

INTERSHIP PROJECT REPORT
(PROJECT TERM AUGUST – DECEMBER 2019)

CHATBOT PROJECT

SUBMITTED BY
DILIP CHALAMALASETTY
REG NUMBER: 11611851

COURSE CODE: CSE 447 – INDUSTRY CO -OP PROJECT -1

UNDER THE GUIDANCE OF
KIRTI BALA | ASSISTANT PROFESSOR, LPU
AND
RAHUL VANPULLY | ASSOCIATE PROGRAM MANAGER, eClerx Service Ltd

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING



L OVELY
P ROFESSIONAL
U NIVERSITY

Transforming Education Transforming India

MARKS GIVEN BY THE EXTERNAL SUPERVISOR

CONTINUOUS ASSESSMENT (CA) for INTERNSHIP/OJT

(By external Supervisor from organization)

Name of the student DILIP CHALAMALASETH Registration Number 11611851

Internship Project Title (if/any): CHATBOT PROJECT

Name of Organization & Address: ECLERX SOLUTIONS LTD., A- BLDG. NO. 11,
6TH FLOOR, MINDSPACE, AIROLI, NAVI MUMBAI - 400708

Name of External Internship in-charge (with mobile number):

RAHUL VANPULLY Contact No: 9967225590

S.No.	Criteria	Marks Obtained	Maximum Marks
1	Student conduct during internship	10	10
2	Punctuality and Enthusiasm	18	20
3	Technical Skill & Knowledge	18	20
4	Performance	48	50
	Total	94	100

Date 04-12-2019

Authorized Signatory 

Name RAHUL VANPULLY

Designation ASSOCIATE PROGRAM MANAGER

Company Seal

DECLARATION

I hereby declare that the project work entitled “**SARA (Chatbot)**” is an authentic record of our own work carried out as requirements of Internship for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Associate Professor Kirthi Bala and Mr Rahul Vanapully (Associate Process Manager, eClerx Service Ltd) during (August to December 2018). All the information furnished in this internship project report is based on my own intensive work and is genuine.

Name of Student: **DILIP CHALAMALASETTY**

Registration Number: 11611851

DILIP CHALAMALASETTY

(Signature of Student)

Date: 30 Nov 2019

ACKNOWLEDGEMENT

I take this opportunity to express our gratitude and respect to all those who have helped me throughout our working period on the real time company environment. Doing internship in eClerx help us a lot to understand the new technology and how to grow in a corporate world. My special thanks is to our mentor Mr. Rahul Vanapally(Associate Process Manager), who helped me a lot to show us the right path how to work in a company and to Learn the various aspects of application.

I owe my regards to the entire faculty of the department of Computer Science at LPU from where I learnt the basics of Computer Science and I express my sincere thanks to all our course mates who supported us in the project through various informal discussions which were very valuable to the successful completion of the project.

-Dilip Chalamalasetty

Table of Contents

CHAPTER 1: INTRODUCTION.....	6
1.1 ABOUT ORGANIZATION	6
CHAPTER 2: INTRODUCTION TO PROJECT UNDERTAKEN	8
2.1 TECHNOLOGIES USED TO DEVELOP THIS PRODUCT:.....	8
2.2 SOFTWARES REQUIRED TO INSTALL:	8
2.2.1 Angular CLI:	8
2.2.1.1 Use of this software in project:.....	9
2.2.2 Node Runtime environment:.....	9
2.2.2.1 Use of nodejs and Koa in Project:	9
2.2.3 Elastic search and kibana	10
2.2.3.1 Use of elastic search in our project:.....	10
2.2.4 Mongo dB and robo3t:	10
2.2.4.1 Use of MongoDB in our project:	10
2.2.5 Anaconda:.....	11
2.2.5.1 About Spacy.....	11
2.2.5.1.1 Tokenization:.....	11
2.2.5.1.2 Parts of Speech Tagging	12
2.2.5.1.3 Dependency Parsing	12
2.2.5.1.4 Lemmatization:	13
2.2.5.1.5 Named Entity Detection:.....	13
2.2.5.1.6 Use of Spacy in Project:	14
2.2.5.2 About Rasa:	14
2.2.5.2.1 File structure of Rasa:	14
2.2.5.2.2 RASA FUNCTIONALITIES	15
2.2.5.2.2.1 COMMANDS FOR RUNNING THE RASA SERVER:.....	15
2.2.5.2.2.2 CONVERTING MD FILES TO JSON AND VICE VERSA:	16
2.2.5.2.2.3 NLU.MD NLU DATA FORMAT:	17
2.2.5.2.2.4 RASA ARCHITECTURE:	17
2.2.5.2.2.5 IMPORTANT LINKS TO DIFFERENT MODULES:	18
CHAPTER 3: PROJECT RESEARCH WORK.....	19
3.1 PANDORABOTS PLATFORM:.....	19
3.1.1 FEATURES:	19
3.1.1.2 DEPLOY:	19
3.1.2 FILES REQUIRED:	23
3.1.2.1 SET_FILES:	23
3.1.2.2 MAP_FILES:	23
3.1.2.3 SUBSTITUTION_FILES:	23
3.1.2.4 UDC.AIML - ULTIMATE DEFAULT CATEGORY:	23
3.2 PYTHON NATURAL LANGUAGE PROCESSING AND UNDERSTANDING MODULES RESEARCH	24
3.2.1 PROS OF EACH MODULE.....	24
3.2.2 CONS OF EACH MODULE	25
CHAPTER 4: DFD SECTION:.....	26
4.1 LEVEL-0 DATA FLOW DIAGRAM	26
4.2 LEVEL-1 DATA FLOW DIAGRAM	26
4.3 LEVEL-2 DATA FLOW DIAGRAM	27
CHAPTER 5: CONCLUSION.....	28
CHAPTER 6: BIBLIOGRAPHY	28

CHAPTER 1: INTRODUCTION

1.1 About Organization

eClerx helps businesses work smarter by its innovative business process management, change management, data-driven insights and advanced analytics powered by subject matter experts and smart automation. And it has eClerx Digital which is the trusted brand of choice to world's largest global brands for creative production, eCommerce/web operation and analytics and insights service.

It has digital delivery employees at five production hubs which are there in Mumbai, Pune, Chandigarh, Verona and Phuket. It was founded in 2000 by Anjan Malik and PD Mundhra.

eClerx provides critical business operations services to over fifty global fortune 500 client, including several of the world leading companies across financial services, cable and telecommunication, retail, fashion, media, & entertainment, travel and leisure, software and high-tech.

eClerx Digital team of 3000+ full-time digital delivery employees at our five production hubs in Mumbai, Pune, Chandigarh, Verona and Phuket apply deep digital expertise to effectively support the most demanding global clients by employing a follow the sun delivery model. eClerx Digital's innovative delivery model drives the "metrics that matter" for our clients: improved acquisition, conversion and retention and overall lifetime value of your customer 24x7x365.

eClerx Marketing- Maximize your marketing efforts to build awareness, generate leads, and increase sales.

eClerx ecommerce- Remove the friction from the user experience to reduce cart abandonment and increase average order value

.

eClerx Business Intelligence & Analytics - Make sense of and use your data for personalization, consistently accurate forecasting, and to direct resources towards profitable activities.

The industries eClerx serve include: financial services, cable and telecommunications, retail, fashion, media and entertainment, manufacturing, travel and leisure, software and high tech.

The popular clients which eClerx provides its products and services are Dell, Citibank, Autodesk, HSBC, Paypal, Adobe, Timberland, Comcast, etc.

CHAPTER 2: INTRODUCTION TO PROJECT UNDERTAKEN

SARA is a chatbot that is on development phase. It is developing in a way that it is helpful for end customers interactions to answer their questions. Our Team is trying to make it more interactive so that the end customers will benefit from it. Its services are provided from eClerx client website.

2.1 TECHNOLOGIES USED TO DEVELOP THIS PRODUCT:

CATEGORY	TECHNOLOGIES
WEB DEVELOPMENT	Angular 8, Nodejs, Koa server
NLP AND NLU	Spacy, Rasa
NO SQL DATABASES	Elasticsearch, MongoDB

2.2 SOFTWARES REQUIRED TO INSTALL:

- Angular 8 CLI
- Node runtime environment
- Elastic search
- Kibana
- MongoDB
- Robo 3t
- Anaconda

2.2.1 Angular CLI: This Framework is used to design the web layout for the project, Typescript is the preferred language in the Angular 8. Advantages are mentioned below

- Component-based architecture that provides a higher quality of code
- Google Long-Term Support
- Seamless updates using Angular CLI
- High Performance
- Loved by millions of developers

- Unit-test friendly
- Reusability

2.2.1.1 Use of this software in project:

- By using this framework, our team create the interface of the chatbot.
- Routing played the key role in navigating and rendering the components using router outlets.
- Sharable data is accessible through user data services.
- Created ts (type script) files for every component to implement the functionality.
- Imported plugins for making chats and api connections
- Create the admin portal for the chatbot, through which admin can see all the chat logs of all the users.
- Admin portal is divided into four sections
 1. Landing home page
 2. Dashboard
 3. Chat logs
 4. Statistics
 5. Priority messages
 6. Profile page

2.2.2 Node Runtime environment: Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

- nodejs provide the non-blocking I/O
- Initially JavaScript is restricted to client-side programming only which is run by the browser only but nodejs changes everything about the JavaScript giving JavaScript power of programming both for client and server side.
- Node js supports npm packages which attracts a greater number of developers to develop packages for the nodejs.

2.2.2.1 Use of nodejs and Koa in Project:

- By using nodejs we made the api for Chabot application

- List of api's
 - Login – which made available by running the nodejs and koa server and accepts get and posts requests from the angular.
 - Register – Which is used to register the user, The user data is taken from the register form which is made by using angular, and the data is post to the api url .
 - Chat logs – This api returns the chat logs of user.

2.2.3 Elastic search and kibana

- Elastic search is no sql database
- While indexing the documents it offers many different options for the developer as compared to other sql databases.
- Kibana is environment for writing and executing the elastic search queries

2.2.3.1 Use of elastic search in our project:

- Aggregations concept of elastic search mainly bucket aggregation is used to index the document indexed as a number of groups based in the criteria.
- While users typing the message in the text box which is developed by the angular ,elastic search server triggered with query every single second to get the best suited suggestions to the user and display that suggestions as an pop attached window which is very helpful to the chatbot user .

2.2.4 Mongo dB and robo3t:

- Mongo dB is an no sql database
- Robo3t is the environment in which we can write the query and see all the documents and collections.

2.2.4.1 Use of MongoDB in our project:

- Answer, questions, chat logs all are stored in the MongoDB.
- Admin data also stored in MongoDB
- All the APIs which are designed by using node js used to store the information in MongoDB only by using mongoose package.

2.2.5 Anaconda: Anaconda come with huge number of packages the two most important packages which our team used in the project is Spacy and Rasa.

2.2.5.1 About Spacy

- Spacy package is NLP package is mostly used for pre-processing of the text like
 1. Tokenization
 2. Parts of Speech Tagging
 3. Dependency Parsing
 4. Lemmatization
 5. Named entity detection

2.2.5.1.1 Tokenization:

- Segmenting text into words, punctuations marks etc.
- There are different kinds of tokenizers that are supported by spacy comity, some of them are listed below.
 1. White space tokenizer
 - The `whitespace` tokenizer divides text into terms whenever it encounters any whitespace character.
 2. Standard tokenizer
 - The `standard` tokenizer divides text into terms on word boundaries, as defined by the Unicode Text Segmentation algorithm. It removes most punctuation symbols. It is the best choice for most languages.
 3. Letter tokenizer
 - The `letter` tokenizer divides text into terms whenever it encounters a character which is not a letter.
 4. Lowercase tokenizer etc
 - The `lowercase` tokenizer, like the `letter` tokenizer, divides text into terms whenever it encounters a character which is not a letter, but it also lowercases all terms.

2.2.5.1.2 Parts of Speech Tagging

- Assigning word types to tokens, like verb or noun.
- Spacy have three kind of models en_core_web_sm, en_core_web_md, en_core_web_lg the size of the models increases in the increases order the main difference between each model is the features provided.
- Pos tagging is the process of identifying the parts of speech of the particular word this tag is identified by the model which is previously trained on the large amount of the data
- These parts of speech tag is very useful for further text pre-processing steps.

2.2.5.1.3 Dependency Parsing

- Assigning syntactic dependency labels, describing the relations between individual tokens, like subject or object.

```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple is looking at buying U.K. startup for $1 billion")
for token in doc:
    print(token.text, token.pos_, token.dep_)
```

output:

```
Apple PROPN nsubj
is AUX aux
looking VERB ROOT
at ADP prep
buying VERB pcomp
U.K. PROPN compound
startup NOUN dobj
for ADP prep
$ SYM quantmod
1 NUM compound
billion NUM pobj
```

2.2.5.1.4 Lemmatization:

- Assigning the base forms of words. For example, the lemma of “was” is “be”, and the lemma of “rats” is “rat”.

```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple is looking at buying U.K. startup for $1 billion")

for token in doc:
    print(token.text, token.lemma_, token.pos_, token.tag_, token.dep_,
          token.shape_, token.is_alpha, token.is_stop)
```

Output

TEXT	LEMMA	POS	TAG	DEP	SHAPE	ALPHA	STOP
Apple	apple	PROPN	NNP	nsubj	Xxxxx	True	False
is	be	VERB	VBZ	aux	xx	True	True
looking	look	VERB	VBG	ROOT	xxxx	True	False
at	at	ADP	IN	prep	xx	True	True
buying	buy	VERB	VBG	pcomp	xxxx	True	False
U.K.	u.k.	PROPN	NNP	compound	X.X.	False	False
startup	startup	NOUN	NN	dobj	xxxx	True	False
for	for	ADP	IN	prep	xxx	True	True
\$	\$	SYM	\$	quantmod	\$	False	False
1	1	NUM	CD	compound	d	False	False
billion	billion	NUM	CD	pobj	xxxx	True	False

2.2.5.1.5 Named Entity Detection:

A named entity is a “real-world object” that’s assigned a name – for example, a person, a country, a product or a book title. spaCy can recognize various types of named entities in a document, by asking the model for a prediction. Because models are statistical and strongly depend on the examples they were trained on, this doesn’t always work perfectly and might need some tuning later, depending on your use case.

2.2.5.1.6 Use of Spacy in Project:

- Spacy is used to in the pipeline of the rasa.
- All the pre processing work will be handled by the spacy.
- All the pre-processing steps are mentioned above.
- We should need to configure depending upon our need.

2.2.5.2 About Rasa:

- Rasa is the NLU (Natural Language Understanding) package which is mainly used to find the intent, entity extraction and find the action that need to take.

2.2.5.2.1 File structure of Rasa:

- Domain.yml – this file contains all the list of intents, entities, slots, templates, forms and etc, these are only declared in this section.
- Data/core/stories.md – This is the core data which on our model will be trained , this consists of blocks of stories each block consists of
Story name
*intent 1
-action need to be taken
*intent 2
- action need to be taken
- Data/nlu/nlu.md – This file consists of all the sentences that our chatbot need to face in future in simple words these are the questions which chatbot handles and answers, this file is also used to train our model at training state.
The main use of this file in our project is it gives intent classification ability to our rasa model.
- Actions.py – this is file where we write code to our customizable actions, like switching on the electrical appliances, booking ticket, and etc..
- Endpoints.yml – this is the files where we configure the endpoints like for example mongo dB endpoint which is tracker store which is used to store the chat logs of all the users directly after the chat, and we can also configure our customize actions server in this file only.

- Config.yml – This is the important file to the rasa project, this contains the pipeline of the project, pipeline means what are the different stages that the user inputted message goes before answered by bot.

2.2.5.2.2 RASA FUNCTIONALITIES

2.2.5.2.2.1 COMMANDS FOR RUNNING THE RASA SERVER:

- `rasa run -vv --endpoints endpoints actions`
- Used to run the custom action server and end points file also.
- We can configure the tracker store in the endpoints file to store the chat logs in the preferred Database
- e.g. Mongo db.

```
tracker_store:
  type: mongod
  url: mongodb://localhost:27017
  db: rasa
```

Above is the configuration to store the chat logs in the mongo db this code should be present in **endpoints.yml** file of root directory of rasa project.

- `rasa run -m models --enable-api --cors "*" --debug`
 - This is used to run the rasa server, it enables many api's by default.
 - **LIST OF ENABLED API**

```
2019-11-29 00:53:54 DEBUG    rasa.core.utils - Available web server routes:
  /conversations/<conversation_id>/messages      POST
    add_message
  /conversations/<conversation_id>/tracker/events  POST
    append_events
  /webhooks/rasa                                GET
    custom_webhook_RasaChatInput.health
  /webhooks/rasa/webhook                        POST
    custom_webhook_RasaChatInput.receive
  /webhooks/rest                                GET
    custom_webhook_RestInput.health
  /webhooks/rest/webhook                       POST
    custom_webhook_RestInput.receive
  /model/test/intents                          POST
    evaluate_intents
  /model/test/stories                         POST
    evaluate_stories
  /conversations/<conversation_id>/execute       POST
    execute_action
```

```

▪ /domain GET
  get_domain
▪ / GET
  hello
▪ /model PUT
  load_model
▪ /model/parse POST
  parse
▪ /conversations/<conversation_id>/predict POST
  predict
▪ /conversations/<conversation_id>/tracker/events PUT
  replace_events
▪ /conversations/<conversation_id>/story GET
  retrieve_story
▪ /conversations/<conversation_id>/tracker GET
  retrieve_tracker
▪ /status GET
  status
▪ /model/predict POST
  tracker_predict
▪ /model/train POST
  train
▪ /model DELETE
  unload_model
▪ /version GET
  version
2019-11-29 00:53:54 INFO root - Starting Rasa server on http://localhost:5005

```

▪ SOME USES OF THIS API:

- We can train the model the model
 - We can test the model and so many.
- `rasa interactive -vv --endpoints endpoints.yml`
- Which enables interactive mode by which we can chat with chat bot and update its prediction.

2.2.5.2.2.2 CONVERTING MD FILES TO JSON AND VICE VERSA:

▪ MD TO JSON

- `python -m rasa_nlu.convert --data_file XYZ --out_file ABC --format json`

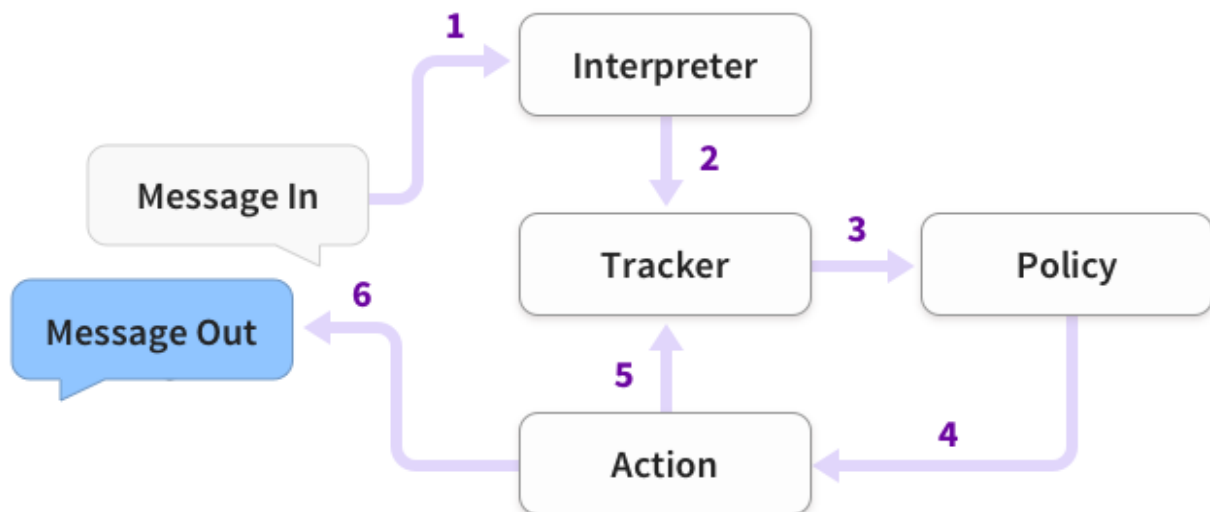
▪ JSON TO MD

- `python -m rasa_nlu.convert --data_file XYZ --out_file ABC --format md`

2.2.5.2.2.3 NLU.MD NLU DATA FORMAT:

- **INTENTS:**
 - Which declares the different types of intents.
 - Each intent will be associated with its sentences to train the model.
- **Synonym:**
 - This section is used to handle the synonym, mostly it is used to save the most wrong spelled word so that our model understands.
- **Lookup:**
 - This section handles the entities and there values.
- **Regex:**
 - This section is specially for handling the regular expression.

2.2.5.2.2.4 RASA ARCHITECTURE:



The Steps are:

1. The message is received and passed to an **Interpreter**, which converts it into a dictionary including the original text, the intent, and any entities that were found. This part is handled by NLU.

2. The Tracker is the object which keeps track of conversation state. It receives the info that a new message has come in.
3. The policy receives the current state of the tracker.
4. The policy chooses which action to take next.
5. The chosen action is logged by the tracker.
6. A response is sent to the user.

SLOTS:

Slots are your bot's memory. They act as a key-value store which can be used to store information the user provided (e.g their home city) as well as information gathered about the outside world (e.g. the result of a database query).

2.2.5.2.2.5 IMPORTANT LINKS TO DIFFERENT MODULES:

- **Hunshell**
 - <https://github.com/blatinier/pyhunspell> (Link to the library)
 - Good for suggestions and spell checking
 - We can add our own words to the hunshell model which will be suggested also.

CHAPTER 3: PROJECT RESEARCH WORK

3.1 PANDORABOTS PLATFORM:

3.1.1 FEATURES:

This section will contains features that are provided by Pandorabots platform.

3.1.1.1 LOG REVIEW:

- Chat logs are displayed dating back 30 days and available for download, if we require longer or custom storage, dashboards, or analytics we need to contact pandorabots.
- We can download our chat log anytime when we require.
- Priority logs are very useful for analyzing the not answered messages.
 1. This is the separate section in the logs section which consists of unanswered question and the bot given answer to that questions.
 2. We can directly edit the answer for the unanswered questions in that the section itself which is used by the bot for the next time when it encounters the same question.

3.1.1.2 DEPLOY:

- Contains so many Integrations and pandorabots api.
- List of integration channels
 1. Chat Widget Browser Integration
 2. Slack Bot
 3. Messenger Bot
 4. What's App Bot
 5. Twitter Bot
 6. Skype Bot
 7. We ChatBot
 8. Line Bot
 9. SMS Bot
 - 10.KIK Bot

- Pandorabots can develop and provide custom integrations for Enterprise Tier platform users.

3.1.1.3 CLUB HOUSE:

- The Clubhouse lets us beta test our bot-in-progress in a bot master only environment.
- In Internal Bot Directory or clubhouse our bot is been available to all the pandorabots account holders to chat.
- We can collect chat logs to review and improve your bot.

3.1.1.4 INDIVIDUAL BOT STATISTICS:

- The number of unread logs will be displayed next to our bot's name. Click on your bot's name to view other key stats

3.1.1.5 GLOBAL BOT STATISTICS:

- Usage statistics like total number of monthly interactions, clients, sessions, and the average interactions per session are displayed dating back 30 days.
-

3.1.1.6 BOT INDICATOR LIGHT:

- Green: Published without errors.
- Yellow: If Bot is in sandbox version it means not yet published
- Red: Compilation errors in AIML files

3.1.1.7 META DATA:

- Which says how the message is handled by the Bot.
- Trace option which gives us route of answering the message.

3.1.1.8 .BOT DOMAIN:

- .BOT is a new generic top-level domain (gTLD) from Amazon Registry Services.
- Currently these domains are available for anyone who owns ,operates or manages bots published using only these supported tools

1. Pandorabots

2. Amazon Lex
3. Dialog flow
4. Gupshup
5. Microsoft Bot Framework

3.1.1.9 AIML VERSION:

- Presently pandorabots does not support AIML 1.0 version, it now supports latest AIML 2.0 version.

3.1.1.10 LIMITS:

- The maximum file size for upload is 2MB.
- The Free Tier allows up to 2 bots.

3.1.1.11 BILLING:

- Billing cycle is monthly using UTC time. Charges are typically made between the 5th to 6th days of the month.

DIFFERENT KINDS OF PLANS:

	SANDBOX	DEVELOPER	PRO	ENTERPRISE
DEVELOPER SANDBOX	+	+	+	+
UNLIMITED SANDBOX MESSAGES	+	+	+	+
API ACCESS		+	+	+
CHAT WIDGET AND LANDING PRICE		+	+	+
3 RD PARTY CHANNELS		\$9/CHANNEL	+	+MORE
CHANNEL MESSAGES		10,000/MONTH	100,000/MONTH	VARIES
ADDITIONAL CHANNEL MESSAGES		\$3/1000 MESSAGES	\$2/ 1000 MESSAGES	VARIES
SUPPORT	PUBLIC OFFICE HOURS	EMAIL	EMAIL,CHAT AND PHONE	SLA AND PREMIUM SUPPORT
CHATBOT LIBRARIES	OPEN SOURCE	OPEN SOURCE	OPEN SOURCE	MITSUKU MODULE
PRICE	FREE	\$19/MONTH	\$199/MONTH	NOT AVAILABLE



-THIS REPRESENTS INCLUDED FEATURE

3.1.1.12 MITSUKU MODULE: These are the modules which are used to develop MITSUKU a four-time winner of the Loebner Prize Turing Test, is the world's best conversational chatbot.

Message: A “message” is one input/output interaction between a client (the person chatting with your bot)

3.1.2 FILES REQUIRED:

This sections will contain system requirements which are useful for developer's team to develop chatbot on PANDORABOT platform.

3.1.2.1 SET_FILES:

- Location : /Sets/ (In Pandorabots file explorer)
- PANDORABOTS Set files contains set of values which should be unique. (Because these are used as keys in mapping objects in MAP_FILES.)

3.1.2.2 MAP_FILES:

- Location : /Maps/ (In Pandorabots file explorer)
- PANDORABOTS Map files contains mapping which are [key, value] pairs.(Here the key values are from the Set files)

3.1.2.3 SUBSTITUTION_FILES:

- Location : /Maps/ (In Pandorabots file explorer)
- PANDORABOTS Substitution files consists of substitution of set of phrases.

3.1.2.4 UDC.AIML - ULTIMATE DEFAULT CATEGORY:

- Location : /AIML/ (In Pandorabots file explorer)
- Use :
 1. This file is used by the compiler when the client message did not match any pattern in our data.
 2. Priority logs consists of logs which our chat bot did not find answer, in that case chat bot give the message which is specified in UDC.AIML file.

3.2 PYTHON NATURAL LANGUAGE PROCESSING AND UNDERSTANDING MODULES RESEARCH

3.2.1 PROS OF EACH MODULE

MODULE NAME	PROS
NLTK	<ul style="list-style-type: none">• Supporting tasks such a classification, tokenization, stemming, and tagging, parsing, and semantic reasoning.• Supports Porter stemmer, Lancaster stemmer, Word Net stemmer,NonEnglishstemmers,Snowball stemmers(supports for 17 languages)• Provides easy-to-use interfaces to over 50 corpora and lexical resources such as Word Net.• We can take more advantage of concepts behind NLP by referring the book https://www.nltk.org/book/
SPACY	<ul style="list-style-type: none">• spaCy is designed specifically for production use.• It is actually developed by Cython which makes it faster.• Supported Features Tokenization, Part-of-speech (POS) Tagging, Dependency Parsing, Lemmatization, Sentence Boundary Detection (SBD), Named Entity Recognition (NER),Entity Linking (EL), Rule-based Matching, Training, Serialization, Integrated word vectors.
TEXT BLOB	<ul style="list-style-type: none">• It is built on the shoulders of NLTK and Pattern, therefore making it Simple for beginners by providing an intuitive interface to NLTK.•It provides language translation and detection which is powered by Google Translate (not provided with Spacy)

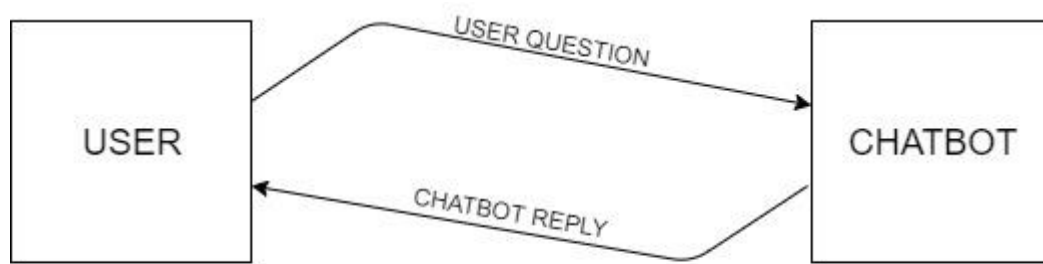
CoreNLP	<ul style="list-style-type: none"> •The library is really fast and works well in product development Environments. Moreover, some of CoreNLP components can be Integrated with NLTK which is bound to boost the efficiency. • It is actually written in Java. Still, it's equipped with wrappers for many different languages, including Python
----------------	---

3.2.2 CONS OF EACH MODULE

PACKAGE	CONS
NLTK	<ul style="list-style-type: none"> • Difficult to use for Production level . •To effectively use the toolkit, we really need to understand the concepts behind it, which can be a lot to absorb if we're just starting with NLP.
SPACY	It supports the smallest number of languages (Eleven).
TEXTBLOB	<ul style="list-style-type: none"> •It is little slower in the comparison to spacy but faster than NLTK.(Spacy > TextBlob > NLTK) •It does not provide features like dependency parsing, word vectors etc. which is provided by spacy

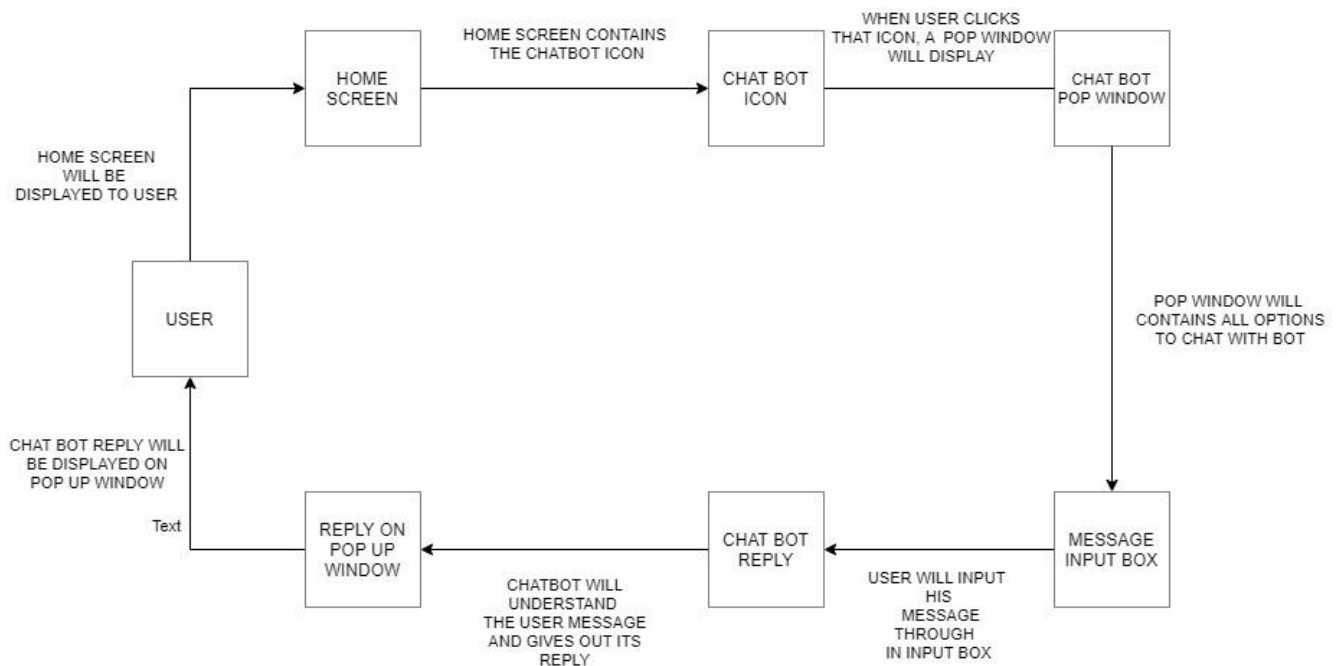
CHAPTER 4: DFD SECTION:

4.1 LEVEL-0 DATA FLOW DIAGRAM



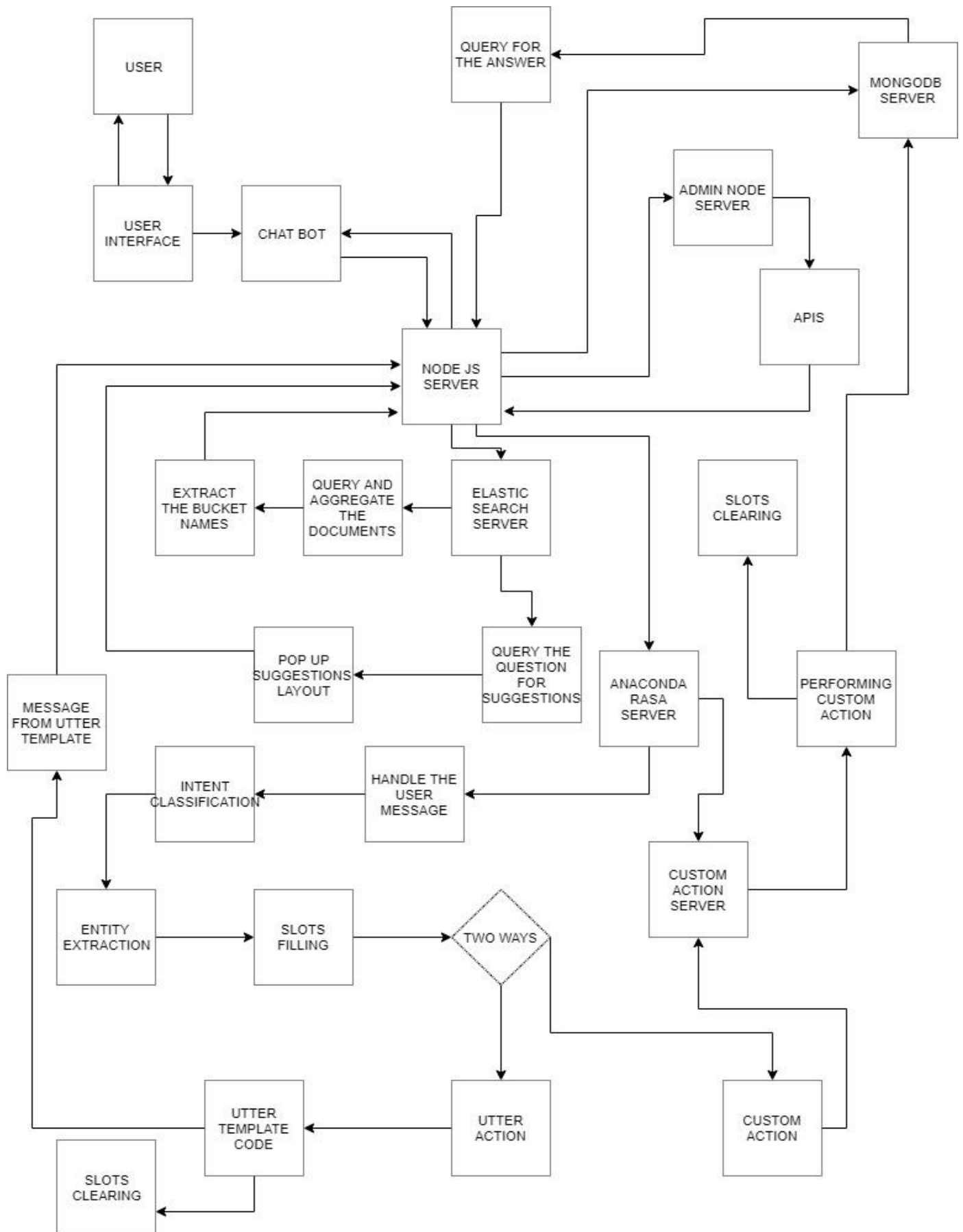
LEVEL -0 CONTEXT LEVEL DATA FLOW DIAGRAM

4.2 LEVEL-1 DATA FLOW DIAGRAM



LEVEL 1 DFD

4.3 LEVEL-2 DATA FLOW DIAGRAM



CHAPTER 5: CONCLUSION

From this Internship I had experienced the live project, and how the production of particular software takes place from research to implementation. While coming to technologies I worked on RASA, SPACY python machine learning modules, Elasticsearch no sql database and nodejs web development technologies for creating api's.

CHAPTER 6: Bibliography

- <https://stackoverflow.com>
- <https://spacy.io/usage>
- <https://rasa.com/docs/getting-started/>
- <https://rasa.com/docs/rasa/>
- <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>
- <https://nodejs.org/en/docs/>