1. HOTEL MANAGEMENT SYSTEM:

Generating SRS DOCUMENT:

03/9/24   OOM LAB

Generating SRS Document:

1) Software Requirement specification for Hotel Management system:

1. Introduction:

1.1 Purpose:

purpose of this document is to define the requirements for Hotel Management system, outlining the function -alities required for efficient hotel operations, such as reservation, check-in and check out, billing and guest management.

1.2 scope:

The system will cater to both hotel staff and customers, managing room bookings, customer profiles, billing and staff management data to day hotel operations, reduce manual labor and provide guests with a seamless experience.

1.3 overview:

The hotel Management system will include modules for front desk operations room management, customer relationship management, house keeping and billing. It will access through desktop or mobile devices; ensuring ease of use for both hotel employees.

2) General Description:

2.1 Product perspective:

The hotel integrates with existing hotel infrastru
and online booking platforms, proving a centralized
system for managing hotel operations.

2.2 User characteristics:

user will include hotel staff. The interface will be
user-friendly for non-techinal users.

2.3 system constrains:

The system will available for 24/7 hours.
support multiple users and handle high traffic during
peak seasons.

3 functional Requirements:

Room booking and reservation management.
customer check in-checkout process.
House keeping status tracking.
Invoice generation and billing.
customer profile and history management
Online booking integration.

4. Interface Requirements.

· Graphical user Interface (GUI) for hotel staff.
· web-based interface.
· Integration with external booking platforms.

5. performance Requirements:

- Must handle at least 500 concurrent users.
· Respone time for room availability should be less
than 2 seconds.

---

5 Design
Must
The
featur
6 Non--
securi
for s
scalab
hotel
7 Rolar
8. per li
Devel
Esti

1)
card

1 Int
1:1 pu
The
CAS
for
wh
an
1.2 Sc
Th
O
Q

5) Design constrains:

Must be compatible with existing hotel hard coarer
The software should be modular to allow future
feature expansion.

6) 7) Non-functional Attributes.

Security: secure data handling with encryption
for sensitive data. details payment.

Scalability: Must be able to scale across multiple
hotel branches.

7) 8) Reliability: 99.9% uptime.

8. Perliminary schedule and Budget:

Development Timeline: 6 months.
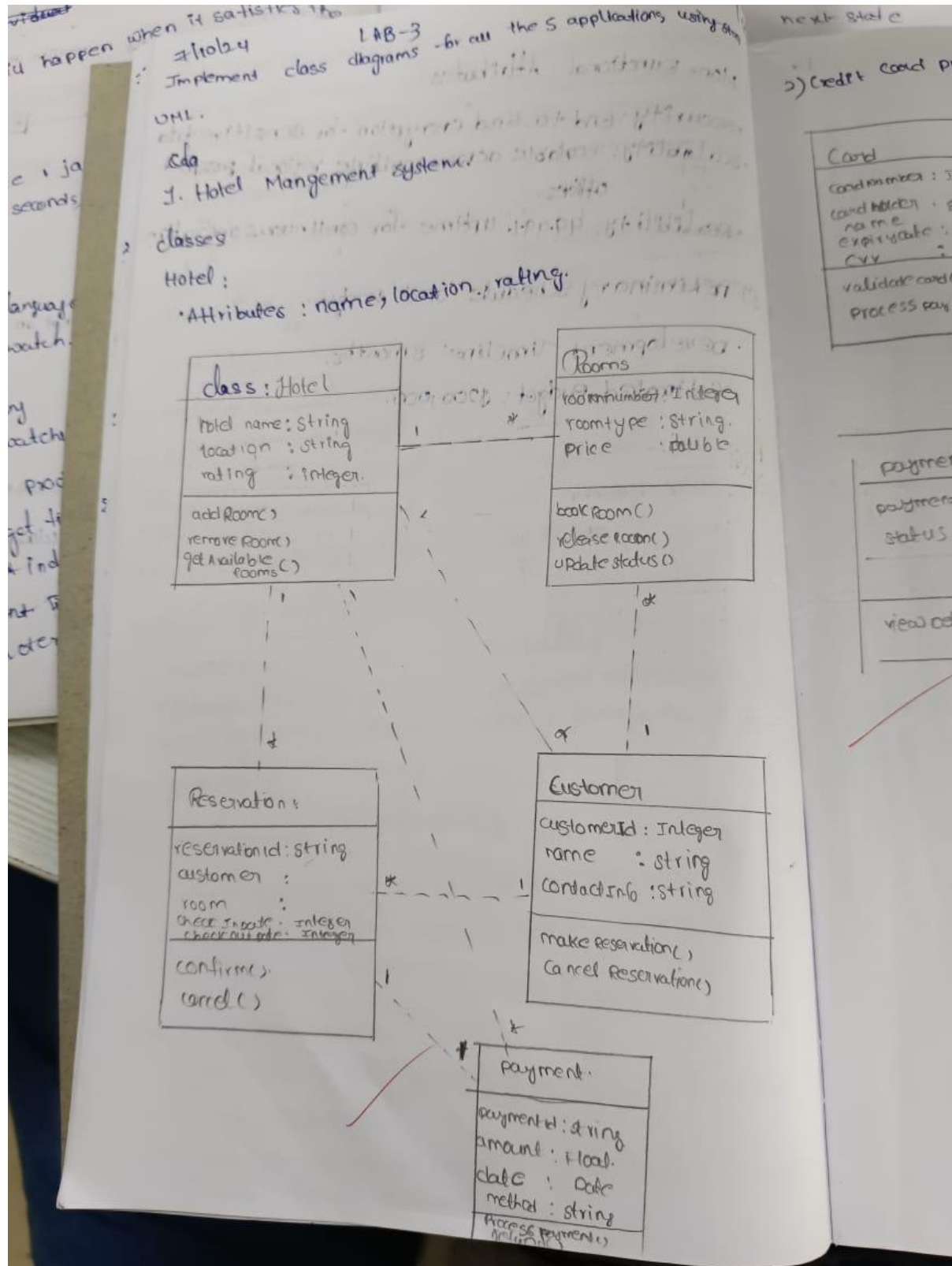Estimated Budget : $100,000 for initial setup and eflo

2) Software Requirement specification for cre
Card processing.

1 Introduction.

1:1 purpose:

The purpose of this document is to do alt line
cred the functional and Non-functional requirem
for a secure credit card processing system,
which transactions between merchan

CLASS DIAGRAM:

LAB-3

Implement class diagrams for all the 5 applications using UML.

&dg

1. Hotel Mangement system

classes

Hotel:

•Attributes: name, location, rating.

class : Hotel

hotel name : String
location : string
rating : Integer

add Room( )
remove Room( )
get Available
rooms( )

Rooms

room number : Integer
roomtype : String
price : double

book Room( )
release Room( )
update status ( )

Reservation :

reservation id : String
customer :
room :
check in date : Integer
checkout date : Integer

confirm( )
cancel ( )

Customer

customerId : Integer
name : string
contact info : string

make Reservation( )
Cancel Reservation( )

Payment

payment el : string
amount : Float
date : Date
method : string
Process payment( )

Card

Card number : I
card holder :
name
expiry date :
CVV :
validate card (
Process pay

payment

payment
status

view o

CLASS DIAGRAM STAR UML:

Handwritten notes page:

Draw the class diagram in Star UML for the applications listed below - class diagram should include attribute, operations, show multiplicity, association class, association, bags, sequence, aggregation, composition.

3. Library Management System.
4. Stock Maintance System.
5. Passport Automation System.

Library Management System:

LiBrary
+ name : string
+ address : string
+ add - book ()
+ remove - book ()
+ manage - book ()

Reading Material
+ id : int
+ name : string
+ status : string
+ borrow ()
+ return ()
+ rate ()

+ has ▶
◀ + are present in {ordered}

USE
+ id : int
+ type : string
+ date : date
+ read ()
+ lend ()
+ return ()

◀ + managed by
+ manages ▶

LiBraRian
+ name : string
+ id : int
+ issue ()
+ manage ()

USERS
+ id : int
+ name : string
+ dept : string
+ borrow ()
+ return ()
+ read ()

BOOKS
+ P. id : int
+ request . ecopy ()
+ borrow ()
+ return ()
+ rate ()

◀ has

PAPERS
+ type : string
+ download ()
+ borrow ()
+ return ()
+ rate

PPT Slides
+ topic : string
+ download ()
+ borrow ()
+ return ()
+ rate ()

Students
+ sem : int
+ study ()

Staff
+ post : string
+ share ()