ASSIGNMENT-1

Q1. State various phases of software development process.

Software life cycle models describe phases of the software cycle and the order in which those phases are executed. Each phase produces deliverables required by the next phase in the life cycle. Requirements are translated into design. Code is produced according to the design which is called development phase. After coding and development the testing verifies the deliverable of the implementation phase against requirements. The testing team follows Software Testing Life Cycle (STLC) which is similar to the development cycle followed by the development team.
There are following six phases in every Software development life cycle model:

1.Feasibility Study: - The main aim of feasibility study is to determine whether it would be financially and technically feasible to develop the product.

At first project managers or team leaders try to have a rough understanding of what is required to be done by visiting the client side. They study different input data to the system and output data to be produced by the system. They study what kind of processing is needed to be done on these data and they look at the various constraints on the behavior of the system.

After they have an overall understanding of the problem they investigate the different solutions that are possible. Then they examine each of the solutions in terms of what kind of resources required, what would be the cost of development and what would be the development time for each solution. Based on this analysis they pick the best solution and determine whether the solution is feasible financially and technically. They check whether the customer budget would meet the cost of the product and whether they have sufficient technical expertise in the area of development.

2.Requirements Analysis And Specification: - The aim of the requirements analysis and specification phase is to understand the exact requirements of the customer and to document them properly.

This phase consists of two distinct activities, namely

- **Requirements gathering and analysis:** The goal of the requirements gathering activity is to collect all relevant information from the customer regarding the product to be developed. This is done to clearly understand the customer requirements so that incompleteness and inconsistencies are removed. The requirements analysis activity is begun by collecting all relevant data regarding the product to be developed from the users of the product and from the customer through interviews and discussions.

- **Requirements specification:** During this activity, the user requirements are systematically organized into a Software Requirements Specification (SRS) document. The customer requirements identified during the requirements gathering and analysis activity are organized into a SRS document. The important components of this document are functional requirements, the nonfunctional requirements, and the goals of implementation.

3.Design: - The goal of the design phase is to transform the requirements specified in the SRS document into a structure that is suitable for implementation in some programming language. In technical terms, during the design phase the software architecture is derived from the SRS document.

Two distinctly different approaches are available:

- **Traditional design approach:** Traditional design consists of two different activities; first a structured analysis of the requirements specification is carried out where the detailed structure of the problem is examined. This is followed by a structured design activity. During structured design, the results of structured analysis are transformed into the software design.

- **Object-oriented design approach:** In this technique, various objects that occur in the problem domain and the solution domain are first identified, and the different relationships that exist among these objects are identified. The object structure is further refined to obtain the detailed design.

***4.Coding And Unit Testing*:-** The purpose of the coding and unit testing phase (sometimes called the implementation phase) of software development is to translate the software design into source code. Each component of the design is implemented as a program module.

The end-product of this phase is a set of program modules that have been individually tested. During this phase, each module is unit tested to determine the correct working of all the individual modules. It involves testing each module in isolation as this is the most efficient way to debug the errors identified at this stage.

***5.Integration And System Testing*: -** Integration of different modules is undertaken once they have been coded and unit tested. During the integration and system testing phase, the modules are integrated in a planned manner.

The different modules making up a software product are almost never integrated in one shot. Integration is normally carried out incrementally over a number of steps. During each integration step, the partially integrated system is tested and set of previously planned modules are added to it. Finally, when all the modules have been successfully integrated and tested, system testing is carried out. The goal of system testing is to ensure that the developed system conforms to its requirements laid out in the SRS document.

System testing usually consists of three different kinds of testing activities:

- **α – testing:** It is the system testing performed by the development team.
- **β – testing:** It is the system testing performed by a friendly set of customers.
- **Acceptance testing:** It is the system testing performed by the customer himself after the product delivery to determine whether to accept or reject the delivered product.

6.Maintenance: - Maintenance of a typical software product requires much more than the effort necessary to develop the product itself. Many studies carried out in the past confirm this and indicate that the relative effort of development of a typical software product to its maintenance effort is roughly in the 40:60 ratio.

Maintenance involves performing any one or more of the following three kinds of activities:

- Correcting errors that were not discovered during the product development phase. This is called corrective maintenance.

- Improving the implementation of the system, and enhancing the functionalities of the system according to the customer’s requirements. This is called perfective maintenance.

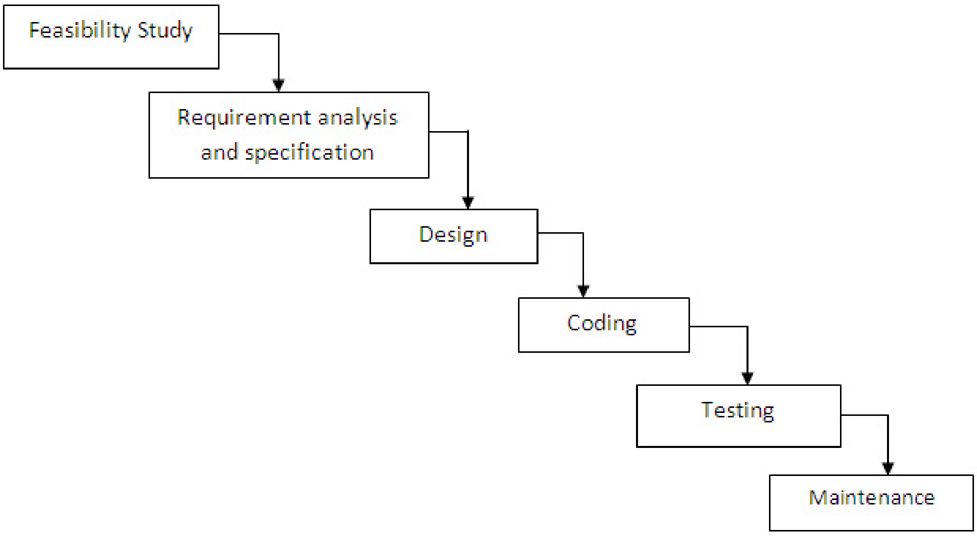
- Porting the software to work in a new environment. For example, porting may be required to get the software to work on a new computer platform or with a new operating system. This is called adaptive maintenance.

Q2.Write short note on the following models:

a)*Waterfall model:-*

The classical waterfall model is intuitively the most obvious way to develop software. Though the classical waterfall model is elegant and intuitively obvious, it is not a practical model in the sense that it can not be used in actual software development projects. Thus, this model can be considered to be a theoretical way of developing software. But all other life cycle models are essentially derived from the classical waterfall model. So, in order to be able to appreciate other life cycle models it is necessary to learn the classical waterfall model. Classical waterfall model divides the life cycle into the following phases.

- Feasibility Study
- Requirements Analysis and Specification
- Design
- Coding and Unit Testing
- Integration and System Testing
- Maintenance



The appropriate conditions for this model:

1. The requirements are well documented and have no ambiguity.
2. The product definition is stable.
3. All required resources are available.

Advantages:

1. Clear and easy to understand.
2. Clearly demarcated phases.

3. Easy to manage as each phase has clearly defined tasks.
4. Works well for short projects with clear requirements.

Disadvantages:

1. No overlapping of phases so people associated with a later stage sit idle until the previous step is completed.
2. Working software is produced at a very later stage.
3. Time required is high.
4. Cannot incorporate changes at any later stage.
5. Not suitable for models where requirements can change.

***b)<u>Prototype model</u>*:-**

Prototype: A prototype is a toy implementation of the system. A prototype usually exhibits limited functional capabilities, low reliability, and inefficient performance compared to the actual software. A prototype is usually built using several shortcuts. The shortcuts might involve using inefficient, inaccurate, or dummy functions. The shortcut implementation of a function, for example, may produce the desired results by using a table look-up instead of performing the actual computations. A prototype usually turns out to be a very crude version of the actual system.

Need for a prototype in software development: There are several uses of a prototype. An important purpose is to illustrate the input data formats, messages, reports, and the interactive dialogues to the customer. This is a valuable mechanism for gaining better understanding of the customer's needs:

- how the screens might look like
- how the user interface would behave
- how the system would produce outputs

Another reason for developing a prototype is that it is impossible to get the perfect product in the first attempt. Many researchers and engineers advocate that if you want to develop a good product you must plan to throw away the first version. The experience gained in developing the prototype can be used to develop the final product.

Applications: A prototyping model can be used when technical solutions are unclear to the development team. A developed prototype can help engineers to critically examine the technical issues associated with the product development. Often, major design decisions depend on issues like the response time of a hardware controller, or the efficiency of a sorting algorithm, etc. In such circumstances, a prototype may be the best or the only way to resolve the technical issues.

Examples for prototype model: A prototype of the actual product is preferred in situations such as:

- <u>Example 1: User requirements are not complete</u> In any application software like billing in a retail shop, accounting in a firm, etc the users of the software are not clear about the different functionalities required. Once they are provided with the prototype implementation, they can try to use it and find out the missing functionalities.
- <u>Example 2: Technical issues are not clear</u> Suppose a project involves writing a compiler and the development team has never written a compiler. In such a case, the team can consider a simple

language, try to build a compiler in order to check the issues that arise in the process and resolve them. After successfully building a small compiler (prototype), they would extend it to one that supports a complete language.

c)*<u>RAD model</u>:-*

The RAD (Rapid Application Development) model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product.

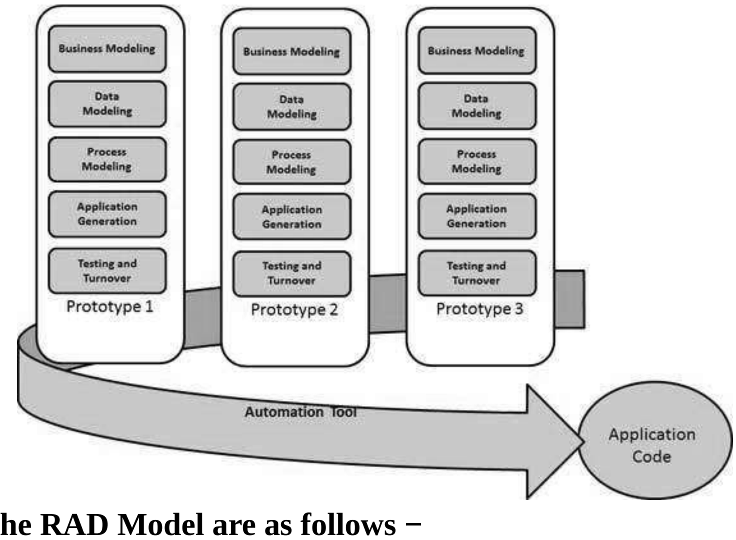
Rapid Application Development focuses on gathering customer requirements through workshops or focus groups, early testing of the prototypes by the customer using iterative concept, reuse of the existing prototypes (components), continuous integration and rapid delivery.

Rapid application development is a software development methodology that uses minimal planning in favor of rapid prototyping. A prototype is a working model that is functionally equivalent to a component of the product.

In the RAD model, the functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery. Since there is no detailed preplanning, it makes it easier to incorporate the changes within the development process.

RAD projects follow iterative and incremental model and have small teams comprising of developers, domain experts, customer representatives and other IT resources working progressively on their component or prototype.

The most important aspect for this model to be successful is to make sure that the prototypes developed are reusable.



The advantages of the RAD Model are as follows −

- Changing requirements can be accommodated.
- Progress can be measured.
- Iteration time can be short with use of powerful RAD tools.
- Productivity with fewer people in a short time.
- Reduced development time.
- Increases reusability of components.
- Quick initial reviews occur.

- Encourages customer feedback.
- Integration from very beginning solves a lot of integration issues.

The disadvantages of the RAD Model are as follows −

- Dependency on technically strong team members for identifying business requirements.
- Only system that can be modularized can be built using RAD.
- Requires highly skilled developers/designers.
- High dependency on modeling skills.
- Inapplicable to cheaper projects as cost of modeling and automated code generation is very high.
- Management complexity is more.
- Suitable for systems that are component based and scalable.
- Requires user involvement throughout the life cycle.
- Suitable for project requiring shorter development times.

d)*__Evolutionary model__*:-

Evolutionary software models are iterative. They are characterized in manner that enables the software engineers to develop increasingly more complete version of a software. That is, initially a rapid version of the product is being developed and then the product is developed to more accurate version with the help of the reviewers who review the product after each release and submit improvements. Specification, development and validation are interleaved rather than separate in evolutionary software process model.

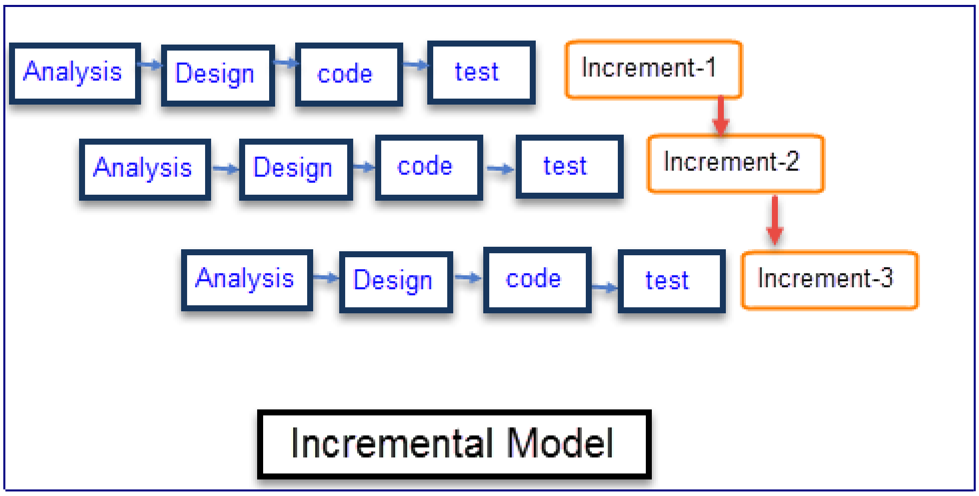
These models are applied because as the requirements often change so the end product will be unrealistic, where a complete version is impossible due to tight market deadlines it is better to introduce a limited version to meet the pressure. Thus the software engineers can follow a process model that has been explicitly designed to accommodate a product that gradually complete over time.

All the models suffer from the disadvantage that the duration of time from start of the project to the delivery time of a solution is very high. Evolutionary model attacks this problem in a slightly different approach. Evolutionary model suggests breaking down of work into smaller chunks, prioritizing them and then delivering those chunks to the customer one by one. The number of chunks is huge and is the number of deliveries made to the customer. The main advantage is that the customer's confidence increases as he constantly gets deliverables from the beginning of the project to verify and validate his requirements. The model allows for changing requirements as well as all work in broken down into maintainable work chunks.

e)*__Incremental model__*:-

Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance.

Each iteration passes through the **requirements, design, coding and testing phases**. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.



The system is put into production when the first increment is delivered. The first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments. Once the core product is analyzed by the client, there is plan development for the next increment.

Characteristics of an Incremental module includes

- System development is broken down into many mini development projects
- Partial systems are successively built to produce a final total system
- Highest priority requirement is tackled first
- Once the incremented portion id developed, requirements for that increment are frozen

Applications:

- Requirements of the system are clearly understood
- When demand for an early release of a product arises
- When software engineering team are not very well skilled or trained
- When high-risk features and goals are involved
- Such methodology is more in use for web application and product based companies

Advantages of Incremental Model:

- The software will be generated quickly during the software life cycle
- It is flexible and less expensive to change requirements and scope
- Thought the development stages changes can be done
- This model is less costly compared to others
- A customer can respond to each building
- Errors are easy to be identified

Disadvantages of Incremental Model:

- It requires a good planning designing
- Problems might cause due to system architecture as such not all requirements collected up front for the entire software lifecycle
- Each iteration phase is rigid and does not overlap each other

- Rectifying a problem in one unit requires correction in all the units and consumes a lot of time

f)*Spiral model*:-

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

- **Identification:** This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase.

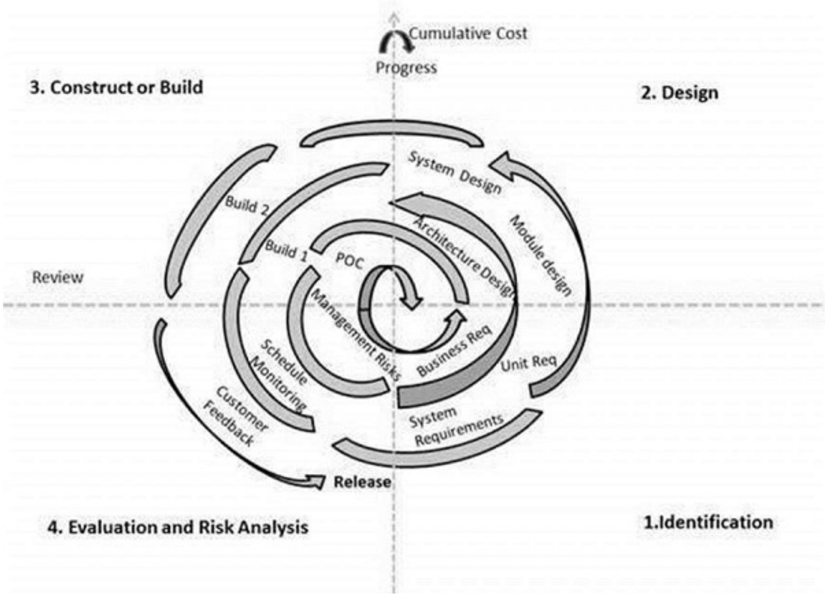
This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral, the product is deployed in the identified market.

- **Design:** The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.

- **Construct or Build:** The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.

- **Evaluation and Risk Analysis:** Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.



The advantages of the Spiral SDLC Model are as follows −

- Changing requirements can be accommodated.
- Allows extensive use of prototypes.
- Requirements can be captured more accurately.
- Users see the system early.
- Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

The disadvantages of the Spiral SDLC Model are as follows −

- Management is more complex.
- End of the project may not be known early.
- Not suitable for small or low risk projects and could be expensive for small projects.
- Process is complex
- Spiral may go on indefinitely.
- Large number of intermediate stages requires excessive documentation.

ASSIGNMENT-2
FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENT

Q1. Consider following case study and find out functional and non functional requirements

CASE STUDY

The SE VLabs Institute has been recently setup to provide state-of-the-art research facilities in the field of Software Engineering. Apart from research scholars (students) and professors, it also includes quite a large number of employees who work on different projects undertaken by the institution. 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

Identification of functional requirements:-

The above problem statement gives a brief description of the proposed system. From the above, even without doing any deep analysis, we might easily identify some of the basic functionality of the system:

- **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. On successful registration, a user ID and password would be provided to the member. He has to use this credentials for any future transaction in LIS.
- **Search book:** Any member of LIS can avail this facility to check whether any particular book is present in the institute's library. A book could be searched by its:
    - Title
    - Authors name
    - Publisher's name
- **User login:** A registered user of LIS can login to the system by providing his employee ID and password as set by him while registering. After successful login, "Home" page for the user is shown from where he can access the different functionalities of LIS: search book, issue book, return book, reissue book. Any employee ID not registered with LIS cannot access the "Home" page -- a login failure message would be shown to him, and the login dialog would appear again. This same thing happens when any registered user types in his password wrong.
- **Issue book:** Any member of LIS can issue a book against his account provided that:
    - The book is available in the library i.e. could be found by searching for it in LIS
    - No other member has currently issued the book
    - Current user has not issued the maximum number of books that can

  If the above conditions are met, the book is issued to the member.
  Note that this FR would remain incomplete if the "maximum number of books that can be issued to a member" is not defined. We assume that this number has been set to four for students and research scholars, and to ten for professors.
  Once a book has been successfully issued, the user account is updated to reflect the same.
- **Return book:** A book is issued for a finite time, which we assume to be a period of 20 days. That is, a book once issued should be returned within the next 20 days by the corresponding member of LIS. After successful return of a book, the user account is updated to reflect the same.
- **Reissue book:** Any member who has issued a book might find that his requirement is not over by 20 days. In that case, he might choose to reissue the book, and get the permission to keep it for another 20 days. However, a member can reissue any book at most twice, after

which he has to return it. Once a book has been successfully reissued, the user account is updated to reflect the information.

Identification of non-functional requirements

Having talked about functional requirements, let's try to identify a few non-functional requirements.

- **Performance Requirements:**
  - This system should remain accessible 24x7
  - At least 50 users should be able to access the system altogether at any given time
- **Security Requirements:**
  - This system should be accessible only within the institute LAN
  - The database of LIS should not store any password in plain text -- a hashed value has to be stored
- **Software Quality Attributes**
- **Database Requirements**
- **Design Constraints:**
  - The LIS has to be developed as a web application, which should work with Firefox 5, Internet Explorer 8, Google Chrome 12, Opera 10
  - The system should be developed using HTML 5

Q2.Note down all functional and non- functional requirements on following :

EXAM REGISTRATION SYSTEM

Purpose:

The developed product is a web-based examination registration system. It assists an examiner to create an exam by adding and modifying questions, supplying possible results of a question and selecting the intended group of students. Moreover the examination will have an overview of the results of an exam by student, group or other factors. A student can solve exams for which they are enrolled and view results and corrections of previous exams. The product benefits the examiner by simplifying and speeding up the work of creating and correcting an exam and provides an easy way to follow the results of a student or group of students. An administrator will setup and maintain the enrolments and courses. The product will be entirely accessed by a user-friendly web-interface which means that a minimum amount of computer knowledge is required to interact with it.

Functional Requirements:

a)Authenticate

Primary Actor: Student/Faculty

Pre-Condition: Nil

Post-Condition: Fetch and load the details of the user

Main Scenario:

1. Start the application. User prompted for login and password.

2. User gives the login and password.
3. System does authentication.
4. Selection screen is displayed.

Alternate Scenario:

5(a). Authentication fails
5(a)1. Prompt user about wrong password
5(a)2. Allow user to re-enter the password.

b)Registration

Primary Actor: Student/Faculty
Pre-Condition: Student/Faculty selected the desired course.
Post-Condition: Store details in database.
Main Scenario:

1. Student/Faculty fills in the other details
2. Registers for the selected course.

Alternate Scenario:

3(a).Registration fails.
3(a)1. System asks the user to re-enter the details.
4(a). Connection gets terminated.
4(a)1.System redirects the user to Login or Selection Page.

c)Take Test

Primary Actor: Student, System
Pre-Condition: Student enrolled in that course
Post-Condition: Store details in database
Main Scenario:

1. Questions are selected randomly and given to the student
2. Student submit the answers
3. System evaluates the answers
4. System displays result of the student

Alternate Scenario:

None

d)Feed Questions

Primary Actor: Faculty, System
Pre-Condition: Faculty selected the desired course.
Post-Condition: Store details in database.
Main Scenario:

1. Faculty Provides the questions and answers to the questions
2. System stores the details in database

Alternate Scenario:

None

<u>Nonfunctional Requirements:-</u>

a)Performance Requirements

Performance requirements define acceptable response times for system functionality.

1. The load time for user interface screens shall take no longer than two seconds.
2. The log in information shall be verified within two seconds.
3. Queries shall return results within five seconds.

b)Security and Safety Requirements

Security: The files in which the information regarding securities and portfolios should be secured against malicious deformations.
Fault Tolerance: Data should not become corrupted in case of system crash or power failure.

c)Business Rules

Server Administrator: under extreme circumstances the administrator has the privileges to back up the data's but can't modify the contents.
User: has the rights to claim the payment made if the ticket is under waiting list on the date of the journey but 10% will be deducted. The user will be able to book for just 6 persons at a time.

BOOK BANK SYSTEM

Purpose:

A Book Bank lends books and magazines to member, who is registered in the system. Also it handles the purchase of new titles for the Book Bank. Popular titles are brought into multiple copies. Old books and magazines are removed when they are out or date or poor in condition. A member can reserve a book or magazine that is not currently available in the book bank, so that when it is returne d or purchased by the book bank, that person is notified. The book bank can easily create, replace and delete information about the tiles, members, loans and reservations from the system.

Functional Requirements:

a)Search book details

The librarian initiates this use case when any member returns or request the book and checking if the book is available.

Precondition: The librarian should enter all Book details .

Normal Flow: Build message for librarian who search the book.

Post Condition: Send message to respective member who reserved the book.

b) Book issue

Initiated by librarian when any member wants to borrow the desired book. If the book is available, the book is issued.

Precondition: Member should be valid member of library.

Normal Flow: Selected book will be issued to the member.

Alternative Flow: If book is not available then reserved book use case should be initiate.

Post Condition: Update the catalogue.

c)Book order

Initiated by librarian when the requested book is not available in the library at that moment. The book is reserved for the future and issued to the person when it is available.

Precondition: Initiated only when book is not available.

Normal Flow: It reserved the book if requested.

Post Condition : Mention the entry in catalogue for reservation

d)Book return

Invoked by the librarian when a member returns the book.

Precondition: Member should be valid member of library.

Normal Flow: Librarian enters bookid and system checks for return date of the book.

Alternative Flow: System checks for return date and if it returned late fine message shows.

Post Condition: Check the status of reservation.

e)Book entry

It is used when new books or magazines are added to the library.

Precondition: Not available or more copies are required.

Normal Flow: Enter bookid,author information, publication information, purchased date, prize and number of copies.

Post Condition: Update the information in catalogue.

<u>Nonfunctional Requirements:-</u>

a)Performance Requirements

Performance requirements define acceptable response times for system functionality.

a) The system should be as much user friendly as possible.
b) This system should remain accessible for 14 hours per day .
c) At least 120 users should be able to access the system altogether at any given time

b)Security and Safety Requirements

a) This system should be accessible only within the institute LAN

b) The database should not store any password in plain text.

c)Design Constraints:

a) Front End Client : The Student and Librarian online interface is built using Visual studio.

b) Back End : The software should use Oracle 11 g database.

Comparison of Various SDLC Models

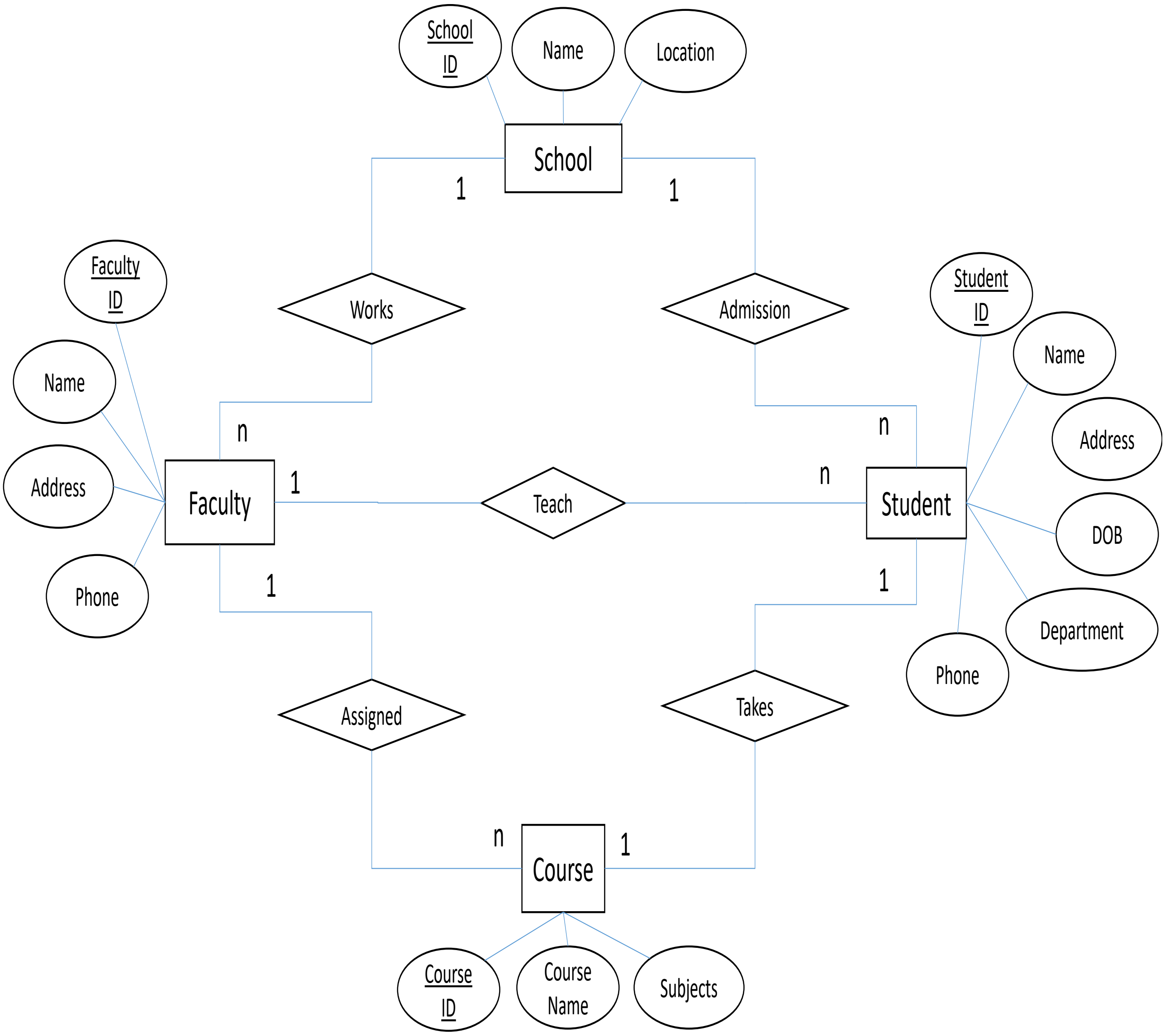
Properties of Model	***Water-Fall Model***	***Incremental Model***	***Spiral Model***	***Rad Model***
Planning in early stage	Yes	Yes	Yes	No
Returning to an earlier phase	No	Yes	Yes	Yes
Handle Large-Project	Not Appropriate	Not Appropriate	Appropriate	Not Appropriate
Detailed Documentation	Necessary	Yes but not much	Yes	Limited
Cost	Low	Low	Expensive	Low
Requirement Specifications	Beginning	Beginning	Beginning	Time boxed release
Flexibility to change	Difficult	Easy	Easy	Easy
User Involvement	Only at beginning	Intermediate	High	Only at the beginning
Maintenance	Least	Promotes Maintainability	Typical	Easily Maintained
Duration	Long	Very long	Long	Short
Risk Involvement	High	Low	Medium to high risk	Low
Framework Type	Linear	Linear + Iterative	Linear + Iterative	Linear
Testing	After completion of coding phase	After every iteration	At the end of the engineering phase	After completion of coding
Overlapping Phases	No	Yes (As parallel development is there)	No	Yes
Maintenance	Least Maintainable	Maintainable	Yes	Easily Maintainable
Re-usability	Least possible	To some extent	To some extent	Yes
Time-Frame	Very Long	Long	Long	Short
Working software availability	At the end of the life-cycle	At the end of every iteration	At the end of every iteration	At the end of the life cycle
Objective	High Assurance	Rapid Development	High Assurance	Rapid development
Team size	Large Team	Not Large Team	Large Team	Small Team
Customer control over administrator	Very Low	Yes	Yes	Yes

ASSIGNMENT-3

Q. Draw an ER diagram for the following problem:

(1) You have been asked to implement a database for a school management system (SMS). This primarily consists of maintaining students' information like name, address, date of birth, roll number, deparment, and so on. Details about the school to be stored includes school's name, location. Although it is unlikely that there would be two schools at the same place with same name, but our SMS would like to accomodate this possibility.

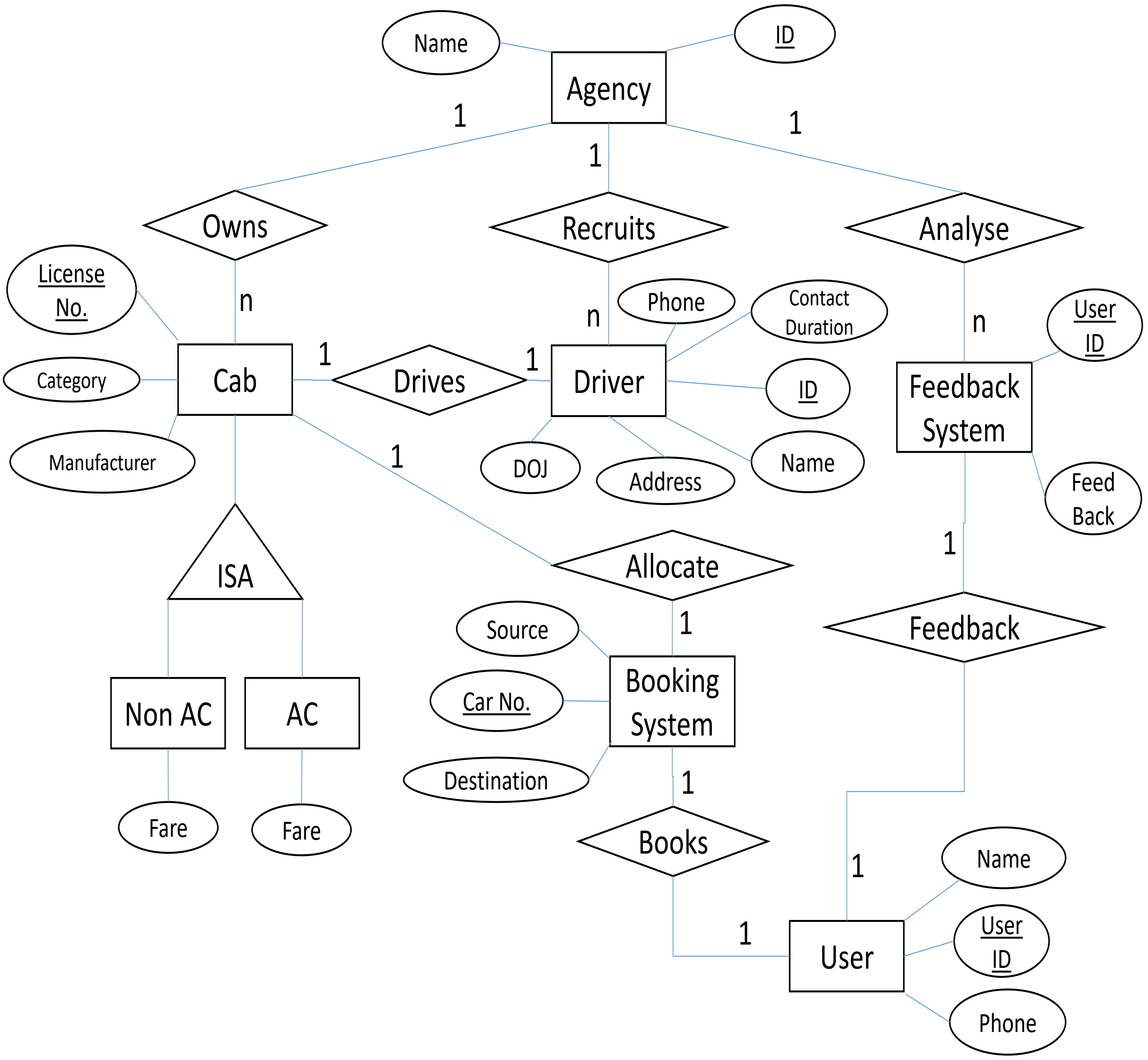
Faculty members works in the school. They teaches the students. A faculty member normally teaches multiple students at a time. Also, he can teach multiple courses to the students.



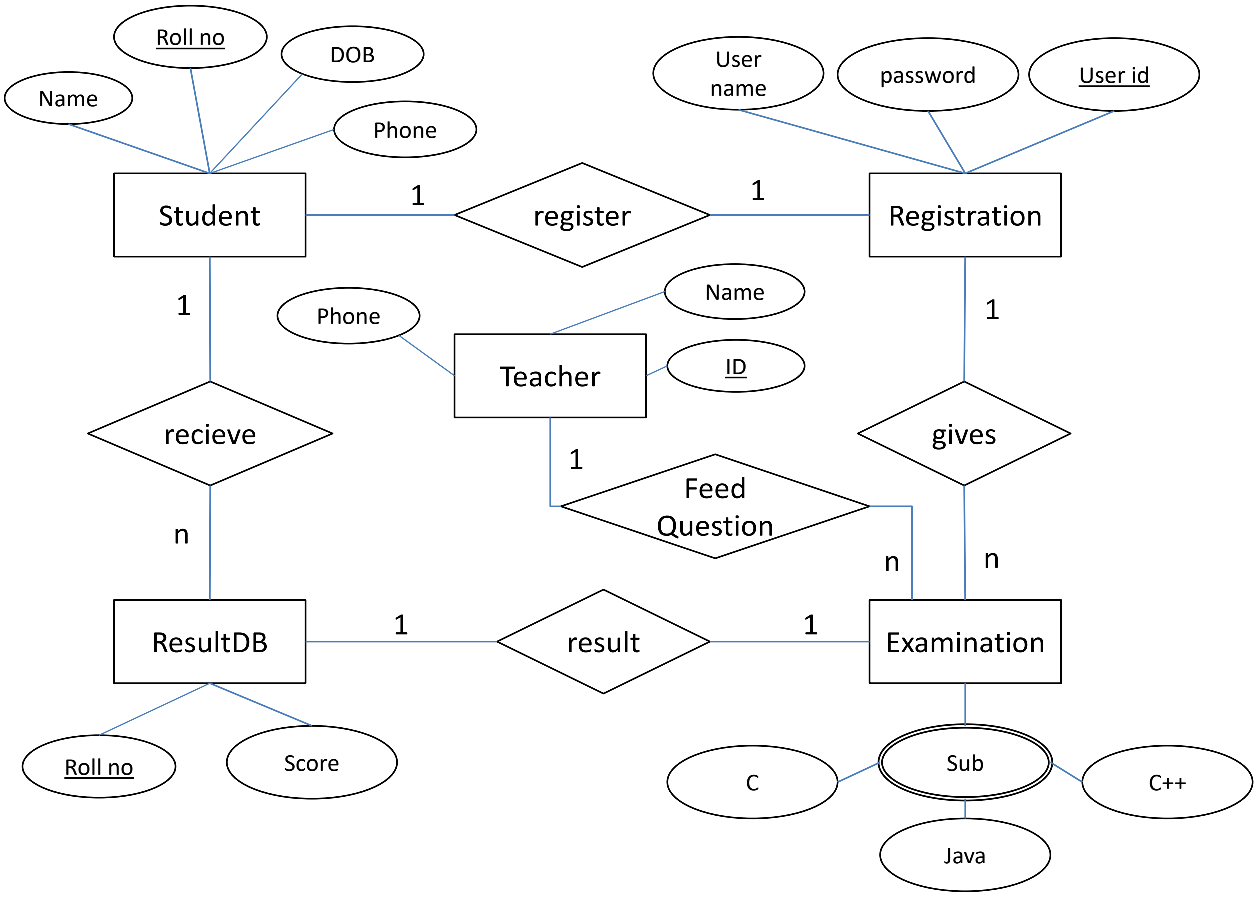
(2) The latest cab services agency in the city has approached you to develop a Cab Management System for them. They would be using this software to efficiently manage and track different cabs that are operated by them.

Cabs are solely owned by the agency. They hire people in contracts to drive the cabs. A cab can be uniquely identified by, like any other vehicle in the country, its license plate. A few different categories of cars are available from different manufacturers. And a few of them are AC cars.Cab drivers are given a identification card while joining. The ID card contains his name, permanent address, phone number, date of joining, duration of contract. Also, an unique alphanumeric code is assigned to each number.The agency provides service from 8 AM to 8 PM. Whenever any passenger books a cab, an available cab is allocated for him. The booking receipt given to the passenger contains the car #, source and destination places. Once he reaches the destination, he signs on a duplicate copy of the receipt and gives back to the driver. Driver must submit this duplicate copy signed by the passenger at the agency for confirmation.

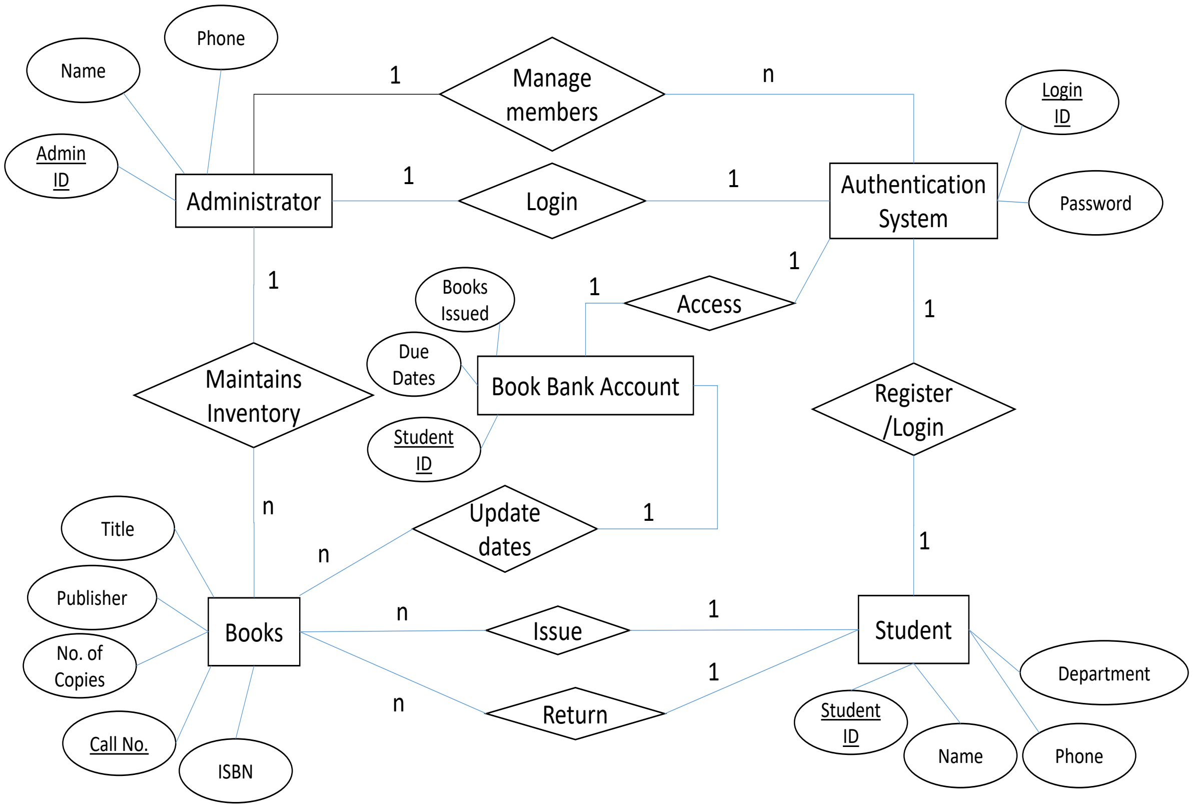
To evaluate their quality of service, the agency also wants a (optional) customer satisfaction survey, where passengers can provide feedback about their journey through the agency's website.



(3) Exam Registration System



(4) Book Bank System



ASSIGNMENT-4
Estimation of Project Metrics

Q1. Identify the unique operators and operands from following snippet of code:

```
int main (int argc, char** argv)
{
        int x=10;
        int y= 20;
        int sum;
        sum= x+y;
        printf("Sum of %d and %d is: %d\n", x, y, sum);
        return 0;
}
```

- Unique Operators: **int, main, (), {}, printf, char, **, =, +, ",", ";", *return***
- Unique Operands: **argc, argv, x, 10, y, 20, sum, x+y, *"Sum of %d and %d is: %d\n", 0***

Q2. The Absolute Beginners Inc. is again at your door! This time their demand is, however, simple. They have a C program, which computes the area of a circle (code shown below). They want it to be rewritten in Java.

```
int main (int argc,char** argv){
        int radius=12.34;
        printf("Area of the circle with radius %f is: %f\n", radius, area(radius));
        return 0;
}
float area(float r){
        return  22*r* r/7;
}
```

Using Halstead's metrics estimate the effort required to recreate this program.

- Unique Operators: **int, float, main, (), {}, printf, char, **, =, *, /, ",", ";", *return, area***
- Unique Operands: **argc, argv, radius, 12.34,*"Area of the circle with radius %f is: %f\n", 22*r*r/7, 0***

Hence,
Total number of Unique Operators : n1 = 15
Total number of Unique Operands : n2 = 7

Total number of Operators : N1 = 26
Total number of Operands : N2 = 9

Program Volume = $N*\log_2(n) = (26+9)*\log_2(15+7) = 35*\log_2(22)$
Programming Difficulty = (n1/2)*(N2/n2) = (15/2)*(9/7) = 135/14

Programming Effort = Difficulty*Volume = $(135/14) * 35*\log_2(22)$ = **1505.06**

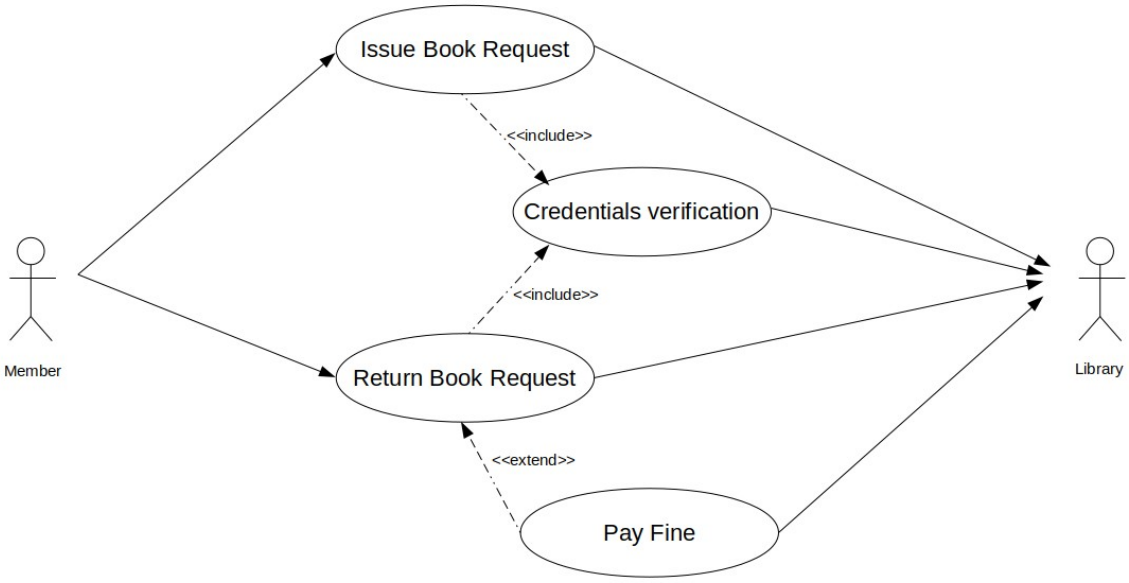
Conslusion:

Effort required to rewrite the given program is 1505.06

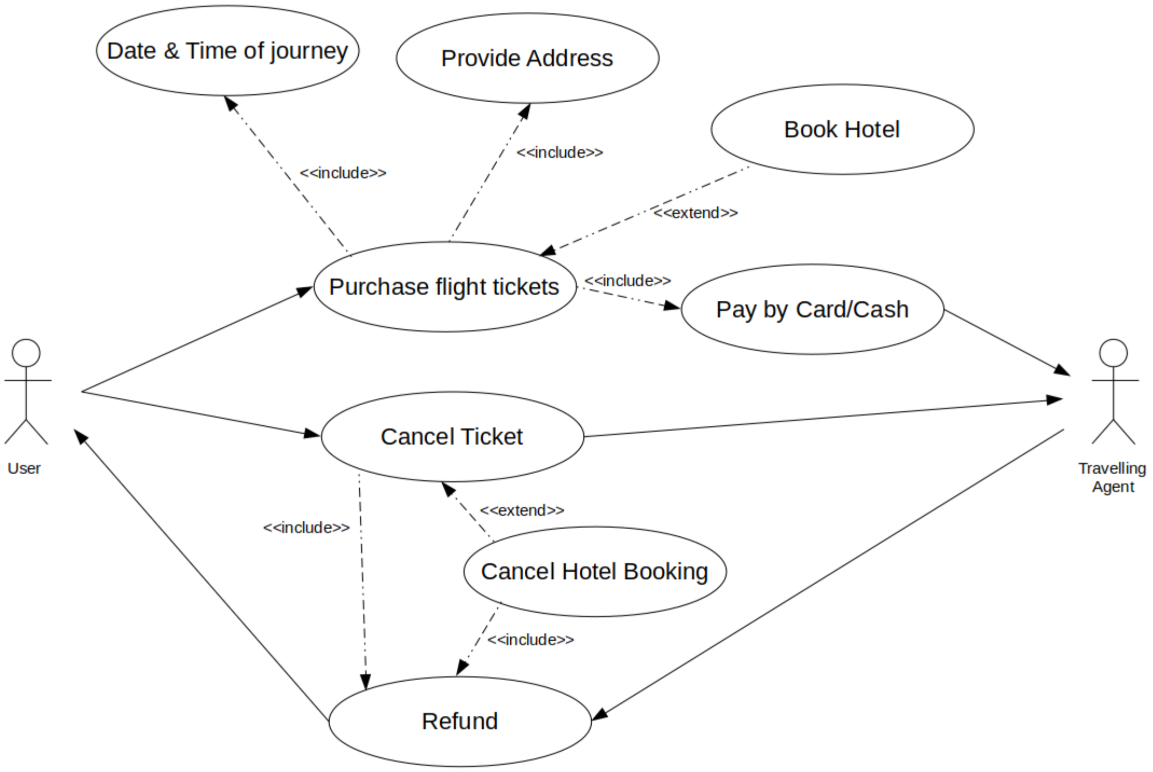
ASSIGNMENT-5
USE CASE DIAGRAM

Q. Draw use case diagram for the following problem:

1.*Consider a library, where a member can perform two operations: issue book and return it. A book is issued to a member only after verifying his credentials.*



2.*Consider your neighboring travel agent from whom you can purchase flight tickets. To book a ticket you need to provide details about your journey i.e. on which date and at what time you would like to travel. You also need to provide your address. You can pay either by cash or by card. You can also cancel a booked ticket later if you decide to change your plan. In that case you need to book a new ticket again. Your agent also allows you to book a hotel along with flight ticket. While canceling a flight ticket you can also cancel hotel booking. Appropriate refund as per policy is made in case of cancellation.*



ASSIGNMENT - 6

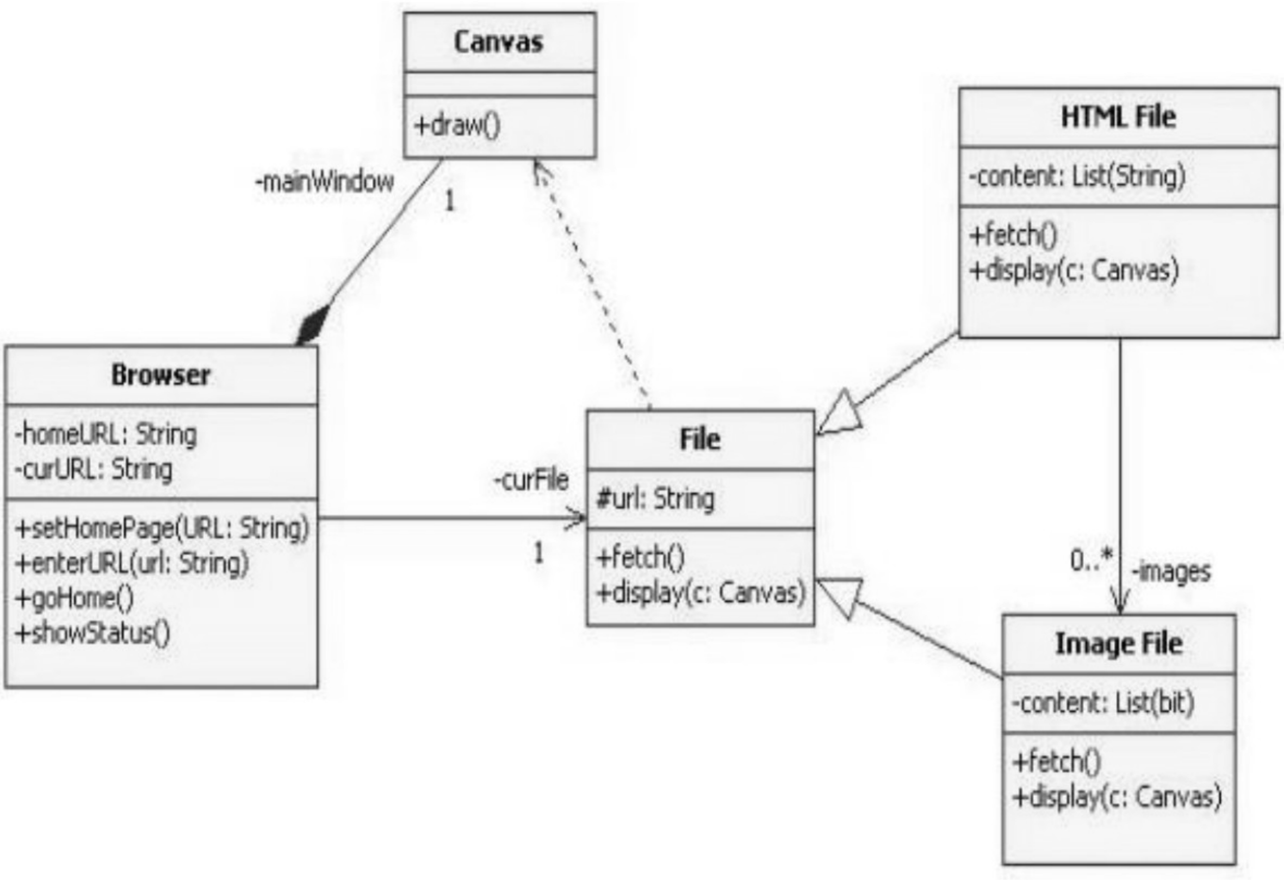
UML CLASS Diagram

1. Draw a UML Class diagram for the following problem

A web browser is a software that helps us access a resource (web page) available on the World Wide Web and identified by a URL. A web browser consists of different sub-components, which can be primarily categorized into browser rendering engine, and browser control.

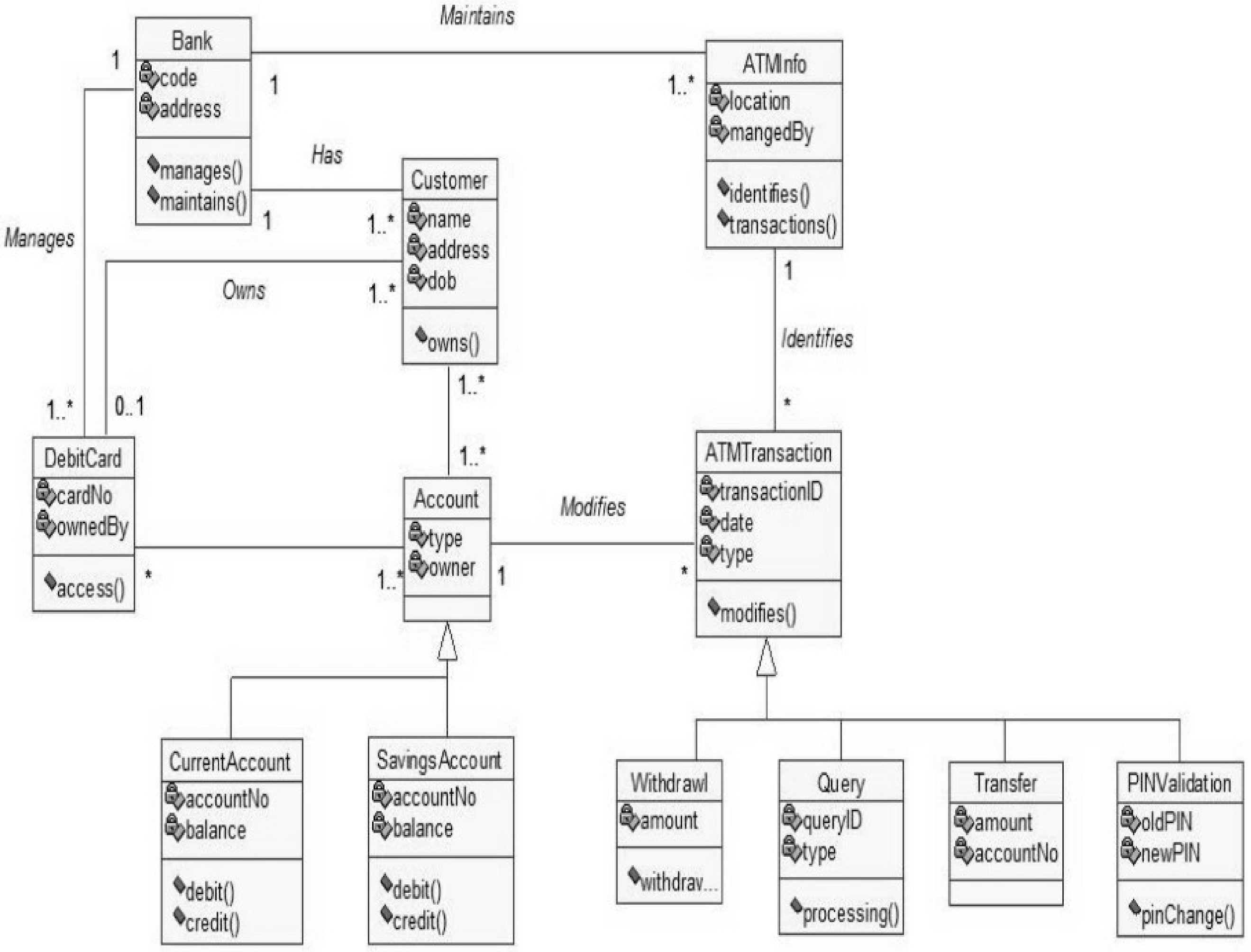
The rendering engine is responsible for displaying a requested page in the web browser. The rendering engine itself is quite a complex piece of software, which knows how to display a web page based on the HTML elements present in the page, and CSS rules defined (if any). Today browsers are not only limited to displaying text and images, but can provide access to audio and video components also.

The web browser control, too, consists of several sub-components including navigation, window control, event handlers, page display. The navigation control aids users to request for web pages (resources) by specifying a URL, navigate to other resources through internal and external hyperlinks, move across pages visited earlier. Event handlers are responsible to identify the kind of activity that user is trying to do, and perform it. For example, when a user clicks on a hyperlink, event handlers identify the URL of the target resource, and delegates loading of the resource to other components. A resource that has been retrieved by the web browser is then displayed in its page display area. Window control, in association with the rendering engine, helps in controlling various aspects of page display like changing font-size, resolution, and so on, apart from resizing or closing the window.



2. Draw a UML Class diagram for the following problem

An automated teller machine (ATM) lets customers to withdraw cash anytime from anywhere without requiring involvement of any banking clerk or representative. Customer must insert his ATM card into he machine and authenticate himself by typing in his personal identification number (PIN). He cannot avail any of the facilities if the PIN entered is wrong. Authenticated customers can also change their PIN. They can deposit cash to their account with the bank. Also they can transfer funds to any other account. The ATM also provides options to the user to pay electricity or phone bill. Every day morning the stock of cash in the ATM machine is replenished by a representative from the bank. Also, if the machine stops working, then it is fixed by a maintenance guy.



ASSIGNMENT – 7

UML Sequence Diagram

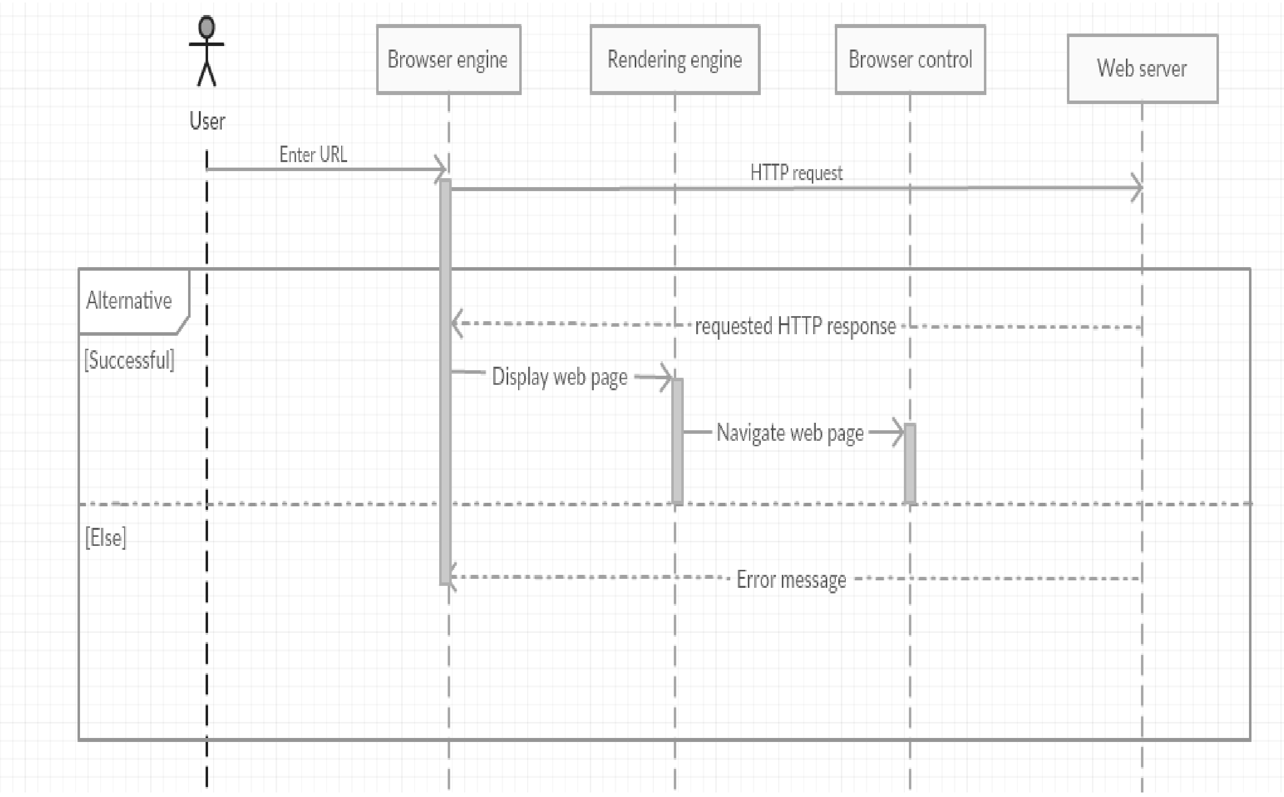
1. Draw a UML Sequence diagram for the following problem

A web browser is a software that helps to access a resource (web page) available on the World Wide Web and identified by a URL. Whenever a user types in the URL of a web page in the browser's address bar and clicks the "Go" button, the browser sends a HTTP request to the concerned web server. If the requested resource is available and accessible, the web server sends back a HTTP response to the requesting web browser. In case of any error, a HTTP response is sent indicating the error.

When the web browser receives a HTTP response, it displays the web page to the user. In very simple terms a web browser can be thought of consisting the following sub-components: rendering engine, and browser control.

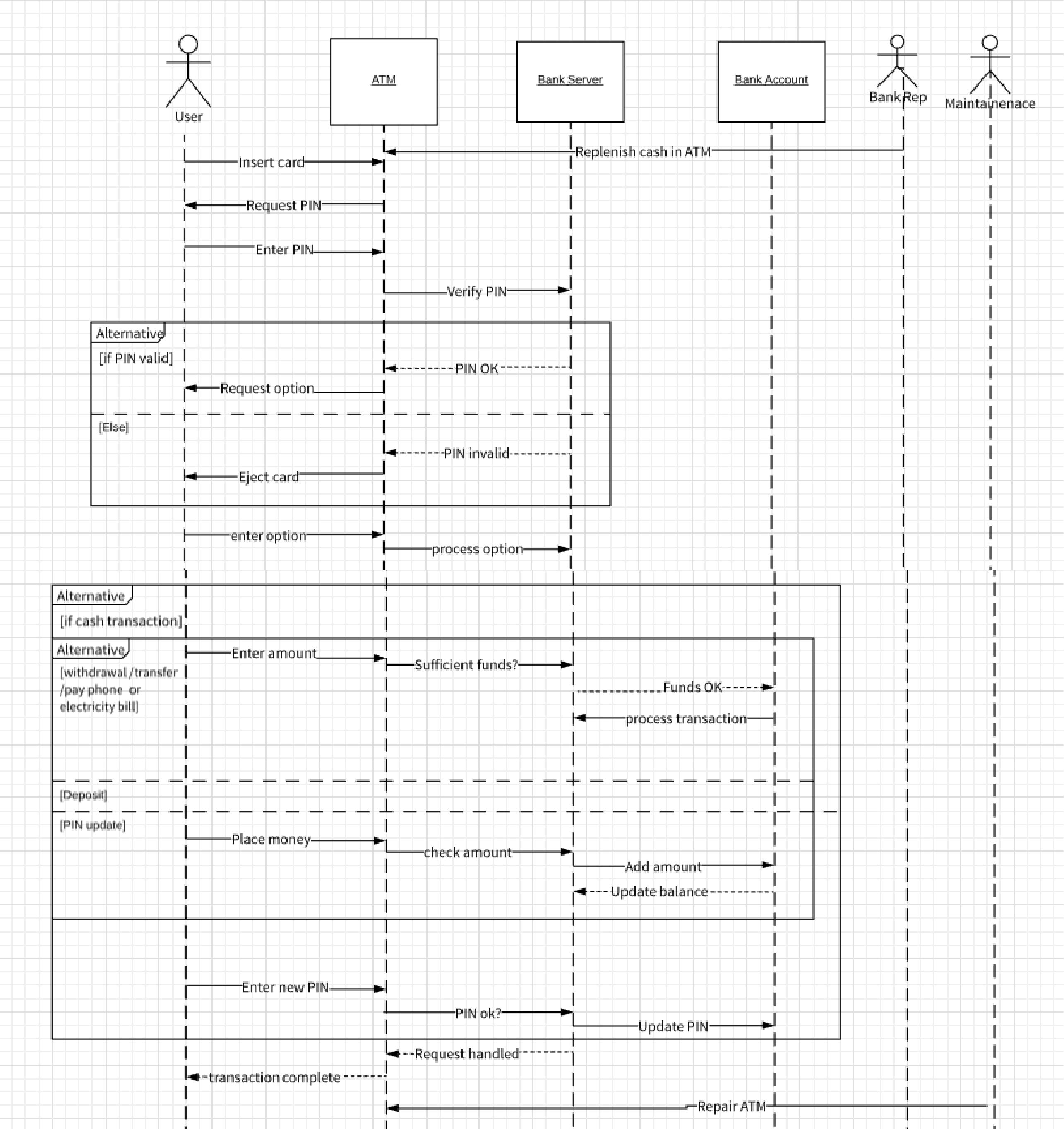
Once a HTTP response has been obtained from the server, the rendering engine decides the layout of the contents and actually displays the requested page. This is done keeping in mind the different HTML elements that are present in the page, and corresponding CSS rules, if any.

The browser control provides facilities like navigating across pages (by following hyperlinks), reload a page, and handles other events related to the window display, for example, resizing the browser window.



2. **Draw a UML Sequence diagram for the following problem**

An automated teller machine (ATM) lets customers to withdraw cash anytime from anywhere without requiring involvement of any banking clerk or representative. Customer must insert his ATM card into he machine and authenticate himself by typing in his personal identification number (PIN). He cannot avail any of the facilities if the PIN entered is wrong. Authenticated customers can also change their PIN. They can deposit cash to their account with the bank. Also they can transfer funds to any other account. The ATM also provides options to the user to pay electricity or phone bill. Every day morning the stock of cash in the ATM machine is replenished by a representative from the bank. Also, if the machine stops working, then it is fixed by a maintenance guy.

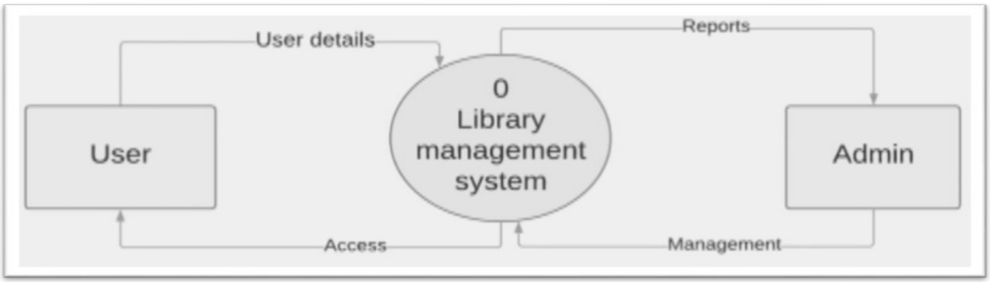


ASSIGNMENT-8

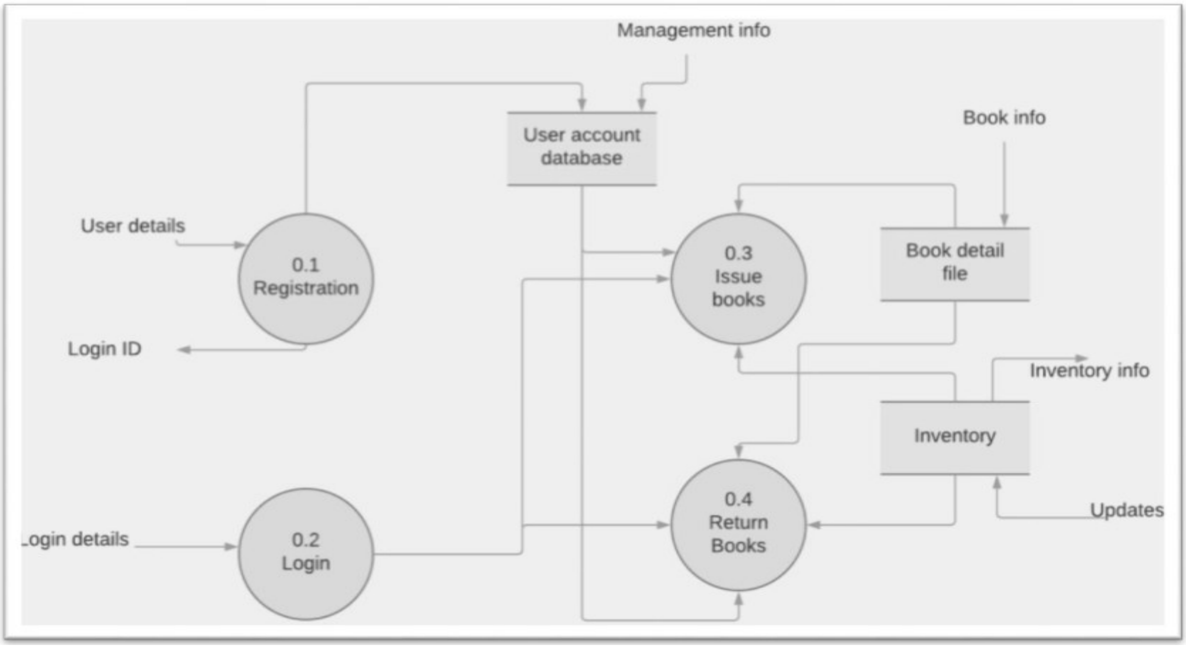
Data Flow Diagram

Library Management System Data Flow Diagram

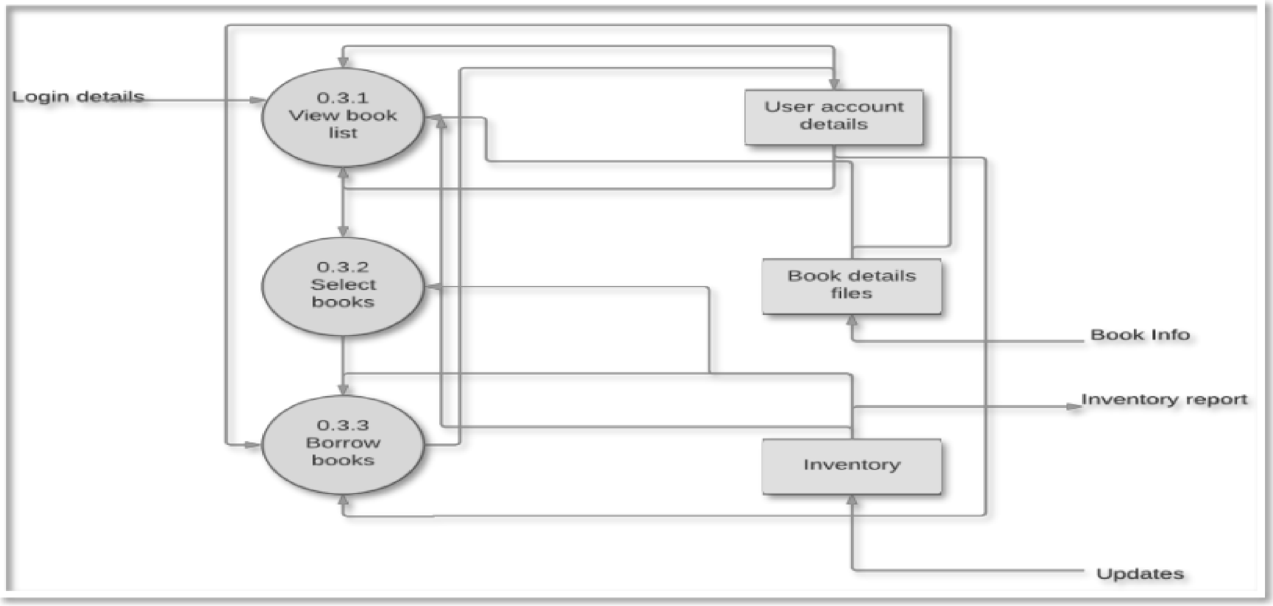
LEVEL 0:



LEVEL 1:



LEVEL 2 for process 0.3:



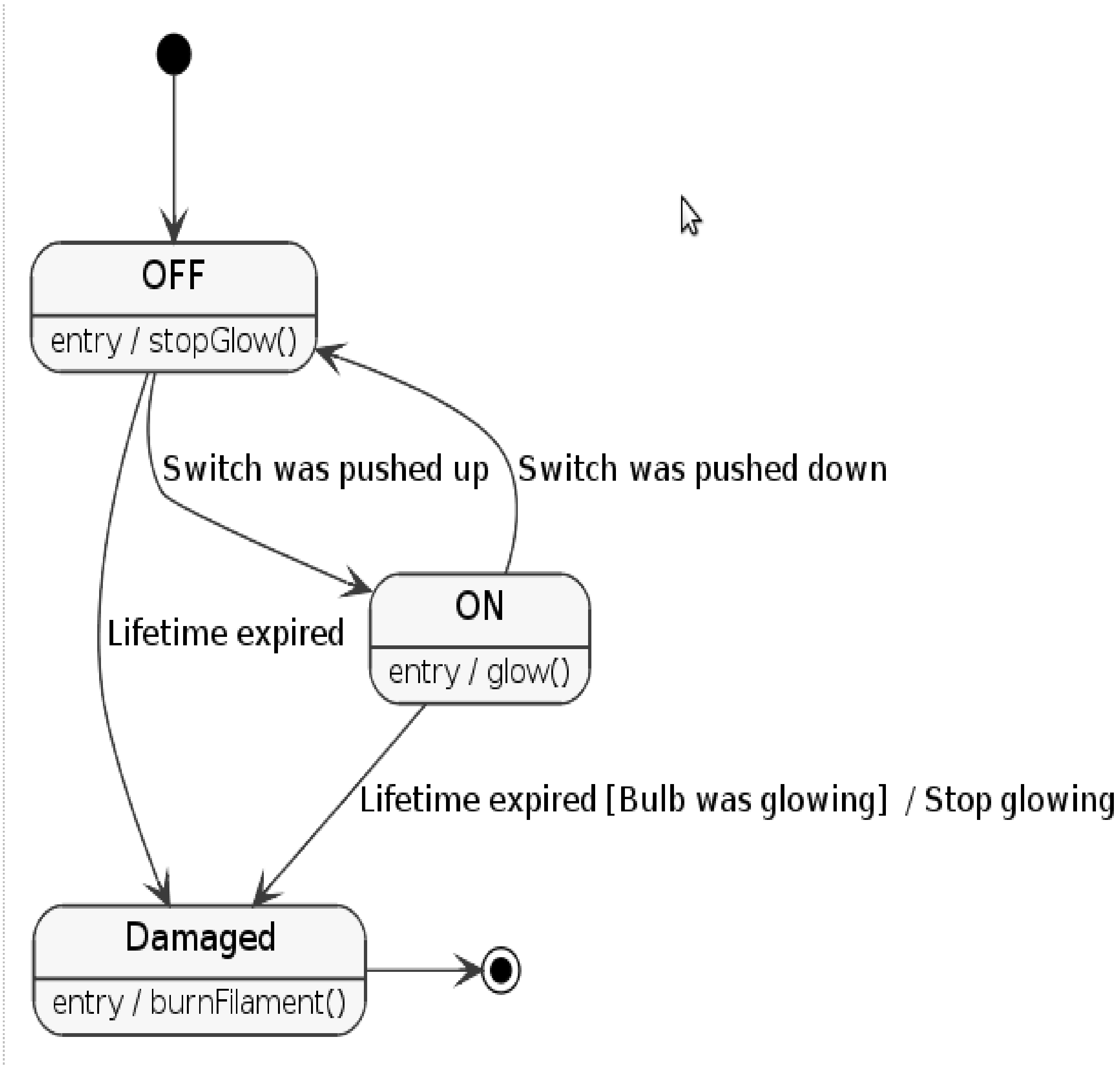
<u>ASSIGNMENT-9</u>

State Chart and Activity Diagram

1. Draw a statechart diagram to graphically represent the following system

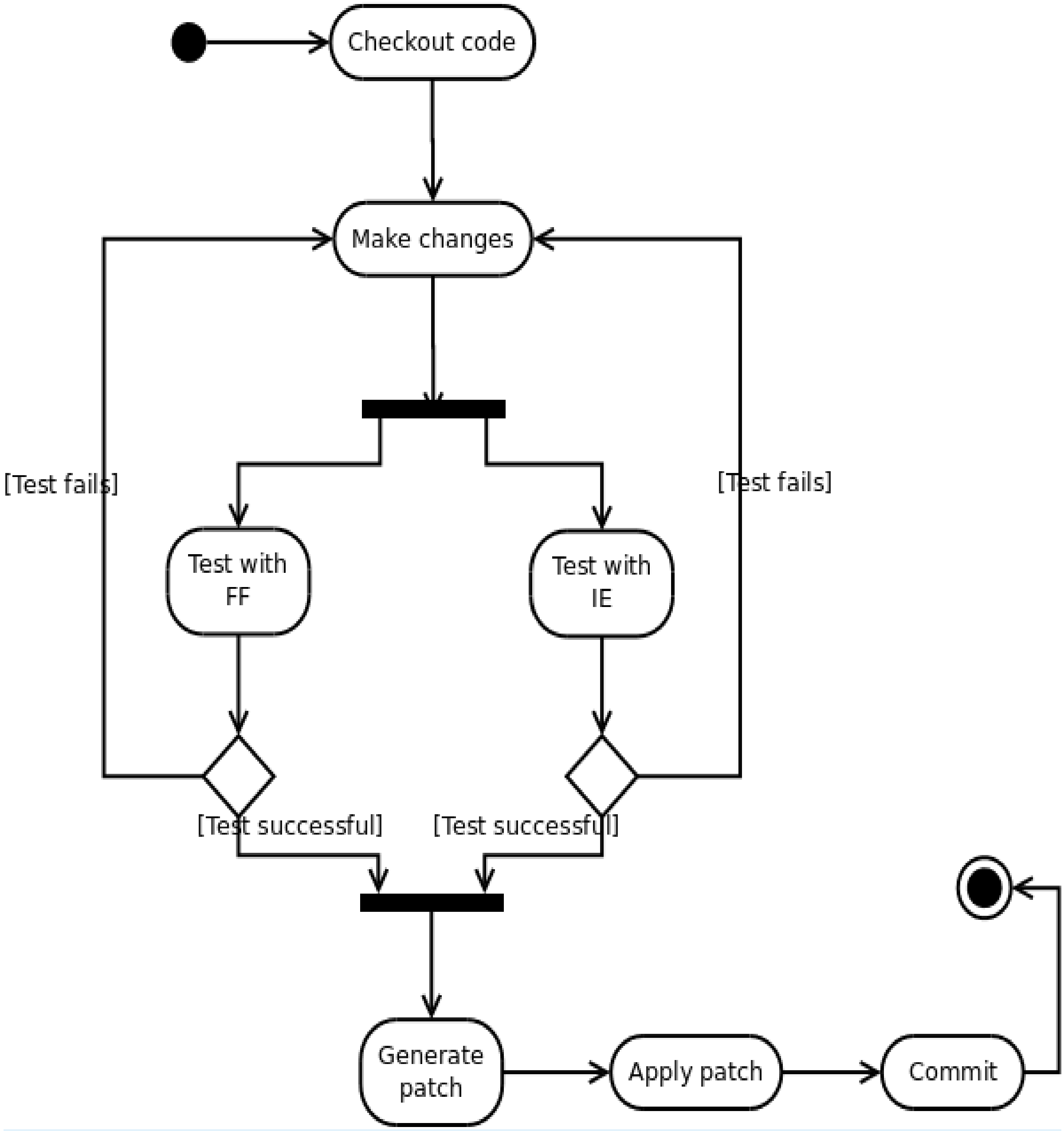
Consider a bulb with a push down switch. The bulb initially remains off. When the switch is pushed down, the bulb is on. Again when the switch is pushed up, the bulb turns off. The lifecycle of the bulb continues in this way until it gets damaged.
Think about these points:

- *What are the different states of the bulb?*
- *What activities are performed in each state?*
- *What action does make the bulb move from one state to another?*



2. Draw an activity diagram to graphically represent the following workflow

Let us consider the development activities of SE Virtual Labs. The process begins by checking out the code from Subversion repository. Necessary modifications are then made to the checked out code (local copy). Once the developer is done with his changes, the application has to be tested to verify whether the new functionality are working fine. This test has to be performed with two of the more popular web browsers: Firefox and Internet Explorer, to support cross-browser accessibility. If testing fails in at least one of the two browser, developer goes back to his code, and fixes it. Only when all the browsers pass the test, a patch is generated from the local copy, and applied to the production code. The local copy is then committed resulting in update of the SVN repository. Note that, if the local copy is committed before generating a patch file, then local changes would get registered, and one won't be further able to generate the patch file.



ASSIGNMENT-10

Estimation of Test Coverage Metrics and Structural Complexity

Identification of basic blocks from a program and determining it's cyclomatic complexity
Consider the following simple C program.

```
//Sum of first n natural numbers (not in the best possible way though)
#include<stdio.h>
int main( int argc, char** argv )
{
int i;
int sum;
sum=0;
for(i=1;i<=n;i++)
sum+=i;
printf(“Sum of first %d numbers is : %d\n”,n,sum);
return 0;
}
```

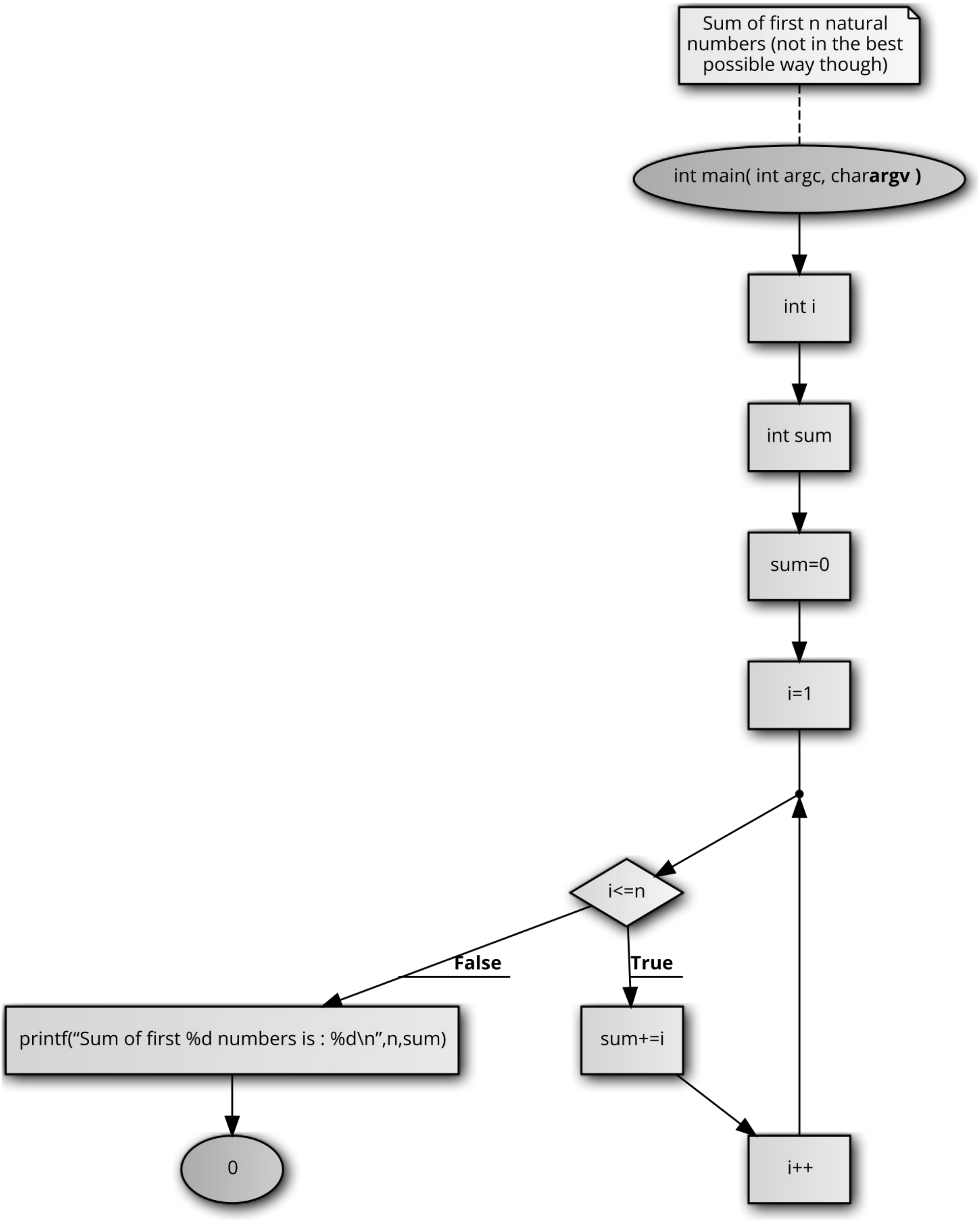
Any sequence of instructions in a program could be represented in terms of basic blocks and a CFG could be drawn using those basic blocks. For the given C program -
1. Identify the basic blocks and verify whether your representation matches with the otput produced after compiling your program.
2. Draw a Control Flow Graph (CFG) using these basic blocks. Again, verify how the CFG generated after compilation relates to the basic blocks identified by the compiler.
3. Calculate McCabe’s complexity from a CFG.

Solution 1:

The basic block found in the given C program is -

1. *int main(int argc, char** argv)*
2. *int i*
3. *int sum*
4. *sum=0*
5. *i=1*
6. *i<=n*
7. *i++*
8. *sum+=i;*
9. *printf(“Sum of first %d numbers is : %d\n”,n,sum);*
10. *return 0*

Solution 2: CONTROL FLOW GRAPH


Sum of first n natural numbers (not in the best possible way though)
int main(int argc, char**argv)
int i
int sum
sum=0
i=1
i<=n
False
True
printf("Sum of first %d numbers is : %d\n",n,sum)
sum+=i
0
i++

Solution 3.

No of edges(E) = 10
No of nodes(N) = 10
McCabe's conplexity obtained from the CFG is M = E – N + 2

= 10 – 10 + 2

= **2**