

Software Requirements Specification  
Pineapple Chatbot  
Version 3.0

Kuong Thong, Enrique Castillo, Julio Aguilar, Jose Lopez, Yilin Ruan  
CSULA ITS Department

December 10, 2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Purpose . . . . .	4
1.2	Intended Audience and Reading Suggestions . . . . .	4
1.3	Product Scope . . . . .	4
1.4	Definitions, Acronyms, and Abbreviations . . . . .	5
1.5	References . . . . .	5
<b>2</b>	<b>Overall Description</b>	<b>6</b>
2.1	Product Perspective . . . . .	6
2.2	Product Functions . . . . .	6
2.3	User Classes and Characteristics . . . . .	6
2.4	Operating Environment . . . . .	7
2.5	Design and Implementation Constraints . . . . .	7
2.6	User Documentation . . . . .	7
2.7	Assumptions and Dependencies . . . . .	7
<b>3</b>	<b>External Interface Requirements</b>	<b>8</b>
3.1	User Interfaces . . . . .	8
3.2	Hardware Interfaces . . . . .	8
3.3	Software Interfaces . . . . .	8
3.4	Communications Interfaces . . . . .	8
<b>4</b>	<b>Requirements Specification</b>	<b>9</b>
4.1	Functional Requirements . . . . .	9
4.2	External Interface Requirements . . . . .	10
4.3	Logical Database Requirements . . . . .	10
4.4	Design Constraints . . . . .	10
<b>5</b>	<b>Other Nonfunctional Requirements</b>	<b>11</b>
5.1	Performance Requirements . . . . .	11
5.2	Safety Requirements . . . . .	11
5.3	Security Requirements . . . . .	11
5.4	Software Quality Attributes . . . . .	11
5.5	Business Rules . . . . .	11
<b>6</b>	<b>Other Requirements</b>	<b>11</b>

## Revision History

Name	Date	Description	Version
Initial Draft	10/15/25	Initial skeleton draft created for Snapshot 1	1.0
Expanded Draft	11/5/25	Added detailed requirements, Bayesian model, database structure, UI details	2.0
Final Release	12/10/25	Completed all sections, refined requirements, added glossary and references	3.0

# 1 Introduction

## 1.1 Purpose

The Pineapple Chatbot provides general information extracted from pages across the `calstatela.edu` domain. Version 3.0 expands functionality with Bayesian classification, improved NLP accuracy, enhanced UI design, and structured SQL database integration.

## 1.2 Intended Audience and Reading Suggestions

This document is intended for developers, testers, marketing staff, project managers, and documentation authors. Readers seeking to understand how the chatbot functions should read the entire document.

Testers may refer to the requirement list and system flow. Investors may refer to high-level descriptions and system scope. Customers can reference the chatbot capabilities.

## 1.3 Product Scope

The Pineapple Chatbot uses NLP and statistical models to interpret user queries and return the most relevant webpages or extracted information.

### User Capabilities

- Ask CSULA-related questions
- Receive text answers or direct website links
- Obtain professor emails when recognized
- View responses in a chat-style interface

### System Capabilities

- Entity extraction using spaCy
- Bayesian category scoring for unclear queries
- Keyword extraction and stop-word removal
- Ranked URL selection using hybrid scoring
- SQL database storage of categorized URLs
- Response generation and formatting

## 1.4 Definitions, Acronyms, and Abbreviations

- **NLP** – Natural Language Processing
- **spaCy** – Python NLP library used for entity recognition
- **Entity** – PERSON, ORG, LOCATION, etc.
- **Intent** – User's goal (e.g., find professor email)
- **Bayesian Model** – Probability-based classifier
- **URL Scoring** – Algorithm selecting the best info source
- **CSULA** – California State University, Los Angeles

## 1.5 References

- spaCy Documentation: <https://spacy.io>
- CSULA Website: <https://www.calstatela.edu>
- Python Standard Library
- Naive Bayes Classification Research

## 2 Overall Description

### 2.1 Product Perspective

Pineapple Chatbot is a standalone web application supported by:

- A chat-based frontend UI
- A Python backend with NLP processing
- SQL database containing categorized URLs

### 2.2 Product Functions

#### User Functions

- Submit questions via a text input field
- View chatbot replies in styled chat bubbles
- Click links provided in chatbot messages
- Receive professor-related details such as emails

#### System Functions

- Extract entities using spaCy
- Identify intent
- Reduce keywords and remove stop-words
- Classify using Bayesian scoring
- Retrieve URLs using SQL database queries
- Rank URLs using hybrid scoring
- Format and send chatbot responses

### 2.3 User Classes and Characteristics

**General Users:** Students or visitors seeking campus information **Developers:** Maintain backend code, NLP pipeline, database, and UI logic **Testers:** Validate reliability, accuracy, and UI behavior

## **2.4 Operating Environment**

- Python 3.x
- spaCy NLP models
- Flask or Node.js backend
- SQL database (MySQL or SQLite)
- Modern browsers (Chrome, Firefox, Edge)
- Optional CSULA VPN for internal hosting

## **2.5 Design and Implementation Constraints**

- English-only input supported
- NLP accuracy limited by spaCy model
- UI not optimized for mobile
- Link dataset must be updated periodically
- Response speed dependent on database size

## **2.6 User Documentation**

No user guides yet; chatbot will be made public on the CSULA website.

## **2.7 Assumptions and Dependencies**

- CSULA website structure remains stable
- spaCy NER continues to detect PERSON, ORG, etc.
- Database remains reachable
- Questions are written in general English

## **3 External Interface Requirements**

### **3.1 User Interfaces**

UI Components include:

- Chat window
- Scrollable message history
- Text input field with send button
- Styled chat bubbles
- Highlighted, clickable links

### **3.2 Hardware Interfaces**

Standard laptop or desktop with internet access.

### **3.3 Software Interfaces**

- spaCy NLP library
- SQL database with URLs and categories
- Backend REST API returning JSON
- Python NLP service

### **3.4 Communications Interfaces**

Internet access required. VPN may be necessary for internal CSULA hosting. Not compatible with Internet Explorer.

## **4 Requirements Specification**

### **4.1 Functional Requirements**

#### **1. Input Module**

- 1.1 System shall accept user text input.
- 1.2 System should allow user feedback on responses.
- 1.3 System may support multi-sentence input.

#### **2. Output Module**

- 2.1 System shall display the user's question.
- 2.2 System shall include clickable links in responses.
- 2.3 System shall display an error for unknown queries.

#### **3. Logic Module**

- 3.1 Parse keywords from user input.
- 3.2 Recognize intent using spaCy.
- 3.3 Respond appropriately to unclear or unintelligible inputs.
- 3.4 Send keywords and intent to the storage module.
- 3.5 Receive possible URLs from storage module.
- 3.6 Select the most relevant URL using scoring.
- 3.7 Send entities, intent, and keywords to data extraction module.

#### **4. Data Extraction Module**

- 4.1 Extract information using entities, keywords, and intent.
- 4.2 Send extracted information to the output module.
- 4.3 Parse and structure text; label using spaCy.

#### **5. Storage Module**

- 5.1 Send SQL queries.
- 5.2 Receive responses from database.
- 5.3 Return information to logic module.

## 4.2 External Interface Requirements

- Input must support standard English text.
- URL responses must be HTML-safe.
- Chat must support sequential messaging.

## 4.3 Logical Database Requirements

**Database Tables:**

### TABLE urls

- id
- url
- keywords
- category

### TABLE categories

- category\_id
- name
- keyword\_distribution

### Requirements

- Supports full-text search
- Allows batch insertion
- Stores at least 500+ URL entries

## 4.4 Design Constraints

- spaCy model loads on startup
- Query processing must finish within 3 seconds
- Category scoring must reach 80–90% accuracy

## 5 Other Nonfunctional Requirements

### 5.1 Performance Requirements

- System must respond within 1–3 seconds
- Database queries must complete under 200ms
- UI must remain responsive during processing

### 5.2 Safety Requirements

Feedback system must keep reports anonymous to protect user identity. Chatbot must avoid exposing sensitive data or giving unsafe guidance.

### 5.3 Security Requirements

- Backend must sanitize all inputs
- Database requires authentication
- Responses must be delivered via HTTPS (future deployment)
- Admin interface (future) requires login

### 5.4 Software Quality Attributes

- **Usability:** Simple chat interface
- **Reliability:** NLP results consistent per input
- **Maintainability:** Modular architecture
- **Scalability:** Database supports growth

### 5.5 Business Rules

- Bot must prioritize official CSULA information
- Answers must originate from the calstatela.edu domain
- Professor data must be accurate and up-to-date

## 6 Other Requirements

Future possibilities include multilingual support and CSULA API integration.

**FINAL VERSION 3.0 COMPLETE**