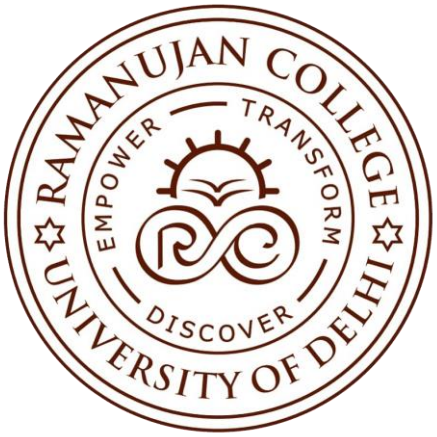


SOFTWARE MODELLING PRACTICAL

Name: Harsh Dubey

College Roll no: 2023713

University Roll no: 23020107023

Semester: IV

Course: B.Voc. Software Development

Project: WATCH STORE

PRODUCT MANAGEMENT

Function Point Calculation

Function Type	Count	Weight (Average)	Total FP
External Inputs (EI)	8	4	8 x 4 = 32
External Outputs (EO)	6	5	6 x 5 = 30
External Inquiries (EQ)	5	4	5 x 4 = 20
Internal Files (ILF)	3	7	3 x 7 = 21
External Interface (EIF)	2	5	2 x 5 = 10

Total Function Points: 113 **FP**

Function Type Description

Function Type	Description
External Inputs (EI)	Customer login, order placement, payment submission
External Outputs (EO)	Order confirmation, invoice generation, promotional notifications
External Inquiries (EQ)	View order status, search for available watches
Internal Files (ILF)	Customer database, watch inventory, sales record storage
External Interface (EIF)	Payment gateway APIs, shipping service APIs

Efforts (For Manpower)

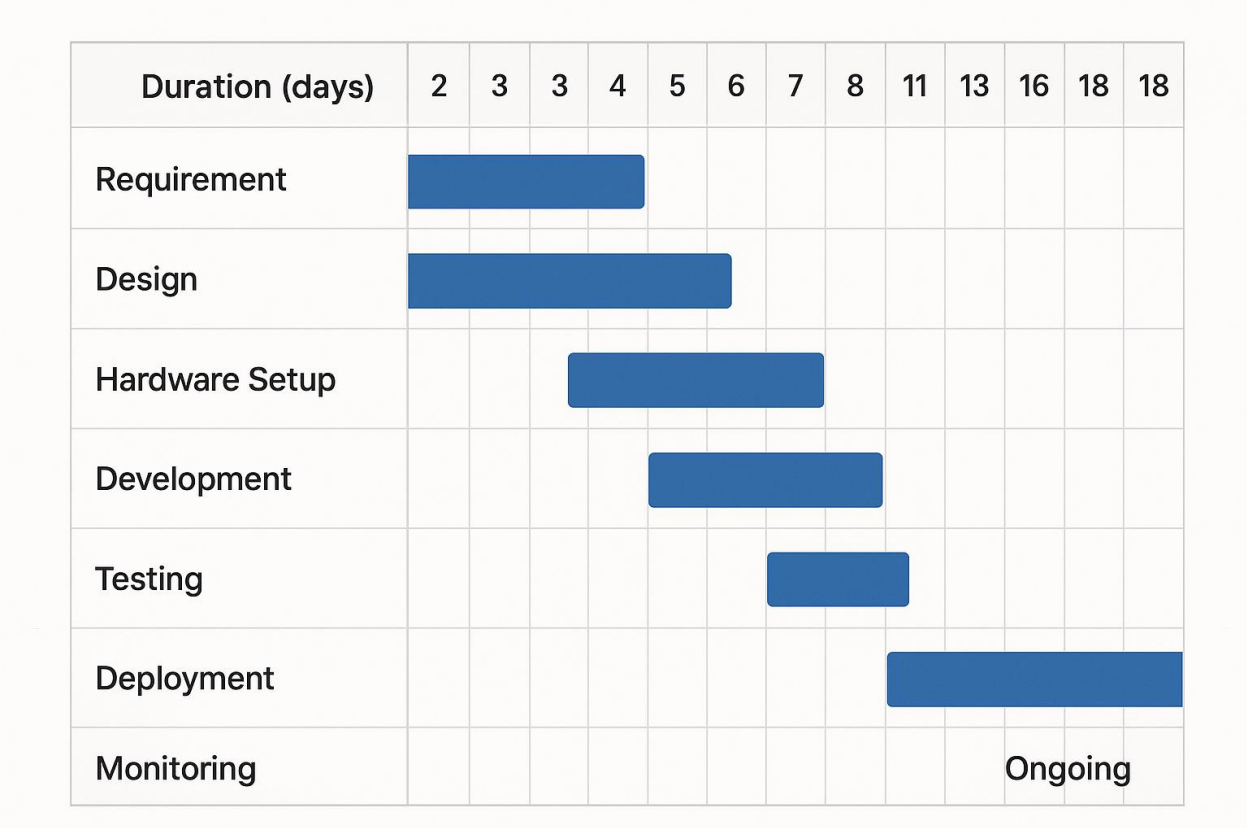
Efforts = Total FP * Productivity Factor

Efforts = 113 * 2.5 = 282.5 person-hours

Schedule and Timeline Chart (Gantt Chart style)

Phase	Duration (days)	Start (Day)	End (Day)
Requirement	2	1	2
Design	3	3	5
Hardware Setup	2	6	7
Development	7	8	14
Testing	2	15	16
Deployment	1	17	17
Monitoring	Ongoing	18	-

Gantt Chart



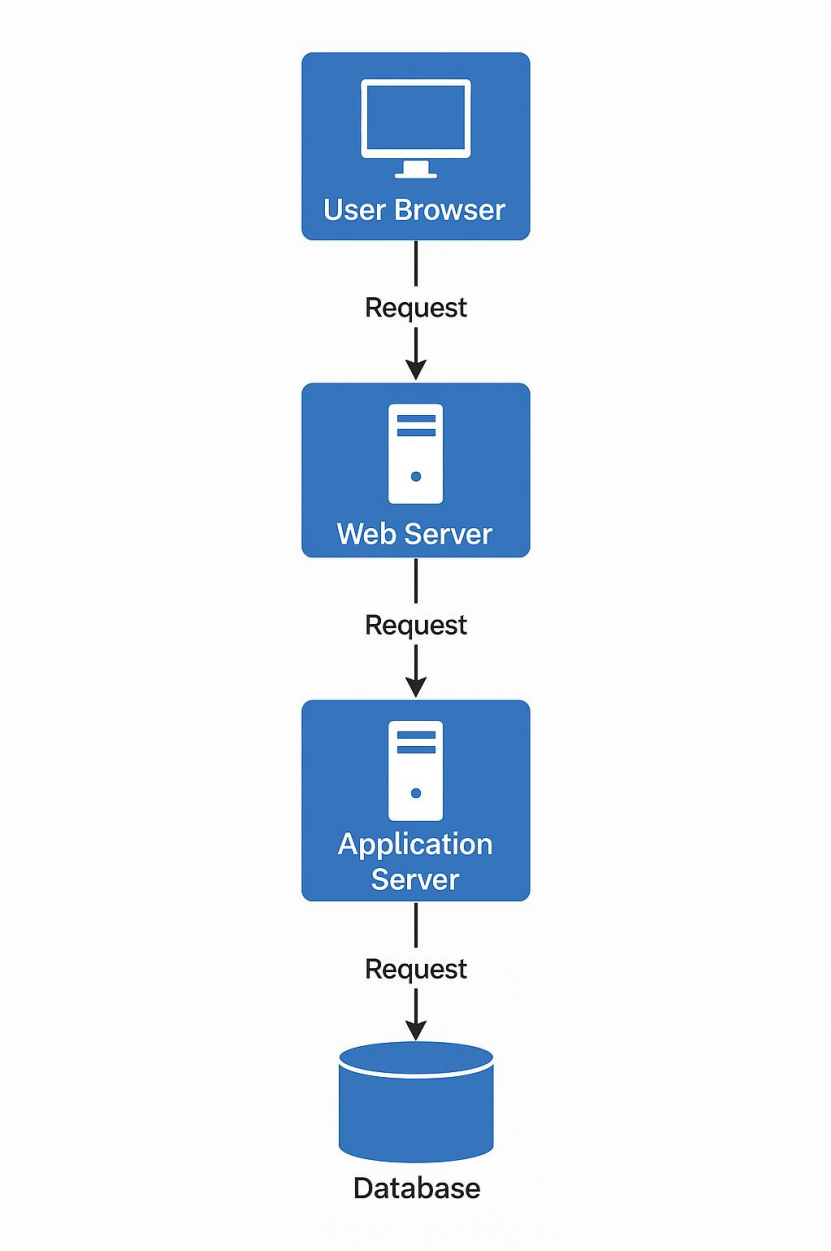
Risk Management

List of Risk	Risk Type	Probability	Impact	Mitigation
Server downtime	System Reliability	Medium	High	Use cloud hosting with backups
Payment gateway failure	Payment Reliability	Medium	High	Integrate backup payment gateways
Cybersecurity threats	Cybersecurity	Low	High	Secure API, SSL encryption, regular audits
Inaccurate stock availability	Operational Risk	Medium	Medium	Real-time inventory sync
Poor customer experience	Usability	Medium	Medium	Usability testing and design improvements

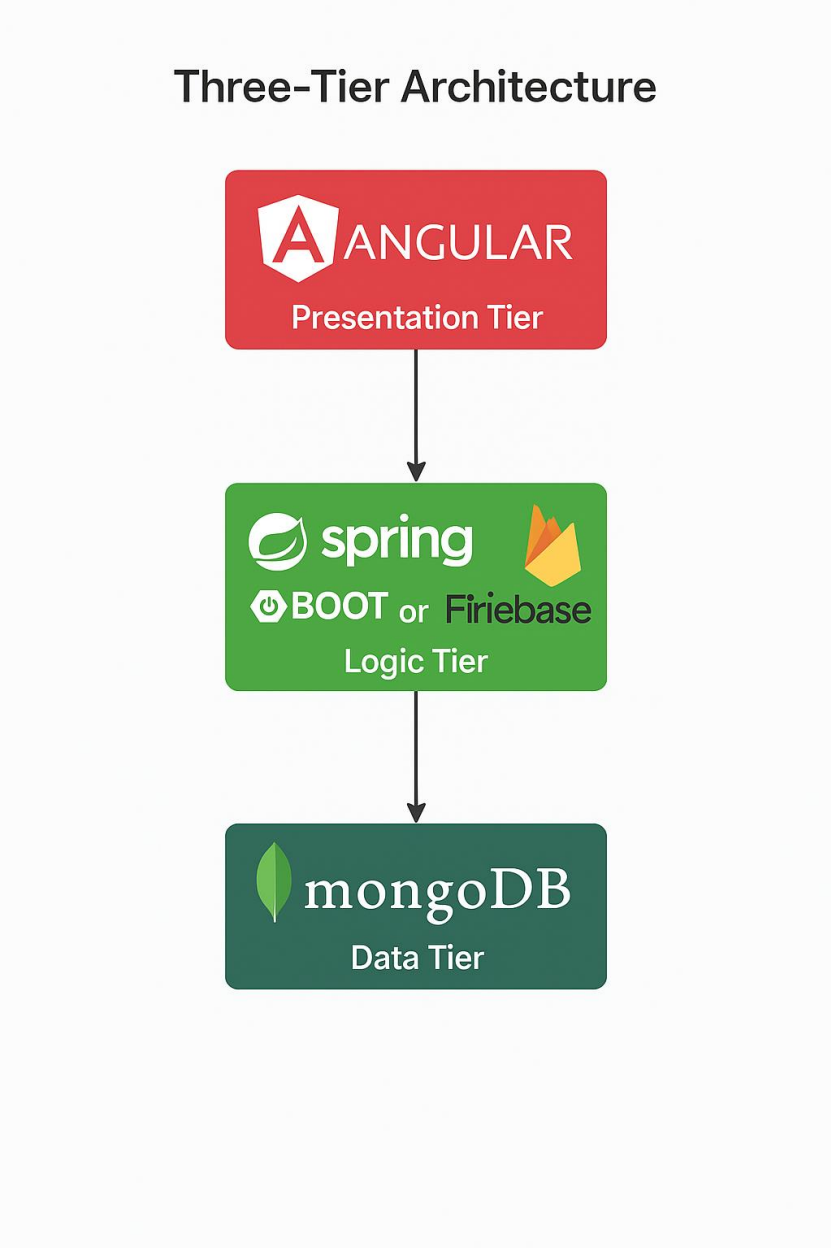
DESIGN ENGINEERING:

a. Architectural Design

Follows three-tier architecture to process order, display user specific watches.



Tech Stack Specific Architecture. Angular (Frontend), Spring Boot or Firebase (either one), MongoDB (Data layer, mainly for storing payment details).



b. Data Design, Component Level Design

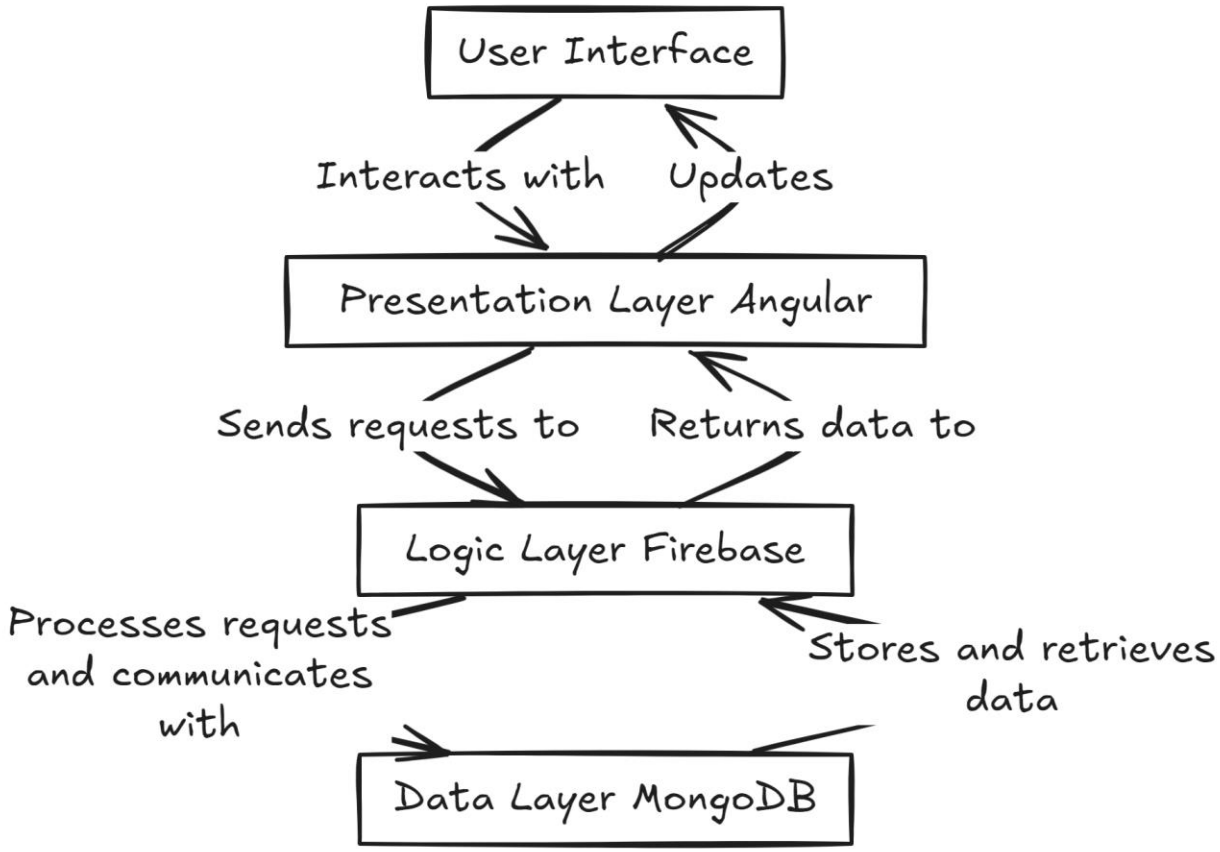
System Design Specification

1. Data Design

- The system includes non-relational database tables for Customers, Products, Orders, and Payments.
- Sensitive user and transaction information is stored securely.

2. Component Level Design

- **User Authentication Module:** Handles user login, registration, and password management.
- **Product Management Module:** Manages product listings, search, and filtering operations.
- **Order Management Module:** Manages the shopping cart, checkout, and order tracking.
- **Payment Integration Module:** Handles online payments through integrated payment gateways.
- **Admin Dashboard Module:** Manages inventory, reviews orders, and monitors performance.



Request Path