Business Case & Draft Plan

Project LS1

Business Case

Executive Summary

Modern warehouses have a recurring operational problem: ineffective container packing based on human judgment. This leads to wasted space, increased packing times, and variable quality. Our team addresses this by introducing PackPilot, an AI-powered warehouse packing assistant that revolutionizes the packing procedure with intelligent automation and real-time visual directions. PackPilot maximizes speed, accuracy, and space use by converting complex optimization algorithms into actionable commands, consequently decreasing operational expenses and training loads.

1. What: Our Solution and the Problem

Workers now rely on experience or instinct in stacking the containers with the boxes, a system that is prone to human error and not very efficient. Every packing job becomes a new logistical problem, with resulting issues such as wasted container space, variable quality stacking, and added pressure on the staff.

PackPilot remedies this by providing a more intelligent, AI-powered way of warehouse packing. It's a web-based, easy-to-use system that helps both employees and managers. It maximizes box placement by considering size, weight, and other limitations with advanced algorithms. It then delivers straightforward, sequential visual guidance via 2D and 3D interfaces—it eliminates the need for you.

2. Why: The Need for Change

The magnitude of inefficiency is great. In big logistics operations, even the smallest packing mistakes compound into major time and cost impacts. Packing faults, for example, can cut container utilization and cause shipment delay, customer complaints, and added fuel and labor expenditure. Conventional training for packing positions is similarly labor-intensive and will not ensure consistent performance results among a varied workforce.

PackPilot resolves this by reducing the amount of human decision-making that is susceptible to errors. For the employees, it serves as a “GPS for packing,” offering real-time direction on box placement. For the managers, it streamlines the item entry, assignment, and review of work. The system provides pre-built task history logs, interactive UI feedback, and controlled input authentication that reduces errors. It is particularly useful for the onboarding of inexperienced employees, as the system both supports learning and facilitates operations.

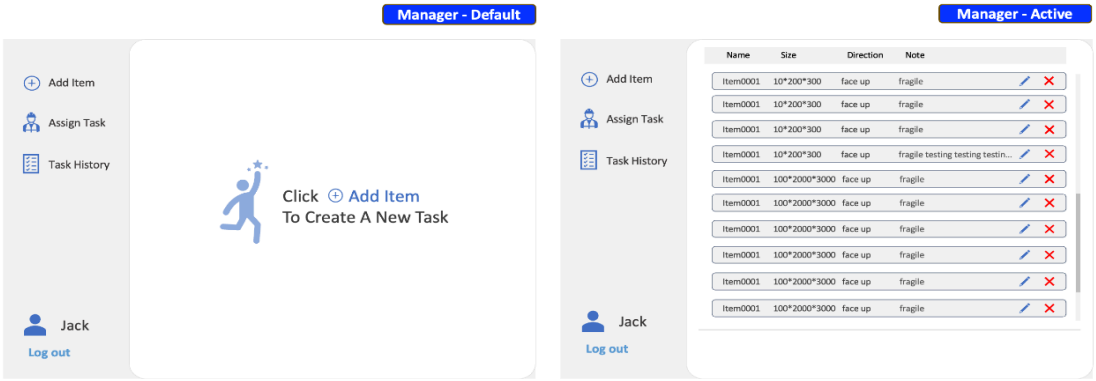
3. Goals: How We Measure Success

Our metrics are closely linked with measurable performance changes and usage adoption:

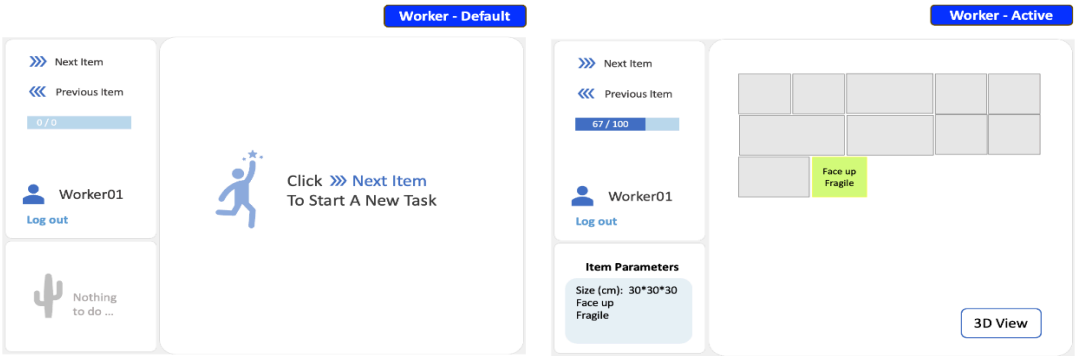
- Operational Efficiency: Time taken for packing decreased using modern methods.
- Error Mitigation: Packing errors and misplaced items.
- Space Utilization: Improved utilisation per container.
- User Adoption & Usability: New warehouse employees capable of utilizing PackPilot efficiently after one shift.
- Scalability: System operates correctly with different product varieties and global package standards.

The following diagram shows the PackPilot system architecture:

- Manager Console: item input, assignment of tasks, and tracking history.



- Worker Console: Provides 2D/3D visual guidance for item placement.



Conclusion

PackPilot is not just software—it is also a redesign of warehouse packing for the age of AI. By bringing optimization theory into practice, it provides a revolutionary leap in packing performance, reliability, and end-user experience. With the use of PackPilot, warehouses can minimize waste, save time, and make their workforce more productive, enabling concrete value creation in logistics operations.

Plan for first milestone

1.1 First Milestone Plan

Milestone 1 aims to establish project fundamentals through planning, requirements gathering, technical design, and initial UI development to create a functional login interface prototype.

Milestone 1	Activities	Projected Outputs
Project Plan	Create project timeline	Gantt chart with key milestones
	Establish team structure	Team roles and responsibilities document
	Develop a communication strategy	Communication plan
Requirements Analysis	Gather client requirements	Product Requirements Document (PRD)
	Map user flows	User journey diagram
	Create user stories	User story with acceptance criteria
Technical plan	Select technology stack	Technology stack documentation
	Design system architecture	System architecture diagram
	Define API endpoints	API design document
	Create database schema	Database schema design
Development	Set up dev environment and build the main application layout	Basic UI framework implementation
	Develop login/sign-up interface	Functional login/registration UI
	Define API contracts	API specification document
	Configure database connections	Working database connection
Testing Plan	Identify key features for testing	Feature testing checklist

1.2 Team Organization

Internal Roles

Role	Responsibilities	Assignment Approach
Project Coordinator	Meeting facilitation, risk management, progress reporting, team coordination	Shared/Rotating weekly among all members

Requirements Analyst	Requirement gathering, feature specification, user story and acceptance criteria definition	Collaborative effort by all team members
Full-Stack Developer	Frontend UI implementation, backend API development, database design, integration testing	All team members contribute based on current project needs
3D/2D Visualization Specialist	Implementation of warehouse visualization components, algorithm output rendering, and interactive display features	Team members with relevant expertise

External Stakeholders

Role	Responsibilities	Entity
Algorithm Specialist	Development of optimization algorithms, performance tuning, and integration support	External Developer
Supervisor	Technical guidance, project evaluation, and client communication facilitation	Course Instructor
Client	Business requirements definition, feedback on deliverables, final acceptance	Client Representative

1.3 Communication Plan

	Objectives	Frequency & Method	Platforms	Key Activities
Internal	Ensure timely and efficient internal team communication to maintain task progress and quality.	• Weekly student meetings (Fri 5-6 pm) • Ongoing communication on WeChat • Additional meetings as needed	Microsoft Teams, WeChat, Google Docs, GitHub, Email	Progress review, task assignments, issue discussion, supervisor feedback, minutes preparation.
External	Maintain effective communication with external stakeholders to obtain timely feedback and ensure project deliverables meet customer needs and expectations.	• Algorithm Specialist: Communication by email or meetings when technical issues arise • Supervisor and Client meetings: Every two weeks (Fri 4–5 pm) • Additional communication by email or WeChat as needed	Microsoft Teams, WeChat, GitHub, Email,	Technical guidance, progress updates, milestone reviews, client feedback, issue resolution. (The previous note-taker chairs the meeting and appoints the next in rotation.)