12/08/25

OOM Lab

Application Problems

1. Hotel Management System
2. Credit Card processing
3. Library Management System
4. Stock maintaince system
5. Passport automation Sys

#Developing problem statement

- Develop a complete IEEE standard SRS document with several requirements (SRR) → Software Requirement Specs

- Intro :

→ Purpose of the document

→ Scope of the document

→ Overview of the document

- General Description :

- Functional Requirement

- Interface Requirement

- Performance Requirement

- Design Constraints

- Non-financial attributes

- Preliminary Schedule budget

SRS Document for {Hotel Management}

1. Introduction

1.1 Purpose of this Document

- To define requirement of HMS
- To serve as a reference for developer, tester and stakeholders
- To ensure the system needs of hotel staff and needs

1.2 Scope of this Document

- Provide official management of hotel operation
- Saves time, reduces errors, enhance customer satis-faction
- Helps estimate development cost

1.3 Overview

- HMS is a software application for handling hotel operations
- Offers easy access to HMS, data
- Enhance overall productivity.

2. General Description

- Automates room booking, check in / check out
- Uses characteristic : hotel staff (admin) customer
- Benefit : reduces paperwork, ensure faster service
- Supports customer community.

3. Functional Requirement

- Customer registration, room-booking, check-in, check-out etc
- Automatic bill generation, including taxes and service
- Room availability & management
- Data storage and retrieval for judiciary

4. Interface Requirement

- User interface : Simple GUI
- S/w interface : Integration with payment gateway, email etc
- Hardware : Connect to ~~printer~~, printers, fax, etc

5. Performance Requirement

- ~~User interface~~
- ~~Should handle interface : Integration with payment~~
- Should handle minimum 500 customer
- Response time for any action < 2 secs
- Error rate ↓ . concurrent access to users

6. Design constraints

- Must use RDBMS
- desktop platform + web based platform
- minimum 8GB RAM
- comply with SSL certificate

7. Non Functional Requirement

- Security : encrypted customer and payment data
- Reliability : 24/7 runtime
- Scalability : support for multiple branches
- Portability : Compatible with windows and Linux systems.

8. Preliminary Schedule and Budget

Schedule

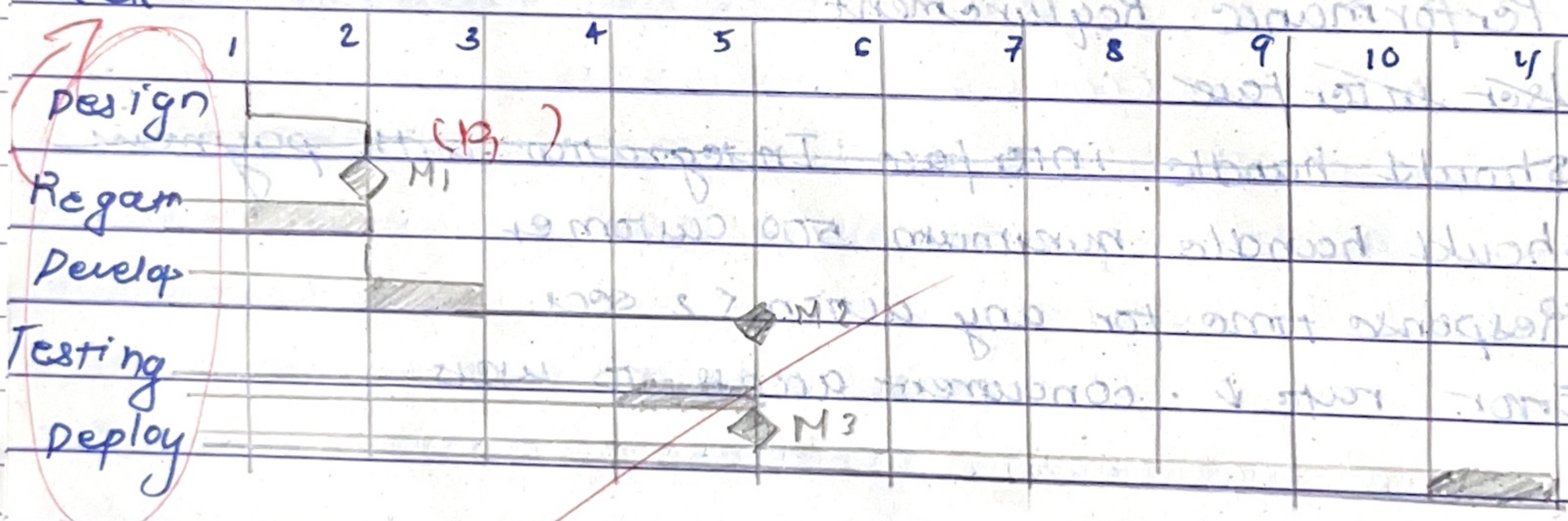
- Req analysis : 2 weeks
- Design : 3 weeks
- Development : 6 weeks

Budget

→ 5 - 7 lakhs

- covers development

Scheduling timeline :



SRS Library Management System :

1. Introduction

1.1 Purpose of this Document

- To define req^r of LMS
- To provide a clear reference for development
- To ensure the system meets the need of staff

1.2 Scope of this document

- Automate borrowing, returning, catalog
- saves time, reduces manual error, accessibility
- Development cost

1.3 Overview

- LMS is designed to maintain record of books.
- Provide search, issue/return, fine calculation
- Improves efficiency, in day-to-day ops

2. General Description

- Handles book cataloging, issue/return
- user char, librarians, staff etc
- Benefits: quick book search, reduced paperwork
- supports user community by providing.

3. Functional Req:

- Book search by title, author ISBN
- Borrowing and Returning
- Fine calculation of late return
- user account management

4. Interface Requirement
- user interface : Dashboard for admin, search and account view
- s/w inter : integration with email
- H/w interface : work on PC's, barcode scanner

5. Performance Requirement
- should handle upto 1000 transactions per day
- Book search result
- Error rate for run ↓ 0.5%
- Database should support multiple users

6. Design Constraints
- Must use RDBMS
- Webbase app accessibility
- 8 GB RAM, dual core processor
- must comply with SSL certificate

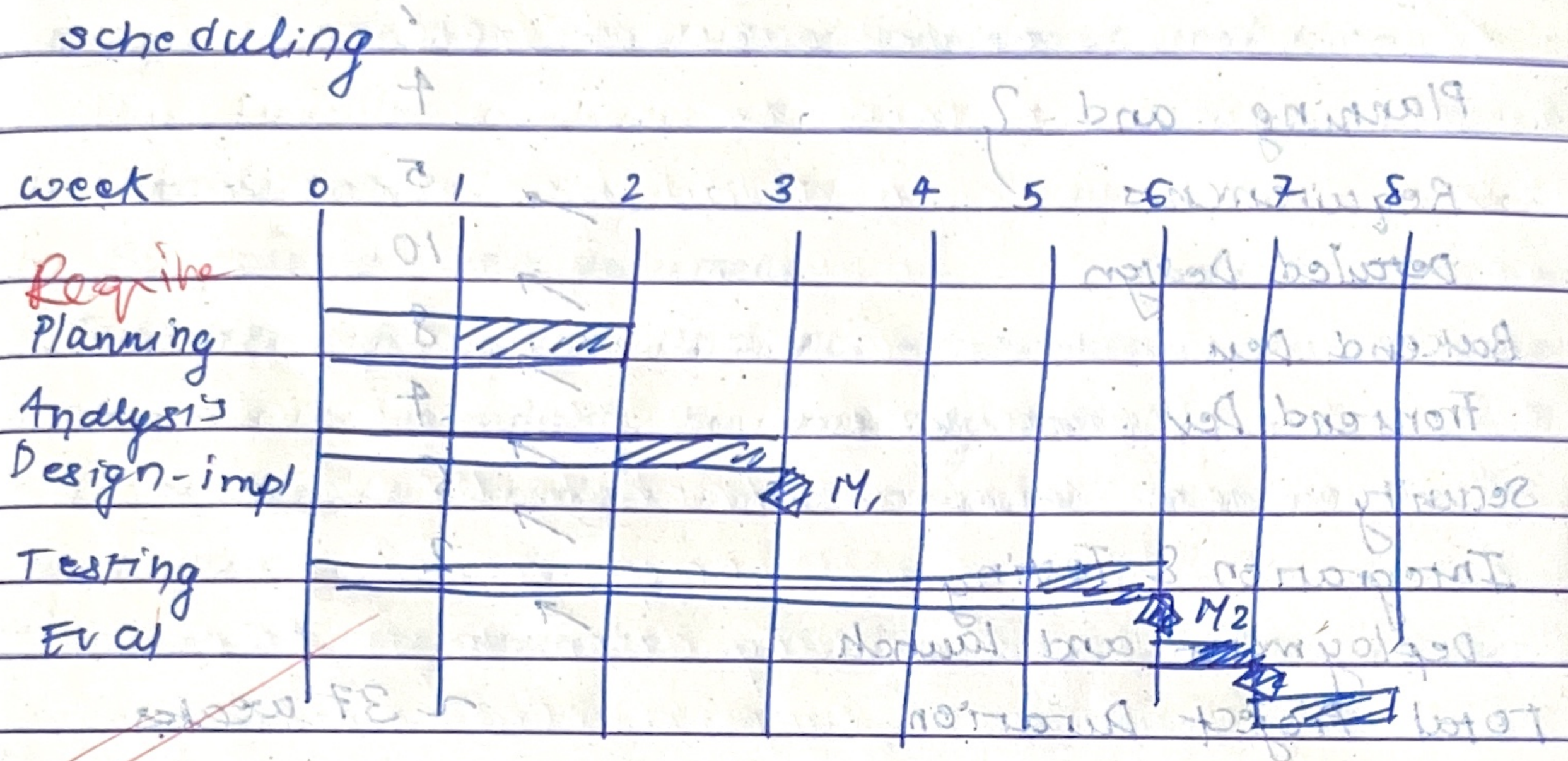
7. Non-functional Requirements
- Security : Access control for admin and users
- Reliability : Backup of book and transaction data
- Portability : Run on windows / Linux
- Scalability : Extendable to support e-books

8. Preliminary budget, scheduling
- schedule :
  - Requirement analysis : 2 weeks
  - Design : 2 weeks
  - Development : 3 weeks
  - Deployment : 1 week

Budget :

- 3-5 lakhs
- covers development + Testing

scheduling



Schedule	Duration (weeks)
Requirement analysis	3 weeks
Design	4 weeks
Development	10 weeks
Testing	4 weeks
Deployment	2 weeks

Budget ($30,000 / 30 Lakhs)

H/w : 5 lakhs

S/w : 3 lakhs

Development: 12 lakhs

Testing : 5 lakhs

Training : 3 lakhs

Maintenance : 2 lakhs / year

For CREDIT CARD PROCESSING

Preliminary Schedule and Budget :

Schedule Phase	Duration (weeks)
Planning and Requirements	4
Detailed Design	5
Bakend Dev	10
Frontend Dev	8
Security	4
Integration & Testing	6
Deployment and launch	2
Total Project Duration	~ 37 weeks

Budget

Personnel costs : 75 lakhs
Infra & Tools : 5 lakhs
PCIDSS certification : 10 lakhs
contingency (15%) : 13.5 lakhs
Total budget : 1 crore

SRS Document for Credit

→ Problem Statement:
There is a need for a
to process credit card t

1. Introduction
1.1 Purpose of this docume
· To define functional an
that will be needed by
· To declare a budget
will take
· ensure all expectations

1.2 Scope of this document
· Covers transaction auth
· will integrate into mo
· exclude bank side pr

1.3 Overview
· Sec 2 describe gener
· Sec 3-7 detailed sys
· Sections 8 outlines B

2. General Description
· Supports AMEX, Visa,
· Real time auth and
· Data encryption an
· cross-platform compa

SRS Document for Credit and card Processing

→ Problem Statement :

There is a need for a secure and efficient system to process credit card transactions in real time

1. Introduction

1.1 Purpose of this document

- To define functional and non-functional requirement that will be needed by the system
- To declare a budget and timeline the project will take
- ensure all expectations are aligned and documented

1.2 Scope of this document.

- Covers transaction auth, fraud detection, payment...
- will integrate into mobile, PC, web app.
- exclude bank side processing

1.3 Overview

- Sec 2 describe general description and background
- Sec 3-7 detailed system functionality, interfaces
- Sections 8 outlines Budget & timeline.

2. General Description

- Supports AMEX, Visa, Digital wallets etc
- Real time auth and settlements
- Data encryption and tokenization
- cross-platform compatibility

3. Functional Requirements
- User can initiate, authorize, complete transaction system verify car validity, available balance
- Admin panel provides access to transaction history blocking of cards etc
- Aids fraud detection, calculation the times pin entered wrongly.

4. Interface Requirements
- User interface : clean web/mobile UI
- Admin : Dashboard with access control
- External interface : integration with API
- POS H/w interface : supports card swipping, Tap to read etc.

5. Performance requirement :
- Txn time ≤ 2 seconds
- Sys must handle minimum 1000 user.
- 99.9% uptime + failure support
- Response by API < 2 seconds

6: Design Constraints :
- Comply with PCI-DSS standards
- use only HTTPS for all communication
- Deployable on this cloud infrastructure.

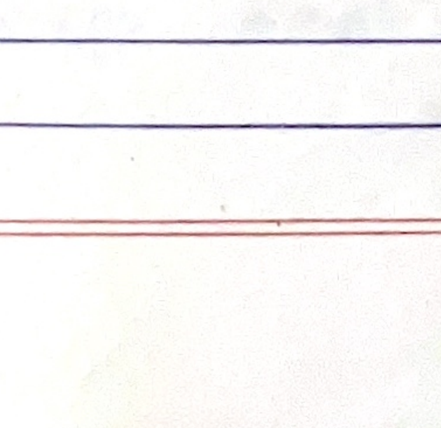
7. Non-functional Requirements :
- Security : End-End encryption and multi-factor auth
- Scalability : Horizontal scaling for ↑ volume txn.

- Maintability : Mo
- Availability : 24/

8. Preliminary Sche

- Budget
  - Total LOC = 25
  Avg cost / LOC
  ∴ Total co

Timeline :
- Requirement ana
- Design : 3 week
- Development : 2
- Deployment : 1 w



, complex transaction
available balance
to transaction

apon the time pin

bile UI

ss control
tion with API
s card swipping,

1000 user
eport
nd

d

unication
infrastructure
s:
and multi-fac

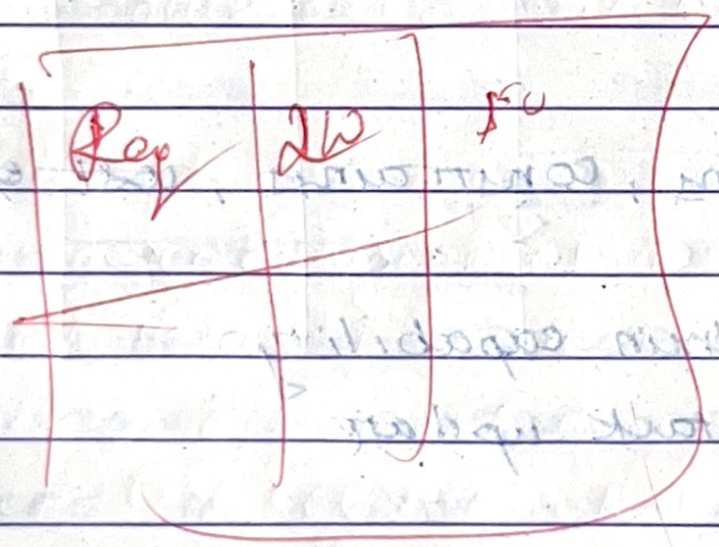
for 1 volume

- Maintainability : Modular codebase for easy update
- Availability : 24/7 system access

8. Preliminary Scheduling & Budget

- Budget
  - Total LOC = 25,000
  - Avg cost / LOC = $0.10
  - ∴ Total cost = 25,000 × 0.10
    = 2500
    to INR = 25,00,000

Timeline :
- Requirement analysis : 2 weeks : RA
- Design : 3 weeks
- Development : 2 weeks
- Deployment : 1 week



SRS Document for Stock Management

Problem Statement:
Business require reliable system to update inventory levels, in order to reduce losses

1. Introduction:
1.1 Purpose of this Document.
- Design requirement for stock management system
- serve as guide for design, development
- Align stakeholder expectation

1.2 Scope of this Document:
- Covers inventory taking, stock updates recorder alert
- used by warehouse, retail store, supply chain manager

1.3 Overview:
- Describes system/functions, constraints, cost estimation
- include both user + admin capability
- Emphasize real time stock update

2. General Description
- support multiple warehouse and product categories
- Real time stock level updates on scales
- Barcode, RFID, support
- Role base access control

3. Functional Requi
- Add, update, rem
- Track stock inflow
- Generate alert whe
- Provide Report

4. Interface Require
- UI: Intuitive da
- Admin I: Device
-gement:
- Hardware I: Bar
-gration.

5. Performance Req
- stock update late
- roughly 100 peopl
use
- There should be fas
- Handle upto 10,00

6. Design Constraint
- should be simple
- compliance with
- Backend in Python

7. Non-functional R
- Security: role based
- Scalability: Ability
- Maintability: modul
- usability: simple

3. Functional Requirement

- Add, update, remove stocks
- Track stock inflow, outflow with time stamp
- Generate alert when stock lower then order.
- Provide Report

4. Interface Requirement:

- UI : Intuitive dashboard for inventory update
- Admin I : Detailed analytics and user management.
- Hardware I : Barcode and RFID scanner interface.

5. Performance Requirement :

- stock update latency < 1s
- roughly 100 people should be able to concurrently use
- There should be faster access for admin
- Handle upto 10,000 SKU's

6. Design Constraint

- should be simple to use
- compliance with data protection law
- Backend in Python

7. Non-functional Requirement

- Security : role based access
- Scalability : Ability to add warehouse down time.
- Maintability : modular Design with clear Sloc.
- usability : simple UI.

8. Preliminary budget + scheduling.

→ Budget

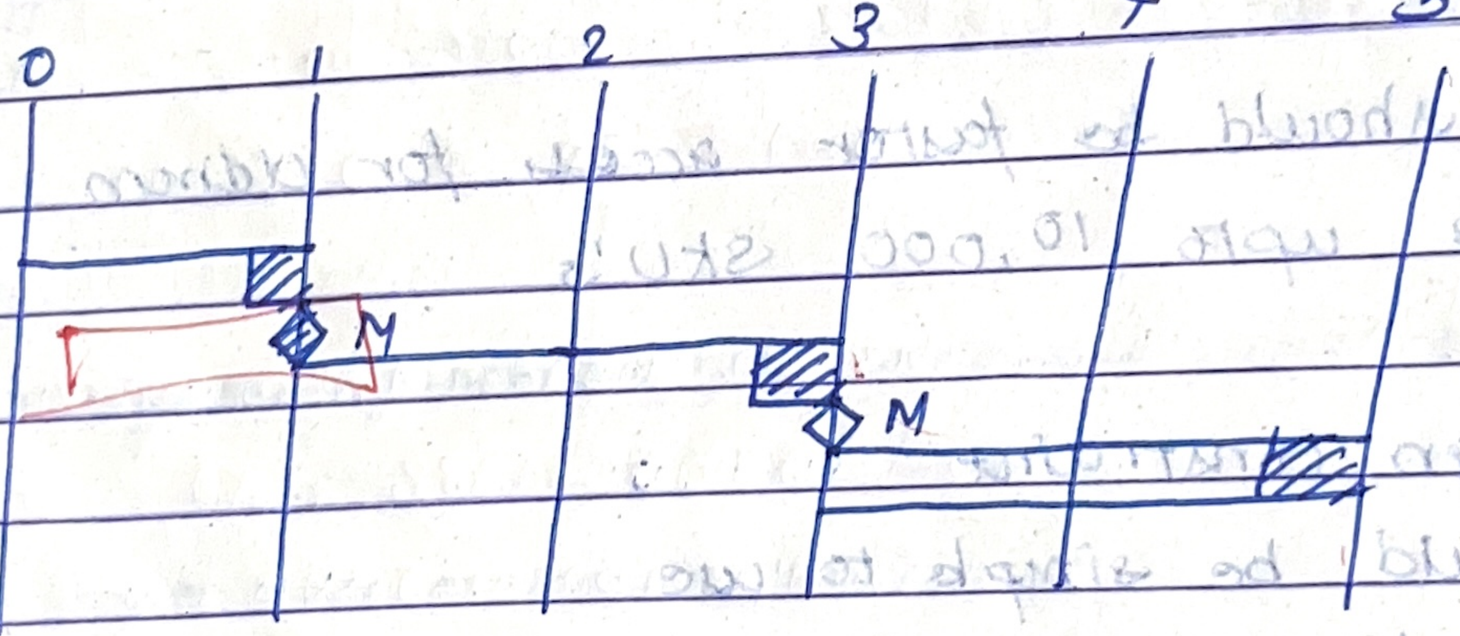
- Total LOC: 18,000
- cost per LOC: 8.3

∴ Total cost: 1,49,400,

→ Scheduling:

RDA → 1 week
Implementation → 2 weeks
Testing → 1 week, → M2 ⇒

week



1) -
1.1
• D
• E
• en
1.2 Sc
• Cle
stru
• exclu
1.3 Over
• Sections
design t

2) General
web base
• Link with
• Tracker f
• Send stat

3. Functional
• Apply, uploa
• Admin proce
• Police upda

SRS for Passport System :

Problem Statement

- Manual passport process are slow and error prone, automated system needed for faster, accurate application and verification.

1) Introduction

1.1 Purpose of Document

- Define system requirement
- Guide for developers, tester and government
- ensure process assignment and compliance

1.2 Scope

- Covers application, appointments, uploads, status tracking, police verification
- excludes painting and delivery.

1.3 Overview :

- Sections describe system features, performance, design limits and cost

2) General Description :

- Web based for citizen and staffs
- Link with Aadhar, Pan, Digilockers.
- Tracker full application life cycle
- Send status updates.

3. Functional Requirements

- Apply, upload docs, and book appointments
- Admin process and tracks application
- Police update verification status.

- user views and track progress.

4) Interface Requirement:

- Citizen UI : clean + multilingual
- Admin Panel : Dashboard with Filter
- Mobile view : Fully responsive

5) Performance Requirement:

- 5006 + users to us simultaneously
- < 2 sec response time
- 99.9% uptime
- secure file handling

6) Design Constraints

- Follow IT act data, laws
- use open source stack
- Hosted on NIC or gov cloud.

7) Non-functional Requirements

- Security : OTP, SSL, aadhar login.
- Scalability : nationwide usage
- Accessibility : WCAG, complains
- Reliability : two backups

8) Budget and Time Table :

→ Budget

Total LOC = 30,006

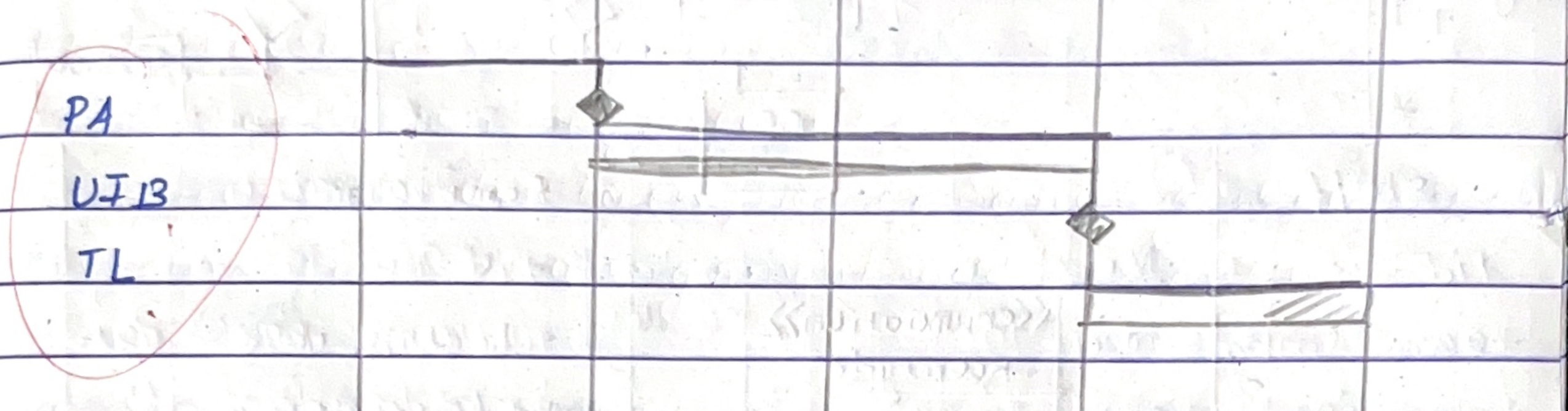
cost per LOC = ₹ 3

∴ Total cost ⇒ 30,006 x ₹ 3

⇒ 21,79,000

⇒ Scheduling :

- Planning + approval (PA) ⇒ 1 week
- UI and backend setup (UIB) ⇒ 2 weeks
- Testing + Launch (TL) ⇒ 1 week



8. Preliminary Schedule and Budget

Schedule (8 months)

Analysis : 1 month
Design : 1 month
Implementation : 3 months
Testing : 2 months
Deployment : 1 month

Budget (~ 50 lakhs)
Infrastructure : 15 lakhs
Development : 20 lakhs
Testing : 7 lakhs
Training goals : 5 lakhs
Maintenance : 3 lakhs/year