Stock Maintenance System (SMS)

Introduction:

→ Purpose :- The purpose of the (SMS) is to streamline & automate the process of tracking & managing Inventory levels, orders, & sales in real time

→ Scope :- The system will be used by retail stores, warehouses, & e-commerce platform to maintain stock levels & track products from suppliers to consumers. It will key functions such as stock monitoring, order processing & report generation.

→ Overview: SMS will be web or desktop-based application that allow users to:
- → Add, remove & update inventory data.
- → Track stock levels in real-time.
- → Create reports for stock analysis, reordering, & sales.
- → Handle stock transfer b/w different locations.

→ General Description :-
- Users :- Store manager, warehouse operator & Inventory clerk.
- Inventory management, Order processing, Reporting.
- Technology Stack :- Can use MERN stack (MongoDB, React for web based or Java for desktop-based systems.
- Security :- Role-based access control (Admin, Manager)

→ Functional Requirements :-
- Inventory management : Ability to add, edit & delete items from stock.
- Stock monitoring :- Real time update on stock with safety & alert when Inventory levels are low.
- User Roles :- Different roles with specific access levels (Admin, Manager, Clerk).

→ Interface Requirements :-

- UI :- A user-friendly, responsive web-based interface with simple navigation for non-technical users.
- Admin Interface: More detailed, with functionalities for stock adjustments, report generation, & supplier management.
- API :- RESTful API's for integrating the stock system with 3rd party platforms such as e-commerce & accounting s/w

→ Performance Requirements :-

- Response Time :- The system should respond within 2 seconds for inventory queries & order generation.
- Availability :- The system should have 99.9% uptime to ensure availability for stock management.

→ Design Constraints :-

- Scalability :- The system should be scalable to accomodate additional users & stock items as the business grows
- Data Storage :- Relational or NoSQL database should be able to handle large datasets & fast retrieval.

→ Non-Functional Attributes :-

- Security :- Use encryption for sensitive data like supplier details & financial records.
- Reliability : System must be reliable with minimal downtime.
- Usability :- The system should be intuitive & easy to use, requiring minimal training

Date: / /

→ Preliminary Schedule :-

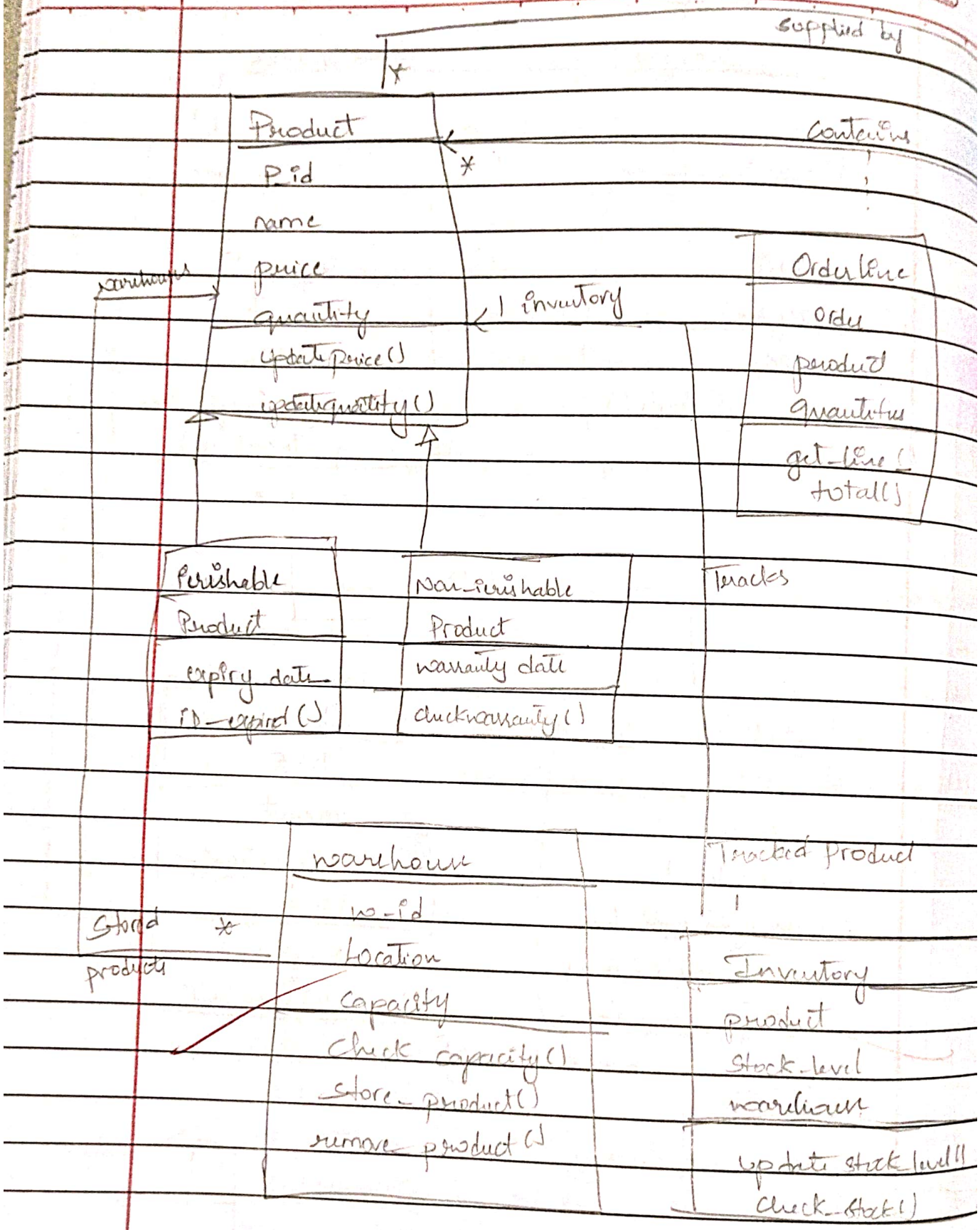
- Planning & Requirement Gathering : 2 weeks
- Design & prototyping : 3 weeks
- Development : 6-8 weeks
- Testing & QA : 3 weeks
- Deployment : 1 week
- Total Estimated Duration : 15-17 week.

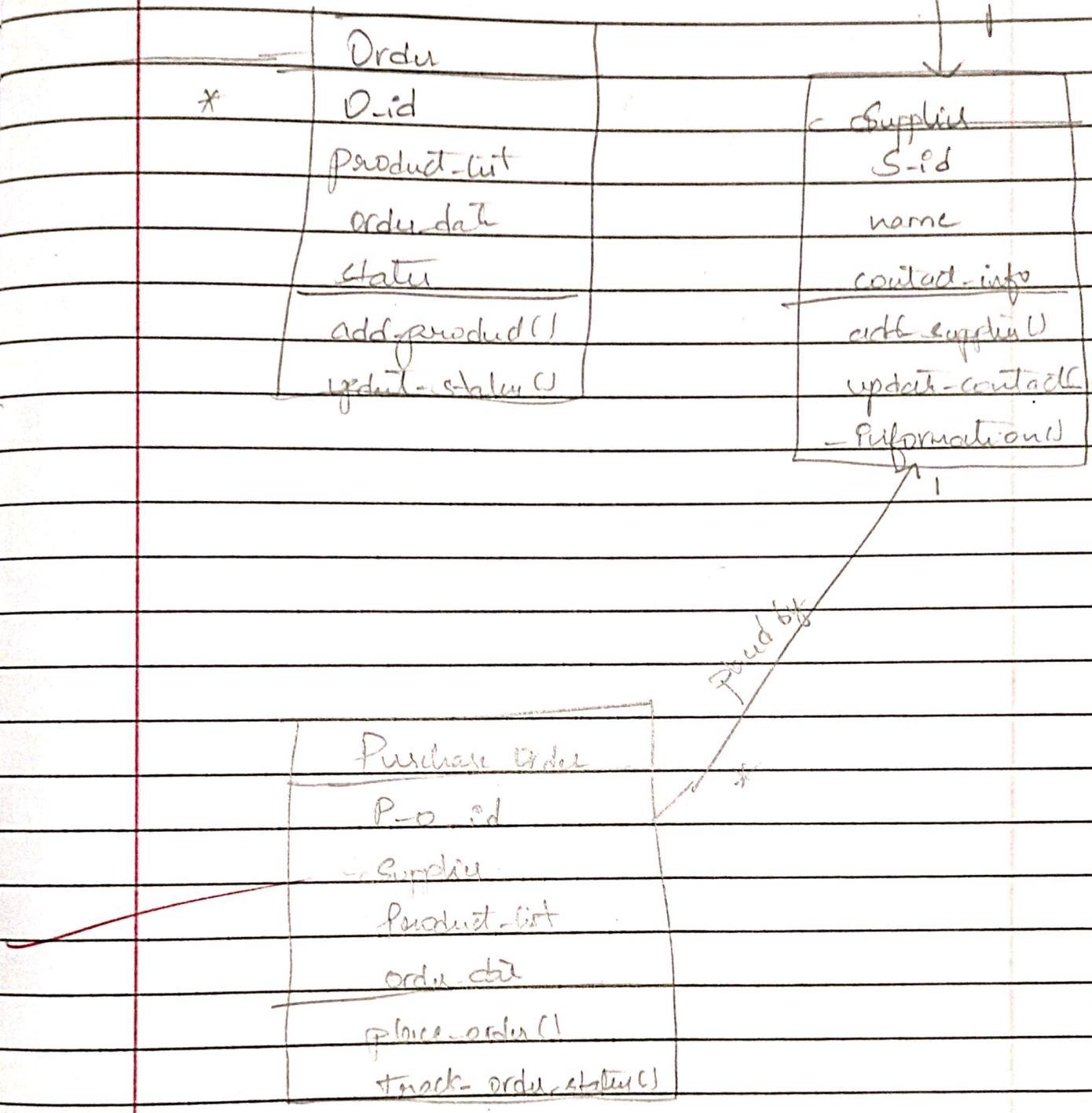
Budget :-

→ Development Team Costs : $50,000
→ HW costs : $5000
→ SW ~~costs~~ Licenses : $2000
→ Maintenance costs : $10,000

Total Initial Budget Estimation : $67000

5) Stock Management System




Ordu
O_id
product_list
ordu_date
status
add_product()
updat_status()
Supplier
S_id
name
contact_info
add_supplier()
update_contact
_information()
Paid by
Purchase Order
P_O_id
Supplier
Product_list
ordu_date
place_order()
track_ordu_status()

Day 15/10/24