

Software Requirements Specification

Pineapple Chatbot

Version 2.0

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

March 22, 2020

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 4 |
| 1.1 | Purpose | 4 |
| 1.2 | Intended Audience and Reading Suggestions | 4 |
| 1.3 | Product Scope | 4 |
| 1.4 | Definitions, Acronyms, and Abbreviations | 4 |
| 1.5 | References | 5 |
| 2 | Overall Description | 6 |
| 2.1 | Product Perspective | 6 |
| 2.2 | Product Functions | 6 |
| 2.3 | User Classes and Characteristics | 6 |
| 2.4 | Operating Environment | 6 |
| 2.5 | Design and Implementation Constraints | 7 |
| 2.6 | User Documentation | 7 |
| 2.7 | Assumptions and Dependencies | 7 |
| 3 | External Interface Requirements | 8 |
| 3.1 | User Interfaces | 8 |
| 3.2 | Hardware Interfaces | 8 |
| 3.3 | Software Interfaces | 8 |
| 3.4 | Communications Interfaces | 8 |
| 4 | Requirements Specification | 9 |
| 4.1 | Functional Requirements | 9 |
| 4.2 | External Interface Requirements | 10 |
| 4.3 | Logical Database Requirements | 10 |
| 4.4 | Design Constraints | 10 |
| 5 | Other Nonfunctional Requirements | 11 |
| 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 |
| 6 | Other Requirements | 11 |

Revision History

| Name | Date | Description | Version |
|----------------|------|--|---------|
| Initial Draft | TBD | Initial skeleton draft created for Snapshot 1 | 1.0 |
| Expanded Draft | TBD | Added detailed requirements, Bayesian model, database structure, UI, updated constraints | 2.0 |

1 Introduction

1.1 Purpose

The Pineapple Chatbot provides general information extracted from pages across the `calstatela.edu` domain. Version 2.0 expands the chatbot's functionality by adding Bayesian classification, improved NLP accuracy, early UI elements, and a transition from static links to a structured database.

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 works should read the entire document from top to bottom.

Testers should review the requirement list and data flow diagrams. Customers may review the capabilities of the Pineapple Chatbot. Investors may examine the product description and future goals.

1.3 Product Scope

The Pineapple Chatbot uses NLP techniques (primarily spaCy) and simple statistical models to interpret user intent and provide the most relevant webpage or text-based response.

Version 2.0 includes:

- Entity extraction using spaCy
- Intent identification
- Expanded link scoring model (Bayesian + keyword scoring)
- Early-stage user interface
- Integration with a preliminary URL database
- Improved handling of unclear queries
- Queueing of user inputs

1.4 Definitions, Acronyms, and Abbreviations

- **Bayesian Model** – Statistical classifier comparing keywords with category distributions.
- **Keyword Reduction** – Removing stop-words to isolate meaningful terms.
- **Category Boosting** – Increasing link scoring weight based on category match.
- **NER** – Named Entity Recognition.

1.5 References

- [spaCy Documentation](#)
- [Python Standard Library](#)
- [Naive Bayes text classification examples](#)
- [CSULA website structure and categories](#)

2 Overall Description

2.1 Product Perspective

Pineapple Chatbot is a standalone web-based chatbot connected to backend NLP models. It currently runs locally with minimal UI; future versions will use a hosted deployment.

2.2 Product Functions

User Functions

- Submit questions through the web interface
- Receive chatbot replies through chat bubbles
- Receive links or direct answers for professor-related queries

System Functions (New)

- Bayesian classification of queries
- Keyword extraction and stop-word filtering
- Category matching
- URL ranking using hybrid scoring
- URL storage and retrieval from SQL database

2.3 User Classes and Characteristics

General Users: Students using the chatbot for quick campus resource lookup. **Developers:** Maintain NLP code, database structure, and UI logic. **Testers:** Validate classification accuracy and link relevance; create TestRail cases.

2.4 Operating Environment

- Python 3.x
- spaCy NLP models
- Web server (Node, Flask)
- MySQL or SQLite database
- Chrome, Firefox, Edge

2.5 Design and Implementation Constraints

- English-only supported
- Entity accuracy limited by spaCy model
- Website not mobile-optimized
- VPN may be required for hosting
- Link dataset must be updated periodically
- Response time depends on DB + model ranking

2.6 User Documentation

No user manuals yet. Chatbot will be published on the CSULA website.

2.7 Assumptions and Dependencies

- CSULA website structure remains consistent
- spaCy NER model can detect PERSON, ORG, etc.
- Database is reachable
- User questions use general English grammar

3 External Interface Requirements

3.1 User Interfaces

Version 2.0 includes a basic web UI:

Components:

- Text input field
- Send button
- Scrollable chat window
- Chat bubbles

Behavior:

- Auto-scroll
- Loading indicator
- Responses may include text + clickable links

3.2 Hardware Interfaces

- Standard laptop/desktop
- Internet access

3.3 Software Interfaces

- spaCy NLP library
- MySQL database
- REST API returning JSON
- Python NLP backend

3.4 Communications Interfaces

Internet access required. VPN may be required. Internet Explorer is not supported (Chrome, Firefox, Edge recommended).

4 Requirements Specification

4.1 Functional Requirements

1. Input Module

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

2. Output Module

- 2.1 Display user's question back to them.
- 2.2 Include a clickable link where available.
- 2.3 Display error message for unknown queries.

3. Logic Module

- 3.1 Parse user input for keywords.
- 3.2 Recognize intent using spaCy.
- 3.3 Respond appropriately even to unclear input.
- 3.4 Send keywords and intent to storage module.
- 3.5 Receive list of URLs from storage module.
- 3.6 Determine the most relevant URL.
- 3.7 Send entity, intent, keywords to data extraction module.

4. Data Extraction Module

- 4.1 Extract correct information from URLs.
- 4.2 Send extracted information to output module.
- 4.3 Parse and organize text; use spaCy for labeling.

5. Storage Module

- 5.1 Send database queries.
- 5.2 Receive responses.
- 5.3 Return data to logic module.

4.2 External Interface Requirements

- Web input must support English text.
- URL responses must be HTML-safe.
- Chat must support sequential message queues.

4.3 Logical Database Requirements

Database Tables:

- **urls**
 - id
 - url
 - keywords
 - category
- **categories**
 - category_id
 - name
 - keyword_distribution

Requirements:

- Full-text search support
- Batch inserts allowed
- At least 500+ URL entries

4.4 Design Constraints

- spaCy model loads at startup
- Queries must finish under 3 seconds
- Category scoring accuracy 80–90%

5 Other Nonfunctional Requirements

5.1 Performance Requirements

- System response time: 1–3 seconds
- DB query time: under 200ms
- UI remains responsive during backend calls

5.2 Safety Requirements

Feedback system must protect user identity. Chatbot must not expose sensitive information or unsafe guidance.

5.3 Security Requirements

- Sanitize all input
- DB requires authentication
- HTTPS required once deployed
- Admin interface requires login

5.4 Software Quality Attributes

- Usability: Simple chat interface
- Reliability: Consistent NLP responses
- Maintainability: Modular architecture
- Scalability: Database expandable

5.5 Business Rules

- Bot must prioritize official CSULA info
- All answers must come from `calstatela.edu`
- Professor data must remain accurate

6 Other Requirements

Not applicable.

END OF VERSION 2.0