

Hotel Management System

Lab 1

Problem Statement -

Hotels face issues with processes that can be scattered - reservations, billing and house keeping.

Manual handling make it prone to errors.

The Hotel Management System (HMS) is to centralize these tasks into secure, reliable, and user-friendly platform.

SRS Document, IEEE Standard

Introduction

Purpose :- The aim of this HMS is to automate such as reservations, room management, billing and reporting while improving efficiency and guest and employee satisfaction.

Scope - The HMS should support online booking, check ins/out, billing, staff loads and manage dashboards. It should be scalable from small to large hotels.

Overview - The system will include a custom portal, staff portal, admin dashboard, and integrate with payment gateways & notifications.

General Description

- Client-Serve architecture with a central database, integrate with APIs for payments and notifications.
- System functions include rooms reservations;

availability tracking, billing, housekeeping updates, reporting

- Users include staff, guest and managers
- Constraints include 24/7 uptime, high data security and multi-support
- Assumptions include internet availability and reliable APIs

Specific Requirements

Functional

- Online booking with real time updates
- Staff check in/out management
- Automated billing and payment
- Housekeeping status updates
- Notification via SMS

Interface

- Web and mobile: Responsive UI
- Payment gateway, email/SMS API integration
- Secure HTTPS communication

Performance

- Handle 1000 concurrent users
- Booking /check in results in 3 seconds
- 99.9% uptime

Design Constraints

- Stack: Django/Java backend
- SQL database
- React/Angular: Frontend
- Role based access
- Cloud or on-premises deployment

Non-functional

- 1 Encrypted secure login
- 2 Intuitive interface
- 3 Reliable backups and fault tolerance
- 4 Scalability and multi branch support

Preliminary Schedule and Budget

Design and analysis : 1 month

Core Modules : 2 months

Reports and House Keeping : 1 month

Integration / testing : 1 month

Deployment training : 1 month

Budget

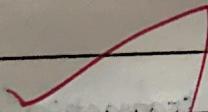
Development \$ 30000

Testing : \$ 7000

Deployment and training : \$ 5000

Maintenance : 1 year : \$ 8000

Total cost: \$ 50,000



Credit Card Processing

Lab 2

Problem Statement

In today's economy, customers expect payments to be fast, seamless and secure. However many businesses struggle with outdated payment systems that are slow, prone to errors and vulnerable to a ~~fast~~ fraud. A modern credit card processing system should handle transactions in real time, verify and authorize them securely, detect fraud activity quickly and ensure funds are settled accurately.

IGE SR Standard System : financial

Introduction

Purpose - To enable secure authorization, processing and settlement of credit card transactions for merchants and customers

Scope - The system will support transaction validation, fraud detection, payment settlement and reporting

Overview - Includes a merchant portal, payment gateway, fraud detection engine and settlement module

General Description

- Works as a 'middleman' between merchants, banks and card networks
- It has functions such as transaction authorization, clearing, settlement, fraud monitoring and reporting

Functional Requirements

- Automate and authenticate transactions
- Process payments in real time
- Provide Fraud detection alerts
- Generate transaction reports

Interface

- 1 Merchant APIs, mobile and web portals
- 2 Integration with banking networks and card issuers

Design Constraints

should be compliant with banking standards

Database: SQL with audit logging

Non-functional

Security

Reliability

Scalability

~~Schedule~~ 6 months

→ Requirements 1 month

Core Modules 2 "

Fraud Detection 1 "

Testing and Integration 1 "

Deployment 1 ,

Budget \$80,000

Requirements and Design \$10,000

Development \$40,000

Security and compliance \$10,000

Testing and QA \$7,000

Deployment and support \$13,000

estimated timeline 3 months

and each iteration

open with the first

iteration of the project

etc.

etc. you have time 3 months

and each iteration

open with the first

iteration of the project

etc. you have time 3 months

and each iteration

open with the first

iteration of the project

etc. you have time 3 months

and each iteration

open with the first

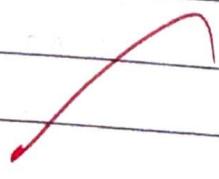
iteration of the project

etc. you have time 3 months

and each iteration

open with the first

iteration of the project



Library Management System

Problem Statement:

Manual Library operations like tracking books, managing members and calculating fines are time consuming and prone to errors.

A Library Management system is needed to digitalize catalogs, automate lending and returns and streamline operations.

Introduction:

Purpose: - To digitalise and streamline book cataloging, borrowing, returning and user management for librarians

Scope: System will manage inventory, user accounts, loan periods, fines and generate reports

Overview: Includes librarian portal, member portal, database of books and search functionality

General Description:

- Client-Server with central book Catalog & database of books
- Functions include catalog management, borrowing/returns, user registration etc

Functional Requirements

- Add, update and delete book Records
- Register member and maintain accounts

- Track borrowing and returning
- Calculate fines for late returns

Interface

- Web and desktop UI
- Barcode scanner integration for books
- Secure login system

Performance

- Support 100+ concurrent users
- Search results in <2s
- 99.9% uptime

Design Constraints

1. Database MySQL

Role based access (member, librarian, admin)

Non-functional

- Usability
- Security
- Reliability
- Scalability

Schedule

5 months → Total

Design : 1 month

Development : 2 months

Testing and Deployment : 2 months

Budget: \$35,000

Requirements and Design: \$4000

Development: \$10000

Testing and QA: \$5000

Deployment and training: \$4000

Maintenance: \$4000

Stock Maintenance System

Problem Statement

Manual processing system: Tracking stock manually is inefficient and prone to errors. A Stock Maintenance System is needed to monitor inventory in real time, automate stock updates, and provide accurate reports for decision making.

IEG Standard SRS

Introduction

Purpose: Automate stock tracking, replenishment, and reporting to reduce errors and improve efficiency.

Scope - The system will manage product details, monitor stock levels, generate alerts for low stock.

Overview - Includes admin dashboard, stock database and reporting module with optional barcode integration.

General Description

- Centralised inventory db
- Functions include stock entry/update, reorder alerts
- Users include store managers, warehouse staff and administration

Functional Requirements

Add/Update Stock Records

Generate low stock alerts

track stock movement

Generate reports

Interface

- Web and desktop UI

- Barcode / RFID integration

Design Constraints

- SQL Database

- Role based access

Non Functional Requirements

1 Reliable

2 Scalable

3 User friendly

Schedule: 5 months

Design: 1 month

Development: 2 months

Testing and Deployment: 2 months

Budget \$35,000

Requirements and Design: \$4000

Development: \$18000

Testing and QA: \$5000

Deployment and training: \$4000

Maintenance: \$4000

Passport Automation System

Problem Statement

Manual Passport application and processing often lead to long queues, delays and errors in document handling. A passport Automation System is required to digitize the process allowing online applications, secure document uploads, automated tracking and faster verification.

Introduction

Purpose - Simplify and digitise passport application, verification & issue.

Scope - The System will support online applications, document verification, payment integration and tracking.

Overview - Includes applicant portal, admin portal, verification system, and integration with government databases.

General Description

E-governance application integrated with gov records. Functions include application submission, status tracking, payment verification and passport issuance. Users include applicants, passport officials and administrators.

Functional Requirements

- Submit applications online
- Upload and verify documents
- Process payments securely

- Track application status

Generate Appointment schedule

Interface

1 Web portal integration

2 Mobile app interface

3 Payment gateway integration

Performance

Must handle 1000+ concurrent users

Design constraints

Must have e-governance security protocols

Non-functional requirements

Must be secure, reliable, user-friendly, scalable

Schedule : 6 months

Design : 1 month

Development : 3 months

Testing : 1 month

Deployment : 1 month

Budget : \$100,000

Requirements and Design : \$2000

Development : \$55,000

Testing and QA : \$10,000

Deployment and Training : \$13,000

Maintenance : \$10,000

✓
16/9