

BHARATI VIDYAPEETH'S COLLEGE OF ENGINEERING

A-4, PASCHIM VIHAR, NEW DELHI - 110063

CERTIFICATE

This is to Certify that Mr./Miss..... ARYAN GUPTA

Roll No. 0 4 5 1 1 5 0 2 7 1 9 Branch:..... CSE-1

has satisfactorily completed his / her terms work in

Software Engineering Lab subject

Head of the Department

Date

Professor
Incharge

INDEX

NAME : ARYAN GUPTA

Roll.No.

0	4	5	1	1	5	0	2	7	1	9
---	---	---	---	---	---	---	---	---	---	---

EXPT. NO.	NAME OF THE EXPERIMENT	DATE OF PERFORMANCE	DATE OF SUBMISSION	REMARKS
1	Explain UML and all its types. Design an ATM class using StarUML	7/9/21	14/9/21	
2	Write Down the Problem Statement for library management system and design the solution	14/9/21	21/9/21	
3	Draw the Sequence, Collaboration, Activity, Component and Deployment diagram for library management system	21/9/21	28/9/21	
4	To perform the function orientation diagram: ER Diagram	28/9/21	5/10/21	
5	To prepare a feasibility report on University Management System	5/10/21	26/10/21	
6	To do the requirement analysis and develop Software Requirement Specification(SFS) sheet for suggested system	26/10/21	2/11/21	
7	To perform the function oriented diagram: Data Flow Diagram(DFD)	2/11/21	16/11/21	
8	To draw the Behavioral Diagram: State Chart Diagram	16/11/21	30/11/21	
9				
10				
11				
12				

Any Additional individual work done by students.

1	To study IOT Testing	30/11/21	7/12/21	
2	To implement Clone Testing	7/12/21	21/12/21	
3				

EXPERIMENT-1

ATM class using star UML

AIM-

Explain UML and all its type. Design an ATM class using star UML

SOFTWARE USED-

Star UML

THEORY-

HOW TO DOWNLOAD AND INSTALL STAR UML?

1. Star UML can be downloaded from <http://staruml.io/download>. Select the OS version suitable for your system and click on the download button. Mine is Windows and is doing it for Windows OS. The size of the file is 127 MB.
2. Run the .exe file.
3. After installation, you can see the Star UML icon on the desktop screen.
4. The software will open directly and it will ask for buy now option or evaluate the option. Click on evaluate.
5. The screen appears like this for the evaluated version.
6. If you check the C Drive, you can see the Star UML folder in the program file's location.

UML DIAGRAM-

UML stands for **Unified Modeling Language**. It's a rich language to model software solutions, application structures, system behavior and Business Processes.

UML Diagram Types-

There are two main categories:

1. Structure diagrams

- Class Diagram
- Component Diagram
- Deployment Diagram
- Object Diagram
- Package Diagram
- Profile Diagram
- Composite Structure Diagram

2. Behavioral diagrams

- Use Case Diagram
- Activity Diagram
- State Machine Diagram
- Sequence Diagram
- Communication Diagram
- Interaction Overview Diagram
- Timing Diagram

Class Diagram

Class diagram are the backbone of almost every object-oriented method, including UML. They describe the static structure of a system.

Package Diagram

Package diagrams are a subset of class diagrams, but developers sometimes treat them as a separate technique. Package diagrams organize elements of a system into related groups to minimize dependencies between packages.

Object Diagram

Object diagrams describe the static structure of a system at a particular time. They can be used to test class diagrams for accuracy.

Composite Structure Diagram

Use case diagrams model the functionality of a system using actors and use cases.

Activity Diagram

Activity diagrams illustrate the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system.

Sequence Diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time.

Interaction Overview Diagram

Interaction overview diagrams are a combination of activity and sequence diagrams. They model a sequence of actions and let you deconstruct more complex interactions into manageable occurrences.

Timing Diagram

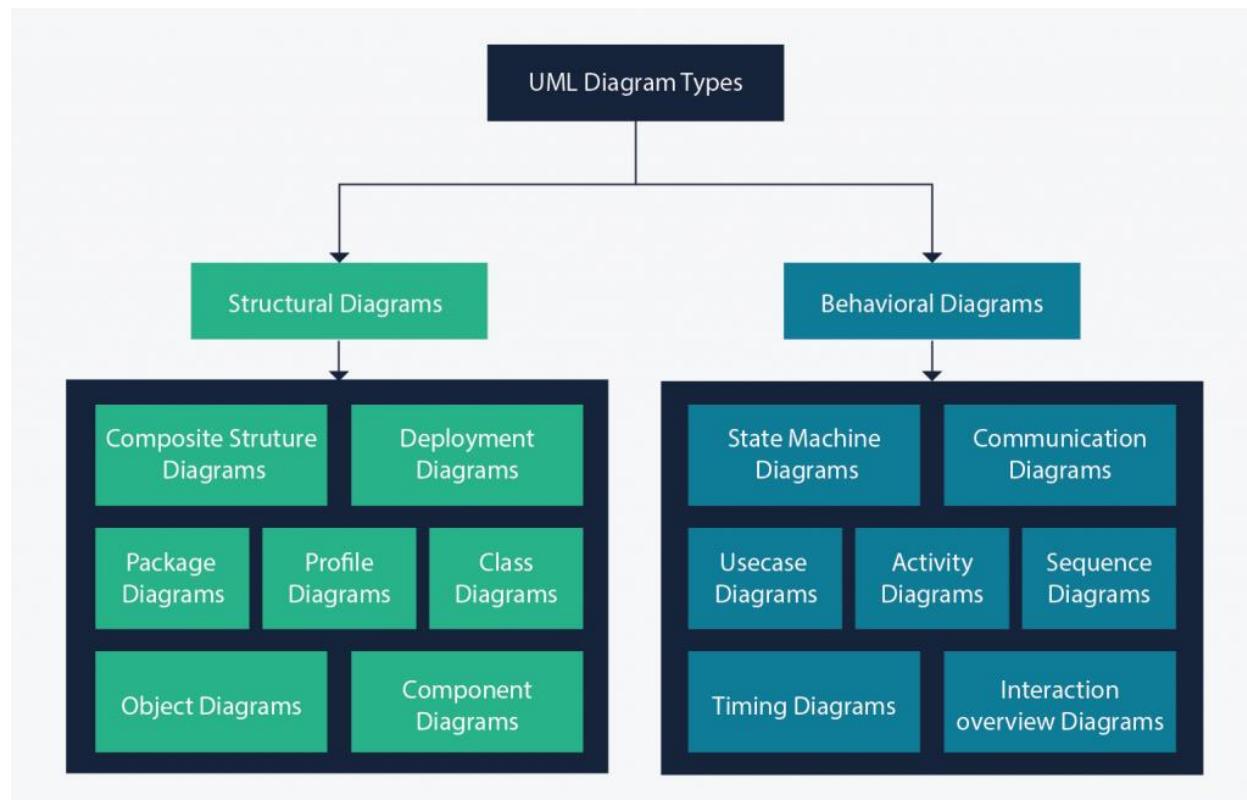
A timing diagram is a type of behavioral or interaction UML diagram that focuses on processes that take place during a specific period of time.

Communication Diagram

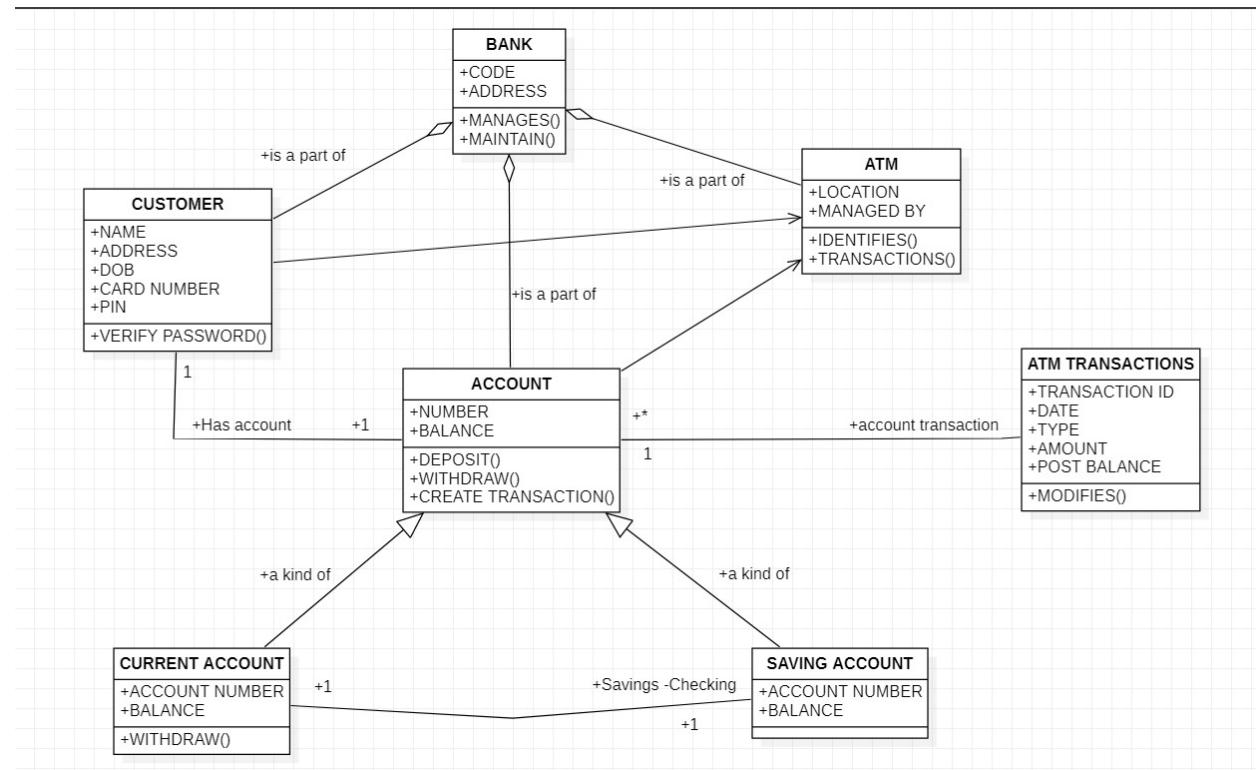
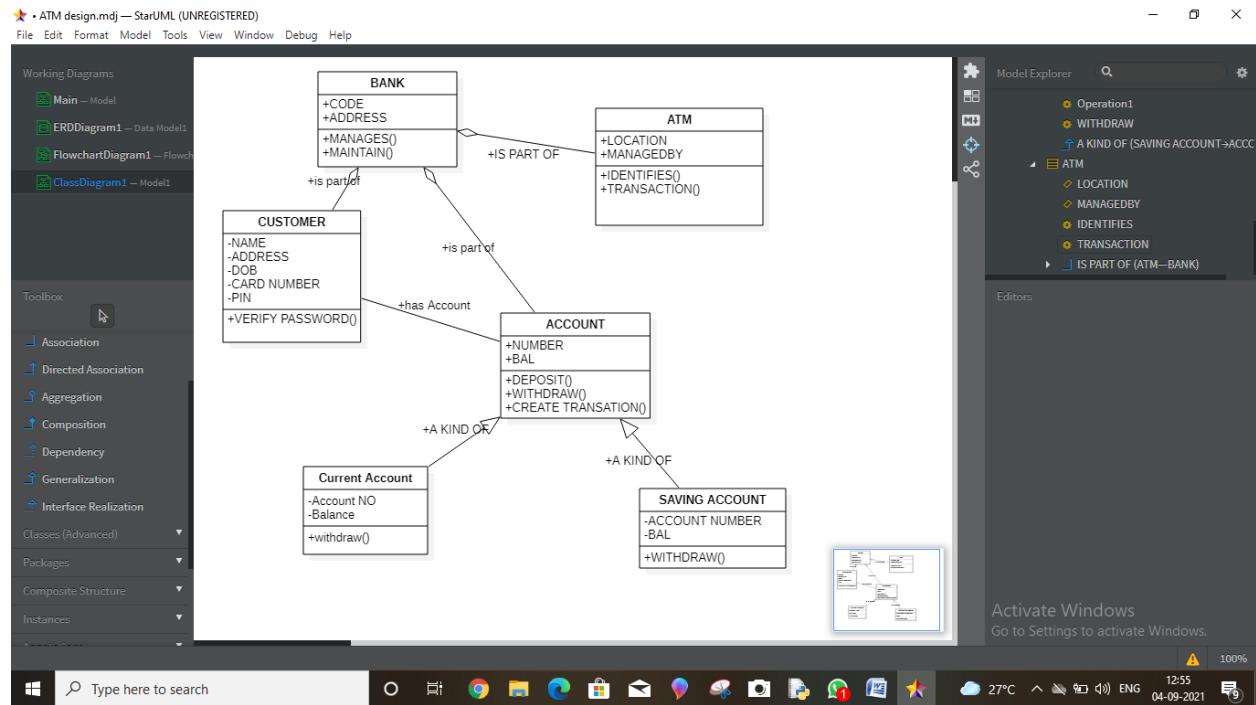
Communication diagrams model the interactions between objects in sequence. They describe both the static structure and the dynamic behavior of a system. In many ways, a communication diagram is a simplified version of a collaboration diagram introduced in UML 2.0.

State Diagram

State chart diagrams, now known as state machine diagrams and state diagrams describe the dynamic behavior of a system in response to external stimuli. State diagrams are especially useful in modeling reactive objects whose states are triggered by specific events



OUTPUT-



EXPERIMENT-2

AIM:

Write down the problem statement for library management system and design the system using UML diagram on Star UML.

Software Used: Star UML

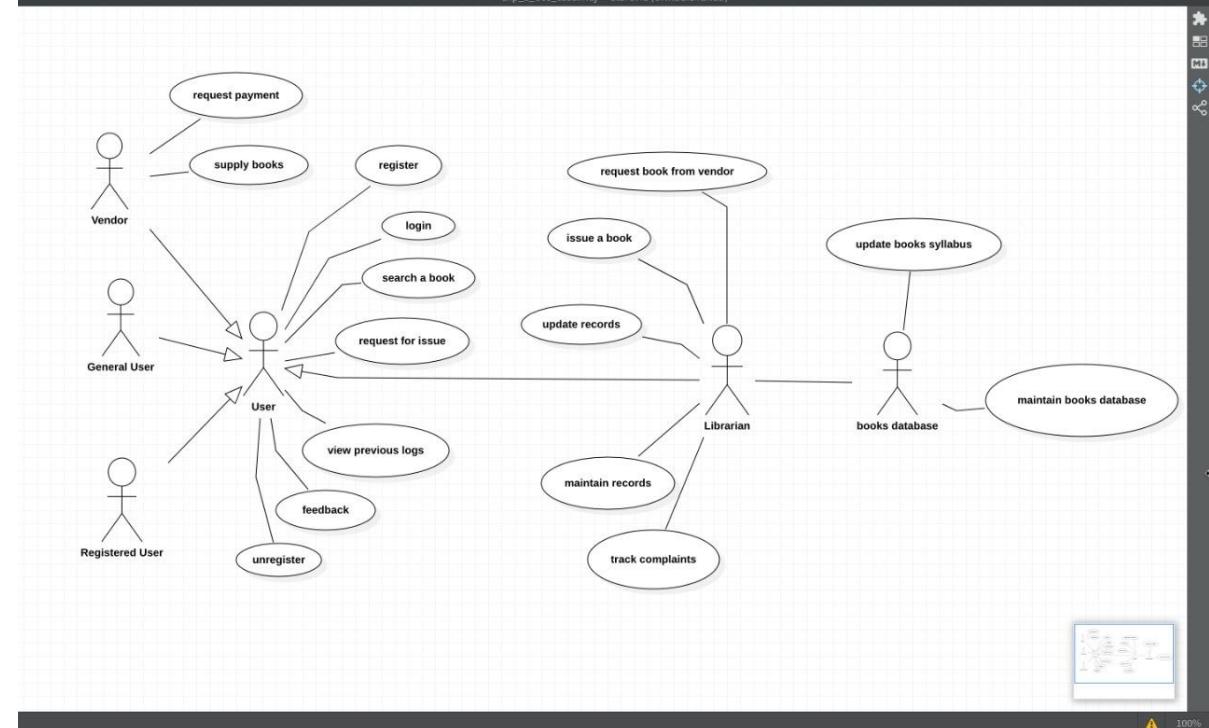
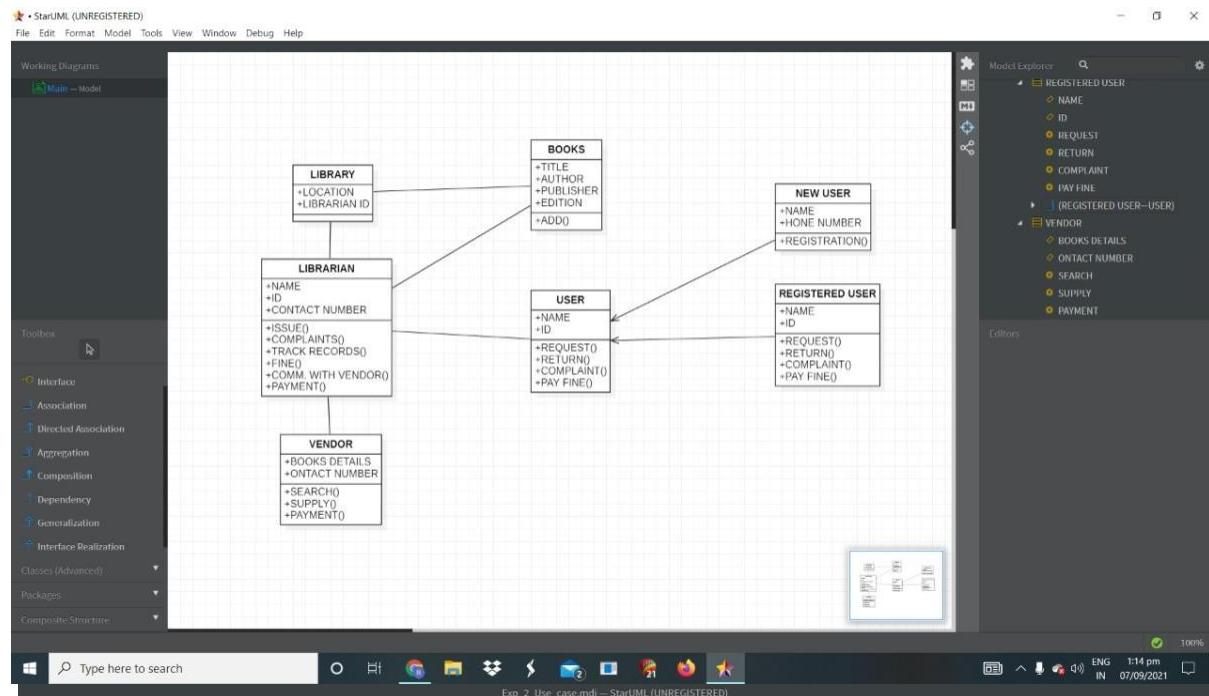
Theory:

Library Management System-

A library management system is ERP software that is designed & developed to manage all the in-house functions of a library. A librarian requires maintaining a database of new books and the books that are borrowed by users along with their due dates. This system completely automates all your library's activities.

Problem statement- A library database system is an infrastructure that allows users to search books and book content, add/remove, and download selected books. The problem faced is that library users require an efficient method to find a specific book or keyword(s) within a book given a continuously expanding library.

OUTPUT:



EXPERIMENT-3

AIM: Draw the sequence, collaboration, activity, component, deployment diagram for library management system.

Software Used: Star UML

Theory:

Sequence diagram: This diagram, as the name suggests, contains the sequence of flow of actions that are processed through a system and the life lines of the entities, when and how are they accessed. It also contains the security like which entity can process which entity and which one is visible, etc. There can be many number of sequence diagrams per each activity being done.

Collaboration diagram: This diagram is a polymorphic form of the sequence diagram in which the representation is different but application is the same. If we are able to create one sequence diagram, then its very simple to create its collaboration diagram with a single key click that varies from to software. There can be many number of collaboration diagrams per each activity being done because there can be many number of sequence diagrams.

Activity diagram: This diagram denotes the structural flow of the activities in the form of flow chart with decision boxes enhanced and hence is also used for troubleshooting like raising exceptions when a particular action is done and the alternative to be done when something abnormal is done. There can be only one activity diagram for the entire system including all the activities that a system can perform.

Component diagram: Component diagram represents the components in which the particular application needs to be installed or implemented on. It also shows the type of relation that exists among the various components that are represented. Hence, only a single component diagram representing all the components and their relations is needed for the entire system.

Deployment diagram: Deployment diagram is employed when we need to deploy the application we developed. A single deployment diagram is possible for a single system.

Problem Statement:

Library Management System is library management software for the purpose of monitoring and controlling the transactions in a library. Library management system gives us the complete information about the library and the daily transactions done in a Library. We need to maintain the record of new s and retrieve the details of books available in the library which mainly focuses on basic operations in a library like adding new member, new books, and up new information, searching books and members and facility to borrow and return books. It features a familiar and well thought-out, an attractive user interface, combined with strong searching, insertion and reporting capabilities.

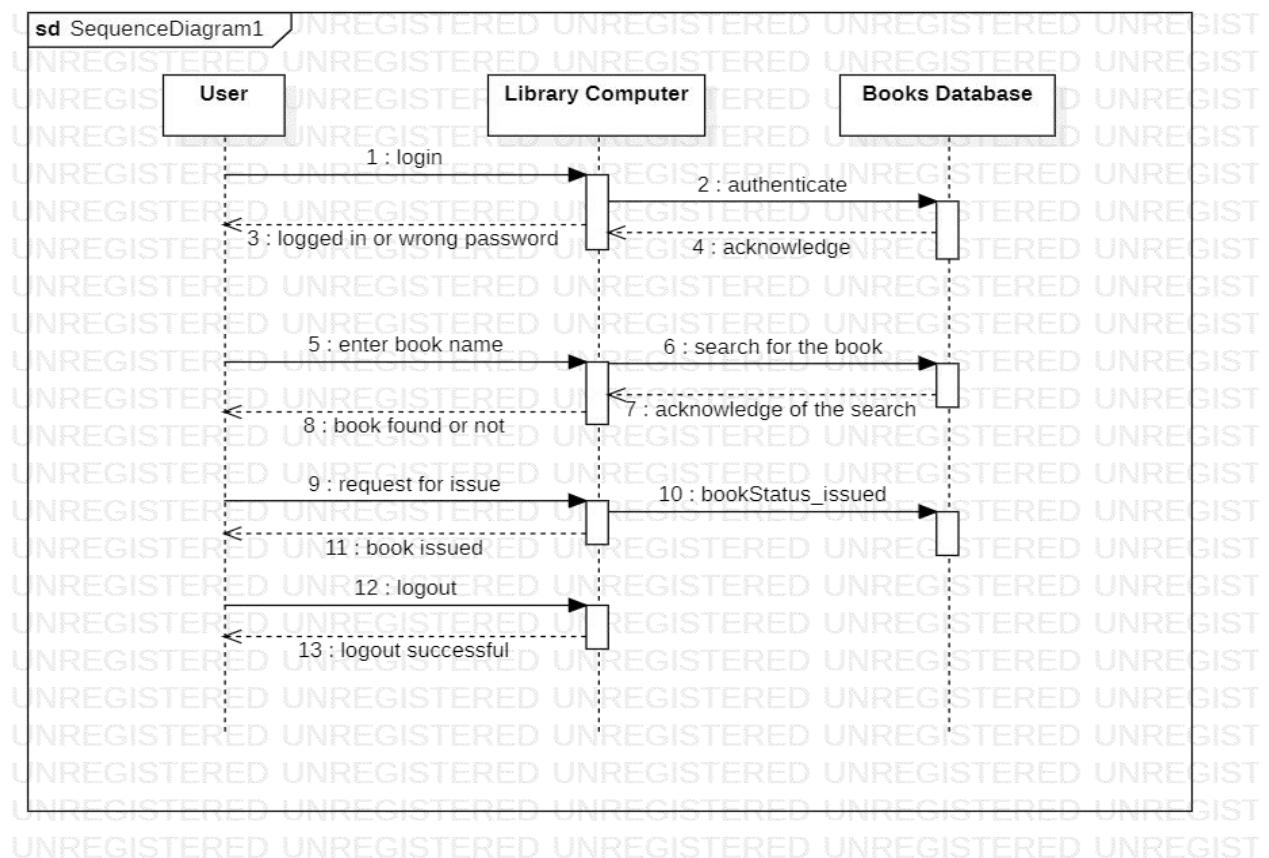
The following are the brief description on the functions achieved through this system:

End-Users:

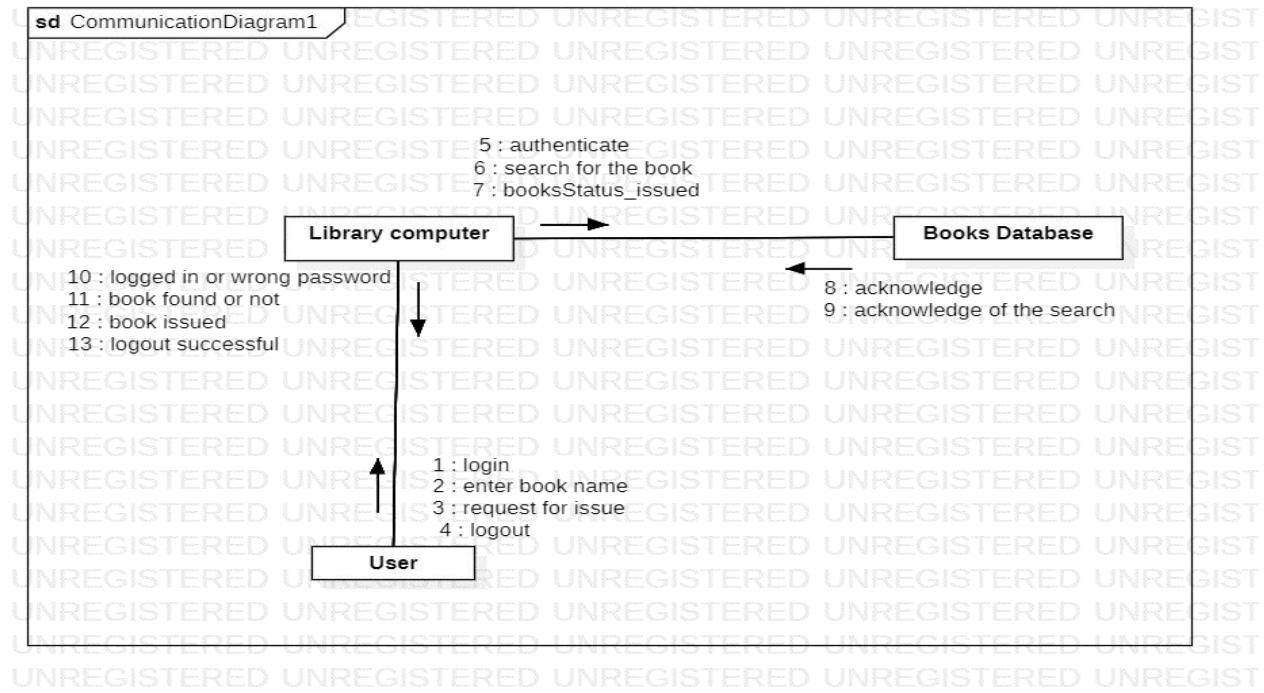
- Librarian: To maintain and update the records and also to cater the needs of the users.
- Reader: Need books to read and also places various requests to the librarian.
- Vendor: To provide and meet the requirement of the prescribed books.

SEQUENCE

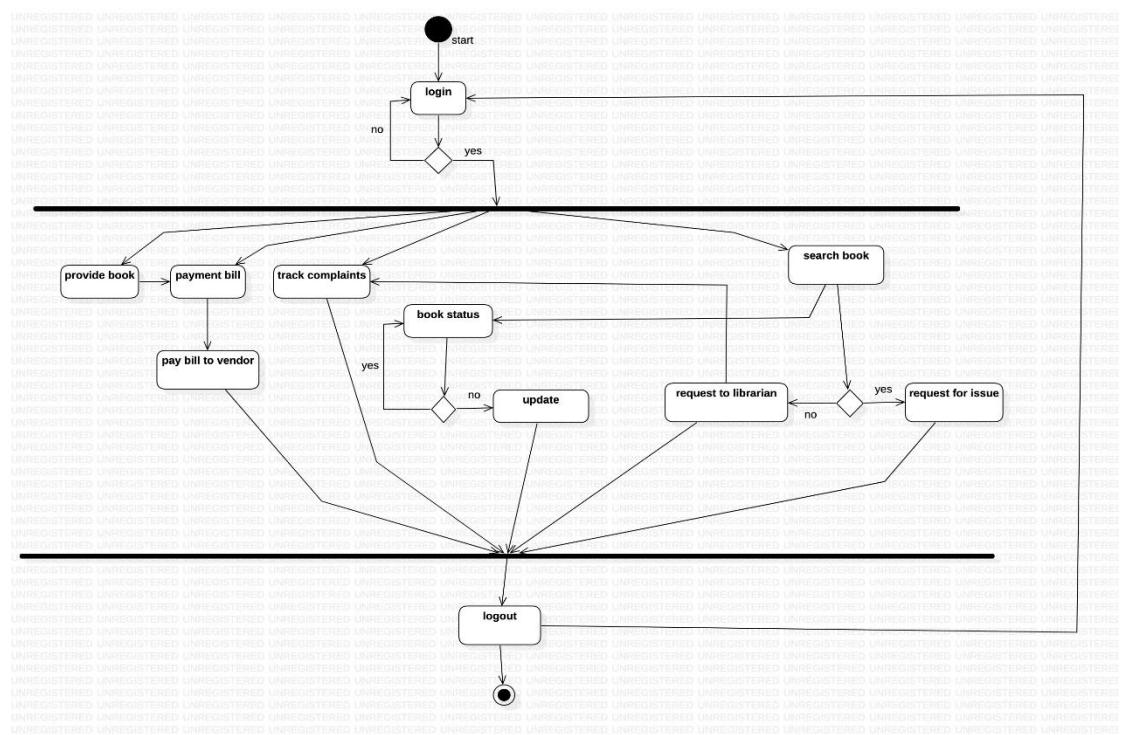
DIAGR



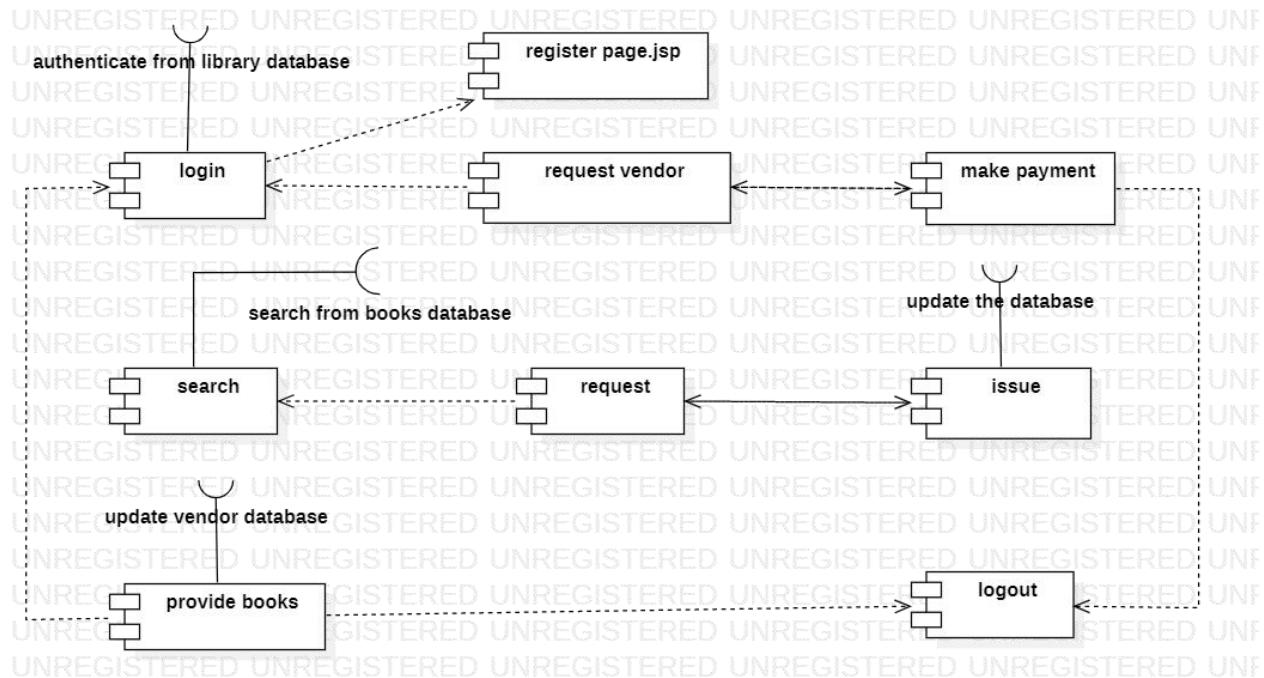
COLLABORATION DIAGRAM



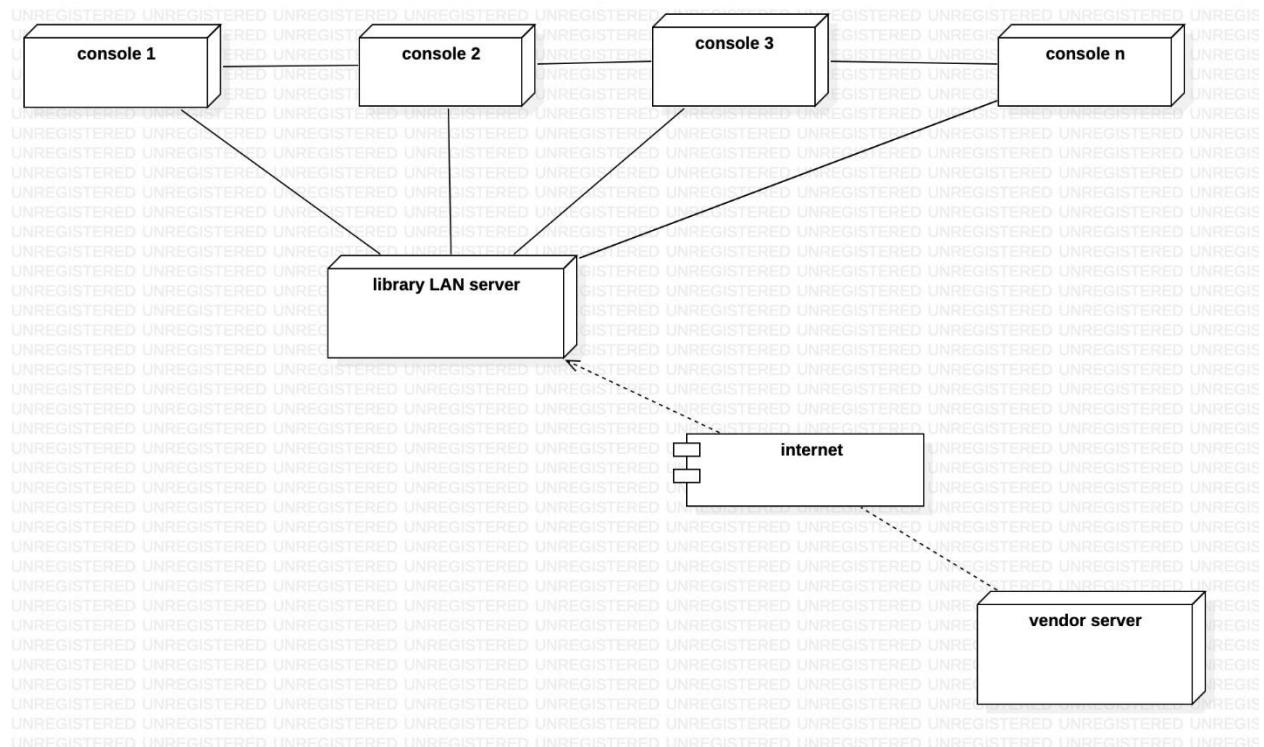
ACTIVITY DIAGRAM



COMPONENT DIAGRAM



DEPLOYMENT DIAGRAM



Experiment: 4

Aim: To perform the function orientation diagram: ER diagram

Software Used: Star UML

Theory:

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

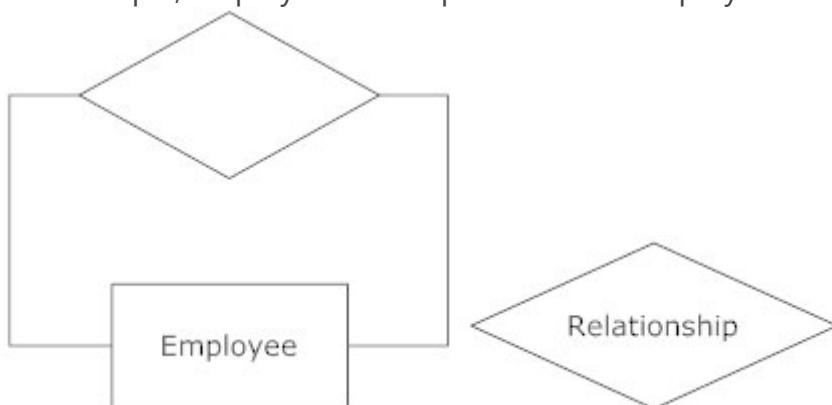
By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.
ER diagrams are used to sketch out the design of a database.

An ER diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

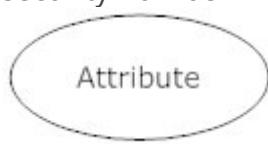
- **Entities**, which are represented by rectangles. An entity is an object or concept about which you want to store information. A weak entity is an entity that must be defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.



- **Actions**, which are represented by diamond shapes, show how two entities share information in the database. In some cases, entities can be self-linked. For example, employees can supervise other employees.



- **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.



A multivalued attribute can have more than one value.

For example, an employee entity can have multiple skill values.



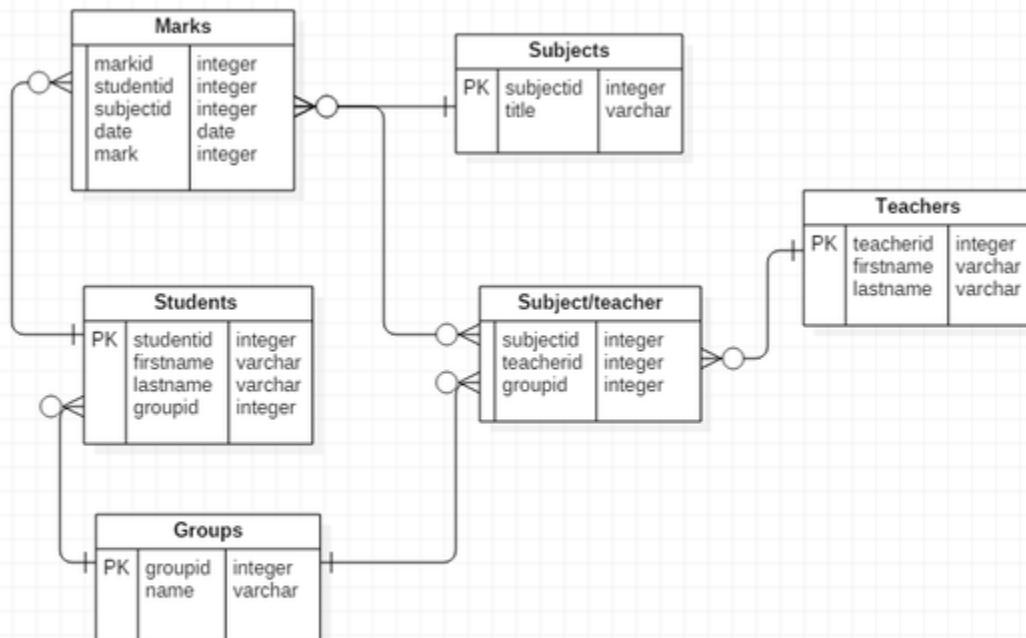
A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual



salary.

- **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the diagram.
- **Cardinality** specifies how many instances of an entity relate to one instance of another entity. Ordinality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinality describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinality specifies the absolute minimum number of relationships.

Entity Relationship Diagram:



Experiment – 5

Aim: Prepare a feasibility report on University management system.

Software Used: MS Word

Theory:

A feasibility study is an assessment of the practicality of a proposed project or system. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success.

A feasibility report is a testimony that attempts to create some sort of action. Feasibility reports are created to persuade/help the decision makers to choose between available options. Remember that your option is not the only one, the decision makers will probably have many to choose from. A feasibility report also determines whether or not the investigated task can be done with the amount of resources available OR how many resources will be necessary in order to complete the task. A feasibility may be useful in a lot of different situations such as event planning, finances, or even remodeling your home.

Following are the important features of feasibility report:

1. Very Articulate
2. Highly Systematic
3. Focused
4. Strategic in nature
5. Cost effective
6. Value adding
7. Organised
8. Flexible

Thus, to get a crystal-clear overview of a software the initial stage is to build its feasibility report. Therefore, let's consider the case of a “University management system” in which will be capable of managing the entire database.



Respect

Excellence

Integrity

Leadership



FEASIBILITY REPORT

UNVIERSTY MANAGEMENT SYSTEM

Paschim Vihar, West Delhi
NEW DELHI



National Project Management System
Business Projects-IT-Enabled
Feasibility Phase

Authority Signatures

Prepared by:	<i>Ayush Mittal</i>		
	Signature		
Please print:	Ayush Mittal Name	Project-Analyst Position	21/11/2021 Date
<hr/>			
Prepared by:	<i>Aryan Gupta</i>		
	Signature		
Please print:	Aryan Gupta Name	Co-founder Position	21/11/2021 Date
<hr/>			
Prepared by:	<i>Parth Garg</i>		
	Signature		
Please print:	Parth Garg Name	Manager Position	20/11/2021 Date

Table of Contents

Authority Signatures.....	3
Table of Contents.....	4
 1. Executive Summary.....	5
 2. Problem Statement.....	5
 3. Project Business Requirement.....	5
 4. Software Description.....	5
 5. Requirements.....	6
 6. Cost Estimation for implementation.....	7
 7. Future Scope.....	7

1. Executive Summary

The reason for a University Management System is to facilitate the use and measure of work at the college. It assists with keeping up a solitary record of the apparent multitude of subtleties of the subsidiary universities and their information base. This item keeps up a solitary information base for various divisions like assessment cell, confirmation cell and so forth to maintain a strategic distance from excess. It helps keeps up the consistency and a legitimate usefulness of the administration area.

- Management of such a large amount of data is impossible in handwritten manuscripts or some file formats.
- Storing such a huge data in the above mentioned way causes issues like data misplacing, time wastage in finding the data when required urgent, etc.
- For further analysis many more options of storing database, teams for implementing the whole software and low-cost procedure may be added to emphasize and ease more workload.

2. Problem Statement

Management of such a lot of information is unthinkable in manually written original copies or some document designs. Storing such a gigantic information in the previously mentioned manner causes issues like information losing, time wastage in finding the information when required earnest, and so on. For further investigation a lot more alternatives of putting away information base, groups for actualizing the entire programming and minimal effort strategy might be added to underline and straightforwardness more outstanding task at hand.

3. Project Business Requirement

The Product has a colossal degree in the advanced period since these days everybody needs to get digitalized. Everybody needs to accomplish the degree of precision and needs their framework to control all the orders for them. This Product is demonstrated useful for all the colleges to keep up their information base so as to eliminate excess and botches. This single programming will be sufficiently competent to deal with all the information base of the whole college and the other partnered schools information base to keep up records in various areas like the affirmation, assessment furthermore, different undertakings.

4. Software Description

UNIVERSITY MANAGEMENT SYSTEM deals with the maintenance of University data, records, instructions, and student information within the University. UMS is an automation system, which is used to store the information, students' record, and information of courses.

Starting from registration of a new student in the college, it maintains all the details regarding the students', instructors' and course details. The project collects related information from all the departments of an organization and maintains database, which are used to generate reports in various forms to measure Individual and overall performance of the students.

Features of a university management system:

- University Administration
- University Employee Management
- College Management
- College Administration
- College Employee Management
- Department Management
- Student Management
- Courses Management
- Society Management
- Head of Department Management
- Instructor Management

5. Requirements

- **Performance:** Developer group dealt with the presentation by composing upgraded code and taking care of the information base of the students. They attempt to give flexibility, accessibility, compactness and ease of use to the product.
- **Security:** The product is made keeping the security in mind. The other individual subtleties are additionally secured by programmers. The students' information will be sheltered from the gatecrashers and programmers. On the off chance that disappointment of programming occurs, all the information will be spared and recovered upon the restart of the application.
- **Storage:** The storage of a large data of the students will require Cloud database. Here in it'll be easy to add, delete, update directly into server from anywhere by the owner. Changes will only be made with proper proofs.
- **Accessibility:** This will be provided to faculties as well as the students as viewers only. Editing access will be with the Admin only with owner tag. This will make sure not fair play between Admin and others.

6. Cost Estimation for implementation

The product sent accompanies an assortment of groups that dealt with the task for the final output software. Beginning from planning to building up the product is a joined exertion of the apparent multitude of groups.

Table -Cost Estimation Table

S. No.	Team	Work Description	Expenditure (₹)
1	Creative Team	Creating eye catching thoughts for crowds/clients.	95,000
2	Marketing Team	Arranging efforts to advance items and administrations to clients	1,20,000
3	Designing Team	Drawing and planning of the required design.	1,00,000
4	Developing Team	Planning the product item in view of end-clients.	1,30,000
5	Testing Team	Troubleshooting the product to guarantee the no disappointment	90,000
TOTAL COST			Rs. 5,35,000

7. Future Scope

With all the features we have introduced, many more features like in colleges where attendance management is strict that can be added, whoever gives e-payment their data can be stored separately. Also, more teams can be introduced and price reduction can be made possible. We can improve accessibility by introducing Android or iOS applications of this software.

EXPERIMENT - 6

Software Requirements Specification

for

Hotel Management System

Prepared by:

**04511502719 ARYAN GUPTA
00811502719 PARTH GARG
01811502719 BHAVUK KHANNA
20211502719 AYUSH MITTAL**

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction.....	1
1.1 Purpose	1
1.2 Document Conventions	2
1.3 Project Scope	2
1.4 References	3
2. Overall Description.....	3
2.1 Product Perspective	3
2.2 User Classes and Characteristics	5
2.3 Operating Environment	6
2.4 Design and Implementation Constraints	6
2.5 Assumptions and Dependencies	6
3. System Features	6
3.1 System Feature	6
4. Data Requirements	7
5. External Interface Requirements	8
5.1 User Interfaces	8
5.2 Software Interfaces	11
5.3 Hardware Interfaces.....	11
5.4 Communications Interfaces	12
6. Quality Attributes	16
7. Other Requirements	17
Appendix A: Glossary.....	17

1. Introduction

The regular hotel management system project entirely in an android app. This android application allows the hotel manager to handle all hotel activities in his android phone. Interactive guy and the ability to manage various hotel bookings and rooms from an android phone makes this hotel management system very flexible and convenient. The hotel is a very busy person and does not have the time to sit and manage the entire thing sitting at a single computer. This application gives him the power and flexibility to manage the entire system from a single android phone. Hotel management android project provides room booking, staff management and other necessary hotel management features top make it a complete portable hotel management solution.

1.1 Purpose

The Software Requirements Specification (SRS) will provide a detailed description of the requirements for the Hotel Management System (HMS). This SRS will allow for a complete understanding of what is to be expected from the newly introduced system which is to be constructed. The clear understanding of the system and its' functionality will allow for the correct software to be developed for the end user and will be used for the development of the future stages of the project. This SRS will provide the foundation for the project. From this SRS, the Hotel Management System can be designed, constructed, and finally tested.

This SRS will be used by the system development team which is constructing the HMS and the hotel end users. The Project team will use the SRS to fully understand the expectations of this HMS to construct the appropriate software. The hotel end users will be able to use this SRS as a “test” to see if the constructing team will be constructing the system to their expectations. If it is not to their expectations the end users can specify how it is not to their liking and the team will change the SRS to fit the end users' needs.

1.2 Document Conventions

The document is prepared using Microsoft Word 2016 and has used the font type 'Times New Roman'. The fixed font size that has been used to type this document is 12pt with 1.5 line spacing. It has used the bold property to set the headings of the document. Use case scenario is written according to Alistair Cockburn's template. UML diagrams have been created according to UML 2.0 standards. Standard IEEE template is the template used to organize the appearance of the document and its flow.

1.3 Project Scope

The software product to be produced is a Hotel Management System which will automate the major hotel operations. The first subsystem is a Reservation and Booking System to keep track of reservations and room availability. The second subsystem is the Tracking and Selling Food System that charges the current room. The third subsystem is a General Management Services and Automated Tasks System which generates reports to audit all hotel operations and allows modification of subsystem information. These three subsystems' functionality will be described in detail in section 2-Overall Description.

There are two end users for the HMS. The end users are the hotel staff (customer service representative) and hotel managers. Both user types can access the Reservation and Booking System and the Food Tracking and Selling System. The General Management System will be restricted to management users.

The Hotel Management System's objectives is to provide a system to manage a hotel that has increased in size to a total of 100 rooms. Without automation the management of the hotel has become an unwieldy task. The end users' day-to-day jobs of managing a hotel will be simplified by a considerable amount through the automated system. The system will be able to handle many services to take care of all customers in a quick manner. The system should be user appropriate, easy to use, provide easy recovery of errors and have an overall end user high subjective satisfaction.

1.4 References

Books References:

1. IanSommerville, Software Engineering 8th edition.
2. ElmasriNavthe, Foundation of database system 3rd edition.
3. RaguRamakrishnan/johnesGehrke, Database management system 3rd edition.
4. Gerald W. Latin, Modern hotel management, W.H. Freeman 2011.
5. Michael J. O'Fallon, Denney G. Rutherford, Hotel Management and Operations illustrated edition John Wiley & Sons, 2001

Internet References:

1. [www.softwareadvice.com /hotel management](http://www.softwareadvice.com/hotel-management)
2. [http://ww.high-level-software.com /features](http://www.high-level-software.com/features)
3. " Hotel Management Case Study", March.6, 2010. [Online]. Available:
<http://www.scribd.com/doc/27927992/Hotel-Management-Case-Study>, [Accessed: June.28, 2014]
4. " High-Level-Software Features", [Online]. Available:
<http://www.high-level-software.com/features/>, [Accessed: June.25, 2014]
5. Lauesen, S, (2003), Task Descriptions as Functional Requirements, IEEE Computer Society, Retrieved from <http://www.itu.dk/~slauesen/Papers/IEEEtasks.pdf>

2. Overall Description

2.1 Product Perspective

The Hotel Management System is a new self-contained software product which will be produced by the project team in order to overcome the problems that have occurred due to the current manual system. The newly introduced system will provide an easy access to the system and it will contain user friendly functions with attractive interfaces. The system will give better options for the

problem of handling large scale of physical file system, for the errors occurring in calculations and all the other required tasks that has been specified by the client. The outcome of this project will increase the efficiency of almost all the tasks done at the Hotel in a much convenient manner.

2.2 User Classes and Characteristics

2.2.1 User Classes

There are three user levels in Hotel Management System.

1. Owner
2. Manager
3. Receptionist

2.2.2 Characteristics of User Classes

Owner: -

Hotel owner has the privilege of Monitoring and authorization of all the tasks handle by the system. He can access every function performed by the system. Owner of the company as well as the system can access to the administration panel which is consider the core of the system. As the main authorized person of the company owner gets the ability to manage the other users including their user levels and privileges. Taking backups of the system and restoring system can also be done by the Owner. Meanwhile he will be able to take all the kinds of reports available in the system. As the owner of the system and the company he has the power to set room rates as well. Hotel owner has the sole right of deleting a staff member from the system database.

Manager:

Manager is responsible for managing resources available in hotel management system. Manager also has most of the privileges mentioned above except the things regarding the payment handling. The reason for using a Manager is to reduce the workload done by the owner that cannot be assigned to the receptionist, as those tasks seem much responsible. The user level, Manager has the authority to take all the reports available in the system but here also except the reports related to

Software Requirements Specification for Hotel Management System

financial stuff, hotel income. Manager has other abilities that receptionist, user level has. Such as, adding new staff member to the system, modifying them or removing them, adding new guests to the system, modifying them and removing them from the system, adding new inventory to the system, modifying them and removing them. Adding new room types to the system, modifying them and removing them.

Receptionist:

As a hotel receptionist, he or her role will be to attain the goals of bookings and to ensure that all guests are treated with a high standard of customer service. Hierarchically receptionist role has the least accessibility to the system functions. Receptionist plays the boundary role of the system. He or she can perform limited functions such as registering new guest to the system, make reservations, Sending e-mail reminders to clients for booking confirmation. Management of hotel will prefer to hire receptionist who have a good standard of general education and possibly in subjects such as English, math and IT.

2.3 Operating Environment

Hardware and software requirements

Hardware: -

1. Operating System Supports all known operating systems, such as Windows, Linux, Mac
2. Computer 512MB+ RAM, monitor with minimum resolution of 1024x768, keyboard, and mouse
3. Hard Drive should be in NTFS filesystem formatted with minimum 10 GB of free space
1. A Laser printer will need to be used to print these reports and notes

Software: -

1. Software is designed to run on any platform above Microsoft Windows 7 (32bit).
2. Microsoft .NET Frameworks 4.0 or above.
3. Microsoft SQL Server Management Studio Express 2010.

2.4 Design and Implementation Constraints

Software development crew provides their best effort in developing the system. In order to maintain the reliability and durability of system, some design and implementation constraints are applied. Availability of an android app for hotel management system could make the system portable but due to time constraint it is not possible. System will need a minimum memory of 512MB. But it is recommended to have a memory of 1GB. When designing interfaces of system, we had the capability of work with new tools such as Dev Express. Considering the client's budget, we decided to create those interfaces in a simple realistic manner using affordable technology.

2.5 Assumptions and Dependencies

Some software used in implementing the system is with high cost and the client has agreed to afford the amount of money needed to purchase them. It's assumed that client won't change that decision on the next phases of the software development. Although we assume that client is using windows 7 or windows 8. Otherwise if client use an open source operating system, there is a need of changing the SRS accordingly.

3. System Features

1. Sometimes it happens that the rooms get booked soon when one visits the place therefore user can make advance booking using this system.
2. It saves user time in search of rooms.
3. The system is useful as it calculates an exact cost for requested number of days.
4. It saves organization resources and expenses.
5. This system is effective and saves time and cost of users.
6. The system is portable i.e. can be used from anywhere.
7. Easy registration.

3.1.1 Functional Requirements

1. Make Reservation
2. Search Room

3. Add Payment
4. Issue Bills
5. Manage Guest (Add, Update Guest)
6. Manage Room Detail (Add, Update, Delete)
7. Manage Inventory (Add, Edit, Delete)
8. Manage Staff (Add, Update, Delete, View)
9. Set Rates
10. Retrieves reports (Staff payment, Income)
11. Manage Users (Add, Update, Delete)
12. Taking Backups
13. Email notification

4. Data Requirements

The logical database requirements include the retention of the following data elements. This list is not a complete list and is designed as a starting point for development.

1. Booking/Reservation System
2. Customer first name
3. Customer last name
4. Customer address
5. Customer phone number
6. Number of occupants
7. Assigned room
8. Default room rate
9. Rate description
10. Guaranteed room (yes/no)
11. Credit card number
12. Confirmation number
13. Automatic cancellation date

14. Expected check-in date
15. Expected check-in time
16. Actual check-in date
17. Actual check-in time
18. Expected check-out date
19. Expected check-out time
20. Actual check-out date
21. Actual check-out time
22. Customer feedback
23. Payment received (yes/no)
24. Payment type
25. Total Bill

5. External Interface Requirements

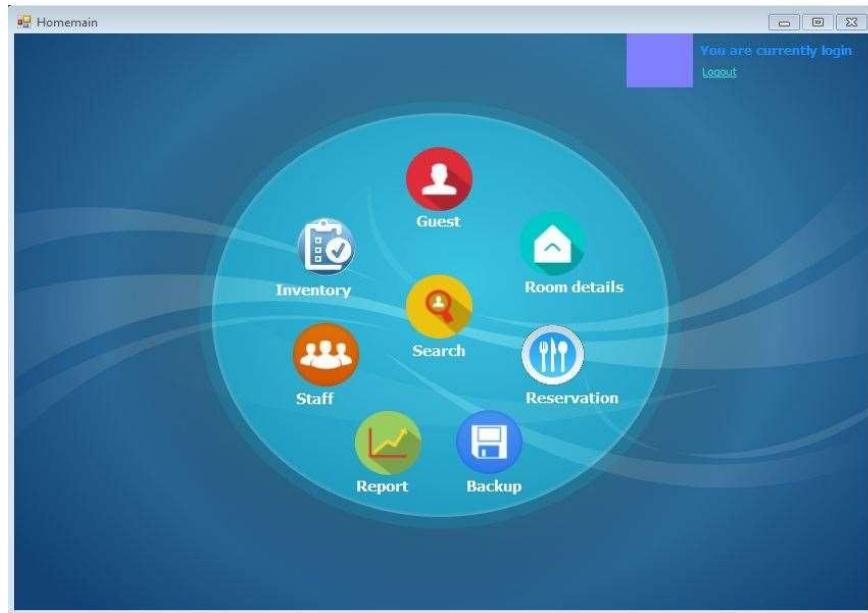
5.1 User Interfaces

Screen Name	Description
Login	Log into the system as a CSR or Manager
Reservation	Retrieve button, update/save reservation, cancel reservation, modify reservation, change reservation, adjust room rate, accept payment type/credit card
Check-in	Modify room stay (e.g., new credit card), check-in customer (with or without a reservation), adjust room rate, special requests, accept payment type/credit card
Checkout	Checkout customer, generate bill
Hotel Payment	Accept payment for room and food
Room Service/Restaurant	Create order, modify order, view order, cancel order, generate meal bill

Software Requirements Specification for Hotel Management System

Screen Name	Description
Customer Record	Add or update customer records
Administer Rooms	Availability and rates
Administer User	Create, modify, and delete users; change password
Administer Meals	Create, modify, and delete meal items and prices
Reports	Select, view, save, and delete reports

User friendly dashboard of system



Login interface is used to login to the system using username and password for three different users.



Software Requirements Specification for Hotel Management System

Adding new guest to the system

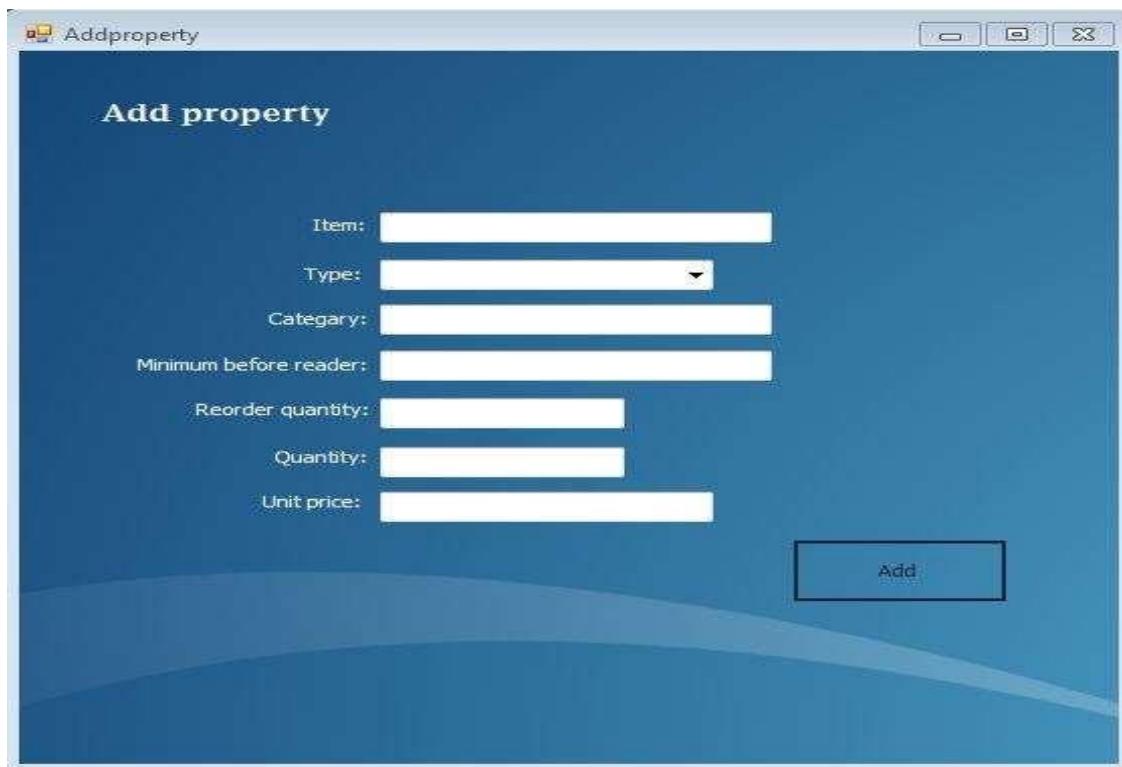
The screenshot shows a Windows application window titled "Home". The interface is divided into several sections:

- Guest Information:** Contains fields for Name (dropdown), Address (text input), and Country (dropdown).
- Other Information:** Contains fields for ID type (dropdown), ID number (text input), Gender (radio buttons for Male and Female), VIP status (dropdown), No.of Adults (text input), and No.of Children (text input).
- Contact Information:** Contains fields for Phone (text input), Mobile (text input), and Email (text input).
- Stay Information:** Contains fields for Room(s) (dropdown), Special Request (text input), Arrival/Check In (date picker set to Saturday, July 26, 2014), Departure/Checked out (date picker set to Saturday, July 26, 2014), and Rate Type (dropdown).

At the bottom right are two buttons: "Temp Reserve" and "Reserve".

Software Requirements Specification for Hotel Management System

Adding new property to the system



5.2 Software Interfaces

The computer this software going to be install need to have Windows Operating System equal or above, Windows 7. On that Windows platform .Net 4.0 will be installed and that will be the platform the software will be run. There will be an ADO.NET data transmission with the Microsoft SQL Server Management Studio Express 2010 R2 edition that will be installed in the same computer.

5.3 Hardware Interfaces

A specific computer must match with the above-mentioned requirements in order to gain the maximum benefits from the system in an efficient manner.

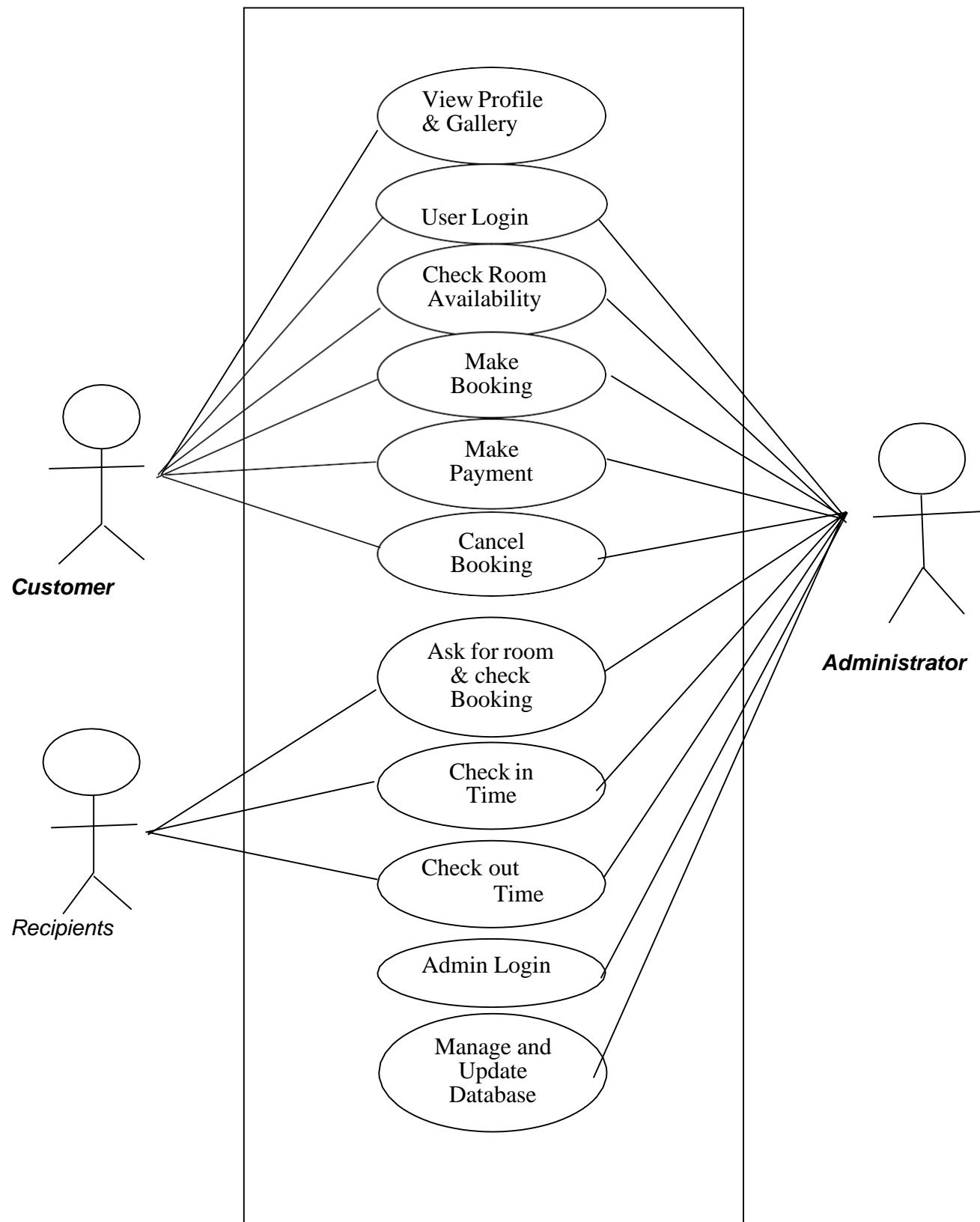
Reservation alerts will be sent to the one of the members of hotel staff as an e-mail notification. So, there is a need of broadband internet connection. Client should able to keep a stable internet connection. A laser printer will be needed when printing bills and several reports

5.4 Communications Interfaces

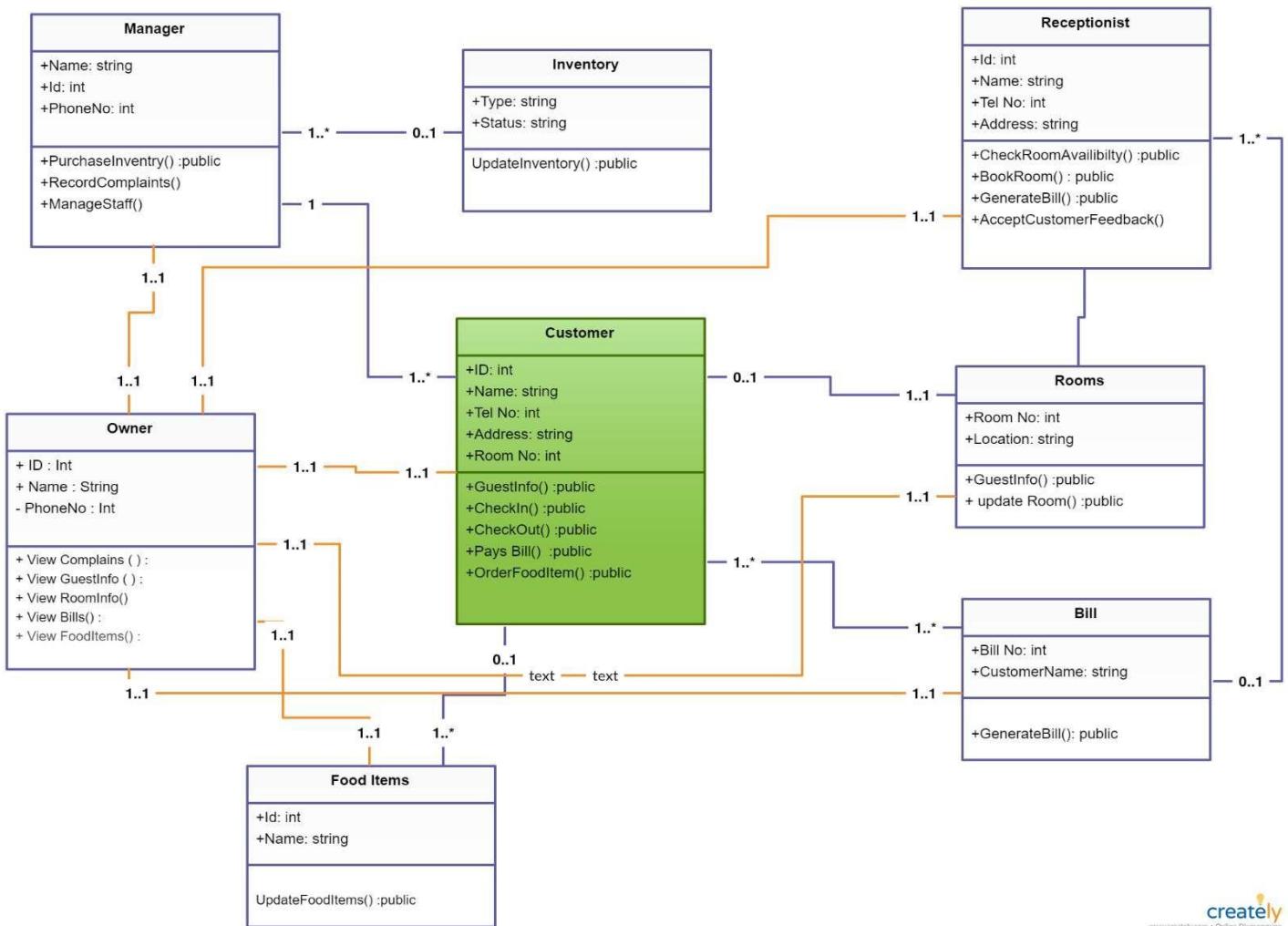
When a specific reservation reserved at the same time an e-mail notification will be sent to both relevant staff member's e-mail account and guest's account. Guest will be notified in the check-out date. To achieve that functionality, it requires having a stable internet connection. Mostly a broadband connection with the client's computer will provide the efficient service.

5.4.1 USE CASE DIAGRAM

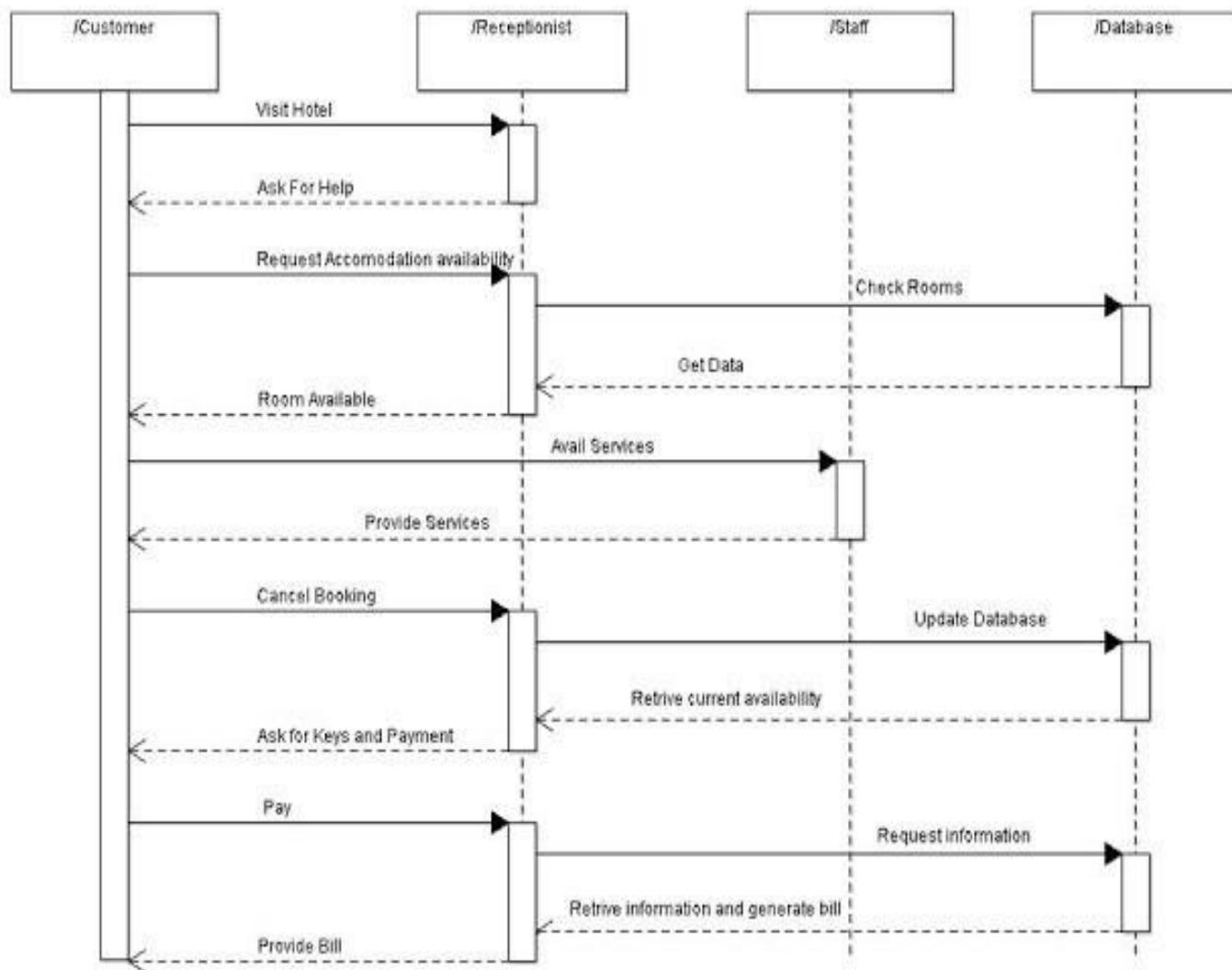
Software Requirements Specification for Hotel Management System



5.4.2 CLASS DIAGRAM



5.4.3 SEQUENCE DIAGRAM



Sequence diagram for Hotel Management System

6. Quality Attributes

1. **Availability:** - The system shall be available during normal hotel operating hours
2. **Correctness:** - extent to which program satisfies specifications, fulfills user's mission objectives
3. **Efficiency:** - How much less number of resources and time are required to achieve a particular task through the system.
4. **Flexibility:** - Ability to add new features to the system and handle them conveniently.
5. **Integrity:** - How the system would insecure the information in the system and how it avoids the data losses. Referential integrity in database tables and interfaces
6. **Maintainability:** - How easy is to keep the system as it is and correct defects with making changes.
7. **Portability:** - The Hotel Management System shall run in any Microsoft Windows environment
8. **Reliability:** - Specify the factors required to establish the required reliability of the software system at time of delivery. Mean time between failures and mean time to recovery
9. **Reusability:** - What is the ability to use the available components of the system in other systems as well.
10. **Testability:** - Effort needed to test to ensure performs as intended
11. Usability: - How easily a person can be taken the benefits of the system and the user friendliness.
12. **Robustness:** – Strength of the system to handle system functions accurately and maintain the database without facing to unexpected failures
13. **Maintainability:** – What design, coding standards must be adhered to exclusions created

7. Other Requirements

When the system is completely developed and submitted to the client, few sessions will be required to make the users of the system understand about the functionality of it and some time to adapt to the system. After those sessions, it's required that a member from the development team should spend sometime in the system background for an agreed time period. That time period will be used in identifying new bugs that could not be reached in the earlier phases of the development process. Client should have a valid e-mail account in order to receive reservation e-mail notifications.

Appendix A: Glossary

Check-out – settle one's hotel bill before leaving

Check-in – the process whereby a guest announces their arrival at the hotel

Experiment-07

AIM

To perform the function oriented diagram: Data flow Diagram (DFD).

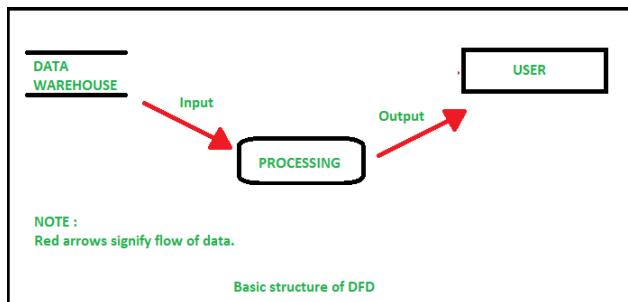
SOFTWARE USED

THEORY

Function Oriented Design is an approach to software design where the design is decomposed into a set of interacting units where each unit has a clearly defined function.

Data Flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.



Rules for creating DFD

- The name of the entity should be easy and understandable without any extra assistance (like comments).
- The processes should be numbered or put in ordered list to be referred easily.
- The DFD should maintain consistency across all the DFD levels.
- A single DFD can have maximum processes upto 9 and minimum 3 processes.

Symbol	Name	Function
	Data flow	Used to Connect Processes to each other , to sources or Sinks; te arrow head indicates direction of data flow.
	Process	Performs Some transformation of Input data to yield output data.
	Source of Sink (External Entity)	A Source of System inputs or Sink of System outputs.
	Data Store	A repository of data; the arrow heads indicate net inputs and net outputs to store.

Symbols for Data Flow Diagrams

DFD uses hierarchy to maintain transparency thus multilevel DFD's can be created. Levels of DFD are as follows:

- 0-level DFD
- 1-level DFD
- 2-level DFD

0-level DFDM

It is also known as fundamental system model, or context diagram represents the entire software requirement as a single bubble with input and output data denoted by incoming and outgoing arrows. Then the system is decomposed and described as a DFD with multiple bubbles. Parts of the system represented by each of these bubbles are then decomposed and documented as more and more detailed DFDs.

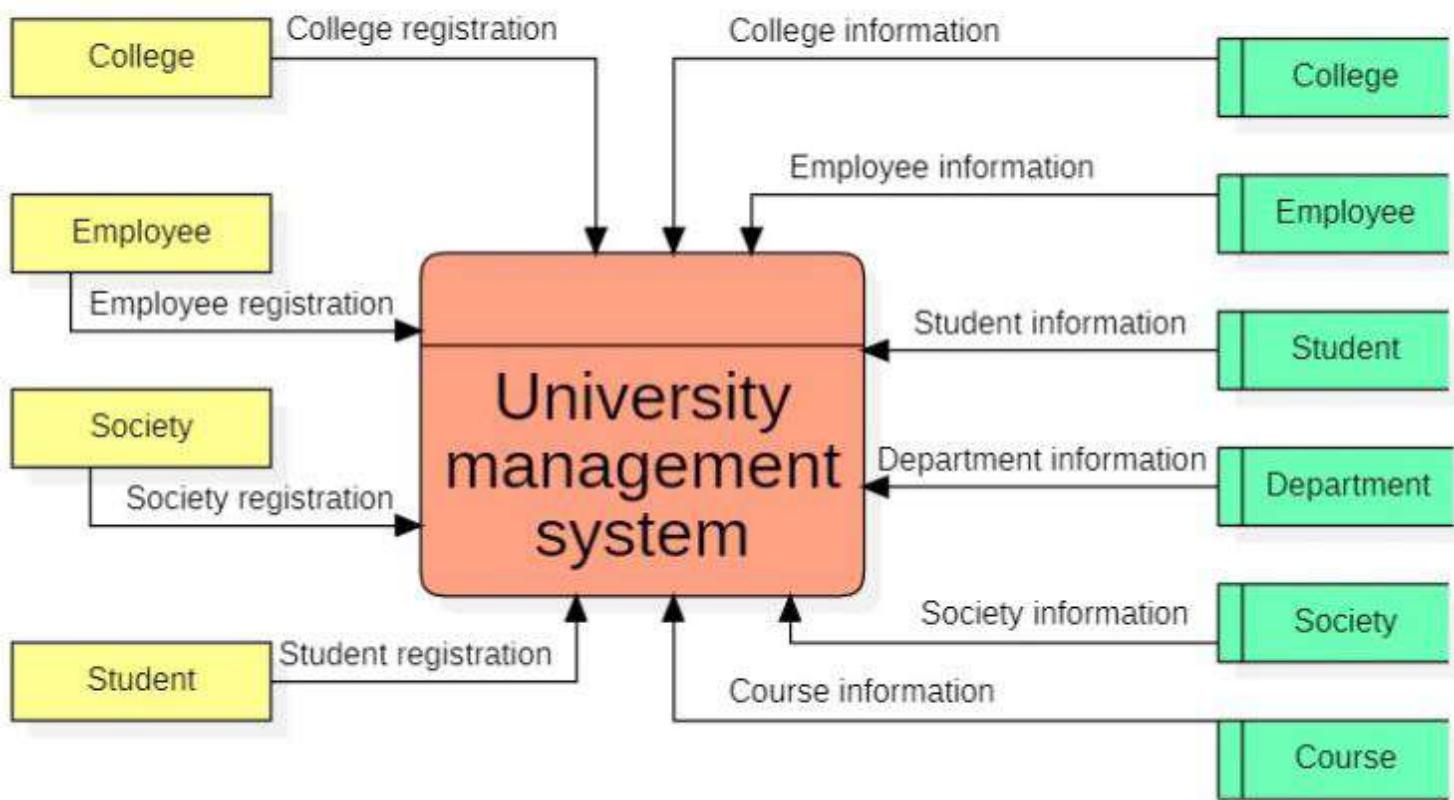
1-level DFD

In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main objectives of the system and breakdown the high-level process of 0-level DFD into subprocesses.

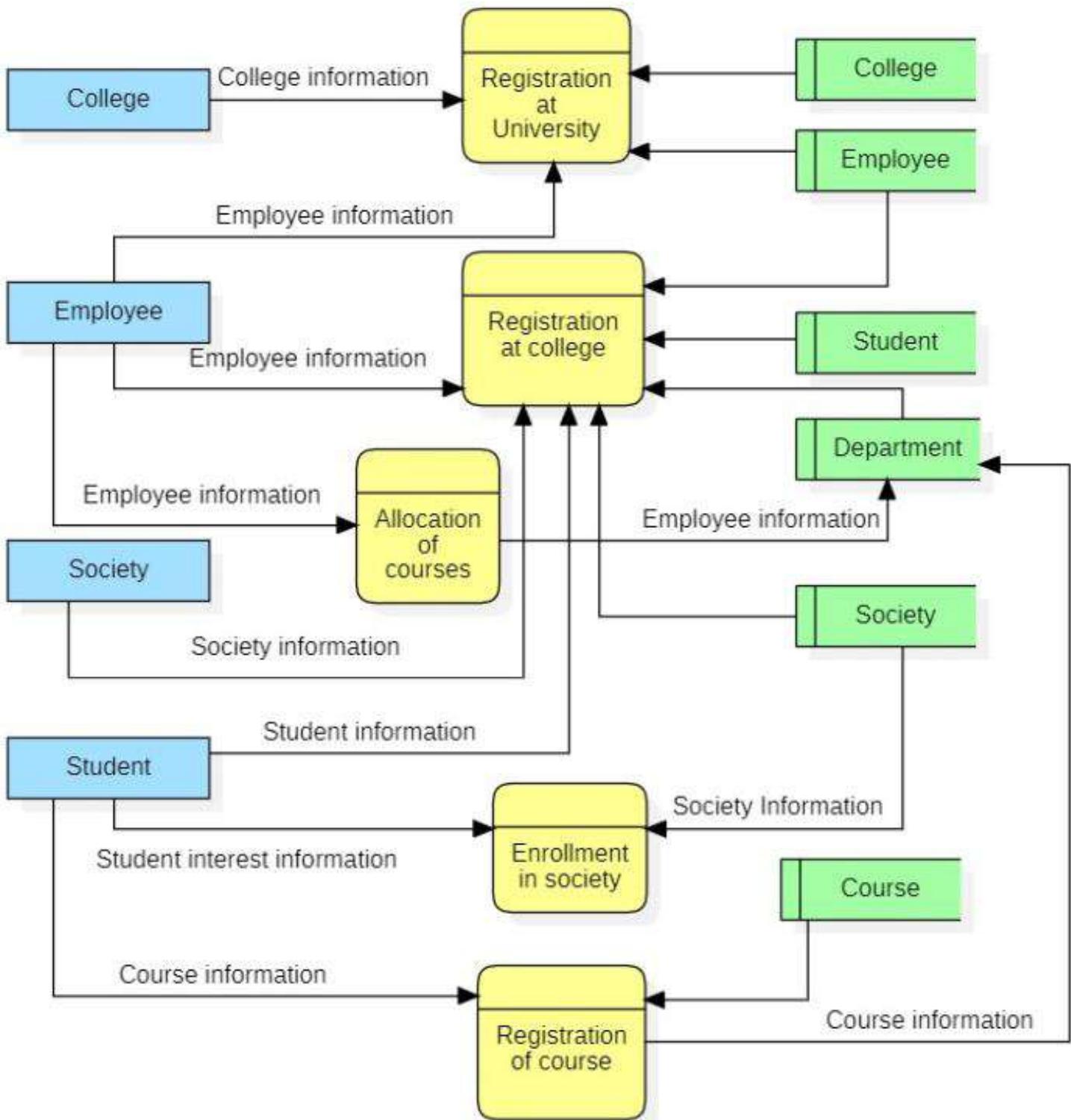
2-Level DFD

2-level DFD goes one process deeper into parts of 1-level DFD. It can be used to project or record the specific/necessary detail about the system's functioning.

0 - Level DFD

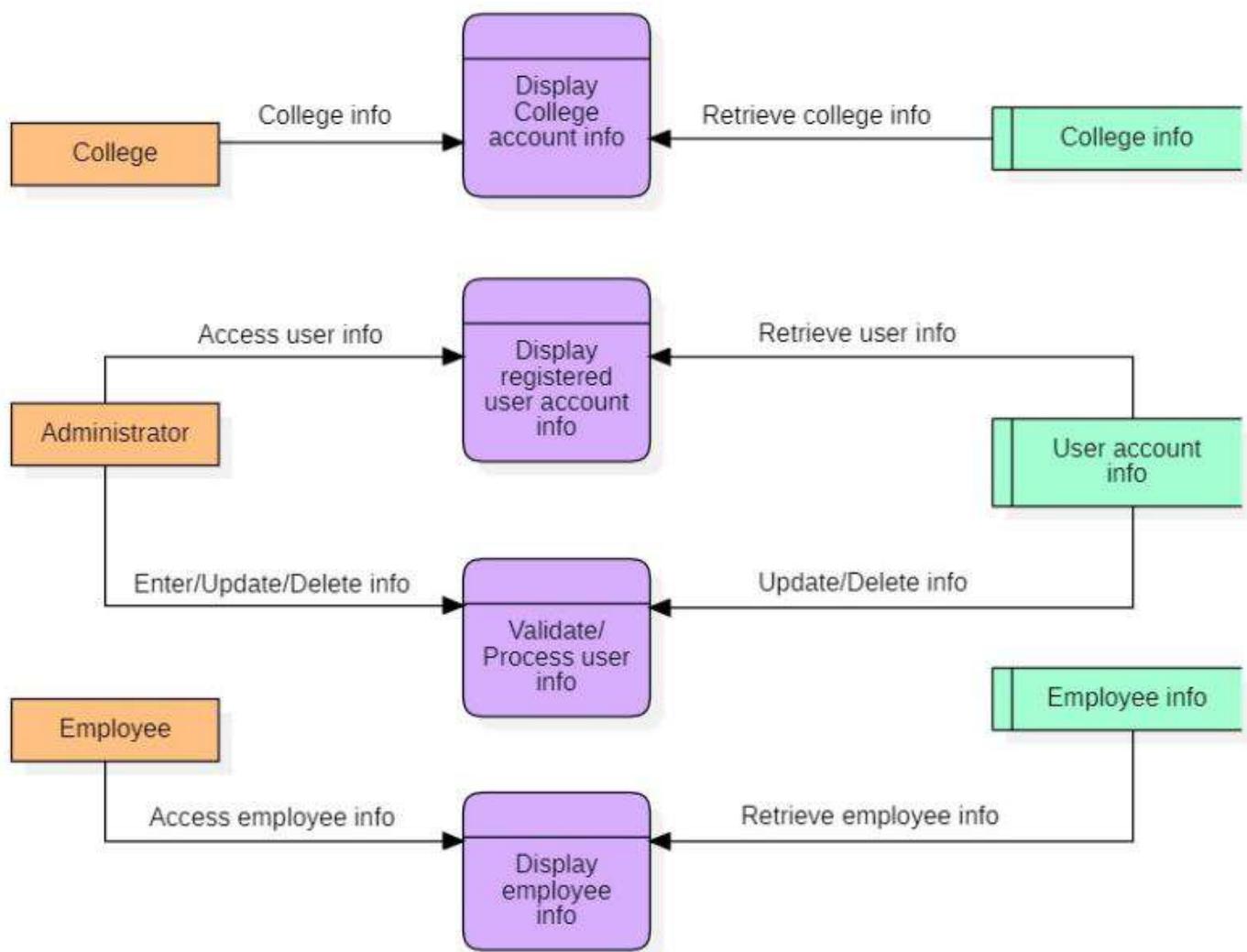


1 - Level DFD



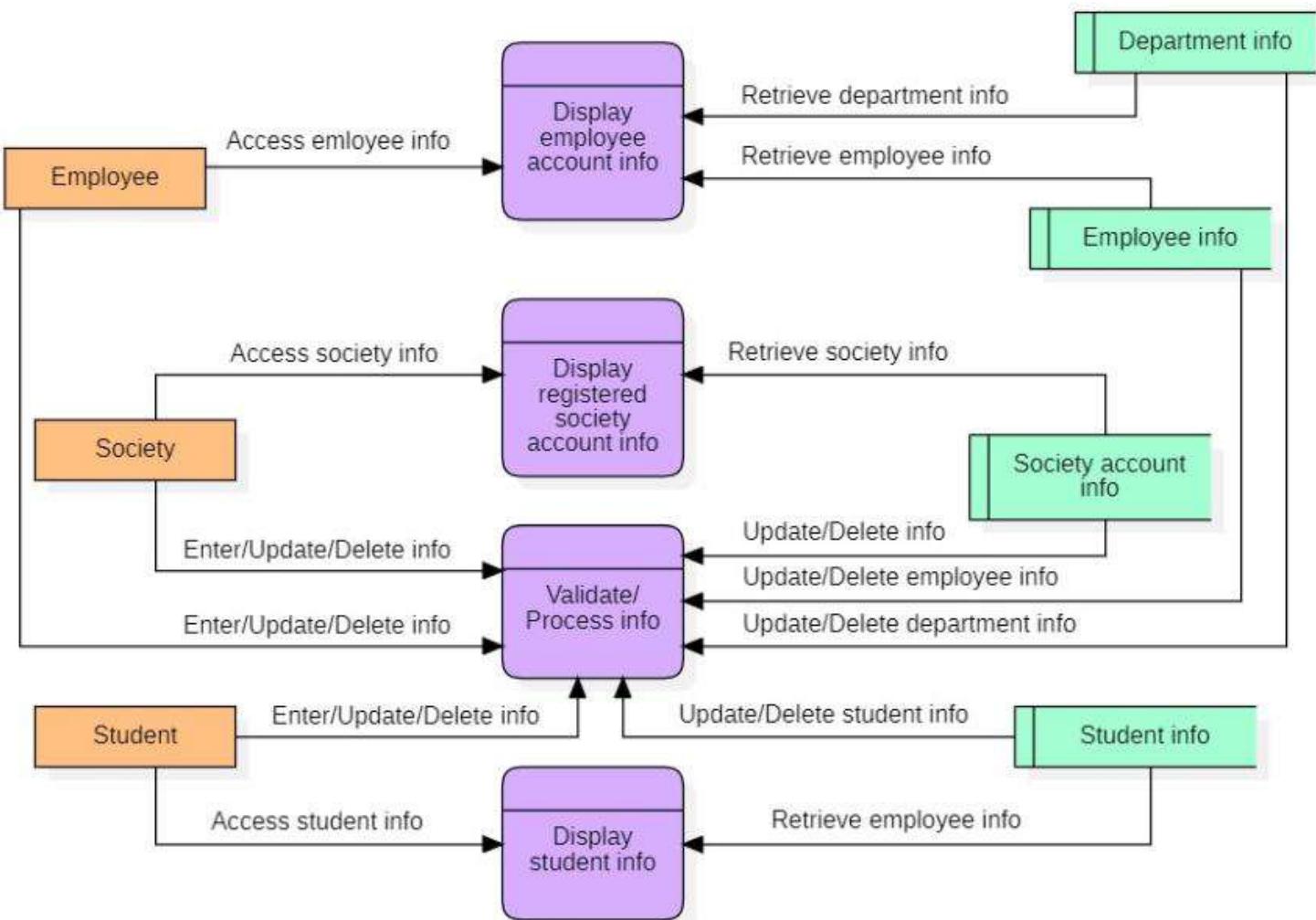
2 - Level DFD

1. Registration at University



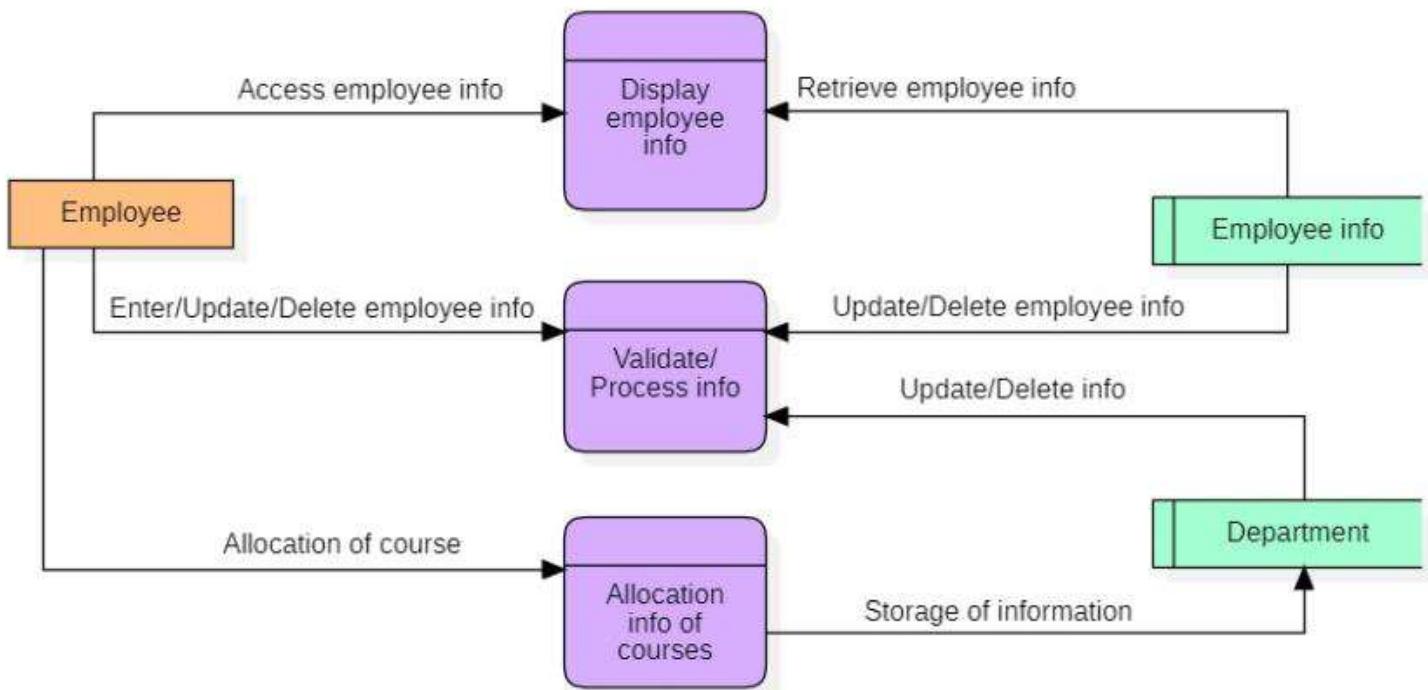
2 - Level DFD

2. Registration at College



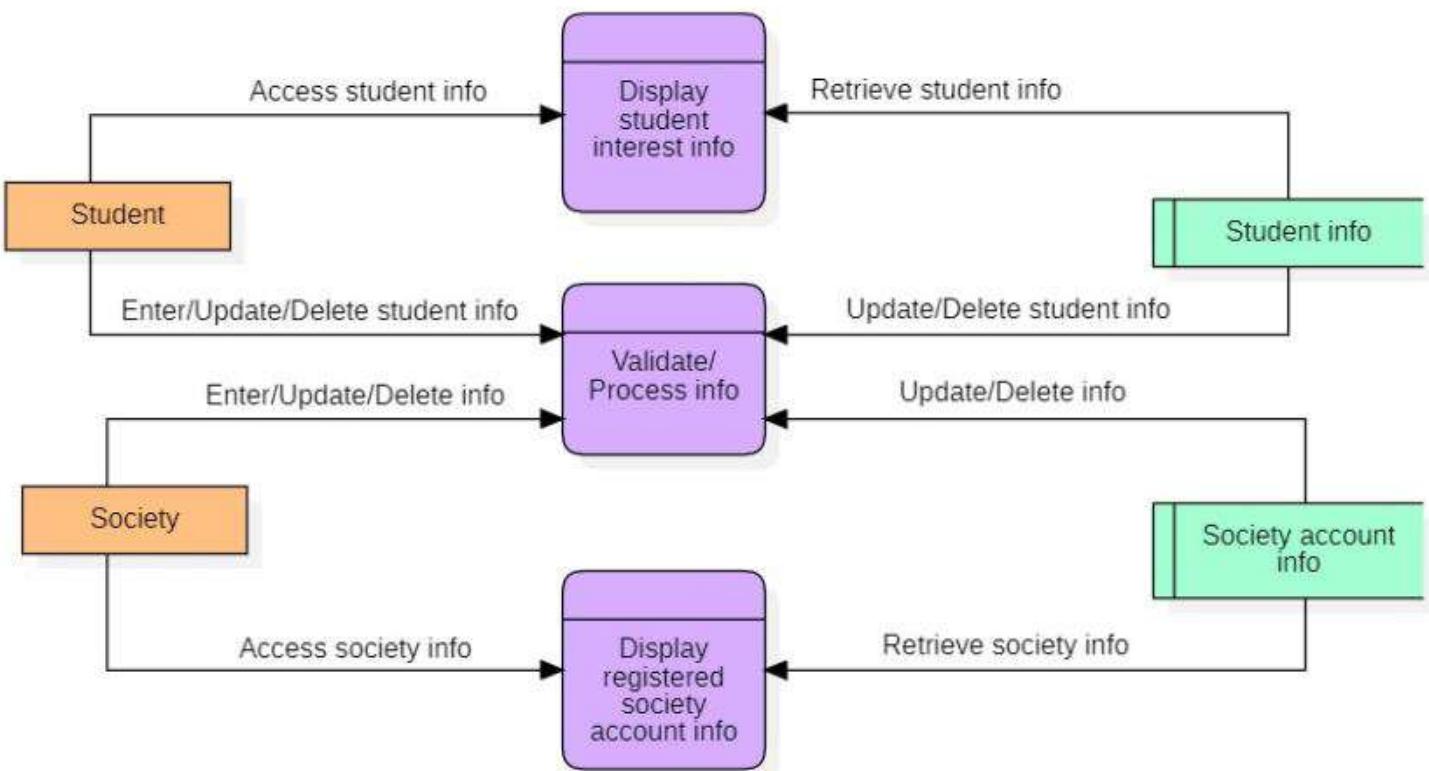
2 - Level DFD

3. Allocation of courses



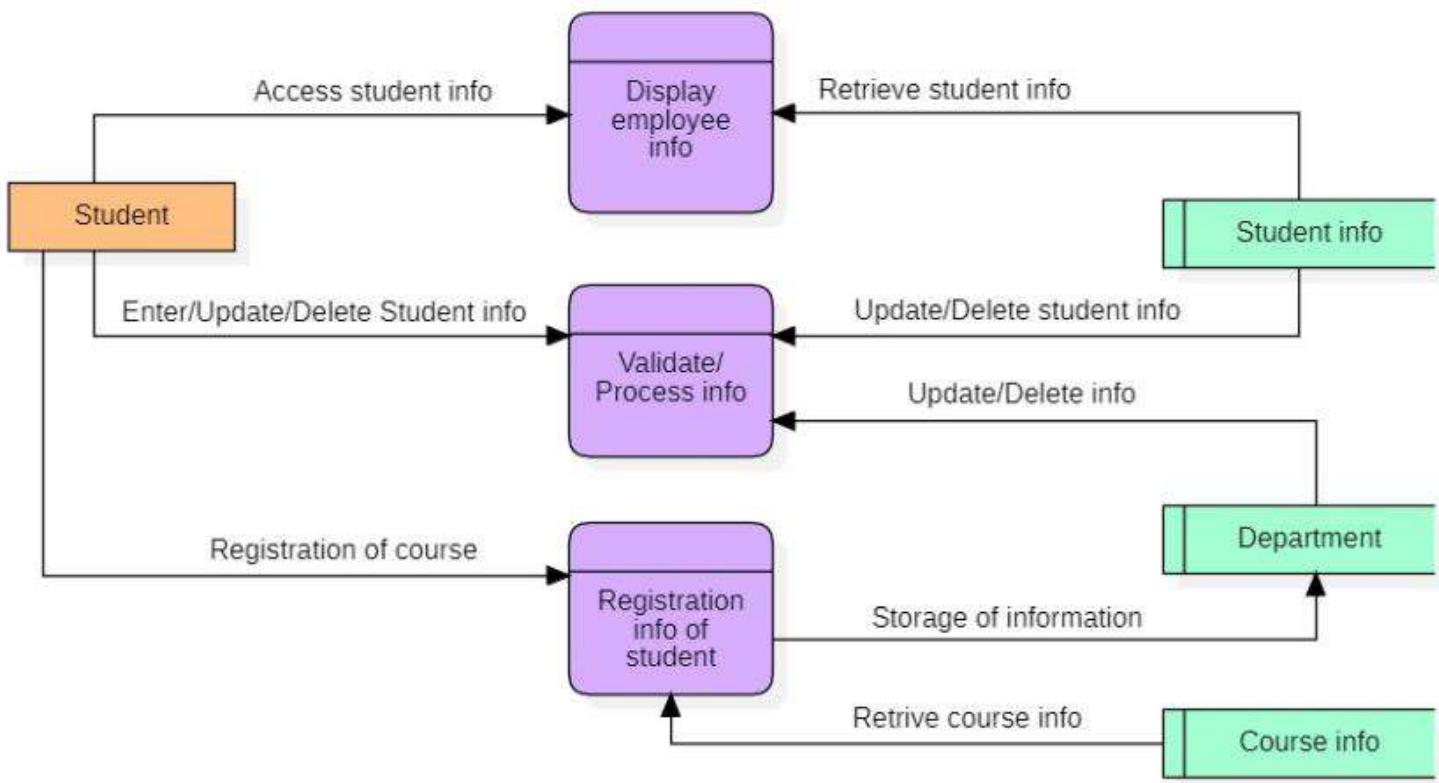
2 - Level DFD

4. Enrollment in society



2 - Level DFD

5. Registration of course



Experiment-08

AIM

To draw the Behavioral Diagram: State Chart Diagram.

Software used

StarUML

Theory

Behavioral Diagrams

A behavioral diagram shows how the system works ‘in motion’, that is how the system interacts with external entities and users, how it responds to input or event and what constraints it operates under.

Behavioral Diagram	Brief Description
Activity Diagram	It is a graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency
Use Case Diagram	It describes a system’s functional requirements in terms of use cases that enable you to relate what you need from a system to how the system delivers on those needs.
State Machine Diagram	It shows the discrete behavior of a part of a designed system through finite state transitions.
Sequence Diagram	It shows the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.
Communication Diagram	It shows interactions between objects and/or parts (represented as lifelines) using sequenced messages in a free-form arrangement.
Interaction Overview Diagram	It depicts a control flow with nodes that can contain other interaction diagrams.
Timing Diagram	It shows interactions when the primary purpose of the diagram is to reason about time by focusing on conditions changing within and among lifelines along a linear time axis.

State Chart Diagram

State chart diagram is one of the five UML diagrams used to model dynamic nature of a system. They define different states of an object during its lifetime. And these states are changed by events. So State chart diagrams are useful to model reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. So the most important purpose of State chart diagram is to model life time of an object from creation to termination.

Statechart diagrams are also used for forward and reverse engineering of a system. But the main purpose is to model reactive system.

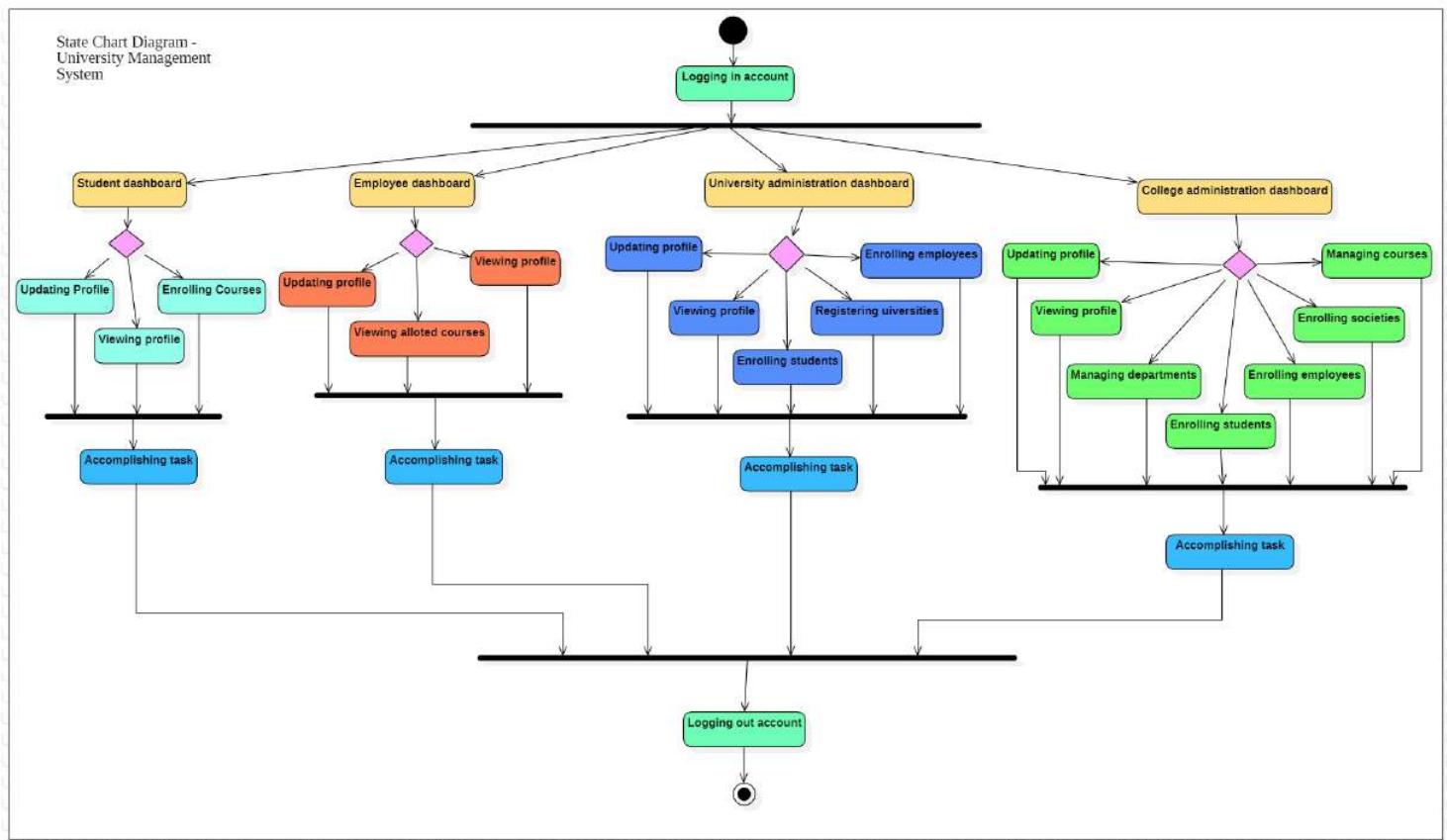
Following are the main purposes of using Statechart diagrams:

- To model dynamic aspect of a system.
- To model life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model states of an object.

Result

The experiment was studied and performed successfully.

State Chart Diagram -
University Management
System



Content Beyond Syllabus: 1

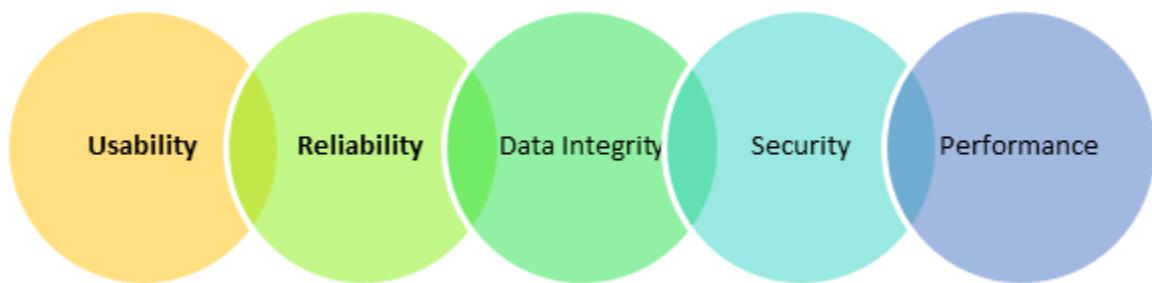
Aim: To study IOT Testing

The Internet of Things popularly known as IoT is the network that consists of devices, vehicles, buildings or any other connected electronic devices. This interconnection facilitates collection and exchange data. The 4 common components of an IoT system are

1. Sensor
 2. Application
 3. Network
 4. Backend (Data Center)
-

IOT testing is a type of testing to check IOT devices. Today there is increasing need to deliver better and faster services. There is a huge demand to access, create, use and share data from any device. The thrust is to provide greater insight and control, over various interconnected IOT devices. Hence, IOT testing framework is important.

Types of Testing in IOT



Testing for IoT devices broadly revolves around Security, Analytics, Device, Networks, Processors, Operating Systems, Platforms and Standards.

Let's investigate the broad testing types

Usability Testing:

There are so many devices of different shape and form factors are used by the users. Moreover, the perception also varies from one user to other. That's why checking usability of the system is very important in IoT testing.

Compatibility Testing:

There are lots of devices which can be connected through IOT system. These devices have varied software and hardware configuration. Therefore, the possible combination are huge. As a result, checking the compatibility in IOT system is important.

Reliability and Scalability Testing:

Reliability and Scalability is important for building an IOT test environment which involves simulation of sensors by utilizing virtualization tools and technologies.

Data Integrity Testing:

It's important to check the Data integrity in IOT testing as it involves large amount of data and its application.

Security testing:

In the IOT environment, there are many users accessing a massive amount of data. Thus, it is important to validate user via authentication, have data privacy controls as part of security testing.

Performance Testing:

Performance testing is important to create strategic approach for developing and implementing an IOT testing plan.

Following chart gives applicability of different testing types for various components of IOT.

IOT elements Testing Types	Sensor	Application	Network	Backend (Data Center)
Functional Testing	True	True	False	False
Usability Testing	True	True	False	False
Security Testing	True	True	True	True
Performance Testing	False	True	True	True
Compatibility Testing	True	True	False	False
Services Testing	False	True	True	True
Operational Testing	True	True	False	False

IOT Testing Process: Example Test Conditions

Test Categories	Sample Test Conditions
Components Validation	<ul style="list-style-type: none"> • Device Hardware • Embedded Software • Cloud infrastructure • Network Connectivity • Third-party software • Sensor Testing • Command Testing • Data format testing • Robustness Testing • Safety testing
Function Validation	<ul style="list-style-type: none"> • Basic device Testing • Testing between IOT devices • Error Handling • Valid Calculation
Conditioning Validation	<ul style="list-style-type: none"> • Manual Conditioning • Automated Conditioning •

Test Categories	Sample Test Conditions
Performance Validation	<ul style="list-style-type: none"> Conditioning profiles
Security and Data Validation	<ul style="list-style-type: none"> Data transmit Frequency Multiple request handing Synchronization Interrupt testing Device performance Consistency validation
Gateway Validation	<ul style="list-style-type: none"> Validate data packets Verify data loses or corrupt packets Data encryption/decryption Data values Users Roles and Responsibility & its Usage Pattern
Analytics Validation	<ul style="list-style-type: none"> Cloud interface testing Device to cloud protocol testing Latency testing
Communication Validation	<ul style="list-style-type: none"> Sensor data analytics checking IOT system operational analytics System filter analytics Rules verification
	<ul style="list-style-type: none"> Interoperability M2M or Device to Device Broadcast testing Interrupt Testing Protocol

Challenges of IOT testing

- You need to check both the network and internal communication
- Security is a big concern in IOT platform as all the task are operated using via Internet.
- The complexity of the software and system may hide bug present in the IOT technology

- Resource considerations such as limitations in memory, processing power, bandwidth, battery life, etc.

Best practices for effective IOT software testing

- Gray Box testing should be used with IOT testing as it allows to design effective test case. This allows you to know the OS, the architecture, third-party hardware, new connectivity and hardware device limitation.
- Real Time Operating System is vital to delivering the scalability, modularity, connectivity, security, which is important for IOT
- IoT Testing should be automated.

IOT Testing Tools:

Two most effective IOT testing tools are:

1.Shodan

2. Thingful

Conclusion:

- IOT is a connection of identifiable embedded devices with the existing Internet infrastructure.
- The complexity of the software and system may hide bug present in the IOT technology
- Gray Box testing should be used with IOT testing as it allows to design effective test case.
- IoT Testing ensures that the users get the improved user experience across all the connected IOT devices.
- As there is no test plan, part of the properties to be tested cannot be measured. So, errors/bugs may not be detected easily.

Content Beyond Syllabus-2

Aim: Implement Clone Testing.

Software Used: Star UML.

Theory:

Software engineering means building, evolving and maintaining software systems. Software engineering means set of problem solving skills, techniques, technology and methods applied upon a variety of domains to evolve and create useful systems that solve many problems like practical problems. Software engineer handles software engineering projects that discover, create, build software and tells its behaviour.

Software Cloning

Software clones are the regions of source code which are highly similar; these regions of similarity are called clones, clone classes, or clone pairs. While there are several reasons why two regions of code may be similar, the majority of the clone analysis literature attributes cloning activity to the intentional copying and duplication of code by programmers; clones may also be attributable to automatically generated code, or the constraints imposed by the use of a particular framework or library. In addition to these, some other issues, including programmer's behaviour such as laziness and the tendency to repeat common solutions, technology limitations, code understandability and external business forces have influences on code cloning. Cloning is the unnecessary duplication of data whether it is at design level or at coding level. Cloning works at the cost of increasing lines of code without adding to overall productivity. Same software bugs and defects are replicated that reoccurs throughout the software at its evolving as well as its maintenance phase. It results in excessive maintenance costs as well. So cut-paste programming form of software reuse deceptively raise the number of lines of code without expected reduction in maintenance costs associated with other forms of reuse. So, clones, is a promising way to reduce the maintenance cost in future.

Background of Software Cloning

Basic clone detection terms:

- 1) **Code Fragment:** A code fragment is a piece of code including function definition, begin-end block, or sequence of statements. We use file name and begin-end line numbers in the original code base to identify code fragment.
- 2) **Code Clone:** A code clone is a similar or duplicate code fragment in a source code or created either by replication or some modifications.
- 3) **Clone Pair:** if there is an equivalence relation between two code segments, then they form a clone pair.
- 4) **Clone Class:** It is defined as collection of similar code segments. Each code segment in a clone class forms a clone pair with other code segment of that class.

5) Exact Clones: Two or more code fragments are called exact clones if they are identical to each other with some differences in comments and whitespace or layout.

6) Renamed Clones: People use the term renamed clones when identifier names, literals values, comments or whitespace changes in the copied fragments. Thus, a renamed clone is essentially a Type II clone.

7) Near-Miss Clones: These are those clones where the copied fragments are very similar to the original. Editing activities such as changing in comments, layouts, changing the position of the source code elements through blanks and new lines, changing the identifiers and literals.

8) Semantic Clones: Semantic Clones are defined as functionally identical code fragments.

Types of Code Clones

There are basically four types of code clones. They are explained following:-

1) Type 1(Exact clones): In Type I clones, a copied code fragment is the same as the original. These code clones are identical code clones with some modification in white space and comments. Type I is widely known as Exact clones.

```
if (a >= b)
{ c = d + b;
```

```
// Comment1 d = d + 1;} else c = d - a;
//Comment2 An exact copy clone of this original copy could be as follows:
if (a>=b) { // Comment1' c=d+b; d=d+1;} else
// Comment2'
if (a >= b) { c = d + b; // Comment1 d = d + 1;} else c = d - a;
//Comment2 An exact copy clone of this original copy could be as follows: if (a>=b) { //
Comment1' c=d+b; d=d+1;}
else // Comment2'
```

2) Type 2(Renamed/parameterized clones): A Type II clone is a code fragment that is the same as the original except for some possible variations about the corresponding names of user-defined identifiers (name of variables, constants, class, methods and so on), types, layout and comments. Let us consider the following code fragment.

```
c = d + b; // Comment1 d = d + 1 ;
else c = d - a; //Comment2 A Type II clone for this fragment can be as follows:
if (m >= n) { // Comment1' y = x + n; x = x + 5; //Comment3 }
else y = x - m; //Comment2' We see that the two code segments change a lot in their
shape, variable names and value assignments. However, the syntactic structure is still
similar in both segments.
```

3) Type 3(Near Miss Clones):

These code clones are copied fragments by changing, adding or removing statements.

Consider the original code segment:

```
if (a >= b) { c = d + b; // Comment1 d = d + 1; }
else c = d - a;
```

//Comment2 If we now extend this code segment by adding a statement e = 1 then we can get, if (a >= b) { c = d + b; // Comment1 e = 1; // This statement is added d = d + 1; } else c = d - a; //Comment2 This copied fragments with one statement inserted is called Type III code clone of the original with a gap of one statement inserted.

4) Type 4(Semantic Clones):

These code clones are based on function similarity but they are different in syntax.

These clones are termed as Type IV semantic clones In this type of clones, the cloned fragment is not necessarily copied from the original. Two code fragments may be developed by two different programmers to implement the same kind of logic making the code fragments similar in their functionality. Functional similarity reflects the degree to which the components act alike. Let us consider the following code fragment 1, where the final value of 'j' is the factorial value of the variable VALUE.

Fragment 1: int i, j=1; for (i=1; i<=VALUE; i++) j=j*i;

Now consider the following code fragment 2, which is actually a recursive function that calculates the factorial of its argument n.

Fragment 2: int factorial(int n) { if (n == 0) return 1 ;
else return n * factorial(n-1) ; }

II. CLONE TESTING

A code clone means similar or duplicate code in a source code or code that is created either by replication or some modifications .These cloned code needs high maintenance cost of software and also cause the code bloating. This is because when changes are performed on one clone, then the same action is performed on respected clone, this will increase the maintenance. These clones can also increase risk of faults in system. Past research conclude that around 7% to 23% of the source code in a software system contains code clone. There are number of techniques and tools to detect the code clones, but it is not effective to remove the clones. Because code clones are needed for software to function properly. So we can apply the principal of refactoring or modularity to improve the reusability and maintainability of software from clone code.

Clone detection techniques

There are basically 4 types, that are:- textual, lexical, syntactic and semantic.

- 1) Textual approach:** In Textual approaches there is little need of normalization or transformation of code. In this, basically line to line comparison is done, which

basically based on two types, one is simple line matching and other one is parameterized line matching. This technique is basically string based.

- 2) Lexical approach: In lexical technique we convert source code into tokens using lexical rules. These tokens are then compared.
- 3) Syntactic approach: In syntactic technique an abstract tree is generated. Using parser source code is converted into parse tree. Abstract tree is then processed either using tree matching or metric to find the clones.
- 4) Semantic approach: In this approach, a source code is represented as program dependency graph. Nodes represent the statements and expressions and edges represent control and data dependencies.