Capstone Project Final Report **Admission Supporting Chatbot**

**FUTO Team**

**Group Members**

Nguyễn Văn Du – SE4312

Dương Việt Anh - SE04154

Đỗ Thị Thu Quỳnh – SE04133

Đào Mạnh Tuấn – SE04002

**Supervisor** Lecturer: Nguyễn Tất Trung

**Capstone Project code** ASC

**Hanoi, August 28th, 2018**

**Table of contents**

**ACKNOWLEDGEMENTS ............................................................................................................... 4 DEFINITIONS AND ACRONYMS ................................................................................................. 5**

**CHAPTER 1 : INTRODUCTION .................................................................................................... 7** 1.1 PURPOSE................................................................................................................................... 7 1.2 PROJECT INFORMATION ............................................................................................................ 7 1.3 THE PEOPLE .............................................................................................................................. 7

**CHAPTER 2 : PROJECT PLAN ................................................................................................... 12** 2.1 PURPOSE................................................................................................................................. 12

**CHAPTER 3 : SOFTWARE REQUIREMENT SPECIFICATION ........................................... 18** 3.1 PURPOSE................................................................................................................................. 18 3.2 FUNCTIONAL REQUIREMENTS ................................................................................................ 18

3.4 DATABASE DOCUMENT DIAGRAM ........................................................................................... 67

**CHAPTER 4 : SOFTWARE DESIGN ........................................................................................... 70** 4.1 PURPOSE................................................................................................................................. 70 4.2 ARCHITECTURE OVERVIEW .................................................................................................... 70

1.4 BACKGROUND .......................................................................................................................... 7 1.5 THE EXISTING SYSTEM ............................................................................................................. 8 1.6 THE PROPOSAL OF SYSTEM ....................................................................................................... 9

2.2 PROJECT MANAGEMENT PLAN ............................................................................................... 14

3.3 NON-FUNCTIONAL REQUIREMENT .......................................................................................... 65

4.3 DESIGN OF ASC SYSTEM .................................................................................................................... 73

**CHAPTER 5 : SOFTWARE TESTING DOCUMENTATION ............................................................... 192** 5.1 PURPOSE ........................................................................................................................................... 192

**CHAPTER 6 : USER MANUAL ................................................................................................................ 220** 6.1 DEPLOYMENT GUIDELINES ............................................................................................................... 220

5.2 TEST PLAN ........................................................................................................................................ 193

5.3 TEST CASE ........................................................................................................................... 200

5.4 TEST REPORT .................................................................................................................................... 206

6.2 USER GUIDELINES ............................................................................................................................. 223

**Acknowledgements**

*We wish to express our deepest gratitude to our supervisor Mr. Nguyen Tat Trung for his continuously sharing and motivating throughout the project. Under the instructions of Mr. Trung, the project team has always felt comfortable and confident. Apart from all the technologies and methodologies, Mr. Trung shared with us the philosophy of keeping the right attitude towards problems. We believe that was the most important factor that led to the success of this project.*

*We would also like to thank the instructors at FPT University for all the classes. We hope you will find this project as a reflection of the knowledge and experiences you have given us during this period of four years.*

*Finally, we truly appreciate the random fellows on the internet who has discussed with us. Without them, it would have likely been a real struggle to solve every tiny problem.*

**Definitions and Acronyms Terminologies**

**Terminology Definition Chatbot (also Bots)** Software that is used to interact between a computer and a human in

natural language. **Broadcast** A message proactively to users. **Nodes** A node is where different points of a dialog or workflow intersect **Utterance** Anything someone using chatbot says to it. **Intent** The few essential words that describe what the user wants the chatbot to

do. **Entity** Entities are the fields, data, or words the developer designates are

necessary for a chatbot to complete the user’s request. An entity could be a date, a time, a location, a description or any number of things. **Chat logs** Histories of all recorded human-to-bot interactions.

**Acronyms**

**Acronym Definition API** Application Programming Interface **BDD** Behavior-Driven Development **CPU** Central Processing Unit **NLP** Natural Language Processing **NLU** Natural Language Understanding **NER** Named Entity Recognition **BoW** Bag-of-Words **SVM** Support Vector Machine **TF-IDF** Term Frequency- Inverse Document Frequency **DBMS** Database Management System **DOM** Document Object Model **GUI** Graphical User Interface **HDD** Hard Disk Drive **HTTP** Hypertext Transfer Protocol **HTTPS** Hypertext Transfer Protocol Secure **IDE** Integrated Development Environment **UML** Unified Modeling Language **N/A** Not Available **OS** Operating System **PM** Project manager **REST** Representational State Transfer **SRS** Software Requirement Specification **TDD** Test-Driven Development

**UC** Use Case **UI** User Interface **UX** User Experience **URL** Uniform Resource Locator **FAQ** Frequently Asked Questions **PSID** Page-Scoped ID **Fb** Facebook **FU** FPT University **JWT** JSON Web Token **ASC** Admission Supporting Chatbot

**Chapter 1 : Introduction 1.1 Purpose**

This chapter provides an overview of the project including background information, a literature review of the existing system and raising a proposal for ideas of improvement.

**1.2 Project Information**

- Project name: Admission Supporting Chatbot - Project code: **ASC** - Project group name: **FUTO** - Product type: **Chatbot Application** - Timeline: **May 14th 2018 - August 28th 2018**

**1.3 The people**

**Team members**

**Full name Student**

**code Phone E-mail Role in**

**Group**

1 Nguyễn Văn Du SE04312 0904423162 dunvse04312@fpt.edu.vn Leader

2 Đỗ Thị Thu Quỳnh SE04133 0966343635 quynhdttse04133@fpt.edu.vn Member

3 Dương Việt Anh SE04154 01645500597 anhdvse04154@fpt.edu.vn Member

4 Đào Mạnh Tuấn SE04002 0974481558 tuandmse04002@fpt.edu.vn Member

*Table 1-2: Team member's information*

**1.4 Background**

In early 2016, the admission office of FPT University in Hanoi created a Facebook page named "Đại học FPT Hà Nội"1 to connect with students and their family. According to statistics of the admissions counselor2, so far, the page has nearly 84,000 followers, an average of 40 people inbox a day, and the peak period of enrollment is up to 60 people inbox a day.

1 “Đại học FPT Hà Nội” Facebook page: https://www.facebook.com/DaihocFPTHaNoi/

2 The admissions counselor ‘s Facebook page: https://www.facebook.com/laiiihonganh (Lại Hồng Anh)

**Supervisor**

**Full name Phone E-Mail Title**

Supervisor Nguyễn Tất Trung 0904399139 trungnt77@fe.edu.vn Lecturer

*Table 1-1: Supervisor's information*

*Figure 1-1: FU Facebook page*

The communication of potential students with FU offices is performed manually and it is a very time-consuming procedure. The communication will require a staff to expend several hours to find suitable answers and contact each student. It would be useful to reduce costs and time. Addressing this issue, the project aims to reduce the burden on the head of admissions, and potentially other users, by developing a convincing chatbot. The main goal of such a system is to conveniently retrieve information without having to look or browse several web pages to fetch answers to frequently asked questions.

**1.5 The Existing System**

While there are already chatbots that allow users to communicate, they all have certain problems. Below, we will go over a few chatbots application and their problems.

**Features Ana1 Harafunel2**

Support Vietnamese chatting. ✓

Serve in specific admission domain. ✓

Throw random greeting responses.

The accuracy of a domain specific bot can further improve.

Having the flow of conversations and context.

Allow the end user to leave personal information. ✓ ✓

Allow the end user to rate and review. ✓

1 Ana Facebook page: https://www.facebook.com/Study.at.US/ 2 Harafunel website: https://harafunnel.com/

Having dashboard administration. ✓ ✓

Allow the admin to view statistics. ✓

Allow the admin to manage chat logs. ✓ ✓

Allow the admin to send messages to all end users at time. ✓

Allow admin to categorize end users into different groups. ✓

*Table 1-3: The existing system information*

**1.6 The Proposal of system**

As we seek that not all bots are born the same. Bots differ from one another in many aspects.

1

*Figure 1-2: Major types of Bots* Furthermore, it is very important to know how to choose the use cases where bot can be utilized.

*Figure 1-3: Major use cases of Bots*

Considering the admission problems of FPT University, the proposal investigates the implementation of ASC system as a domain-specific chatterbox named FAQ bot, our work will show

1 Designing Bots Book: https://www.amazon.com/Designing-Bots-Creating-Conversational- Experiences/dp/1491974826

how a chatbot can work as domain specific information system and experiments on how the system’s accuracy could be improved based on a specific domain.

*Figure 1-4: Our idea* ASC is a chatbot designed for Vietnamese. Unlike existing applications, all basic functions will be usable for free and there have a delightful conversation. In addition, the application was evaluated by keeping logs of questions and answers and by feedback received by potential end users that used it.

**System functions**

• Allow the end user to chat in private.

• Allow the end user to rate and feedback.

• Allow the staff to login by ASC account.

• Allow the staff to view and update personal profile.

• Allow the staff to view some statistics.

• Allow the staff to search for account and response of the chatbot.

• Allow the staff to view facebook information, feedback and chat logs of the end user.

• Allow the staff to request to edit data.

• Allow the staff to receive notifications related to his/her activities.

• Allow the admin to manage data.

• Allow the admin to add tags or remove tags from end user.

• Allow the admin to manage ASC accounts including change role, status or create a new account.

• Allow the admin to collect end user’s personal information, then export to a file.

• Allow the admin to send messages to all potential end users with different labelling tags.

**Out of scope functions**

Because of the time limitation, we will not implement these following functions in the ASC initial version 1.0.0 release of the project but not permanently excluded, and will be developed in future version 2.0.0, although we are aware that they so also important:

• ASC chatbot will analyze end user’s information to recommend what should to do.

• ASC chatbot will be able to self-study, without manual training.

• ASC chatbot will be integrated into website besides Facebook Messenger.

• ASC chatbot will support English language.

• Staff can edit chatbot scenario to response the end user.

Finally, we use some new approaches to solve the stated problems. Having awareness of the importance of new technologies, we aim at applying them to our project.

**Special approaches**

• For backend system

o Using HTTP Methods & API Routes for building a Node.js RESTful API. o Using HTTP Status Codes Correctly if something goes wrong while serving a

request, we must set the correct status code for that in the response. o Using HTTP headers to Send Metadata. o Using Express Framework for Node.js REST API, to create browser

applications, and as such, it supports templating and rendering. o Using Microsoft Azure, Linux Virtual Machine for storage, hosting and

deployment. o Using real-time WebSocket technology with Socket.IO library. o Using Elasticsearch for searching and filtering. o Using Memcached in-memory data store for caching.

• For frontend system:

o Using ReactJS for web component rendering. o Using Redux for managing application state.

**Chapter 2 : Project Plan 2.1 Purpose**

This chapter provides an overview of the project plan including project organization and project management plan.

**Software Process Model**

*Figure 2-1: Iterative and Incremental Software Process Model*

ASC project uses the Iterative and Incremental Software Process Model. The Iterative and Incremental Software Process Model is mostly used when the scope of the project is big, the major requirements are defined clearly, some more details will be added later in software development. By using this software process model, we break down the developing system task into series of smaller tasks which will be completed separately, allowing us to take advantage of what was learned during development of earlier parts of the system. In addition, the iterative model is easier than other models when the issues are discovered. They are fed back to the team, and solutions will be found while the project is still in development.

**Roles and Responsibilities**

**2.1.2.1 Organization Structure**

**Role Responsibility**

Project Manager Planning, developing schedules, coordinating communication, generally responsible for keeping the team’s focus on the main goal.

Technical Leader

Responsible for choosing and deciding what technologies should be used, as well as for overseeing the work being done by other developers.

Quality Assurance Manager

Ensuring the product meets the certain standards of quality from requirements.

Test Leader

Responsible for test execution, including test set-up and test run, evaluation of test run and error recovery, defect logging and test results recording.

Developer Involve to code the product and review code of other developers.

Designer Involve to design product’s user interface.

Tester Involve to test the product.

Business Analyst Analyzes an organization or business domain and documents its

business or processes or systems.

*Table 2-1: Project Structure*

**2.1.2.2 Project Team Member**

**Team Member Role**

DuNV Project Manager, Developer, Business Analyst, Designer.

QuynhDTT Developer, Quality Assurance Manager, Business Analyst, Test

Leader.

AnhDV Technical Leader, Developer, Tester.

TuanDM Developer, Tester.

*Table 2-2: Project Team Member*

**Tools and Techniques**

**Programming languages** NodeJS, JavaScript

**Framework** ExpressJS, Socket.IO, Elasticsearch

**API** Wit.ai

**DBMS** MongoDB

**IDEs/Editors** Visual Studio Code, Atom

**UML tools** Visual Paradigm, Hackolade, Microsoft Visio, Astah

**Version Control** Gitlab

**Deployment server** Microsoft Azure, Linux Virtual Machines

**Project management tool** Microsoft Project, Trello, Backlog

**Process model** Iterative and Incremental Software Process Model

**Development process** Test-driven development, Behavior Driven Development

*Table 2-3: Project Team Member*

**2.2 Project Management Plan**

**Tasks**

Refers to “ASC\_ProjectManagement.mpp” file.

*Figure 2-2: ASC Project Management file*

**Meeting Minutes**

All meeting minutes will be written following this template:

*ASC* ***Meeting/Project Name: Date of Meeting:*** *18/05/2018* ***Time: (Type)*** *1 hours (Face-to-face)* ***Meeting Called by:*** *DuNV* ***Location:*** *FPT University – P205* ***Note Taker:*** *QuynhDTT* ***Time Keeper:*** *DuNV* 1. Meeting Objective

1. Assign new task for member 2. Do report No.2

2. Attendance ***Name Roles E-mail Phone*** Nguyen Van Du Project Manager dunvse04312@fpt.edu.vn 0904423162 Do Thi Thu Quynh Developer, Tester Leader quynhdttse04133@fpt.edu.vn 0966343635

Duong Viet Anh Developer, Tester anhdvse04154@fpt.edu.vn 01645500597

Dao Manh Tuan Developer, Tester tuandmse04002@fpt.edu.vn 0974481558

3. Done task

1. Build project structure 2. Configure git repository with security 3. Do report No.1

4. New task

1. Backend

a. Training Data with Wit.ai b. Setup Environment c. Code Modules Integration 2. Frontend

a. Setup environment b. Code Dashboard Homepage c. Code Login page 3. Do report No.2

5. Risk & Difficulty N/A

*Table 2-4: Meeting Minutes Template*

**Coding Conventions**

We strictly follow Eslint JavaScript Style Guide.

Please refer to **JavaScript Style Guide - Eslint.pdf** file or the official website at https://www.npmjs.com/package/eslint

**Risk Management Plan**

**No Description Avoidance plan Contingency plan Status**

**R1** Data loss - Use GitLab for version control. - Teach members how to use Git

and resolve conflicts. - Always have important backups.

Closed

**R2** Illness or absence of team members

- Restore backed up data from GitLab.

Closed

**R3** Misunderstan

ding of requirements

- Provide schedules in advance. - For long periods of absence,

members should notify the group in advance.

- Assign the tasks of

absent member to other members. - Work overtime if

necessary.

Closed

**R4** Requirement changed

- Discuss requirements carefully

with the customer. - Always ask for clarification if

requirement is unclear. - Comment need to meet reality and

possibility.

- Make sure idea’s

business logic is carefully analyzed.

Closed

**R5** Conflict

between team members

- Every new update of requirement has to be reviewed by all team members and supervisor. - Team member has to analyze

requirement carefully before raise up to team.

- If requirement has new update, all members have to join the meeting to aware and make decision.

Closed

**R6** Failure to meet deadline

- Everything must be documented. - Every team member has to express

clearly and carefully.

- Make sure any

miscommunication has to be resolved. - Plan and develop schedule

Closed carefully - Assign tasks carefully - Define punishment for team members who neglect work

*Table 2-5: Risk Management*

- Find the root cause of the problem - Reassign tasks - Focus on important

functions first

**Communication Plan**

*Weekly meeting schedule:* We use Iterative and Incremental Process Model, then we divide the system into two sub-systems (ASC Backend and ASC Frontend), each sub-system is divided into a series of small tasks. Each task is logged to TRELLO then estimated depending on difficulty and the amount of work by the whole team, after that the task will be assigned to team members by the Team Leader and depending on difficulty the Technical Leader will assign deadlines for each task. We will have a meeting every Sunday to inform to all team about what each member finished last week, the status (fast, on time or slow), the issues met and how to solve them. If any member raises any issue, the whole team will help to find out a solution together. After that, the team will define detailed stories for next week tasks.

*Daily meeting schedule*: Each sub-system has one development team with different schedule. When starting work-day, each team will have a stand-up meeting to inform to others: “What did I do yesterday?”, “What will I do today?” and “Is there any difficulty?”. By focusing on what each person accomplished yesterday and will accomplish today, the team gains an excellent understanding of what has been done and what remains.

*Unscheduled meeting*: If someone has an important problem that he wants to solve immediately, we will have a meeting for discussion, usually via some online channel: Slack, Facebook or Phone.

*Communication channel*: Our main communication channel is Slack. On the other hand, we use face-to-face meeting, Facebook group and comment on TRELLO issues. However, we sometimes make a phone call or instant message if someone has a problem.

*Figure 2-3: Communication via Slack*

**Chapter 3 : Software Requirement Specification**

**3.1 Purpose**

This chapter outlines functional and non-functional requirements of our website. It also provides some format constraints in common requirements and project success criteria. All members will work based on the information provided in this chapter.

**3.2 Functional Requirements**

**Business Rules**

**No Description**

**B1** Username must not be empty.

**B2** Username must be unique.

**Use Case Diagram**

*Figure 3-1:Use case diagram of ASC system*

**B3** User’s full name must not be empty.

**B4** User’s address must not be empty.

**B5** The email must not be empty.

**B6** Email address must be valid.

**B7** Each email address may be used for only one account.

**B8** Re-enter password must match with password.

**B9** Password must not be empty.

**B10** Password reset token expires after 24 hours.

**B11** The password length must not be less than 6.

**B12** Deactivation must note reasons.

**B13** Change deactivated status to active status must delete note and deactivated time.

**B14** Phone number must be valid.

**B15** Phone number must be numeric string only.

**B16** Exported files must be Microsoft Excel files with valid format.

**B17** The admin must fill out username and email field.

**B18** Every user accessing dashboard has a role: staff or admin.

**B19** The administrator is able to: active/ deactivate account, change account role, change

request state.

**B20** Staff can request to edit chatbot answer.

**B21** An answer editing request has 3 states: “PENDING”, “REJECTED”, “ACCEPTED”

**B22** Answer editing request form must have: old answer, new answer, created date, creator.

**B23** Only “PENDING” editing answer can be accepted by the admin, after that editing

answer’s state is “ACCEPTED”.

**B24** Only “PENDING” editing answer can be rejected by the admin, after that editing

answer’s state is “REJECTED”.

**B25** Answer editing request status only can be changed from “PENDING” to

“ACCEPTED” or “REJECTED”.

**B26** To review an answer request, the staff must have at least an editing answer.

**B27** The chatbot always has a state: “ON” or “OFF”.

**B28** The chatbot will not automatically reply in “OFF” state.

**B29** The chatbot will automatically reply in “ON” state.

**B30** When admin deactivates account. System logs that account out of ASC system.

**B31** Rating must be 1 in 5 states: “EXCELLENT”, “VERY GOOD”, “GOOD”, “BAD”,

“VERY BAD”.

**B32** When no reply after 10 minutes, the chatbot auto finishes the conversation.

*Table 3-1: Business rules*

**Use Cases**

**Actor Description**

End User Someone who uses the application to engage in a conversation.

Staff Everyone who has a ASC Account such as FU staffs.

Admin Administrator is a staff, has the highest permission level and is responsible for

managing administrative tasks.

*Table 3-2: Actor description*

**ID Actor Name Description**

UC-1.0 End User Chat End User asks a question and

chatbot replies automatically.

UC-2.0 End User Leave Feedback End User is able to leave feedback, which is comprised of a text message and a rating.

UC-3.0 End User Leave Personal Information End User leaves personal

information.

UC-4.0 Staff, Admin Reset Password Reset a forgotten password

using email.

UC-5.0 Staff, Admin Login Login to dashboard in ASC

account.

UC-6.0 Staff, Admin View Statistics View feedbacks, number of end user and related admission information statistics.

UC-7.0 Staff, Admin View Profile View personal profile.

UC-8.0 Staff, Admin Update Profile Update personal profile.

UC-9.0 Staff, Admin Change Password Change password of account.

UC-10.0 Staff, Admin View Account List View a list of accounts in the

system.

UC-11.0 Staff, Admin View Account Information

Detail

View detailed information regarding specific account.

UC-12.0 Staff, Admin Search Account Search for account by

username or full name.

UC-13.0 Staff, Admin Filter Account Filter account by role or status.

UC-14.0 Staff, Admin View Response & Answer List View chatbot responses

regarding answers.

UC-15.0 Staff, Admin Filter Response & Answer Filter chatbot response and

answer by intent.

UC-16.0 Staff, Admin Search Response Search for chatbot response.

UC-17.0 Staff, Admin Request to Edit Answer Request to edit a chatbot

answer.

UC-18.0 Staff, Admin Receive Notification Notification of changes to events automatically sent by the system as a result of an edit.

UC-19.0 Staff, Admin View Notification List View a list of notifications.

UC-20.0 Staff, Admin View Notification Detail View some details of the

notification.

UC-21.0 Staff, Admin View Fb Count Page View all End User interacting

with the chatbot.

UC-22.0 Staff, Admin View Chat Logs View a conversation history between End User and the chatbot.

UC-23.0 Staff, Admin View Fb Account Infor View information of Fb Account and his/her satisfaction ratings and reviews.

UC-24.0 Staff, Admin View Providing Infor View providing information

of End User.

UC-25.0 Staff, Admin Add Tag Add a tag to categorize and

filter End User.

UC-26.0 Staff, Admin Remove Tag Remove a tag from End User.

UC-27.0 Staff, Admin Export End User List Export a list of End Users into

Excel file.

UC-28.0 Staff, Admin Change Bot Status Turn chatbot on or off.

UC-29.0 Staff, Admin Logout Log out of the ASC

dashboard.

UC-30.0 Admin View Answer Editing Requests View a list of answers edited

by staff.

UC-31.0 Admin Accept Answer Editing

Request

Accept an answer editing request with status Pending.

UC-32.0 Admin Reject Answer Editing Request Reject an answer editing

request with status Pending.

UC-33.0 Admin Create Account Create a new ASC account.

UC-34.0 Admin Delete Account Delete an existing account.

UC-35.0 Admin Change Account Role Change role of account:

Admin or Staff.

UC-36.0 Admin Change Account Status Change status to be a deactivated account or active account.

UC-37.0 Admin Broadcast Send a single message to all of

the chatbot end users.

*Table 3-3: Use Case list*

**3.2.3.1 End User**

*Figure 3-2:Use case diagram of End User actor*

***3.2.3.1.1 Chat***

***3.2.3.1.1.1 Training the Chatbot***

*Figure 3-3: Training the Chatbot activity diagram*

**Explanation:**

(1): Define the intents chatbot to extract from natural language utterances. (2): Collect real end user utterances that the chatbot to map to intents.

(3): Assign the utterances collected to the different intents. (4): Randomly divide the utterances into two sets, a training set and a test set. A 70% training and 30% test is a typical split.

(5): Train the chatbot using the training set.

(6): Once training is complete, run the test set against the trained classifier and collect performance metrics such as accuracy, precision, and recall.

(7): Perform Error Analysis: review the results in previous step to understand why the classifier missed certain utterances. Update training data accordingly and go back to step (5)

(8): If satisfied with the results produced by the trained system, the system is now ready to be released. (9): When the chatbot is in use, continue to collect end user utterances. Then map results collected to new training/test data. Go back to step (4) and iterate.

**Detail Specification:**

I. Define intent

An intent represents the purpose of an end user’s input.

To define intent , we go to my customer FPT Admission Office to see what they think are the top reasons end users are contacting them. The typical support material that the reps use we will get closer to understanding the end user intent is website of FU1. It is largely divided into “Question” and “Answer”. Data named “Answer” are normal article data from the website with section, title, body attributes. Data named “Question” are the end user’s comments.

*Figure 3-4: Answer data*

1 FU website: http://daihoc.fpt.edu.vn/

*Figure 3-5: Question Data*

Since the raw data had too many topics such as admission procedures, tuition fees, scholarships, and other classes making the end user is interested in so much, we grouped them into larger classes, also called intents.

Besides the intent, we create a lot of entities. An entity represents a term or object in the end user’s input that provides clarification or specific context for a particular intent.

II. Collect training data

To get all real comments of potential end user from the “dai.hoc.fpt.edu.vn” website for training , we must create a crawler tool to accquire data.

Input: “http:daihoc.fpt.edu.vn/tuyen-sinh/”

Output: data.json

A sample of raw data is provided below:

III. Clean and assign data collected to the different intent.

To get high quality labeled training data, we must clean the misunderstanding and ambigious utterances. Only keeping the utterance having an intent and relating our model domain. To label training data, we attach each of utterance into diffrent intent and entity.

***3.2.3.1.1.2 Chat process***

*Figure 3-6: Chat state machine diagram*

*Figure 3-7: Chat activity diagram*

*Use Case Specification*

UC ID and Name: **UC-1.0 Chat**

Created by: **QuynhDTT** Date Created: **24/05/2018**

Primary Actor: **End User** Secondary Actors:

Trigger: N/A

Description: Allow End User to send a question and receive the answer to their question

immediately.

Preconditions: **PRE-1.1.** The end user starts the Facebook Messenger application.

Post conditions: **POST-1.1.** The answer to the end user’s question is sent.

Normal Flow: **1.0 Chat**

1. End User enters a question then clicks on the send button.

2. System generates an answer displayed in the chat window to reply

his/her question.

Alternative Flows: **1.0 End User violates some business rule in step 1**

1. System sends a goodbye message to finish the conversation.

Exceptions: **1.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: High

Business Rules: **B32**

Other Information: N/A

Assumptions: N/A

***3.2.3.1.2 Leave Feedback***

*Figure 3-8: Feedback state machine diagram*

*Use Case Specification*

UC ID and Name: **UC-2.0 Leave Feedback**

Created by: **QuynhDTT** Date Created: **24/05/2018**

Primary Actor: **End User** Secondary Actors:

Trigger: N/A

Description: Allow End User to rate and review for chatbot in the end of conversation.

Preconditions: **PRE-2.1.** End User has already chat.

**PRE-2.2.** End User has not sent any message for 10 minutes.

Post conditions: **POST-2.1.** System saves new feedback to database.

Normal Flow: **2.0 Leave Feedback**

1. After 10 minutes finishing conversation, system sends a message

with 5 rating quick replies.

2. End User chooses one of them. 3. System asks for reasons that they chose it.

4. End User gives their review for chatbot. 5. System saves his/her feedback to database.

Alternative Flows: **2.0 End User violates some business rule in step 2**

1. System sends the help message” (Fb frirst name) vui lòng chọn 1

trong số các lựa chọn bên dưới giúp Futo.”

**2.0 End User violates some business rule in step 4**

2. System sends a goodbye message to finish the conversation.

Exceptions: **2.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: Medium

Business Rules: **B31, B32**

Other Information: N/A

Assumptions: N/A

***3.2.3.1.3 Leave Personal Information***

*Figure 3-9: Leave Personal Information state machine diagram*

*Figure 3-10: Leave Personal Information activity diagram*

*Use Case Specification*

UC ID and Name: **UC-3.0 Leave Personal Information**

Created by: **QuynhDTT** Date Created: **24/05/2018**

Primary Actor: **End User** Secondary Actors:

Trigger: N/A

Description: Allow End User to leave personal information to contact.

Preconditions: **PRE-3.1.** End User sends a message to want to talk to some real person.

Post conditions: **POST-3.1.** System saves end user’s personal information to database.

Normal Flow: **3.0 Leave Personal Information**

1. System asks some questions about getting end user’s personal

information. 2. End User replies to each of them.

3. System saves information that end user provides.

Alternative Flows: **3.0 End User violates some business rule in step 2**

1. System sends a goodbye message to finish the conversation.

Exceptions: **3.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: Medium

Business Rules: **B32**

Other Information: N/A

Assumptions: N/A

**3.2.3.2 Staff**

*Figure 3-11: Use case diagram of Staff actor*

***3.2.3.2.1 Reset password*** *Use Case Specification*

UC ID and Name: **UC-4.0 Reset password**

Created by: **AnhDV** Date Created: **24/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to reset a forgotten password.

Preconditions: **PRE-4.1**. Staff account has been set up and username and password has

been generated.

Post conditions: **POST-4.1**. New password is reset and sent to staff via his/her email.

Normal Flow: **4.0 Reset password**

1. Staff visits the dashboard page. 2. System displays Login page.

3. Staff clicks “Quên mật khẩu”. 4. System displays Forgot Password page.

5. Staff enters his/her email, then clicks “Xin cấp lại mật khẩu”. 6. System displays a message that asks the staff to check email and

follow instruction. 7. Staff clicks on the link in email.

8. System displays Reset Password page. 9. Staff is required to input password twice, then clicks “Xác nhận”.

10. System displays message “Thay đổi mật khẩu thành công”. 11. System displays Login page.

Alternative Flows: **4.0 User violates some business rule in step 5**

1. System displays help message popup. 2. Staff enters email address complying the business rules then clicks

“Xin cấp lại mật khẩu”. 3. Go to step 6 in normal flow.

**4.0 User violates some business rule in step 9**

1. System displays help message popup.

2. Staff enters password complying the business rules then clicks

“Xác nhận”.

3. Go to step 10 in normal flow.

Exceptions: **4.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: Low

Business Rules: **B5, B6, B8, B10, B11**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.2 Login*** *Use Case Specification*

UC ID and Name: **UC-5.0 Login**

Created by: **TuanDM** Date Created: **24/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff who has ASC account to login.

Preconditions: **PRE-5.1**. Staff account has been set up and username and password has

been generated.

Post conditions: **POST-5.1**. The staff is logged into ASC successfully.

**POST-5.2**. System displays Dashboard Home Page.

Normal Flow: **5.0 Login**

1. Staff visits the website. 2. Staff fills valid username and password then clicks “Đăng nhập”.

3. System displays Dashboard Home page.

Alternative Flows: **5.0 Staff violates some business rule in step 2**

1. System displays help message popup. 2. Staff enters username and password complying the business rules

then clicks “Đăng nhập”. 3. Go to step 3 in normal flow.

Exceptions: **5.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: High

Business Rules: **B1, B9**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.3 View Statistics*** *Use Case Specification*

UC ID and Name: **UC-6.0 View Statistics**

Created by: **AnhDV** Date Created: **24/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to view statistics about majors, feedbacks, number of end

users.

Preconditions: **PRE-6.1.** Staff can access the system.

**PRE-6.2.** Staff logins successfully.

Post conditions: **POST-6.1.** System displays statistics.

Normal Flow: **6.0 View Statistics**

1. From index page, system displays statistics using 4 cards and 2

charts: 2. Card about the number of all end users asking.

3. The number of new end users asking today. 4. The number of old end users asking today.

5. The number of end users proving personal information. 6. Pie chart about FU academic majors.

7. Bar chart about the number of end user’s feedback.

Alternative Flows: N/A

Exceptions: **6.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.4 View Profile*** *Use Case Specification*

UC ID and Name: **UC-7.0 View Profile**

Created by: **DuNV** Date Created: **24/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow all staffs to view his/her personal profile.

Preconditions: **PRE-7.1.** Staff can access the system.

**PRE-7.2.** Staff logins successfully.

Post conditions: **POST-7.1.** System displays the staff profile page.

Normal Flow: **7.0 View Profile**

1. From the website header, staff clicks on avatar icon in the top right

corner. 2. Staff clicks “Trang cá nhân”.

3. System displays the staff profile page with following tab:

o Thông tin tài khoản o Tên tài khoản o Vai trò o Thông tin liên hệ o Họ và tên o Email o Số điện thoại o Địa chỉ

Alternative Flows: N/A

Exceptions: **7.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.5 Update Profile*** *Use Case Specification*

UC ID and Name: **UC-8.0 Update Profile**

Created by: **QuynhDTT** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to update his/her basic information after signing in.

Preconditions: **PRE-8.1.** Staff can access the system.

**PRE-8.2.** Staff logins successfully.

Post conditions: **POST-8.1.** New staff profile is saved into database.

**POST-8.2.** New updated profile is displayed in profile page.

Normal Flow: **8.0 Update Profile**

1. From the website header, staff clicks on avatar icon in the top right

corner.

2. Staff clicks “Trang cá nhân”. 3. System displays the Staff profile.

4. Staff can edit information about fullname, address, phone, email. 5. Staff clicks “Lưu thông tin cá nhân” to save the changed

inforamtion. 6. System saves changed into database then displays updated profile

information.

Alternative Flows: **8.0 Staff violates some business rule in step 4**

1. System displays help message under each input area. 2. Staff enters fullname, email, phone and address complying the

business rules then clicks “Lưu thông tin cá nhân”. 3. Go to step 6 in normal flow.

Exceptions: **8.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Low

Business Rules: **B3, B4, B5, B6, B7, B14, B15**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.6 Change Password*** *Use Case Specification*

UC ID and Name: **UC-9.0 Change password**

Created by: **AnhDV** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to change his/her account password.

Preconditions: **PRE-9.1.** Staff can access the system.

**PRE-9.2.** Staff logins successfully.

Post conditions: **POST-9.1.** New password is saved into database.

**POST-9.2.** Staff logins successfully with new password.

Normal Flow: **9.0 Change Password**

1. From the website header, staff clicks on avatar icon in the top right

corner. 2. Staff clicks “Thay đổi mật khẩu”

3. System displays the Change Password page. 4. Staff enters current password, new password and re-enter password

in the corresponding text boxes. 5. Staff clicks “Đổi mật khẩu”.

**6.** System saves new data into database.

Alternative Flows: N/A

Exceptions: **9.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: Low

Business Rules: **B8, B9, B11**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.7 View Account List*** *Use Case Specification*

UC ID and Name: **UC-10.0 View Account List**

Created by: **AnhDV** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to view the list of accounts in the system.

Preconditions: **PRE-10.1.** Staff can access the system.

**PRE-10.2.** Staff logins successfully. **PRE-10.3.** Account data has been added to the system by the admin.

Post conditions: System displays the account list screen with necessary elements.

Normal Flow: **10.0 View Account List**

1. Staff clicks on “Tất cả tài khoản” on menu in the left side bar of the

screen.

2. System displays list of all accounts.

Alternative Flows: N/A

Exceptions: **10.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.8 View Account Information Detail*** *Use Case Specification*

UC ID and Name: **UC-11.0 View Account Information Detail**

Created by: **AnhDV** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to view detail information of an account.

Preconditions: **PRE-11.1.** Staff can access the system.

**PRE-11.2.** Staff logins successfully.

**PRE-11.3.** Account data has been added to the system by the admin.

Post conditions: System displays a model with chosen account information.

Normal Flow: **11.0 View Account Information**

1. Staff clicks on “Tất cả tài khoản” on the left side of the screen. 2. System displays list of all accounts.

3. Staff clicks “Xem” button on each row in table. 4. System displays account information.

Alternative Flows: N/A

Exceptions: **11.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Low

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.9 Search Account*** Use Case Specification

***3.2.3.2.9.1 Search Account by Username***

UC ID and Name: **UC-12.0 Search Account by Username**

Created by: **TuanDM** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to search account(s) belong to username.

Preconditions: **PRE-12.1.** Staff can access the system.

**PRE-12.2.** Staff logins successfully.

**PRE-12.3.** Account data has been added to the system by the admin.

Post conditions: N/A

Normal Flow: **12.0 Search Account by Username**

1. Staff clicks on “Tất cả tài khoản” on the left side of the screen.

2. System displays table of all accounts. 3. In the “Tài khoản” text box, staff enters text.

4. System displays account(s) that related to search keyword.

Alternative Flows: N/A

Exceptions: **12.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.9.2 Search Account by Fullname***

UC ID and Name: **UC-12.0 Search Account by Fullname**

Created by: **TuanDM** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to search account by fullname.

Preconditions: **PRE-12.1.** Staff can access the system.

**PRE-12.2.** Staff logins successfully.

**PRE-12.3.** Account data has been added to the system by the admin.

Post conditions: N/A

Normal Flow: **12.0 Search Account by Fullname**

1. Staff clicks on “Tất cả tài khoản” on the left side of the screen. 2. System displays account list.

3. In the “Họ và tên” text box, staff enters text. 4. System displays account list that related to search keyword.

Alternative Flows: N/A

Exceptions: **12.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.10 Filter Account*** Use Case Specification

***3.2.3.2.10.1 Filter Account by Role***

UC ID and Name: **UC-13.0 Filter Account by Role**

Created by: **TuanDM** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to filter accounts by role.

Preconditions: **PRE-13.1.** Staff can access the system.

**PRE-13.2.** Staff logins successfully. **PRE-13.3.** Account data has been added to the system by the admin.

Post conditions: N/A

Normal Flow: **13.0 Filter Account by Role**

1. Staff clicks role dropdown button

2. System displays role list in dropdown list. 3. Staff selects a specific role.

4. System hided role list and displays selected role and accounts list

related to selected role.

Alternative Flows: N/A

Exceptions: **13.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.10.2 Filter Account by Status***

UC ID and Name: **UC-13.0 Filter Account by Status**

Created by: **TuanDM** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to filter account by status.

Preconditions: **PRE-13.1.** Staff can access the system.

**PRE-13.2.** Staff logins successfully. **PRE-13.3.** Account data has been added to the system by the admin.

Post conditions: N/A

Normal Flow: **13.0 Filter Account by Status**

1. Staff clicks on status dropdown button. 2. System displays status list in dropdown list.

3. Staff selects a specific status. 4. System hided status list and displays selected state and accounts list

related to selected status.

Alternative Flows: N/A

Exceptions: **13.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.11 View Response & Answer List*** *Use Case Specification*

UC ID and Name: **UC-14.0 View Response & Answer List**

Created by: **AnhDV** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to view list of responses regarding answers.

Preconditions: **PRE-14.1.** Staff can access the system.

**PRE-14.2.** Staff logins successfully.

Post conditions: System displays response and answer.

Normal Flow: **14.0 View Response & Answer List**

1. Staff clicks on “Quản lí dữ liệu của Chatbot” on the left side of the

screen.

2. System displays list of responses regarding answers.

Alternative Flows: N/A

Exceptions: **14.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Low

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.12 Filter Response & Answer*** *Use Case Specification*

UC ID and Name: **UC-15.0 Filter Response & Answer**

Created by: **QuynhDTT** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to filter response and answer.

Preconditions: **PRE-15.1.** Staff can access the system.

**PRE-15.2.** Staff logins successfully.

**PRE-15.3.** Response and Answer data has been added to the system by the admin.

Post conditions: **POST-15.1.** System displays filtered response and answer.

Normal Flow: **15.0 Filter Response & Answer.**

1. Staff clicks on intent dropdown butoon. 2. System displays intent list in dropdown list.

3. Staff selects a specific intent. 4. System hided intent list and displays selected intent and responses

regarding answers list related to selected intent.

Alternative Flows: N/A

Exceptions: **15.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Low

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.13 Search Response*** *Use Case Specification*

UC ID and Name: **UC-16.0 Search Response**

Created by: **QuynhDTT** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to search chatbot response(s).

Preconditions: **PRE-16.1.** Staff can access the system.

**PRE-16.2.** Staff logins successfully.

**PRE-16.3.** Response data has been added to the system by the admin.

Post conditions: **POST-16.1.** System displays searched responses regarding answers.

Normal Flow: **16.0 Search Response**

1. Staff clicks on “Quản lí dữ liệu của Chatbot” on the left side of the

screen. 2. System displays responses regarding answers list.

3. In the response textbox, Staff enters text.

4. System displays response regarding answers list related to search

keyword.

Alternative Flows: N/A

Exceptions: **12.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Low

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.14 Request to Edit Answer***

*Figure 3-12: Answer Editing Request state machine diagram Use Case Specification*

UC ID and Name: **UC-17.0 Request to Edit Answer**

Created by: **QuynhDTT** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to request to edit an answer of chatbot.

Preconditions: **PRE-17.1.** Staff can access the system.

**PRE-17.2.** Staff logins successfully.

**PRE-17.3.** Answer data has been added to the system by the admin.

Post conditions: **POST-17.1.** An answer editing request is successfully submitted and

noticed to admin.

Normal Flow: **17.0 Edit Answer**

1. Staff clicks “Quản lí dữ liệu của Chatbot” on the left side of the

screen.

2. System displays response regarding answers list. 3. In each row of answer, Staff clicks on edit button.

4. Staff enters new answer. 5. Staff clicks “Lưu”.

6. System saves new answer editing request into database with

pending status.

Alternative Flows: N/A

Exceptions: **17.0-E1 – Cannot communicate with server**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Low

Business Rules: **B20, B21, B22**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.15 Receive notification*** *Use Case Specification*

UC ID and Name: **UC-18.0 Receive notification**

Created by: **TuanDM** Date Created: **08/06/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: An chatbot editing answer, end user’s request or changing account detail

from admin, which is related to Staff, is created.

Description: Allow Staff to receive notification.

Preconditions: **PRE-18.1.** Staff can access the system.

**PRE-18.2.** Staff logins successfully.

**PRE-18.3.** Admin deactivates account successfully.

**PRE-18.4.** Admin changes role of account. **PRE-18.5.** Staff request to edit chatbot answer.

**PRE-18.6.** Admin rejects answer editing request. **PRE-18.7.** Admin accepts answer editing request.

**PRE-18.7.** End User requests to contact.

Post conditions: **POST-18.1.** Staff receives notification.

Normal Flow: **18.0 Account is deactivated.**

1. Admin deactivates staff’s account. 2. Staff receives notification.

**18.1 Role of account is changed.**

1. Admin changes staff’s role.

2. Staff receives notification.

**18.2 Staff request to edit Answer.**

1. Staff submits a request for editing answers. 2. Admin receives notification.

**18.3 Answer is rejected.**

1. Admin rejects answer editing request.

2. Staff receives notification that your request has been rejected.

**18.4 Answer is accepted.**

1. Admin accepts answer editing request. 2. Staff receives notification that your request has been accepted.

**18.5 End User requests to contact.**

1. End User send a call back or chatting request with staff . 2. Admin receives notifications.

Alternative Flows: N/A

Exceptions: **18.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: High

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.16 View notification list*** *Use Case Specification*

UC ID and Name: **UC-19.0 View notification list**

Created by: **TuanDM** Date Created: **08/06/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to view notifications about account status, answer editing

request status, end user’s contact request.

Preconditions: **PRE-19.1.** Staff can access the system.

**PRE-19.2.** Staff logins successfully.

**PRE-19.3.** Staff has an action history.

Post conditions: **POST-19.1.** System resets notification count to zero.

Normal Flow: **19.0 View notification list**

1. On top right of navigation bar, Staff clicks on the notification icon.

2. System displays dropdown menu to show all notifications. 3. Staff scrolls down to view more notifications.

4. System changes status of all notifications to read.

Alternative Flows: N/A

Exceptions: **19.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.17 View notification detail*** *Use Case Specification*

UC ID and Name: **UC-20.0 View notification detail**

Created by: **TuanDM** Date Created: **08/06/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: Mark selected notification as read.

Description: Allow Staff to view specific content to the notification.

Preconditions: **PRE-20.1.** Staff can access the system.

**PRE-20.2.** Staff logins successfully.

**PRE-20.3.** Staff has notification(s).

Post conditions: **POST-20.1.** Redirect to specific content correspond to each notification.

Normal Flow: **20.0 View notification detail**

1. Staff clicks on the notification icon on top right of navigation bar.

2. Staff clicks on a notification in notification list.

Alternative Flows: N/A

Exceptions: **20.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.18 View Facebook Count Page*** *Use Case Specification*

UC ID and Name: **UC-21.0 View Facebook Count Page**

Created by: **AnhDV** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to view list of end user’s information.

Preconditions: **PRE-21.1.** Staff can access the system.

**PRE-21.2.** Staff logins successfully.

Post conditions: System displays all end users interacting with the chatbot.

Normal Flow: **21.0 View Facebook Count Page**

1. Staff clicks “Quản lí dữ liệu từ Facebook” on the left side of the

screen. 2. System displays a table of all Facebook accounts having a conversation with the chatbot. Some information is displayed with each of account:

3. Fb name 4. Count of his/her messages

5. Current Bot status 6. Labeling tag

Alternative Flows: N/A

Exceptions: **21.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Low

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.19 View Chat Logs*** *Use Case Specification*

UC ID and Name: **UC-22.0 View Chat Logs**

Created by: **AnhDV** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to view conversation history between specific end user and the

chatbot.

Preconditions: **PRE-22.1.** Staff can access the system.

**PRE-22.2.** Staff logged in.

Post conditions: System displays conversation history.

Normal Flow: **22.0 View Chat Logs**

1. Staff clicks “Lịch sử hội thoại” on each row of facebook account

table.

2. System displays the history of all messages from end user. 3. Staff clicks “Xem thêm” to view more messages.

Alternative Flows: N/A

Exceptions: **22.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Low

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.20 View Fb Account Infor*** *Use Case Specification*

UC ID and Name: **UC-23.0 View Fb Account Infor**

Created by: **AnhDV** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to view a specific facebook account information detail

corresponding with his/her feedback(s).

Preconditions: **PRE-23.1.** Staff can access the system.

**PRE-23.2.** Staff logged in.

Post conditions: System displays information about facebook account.

Normal Flow: **23.0 View Fb Account Infor**

1. Staff clicks “Lịch sử hội thoại” on each row of facebook account

table.

2. System displays some tabs. 3. Clicks “Thông tin Facebook” tab

4. System displays facebook name, gender and his/her feedbacks.

Alternative Flows: N/A

Exceptions: **23.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Low

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.21 View Providing Infor*** *Use Case Specification*

UC ID and Name: **UC-24.0 View Providing Infor**

Created by: **AnhDV** Date Created: **26/05/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to view providing personal information of end user.

Preconditions: **PRE-24.1.** Staff can access the system.

**PRE-24.2.** Staff logged in.

Post conditions: System displays the providing information of end user.

Normal Flow: **24.0 View Providing Infor**

1. Staff clicks “Lịch sử hội thoại” on each row of facebook account

table. 2. System displays some tabs.

3. Staff clicks “Thông tin đã cung cấp” tab. 4. System displays end user’s providing information.

Alternative Flows: N/A

Exceptions: **24.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: High

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.22 Add Tag*** *Use Case Specification*

UC ID and Name: **UC-25.0 Add Tag**

Created by: **QuynhDTT** Date Created: **01/06/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to add item tags to end user.

Preconditions: **PRE-25.1.** Staff can access the system.

**PRE-25.2.** Staff logged in.

Post conditions: **POST-25.1.** End User is tagged with some label.

Normal Flow: **25.0 Add Tag**

1. Staff clicks “Lịch sử hội thoại” on each row of facebook account

table. 2. System displays some tabs.

3. Staff clicks “Gãn nhãn phân loại”. 4. System displays all labelling tag with the end user.

5. Staff clicks inside add tags field and choose an existing tag from the dropdown list or type a new tag then choose recently generated tag from the dropdown list. 6. Staff clicks “Lưu” button.

7. System saves changed data into database.

Alternative Flows: N/A

Exceptions: **25.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.23 Remove Tag*** *Use Case Specification*

UC ID and Name: **UC-26.0 Remove Tag**

Created by: **QuynhDTT** Date Created: **01/06/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to remove a tag from the end user.

Preconditions: **PRE-26.1.** Staff can access the system.

**PRE-26.2.** Staff logged in.

Post conditions: **POST-26.1.** Tag is removed successfully.

Normal Flow: **26.0 Remove Tag**

1. Staff clicks “Lịch sử hội thoại” on each row of facebook account

table. 2. System displays some tabs.

3. Staff clicks “Gán nhãn phân loại” tab. 4. System displays all tags labeling with the end user.

5. Staff selects a tag, then clicks “x”. 6. Staff clicks “Lưu” button.

7. System saves changed to database.

Alternative Flows: N/A

Exceptions: **26.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.24 Export End User List*** *Use Case Specification*

UC ID and Name: **UC-27.0 Export End User List**

Created by: **QuynhDTT** Date Created: **01/06/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to export list of all End Users in the system.

Preconditions: **PRE-27.1.** Staff can access the system.

**PRE-27.2.** Staff logged in.

Post conditions: **POST-27.1.** Excel file is generated.

Normal Flow: **27.0 Export End User List**

1. Staff clicks “Quản lí dữ liệu từ Facebook” on the left side of the

screen.

2. Staff clicks “Tải xuống dữ liệu” button on top right of the screen. 3. System auto generates end user’s information list into excel file.

Alternative Flows: N/A

Exceptions: **27.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: Medium

Business Rules: **B16**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.25 Change Bot Status***

*Figure 3-13: Bot state machine diagram*

*Use Case Specification*

UC ID and Name: **UC-28.0 Change Bot Status**

Created by: **AnhDV** Date Created: **01/06/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to turn on/off chatbot.

Preconditions: **PRE-28.1.** Staff can access the system.

**PRE-28.1.** Staff logged in.

Post conditions: **POST-28.1.** System turns chatbot on/off corresponding with staff request.

Normal Flow: **28.0 Turn on chatbot**

1. Staff chooses a specific end user in facebook account table, then

clicks “ON” toggle button.

2. Whenever the end user asking, chatbot will automatically reply.

**28.0 Turn off chatbot**

1. Staff chooses a specific end user in facebook account table, then

clicks “OFF” toggle button.

2. Whenever the end user asking, chatbot will not automatically reply.

Alternative Flows: N/A

Exceptions: **28.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: Medium

Business Rules: **B27, B28, B29**

Other Information: N/A

Assumptions: N/A

***3.2.3.2.26 Logout*** *Use Case Specification*

UC ID and Name: **UC-29.0 Logout**

Created by: **AnhDV** Date Created: **01/06/2018**

Primary Actor: **Staff** Secondary Actors:

Trigger: N/A

Description: Allow Staff to log out of ASC system.

Preconditions: **PRE-29.1.** Staf can access the system.

**PRE-29.1.** Staff logged in.

Post conditions: **POST-29.1.** Staff logs out of system.

Normal Flow: **29.0 Logout**

1. On Navigation Bar, Staff clicks on profile avatar.

2. Staff clicks “Đăng xuất”. 3. System logs Staff out of system and redirects to Login page.

Alternative Flows: N/A

Exceptions: **29.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.3 Admin***

*Figure 3-14: Use case diagram of Admin actor*

***3.2.3.3.1 View Answer Editing Requests*** *Use Case Specification*

UC ID and Name: **UC-30.0 View Answer Editing Requests**

Created by: **QuynhDTT** Date Created: **30/05/2018**

Primary Actor: Admin Secondary Actors:

Trigger: N/A

Description: Admin views a list of all chatbot answer editing requests.

Preconditions: **PRE-30.1**. Admin logins into ASC administration successfully.

Post conditions: **POST-30.1**. System displays a list of requests.

Normal Flow: **30.0 View Answer Editing Request List**

1. Admin clicks “Kiểm duyệt dữ liệu Chatbot” tab on the left side in

the window screen. 2. System displays all requests as table.

Alternative Flows: N/A

Exceptions: **30.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Medium

Business Rules: **B19, B26**

Other Information: N/A

Assumptions: N/A

***3.2.3.3.2 Accept Answer Editing Request*** *Use Case Specification*

UC ID and Name: **UC-31.0 Accept Answer Editing Request**

Created by: **QuynhDTT** Date Created: **30/05/2018**

Primary Actor: Admin Secondary Actors:

Trigger: N/A

Description: Admin accepts an answer editing request.

Preconditions: **PRE-31.1**. Admin logins into ASC administration successfully.

**PRE-31.2**. Answer editing request status is pending.

Post conditions: **POST-31.1**. Answer editing request status is changed to accepted.

**POST-31.2**. System pushes new status to Websocket server to notify staff.

Normal Flow: **31.0 Accept an answer opening request**

1. Admin navigates to Request Management.

2. System displays a list of answer editing request. 3. Admin chooses an answer editing request from list.

4. Admin chooses option “Chấp nhận”. 5. System changes chosen answer editing request to accepted.

6. System sends a notification about his/her answer editing request has

been accepted.

Alternative Flows: N/A

Exceptions: **31.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Low

Business Rules: **B23, B25, B26**

Other Information: N/A

Assumptions: N/A

***3.2.3.3.3 Reject Answer Editing Request***

UC ID and Name: **UC-31.0 Reject Answer Editing Request**

Created by: **QuynhDTT** Date Created: **30/05/2018**

Primary Actor: Admin Secondary Actors:

Trigger: N/A

Description: Admin rejects an answer editing request.

Preconditions: **PRE-31.1**. Admin logins into ASC administration successfully.

**PRE-31.2**. Answer editing request status is pending.

Post conditions: **POST-31.1**. Answer editing request status is changed to rejected.

**POST-31.2**. System pushes new status to Websocket server to notify staff.

Normal Flow: **31.0 Reject answer editing request**

1. Admin navigates to Request Management. 2. System displays a list of answer editing request.

3. Admin chooses an answer editing request from list. 4. Admin chooses option “Từ chối”.

5. System displays admin message textbox. 6. Admin fills in message.

7. Admin clicks “Gửi phản hồi”. 8. System changes chosen answer editing request to rejected.

9. System sends a notification about his/her answer editing request has

been rejected.

Alternative Flows: N/A

Exceptions: **31.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Low

Business Rules: **B24, B25, B26**

Other Information: N/A

Assumptions: N/A

***3.2.3.3.4 Create Account*** *Use Case Specification*

UC ID and Name: **UC-32.0 Create Account**

Created by: **QuynhDTT** Date Created: **30/05/2018**

Primary Actor: Admin Secondary Actors:

Trigger: N/A

Description: Admin creates a new account.

Preconditions: **PRE-32.1**. Admin logins into ASC administration successfully

Post conditions: **POST-32.1**. System displays Add Account page.

**POST-32.2**. A new account is created.

Normal Flow: **32.0 Create Account**

1. Admin clicks “Thêm mới tài khoản” tab on the left side in the

window screen. 2. Add Account page is displayed.

3. Admin enters username, email, fullname and role, then clicks “Tạo

tài khoản”.

4. System saves the new account with his/her information into

database and sends password to his/her email.

Alternative Flows: **32.0 Admin violates some business rule in step 3**

1. System displays help message under each input area.

2. Admin enters usename, email and fullname complying business

rules, then clicks “Tạo tài khoản”.

3. Go to step 4 in normal flow.

Exceptions: **32.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Medium

Business Rules: **B1, B2, B5, B6, B7, B17**

Other Information: N/A

Assumptions: N/A

***3.2.3.3.5 Delete Account*** *Use Case Specification*

UC ID and Name: **UC-33.0 Delete Account**

Created by: **QuynhDTT** Date Created: **30/05/2018**

Primary Actor: Admin Secondary Actors:

Trigger: N/A

Description: Account is deleted from the system.

Preconditions: **PRE-33.1**. Admin logins into ASC administration successfully.

Post conditions: **POST-33.1**. Selecting account is deleted.

Normal Flow: **33.0 Delete Account**

1. Admin clicks “Tất cả tài khoản” tab on the left side in the window

screen. 2. Accounts is displayed as table.

3. Admin clicks on “Xoá” button on each row. 4. System deletes the selected account from system.

Alternative Flows: N/A

Exceptions: **33.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Medium

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.2.3.3.6 Change Account Role*** *Use Case Specification*

UC ID and Name: **UC-34.0 Change Account Role**

Created by: **QuynhDTT** Date Created: **30/05/2018**

Primary Actor: Admin Secondary Actors:

Trigger: N/A

Description: Admin changes role of account.

Preconditions: **PRE-34.1**. Admin logins into ASC administration successfully.

Post conditions: **POST-34.1**. Role of his/her account is changed.

Normal Flow: **34.0 Change Account Role**

1. Admin clicks “Tất cả tài khoản” tab on the left side in the window

screen.

2. Accounts is displayed as table. 3. Admin changes specific role from role dropdown list.

4. System saves the new updated role.

Alternative Flows: N/A

Exceptions: **34.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Medium

Frequency of Use: Medium

Business Rules: **B18, B19**

Other Information: N/A

Assumptions: N/A

***3.2.3.3.7 Change Account Status*** *Use Case Specification*

UC ID and Name: **UC-35.0 Change Account Status**

Created by: **QuynhDTT** Date Created: **30/05/2018**

Primary Actor: Admin Secondary Actors:

Trigger: N/A

Description: Admin changes status of account.

Preconditions: **PRE-35.1**. Admin logins into ASC administration successfully

Post conditions: **POST-35.1**. Account status is changed

Normal Flow: **35.0 Change Account Status**

1. Admin clicks “Tất cả tài khoản” tab on the left side in the window

screen.

2. Accounts is displayed as table. 3. Admin select a specific user click “Hoạt động” | “Ngừng” depends

on account current status. 4. Admin fill reason that causes admin action.

5. Status of account is updated.

Alternative Flows: N/A

Exceptions: **35.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: Low

Frequency of Use: Low

Business Rules: **B12, B13, B19, B30**

Other Information: N/A

Assumptions: N/A

***3.2.3.3.8 Broadcast*** *Use Case Specification*

UC ID and Name: **UC-36.0 Broadcast**

Created by: **QuynhDTT** Date Created: **30/05/2018**

Primary Actor: Admin Secondary Actors:

Trigger: N/A

Description: Admin sends information to all end user in the system.

Preconditions: **PRE-36.1**. Admin logins into ASC administration successfully.

Post conditions: **POST-36.1**. A single message is sent to all end users in the system.

Normal Flow: **36.0 Broadcast**

1. Admin navigates to Facebook Data Management.

2. System displays broadcast form. 3. Admin inputs broadcast content, selects appreciate label, then

clicks “Gửi”.

4. System sends a single message to all chosen end users.

Alternative Flows: N/A

Exceptions: **36.0-E1 – Cannot communicate with server.**

1. ASC displays error message.

Priority: High

Frequency of Use: High

Business Rules: **N/A**

Other Information: N/A

Assumptions: N/A

***3.3* Non-functional Requirement**

**Security**

✓ Token-Based Authentication, relies on a signed token that is sent to the server on each request. ✓ Apply SSL encryption to restrict user access and prevent man-in-the-middle attacking. ✓ All passwords are encrypted by MD5 algorithm. ✓ The security matrix is as the following table:

**Function End User Staff Admin**

Chat ✓

Leave Feedback ✓

Leave Personal Information ✓

Reset Password ✓ ✓

Login ✓ ✓

View Statistics ✓ ✓

View Profile ✓ ✓

Update Profile ✓ ✓

Change Password ✓ ✓

View Account List ✓ ✓

Search Account ✓ ✓

View Account Information Detail ✓ ✓

Filter Account ✓ ✓

View Response & Answer List ✓ ✓

Filter Response & Answer ✓ ✓

Search Response ✓ ✓

Request to Edit Answer ✓ ✓

Receive Notification ✓ ✓

View Notification List ✓ ✓

View Notification Detail ✓ ✓

View Facebook Count Page ✓ ✓

View Chat Logs ✓ ✓

View Fb Account Infor ✓ ✓

View Providing Infor ✓ ✓

Add Tag ✓ ✓

Remove Tag ✓ ✓

Export End User List ✓ ✓

Change Bot Status ✓ ✓

Logout ✓ ✓

View Answer Editing Requests ✓

Reject Answer Editing Request ✓

Accept Answer Editing Request ✓

Create Account ✓

Delete Account ✓

Change Account Role ✓

Change Account Status ✓

Broadcast ✓

*Table 3-4: Security matrix*

**Accuracy**

**Usability**

✓ The dashboard interface should be elegant, simple: links and buttons look like they are clickable, and are easily clickable, search box is wide enough for users to see what they have typed, interface does not break at various zoom levels. ✓ Chatbot avatar design is distinct, colorful and sets of words chatbot using are simple, friendly

to create a friendly feel for the conversation and give the chatbot a bit of a persona. ✓ Having human in the loop to resolve ambiguity and provide response supervision. ✓ Having response buttons, rating buttons and clickable menus to help guide end users to their

answer when needed. ✓ Using typing indicators allow the chatbot to interact with an end user in a more natural

conversation.

**3.4 Database document diagram**

✓ The overall accuracy is calculated by diving total number of correct answers by the number

of questions asked.

**Maintainability & Extensibility**

✓ Follow the coding convention of Airbnb JavaScript Style Guide to help improve readability

of source code and make the website more maintainable. ✓ Client-side web application uses Redux architecture. It complements React's compostable

view components by utilizing a unidirectional data flow.

**Availability and Scalability**

✓ Using Azure Load Balancer for high availability. ✓ Using Microsoft Azure Virtual Machines for automatic horizontal scalability. ✓ Using Azure Monitor to monitor all services and alert when something wrong happened.

**Performance**

✓ Using HTTP/2, which uses a single, multiplexed connection to allow multiple requests to be sent on the same connection instead of establishing new connection for each request, increase speed of request-response. ✓ Web frontend uses React which uses several clever techniques to minimize the number of

costly DOM operations required to update the UI. ✓ Web frontend is a Single Page Application with the goal of providing a more fluid user

experience. ✓ Node.js brings event-driven programming to web servers, enables development of fast web

servers in JavaScript. ✓ Using Microsoft Azure, Linux Virtual Machine for storage, hosting and deployment.

*Figure 3-15: ASC database document diagram*

**Chapter 4 : Software Design 4.1 Purpose**

This chapter is to give the developer team the overview and detailed design of what the system’s architecture is, and how they should be implemented. This chapter consists of:

✓ Architecture overview ✓ Component diagram

✓ Detailed design ✓ Detailed description of components

✓ Database design

**4.2 Architecture Overview**

**System Architecture**

**4.2.1.1 Diagram**

*Figure 4-1: ASC System Architecture*

**4.2.1.2 Explanation**

(1) ASC Frontend sends a request to ASC Backend.

**Implementation**

Request URL: https://futo-dashboard.luzotech.com/data/facebook

Request Method: GET

Status Code: 200

(2) ASC Backend requests to send mail using library.

(1) Messenger’s built-in NLP automatically detects meaning and intent in every text message. (2) ASC Backend requests to Wit.AI API to get all entities detected in the message.

(3) ASC Backend sends messages using Facebook Webhook.

**Implementation**

Request URL: https://graph.facebook.com/v3.0/me/messages?access\_token=

Request Method: POST

**System Architecture Explanation**

The entire project will be deployed on Microsoft Azure. We aim at delivering a secured, responsive, and highly available system. In the following section, we will explain the function and mechanism of each unit in the system architecture design.

**4.2.2.1 CloudFlare CDN**

*Figure 4-2: CloudFlare* **CloudFlare** provides a content delivery network and distributed domain name server (DNS) services, sitting between the visitor and the CloudFlare user’s hosting provider, acting as a reverse proxy for websites**. We use it to provide addition layer of protection for ASC via Cloud Flare’s SSL and resolve domain name to IP address point to server.**

**4.2.2.2 Nginx**

*Figure 4-3: Nginx* **Nginx** is a web server. It can act as a reverse proxy server for HTTP, HTTPS, SMTP, POP3, and IMAP protocols, as well as a load balancer and an HTTP cache. **We use Nginx as a web server to serve static assets of ASC Frontend System.**

**4.2.2.3 React**

*Figure 4-4: React* **React** is an open-source JavaScript library for building user interfaces, allows developers to create large web applications that use data which can change over time, without reloading the page. **We use it to build entire website view instead of plain HTML/JavaScript.**

**4.2.2.4 Redux**

*Figure 4-5: Redux* **Redux** is a predictable state container for JavaScript apps. **Redux** evolves the ideas of **Flux** what is the application architecture that Facebook uses for building client-side web applications. It complements **React** view components by utilizing a unidirectional data flow. **We use Redux to take advantages of better separation of concerns, less time debugging.**

**4.2.2.5 MongoDB**

*Figure 4-6: MongoDB* **MongoDB** is a free and open-source cross-platform document-oriented database program. **We use it to manage ASC operational data.**

**4.2.2.6 Elasticsearch**

*Figure 4-7: Elasticsearch* **Elasticsearch** is a search engine that provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. **We use it to implement full-text search for end user’s name and name of school.**

**4.2.2.7 Node.js**

*Figure 4-8: Node.js*

**Node.js** is an open-source, cross-platform JavaScript run-time environment for executing JavaScript code server-side.

**4.2.2.8 Socket.IO**

*Figure 4-9: Socket.IO* **Socket.IO** is a JavaScript library for real-time web applications. It enables real-time, bi- directional communication between web clients and servers. It has two parts: a client-side library that runs in the browser, and a server-side library for node.js**. We use Socket.IO to push real-time updates to our Web application when there are changes of answer information, when dashboard user’s status is updated, etc....**

**4.3 Design of ASC System**

**Architecture Layers Design**

**4.3.1.1 ASC– API Layer Design** *Figure 4-10: ASC - API Layer Design*

**No Directory/File Description Convention**

1 app.js Entry point for the application. Everything in the app

gets triggered through this file. N/A

2 config/ This folder contains additional configuration files. XXXConfig.js

3 controllers/ Contain Authentication, Account, Request,

LogMessage, UserInformation, Utterance, Bot. XXXController.js

4 constans/ Contain the constants. N/A

5 bot\_modules/ Contain modules related to bot. N/A

6 doc/ Contains documentation that uses the API. N/A

7 models/ Contain all Mongoose models. N/A

8 services/ Contains files that directly interact with database

like AccountService.js, RequestService.js. XXXService.js

9 views/ Contain all ejs files. N/A

10 routes/ Contain all route logic for RESTful API endpoint. N/A

11 templates/ Contain all template. N/A

12 utilities/ Contain helper files. N/A

13 test/ Contain unit tests, integration tests, fixtures, and

others test apparatus. N/A

14 package.json Contain meta data about application: the list of

dependencies, npm run script. N/A

15 .eslintrc.json Contain rules for ESLint, which a set of rules to keep

our project’s code clean and well conventional. N/A

*Table 4-1: ASC-API Layer Detail*

**4.3.1.2 ASC – Frontend Design**

*Figure 4-11: ASC – Frontend Design*

**No Directory/File Description Convention**

1 node\_modules/ This folder contains all library of node

modules N/A

2 public/ This folder contains resource about

html, css, javascript N/A

3 src/ This folder contains all source code of

project N/A

4 src/actions/ This folder contains all Redux actions

which components dispath N/A

5 src/components/ This folder contains some components

which is common UI for whole app N/A

6 src/constants/ This folder contains all constants

variable of project N/A

7 src/img/ This folder contains some image N/A

8 src/pages/ This folder contains all pages of

application, each page map with a route N/A

9 src/reducers/ This folder contains all Redux Reducer

of projects N/A

10 src/utils/ This folder contains some file to

connect with api N/A

**Database Design**

**4.3.2.1 Explanation of database decision**

As our system have following characteristic:

o Users manage data frequently on dashboard. o Flexible data model. o System may need to scale. o Provide real-time experience for statistics.

So, we chose MongoDB because:

o MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from

document to document and data structure can be changed over time. o MongoDB is free and open-source. o MongoDB enables scaling by shards (partitions data by key ranges and distributes the

data among two or more database instances). o MongoDB allows stores sub-document inside document avoiding join multiple tables,

therefore providing faster write and read time.

11 src/App.css The file contains some css of App

component N/A

12 src/App.js

This root component of application. All others components are show through this component

N/A

13 src/index.js

Entry point for the application. Everything in the app gets triggered through this file.

N/A

14 src/

registerServiceWorker.js

This file to register service with opera system to help app always run. N/A

15 src/routes.js This configurations. file contains route matching

N/A

16 .gitignore Specifies intentionally untracked files

to ignore. N/A

17 package.json

This file contains information of project and all library with it is version of node modules

N/A

18 package-lock.json The modules file tree describes that was the generated exacted node

N/A

*Table 4-2: ASC- Frontend Layer Detail*

**4.3.2.2 Database Diagram**

*Figure 4-12: ASC database document diagram*

**4.3.2.3 Data Dictionary**

**Collection Field Type Description**

accounts \_id ObjectId MongoDB

ObjectId, Unique for each document

username String User’s name

email String User’s email

password String User’s password

role String Role of user

fullname String Fullname of user

phone String Contact phone of

user

address String Address of user

active Boolean Status of user:

active or not

visible Boolean Status of user:

visible or not

deactivatedTime Date Time that user is

deactivated

deactivatedReason String Reason if user is

deactivated

resetPasswordExpires Date Time that mark all token create after this time is valid

resetPasswordToken String Password reset

token

requests \_id ObjectId MongoDB

ObjectId, Unique for each document

answerId String Id of answer that

user edited

editedContent String Content of answer

that user edited

createdDate Date Time when request

is created

createdBy String User created

request

state String State of request: pending, rejected or approved

reviewer String User that reviewed

request

reviewedDate Date Time when request

is reviewed

answers \_id ObjectId MongoDB

ObjectId, Unique for each document

answers ArrayString List of answers

content String Content of answer

createdBy String User when created

answer

createdDate Date Time when created

answer

reviewedDate Date Time when

reviewed answer

reviewer String User when

reviewed answer

logmessages \_id ObjectId MongoDB

ObjectId, Unique for each document

senderId String psid of user

sending messages

messages BsonArray List of messages

content String Text content in

message

isBot Boolean Whether or not the message is sent by bot.

time Date Time when sent

message

responses \_id ObjectId MongoDB

ObjectId, Unique for each document

intent String Intent of end user’s

message

response String Response of

chatbot

entityValue BsonArray List of entities and

values

entity String Entity that extracted in message

value String Value of each

entity

\_id ObjectId MongoDB

ObjectId, Unique for each document