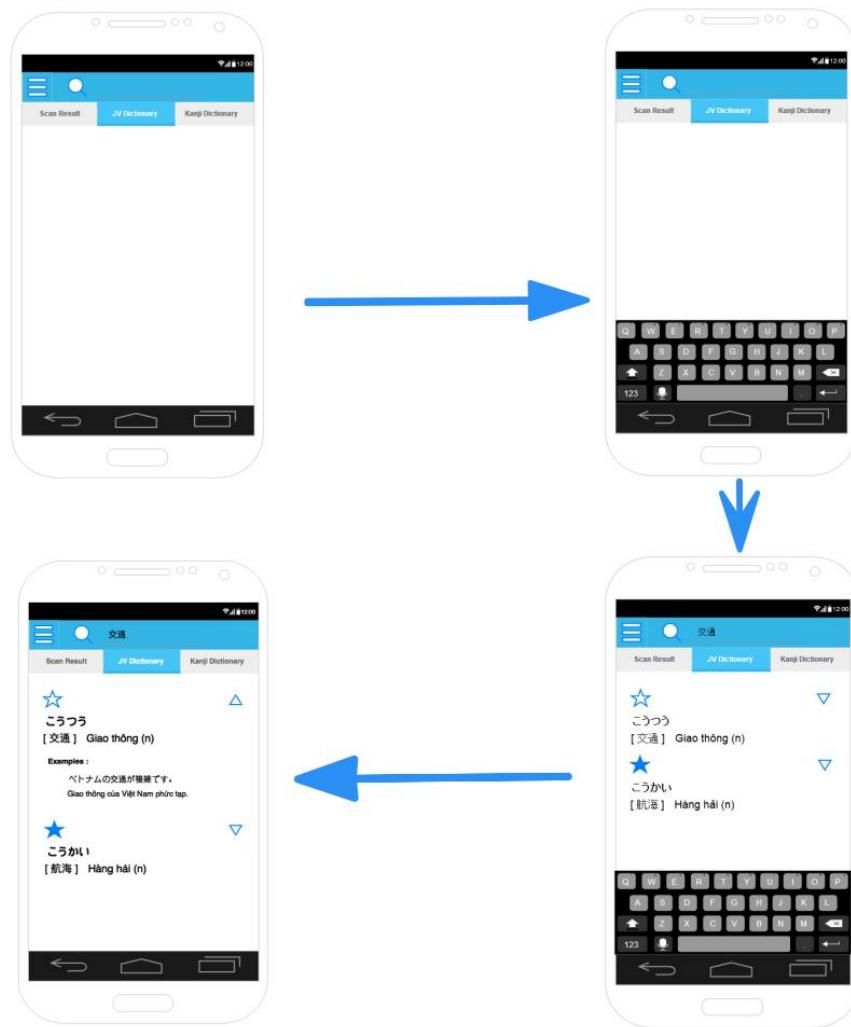
UC - 4 Search in Japanese dictionary

Figure 15: Screens of UC - 4' main flow

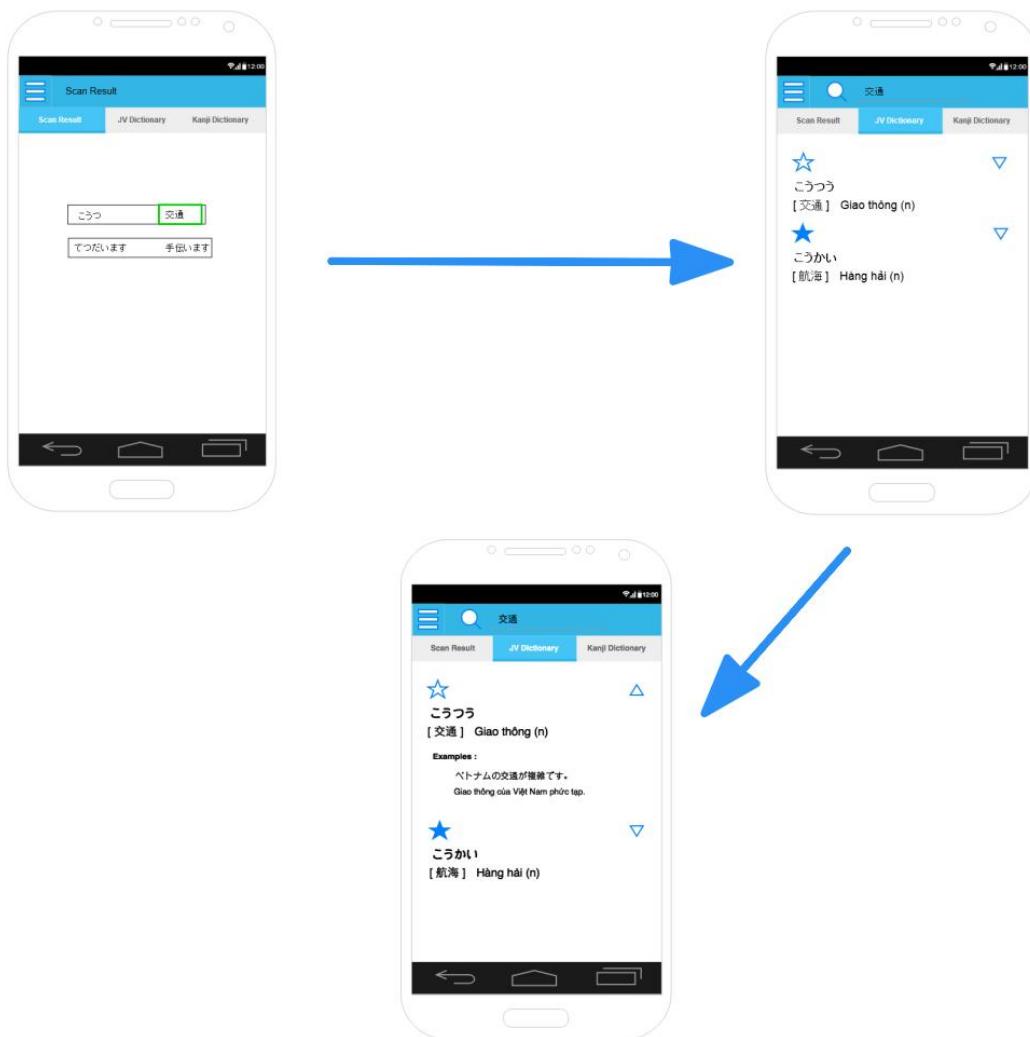


Figure 16: Screens of UC - 4' alternative flow 1

Use Case ID		UC - 4
Use Name		Search in Japanese dictionary
Actor		User
Description		User searches for word's meaning in Japanese dictionary
Precondition		System is in dictionary screen
Trigger		N/A
Post-Condition		Show information of the word Save the searched word, if it is available in the dictionary, to History
Main flow		
Step	Actor	Action
1	User	Tap to open tab Japanese dictionary
2	User	Tap on the magnifier icon in the right of the text field on top of the screen

3	System	Show keyboard
4	User	Type the word his want to search, then press Search key
5	System	<p>Search the word in text field in Japanese dictionary</p> <ul style="list-style-type: none"> <li>- If the word are Hiragana / Katakana / Kanji: Search all the words which have the first part matchs the searching word. For example: search for 交通 will return</li> <li>- If the word are Romaji: Search all the words which's Hiragana writting have the first part matchs the searching word. For example: search for Aki will return 空き (あき) 、あきあきました、秋風 (あきかぜ)</li> </ul>
6	System	Show a list of output words in form of extendable list view, each entry includes a Star icon, word, its reading, the first meaning and word class.
7	User	Tap on the Extend Icon of a entry
8	System	<p>Extend the entry and view full information of the word (word, star icon, Hiragana writting, meaning, example...)</p> <p>Save the word in this entry to History</p>
Alternative flow 1		
Precondition		System is Dictionary screen
1	User	User selects text in scanned document as in UC – 3
2	System	Write the recognized text in user selected area to the text field in the top of Dictionary screen
3	User	Tap the tab Japanese dictionary
4	System	View Japanese dicitonary tab.
5	System	Follow steps 5 to 8 in main flow
Alternative flow 2		
Precondition		User has typed a word and searched in Kanji dictionary tab
1	User	Tap the tab Japanese dictionary
2	System	Follow step 5 to 8 of the main flow
Alternative flow 3		
Precondition		
1	User	After step 8 of the main flow, user tap the Extend icon of the extending entry
2	System	Change the entry to the short form with Star icon, word and a brief information
Exceptions		In step 6 of main flow, if there is no word found, display message "There is no result!"

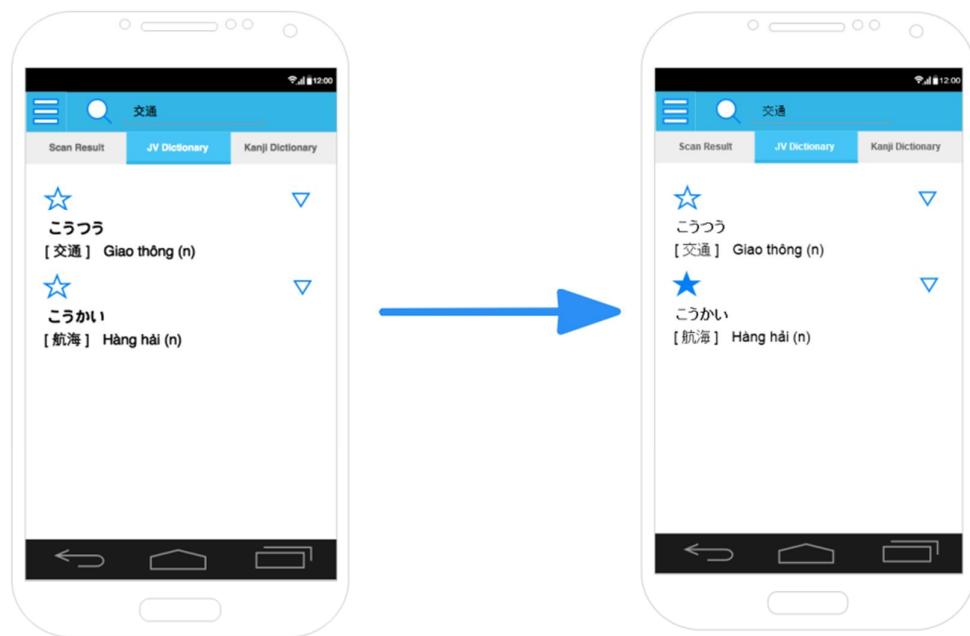
*UC - 4.1 Add to favorite list (Japanese dictionary screen)*

Figure 17: Screens of UC - 4.1

Use Case ID		UC – 4.1
Use Name		Add to favorite list
Actor		User
Description		User add a word in Japanese dictionary to favorite list
Precondition		System is in tab Japanese dictionary User has searched a word in Japanese dictionary, and System is showing a list of words, as in steps 6 of main flow of UC - 5
Trigger		N/A
Post-Condition		The word is added to favorite list
Main flow		
Step	Actor	Action
1	User	Tap on the grey start icon in the entry of the word that he want to add to favorite list
2	System	Change the star to yellow Add the word in the entry to favorite list in database
Alternative flow 1		
Precondition		System is Japanese tab User has expanded a entry, as in steps 8 of main flow of UC - 5
1	User	Tap on the grey star icon of the expanding entry
2	User	Change the star to yellow Add the word in the entry to favorite list

Exceptions	
------------	--

UC – 4.2 Remove from favorite list (Japanese dictionary screen)

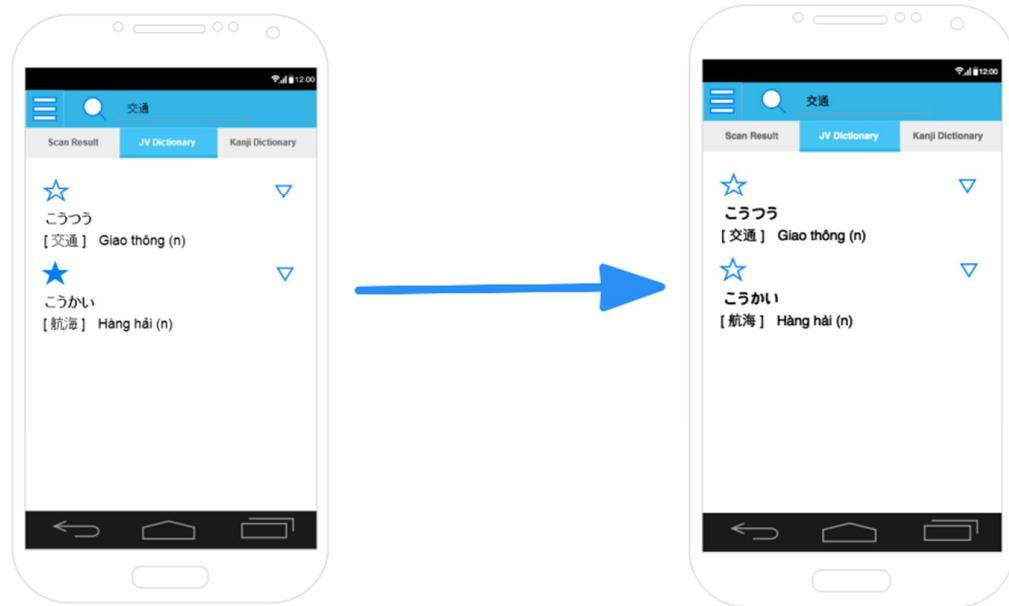


Figure18: Screens of UC - 4.2

Use Case ID	UC – 4.2	
Use Name	Remove from favorite list	
Actor	User	
Description	User removes a word in Japanese dictionary form favorite list	
Precondition	System is in tab Japanese dictionary User has searched a word in Japanese dictionary, and System is showing a list of words, as in steps 6 of main flow of UC - 5	
Trigger	N/A	
Post-Condition	The word is removed from favorite list	
Main flow		
Step	Actor	Action
1	User	Tap on the yellow start icon in the entry of the word that he want to add to favorite list
2	System	Change the star to grey Remove the word in the entry from favorite list
Alternative flow 1		
Precondition		System is Japanese tab User has expanded a entry, as in steps 8 of main flow of UC - 5

1	User	Tap on the yellow star icon of the expanding entry
2	User	Change the star to grey Add the word in the entry to favorite list in database
Exceptions		

UC - 5 Search in Kanji dictionary

Figure 19: Screens of UC - 5' main flow

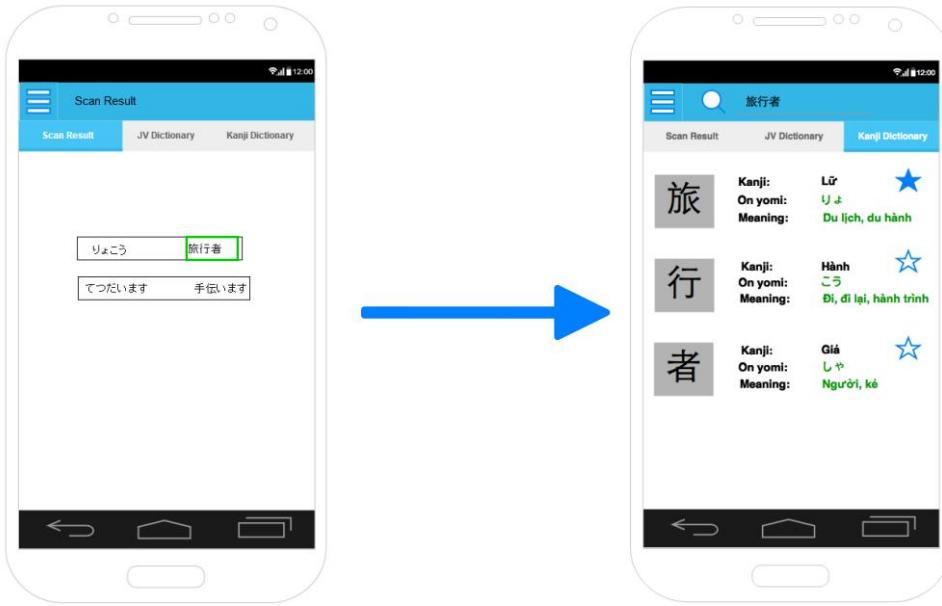


Figure 20: Screens of UC - 5' alternative flow 1

Use Case ID	UC – 5	
Use Name	Search in Kanji dictionary	
Actor	User	
Description	User searches for Kanji information in Kanji dictionary	
Precondition	System is in Dictionary screen	
Trigger	N/A	
Post-Condition	Show information of all Kanji in user's input Save the searched Kanji, if it is available in the dictionary, to History.	
Main flow		
Step	Actor	Action
1	User	Tap to open tab Kanji dictionary
2	User	Tap on the magnifier icon in the right of the text field on top of the screen
3	System	Show keyboard
4	User	Type the word his want to search, then press Search key
5	System	If the text in the text field is Japanese characters, extract all Kanji from the text. For example: Extract Kanji from 秋は涼しい return 2 Kanji 秋、涼. Search for these Kanji in Kanji dictionary If the text in the text field is not Japanese, search all the Kanji that have other reading the same as the text.
6	System	Show a list of found Kanji. With each Kanji, show Kanji, On yomi, Kun yomi, meaning, other reading (if the Kanji

		has) and Star icon Save all Kanji to History
<b>Alternative flow 1</b>		
	Precondition	Dictionary screen
1	User	Select text in scanned document as in UC - 3
2	System	Write the recognized text in user selected area to the text field in the top of Dictionary screen
3	User	Tap the tab Kanji dictionary
4	System	View tab Kanji dicitory
3	System	Follow steps 5 to 6 in main flow
<b>Alternative flow 2</b>		
	Precondition	User has typed and searched a word in Japanese dictionary
1	User	Tap the tab Kanji dictionary
2	System	Follow step 5 to 6 of the main flow
	Exceptions	In step 6 of main flow, if there is no word found, display message "There is no result!"

UC – 5.1 Add to favorite list (Kanji dictionary screen)

Use Case ID	UC – 5.1	
Use Name	Add to favorite list	
Actor	User	
Description	User adds a word in Japanese dictionary to favorite list	
Precondition	System is in tab Kanji dictionary User has searched Kanji dictionary, and System is showing a list of words, as in steps 6 of main flow of UC - 6	
Trigger	N/A	
Post-Condition	The word is added to favorite list	
<b>Main flow</b>		
Step	Actor	Action
1	User	Tap on the grey start icon in the entry of the word that he want to add to favorite list
2	System	Change the star to yellow Add the word in the entry to favorite list in database
Exceptions		

UC – 6.2 Remove from favorite list (Kanji dictionary screen)

Use Case ID	UC – 5.2
Use Name	Remove from favorite list
Actor	User

Description		User removes a Kanji from favorite list
Precondition		System is in tab Kanji dictionary User has searched in Kanji dictionary, and System is showing a list of words, as in steps 6 of main flow of UC - 6
Trigger		N/A
Post-Condition		The word is added to favorite list
Main flow		
Step	Actor	Action
1	User	Tap on the yellow start icon in the entry of the word that he want to add to favorite list
2	System	Change the star to grey Remove the word in the entry from favorite list
Exceptions		

### 3.1.2.3.2. Screen description

Description of Result tab

Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
Header	Include the Menu icon and the title Anki Pan	-	-		-	-
Document area	Show the document that user captured. Text lines in the document are surrounded by white rectangles.	-	-	Image	-	-
Magnifier	Zoom in the area that user's finger is on when he is selecting text.	-	-		-	-

Description of Japanese tab

Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
Header	Include the Menu icon and the title Anki Pan	-	-		-	-
Input text area	Text area for user to input words	N	N	Text	Text	-
Word lists	Show a list of words that match user's input	-	-	Extendable list	-	-

Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
Star icon	User taps on Star icon to add / remove words from favorite list	-	-	Image icon	-	-
Extend icon	User taps on Extend icon to view full information of words (word type, example, meaning,...). Tap again to back to short form.	-	-	Image icon	-	-

### Description of Kanji tab

Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
Header	Include the Menu icon and the title Anki Pan	-	-		-	-
Input text are	Text area for user to input words	N	N	Text	Text	-
Word lists	Show a list of Kanji in user's input. Each Kanji is show with the Kanji in big size, On yomi, Kun yomi and the meaning of the Kanji, and other way of reading if any.	-	-	List	-	-
Star icon	User taps on Star icon to add / remove words from favorite list	-	-	Image icon	-	-

### 3.1.2.4 History

#### 3.1.2.4.1. User cases

##### UC - 6 View History

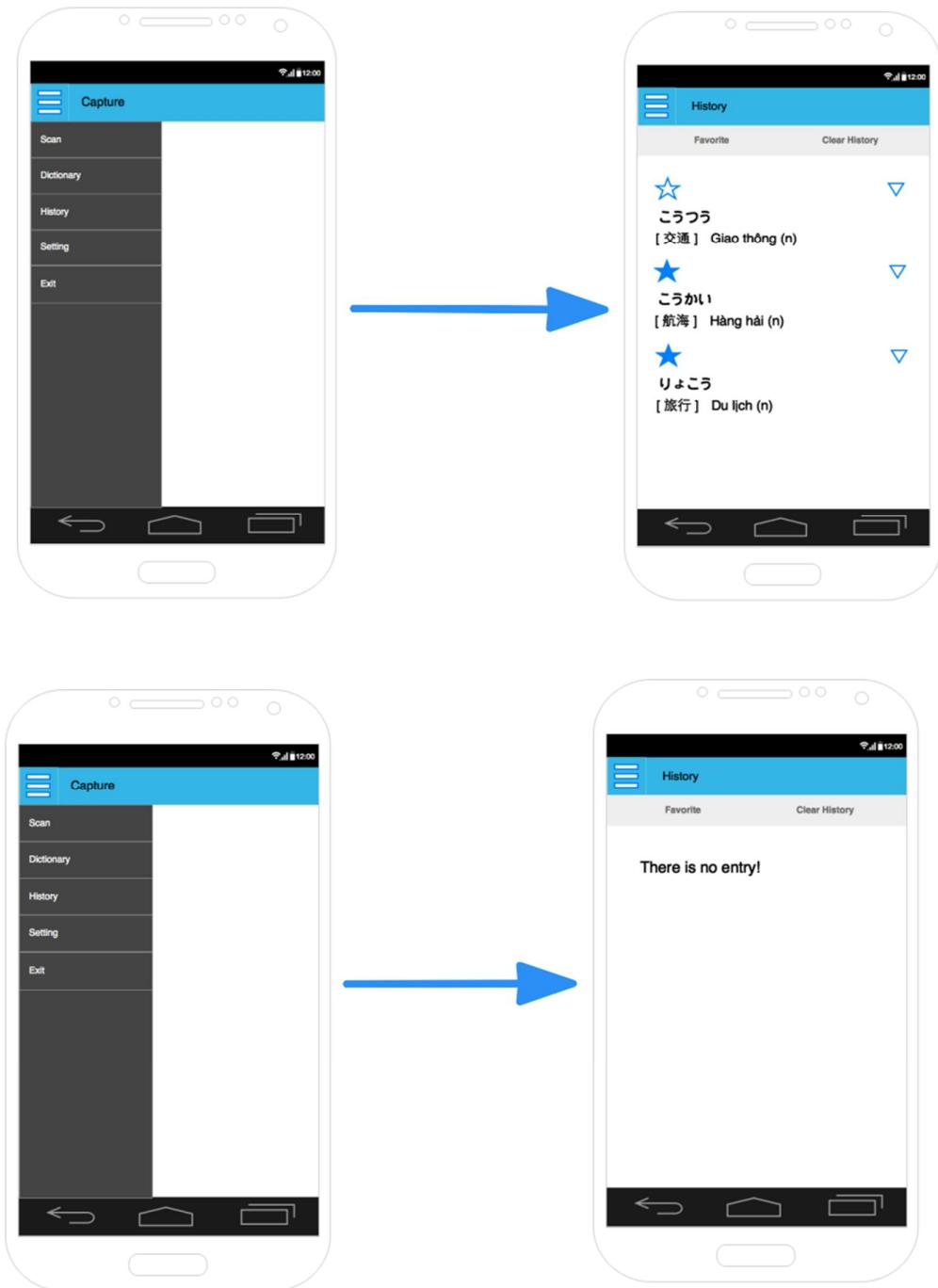


Figure 21: Screens of UC - 6

Use Case ID	UC - 6
-------------	--------

Use Name	View history	
Actor	User	
Description	User view all words that he has searched	
Precondition	N/A	
Trigger	N/A	
Post-Condition	All words in history are shown on screen	
Main flow		
Step	Actor	Action
1	User	Tap on the Menu icon in the top left of the screen
2	System	Show the screen list
3	User	Tap on entry History
4	System	Show History screen with all words user has search, in form of extendable list view. - In the entry of Japanese word: show Star icon, the word, its reading, the first meaning and word class (as in step 6 of UC – 5. Search in Japanese dictionary) - In the entry of Kanji: show Star icon and the Kanji If there is no word in History, show message “There is no word in History”
Alternative flow 1		
Exceptions	N/A	

UC – 6.1 View word's information (History screen)

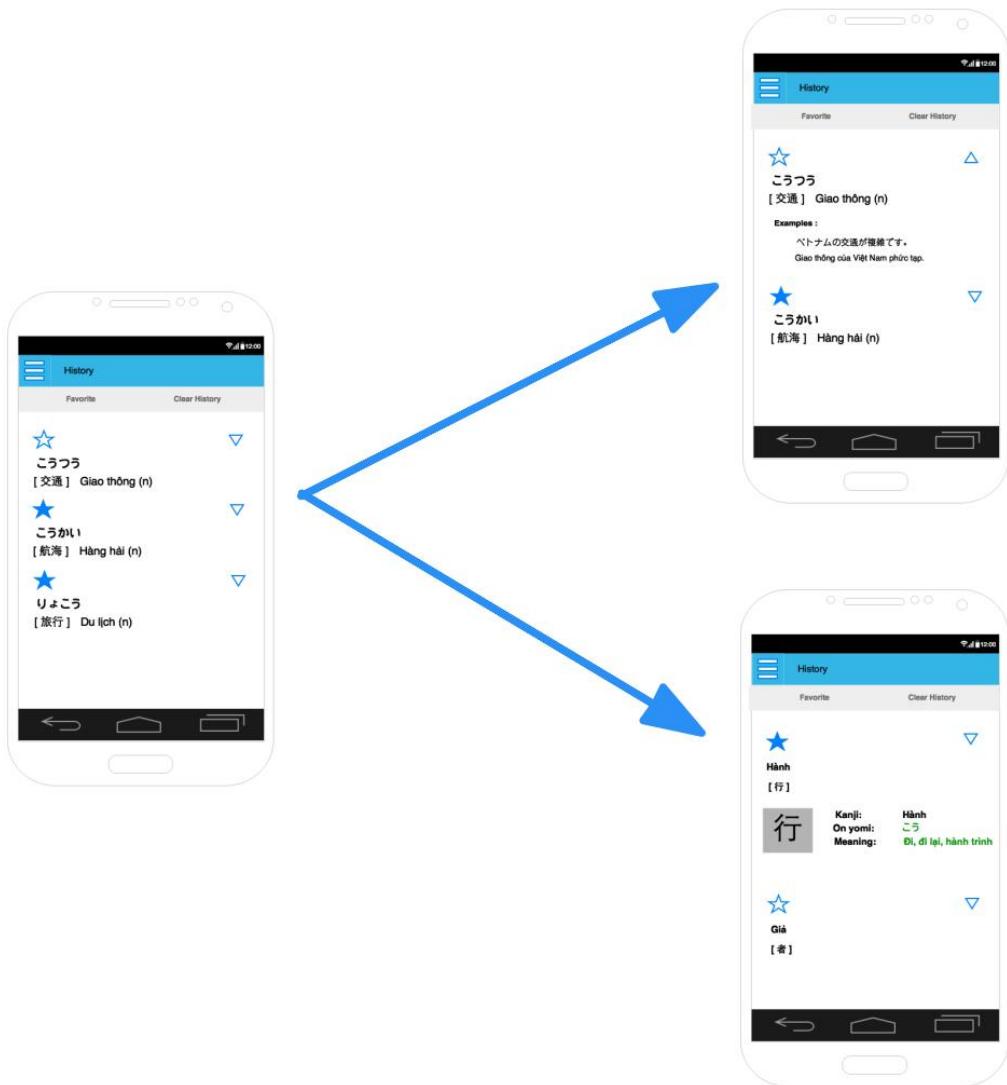


Figure 22: Screens of UC - 6.1

Use Case ID		UC - 6.1
Use Name		View word's information
Actor		User
Description		User looks for information of a word in history
Precondition		System is in History screen
Trigger		N/A
Post-Condition		Information of the word is shown on screen
Main flow		
Step	Actor	Action
1	User	Tap on the Extend icon on the right of the entry of the word that he want to see detail information
2	System	Extend the entry and view detail information of the word - If the word in the entry is from Japanese dictionary,

		<p>then search in Japanese dictionary, and show all its information (Star icon, word, Hiragana / Kanji writing, meaning, word class, example...) as show in step 8 of UC – 5. Search in Japanese dictionary</p> <ul style="list-style-type: none"> <li>- If the word in the entry is from Kanji dictionary, then search in Kanji dictionary and show Kanji, On yomi, Kun yomi, other reading (if it has), meaning, as show in step 6 of UC – 6. Search in Kanji dictionary</li> </ul>
<b>Alternative flow 1</b>		
<b>Precondition</b>		
1	User	Tap on the Extend icon of the entry again
2	System	<p>Show short form of the entry</p> <ul style="list-style-type: none"> <li>- if the word in the entry is Japanese word: show Star icon, word, its reading, the first meaning and word class.</li> <li>- if the word in the entry is Kanji: show Star icon and the Kanji</li> </ul>
<b>Exceptions</b>		N/A

#### UC – 6.2 Add to favorite list ( History screen)

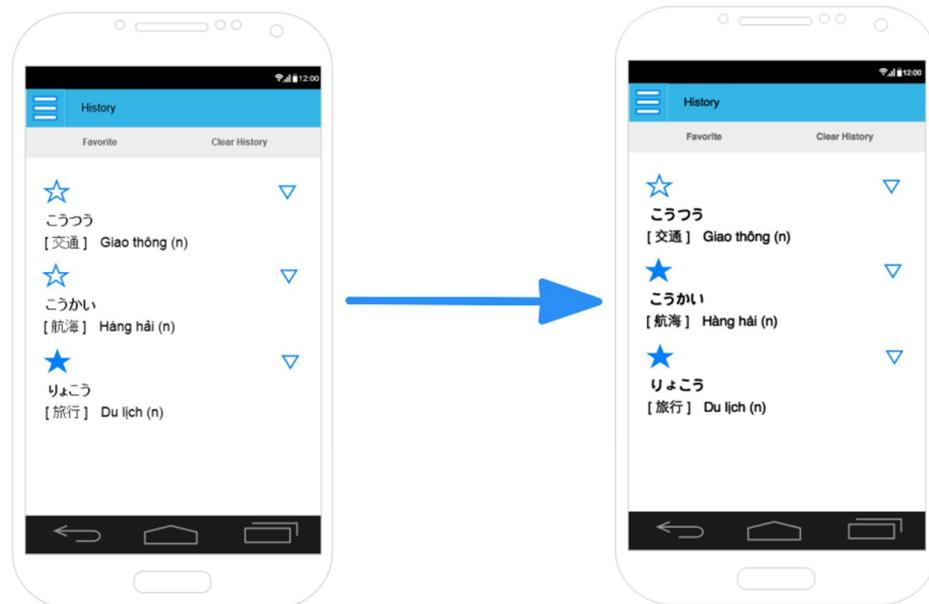


Figure 23: Screens of UC - 6.2

Use Case ID	UC – 6.2
Use Name	Add to favorite list
Actor	User

Description		User adds a word favorite list
Precondition		System is in History screen
Trigger		N/A
Post-Condition		The word is added to favorite list
Main flow		
Step	Actor	Action
1	User	Tap on the grey start icon in the entry of the word that he want to add to favorite list
2	System	Change the star to yellow Add the word in the entry to favorite list in database
Alternative flow 1		
Precondition		System is in History screen
1	User	Tap on the Extend icon on the right of a entry
2	System	Expand the entry and show detail information of the word in the entry, as in step 2 of UC - 7.1
3	User	Tap on grey star icon in the entry
4	System	Change the star to yellow Add the word in the entry to favorite list
Exceptions		

UC – 6.3 Remove from favorite list (History screen)

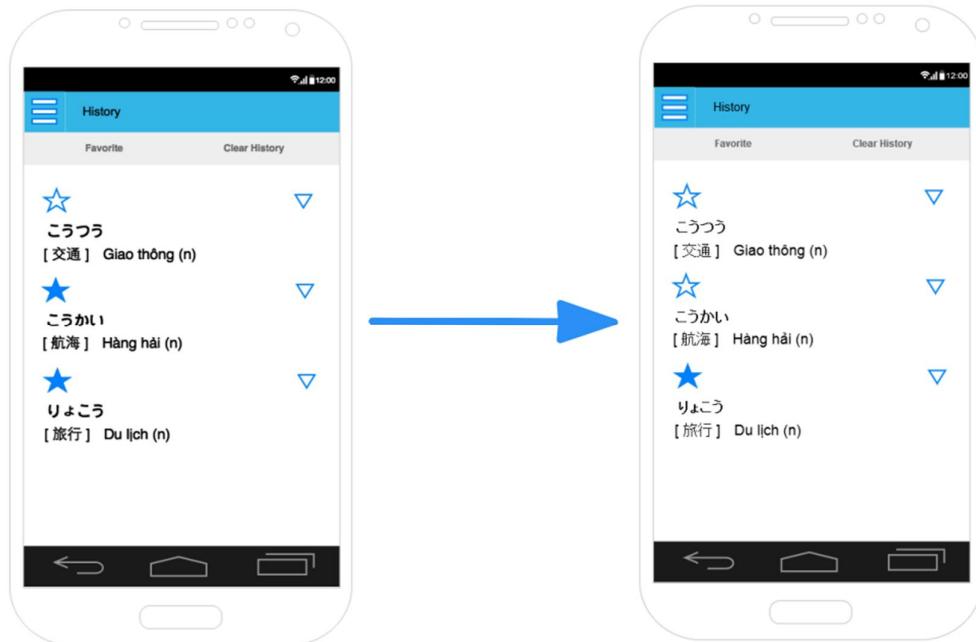


Figure 24: Screens of UC - 6.3

Use Case ID	UC – 6.3
Use Name	Remove form favorite list

Actor		User
Description		User removes a word from favorite list
Precondition		System is in History or Favorite screen
Trigger		N/A
Post-Condition		The word is removed from favorite list
Main flow		
Step	Actor	Action
1	User	Tap on the yellow start icon in the entry of the word that he want to remove from favorite list
2	System	Change the star to grey Remove the word in the entry from favorite list in database
Alternative flow 1		
Precondition	System is in History or Favorite screen	
1	User	Tap on the Extend icon on the right of a entry
2	System	Expand the entry and show detail information of the word in the entry, as in step 2 of UC - 7.1
3	User	Tap on yellow star icon in the entry
4	System	Change the star to grey Remove the word in the entry from favorite list in database
Exceptions		

#### UC – 6.4 View favorite words only

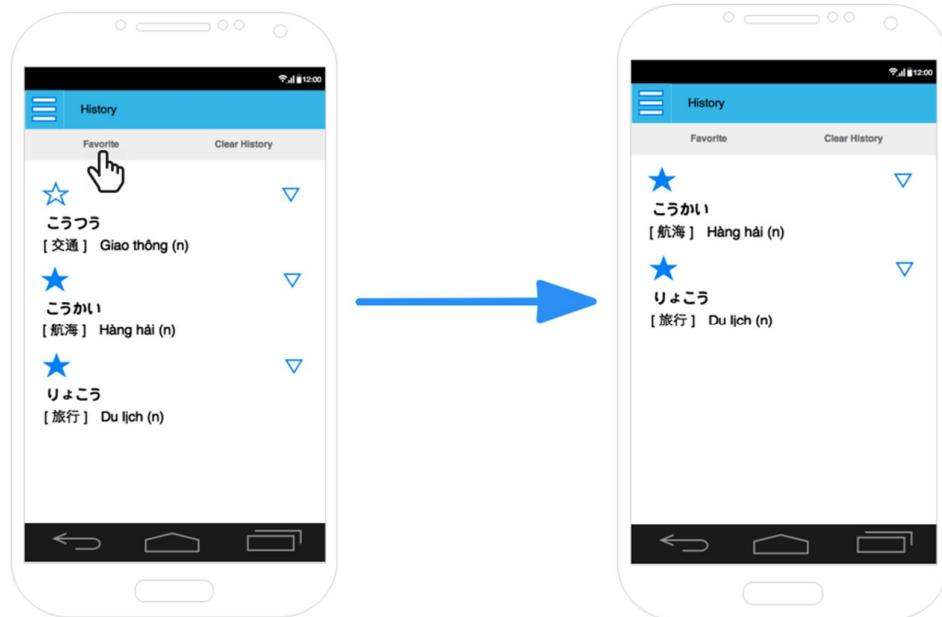


Figure 25: Screens of UC - 6.4

Use Case ID		UC – 6.4
Use Name		View favorite words only
Actor		User
Description		View all word that user put to favorite list
Precondition		System is in History screen
Trigger		N/A
Post-Condition		View all word in favorite list
Main flow		
Step	Actor	Action
1	User	Tap on the Favorite button
2	System	<p>Show all words in favorite list. The words are shown in form of extendable list view.</p> <ul style="list-style-type: none"> <li>- In the entry of Japanese word: show Star icon, the word, its reading, the first meaning and word class (as in step 6 of UC – 5. Search in Japanese dictionary)</li> <li>- In the entry of Kanji: show Star icon and the Kanji</li> </ul>
Alternative flow 1		
Precondition		
1	System	If there is no word in Favorite list, show message "There is no word in Favorite list"
Alternative flow 2		
Precondition		
1	User	After step 2 of the main flow, user tap the Favorite button again
2	System	Back to History screen with all favorite and not favorite words in the History
Exceptions		N/A

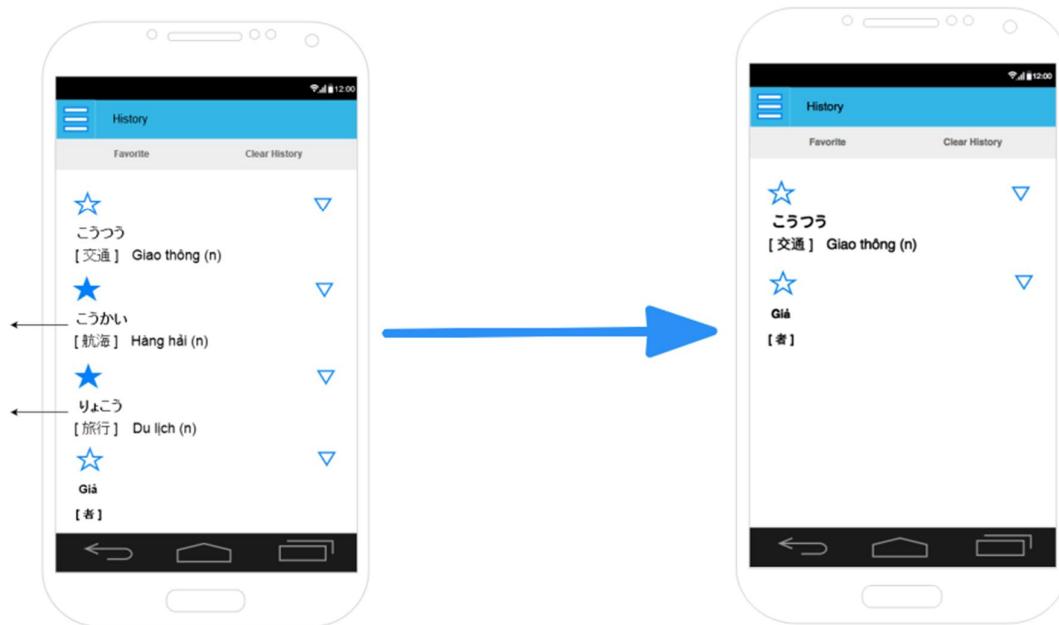
*UC – 6.5 Delete a word from history*

Figure 26: Screens of UC - 6.5

Use Case ID		UC – 6.5
Use Name		Delete 1 word from History
Actor		User
Description		User deletes one word from History
Precondition		System is in History screen
Trigger		N/A
Post-Condition		The deleted word is delete from History table of database
Main flow		
Step	Actor	Action
1	User	Tap on one entry in the History and swap it to the left
2	System	Show confirm message form "Do you want to delete the word?", with 2 button "Yes", "No"
3	User	Press "Yes" button
4	System	Remove entry with the word from screen Delete the word in History table of database
Alternative flow		
Precondition		
1	User	In step 3 of main flow, user press "No" button
2	System	Close confirm message form
Exceptions		N/A

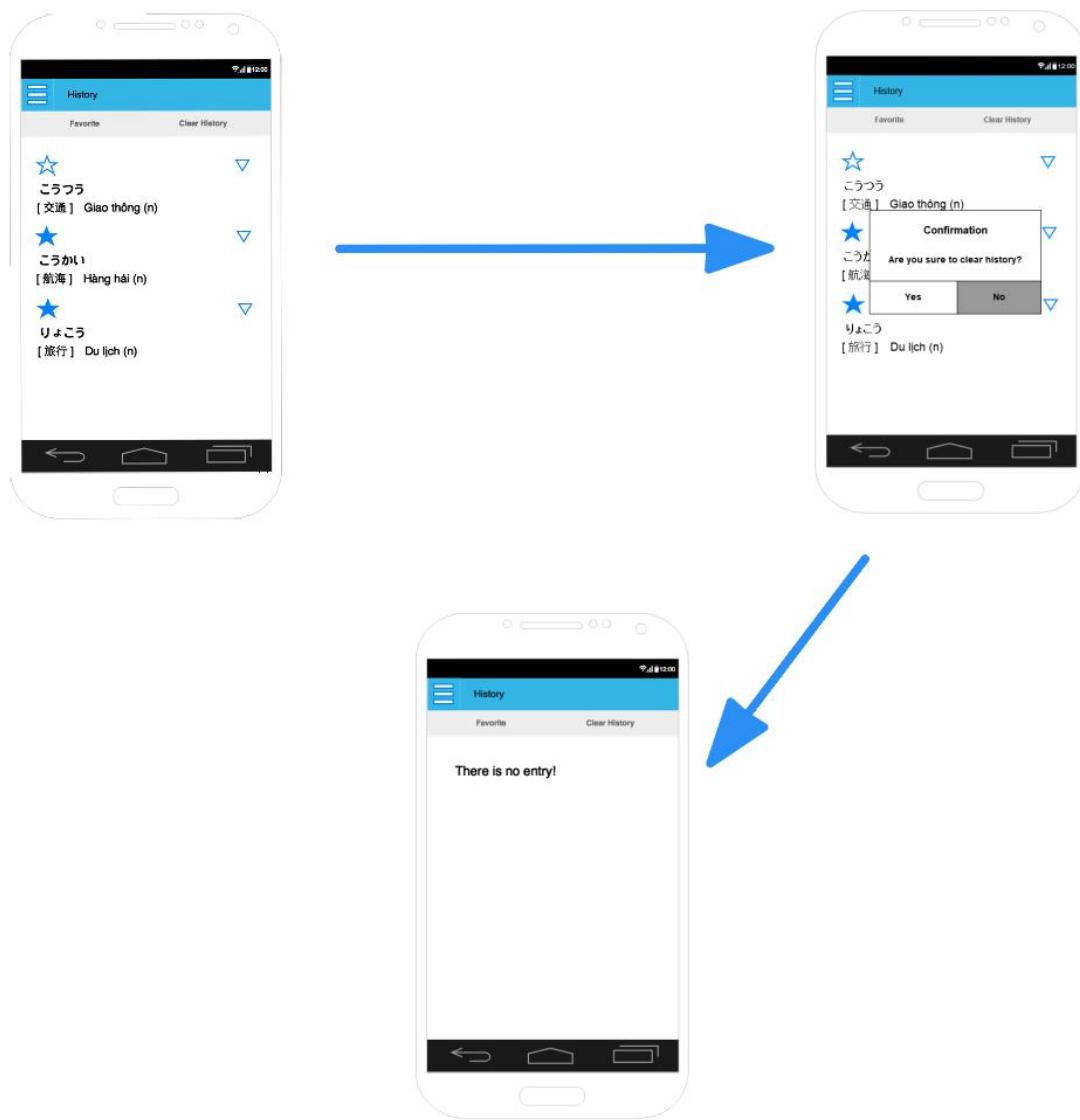
UC – 6.6 Delete all history

Figure 27: Screens of UC - 6.6

Use Case ID	UC – 6.6	
Use Name	Delete all History	
Actor	User	
Description	User deletes all words from History	
Precondition	System is in History screen	
Trigger	N/A	
Post-Condition	All words are deleted from History table of database	
Main flow		
Step	Actor	Action
1	User	Tap on Clear history button

2	System	Show confirm message form "Do you want to delete all history?", with 2 button "Yes", "No"
3	User	Press "Yes" button
4	System	Show message "There is no word in history" Delete all words in History table in database
Alternative flow		
Precondition		
1	User	In step 3 of main flow, user press "No" button
2	System	Close confirm message form
Exceptions		N/A

### 3.1.2.4.2 Screen description

Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
Header	The title bar of the application	-	-		-	-
Favorite button	The favorite button of each word or kanji entry. Tapping on this will add or remove the favorite state	-	-	Button	-	-
Word (Kana)	The word entries from searching query, always written in Hiragana or Katakana, there may be some entries have the same value	Y	Y	Label	Text	10
Word (Kanji)	The word entries above written in Kanji, no entries have the same value.	N	Y	Label	Text	7
Meaning	The meaning of the word entry. Also contain word type (noun, verb, adjective,...)	Y	Y	Label	Text	30
Extend button	The extend button for each word. Tapping on this will expand or collapse the content of the words to displays examples. The button direction is reserved after each time user tap	-	-	Button	-	-

Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
	on it.					
Example	The example area of each word entry. Example contains a Japanese sentence following by a Vietnamese sentence translated from the above one.	Y	Y	Label	Text	200
Confirmation box	The dialog box appears when user choose to clear history entry(ies)	-	-	Dialog box	-	-

### 3.1.2.5 Setting

#### 3.1.2.5.1. User cases

##### UC - 7: View Setting screen

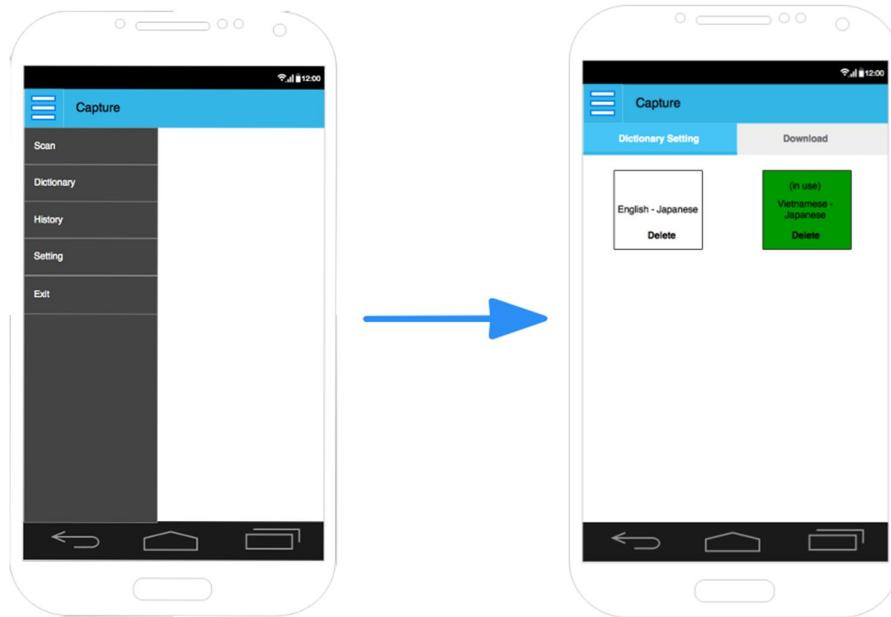


Figure 28: Screens of UC - 7

Use Case ID	UC - 7
Use Name	View Setting screen
Actor	User
Description	User views setting screen
Precondition	
Trigger	N/A

Post-Condition		Setting screen is shown
Main flow		
Step	Actor	Action
1	User	Tap on Menu icon in top left of the screen
2	User	Tap on Setting item
3	System	Open tab Dictionary setting of Setting screen In tab Dictionary setting, list all dictionary packages that user has downloaded. The dictionary package is chosen to use for searching word' information in Dictionary screen is marked.
Exceptions		N/A

UC - 7.1 Select a dictionary package

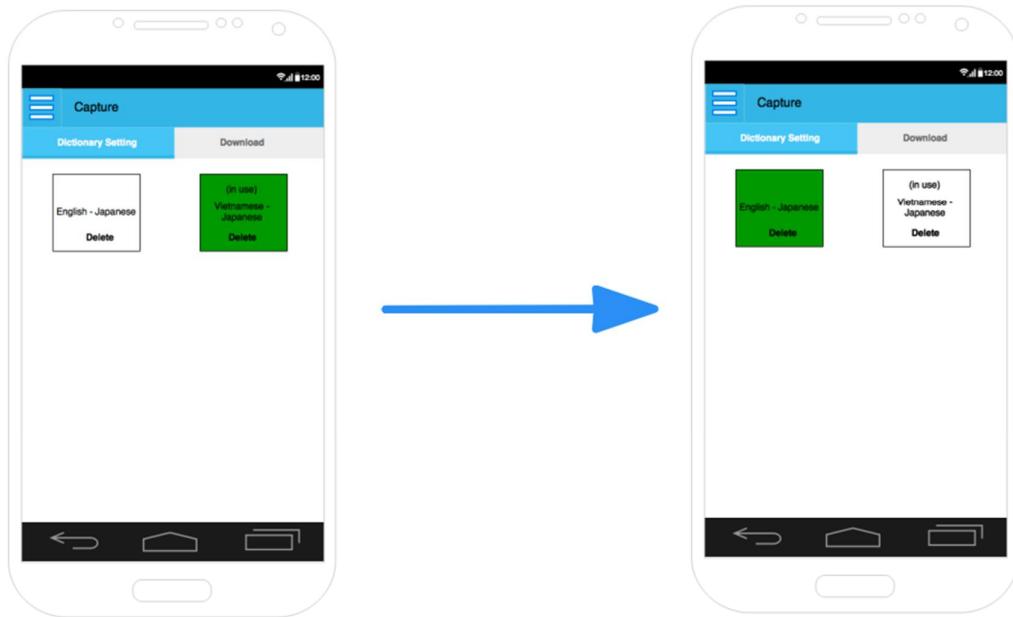


Figure 29: Screen of UC - 7.1

Use Case ID	UC - 7.1
Use Name	Select a dictionary package
Actor	User
Description	User select a dictionary package to use in Dictionary screen
Precondition	System is in Setting screen
Trigger	N/A
Post-Condition	The selected dictionary package is used in Dictionary screen

Main flow		
Step	Actor	Action
1	User	If system is not in tab Dictionary Setting, tap on tab Dictionary Setting
2	System	Show tab Dictionary Setting with all dictionary package that user has download, and the dictionary package which is in use is marked.
3	User	Tap on the dictionary package that he want to use in Dictionary screen
4	System	Mark the selected dictionary package Change the dictionary package use for Dictionary screen to selected one.
Exceptions		N/A

UC - 7.2 Delete a dictionary package

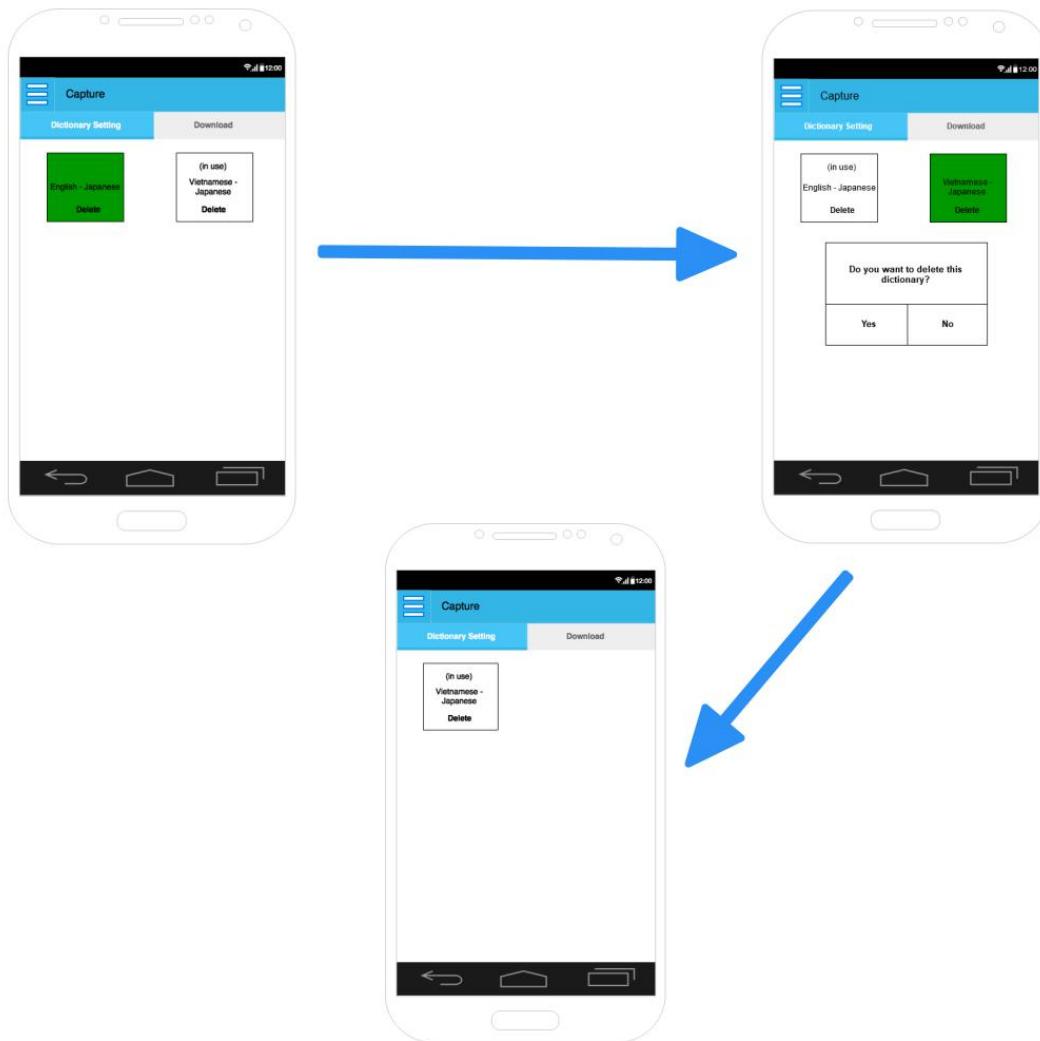


Figure 30: Screens of UC - 7.2' main flow



Figure 31: Screens of UC - 7.2' alternative flow

Use Case ID	UC - 7.2	
Use Name	Delete a dictionary package	
Actor	User	
Description	User delete a dictionary package that he has downloaded	
Precondition	System is in tab Dictionary setting of Setting screen	
Trigger	N/A	
Post-Condition	The deleted dictionary package is removed from database	
Main flow		
Step	Actor	Action
1	User	Tap on Delete button of the dictionary package that he wants to delete
2	System	Show confirm message form: "Do you want to delete [Dictionary package's name] dictionary package?" with 2 button "Yes" and "Cancel"
3	User	Tap on Yes button
4	System	Remove the dictionary package from screen Delete the dictionary package from database
Alternative flow 1		
Precondition		
1	User	In step 3 of the main flow, user tap on "Cancel" button
2	System	Close confirm form
Alternative flow 2		

Precondition		System is in Dictionary setting of Setting screen
1	User	In step 1 of the main flow, user tap on Delete button of the dictionary package that is using for Dictionary screen
2	System	Show warning message form "This dictionary package is in use!" with OK button
3	User	Tap on Ok button
4	System	Close warning form
Exceptions		N/A

UC – 7.3 View dictionary package's description

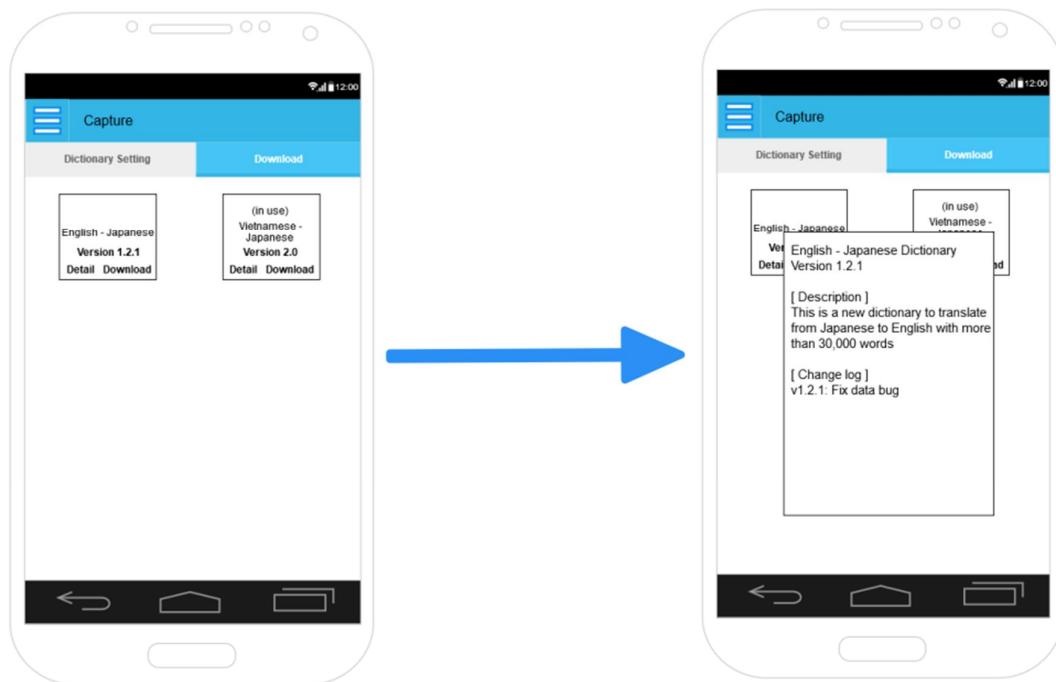


Figure 32: Screens of UC - 7.3

Use Case ID	UC – 8.3	
Use Name	View dictionary package's description	
Actor	User	
Description	User views description of a dictionary package	
Precondition	System is in Setting screen	
Trigger	N/A	
Post-Condition	The selected description of the dictionary package is shown on screen	
Main flow		
Step	Actor	Action
1	User	If system is not in tab Download, tap on tab Download

2	System	Send request to AKServer to get information of all Dictionary package available Show tab Download with all dictionary packages available on server and has not been downloaded
3	User	Tap Detail button of a package that he wants to view detail description
4	System	Pop out a form with description of the package, with OK button
5	User	Tap on OK button
6	System	Close the form
Exceptions		N/A

UC – 7.4 Download a dictionary package

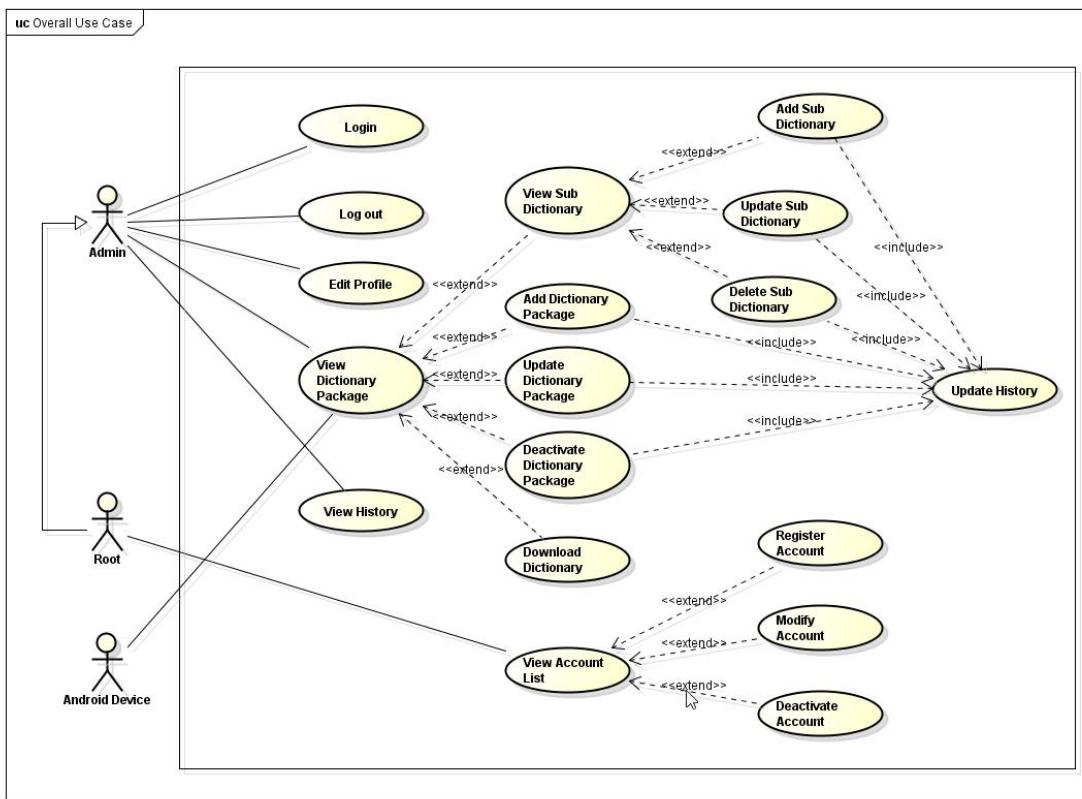
Use Case ID	UC – 8.4	
Use Name	Download a dictionary package	
Actor	User	
Description	User downloads a new dictionary package	
Precondition	System is in tab Download of Setting screen	
Trigger	N/A	
Post-Condition	The dictionary package is saved in the application	
Main flow		
Step	Actor	Action
1	User	Tap on Download button of the dictionary package that he wants to download
2	System	Show confirm message form: “Do you want to download [Dictionary package’s name] dictionary package?” with 2 button “Yes” and “Cancel”
3	User	Tap on Yes button
4	System	Request to AKServer to get the dictionary package Show status bar while downloading the dictionary package
5	System	Close the status bar when download is completed
Alternative flow 1		
Precondition		
1	User	In step 3 of the main flow, user tap on “Cancel” button
2	System	Close confirm form
Exceptions		If application is not connecting to the internet, show message form “Please connect to the internet to download this dictionary package”

### 3.1.2.5.2. Screen description

Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
Title bar	The title bar of the application	-	-	Label	-	-
Menu bar	The tab bar of the dictionary feature, contain 2 tab buttons: Setting Dictionary Downlaod	-	-	Button	-	-
Dictionary item	A dictionary item, contains its name, current version, tapping on this item will set the dictionary to “in use” state	-	-	Component	-	-
Delete button	The delete button of each dictionary item in setting tab	-	-	Button	-	-
Error dialog	The dialog that show error when user tries to delete a dictionary that are currently “in use”	-	-	Dialog box	-	-
Detail button	The detail button of each dictionary item in download tab	-	-	Button	-	-
Detail box	The box containing all description about the selected dictionary	-	-	Box	Text	500
Download /Update button	The download button of each dictionary. Will change to update button if the dictionary is already exist in user's device	-	-	Button	-	-
Confirmation box	The dialog box appears when user choose to delete dictionary	-	-	Dialog box	-	-

## 3.2. Server

### 3.2.1 User case diagram



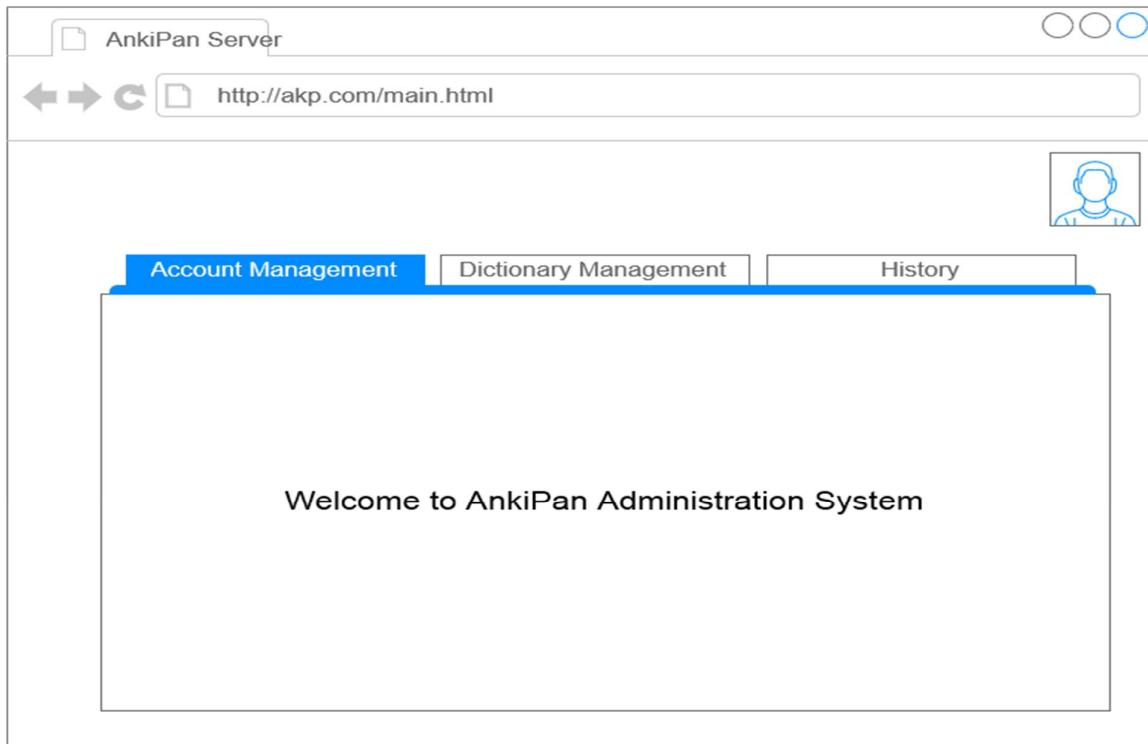
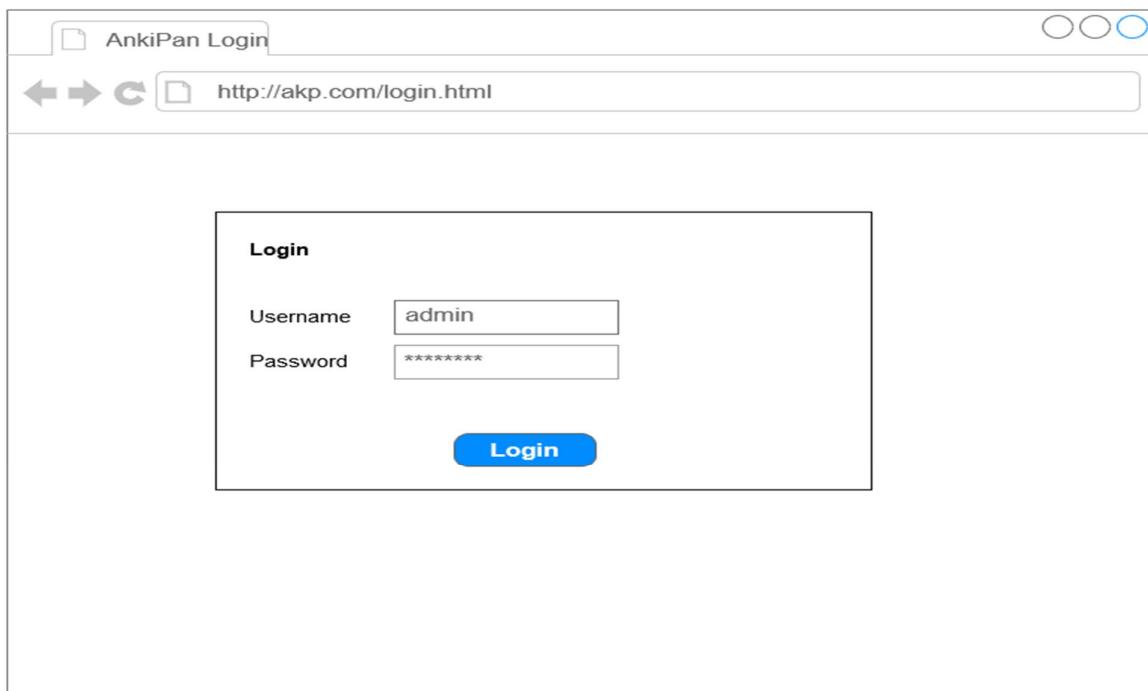
*Figure 33: User case diagram for server*

### 3.2.2. Functions

#### 3.2.2.1 Login

##### 3.2.2.1.1 User cases

UC - 8. Login



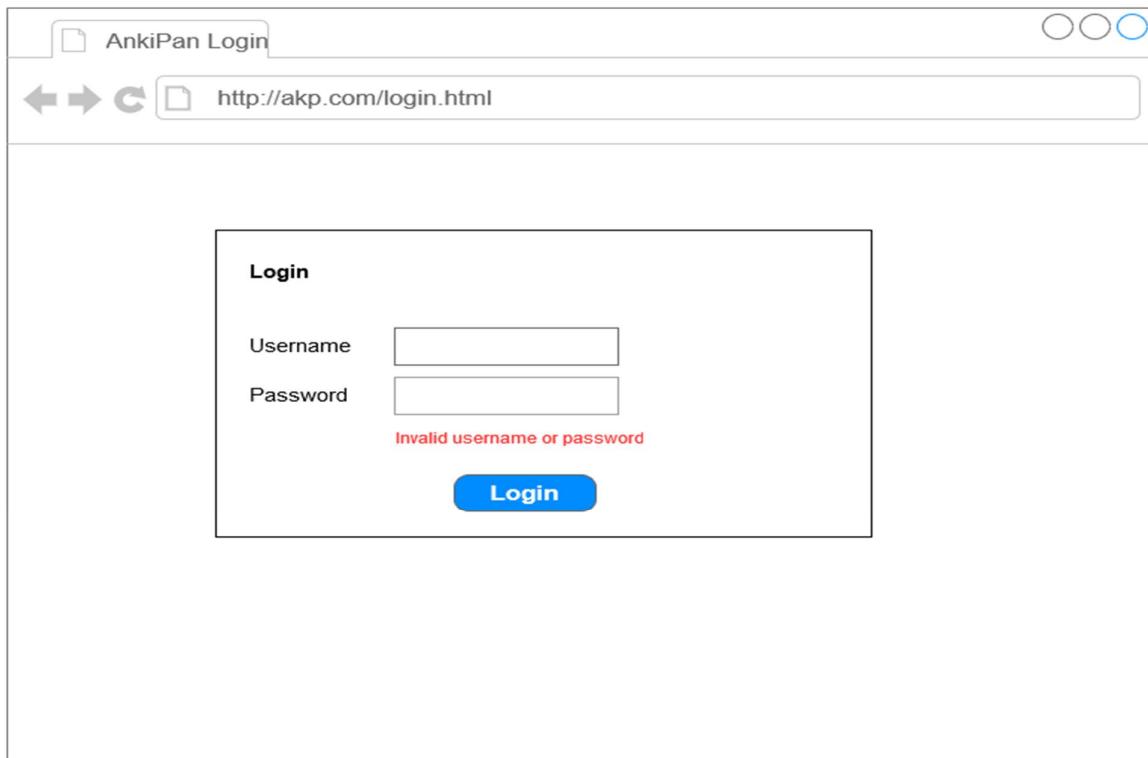


Figure 34: Screens of UC - 8

Use Case ID		UC – 8
Use Name		Login
Actor		Admin
Description		Admin login into AnkiPan Server
Precondition		Internet connection
Trigger		N/A
Post-Condition		The admin's session is stored
Main flow		
Step	Actor	Action
1	Admin	Type Ankipan's URL into the address box of the browser, then press "Enter" or use "Go" button
2	System	Display the index page, which allows admin to login
3	Admin	Enter his/her username and password, click "Login" button
4	System	Validate input information, then grant access to AnkiPan Server
Alternative flow 1		
Precondition		In step 3, admin entered incorrect information
1	User	Entered incorrect information
2	System	Validated information, then display the login page again with error message "Incorrect username and

		password”
Exceptions	N/A	

### 3.2.2.1.2. Screen description

Screen	Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
Login page	Username	The username field of the login form. Default value: none.	Y	N	Text field	Text	20
	Password	The password field of the login form, mask every character user typed in with asterisk (*). Default value: none	Y	N	Password field	Text	20
	Login button	The button for logging into the system	-	-	Button	-	-
Main page	Navigation bar	The main navigation bar of the entire system, containing 3 item: Account list (only for root admin), Dictionary list, History list	-	-	Navigation bar	-	-
	Welcome text	The welcome text display in the content pane after user login into the system Default value: “Welcome to AnkiPan Administration System”	Y	Y	Text	Text	50
	Avatar	The profile picture of the current user	-	-	Picture	-	-
Login fail page	Error message	The error message when user enter incorrect username or password then login	Y	Y	Text	Text	30

### 3.2.2.2 Log out

#### 3.2.2.2.1 User cases

##### UC – 9.2. Log out

Use Case ID	UC – 9.2
Use Name	Log out
Actor	Admin

Description		Admin log out from AnkiPan Server
Precondition		Internet connection
Trigger		N/A
Post-Condition		The admin's session is ended
Main flow		
Step	Actor	Action
1	Admin	Click "Log out" link on the upper right of the browser screen
2	System	Ended admin's session, then redirect admin to the index page
Alternative flow 1		
Precondition		
		N/A
Exceptions		N/A

### 3.2.2.3 Manage profile

#### 3.2.2.3.1 User cases

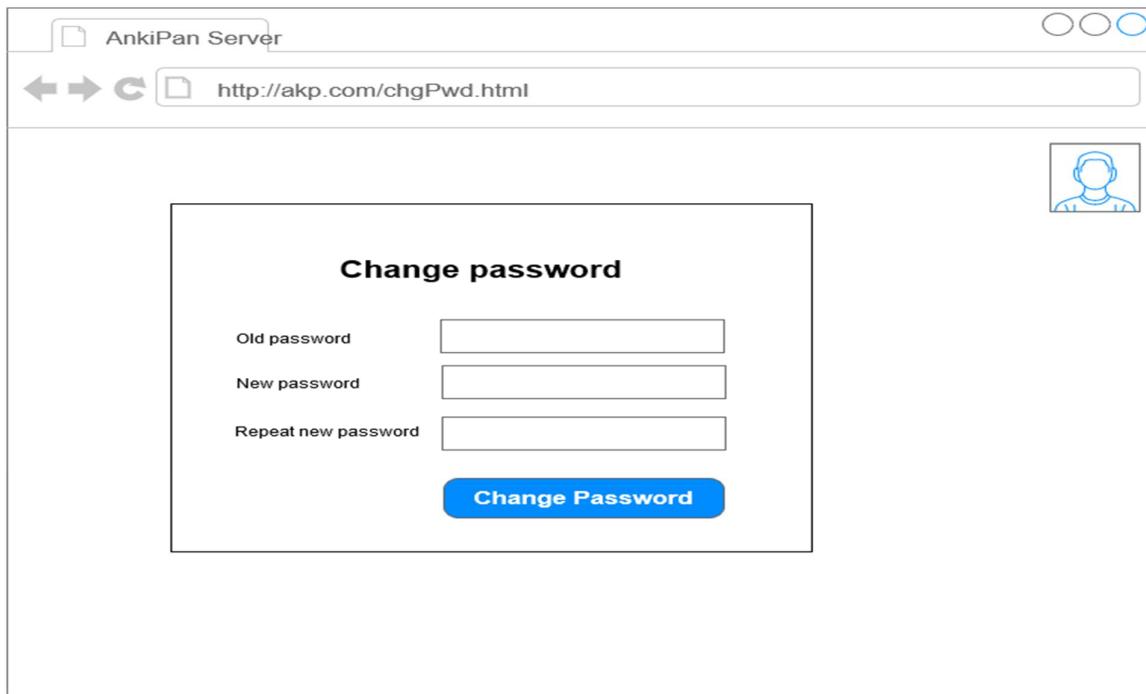
##### UC – 10.1 Edit Profile

AnkiPan Server

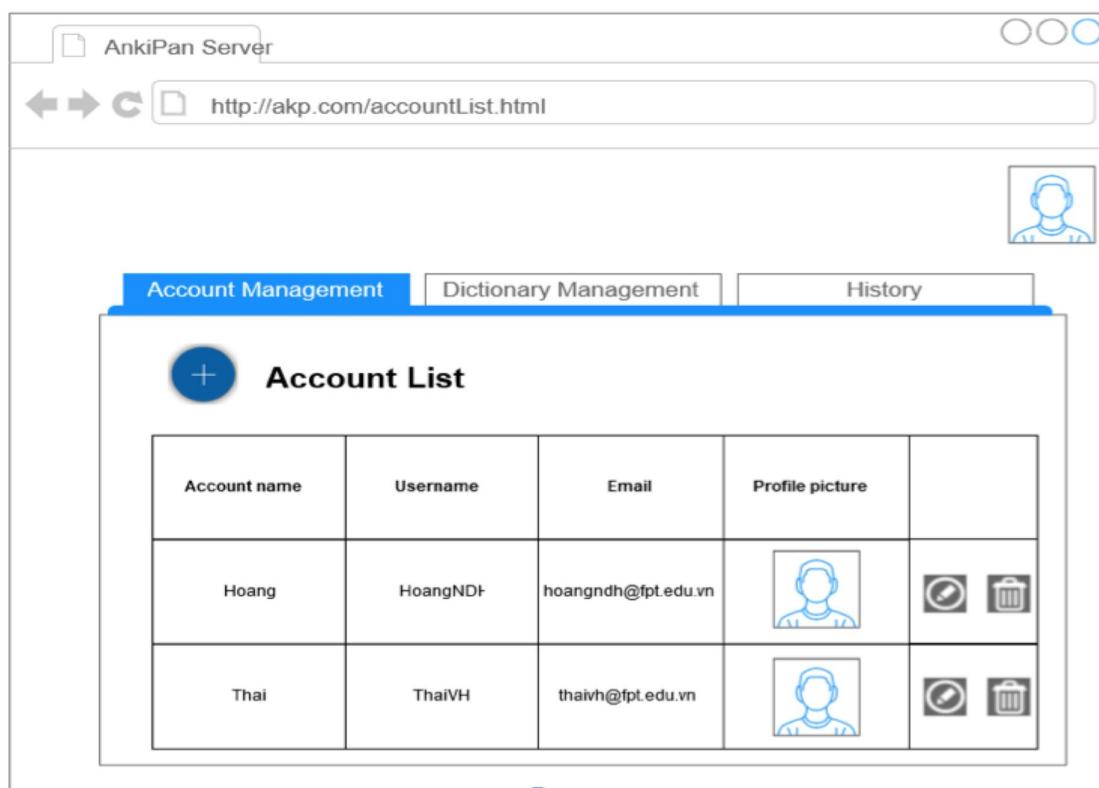
http://akp.com/editProfile.html

Edit Profile

Account name	<input type="text"/>
Username	<input type="text"/>
Password	<input type="password"/> <b>Change Password</b>
Email	<input type="text"/>
Profile picture	<input type="file"/>
<b>Update</b>	



Use Case ID	UC – 10.1	
Use Name	Edit Profile	
Actor	Admin	
Description	Admin edit owned profile	
Precondition	Internet connection	
Trigger	N/A	
Post-Condition	The admin's session is ended	
Main flow		
Step	Actor	Action
1	Admin	Click "Welcome" link or the admin's logo on the upper right of the browser screen
2	System	Load admin's information, then display the edit profile form, with user information entered.
3	Admin	Edit information, and click Update
4	System	Update admin's new information into the database, then redirect to the main screen
Alternative flow 1		
Precondition		At step 3 of the main flow, user leave any field blank
1	Admin	Leave any field blank
2	System	Surround the field with red lines, notify admin to enter that field
Exceptions		N/A

**UC 10.2 – View Account List**

Use Case ID	UC – 10.2	
Use Name	View Account List	
Actor	Root Admin	
Description	Root Admin view other admin's account list	
Precondition	Internet Connection	
Trigger	N/A	
Post-Condition	N/A	
Main flow		
Step	Actor	Action
1	Admin	Click on "Account List" link on toolbar
2	System	Get the admin's account list
4	System	Display the account list on the screen
Alternative flow 1		
Precondition		
1	User	Admin try to access the account list link
2	System	Block admin and redirect to main page
Exceptions		N/A

UC 10.3 – Register account

The screenshot shows a web browser window with the title "AnkiPan Server". The address bar displays the URL "http://akp.com/register.html". The main content area is a registration form titled "Register". The form includes fields for "Account name", "Username", "Password", "Confirm Password", "Email", "Role" (with radio buttons for "Root admin" and "Admin"), and "Profile picture". Below the form is a blue "Register" button. In the top right corner of the page, there is a small user icon.

Use Case ID	UC – 10.3	
Use Name	Register account	
Actor	Root admin	
Description	Root admin create new account for other admin access	
Precondition	Internet Connection	
Trigger	N/A	
Post-Condition	New account is created and added into the database	
Main flow		
Step	Actor	Action
1	Root admin	Click "Create new account" link on the tab bar
2	System	Display the add new account form
3	Root admin	Enter new admin's information into fields, then click "Register"
4	System	Validate input
5	System	Add new account into the database, then return to account list page
Alternative flow 1		
Precondition		
1	Root admin	In step 3 of the main flow, root admin leaves any field blank
2	System	Surround the empty field with red lines, require root

		admin to fill that field
<b>Alternative flow 2</b>		
<b>Precondition</b>		
1	Root admin	In step 3 of the main flow, root admin fill incorrect email, then click "Register"
2	System	Show error message "Invalid email address"
3	Root admin	Tap on Ok button
4	System	Close error message
<b>Exceptions</b>		N/A

#### UC 10.4 – Modify account

The screenshot shows a web browser window with the title bar 'AnkiPan Server'. The address bar displays the URL 'http://akp.com/editProfile.html'. The main content area is a form titled 'Edit Profile' with four input fields: 'Account name', 'Username', 'Email', and 'Profile picture'. Below the fields is a large blue 'Update' button. In the top right corner of the browser window, there is a small user icon.

Use Case ID	UC – 10.4	
Use Name	Modify account	
Actor	Root admin	
Description	Root admin modify other admin account	
Precondition	Internet Connection Root admin is at account list page	
Trigger	N/A	
Post-Condition	Admin's account is modified and save into the database	
<b>Main flow</b>		
Step	Actor	Action

1	Root admin	On the account list page, click the pen button at the right of admin account
2	System	Display the update account page
3	Root admin	Enter admin's new information into fields, then click "Update"
4	System	Validate input
5	System	Update new account into the database, then return to account list page
Alternative flow 1		
Precondition		
1	Root admin	In step 3 of the main flow, root admin leaves any field blank
2	System	Surround the empty field with red lines, require root admin to fill that field
Alternative flow 2		
Precondition		
1	Root admin	In step 3 of the main flow, root admin fill incorrect email, then click "Update"
2	System	Show error message "Invalid email address"
3	Root admin	Tap on Ok button
4	System	Close error message
Exceptions		N/A

#### UC 10.5 – Deactivate account

Use Case ID	UC – 10.5	
Use Name	Deactivate account	
Actor	Root admin	
Description	Root admin deactivate other admin's account, preventing him/her from accessing the system	
Precondition	Internet Connection Root admin is at account list page	
Trigger	N/A	
Post-Condition	Admin's account is deactivated and cannot be used to access the system again.	
Main flow		
Step	Actor	Action
1	Root admin	On the account list page, click the trash bin icon at the right of the admin's account information
2	System	Deactivate the account, then display the list again, with that account removed
Alternative flow 1		
Precondition		

		N/A
Exceptions		N/A

### 3.2.2.3.2 Screen description

Screen	Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
Account List page	Account table	The table containing account information about admin (including root admin)	-	-	Table	-	-
	Account name column	The account name column of the list of admins	-	-	Table header	-	-
	Username column	The username column of the list of admins	-	-	Table header	-	-
	Email column	The email column of the list of admins	-	-	Table header	-	-
	Role column	The role column of the list of admins	-	-	Table header	-	-
	Profile picture column	The profile picture column of the list of admins	-	-	Table header	-	-
Edit profile page	Account name	The account name of the current admin, this field is fixed Default value: current admin's account name	Y	Y	Text field	Text	30
	Username	The account name of the current admin Default value: current admin's username	Y	N	Text field	Text	30
	Password button	The change password button	-	-	Button	-	-
	Email	The emai of the current admin Default value: current admin's email	Y	N	Text field	Text	50
	Profile Picture	The profile picture of the current admin Default value: none	-	-	File	-	-

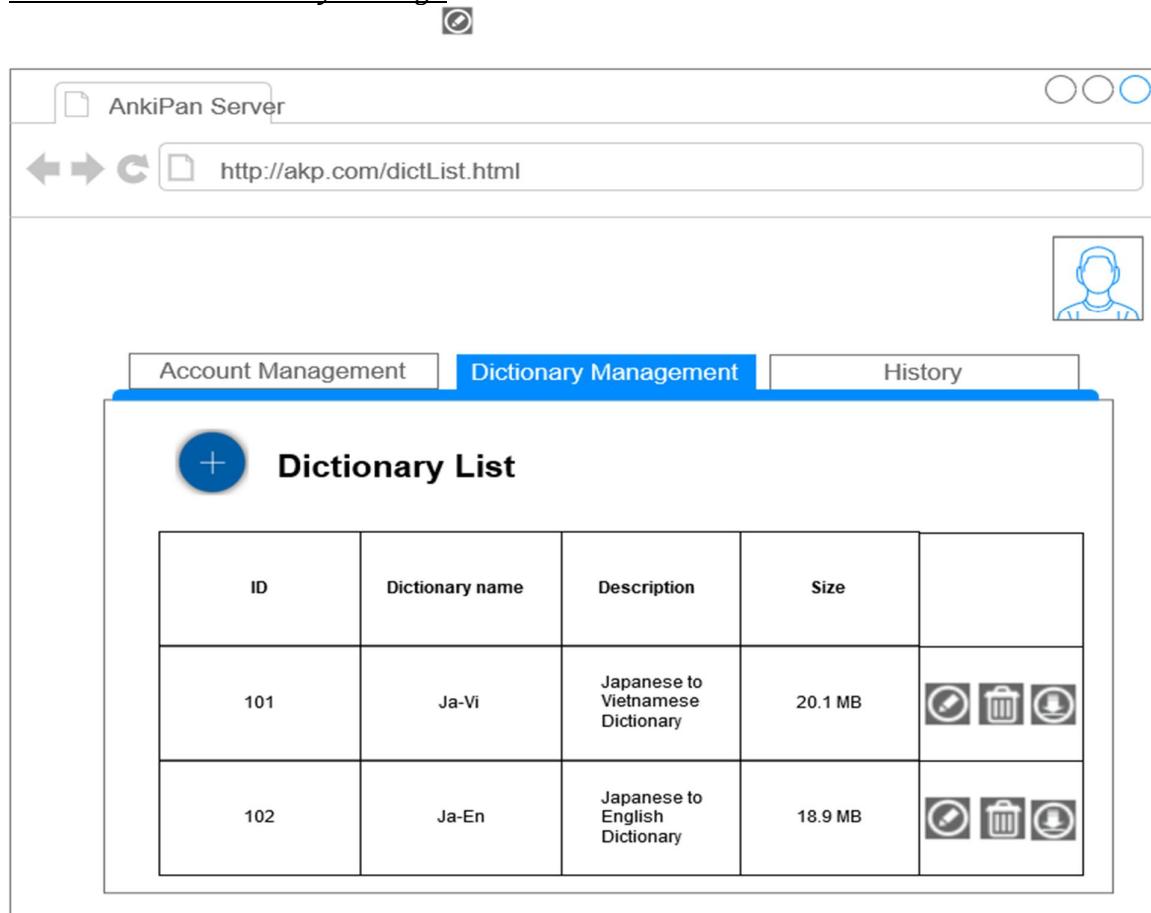
Screen	Field Name	Description	Mand atory	Read only	Control type	Data Type	Length
Change password page	Old password	The old password of the current user, mask all character with asterisk (*) Default value: none Constant: more than 6 characters, no space	Y	N	Password field	Text	30
	New password	The new password of the current user, mask all character with asterisk (*) Default value: none	Y	N	Password field	Text	30
	Repeat new password	The old password of the current user, mask all character with asterisk (*) Default value: none	Y	N	Password field	Text	30
Register Page	Account name	The account name of the new admin Default value: none	Y	N	Text field	Text	30
	Username	The username of the new admin Default value: none	Y	N	Text field	Text	30
	Password	The password of the new admin, mask all field with asterisk (*) Default value: none	Y	N	Password field	Text	30
	Confirm password	The password confirmation of the new admin, mask all field with asterisk (*) Default value: none	Y	N	Password field	Text	30
	Email	The email of the new admin Default value: none Constant: contain @ and domain name	Y	N	Text field	Text	30
	Role	The role of the new admin, containing 2 items which root admin will choose 1: admin and root admin	-	-	Radio box	-	-

Screen	Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
	Profile Picture	The profile picture of the new admin	-	-	File	-	-
	Register button	The register button of the register form	-	-	Button	-	-

### 3.2.2.4 Manage dictionary packages

#### 3.2.2.4.1 User cases

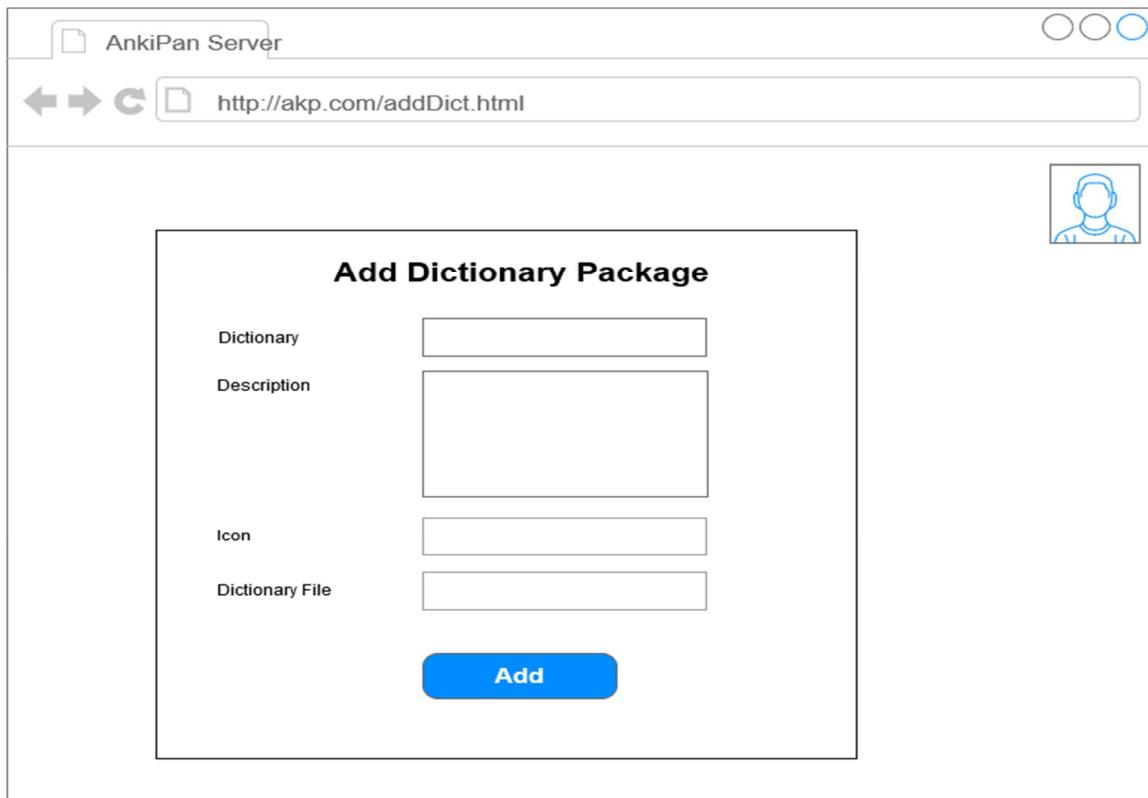
UC 11.1 - View Dictionary Package



Use Case ID	UC - 11.1
Use Name	View Dictionary Package
Actor	Admin
Description	Admin view dictionary package list
Precondition	Internet Connection
Trigger	N/A

Post-Condition		N/A
Main flow		
Step	Actor	Action
1	Admin	Click on “Dictionary Package List” link on taskbar
2	System	Get the dictionary package list
4	System	Display the dictionary package list on the screen
Alternative flow 1		
Precondition		
		N/A
Exceptions		N/A

### UC 11.2 – Add Dictionary Package



Use Case ID	UC – 11.2
Use Name	Add Dictionary Package
Actor	Admin
Description	Admin add new dictionary package
Precondition	Internet Connection Admin is at the dictionary package list
Trigger	N/A
Post-Condition	New dictionary package is created and added into the database

Main flow		
Step	Actor	Action
1	Admin	On the dictionary package list, click the + button
2	System	Display the add new dictionary form
3	Admin	Enter new dictionary package's information into fields, then click "Add"
4	System	Validate input
5	System	Add new dictionary package into the database, then return to dictionary package list page

Alternative flow 1		
Precondition		
1	Admin	In step 3 of the main flow, admin leaves any field blank
2	System	Surround the empty field with red lines, require admin to fill that field
Exceptions		N/A

### UC 11.3 – Update Dictionary Package

The diagram illustrates the 'Update Dictionary Package' user interface. At the top, a browser window displays the URL <http://akp.com/updateDict.html>. The main content is a form titled 'Update Dictionary Package' containing six input fields: 'Dictionary ID', 'Dictionary', 'Description', 'Version', 'Icon', and 'Dictionary File'. Below these fields is a large blue 'Update' button. In the top right corner of the form area, there is a small icon of a person.

Use Case ID	UC – 11.3
Use Name	Update Dictionary Package

Actor		Admin
Description		Admin update the dictionary package
Precondition		Internet Connection Admin is at dictionary package list page
Trigger		N/A
Post-Condition		Dictionary package is updated and save into the database
Main flow		
Step	Actor	Action
1	Admin	On the dictionary package list page, click the pen icon at the right of the dictionary package
2	System	Display the update dictionary package page
3	Admin	Enter new dictionary package's information into fields, then click "Update"
4	System	Validate input
5	System	Update new dictionary package's information into the database, then return to dictionary package list page
Alternative flow 1		
Precondition		
1	Admin	In step 3 of the main flow, Admin leaves any field blank
2	System	Surround the empty field with red lines, require admin to fill that field
Exceptions		N/A

UC 11.4 – Deactivate Dictionary Package

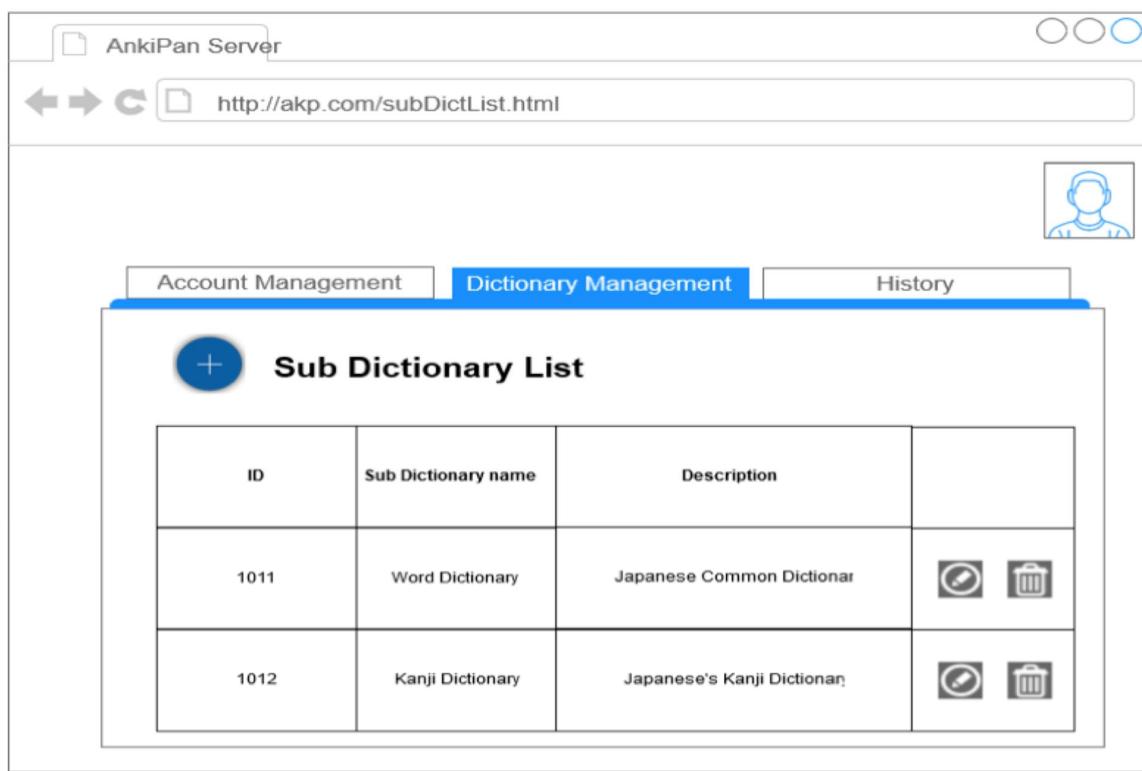
Use Case ID		UC – 11.4
Use Name		Deactivate Dictionary Package
Actor		Admin
Description		Admin deactivate the dictionary package
Precondition		Internet Connection Admin is at dictionary package list page
Trigger		N/A
Post-Condition		Dictionary Package is deactivated and any device cannot get its information
Main flow		
Step	Actor	Action
1	Admin	On the dictionary package list page, click the trash bin icon at the right of the dictionary package information
2	System	Deactivate the dictionary package, then display the dictionary package list again, with that dictionary package removed

Alternative flow 1		
Precondition		
		N/A
Exceptions		N/A

### 3.2.2.4.2 Screen description

Screen	Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
Dictionary List page	Dictionary table	The table containing dictionary information	-	-	Table	-	-
	ID column	The ID column of the list of dictionary package	-	-	Table header	-	-
	Dictionary name column	The name column of the list of dictionary package	-	-	Table header	-	-
	Description column	The description column of the list of dictionary package	-	-	Table header	-	-
	Size	The size column of the list of dictionary package	-	-	Table header	-	-
	Toolbox colum	The toolbox column of the list of dictionary package, containing button related to the dictionary of the same row	-	-	Table header	-	-
Add dictionary page	Dictionary name	The name of the new dictionary Default value: none	Y	Y	Text field	Text	30
	Description	The descriprion of the new dictionary Default value: none	Y	N	Text area	Text	250
	Icon	The icon of the new dictionary	-	-	File	-	-
	Dictionary file	The file of the new dictionary	-	-	File	-	-
	Add button	The add dictionary form's button	-	-	Button	-	-
Update dictionary	Dictionary ID	The ID of the current dictionary, this field is	Y	Y	Text field	Text	5

Screen	Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
page		fixed Default value: current dictionary's ID					
	Dictionary name	The name of the current dictionary Default value: current dictionary's name	Y	N	Text field	Text	30
	Description	The description of the current dictionary Default value: current dictionary's description	Y	N	Text area	Text	250
	Version	The version of the current dictionary Default value: current dictionary's version	Y	N	Text field	Text	10
	Icon	The icon of the new dictionary	-	-	File	-	-
	Dictionary file	The file of the new dictionary	-	-	File	-	-
	Update button	The add dictionary form's button	-	-	Button	-	-

**UC 11.5 – View Sub Dictionary**

Use Case ID	UC – 11.5	
Use Name	View Sub Dictionary	
Actor	Admin	
Description	Admin view sub dictionary list	
Precondition	Internet Connection Admin is at dictionary package list page	
Trigger	N/A	
Post-Condition	N/A	
Main flow		
Step	Actor	Action
1	Admin	On the dictionary package list page, click on the dictionary package
2	System	Get the sub dictionary list
4	System	Display the sub dictionary list on the screen
Alternative flow 1		
Precondition		
		N/A
Exceptions		N/A

**UC 11.6 – Add Sub Dictionary**

AnkiPan Server

http://akp.com/addSubDict.html

Add Sub Dictionary

Sub Dictionary

Description

Dictionary ID

Add

Use Case ID	UC – 11.6	
Use Name	Add Sub Dictionary	
Actor	Admin	
Description	Admin add new sub dictionary	
Precondition	Internet Connection Admin is at the sub dictionary list	
Trigger	N/A	
Post-Condition	New sub dictionary is created and added into the database	
Main flow		
Step	Actor	Action
1	Admin	On the sub dictionary list, click the + button
2	System	Display the add new sub dictionary form
3	Admin	Enter new sub dictionary's information into fields, then click "Add"
4	System	Validate input
5	System	Add new sub dictionary into the database, then return to sub dictionary list page
Alternative flow 1		

Precondition		
1	Admin	In step 3 of the main flow, admin leaves any field blank
2	System	Surround the empty field with red lines, require admin to fill that field
Exceptions		N/A

UC 11.7 – Update Sub Dictionary

The screenshot shows a web browser window with the following details:

- Address bar: AnkiPan Server → http://akp.com/addSubDict.html
- User icon in the top right corner.
- Main content area:
  - Update Sub Dictionary** title.
  - Form fields:
    - Sub Dictionary ID
    - Sub Dictionary
    - Description
    - Dictionary ID
  - A blue **Add** button at the bottom.

Use Case ID	UC – 11.7	
Use Name	Update Sub Dictionary	
Actor	Admin	
Description	Admin update the sub dictionary	
Precondition	Internet Connection Admin is at sub dictionary list page	
Trigger	N/A	
Post-Condition	Sub dictionary is updated and save into the database	
Main flow		
Step	Actor	Action
1	Admin	On the account list page, click the pen icon at the right of the sub dictionary
2	System	Display the update sub dictionary page

3	Admin	Enter new sub dictionary's information into fields, then click "Update"
4	System	Validate input
5	System	Update new sub dictionary's information into the database, then return to sub dictionary list page
Alternative flow 1		
Precondition		
1	Admin	In step 3 of the main flow, Admin leaves any field blank
2	System	Surround the empty field with red lines, require admin to fill that field
Exceptions		N/A

#### UC 11.8 – Delete Sub Dictionary

Use Case ID	UC – 11.8	
Use Name	Delete Sub Dictionary	
Actor	Admin	
Description	Admin delete the sub dictionary	
Precondition	Internet Connection Admin is at sub dictionary list page	
Trigger	N/A	
Post-Condition	Sub Dictionary is deleted from the database	
Main flow		
Step	Actor	Action
1	Admin	On the sub dictionary list page, click the trash bin icon at the right of the sub dictionary information
2	System	Delete the sub dictionary, then re-display the sub dictionary package list
Alternative flow 1		
Precondition		
		N/A
Exceptions		N/A

#### UC 11.9 – Download Dictionary

Use Case ID	UC – 11.9
Use Name	Download Dictionary
Actor	Admin, Android Device
Description	Admin and any Android Device perform downloading the dictionary
Precondition	Internet Connection
Trigger	N/A
Post-Condition	Dictionary is downloaded to the user's device

Main flow		
Step	Actor	Action
1	Admin	On the dictionary package list, click the download icon at the right of the dictionary package's information
2	System	Get the dictionary data
3	System	Transfer the dictionary data to user device
Alternative flow 1		
Precondition		
1	Android Device	At the step 1 of the main flow, Android device request an URL with dictionary package ID to the system
2	System	Get the dictionary data
3	System	Transfer the dictionary data to user device
Exceptions		N/A

### 3.2.2.4.2 Screen description

Screen	Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
Sub Dictionary List page	Sub Dictionary table	The table containing sub dictionary information	-	-	Table	-	-
	ID column	The ID column of the list of sub dictionary	-	-	Table header	-	-
	Sub Dictionary name column	The name column of the list of sub dictionary	-	-	Table header	-	-
	Description column	The description column of the list of sub dictionary	-	-	Table header	-	-
	Version	The version column of the list of sub dictionary	-	-	Table header	-	-
	Toolbox colum	The toolbox column of the list of sub dictionary, containing button related to the dictionary of the same row	-	-	Table header	-	-
Add sub dictionary page	Sub Dictionary name	The name of the new sub dictionary Default value: none	Y	Y	Text field	Text	30

Screen	Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
	Description	The description of the new sub dictionary Default value: none	Y	N	Text area	Text	250
	Dictionary ID	The ID of the dictionary package, this field is fixed Default value: the dictionary package's ID	Y	Y	Text field	Text	30
	Add button	The add sub dictionary form's button	-	-	Button	-	-
Update dictionary page	Sub Dictionary ID	The ID of the current sub dictionary, this field is fixed Default value: current sub dictionary's ID	Y	Y	Text field	Text	5
	Sub Dictionary name	The name of the current sub dictionary Default value: current sub dictionary's name	Y	N	Text field	Text	30
	Description	The description of the current sub dictionary Default value: current sub dictionary's description	Y	N	Text area	Text	250
	Version	The version of the current sub dictionary Default value: current sub dictionary's version	Y	N	Text field	Text	10
	Dictionary ID	The ID of the dictionary package, this field is fixed Default value: the dictionary package's ID	Y	Y	Text field	Text	30
	Update button	The add sub dictionary form's button	-	-	Button	-	-

### 3.2.2.5 History Logging

#### 3.2.2.5.1 User cases

*UC 12.1 – View History List*

ID	Action date	Account name	Action	Dictionary ID
1	18/6/2015	Hoang	Add Ja-Vi dictionary	101
2	20/6/2015	Thai	Add Kanji sub dictionary	101

Use Case ID	UC – 11.5	
Use Name	View History List	
Actor	Admin	
Description	Admin view history list	
Precondition	Internet Connection	
Trigger	N/A	
Post-Condition	N/A	
Main flow		
Step	Actor	Action
1	Admin	Click the History List link on the tab bar
2	System	Get the history list
4	System	Display the history list on the screen
Alternative flow 1		
Precondition		

		N/A
Exceptions		N/A

UC 12.2 – Update History

Use Case ID		UC – 12.2
Use Name		Update History
Actor		System
Description		System log history into the database
Precondition		N/A
Trigger		UC – 11.2, 11.3, 11.4, 11.6, 11.7, 11.8
Post-Condition		New history entry is generated and added into the database
Main flow		
Step	Actor	Action
1	Admin	Complete any use case in Trigger
2	System	Generated new history entry
5	System	Add new history entry into the database
Alternative flow 1		
Precondition		
		N/A
Exceptions		N/A

**3.2.2.5.2 Screen description**

Screen	Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
History List page	History table	The table containing history information	-	-	Table	-	-
	ID column	The ID column of the list of history entries	-	-	Table header	-	-
	Action date column	The action date column	-	-	Table header	-	-
	Account name column	The account name of admins who created the entry	-	-	Table header	-	-
	Action	The action of the admins	-	-	Table header	-	-

Screen	Field Name	Description	Mandatory	Read only	Control type	Data Type	Length
	Dictionary ID	The ID of dictionary package that has been made change	-	-	Table header	-	-

## 4. NON – FUNCTIONAL REQUIREMENTS

### 4.1. Usability

- User interface should be clear and easy to user
- User need no more than 10 minutes to learn how to use the application
- User need no more than 10 minutes to learn how to use the server

### 4.2. Availability

- Available time of the application:  $\geq 95\%$
- Available time of the server:  $\geq 95\%$

### 4.3. Performance

- Time for image processing: less than 10s
- Time for searching a word in dictionary: less than 3s

### 4.4. Interface

#### 4.4.1. User interface

- User interface will be designed base on flat design
- User interface for application must be simple and clear.

#### 4.4.2. Hardware interface

- The application can run on smart phone that installs Android version 4.2 or higher

#### 4.4.3. Software interface

- Tesseract open ORC library for Japanese recognition
- OpenCV library for image processing

### 4.5. Security

- Secure user accounts on server

## IV. SOFTWARE DETAIL DESIGN

### 1. ARCHITECTURE DESIGN

#### 1.1. Introduction

##### 1.1.1. Purpose

The Software Architecture Document (SAD) provides a comprehensive architectural overview of the Ankipan application 1.0 offered by Ankipan team. It present a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

##### 1.1.2. Scope

The Scope of this SAD is to depict the architecture design of the Ankipan application create by Ankipan team.

##### 1.1.3. Overview

In order to provide an overview of this document, the SAD contains the following subsection:

1. Section 2: describes the use of each view in 4+1 views
2. Section 3: describe the architecture constrains if the system
3. Section 4: describe the functional requirements with its flow and notice.
4. Section 5: describe the most important use-case realization, and architecture of each function
5. Section 6: describe the dynamic aspect of the system, explain the system process and how they communicate
6. Section 7: describe how the system deployed, the topology of software component of the physical layer, as well as the connection between these components
7. Section 8: describe the layers and subsystems of the application
8. Section 9: describe the layers and subsystems of the server
9. Section 10: describe the Image processing lib
10. Section 11: describe any significant persistent element
11. Section 12: describe any non-functional requirement as known as performance and constraints
12. Section 13: describe any aspects related to the quality of service (QoS) attribute

## 1.2. Description of Architecture Design

This document depicts the architecture using the views defined in the “4+1” model [KRU41] as below:

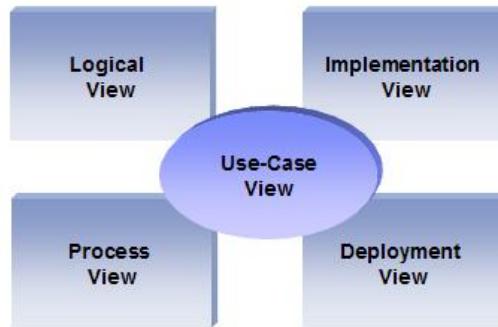


Figure 35: View Model 4 + 1

### Use case view

**Audience:** all the stack holder of the system

**Area:** describes the set of the scenarios and/ or use cases that represent some significant, central functionality of the system.

**Related Artifacts:** Use-Case Model

### Logical view

**Audience:** Designer

**Area:** Function Requirements: describes the design's object model. Also describes the most important use-case realizations

**Related Artifacts:** Class diagram, Sequence diagram, Communication diagram

### Implementation view

**Audience:** Programmers

**Area:** Software components: describes the layers and subsystem of the application

**Related Artifacts:** Component diagram, package diagram

### Deployment view

**Audience:** Manager

**Area:** Topology: describes the mapping of the software onto the hardware and shows the system's distributed aspects.

**Related Artifacts:** Deployment diagram

### Process view

**Audience:** Integrators

**Area:** non-functional requirements: describes the designer's concurrency and synchronization aspects

**Related Artifacts:** Activity diagram

## 1.3. Architecture Goals & Constraints

This section describe the software requirements and objectives that have show significant impact in the architecture

### 1.3.1. Technical Platform

This application will be deployed as an Android standalone application.

The AnkiPan Web Application will be deployed onto a JEE 7 Platform (Apache Tomcat 8.0.9, as it is already the application server used for internal application)

### 1.3.2. Communication

AnkiPan application will send a String to Google API to translate text and, gets data's information from our own server.

### 1.3.3. Security

The system must be secure because of user's information stored in our own web server. The system must be secured, especially user data

### 1.3.4. Persistence

Data in web server will be stored by relational database.

But data in application, specifically dictionary data will be stored with no relation, because of performance.

Login using username and password

### 1.3.5. Reliability/Availability

Because this application is a standalone app, so we have no trouble with Android application's Availability

The availability is the key requirement, as it is required to perform downloading action from any Android device.

Targeting availability is 14/7: 14 hours a day, 7 days a week

Time left (10 hours) is reserved for any maintenance activities

### 1.3.6. Performance

Any process related to password must take about 1-2 seconds to get response from server (for security reason)

## 1.4. User Case View

This section lists all use cases of AKP system.

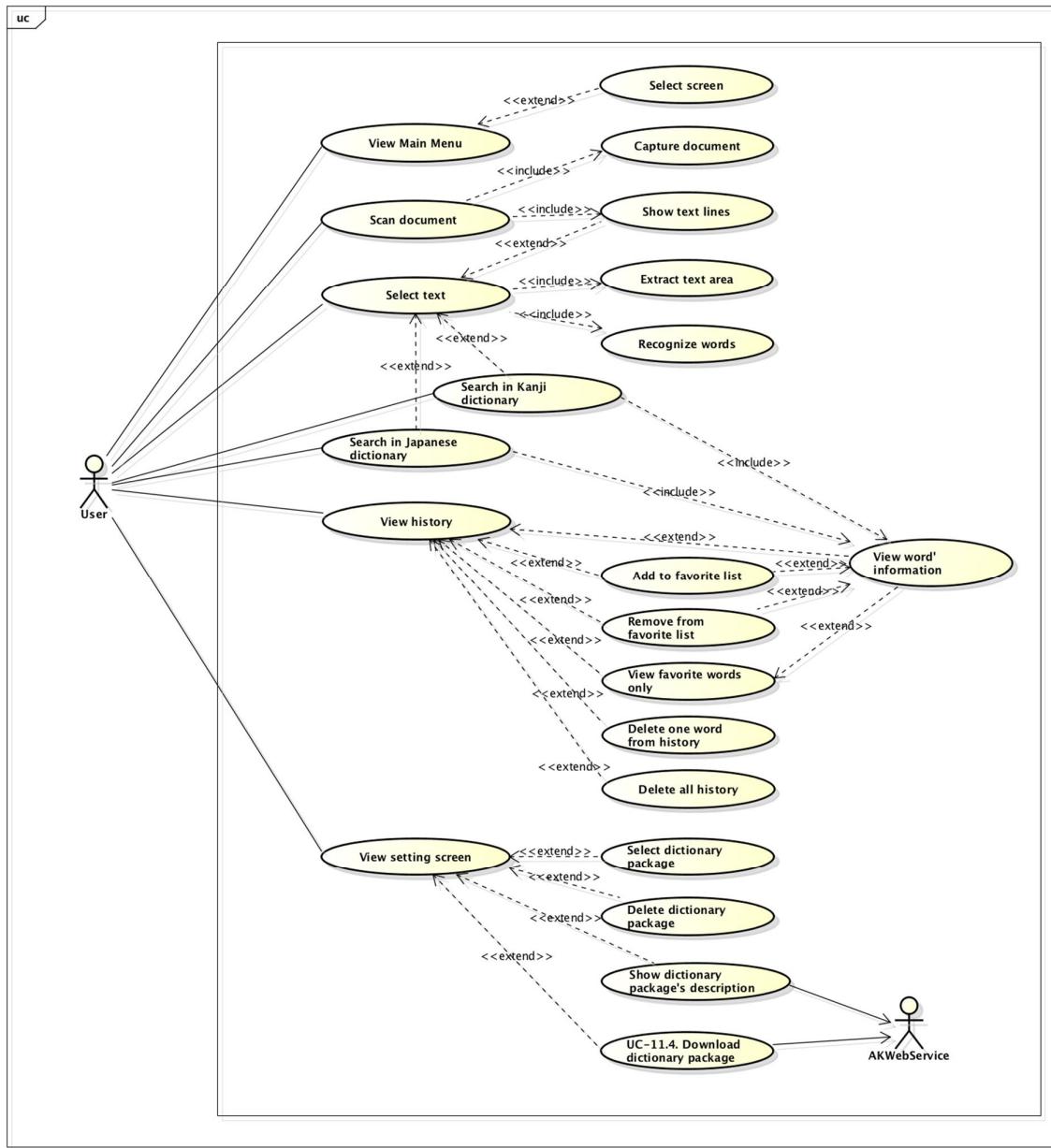


Figure 36: Overall application Use Case

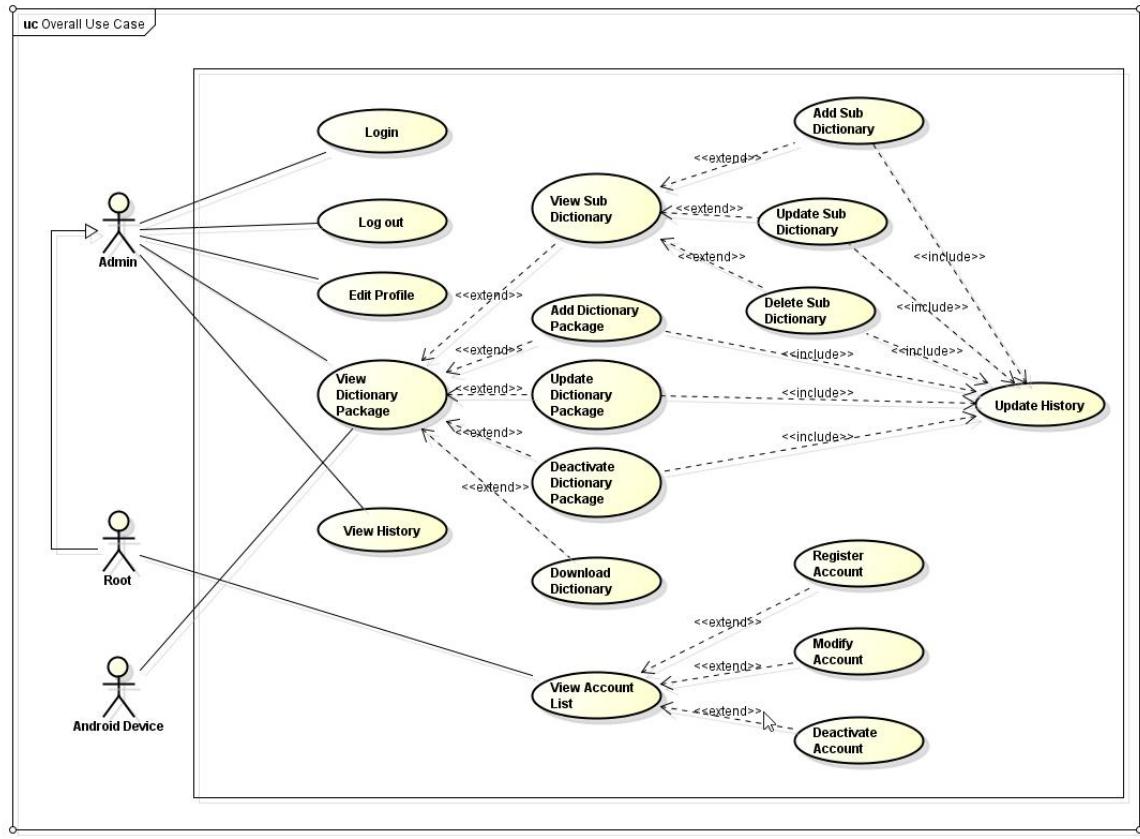


Figure 37: Overall webserver Use Case

## 1.5. Deployment View

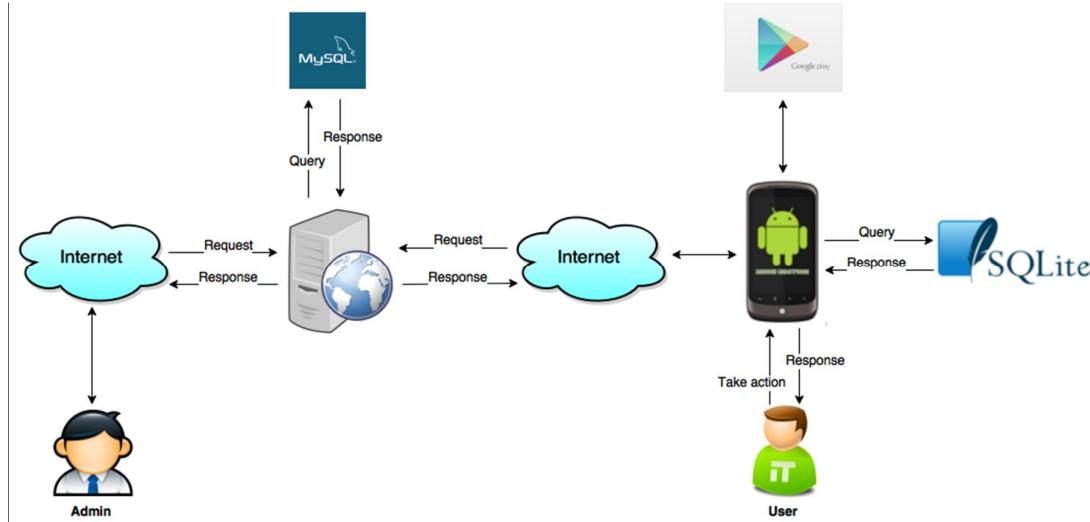


Figure 38: Overview of Anki Pan system

AKP system includes Anki Pan application, server and 2 main kinds of user: admin and user.

User uses Anki Pan application. Anki Pan application queries data from SQLite database to response user's request in need. New version of Anki Pan application will be uploaded to Google play store for user to download and update the application. Admins of Anki Pan system upload dictionary packages to MySQL database on server. User can download dictionary packages from server to application.

## 1.6. Implementation View

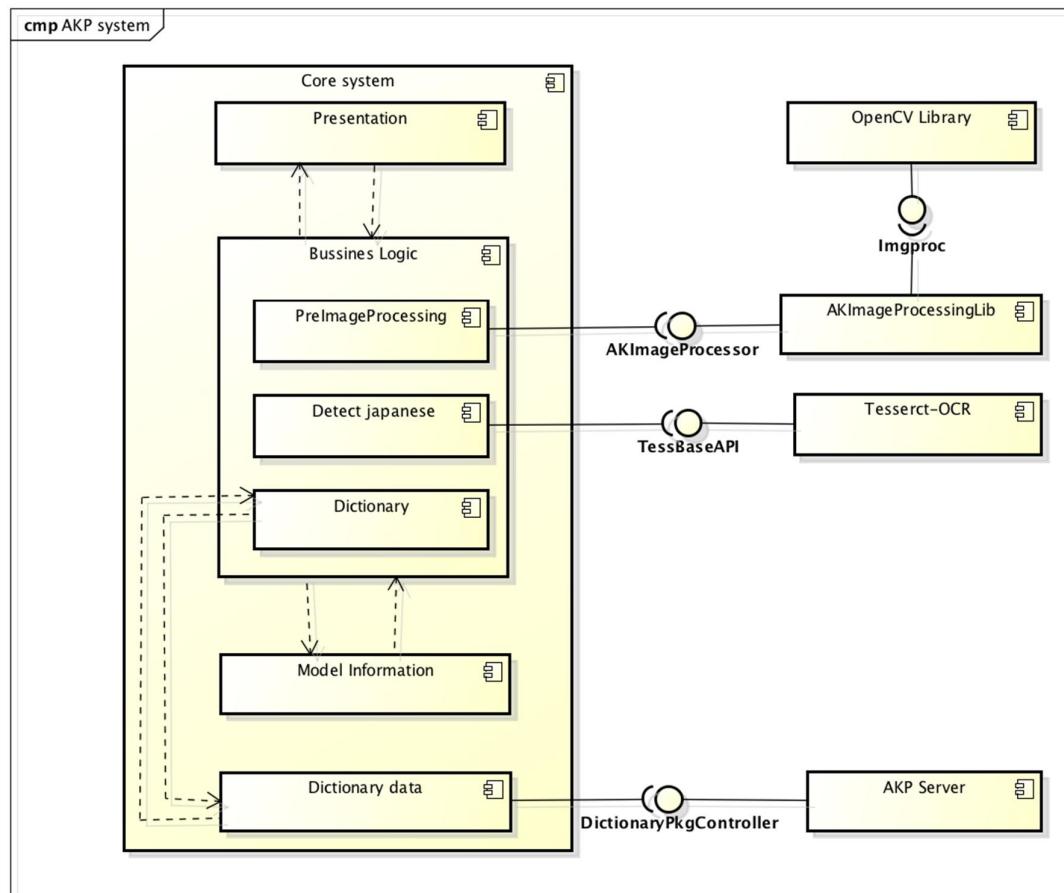


Figure 39: Component Diagram

The system includes a core system to implement required functions on application and a server to upload dictionary packages. Development team will also develop a library to process image, using OpenCV library. Tesseract open source OCR will be used to recognize text.

## 1.7. Detail Design of Application

### 1.7.1. Overview

This application base on MVC model and using Android Studio IDE to program.

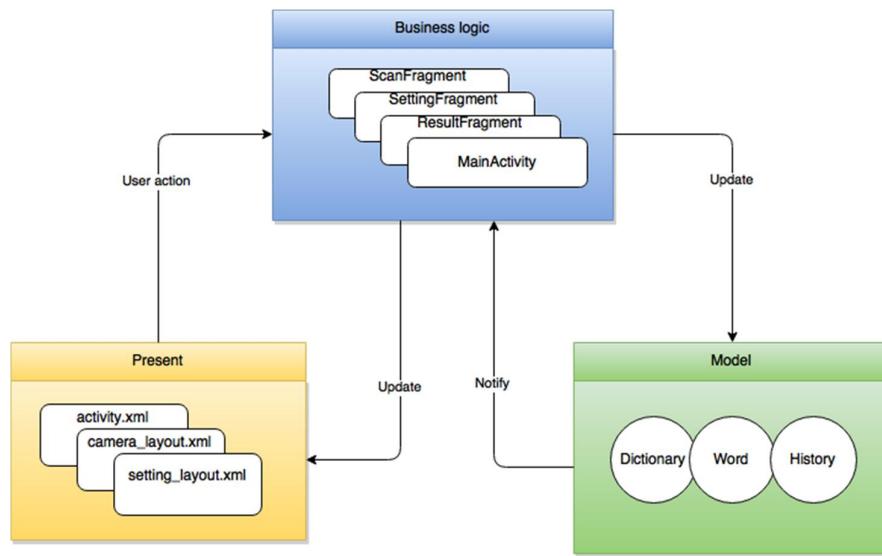


Figure 40: Overview of application architecture

## 1.7.2. Design

### 1.7.2.1. View Main Menu

#### 1.7.2.1.1. Use Case specification

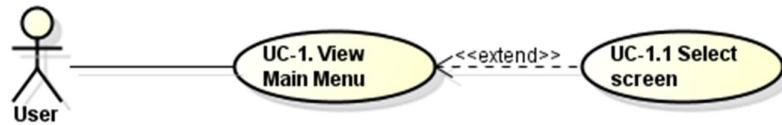


Figure 41: MVC model

- ❖ **Description:** User can be view all application's menu by tapping Menu button on the left top of display screen.

### 1.7.2.1.2. Logical view

#### ❖ Class diagram

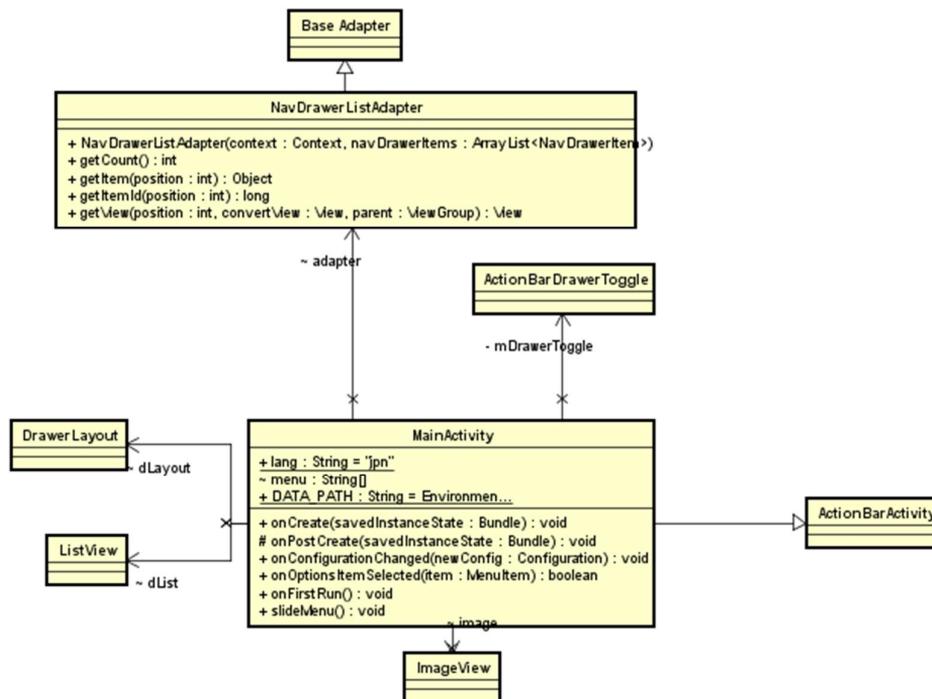


Figure 42: View Main Menu Classes Diagram

#### ❖ Class Specification:

##### 1. MainActivity class:

###### a. Attributes:

No	Name	Type	Description
1	lang	Static final String	This variable contain Tesseract's train data file extension
2	menu	String[]	This variable contain all of view screen's name
3	DATA_PATH	Static final String	Contain application's folder destination in external memory of smartphone

###### b. Methods

OnCreate:

Purpose Return	Create an activity when at its first call			
	Type	Description		
void				
Parameters	No	Name	Type	Description
	1	savedInstanceState	Bundle	Contain bundle that transfer from calling method

## OnPostCreate:

Purpose	Called when activity start-up is complete (after onStart() and onRestoreInstanceState(Bundle) have been called). Applications will generally not implement this method; it is intended for system classes to do final initialization after application code has run.		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	savedInstanceState	Bundle

If the activity is being re-initialized after previously being shut down then this Bundle contains the data it most recently supplied in onSaveInstanceState(Bundle)

## onOptionsItemSelected:

Purpose	Called by the system when the device configuration changes while your activity is running. Note that this will only be called if you have selected configurations you would like to handle with the configChanges attribute in your manifest. If any configuration change occurs that is not selected to be reported by that attribute, then instead of reporting it the system will stop and restart the activity (to have it launched with the new configuration).		
Return	Type		Description
	boolean		Return false to allow normal menu processing to proceed, true to consume it here.
Parameters	No	Name	Type
	1	item	MenuItem

The menu item that was selected.

## onConfigurationChange:

Purpose	Called whenever an item in your options menu is selected. The default implementation simply returns false to have the normal processing happen (calling the item's Runnable or sending a message to its Handler as appropriate). You can use this method for any items for which you would like to do processing without those other facilities.		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	newConfig	Configuration

The new device configuration.

## onFirstRun:

Purpose	Called on the first time running of application		
---------	---	--	--

Return	Type			Description
	void			
Parameters	No	Name	Type	Description

slideMenu:

Purpose	Called when application open			
Return	Type			Description
	void			
Parameters	No	Name	Type	Description

**2. NavDrawerListAdapter:***a. Methods*

getCount:

Purpose	How many items are in the data set represented by this Adapter			
Return	Type			Description
	int			
Parameters	No	Name	Type	Description

getItem:

Purpose	Get the data item associated with the specified position in the data set			
Return	Type			Description
	Object			
Parameters	No	Name	Type	Description
	1	position	int	Position of the item whose data we want within the adapter's data set

getView

Purpose	Get a View that displays the data at the specified position in the data set. You can either create a View manually or inflate it from an XML layout file. When the View is inflated, the parent View (GridView, ListView...) will apply default layout parameters unless you use inflate(int, android.view.ViewGroup, boolean) to specify a root view and to prevent			
---------	--	--	--	--

	attachment to the root.			
Return	Type			Description
	View			A View corresponding to the data at the specified position
Parameters	No	Name	Type	Description
	1	position	int	The position of the item within the adapter's data set of the item whose view we want
	2	convertView	View	The old view to reuse, if possible. Note: You should check that this view is non-null and of an appropriate type before using. If it is not possible to convert this view to display the correct data, this method can create a new view
	3	parent	ViewGroup	The parent that this view will eventually be attached to

### ❖ Sequence Diagram:

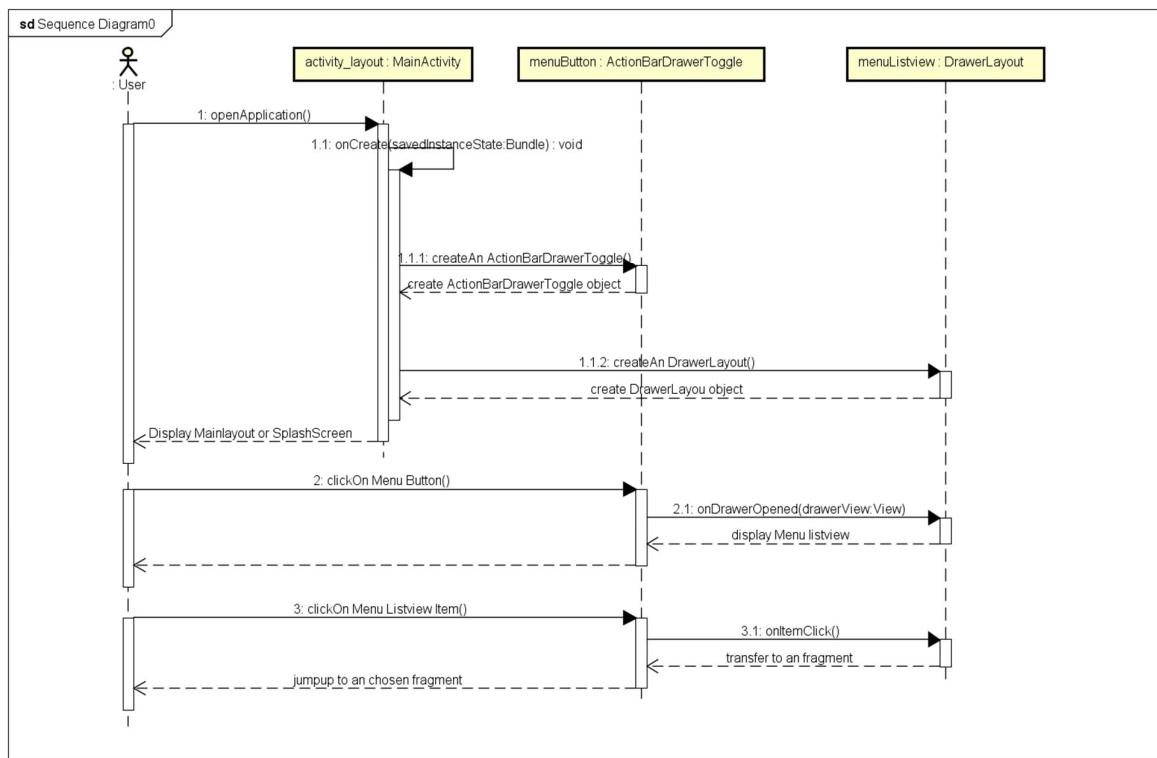


Figure 43: View Main Menu Sequence Diagram

powered by Astah

### 1.7.2.1.3. Activity Diagram

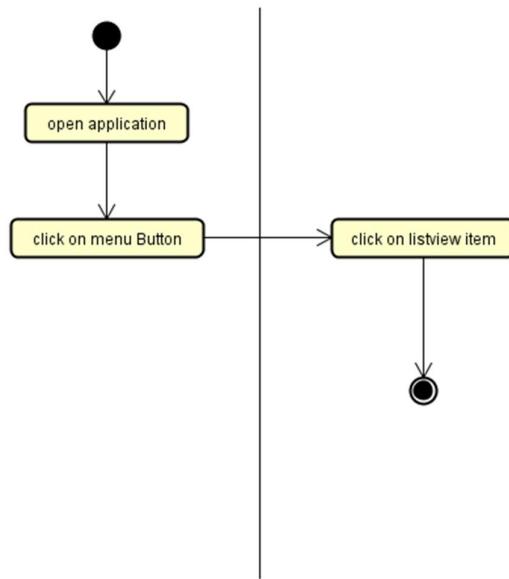


Figure 44: View Main Menu Activity Diagram

### 1.7.2.2. Scan Document

#### 1.7.2.2.1. Use Case specification

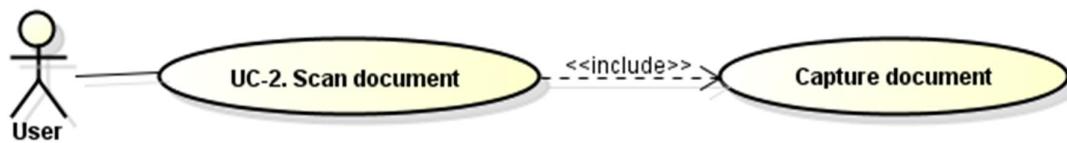


Figure 45: Scan Document Use Case

- ❖ **Description:** User can scan document by capture an image.

#### 1.7.2.2.2. Logical view

- ❖ **Class Diagram:**

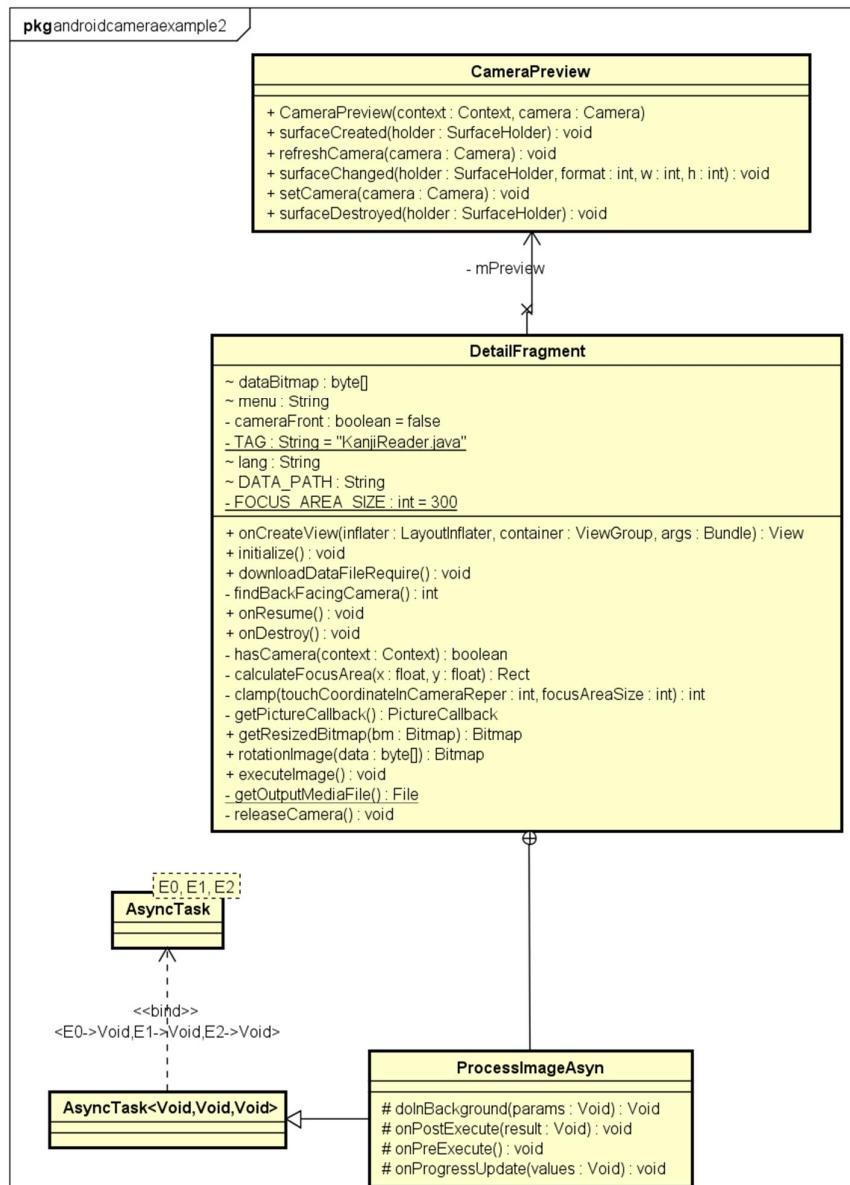


Figure 46: Scan Document Class Diagram

## ❖ Class Specification:

### 1. DetailFragment class:

#### a. Attributes:

No	Name	Type	Description
1	Bitmap	Data[]	This variable contain image information's datas
2	TAG	String	This variable contain an TAG name
3	DATA_PATH	String	Contain application's folder destination in external memory of smartphone
4	Lang	String	This variable contain all of view screen's name
5	FOCUS_AREA_SIZE	int	Contain area size to focus in display

*b. Methods*

downloadDataFileRequire:

Purpose	Check if lang.train data file exists if not, download and initial it from internet		
Return	Type		Description
	void		
Parameters	No	Name	Type
			Description

findBackFacingCamera:

Purpose	Check if device has an back Camera		
Return	Type		Description
	int		If device has no camera return -1. If it has, return camera's id
Parameters	No	Name	Type
			Description

onResume:

Purpose	Reopen Camera		
Return	Type		Description
	void		
Parameters	No	Name	Type
			Description

onDestroy:

Purpose	Release Camera		
Return	Type		Description
	void		
Parameters	No	Name	Type
			Description

hasCamera:

Purpose	Check if Camera is exists or not (call in <b>findBackFacingCamera</b> method)		
Return	Type		Description
	boolean		If device has an camera return true, and false if it is not
Parameters	No	Name	Type
	1	context	Context

calculateFocusArea:

Purpose	Calculate an focus area in display screen		
Return	Type		Description
	Rect		Position of x, y and width, height of focus rectangular
Parameters	No	Name	Type
	1	x	float
	2	y	float

getResizedBitmap:

Purpose	Calculate and resize an image to fit display		
Return	Type		Description
	Bitmap		
Parameters	No	Name	Type
	1	bitmap	Bitmap

rotationImage:

Purpose	Rotate image to portrait mode		
Return	Type		Description
	Bitmap		Image after rotation
Parameters	No	Name	Type
	1	data	byte[]

executeImage:

Purpose	Scan and get text lines		
Return	Type		Description
	void		
Parameters	No	Name	Type
			Description

releaseCamera:

Purpose	Release camera after capture or change activity		
Return	Type		Description
	void		
Parameters	No	Name	Type
			Description

## 2. Camera Preview:

*a. Methods:*

surfaceCreated:

Purpose	This is called immediately after the surface is first created. Implementations of this should start up whatever rendering code they desire. Note that only one thread can ever draw into a Surface, so you should not draw into the Surface here if your normal rendering will be in another thread.		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	holder	SurfaceHolder
			The SurfaceHolder whose surface is being created.

refreshCamera:

Purpose	Stop and reopen camera		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	camera	Camera
			Object camera need to refresh

surfaceChanged:

Purpose	This is called immediately after any structural changes (format or size) have been made to the surface. You should at this point update the imagery in the surface. This method is always called at least once, after surfaceCreated(SurfaceHolder).		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	holder	SurfaceHolder
			The SurfaceHolder whose surface has changed.
	2	format	int
			The new PixelFormat of the surface.
	3	width	int
			The new width of the surface.
	4	height	int
			The new height of the surface.

surfaceDestroyed:

Purpose	This is called immediately before a surface is being destroyed. After returning from this call, you should no longer try to access this surface. If you have a rendering thread that directly accesses the surface, you must ensure that thread is no longer touching the Surface before returning from this function.		
---------	--	--	--

Return	Type			Description
	void			
Parameters	No	Name	Type	Description
	1	holder	SurfaceHolder	The SurfaceHolder whose surface is being destroyed.

### 3. ProcessImageAsyn class:

#### a. Methods:

doInBackground:

Purpose	Override this method to perform a computation on a background thread. The specified parameters are the parameters passed to execute(Params...) by the caller of this task. This method can call publishProgress(Progress...) to publish updates on the UI thread.			
Return	Type			Description
	void			A result, defined by the subclass of this task.
Parameters	No	Name	Type	Description
	1	params	Void	The parameters of the task.

onPostExecute:

Purpose	Runs on the UI thread after doInBackground(Params...). The specified result is the value returned by doInBackground(Params...). This method won't be invoked if the task was cancelled.			
Return	Type			Description
	void			
Parameters	No	Name	Type	Description
	1	result	Result	The result of the operation computed by doInBackground(Params...).

onPreExecute:

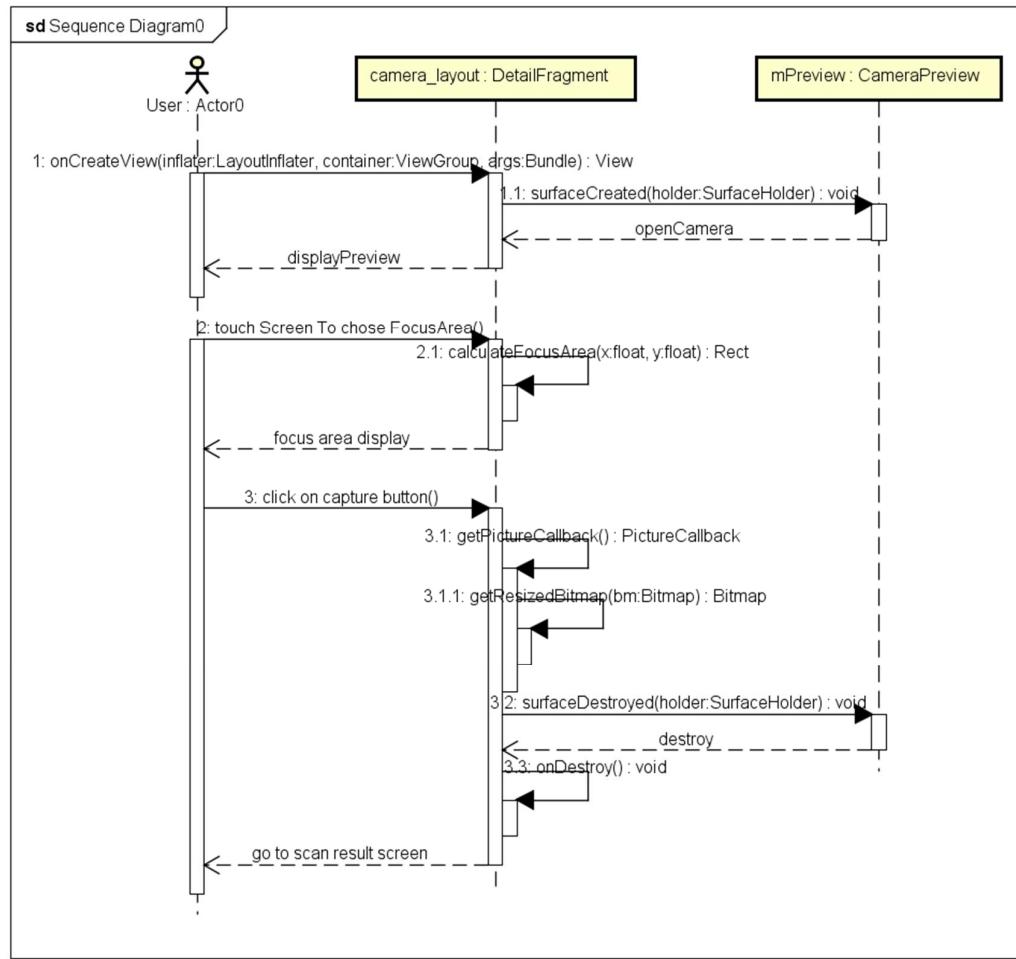
Purpose	Runs on the UI thread before doInBackground(Params...).			
Return	Type			Description
	void			
Parameters	No	Name	Type	Description

onProgressUpdate:

Purpose	Runs on the UI thread after publishProgress(Progress...) is invoked. The specified values are the values passed to publishProgress(Progress...).			
Return	Type			Description

	void			
Parameters	No	Name	Type	Description
	1	value	Void	The values indicating progress.

## ❖ Sequence diagram



powered by Astah

Figure 47: Scan Document Sequence Diagram

### 1.7.2.2.3. Activity Diagram:

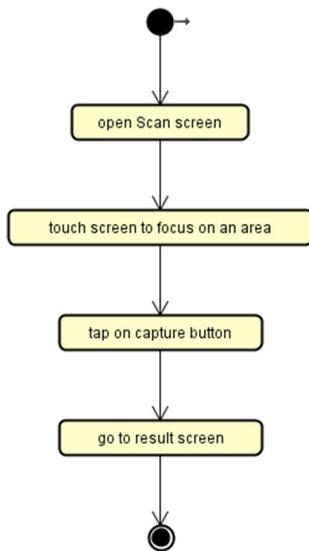
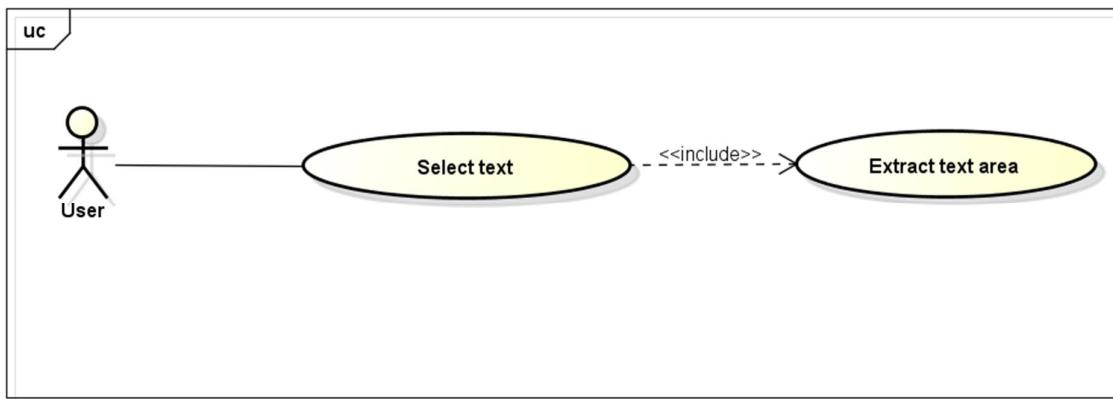


Figure 48: Scan Document Activity Diagram

### 1.7.2.3. Select text

#### 1.7.2.3.1. Use Case specification



Description: After capture image by camera, user can chose text area by touch to start-area-point to end-area-point. Then, the chosen area will be high light and recognize text from this area.

#### 1.7.2.3.2. Logical View

##### ❖ Class diagram

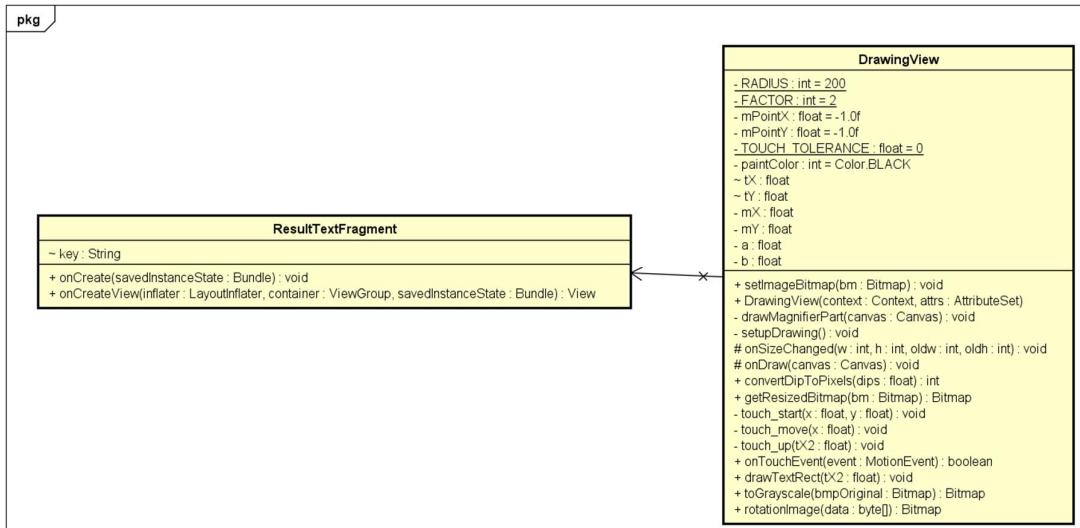


Figure 49: Class diagram of user case Select text

## ❖ Class specification

### 1. Drawing view

#### a. Attribute

No	Name	Type	Description
1	paintColor	int	Color value that defined in class Color in Android
2	tX	float	Current x position that user pointing in
3	tY	float	Current y position that user pointing in
4	mX	float	X position that use in
5	mY	float	
6	a	float	X start point
7	b	float	Y start point

#### b. Methods

##### setImageBitmap

Purpose	Called when init view, this method set src for view in layout xml		
Return	Type	Description	
	void		
Parameters	No	Name	Type
	1	bm	Bitmap
			Description
			This parameter refer an Bitmap variable

##### drawMagnifierPart

Purpose	Called when user touch in View, that will create an magnifier to scale touch area		
Return	Type	Description	
	void		

Parameters	No	Name	Type	Description
	1	canvas	Canvas	This parameter refer an Canvas variable that will be scale on

**setupDrawing**

Purpose	This method call when init some variable use to draw on canvas object that set image to		
Return	Type void		Description
Parameters	No	Name	Type
			Description

**onDraw**

Purpose	This method override from super method, call when user have any action to View		
Return	Type void		Description
Parameters	No	Name	Type
	1	canvas	Canvas

**touch\_start**

Purpose	This method call when user start to touch on View		
Return	Type void		Description
Parameters	No	Name	Type
	1	x	float
	2	y	float

**touch\_move**

Purpose	This method call when user start to move finger on View		
Return	Type void		Description
Parameters	No	Name	Type
	1	x	float
	2	y	float

**touch\_up**

Purpose	This method call when user end of touch		
Return	Type void		Description
Parameters	No	Name	Type
	1	x	float
	2	y	float

**drawTextRect**

Purpose	This method call after user finish action on View, after that, this method start to run and execute information which is collect from these step above		
---------	--	--	--

Return	Type	Description		
	void			
Parameters	No	Name	Type	Description
	1	tX2	float	Refer an x position of the last touch point

### 1.7.2.3.1. Sequence diagram

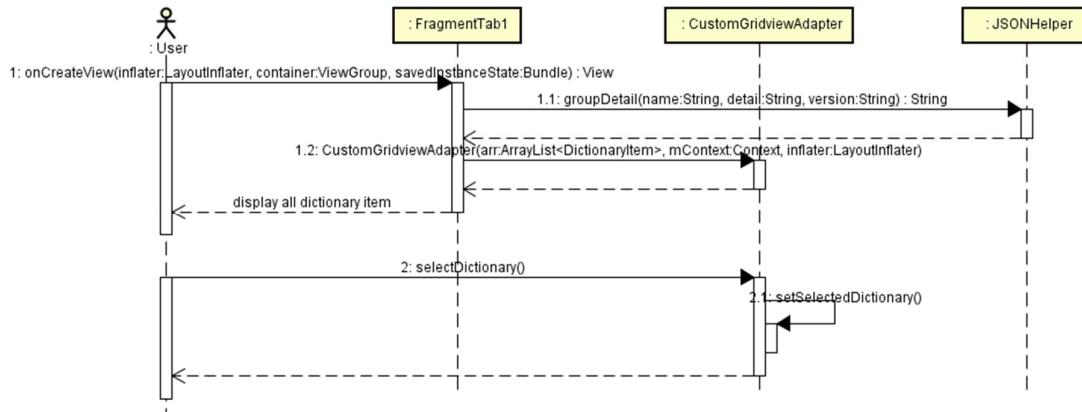


Figure 50: Sequence diagram of user case Select text

### 1.7.2.3.4. Activity diagram

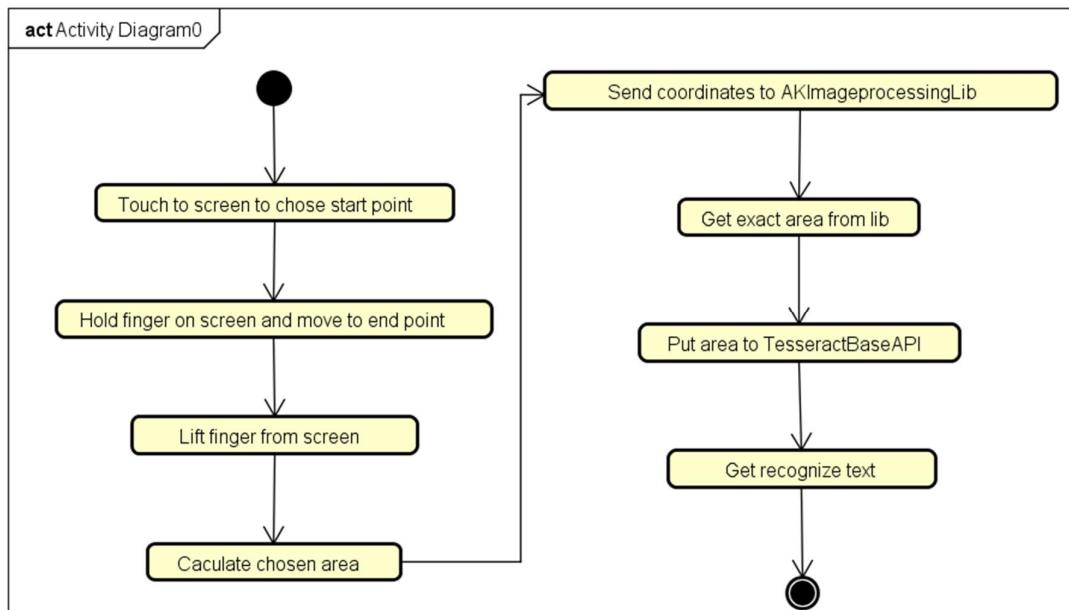


Figure 51: Activity diagram of user case Select text

### 1.7.2.4. Search in Kanji Dictionary

#### 1.7.2.4.1. Use Case specification

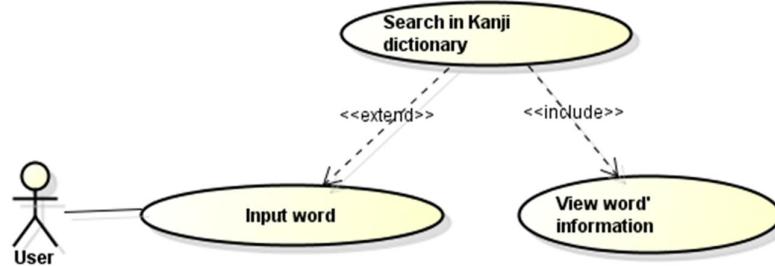


Figure 52: Search Kanji Dictionary Use Case

- ❖ **Description:** After input text or scan text in the previous step, user can search word meaning that already existed in database by click on search button on keyboard.

#### 1.7.2.4.2. Logical View

##### ❖ Class Diagram:

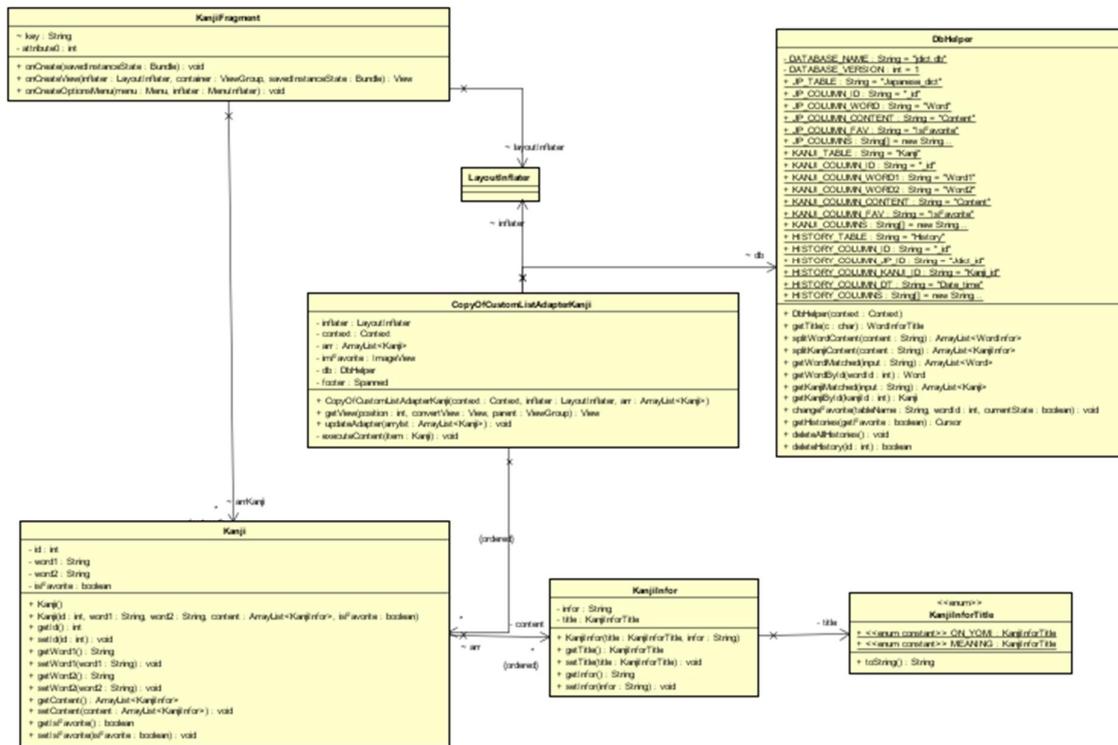


Figure 53: Search Kanji Dictionary Class Diagram

##### ❖ Class Specification:

###### 1. KanjiInfor:

###### a. Attribute:

No	Name	Type	Description
1	Infor	String	This variable contain kanji word's information
2	Title	KanjiInforTitle	This variable contain kanji word's title (like Onyomi and meaning)

b. Method:

getTitle:

Purpose	Called when want to get title of kanji word			
Return	Type		Description	
	KanjiInforTitle			
Parameters	No	Name	Type	Description

setTitle:

Purpose	Called when want to set title of kanji word			
Return	Type		Description	
	void			
Parameters	No	Name	Type	Description
	1	title	KanjiInforTitle	

getInfor:

Purpose	Called when want to get information of kanji word			
Return	Type		Description	
	String			
Parameters	No	Name	Type	Description

setInfor:

Purpose	Called when want to set information of kanji word			
Return	Type		Description	
	Void			
Parameters	No	Name	Type	Description
	1	infor	String	

## 2. Kanji class:

a. Attribute:

No	Name	Type	Description
1	Id	int	This variable contain id of kanji word in

			database
2	word1	String	This variable contain kanji word's in Japanese
3	word2	String	This variable contain kanji word's in other language
4	isFavorite	boolean	If this kanji word is mark to favorite or not
5	Content	ArrayList<KanjiInfor>	Contain content of word (like meaning, onyomi)

a. Methods

Kanji:

Purpose	Constructor create an Kanji object			
Return	Type			Description
	Kanji			
Parameters	No	Name	Type	Description
	1	id	Int	
	2	word1	String	
	3	word2	String	
	4	isFavorite	boolean	
	5	content	ArrayList<KanjiInfor>	

### 3. CopyOfCustomListAdapterKanji Class:

a. Attributes:

No	Name	Type	Description
1	Inflater	LayoutInflater	This variable LayoutInflater of Activity
2	Context	Context	This variable contain context of application
3	Arr	ArrayList<Kanji>	This variable contain arraylist of kanji object
4	imFavorite	ImageView	Point to imageview in xml layout that display this kanji is favorite or not
5	Db	DbHelper	Point to database connector already open
6	Footer	Spanned	Footer of kanji word

a. Methods

getView:

Purpose	Call when create an adapter and fill the list view		
Return	Type		
	View		
Parameters	No	Name	Type

	1	position	int	
	2	convertView	View	
	3	parent	ViewGroup	

updateAdapter:

Purpose	Called when adapter change and new listview have to fill		
Return	Type	Description	
	void		
Parameters	No	Name	Type
	1	arrylst	ArrayList<Kanji>
			New arraylist have to fill in adapter

executeContent:

Purpose	Method change Kanji's word content from String to Html		
Return	Type	Description	
	void		Change content of kanji word from enum to Html form with more color and symbol like interface design
Parameters	No	Name	Type
	1	item	Kanji

#### 1.7.2.4.3. Sequence Diagram:

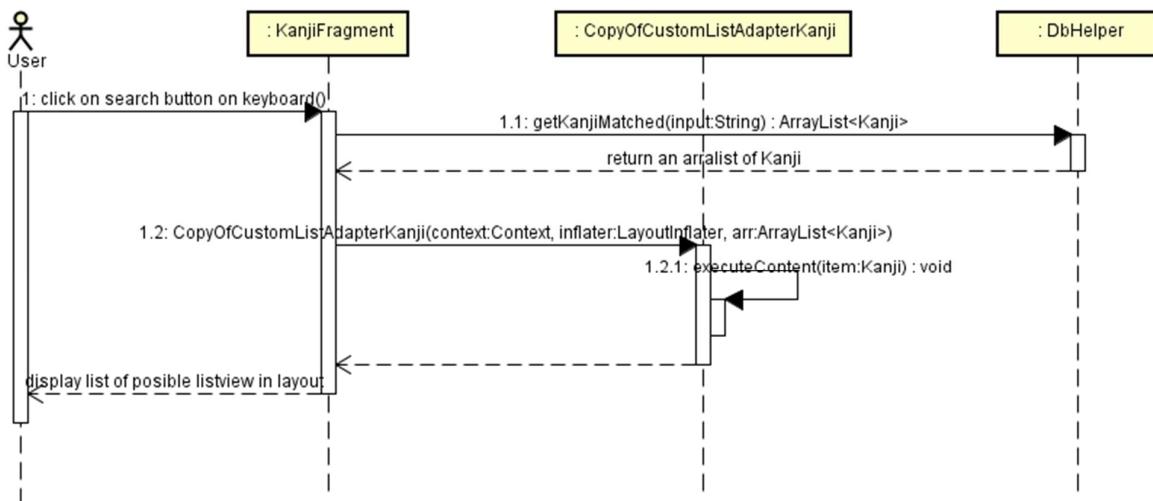


Figure 54: Search Kanji Dictionary Sequence Diagram

#### 1.7.2.4.4. Activity Diagram

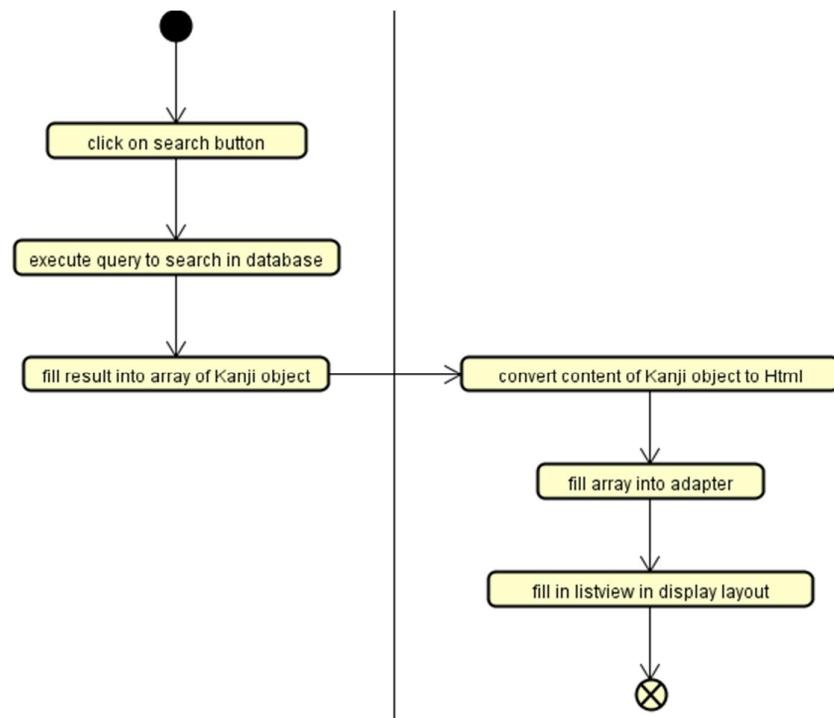


Figure 55: Search Kanji Dictionary Activity Diagram

#### 1.7.2.5. Search in Japanese Dictionary

##### 1.7.2.5.1. Use Case Specification

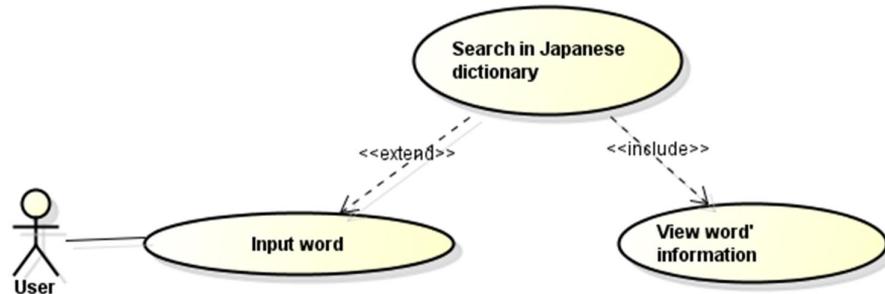


Figure 56: Search Japanese Dictionary Use Case

##### 1.7.2.5.2. Logical View

###### ❖ Class Diagram:

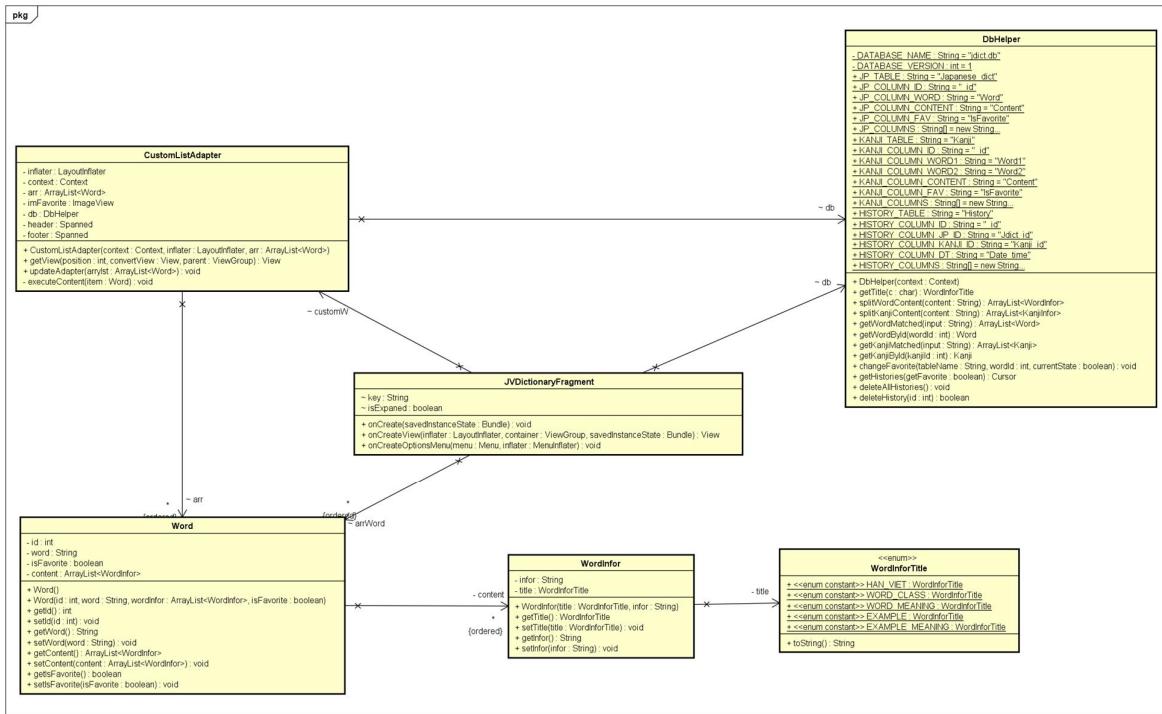


Figure 57: Search Japanese Dictionary Class Diagram

## ❖ Class Specification:

### 1. WordInfor Class:

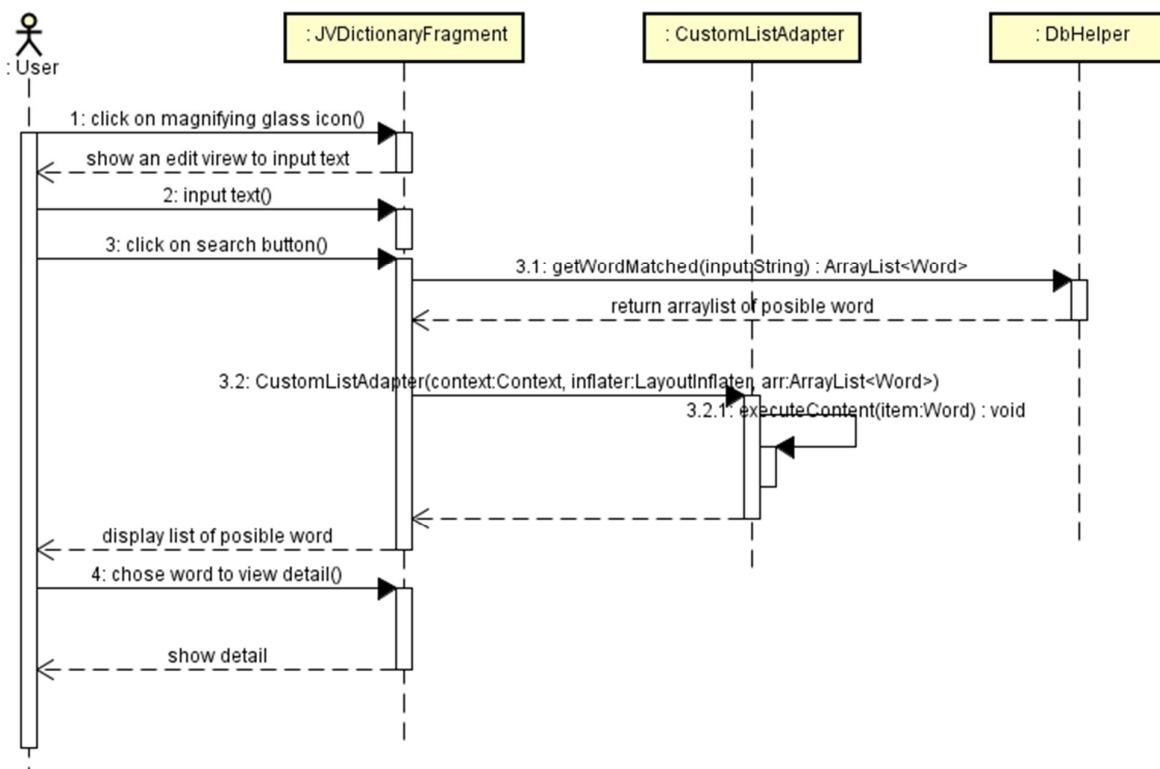
#### a. Attribute:

No	Name	Type	Description
1	Infor	String	This variable contain kanji word's information
2	Title	WordInforTitle	This variable contain word's title (like example and meaning)

#### b. Method:

`getTitle:`

Purpose	Called when want to get title of word		
Return	Type		Description
Parameters	WordInforTitle	Return an ENUM type variable	
	No	Name	Type

**1.7.2.5.3. Sequence Diagram:***Figure 58: Search Japanese Dictionary Sequence Diagram*

#### 1.7.2.5.4. Activity Diagram

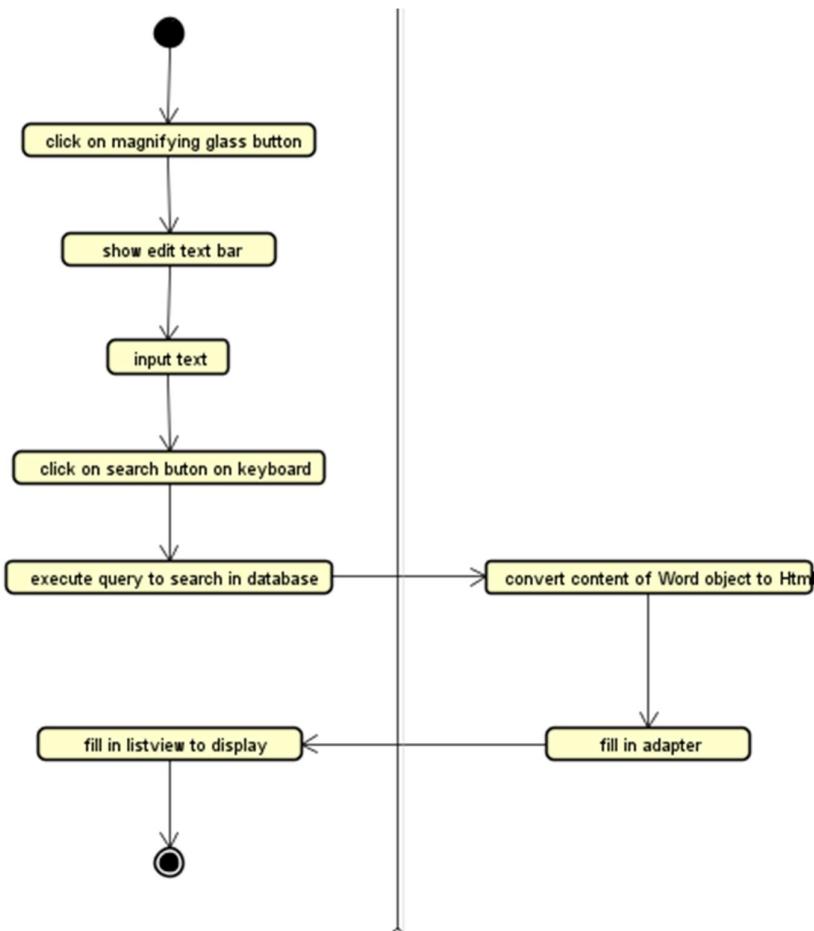


Figure 59: Search Japanese Dictionary Activity Diagram

#### 1.7.2.6. Select Dictionary Package

##### 1.7.2.6.1. Use Case Specification

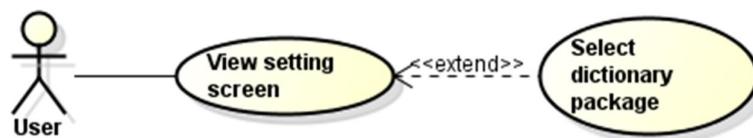


Figure 27 - Select Dictionary Package Use Case

- ❖ **Description:** User can select what dictionary package that search in.

##### 1.7.2.6.2. Logical View

- ❖ **Class Diagram:**

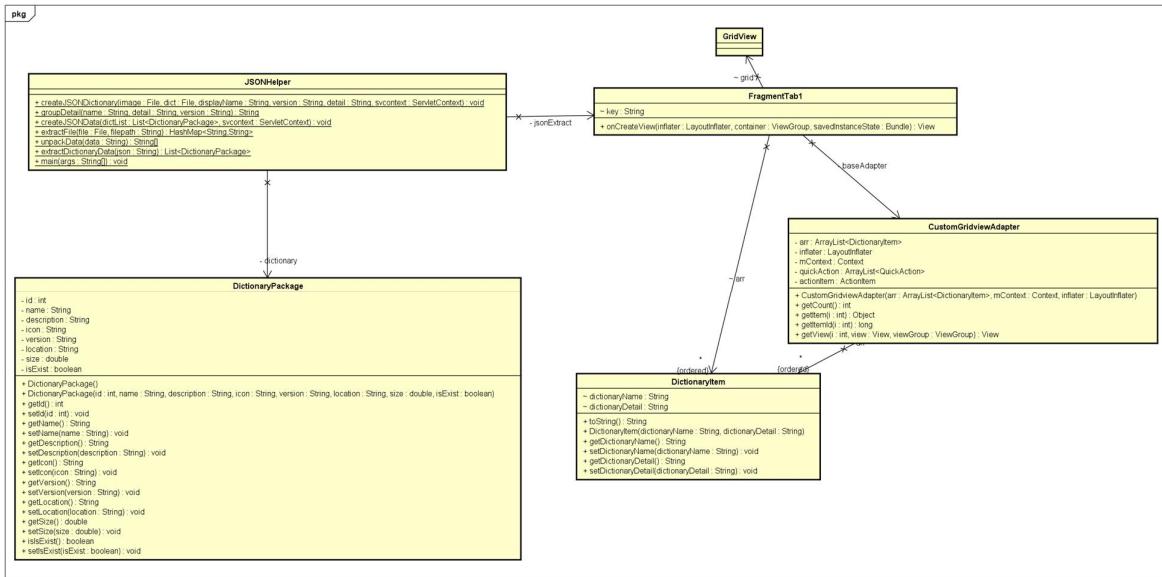


Figure 60: Select Dictionary Package Class Diagram

## ❖ Class Specification:

### 1. CustomGridviewAdapter:

#### a. Attributes:

No	Name	Type	Description
1	arr	ArrayList<DictionaryItem>	
2	inflater	LayoutInflator	
3	mContext	Context	
4	quickAction	ArrayList<Quick Action>	Contain list of quickaction object that add to grid view item tooltips
5	actionItem	ActionItem	Specific tooltips item

#### b. Methods

getView:

Purpose	Call when create an adapter and fill the list view		
Return	Type	Description	
	View	In this method, we create an view that contain information of a dictionary item and it's detail tooltips	
Parameters	No	Name	Type
	1	i	int
	2	view	View
	3	viewGroup	ViewGroup

## 2. JSONHelper:

### a. Method:

extractFile:

Purpose	Extract a json extension file into db extension file and hash map data.		
Return	Type	Description	
	HashMap<String, String>		Return information about dictionary data
Parameters	No	Name	Type
	1	file	File
	2	filepath	String
			Description
			The json extension file that hold dictionary information
			The filepath that will transfer the db extension file into

extractDictionaryData

Purpose	Extract a json string into a list of dictionary package data		
Return	Type	Description	
	List<DictionaryPackage>		Return a list of dictionary package data
Parameters	No	Name	Type
	1	json	String
			Description
			The json string that hold the information of a list of dictionary package

### 1.7.2.6.3. Sequence Diagram:

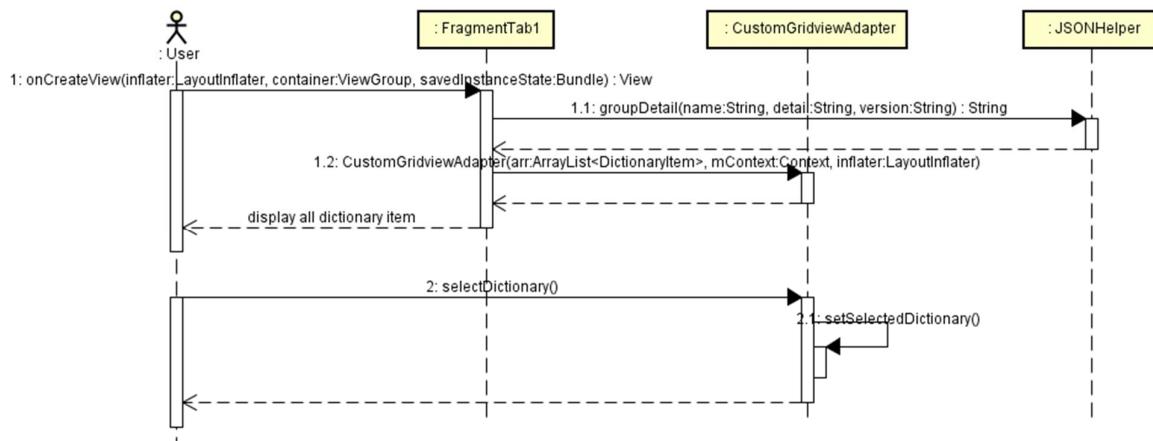


Figure 61: Select Dictionary Package Sequence Diagram

#### 1.7.2.6.4. Activity Diagram

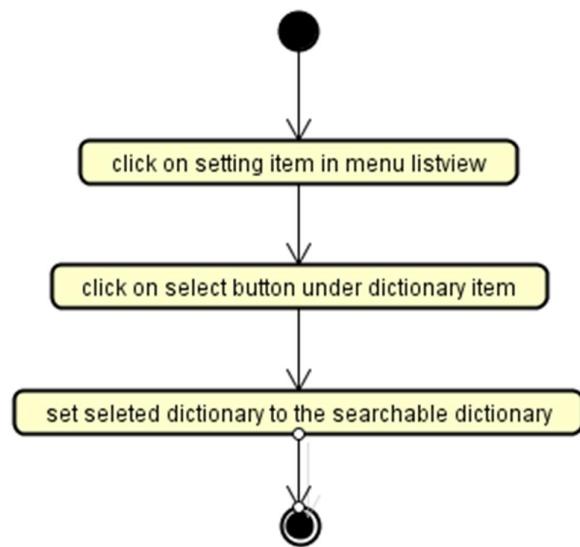


Figure 62: Select Dictionary package Activity Diagram

#### 1.7.2.7. Delete Dictionary Package

##### 1.7.2.7.1. Use Case Specification

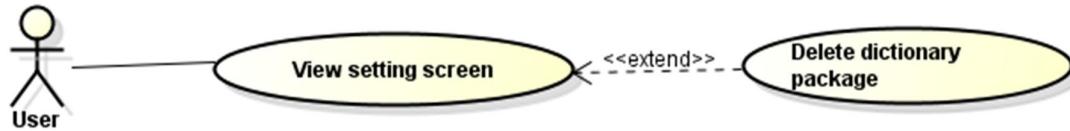


Figure 32 - Delete Dictionary package Use Case

- ❖ **Description:** After download a Dictionary package, user can delete these dictionary from external memory.

##### 1.7.2.7.2. Logical View

- ❖ **Class Diagram:**

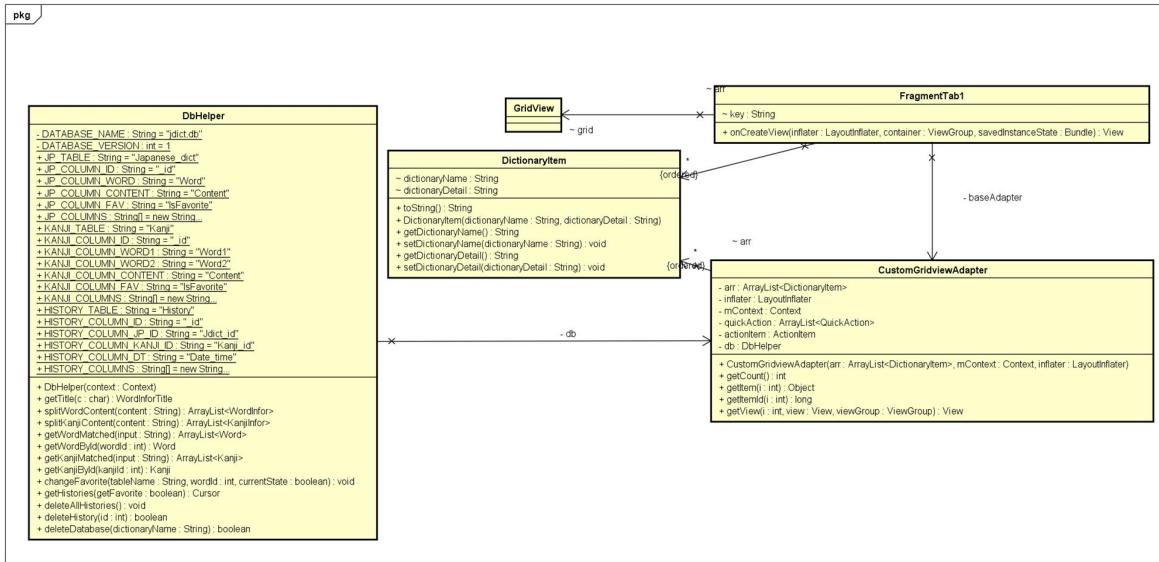


Figure 63: Class diagram for Delete dictionary package user case

## ❖ Class Specification:

### 1. DBHelper:

#### a. Method:

deleteDatabase:

Purpose	Call when user want to delete a dictionary package		
Return	Type	Description	
	boolean	Return true if success, false if its fail	
Parameters	No	Name	Type
	1	dictionaryName	String
			Description
			Name of dictionary package

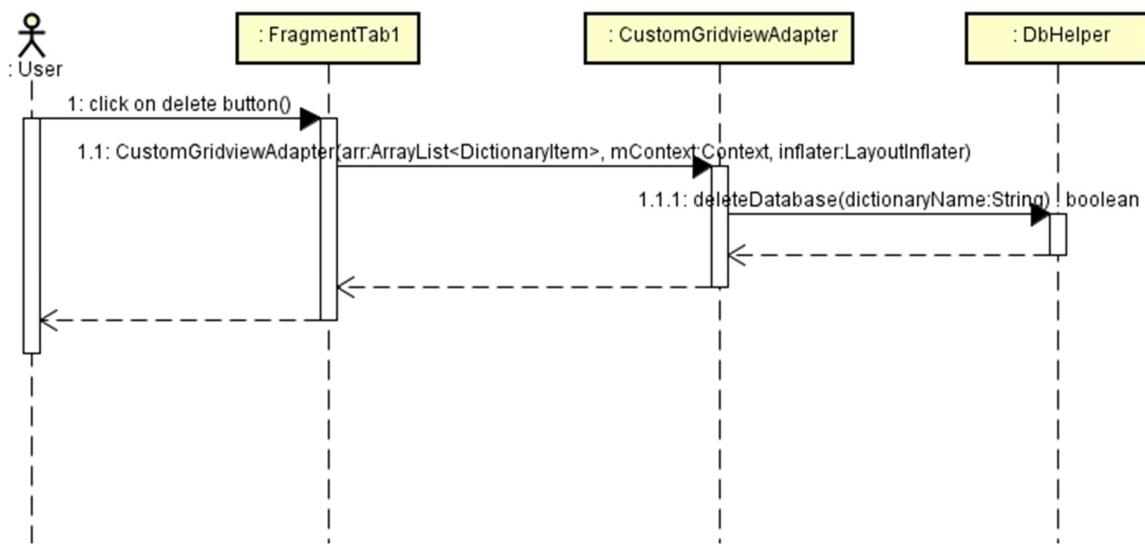
**1.7.2.7.3. Sequence Diagram:**

Figure 64: Sequence diagram of deleting dictionary package

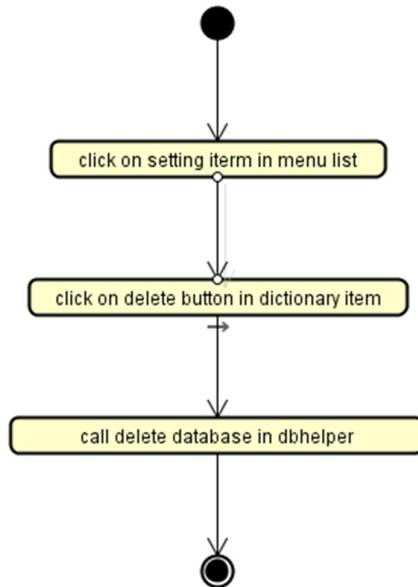
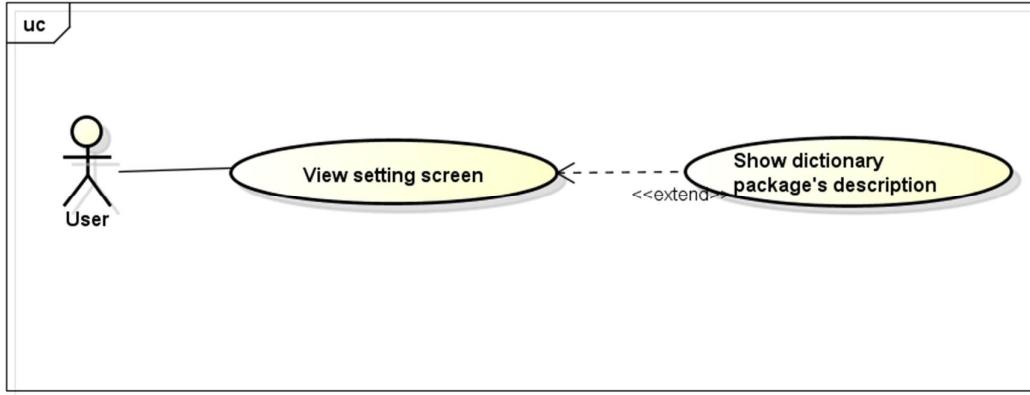
**1.7.2.7.4. Activity Diagram**

Figure 65: Activity diagram of deleting dictionary package

### 1.7.2.8. Show dictionary package's description

#### 1.7.2.8.1. Use Case Specification



- ❖ **Description:** User can see detail information of dictionary package from tooltips that display in setting screen.

#### 1.7.2.8.2. Logical View

##### ❖ Class Diagram:

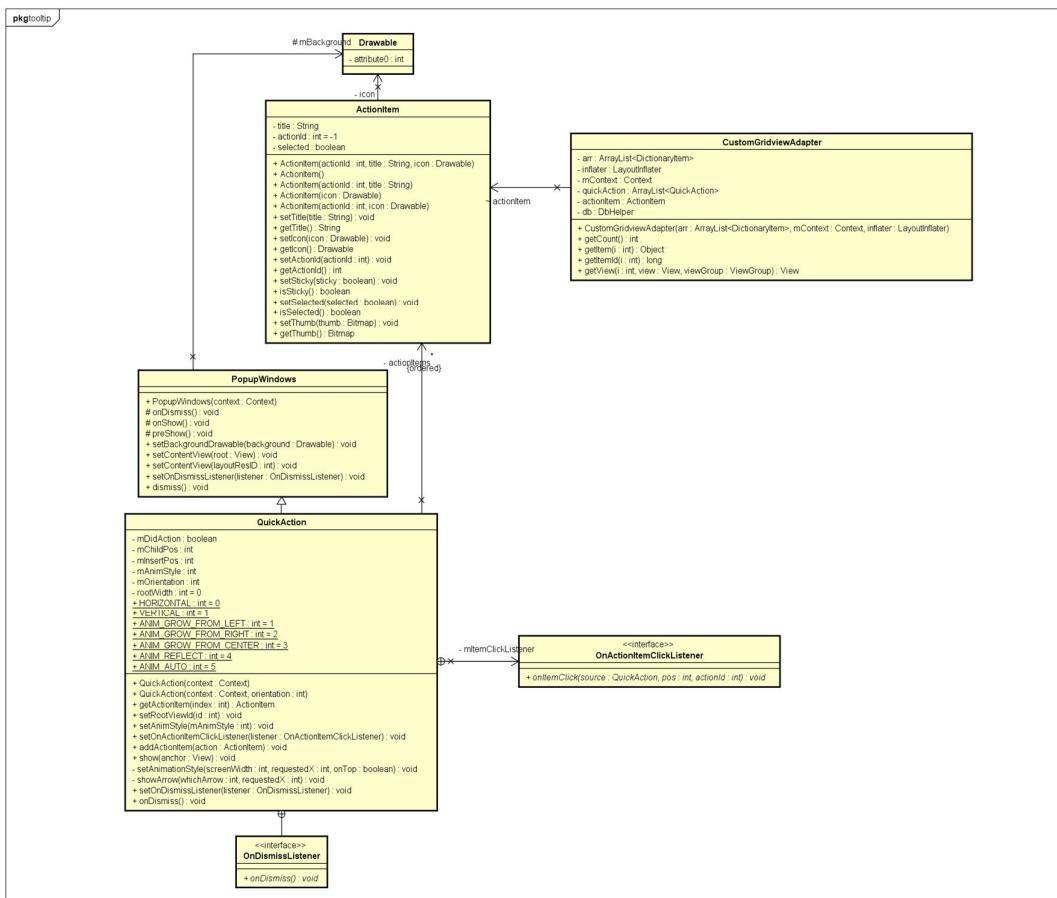


Figure 66: Class diagram of user case Show dictionary package's description

❖ **Class Specification:**

**1. ActionItem class:**

a. Attribute:

No	Name	Type	Description
1	title	String	Title of the button on tooltips
2	actionId	int	
3	selected	boolean	This tooltips is open or not

**2. QuickAction class:**

a. Attribute:

No	Name	Type	Description
1	mDidAction	boolean	The tooltips is showing or not
2	mChildPos	int	Position of child view
3	mInsertPos	int	Count the number of chikd view added
4	mAnimStyle	int	Animation style pass over
5	mOrientation	int	Orientation pass over (HORIZONTAL, or VERTICAL)
6	rootWidth	int	
7	HORIZONTAL	int	
8	VERTICAL	int	
9	ANIM_GROW_FROM_LEFT	int	
10	ANIM_GROW_FROM_RIGHT	int	
11	ANIM_GROW_FROM_CENTER	int	
12	ANIM_REFLECT	int	
13	ANIM_AUTO	int	

b. Method:

getActionItem:

Purpose	Return list of action item		
Return	Type		Description
	List<ActionItem> actionItems		
Parameters	No	Name	Type
			Description

setOnActionItemClickListener:

Purpose	Set action for clicking action
---------	--------------------------------

Return	Type			Description
	void			
Parameters	No	Name	Type	Description
	1	action	Action Item	

addActionItem:

Purpose	Add an action Item			
Return	Type			Description
	void			
Parameters	No	Name	Type	Description
	1	action	Action Item	

show:

Purpose	Show tooltips			
Return	Type			Description
	void			
Parameters	No	Name	Type	Description
	1	anchor	View	View that user click on

setAnimationStyle:

Purpose	Show tooltips			
Return	Type			Description
	void			
Parameters	No	Name	Type	Description
	1	screenWidth	int	Display screen width
	2	requestedX	int	X position of view
	3	onTop	boolean	

showArrow:

Purpose	Show arrow point to clicked view			
Return	Type			Description
	void			
Parameters	No	Name	Type	Description
	1	whichArrow	int	Type of arrow
	2	requestedX	int	X position of view

onDismiss:

Purpose	Dismiss tooltips			
---------	------------------	--	--	--

Return	Type	Description		
	void			
Parameters	No	Name	Type	Description

#### 1.7.2.8.3. Sequence Diagram:

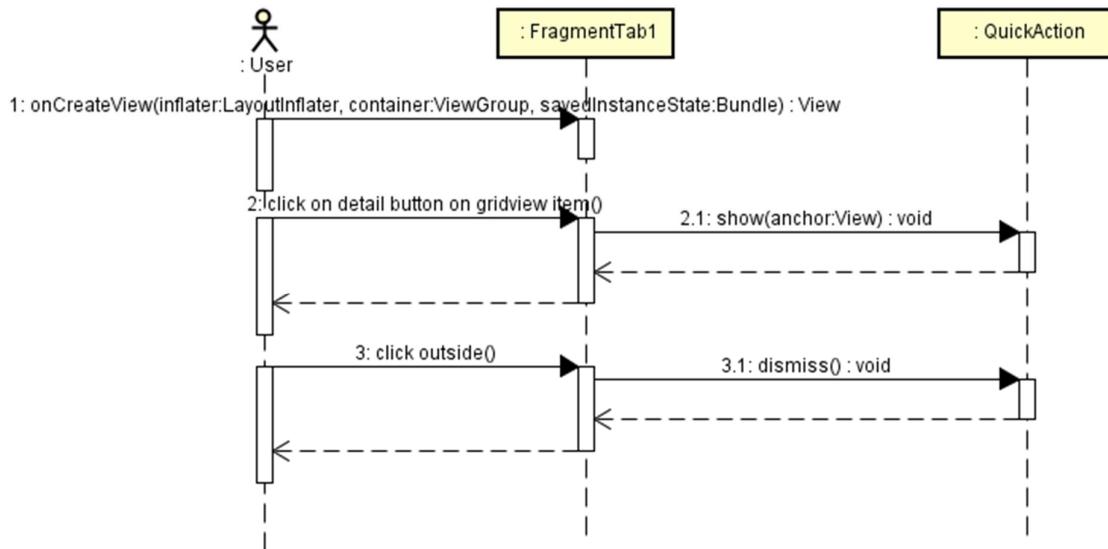


Figure 67: Sequence diagram of user case View dictionary package's description

#### 1.7.2.8.4. Activity Diagram

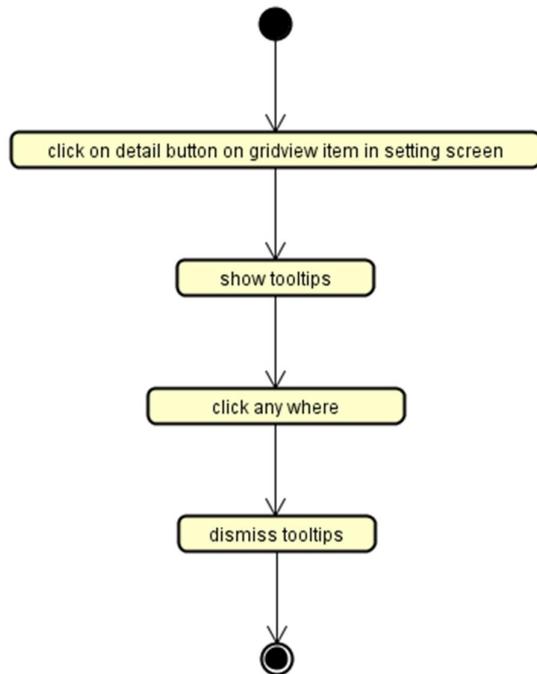


Figure 68: Activity diagram of user case View dictionary package's description

### 1.7.2.9. View History

#### 1.7.2.9.1. Use Case Diagram:

- ❖ **Description:** View user's searching history.

#### 1.7.2.9.2. Logical view

- ❖ **Class diagram**

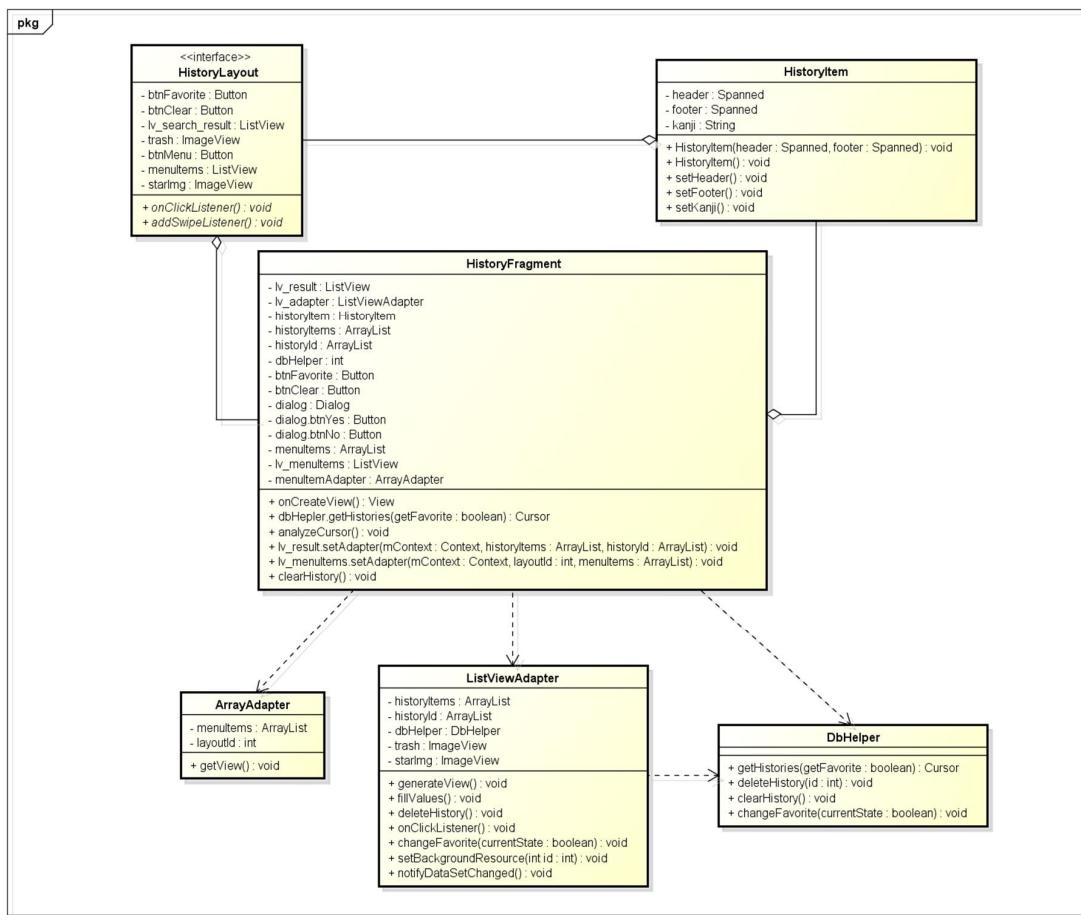


Figure 69: Class diagram of User case View history

- ❖ **Class specification:**

##### 1. HistoryFragment class

###### a. Attribute:

No	Name	Type	Description
1	lv_result	ListView	display all history items
2	lv_adapter	ListView adapter	bind data to every item of ListView (lv_result)
3	historyItem	History Item	store information (header, footer) to display in each ListView item

4	historyItems	ArrayList	list of historyItems
5	historyId	ArrayList	store the id of each history item.
6	dbHelper	DbHelper	used for accessing database

*b. Method:*

lv\_result.setAdapter

Purpose	Create actor's account after actor logins into system.		
Return	Type	Description	
	void		
Parameters	No	Name	Type
	1	mContext	Context
	2	historyItems	ArrayList
	3	historyId	ArrayList

analyzeCursor

Purpose	analyze the Cursor information to add to listview		
Return	Type	Description	
	void		
Parameters	No	Name	Type
			Description

**2. HistoryItem class***a. Attribute:*

No	Name	Type	Description
1	header	Spanned	store data of header TextView in the layout
2	footer	Spanned	store data of footer TextView in the layout

*b. Method*

HistoryItem

Purpose	Create a HistoryItem object		
Return	Type	Description	
	void	this is a constructor to create a HistoryItem object with null attribute	
Parameters	No	Name	Type
			Description

setHeader

Purpose	assign value for header attribute		
Return	Type		Description
	void		Assign value for header attribute
Parameters	No	Name	Type
	1	header	Spanned
			The header attribute of HistoryItem object

**setFooter**

Purpose	assign value for footer attribute		
Return	Type		Description
	void		assign value for footer attribute
Parameters	No	Name	Type
	1	footer	Spanned
			the footer attribute of HistoryItem object

**3. DBHelper class***a. Method:***getHistories**

Purpose	get all history items		
Return	Type		Description
	void		get all history items
Parameters	No	Name	Type
	1	getFavorite	boolean
			false: get all items true: get items that are favorite

**4. ListViewAdapter class***a. Attribute:*

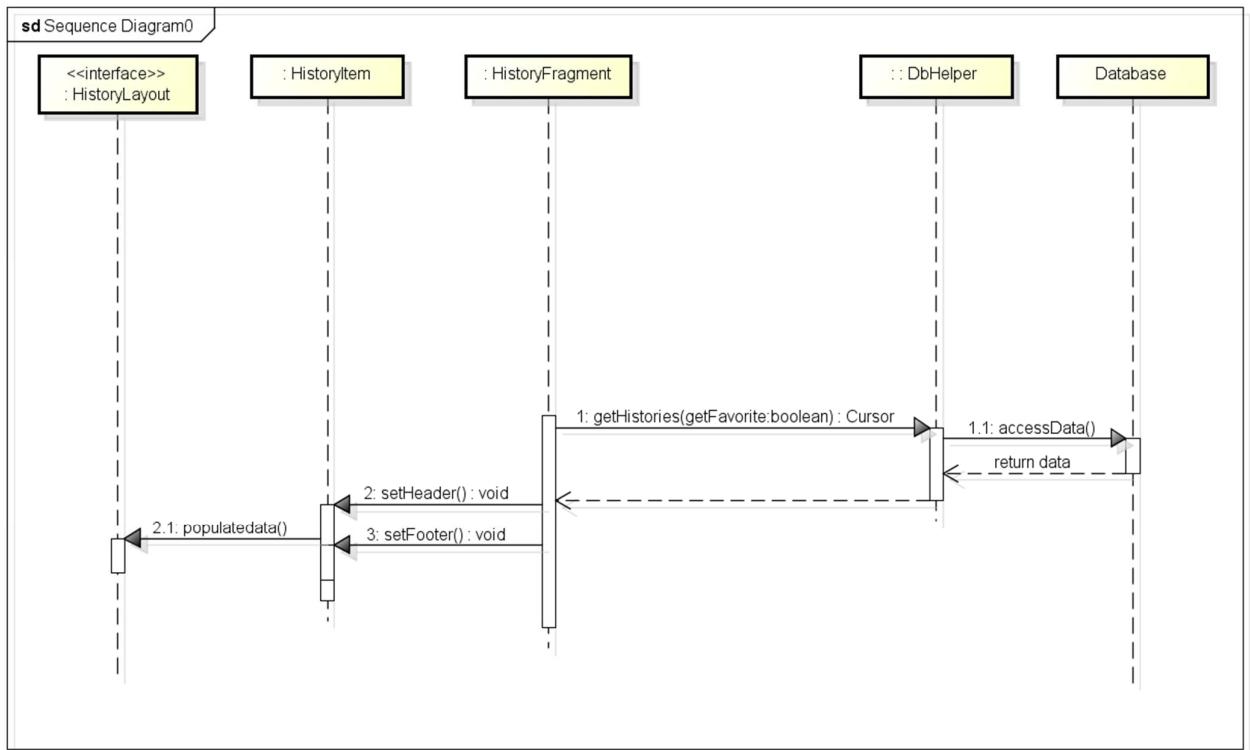
No	Name	Type	Description
1	historyItems	ArrayList	list of historyItem. Each element will be bind to one lisview item
2	historyId	ArrayList	list of history item id (id in the database)
3	dbHelper	Dbelper	access database

*b. Method:***fillValues**

Purpose	fill values into listview item	
Return	Type	Description

	void			
Parameters	No	Name	Type	Description
	1	id	Int	the id of the listview item
	2	view	View	the listview item

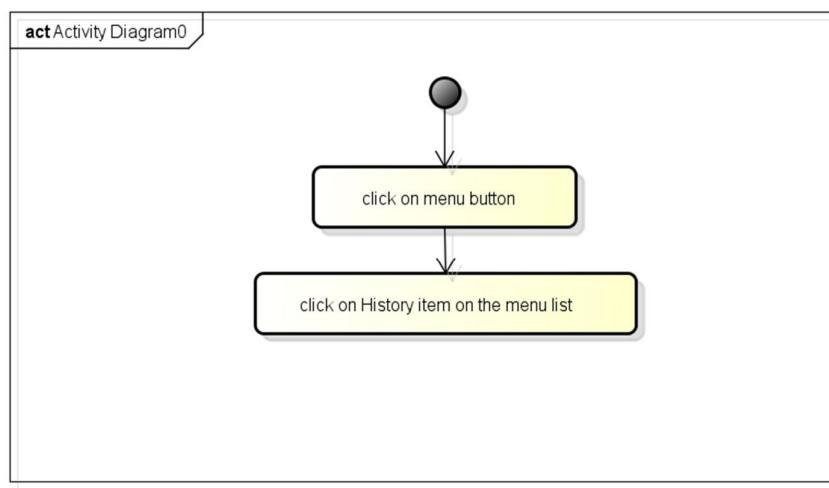
#### 1.7.2.9.3. Sequence diagram



powered by Astah

Figure 70: Sequence diagram of user case View history

#### 1.7.2.9.4. Activity diagram



powered by Astah

Figure 71: Activity diagram of UC view history

### 1.7.2.10. View favorite words only

#### 1.7.2.10.1. User case

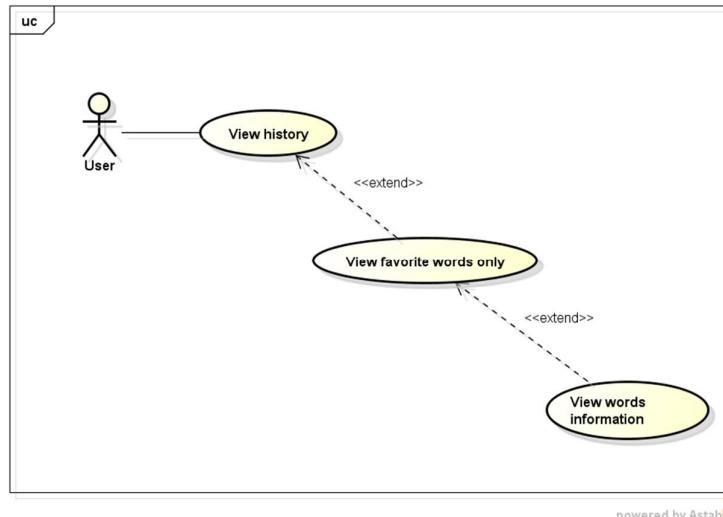


Figure 72: User case View farorite words

#### 1.7.2.10.1. Logical view

##### ❖ Class diagram

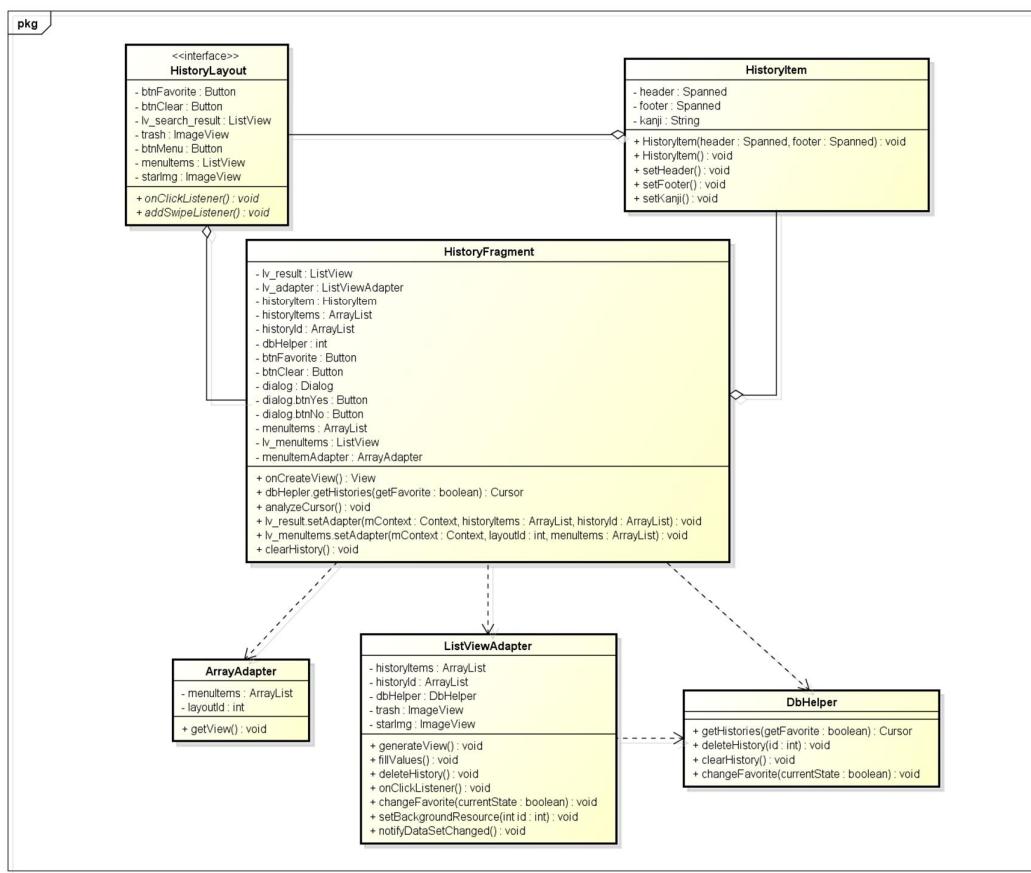


Figure 73: Class diagram for user case View favorite

❖ **Class specification:**

**1. HistoryFragment class**

*a. Attribute:*

No	Name	Type	Description
1	lv_result	ListView	display all history items
2	btnFavorite	Button	click to filter favorite words click again do disable filter
3	lv_adapter	ListView Adapter	bind data to every item of ListView (lv_result)
4	historyItem	History Item	store information (header, footer) to display in each ListView item
5	historyItems	ArrayList	list of historyItems
6	historyId	ArrayList	store the id of each history item
7	dbHelper	DbHelper	used for accessing database

*b. Method:*

btnFavorite.setOnClickListener

Purpose	filter by favorite words		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	View.OnClickListener	OnClickListener

the OnClickListener, executes events inside it when the view is clicked

lv\_result.setAdapter

Purpose	Create actor's account after actor logins into system.		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	mContext	Context
	2	historyItems	ArrayList

list of HistoryItem objects to display on layout

analyzeCursor

Purpose	analyze the Cursor information to add to listview		
Return	Type		Description
	void		
Parameters	No	Name	Type

--	--	--	--	--

## 2. HistoryItem class

### a. Attribute:

No	Name	Type	Description
1	header	Spanned	store data of header TextView in the layout
2	footer	Spanned	store data of footer TextView in the layout

### b. Method:

#### HistoryItem

Purpose	Create a HistoryItem object		
Return	Type		Description
	void		
Parameters	No	Name	Type
			Description

#### setHeader

Purpose	assign value for header attribute		
Return	Type		Description
	void		assign value for header attribute
Parameters	No	Name	Type
	1	header	Spanned

#### setFooter

Purpose	assign value for footer attribute		
Return	Type		Description
	void		assign value for footer attribute
Parameters	No	Name	Type
	1	footer	Spanned

## 3. DBHelper class

### a. Method:

#### getHistories

Purpose	get all history items	
Return	Type	Description

Parameters	No	Name	Type	Description
	1	getFavorite	boolean	false: get all items true: get items that are favorite

#### 4. ListViewAdapter class

##### a. Attribute:

No	Name	Type	Description
1	historyItems	ArrayList	list of historyItem. Each element will be bind to one lisview item
2	historyId	ArrayList	list of history item id (id in the database)
3	dbHelper	Dbhelper	access database

##### b. Method:

fillValues

Purpose	fill values into listview item		
Return	Type		Description
Parameters	void		
	No	Name	Type
	1	id	int
	2	view	View
			Description
			The id of the listview item
			The listview item

### 1.7.2.10.2. Sequence diagram

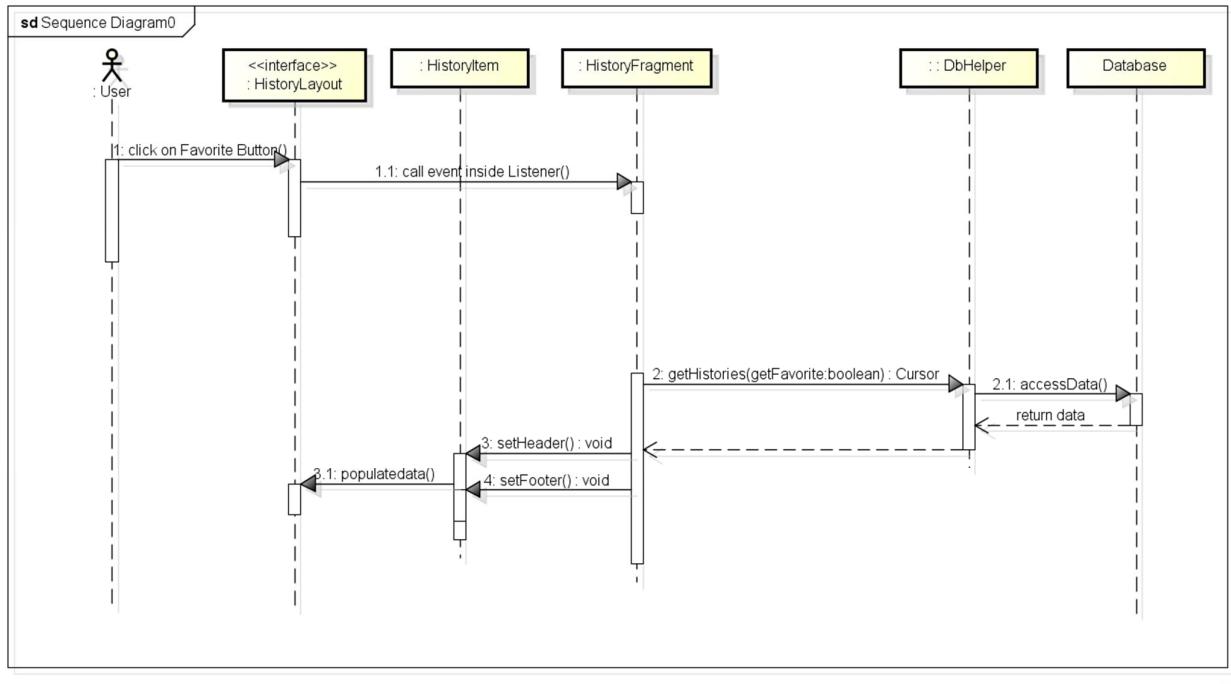


Figure 74: Sequence diagram of user case View favorite

### 1.7.2.10.4. Activity diagram

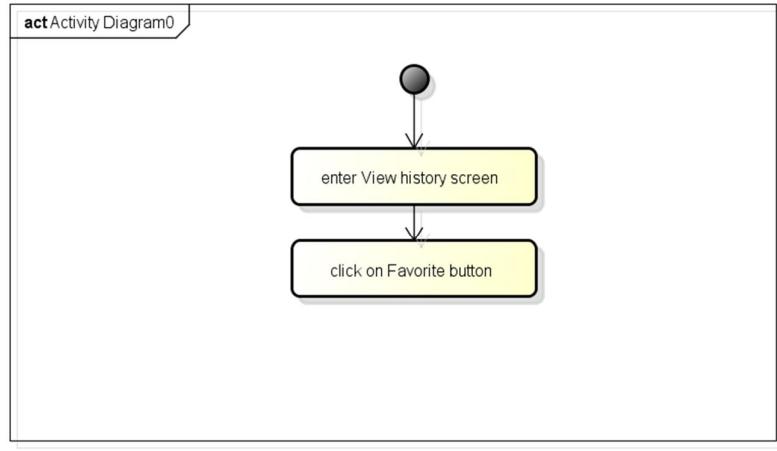
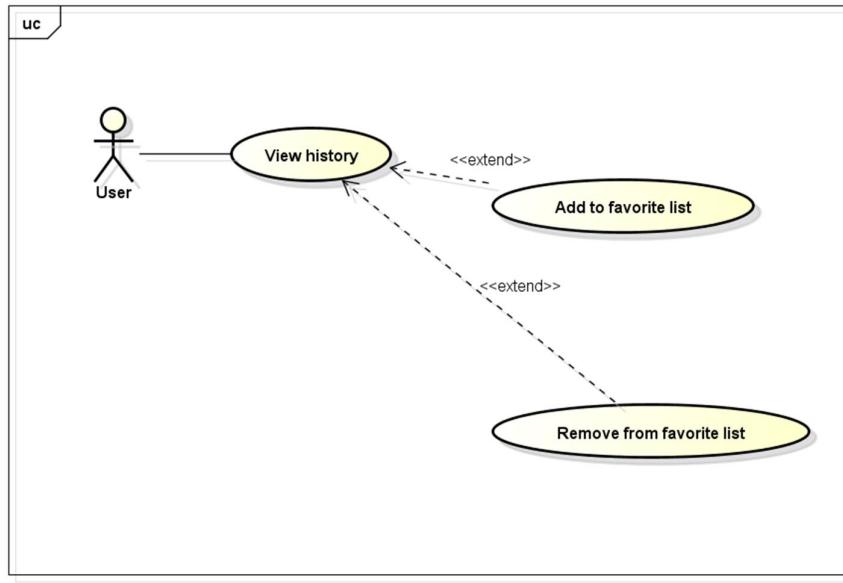


Figure 75: Activity diagram of user case View favorite words

### 1.7.2.11. Add to favorite list/ Remove from favorite list

#### 1.7.2.11.1. User case diagram

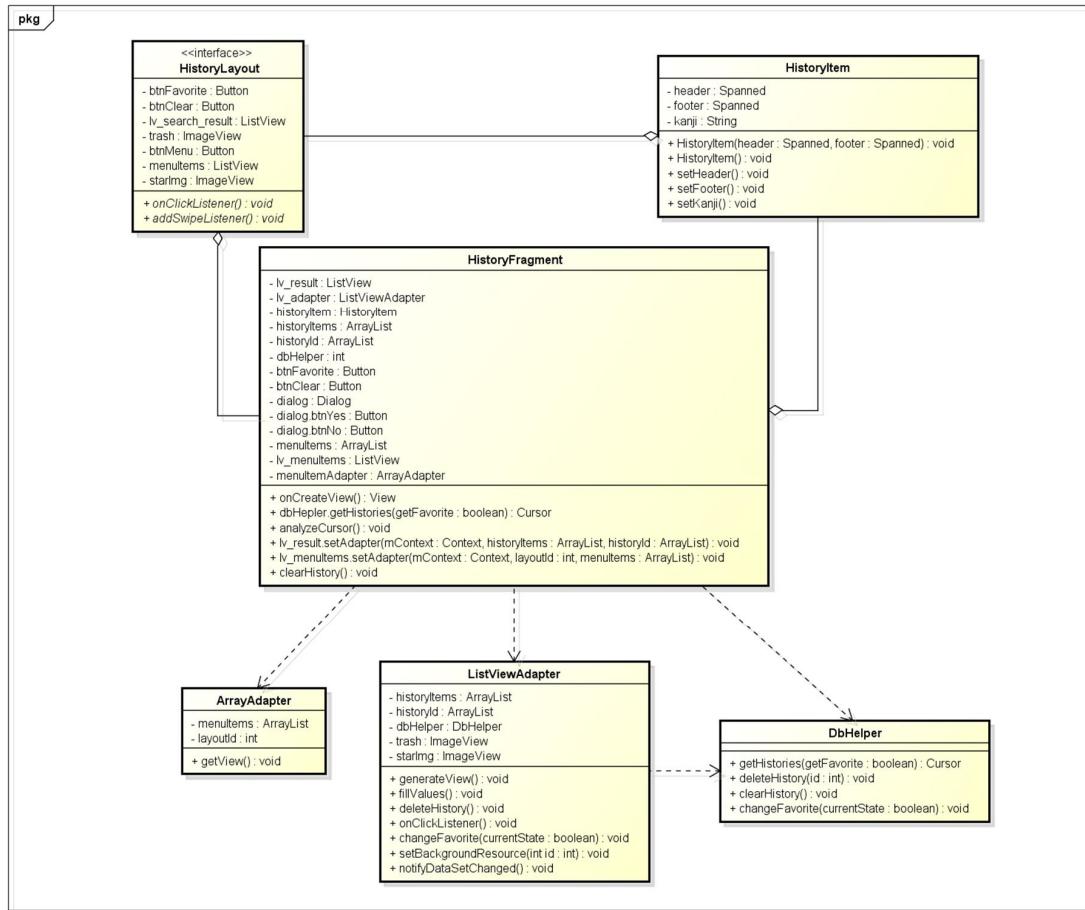


powered by Astah

Figure 76: User case Add / remove from favorite list

### 1.7.2.11.2. Logical view

#### ❖ Class diagram:



powered by Astah

Figure 77: Class diagram of user case Add / Remove from favorite

❖ **Class specification:**

**1. ListViewAdapter class**

a. Attribute:

No	Name	Type	Description
1	historyId	ArrayList	store the id of each history item.
2	starImg	ImageView	click to change favorite state

b. Method:

setBackgroundResource

Purpose	change the star image that suits the favorite state		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	id	int
		The resource id of the image	

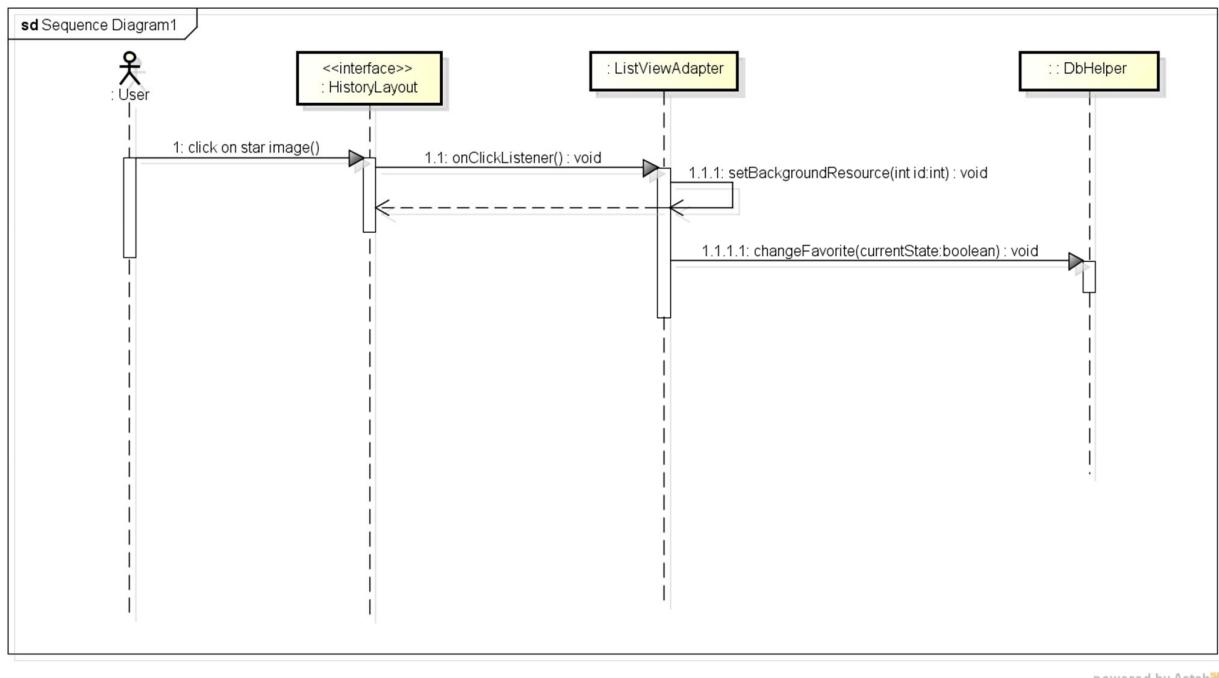
**2. DBHelper class**

a. Method:

changeFavorite

Purpose	change the favorite state		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	tableName	String
	2	id	int
		the table that contains the word/kanji	
		the word/kanji ID	
		the current favorite state of that word/kanji	

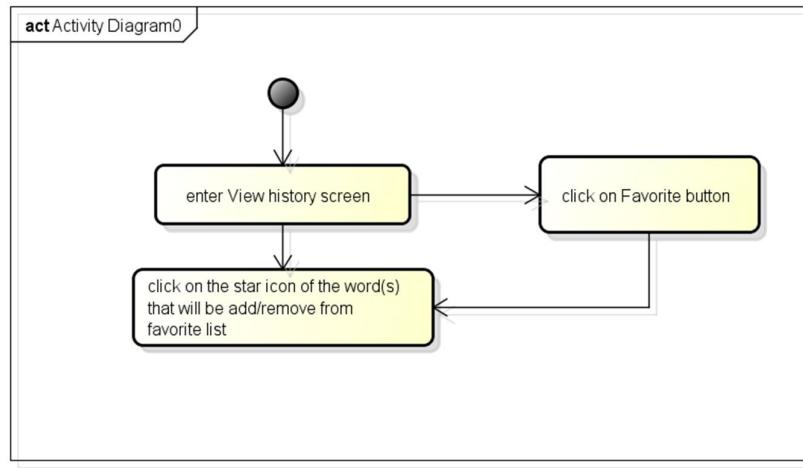
### 1.7.2.11.3. Sequence diagram:



powered by Astah

Figure 78: Sequence diagram of user case Add / Remove from favorite

### 1.7.2.11.4. Activity diagram



powered by Astah

Figure 79: Activity diagram of user case Add / remove from favorite list

### 1.7.2.12. Delete one word from history

#### 1.7.2.12.1. User case

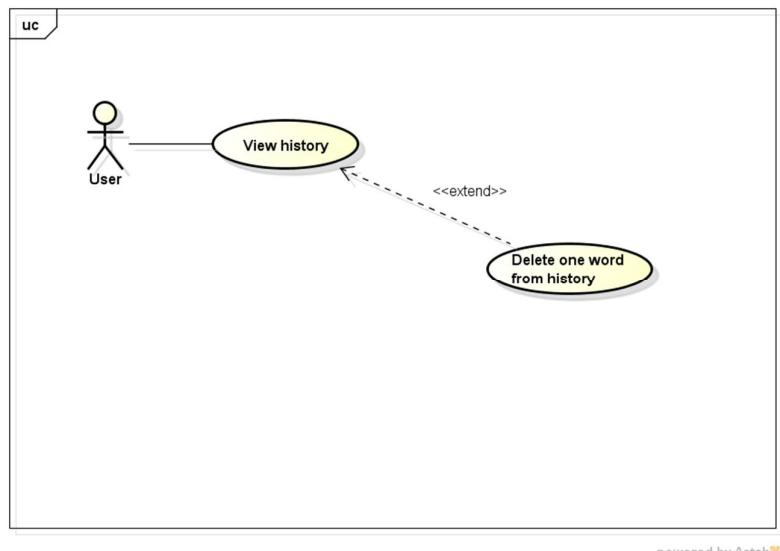


Figure 80: User case Delete word from histor

#### 1.7.2.12.2. Logical view

- ❖ Class diagram:

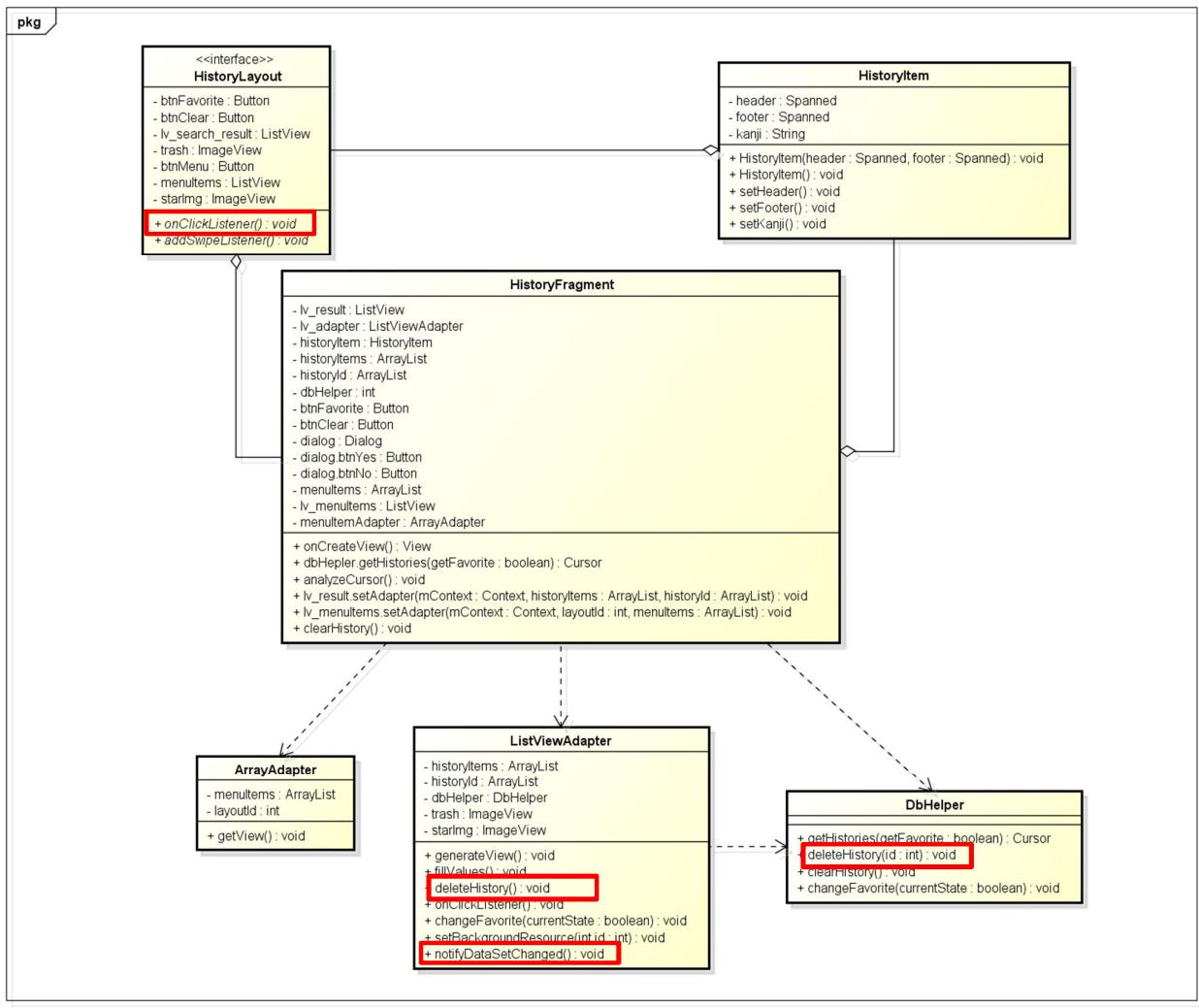


Figure 81: Class diagram of user case Delete a word from history

powered by Astah

## ❖ Class specification:

### 1. ListViewAdapter class

#### a. Method

historyItems.remove

Purpose	remove an item from historyItems ArrayList		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	position	int
			The position of the item in the array list

historyId.remove

Purpose	remove an item from historyId ArrayList		
---------	---	--	--

Return	Type	Description		
	<b>void</b>			
Parameters	No	Name	Type	Description
	1	position	int	The position of the item in the array list

## 2. DBHelper class

### a. Method

deleteHistory

Purpose	remove a history record from history table		
Return	Type	Description	
	<b>void</b>		
Parameters	No	Name	Type
	1	id	int
			The id of history record

### 1.7.2.12.2. Sequence diagram:

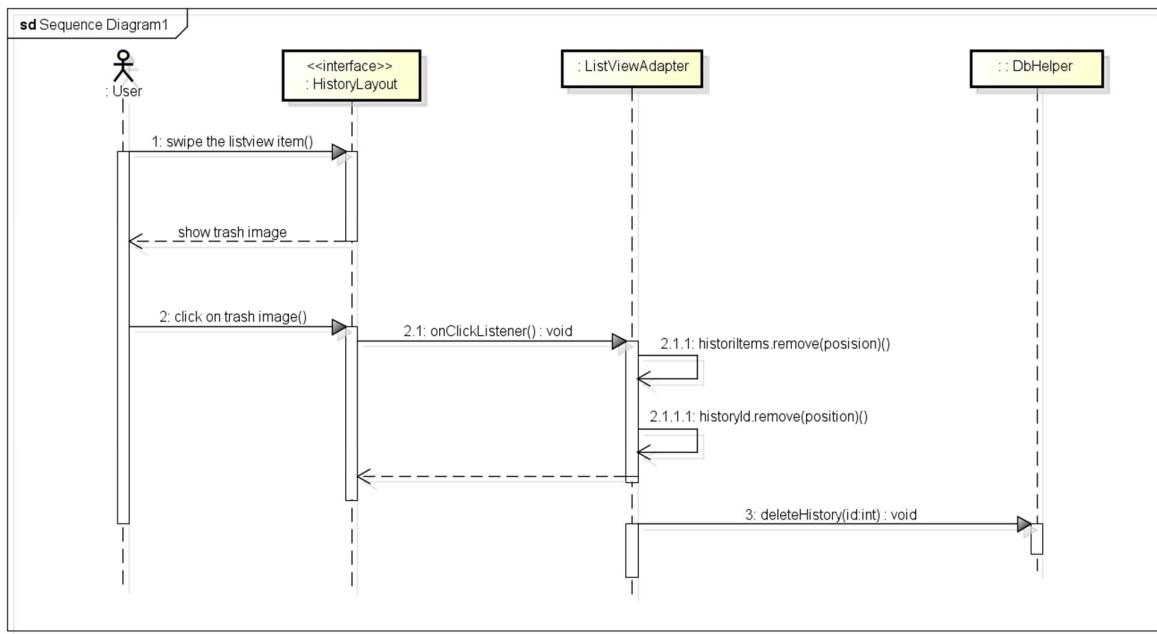


Figure 82: Sequence diagram of user case Delete a word from history

### 1.7.2.12.3. Activity diagram

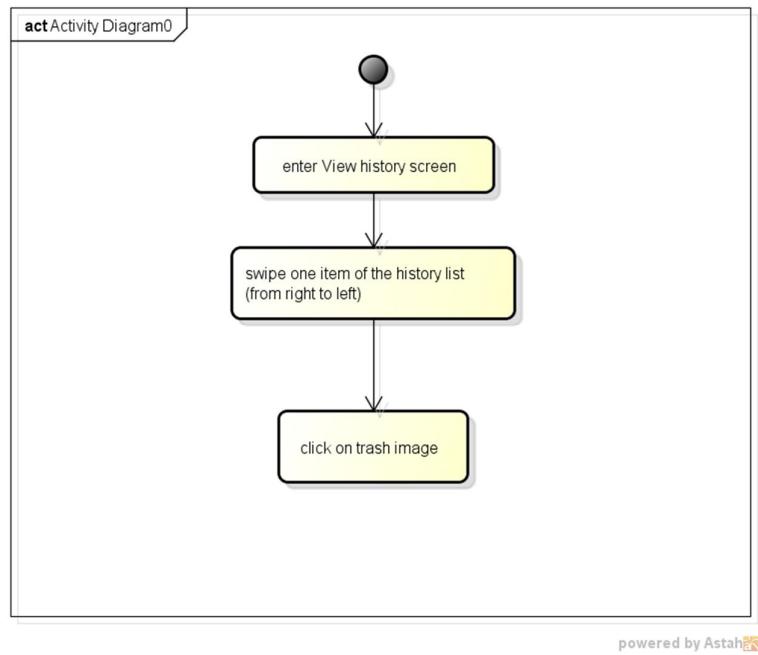
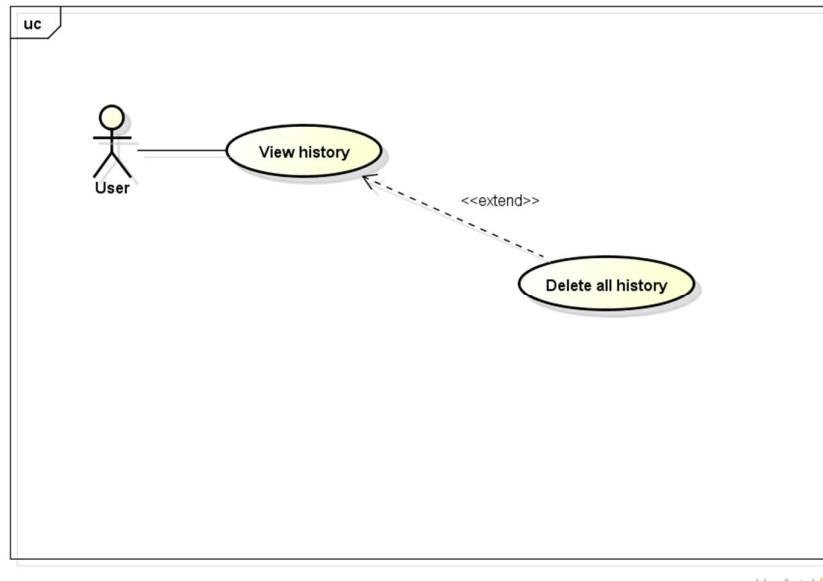


Figure 83: Activity diagram of user case delete word from history

#### 1.7.2.14. Clear History

##### 1.7.2.14.1. User case



##### 1.7.2.13.1. Logical view

❖ Class diagram:

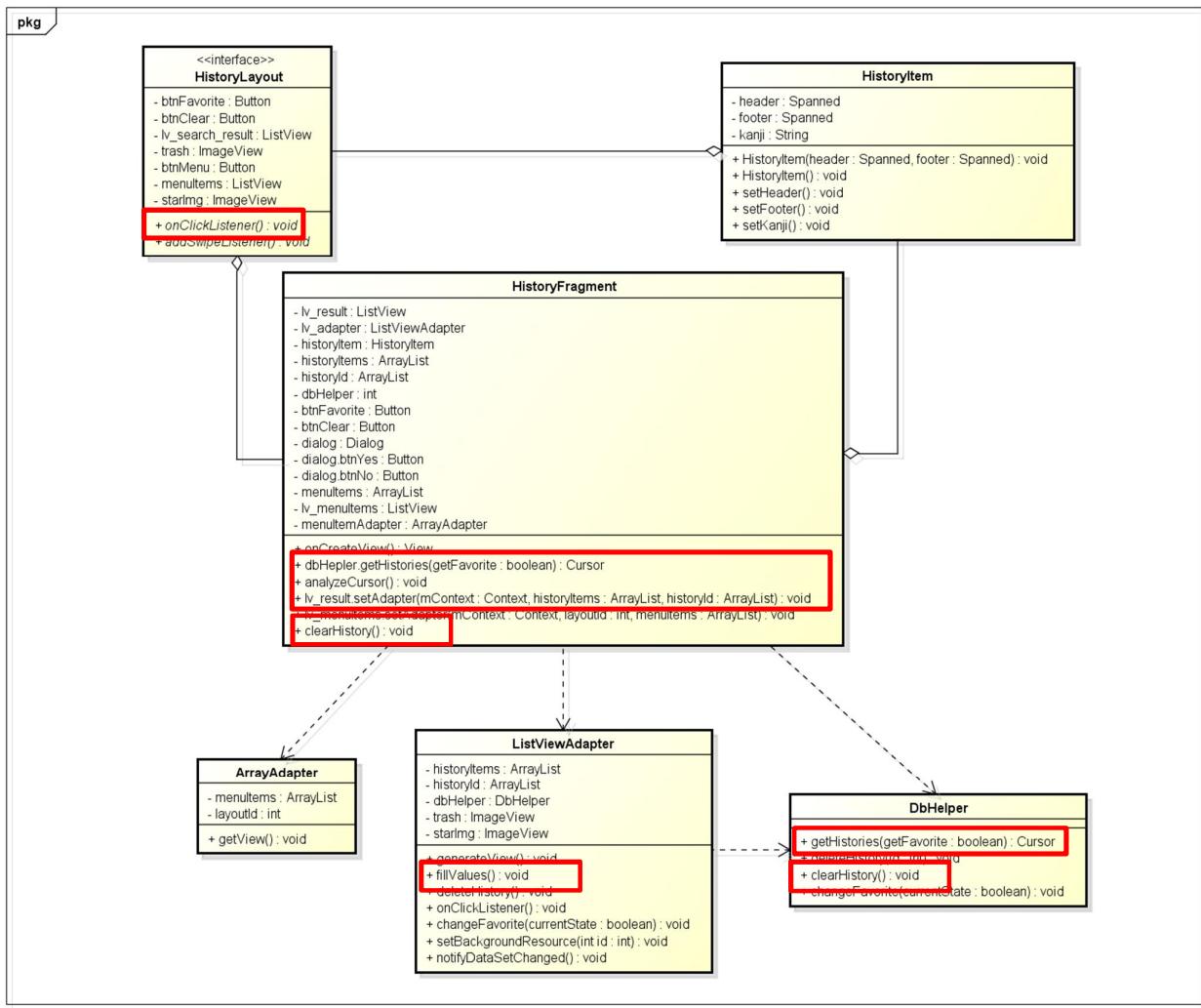


Figure 84: Class diagram of user case clear history

#### ❖ Class specification:

##### 1. HistoryFragment class

###### a. Attribute:

No	Name	Type	Description
1	lv_result	ListView	display all history items
2	lv_adapter	ListView Adapter	bind data to every item of ListView (lv_result)
3	historyItem	History Item	store information (header, footer) to display in each ListView item
4	historyItems	ArrayList	list of historyItems
5	historyId	ArrayList	store the id of each history item.
6	dbHelper	DbHelper	used for accessing database

7	dialog	Dialog	dialog to confirm deletion
---	--------	--------	----------------------------

b. Method:

clearHistory

Purpose	clear all history		
Return	Type		Description
	void		clear all history
Parameters	No	Name	Type
			Description

dialog.create.show

Purpose	confirm the deletion		
Return	Type		Description
	void		
Parameters	No	Name	Type
			Description

lv\_result.setAdapter

Purpose	Create actor's account after actor logins into system.		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	mContext	Context
	2	historyItems	ArrayList
	3	historyID	ArrayList

analyzeCursor

Purpose	analyze the Cursor information to add to listview		
Return	Type		Description
	void		
Parameters	No	Name	Type
			Description

## 2. HistoryItem class

a. Attribute:

No	Name	Type	Description
----	------	------	-------------

1	header	Spanned	store data of header TextView in the layout
2	footer	Spanned	store data of footer TextView in the layout

b. Method:

HistoryItem

Purpose	Create a HistoryItem object		
Return	Type		Description
	void		this is a constructor to create a HistoryItem object with null attribute
Parameters	No	Name	Type
			Description

setHeader

Purpose	assign value for header attribute		
Return	Type		Description
	void		assign value for header attribute
Parameters	No	Name	Type
	1	header	Spanned

setFooter

Purpose	assign value for footer attribute		
Return	Type		Description
	void		Assign value for footer attribute
Parameters	No	Name	Type
	1	footer	Spanned

## 1. DBHelper class

a. Method:

getHistories

Purpose	get all history items		
Return	Type		Description
	void		get all history items
Parameters	No	Name	Type
	1	getFavorite	boolean

## 2. ListViewAdapter class

*a. Attribute:*

No	Name	Type	Description
1	historyItems	ArrayList	list of historyItem. Each element will be bind to one lisview item
2	historyId	ArrayList	list of history item id (id in the database)
3	dbHelper	Dhelper	access database

*b. Method:*

fillValues

Purpose	fill values into listview item			
Return	Type	Description		
<b>void</b>				
Parameters	No	Name	Type	
	1	id	int	The id of the listview item
	2	view	view	The listview item

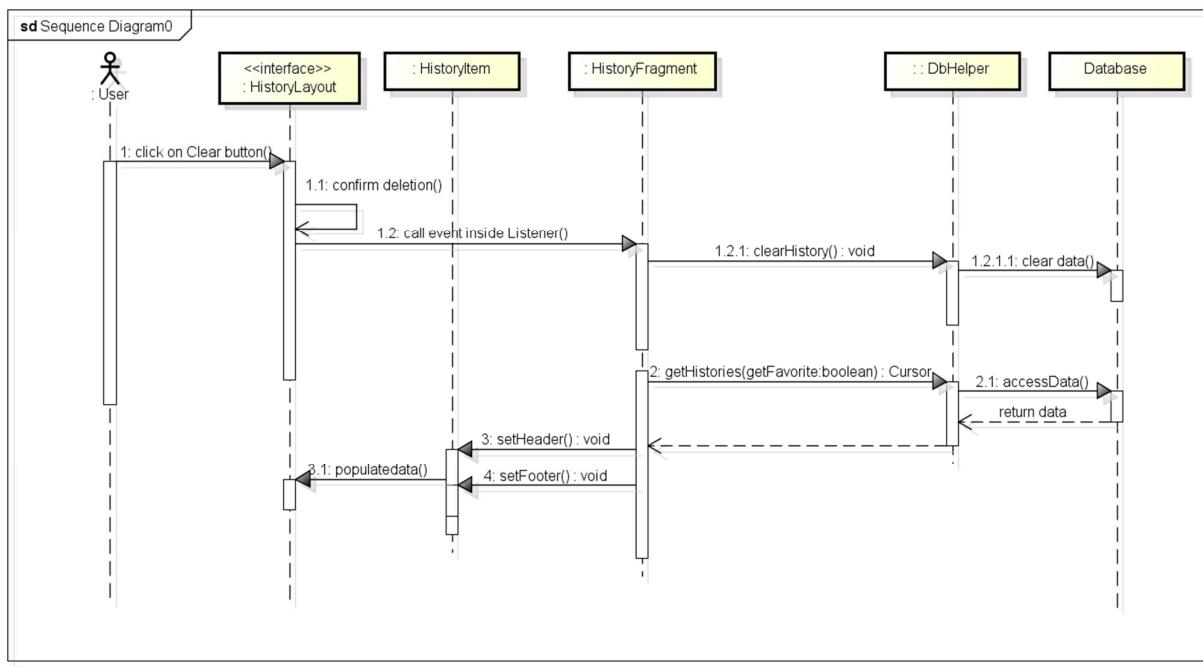
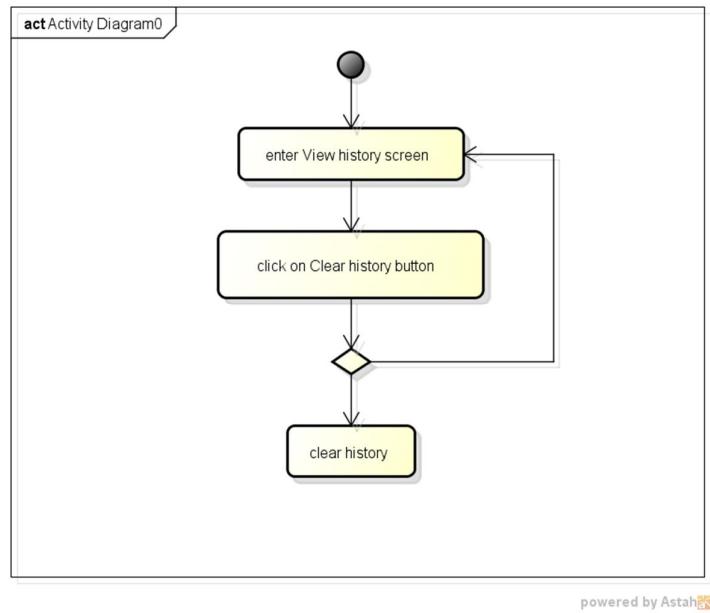
**1.7.2.13.3. Sequence diagram**

Figure 85: Sequence diagram of user case clear history

**1.7.2.13.4. Activity diagram**



powered by Astah

Figure 86: Activity diagram of user case clear history

## 1.8. Detail design of Server

### 1.8.1. Overview

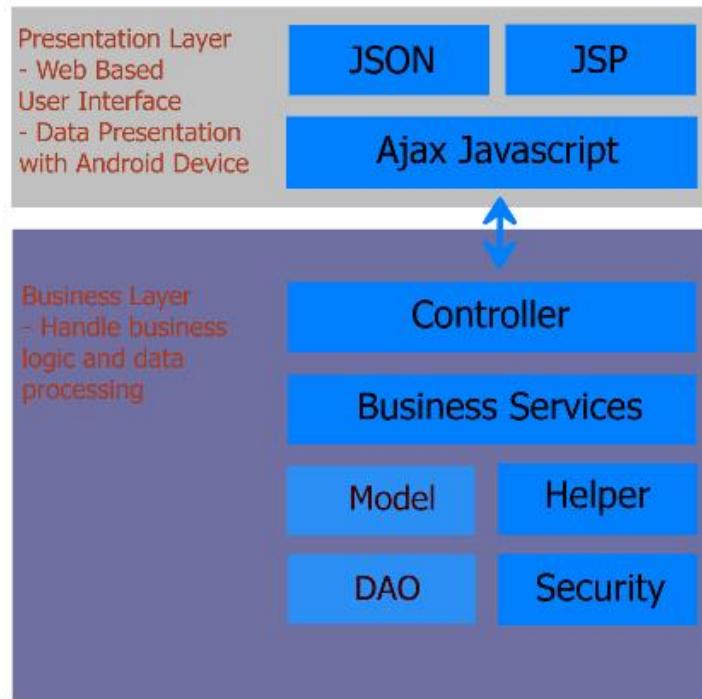


Figure 87: Overview of server architecture

### 8.1.2. Detail design

#### 8.1.2.1. Dictionary Management

##### 8.1.2.1.1. Use Case Specification

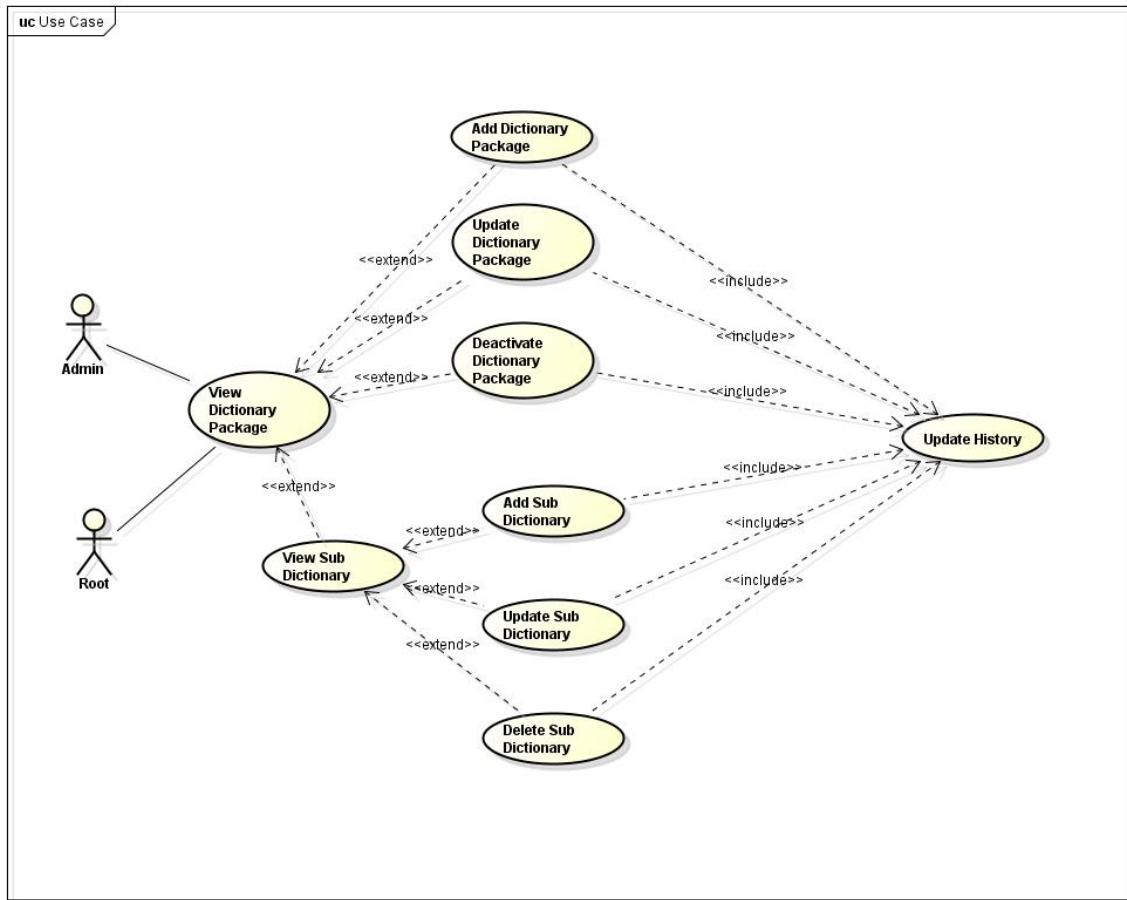


Figure 88: User cases of Dictionary management

##### 8.1.2.1.2. Logical view

- ❖ Class diagram

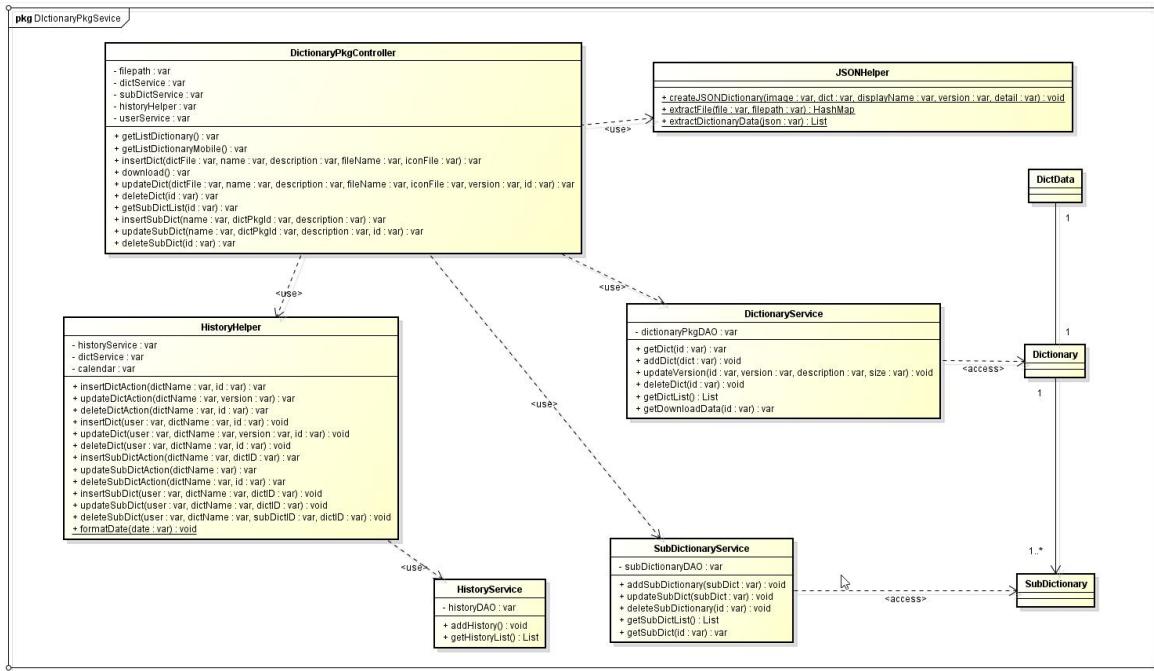


Figure 89: Dictionary Service Class Diagram

## ❖ Class Specification

### 1. DictionaryPkgController

#### a. Attribute:

No	Name	Type	Description
1	filepath	String	This variable contain the current filepath for storing and downloading dictionary json file
2	dictService	DictionaryPkgService	This variable is the service of dictionary package service
3	subDictService	SubDictionaryService	This variable is the service of sub dictionary service
4	historyHelper	HistoryHelper	The helper class of the history service
5	userService	UserService	The user service, help validate user session

#### b. Method:

getDictionaryList:

Purpose	Get the list of dictionary package and return the view of that list		
Return	Type	Description	
	String	The page view of the dictionary list	
Parameters	No	Name	Type

## getDictionaryListMobile:

Purpose	Get the list of dictionary package and return the json string of that list (for Android device's check for update purpose)			
Return	Type			Description
	String			The json string of the dictionary list
Parameters	No	Name	Type	Description

## getSubDictionaryListMobile:

Purpose	Get the list of sub dictionary package related to a specific sub dictionary and return the json string of that list (for Android device's data integration purpose)			
Return	Type			Description
	String			The json string of the sub dictionary list
Parameters	No	Name	Type	Description
	1	id	String	The id of the dictionary package

## insertDict:

Purpose	Add new dictionary to the database			
Return	Type			Description
	void			
Parameters	No	Name	Type	Description
	1	dictFile	MultipartFile	The dictionary file
	2	name	String	The dictionary name
	3	description	String	The description of the dictionary
	4	fileName	String	The filename of the dictionary which will be stored into the storage
	5	iconFile	MultipartFile	The icon file of the corresponding dictionary

## download:

Purpose	Download the specified dictionary			
Return	Type			Description
	void			
Parameters	No	Name	Type	Description

## updateDict:

Purpose	Update the specified dictionary by uploading new dictionary file, new icon file, and update the dictionary data		
Return	Type		Description
	String		The view page of the dictionary list
Parameters	No	Name	Type
	1	dictFile	MultipartFile
	2	name	String
	3	description	String
	4	fileName	String
	5	iconFile	MultipartFile
	6	version	String
	1	dictFile	MultipartFile

## deleteDict:

Purpose	Delete the specified dictionary by setting it into inactive state, allowing the preservation of the history list		
Return	Type		Description
	String		The view page of dictionary list
Parameters	No	Name	Type
	1	id	String

## getSubDictList:

Purpose	Get the list of the sub dictionary and return the view of the sub dictionary list		
Return	Type		Description
	String		The view page of the sub dictionary list
Parameters	No	Name	Type
	1	id	String

## insertSubDict:

Purpose	Add new sub dictionary to the database		
Return	Type		Description
	String		The view page of the sub

				dictionary list
Parameters	No	Name	Type	Description
	1	name	String	The sub dictionary name
	2	dictPkgId	String	The dictionary package ID
	3	description	String	The description of the sub dictionary

updateSubDict:

Purpose	Update the specific sub dictionary data			
Return	Type		Description	
	String		The view page of the sub dictionary list	
Parameters	No	Name	Type	Description
	1	name	String	The sub dictionary name
	2	dictPkgId	String	The dictionary package ID
	3	description	String	The description of the sub dictionary
	4	id	String	The ID of the sub dictionary

deleteSubDict:

Purpose	Delete the specific sub dictionary, by remove its data from the database			
Return	Type		Description	
	String		The view page of the sub dictionary list	
Parameters	No	Name	Type	Description
	1	id	String	The sub dictionary id

## 2. DictionaryPkgService

### a. Attribute:

No	Name	Type	Description
1	dictionaryPkgDAO	DictionaryPkgDAO	This variable contain the dictionary DAO to access to the database

### a. Methods

getDict:

Purpose	Get the specific dictionary package	
Return	Type	
	DictionartPackage	

Parameters	No	Name	Type	Description
	1	id	int	The dictionary package id

addDict:

Purpose	Add new Dictionary into the database		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	dict	DictionaryPackage
			The dictionary package data

updateVersion:

Purpose	Update the version and dictionary data (Dictionary file is already uploaded in the controller class)		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	id	int
	2	version	String
	3	size	double
			The size of the dictionary file

deleteDict:

Purpose	Delete the specific dictionary package, by setting its active state to false		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	id	int
			The dictionary packge id

getDictList:

Purpose	Get the list of dictioney package		
Return	Type		Description
	List<DictionaryPackage>		The list of the dictionary package
Parameters	No	Name	Type
			Description

getDownloadData:

Purpose	Get the download data of the dictionary package		
Return	Type		Description
	DictData		The dictionary data for downloading

Parameters	No	Name	Type	Description
	1	id	int	The dictionary package id

### 3. SubDictionaryService

a. Attribute:

No	Name	Type	Description
1	subDictionary DAO	SubDictionaryDAO	This variable contain the sub dictionary DAO to access to the database

b. Methods:

addSubDictionary:

Purpose	Add new sub dictionary to the database		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	subDict	SubDictionary

updateSubDict:

Purpose	Update the specific sub dictionary data		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	subDict	SubDictionary

deleteSubDictionary:

Purpose	Delete sub dictionary data, by removing its record from the database		
Return	Type		Description
Parameters	No	Name	Type
	1	id	int

getSubDictList:

Purpose	Get the list of the sub dictionary		
Return	Type		Description
	List<SubDictionary>		The list of the sub dictionary
Parameters	No	Name	Type

getSubDict:

Purpose	Get the sub dictionary data		
Return	Type		Description
	SubDictionary		The sub dictionary ata
Parameters	No	Name	Type
	1	id	int
			The id of the sub dictionary package

#### 4. JSONHelper

##### a. Method

createJSONDictionary:

Purpose	Create a json extension file with the dictionary icon file and the dictionary package file itself. Both the icon file and the dictionary file will be converted into bytes array and put into the json extension file.		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	image	File
Parameters	2	dict	File
	3	displayName	String
Parameters	4	version	String
	5	detail	String
			The description of the dictionary

extractFile:

Purpose	Extract the json extension file into dictionary file and icon file, while returning the hash map containing the dictionary data		
Return	Type		Description
	HashMap<String, String>		The hash map of the dictionary data
Parameters	No	Name	Type
	1	file	File
Parameters	2	filepath	String
			The filepath of the dictionary that will be saved into

extractDictionaryData:

Purpose	Extract a Dictionary json data string into the list of the dictionary data (for Android's check for update purpose). Using Gson library		
Return	Type		Description

	List<DictionaryPackage>			The list of dictionary package
Parameters	No	Name	Type	Description
	1	json	String	The json string of the dictionary list

### 8.2.1.3. Sequence Diagram

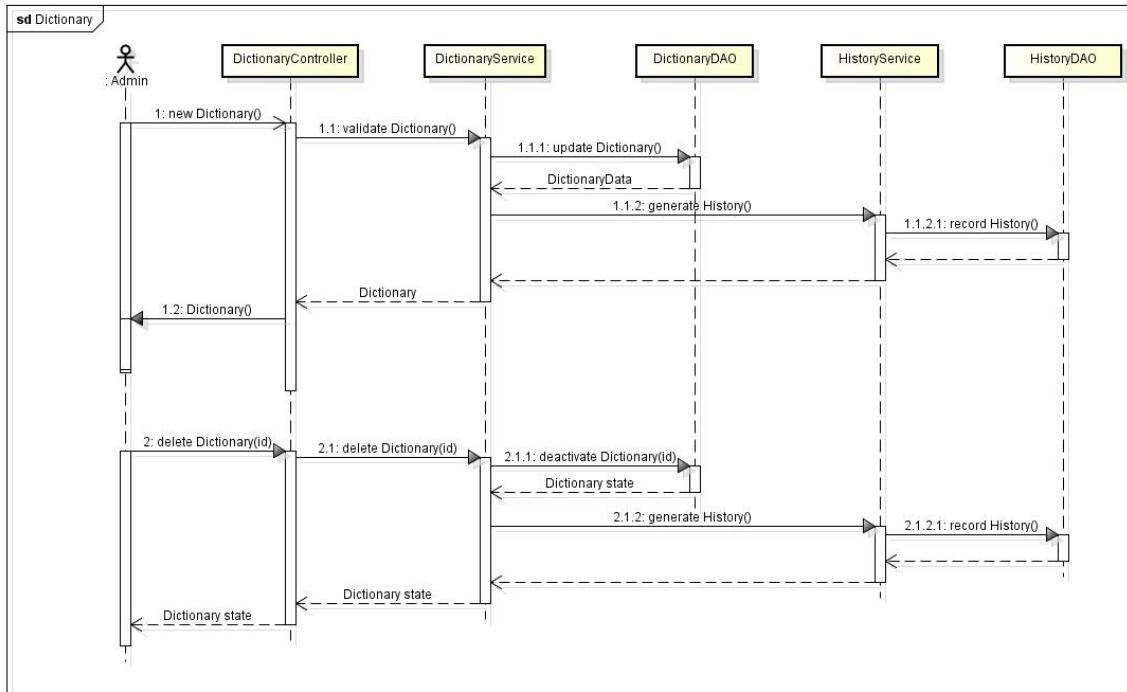


Figure 90: Dictionary Service Sequence Diagram

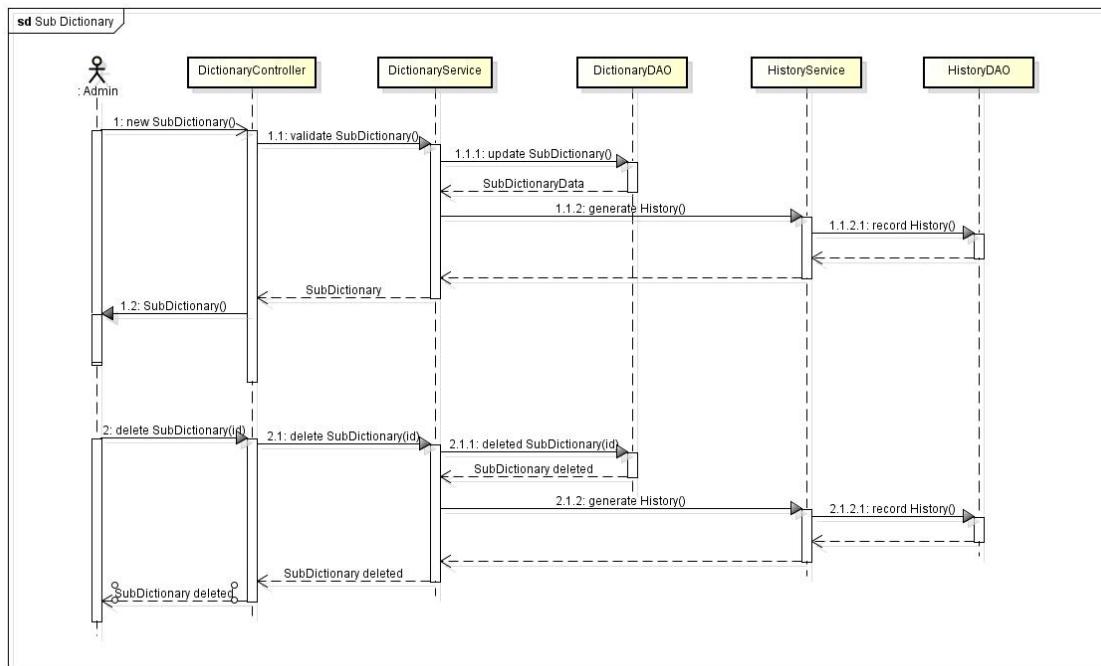
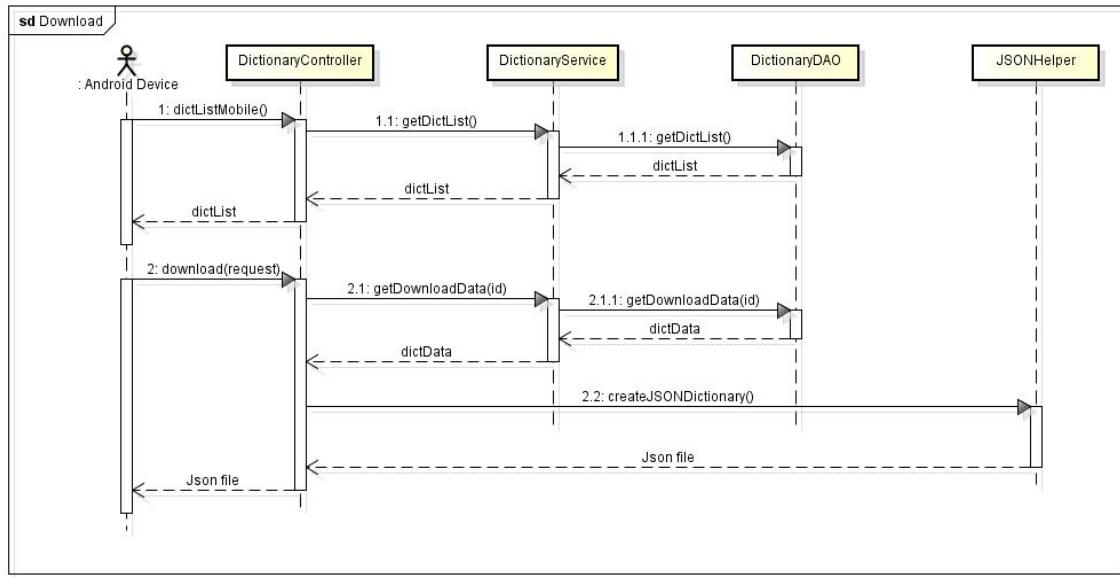
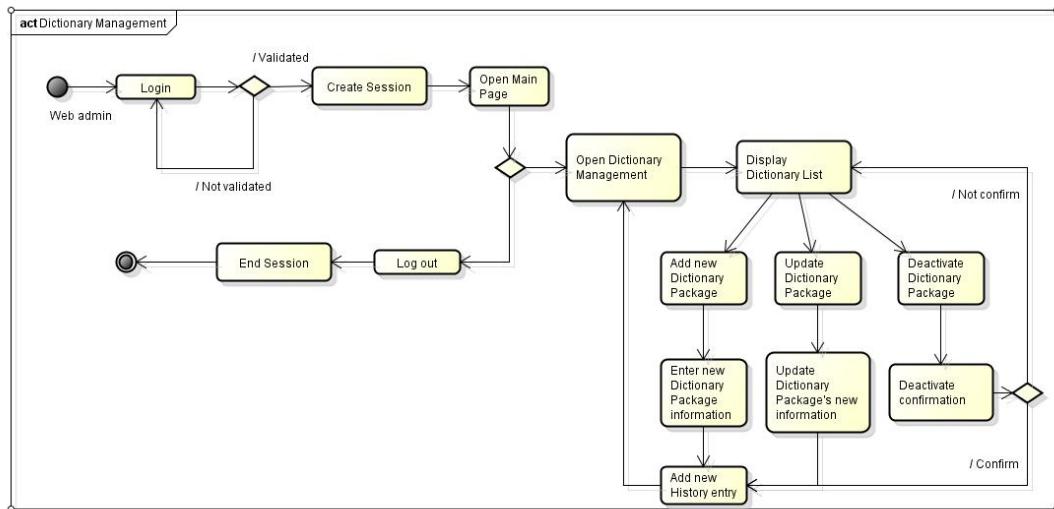


Figure 91: Sub Dictionary Service Sequence Diagram



*Figure 92: Download Dictionary Sequence Diagram*  
(Reference to 5.4 History Logging for detail design of HistoryService)

#### 1.8.2.1.4. Activity diagram



*Figure 93: Dictionary Service Activity Diagram*

After every action evolving dictionary management, the system will always have to log the history data.

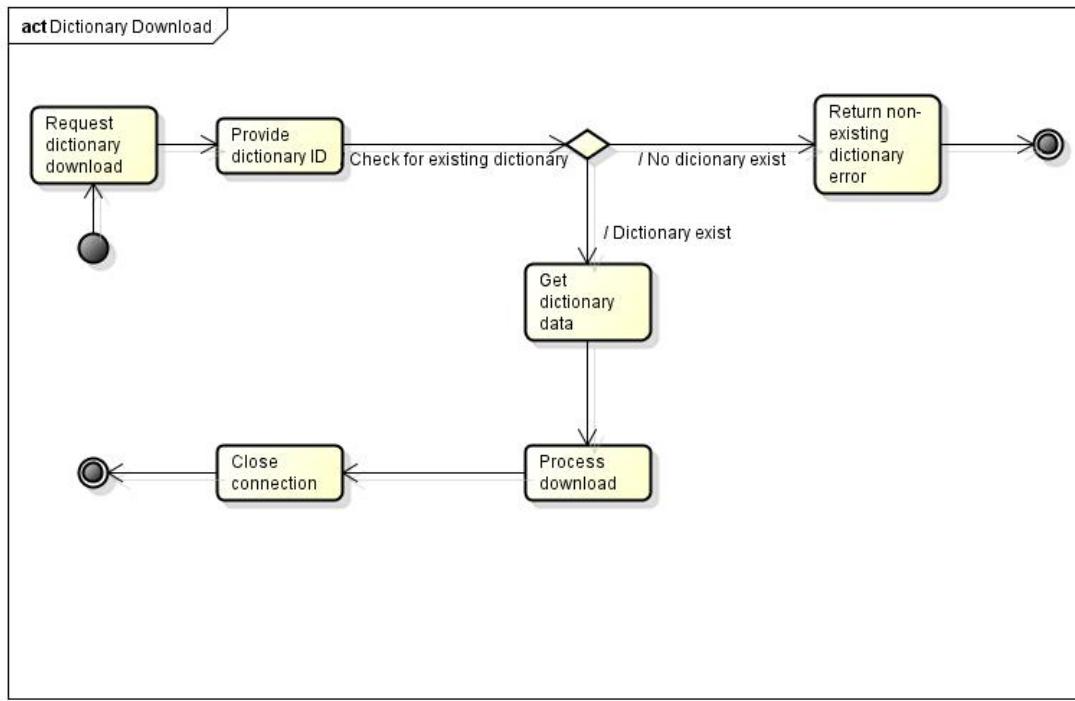


Figure 94: Download Dictionary Activity Diagram

### 1.8.2.2. Account management

#### 1.8.2.2.1. Use Case specification

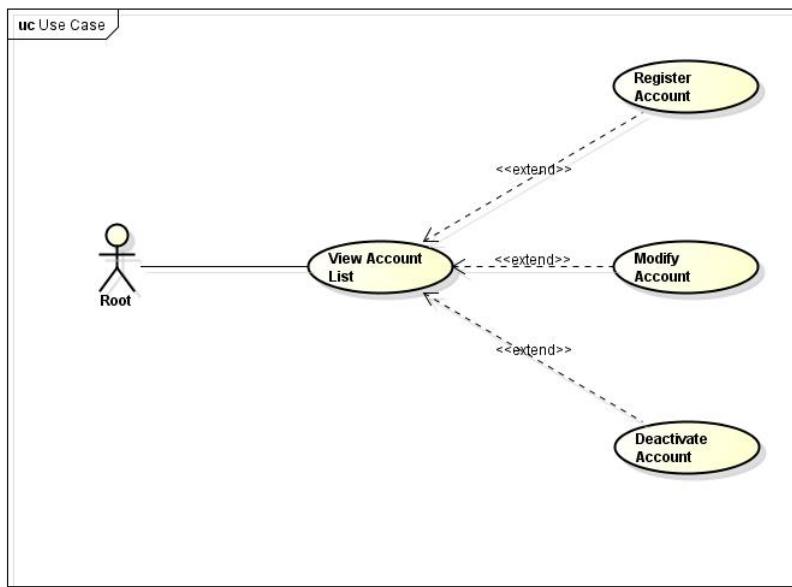


Figure 95: Account management user cases

### 1.8.2.2. Logical view

#### ❖ Class diagram:

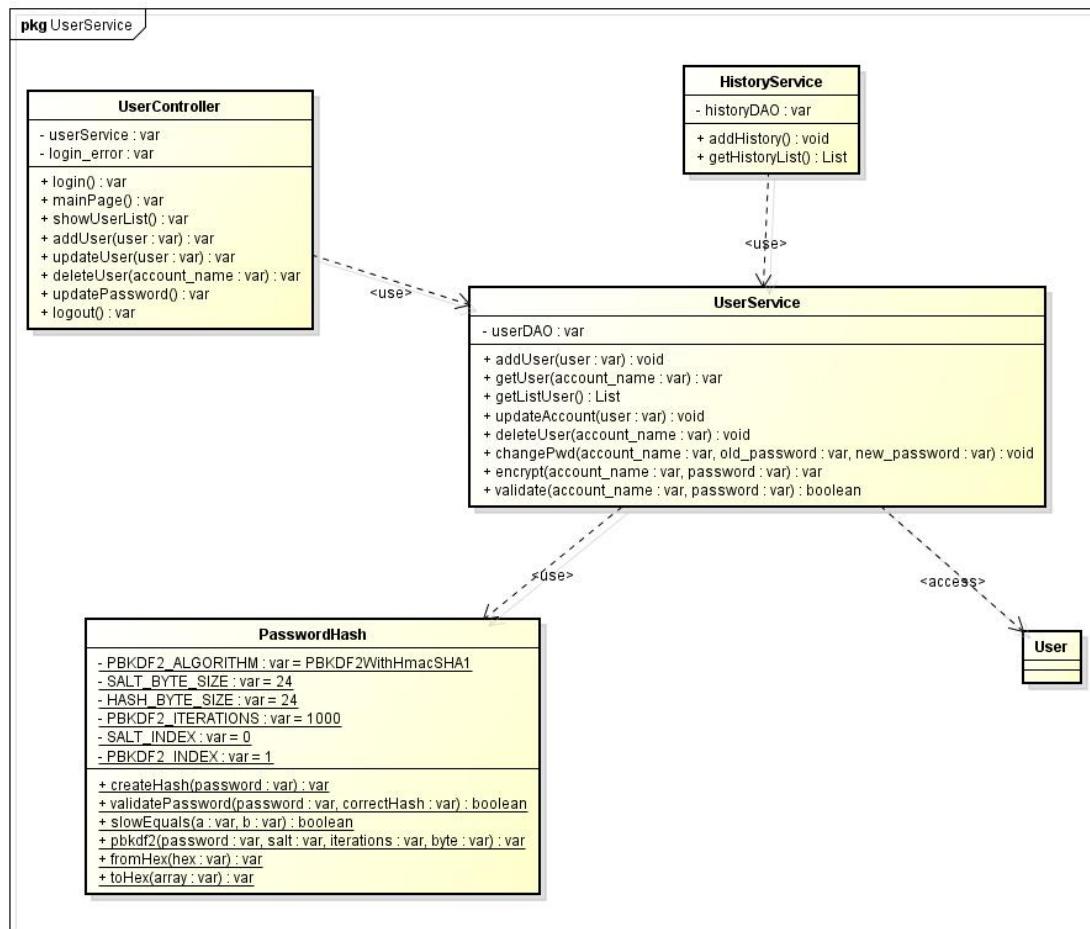


Figure 96: Account Management Class Diagram

#### ❖ Class Specification:

##### 1. UserController

###### a. Attribute:

No	Name	Type	Description
1	1	userService	UserService
2	2	login_error	String

###### b. Method:

login:

Purpose	Login user into the system, set the session of the user including his account_name and role	
Return	Type	Description

	String			The main page of the system if login successful, otherwise the index login
Parameters	No	Name	Type	Description

mainPage:

Purpose	Show the main page of the system to the user, while checking for user's authority			
Return	Type			Description
	String			The main view page
Parameters	No	Name	Type	Description

showUserList:

Purpose	Get the list of user data (without sensitive information) and display it on the userList page			
Return	Type			Description
	String			The view page of the user list
Parameters	No	Name	Type	Description

addUser:

Purpose	Add new user into the database, while hashing their password before storing it into the database			
Return	Type			Description
	String			The view page of the user list
Parameters	No	Name	Type	Description
	1	user	User	The user data

updateUser:

Purpose	Update the specific user's data (without changing their password – at the different method) into the database			
Return	Type			Description
	String			The view page of the user list
Parameters	No	Name	Type	Description
	1	user	User	The updated user data (without password)

deleteUser:

Purpose	Delete the specific user, by deactivated his account in the database.		
Return	Type		Description
	String		the view page of the user list
Parameters	No	Name	Type
	1	account_name	String

updatePassword:

Purpose	Update user password, by validating it's old password, then hashing it's new password and store the new hashed password into the database. Close the windows after changing the password		
Return	Type		Description
	void		
Parameters	No	Name	Type
			Description

logout:

Purpose	Log out the user from the system		
Return	Type		Description
	String		the index view page
Parameters	No	Name	Type
			Description

## 2. UserService

a. Attribute:

No	Name	Type	Description
1	userDAO	UserDAO	The user's DAO, which allow access to the database

b. Method

addUser:

Purpose	Root admin register new admin account for dictionary management		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	user	User

getUser:

Purpose	Get the information of the specific user		
Return	Type		Description

	User			The user data
Parameters	No	Name	Type	Description
	1	account_name	String	The account name of the user

getListUser:

Purpose	Get the list of all users in the database			
Return	Type		Description	
	List<User>		The list of user data	
Parameters	No	Name	Type	Description

updateAccount:

Purpose	Update new information for user data (without password change)			
Return	Type		Description	
	void			
Parameters	No	Name	Type	Description
	1	user	User	The user data

deleteUser:

Purpose	Delete user data, by deactivate it, to prevent that user from accessing into the system			
Return	Type		Description	
	void			
Parameters	No	Name	Type	Description
	1	account_name	String	The account name of the user

changePwd:

Purpose	Allow user to change their password, the old password will be validated, and the new password will be hashed, and update it into the database			
Return	Type		Description	
	void			
Parameters	No	Name	Type	Description
	1	account_name	String	The account name of the user
	2	old_password	String	The user's old password
	3	new_password	String	The user's new password

encrypt:

Purpose	Encrypt the user's password into the hash string
---------	--

Return		Type		Description
		String		The hash of the user password
Parameters	No	Name	Type	Description
		account_name	String	The account name of the user
	2	password	String	The password of the user

validate:

Purpose	Validate the current user, return true if the password hash is match the same with the hash in the database, otherwise return false		
Return	Type		Description
	boolean		the validation result of the hash check
Parameters	No	Name	Type
	1	account_name	String
	2	password	String

### 3. PasswordHash

#### a. Attribute

No	Name	Type	Description
1	PBKDF2_ALGORITHM	String	The PBKDF2 with Hmac SHA1 algorithm string declaration, which will be used for password hashing
2	SALT_BYTE_SIZE	int	The desired byte size of the salt
3	HASH_BYTE_SIZE	int	The desired hashed string size of the password
4	PBKDF2_ITERATIONS	int	The iteration number of the algorithm. The larger the number, the more costly to hash a password for the system and also for the attacker
5	SALT_INDEX	int	The index of the salt in the string got from database
6	PBKDF2_INDEX	int	The index of the hashed password in the string got from database

#### b. Method

createHash:

Purpose	Hash the password using the secret random salt	
Return	Type	Description
	String	The hashed password, along with the hashed salt

Parameters	No	Name	Type	Description
	1	password	String	The un-hashed password of the user

validatePassword:

Purpose	Validate the current password with the given pbkdf2 hashed password using salt			
Return	Type		Description	
	boolean		the validation result of the hash check	
Parameters	No	Name	Type	Description
	1	password	String	The password of the user
	2	correctHash	String	The correct hashed password, along with the salt

slowEquals:

Purpose	Compare the two bytes array in length-constant time. This comparison method is used so that password hashes cannot be extracted from an on-line system using a timing attack then attacked off-line			
Return	Type		Description	
	boolean		the validation result of the comparison	
Parameters	No	Name	Type	Description
	1	a	byte[]	The 1 <sup>st</sup> bytes array
	2	b	byte[]	The 2 <sup>nd</sup> bytes array

pbkdf2:

Purpose	Compute the PBKDF2 hash of the password			
Return	Type		Description	
	byte[]		The PBKDF2 hash of the password	
Parameters	No	Name	Type	Description
	1	password	char[]	The password of the user
	2	salt	byte[]	The salt for that password
	3	iteration	int	The iteration count (slowness factor)
	4	bytes	int	The length of the hash to compute in bytes

fromHex:

Purpose	Converts a string of hexadecimal characters into a byte array.	
Return	Type	Description

	byte[]			The hex string decoded into a byte array
Parameters	No	Name	Type	Description
	1	hex	String	The string hex

toHex:

Purpose	Converts a byte array into a hexadecimal string.			
Return	Type		Description	
	byte[]		A length*2 character string encoding the byte array	
Parameters	No	Name	Type	Description
	1	array	byte[]	The byte array to convert

### 1.8.2.2.3. Sequence diagram

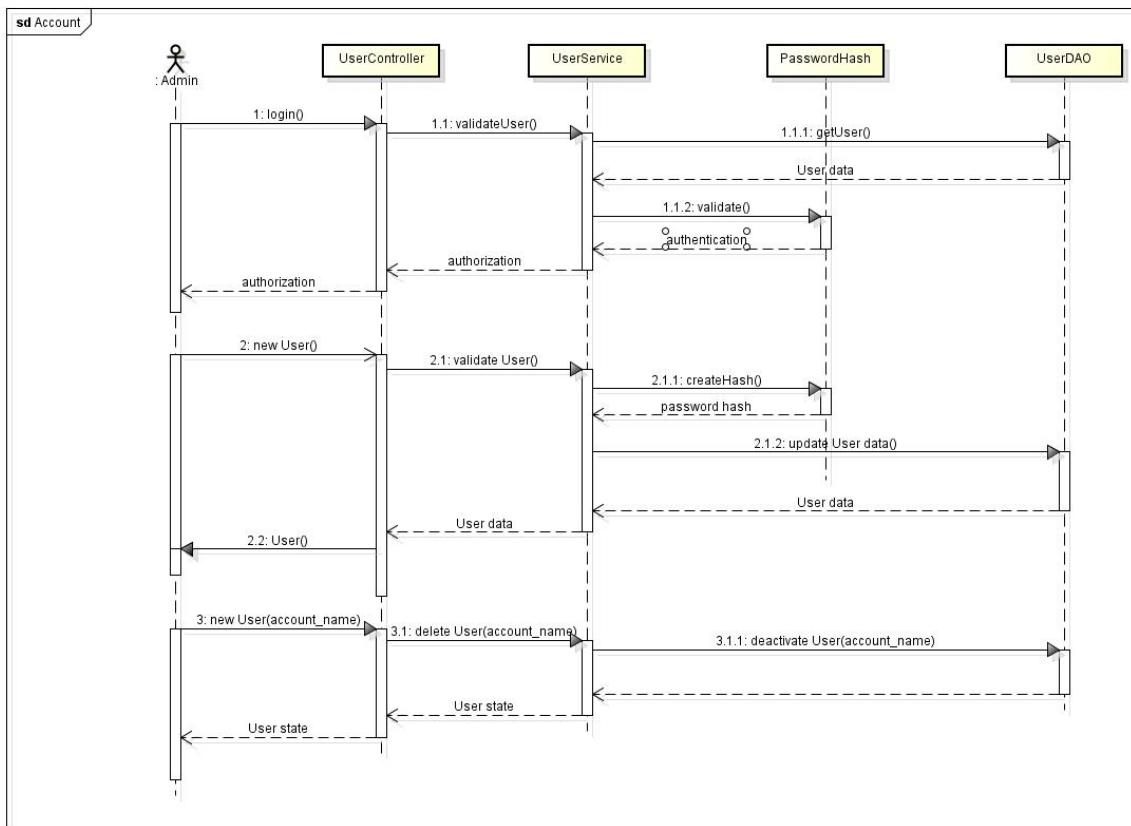


Figure 97: Account Management Sequence Diagram

#### 1.8.2.2.4. Activity diagram

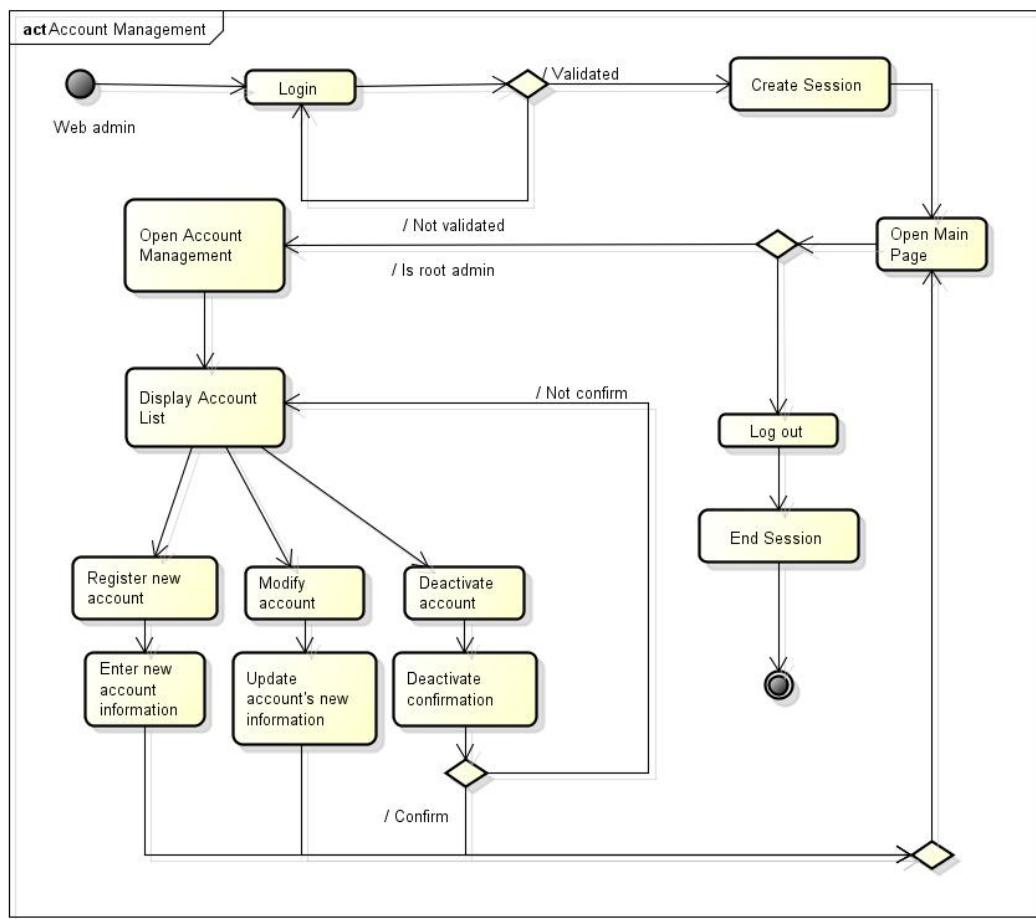


Figure 98: Account Management Activity Diagram

#### 1.8.2.3. History Logging

##### 1.8.2.3.1. Use Case diagram



##### 1.8.2.3.2. Logical view

###### ❖ Class diagram

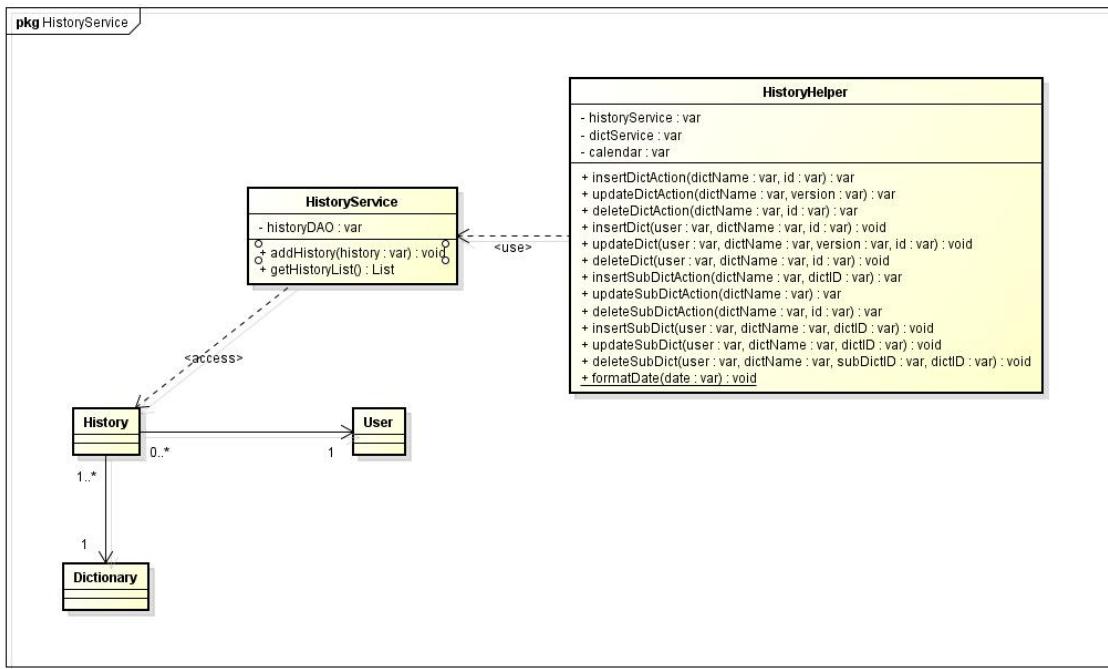


Figure 99: History Service Class Diagram

## ❖ Class specification

### 1. HistoryService

#### a. Attribute

No	Name	Type	Description
1	historyDAO	HistoryDAO	The history DAO to access history data in database

#### b. Method

`addHistory:`

Purpose	Add new history entry into the database		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	history	History
			Description
			The history entry

`getHistoryList:`

Purpose	Get the history entry list		
Return	Type		Description
	List<History>		The history entry list
Parameters	No	Name	Type
			Description

--	--	--	--	--

## 2. HistoryHelper

### a. Attribute

No	Name	Type	Description
1	historyService	HistoryService	The history service
2	dictService	DictionaryPkgService	The dictionary package service
3	calendar	Calendar	The calendar which is used for tracking current time in server

### b. Method

insertDictAction:

Purpose	Generate an “add dictionary” string with given dictionary name and dictionary id		
Return	Type		Description
	String		The generated string
Parameters	No	Name	Type
	1	dictName	String
	2	id	int

updateDictAction:

Purpose	Generate an “update dictionary” string with given dictionary name and dictionary version		
Return	Type		Description
	String		The generated string
Parameters	No	Name	Type
	1	dictName	String
	2	version	String

deleteDictAction:

Purpose	Generate an “delete dictionary” string with given dictionary name and dictionary id		
Return	Type		Description
	String		The generated string
Parameters	No	Name	Type
	1	dictName	String
	2	id	int

insertDict:

Purpose	Perform adding new history entry with “add dictionary” message		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	account name	String
	2	dictName	String
	3	id	int

updateDict:

Purpose	Perform adding new history entry with “update dictionary” message		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	account name	String
	2	dictName	String
	3	version	String
	4	id	int

deleteDict:

Purpose	Perform adding new history entry with “delete dictionary” message		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	account name	String
	2	dictName	String
	3	id	int

insertSubDictAction:

Purpose	Generate an “add sub dictionary” string with given sub dictionary name and sub dictionary id		
Return	Type		Description
	String		The generated string
Parameters	No	Name	Type
	1	dictName	String
	2	dictID	int

updateSubDictAction:

Purpose	Generate an “update sub dictionary” string with given sub dictionary name		
Return	Type		Description
	String		The generated string
Parameters	No	Name	Type
	1	dictName	String

deleteSubDictAction:

Purpose	Generate an “delete sub dictionary” string with given sub dictionary name and sub dictionary id		
Return	Type		Description
	String		The generated string
Parameters	No	Name	Type
	1	dictName	String

insertSubDict:

Purpose	Perform adding new history entry with “add sub dictionary” message		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	account name	String
	2	dictName	String
	3	dictID	int

updateSubDict:

Purpose	Perform adding new history entry with “update sub dictionary” message		
Return	Type		Description
	void		
Parameters	No	Name	Type
	1	account name	String
	2	dictName	String
	3	dictID	int

deleteSubDict:

Purpose	Perform adding new history entry with “delete sub dictionary” message		
Return	Type		Description
	void		

Parameters	No	Name	Type	Description
	1	account name	String	The user that deleted sub dictionary
	2	dictName	String	The sub dictionary name
	3	subDictID	int	The sub dictionary id
	4	dictID	int	The dictionary package id containing the deleted sub dictionary

formatDate:

Purpose	Format a long date into string		
Return	Type		Description
	String		The generated string
Parameters	No	Name	Type
	1	account name	String
	2	dictName	String
	3	subDictID	int
	4	dictID	int

### 1.8.2.3.3. Sequence diagram

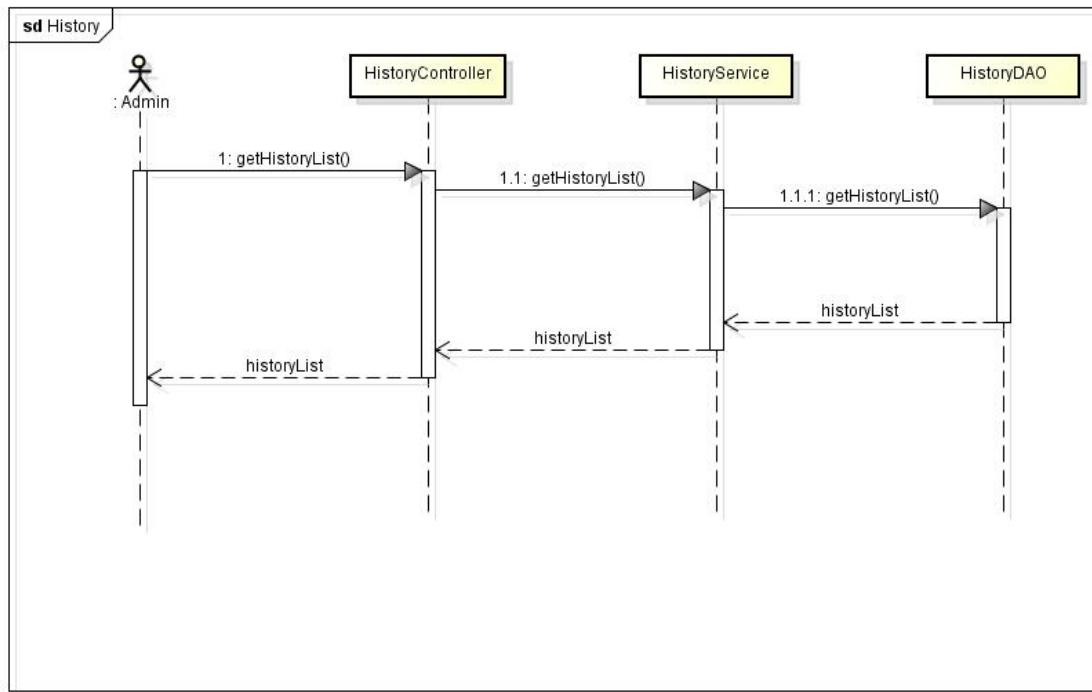
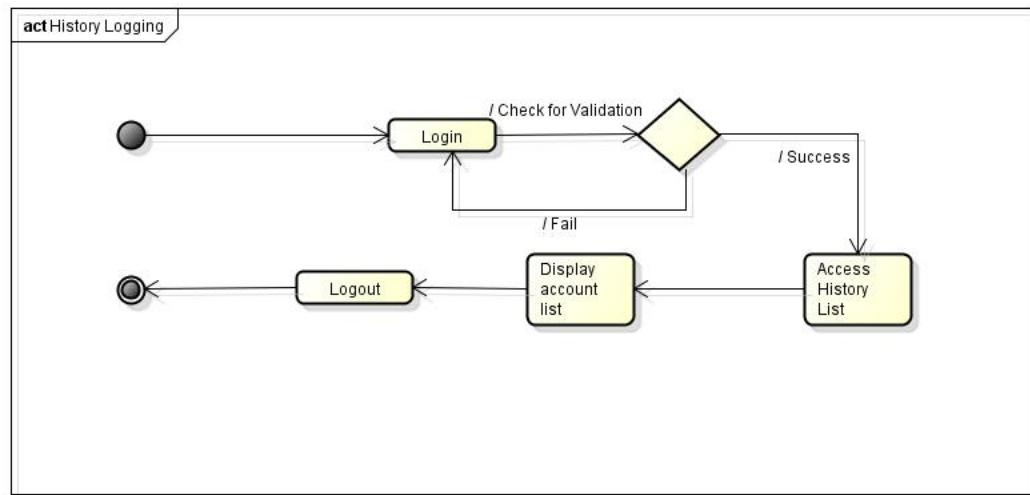


Figure 100: History Logging Sequence Diagram

### 1.8.2.3.4. Activity diagram



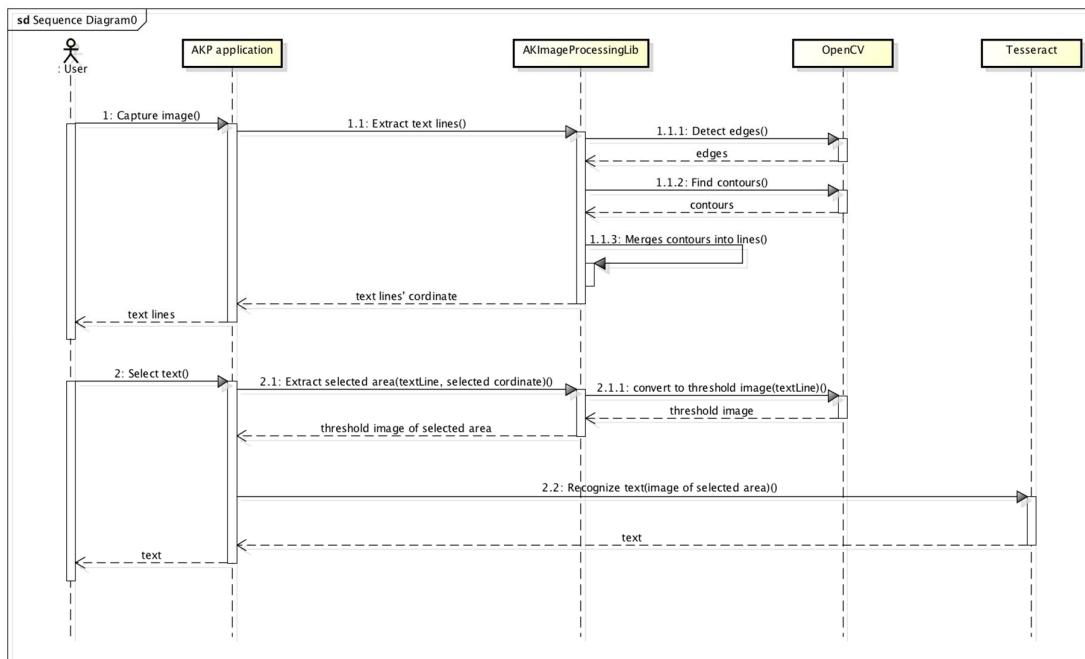
*Figure 101: History Logging Activity Diagram  
(Reference section 5.2.7 for history logging entries)*

## 1.9. AKP image processing library

### 1.9.1. Purpose

AKP library is developed to pre-process image before passing to Tesseract to improve text recognition accuracy. We use OpenCV library to support in processing image.

The diagram below shows the sequence of text recognition in Anki Pan application



*Figure 102: Text recognition in Anki Pan application*

### 1.9.2. Tesseract' shortcoming

In fact, Tesseract does various image processing operations, using Leptonica library, before doing the actual OCR. It generally does a good job, but still there are cases where it is not good enough, which leads to significant reduction of accuracy. According to Tesseract wiki on code.google.com<sup>[7]</sup>, Tesseract has some shortcoming, as described below, that cause problem in recognizing characters.

#### 1.9.2.1. Text size

Tesseract can not perform well with unsuitable text size. Characters with x-height less than 8 pixel is considered as noise and will be removed. Characters with x-height of 10 pixel have very little chance of being recognized correctly. Generally, Tesseract performs best with characters that is from 20 pixel in x-height.

#### 1.9.2.2. Binarization

Binarization, in this case, means converting an image to black and white. Tesseract can make mistake in this steps, particularly if the image background is of uneven darkness. As showed in figure below, the bottom left of the image is converted into black, which loses a part of the document.

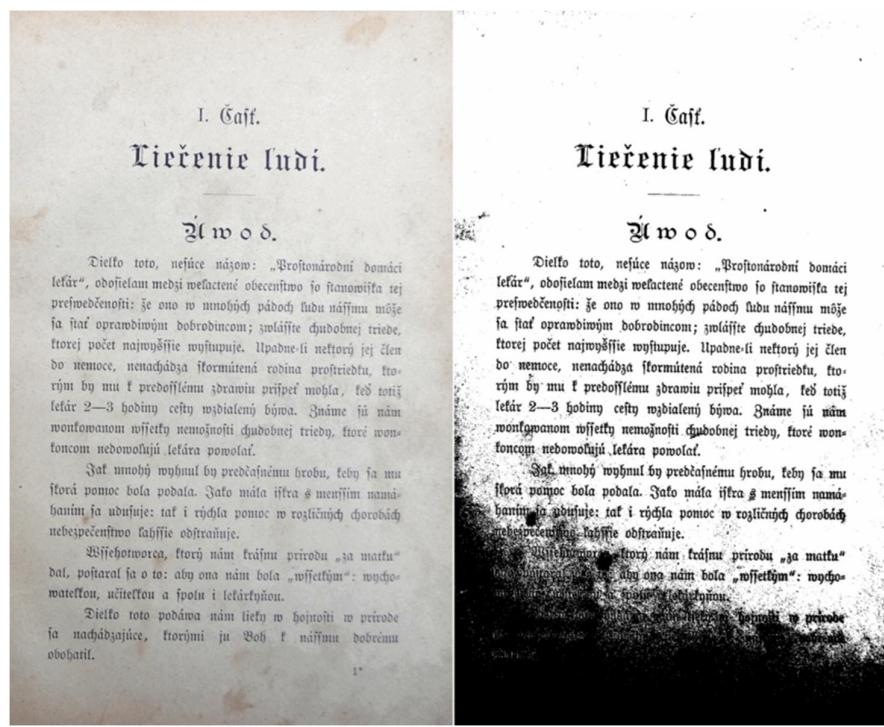


Figure 103: Sample of image binarized by Tesseract

#### 1.9.2.3. Noise

Noise is unexpected random variation of brightness or color. Certain type of noise can not be remove by Tesseract, which can cause accuracy to drop.

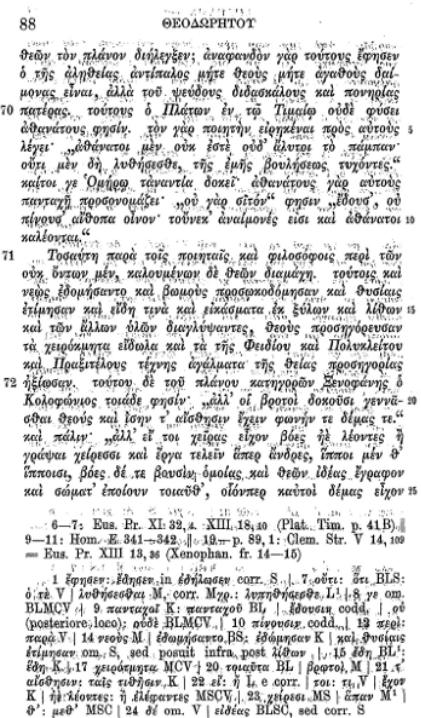


Figure 104: Sample of image binarized by Tesseract

#### 1.9.2.4. Orientation / Skew

This is when text in the image is not straight. The quality of Tesseract's line segmentation reduce significantly if the image is too skew, which severely impact the quality of OCR.

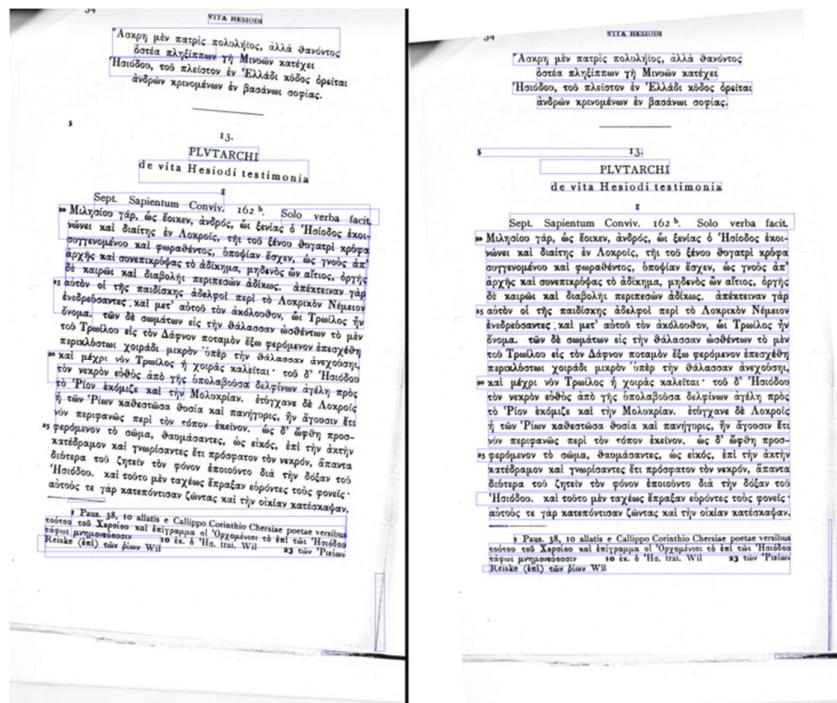


Figure 105: Tesseract's Text line segmentation performed on skew and straight image

### 1.9.2.5. Segmentation

Tesseract is optimized to recognize line of text. If the images are invoice, poster,... that include more than just text in background, it is necessary to extract text area from those image before passing to Tesseract.

### 1.9.3. Tesseract's recognition of Japanese

We conducted a test for Tesseract's accuracy of Japanese recognition. First, we developed a tool to create text's images, and then input these images to Tesseract.

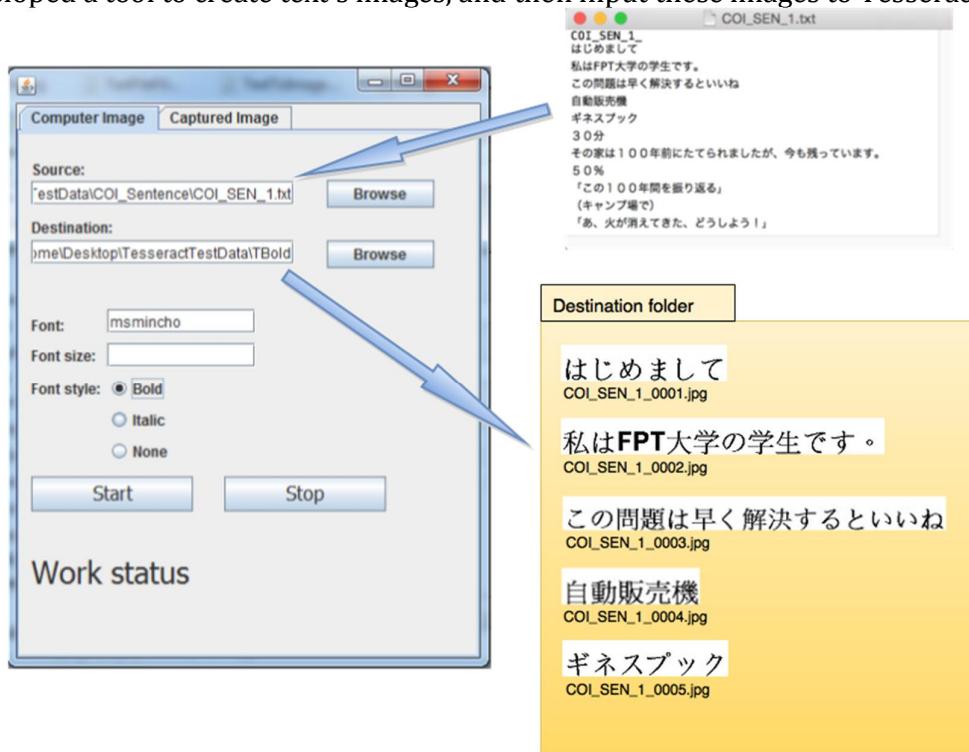


Figure 106: AKP\_CreateImageTool

We have tested with 3 syllabaries (Hiragana, Katakana, Furigana), numbers, and other character of Japanese writing system. We compared the recognition result of Tesseract with the correct text in each image and summarize in the table below.

Image sets	Accuracy (%)					SUMMARY
	Bold Text (size 40)	Italic text (size 40)	Text with size 14	Text with size 40*	Text with size 60	
<b>HIRAGANA</b>						
Monograph Hiragana (か、き、く、...)	78.26	76.08	76.08	78.26	78.26	
Diacritic Hiragana (が、ぎ、ぐ、...)	48.00	32.00	56.00	80.00	44.00	
Digraph Hiragana (しゃ、しゅ、...)	0.00	0.00	17.85	0.00	4.16	
<b>Average:</b>	<b>42.09</b>	<b>36.03</b>	<b>49.98</b>	<b>52.75</b>	<b>42.14</b>	<b>44.60</b>

<b>KATAKANA</b>					
Monograph Katakata (カ、キ、ク、...)	84.78	84.78	84.78	89.13	80.04
Diacritic Katakana (ガ、ギ、グ、...)	56.00	56.00	28.00	40.00	44.00
Digraph Katakana (シャ、シュ、...)	6.54	6.54	6.54	6.54	4.16
<b>Average:</b>	<b>49.11</b>	<b>49.11</b>	<b>39.77</b>	<b>45.22</b>	<b>42.73</b>
<b>KANJI</b>					
Simple Kanji (立、私、...)	67.24	67.24	67.24	65.51	62.07
Complicated Kanji (教、葉、...)	70.00	42.50	42.50	43.75	33.75
<b>Average:</b>	<b>68.62</b>	<b>54.87</b>	<b>54.87</b>	<b>54.63</b>	<b>47.91</b>
<b>TEXT LINE</b>					
1 text line: (火が消えてき た...)	100%	18.18	9.09	18.18	9.09
					<b>11.11</b>
<b>OTHER CHARACTERS</b>					
Special characters (「、！、...」)	35.71	21.42	42.85	42.86	28.57
Numbers (1、2、...)	80.00	80.00	60.00	80.00	80.00
<b>Average:</b>	<b>57.86</b>	<b>50.71</b>	<b>51.43</b>	<b>61.43</b>	<b>54.29</b>
* Text with size 40px equivalent to 20px in x-height					

Beside the weak points of Tesseract listed in section 2, analyzing the test result, we realized that:

- When the text size is too big, a kanji which is combined from some single characters will be recognized as those single character. For example 休 is a combination of Katakana characters イ and Kanji character 木 will be recognize as イ and 木.
- There are some character that Tesseract can not recognize, for example: り or シ

#### 1.9.4. AKP Image Processing Library

This version of AKPImageProcessingLib is able to:

- Extract text lines from simple images which contain black text only. This is to reducing time for Tesseract to processing the whole image, and for better binarization.

- Binarization, in other word convert to threshold image. This is to repair the shortcoming of Tesseract in binarization step.
- Remove noise. This is done in both functions of text line extraction and binarization.
- Extract the exact text area that user selected. When user use finger to select text on screen, they might miss some strokes of the character. AKImageProcessor extracts the text area that cover all the expected characters, with no missed strokes.

Below is class diagram of AKImageProcessingLib

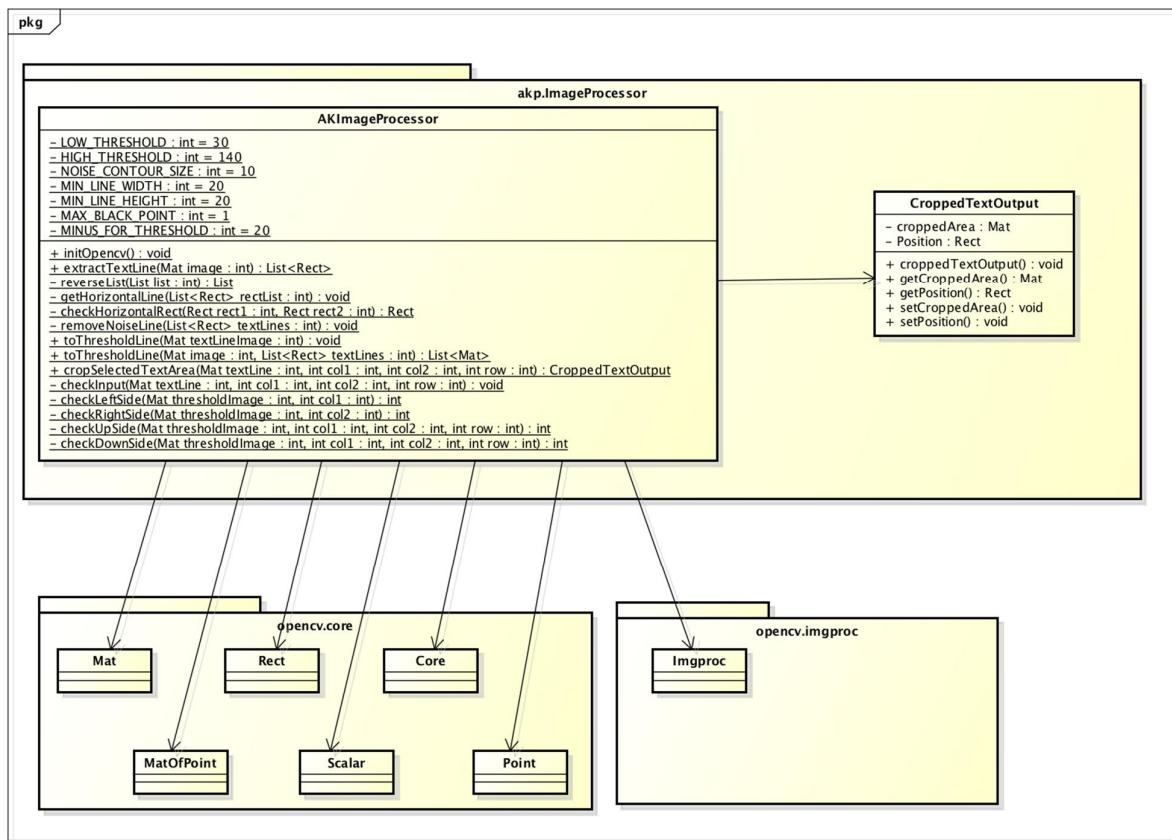


Figure 107: Class diagram of AKImageProcessingLib

#### 1.9.4.1. Extract text line

The function of extract text line follow the sequence described below:

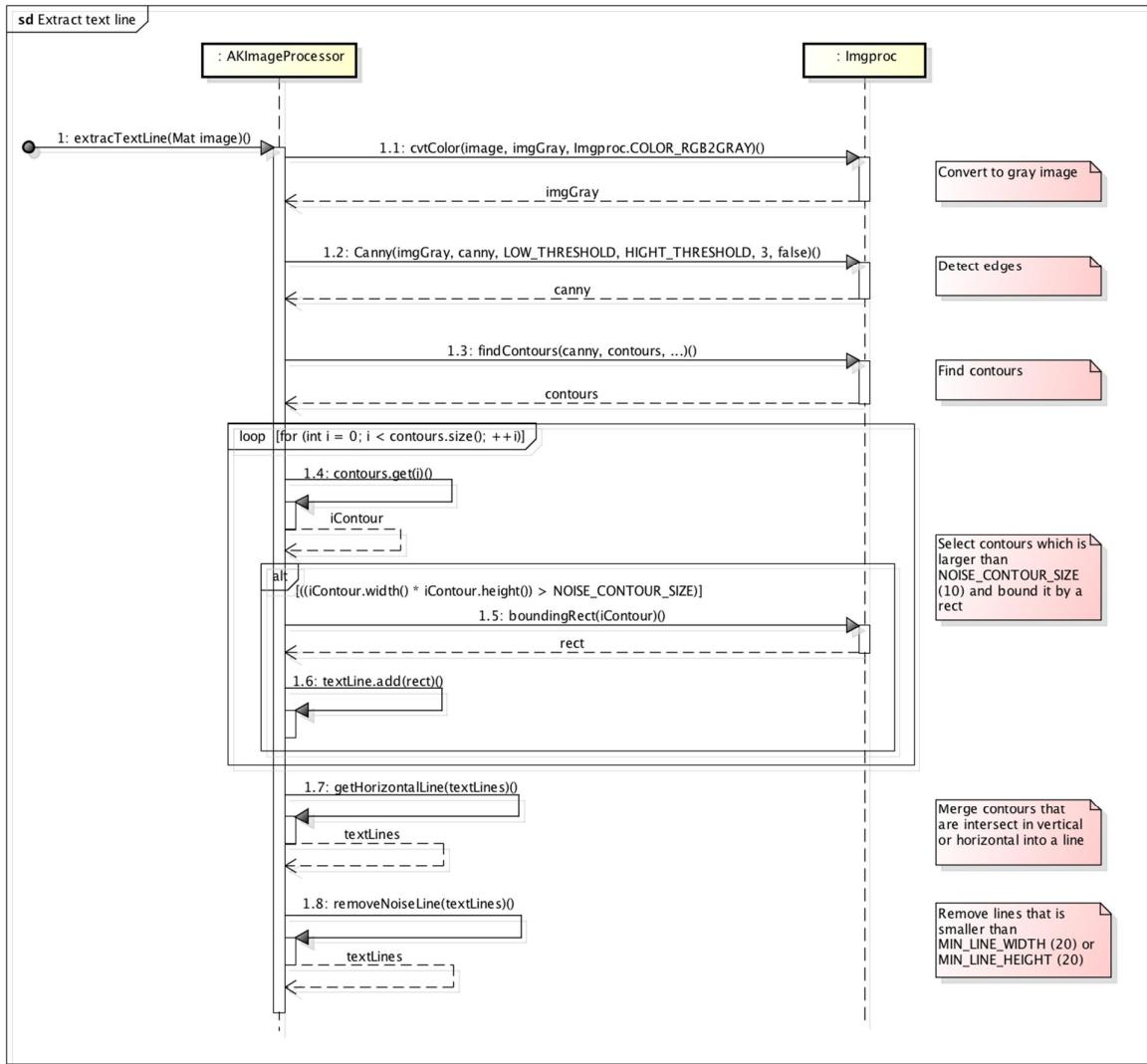


Figure 108: Sequence diagram of extracting text line

#### 1.9.4.1.1. Convert to gray image

Function `cvtColor()` converts color input image to gray image for further process.

#### 1.9.4.1.2. Detect edge by Canny

The method of edge detection is to compute the gradient of pixel intensity find edges<sup>[8]</sup>. Edge occurs when the gradient is greatest. In text image, there is great gradient between background and text because text is often colored so that they stand out of the background, therefore edges of character strokes will be detected.

Edge detection can be done by `Imgproc.Sobel` or `Imgproc.Canny`. In `AKImageProcessor`, we apply `Imgproc.Canny` function. `Sobel` and `Canny` implement similar algorithm to detect edges. The advantage of `Canny` over `Sobel` is that it is better at removing noise and therefore performs better detection<sup>[9]</sup>. Besides, we tried both `Canny` and `Sobel`, and the results showed that `Sobel` often lose a part of the character strokes (an edge of the stroke).

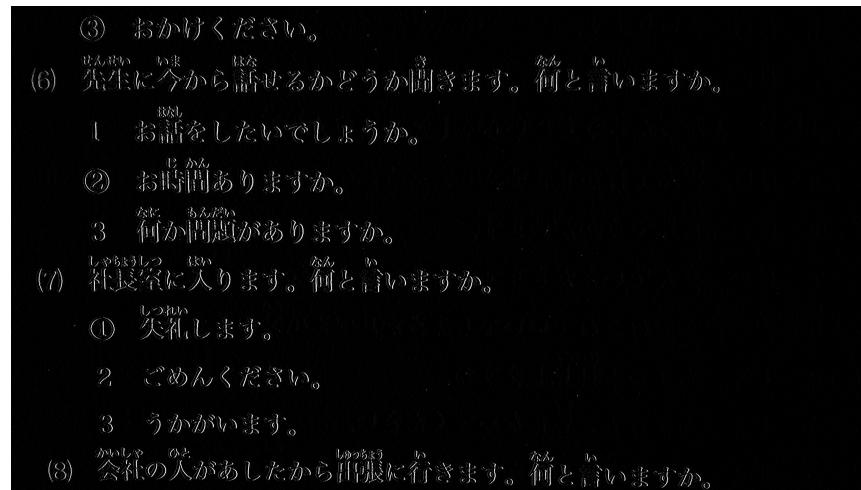


Figure 109: Result of edge detection by Sobel

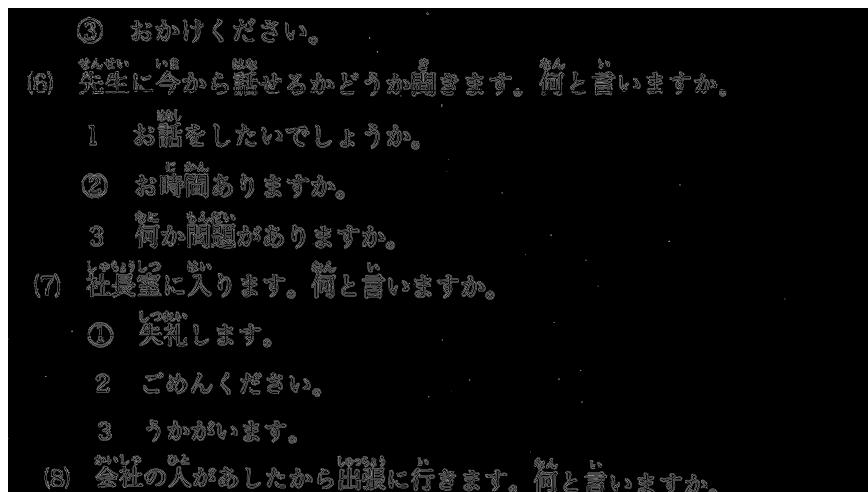


Figure 110: Result of edge detection by Canny

#### 1.9.4.1.3. Find contours

Contour is a curve that connect continuous points which have the same color or intensity. We use findContours function to determine contours from edges outputed by Canny. As Canny detect edges of characters, these contours can be considered as parts of character strokes.

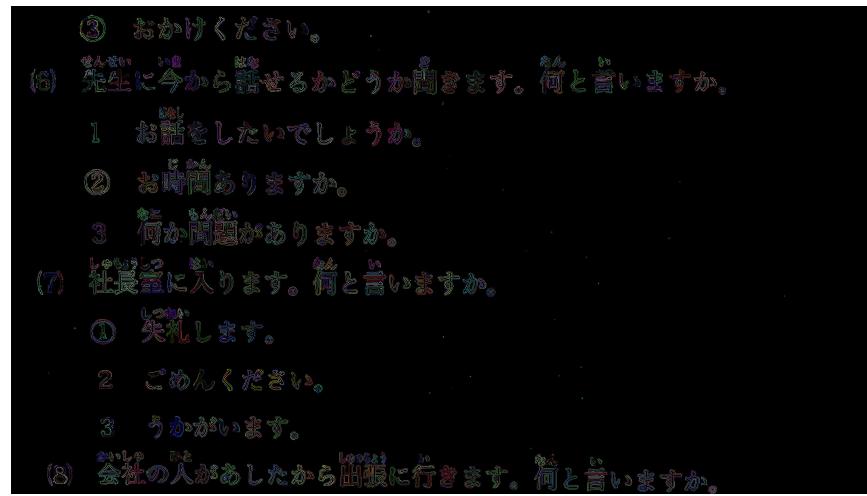


Figure 111: Detected contours, which is a part of character strokes

#### 1.9.4.1.4. Get horizontal line

After we got all contours from step 1.3, we bound contours by rectangles, then merge rectangle that in the same text line into 1 line. 2 rectangle are considered as in the same line if they are intersect in vertical.

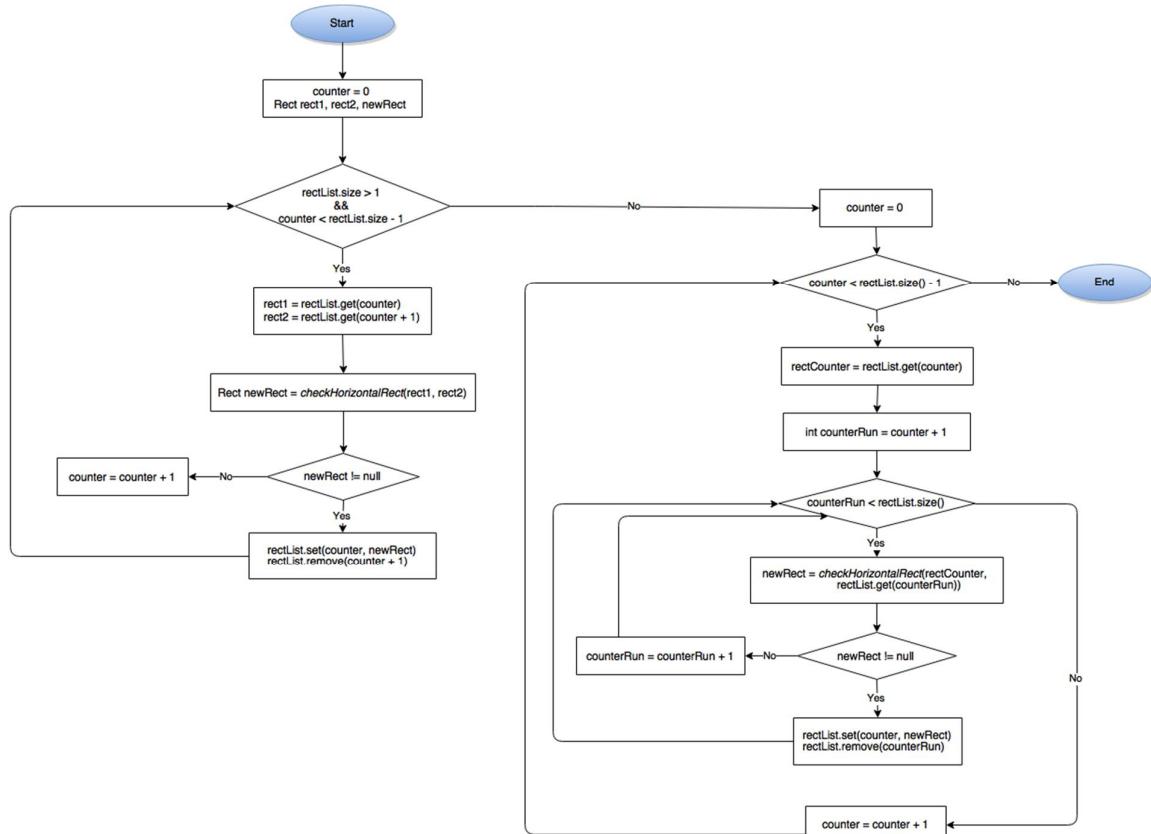


Figure 112: Flowchart of function getHorizontalLine

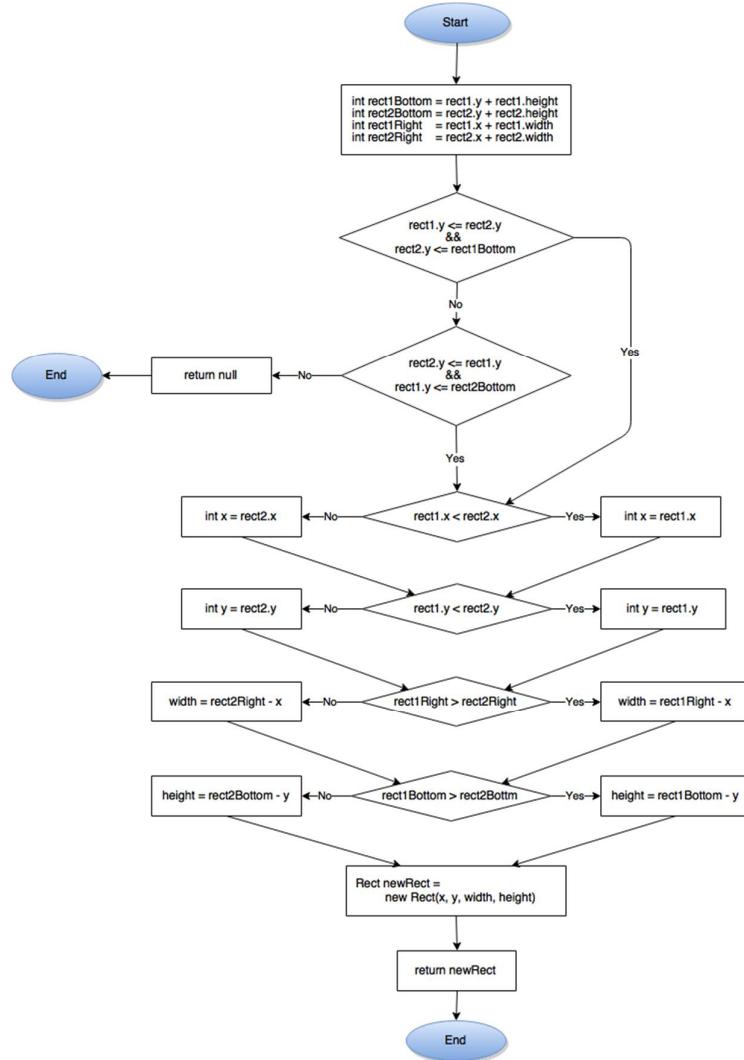
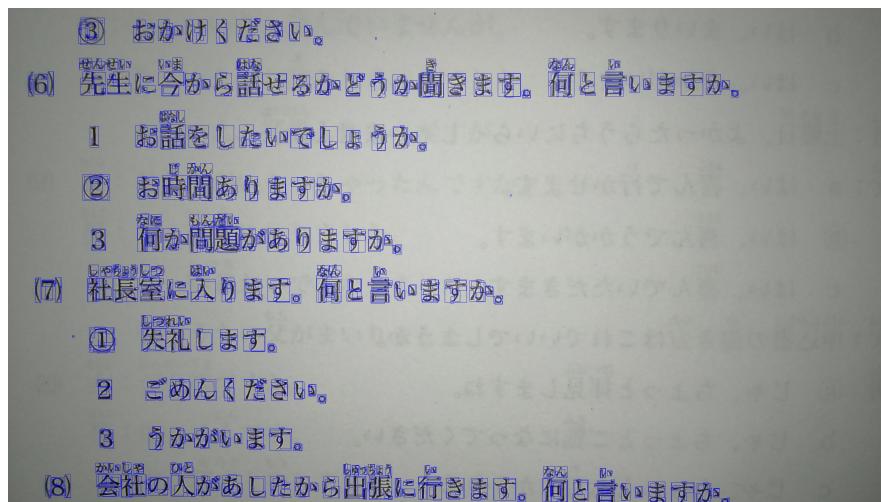
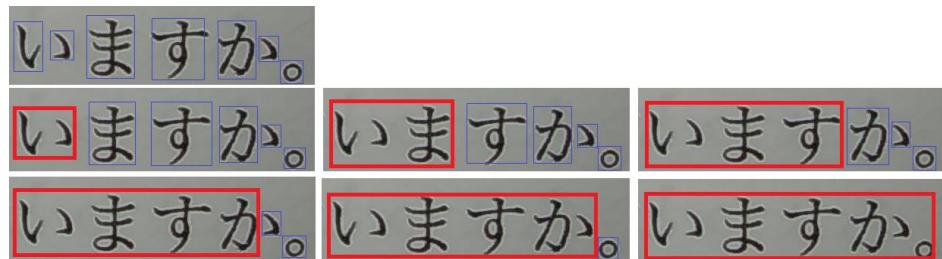


Figure 113: Flowchart of function checkHorizontalRect

Image with contours bounded by rectangles:



Steps of merging contours is illustrate in figure below:

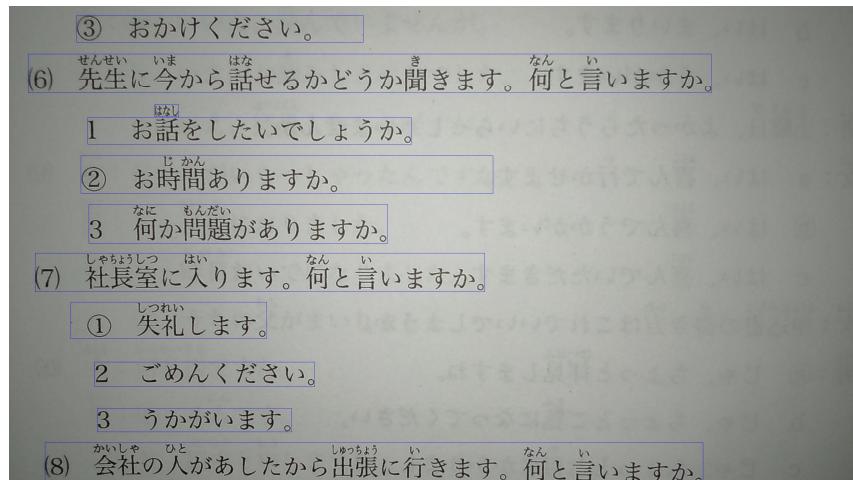


*Figure 114: Example of merging contours into line*

#### **1.9.4.1.5. Remove noise**

- In the loop, all the contours whose area are smaller than NOISE\_CONTOUR\_SIZE (10 as defined) will not be consider as a contours of a character stroke and will not be add to textLines list.
  - In removeNoiseLine functions, text line with height or width smaller than MIN\_LINE\_HEIGHT (20 as defined) or MIN\_LINE\_WIDTH (20 as defined) respectively will be removed. We set the MIN as 20px because Tesseract may not perform well with x-height smaller than 20px, as depicted in section 2.1.

#### **1.9.4.1.6. Result of extractTextLine**



*Figure 115: Final result of extractTextLine()*

### 1.9.4.2. Binarize image

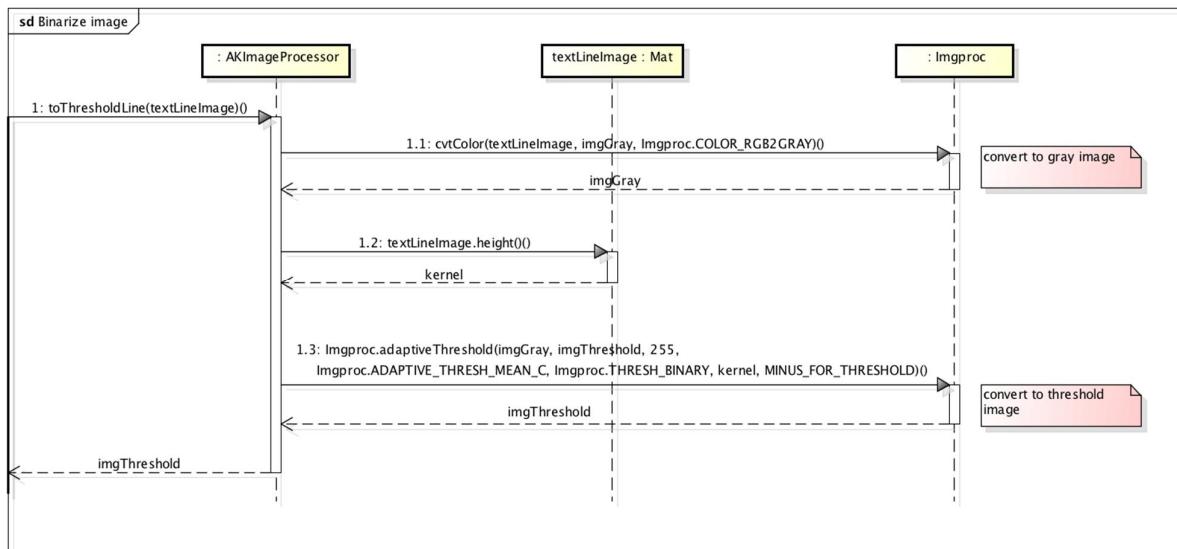


Figure 116: Sequence diagram of binarizing image

The method of image binarization is to compare the intensity value of a pixel to a threshold value. If the intensity of the pixel is smaller than the threshold value, then set it to a determined value, otherwise set it to other determined value. By doing this, the image will be segmented into 2 different area.

OpenCV provides 2 kind of bilarization:

- Apply 1 threshold value for all pixel (`threshold()`). In this, we tried Otsu algorithm, which calculate threshold value base on intensity of all pixel and then apply the threshold value to the whole image. However, the output result was not so good, in case of uneven darkness background.

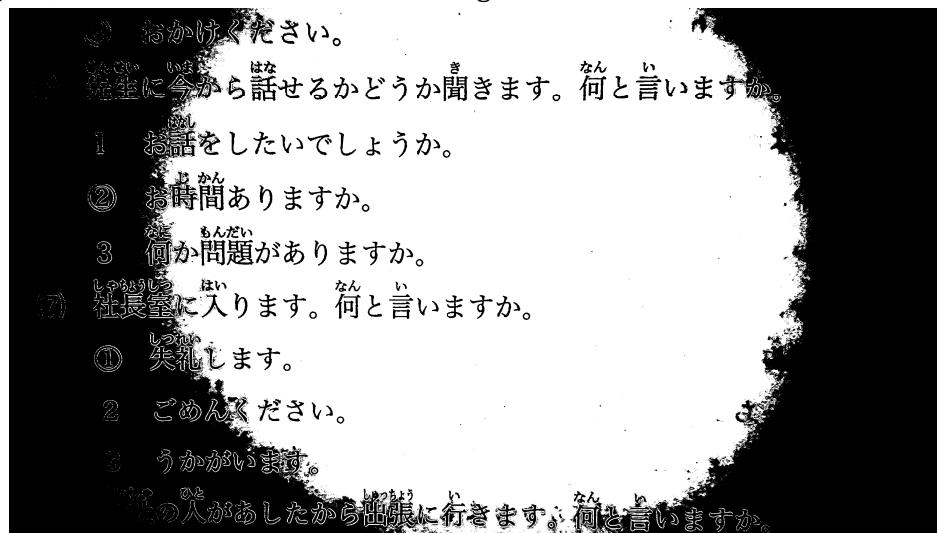


Figure 117: Image binarized by Otsu algorithm

- Calculate threshold value for each single pixel base on it neighbors (`adaptiveThreshold()`). After text lines were extracted from images, we passed the height of each lines to the functions to set as the kernel of comparison. The

result was better than using threshold function, so finally we decided to use this one.

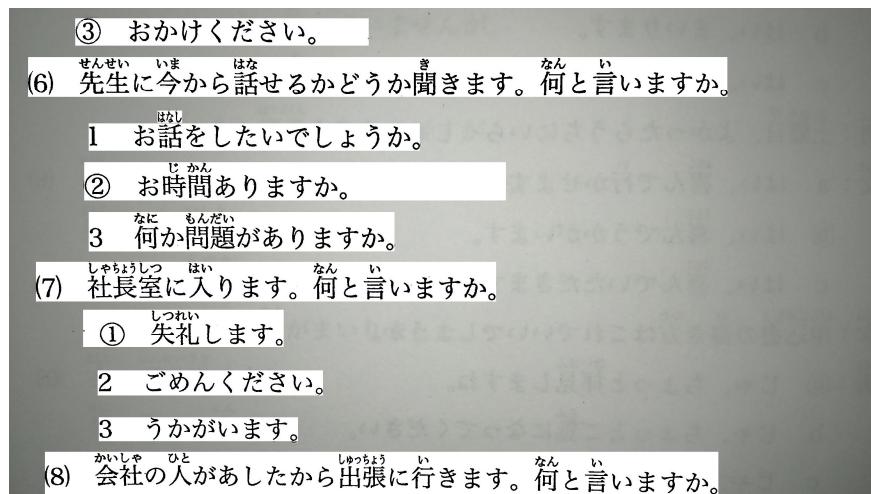


Figure 118: text lines converted to threshold image by adaptiveThreshold

In compare to image converted by Tesseract:

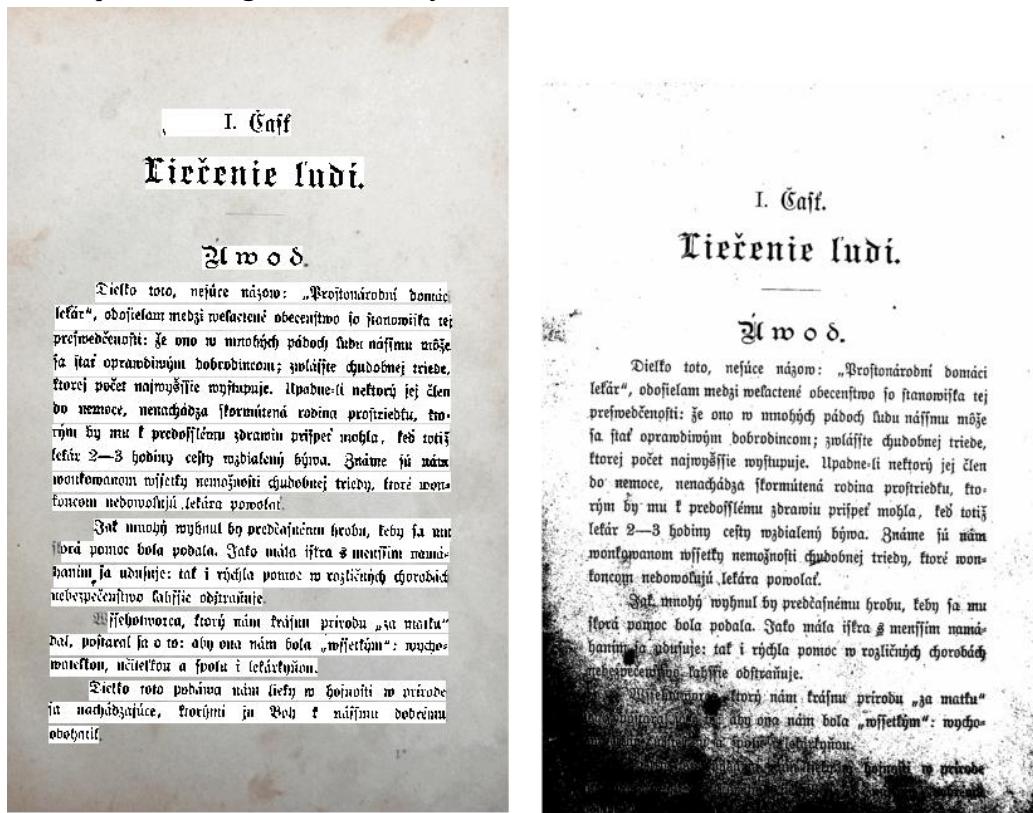


Figure 119: Image converted to threshold by AKImageProcessingLib (left) and Tesseract (right)

#### 1.9.4.3. Extract selected text area

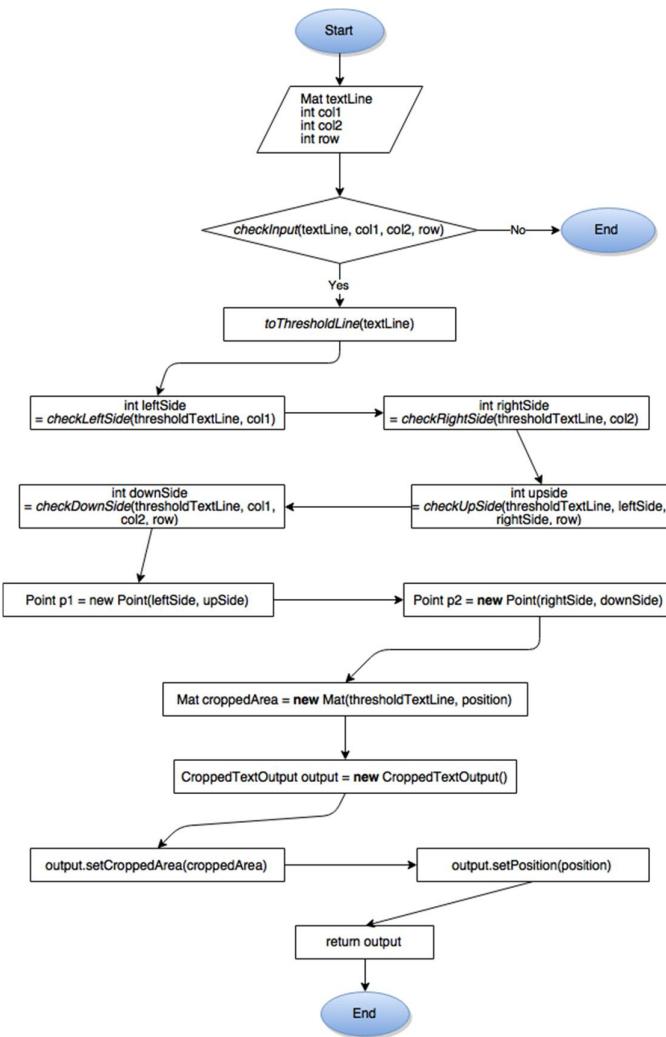


Figure 120: Flow of text line extraction

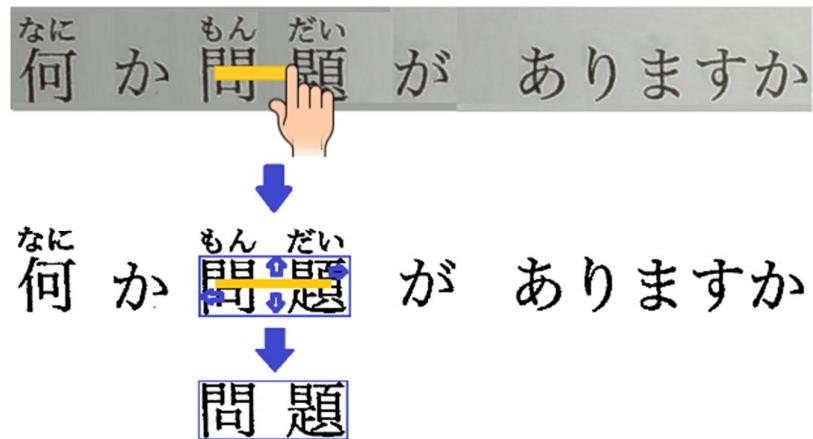


Figure 121: Illustration of text line selection steps

### 1.9.5. AKP image processing library's limitation

This version of AKImageProcessingLib is unable to deskew image. We did try some OpenCV functions but the results for Japanese documents were not as good as English document. In next version we will work more on this issue.

This version does not support document with text printed in vertical direction, which is quite popular format of Japanese document.

## 1.10. Size And Performance

### 1.11. Quality

Anki pan system architecture design aimed at the quality goals as specified below.

#### 1.11.1. Scalability:

- ❖ **Description:** System's reaction when user need more than one language.
- ❖ **Solution:** Sever can provide more than one dictionary to download, and component design allow developer add more function to application.

#### 1.11.2. Reliability / Availability:

- ❖ **Description:** Application may require Internet connection. And it may not read image precisely.
- ❖ **Solution:** Incased there no Internet connection, application change to offline mode that provide offline function to use. And, application can show tutorial that make user easy to use.

#### 1.11.3. Portability:

- ❖ **Description:** Ability to be reused in another environment.
- ❖ **Solution:** Application was programed in java language, so it can running on any environment that contain JDK (java development kit).

#### 1.11.4. Security:

- ❖ **Description:** User may leaked their information.
- ❖ **Solution:** All data or user's Information store in user's device.

## 2. DATABASE DESIGN

### 2.1. Introduction

#### 2.1.1. Purpose

This document is created to define the architecture for the entire data structure and business solution.

What is written in this document will make the developer get to know specifically about the database system components. Beside it, the tester implement the unit test base on this design document.

#### 2.1.2. Overview

This document is a detail description of the database design of Anki pan system. It depicts all tables to be created, information stored in each tables, and the relationship between tables.

The database data is designed and improved based on the original database of "Jdict" app – made by Le Hoang.

## 2.2. Database

### 2.2.1. Application database

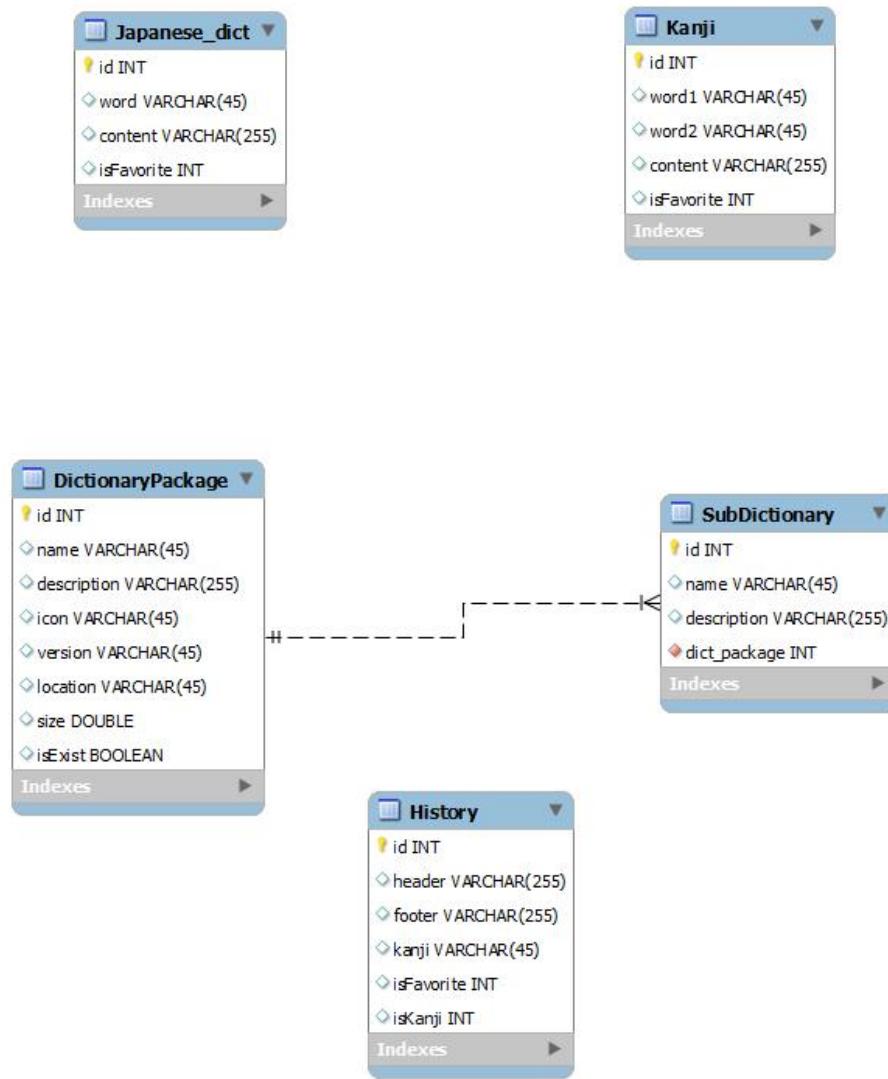


Figure 122: ERD of Application database

No	Table	Description
1	DictionaryPackage	List of packages and their detailed information
2	SubDictionary	List of sub dictionaries and their detailed information
3	Japanese_dict	List of Japanese words and their meaning, type of word, example, etc.
4	Kanji	List of Kanji words , “Han – Viet” pronunciation, meaning, etc.
5	History	Store user's searching history

### ❖ DictionaryPackage table

No	Field name	Type	Length	Size	Null	Unique	P/F Key	Default	Description
1	id	INT				x	P		unique id for each package
2	name	VARCHAR	45						package name
3	description	VARCHAR	255						detailed description about the package
4	icon	VARCHAR	45						path to the location of the package icon
5	version	VARCHAR	45						package version
6	location	VARCHAR	45						package's directory
7	size	DOUBLE							package size

### ❖ SubDictionary table

No	Field name	Type	Length	Size	Null	Unique	P/F Key	Default	Description
1	id	INT				x	P		unique id for each sub_dictionary
2	name	VARCHAR	45						sub_dictionary name
3	description	VARCHAR	255						detailed description about sub_dictionary
4	dict_package	INT							dict_package id that includes the sub_dict

### ❖ History table

No	Field name	Type	Length	Size	Null	Unique	P/F Key	Default	Description
1	id	INTEGER				x	x		The unique ID of each history item
2	header	VARCHAR	255						The header HTML String to be displayed in History
3	footer	VARCHAR	255						The footer HTML String to be displayed in History
4	kanji	VARCHAR	45						The single kanji character of the word (if it is kanji)
5	isFavorite	INTEGER							Favorite status

6	isKanji	INTEGER							Check if the history item is JWord or Kanji
---	---------	---------	--	--	--	--	--	--	---

❖ **Japanese\_dict table**

No	Field name	Type	Length	Size	Null	Unique	P/F Key	Default	Description
1	id	INTEGER				x	P		Unique id of each word
2	word	VARCHAR	32			x			The word text
3	content	VARCHAR	256						The word's type, meaning, examples, etc
4	isFavorite	INTEGER							Values: 0 or 1 to check if the user marks the word as favorite

❖ **Kanji table**

No	Field name	Type	Length	Size	Null	Unique	P/F Key	Default	Description
1	id	INTEGER				x	P		Unique id of each word
2	word1	VARCHAR	32						The “Han” pronunciation
3	word2	VARCHAR	32			x			The kanji text
4	content	VARCHAR	256						The “on” pronunciation and the kanji’s meaning
5	isFavorite	INTEGER							Values: 0 or 1 to check if the user marks the kanji as favorite

## 2.2.2. Server database

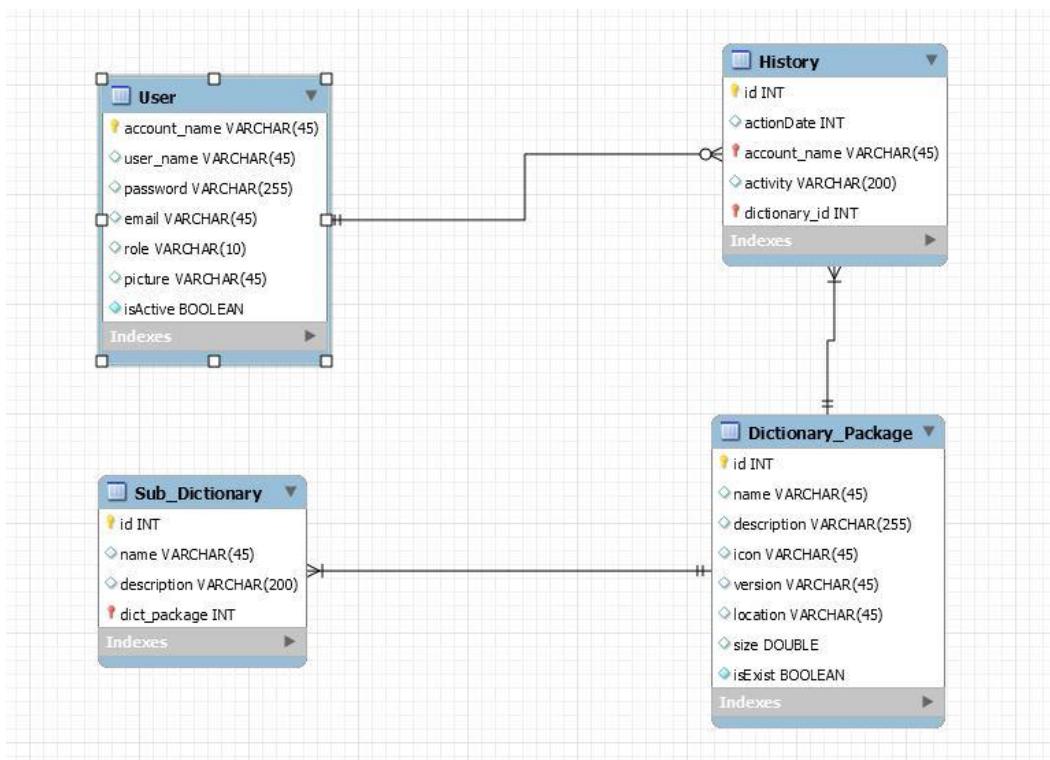


Figure 123: ERD of server database

No	Table	Description
1	User	Store all information of Admin and Root admin accounts
2	History	Store history of updating dictionary database
3	Dictionary package	Store information of dictionary packages available on server
4	Sub_Dictionary	Store information of sub dictionaries belong to dictionary packages

### ❖ User table

No	Field name	Type	Length	Size	Null	Unique	P/F Key	Default	Description
1	account_name	VARCHAR	45			x	P		unique account name for each user
2	user_name	VARCHAR	45						the user's real name
3	password	VARCHAR	255						the account password
4	email	VARCHAR	45						user's email address
5	role	VARCHAR	10						user's role: admin/

									root
6	picture	VARCHAR	45						user's profile picture location
7	isActive	BOOLEAN							account status: true: is currently used

#### ❖ Dictionary package table

No	Field name	Type	Length	Size	Null	Unique	P/F Key	Default	Description
1	id	INT				x	P		unique id for each package
2	name	VARCHAR	45						package name
3	description	VARCHAR	255						detailed description about the package
4	icon	VARCHAR	45						path to the location of the package icon
5	version	VARCHAR	45						package version
6	location	VARCHAR	45						package's directory
7	size	DOUBLE							package size
8	isExist	BOOLEAN							package status true if it currently exists in the database

#### ❖ Sub\_Dictionary table

No	Field name	Type	Length	Size	Null	Unique	P/F Key	Default	Description
1	id	INT				x	P		unique id for each sub_dictionary
2	name	VARCHAR	45						sub_dictionary name
3	description	VARCHAR	200						detailed description about sub_dictionary
4	dict_package	INT							dict_package id that includes the sub_dict

#### ❖ History table

No	Field name	Type	Length	Size	Null	Unique	P/F Key	Default	Description
1	id	INT				x	P		unique id for each

								history item
2	acionDate	INT						the date that the change is made
3	account_name	VARCHAR	45					the account name that make the change
4	activity	VARCHAR	200					activity details
5	dictionary_id	INT						the package id that is affected

## 2.3. Code Design

No	Table	Description
1	Japanese_dict.content	Contain the word's type, meaning, examples, etc.
2	Kanji.content	Contain the on-yomi and the word meaning
3	User.password	Save account password

### 2.3.1. Japanese\_dict.content

This field contains the word's other writing, type, meaning, examples. Each attribute starts with a special character (∴/ ☆/ ◆/ ≈/):

- Word's other writing: The other way of writing the word in Japanese. Start with character '∴'
- Word's type: type of word. Start with '☆'
- Word's meaning: Start with '◆'
- Examples: word's example sentences. There will be one or many examples. Each example starts with '≈' character.
- Examples meaning: start with ':'

### 2.3.2. Kanji.content

Contain the on-yomi and the kanji meaning, separated by '\*' character

### 2.3.3. User.password

Store 2 information to encrypt the password: the password and the salt code. Both fields are stored as hex string.

User's password inputted from web browser will be converted from string to char array while salt will be converted from hex string to byte array. Then we use this 2 values to encrypt password to a byte array, and then convert it to hex string.

The 2 hex string of hashed password and salt will be stored together into the table.

### 3. INTERFACE DESIGN

#### 3.1. Introduction

This document describes the User Interface for the AnkiPan Application an Server.

This document is divided into the following sections:

- Structure and layout describes the structure of the application, screen flow
- Screen Descriptions, which comprises the main part of the document, contains a detailed description of each screen and its component parts.

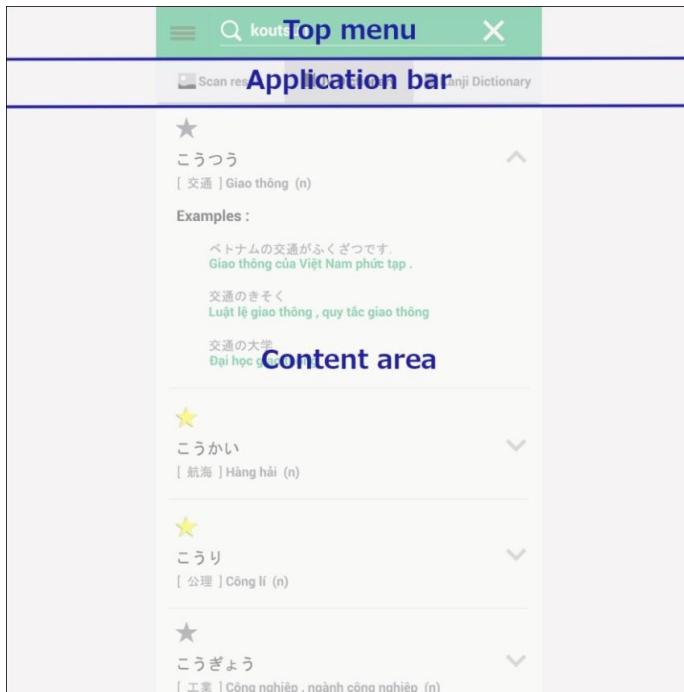
#### 3.2. Interface Design of Application

##### 3.2.1. Structure and Layout

The overall structure of the application is relatively simple, as shown in the following diagram.

No	Screen Name	Description
1	Capture Screen	Where user can take a photograph .
2	Result screen.	The Japanese letters will be captured in image . User can select the japanese letters in this image.User uses these letters as a input text for searching in dictionary.
3	Japanese Dictionary Screen	Where user searchs input word in Japanese dictionary.
4	Kanji Dictionary Screen	Where user searchs input word in Kanji dictionary.
5	History Screen	Where displays the history of searched result.
6	Dictionary Setting Screen	Where user can select dictionary packages to change type of dictionary and delete some dictionary in used package.
7	Download Dictionary Screen	Where user can download new dictionaries or update new version.

The content is displayed in the layout that is shown below :



- **Top menu**

Provide a menu button as a navigation , search bar or title

- **Application bar**

Provide option tabs as the features that user can choose

- **Content Area**

Display the results

Screen flow :

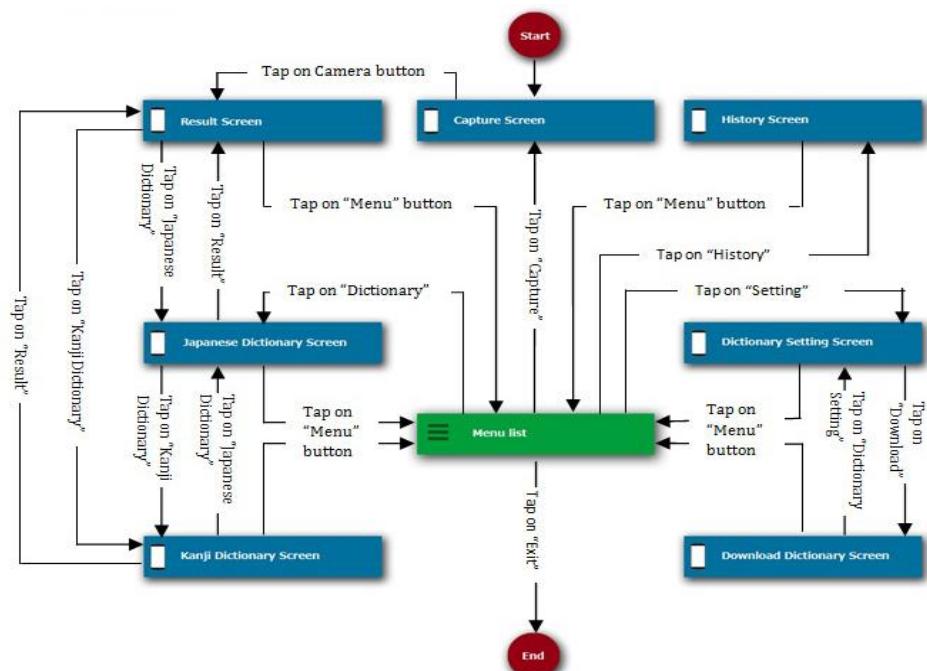


Figure 124: Screen flow of Anki Pan application

### 3.2.2. Screen Description

#### 3.2.2.1. Capture screen

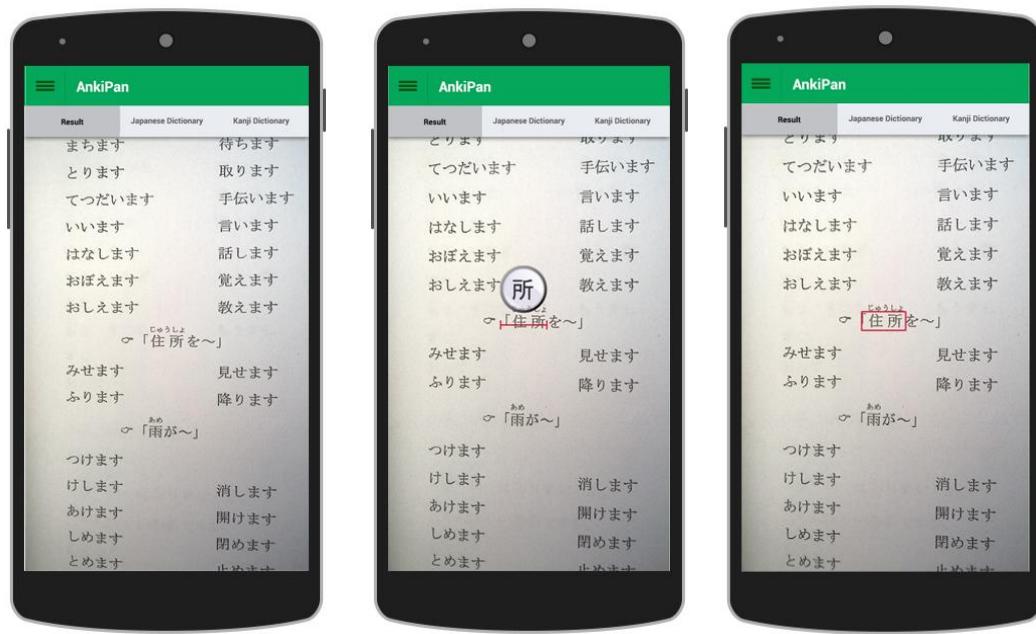
This is the first screen after user launched AnkiPan application where user can capture image ans Japanese document.



No	Elements	Required	Type	Description
1	<b>Menu</b>	Touch on	List view	List of the features ( Capture , Dictionary ,History , Setting , Exit ) when user taped on it.The menu button like a navigation tool to direct to other screens .
2	<b>Title</b>		Label	The title is “AnkiPan”
3	<b>Ruler</b>			Ruler helps user to adjust the balance for photos . User should set parallel the ruler to the textline before taking the photos.
4	<b>Camera button</b>	Touch on	Button	User taps on capture button to take the photo.

#### 3.2.2.2. Result screen

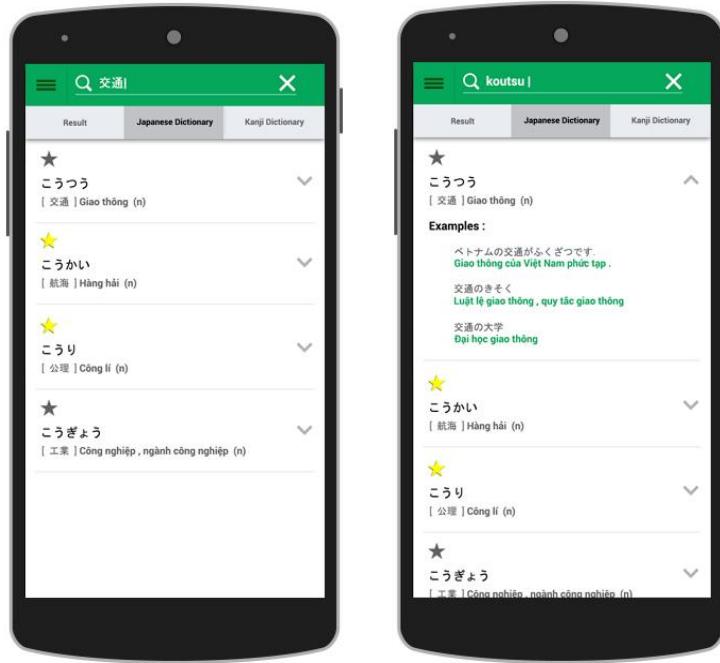
When user captures the japanese letter , AnkiPan will display the image. User can touch on the image to select letters. The magnifier will zoom in the selected letter. Application marks the start point and end point of the selected letters.When user finished , the selected words will be ready for searching.



No	Elements	Required	Type	Default	Description
1	<b>Menu</b>	Touch on	List view		List of the features (Capture ,Dictionary ,History , Setting , Exit ) when user taped on it.The menu button like a navigation tool to direct to other screens .
2	<b>Title</b>		Label		The title is “AnkiPan”
3	<b>Tab bar</b>	Touch on	Sliding tab	[Result] is selected	Tabs make it easily to switch between dictionaries.
4	<b>Magnifier</b>	Touch on			Zoom in selected letters
5	<b>Red line</b>				Marks start point to end point when user touch on screen.
6	<b>Red frame</b>				Wraps the content that user selected.

### 3.2.2.3. Japanese dictionary screen

After selecting letters in image , user can use tab bar to select . AnkiPan provides 2 dictionary Japanese Dictionary and Kanji Dictionary , user inputs text in search bar directly and selects dictionary in tab bar.The result will be displayed below.

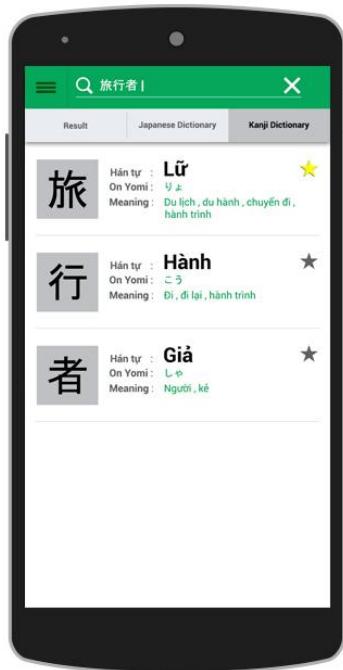


No	Elements	Required	Type	Default	Description
1	<b>Menu</b>	Touch on	List view		List of the features ( Capture , Dictionary ,History , Setting , Exit ) when user taped on it.The menu button like a navigation tool to direct to other screens .
2	<b>Search bar</b>	Touch on	Search bar		Input text for translation. The selected letters in captured image also are displayed here
3	<b>Tab bar</b>	Touch on	Sliding tab	[Japanese Dictionary] is selected	Tabs make it easy to switch between dictionaries.
4	<b>Star</b>	Touch on	Button		Marks the searched word in history and favorite. Tap again to unmark
5	<b>Hiragana</b>		TextView		Displays hiragana of searched words.
6	<b>Kanji</b>		TextView		Displays kanji of searched words if it is possible.
7	<b>Meaning</b>		TextView		Displays the meaning of search words.
8	<b>Type of word</b>		TextView		Displays the type of search word

<b>9</b>	<b>Examples</b>		TextView		Displays some examples
<b>10</b>	<b>Expandable Button</b>	Touch on	Expandable list view		Shows more and hides information about searched words.

### 3.2.2.4. Kanji dictionary screen

User can select words from captured image or type on search bar directly to search .The result will be displayed as below. When user select two or more words, AnkiPan will return corresponding Sino-Vietnamese words exactly.

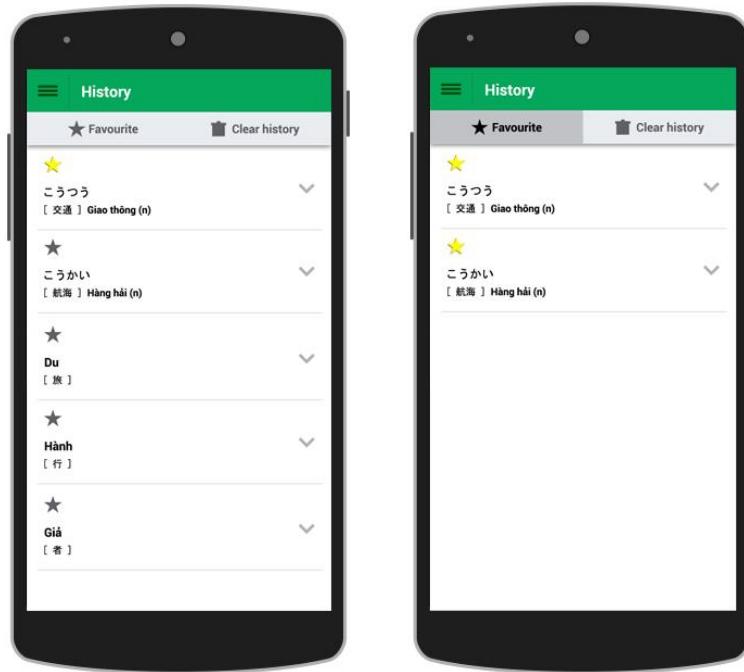


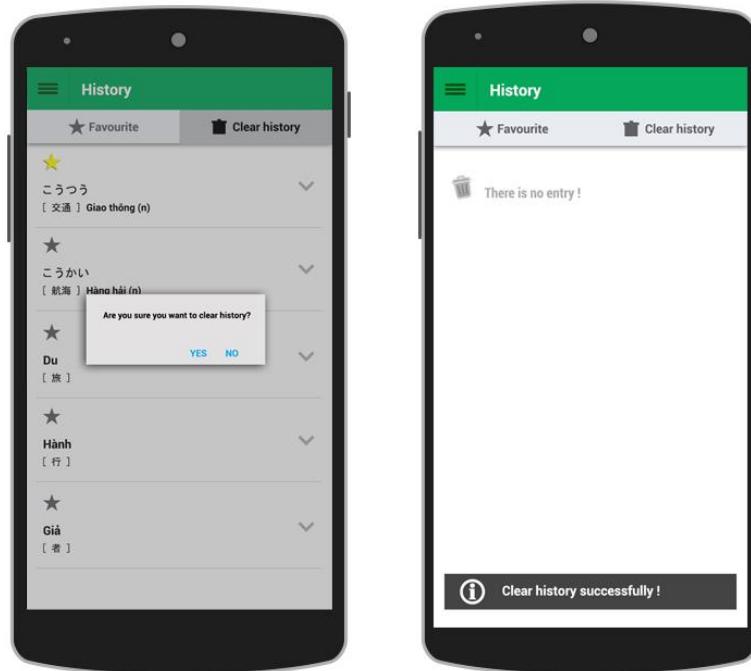
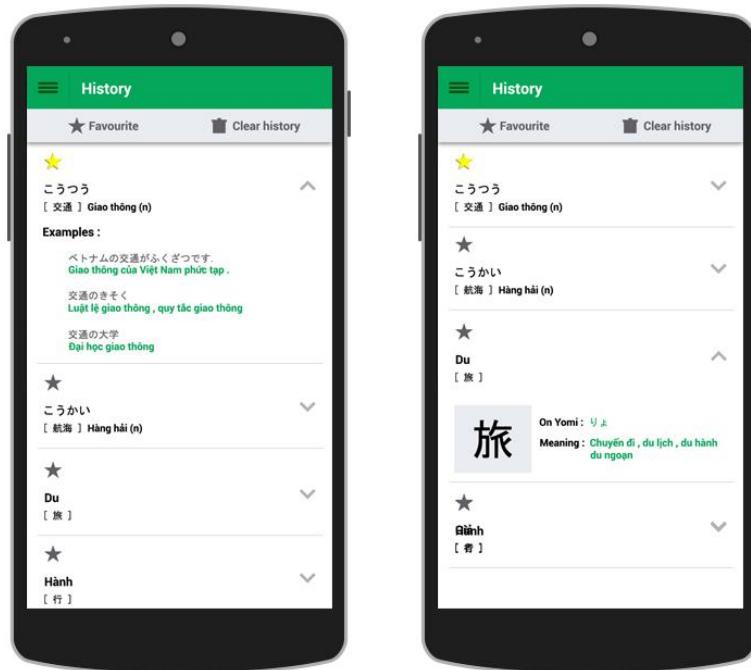
No	Elements	Required	Type	Default	Description
<b>1</b>	<b>Menu</b>	Touch on	ListView		List of the features ( Capture,Dictionary ,History,Setting,Exit)when user taped on it.The menu button like a navigation tool to direct to other screens .
<b>2</b>	<b>Search bar</b>	Touch on	SearchBar		Input text for translation. The selected letters in captured image also are displayed here
<b>3</b>	<b>Tab bar</b>	Touch on	Sliding tab	[Kanji Dictionary] tab is selected.	Tabs make it easy to switch between dictionaries

<b>4</b>	<b>Big Kanji</b>		TextView		Displays Kanji of searched word in big size.
<b>5</b>	<b>Sino-Vietnamese</b>		TextView		Displays Sino-Vietnamese of search word in big size.
<b>6</b>	<b>On Yomi</b>		TextView		Displays on yomi of corresponding kanji
<b>7</b>	<b>Meaning</b>		TextView		Displays the meaning of corresponding kanji.
<b>8</b>	<b>Star</b>	Touch on	Button		Marks the searched word in history and favourite. Taps again to unmark.

### 3.2.2.5. History screen

History help user saving searched word so that user can review those words. User can use favourite tab to display all marked words. By touching on star button user can switch between marked and unmarked status. When history is so long with many records , user can clear history to delete all records .



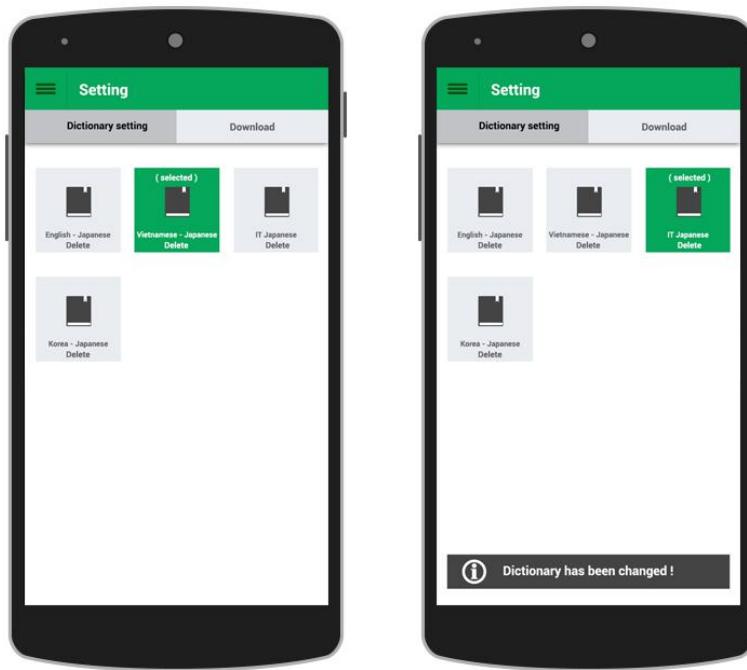


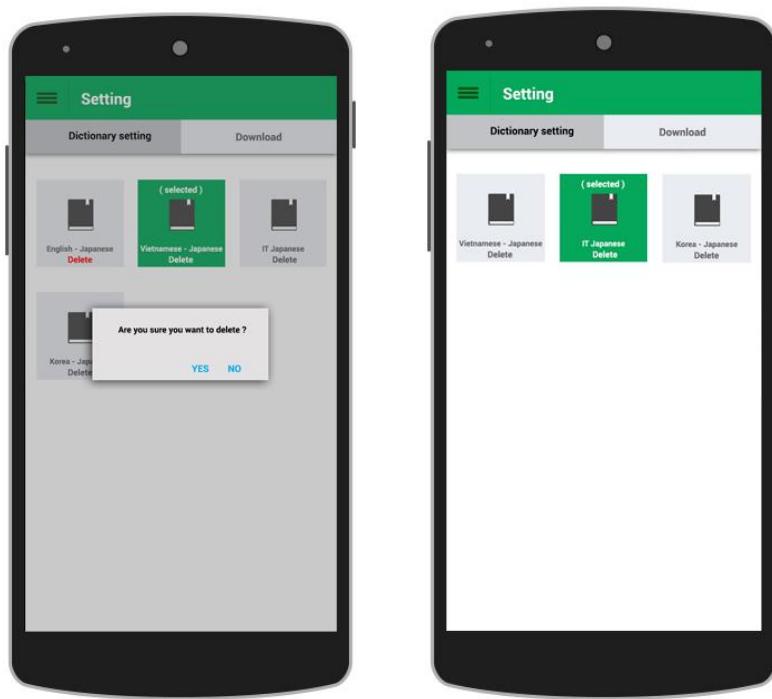
No	Elements	Required	Type	Description
1	<b>Menu</b>	Touch on	List View	List of the features ( Capture , Dictionary ,History , Setting , Exit ) when user taped on it.The menu button like a navigation tool to direct to other screens .
2	<b>Title</b>		Label	Title is "History"
3	<b>Favorite</b>	Touch on	Button	Displays the list of marked words.

<b>4</b>	<b>Clear history</b>	Touch on	Button	Deletes all records in history.
<b>5</b>	<b>Star</b>	Touch on	Button	Marks the searched word in history and favorite. Tap again to unmark.
<b>6</b>	<b>Hiragana</b>		TextView	Display hiragana of saved words.
<b>7</b>	<b>Kanji</b>		TextView	Display kanji of saved words.
<b>8</b>	<b>Meaning</b>		TextView	Display the meaning of saved words.
<b>9</b>	<b>Sino-Vietnamese</b>		TextView	Display the Sino-Vietnamese of saved words.
<b>10</b>	<b>Expandable button</b>	Touch on	Button	Show or hide more information of saved words.
<b>11</b>	<b>Confirmation</b>		ShowDialog	Ask user before do deleting process
<b>12</b>	<b>Notice</b>		TextView	Notice "There is no record" when clear history successfully.

### 3.2.2.6. Dictionary setting screen

Dictionary setting screen display all dictionaries which user had. The dictionary that user is using will be selected as default .In this screen , user can change dictionary and delete dictionary.





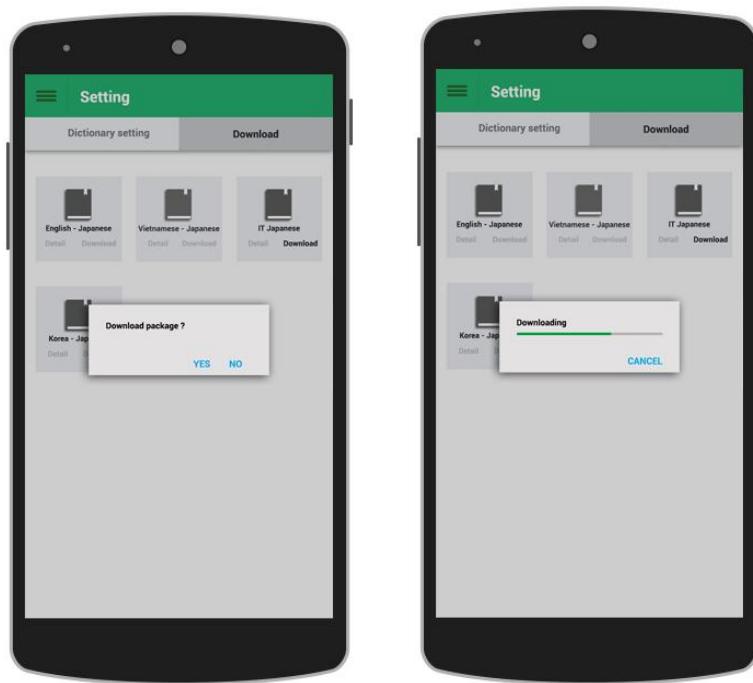
No	Elements	Required	Type	Default	Description
1	<b>Menu</b>	Touch on	List View		List of the features ( Capture , Dictionary ,History , Setting , Exit ) when user taped on it.The menu button like a navigation tool to direct to other screens .
2	<b>Title</b>		Label		Title is "Setting"
3	<b>Tab bar</b>	Touch on	Sliding tab	[Dictionary setting ] tab is selected	Tabs make it easy to switch between dictionary setting and download.
4	<b>Types of Dictionary</b>	Touch on		Dictionary that is using is marked	Display all dictionaries in package . The dictionary that is using will be marked as default.
5	<b>Status</b>		Label	[selected]	[selected] status is displayed when a dictionary is selected.
6	<b>Delete</b>	Touch on	Button		Delete corresponding dictionary.

7	<b>Confirmation</b>		ShowDialog		Asks user before deleting dictionary
8	<b>Notice</b>		ShowDialog		Notice to user when change dictionary or delete dictionary successfully.

### 3.2.2.7. Dictionary download screen

Download screen display all dictionary packages. User can download new package or update new version here.





No	Elements	Required	Type	Default	Description
1	<b>Menu</b>	Touch on	List View		List of the features ( Capture , Dictionary ,History , Setting , Exit ) when user taped on it.The menu button like a navigation tool to direct to other screens .
2	<b>Title</b>		Label		Title is "Setting"
3	<b>Tab bar</b>	Touch on	Sliding tab	[Download ] tab is selected	Tabs make it easy to switch between dictionary setting and download.
4	<b>Dictionary packages</b>				Display all dictionary packages that user can download or update .
5	<b>Download</b>	Touch on	Button		Download dictionary.
6	<b>Detail</b>	Touch on	Button		Displays information for new version or new packages.
7	<b>Confirmation</b>		ShowDialog		Asks user before downloading and updating dictionary.
8	<b>Progress bar</b>		Progress bar		Displays the downloading process and updating process.

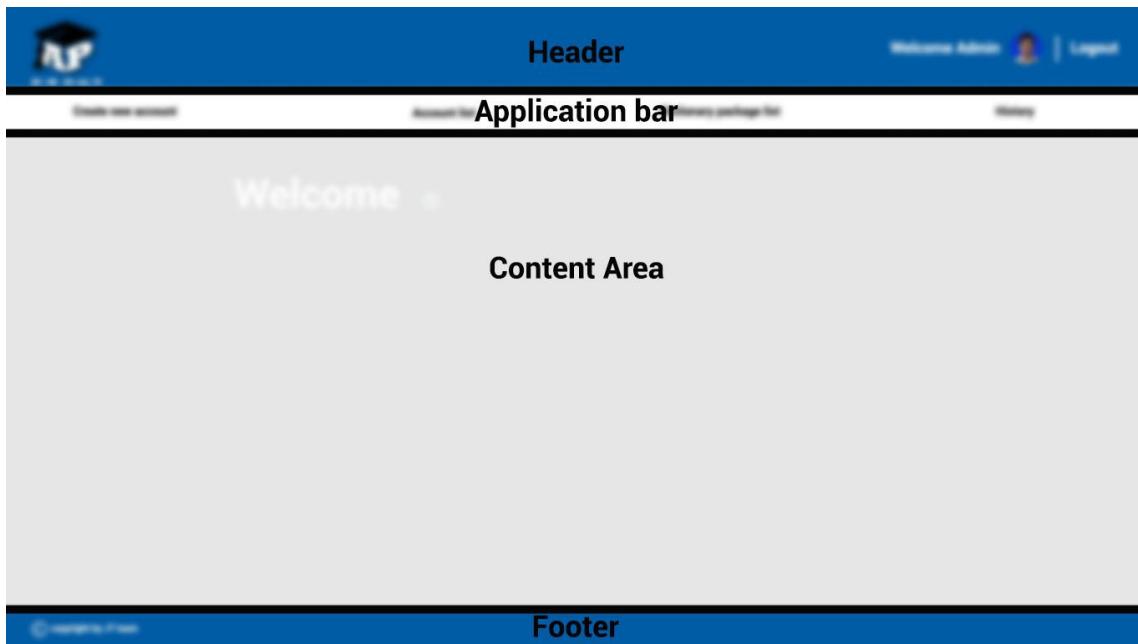
### 3.3. Interface Design of Server

#### 3.3.1. Structure and layout

The overall structure of the website is relatively simple, as shown in the following diagram.

No	Screen Name	Description
1	Login page	Where user enters username and password to log in website .
2	Home page	The first page after logging facilitate navigation to other pages on the site.
3	Account list page	Where user can see the account list , create new account , edit account , delete account.
4	Package list page	Where displays all dictionary packages, create new package , edit package , delete package.
5	Package detail page	Where displays the detail of specific package that include all sub-dictionary in a package.
6	History page	Where displays the history of actions.

The content is displayed in the layout that is shown below :



- **Header**



Shows :

\_ Ankipan logo

- \_ Current user
- \_ Avatar
- \_ Links to sign out.

- **Application bar**



A tab bar that provides links to other pages :

- \_ "Create new account" tab ( If user is root admin )
- \_ "Account list" tab
- \_ "Dictionary package list" tab
- \_ "History" tab

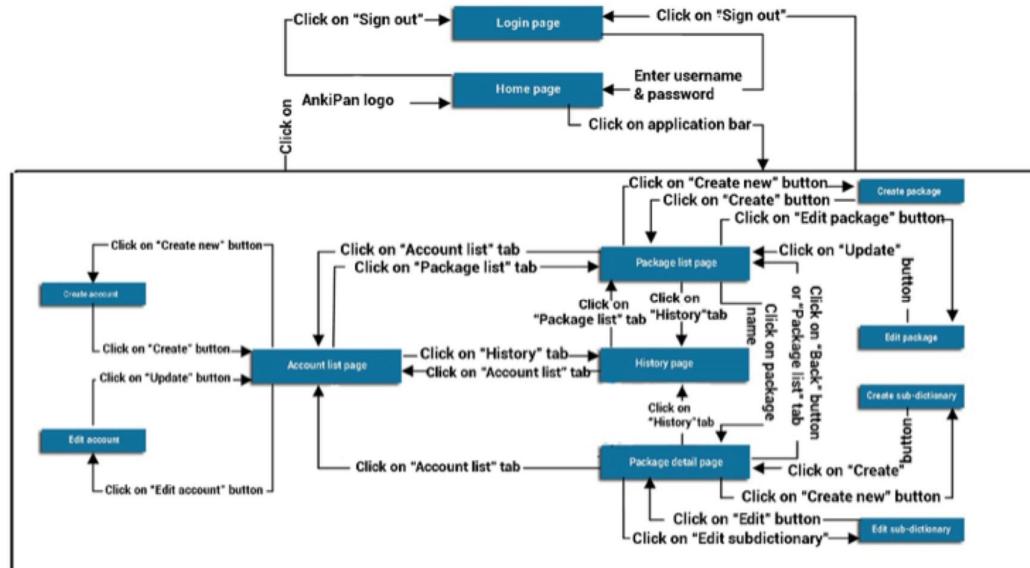
- **Content area**

The content of each pages will be described in topic **3. Page content description**

- **Footer**



Screen flow:

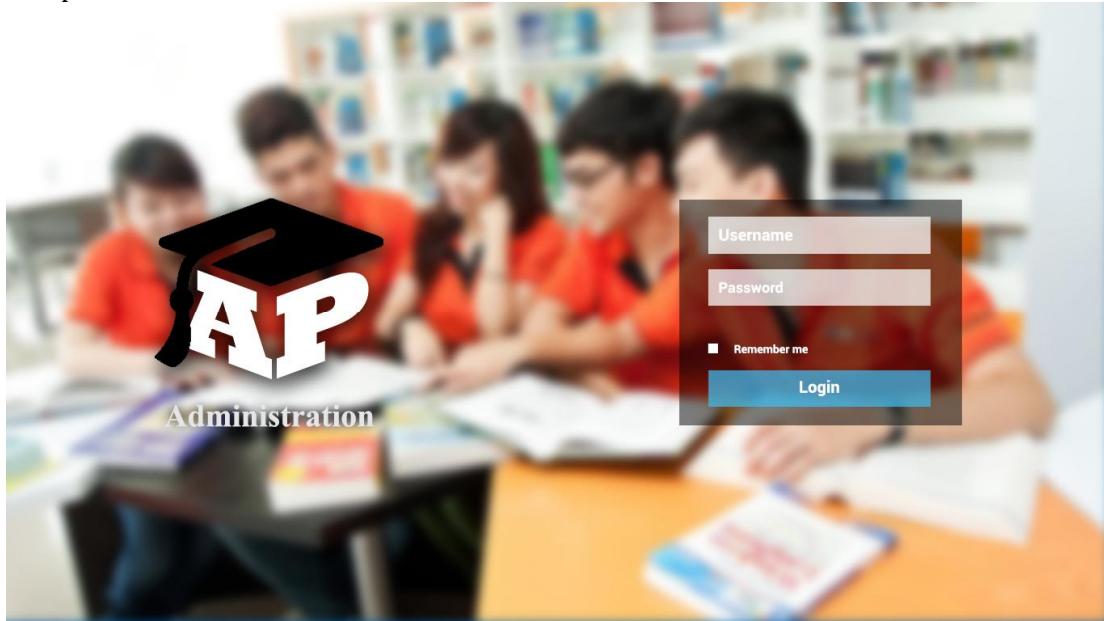


*Figure 3: Screen flow of server*

### 3.3.2. Page content description

#### 3.3.2.1. Login page

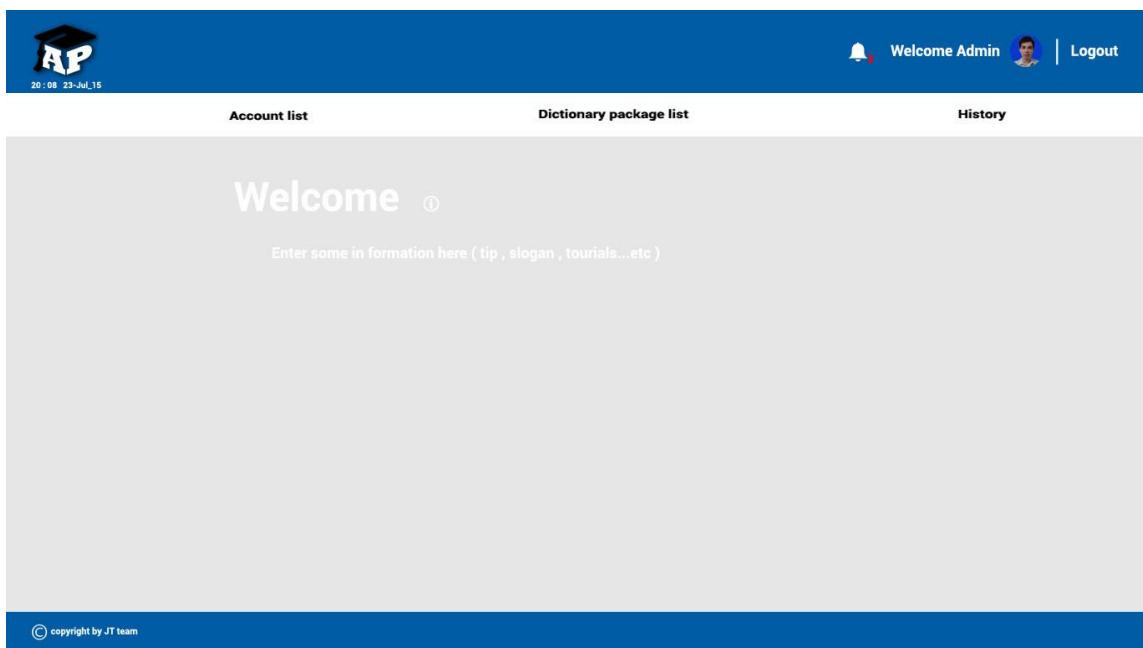
This is the first page where user enters username and password to log in website. In the first page , user can see the AnkiPan logo and login form .User can save username and password in Cookies.



No	Components	Elements	Type	Description
1	AnkiPan logo	AnkiPan logo	Image	The logo of AnkiPan application.
2	Login form	Username	TextField	Where user enters username .
		Password	PasswordChar	Where user enters password.
		Remember password	Check box	Save password to the web browse's cookies.
		Login button	Button	Where user click on to submit username and password.

#### 3.3.2.2. Home page

This is a landing page , where displays all application links to other pages . The greeting will be displayed in the content area .



### 3.3.2.3. Account list page

#### 3.3.2.3.1. Account list

User click on account list tab to link to account list page. This page will display the list of accounts as a table.

Account list				
Account name	Username	Email	Role	Picture
<input checked="" type="checkbox"/> Lê Thanh Bình	admin	binhlitse02917@fpt.edu.vn	Root Admin	
<input type="checkbox"/> accountName1	username1	email1@gmail.com	Admin	

No	Column	Description
1	Account name	Displays all account names. This column is sorted by name.
2	User name	Displays all user names . This column is sorted by name.

<b>3</b>	Email	Displays email of accounts.
<b>4</b>	Role	Displays the role of accounts.
<b>5</b>	Picture	Displays the picture of account as avatar if it is possible.

### 3.3.2.3.2. Create new account button

Root Admin click on “Create new account” button to create new account.



### 3.3.2.3.3. Create new account form

In the application bar , Root Admin clicks on create new account tab to link to Create New Account page . In this page , Root Admin can create new accounts.

The screenshot shows the 'Create new Account' dialog box. It contains the following fields:

- Account name: Text input field.
- Username: Text input field.
- Password: Text input field.
- Re-enter password: Text input field.
- Email: Text input field.
- Profile picture: A file selection input field with a placeholder "No file selected".
- Create: A blue button at the bottom right.

No	Elements	Type	Description
<b>1</b>	Account name	TextField	Where Root Admin enters the account name.
<b>2</b>	User name	TextField	Where Root Admin enters the username.
<b>3</b>	Password	Password Char	Where Root Admin enters the password.
<b>4</b>	Re-enter Password	Password Char	Where Root Admin re-enters the password.
<b>5</b>	Email	TextField	Where Root Admin enters the email of user account.

<b>6</b>	Profile Picture	File Browse	Where Root Admin select picture to set avatar.
<b>7</b>	Create button	Button	Click on to submit information.

### 3.3.2.3.4. Tool box

When user selected an account , a Tool box will be displayed .It contain : “Change password” button , “Edit profile” button , “Delete account” button.



- Change password button , link to “Change password” form.
- Edit profile button , link to “Edit profile ” form.
- Delete account button , delete account.

### 3.3.2.3.5. Change password form

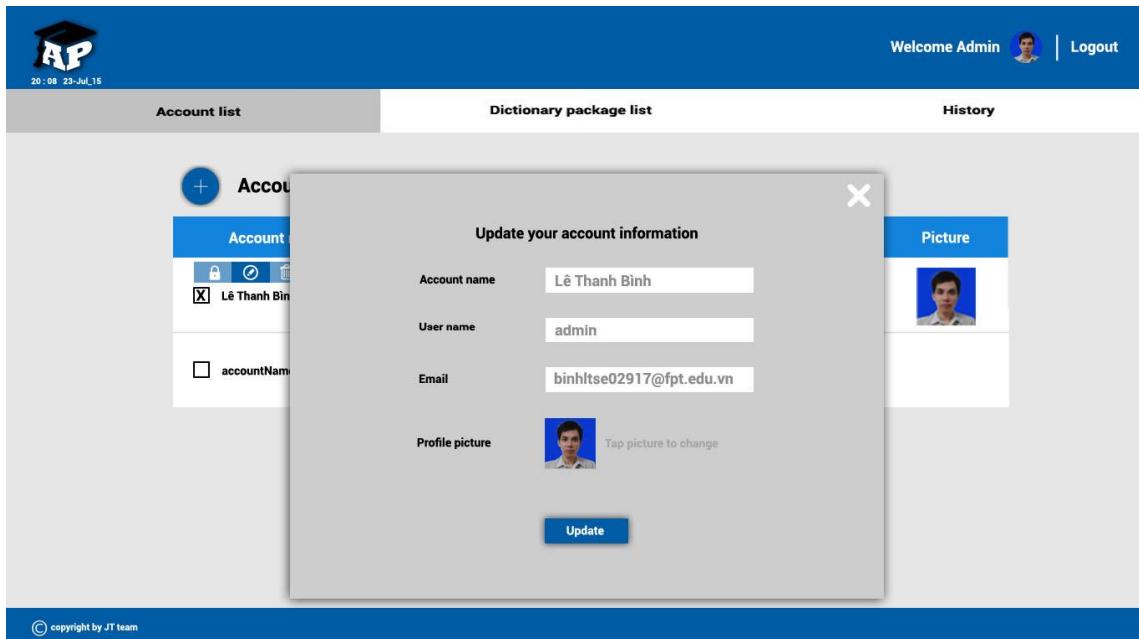
In account list page , after selecting an account , root Admin click on “Change Password” button in the tool bar, change password form will be displayed. Root Admin enters new password to change password.

No	Elements	Type	Description
<b>1</b>	Old password	Password char	Where displays old password
<b>2</b>	New password	Password char	Where user enters the new password.

3	Re-enter new password	Password char	Where user re-enters the new password.
4	Change button	Button	Click on to submit new password, the password will be changed .

### 3.3.2.3.6. Edit profile form

In account list page , after selecting an account , root Admin click on “Edit profile” button in the Tool box, “Edit profile” form will be displayed. Root Admin enters new information to update profile.



No	Elements	Type	Description
1	Account name	TextField	Where displays selected account name.
2	User name	TextField	Where displays user name of account that has been selected
3	Email	TextField	Where displays email of account that has been selected.
4	Profile picture	File browse	Displays old picture of account that has been selected . Root admin can change picture.
5	Update button	Button	Clicks on to update new information.

### 3.3.2.4. Package list page

#### 3.3.2.4.1. Package list

User click on “Package list” tab to link to package list page. This page will display the list of package as a table.

The screenshot shows a web application interface for managing dictionary packages. At the top, there's a blue header bar with the logo 'AP' and the date '20:08 23-Jul\_15'. On the right side of the header are links for 'Welcome Admin' (with a user icon) and 'Logout'. Below the header, there are three tabs: 'Account list', 'Dictionary package list' (which is currently selected and highlighted in grey), and 'History'. The main content area is titled 'Package list' and contains a table with two rows of data. The columns are labeled 'ID', 'Name', 'Version', 'Description', and 'Icon'. Row 101 shows a package named 'Vietnamese' with version 1.21, described as 'A dictionary package that includes Vietnamese and Japanese', and has a small profile picture of a person next to it. Row 102 shows a package named 'English' with version 1.0, described as 'A dictionary package that includes English and Japanese'. Each row also includes a set of icons for edit, delete, and details.

No	Column	Description
1	ID	Displays all package ID. It is also a hyper link that links to "Package detail" page .
2	Name	Displays the package name. It is also a hyper link that links to "Package detail" page .
3	Version	Displays the package version.
4	Description	Describes the package simply and sortly.
5	Icon	Display the package icon.

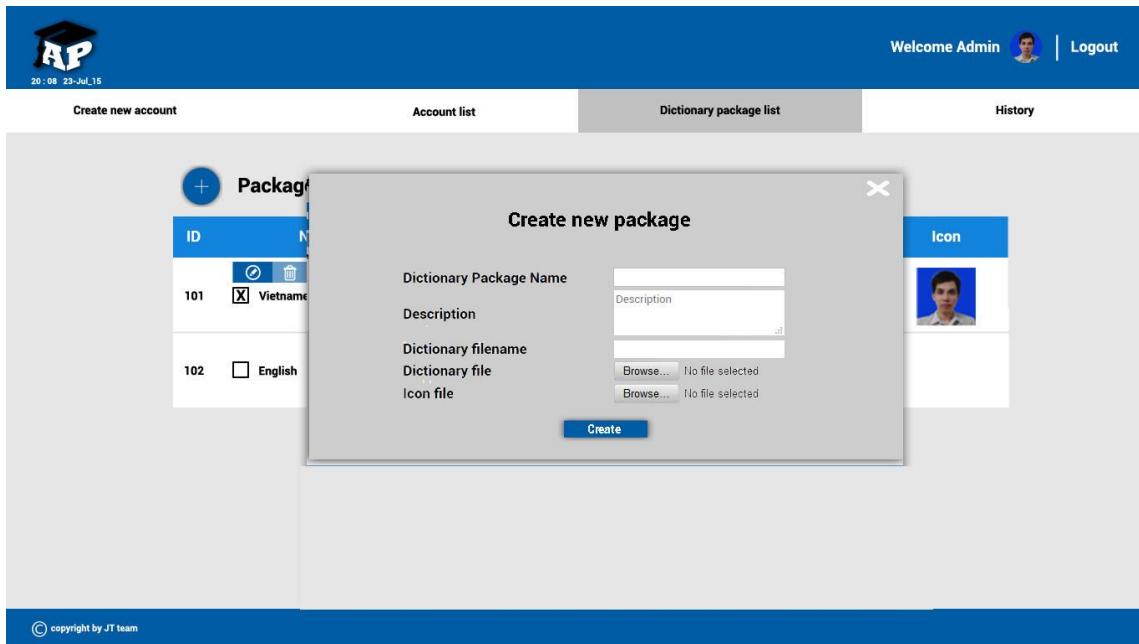
### 3.3.2.4.2. Create new package button

When user click on “Create new package” button , the “Create new package” form will be displayed .



### 3.3.2.4.3. Create new package form

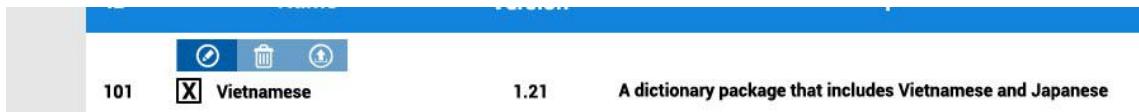
When user click on “Create new package” button , the “Create new package” form will be displayed. User enter the package information , click “Create” button to create new package.



No	Elements	Type	Description
1	Dictionary Package name	TextField	Where user enter the name of package.
2	Description	TextArea	Where user describe new package.
3	Dictionary filename	TextField	Where user enter the filename.
4	Dictionary file	File Browse	Where user select dictionary file to submit.
5	Icon file	File Browse	Where user select dictionary icon.
6	Create button	Button	Click on to create new package.

#### 3.3.2.4.4. Tool box

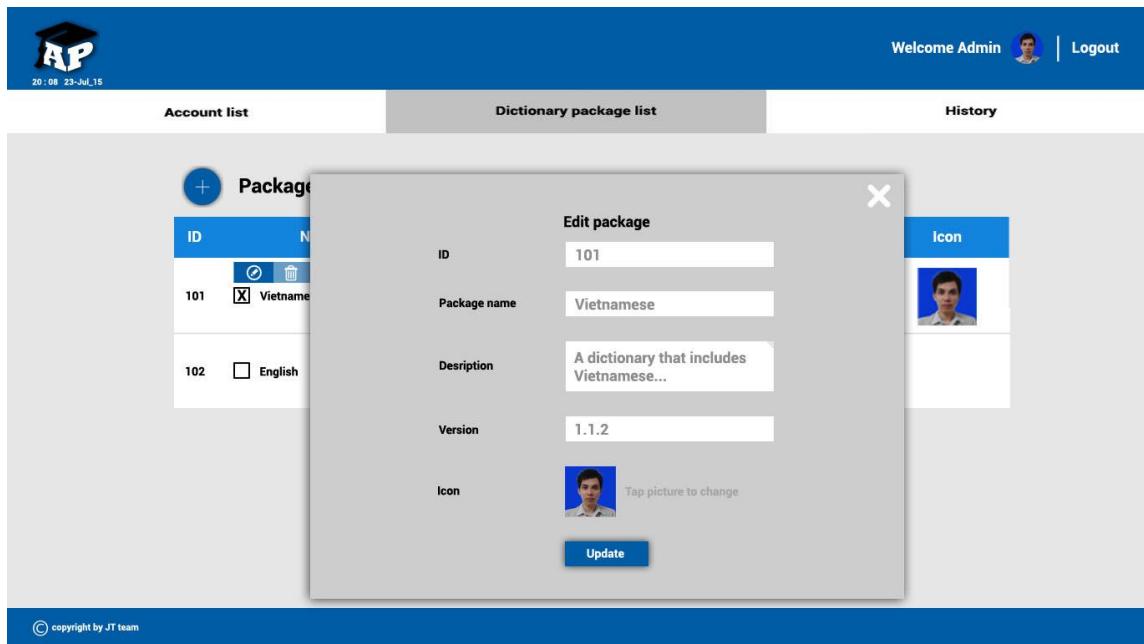
When user selected an package , a Tool box will be displayed .It contain : “Edit package” button , “Delete package” button , “Download package” button.



- Edit package button , link to “Edit package” form.
- Delete package button , delete package immediately.The package list will be updated immediately.
- Download package button , download packge to your computer.

#### 3.3.2.4.5. Edit package form

In package list page , after selecting a package , user click on “Edit package” button in the Tool box, “Edit package” form will be displayed. User enters new information to update package.



No	Elements	Type	Description
1	ID	TextField	Where displays selected package's ID.
2	Package name	TextField	Where displays the name of package.
3	Description	TextArea	Where displays sort description of selected package.
4	Version	TextField	Where displays the version of package.
5	Icon	Image	Where display package's icon.
6	Update button	Button	Clicks on this button to update package.

### 3.3.2.5. Package Detail page

#### 3.3.2.5.1. Package detail

In the package list page , user click on ID or package name of the package to selected a package . The package detail page will be displayed. The package detail page displays all the sub-dictionary that selected package contains.

The screenshot shows a web application interface for managing dictionary packages. At the top, there's a blue header bar with the logo 'AP' and the date '20.08.23-Jul\_15'. On the right side of the header, there are links for 'Welcome Admin' (with a user icon) and 'Logout'. Below the header, there are three tabs: 'Account list', 'Dictionary package list' (which is currently selected), and 'History'. The main content area is titled 'Package detail' and contains a table with two rows of sub-dictionary information. The columns are labeled 'ID', 'Name', 'Version', 'Description', 'Size', and 'File name'. The first row represents 'JV Dictionary' (ID 1011, Version 1.0, Description: 'A dictionary that translates from Japanese to Vietnamese', Size: 19.5, File name: 'jdict.db'). The second row represents 'Kanji' (ID 1012, Version 1.2, Description: 'A dictionary that translates from Kanji to Han Tu', Size: 0.05, File name: 'NhatHan.db'). Each row has a small circular icon with a plus sign and a delete icon.

No	Column	Description
1	ID	Displays sub-dictionary's ID.
2	Name	Displays the sub-dictionary name.
3	Version	Displays the sub-dictionary version.
4	Description	Display sort description.
5	Size	Displays the size of sub-dictionary.
6	File name	Displays the file name of sub-dictionary.

### 3.3.2.5.2. Create new sub-dictionary button

In the package detail page , user click on “Create new sub-dictionary” button to create new sub-dictionary.



### 3.3.2.5.3. Create new sub-dictionary form

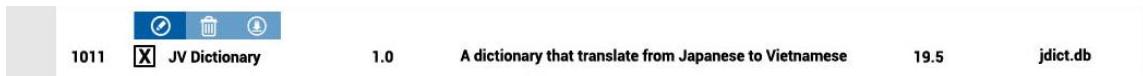
When user click on “Create new sub-dictionary” button , the create new form will be displayed . User enter the sub-dictionary information to create new subdictionary.

The screenshot shows a web application interface for managing dictionary packages. At the top, there's a blue header bar with the logo 'AP' and the date '20.08.23-Jul\_15'. On the right side of the header, there are links for 'Welcome Admin' (with a user icon) and 'Logout'. Below the header, there are three tabs: 'Account list', 'Dictionary package list' (which is currently selected), and 'History'. The main content area displays a table of dictionary packages. One row is selected, showing details: ID 1011, Name 'JV Dictionary', Type 'Dictionary Package', Version 1.0, Description 'A dictionary that translate from Japanese to Vietnamese', and Size 19.5. To the right of this row, a modal window titled 'Create new sub-dictionary' is open. It has fields for 'Name' (containing 'NhatHan'), 'Description' (containing '101'), and 'Dictionary Package' (containing '101'). A 'Create' button is at the bottom of the modal. A sidebar on the right lists file names: 'jdict.db' and 'NhatHan.db'. At the bottom of the page, there's a copyright notice '(C) copyright by JT team'.

No	Elements	Type	Description
1	Name	TextField	Where user enter the name of sub-dictionary.
2	Description	TextArea	Where user describe new sub-dictionary.
3	Dictionary package	TextField	Where display ID of the selected package.
4	Create button	Button	Click on to create new sub-dictionary.

#### 3.3.2.5.4. Tool box

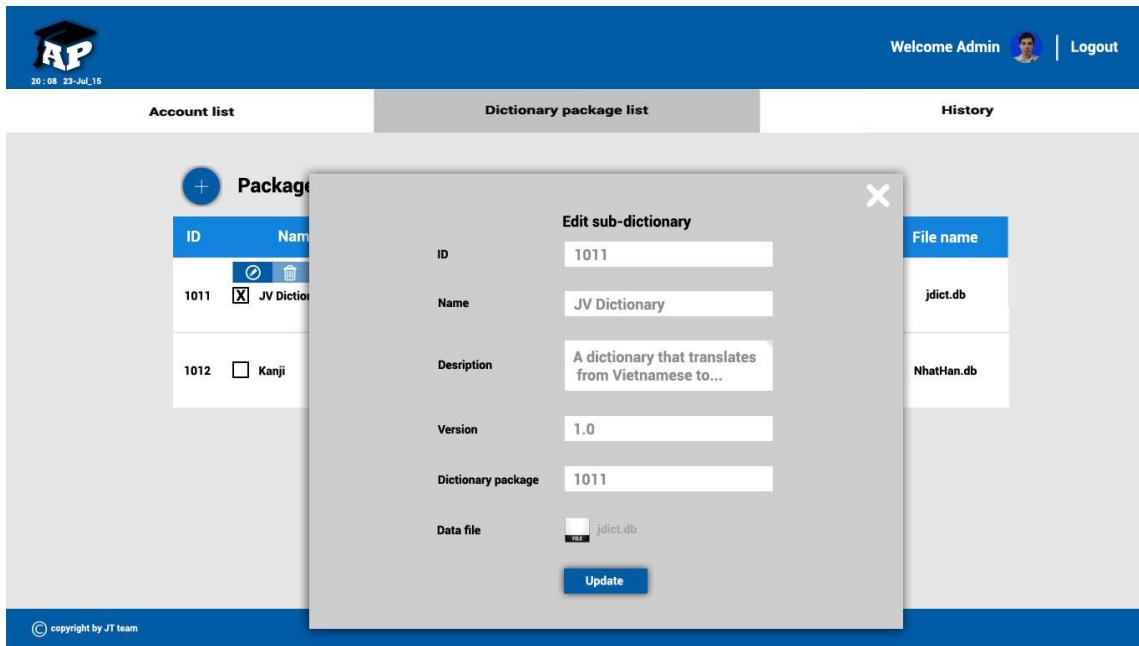
When user selected a sub-dictionary, a Tool box will be displayed .It contain : “Edit sub-dictionary” button , “Delete sub-dictionary” button , “Download sub-dictionary” button.



- Edit sub-dictionary button , link to “Edit sub-dictionary” form.
- Delete sub-dictionary button , delete sub-dictionary immediately.The sub-dictionary list will be updated immediately.
- Download sub-dictionary button , download sub-dictionary file to your computer.

#### 3.3.2.5.5. Edit sub-dictionary form

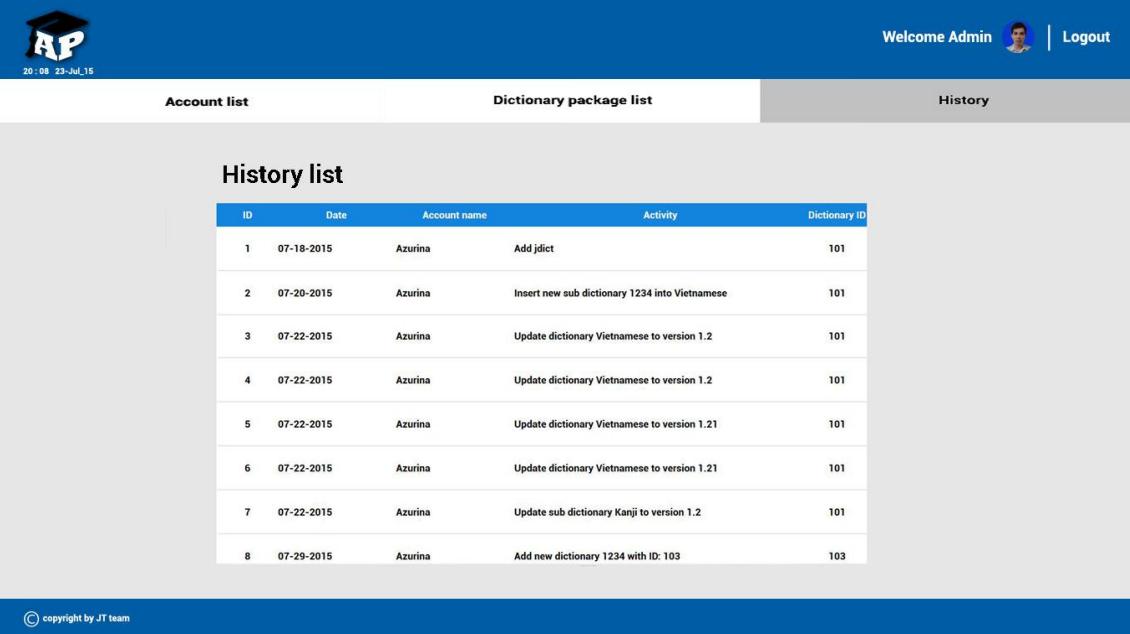
In package detail page , after selecting a sub-dictionary , user click on “Edit sub-dictionary” button in the Tool box, “Edit sub-dictionary” form will be displayed. User enters new information to update sub-dictionary.



No	Elements	Type	Description
1	ID	TextField	Where displays the ID of selected sub-dictionary.
2	Name	TextField	Where displays the name of sub-dictionary.
3	Description	TextArea	Where displays sort description of selected sub-dictionary.
4	Version	TextField	Where displays the version of sub-dictionary.
5	Dictionary package	TextField	Where display the package that contains the selected sub-dictionary.
6	Data file	File Browse	Where display the current file of selected sub-dictionary. User can change new file.
7	Update button	Button	Clicks on to update sub-dictionary.

### 3.3.2.6. History page

User click on "History" tab to link to history page. This page will display the list of actions as a table.



The screenshot shows a web-based application with a dark blue header bar. On the left is a logo with 'AP' and a small house icon. In the center is the text '20 : 08 23-Jul\_15'. On the right, there are links 'Welcome Admin' with a user icon, 'Logout', and a vertical separator line.

Below the header, there are three tabs: 'Account list', 'Dictionary package list', and 'History'. The 'History' tab is selected, indicated by a grey background.

### History list

ID	Date	Account name	Activity	Dictionary ID
1	07-18-2015	Azurina	Add jdict	101
2	07-20-2015	Azurina	Insert new sub dictionary 1234 into Vietnamese	101
3	07-22-2015	Azurina	Update dictionary Vietnamese to version 1.2	101
4	07-22-2015	Azurina	Update dictionary Vietnamese to version 1.2	101
5	07-22-2015	Azurina	Update dictionary Vietnamese to version 1.21	101
6	07-22-2015	Azurina	Update dictionary Vietnamese to version 1.21	101
7	07-22-2015	Azurina	Update sub dictionary Kanji to version 1.2	101
8	07-29-2015	Azurina	Add new dictionary 1234 with ID: 103	103

At the bottom left of the main content area, there is a small copyright notice: '© copyright by JT team'.

No	Column	Description
1	ID	The ID of action.
2	Date	The date that action happened in.
3	Account name	The account name that do the action.
4	Activity	The name of action.
5	Dictionary ID	The ID of dictionary that related with action.

## V. EVALUATION

### 1. TEST PLAN

#### 1.1. Introduction

##### 1.1.1. Purpose

This document describes the plan for testing the developed software system against the software requirements as defined in the Software Requirements Specification and Detailed Document. The purpose of these tests is to make sure that the software system developed during the Japanese Translator project complies with the requirements. The planning of these test will be done in the Planning phase, they will be designed in the Design phase and finally must be executed in the Coding phase and Testing phase of each cycle in the project.

##### 1.1.2. Background information

Japan is one of the most advanced countries in the world. Besides, Japanese culture is spreading very quickly through manga, anime,... Therefore, there are more and more people study Japanese. However, for most of them, reading Japanese is a real challenge because of a lots Chinese-like characters called Kanji. So "Anki Pan" project was kicked off to serve purpose of learning Japanese easier and simpler.

This project will develop an Android application that has following main function:

- Capture image
- Detect Japanese characters in image captured by users
- Select text in captured image
- Search for meaning of Japanese words and Kanji in dictionary
- Download dictionary: allow user to download dictionary package, each package include a Japanese dictionary and a Kanji dictionary.
- Save words to history
- Add to favorite list

A server will also be developed to support users download dictionary database. The server has below main functions:

- Login as admin or root admin
- Logout
- Manage user' profile (add, delete, edit information)
- Upload dictionary packages, each package includes 2 kind of dictionaries: Japanese dictionary and Kanji dictionary
- Manage dictionary packages (edit, delete)

Project team will develop a library for image processing (AKImageProcessingLib), using OpenCV open library. AKImageProcessingLib is created to:

- Extract all text lines from image captured by user
- Convert image to threshold image

Tesseract open source OCR is used for Japanese recognition. Threshold image converted by AKImageProcessingLib will be passed to Tesseract to detect Japanese characters.

### **1.1.3. Scope of testing**

The scope of this test will be limited to testing the application on Android and testing of server.

#### **1.1.3.1. Target of Test**

Functional items and Non-functional items will be verified and passed by AKP development team, via test stages, including the requirements of the following functions of Android application:

- Capture image
- Detect Japanese characters in image captured by users
- Select text in captured image
- Search for meaning of Japanese words and Kanji in dictionary
- Download dictionary: download dictionary package, each package include a Japanese dictionary and a Kanji dictionary.
- Save words to history
- Add to favorite list

And the following functions of server:

- Login as admin or root admin
- Logout
- Manage user' profile (add, delete, edit information)
- Upload dictionary packages, each package includes 2 kind of dictionaries: Japanese dictionary and Kanji dictionary
- Manage dictionary packages (edit, delete)

#### **1.1.3.2. Test stage**

No	Test Stages	Description
1	Unit Test	Unit Test will be performed by AKP development team
2	Integration Test	Integration Test will be performed by AKP tester. After the Unit Test is finished testers will execute the UT Gate

		<p>based on the UT Gate checklist for each function. Integration Test will only start if the result of UT Gate is passed.</p> <p>This test stage focuses on specific areas of use cases when all requirements are completed, integration test should be performed to ensure all components incorporate well.</p>
3	System Test	System test will be execute by AKP testers. Testers will perform complete, end-to-end system testing staged in pre-production environment to validate that functions and system interfaces perform properly in production environment.
4	Acceptance Test	Acceptance Test will be conducted by users. AKP team's will fix bugs reported by user in acceptance test phrase.

#### 1.1.4. Constraints

- Working processing must be report Personal weekly report to PM and Supervisor.
- Deadline for testing only can be met if development progress is on time.
- Test execution can be performed when system passes Unit Test Inspection.
- At least one round of testing must be performed for requirements.

#### 1.1.5. Risk List

No	Description	Source	Probability	Exposure	Trigger
1	Test phrases is not completed due to testers' lack of experience	Human	Medium		

#### 1.1.6. Training Needs

No	Training items	Description
1	Testing process	<p>Learn &amp; study by MaiNT: Understanding: Unit Test, System Test, Acceptance Test, Integration Test Learn from web: testing.vn</p>
2	Test tools	Learn & study by BinhLT

		Understanding: test tool for server
3	Test model	Learn & study by developer: Understanding V-model

## 1.2. Requirement For Test

### 1.2.1. Test items

Developer will be performed unit test to ensure that all functions are working as expected. Tester will be under taking the responsibility of testing and validating the operation of the part that developed. The functions developed require completing design and execution of function to ensure proper coverage.

The listing below identifies those items that have been identified as targets for testing. This list represents what will be tested.

#### 1.2.1.1. Functional Items

##### ❖ Application

1. Main menu 2. Capture image 3. Result <ul style="list-style-type: none"> <li>• Show text line</li> <li>• Select text from image</li> <li>• Zoom in selecting area</li> </ul> 4. Dictionary <ul style="list-style-type: none"> <li>• Search Japanese word</li> <li>• Suggest word</li> <li>• Search Kanji</li> <li>• Add to favorite list</li> <li>• Remove from favorite list</li> </ul>	5. History <ul style="list-style-type: none"> <li>• Show searched words</li> <li>• Show favorite words (bookmarked words)</li> <li>• Show words' information</li> <li>• Add to favorite list</li> <li>• Remove from favorite list</li> <li>• Delete a word</li> <li>• Delete all history</li> </ul> 6. Setting <ul style="list-style-type: none"> <li>• Select dictionary package</li> <li>• Download dictionary package</li> <li>• Delete dictionary package</li> <li>• Show dictionary package's description</li> </ul> 7. Exit
--	---

##### ❖ Server

Root admin & Admin <ul style="list-style-type: none"> <li>1. Login</li> <li>2. Logout</li> <li>3. Manage profile <ul style="list-style-type: none"> <li>• Edit profile (user name, avatar,</li> </ul> </li> </ul>	Root admin <ul style="list-style-type: none"> <li>1. Manage account: <ul style="list-style-type: none"> <li>• Add new account</li> <li>• Delete account</li> </ul> </li> </ul>
---	--

email) <ul style="list-style-type: none"> <li>• Change password</li> </ul> 4. Upload dictionary package 5. Edit dictionary package 6. Delete dictionary package	
--	--

### 1.2.1.2. Non functional Items

1. User interface is clear and easy to understand. User need no more than 10 minutes to learn how to use the application
2. Time for processing image is less than 10s

### 1.2.2. Acceptance test criteria

(Reference to FSoft norm 2013 for development project)

- The criteria for Test team to accept source code after Unit test of Development team:
  - Number of UTC/KLOC : 80 - 90 Case/1KLOC
  - Weight defects/KLOC : 8-9 bug/1KLOC
  - Part coverage : 100%
- The criteria for accept the result of System test:
  - Number of UTC/KLOC : 34 - 46 Case/1KLOC
  - Weight defect/KLOC : 2 - 4 bug/1KLOC
  - Part coverage : 100%
- The criteria for accept the result of Acceptance test:
  - Number of UTC/KLOC : 1 - 2 Case/1KLOC
  - Weight defect/KLOC : 0 - 1 bug/1KLOC
  - Part coverage : 100%

## 1.3. Test strategy

### 1.3.1. Test type

#### 1.3.1.1. Function Testing

Unit testing searches for defects in each function, and verifies the function of Anki Pan application and server.

Application and tests whether its functions are implemented properly as required or not.

Test Objective	Ensure proper target-of-test functionality, including navigation, data entry, processing, and retrieval.
----------------	--

<b>Technique</b>	Execute each use case, use-case flow, or function, using valid and invalid data, to verify the following: <ul style="list-style-type: none"> <li>- The expected results occur when valid data is used.</li> <li>- The appropriate error or warning messages are displayed when invalid data is used.</li> </ul>
<b>Completion criteria</b>	All planned tests have been executed.
<b>Special considerations</b>	Identify or describe those items or issues (internal or external) that impact the implementation and execution of function test.

### 1.3.1.2. User Interface Testing

Integration testing tests interfaces between components (Mobile screen), interactions with different parts of a system, and interfaces between systems.

<b>Test Objective</b>	Verify the following: Navigation through the target-of-test properly reflects business functions and requirements, including window-to-window, field-to-field, and use of access methods (tab keys, mouse movements, accelerator keys)
<b>Technique</b>	Create or modify tests for each window (server) and each screen (application) to verify proper navigation and object states for each application window and objects
<b>Completion criteria</b>	Each window(server) and each screen(application) successfully verified to remain consistent with benchmark version or within acceptable standard
<b>Special considerations</b>	Not all properties for custom and third party objects can be accessed.

### 1.3.1.3. Data and Database Integrity Testing

Data and Database Integrity Testing test integrity of data, accuracy of data and save data to the database.

<b>Test Objective</b>	Ensure database access methods and processes function properly and without data corruption.
<b>Technique</b>	<ul style="list-style-type: none"> <li>- Invoke each database access method and process, seeding each with valid and invalid data or requests for data.</li> <li>- Inspect the database to ensure the data has been populated as intended, all database events occurred</li> </ul>

	properly, or review the returned data to ensure that the correct data was retrieved for the correct reasons
<b>Completion criteria</b>	All database access methods and processes function as designed and without any data corruption.
<b>Special considerations</b>	<ul style="list-style-type: none"> <li>- Testing may require a DBMS development environment or drivers to enter or modify data directly in the databases.</li> <li>- Processes should be invoked manually.</li> <li>- Small or minimally sized databases (limited number of records) should be used to increase the visibility of any non-acceptable events.</li> </ul>

#### 1.3.1.4. Regression Testing

Regression testing is to validate modified parts of the software, to make sure that the modification does not cause errors in other parts.

<b>Test Objective</b>	Regression testing is to validate modified parts of the software, to make sure that the modification does not cause errors in other parts.
<b>Technique</b>	<ul style="list-style-type: none"> <li>- Reuse the set of test cases from an existing test suite to test a modified module.</li> <li>- Construct a program-analysis infrastructure. We are building an extensible infrastructure to implement and evaluate a program-analysis. Basing on the analysis result, we identify scope of regression test.</li> </ul>
<b>Completion criteria</b>	All selected test cases(cases can cause bugs when changed requirement) are performed and passed.
<b>Special considerations</b>	N/A

#### 1.3.2 Test stages

<b>Types of Test</b>	<b>Stages of Test</b>			
	<b>Unit</b>	<b>Integration</b>	<b>System</b>	<b>Acceptance</b>
Function Test	X	X	X	
User Interface test		X	X	

Database Integrity Testing		X	X	
Regression test	X	X	X	

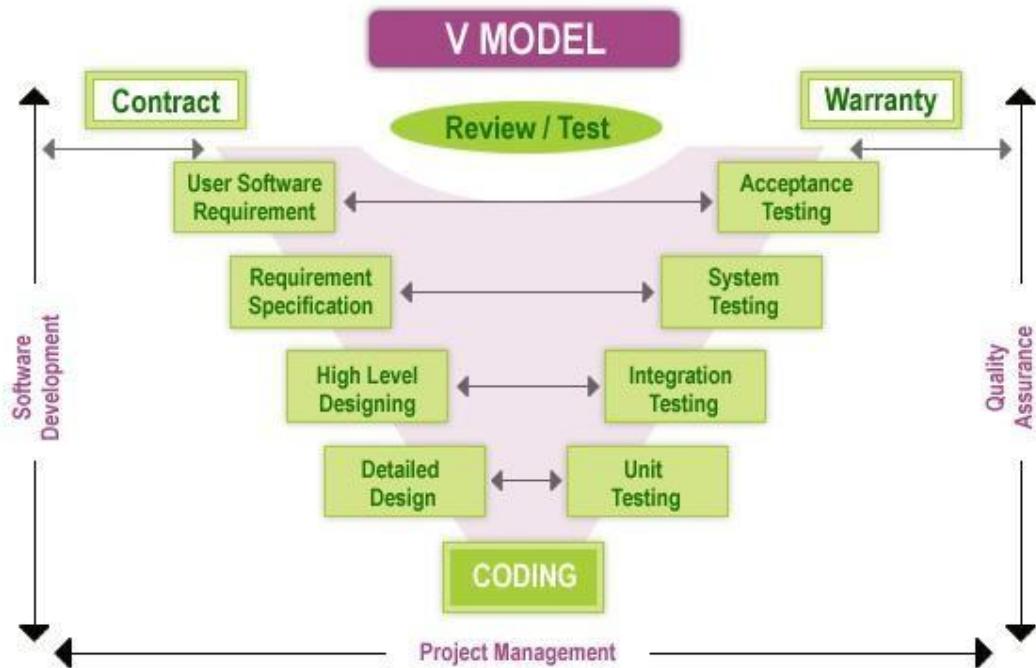


Figure 125: V-Model

## 1.4. Resource

### 1.4.1 Human resource

Doer	Roles	Specific Responsibilities / Comments
MaiNT	Test leader	Manage Test resource and assign test tasks Create TP ST IT TC Review TC Execute test
NhungBH	PM + Developer	Review test plan Review test result Create UT & execute test Report test result
ThaiVH	Technical leader	Create UT & execute test Report test result
NghiaPT	Developer	Create UT & execute test Report test result

HoangND	Developer	Create UT & execute test Report test result
BinhLT	Designer + Tester	Create integration test cases Execute test cases Report test result

#### 1.4.2 Environment

Hardware	Version
LG smart phone	Android ver4.4.2
Sony smart phone	Android 5.0.2
Laptop	Window 7

Hardware	Version
OpenCV Manager	2.18
Google Chrome	44.0.2403.107
Fire fox	39.0

#### 1.5. Test Milestones

Milestone Task	Effort (pd)	Start Date	End Date
<b>❖ Increment 1</b>			
Create test plan		08/06/2015	10/06/2015
Submit test plan version 1.0		11/06/2015	11/06/2015
<b>❖ Increment 2</b>			
Create ST test cases for Dictionary		24/06/2015	25/06/2015
Create ST test cases for History		26/06/2015	29/06/2015
Create & execute UT test case for image processing library		9/7/2015	9/07/2015
Create & execute UT test case for server		10/07/2015	10/07/2015
Create & execute UT test case for Dictionary		10/07/2015	10/07/2015
Create & execute UT test case for Scan		11/07/2015	11/07/2015

Report UT result		11/07/2015	11/07/2015
<b>❖ Increment 3</b>			
Review & update & submit test plan ver2.0		14/07/2015	14/07/2015
Create ST test cases for server		20/07/2015	21/07/2015
Execute & report ST for server		22/07/2015	23/07/2015
Create & execute UT Result screen's functions		24/07/2015	24/07/2015
Create & execute UT History functions		31/07/2015	31/07/2015
Create & execute UT Dictionary functions		31/07/2015	31/07/2015
Create & execute UT functions of cropping text area		30/07/2015	30/07/2015
Create & execute UT Setting functions		1/8/2015	1/8/2015
Create & execute IT		30/07/2015	1/08/2015
Report UT & IT result		2/08/2015	2/08/2015
<b>❖ Increment 4</b>			
Execute & report ST test cases for Application		5/08/2015	7/08/2015
Submit final report of testing		7/08/2015	7/08/2015

## 1.6. Deliverables

No	Deliverables	Delivered Date	Delivered by	Delivered to
<b>❖ Increment 1</b>				
1	Test plan ver1.0	11/06/2015	MaiNT	NhungBH
<b>❖ Increment 2</b>				
2	UT test cases and result	11/07/2015	ThaiVH	NhungBH
3	ST test cases ver1.0	29/06/2015	MaiNT	NhungBH
<b>❖ Increment 3</b>				
4	Test plan ver2.0	4/07/2015	MaiNT	NhungBH
5	UT test cases and result	2/08/2015	ThaiVH	NhungBH
6	ST test case ver2.0	23/07/2015	MaiNT	NhungBH
7	ST result of server	23/07/2015	MaiNT	NhungBH

8	IT test cases ver1.0	2/08/2015	BinhLT	NhungBH
9	IT result	2/08/2015	BinhLT	NhungBH
<b>❖ Increment 4</b>				
9	ST result of application	7/08/2015	MaiNT	NhungBH
10	Final report of testing	7/08/2015	NhungBH	Supervisor

## 2. TEST REPORT

### 2.1. Integration test

#### INTEGRATION TEST REPORT

Project Name	AnKiPan	Creator	Ngo Thi Mai
Project Code	AKP	Reviewer/Approver	Bui Hong Nhung
Document Code	AKP_Integration test report_v1.0	Issue Date	2/8/15
Notes	"System test round 1 includes modules: Application: Web service: 1. Main Menu 1. Login 2. Setting 2. Update profile 3. History 3. Manage accounts 4. Scan 4. Manage dictionaries 5. Dictionary 5. History 6. Scan result 6. Logout"		

No	Module code	Pass	Fail	Untested	N/A	Number of test cases
1	Scan+Scan result	3	0	0		3
2	Text recognition +History	5	0	0		5
3	Dictionary Screen+History	4	0	0		4
4	History	7				7
5	Download + Dictionary	2	0	0		2
6	Create account+Add account	2	0	0		2
7	Account list+Delete account	2	0	0		2
8	Dictionary Package list+History	5	0	0		5
<b>Sub total</b>		<b>30</b>				<b>30</b>
Test coverage		<b>100.00 %</b>				
Test successful coverage		<b>100.00 %</b>				

Figure 126: Report of Integration test

### 2.2. System test

#### ❖ System test round 1

## SYSTEM TEST REPORT

<b>Project Name</b>	AnKiPan	<b>Creator</b>	Ngo Thi Mai
<b>Project Code</b>	AKP	<b>Reviewer/Approver</b>	Bui Hong Nhung
<b>Document Code</b>	AKP_SystemTestReport_v1.0	<b>Issue Date</b>	2/8/15
<b>Notes</b>			
System test round 1 includes modules: Application: Web service: 1. Main Menu 1. Login 2. Setting 2. Update profile 3. History 3. Manage accounts 4. Scan 4. Manage dictionaries 5. Dictionary 5. History 6. Scan result 6. Logout			

No	Module code	Pass	Fail	Untested	N/A	Number of test cases
1	MainMenu	30	0	0	0	30
2	Setting	7	0	42	0	49
3	History	9	0	17	0	26
4	Scan	24	1	0	0	25
5	Dictionary	42	1	0	0	43
6	ScanResult	23	4	13	0	40
7	Login	8	0	0	0	8
8	Update profile	10	0	5	0	15
9	Manage accounts	35	3	0	0	38
10	Manage dictionaries	24	0	0	0	24
11	History	1	0	0	0	1
12	Logout	2	0	0	0	2
<b>Sub total</b>		215	6	72	0	301
Test coverage				73.42%		
Test successful coverage				71.43%		

*Figure 127: Report of system test round 1*

### ❖ System test round 2

## SYSTEM TEST REPORT

<b>Project Name</b>	AnKiPan	<b>Creator</b>	Ngo Thi Mai
<b>Project Code</b>	AKP	<b>Reviewer/Approver</b>	Bui Hong Nhung
<b>Document Code</b>	AKP_SystemTestReport_v1.0	<b>Issue Date</b>	2/8/15
<b>Notes</b>			
System test round 1 includes modules: Application: Web service: 1. Main Menu 1. Login 2. Setting 2. Create New Account 3. History 3. Account List 4. Scan 4. Dictionary package list 5. Dictionary 5. Update profile 6. Scan result 6. History List 7. Logout			

No	Module code	Pass	Fail	Untested	N/A	Number of test cases
1	MainMenu	30	0	0	0	30
2	Setting	49	0	0	0	49
3	History	26	0	0	0	26
4	Scan	26	0	0	0	26
5	Dictionary	41	3	0	0	44
6	ScanResult	41	0	0	0	41
7	Login	8	0	0	0	8
8	Create New Account	18	3	0	0	21
9	Account List	18	0	0	0	18
10	Dictionary package list	24	0	0	0	24
11	Update profile	10	0	5	0	15
12	History List	1	0	0	0	1
13	Logout	2	0	0	0	2
<b>Sub total</b>		294	6	5	0	305
Test coverage				98.36%		
Test successful coverage				96.39%		

*Figure 128: Report of system test round 1*

## 3. CHECK LIST

Below is a sample of our check list:

**Test Plan Review Checklist**

Project Code: AnkiPan Application  
 Version of the work product: 1.0  
 Reviewer(s): MaiNT  
 Review date: :<08/08/2015>

Question	Yes	No	N/A	Note	Priority
<b>DOCUMENT CONTROL</b>					
Verify whether document control procedures have been followed by checking for the following:	x				Mandatory
Does the title page contain the document name, version number, release date and issued date.?	x				Optional
Does the header and footer correctly specify the name, and version of the document?	x				Optional
Does the page – numbering scheme indicate the total number of pages in the document?	x				Optional
Is the history traceable?	x				Optional
Does it include list of reference documents?	x				Optional
Are all documents checked Spelling and Grammar in MS Word or similar tools?	x				Optional
<b>TEST PLAN CHECKLIST</b>					
Have products that are to be tested been identified?	x				Mandatory
Is the overall defect expectation or defect expectation rate stated in the plan ?	x				Mandatory
Does the plan provide clearly scope for the following?	x				Mandatory
Unit testing	x				Optional
Integration testing	x				Optional
System testing	x				Optional
Acceptance testing	x				Optional
Does the plan reflect the requirements as specified in the SRS?	x				Mandatory
Have the test types addressed the following?					Optional
Function testing	x				Optional
User interface testing	x				Optional
Data and Database Integrity Testing	x				Optional
Performance testing	x				Optional
Security and Access Control Testing		x			Optional
Portability		x			Optional
Have acceptance criteria for requirements been identified?	x				Mandatory
Do the types of test reflect all requests specified in SRS?	x				Mandatory
Is it a maintenance project? If yes, Has the regression test been identified?	x				Mandatory
Trigger for Regression test:	x				Mandatory
Period for Regression test	x				Mandatory
Scope of Regression test	x				Mandatory
Does the description of each type of test include the following?					Optional
Test objectives	x				Mandatory
Techniques	x				Mandatory
Completion criteria		x			Optional
Special consideration		x			Optional
Have the testing environment, testing tools and testing software and hardware been described?	x				Mandatory
Are the Pass/Fail criteria of tested products defined in Special consideration correspondent with requirements in SRS?	x				Optional
Are Doer and Responsibilities identified correctly?	x				Optional
Have the test deliverables been defined?	x				Mandatory
Does the testing schedule correspond to the development schedule?	x				Mandatory
Have staffing and training needs been specified?	x				Mandatory
Have the risks of testing been identified?	x				Mandatory
<Add more rows if needed>					

\* Comments

Figure 129: Check list for test plan

## VI. IMPLEMENTATION

### 1. DEVELOPMENT

Application and server were developed through 4 Increment as specified in Project Plan. In the end of each Increment, we held a meeting to evaluate the product and defined goals for next Increment.

We have meeting with Supervisor every week to report project progress. All unsolved issues was informed to the Supervisor to get his instruction.

All versions of source code are stored on Google drive.

### 2. PROGRESS REPORT

Project progress reports are delivered to Supervisor every week. Below is the sample of our progress report:



## PROJECT PROGRESS REPORT 1

Project Name	Anki Pan	Project code	AKP
Creator	Bui Hong Nhung	Project Manager	Bui Hong Nhung
Deliver date	22/05/2015	Receiver	Mr. Tran Binh Duong
Report period	15/05/2015 – 21/08/2015		

## Progress description

Item	Information	Note
Project start-date	4/05/2015	
Estimated end-date	15/08/2015	
Team size	6 members	Bui Hong Nhung Vu Hong Thai Pham Trong Nghia Nguyen Duc Huy Hoang Le Thanh Binh Ngo Thi Mai
Estimated effort	2410 man-hours	
Effort in this process	273 man-hours	
Effort spent until now	553 man-hours	
Total effort left	1857 man-hours	

## Task status (match / missed)

Task	Detail	Members in charge	Deliverables	Status	Schedule update
Create Introduction document	Conducting survey about - Demand of using smartphone for reading and translating Japanese. - Desired functions of the app. Search information about number of people studying Japanese in Vietnam	NghiaPT MaiNT NhungBH	Report 1: Introduction document	Done	No
Develop Image	- Study openCV library - Research on image processing	HoangNDH	image1.jpg image2.jpg	In process	Continue as

Figure 130: Sample of progress report (English version)



## プロジェクト進捗報告書 1

プロジェクト名	Anki Pan	プロジェクトコード	AKP
作成者	Bui Hong Nhungh	プロジェクトリーダー	Bui Hong Nhungh
日付	22/05/2015	受領者	Mr. Tran Binh Duong
報告期間	15/05/2015 - 21/08/2015		

## 進捗詳細

項目	情報	備考
プロジェクト開始日	4/05/2015	
終了予定日	15/08/2015	
チームサイズ	6人	Bui Hong Nhungh Vu Hong Thai Pham Trong Nghia Nguyen Duc Huy Hoang Le Thanh Binh Ngo Thi Mai
工数合計（予定）	2410	
工数合計（実際）	273 man-hours	
現在までの消化工数	553 man-hours	
残工数合計	1857 man-hours	

## タスク実行の状況（進捗度、遅延など）

タスク	詳細	責任者	納品物	ステータス	スケジュール変更
紹介書作成	調査を行った - 日本語を読むかつ翻訳するためスマートフォンを使う要求 - 望むアプリ機能 日本語を勉強するベトナム人の人数を調べた。	NghiaPT MaiNT NhunghBH	レポート 1 : 紹介書	完成	なし

Figure 131: Sample of progress report (Japanese version)

## VII. INSTALLATION GUIDE AND USER MANUAL

### 1. INSTALLATION GUIDE

#### 1.1. English Version

This section contains the list of prerequisites that must be fulfilled before install.

- A smart phone running on Android ver 4.4.2 or higher
- 50MB free or higher

#### 1.1.2. Installation procedures

1. Connect to internet
2. Tap on **Google play** icon on screen
3. Search Anki Pan
4. Tap on **Install** button
5. Tap **Accept** button on confirmation dialog

#### 1.1.3. Uninstall procedures

1. Tap **Apps** in the upper right hand corner
2. Tap on **Setting**
3. Tap **Applications** to the left of the screen.
4. Tap **Manage applications** on the right side of the screen.
5. Tap on **Anki Pan**
6. Tap the **Uninstall** button.
7. To confirm you want to uninstall this app tap **OK**.
8. Click **OK** once finished.

### 1.2. Japanese Version

#### 1.1.1. 前提条件

このセクションにはアプリをインストールするための前提条件を表します。

- Android バージョン 4.4.2 以上で動くスマートフォン
- 50MB 以上のフリー の容量

#### 1.1.2. インストール手順

1. インターネットに接続する
2. 画面に現れるグーグルプレイアイコンに押す

3. Anki Pan を検索する
4. インストールボタンに押す
5. 確認ダイアログのはいボタンに押す

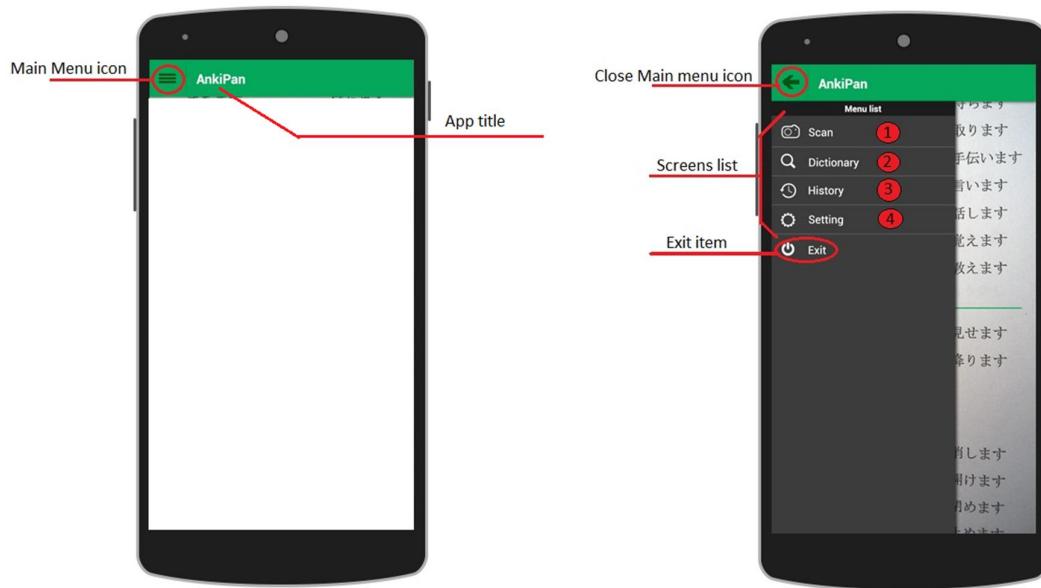
### 1.1.3. アンインストール手順

1. 右上コーナーにアプリを押す
2. 設定に押す
3. 画面の左に現れるアプリケーションに押す
4. 画面の右側にあるアプリ管理に押す
5. Anki Pan に押す
6. アンインストールボタンに押す
7. このアプリのアンインストールを確認するため OK に押す
8. アンインストールが終わったら OK に押す

## 2. USER MANUAL

This document guides you how to use Anki pan application. You will need only 10 minutes to learn.

### 2.1. Select a screen



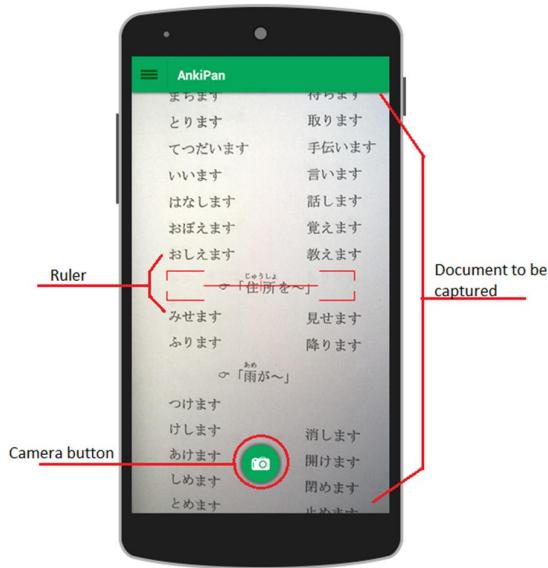
#### To select a screen

1. Tap on Main Menu icon on the upper left corner. The Main Menu will pop out listing all screen of the application.
2. Tap on the screen that you want to use.
  - ① Scan screen: used to capture and scan document
  - ② Dictionary screen: used to show scan result, and to search word in dictionary

- ③ History screen: used to show searched words
- ④ Setting screen: used to select dictionary package or download new dictionary package

If you want to quite the application, tap on Exit.

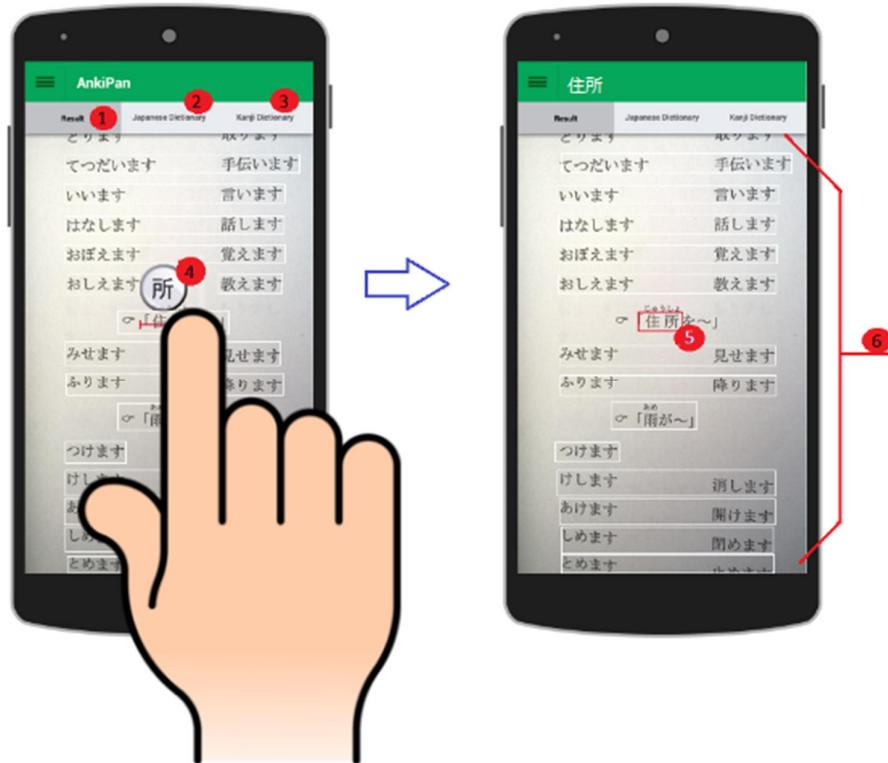
## 2.2. Scan document



### To scan a document

1. Open Scan screen (see 2.1.)
  2. Keep your smart phone above the document
  3. Adjust your smart phone so that the Ruler is parallel with the text line, and the text is at least equal the height of the Ruler
  4. Tap on Camera button
- Then you will see Result screen with your document scanned.

## 2.3. Select text on scanned document



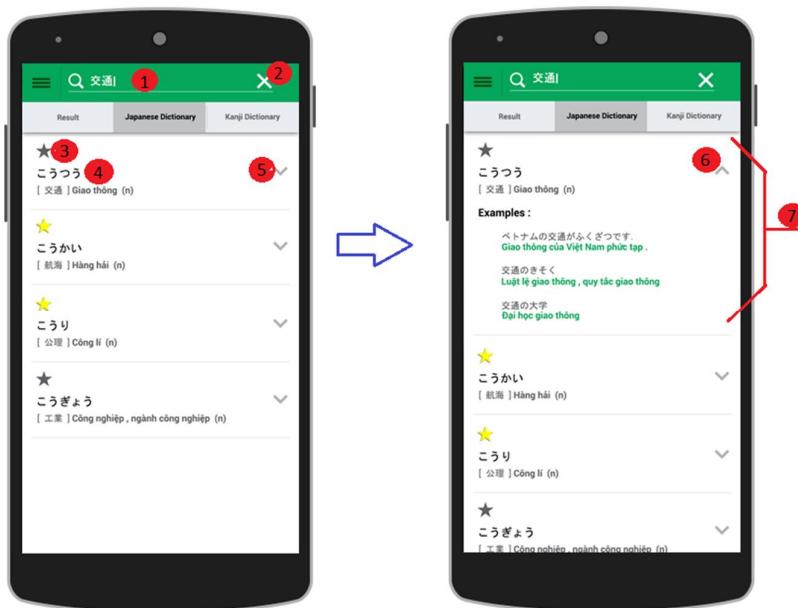
### Screen description

- ① Result tab: shows scanned document. Text lines are surrounded by white rectangles
- ② Japanese Dictionary tab
- ③ Kanji Dictionary tab
- ④ Magnifier: zoom in the text area that is being selected
- ⑤ Scanned document

### To select text on scanned document:

1. Put your finger on the first character of all characters you want to select
2. Swipe your finger to the last character of all character you want to select. While you are moving your finger, the magnifier will zoom in the area that your finger is on, so that you can see the text clear.
3. Lift up your finger. Your selected word will be surrounded by a red rectangle (⑤). Do not worry if you missed stroke of first or last character. The application will process to get all the characters that you might intended to select. The text then will be show in the text field on top of the screen.
4. Now tap on Japanese Dictionary ② or Kanji Dictionary ③ to look up the selected word.

## 2.4. Search word in Japanese dictionary



### To search word in Japanese dictionary

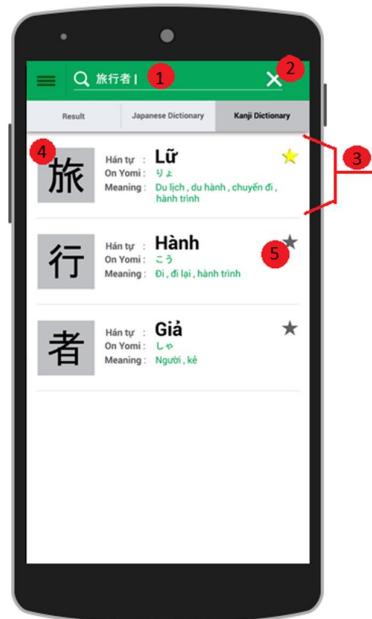
1. Tap on Japanese Dictionary tab in Dictionary screen
2. Type the word to the text field ①. If you want to clear all the text, tap on Clear icon ②

3. After you press Search key on keyboard, the words which have the first part the same as the word you typed will be list on the screen. Each word will be show with a Star icon ③ and the word itself ④
4. If you want to see more about the word, tap the extend icon ⑤. Other information of the word, for example meanings, word type,... will be show, as ⑦
5. If you want to see the short form, tap the close icon ⑥

#### **To add / remove word from favorite list**

Tap on the Star icon of the word to add / remove it from favorite list. If the star is yellow, it mean that the word is in favorite list. If you tap on a yellow star, the star turns to gray and the word will be removed from favorite list. If you tap on a gray star, the star turns to yellow, and the word will be added to favorite list.

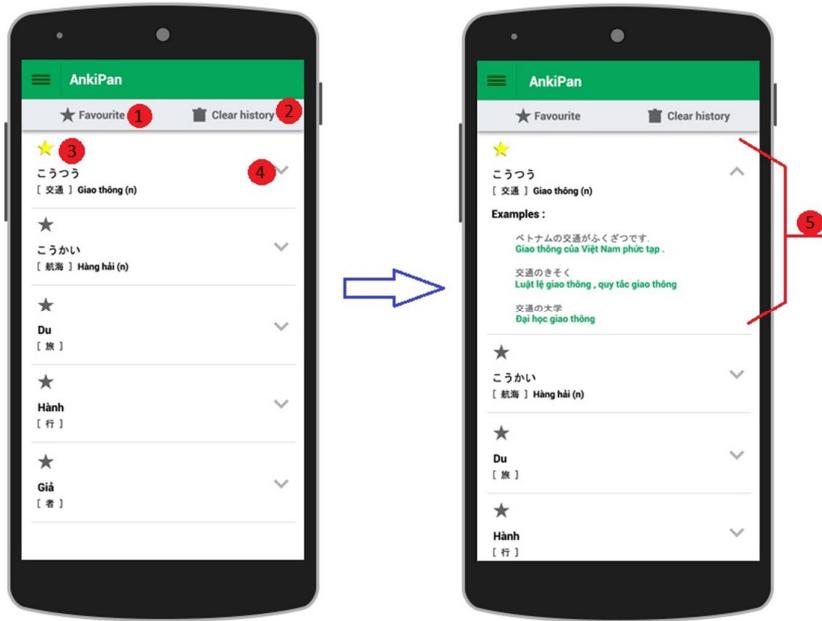
### **2.5. Search kanji in Kanji dictionary**



#### **To search kanji in Kanji dictionary**

1. Type words in the text field 1. If you want to clear all the text, tap Clear button ②
2. After you press the Search key on the keyboard, all kanji in you input will be show. You can see the kanji ④ and its On yomi, Kun yomi, meaning... as in ③
3. To add or remove the kanji from favorite list, tap the Star icon

## 2.6. View history



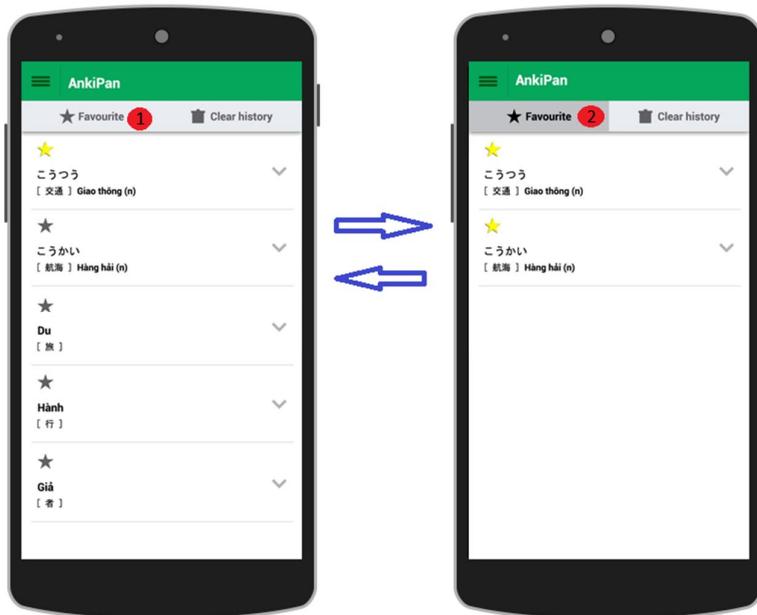
History screen include Favorite button ① and Clear history button ② and a list of word that you have searched

To view History screen, select History screen from Main Menu ( see 2.1)

To view full information of the word by tap on the Extend icon ④

Tap on Star icon ③ to add / remove a word from favorite

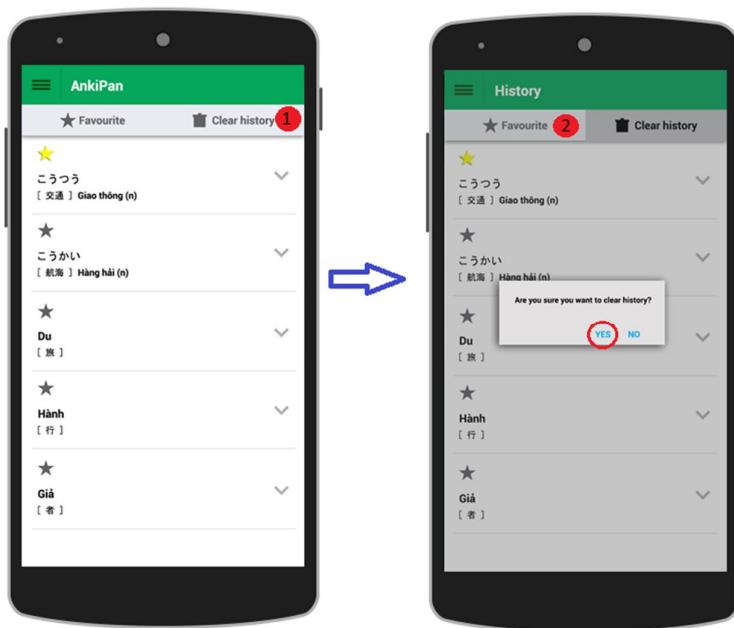
## 2.7. View all favorite words



### To view all favorite word

1. Select History screen from Main Menu (see 2.1)
2. Tap on Favorite button ①. Then all the words in favorite list will be shown on the screen.
3. If you want to back to History screen, tap the Favorite button again.

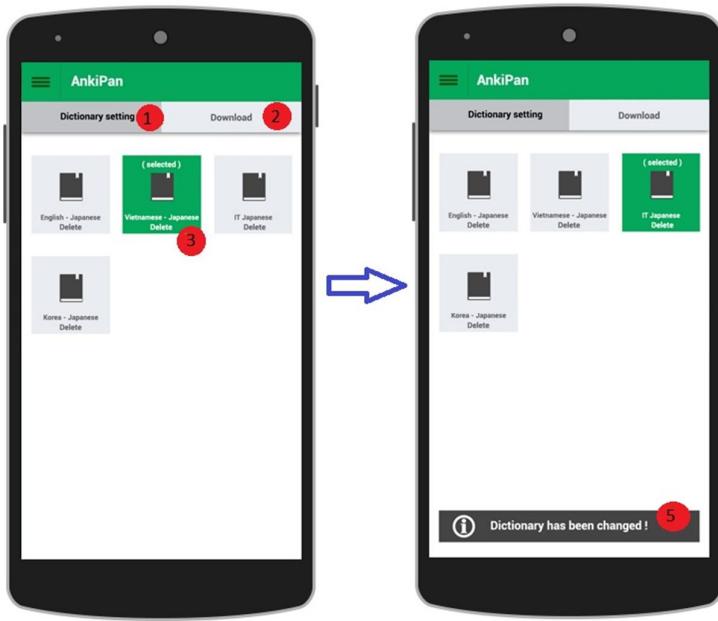
## 2.8. Clear history



### To Clear all history

1. Select History screen from Main Menu (see 2.1)
2. Tap on Clear history button ①
3. To confirm that you want to clear all history, tab on Yes button of confirmation dialog
4. If you want to back to History screen, tap on Favorite button ②

## 2.10. Select Dictionary package



A Dictionary package include a Japanese dictionary and a Kanji dictionary. You can select a Dictionary package to use in Dictionary screen.

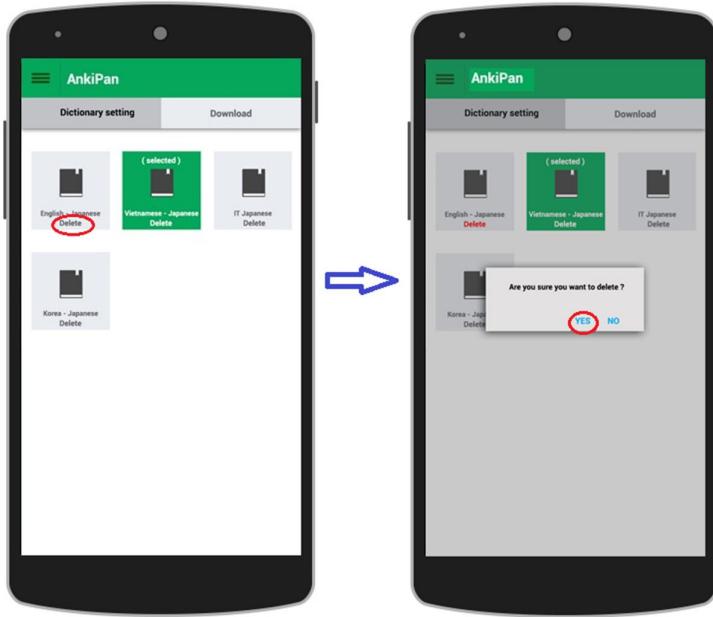
### Screen description:

- ① Dictionary setting tab: used to select or delete Dictionary package
- ② Download tab: used to download new Dictionary package
- ③ Dictionary package that is in use is marked by green background

### To select a Dictionary package

Tap on the Dictionary package that you want to use. The Dictionary package's background will turn to green and confirm message will be shown on footer of the screen.

## 2.11. Delete Dictionary package



### To delete a Dictionary package:

1. Tap on the Delete label of the Dictionary package
2. Tap Yes button of the confirmation dialog

## 2.12. Download Dictionary package



A Dictionary package include a Japanese dictionary and a Kanji dictionary. You can download the Dictionary package that explains words in your own language to use in Dictionary screen.

### To download a Dictionary package:

1. Select Download screen from Main Menu (see 2.1)
2. Tap on Download tab
3. If you want to see description of a Dictionary package, tap on Detail label ①
4. Tap on Download label ②
5. Tap on Yes button of the confirmation dialog. The Dictionary package will be downloaded to the application. Progress bar ③ shows the download progress
6. If you want to stop download, tap Cancel button

## VII. CONCLUSION

### 1. LESSONS LEARN

This project is the last wonderful lectures for us in FPT university. Through this project we learned how to apply all we studied to do the work. We got a lot of experience in document writing, which we did not pay much attention before. We also gained more technical knowledge. Especially, we learned to work which each other in a team. There was pressure and misunderstanding, but we did try to talk and get through that. This is the first time we run a project on our own, and many unexpected problems have occurred, but in the end of the project, we are happy that we did our best and completed the product that we expected.

### 2. FUTURE WORK

We plan to improve AKP image processing library in order to extract text in vertical direction and to work with documents of more complete structure, posters for example.

We also want to provide text recognition and dictionary packages of other languages.

We are in consider of adding translation and text-to-speech functions in next version. In the survey we conducted at the begin of this project, quite many students noted that they want these functions in the application.

## APPENDICES

### 1. SURVEY SUMMARY

# SURVEY SUMMARY

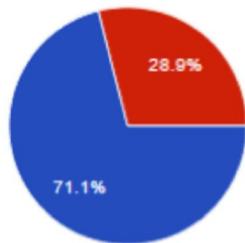
*Demands of using smartphone in reading/ translating Japanese*

**Created by:** Pham Trong Nghia

**Create date:** 05/21/2015

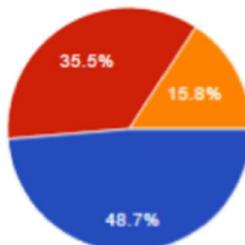
❖ **Total of responses: 78**

➤ **Do you use your smartphone in studying Japanese?**



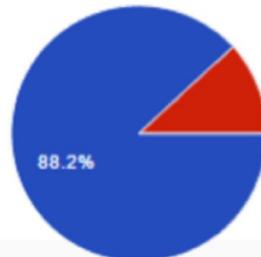
Yes 54 71.1%  
No 22 28.9%

➤ **Do you usually use Japanese document?**



Usually 37 48.7%  
Sometimes 27 35.5%  
Rarely 12 15.8%

➤ **Do you have troubles in understanding words when reading/ writing Japanese documents?**



Yes 67 88.2%  
No 9 11.8%

➤ **Besides, do you have any other troubles in reading Japanese document?**

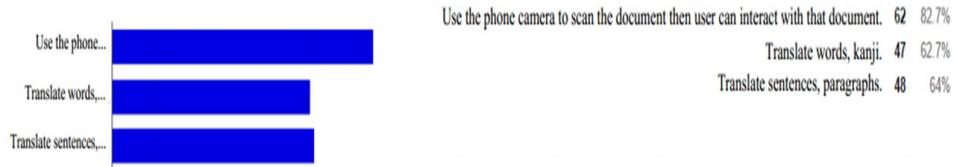
Answers:

- ✓ Reading and translating Kanji.
- ✓ Reading and writing Kanji.
- ✓ Difficult to translate because one words might has many meanings.
- ✓ Do not know how to pronounce the word/ Kanji.
- ✓ Having trouble with kanji.

➤ **Do you find it difficult to read the manual instruction of the consumables made in Japan?**



➤ **What functions do you want a Japanese translating application to have?**



➤ **Other functions:**

Answers:

- ✓ Speech-to-text and translate.
- ✓ Quick translate paragraph.
- ✓ Auto checking grammar errors in document.
- ✓ Small quiz games.
- ✓ Display pronunciation, meaning of a word, sentence or each word in a sentence.
- ✓ Nice and modern interface, good performance.
- ✓ Exactly translate sentences.
- ✓ Should divide words into specialized fields. Have paragraphs for users to practice their translating skills. Use images in word definition explaining.
- ✓ Have audio for word pronunciation. Add news reading function.
- ✓ Add synonyms, antonyms.
- ✓ Speech-to-text, translate then read the translated sentence.
- ✓ Translate kanji written by user.
- ✓ Real-time audio translating.

- ✓ Allow user to write note of the words they searched. Audio translating.
- Capture to translate

## 2. MEETING MINUTES

Below is a sample of our meeting minutes. All meeting minutes is attached in Meeting minutes folder in the CD.

### English version

<b>Project Name</b>	Anki Pan	<b>Project code</b>	AKP	
<b>Project Manager</b>	Bui Hong Nhung	<b>Conductor</b>	Bui Hong Nhung	
		<b>Secretary</b>	Bui Hong Nhung	
<b>Date</b>	20/07/2015	<b>Time</b>	12.30 – 13.00	
<b>Venue</b>	Room 100, Hoa Lac campus			
<b>Meeting topic</b>	Discuss about problems occurred in last weeks and project progress			
<b>Attendees</b>				
No.	<b>Full name</b>	<b>Unit / Group</b>	<b>Position</b>	<b>Attendance</b>
1	Tran Binh Duong	FPT University	Supervisor	Present
2	Bui Hong Nhung	AKP team	Project Manager	Present
3	Vu Hong Thai	AKP team	Technical leader	Present
4	Nguyen Duc Huy Hoang	AKP team	Developer	Present
5	Le Thanh Binh	AKP team	Designer	Absent
6	Pham Trong Nghia	AKP team	Developer	Present
7	Ngo Thi Mai	AKP team	Tester	Absent
<b>Objectives</b>				
Find solutions for occurred issues and specify how to write final report				
<b>Agenda</b>				
<ul style="list-style-type: none"> <li>- Error in the number of XML file of Application</li> <li>- Error in changing password of server</li> <li>- Report progress of the project</li> <li>- How to write Final report</li> </ul>				
<b>Contents</b>				
<b>I. Issues in this week</b> <ol style="list-style-type: none"> <li><b>1. Error in the number of XML file of Application</b> <ul style="list-style-type: none"> <li>- Nhung:           <ul style="list-style-type: none"> <li>+ After the 17<sup>th</sup> XML file is created, the Application can not compile</li> <li>+ We are using 2 open source, OpenCV and Tesseract, but when we import the 3<sup>th</sup> library, the Application can not compiled either.</li> <li>+ Due to the limit in the number of XML file, we are thinking of combining Result screen and Dictionary screen into 1 and remove Translate screen</li> </ul> </li> </ul> </li> </ol>				

- Supervisor
    - + Project team should try another IDE, Android studio for example. Eclipse still has some errors.
    - + Using OpenCV version 2.4.9 instead of version 2.4.10. As I searched on the Internet, there are some bugs in version 2.4.10.
- 2. Error in editing password of server**
- Nhung: Project team decided to use open source for editing password because we could not fix the error.
  - Supervisor: What is the error?
  - Hoang: I have problem in changing charset. The 2 charsets, before and after changing, is not the same
  - Supervisor: Project team ought to find out cause of the error and fix. I expected project team to do this part.
- II. Project progress**
- Nhung: This week, team members were too busy with studying Japanese, so we did not do much work
  - Supervisor: Do you have any solution for this? Do project team work together this week?
  - Hoang: Because each member has different study schedule, so we could not arrange for all team
  - Nhung: Even if we can arrange for a team work time, it is not suitable for everyone, so 1 or 2 member could not jointed. This period, almost work is of the development team, so I will try to arrange for 3 developer to work together.
  - Supervisor:
    - + All team members should support each other
    - + A lesson to learn: when manage a project, you should not let a issue of one member affect progress of all other member. You should left the issue on one side, and do other stuff. When you are back to the issue later, you may realize what was the cause.
    - + Should let Thai to take care more of the technical affair
    - + About combining screens, project team should evaluate if this solution is OK
    - + Project team should be united and sociable, go out together sometimes to relax.

### Conclusions

Discussed items	Decisions
1. Error in the number of XML file	Report to Supervisor result after project team switch to OpenCV version 2.4.9
2. Error in changing password	Find out the root cause of the error and fix. Do not use open source
3. How to write final report	<ul style="list-style-type: none"> <li>- Complete and deliver Final report to University in 15/08</li> <li>- Write down all process of working in this project</li> <li>+ Testing Tesseract</li> <li>+ All algorithms</li> <li>+ About project idea: what is special point, what issue can it solve</li> </ul>

### Issues

No.		Resolution	Target date
	N/A		

### Japanese version

プロジェクト名	Anki Pan	プロジェクトコード	AKP
プロジェクト管理者	Bui Hong Nhung	指揮者	Bui Hong Nhung
		秘書	Bui Hong Nhung

日付	20/07/2015	時間	12.30 – 13.00	
会場	Room 100, Hoa Lac campus			
ミーティングトピック	今週に発生した問題の相談と進歩報告			
<b>出席者</b>				
番	氏名	単位/グループ	担当	出席
1	Tran Binh Duong	FPT 大学	監督者	出席
2	Bui Hong Nhung	AKP チーム	管理者	出席
3	Vu Hong Thai	AKP チーム	技術リーダー	出席
4	Nguyen Duc Huy Hoang	AKP チーム	開発者	出席
5	Le Thanh Binh	AKP チーム	デザイナー	欠席
6	Pham Trong Nghia	AKP チーム	開発者	出席
7	Ngo Thi Mai	AKP チーム	テスター	欠席
<b>目標</b>				
問題の解決法を見出すと最後報告の書き方を確認する				
<b>議題</b>				
<ul style="list-style-type: none"> <li>- XML ファイル数のエーラ (アブリ)</li> <li>- パスワード変更のエーラ (ウェーブサービス)</li> <li>- 進歩報告</li> <li>- 最後報告の書き方</li> </ul>				
<b>内容</b>				
<p>ア. 今週の問題</p> <p>ア 1. XML ファイル数のエーラ (アブリ)</p> <ul style="list-style-type: none"> <li>- Nhung: <ul style="list-style-type: none"> <li>+ 17 番名の XML ファイルを追加すると、アブリがコンパイルできない</li> <li>+ 二つのライブラリー、OpenCV と Tesseract を使っているが、3 番名のライブラリーをインポートすると、アブリもコンパイルできない</li> <li>+ XML ファイル数限で、結果画面と辞書画面と一緒にしたり、翻訳画面を無しにしたりする解決法を考えている</li> </ul> </li> <li>- 監督者 <ul style="list-style-type: none"> <li>+ プロジェクトチームが他の IDE、例えば Android studio、を試したほうみたいと思う。Eclipse はいつかのエーラがあるらしい</li> <li>+ OpenCV 2.4.10 の代わりに OpenCV 2.4.9 にしてください。インターネットで調べた結果、バージョン 2.4.10 はまだバグがあるということ</li> </ul> </li> </ul> <p>ア 2. パスワード変更のエーラ (ウェーブサービス)</p> <ul style="list-style-type: none"> <li>- Nhung: 私たちはパスワード変更の機能にオープンソースを使うに決めた。エーラは直せなかったから</li> <li>- 監督者: エーラはどのようだ?</li> <li>- Hoang: チャセット変更に問題がある。チャセットは変更するあとは元のチャセット違う。</li> <li>- 監督者: 皆さんはこのエーラと原因を見出して、直しなさい。プロジェクトチームはこの機能を自力で開発と思ってきた。</li> </ul> <p>イ. プロジェクト進歩</p> <ul style="list-style-type: none"> <li>- Nhung: 今週、チームメンバーは日本語の勉強で忙しかったので、あまり仕事は進歩していない</li> </ul>				

- 監督者 : 解決法は考えたか?
- Hoang : メンバーは違う大学スケジュールがあるので、全員に手配できない
- Nhung : チームワークの時間が決まっても、全員に適用できないので、1、2人が参加できない。この期間、ほとんどのタスクは開発チームのタスクだから、開発者の3人がチームワークできるように手配する。
- 監督者 :
  - + チームメンバーがお互いにサポートすれば良い
  - + 得られた経験 : プロジェクト管理には、一人の問題を全員に影響させてはいけない。解決できない問題を一時中止して、他のタスクをする。後で振り返って考えるとこの問題の原因が分かるかもしれない
  - + Thaiさんに技術問題をもっと任せてください
  - + 画面を一緒にするについては、皆さんがそれで良いかどうか判断してください
  - + チームは共同で社交的で、たまにも一緒に遊びに行ってリラックスすれば良い

### 結論

相談項目	決定		
ア. XML ファイル数のエーラ	OpenCV 2.4.9 に変わった後で結果はどうか報告しなさい		
イ. パスワード変更のエーラ	原因を見出して、直す。オープンソースは使わない		
ウ. 最後報告の書き方	<ul style="list-style-type: none"> <li>- 8月15日までに完成して、大学に渡す</li> <li>- プロジェクトの過程を全部記述する           <ul style="list-style-type: none"> <li>+ Tesseract テストと結果</li> <li>+ アルゴリズム</li> <li>+ プロジェクトアイディア：特別な点、解決できる問題</li> </ul> </li> </ul>		
<b>問題</b>			
番	詳細	解決法	目標期日
	N/A		

## 3. CONFIGURATION MANAGEMENT

### 3.1. CI Identification and Naming Convention

No	Configuration items	Naming convention
<b>❖ Project Management</b>		
1	Project plan	AKP_ProjectManagementPlan_ver[version number] For example: AKP_ProjectManagementPlan_ver1.0
<b>❖ Requirement</b>		
2	Software requirement specification	AKP_SoftwareRequirementSpecification_ver[version number] For example: AKP_SoftwareRequirementSpecification_ver1.0

<b>❖ Design</b>		
3	Architecture design	AKP_ArchitectureDesign_ver[version number] For example: AKP_ArchitectureDesign_ver1.0
4	Database design	AKP_DatabaseDesign_ver[version number] For example: AKP_DatabaseDesign_ver1.0
5	Interface design	AKP_InterfaceDesign_ver[version number] For example: AKP_InterfaceDesign_ver1.0
	Source code	
6	Source code	AKP_SourceCode_ver[version number][Tested/Untested] For example: AKP_SourceCode_ver1.0Tested
<b>❖ Support Document</b>		
7	User manual	AKP_UserManula_ver[version number] For example: AKP_UserManula_ver1.0
<b>❖ Test</b>		
8	Test plan	AKP_TestPlan_ver[version number] For example: AKP_TestPlan_ver1.0
9	Unit Test Case	AKP_UnitTestCase_ver[version number] For example: AKP_UnitTestCase_ver1.0
10	Integration test case	AKP_IntegrationTestCase_ver[version number] For example: AKP_UnitTestCase_ver1.0
11	System test case	AKP_SystemTestCase_ver[version number] For example: AKP_SystemTestCase_ver1.0
12	Test data	AKP_TestData_ver[version number] For example: AKP_TestData_ver1.0
13	Test result	AKP_TestReport_ver[version number] For example: AKP_TestReport_ver1.0
<b>❖ Report</b>		
14	Personal weekly report	[Name]_WeeklyReport_W[week No]_[Send Date(ddmmyyyy)] For example: NhungBH_WeeklyReport_W4_04062015
15	Team weekly report	AKP_WeeklyReport_W[week No]_[Send

		Date(ddmmyyyy)]_[EN/JP) For example: AKP_WeeklyReport_W4_04062015_EN
❖ Date format		
16	Date wrote in files	DD/MM/YYYY For example: 20/07/2015
❖ File type		
17	MS Word	*.doc, *.docx
28	MS Excel	*.xls
29	MS Power Point	*.ptt
20	MS Project Plan	*.mpp
21	Astah	*.asta
22	Images	*.png, *.jpg, *.jpeg, *.bmp, *.gif

### 3.2. Version Numbering Rules<sup>[6]</sup>

❖ **For documents:**

Each file has a version number as part of its identity. This version number is physically represented as a 2-part string with the following format:

<version>.<revision>

For example, version 1.0 indicates 1 as the version, and 0 as the revision number.

The first version will be numbered 1.0

- **Version number:** appears to the left of the decimal. It is changed only when the core content of the item is significance changed. For example: when an item is completely overhauled, with substantial internal changes, the version 1.0 would become version 2.0.
- **Revision number:** appears to the right of the decimal. It is changed when the existing content is changed, but the main (or core) content is remained. The normal sequence of revision is 1.1, 1.2, and so on.

❖ **For Software source files:**

Software executable and support files are generally identified by name and version number.

The version number is physically represented as a 3-part string with the following format: <version>.<revision><update>

For example, version 1.1a indicates 1 as the version, 0 as the revision number, and a as the update level.

- **Version number:** appears to the left of the decimal. It is changed only when the core content of the item is significance changed, as when moving from one are of the development tool to another, when an Application is completely overhauled, or the user interface changes fundamentally. In this case, version 1.1a would become version 2.0.
- **Revision number:** appears to the right of the decimal. It is changed when new features, functionality or other content are added or significantly changed. In normal case, the core architecture or user interface have been extended or limited in some manner. The most common reason for changing the revision number is adding a new module or other functionality to the software. The normal sequence of revision is 1.0, 1.1 and 1.2 and so on.
- **Update level:** is appended or incremented when the only change to the software item is to correct one or more defects, without the addition of any new function. Version 1.1 would become v1.1a, 1.1b and so on. This updating is overridden when a combination revision, involving bug fixes and new feature additions, is performed. In such a case, the software revision number is incremented and any update indicator is dropped, as in v1.1b to 1.2.

### 3.3. Directory Structure

A folder used to store all project documents is created on Google driver. This folder includes of some main folders as viewed in the following table.

Main folder	Sub-folder	Purpose
Download		Store documents, material downloaded from Internet or other sources
Meeting minutes		Store project meeting minutes
Weekly report	W1_15052015	Store weekly report of week 1 (end at 15/05/2015)
	W2_22052015	Store weekly report of week 2
	W3_29052015	Store weekly report of week 3
	....	

Project report	Report1_Introduction	Store Introduction document
	Report2_ManagementPlan	Store PMP and related files
	Report3_SoftwareDetailDesign	Store SDD and related files
	Report4_SoftwareRequirement	Store SRS and related files
	Report5_SoftwareImplementationAndTest	Store SIT and related files
	Report6_UserManual	Store SUM
Source code	Increment 1	Store source code completed in Increment 1
	Increment 2	Store source code completed in Increment 2
	Increment 3	Store source code completed in Increment 3
Test Tesseract		Store tool, data used to test Tesseract library and Test result.
Templates		Store templates of documents
Training		Store training documents of team members
Others		Store other document of project team

## 4. CODING CONVENTION

### 4.1. Coding Convention

- ❖ Naming class: Use the uppercase letter for the first character of each word. For example: ImageProcessor
- ❖ Naming variables:
  - Start the name with a lowercase letter and follow Camel Case. For example: inputImage
  - Name of a variable must describe itself
- ❖ Catch exception
  - Catch all exceptions that may happened

- Throw all exceptions that may occur

## 4.2. Comment Convention

- Write comment for all functions, include what the functions do, what they returns
- Write comment for all important logic.
- Explain all parameters of each function

## REFERENCES:

- [1] 400万人に迫る！世界で日本語を学んでいるのはどんな人?  
<http://www.wochikochi.jp/topstory/2013/12/learning-japanese.php>
  - [2] Google Translate application on Google play store  
<https://play.google.com/store/apps/details?id=com.google.android.apps.translate&hl=en>
  - [3] CamDictionary application on Google play store  
<https://play.google.com/store/apps/details?id=com.intsig.camdict&hl=en>
  - [4] Jidict application on Google play store  
<https://play.google.com/store/apps/details?id=info.hoang8f.jdict>
  - [5] Risk management template  
<http://www.projectmanagementdocs.com/project-planning-templates/risk-management-plan.html>
  - [6] Version numbering rules  
EC\_FinalReport\_v1.0.pdf  
Author: Easy Chemistry team
  - [7] Tesseract  
[https://en.wikipedia.org/wiki/Tesseract\\_\(software\)](https://en.wikipedia.org/wiki/Tesseract_(software))
  - [8] Canny edge detection  
[http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny\\_detector/canny\\_detector.html](http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html)
  - [9] Sobel detector and Canny detector  
<http://www.egr.msu.edu/classes/ece480/capstone/fall13/group04/docs/danap.pdf>
- Tips to write user manual  
<http://online-learning.com/five-tips-writing-user-manual/>
- StudentGuide\_CapstoneProject\_JS\_v1.0.doc  
Author: FPT university
- FPT University The\_Student\_Guide\_Summer2013.pdf  
Autory: FPT university