# Bank Bot Wireframe Document

Jaideep Jaiswal

26/01/2025

## Document Version Control

| Date<br>Author | Version | Description |
|---|---|---|
| 26/01/2025<br>Jaideep Jaiswal | 1.0 | Initial wireframe for Bank Bot Project |

## Contents

5. Design Elements
    - Buttons and Links
    - Input Fields
    - Notifications
    - Typography and Color Scheme
6. System Architecture
    - Backend
    - Frontend
    - Integration
7. Future Enhancements
    - Additional Features
    - Scalability Considerations
8. Conclusion

# 1 Introduction

## 1.1 Purpose of the Document

The purpose of this document is to provide a comprehensive wireframe for the Bank Bot project. This wireframe will serve as a design blueprint for the development of the system, outlining the layout, interactions, and essential components of the user interface. It is intended for developers, designers, and stakeholders involved in the project to align on the UI/UX aspects before development begins.

## 1.2 Scope of the Project

This wireframe covers the user interface design for the Bank Bot system, which aims to assist users with banking-related queries, transaction alerts, and fraud detection. The system will integrate with existing banking infrastructure to deliver responses to customer inquiries, track transactions, and provide security alerts.

## 1.3 Goals of the Wireframe

- To visualize the layout of key components of the Bank Bot user interface.
- To outline the flow of interactions between the user and the system.
- To establish the design guidelines for implementing the user interface.
- To ensure ease of use, accessibility, and responsiveness.
- To gather feedback from stakeholders and refine the design before development.

## 1.4 Document Structure

This document is divided into the following sections:

- Introduction: Describes the purpose and goals of the wireframe.
- System Overview: Provides an overview of the system architecture and interaction flows.
- Detailed Wireframe Layout: Describes the layout of various pages, including the Home Page, Query Page, Transaction Alerts Page, and Settings Page.
- User Interaction: Discusses how users interact with the system, including inputs, responses, and error handling.
- Design Elements: Details the buttons, input fields, typography, and color scheme used in the wireframe.
- System Architecture: Outlines the backend, frontend, and integration points of the system.
- Future Enhancements: Highlights possible future enhancements and scalability considerations.
- Conclusion: Summarizes the document and its objectives.

# 2 System Overview

## 2.1 User Interface Flow

The user interface follows a simple and intuitive flow, ensuring ease of navigation and quick access to key features:

- **Homepage:** The user accesses the homepage to start interacting with the Bank Bot.
- **Query Input:** The user enters their query in a text box and submits it to the system for processing.
- **Response Display:** The system processes the query and displays an appropriate response in real-time.
- **Transaction Alerts:** The system monitors transactions and displays alerts for suspicious activities.

## 2.2 System Interaction Flow

The system interaction flow is as follows:

- **User Input:** The user provides input via text fields or buttons (e.g., submitting a query, viewing alerts).
- **Backend Processing:** The backend processes the input using a machine learning model or predefined business logic (e.g., categorizing the query, checking transaction status).
- **Response Generation:** The system generates a response based on the user's query or the transaction status.
- **Display Response:** The response is displayed on the user's screen in an organized manner (e.g., text, transaction details, alerts).

## 2.3 Technologies Used

- **Backend:** Flask, Python
- **Machine Learning:** Random Forest, TF-IDF Vectorization
- **Frontend:** HTML5, CSS, JavaScript (with AJAX for real-time interaction)
- **Database:** SQLite or MySQL (for storing user data and transaction history)

# 3 Detailed Wireframe Layout

## 3.1 Home Page

The home page is the first screen users encounter. It includes:

- **Navigation Bar:** Includes links to the homepage, transaction alerts, and user settings.
- **Welcome Message:** A personalized greeting (e.g., "Hello, [User's Name]").
- **Search Bar:** A central search bar where users can type questions (e.g., "What is my current balance?").
- **Footer:** Contains links to privacy policy, terms of service, and support contact details.

## 3.2 Query Page

On the query page, users can submit their queries. It includes:

- **Text Box:** A large text box for the user to type their query.
- **Submit Button:** A button to submit the query.
- **Response Area:** A space to display the response, such as the bot's answer to the user's query.
- **Loading Spinner:** A visual indicator while the system processes the query.

## 3.3 Transaction Alerts Page

This page shows alerts related to transactions. It includes:

- **Transaction List:** A list of recent transactions, with timestamps, amounts, and status (e.g., successful, pending).
- **Suspicious Activity Alerts:** A section dedicated to any flagged suspicious activities.
- **Action Buttons:** Options to acknowledge or resolve alerts (e.g., "Resolve", "Mark as Safe").

## 3.4 Settings Page

The settings page allows users to configure their preferences. It includes:

- **Change Password:** Fields to update account credentials.
- **Notification Preferences:** Options to customize transaction and activity alerts.
- **Theme Settings:** Switch between light/dark themes for the interface.

# 4 User Interaction

## 4.1 User Input

The user interacts with the system primarily through text input fields and selection options. The key points of user input are:

- **Text Fields:** Used for entering queries, username, password, and feedback.
- **Buttons:** Used to submit queries, acknowledge alerts, and change settings.
- **Dropdowns/Check Boxes:** Used for selecting preferences or setting options (e.g., notification preferences).

## 4.2 System Response

Once the user provides input, the system responds in the following ways:

- **Query Response:** Displays the answer to the user query, either in text or with additional transaction details.
- **Transaction Alert:** Pop-up notifications will inform the user of any suspicious or important activities.
- **Error Messages:** If an error occurs, such as an unrecognized query, the system will provide a user-friendly error message (e.g., "Sorry, I couldn't understand your query. Please try again.").

## 4.3 Error Handling

The system handles errors gracefully by:

- Providing clear error messages for user input issues (e.g., "Invalid query format").
- Redirecting the user to the homepage if a critical failure occurs (e.g., model loading error).
- Offering troubleshooting options or support contacts for persistent issues.

# 5 Design Elements

## 5.1 Buttons and Links

The wireframe uses buttons for:

- **Submit Queries:** The "Submit" button is the primary action button for the user.
- **Acknowledge Alerts:** Buttons to resolve or mark alerts as acknowledged.
- **Navigation Links:** Links in the header/footer for easy navigation between pages.

## 5.2 Input Fields

Text input fields are used for:

- User queries.
- Account login (username, password).
- Feedback submission.

## 5.3 Notifications

Notifications are displayed as pop-ups or banners on the screen for:

- Transaction updates.
- Fraud alerts.
- System errors or confirmations.

## 5.4 Typography and Color Scheme

The wireframe suggests:

- **Typography:** Use of clear, readable fonts like Arial or Helvetica for text.
- **Color Scheme:** A professional color palette with blues and whites, accented with red for alerts.

# 6 System Architecture

## 6.1 Backend

The backend will handle all logic, model predictions, and database interactions. It will:

- Integrate with the Random Forest model to classify user queries.
- Interface with a database to fetch transaction data.
- Manage user authentication and session management.

## 6.2 Frontend

The frontend will:

- Display the user interface designed in this wireframe.
- Handle user interactions via forms and buttons.
- Send AJAX requests to the backend for real-time processing.

## 6.3 Integration

The system will integrate the frontend with the backend using:

- RESTful API calls.
- WebSockets for real-time notifications.

# 7 Future Enhancements

## 7.1 Additional Features

The Bank Bot system could evolve to include:

- Multi-language support.
- Integration with voice assistants (e.g., Alexa, Google Assistant).

## 7.2 Scalability Considerations

To ensure scalability:

- Use of microservices for independent system components.
- Cloud deployment to handle increasing user traffic.

# 8 Conclusion

This wireframe document provides a detailed and comprehensive design for the Bank Bot user interface. By following this blueprint, developers will be able to create an intuitive, efficient, and visually appealing interface for end users. The system architecture and integration guidelines ensure that the system is scalable and adaptable to future needs.