lab - 1

19/08/25

1) Hotel Management System

problem statement: (Title)

Hotel face challenges in managing multiple aspects of their operations, including room reservations, guest check-ins and check-outs, billing, inventory management, and staff coordination. many existing solutions are either fragmented or manual, leading to inefficiencies such as double booking, inaccurate billing, delayed service, and poor customer satisfaction.

I. Introduction

→ Purpose:

The document specifies the requirements for the development of Hotel management system to automate & streamline hotel operation such as room booking, guest management, billing, and reporting. the system aims to improve operational efficiency & enhance customer satisfaction.

→ Scope:

The HMS will provide functionalities including room reservation, check-in / check-out process, billing and invoicing, inventory management, staff management, and reporting. It will support hotel administrators, front desk staff, and management personnel.

food
gym
spa
Travel

→ Overview:

this document outlines the functional and non-functional requirements, system interfaces, and constraints for the HMS.

II general description

The HMS will be standalone web-application integrated with payment gateways and possibly third party booking platforms

III functional requirements

- functional

→ the system shall allow guests to search of room availability by date and room type

→ the system shall allow front desk staff to create, update, and cancel bookings.

- non-functional requirements

→ the system shall respond to user requests within 3 seconds

→ the system shall respond to user requests within 99.9 % of the time

Security, Portabi

VI Interface requirements.

User interface : responsive web UI for staff and optionally guests.

Hardware interface : Integration with hotel pos system

Software interface : payment gateway APIs, third-party booking platforms.

V performance requirement.

→ response time ≤ 3 seconds for user actions
→ support 200 concurrent users smoothly
→ scale to 1000 users during peak times
→ reports generated within 5 minutes

VII non-functional attributes

→ usability : easy to use minimal training needed
→ reliability : automatic error recovery.
→ security : role-based access, data encryption
→ portability : works on various OS and cloud platforms.

VIII preliminary schedule

→ requirements : 3 weeks
→ design : 4 weeks
→ development : 8 weeks

preliminary budget

analysis $5,000

design $8,000

development $25,000

testing $7,000

total $58,000.

2) Credit card processing

problem statement:-

current credit card transaction processing systems often face challenges such as slow authorization time, security vulnerabilities, and difficulty in handling high volumes of concurrent transactions these issue can lead to poor user experience.

Introduction

purpose:- this document specifies the requirement for a credit card processing system designed to securely process, authorize & manage credit card transactions.

Scope:-

the ups will handle transaction authorization payment processing, fraud detection, reporting & settlement functionalities.

Description :

ccps is a standalone system integrating with Merchant pos systems, banks, and payment gateways via secure APIs.

Functional requirements :-

→ process credit card payments with authorization response within 2 seconds.

detect and flag fraudulent transactions automatically.

→ Generate daily and monthly transaction reports.

→ support refund & chargeback processes.

performance requirements :-

→ response time per transaction ≤ 2 seconds

→ supports 500 concurrent transactions

→ generate settlement reports within 2 min.

system uptime of 99.9 %

non-functional requirements

→ Usability intuitive interface for merchants & admins

→ automatic error recovery. backup & restore capabilities.

Enterprise requirements

→ Api to integrate with Merchant pos & banking systems.

→ Secure payment gateway connections

Preliminary schedule

→ Requirements analysis 2 weeks

→ System design 3 weeks

→ Implementation 6 weeks.

→ testing 3 weeks

→ deployment & training 1 week.

→ maintenance ongoing

→ Preliminary budget

→ Requirements analysis $4000

→ System design $6,000

→ development $20,000

→ testing $5000

→ deployment & training $2,000

→ total $45,000

design and implementation constraints

- -> Compliance with PCI-DSS
- -> encryption of sensitive data
- -> High availability & fault tolerance.

3) Library Management System:-

Prob. statement :-

the library management system automates cataloging, members, Circulation, reservation, fines, and reporting processes. It ensures data integrity, concurrency control, audit trails, and provides role-based web access for staff and members.

Introduction :-

Purpose :-

- defines requirements for LMS
- for stakeholders, developers, testers, maintainers

Scope :-

- Functions : Catalog, Members, Search, Loans, reservations, fines, notifications, reports, admin
- web based system with concurrent users.

Constraints :-

- enforce concurrency on loans.
- privacy and data retention rules.

functional requirements :

authentication & roles -) register, login, role-based access.

catalog management -) add / update / delete books, copy status.

search & browse -) search by title, author, ISBN, subject, tags

circulation -) issue, return, renew items.

non functional requirements

availability =) $\leq 2s$ response (500 users) ×
↳ 99.5 % uptime

scalability -) scale DB & web tiers

usability =) tasks in ≤ 3 clicks.

maintainability =) modular, REST APIs.

Overview:-

-) The LMS automates cataleg, members management, circulation, fines & reports
-) ensures accuracy, late transcations, better user experience, and audit trails.

Preemptive Schedule

phase 1 (week 1-2) -) requirement analysis & SRS

(week 3-4) -) system design

week 5-8 -) development

week 9-10 -) testing

week 11 -) deployment & training

week 12 -) maintance & support setup.

Preemptive budget

- software tools -) free/open source
- hardware/ server =) shared hosting / local server ₹10,000 - ₹15,000
- development & testing efforts =) 3 months ₹ 2,00,000
- maintance & miscellaneous =) ₹ 20,000/ year.

4) Stock maintenance system

Problem statement

- businesses face difficulty in tracking stock levels, purchases & sales manually
- manual methods lead to errors, overstocking, and stock-outs.
- ensures accuracy, efficiency & real time availability of stock data

Introduction

- stock maintenance system manages stock levels, purchase records, sales records & notification
- provides role-based access: admin, store manager, staff

Purpose

- to define complete requirements of the stock maintance system.
- to provide a guide for developers, testers & users.
- to ensure a clear understanding of system objective & constraints

Scope :-

- automate stock entry, update & withdrawals
- track purchase orders, sales, & supplier information
- provides reports for decision-making
- web based application with user-friendly interface.

overview

- system maintains real-time stock levels
- reduces human errors in stock handling
- provides role-based secured access
- generates alerts & reports for better stock control.

general description

- user → admin, store manager, staff
- system functions : stock entry, update, delete, issues, receive
- environment → web application with database support
- dependencies → barcode scanner, email/sms gateway for alerts.

Functional requirements:

- user authentication & role Management.
- add / update / delete stock items
- track purchase & sales records.
- generate stock level alerts.

Non-functional requirements

- usability → Simple interface for staff with minimal training
- reliability → System must maintain accurate records without data loss.
- security → password protection & role based access.
- availability → 24/7 access with minimal downtime.

Performance requirements

- must handle at least 200 concurrent users.
- Store update / search operations should complete within 2 seconds

Design constraints

- use of relational database
- web-based application
- responsive design
- must comply with company's IT Security policies

Preemptive Schedule

week 1-2 -> requirement gathering & documentation.
week 3-4 -> System design
week 5-8 -> development
week 9-10 -> testing
week 11 -> deployment & training
week 12 -> maintenance setup.

Preemptive budget

software tools => open-source
hardware / Server => ₹15,000
development effort => 3 developers x 3 months
= ₹2,00,000
maintenance & miscellaneous => ₹20,000
total cost => ₹2.3 - ₹2.5 lakhs.

5) passport automation system

problem statement

- manual passport application & processing is slow, prone & inefficient.
- applicant face delays in verification, approval & delivery due to paperwork.

Introduction:

- System designed to manage passport application, verification & issuance online
- role-based users: applicant, verification officer, passport officer, admin.

purpose

- define requirements of passport automation system
- serve as a reference for developers, testers & end-users.
- ensure alignment with government rules & security standards

Scope

- online submission of application & required documents
- appointment scheduling for verification
- passport approval, rejection / request for clarification
- report generation for government agencies.

Overview

- fully web based secure system
- reduces processing time for passport issuance
- eliminates redundant paperwork.

general description

. user -> applicants, police verification officer, passport admin

. system functions -> application submission, verification, tracking

. dependencies -> email / sms gateway, payment gateway

functional requirements

. user registration & login

. online payment of application fee

. passport office approval / rejection

. reports for government agencies

non-functional requirements

. usability -> simple for applicants with minimal training

. reliability -> no data loss, high system uptime

. security -> strong encryption, secure login

. scalability -> handle lakhs of applicants nationwide

Performance requirements

- handle 10,000 + concurrent users
- process an application submission within 5 sec

design constraints

- must comply with government IT security standards
- web + mobile responsive design
- database : RDBMS.

Preemptive Schedule

Week 1-2 -) requirement analysis
Week 3-4 -) system design
Week 5-9 -) development
Week 10-11 -) testing
Week 12 -) Deployment

Preemptive budget

Software tools -) 800 × ₹50,000
hardware/ server -) central server ₹5,00,000
development effort -) ₹6,00,000
Maintenance & security -) ₹ 1,50,000/year
total cost -) ₹12 - 15 lakhs