High Level Design (HLD)

Bankbot

Document Version Control

Date Issued	Version	Description	Author
13/07/2023	1	HLD – V1.0	Diana Laveena DSouza

Contents

Abstract

In today's world, online banking has become increasingly automated, but many people still encounter various issues during their online banking experience. Additionally, some individuals may not be aware of the correct safety measures to protect their accounts. To assist customers with their queries and provide guidance on online banking safety, a bot can be developed.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions before coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - ♦ Security
    - ♦ Reliability
    - ♦ Maintainability
    - ♦ Portability
    - ♦ Reusability
    - ♦ Application compatibility
    - ♦ Resource utilization
    - ♦ Serviceability

1.2 Scope

The HLD documentation presents the system's structure, including the database architecture, application architecture, application flow and technology architecture. The HLD uses non-technical to mildly-technical terms, which should be understandable to the system's administrators.

1.3 Definitions

Term	***Description***
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment

2 General Description

2.1 Product Perspective

To create a bot mobile app which can give answers to all your banking related queries. Keep you updated with your transaction activity so any kind fraud can be avoided. Also give you alert signs if find any suspicious activity in your account.

2.2 Problem Statement

We have to create a bot using Natural Language Processing and deep learning which will be able to give answers for various banking related issues like your bank id, transaction details, Security services, loan policies, your account status etc.

2.3 Data Requirements

Data requirements depend entirely on our problem statement.

- We need a data which contains customer queries, tags and responses.

We use the JSON file and exporting it to a CSV file for further use.

2.5 Tools used

Python programming language and frameworks such as NumPy, Pandas, and Tensorflow are used to build the whole model.

- PyCharm is used as IDE.
- For Text Preprocessing, we used NLTK and sci-kit learn.
- Mysql is used to retrieve, insert, delete and update the database.
- Front-end development is done using HTML.
- Java is the primary programming language used for Android app development. It is a versatile and widely adopted language with a large ecosystem of libraries, frameworks, and tools. Java provides the necessary features and capabilities to build robust and scalable Android applications. It is an object-oriented language that allows developers to write reusable and modular code. The Android Software Development Kit (SDK) provides a collection of tools and libraries specifically designed for Android development. The Android SDK includes the Java Development Kit (JDK), which allows developers to write, compile, and run Java code for Android apps. The SDK also provides a set of APIs (Application Programming Interfaces) that allow developers to interact with various device functionalities and services, such as camera, sensors, network, and more. XML (eXtensible Markup Language) is used in Android development to define the user interface (UI) layout of an app. Android uses a declarative approach for UI development, where XML files are used to describe the structure and appearance of the app's screens and views. XML provides a

clear separation between UI design and code logic, making it easier to maintain and update UI layouts. Additionally, XML is also used for managing app resources, such as strings, colors, dimensions, and styles. Android provides a UI framework based on XML and Java. XML is used to define the layout and appearance of UI elements, while Java code is used to handle UI interactions, event handling, and business logic. This separation of UI design and code allows for better code organization, reusability, and easier collaboration between designers and developers. Java is a widely supported and backward-compatible language, which means that Android apps written in Java can run on a wide range of Android devices, including older versions. This compatibility is essential for reaching a larger user base and ensuring that the app functions correctly across different Android versions and devices.

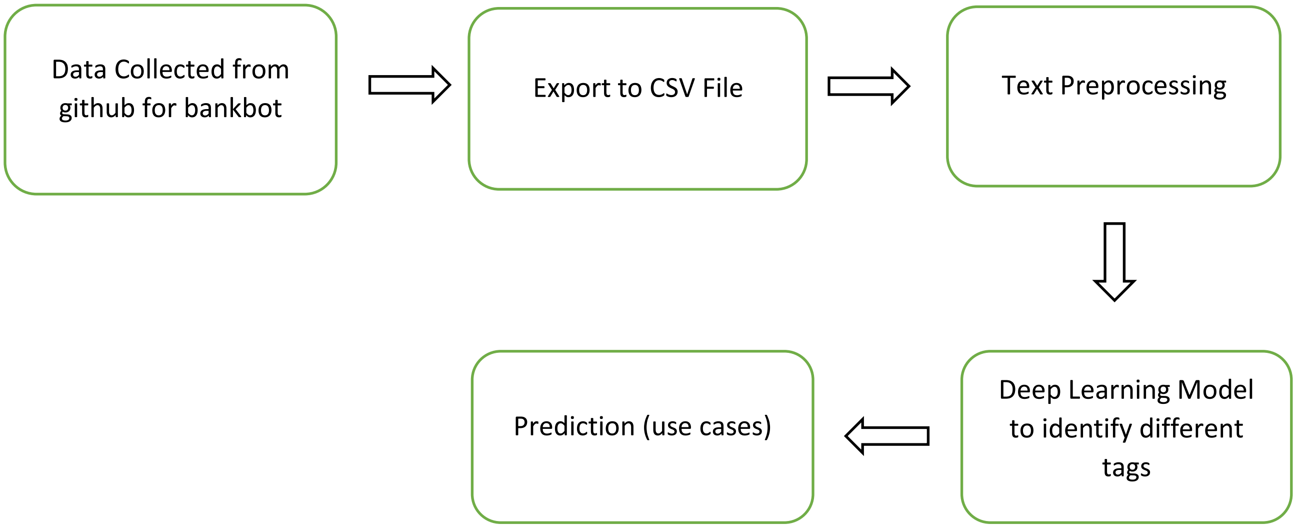
- Python Flask is used for backend development.
- GitHub is used as a version control system.
- ChatGPT is used to get responses.

3 Design Details

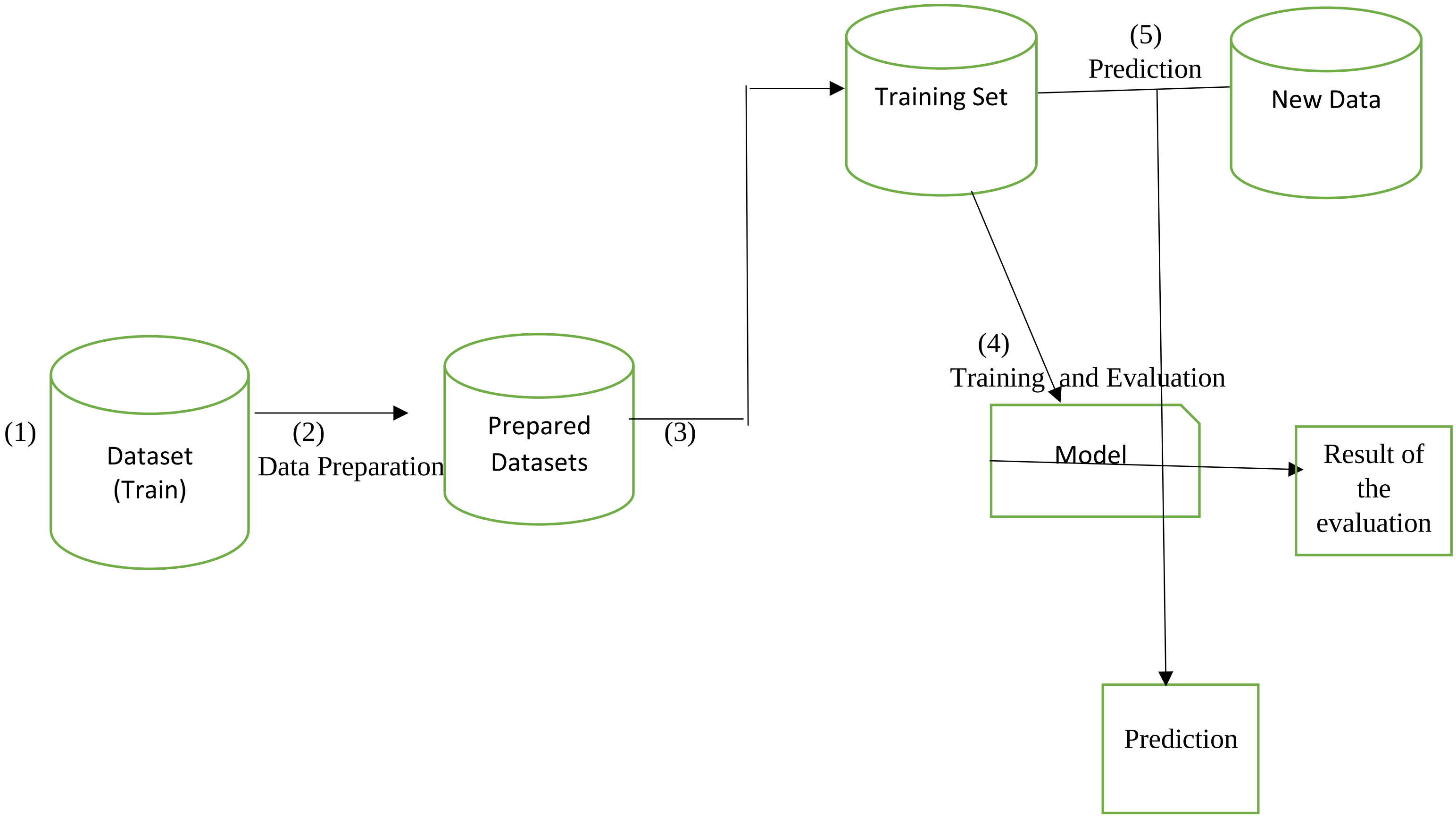
3.1 Process Flow

To identify the different types of tags, we will use the deep learning model for classification. Below is the process flow diagram is as shown below.

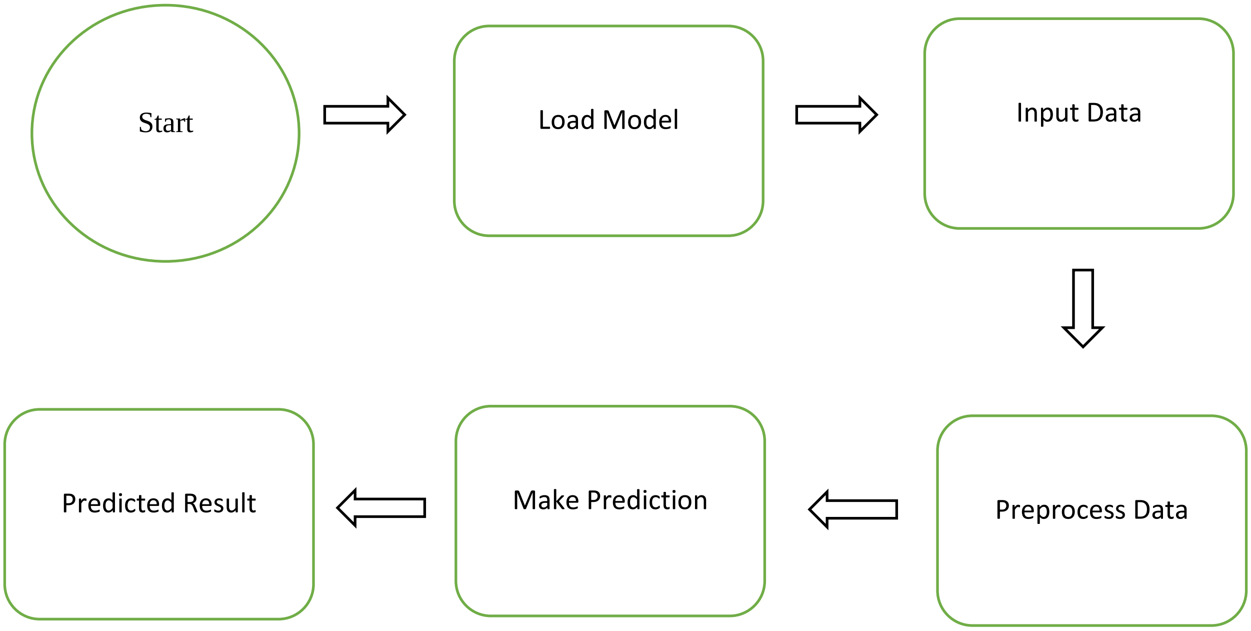
Proposed Methodology


Data Collected from github for bankbot
Export to CSV File
Text Preprocessing
Deep Learning Model to identify different tags
Prediction (use cases)

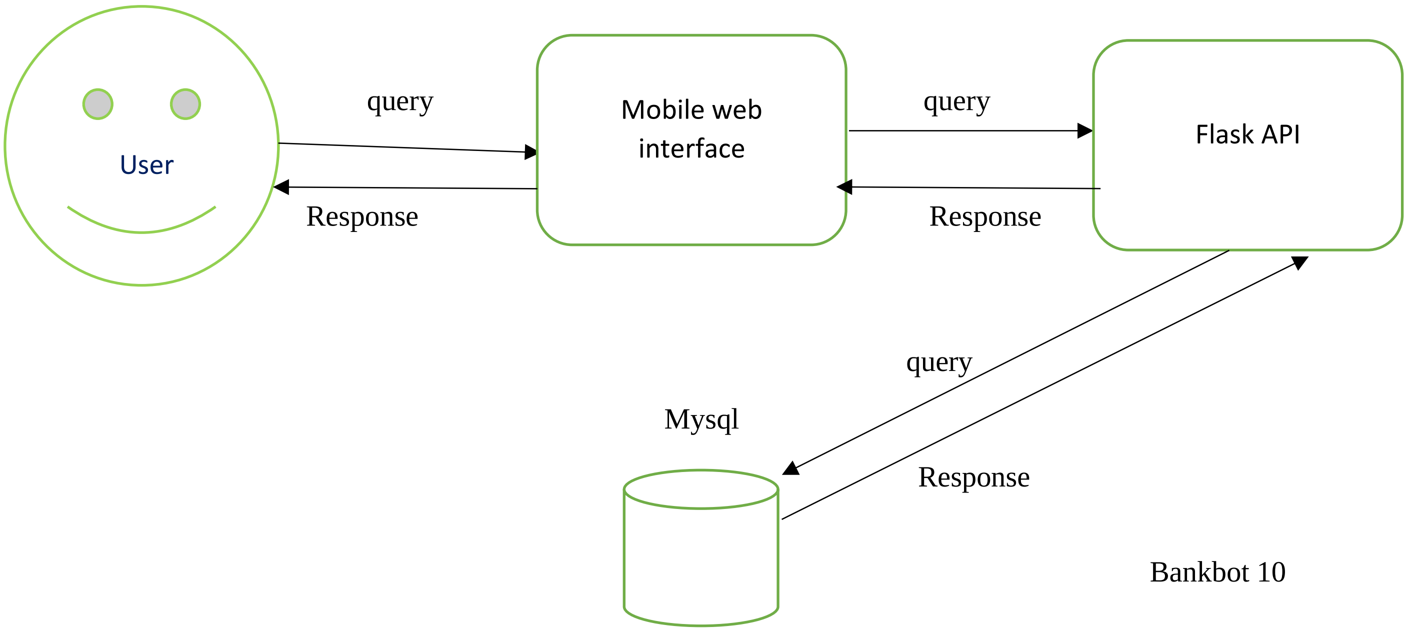
3.1.1 Model Training and Evaluation



3.1.2 Deployment Process



3.1.3 Bankbot Architecture



3.2 Event Log

The system should log every event, so the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging is required.
2. The System should be able to log each system flow.
3. Developer can choose the logging method. You can choose database logging/File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues, so logging is mandatory too.

3.3 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong. An error will be defined as anything that falls outside the normal and intended usage.

4 Performance

The Deep Learning Model is used to identify different tags. The tag response will be the answers to the customer queries. Additionally, it will give transaction details, security changes based on tags.

4.1 Reusability

The code written and the components used should have the ability to be reused with no problems.

4.2 Application Compatibility

The different components for this project will use Python as an interface between them. Each component will have its task to perform, and it is the job of Python to ensure the proper transfer of information.

4.3 Resource Utilization

When any task is performed, it will likely use all the preprocessing power available until that function is finished.

4.4 Deployment

5 Conclusion

The Deep Learning Model will identify different tags. The tag response will be the answers to the customer queries. Additionally, it will give transaction details, security changes based on tags.

6 Reference

- https://computools.com/chatbots-in-banking-how-they-improve-the-customer-experience/
- https://www.forbes.com/sites/blakemorgan/2017/08/06/5-ways-chatbots-can-improve-customer-experience-in-banking/
- https://medium.com/nerd-for-tech/a-financial-digital-assistant-chatbot-fin-bot-using-deep-learning-natural-language-processing-16b19c624541
- https://chat.chatgptdemo.net/