

CSA1011-SOFTWARE ENGINEERING FOR PROJECT ESTIMATION

INDEX

EXP.NO	DATE	LIST OF PROGRAMS	PAGE.NO	MARKS	SIGN
1	12/07/21	a). Identifying Functional Requirements For Online Voting System	2		
	12/07/21	b). Identifying Functional Requirements For Library Management System	4		
2	13/07/21	a). Identifying Non Functional Requirements For Online Shopping System	8		
	13/07/21	b). Identifying Non Functional Requirements For Blood Donor System	10		
3	14/07/21	Estimation Of Effort And Project Duration Using Cocomo Model	13		
4	14/07/21	a).Uml Diagram For Online Voting System	15		
	14/07/21	b).Uml Diagram For Library Management System	19		
5	15/07/21	a).Uml Diagram For Online Shopping System	22		
	15/07/21	b).Uml Diagram For Course Registration System	25		
6	16/07/21	a). Uml Diagram For Blood Donor System	28		

191911219**A.SIRISHA**

	16/07/21	b). Uml Diagram For Hospital Management System	30		
7	17/07/21	a). Uml Diagram For e-Ticketing system	32		
	17/07/21	b). Uml Diagram For Restaurant Management System	35		
8	19/07/21	a). Using Raptor – Draw Flowchart To Display The Position Of The Numbers	37		
	19/07/21	b). Using Raptor - Draw And Validate Flowchart To Swap Two Characters	39		
9	20/07/21	a) Using Raptor - Draw And Validate Flowchart To Find Whether the Given Number Is Prime Or Not	41		
	20/07/21	b). Using Raptor - Draw And Validate Flowchart To Find Whether the Given Input Is Palindrome Or Not	44		
10	21/07/21	Find Cyclomatic Complexity for a graph having number of edges as 12, number of nodes as 13 and number of predicate nodes in the flow graph as 5.	46		

EXP NO: 1(a). FUNCTIONAL REQUIREMENTS FOR ONLINE VOTING SYSTEM**Date:****AIM:**

To perform analysis of the given problem statement for listing the functional requirements

PROBLEM STATEMENT:**ONLINE VOTING SYSTEM:**

Internet has led to discussion of e-democracy and online voting. Many peoples think that the internet could replace representative democracy, enabling everyone to vote on everything and anything by online voting .Online voting could reduce cost and make voting more convenient. This type of voting can be done for e-democracy, or it may be used for finalizing a solution, if many alternatives are present. Online voting make's use of authentication, hence it needs security, and the system must be able to address obtaining, marking, delivering and counting ballots via computer. Advantage of online voting is it could increase voter turnout because of convenience, and it helps to reduce fraud voting.

FUNCTIONAL REQUIREMENTS:

- **Mobility:** The voter should not be restricted to cast his ballot at a single poll-site at his home precinct.
- He shall be able to vote from any poll-site within the nation.

- **Convenience:** The system shall allow the voters to cast their votes quickly, in one session, and should not require many special skills or intimidate the voter (to ensure Equality of Access to Voters).

- **User-Interface:** The system shall provide an easy-to-use user-interface. Also, it shall not disadvantage any candidate while displaying the choices (e.g., by requiring the user to scroll down to see the last few choices).

- **Transparency:** Voters should be able to possess a general knowledge and understanding of the voting process.

- **Flexibility:** The system shall be flexible in that it allows a variety of ballot question formats including open-ended questions (e.g. Write-in candidates and survey questions).

- **Support for Disabled Voters:** The system shall cater to the needs of physically challenged voters (e.g. blind voters).
- **Accuracy:** The system shall record and count all the votes and shall do so correctly.
- **Eligibility:** Only authorized voters, who are registered, should be able to vote.
- **Uniqueness:** No voter should be able to vote more than once.
- **Audit ability:** It should be possible to verify that all votes have been correctly accounted for in the final election tally, and there should be reliable and demonstrably authentic election records, in terms of physical, permanent audit trail (which should not reveal the user's identity in any manner).
- **Voter Confirmation:** The voter shall be able to confirm clearly how his vote is being cast, and shall be given a chance to modify his vote before he commits it.
- **Provisional Ballots:** The voter shall be able to vote with a provisional (electronic) ballot if he has some registration problems, which could be counted if verified by the authorities later.
- **Documentation and Assurance:** The design, implementation, and testing procedures must be well documented so that the voter-confidence in the election process is ensured.

RESULT:

The given problem statement is analyzed and various functional requirements are listed.

EXP NO: 1(b). FUNCTIONAL REQUIREMENTS FOR LIBRARY MANAGEMENT SYSTEM**Date:****AIM:**

To perform analysis of the given problem statement for listing the functional requirements

PROBLEM STATEMENT:**LIBRARY INFORMATION SYSTEM:**

As the size and capacity of the institute is increasing with the time, it has been proposed to develop a Library Information System (LIS) for the benefit of students and employees of the institute. LIS will enable the members to borrow a book (or return it) with ease while sitting at his desk/chamber. The system also enables a member to extend the date of his borrowing if no other booking for that particular book has been made. For the library staff, this system aids them to easily handle day-to-day book transactions. The librarian, who has administrative privileges and complete control over the system, can enter a new record into the system when a new book has been purchased, or remove a record in case any book is taken off the shelf. Any non-member is free to use this system to browse/search books online. However, issuing or returning books is restricted to valid users (members) of LIS only. The final deliverable would a web application (using the recent HTML 5), which should run only within the institute LAN. Although this reduces security risk of the software to a large extent, care should be taken no confidential information (eg., passwords) is stored in plain text.

FUNCTIONAL REQUIREMENTS:**Administrator Menu:**

- Admin should be able to insert, modify and delete books.
- Can accept or reject a new user according to the library policy.
- Can get the information of any member who has borrowed a book.
- Add and edit book categories and arrange books by categories.
- Can record books returned by users.

User Requirements:

- Allow user to view quick reports like book issues/ returned etc. in particular time.
- Stock verification in different criteria or according to different categories.

Usability:

- The system shall allow the users to access the system from the Internet using HTML or its derivative technologies. The system uses a web browser as an interface.

- Since all users are familiar with the general usage of browsers, no specific training is required.
- The system is user friendly and self-explanatory.

New user Registration:

Any member of the institute who wishes to avail the facilities of the library has to register himself with the library information system.

Search book:

The system must be able to search the database based on select search type, must be able to filter book based on keyword entered and must be able to show the filtered book in table view.

Reliability:

The system has to be very reliable due to the importance of data and the damages incorrect or incomplete data can do

Availability:

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

Issue books and return books:

The system must be able to enter issue information in database, must be able to update number of books, must be able to search if book is available or not before issuing books and should be able to enter issue and return date information

Accuracy:

The accuracy of the system is limited by the accuracy of the speed at which the employees of the library and users of the library use the system.

Event addition:

System should be able to add detailed information about events and display information on notice board available in the homepage of site.

RESULT:

The given problem statement is analyzed and various functional requirements are listed.

EXP NO: 2(a). NON-FUNCTIONAL REQUIREMENTS FOR ONLINE SHOPPING SYSTEM**Date:****AIM:**

To perform analysis of the given problem statement for listing the functional requirements

PROBLEM STATEMENT:**ONLINE SHOPPING SYSTEM:**

In day to day life, we will need to buy lots of goods or products from a shop. It may be food items, electronic items, house hold items etc. Now days, it is really hard to get some time to go out and get them by ourselves due to busy life style or lots of works. In order to solve this, B2C E-Commerce websites have been started. Using these websites, we can buy goods or products online just by visiting the website and ordering the item online by making payments online. This existing system of buying goods has several disadvantages. It requires lots of time to travel to the particular shop to buy the goods. Since everyone is leading busy life now a day, time means a lot to everyone. Also there are expenses for travelling from house to shop. More over the shop from where we would like to buy something may not be open 24*7*365. Hence we have to adjust our time with the shopkeeper's time or vendor's time. In order to overcome these, we have e-commerce solution, i.e. one place where we can get all required goods/products online. The proposed system helps in building a website to buy, sell products or goods online using internet connection. Purchasing of goods online, user can choose different products based on categories, online payments, delivery services and hence covering the disadvantages of the existing system and making the buying easier and helping the vendors to reach wider market

NON-FUNCTIONAL REQUIREMENTS:**Maintainability:**

- The system should be developed in such a way that changes can be made easily, whether for bug fixes or to add new functionality.
- The system should be easy enough to maintain that someone else could do it with a manual and a few hours training.

Portability:

- The system should be portable to various operating environments.
- Should the current hosting become too restricting for the system, the system must be portable enough to be moved over to a new server with minimal downtime.

Integrity:

- The system should be able to protect and preserve transactions.

Manageability:

- The system should be developed in such a way that it can be easily reused, deployed and tested.

Usability:

- The user interface of the system should be very user friendly.
- It should not take more than 120 seconds for a new user to register for an account.
- It should not take more than 90 seconds for a registered user to place an order.

Security:

- There needs to be clearly defined roles of the users. These roles are 'customer' and 'administrator'. Each person that goes to the system's website will be required to register if they want to do more than just read / browse site content.
- A secure server will be required to ensure confidentiality of customer's credit card and other details.

Simplicity:

- The system shall be designed to be extremely simple, as complexity is the enemy of security.

System Accountability:

- Ensure that system operations are logged and audited.

Personnel Integrity:

- Those developing and operating the voting system should have unquestionable records of behavior.

RESULT:

The given problem statement is analyzed and various non-functional requirements are listed.

EXP NO: 2(b). NON-FUNCTIONAL REQUIREMENTS FOR BLOOD DONOR SYSTEM USING

Date:

AIM:

To draw use-case diagram, class diagram and activity diagram for blood donor system using Umbrello.

PROBLEM STATEMENT:

In times of medical emergency, blood banks play a major role in deciding the fate of a person. The blood donor system helps the user to order specific blood type from the local blood reserves, while also enabling users to donate blood to the local blood reserves.

List of Non functional requirements.

- **Reliability**
- **Efficiency**
- **integrity**
- **usability**
- **testability**
- **Maintainability portability**
- **Availability.**

NON-FUNCTIONAL REQUIREMENTS:

Reliability:-

- The blood donor system performs its functions in on precision of requirements. so that the organizing the system becomes easy. to get and supply the blood.

Efficiency: -

- The blood donas system works efficiently with proper set of instructions and categories that makes simpler to understand and donate blood or to receive blood..

Integrity:-

- The system provides stronger authentication and access. so that an authorized users can not enter into the system, so that it ensures the security of the system.

usability :-

- The system has an effective user interface, so that user can enter in to system. and perform his activities effectively, and it ensures the user friendly atmosphere with the system.

Testability :-

- it ensures blood group the user with correct to be donated and or to get correct blood, by testing and delivering the right blood to the user.

Maintainabilit:-

- The system is always locates and fixes the errors and impurities present the system. So that it maintains from unintended operations performed by the unauthorital people.

portability:

- The system ensures that the ability to transfer from one hardware /software to another or operating system to another .so that it is accessible for any one in.any where.

Availability: -

- The system ensures 24/7 availability So that the user can get blood at any time in his critical situations,even at the night. or system can get blood from a donor in an emergency situations. at any time to save person who is in the critical situation.

RESULT:

The given problem statement is analyzed and various non-functional requirements are listed.

EXP NO: 3 ESTIMATION OF EFFORT AND PROJECT DURATION USING COCOMO MODEL**Date:****AIM**

To estimate Effort and Project Duration using Basic Cost Constructive Model.

PROBLEM STATEMENT

Estimation of Software Maintenance Cost is carried out by COCOMO. For different modes of operation of COCOMO (Organic, Semi-detached and Embedded),

Calculate the following: a) Effort in Staff months b) Project Duration using the data given below:

KLOC (thousands of lines of code) = 200

Constants to be used for Effort in Staff Months:

Mode	A	b
Organic	2.4	1.05
Semi-Detached	3.0	1.12
Embedded	3.6	1.20

Constants to be used for Project Duration:

Mode	C	d
Organic	2.5	0.38
Semi-Detached	2.5	0.35
Embedded	2.5	0.32

PROBLEM ANALYSIS:

- a. Effort in Staff-Month

$$E = a(KLOC)^b$$

Where KLOC – Kilo Lines of code and a,b are constants.

- b. Project Duration in Months

$$T = c(E)^d$$

Where E is the effort in months and c,d are constants

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
inti,ss[3];
float kloc,E[3],D[3],P[3];
float a[3] = { 2.4, 3.0, 3.6 };
float b[3] = { 1.05, 1.12, 1.20 };
float c[3] = { 2.5, 2.5, 2.5 };
float d[3] = { 0.38, 0.35, 0.32 };
void calc(int i);
void main()
{
    clrscr();
    printf("\nEnter the value of kloc:");
    scanf("%f",&kloc);
    printf("\n\nESTIMATIONS FOR BASIC COCOMO ARE :");
    printf("\n\nFor Organic Mode :");
    calc(0);
    printf("\n\nForSemidetatched Mode :");
    calc(1);
    printf("\n\nFor Embedded Mode :");
    calc(2);
    getch();
}
voidcalc(int i)
```

191911219

A.SIRISHA

```
{
    E[i] = a[i]*(pow(kloc,b[i]));
    D[i] = c[i]*(pow(E[i],d[i]));
    printf("\n\tEffort, E = %f PM",E[i]);
    printf("\n\tDevelopmentTime,D= %fM",D[i]);
}
```

OUTPUT:

```
Enter the value of kloc:170

ESTIMATIONS FOR BASIC COCOMOARE :

For Organic Mode :
    Effort,E=527.451477PM
    DevelopmentTime,D= 27.062590M

For Semidetatched Mode:
    Effort,E=944.548035PM
    DevelopmentTime,D= 27.495930M

For Embedded Mode :
    Effort,E=1709.390503PM
    DevelopmentTime,D= 27.067566M
```

RESULT:

Thus the cocomo model is executed and output is verified successfully.

Exp No: 4(a).**UML DIAGRAM FOR ONLINE VOTING SYSTEM****Date:****AIM:**

To draw USE-CASE diagram for Online voting system using Umbrello.

PROBLEM STATEMENT:

Develop a scenario based modeling and draw the appropriate use case diagrams for the below given "Online Voting System". Internet has led to discussion of e-democracy and online voting. Many peoples think that the internet could replace representative democracy, enabling everyone to vote on everything and anything by online voting .Online voting could reduce cost and make voting more convenient. This type of voting can be done for e-democracy, or it may be used for finalizing a solution, if many alternatives are present. Online voting make's use of authentication, hence it needs security, and the system must be able to address obtaining, marking, delivering and counting ballots via computer. Advantage of online voting is it could increase voter turnout because of convenience, and it helps to reduce fraud voting.

USE CASE SCENARIOS:**The voter scenarios are:**

- The voter login to the system.
- The voter can begin to vote at any time.
- We have only 1 change to change a vote
- Caste vote
- View online help

Admin scenario arc:-

- Admin can login to the system
- Admin has a activities like update and delete or add new candidate
- File can give the results of the elections
- He can filter the results

191911219

A.SIRISHA

- He have the permissions like modified the voter profile etc.

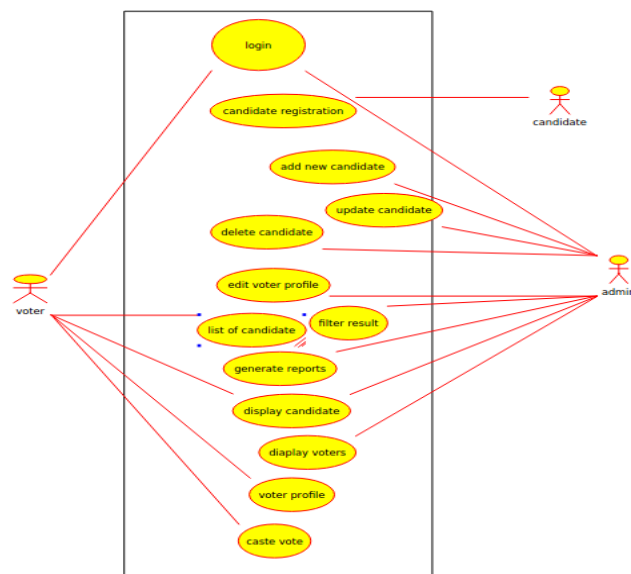
Login:-Both the admin and valid voter can login into the system. The admin should provide a password and login ID to the voters before the election starts.

Voter profile:-The voter can seen their profile but changes not be done by the voter only admin can change the voter profile.

Add candidate:-The admin can have the permissions like to add the candidate or update the details of the candidate .

Caste vote:-The voter can use the vote on the day of polling can be due to their respective candidate.

USE CASE DIAGRAM:



RESULT:

Thus the use-case diagram for online voting system is drawn and verified successfully.

Exp No: 4(b).**UML DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM****Date:****AIM:**

To draw USE-CASE diagram for Library management system using Umbrello.

INTRODUCTION:

In the Unified Modeling Language, a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that it helps those entities (known as actors) achieve
- The scope of your system

Common components include:

- **Actors** - the users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.
- **System** - a specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.
- **Goals** - the end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

USE CASE DIAGRAM OBJECTS

- Actor
- Use case
- System
- Package

The objects are further explained below.

Actor:

Actor in a use case diagram is **any entity that performs a role** in one given system. This could be a person, organization or an external system and usually drawn like skeleton shown below.

Use Case:

A use case **represents a function or an action within the system**. Its drawn as an oval and named with the function.

System:

System is used to **define the scope of the use case** and drawn as a rectangle. This an optional element but useful when your visualizing large systems. For example you can create all the use cases and then use the system object to define the scope covered by your project. Or you can even use it to show the different areas covered in different releases.

Package:

Package is another optional element that is extremely useful in complex diagrams. Similar to **class diagrams**, packages are **used to group together use cases**. They are drawn like the image shown below.

USE CASE SCENARIOS:**LOGIN:**

To interact with the system, LMS will validate its registration with this system.

The actors involved are

- Administrator
- Librarian
- member

View user details:

1. To see the details of the registered user & the books currently borrowed from the library.
2. Member can involve.

191911219

A.SIRISHA

3. User must be logged into the system.

View books:

1. To display the details, when a member, guest or administrator want to see the details on the available books.

2. The Actors involved in step are Administrator, guest and member.

Reserve Books:

1. User can reserve a book by inputting the relevant details and the librarian can also reserve a book for a member

Search books:

1. Member or guest can search for a particular book in the book library by book name or category or author name.

Issue books:

1. This use case can describe the process of issuing a certain book for a member by librarian.

2. Get the member ID and book ID before issue a book.

3. Check the availability.

Return books:

1. This use case describes the process of return a book.

2. If return book is late member should be paid fine.

View Members:

To display the details, when a member, guest or administrator wants to see the details of the registered user.

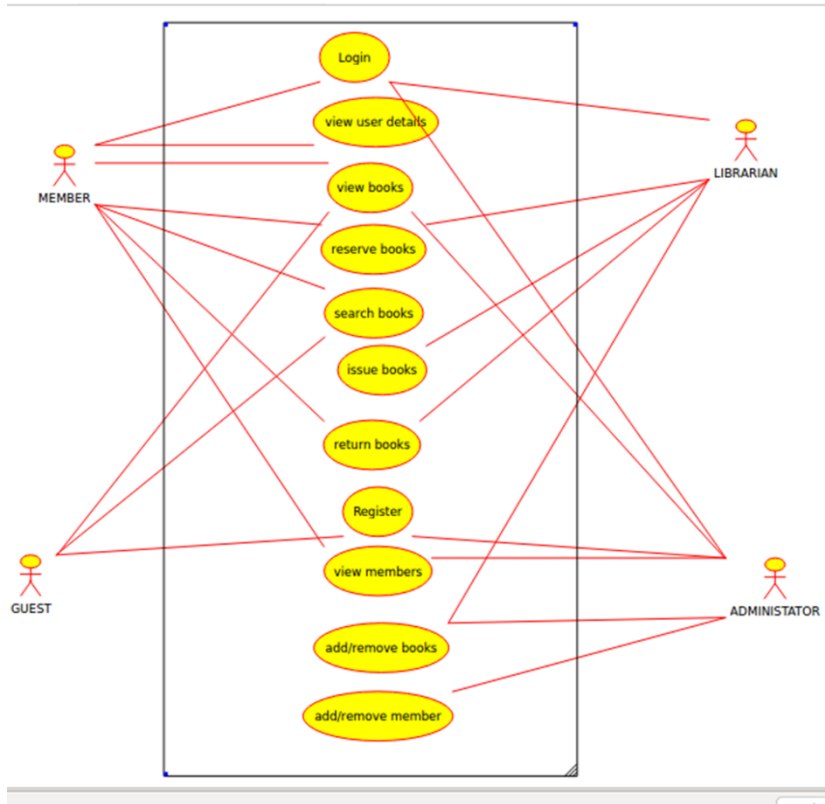
Add /remove members:

Only administrator is allowed to add or remove a member from library data base .to remove a member, member should request to leave the library.

191911219

A.SIRISHA

USE CASE DIAGRAM:



RESULT:

Thus the use-case diagram for online voting system is drawn and verified successfully.

Exp No: 5a.**UML DIAGRAM FOR ONLINR SHOPPING SYSTEM****Date:****AIM:**

To draw USE-CASE diagram for Online voting system using Umbrello.

PROBLEM STATEMENT:**Online shopping system:**

In day to day life, we will need to buy lots of goods or products from a shop. It may be food items, electronic items, house hold items etc etc. Now a days, it is really hard to get some time to go out and get them by ourselves due to busy life style or lots of works. In order to solve this, B2C E-Commerce websites have been started. Using these websites, we can buy goods or products online just by visiting the website and ordering the item online by making payments online. This existing system of buying goods has several disadvantages. It requires lots of time to travel to the particular shop to buy the goods. Since everyone is leading busy life now a days, time means a lot to everyone. Also there are expenses for travelling from house to shop. More over the shop from where we would like to buy something may not be open 24*7*365. Hence we have to adjust our time with the shopkeeper's time or vendor's time. In order to overcome these, we have e-commerce solution, i.e. one place where we can get all required goods/products online. The proposed system helps in building a website to buy, sell products or goods online using internet connection. Purchasing of goods online, user can choose different products based on categories , online payments , delivery services and hence covering the disadvantages of the existing system and making the buying easier and helping the vendors to reach wider market.draw a use case diagram for given scenario.

Login/Register:

- The user is a visiter, who visits the site and buy something from our website by making an online payment (or) otherwise he can register.
- Both customer and admin have different task in online system.

191911219

A.SIRISHA

Admin:

Admin is a responsible person to run the whole system. Admin can add delete, update all information, order information, item information.

Admin scenarios:

- Login module
- Manage items
- Process a order
- Cancel the order
- Process return
- Online help

Customer scenarios:

- Registration for new customer
- Login for existing customers
- Place a order
- Browser the item
- Change the password
- Cancel the order
- Make payment

Courier service scenarios are:

- The courier service has to provide by the deliver the orders to the customers.
- And also return items for the customer's ad send to the back to sales marketing.

Payment:

191911219

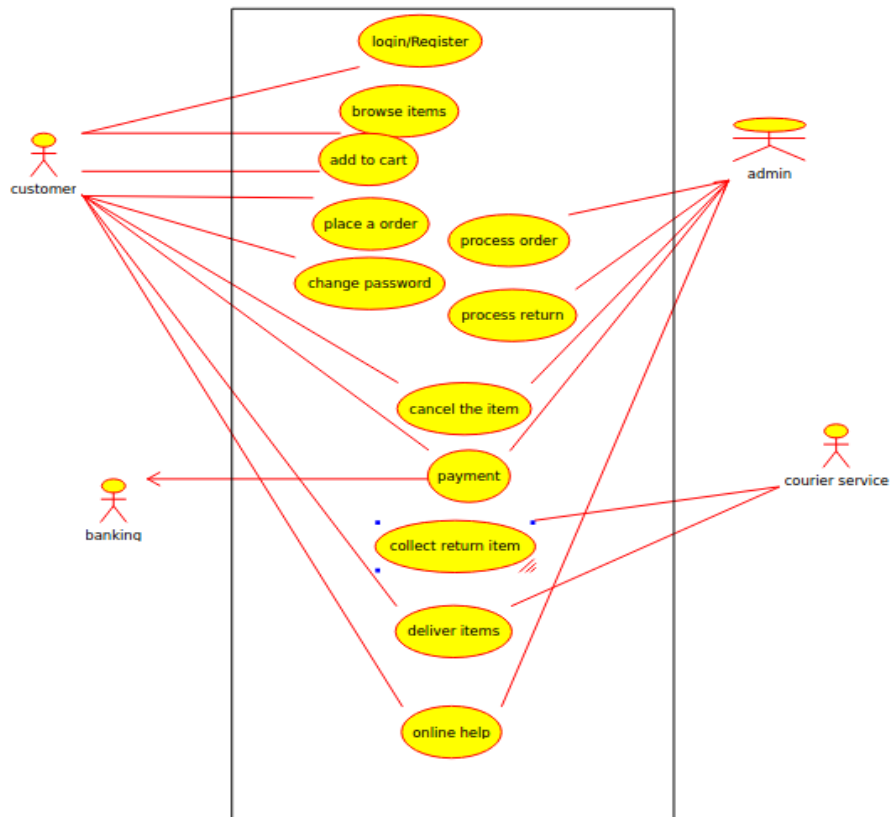
A.SIRISHA

The customer choose the payment option when he was placed a order.The payment is handled by the bank and the payment which could be done either by using credit card and debit card or cash on delivery.

Online help:

When the customers are needs a help the system will provide the online help system.

USE CASE DIAGRAM:



RESULT:

Thus the use-case diagram for online shopping system is drawn and verified successfully.

Exp No: 5b. UML DIAGRAM FOR COURSE REGISTRATION SYSTEM**Date:****AIM:**

To draw USE-CASE diagram for course registration system using Umbrello.

PROBLEM STATEMENT:

The new on-line registration system will allow students to select four course offerings for the coming semester. In addition, each student will indicate two alternative choices in case a course offering becomes filled or cancelled. No course offering will have more than ten students. No course offering will have fewer than three students. A course offering with fewer than three students will be cancelled. Once the registration process is completed for a student, the registration system sends information to the billing system, so the student can be billed for the semester.

Professors must be able to access the on-line system to indicate which courses they will be teaching. They will also need to see which students signed up for their course offering.

USE CASE DIAGRAM:

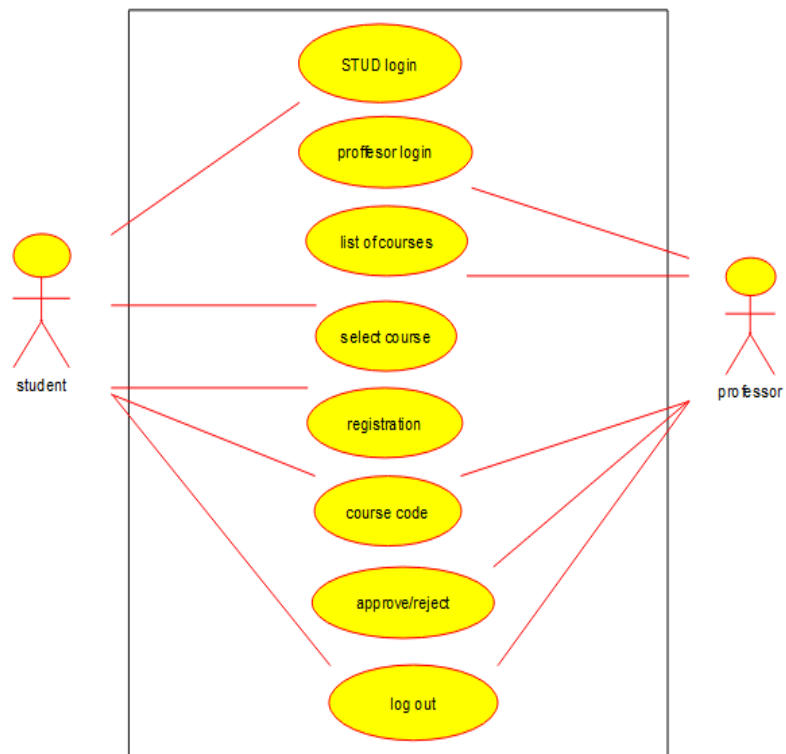
In course registration, there are two users and they can do the required activities after login. The actors are pointed to the use case as they proceed the activity.

Actors :

1. Professor
2. Student

USE CASE :

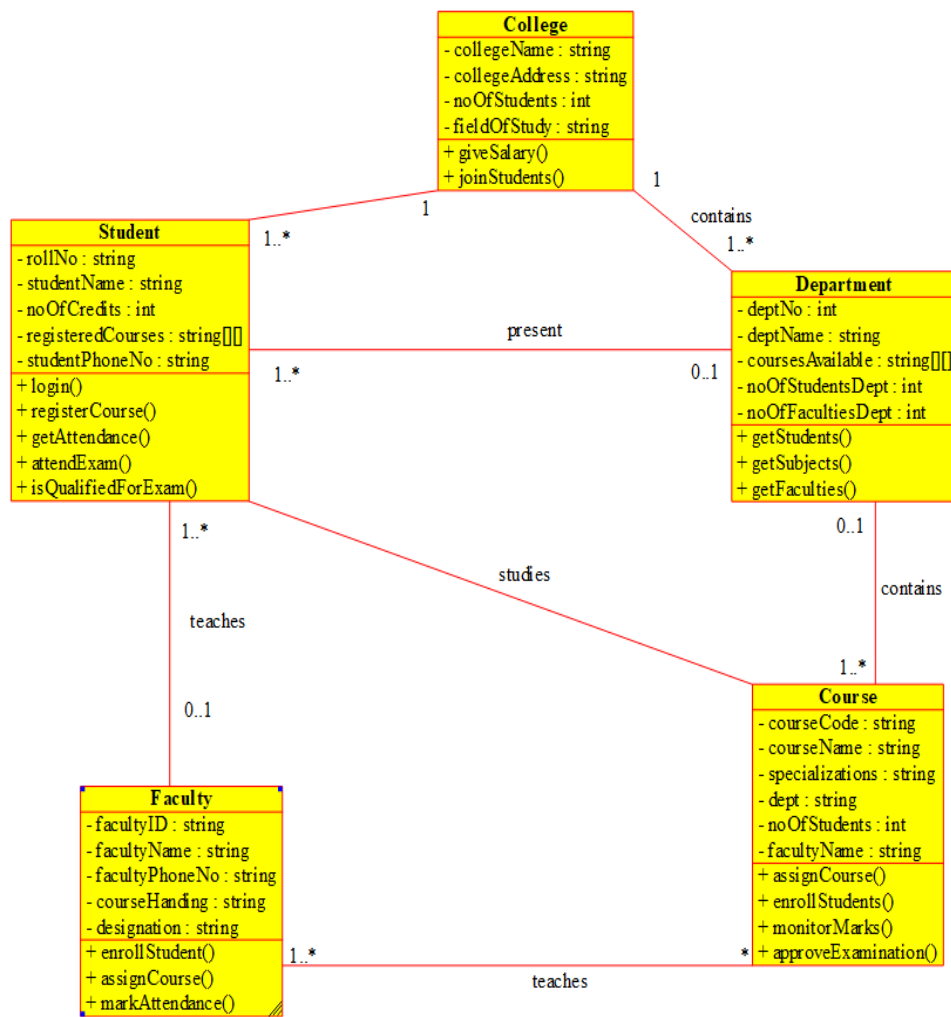
1. Login
2. List of Courses
3. Selected Course
4. Registration
5. Course Code
6. Approved / Rejected
7. Logout



CLASS DIAGRAM:

In online course registration system class diagram describes the structured classdiagram of library their attributes, methods and relationship among objects.

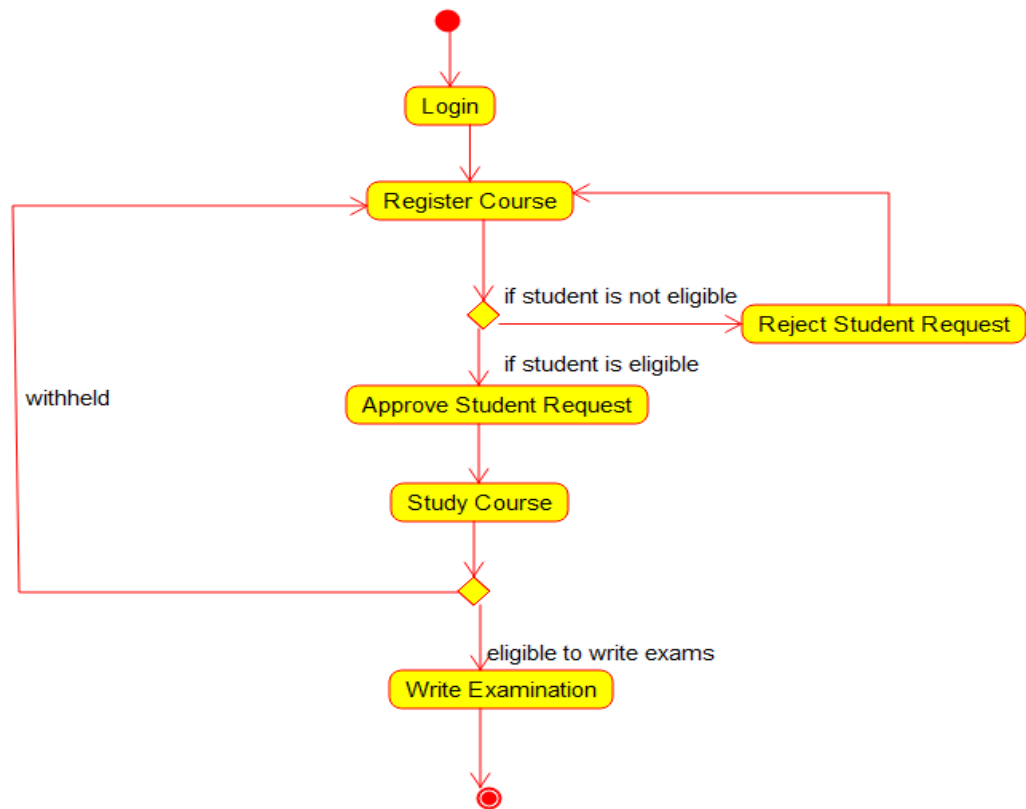
- ☐ Class
- ☐ Attributes
- ☐ Operations



ACTIVITY DIAGRAM:

The activity diagram of online course registration system used to describe flow of activity through series of activity actions.

It describes the activity done by the user with the decision and the condition.

**RESULT:**

Thus the use-case diagram, class diagram and activity diagram for course registration system is drawn and verified successfully.

Exp No: 6(a) .

UML DIAGRAM FOR BLOOD DONOR SYSTEM

Date:

AIM:

To draw use-case diagram, class diagram and activity diagram for blood donor system using Umbrello.

PROBLEM STATEMENT:

In times of medical emergency, blood banks play a major role in deciding the fate of a person. The blood donor system helps the user to order specific blood type from the local blood reserves, while also enabling users to donate blood to the local blood reserves.

Login/Register:

- The user is a visiter, who visits the site and order blood of some types and pay the amount, also to donate blood
- Both customer and admin have different task in online system.

Admin:

Admin is a responsible person to run the whole system. Admin can add delete, update the blood stocks, as well as validate orders, receive payment etc

Admin scenarios:

- Login module
- Manage items
- Process a order
- Cancel the order
- Process return
- Online help

191911219

A.SIRISHA

Customer scenarios:

- Registration for new customer
- Login for existing customers
- Place a order
- Browser the item
- Change the password
- Cancel the order
- Make payment

Courier service scenarios are:

- The courier service has to provide by the deliver the blood to the patient.
- And also deliver the blood fast to the patient.

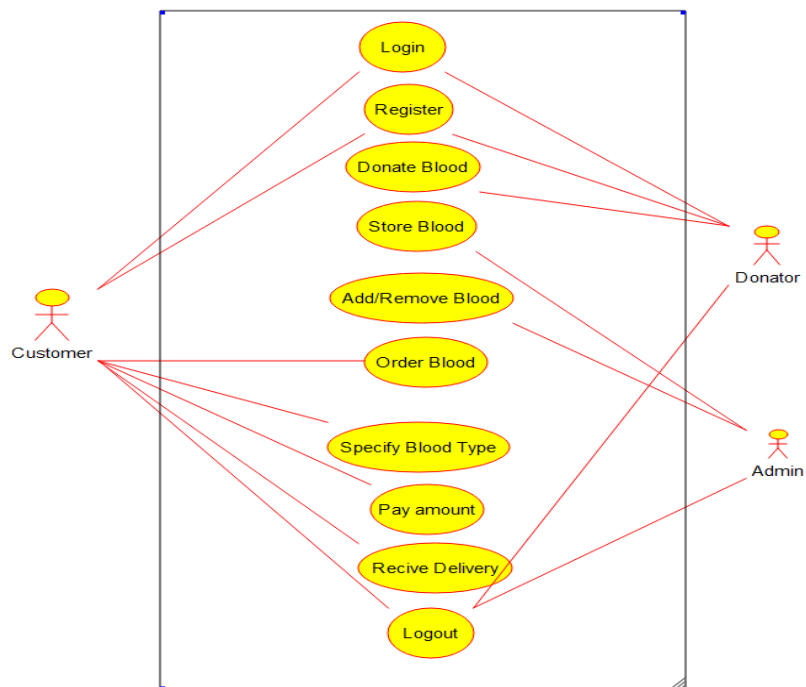
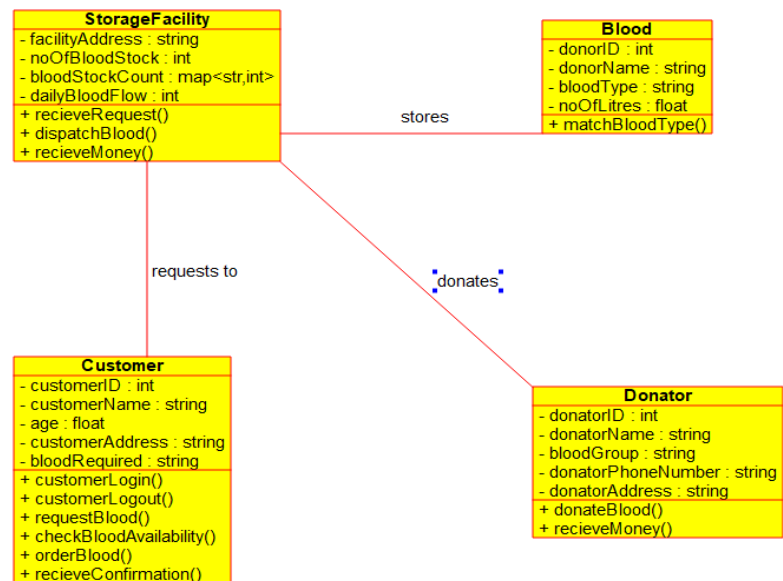
Payment:

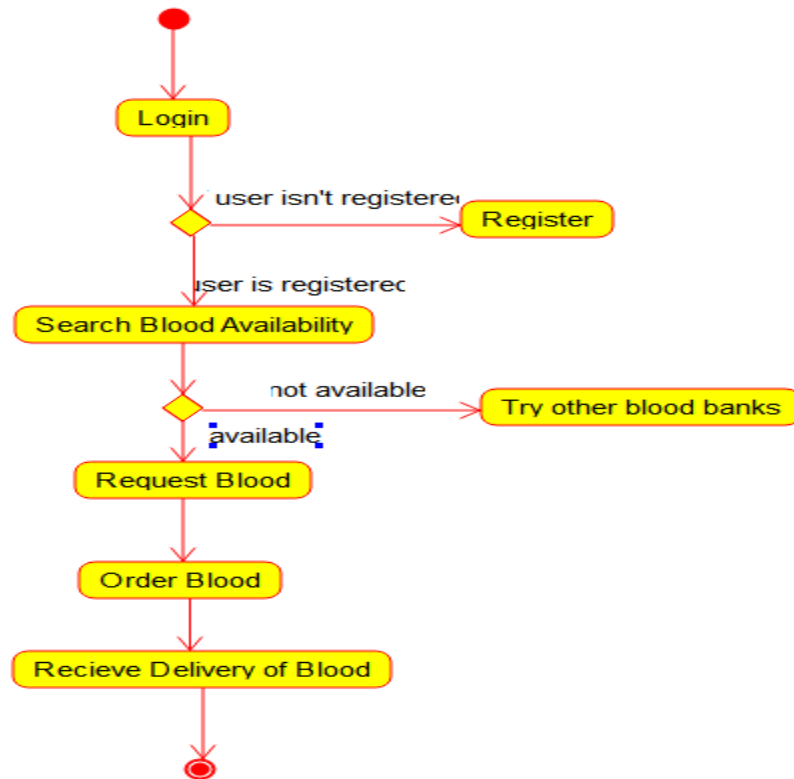
The customer choose the payment option when he was placed aorder.The payment is handled by the bank and the payment which could be done either by using credit card and debit card or cash on delivery.

Online help:

When the customers are needs a help the system will provide the online help system.

USE CASE DIAGRAM:

**CLASS DIAGRAM:**

ACTIVITY DIAGRAM:**RESULT:**

Thus the use-case diagram, class diagram and activity diagram for blood donor system is drawn and verified successfully.

Exp No: 6(b) UML DIAGRAM FOR HOSPITAL MANAGEMENT SYSTEM

Date:

AIM:

To draw use-case diagram, class diagram and activity diagram for Hospital management system using Umbrello.

PROBLEM STATEMENT:

Hospital Management system enables users to set appointments for visiting the doctors, check availability of vacancies in hospitals during times of emergencies, and find remedies for the symptoms and ailments of the patients through treatment by monitoring health conditions of the patient

Login/Register:

- A patient registers to set appointment to a specific doctor or a specific department of expertise, while admin and receptionists confirm appointments.
- Both customer, receptionists and admin have different task in online system.

Admin:

Admin is a responsible person to run the whole system. Admin can add delete, update all information, patient information, appointment information.

Admin scenarios:

- Login module
- Manage items
- Confirm Appointment
- Cancel the appointment
- Process return

191911219

A.SIRISHA

- Online help

Customer scenarios:

- Registration for new patient
- Login for existing patient
- Request Appointment
- Check Doctor's availability
- Change the password
- Cancel appointment
- Make payment

Customer service scenarios are:

- The customer service has to provide details of the hospital to the customers.

Payment:

The customer choose the payment option when he was placed aorder.The payment is handled by the bank and the payment which could be done either by using credit card and debit card or cash on delivery.

Online help:

When the customers are needs a help the system will provide the online help system.

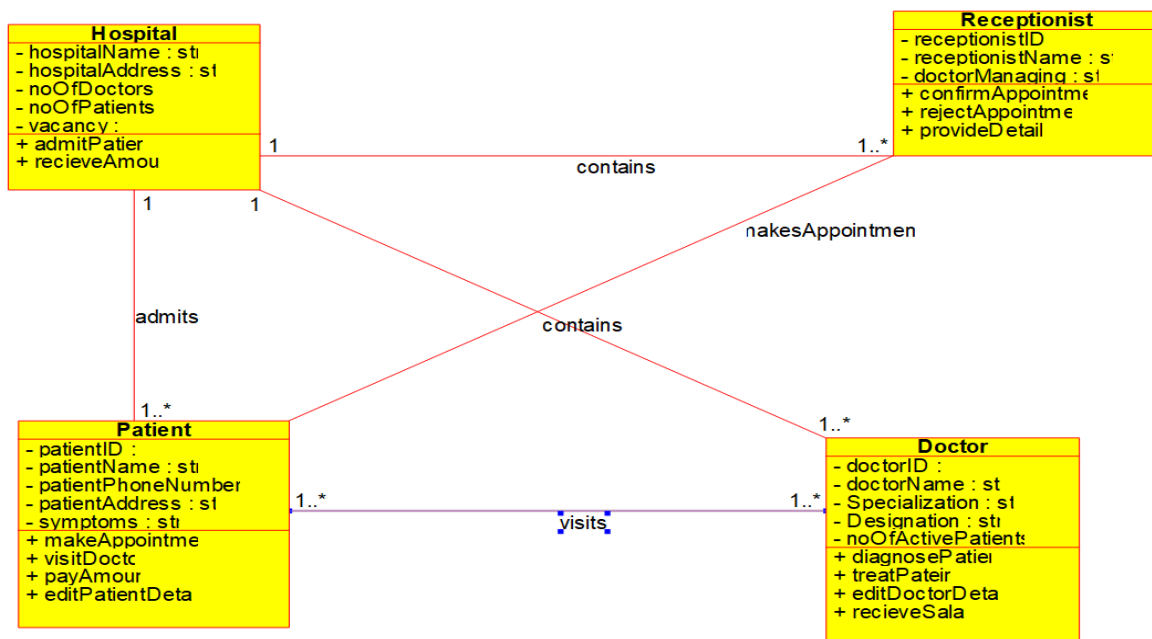
191911219

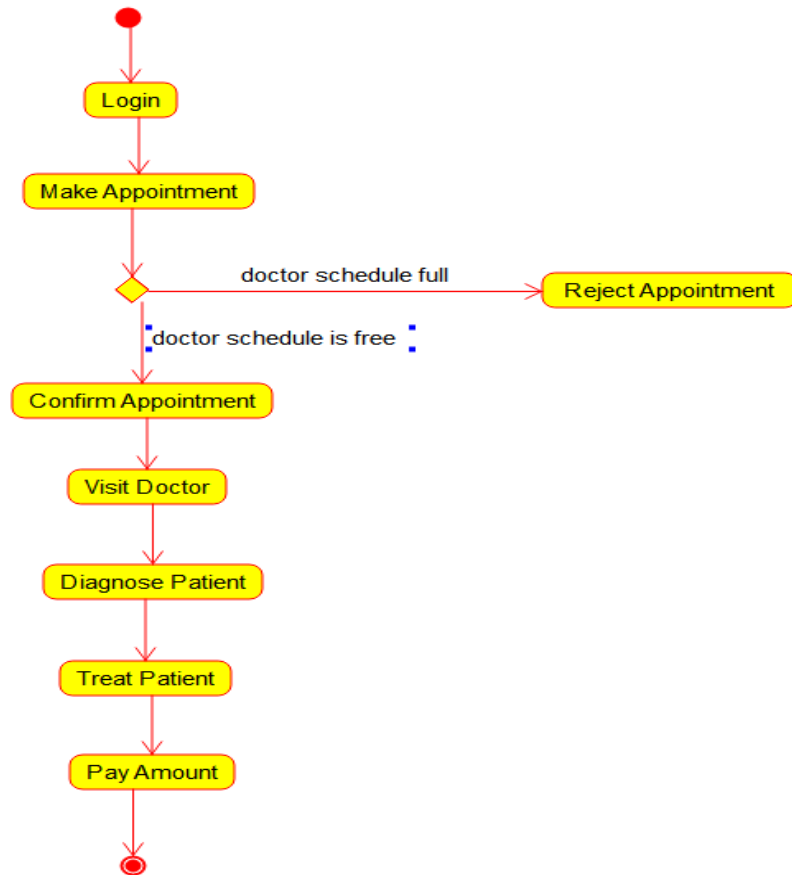
A.SIRISHA

USE CASE DIAGRAM:



CLASS DIAGRAM:



ACTIVITY DIAGRAM:**RESULT:**

Thus the use-case diagram, class diagram and activity diagram for hospital management system is drawn and verified successfully.

Exp No: 7(a) UML DIAGRAM FOR E-TICKETING SYSTEM

Date:

AIM:

To draw use-case diagram, class diagram and activity diagram for e-ticketing using Umbrello.

PROBLEM STATEMENT:

The E-Ticketing system will allow passengers to select trains for the time. In addition, each student will indicate two alternative choices in case a course offering becomes filled or cancelled. No course offering will have more than ten students. No course offering will have fewer than three students. A course offering with fewer than three students will be cancelled. Once the registration process is completed for a student, the registration system sends information to the billing system, so the student can be billed for the semester.

Clerks must be able to access the on-line system to indicate the number of passengers will be travelling in the train. They will also have the ability to hold, approve or cancel tickets of passengers

Login/Register:

- The user is a passenger, who visits the site to book train tickets and check ticket availability
- Both customer and clerk have different task in online system.

Clerk:

Clerk is a responsible person to run the whole system. Clerk can add delete, update the train details, as well as cancel and approve tickets etc.

Clerk scenarios:

- Login module
- Manage train details
- Process a ticket booking
- Cancel the ticket

191911219

A.SIRISHA

- Process return
- Online help

Customer scenarios:

- Registration for new passenger
- Login for existing passengers
- Book a ticket
- Cancel ticket
- Change the password
- Board the train
- Make payment

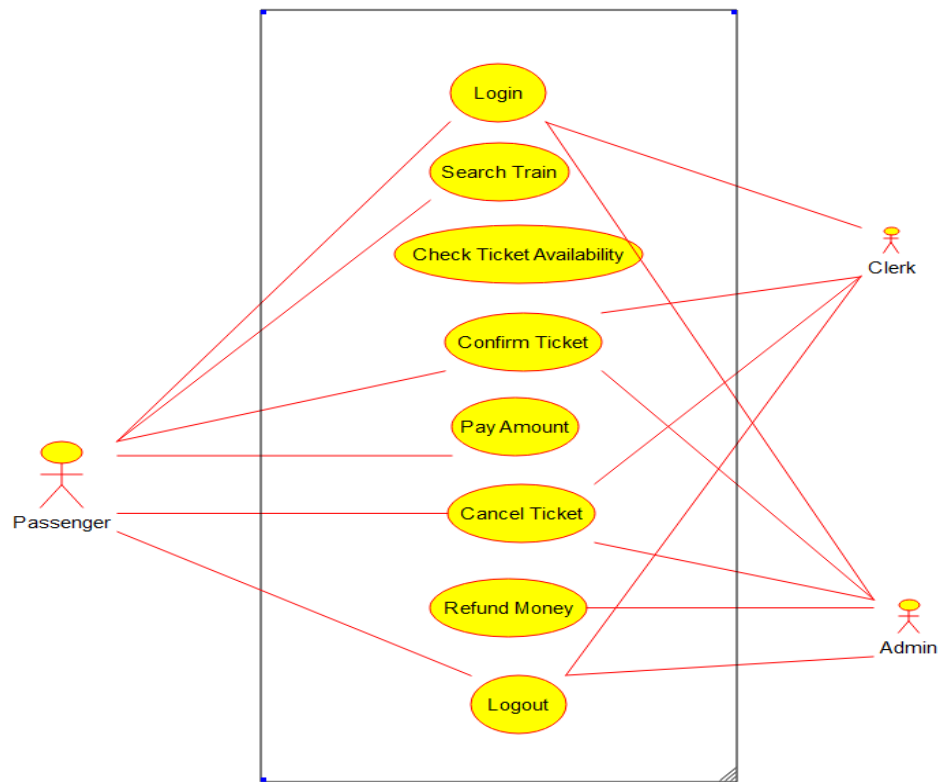
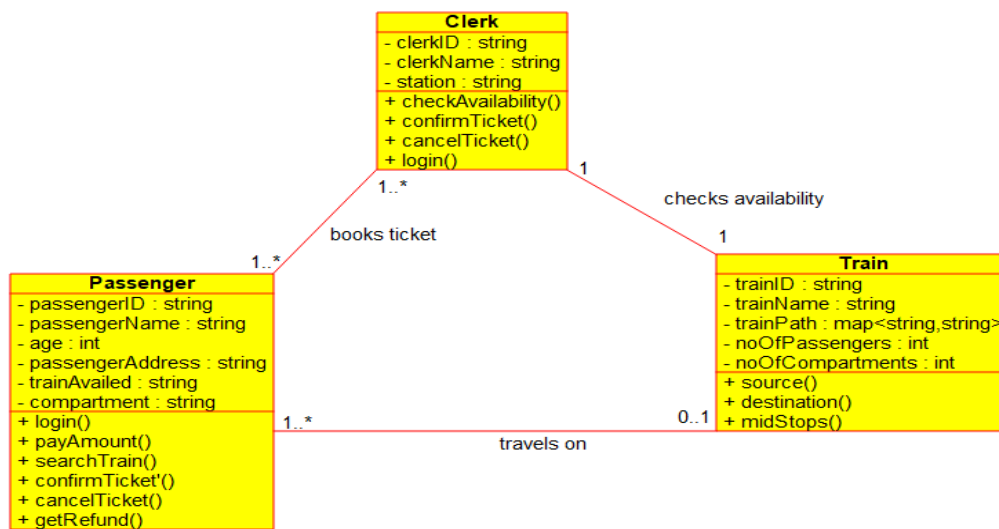
Payment:

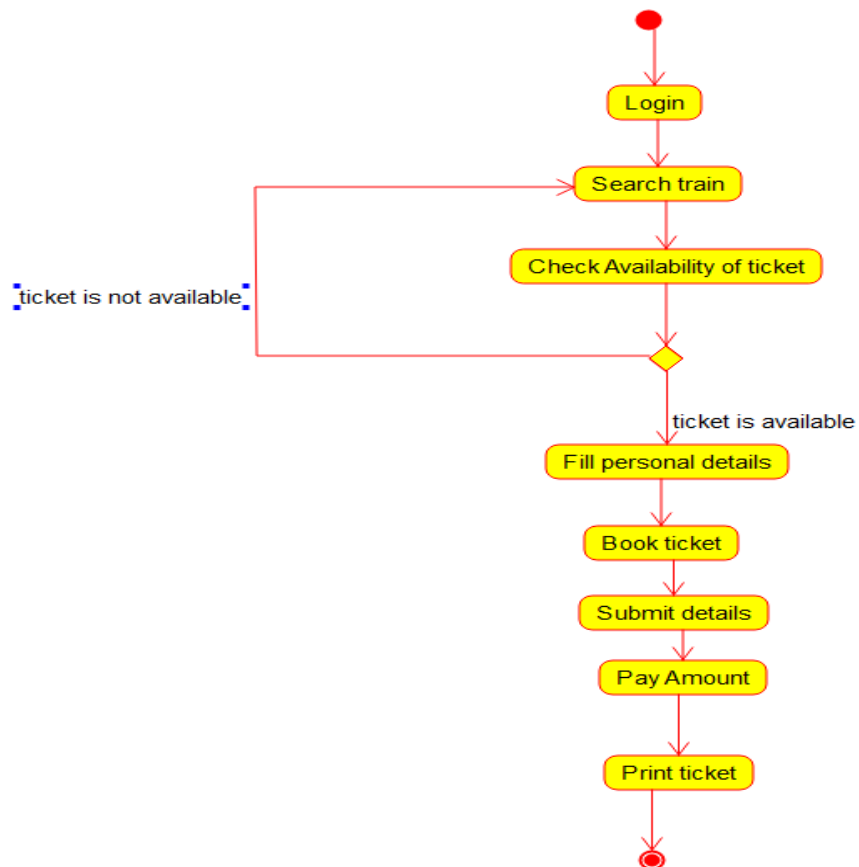
The customer choose the payment option when he was placed aorder.The payment is handled by the bank and the payment which could be done either by using credit card and debit card or cash on delivery.

Online help:

When the customers are needs a help the system will provide the online help system.

USE CASE DIAGRAM:

**CLASS DIAGRAM:**

ACTIVITY DIAGRAM:**RESULT:**

Thus the use-case diagram, class diagram and activity diagram for e-ticketing is drawn and verified successfully.

Exp No: 7(b) UML DIAGRAM FOR RESTAURANT MANAGEMENT SYSTEM

Date:

AIM:

To draw use-case diagram, class diagram and activity diagram for restaurant management system using Umbrello.

PROBLEM STATEMENT:

Restaurant Management system is used to make food delivery easier for delivery, ordering and managing orders in a restaurant. The restaurant management system can be accessed by the user through an application or website, the user must be able to search food, check availability, order food, pay amount and track the progress of food delivery.

Login/Register:

- The user is a visiter, who visits the site and orders food from their favourite restaurant and track the progress of the food through the application or website.
- Both customer and admin have different task in online system.

Admin:

Admin is a responsible person to run the whole system. Admin can add,delete, update the food and restaurant details, as well as validate food order and dispatch the food.

Admin scenarios:

- Login module
- Manage orders
- Process an order
- Cancel the order
- Process return

191911219

A.SIRISHA

- Online help

Customer scenarios:

- Registration for new customer
- Login for existing customers
- Place a order
- Browse the menu
- Change the password
- Cancel the order
- Make payment

Delivery service scenarios are:

- The delivery service has to provide by the deliver the food to the customer.
- And also deliver the food fast to the customer.

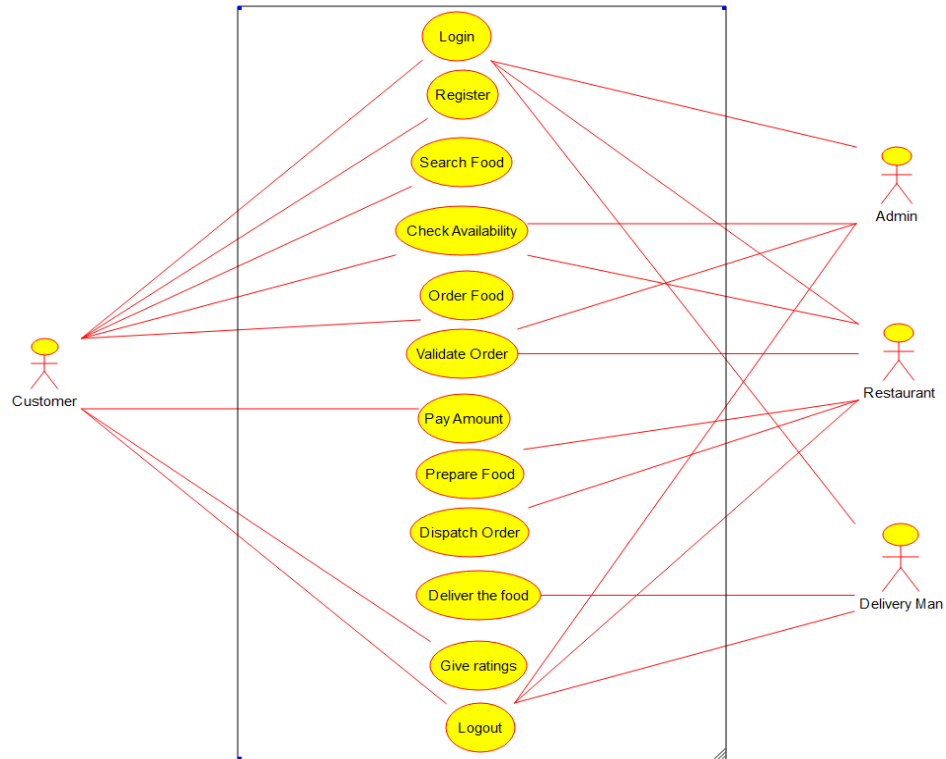
Payment:

The customer choose the payment option when he was placed aorder.The payment is handled by the bank and the payment which could be done either by using credit card and debit card or cash on delivery.

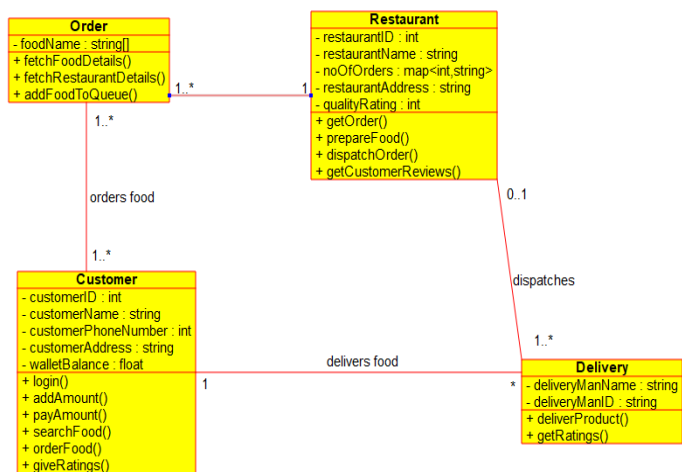
Online help:

When the customers are needs a help the system will provide the online help system.

USE CASE DIAGRAM:



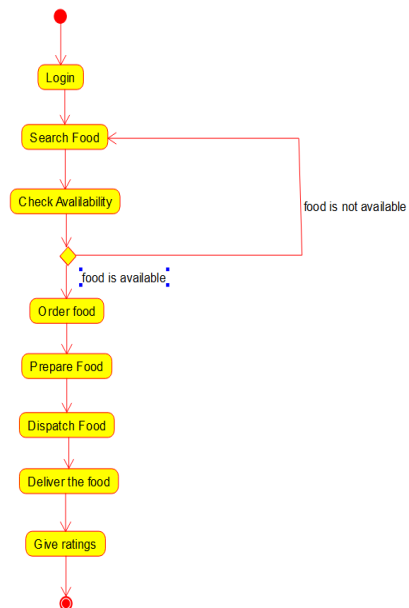
CLASS DIAGRAM:



191911219

A.SIRISHA

ACTIVITY DIAGRAM:



RESULT:

Thus the use-case diagram, class diagram and activity diagram for restaurant management system is drawn and verified successfully.

Exp No: 8(a). Using Raptor – Draw Flowchart To Display The Position Of The Numbers

Date:

AIM:

Draw a flowchart to display the position of the number and validate the process flow using RAPTOR.

STEPS INVOLVED:

STEP 1: A string of numbers is obtained from the user.

STEP 2: stringlength function is used to calculate the length of the string and is displayed.

STEP 3: A variable number is initiated and assigned the value 1.

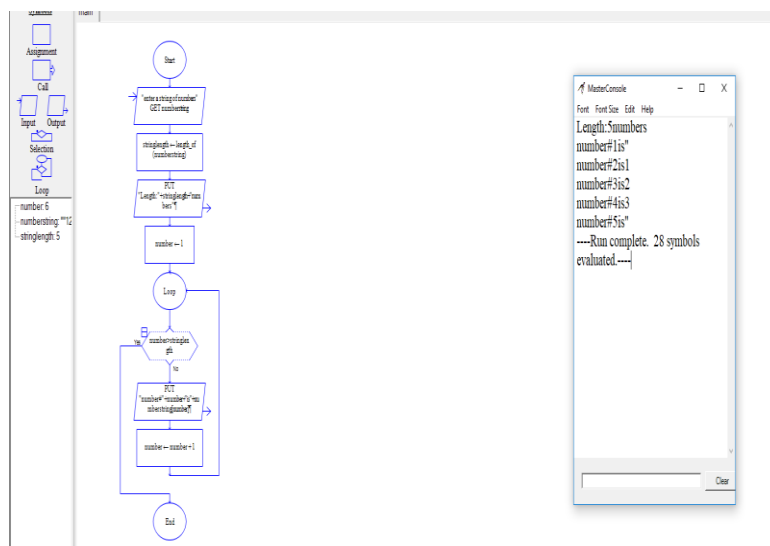
STEP 4: Loop is initiated, where the condition $\text{number} > \text{stringlength}$ is checked.

STEP 5: If above condition is true, each number of the array is displayed along with position with the help of “number” variable.

STEP 6: At the end of each turn, number is incremented by 1.

STEP 7: Once “number” > stringlength, the program ends.

FLOWCHART



RESULT:

The program is executed and verified successfully

Exp No: 8(b) Using Raptor - Draw And Validate Flowchart

- TO SWAP TWO CHARACTERS

Date:

AIM:

Draw a flowchart to swap two characters and validate the process flow using RAPTOR.

STEPS INVOLVED:

STEP 1: Three variables N1,N2, and TEMP are initiated and assigned the value 0.

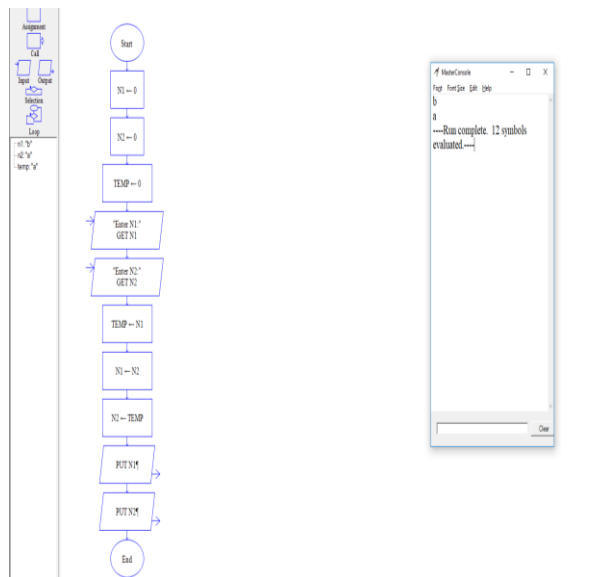
STEP 2: The user enters the value for N1 and N2.

STEP 3: TEMP variable is assigned the value of N1 and then N1 is assigned the value of N2.

STEP 4: N2 is now assigned the value of temp.

STEP 5: The exchanged values of N1 and N2 are returned.

FLOWCHART:



RESULT:

The program is executed and verified successfully

Exp NO: 9(a).

Using Raptor - Draw And Validate Flowchart**- TO FIND GIVEN NUMBER IS PRIME OR NOT****Date:****AIM:**

Draw a flowchart for finding given number is prime or not and validate the process flow using RAPTOR.

STEPS INVOLVED:

STEP 1: Get a number from the user

STEP 2: Assign value of 1 to variable b and 0 to variable A

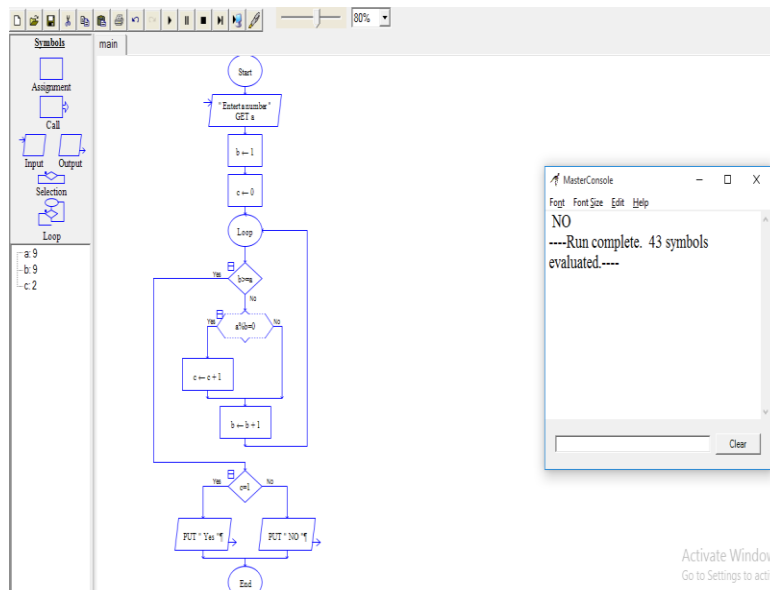
STEP 3: if the value of b is greater or equal to a, the loop is bypassed. If not the loop is initiated.

STEP 4: The value of b divided by a is checked with the value 0.

STEP 5: If $b\%a$ is equal to 0, then the value of c is incremented by 1 and b is incremented by 1.

STEP 6: If not, the value of b is incremented by 1 and the loop runs again.

STEP 7: If the value of c is equal to 1, entered value is a prime number. Else, it is not a prime number.

FLOWCHART:**RESULT:**

The program is executed and verified successfully

Exp No: 9(b). Using Raptor - Draw And Validate Flowchart
- TO CHECK GIVEN NUMBER IS PALINDROME OR NOT

Date:

AIM:

Draw a flowchart to check given number is palindrome or not and validate the process flow using RAPTOR.

STEPS INVOLVED:

STEP 1: A variable “palindrome” is initiated and assigned the value 0.

STEP 2: A number is obtained from the user and is assigned to “n”.

STEP 3: A variable m is created and assigned the value of n.

STEP 4: If n is less than or equal to 0, the program is terminated. Else, a loop is initiated.

STEP 5: The value of palindrome is multiplied by 10.

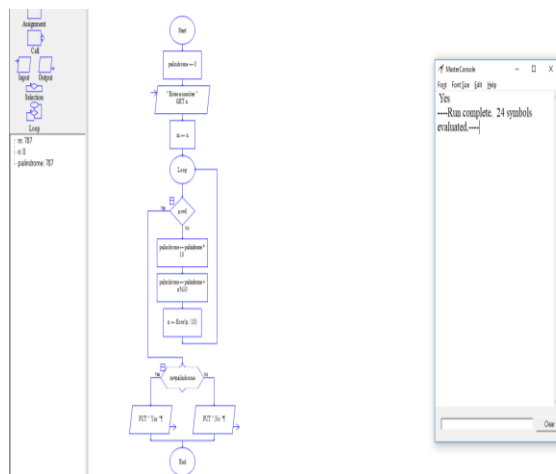
STEP 6: The new value of palindrome is added with the remained of the input number divided by

STEP 7: The value of n is then divided by 10 and the loop is executed until the value of n is less

STEP 8: A condition is then checked. If the value of palindrome and m are the same, then the number is a palindrome.

STEP 9: Program is terminated.

FLOWCHART:



RESULT:

The program is executed and verified successfully

Exp No: 10 Find Cyclomatic Complexity for a graph having number of edges as 12, number of nodes as 13 and number of predicate nodes in the flow graph as 5

Date:

AIM:

To write a C Program to compute the Cyclomatic Complexity.

Cyclomatic Complexity

Cyclomatic complexity is a source code complexity measurement that is being correlated to a number of coding errors. It is calculated by developing a Control Flow Graph of the code that measures the number of linearly-independent paths through a program module.

Lower the Program's cyclomatic complexity, lower the risk to modify and easier to understand. It can be represented using the below formula:

Cyclomatic complexity = $E - N + 2 * P$ where,

E = number of edges in the flow graph.

N = number of nodes in the flow graph.

P = number of nodes that have exit points.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
void main()
{
int E,N,P,CC;
clrscr();
printf("\n Program to find Cyclomatic Complexity:");
printf("\n Enter the number of Edges in the flow graph:");
scanf("%d",&E);
printf("\n Enter the number of Nodes in the flow graph:");
scanf("%d",&N);
printf("\n Enter the number of Predicate Nodes in the flow graph:");
```

191911219

A.SIRISHA

```
scanf("%d",&P);  
  
CC = E - N + (2*P);  
  
printf("\n The Cyclomatic Complexity of the flow graph is : %d", CC);  
  
getch();  
  
}
```

OUTPUT:

```
sse8@sse8-HP-ProDesk-400-G1-SFF:~$ cd Desktop  
sse8@sse8-HP-ProDesk-400-G1-SFF:~/Desktop$ cc cc.c  
sse8@sse8-HP-ProDesk-400-G1-SFF:~/Desktop$ ./a.out  
  
Program to find Cyclomatic Complexity:  
Enter the number of Edges in the flow graph:12  
  
Enter the number of Nodes in the flow graph:13  
  
Enter the number of Predicate Nodes in the flow graph:5  
  
The Cyclomatic Complexity of the flow graph is : 9sse8@sse8-HP-ProDesk-400-G1-SFF:~/Desktop$ █
```

RESULT:

The C Program to compute the Cyclomatic Complexity for the given Flow Graph has been written and verified.