****

**RAMNIRANJAN JHUNJHUNWALA COLLEGE GHATKOPAR (W), MUMBAI - 400 086**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**2022 - 2023**

**s.Y. B. Sc.( I.T.) SEM iv**

**Paper RJSUITp404 – Software Engineering**

**Name :- Rajbhar Sudesh Dinesh Sushiladevi**

**Roll No. 3067**

**Hindi Vidya Prachar Samiti’s**

**RAMNIRANJAN JHUNJHUNWALA COLLEGE**

**Ghatkopar (W), Mumbai-400 086**

***Certificate***



**This is to certify that Mr./Ms. Rajbhar Sudesh Dinesh SushilaDevi Roll No 3067 of S.Y.B.Sc.(I.T.) class has completed the required number of experiments in the subject of Software Engineering in the Department of Information Technology during the academic year 2022 - 2023 .**

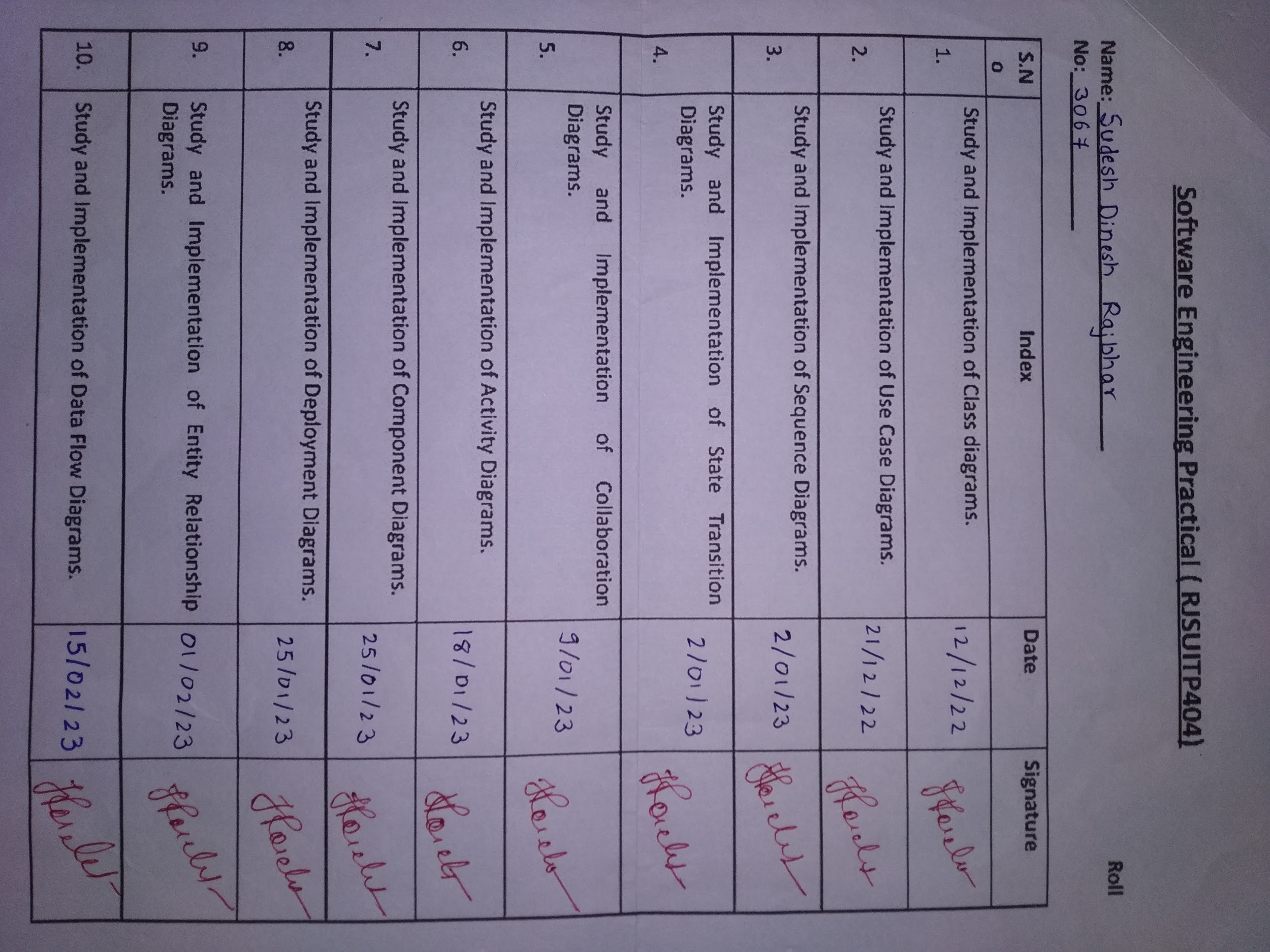
**Professor In-Charge Co-ordinator of IT Department**

**Prof. Bharati Bhole**

**Prof. Archana Bhide**

**College Seal & Date Examiner**

**Software Engineering Practicals Journal**



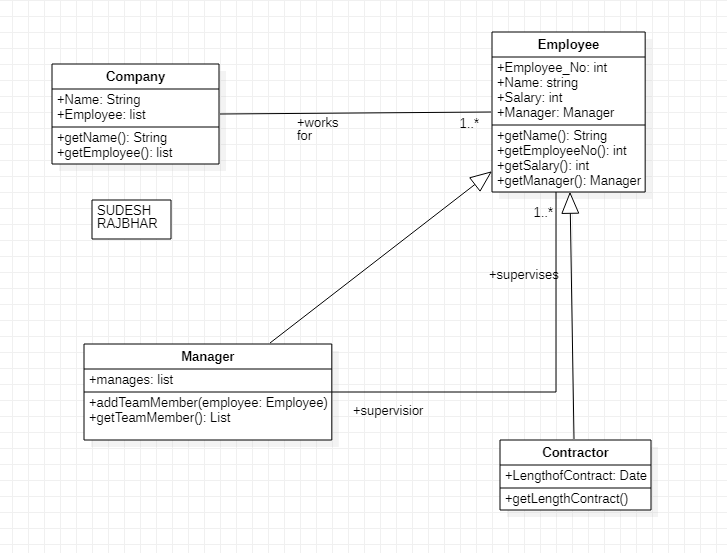
### **Practical 1**

1. Study and implementation of class diagrams.

#### 1)Draw a class diagram using StarUML for the scenario given below.

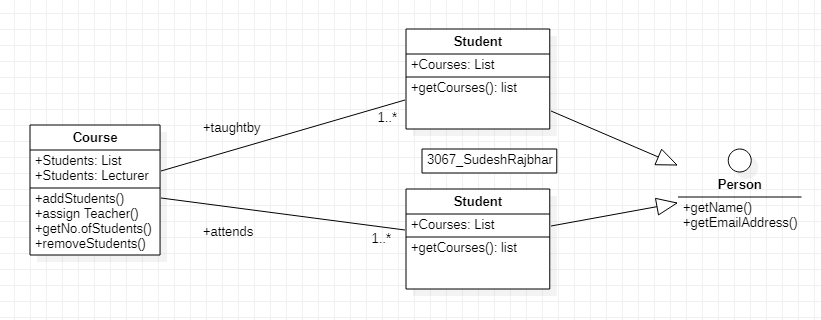
This scenario is from a system that models companies for a payroll or reporting system. Company object has properties such as name and employees\_list and getName and getEmployees as its behavior. Employee object includes employee no, name, salary and manager as its properties getName (), getEmplyoeeNo () ,getSalary() and getManager() as its methods. getManager() accepts the object of manager. Company may have one or more employees.

A manager object keeps managers as list property and adds TeamMember(employee\_list) and getTeamMember() as its behaviors. One or more employees can be managed by manager objects. Some employees are contractual employees who are within a lieu of a contractor object. A contractor object may have length\_ of \_contract as its property and getLength() as its behavior.



#### 2)Draw a class diagram for the scenario given below.

This is an example that models University Courses. Assume three classes’ such as course, lecturer, student and an interface person. Each course object maintains a list of students on that course and lecturer who has been assigned to teach that course. The course object has behavior that allows adding and removing students to and from the course, assigning the teacher and getting a list of currently assigned student and currently assigned teacher. A teacher may teach several courses but a course only has a single teacher .A lecturer object maintains a list of courses that it teaches, courses are attended by 0 or more students and students may attend multiple courses. A person interface will have getName() and getEmailAddress () methods both lecturer and student are shown to be the type of person.

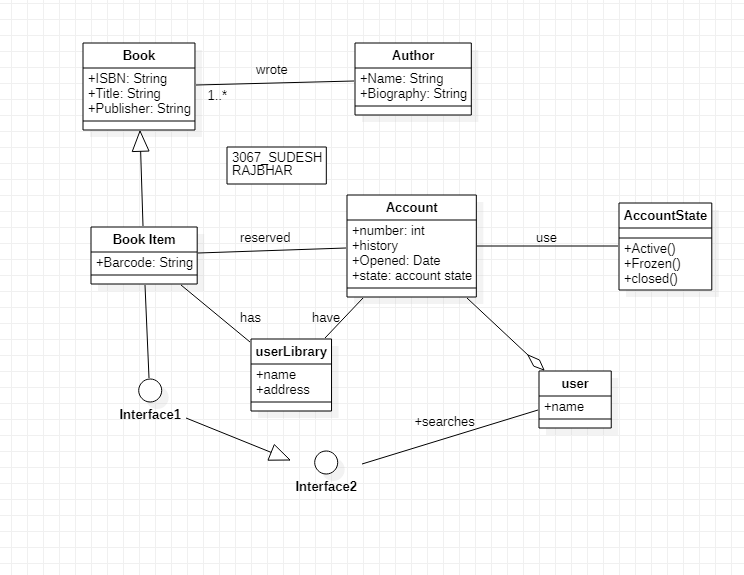


#### 3)Create a class diagram(Use Star UML) for “library management”

using the classes with their attributes and operation given below. Also set the appropriate relationship between the classes using the relationship tools from the toolbox. following the overview of the system given below.

Overview of the system:-

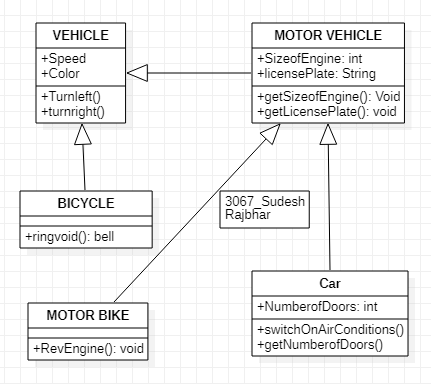
1. It has a class called “Book”. Book has authors so it has an “Author” class.
2. In order to collect book information it has a “BookItem” class which uses some of the properties from the book class.
3. It needs an account for reserving books by the user so it has an “Account class.”
4. In account class there is an attribute named state which uses an enumeration named “AccountState”.
5. It also has a class “Library” to manage the account, user and the books.
6. It has a user class to manage the user detail that has an account in the library and he can borrow and return books to the library.
7. The system also has an interface “Search” where the user searches the book he needed from the “Catalog” class.



#### 4) Draw a class diagram using StarUML for the scenario given below.

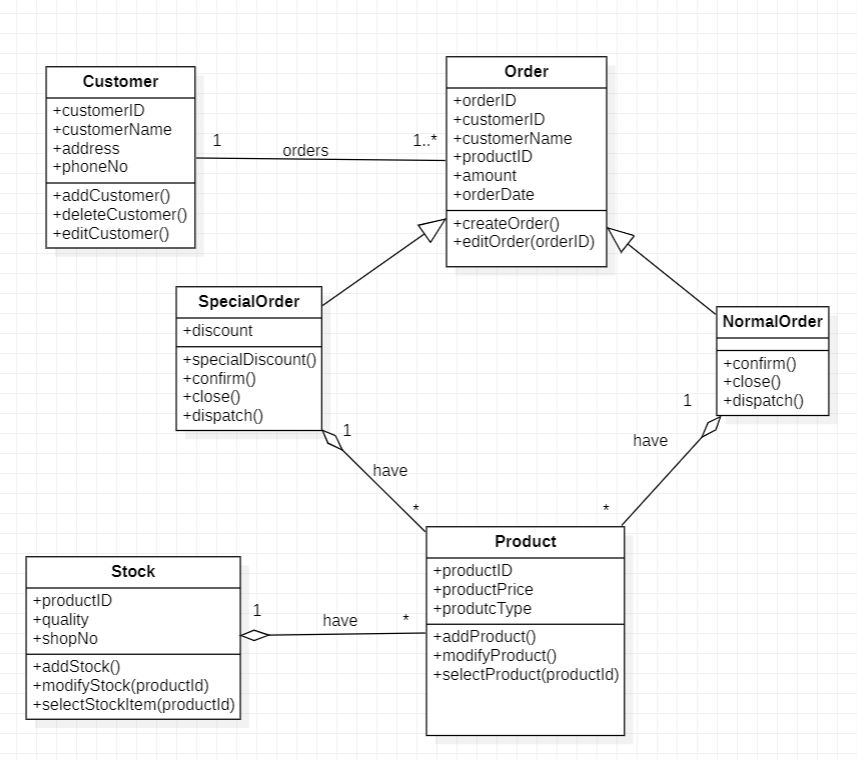
#### 

This scenario shows an inheritance hierarchy of a series of classes and their subclasses. It’s for an imaginary application that must model different kinds of vehicles such as bicycles, motor bikes and cars. All Vehicles have some common attributes (speed and color) and common behavior (turnLeft, turnRight). Bicycle and MotorVehicle are both kinds of Vehicle and are therefore shown to inherit from Vehicle. To put another way, Vehicle is the superclass of both Bicycle and MotorVehicle. In our model MotorVehicles have engines and license plates. Attributes have been added accordingly, along with some behavior that allows us to examine those attributes. MotorVehicles is the base class of both MotorBike and Car; therefore these classes not only inherit the speed and color properties from Vehicle, but also the additional attributes and behavior from MotorVehicle. Both MotorBike and Car have additional attributes and behavior which are specific to those kinds of objects.



#### 5) Draw a class diagram using StarUML for the scenario given below.:

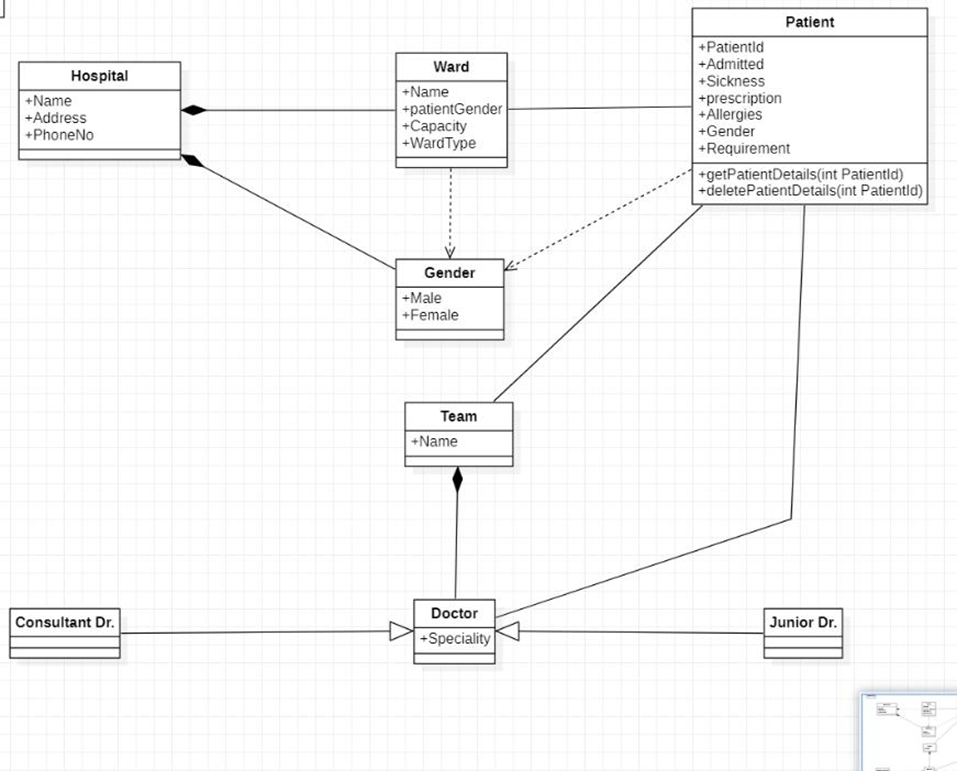
This is an example that models ``ORDER MANAGEMENT”. The Customer object has properties such as CustomerId, CustomerName, Address and Phone and methods such as AddCustomer(),DeleteCustomer() and EditCustomer(). Orderobject includes OrderId, CustomerId, CustomerName, ProductId, Amount and OrderDate as its property and CreateOrder() and EditOrder(OrderId) as its behavior. A customer can place one or many orders. Further there areSpecialOrder objects and NormalOrder objects which have the same methods CreateOrder(), confirm(), close(), dispatch() whereas the SpecialOrder object also has one property named SpecialDiscount. SpecialOrder and NormalOrder objects are both kinds of order and are therefore shown to inherit from order entities. Moreover the system also has Product entities having attributes such as ProductId, ProductPrice, ProductType and methods such as AddProduct(), ModifyProduct() and SelectProduct(ProductId). Stock objects have properties like ProductId, Quality and ShopNo and behavior such as addStock(), ModifyStock(ProductId) and slectStockItem(ProductId). Note that specialOrder and NormalOrder have 1 or more products whereas stock has many products.



#### 6) Draw a class diagram using StarUML for the scenario given below.

This is an example that models “Hospital Management”. The ward object of this system has attributes such as name, patient-gender and capacity. Note that patient- gender is a gender type which is an enumeration containing enums male and female.The system also has Patient entity with attributes such as patient\_id, admitted, sickness \_history, prescriptions, special\_reqs and allergies and gender which is again a gender enumeration type. And operations such as getPatient() and deletePatient(Patient\_id).Ward is a division of a hospital object having attributes such as name address and phone number. In hospital there are number of wards

each of which may be empty or have one or more patients. Each ward has a unique name. This ward is shared by patients who need a similar kind of care. Each patient is on a single ward. The system also has a Doctor entity which is further classified into Consultant Doctor and Junior Doctor. The doctors in the hospital are organized into Teams entities with attribute team\_name. Each team can have two or more doctors. Each patient is under the care of a single team of doctors. A patient may be treated by any number of doctors but all the doctors must belong to the same team that cares for the patient. Note that the team is owned by the hospital.

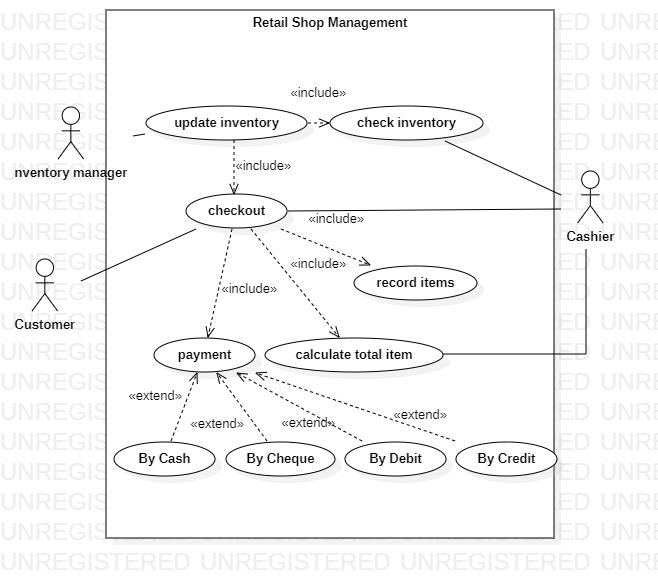


### **Practical 2**

1. Study and implementation of Use Case Diagrams.

#### 1)Construct a Use Case Diagram for Point of sale terminal management system that can be used for buying and selling of goods in the retail shop.

When the customer arrives at the post check point with the items to purchase. The cashier records each item, price and adds the item information to the running sales transaction. The description and price of the current item are displayed. On completion of the item entry the cashier informs the sales total and tax to the customer. The customer chooses payment type (cash, cheque, credit or debit).After the payment is made the system generates a receipt and automatically updates the inventory. The cashier handovers the receipt to the customer.

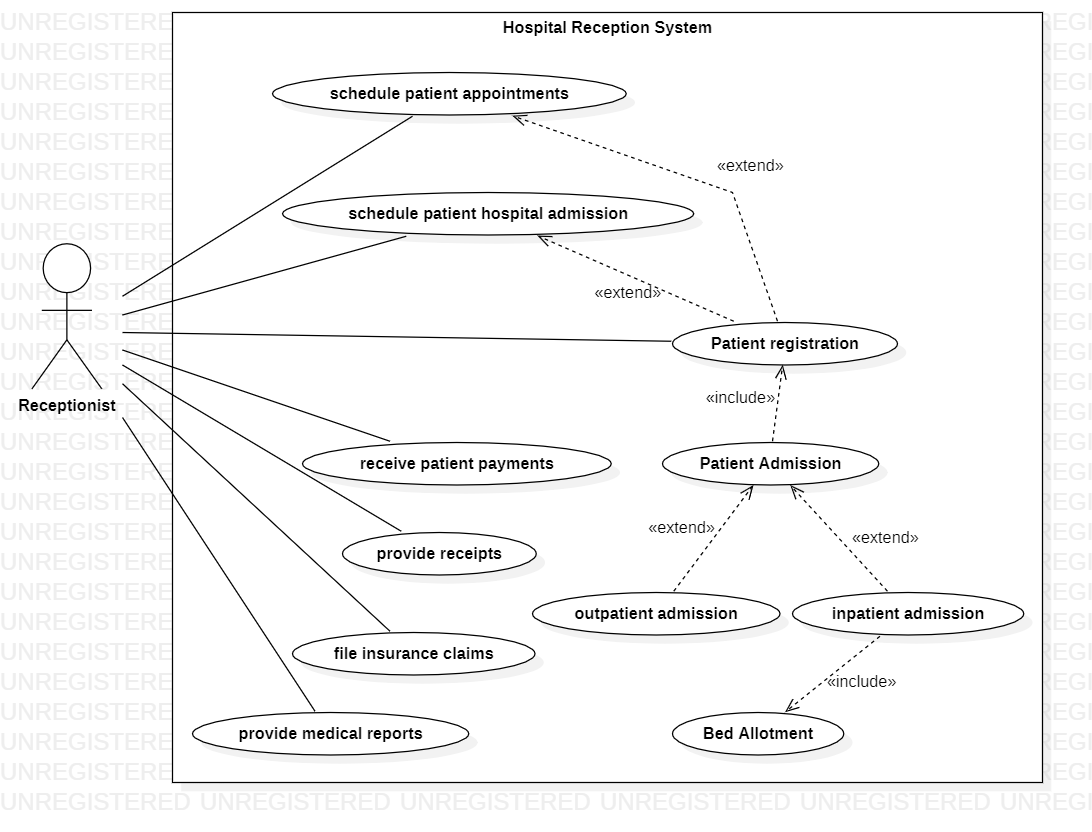


#### 2)Draw a use case diagram for:

"**Hospital Reception Subsystem explained below:**

"Hospital Reception Subsystem" supports some of the many job duties of hospital

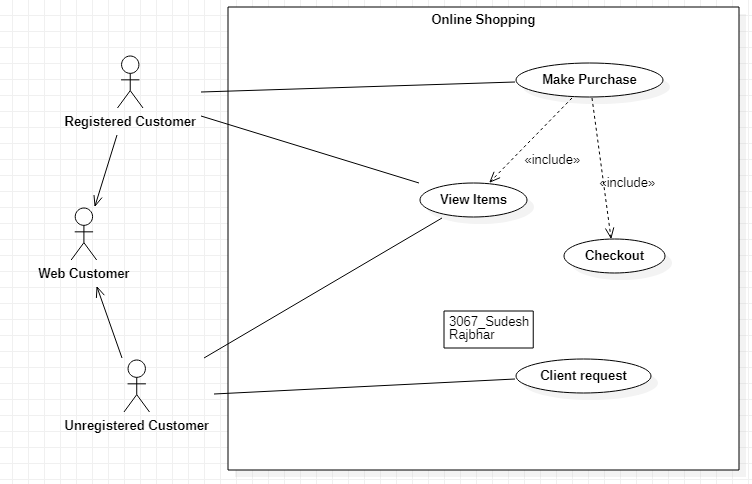
receptionist. Receptionists schedule patient's appointments with the doctor and also schedule patient hospital admission. If a doctor is available and admission to the hospital is possible then the receptionist can extend the service to patient registration by collecting the patient information on patient arrival or over the phone. Patient registration is an integral part of patient Hospital Administration use cases. Hospital administration use cases are further generalized into outpatient hospital admission and inpatient hospital administration. Note that for the patient that will stay in the hospital, he or she should have a bed allotted in a ward. Receptionists might also receive patient's payments, record them in a database and provide receipts, file insurance claims and medical reports.



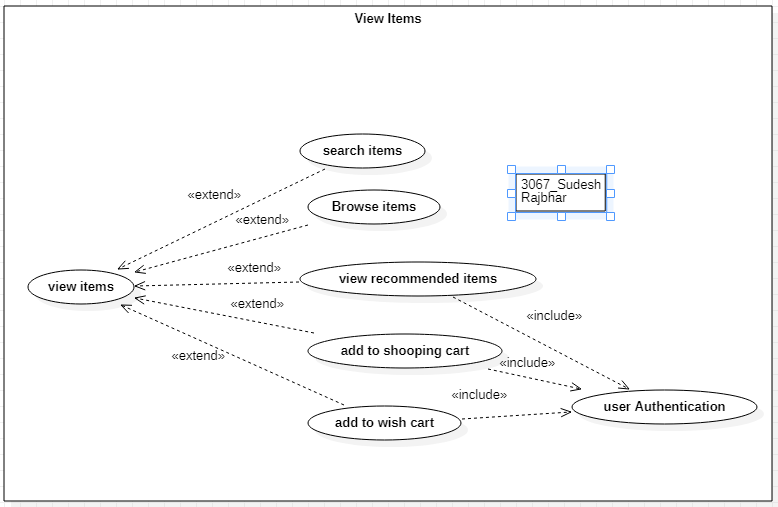
#### **3)Web Customer** [**actors**](about:blank) use some websites to make purchases online.

Top level [**use cases**](about:blank) **are View Items, Make Purchase and Client Register**. View Items use case could be used by the customer as a top level use case if the customer only wants to find and see some products. This use case could also be used as a part of Make Purchase use case. Client Register use case allows customers to register on the web site, for example to get some coupons or be invited to private sales. Note that **Checkout** use case **includes a** [**use case**](about:blank) not available by itself - checkout is part of making a purchase. Except for the **Web Customer** actor there are several other actors which will be described below with detailed use cases. **View Items** use case is [extended](about:blank) by several optional use cases - customers may search for items, browse catalog, view items recommended for him/her, add items to shopping cart or wish list. All these use cases are extending use cases because they provide some optional functions allowing customers to find items. **Customer Authentication** use case is [included](about:blank) in **View Recommended Items** and **Add to Wish List** because both require the customer to be authenticated. At the same time, items could be added to the shopping cart without user authentication. **Checkout** use cases include several required use cases. Web customers should be authenticated. It could be done through a user login page, user authentication cookie ("Remember me") or Single Sign-On (SSO). Web site authentication service is used in all these use cases, while SSO also requires participation of external identity providers. **Checkout** use case also includes **Payment** use case which could be done either by using credit card and external credit payment service or with PayPal.

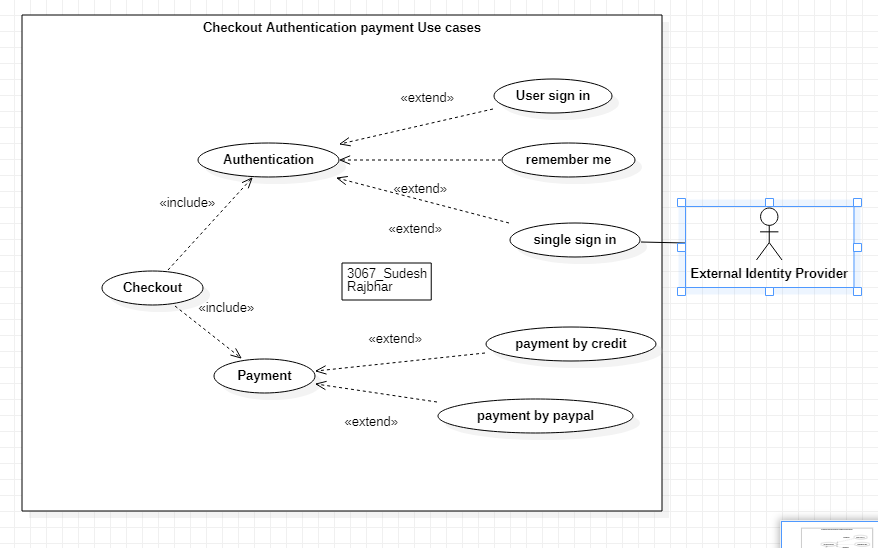
Draw top level use case, View Items Use Case, Checkout, Authentication and Payment Use Cases for the above scenario.

****

**Top level Use case**

****

**View items use case**

****

**Checkout , Authentication,payment Use cases**

#### **4)DRAW A USE-CASE DIAGRAM USING STARUML FOR THE “MUSIC PORTAL SYSTEM**” **DEPICTED BELOW.**

The following narration describes some of the use cases for “**Music Portal System**”. This system has a web **user** as its main actor. The web user can perform first level use cases namely **Search Album, login, logout** and **ViewAccount**. **Registration** use case extends login i.e. if the user doesn’t have a login and wishes to create a new one, he or she can register to get a login. Moreover the **View Album Detail** use case is extending the Search Album use case. Further the View Album Detail use case is extended by two more services viz**. Download Album** and **Buy Album**. Note that to download or to buy an album the user must be a registered member. Moreover the buyalbum and ViewAccount is further extended with R**echarge** use case



### 

### Practical 3

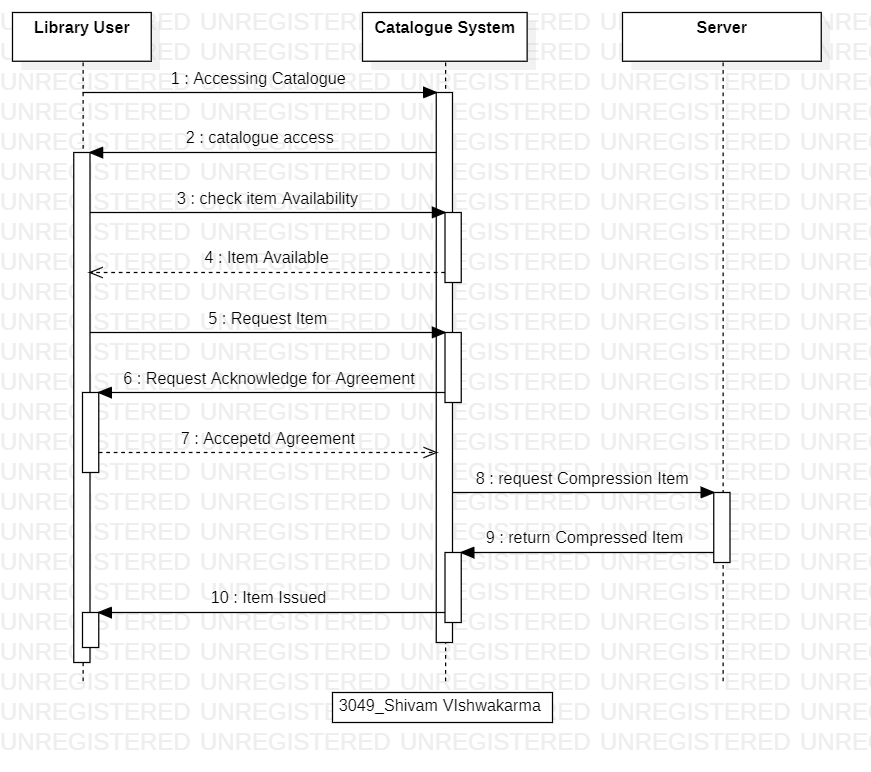
### 

#### 1)Study and implementation of Sequence Diagrams.

the library user accesses the catalog to see whether the item required is available electronically; if it is, the user requests the electronic issue of that item. For copyright reasons, this must be licensed so there is a transaction between the item and the user where the license is

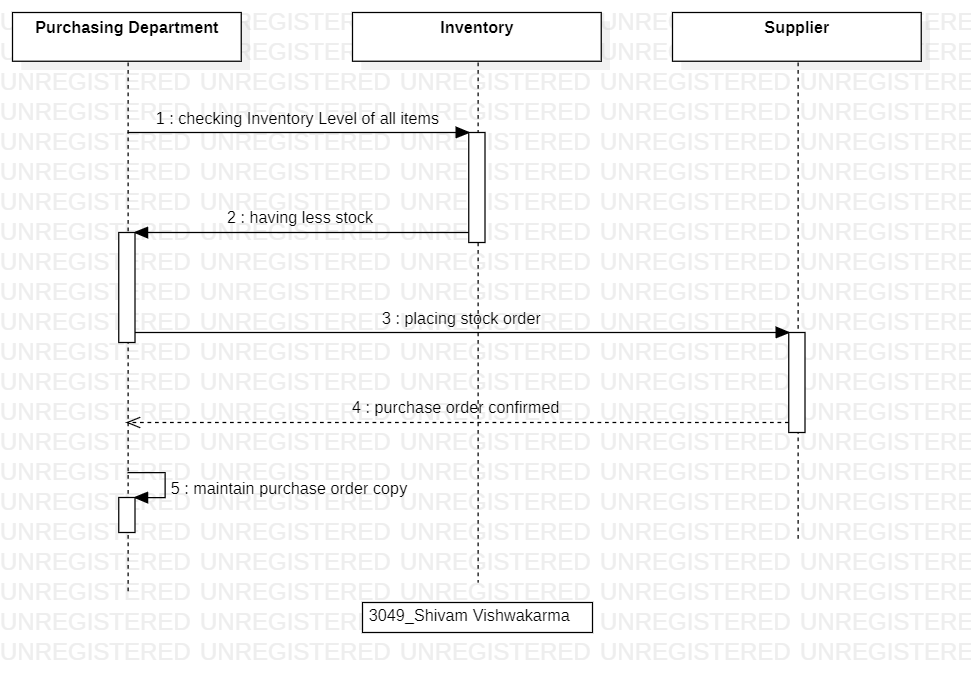
agreed. The item to be issued is then sent to a network server object for

compression before being sent to the library user.



#### 2)Draw a Sequence diagram for Placing Purchase Order :-

Purchasing department checks the inventory level of all the items. If stock is less than the department searches the supplier list, generates the purchase order and sends it to the supplier. Purchasing department also maintains the copy of the purchase order.



### 

### Practical 4

### 

#### 1)Study and implementation of State Transition Diagrams.

1.Draw State machine diagram for vending machine. The active object in the system is the controller. The states of the controller are:

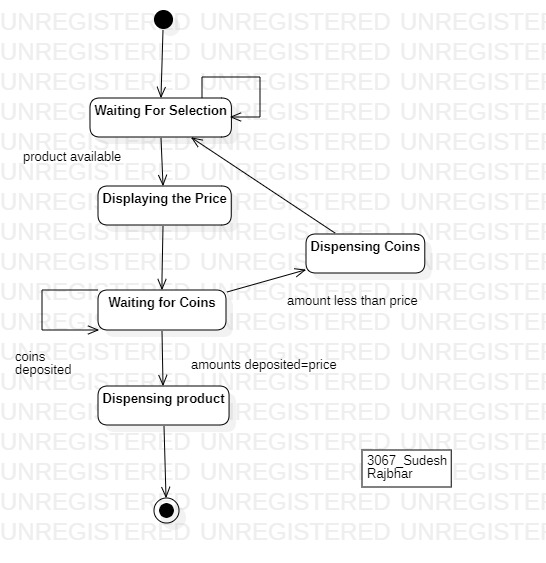
**1.Waiting for selection:** The controller will be in this state till the time the customer is not selecting any product. If the product is available then it will go to displaying price state. Otherwise it will return to the same state and a message “Product not available” will be displayed.

**2.Display Price:** The product which is selected, price of it will be displayed.

**3.Waiting for coins:** The coins are deposited by the customer. When the stipulated time is over, the state will be changed.

**4.Dispensing Product:** If the amount deposited is equal to or greater than price of product then product will be dispensed in this state.

**5. Dispensing Coins:** if the amount deposited is less than price then, an error message “Amount deposited is less” will be displayed and coins will be dispensed.



### 

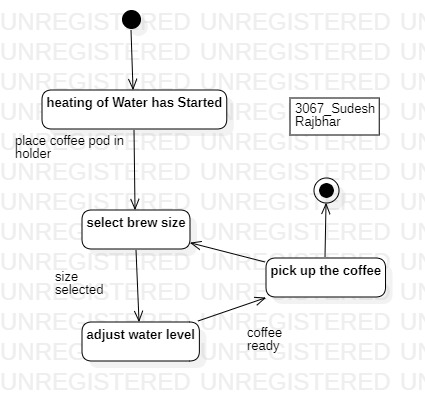
#### 2) Following is an illustration for the State Transition diagram of a coffee brewing

machine. Once the machine is turned ON, it starts to heat the water. You then need

to place a coffee pod in the pod holder. We need to select the brew size and after

where the water output is adjusted and coffee is brewed. Finally, the coffee brew is

Completed.



#### 

#### 3) Following is an illustration for the State Transition diagram of a Toaster oven

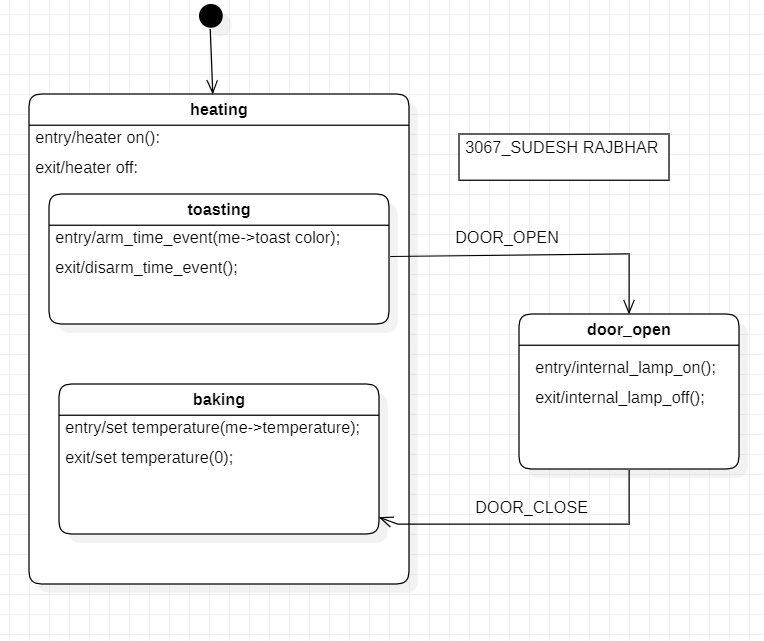
state machine with entry and exit actions. For example, consider the “door\_open”

state which corresponds to the toaster oven behavior while the door is open. This

state has a very important safety critical requirement: Always disable the heater

when the door is open. Additionally, while the door is open, the internal lamp

illuminating the oven should light up



### 

### Practical 5

### 

#### 1)Study and implementation of Collaboration Diagrams.

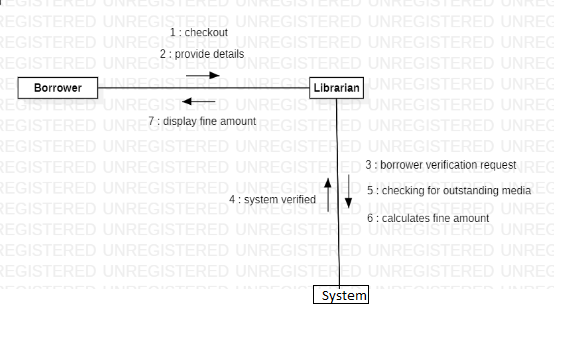
1. Draw a Collaboration diagram to depict the interaction that takes place between

systems when a librarian calculates the fine for the borrower of an online media

library, after the due date is elapsed. Librarian first enters the borrower’s id and the

system then verifies it. It then checks for any outstanding media for the borrower and

calculates the fine amount and displays it.



#### 2)Draw a Collaboration diagram for Defective shipment :-

The S/R clerk sends the verified shipment to the inspector. Inspector searches the quality

criteria and verifies the shipment with it. If the quality is satisfied then the accepted

list is prepared and given to the inventory clerk else the defective items are sent

back to the supplier with a defective item notice, prepared by the inspector.



### 

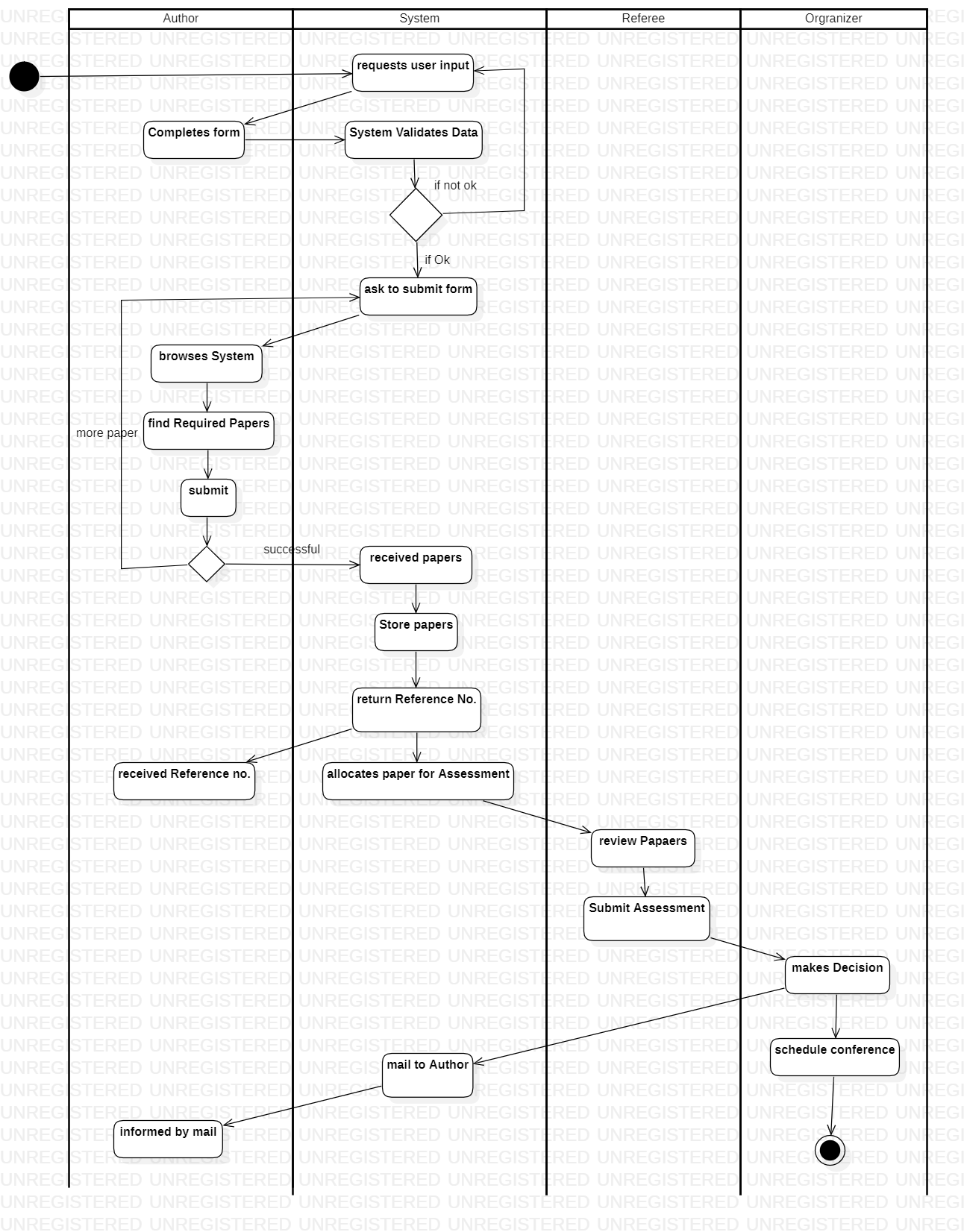
### Practical 6

### 

#### 1)Study and implementation of Activity Diagrams.

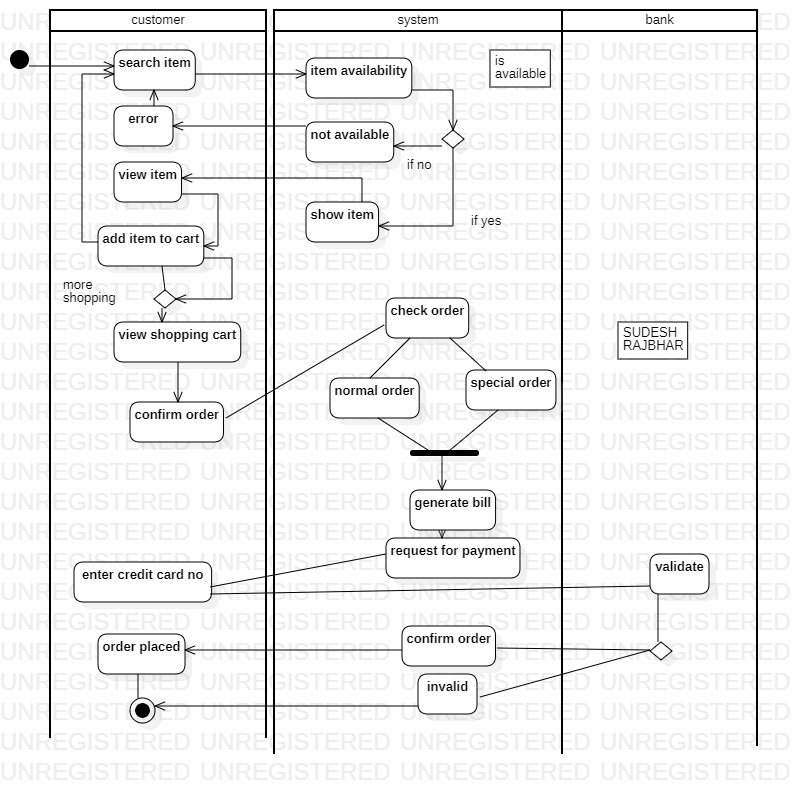
1. Draw an activity diagram from the narrative text on “ONLINE PAPER SUBMISSION SYSTEM”.

The author completes an online form that requests the user to input author name, Correspondence address, email and title of paper. The system validates this data and, if correct, asks the author to submit the paper. The author then browses to find the correct paper on their system and submits it. Once received and stored, the system returns to the author a reference number for the paper. Authors may submit as many papers as they like to be considered for acceptance to the conference up until the deadline date for submissions. Papers are allocated to referees for assessment. They review each paper and submit to the system their decision. Once the program organizer has agreed the decisions, authors are informed by email. Accepted papers are then scheduled to be delivered at a conference. This involves allocating a date, time and place for the presentation of the paper.



#### 2)Derive an Activity diagram using star UML from the case given below on “order processing subsystem“.

Web Customers use some websites to make purchases online. Where customers can search items, View items, add items to the cart, place orders and make payment. For placing an order, the customer first searches the required items from the system. As and when the customer finds the item available in the system he starts adding the item to the chart. The System provides facility to the Customer to add any number of items to the chart. Customer can also view his shopping cart containing items. Once the customer finishes his shopping he can place the order by requesting a system to confirm the order. The system will then check whether the order is normal order or any special order and according to that the system will generate the bill and request for payment. After getting the bill the customer can make payment. The bank will validate the credit card number. If the credit card number is valid the system will confirm the order. Otherwise the process will get terminated.



### Practical 7

### 

#### 1)Study and implementation of component Diagrams.

Passport Automation System is used in the effective dispatch of passports to all of the applicants. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in a cogent manner. The core of the system is to get the online registration form (with details such as name, address etc.,) filled by the applicant whose testament is verified for its genuineness by the Passport Automation System with respect to the already existing information in the database.

SOFTWARE REQUIREMENT SPECIFICATION:

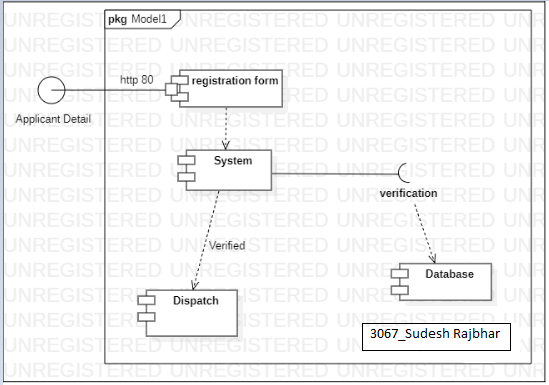
SOFTWARE INTERFACE

* Front End Client - The applicant and Administrator online interface is built usingJSP and HTML. The Administrators local interface is built using Java.
* Web Server - Glassfish application server
* Back End - Oracle database.

HARDWARE INTERFACE

The server is directly connected to the client systems. The client systems have

access to the database on the server.



#### 2)Study and implementation of component Diagrams.

The recruitment system allows the job seekers to enroll their names through the

process of registration. The employee also can get the list of available candidates

and shortlist for their company requirement. Once the applicant enrolls he receives

an id, which helps him in further Correspondence. A fees amount is received from

the job seekers for enrollment. This system makes the task of the job seeker easier

rather than waiting in queue for enrollment. This also reduces the time consumption

for both the job seeker and employee.

SOFTWARE INTERFACE

• Front End Client - The applicant and Administrator uses online interface built using

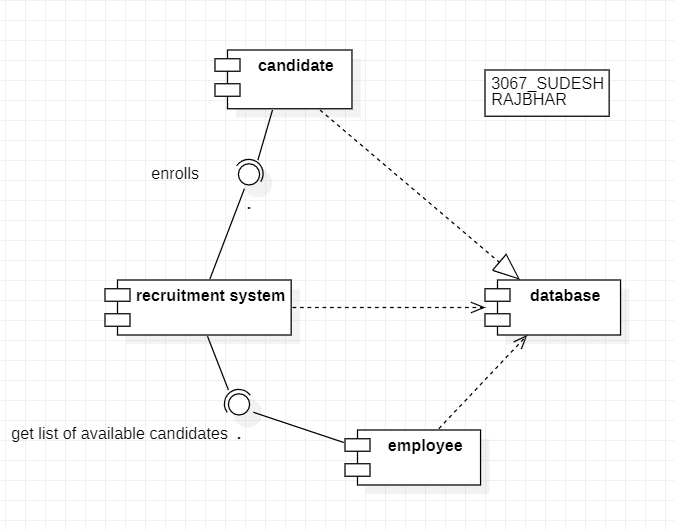
PHP and HTML.

• Web Server - Xampp server

• Back End - MySQL database.

2.3HARDWARE INTERFACE The server is directly connected to the client systems.

The client systems have access to the database in the server.



### Practical 8

### 

#### 1)Study and implementation of deployment Diagrams.

Passport Automation System is used in the effective dispatch of passports to all of the applicants. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in a cogent manner. The core of the system is to get the online registration form (with details such as name, address etc.,) filled by the applicant whose testament is verified for its genuineness by the Passport Automation System with respect to the already existing information in the database.

SOFTWARE REQUIREMENT SPECIFICATION:

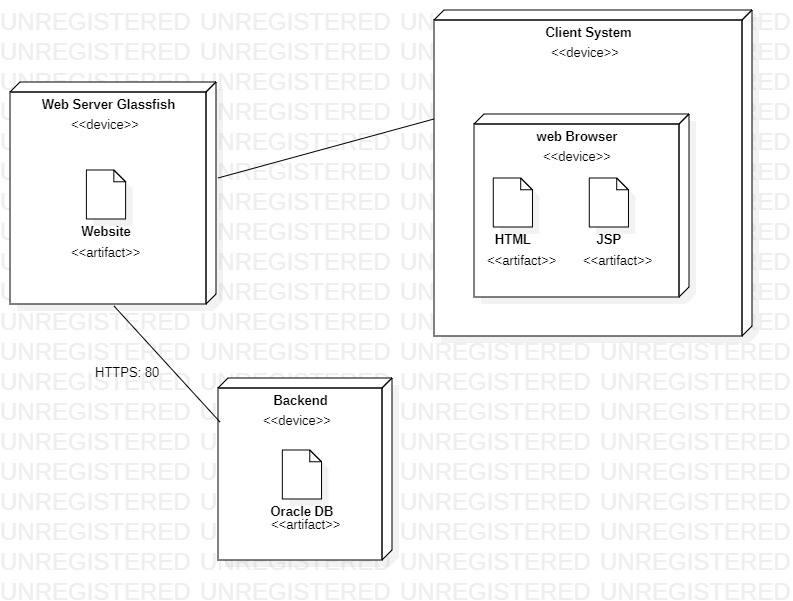
SOFTWARE INTERFACE

* Front End Client - The applicant and Administrator online interface is built usingJSP and HTML. The Administrators local interface is built using Java.
* Web Server - Glassfish application server
* Back End - Oracle database.

HARDWARE INTERFACE

The server is directly connected to the client systems. The client systems have

access to the database on the server.



#### 2)Study and implementation of deployment Diagrams.

The recruitment system allows the job seekers to enroll their names through the

process of registration. The employee also can get the list of available candidates

and shortlist for their company requirement. Once the applicant enrolls he receives

an id, which helps him in further Correspondence. A fees amount is received from

the job seekers for enrollment. This system makes the task of the job seeker easier

rather than waiting in queue for enrollment. This also reduces the time consumption

for both the job seeker and employee.

SOFTWARE INTERFACE

• Front End Client - The applicant and Administrator uses online interface built using

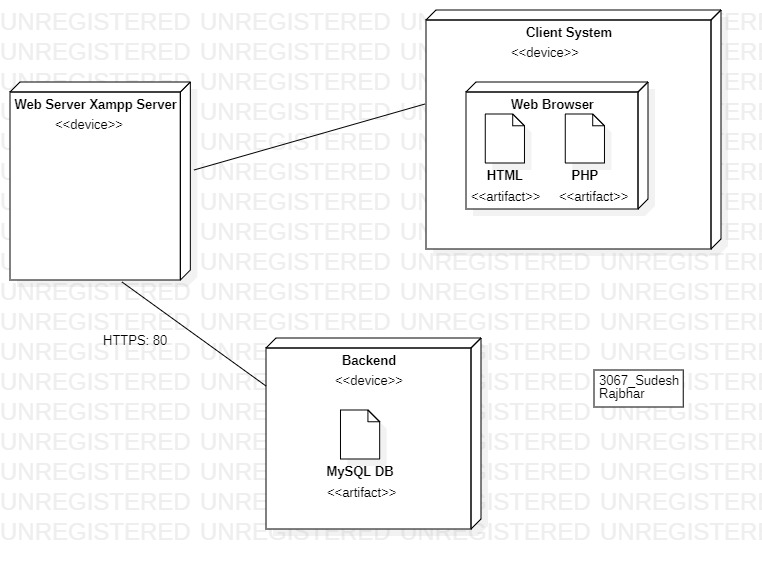
PHP and HTML.

• Web Server - Xampp server

• Back End - MySQL database.

2.3HARDWARE INTERFACE The server is directly connected to the client systems.

The client systems have access to the database in the server.



### Practical 9

### 

#### 1)Study and implementation of Entity Relationship Diagrams.

1.Consider a university database for the scheduling of classrooms for final exams.

This database could be modeled as the single entity set exam, with attributes course-name, section-

number, room-number, and time.

Alternatively, one or more additional entity sets could be defined, along with relationship sets to

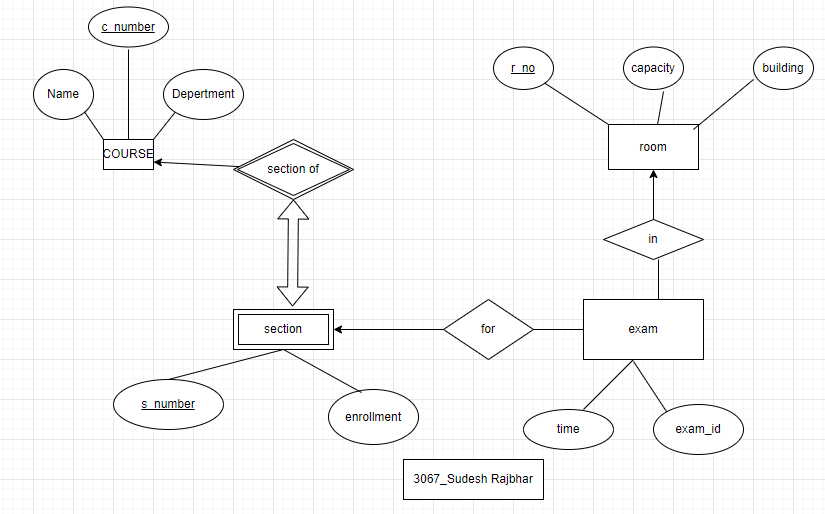
replace some of the attributes of the exam entity set, as

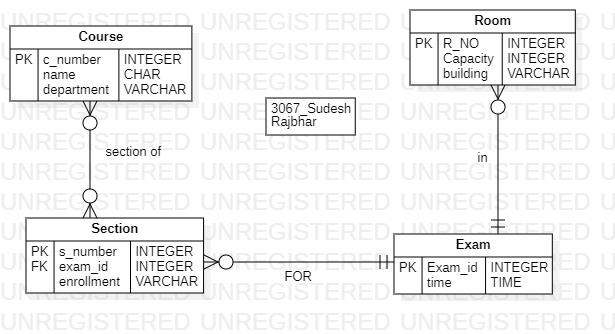
• course with attributes name, department, and c-number

• section with attributes s-number and enrolment, and dependent as a weak entity set on course

• room with attributes r-number, capacity, and building

Show an E-R diagram illustrating the use of all three additional entity sets listed.





#### 2)Study and implementation of Entity Relationship Diagrams.

A university registrar’s office maintains data about the following entities:

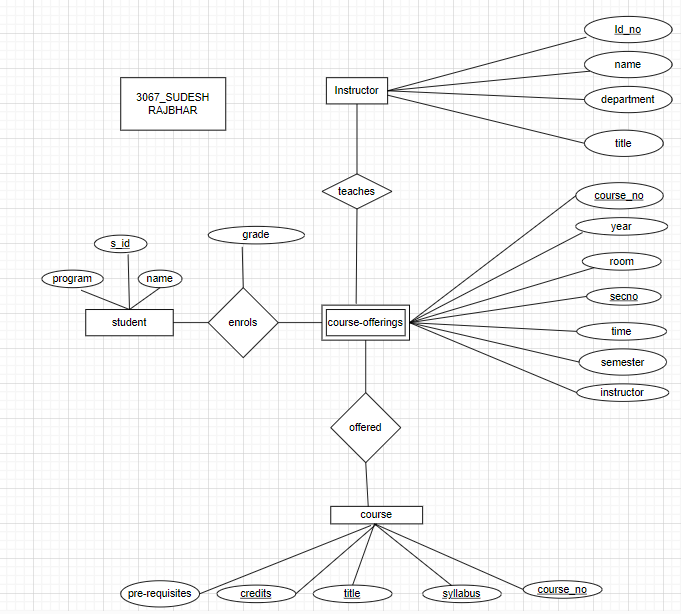
(a) courses, including number, title, credits, syllabus, and prerequisites;

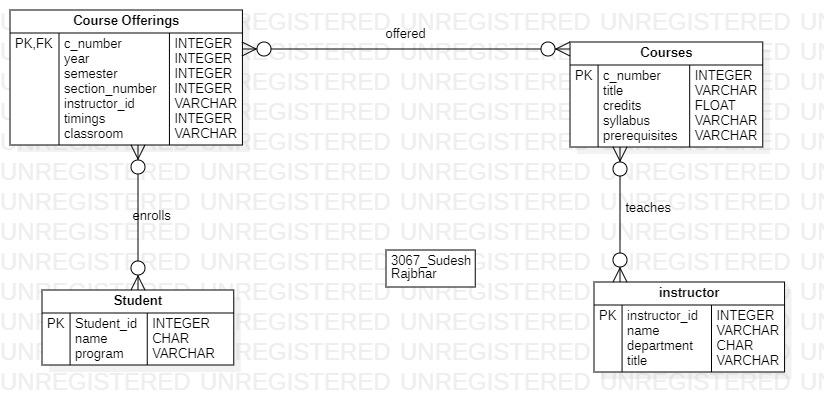
(b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom.

(c) students, including student-id, name, and program; and

(d) instructors, including identification number, name, department, and title.

Further, the enrolment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled.

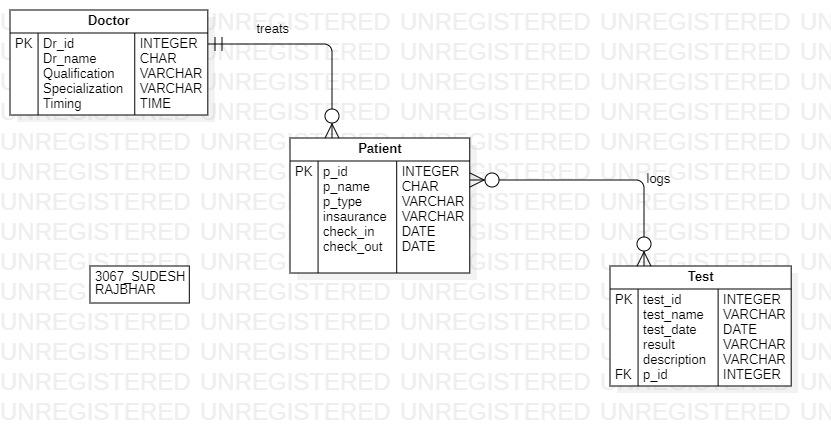




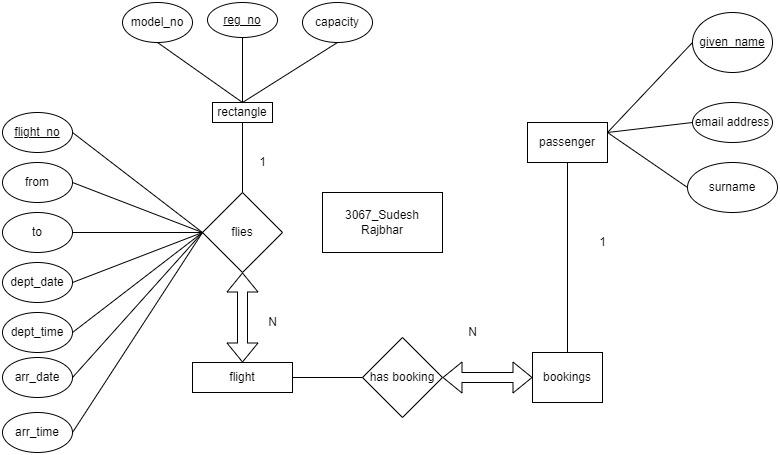
#### 

#### 3) Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors.Associate with each patient a log of the various tests and examinations conducted.





#### 4)The flight database stores details about an airline’s fleet, flights, and seat bookings. Again, it’s a hugely simplified version of what a real airline would use, but the principles are the same. Consider the following requirements list:The airline has one or more airplanes.An airplane has a model number, a unique registration number, and the capacity to take one or more passengers.An airplane flight has a unique flight number, a departure airport, a destination airport, a departure date and time, and an arrival date and time.Each flight is carried out by a single airplane.A passenger has given names, a surname, and a unique email address.A passenger can book a seat on a flight.



### 

### Practical 10

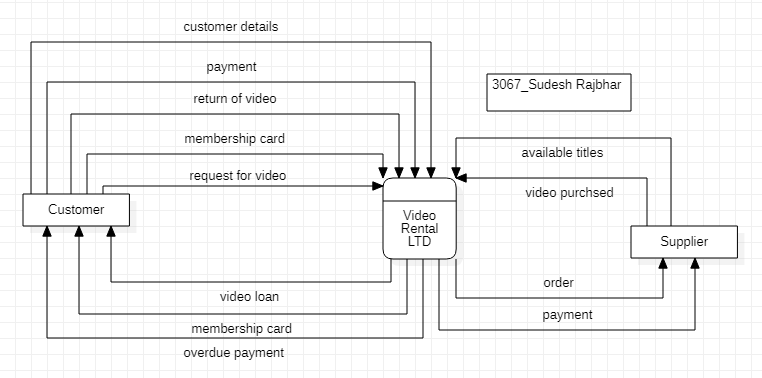
### 

#### 1) Study and implementation of Data Flow Diagrams.

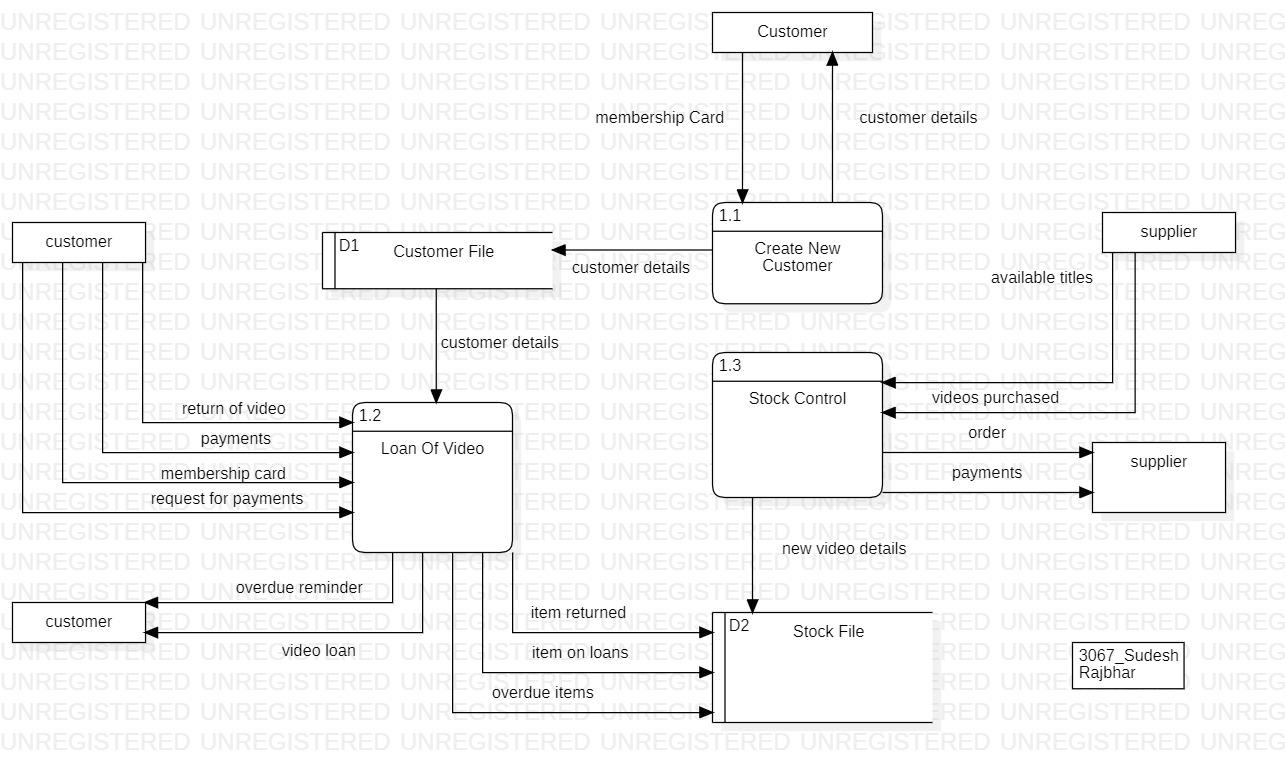
Video-Rental LTD is a small video rental store. The store lends videos to customers for a fee, and purchases its videos from a local supplier.A customer wishing to borrow a video provides the empty box of the video they desire, their membership card, and payment – payment is always with the credit card used to open the customer account. The customer then returns the video to the store after watching it. If a loaned video is overdue by a day the customer#39;s credit card is charged, and a reminder letter is sent to them. Each day after that a further card is made, and each week a reminder letter is sent. This continues until either the customer returns the video, or the charges are equal to the cost of replacing the video. New customers fill out a form with their personal details and credit card details, and the counterstaff give the new customer a membership card. Each new customer#39;s form is added to the customer file.

The local video supplier sends a list of available titles to Video-Rental LTD, who decide whether to send them an order and payment. If an order is sent then the supplier sends the requested videos to the store. For each new video a new stock form is completed and placed in the stock file.

LEVEL 0: CONTEXT DIAGRAM:



LEVEL 1:



LEVEL 2:

