Warehouse Robot Management Dashboard

Design Document

1. System Architecture

The system follows a component-based frontend architecture built using React.js and Redux Toolkit.

Architecture Layers:
- UI Layer: React Components and Pages
- State Layer: Redux Store & Slices
- Data Visualization: Recharts
- Routing: React Router

User Interface → React Components → Redux Store → State Slices

2. UI / UX Decisions

- Card-based layout for robot visualization
- Color-coded statuses for robots
- Auto-refresh UI for real-time updates
- Live charts for analytics
- Clean navigation bar
- Responsive layout

3. Data Flow

1. User allocates a task
2. Action dispatched to Redux
3. Task stored in tasksSlice
4. Task appears in Live Queue
5. Robot is assigned automatically
6. Bot status updates
7. Analytics reflect updated data

4. Key Assumptions

- All data is simulated locally
- No backend integration

- Timers simulate robot movement
- Single warehouse layout

5. Trade-Offs

Redux Toolkit offers predictable state control but introduces learning overhead.
Local simulation speeds development but lacks persistence.

6. State Management Design

Redux Toolkit is used for centralized state management.
Slices used:
- botsSlice
- tasksSlice
- authSlice

Benefits:
- Predictable updates
- Easy debugging
- Scalable design

7. Improvements with More Time

- Real backend integration (Node + MongoDB)
- Authentication with JWT
- Real-time updates with WebSockets
- Export reports
- AI task priority
- Mobile responsiveness