

Hotel Management System

Problem statement:

- Managing hotel operations manually can lead to inefficiencies, errors & a poor guest experience. There is a need for an automated Hotel management system that streamlines reservations, check-ins, billing & customer relationship management to improve operational efficiency & enhance guest satisfaction.

SRS

1. Introduction

1.1 Purpose of this document

This document outlines the software requirement for the Hotel Management System (HMS), aimed at streamlining hotel operation & enhancing guest experiences.

1.2 Scope of the document

The ~~HMS~~ document covers functional & non-functional requirements, interface requirements & performance metrics. It also includes preliminary schedule & budget detailing the expected benefits to Hotel management & guests.

1.3 Overview:

The HMS will manage reservations, check-ins, billing and customer relationships, providing a centralized system for hotel staff & guests.

2. General Description

The HMS is intended for hotel staff, managers & guests. It includes features like reservation management, billing, accounting & customer management by automating their tasks, the system improves.

① Functional Requirements:

• Room Service:

A user friendly staff who is trained to provide the utmost for your room.

• Valet Service:

Our hotel has top of the notch valet service. Our hotel has 2 level basement parking system. With a nice structure & great labelling system.

• Buffet system

A extensive collection of exotic food from across the country.

• Event Management:

A separate management system to handle events like birthday & marriage.

Non-functional Requirements

① Modularity :-

Management is broken into smaller modules like guest manager, chief management.

② Performance :-

Run of efficient software facilitate faster

Domain Requirement

① Multiple language support

② Secure System

③ Should conform to local laws & regulations

④ Inconvenience

✓ Credit Card System

Problem Statement:

The credit card system industry faces challenges with transaction speed, security & integration with various banking system. A liable credit card information system is needed to ensure fast transaction, robust security measures. It should ensure secure storage & management of sensitive information.

Scope:-

Storing & managing customer credit card information securely, processing real-time transactions, including authorization, payment & refunds. Detecting & reporting fraudulent activities. Support multiple types of cards.

Functional Requirements

- ① User registration & authorization:
 - users can register, update & delete their account
- ② Credit card management:
 - add, update & remove credit card detail
 - display credit card balance available credit

③ Transaction processing

- real time transaction authorization & approval
- pref processing refunds & charge back

④ Fraud detection

- automatic detection of suspicious transadm
- immediate alerts for suspicious activities

Non-functional requirements

① Security

- data encryption on sensitive information
- role based access control for system

② Performance

- System should handle high volume of transactions per second unit minimal lag time

③ Compliance

- most comply with regional & international functional requirement

Domain - requirement

- ① Credit card information should comply with international guidelines & national guidelines.
- ② Check validity of information via credit card information

11/10/20

01/10/2024

generate SRS

- ① → Library Management
→ Stock Maintenance
→ Passport customization
- ② Identify the classes required for all 5 applications.

Library management

Problem statement :-

Current library management systems store data in a conventional way which makes searching for a required file hard & a lot time consuming. It's also hard to remind the users to return the book. There is a need for interactive and efficient library management which provides user management, staff management, searching literature or audio & video book.

Scope :-

The proposed library management system provides efficient user management, staff management, searching of literature & audio/video book. The proposed library management system solves the problem of poor data storage & management & also

✓ General description:

- User characteristics: librarians, library
- Features: Catalog management, member registration, transaction processing
- Benefits: Enhanced user experience, reduced manual errors

✓ Functional requirements:

• User Management:

User registration & Authentication:

- Ability for users (student/staff, librarians) to create accounts
- Login, logout, password recovery, & user authentication functionalities

• Book Management:

- Book cataloging: Adding, editing or removing books from the system, Maintain detailed records of books such as ISBN, title, authors, genre.

• Search functionality

Book status

• Inventory Management

- Stock tracking: monitor the no of copies available for each book

• Borrowing & Returning

Borrow request

~~Issue book~~

Return Books

Renewals

• Reservations: user can reserve books that are currently borrowed by others

• Late fees & fines:

Fine Calculation

Payment Process

✓ Non functional Requirements:

• Performance

- Response time

- Scalability

- Concurrent Users

• Reliability

- System Uptime

- Data integrity

- Fault tolerance

• Security

- Authentication & Authorization

- Data Encryption

- Backups & recovery

- Audit Trails

✓ Domain requirements

- Library policies:

- loan duration

- renewal limits

- Borrowing limits

- Reservation policies

- Library resource management

- Resource types

- Cataloguing standards

- Special collections

- Fines & fees policies

- late fees calculation

- grace period

- lost / damaged books

Stock Maintenance System

Problem Statement:

Business struggles with stock management due to manual tracking, leading to stockout or overstocking. A stock maintenance system is essential for real-time inventory tracking, automated recording processes & report capabilities to optimize stock levels & reduce operating costs.

Scope:

Encompasses all the functionalities & processes required to track, manage & optimize the inventory levels of products or materials within an organization. This type of system is crucial for ensuring that stocks are efficiently managed, preventing overstocking or understocking & supporting smooth operations.

General description:

- Uses characteristics: Warehouse managers, Sales personnel
- Features:- Inventory tracking, automated recording, reporting
- Benefits:- Optimized stock levels, reduced costs.

Functional Requirements :-

- user can add, update & delete inventory items
- The system alerts users for low stocks
- generate inventory report

Interface requirement:-

- Dashboard interface for inventory tracing
- API for integration with sales

Performance requirement:-

- inventory update
- report generation

Design constraints:-

- needs to be scalable
- needs to integrate with existing system

Non-functional requirements:-

- Reliability: data integrity
- Usability: code should be modular

Preliminary budget & time

- budget - \$50,000 - \$80,000
- time - 5 months

classes :- product, transactions, warehouse etc.

5. Passport Automation System:-

Problem statement: The passport application process can be tedious & time consuming leading to user frustration & administrators delay. A passport automation system is needed to simplify the application process, enhance document verification, & improve tracking resulting in tasks processing times & better customer service.

Introduction :-

1.1 purpose of this document

This document has the software requirements for the passport automation system aimed at streamlining the automation system.

1.2 Scope of this document

The document covers functional & non functional requirements, interface requirements, & performance metrics to enhance the user experience.

1.3 Overview:

The system will simplify the application process, enhance document verification & improve tracking for applications.

2. General description

- User characteristics: Applicants, administration staff
- Features: Application submission, document verification, tracking
 - Benefits: Streamlined processes, improved customer services

3. Functional Requirements

- Users can submit applications online
- System verifies documents & tracks application
- Generates verification for updates

4. Interface Requirements

- Web-based interface for applicants.
- Backend API for administrative tasks & document verifications

5. Performance Requirements

- Application processing time: < 3 days
- System uptime: 94.5%

6. Design constraints

- Must comply with governmental regulations
- Requires secure storage solution for sensitive information

7. Non-Functional Attributes

- security: strict access controls for sensitive data
- compatibility: work with various governmental databases

8. Preliminary schedule & budget:

- Estimated time: 7 months
- Estimated Budget: \$50,000 - \$80,000

dry storage

Manager
username: string
setPassword()
getPassword()

Recuperist
enriching
int
setDetail()
getDetail()

Hotel
name: string
id: int
room: int
checkRoom(): boolean
bookRoom(): void
cancelRoom(): void

Branch
branchName: string
id: int
brand: string
brandAddress: string
computeBranches()

(overlapping)

Employee

(is abstract)

+ manager: int
+ recuperist: int
+ name: string
+ password: string
+ confirmPassword()

Customer

(uses Recuperist)

Bill

(uses Recuperist)

Workshop

(uses Recuperist)

Stock

(uses Recuperist)

+ stockName: string

+ stockId: int

+ status: string

Order

(uses Recuperist)

+ bill: Bill

+ address: rec

+ roomNum: int

+ checkIn: date

+ checkOut: date

Communication

(uses Recuperist)

AC Room

Non AC Room

Book

+ book-id
+ name
+ auth or
+ publisher
+ price
+ date-of-publication
+ date-of-purchase

Works for

Library

+ library-name
+ library-id
+ get details()

+ requests

(overlapping incomplete)

Librarian

+ name
+ phone
+ paymmt
+ address

-

Magazine

+ issue-no
+ year
+ get issues()

+ get issues()

(overlapping incomplete)

+ issuebook()

+ Create

Member

+ member-id
+ name
+ phone
+ thumb印

+ address

+ issuebook()

+ pay

+ borrow

+ return

+ update info

+ custodial

+ updatemember

Transaction

+ transaction_id
+ member_id
+ book_id
+ issue_date

Bill

+ Bill-dt
+ date

+ amount

+ custodial

+ updatemember

+ pay

+ borrow

+ return

+ updateinfo

+ updatemember

+ custodial

+ updatemember

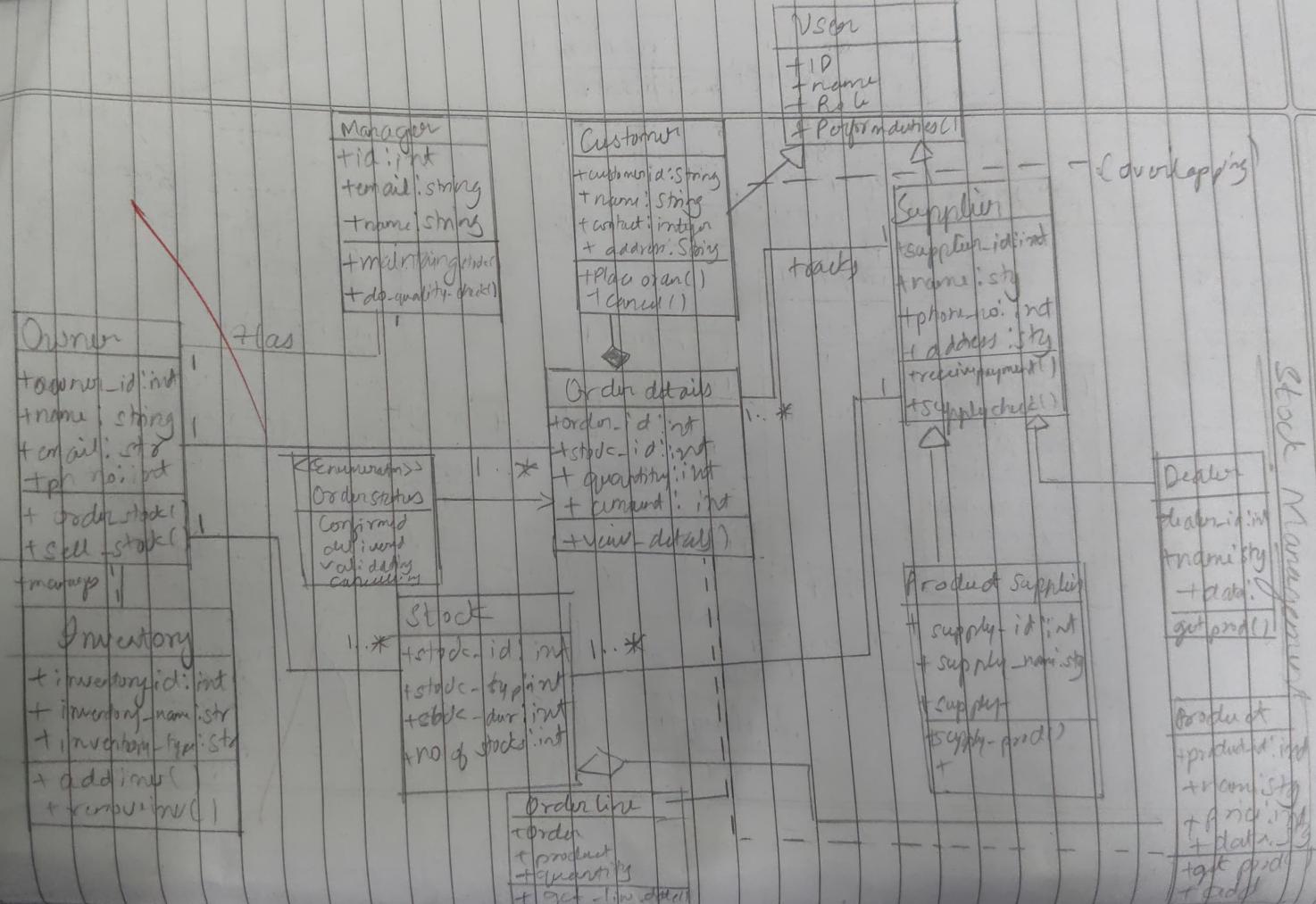
+ pay

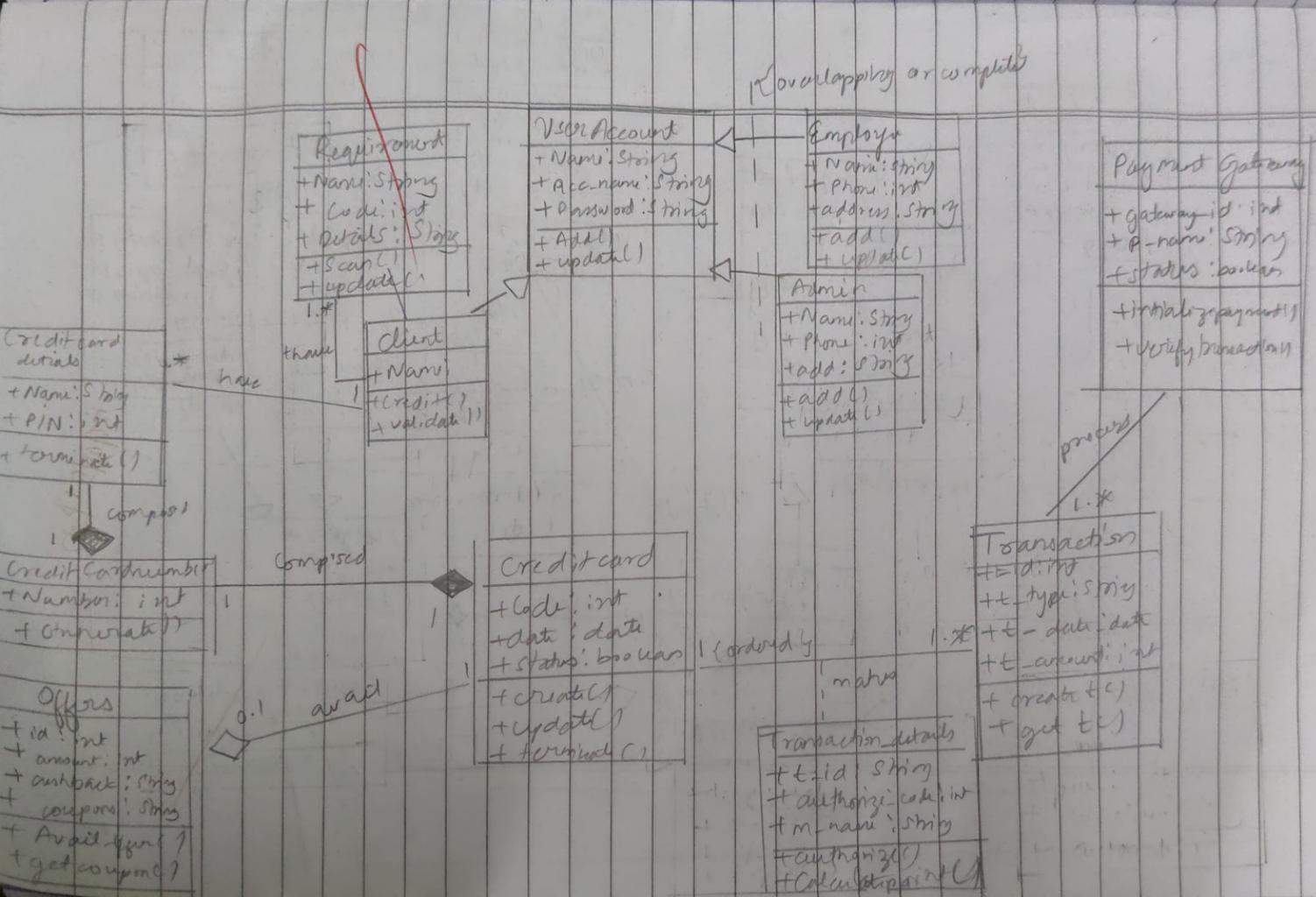
+ borrow

+ return

+ updateinfo

Stock Management





Passport Automation

