

Software Design Document

Pineapple Chatbot

Prepared by:

Kuong Thong, Enrique Castillo, Julio Aguilar, Jose Lopez, Yilin Ruan

CSULA ITS Department

Version History

Version	Date	Summary of Changes
apple	12/4/18	Added Introduction sections (1.1–1.2).
banana	12/5/18	Added early DFD diagrams and transferred initial design content.
cantaloupe	12/6/18	Added database design and sections 8–9.1.
dates	12/7/18	Completed document draft and added validation sections.
elderberry	1/3/19	Updated DFDs to match current model.
farkleberry	3/13/19	Updated feature descriptions and removed outdated content.
grapefruit	3/22/19	Cleaned document structure and updated sections 5–7.
huckleberry	9/25/19	Updated crawling approach.
jackfruit	11/15/19	Removed Amazon Lex dependency.
kiwi	12/2/19	Updated models to reflect spaCy usage.

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Document Conventions	4
1.3	Intended Audience	4
1.4	System Overview	4
2	Design Considerations	4
2.1	Assumptions & Dependencies	4
2.2	Constraints	4
2.3	Goals	4
2.4	Development Methods	5
3	Architectural Strategy	5
4	System Architecture	5
4.1	Level 0 DFD	5
4.2	Level 1 DFD	5
5	Policies and Tactics	6
5.1	Tools & Technologies	6
5.2	Requirements Traceability	6
5.3	Testing Approach	6
5.4	Engineering Trade-offs	6
5.5	Coding Standards	6
5.6	Source Code Organization	6
5.7	System Build Instructions	6
6	Detailed System Design	6
6.1	Input Module	6
6.2	Preprocessor Module	7
6.3	Storage Module	7
6.4	Logic Module	7
6.5	Output Module	7

7 Lower-Level Component Design	7
7.1 Input Module Components	7
7.2 Output Module Components	7
7.3 Storage Module Components	7
7.4 Logic Module Components	7
8 Database Design	8
9 User Interface	8
9.1 Overview	8
9.2 Future Enhancements	8
10 Requirements Validation & Verification	8
11 Glossary	8
12 References	9

1 Introduction

1.1 Purpose

Pineapple Chatbot is an informational chatbot for a standalone web application that helps users find information related to the CSULA website. The chatbot processes user queries, responds with useful answers, and can receive user feedback.

1.2 Document Conventions

This document follows a structured SDD format with enumerated sections, module definitions, and design descriptions.

1.3 Intended Audience

This document is intended for developers, testers, designers, project managers, and documentation staff.

1.4 System Overview

The chatbot crawls university webpages, stores extracted information in a database, and uses natural-language processing to interpret user queries. It returns answers as text, links, or summaries.

2 Design Considerations

2.1 Assumptions & Dependencies

- Backend built using Python.
- Libraries: MySQL Connector, BeautifulSoup, nltk, ChatterBot, spaCy.
- Additional dependencies: NodeJS, React (front-end).

2.2 Constraints

- Early versions lack spell-check, web abstraction, and high-accuracy NLP.
- System quality depends on crawled data; vague queries may yield incorrect results.

2.3 Goals

- Provide helpful, conversational responses.
- Keep design simple (KISS principle).
- Provide detail without overwhelming users.
- Maintain high user satisfaction.

2.4 Development Methods

Pair programming and iterative testing were used to refine modules and improve accuracy.

3 Architectural Strategy

Python is used for crawling, NLP, and chatbot logic due to its open-source ecosystem. MySQL stores processed data because it integrates well with ChatterBot's SQL backends. The chatbot runs inside a web application built with React, which manages user input and output.

4 System Architecture

4.1 Level 0 DFD

Figure 1: DFD Level 0 for Pineapple Chatbot

4.2 Level 1 DFD

Figure 2: DFD Level 1 for Pineapple Chatbot

Modules

1. Input Module – Collects user inputs.
2. Preprocessor Module – Cleans and normalizes text.
3. Logic Module – Identifies user intent and selects a response.
4. Storage Module – Stores and retrieves crawled data.
5. Output Module – Formats and returns the answer to the user.

Data Flow Summary

User query → Preprocessing → Intent recognition → Keyword lookup → Data retrieval → Response → User.

5 Policies and Tactics

5.1 Tools & Technologies

ChatterBot, React, Python, MySQL.

5.2 Requirements Traceability

Crawled data is backed up regularly.

5.3 Testing Approach

User testing and QA validate accuracy and relevance of chatbot responses.

5.4 Engineering Trade-offs

Campus websites store large amounts of information; storing more data increases coverage but slows query speed.

5.5 Coding Standards

Standard Python and JavaScript conventions are followed; version control is used throughout.

5.6 Source Code Organization

- **React App:** User interface
- **Python Backend:** Chatbot logic and server
- **Database:** MySQL storage

5.7 System Build Instructions

1. Start VM
2. Run React frontend (npm start)
3. Run Python backend (python manage.py runserver)
4. Access chatbot via CSULA server interface

6 Detailed System Design

6.1 Input Module

Responsibilities: Convert user input into Statement objects. Constraints: Must handle text and minimal structured formats. Interactions: Passes processed input to Logic Module.

6.2 Preprocessor Module

Responsibilities: Clean whitespace, remove HTML artifacts, convert Unicode to ASCII. Limitations: Must preserve important keywords.

6.3 Storage Module

Responsibilities: Store responses and crawled data. Technologies: MySQL; SQLStorageAdapter recommended. Limitations: Requires credentials; large datasets may slow queries.

6.4 Logic Module

Responsibilities include:

- Recognize intent
- Identify keywords
- Select best-matched response
- Optionally evaluate time-based or mathematical queries

6.5 Output Module

Responsibilities: Format and return final response to user through React UI.

7 Lower-Level Component Design

7.1 Input Module Components

- Creates Statement objects from input
- Accepts text, dictionary, or Statement formats

7.2 Output Module Components

- Returns Statement objects as responses
- Reads from Storage Module

7.3 Storage Module Components

- CRUD operations for stored responses
- Requires DB credentials
- SQL module is primary implementation

7.4 Logic Module Components

- Compares user statements with known statements
- Uses similarity thresholds and exclusion rules

- Selects closest-matching response

8 Database Design

A MySQL database stores the crawled website data.

Key Table: termUrlKeywords

Approximately 300,000 records.

- Columns: `term`, `URL`, `keywords`

Additional tables contain faculty information and classifier labels.

Database enables fast keyword matching, URL lookup, and long-term storage.

9 User Interface

9.1 Overview

- Simple chat-style interface built with React
- Input box for queries
- Response window for answers and links

9.2 Future Enhancements

- Richer UI components
- Improved accessibility

10 Requirements Validation & Verification

- Unit tests for crawler, NLP, and storage
- Integration tests for full query flow
- User testing for clarity and accuracy

11 Glossary

- **Statement:** Processed user input
- **Intent:** Interpreted purpose of a query
- **Similarity Score:** Metric to select the closest matching response

12 References

Original Pineapple Chatbot documents, tutorials for Python NLP libraries, and MySQL documentation.