ChatBox AI
Design Specification

Robert Melena, Miguel Aguirre, Robert Guiterrez

December 9, 2025

1 Purpose

This Design Specification describes each page, component, and tool used in ChatBox AI, and lists all packages, dependencies, and libraries that would be required in a Dockerized deployment.

2 Page-by-Page Breakdown

2.1 Landing / Login Page

- Components: Logo, brief description, login form, register link.

- Actions: User can log in or navigate to registration.

2.2 Registration Page

- Components: Inputs for name, email, password, confirm password.

- Actions: Create account and redirect to the chat dashboard.

2.3 Chat Dashboard

- Components: List of existing chat sessions, button to create new session.

- Actions: Open existing sessions, delete sessions, or start a new one.

2.4 Chat Screen

- Components: Message list, text input, send button, session title, export button.

- Actions: Send messages, receive AI responses, export conversation.

2.5 Settings Page

- Components: Theme toggle, language options, data retention settings.

2.6 Admin Dashboard

- Components: Charts or tables summarizing total users, total sessions, total messages.

3 Backend and API Design

3.1 Example Endpoints

- `POST /api/auth/register`

- `POST /api/auth/login`

- `GET /api/sessions`

- `POST /api/sessions`

- `POST /api/sessions/{id}/messages`

- `GET /api/admin/metrics`

4 Dependencies and Libraries

4.1 Frontend (Example Stack)

- React or Vue.js for the UI.

- CSS framework such as Bootstrap or Material UI.

- Axios or Fetch API for HTTP requests.

4.2 Backend (Example Stack)

- Node.js with Express.

- Libraries for authentication, validation, and logging.

4.3 Database

- PostgreSQL as the primary relational database.

4.4 Other Tools

- Docker and docker-compose for containerization.

- Jira for task management.

- TestRail for test case management and reporting.

5 Docker Packaging Requirements

The Docker image for the backend would include:

- Runtime Node.js.

- Application code.

- Configuration for connecting to the database.

The frontend image would include:

- Compiled static assets served via NGINX.