DECCAN COLLEGE OF ENGINEERING & TECHNOLOGY

(Affiliated to Osmania University & Approved by AICTE, Hyd)

LABORATORY MANUAL
SOFTWARE ENGINEERING LAB

Subject Code :(PC 551 CS)

BE V Semester (AICTE Model Curriculum): 2023-24

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERNG

Faculty: Mrs. Sara Tabassum

DECCAN COLLEGE OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION AND MISSION OF THE INSTITUTION

To create a “Centre for excellence” in the developing rural areas with a view to expose the students to the vast technical and research opportunities of high standards available in the emerging fields of Engineering & Technology, which will contribute to the advancement of society and humankind.

Our Mission is to identify, scientifically evaluate and implement proven, prevention oriented, forward-looking solutions to critical scientific and technological problems and bring technical education of international standard within the reach of all and uplift the weaker Muslim minority community in particular.

We are committed to achieve and maintain excellence in Higher Technical Education, Placement and Research for the benefit of society.

We dedicate and commit ourselves, sustain and foster unmatched excellence in Technical Education. To this end, we will pursue continuous development of infrastructure and enhance state-of-the art equipment to provide our students a technologically up-to-date and intellectually inspiring environment of learning, research, innovation and professional activity and inculcate in them ethical and moral values.

The college is committed to raise the intellectual tone of the young students in understanding and incorporating the basics of rapidly progressing changes in the fields of Science, Engineering & Technology Education with an objective of enhancing their competence by applying their proficiency and skill for Industrial and Economic development.

It is our effort to develop every student into a well qualifies Professional of the future India. Come and join with us to face the challenging world with new vision.

MISSION OF THE DEPARTMENT

The main aim of the department is to impart basic as well as latest and advanced knowledge of Computer Science to the students so as to prepare them to face with ease any tougher assignments they might face in future, be it in higher studies or in jobs.

DECCAN COLLEGE OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAM EDUCATIONAL OBJECTIVES

PEO 1: To make graduates of the Computer Science & Engineering program capable of contributing towards Nation’s development agenda through their ability to solve diverse and complex Computer Science & Engineering problems across a broad range of domains.

PEO 2: To transform graduates of the Computer Science & Engineering program into successful professionals in designing and implementing Products & Services of global standards for becoming entrepreneurs, pursuing higher studies & research.

PEO 3: To enable graduates of the Computer Science & Engineering program for adapting to dynamic changes in technology with potential to solve complex societal problems using the logical and flexible approach in decision making.

DECCAN COLLEGE OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAM OUTCOMES (POS)

PO-1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO-2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO-3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety,and the cultural, societal, and environmental considerations.

PO-4.Conduct investigations of complex problems: Use research- based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO-6.The engineer and society: Apply reasoning in formed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO-7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO-8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO-9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10.Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation ,make effective presentations, and give and receive clear instructions.

PO-11.Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO-12.Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES(PSOS)

PSO-1

The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics and networking for efficient design of computer based systems of varying complexity

PSO-2

Able to analyze and formulate solutions to real world and socially relevant problems over multi disciplinary domains by using latest technologies.

PSO-3

Able to be a technically competent employee, researcher, entrepreneur, excel in competitive exams and zest for higher studies.

SOFTWARE ENGINEERING LAB

PC551CS

Instruction:2 periods per week Duration of SEE: 3 hours
CIE: 25marks SEE: 50marks
Credits: 1

Objectives:

1. To understand the software engineering methodologies for project development.
2. To gain knowledge about open-source tools for Computer Aided Software Engineering (CASE).
3. To develop test plans and test cases to perform various testing.

Outcomes: Student will be able to

1. Analyze and design software requirements in an efficient manner.
2. Use open-source case tools to develop software.
3. Implement the design, debug and test the code.

I. FORWARD ENGINEERING

Students have to form a team with a batch size of two or three and take up a **ca**
project to analyze, plan, design UML models and create a prototypical model
deliverables) by coding the developed designs and finally documenting consid
example of the following domains: -

1. Academics (Course Registration System, Student marks analyzing system)
2. Health Care (Expert system to prescribe medicines for given symptoms, Re
Diagnostics, Patient/Hospital Management System)
3. Finance (Banking: ATM/NetBanking, UPI:PayTM/Phone Pay, Stocks:Zero
4. E-Commerce (various online shopping portals like FlipKart/Amazon/Myntr
5. Logistics (Postal/Courier: IndiaPost/DTDC/UPS/FedEx, Freight:Maersk)
6. Hospitality (Tourism Management: Telangana Tourism/Incredible India, Ev
Management: MeraEvents/BookMyShow/Explara/EventBrite)
7. Social Networking (LinkedIn, FaceBook, Shaadi.com, BharatMatrimony,T
8. Customer Support (Banking Ombudsman, Indian Consumer Complaints For
9. Booking/Ticketing(Food:Zomato/Swiggy/BigBasket/Grofers/JioMart, Hote
Travel: {Cars:Uber/OLA/Zoom, Railways:IRCTC, Buses:OnlineTSRTC/RedI
Flights:MakeMyTrip/Goibibo,Ships:Lakport})

II. REVERSE ENGINEERING

Students have to refer any project repository: GitLab/GitHub, execute the code
observe its functionalities/features/requirements and by the help of any tool der
from the code for understanding the relationships among various
subsystems/classes/components and if the tool partially generates models then i
associating elements to judge/mark the appropriate relationships.

III. TESTING

Prepare Test Plan and develop Test Case Hierarchy to monitor or uncover/report
manual/automated testing tools

Software Required: 1. StarUML/Umbrello, NetBeans/Eclipse IDE, XAMPP/MEAN stack, JUnit, JMeter, Selenium, Bugzilla

COURSE OUTCOMES (CO'S):

SUBJECT NAME : SOFTWARE ENGINEERING LAB CODE : PC551CS

CO No.	**Course Outcomes**	**Taxonomy Level**
PC531CS.1	Interpret a variety of approaches and perspectives of system development.	Understanding
PC531CS.2	Identify the requirements which are relevant to the design of a system.	Applying
PC531CS.3	Model software design with a set of objects and their relationships using structural modeling.	Applying
PC531CS.4	Take part in using advanced & behavioral modeling to develop a case study.	Analysing
PC531CS.5	Design the activities with the help of behavioral modeling.	Creating
PC531CS.6	Develop components through architectural modeling.	Creating

Outcomes

At the end of the course, the student will

□ Understand the role of software

□ Determine the problems occurred due to various software crisis.

□ Understand the need of requirements engineering process.

□ Compare the process of requirements development and requirements management.

□ Determine the importance of requirements classification.

□ Determine the procedure of regression testing.

□ Understand the importance of performance testing.

□ Determine the concepts of software metrics used before software deployment.

LIST OF CONTENTS

SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

There are various development approaches defined and designed which are used/employed during development process of software these approaches are also referred as " Software Development Process Models" (eg: Waterfall model, Incremental model, etc.). Each process model follows a particular life cycle in order to ensure success in process of software development. Software life cycle models describe phases of the software cycle and the order in which those phases are executed. Each phase produces deliverable 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.

Life Cycle Phases

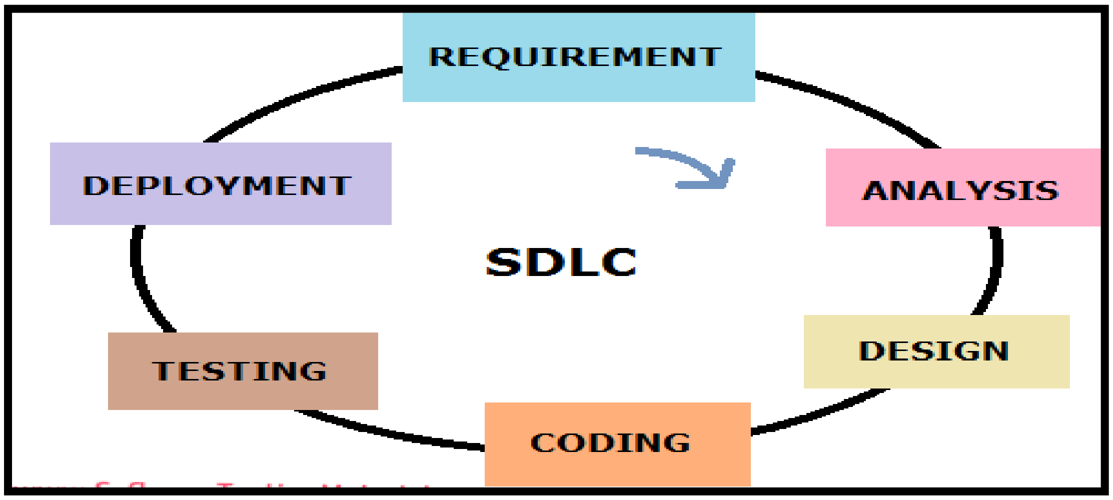
There are following six phases in every software development life cycle model:

1. Requirement gathering and analysis
2. Design
3. Implementation or coding
4. Testing
5. Deployment
6. Maintenance

- Requirement gathering and analysis: Business requirements are gathered in this phase. This phase is the main focus of the project managers and stakeholders. Meetings with mangers, stakeholders and users are held in order to determine the requirements like; who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirements gathering phase.
- After requirements gathering these requirements are analyzed for the their validity and the possibility of incorporating the requirements in the system to be developed is also studied. Finally, a requirements specification document is created which serves the

purpose of guideline for the next phase of the model.

- Design: in this phase the system and software design is prepared from the requirement specifications which were studied in the first phase. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.
- Implementation/ Coding: On receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. This is the longest phase of the software development life cycle.
- Testing: After the code is developed it is tested again the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase unit testing, integration testing, system testing, acceptance testing are done.
- Deployment: After successful testing the product is delivered/ deployed to the customer for their use.
- Maintenance: Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance.



Software Development Life Cycle (SDLC

LIST OF EXPERIMENTS

Do the following exercises for any case study given in the list of sample projects or any other projects:

S.no	**Program Names**
1)	Development of problem statement. (Abstract)
2)	Preparation of Software Requirement Specification Document
3)	Preparation of Software Configuration Management
4)	Performing the UML Designing by using any CASE tools (Star UML).
5)	Perform Forward Engineering of Design Diagram in Star UML
6)	Re-Write the Code and execute the project (in any Standard IDE)
7)	Perform Reverse Engineering of code in Star UML
8)	Develop test cases for various testing techniques.
9)	Perform Automated Testing (Use Testing Tools)
10)	Sample Viva

LIST OF TOOLS REQUIRED

1. **Designing ,(Forward & Reverse Engineering):**

   Rational Rose,StarUM:L/Umbrello,etc..,

2. **Coding**

   NetBeans/Eclipse IDE, Visual Studio etc..

3. **Server:**

   XAMPP/MEAN stack,etc..,

   Tutorial :https://www.youtube.com/watch?v=jLqBiSDNXO0

4. **Connecting Frontend & Backend :**

   Tutorial: https://www.youtube.com/watch?v=F8Ke30Dqq0g

5. **Testing :**

   JUnit, JMeter, Selenium, Bugzilla,etc..,

   Sample Case Studies for Reference only

   1.https://www.youtube.com/watch?v=ZBgTzx46B8s

   2. https://www.youtube.com/watch?v=smmKiv-TloI

\

INSTALLATION GUIDE

Study and usage of any Design phase CASE tool

CASE Tool: STARUML

How to Install StarUML on Windows 10 and higher..

Star UML is a UML (**Unified Modeling Language**) tool, introduce by MKLab. It is an open-source modeling tool that supports the UML framework for system and software modeling. StarUML is based on UML version 1.4, it provides 11 different types of diagram and it accepts UML 2.0 notation. Version 2.0 was released for beta testing under a property license.

StarUML is actively supporting the **MDA (Model Driven Architecture).** It approaches by supporting the UML profile concept and allowing it to generate code for multiple languages. It also provides a number of bug fixes and improved compatibility with the modern versions of the Windows Operating System.

StarUML is mostly used by the Agile and small development teams, professional persons and used by the educational institutes

Diagram Types in StarUML

1.Use Case Diagram

2.Class Diagram

3.Sequence Diagram

4.Collaboration Diagram

5.Statechart Diagram

6.Component Diagram

7.Deployment Diagram

8. Activity Diagram and many others

Features of StarUML

1. It supports multi-platform such as macOS, Windows, and Linux.

2. It involves UML 2.x.standard compliant.

3. Includes Entity-Relationship diagram (ERD), Data-flow diagram (DFD), and Flowchart diagram.

4. It creates multiple windows.

5. It has modern UX and dark and light themes.

6. Featured with retina (High-DPI) display support.

7. Includes model-driven development.

8. It has open APIs.

9. Supports various third-party extensions.

10. Asynchronous model validation.

11. It can export to HTML docs.

Steps to Download and Install StarUML

Step 1: Go on the browser, type in the URL “StarUML” Step 2: Click on the very first search “Download-StarUML”.

Step 3: There will be 3 Operating Systems (OS) options, click on the option as per the devise OS. Step 4: Now, right-click on the downloaded file, select “Show in Folder” option. Step 5: Click on the open file, a popup window opens, click on the “Yes” button.

Step 6: Installation gets start. After installation popup opens to ask to buy a license. If you want to click on the "Buy Now" button or else close that window. StarUML is ready to use.

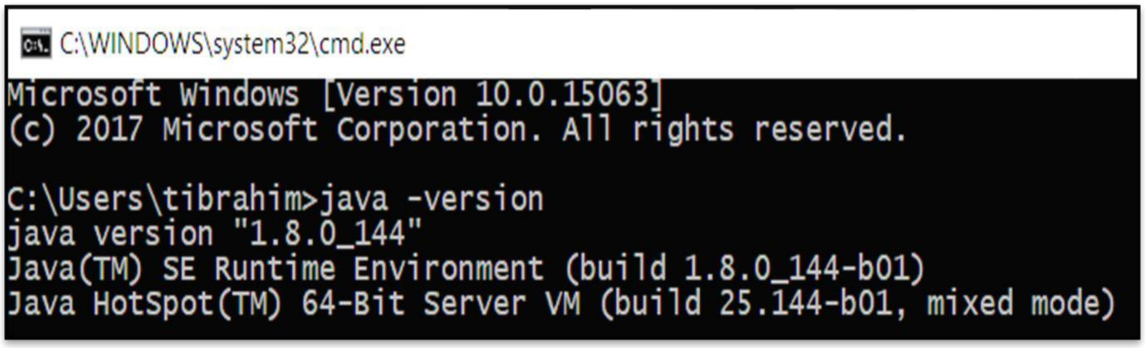
Testing Tools

1.JMeter Installation

Apache JMeter installation on Windows 10.

Step 1: Install Java 8 Download latest Java 8 from Oracle website. You will need to have Oracle account for downloading Java. Run the Java installer package. Verify installed Java version by running this command on Command Prompt.

```
java -version
```

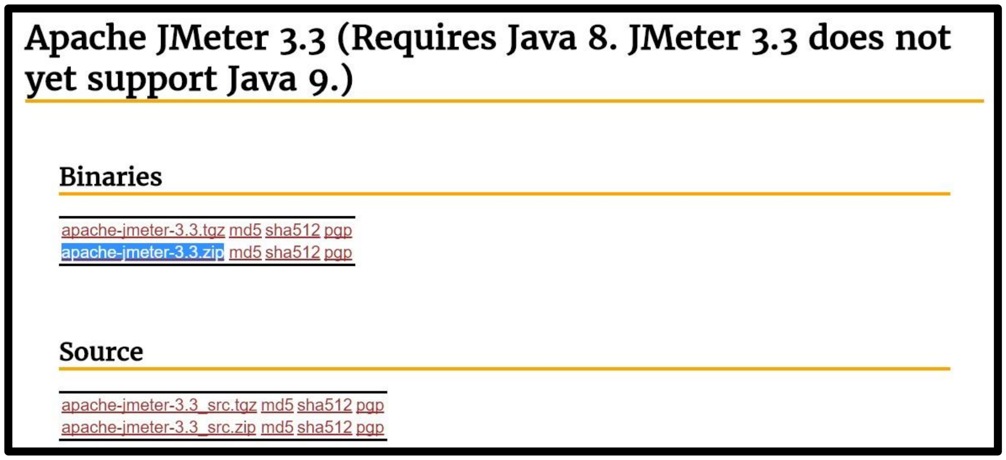


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\tibrahim>java -version
java version "1.8.0_144"
Java(TM) SE Runtime Environment (build 1.8.0_144-b01)
Java HotSpot(TM) 64-Bit Server VM (build 25.144-b01, mixed mode)
```

Step 2: Download Apache JMeter

You can download the latest version of Apache JMeter fromhttp://jmeter.apache.org/download_jmeter.cgi



Apache JMeter 3.3 (Requires Java 8. JMeter 3.3 does not yet support Java 9.)

Binaries

apache-jmeter-3.3.tgz md5 sha512 pgp
apache-jmeter-3.3.zip md5 sha512 pgp

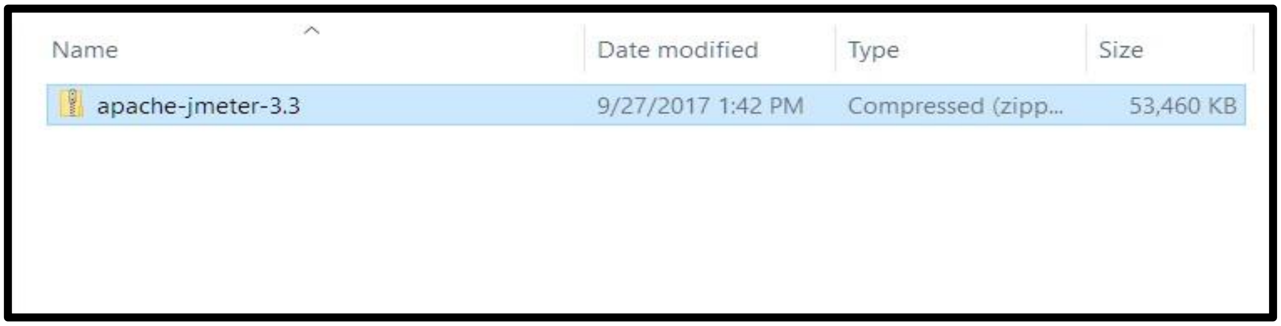
Source

apache-jmeter-3.3_src.tgz md5 sha512 pgp
apache-jmeter-3.3_src.zip md5 sha512 pgp

Download the Binaries zip file.

Step 3: Extract Apache JMeter

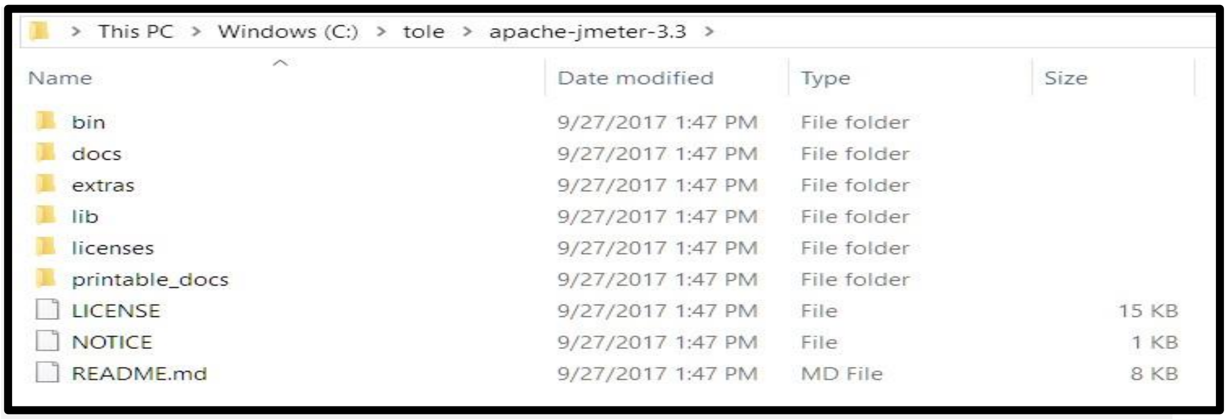
Find the downloaded zip file and extract it to your desired directory.



Name	Date modified	Type	Size
apache-jmeter-3.3	9/27/2017 1:42 PM	Compressed (zipp...	53,460 KB

Downloaded zip file For example, I extract the ***apache-jmeter-3.3.zip*** to ***C:\tole\***.

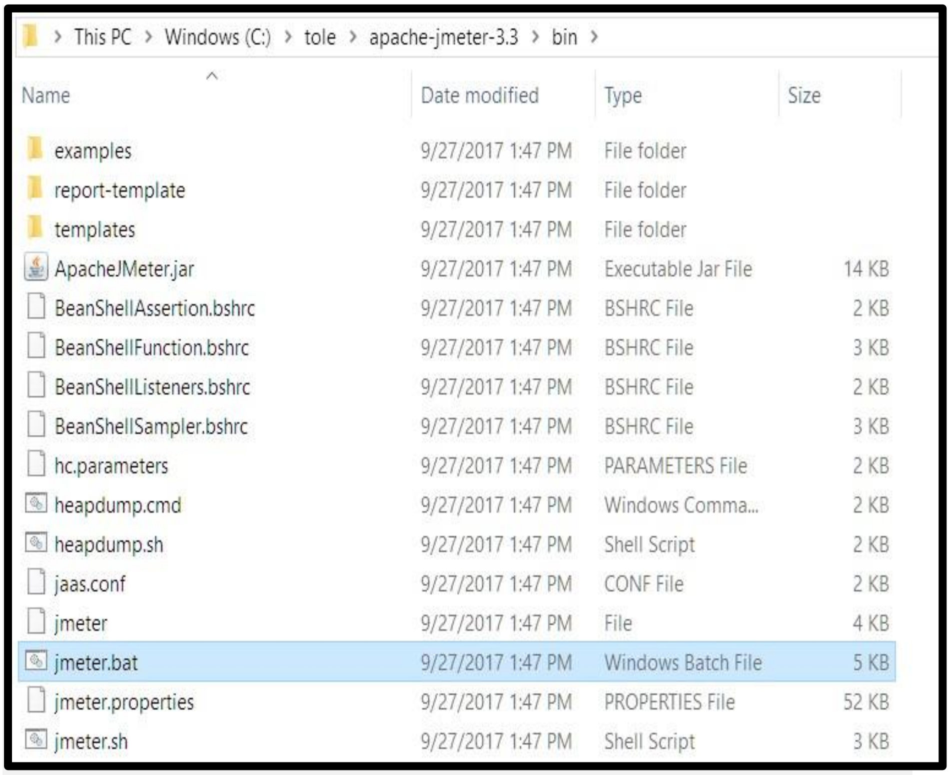
This is the snapshot of the extracted directory.



> This PC > Windows (C:) > tole > apache-jmeter-3.3 >

Name	Date modified	Type	Size
bin	9/27/2017 1:47 PM	File folder	
docs	9/27/2017 1:47 PM	File folder	
extras	9/27/2017 1:47 PM	File folder	
lib	9/27/2017 1:47 PM	File folder	
licenses	9/27/2017 1:47 PM	File folder	
printable_docs	9/27/2017 1:47 PM	File folder	
LICENSE	9/27/2017 1:47 PM	File	15 KB
NOTICE	9/27/2017 1:47 PM	File	1 KB
README.md	9/27/2017 1:47 PM	MD File	8 KB

Apache JMeter extracted directory Now go to ***/bin*** folder. You will find ***jmeter.bat***.

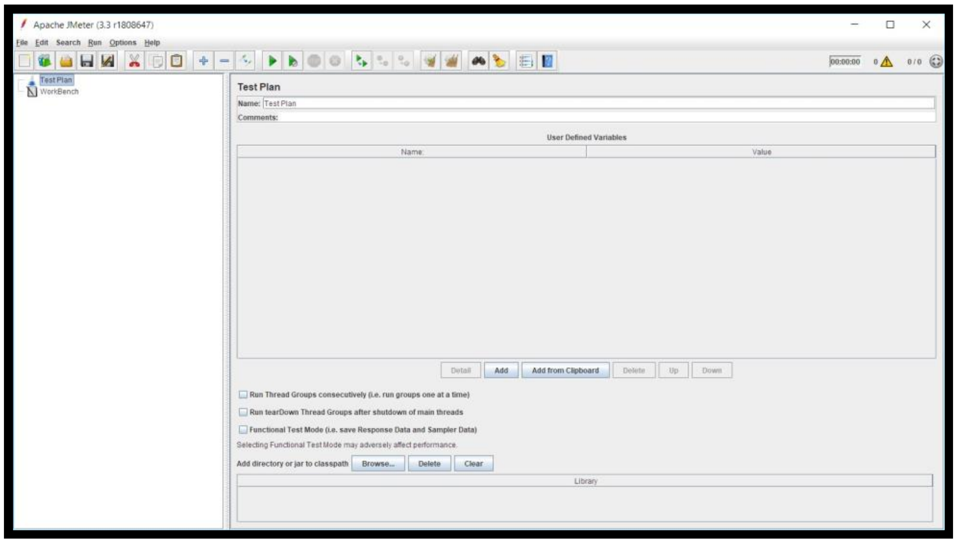


› This PC › Windows (C:) › tole › apache-jmeter-3.3 › bin ›

Name	Date modified	Type	Size
examples	9/27/2017 1:47 PM	File folder	
report-template	9/27/2017 1:47 PM	File folder	
templates	9/27/2017 1:47 PM	File folder	
ApacheJMeter.jar	9/27/2017 1:47 PM	Executable Jar File	14 KB
BeanShellAssertion.bshrc	9/27/2017 1:47 PM	BSHRC File	2 KB
BeanShellFunction.bshrc	9/27/2017 1:47 PM	BSHRC File	3 KB
BeanShellListeners.bshrc	9/27/2017 1:47 PM	BSHRC File	2 KB
BeanShellSampler.bshrc	9/27/2017 1:47 PM	BSHRC File	3 KB
hc.parameters	9/27/2017 1:47 PM	PARAMETERS File	2 KB
heapdump.cmd	9/27/2017 1:47 PM	Windows Comma...	2 KB
heapdump.sh	9/27/2017 1:47 PM	Shell Script	2 KB
jaas.conf	9/27/2017 1:47 PM	CONF File	2 KB
jmeter	9/27/2017 1:47 PM	File	4 KB
jmeter.bat	9/27/2017 1:47 PM	Windows Batch File	5 KB
jmeter.properties	9/27/2017 1:47 PM	PROPERTIES File	52 KB
jmeter.sh	9/27/2017 1:47 PM	Shell Script	3 KB

/bin directory

Double click on that file. The Apache JMeter GUI will be opened. You can create a shortcut for it on your desktop too.



Apache JMeter (3.3 r1808647)

File Edit Search Run Options Help

00:00:00 0 0 / 0

Test Plan
WorkBench

Test Plan

Name: Test Plan

Comments:

User Defined Variables

Name:	Value

Detail Add Add from Clipboard Delete Up Down

☐ Run Thread Groups consecutively (i.e. run groups one at a time)

☐ Run tearDown Thread Groups after shutdown of main threads

☐ Functional Test Mode (i.e. save Response Data and Sampler Data)

Selecting Functional Test Mode may adversely affect performance.

Add directory or jar to classpath Browse... Delete Clear

Library

▯

Apache JMeter GUI is installed.

<u>2.Selenium IDE</u>

Selenium IDE (Integrated Development Environment) is primarily a record/run tool that a test case developer uses to develop **Selenium** Test cases. **Selenium IDE** is an easy to use tool from the **Selenium** Test Suite and **can** even be used by someone new to developing automated test cases for their web applications
It is a Firefox add-on that creates tests very quickly through its record-and-playback functionality.

This feature is similar to that of QTP. It is effortless to install and easy to learn.

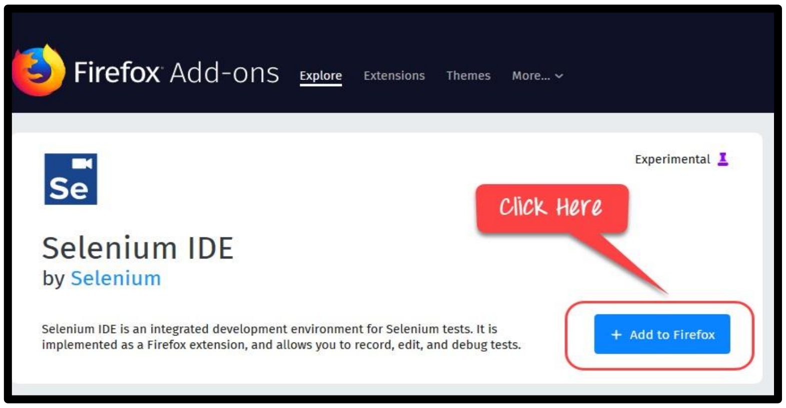
Installation of Selenium IDE

What you need

- Mozilla Firefox
- Active Internet Connection

If you do not have Mozilla Firefox yet, you can download it from http://www.mozilla.org/enUS/firefox/new.

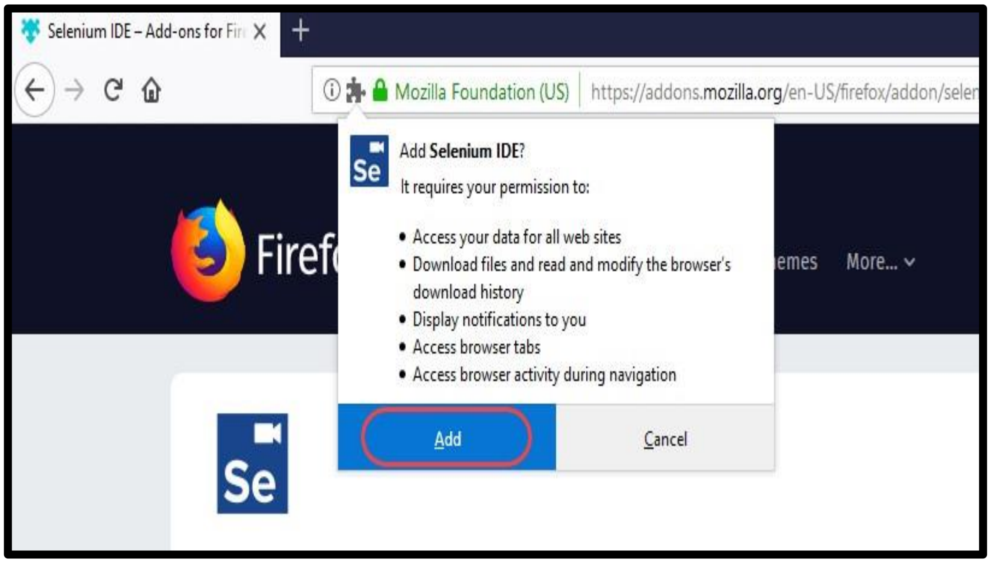
Steps 1) Launch Firefox and navigate to https://addons.mozilla.org/en-US/firefox/addon/seleniumide/. Click on Add to Firefox



Selenium IDE
by Selenium

Selenium IDE is an integrated development environment for Selenium tests. It is implemented as a Firefox extension, and allows you to record, edit, and debug tests.

Steps 2) Wait until Firefox completes the download and then click "**Add.**"

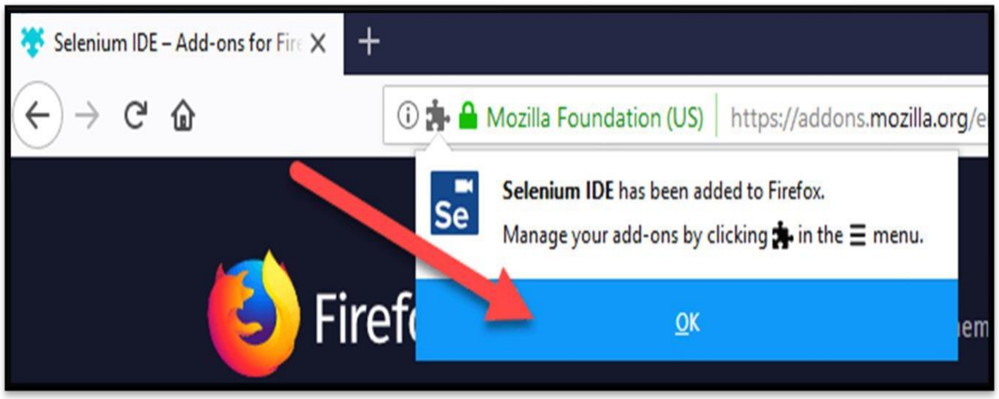

Selenium IDE – Add-ons for Fir
Mozilla Foundation (US)
https://addons.mozilla.org/en-US/firefox/addon/selen
Firefo
emes
More...
Add
Cancel

Add **Selenium IDE**?

It requires your permission to:

- Access your data for all web sites
- Download files and read and modify the browser's download history
- Display notifications to you
- Access browser tabs
- Access browser activity during navigation

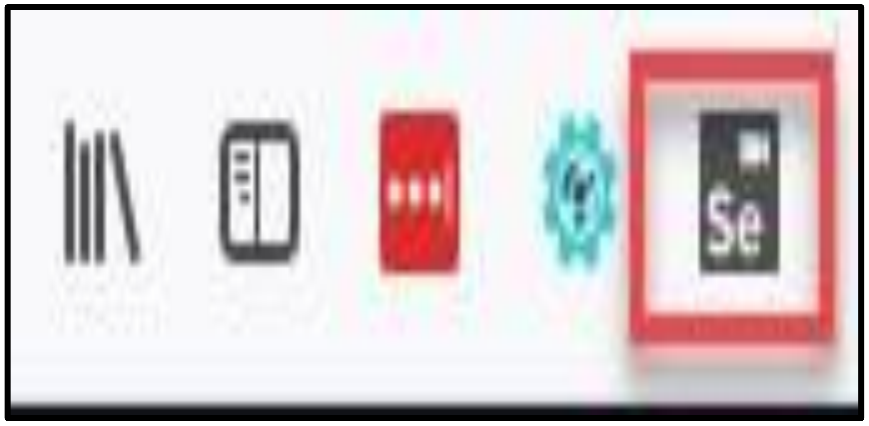
Steps 3) Once install is complete, you will get a confirmation message. Click **"OK"**



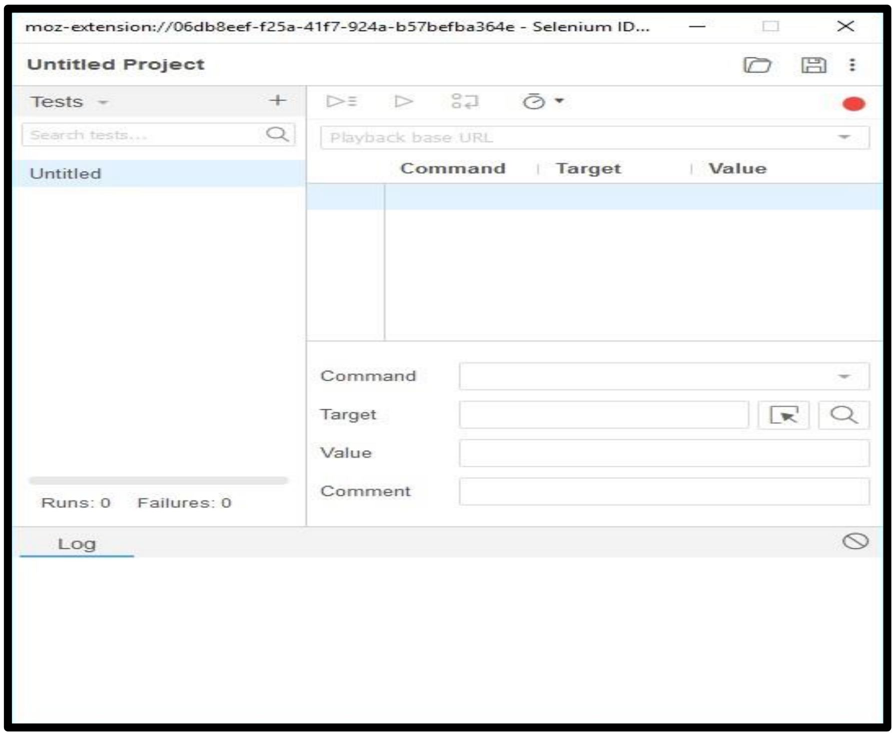
Selenium IDE has been added to Firefox.

Manage your add-ons by clicking in the ☰ menu.

Steps 4) Click on the Selenium IDE icon



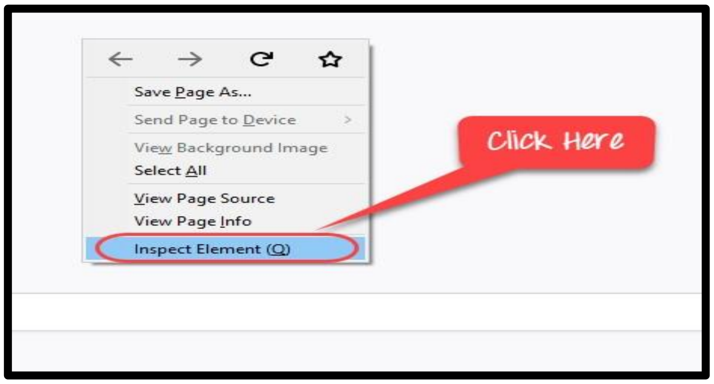
Selenium IDE will open



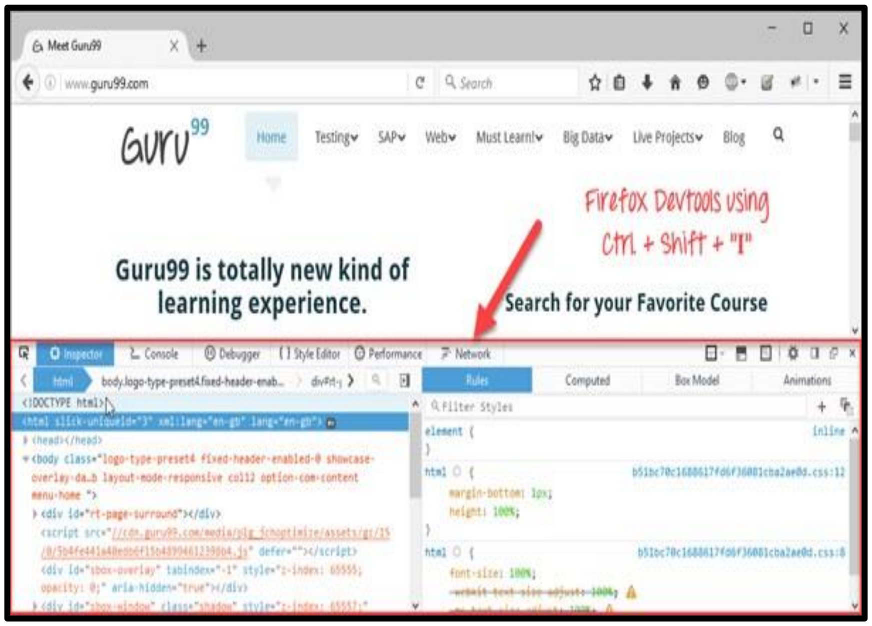
Firefox DevTools in Firefox

Firefox DevTools is a Firefox feature that we will use to **inspect the HTML elements** of the web application under test. It will provide us the name of the element that our Selenese command would act upon.

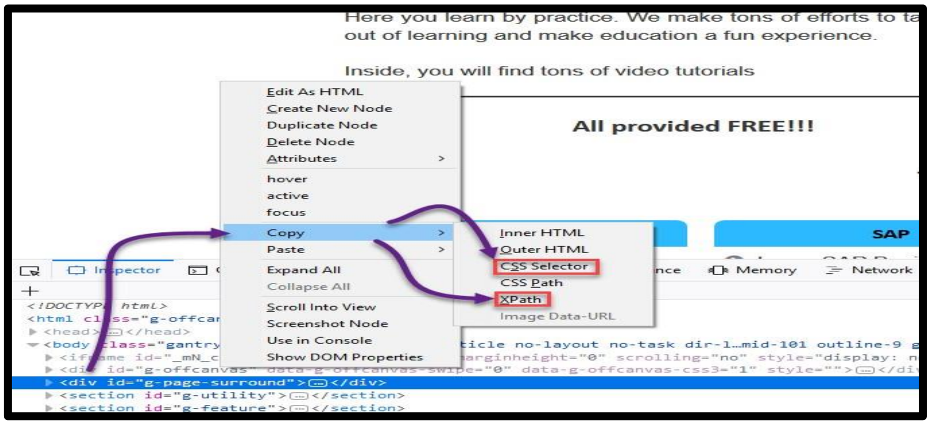
Step 1) Right click anywhere on the page and select Inspect Element. You can also use shortcut Cntrl + Shift + I



Step 2) You will see the Interface



Step 3) You can right click on an element and chose CSS or XPath. This is useful in object identification

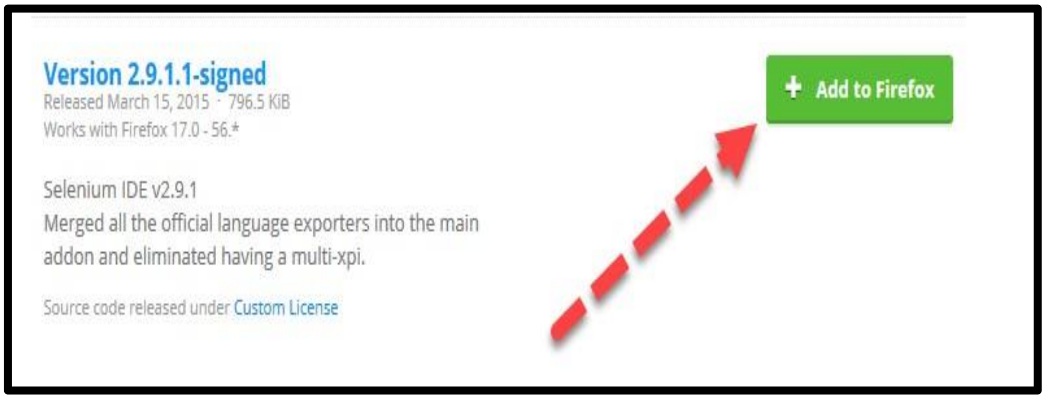


Note: Likewise, you can also use Developer Tools in Chrome to identify object properties

Selenium IDE was deprecated, and the development had stopped. Only recently the project has been resurrected. The new Selenium lacks many features compared to the deprecated IDE. Features are being added but at a slow pace. To explore all the features of Selenium IDE, we recommend you use the old version. To use the old version of IDE

Step 1) Use Firefox 54 Portable Version check here

Step 2) Visit Selenium IDE version https://addons.mozilla.org/en-US/firefox/addon/seleniumide/versions/and install



Version 2.9.1.1-signed
Released March 15, 2015 · 796.5 KiB
Works with Firefox 17.0 - 56.*

Selenium IDE v2.9.1
Merged all the official language exporters into the main addon and eliminated having a multi-xpi.

Source code released under Custom License

The following features may not be available in latest IDE version. We will keep updating the tutorials as the new version is updated.

Plugins:Selenium IDE can support additional Firefox add-ons or plugins created by other users. You can visithere for a list of Selenium add-ons available to date.

Install them just as you do with other Firefox add-ons.

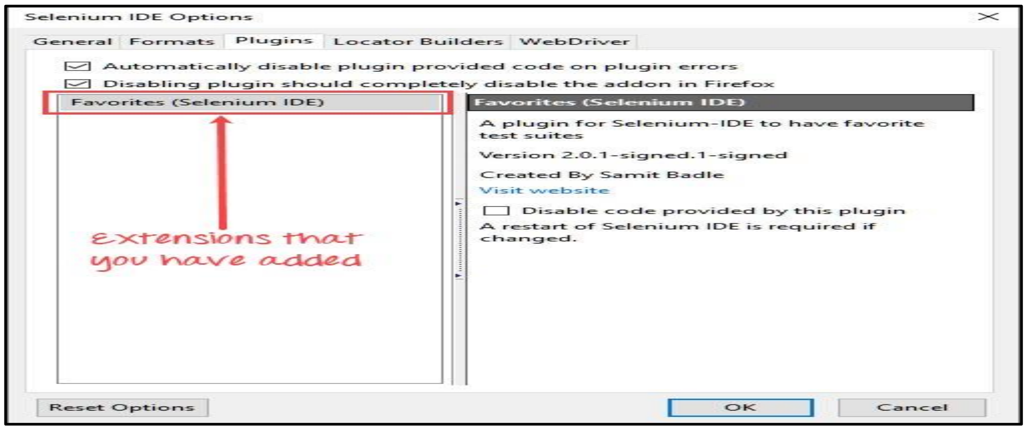
By default, Selenium IDE comes bundled with 4 plugins:

1. Selenium IDE:C#Formatters 2.Selenium IDE:JavaFormatters

3.Selenium IDE:PythonFormatters 4. Selenium IDE: Ruby Formatters

These four plugins are required by Selenium IDE to convert Selenese into different formats.

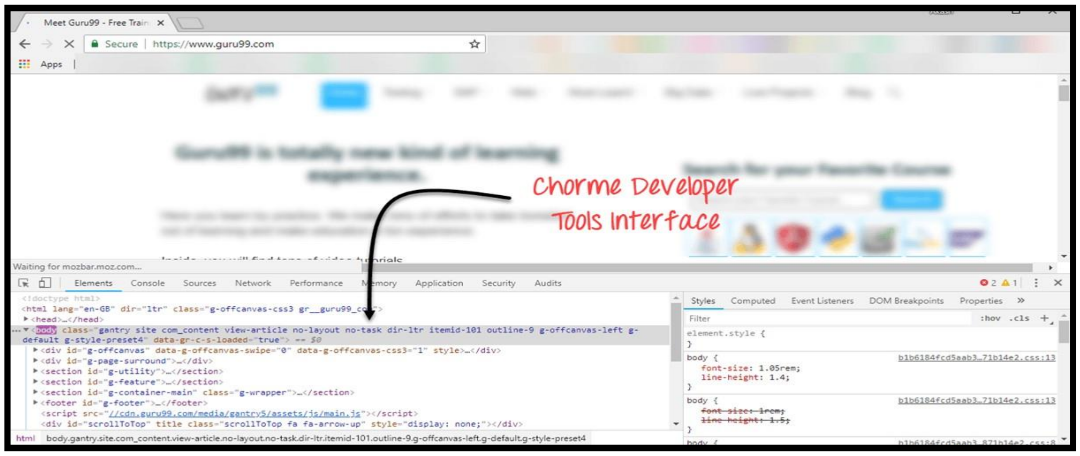
The Plugins tab shows a list of all your installed add-ons, together with the version number and name of the creator of each.



☑ Automatically disable plugin provided code on plugin errors
☑ Disabling plugin should completely disable the addon in Firefox

Favorites (Selenium IDE)

Favorites (Selenium IDE)
A plugin for Selenium-IDE to have favorite test suites
Version 2.0.1-signed.1-signed
Created By Samit Badle
Visit website
☐ Disable code provided by this plugin
A restart of Selenium IDE is required if changed.



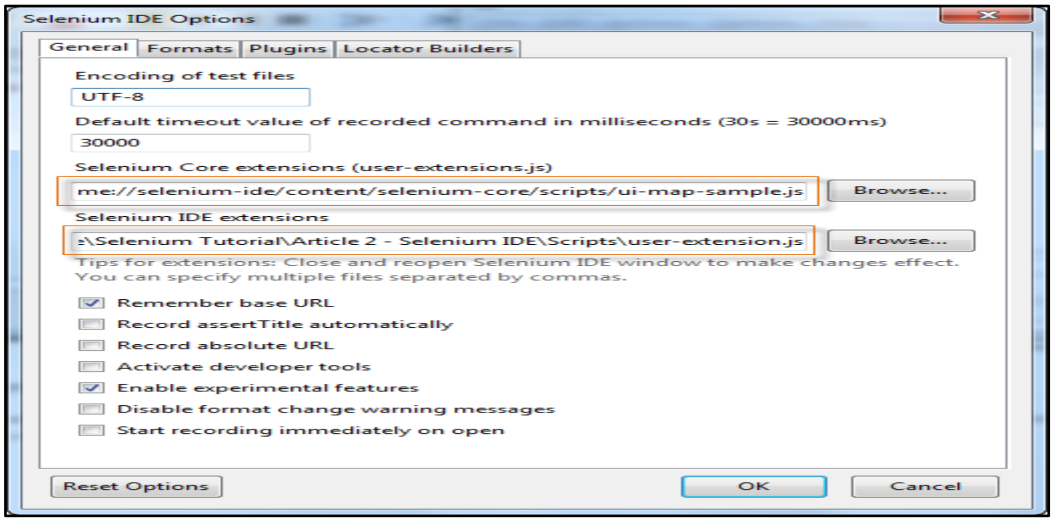
```
<!doctype html>
<html lang="en-GB" dir="ltr" class="g-offcanvas-css3 gr__guru99_co">
▶<head>…</head>
▼<body class="gantry site com_content view-article no-layout no-task dir-ltr itemid-101 outline-9 g-offcanvas-left g-default g-style-preset4" data-gr-c-s-loaded="true"> == $0
  ▶<div id="g-offcanvas" data-g-offcanvas-swipe="0" data-g-offcanvas-css3="1" style>…</div>
  ▶<div id="g-page-surround">…</div>
  ▶<section id="g-utility">…</section>
  ▶<section id="g-feature">…</section>
  ▶<section id="g-container-main" class="g-wrapper">…</section>
  ▶<footer id="g-footer">…</footer>
   <script src="//cdn.guru99.com/media/gantry5/assets/js/main.js"></script>
   <div id="scrollToTop" title class="scrollToTop fa fa-arrow-up" style="display: none;"></div>
html  body.gantry.site.com_content.view-article.no-layout.no-task.dir-ltr.itemid-101.outline-9.g-offcanvas-left.g-default.g-style-preset4
```

```
element.style {
}
body {                                b1b6184fcd5aab3…71b14e2.css:13
   font-size: 1.05rem;
   line-height: 1.4;
}
body {                                b1b6184fcd5aab3…71b14e2.css:13
   ~~font-size: 1rem;~~
   ~~line-height: 1.5;~~
}
```

User Extensions

Selenium IDE can support user extensions to provide advanced capabilities. User extensions are in the form of JavaScript files. You install them by specifying their absolute path in either of these two fields in the Options dialog box.

- Selenium Core extensions (user-extensions.js)
- Selenium IDE extensions


Selenium IDE Options

General | Formats | Plugins | Locator Builders

Encoding of test files
UTF-8
Default timeout value of recorded command in milliseconds (30s = 30000ms)
30000
Selenium Core extensions (user-extensions.js)
me://selenium-ide/content/selenium-core/scripts/ui-map-sample.js
Selenium IDE extensions
:\Selenium Tutorial\Article 2 - Selenium IDE\Scripts\user-extension.js
Tips for extensions: Close and reopen Selenium IDE window to make changes effect.
You can specify multiple files separated by commas.

- [✓] Remember base URL
- [ ] Record assertTitle automatically
- [ ] Record absolute URL
- [ ] Activate developer tools
- [✓] Enable experimental features
- [ ] Disable format change warning messages
- [ ] Start recording immediately on open

CASE STUDIES WITH DESCRIPTION & MODULES

1: Academics (Course Registration System, Student marks analyzing system)

AIM:Course Registration System

Problem Statement:

The software which displays the list of courses available for the mark that the student get and the student can able to allocate best course from the choice available. It displays and allocates courses based on student ranking. The student ranking is based on marks, caste and community. Based on caste and community the ranking may varied. This software allows the student to choose the best college for the available courses.

Modules:

1. Login
2. student_info
3. .course_details
4. course_registration
5. confirmation

AIM :Student marks analyzing system

Problem Statement:

Student marks analyzing system has to be developed for analyzing obtained by the students who scored in Semester Examination The System should provide following functionalities

1. The System obtains following information's from the faculty generates report Roll No, Name, Department, Semester, Marks obtained in each subject.

2. The total for each student should be calculated and ranked based on total and pass in all the subject appeared.

3. The Final report should display rank, percentage, Class, Pass/Fail Status for each student.

Modules:

1. Login
2. student_mark
3. View report

2 : Health Care AIM : Health Care (Expert system to prescribe medicines for given symptoms, Remote Diagnostics, Patient/Hospital Management System)

Problem Statement:

The project is mainly focused on medical field in expert system where the person or patient login to the system and must select what are the symptoms for him that field information will be given to expert system. Expert system will diagonise what type of disease using medical database. The medical database will get all the information about medicine from pharmacy module and generate the prescription for the right symptoms and it gives to expert system.The medical database gives types of disease information to expert system.

Modules:

1. Symptoms
2. Expert System
3. Medical DB
4. Prescription
5. Pharmacy

3: Finance (Banking:ATM/NetBanking, UPI:PayTM/PhonePay,Stocks:Zerodha)

AIM : Banking:ATM/NetBanking

Problem Statement:

This project on atm is based on the processing of debit cards in atm.There are two main section and the administrator section.In withdraw module the requested amount is deducted from the user's account if sufficient balance is available.If the balance in the user's account is insufficient the service is denied.In deposit module the deposited amount is added to the user's balance.In balance enquiry module the amount that is present in the user's account is displayed. The user must enter his pin and account number to perform any of the transactions.

Modules:

Login

Deposit

Withdraw

Changing pin number

Balance enquiry

4 : E-Commerce (various online shopping portals like FlipKart / Amazon /Myntra)

AIM : E-Commerce

Problem Statement: Our E-commerce project aims to revolutionize online shopping by enhancing user interface, optimizing search functionality, ensuring robust security measures, and prioritizing mobile responsiveness. We are committed to improving supply chain efficiency, implementing transparent customer support processes, and personalizing user experiences through advanced recommendation engines. With a focus on sustainability and ethical practices, our project seeks to establish a reliable, customer-centric e-commerce platform that fosters trust and loyalty.

Modules:

1. User Authentication and Authorization:
2. Product Management
3. Shopping Cart
4. C heckout and Payment Gateway Integration
5. Search and Filtering
6. User Reviews and Ratings
7. User Account Management
8. Order Management

5: Logistics (Postal/Courier:IndiaPost/DTDC/UPS/FedEx,Freight:Maersk)

AIM : Logistics

Problem Statement: Our logistics project focuses on optimizing the movement of goods, both domestically and internationally. Integrating major postal and courier services such as IndiaPost, DTDC, UPS, and FedEx, alongside freight solutions from Maersk, the project aims to enhance end-to-end visibility, tracking, and delivery efficiency. Key features include real-time package tracking, automated route optimization, and seamless collaboration with shipping partners. With a commitment to reliability and transparency, our project aims to revolutionize logistics, ensuring timely and secure delivery across diverse channels.

Modules

1. Order Fulfillment

2. Package Tracking
3. Inventory Management
4. Delivery Scheduling
5. Returns Management
6. Billing and Invoicing

6:Hospitality(TourismManagement:Telangana,Tourism/IncredibleIndia,Event Management:MeraEvents/BookMyShow/Explara/EventBrite)

AIM : Hospitality

Problem Statement: hospitality project is designed to enhance the tourism experience in Telangana and contribute to the promotion of Incredible India. By integrating with prominent event management platforms such as MeraEvents, BookMyShow, Explara, and EventBrite, the project aims to create a seamless ecosystem. Key features include personalized tourism itineraries, real-time event bookings, and efficient management of tourist services. With a focus on user engagement and satisfaction, our project seeks to position Telangana as a prime tourism destination while providing event organizers and attendees with a comprehensive and user-friendly platform for event planning and participation.

Modules

1. Tourism Information and Guides
2. Accommodation Booking
3. Transportation Services
4. User Reviews and Ratings

7: Social Networking (LinkedIn, FaceBook, Shaadi.com, Bharat Matrimony, Tinder)

AIM: Social Networking

Problem Statement:

social networking project aims to create a versatile platform catering to personal and professional interactions. By integrating features from leading platforms like LinkedIn, Facebook, Shaadi.com, Bharat Matrimony, and Tinder, the project seeks to provide a comprehensive and inclusive experience. Users can connect with professionals for networking, engage in social interactions, find life partners through matrimonial services, and explore dating opportunities.

The project prioritizes user privacy, security, and personalized experiences, fostering a dynamic community that spans personal and professional spheres.

Modules:

1. User Profiles
2. News Feed
3. Connections and Friend Requests
4. Messaging and Chat
5. Notifications
6. Privacy Settings

8 : Customer Support (Banking Ombudsman,Indian Consumer Complaints Forum)

AIM : Customer Support

Problem Statement:

Our customer support project aims to create a robust platform for addressing grievances and providing assistance in the realms of banking and consumer complaints. By integrating features inspired by the Banking Ombudsman and Indian Consumer Complaints Forum, the project seeks to empower users to resolve issues, seek advice, and advocate for consumer rights. Key functionalities include a user-friendly complaint submission process, an efficient ticketing system, and a knowledge-sharing community. With a commitment to transparency and fair dispute resolution, our project aims to elevate customer support in the banking and consumer sectors.

Modules:

1. Complaint Submission:
2. Ticketing System
3. Document Upload and Verification
4. Dispute Resolution Workflow
5. Status Tracking
6. Feedback and Rating System

9:Booking/Ticketing(Food:Zomato/Swiggy/BigBasket/Grofers/JioMart, Hotel:OYO/Trivago or Travel:{Cars:Uber/OLA/Zoom, Railways:IRCTC, Buses:OnlineTSRTC/RedBus/AbhiBus,Flights:MakeMyTrip/Goibibo, Ships:Lakport})

AIM : Booking/Ticketing

Problem Statement:

This project is about online ticket reservation and consists of two modules. The reservation and the cancellation module. The reservation module allows the user to reserve tickets for a particular train on a particular date. If there is a ticket available, the users can know the vacancy details through the enquiry module. The cancellation module allows user to cancel the tickets for a particular date through reservation officer (system). This module performs status reveal before tickets are being reserved and after they get booked. All these modules together prove to be a flexible online reservation system and it provides complete flexibility to end users and it assumes the desired performance.

Modules:

1. Login
2. Display train list
3. Search for train
4. Reservation
5. Cancellation
6. Train Status

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Here is a template for an empty Software Requirements Specification (SRS) document. You can fill in the details based on your specific project requirements.

[Project Name]

Document Information

- Document Version:[Version Number]
- Date: [Date]
- Authors: [List of Authors]
- Project Sponsor: [Name of Project Sponsor]
- Project Manager: [Name of Project Manager]

1. Introduction

1.1 Purpose

[Describe the purpose of the software, including its goals and objectives.]

1.2 Scope

[Define the scope of the project, outlining the features and functionalities that will be included.]

2. System Overview

2.1 System Description

[Provide a brief description of the overall system, its components, and how it fits into the existing environment.]

3. Functional Requirements

3.1 [Module Name]

3.1.1 Description

[Provide a brief description of the module.]

3.1.2 Features

[List the features and functionalities of the module.]

3.1.3 Use Cases

[Describe the various use cases related to the module.]

3.2 [Next Module Name]

[Repeat the structure for each module in the system.]

4. Non-Functional Requirements

4.1 Performance
[Specify performance-related requirements, such as response times, scalability, etc.]

4.2 Security
[Outline security requirements, including user authentication, data encryption, etc.]

5. User Interfaces

5.1 [Interface Name]

5.1.1 Description
[Describe the purpose and functionality of the interface.]

5.1.2 Screenshots
[Include screenshots or mockups if available.]

5.2 [Next Interface Name]

[Repeat the structure for each user interface in the system.]

6. Constraints

[Identify any constraints or limitations that might affect the development or implementation of the software.]

7. Glossary

[List key terms and their definitions to ensure clarity and understanding.]

Remember to customize this template by replacing the placeholders with actual details related to your project. The SRS document is a living document that evolves as the project progresses, so regular updates and reviews are essential.

Sample(This is a basic template)

Software Requirements Specification

Project: (Case Study: Online Learning Platform)

1. Introduction

1.1 Purpose

The purpose of the Online Learning Platform is to provide an interactive and user-friendly environment for students and instructors to facilitate remote education.

1.2 Scope

The Online Learning Platform will include features for course creation, enrollment, live lectures, quizzes, and discussion forums.

2. System Overview

2.1 System Description

The system will consist of a web-based platform accessible by students and instructors. It will include user authentication, a dashboard for course management, and interactive tools for learning.

3. Functional Requirements

3.1 User Authentication

1. Users (students and instructors) must register with a valid email and password.
2. User authentication will be secure and follow industry standards.

3.2 Course Management

1. Instructors can create, edit, and delete courses.
2. Courses must have a title, description, and cover image.
3. Courses can be categorized by subject and level.

3.3 Enrollment

1. Students can browse available courses.
2. Students can enroll in courses of interest.
3. Instructors can view and manage enrolled students.

3.4 Live Lectures

1. Instructors can schedule live lectures.
2. Students can join live lectures via video conferencing.

3. Recordings of live lectures will be available for later viewing.

3.5 Quizzes

1. Instructors can create quizzes with multiple-choice questions.
2. Students can take quizzes within a specified timeframe.
3. Automated grading for quizzes.

3.6 Discussion Forums

1. Each course will have a discussion forum for students and instructors.
2. Users can post questions, replies, and engage in discussions.

4. Non-Functional Requirements

4.1 Performance

1. The platform must handle at least 1000 simultaneous users.
2. Response time for actions (e.g., enrolling in a course) should be under 3 seconds.

4.2 Security

1. User data must be encrypted during transmission.
2. Access to sensitive data, such as quizzes and grades, should be role-restricted.

5. User Interfaces

5.1 Login Page

- A simple and intuitive login page with fields for email and password.

5.2 Dashboard

- A personalized dashboard for each user displaying enrolled courses, upcoming lectures, and notifications.

6. Constraints

- The platform will be developed using HTML, CSS, JavaScript, and the Django framework.
- Compatibility with modern web browsers (Chrome, Firefox, Safari).

7. Glossary

- **Instructor:** A user with the ability to create and manage courses.
- **Student:** A user enrolled in one or more courses. In real-world scenarios, an SRS document would go through multiple iterations with input from stakeholders, detailed use cases, and additional documentation for testing, deployment, and maintenance.

UNIFIED MODELING LANGUAGE (UML)

1. Model

A model is a simplification of reality.

A model provides the blueprints of a system.

A model may be structural, emphasizing the organization of the system, or it may be behavioral, emphasizing the dynamics of the system.

We build models so that we can better understand the system we are developing.

We build models of complex systems because we cannot comprehend such a system in its entirety**.**

Through modeling, we achieve four aims.

Models help us to visualize a system as it is or as we want it to be. Models permit us to specify the structure or behavior of a system.

1. Static Diagrams
   - a) Use case diagrams
   - b) Class diagrams
   - c) Object diagrams
   - d) Component diagrams
   - e) Deployment diagrams
2. Dynamic diagrams
   - a) Interaction diagrams
     - i) Sequence diagrams
     - ii) Collaboration diagrams
   - b) State machine diagrams
   - c) Activity diagrams Applications of UML:

Models give us a template that guides us in constructing as system.

Models document the decisions we have made

2. Principles of Modeling

The choice of what models to create has a profound influence on how a problem is attacked and how a solution is shaped

Every model may be expressed at different levels of precision The best models are connected to reality

No single model is sufficient. Every nontrivial system is best approached through a small set of nearly independent models

UML is a graphical notation used to visualize, specify, construct and document the artifact of software intensive. UML is appropriate for modeling systems ranging from Enterprise Information Systems to Distributed Web-based Application and even to Hard Real-time Embedded systems. UML effectively starts with forming a conceptual modeling of the language.

There are 2 types of diagrams. They are

UML is intended primarily for software intensive systems. It has been used effectively for such domains as

1. Enterprise Information Systems
2. Banking and Financial Services
3. Telecommunications
4. Transportation
5. Defense and Aerospace
6. Retail
7. Medical Electronics
8. Scientific
9. Distributed Web-based Services

Basic building blocks of UML:

The building blocks of UML can be categorized as

1. Things

2. Relationships and

3. Diagrams

Things:- Things are the most important building blocks of UML. Things can be

a) Structural

b) Behavioral

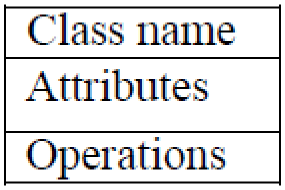
c) Grouping

d) Annotational

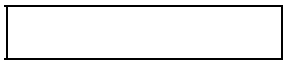
a) Structural Things: They define the static part of the model. They represent physical and conceptual elements.

Following are the structural things –

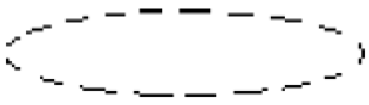
1. *Class*: - It describes a set of objects that share the same attributes, operations, relationships and semantics.



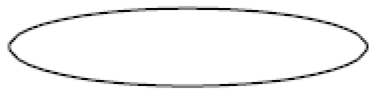
2. *Object*: - It is a collection of operations that specifies a service of a class or a component.



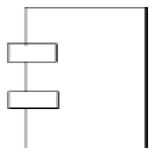
3 *Collaboration*: - It defines interaction between elements.



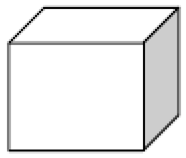
4. *Use case*: - They are used to identify different use case components of a particular software project. It is used to model the operation.



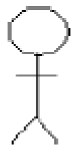
5. *Component*: - It is a physical and replaceable part that confirms to and provides realization of set of interfaces.



6. *Node*: - A physical resource that exists in runtime and represent a computational resource.

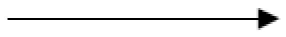


7. *Actor*: - The outside entity that communicates with a system. Typically a person playing a role on an external device.

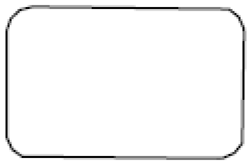


b) Behavioral Things: They consist of dynamic parts of the UML model. The following are behavioral things –

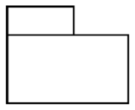
1. *Interaction*: - It is defined as a behavior that consists of a group of message exchanged among elements to accomplish a specific task. message



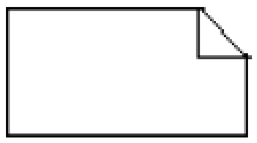
2. *State machine*: - It is useful when the states of an object in its life cycle. It defines the sequence of states and object goes through in response to events.



c) Grouping Things: They can be defined as a mechanism to group elements of UML model together. There is only one grouping thing available i.e., Package.
Package is used for gathering structural and behavioral things



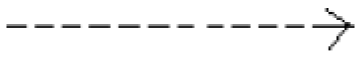
d) Annotational Things: - They can be defined as a mechanism to capture remarks, description and comments of UML model elements. There is only one annotational thing available i.e., Note.
Note is used to render comments, constraints and so on of a UML element.



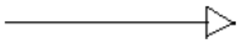
Relationships: -

The relationship is another most important building block of UML. They show how elements are associated with each other and their association describes the functionality of application.
There are 5 types of relationships. They are

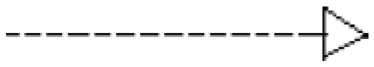
1. Dependency: It is a relationship between two things in which change in one element also affects another



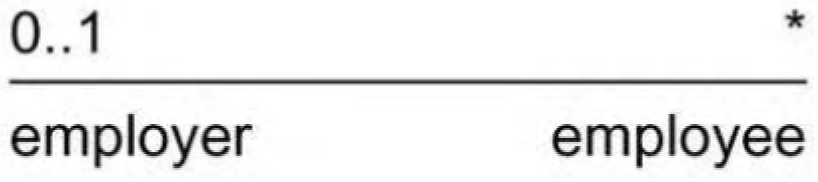
2. Generalization: It can be defined as a relationship which connects a specialized element with a generalized element. It basically describes inheritance relationship in the object. It is a ‘is_a ‘ hierarchy.



3. Realization: It can be defined as a relationship in which two elements are connected. One element describes some responsibility which is not implemented and the other one implement then. This relationship exists in case of interfaces.



4. Association: It is a set of links that connects elements of an UML model.



***Following are Sample Reference UML Diagrams of Banking Applications:

USE CASE DIAGRAM

A use case diagram describes a set of sequences in which each sequence indicates the relation with outside things. A use case involves the interaction of actor and system. There exist 3 types of relationships-

1. Association

2. Dependency

3. Generalization

Use case diagrams can contain

• Actors – "things" outside the system

• Use cases – system boundaries identifying what the system should do.

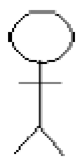
Use case diagram can be used during analysis to capture the system requirements and to understand how the system should work. During the design phase, you can use use-case diagrams to specify the behavior of the system as implemented.

Actor:- An actor represents system users. They help delimit the system requirements and give a clearer picture of what the system should do. An actor is someone or something that

• Interacts with or uses the system.

• Provides input to and receives information from the system.

• Is external to the system and has no control over the use cases.



Customer (actor)

Actors are discovered by examining,

• Who directly uses the system

• Who is responsible for maintaining the system?

• External hardware used by the system.

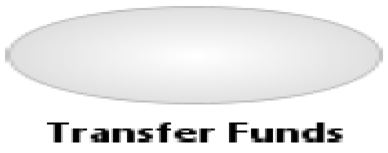
• Other systems that need to interact with the system.

Use case: - A use case can be described as a specific way of using the system from users (actors) perspective. Use case can be characterized as-

• A pattern of behavior the system exhibits.

• A sequence of related transactions performed by an actor and the system.

• Delivering something of value to the actor.



Note: Use cases often start with a “verb”.

Use cases provide a means to,

• Capture system requirements.

• Communicate with end users and domain experts.

• Test the system.

Every graphical representation has a textual description. The description of each use case is written in a use case specification. Use Case specification has :

Precondition – which states how and when the use case starts?

Main Flow – which lists the set of actions performed by the use case?

Alternate Flow – which lists the exceptions that are possible during executing the use case?

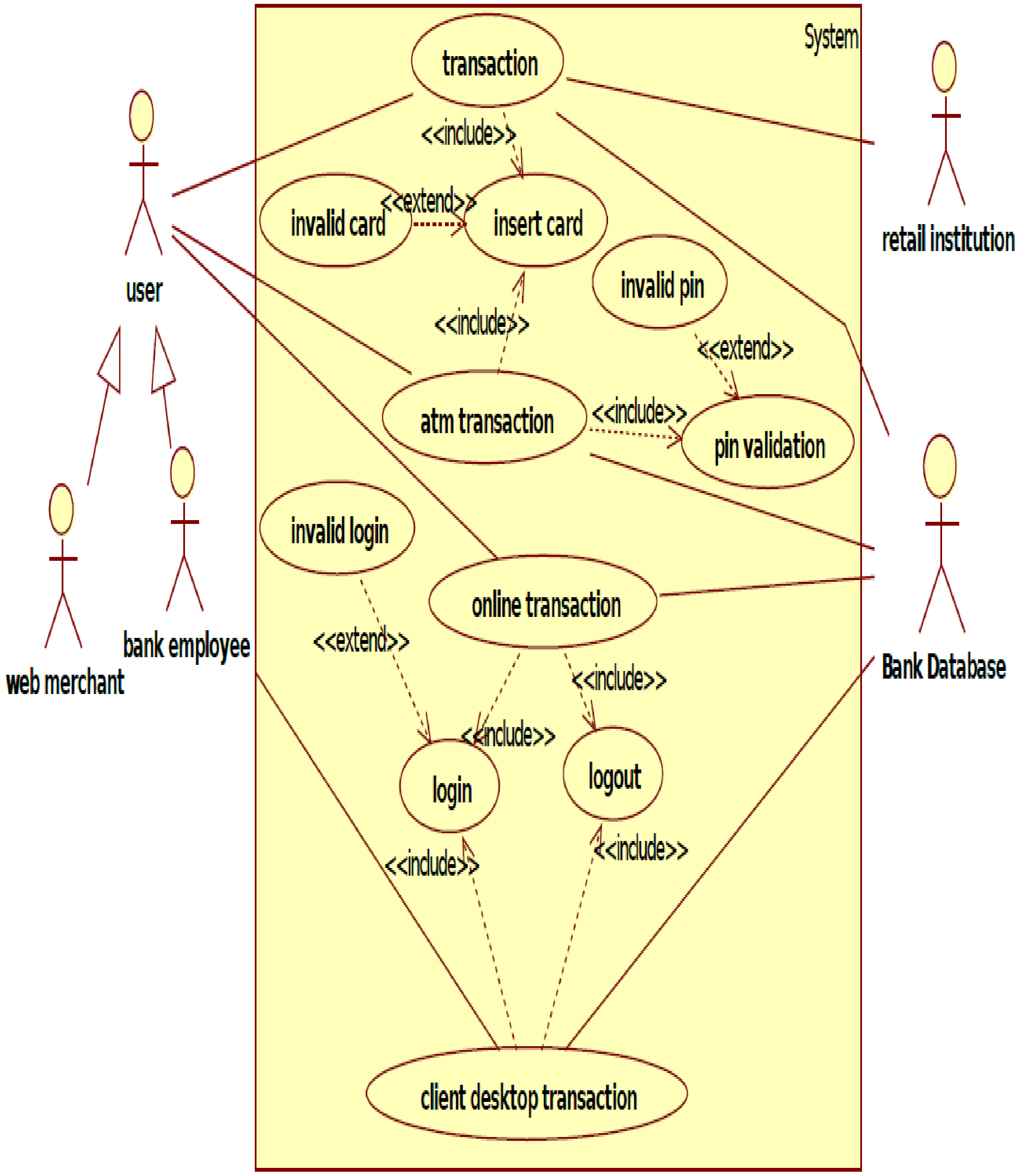
Post condition – which shows the result after the use case completes successfully.

Pseudo code

1. Right click on the model

2. Select Add Diagram – Use case diagram

Use –Case Diagram for Banking Application

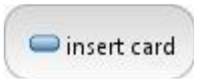


ACTIVITYDIAGRAM

An activity diagram is a special case of state diagram. An activity diagram is like a flow Machine showing the flow a control from one activity to another. An activity diagram is used to model dynamic aspects of the system.

Activity diagram contains: 1.Activity states and action states 2.Transition

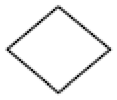
Action state:- These are atomic, executable computation which represents the execution of an action.



Activity state:- They can be decomposed. That is, their activity is represented by other activity diagrams.

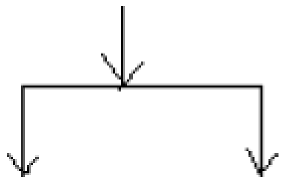
Branching:- In branching, we have one incoming transition and two or more outgoing transitions.

Decision

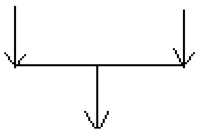


Decision

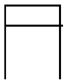
Forking:- It is a process of splitting a single flow of control into multiple flow of controls. Generally a fork has a single incoming flow of control but multi outgoing flow of control.



Joining:- It is exactly opposite of forking. It generally has multiple incoming flows of control but single outgoing flow of control.



Swim-lanes:- They represent the columns in the activity diagram to group the related activities. These are represented in the form of partitioned region.Swim-lanes are helpful when modeling a business work flow because they can represent organizational units or role with in a business model. Swim-lanes are very similar to an object because they provide a way to tell who is performing a certain role.

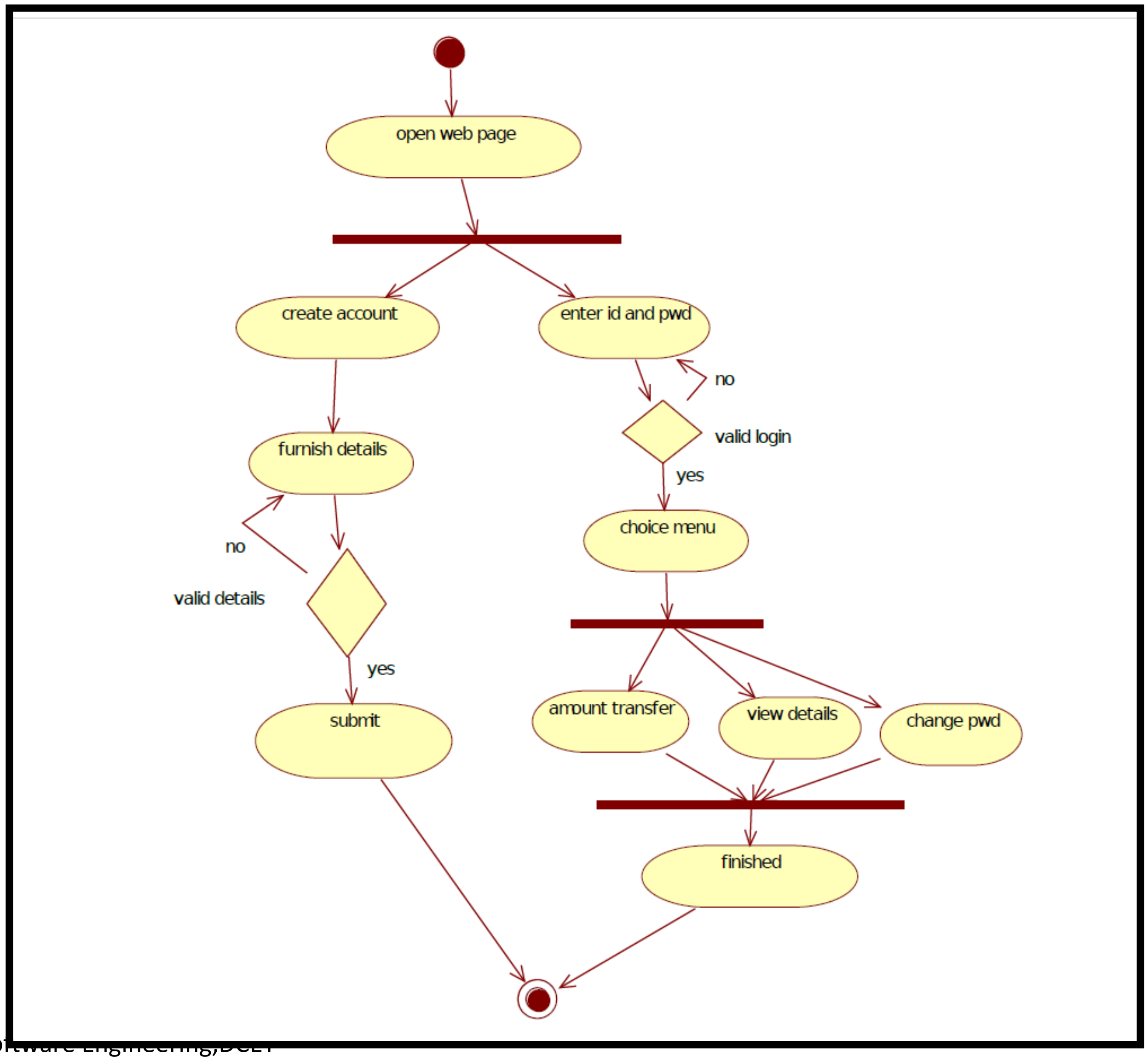


Pseudo code

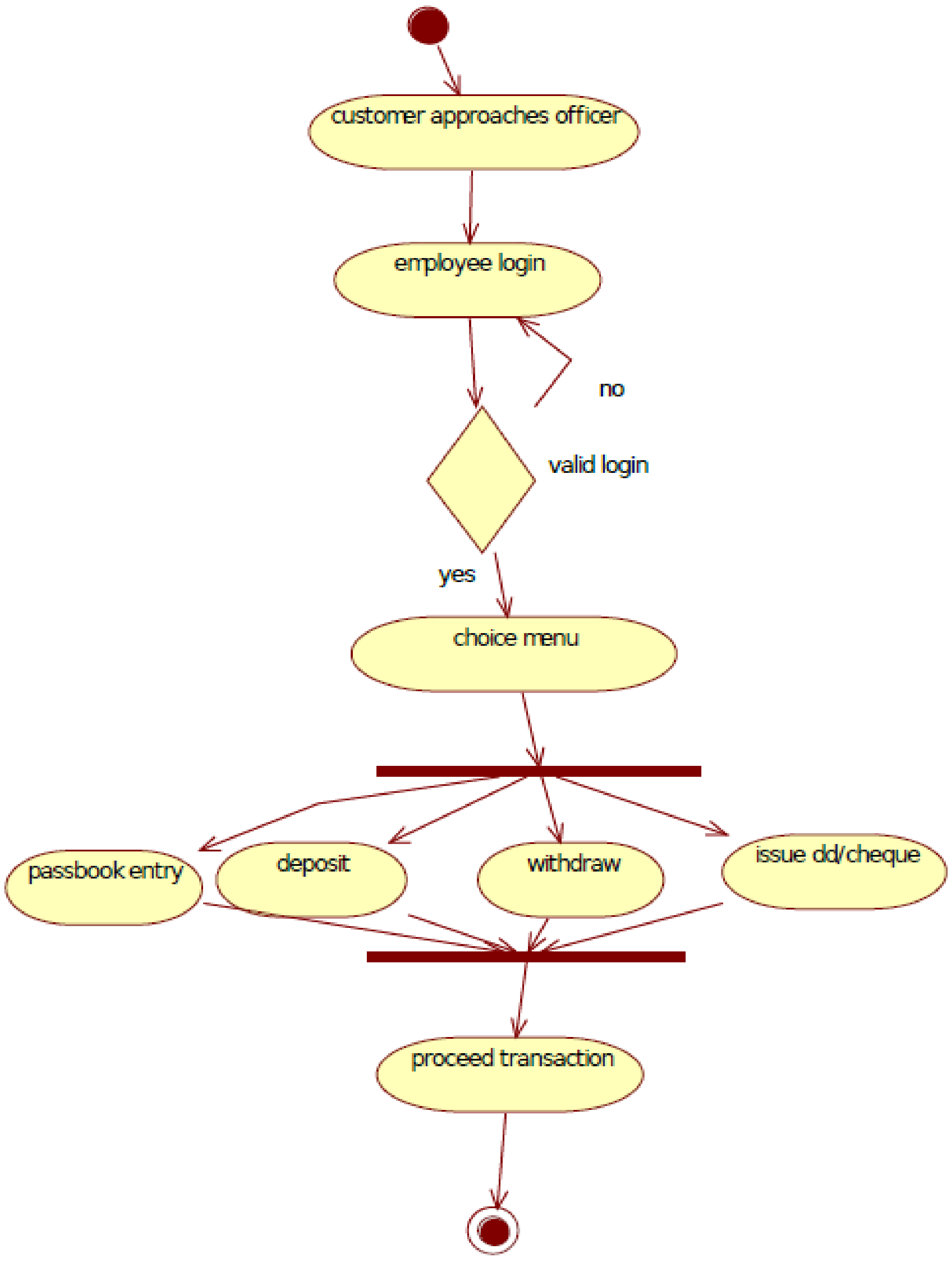
1. Right click on the model

2. Select Add Diagram – Activity diagram

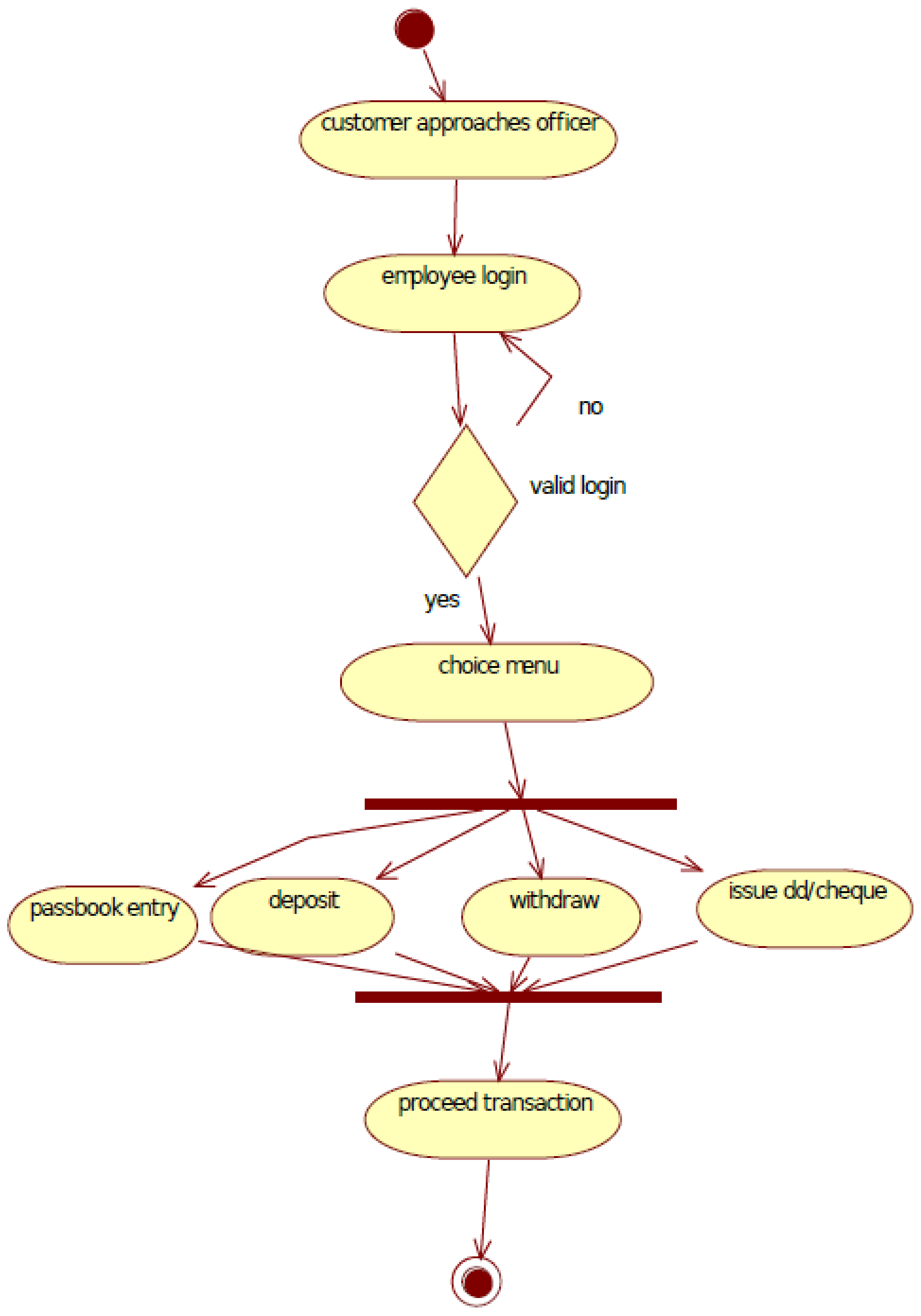
Activity Diagram of Banking Application



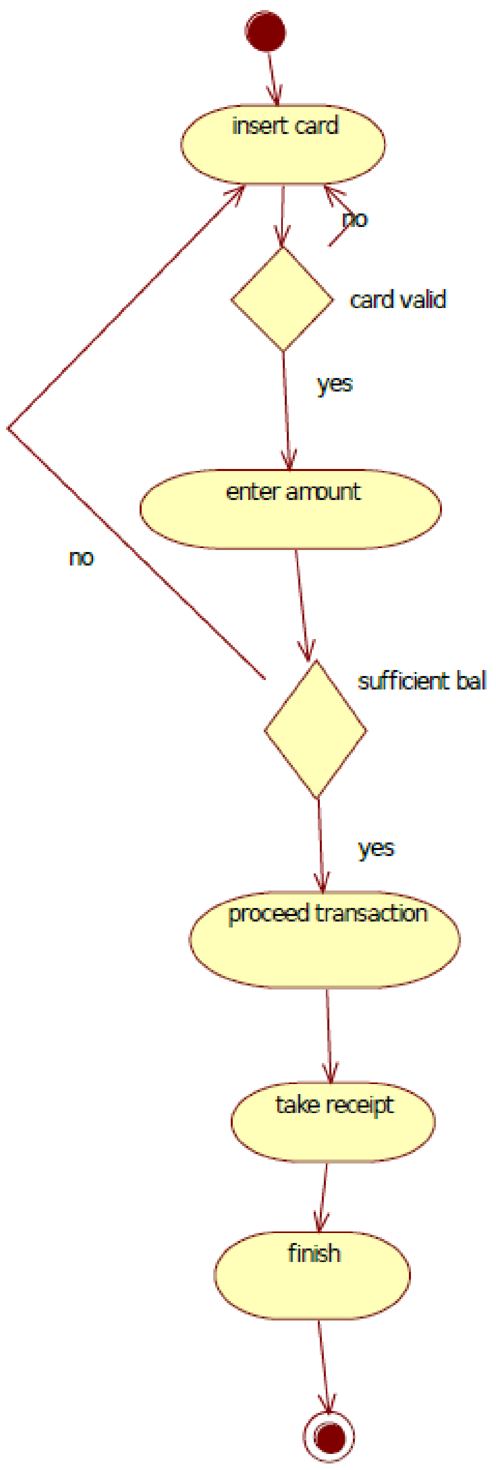
Client Desktop Transaction Activity Diagram:



ATM Transaction Activity Diagram:



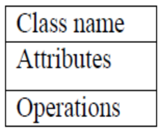
Web Merchant Transaction Activity Diagram:



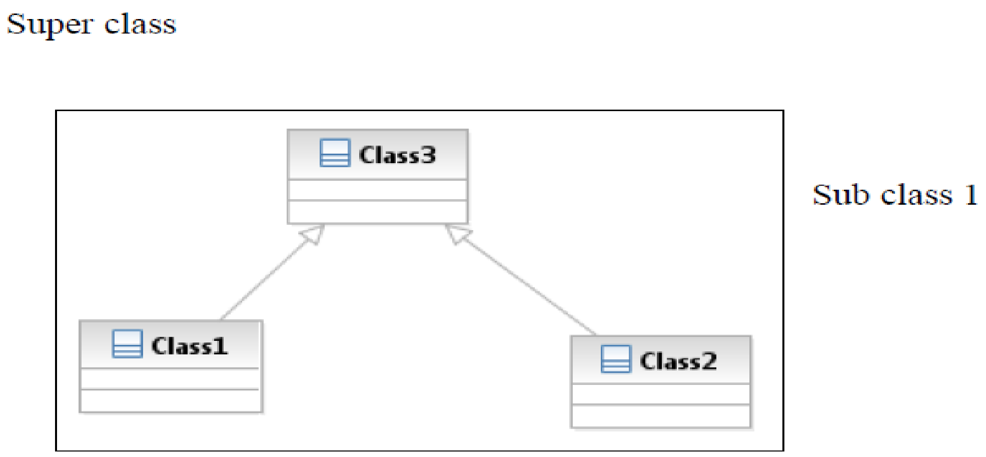
CLASS DIAGRAM

Class diagram contains icons representing classes, interfaces and their relationships.

Class:- A class is a set of objects that share a common structure and common behavior (the same attributes, operations and semantics). A class is a abstraction of real-world items. When these items exist in the real-world, they are instances of the class and are referred to as objects.



Generalization relationship for classes: It shows that sub classes share the structure or behavior defined in one or more super classes. Use a generalize relationship to show "is_a" relationship.



Dependency Relationship: The dependency is a relationship between two model elements in which change in one element will affect the other model element. Typically in class diagrams, a dependency relationship indicates that the operations of the client invoke operation of the supplier.

Cardinality Adornment:- Cardinality specifies how many instances of one class may be associated with single instance of other class. When you apply a cardinality adornment to a class, you are indicating number of instances allowed for that class. A relationship, you are indicating number of links allowed between one instance of a class and the instances of another class.

Valid Values:	**Value Description**
0..0	Zero
0..1	zero or one
0..n	zero or more
1..1	One
1..n	one or more
N	unlimited number

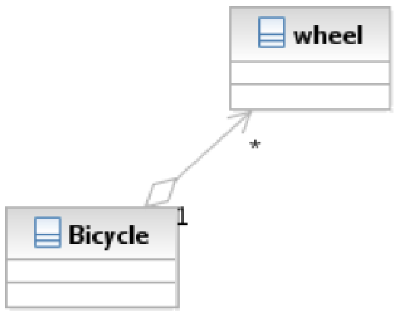
Interface:- An interface specifies the externally visible operations of a class and/or component, and has no implementation of its own. An interface specifies only a limited part of behavior of class or a component.

Association relationship: An association provides a pathway of communications. The communication can be between use cases, actors, classes or interfaces. If two classes are usually considered independently, the relationship is an association.

Unidirectional
Bi directional

An association is an orthogonal or straight solid line.

Aggregate relationship: Use the aggregate relationship to show a whole or part relationship between two classes.



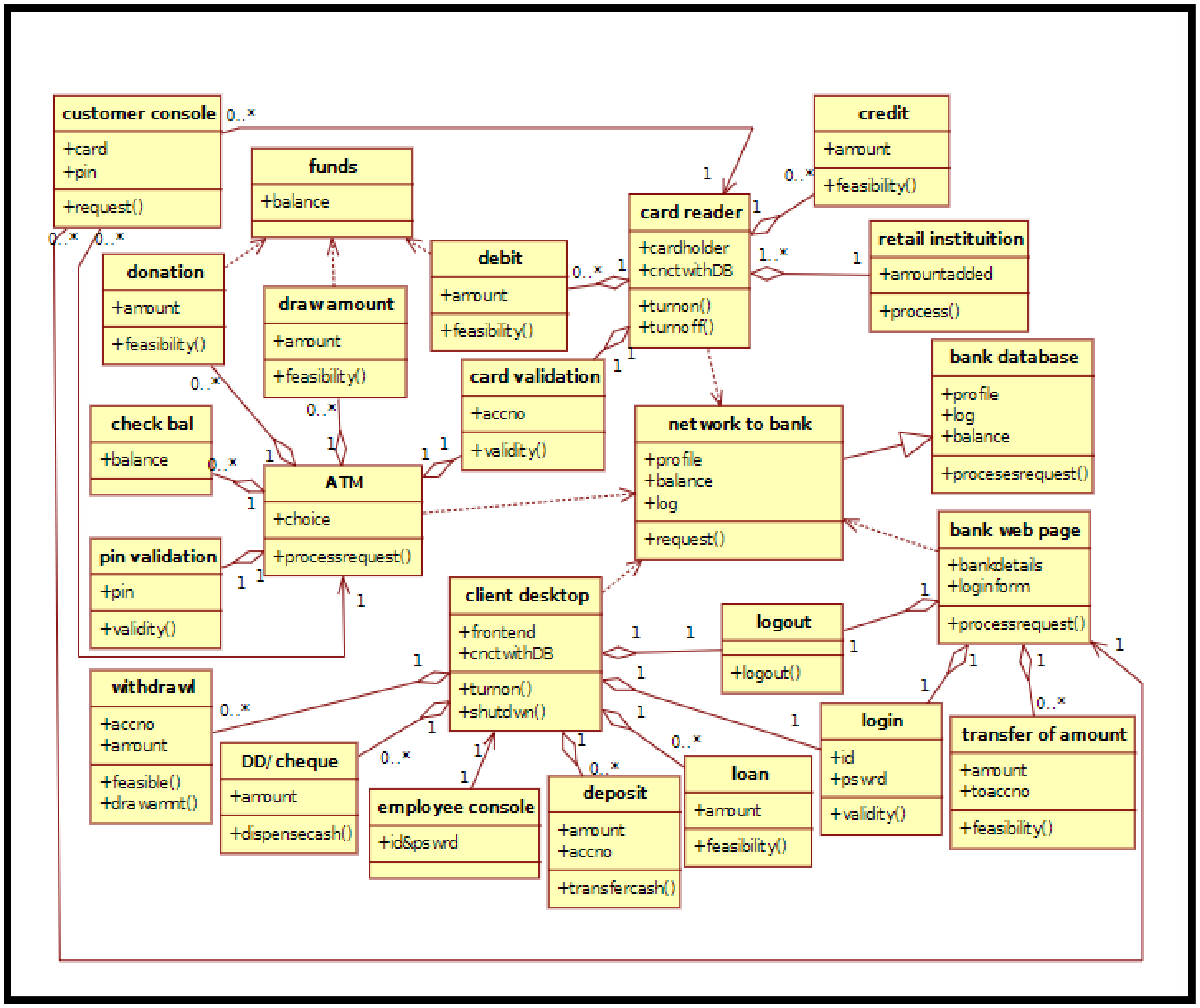
The diamond end represents the class which is whole.

Pseudo code

1. Right click on the model

2. Select Add Diagram – Class diagram

Class Diagram of a Banking Application



OBJECT DIAGRAM

Object diagrams model the instances of things contained in the class diagrams. Object diagrams show a set of objects and their relationships at a point in time. They are used to model the static design view of a system.

Object diagrams contain:

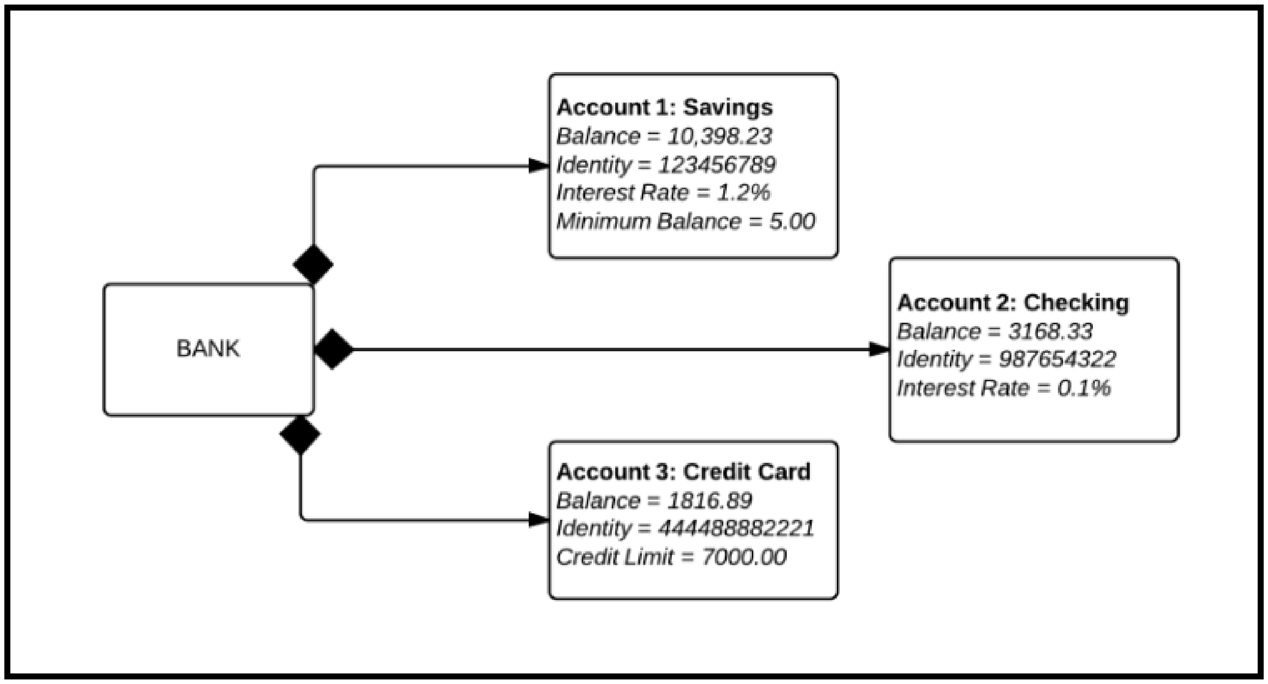
- • Objects
- • Links

Pseudo code

1. Right click on the model

2. Select Add Diagram – Object diagram

Object Diagram for Banking Appplication



INTERACTION DIAGRAM

An interaction is an important sequence of interactions between objects. There are two types of interaction diagrams,

1. Sequence Diagrams.

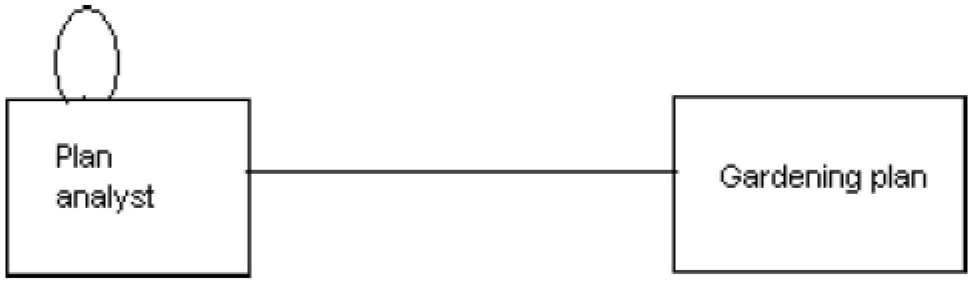
2. Collaboration diagrams.

1. **Sequence Diagram *:*** A sequence diagram is a graphical view of a scenario that shows object interaction in a time based sequence, what happens first, what happens next.

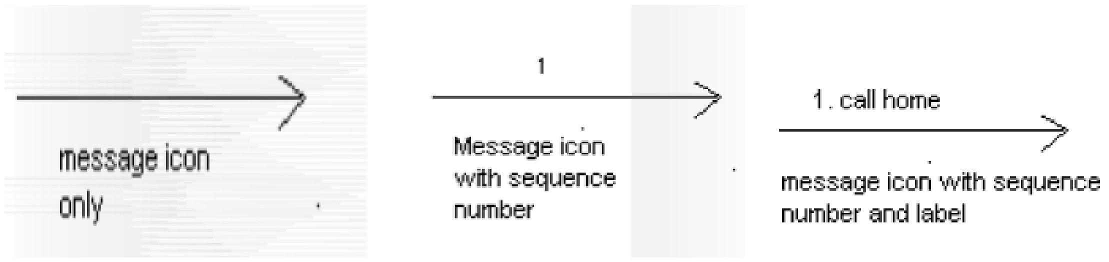
Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interfaces.

A sequence diagram has two dimensions: vertical placement represents time and horizontal placement represents different objects.

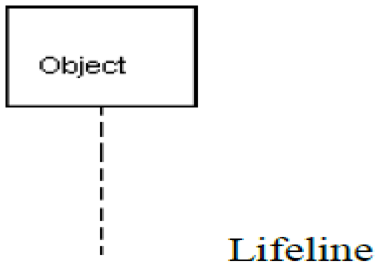
Link: Objects interact through their links to other objects. A link is an instance of an association, analogous to an object being instance of a class. A link should exist between two objects, including class utilities, only if there is a relationship between their corresponding classes.



Message icons: A message icon represents the communication between objects indicating that an action will follow. The message icon is a horizontal, solid arrow connecting two lifelines together.A message icon in a sequence diagram represents exactly one message.


message icon only
1
Message icon with sequence number
1. call home
message icon with sequence number and label

Lifeline: Each object appearing on the sequence diagram contains a dashed vertical line, called lifeline, which represents the location of an object at a particular point in time. The lifeline also serves as a place for messages to start and stop and a place for the focus of control to reside.



Message or Event: a message is a communication carried between two objects that trigger an event. A message is represented in collaboration and sequence diagrams by a message icon which usually indicates its synchronization.

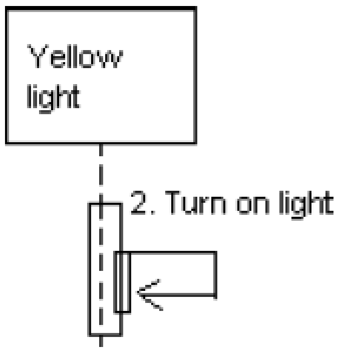
Synchronization types that are supported.

1. Synchronous Call
2. Asynchronous Call
3. Asynchronous Signal
4. Create
5. Delete
6. Reply

Message or Event: a message is a communication carried between two objects that trigger an event. A message is represented in collaboration and sequence diagrams by a message icon which usually indicates its synchronization.

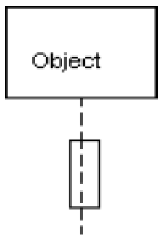
Synchronization types that are supported.

Message to self: It is a tool that sends a message from one object back to the same object. The sender of the message is same as the receiver.

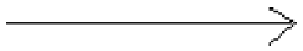


Tools:

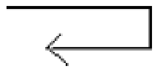
1. Object



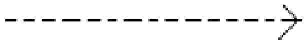
2. Object message



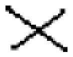
3. Message to self



4. Return message



5. Destruction marker

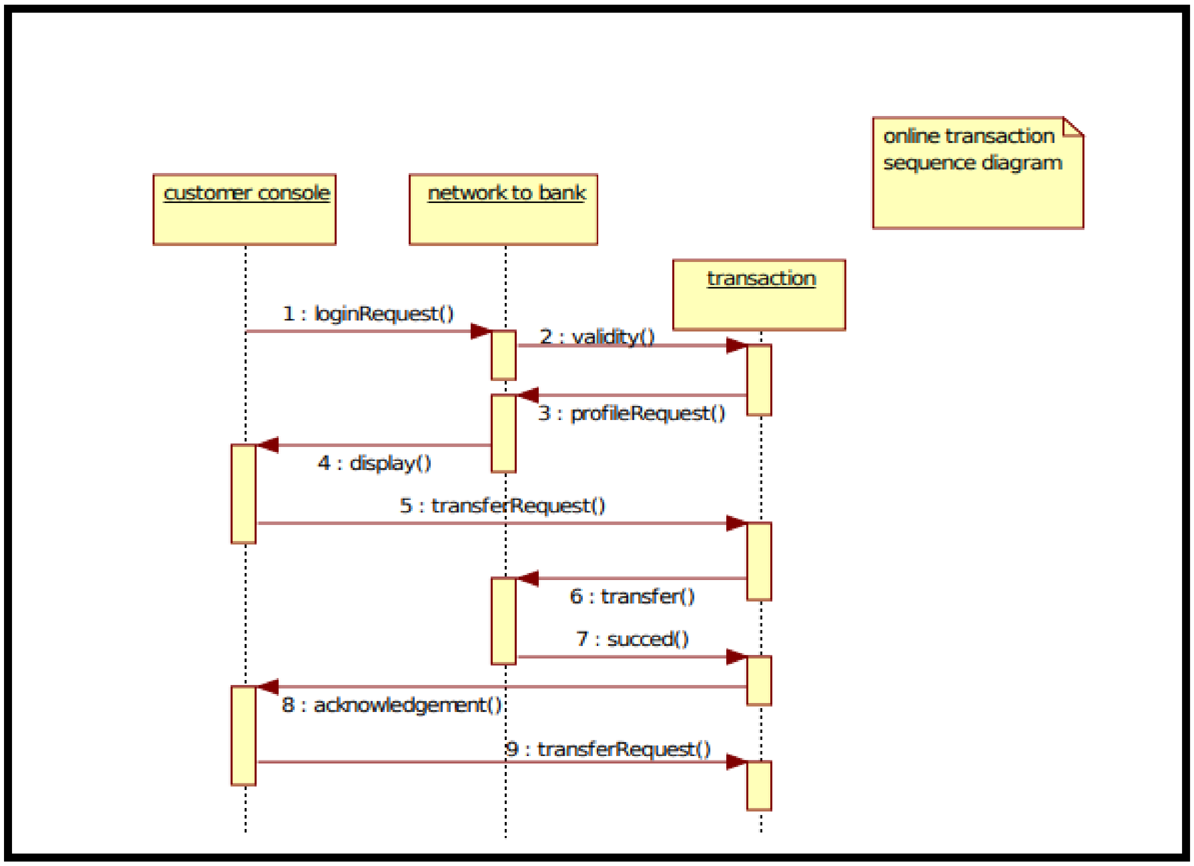


Pseudo code

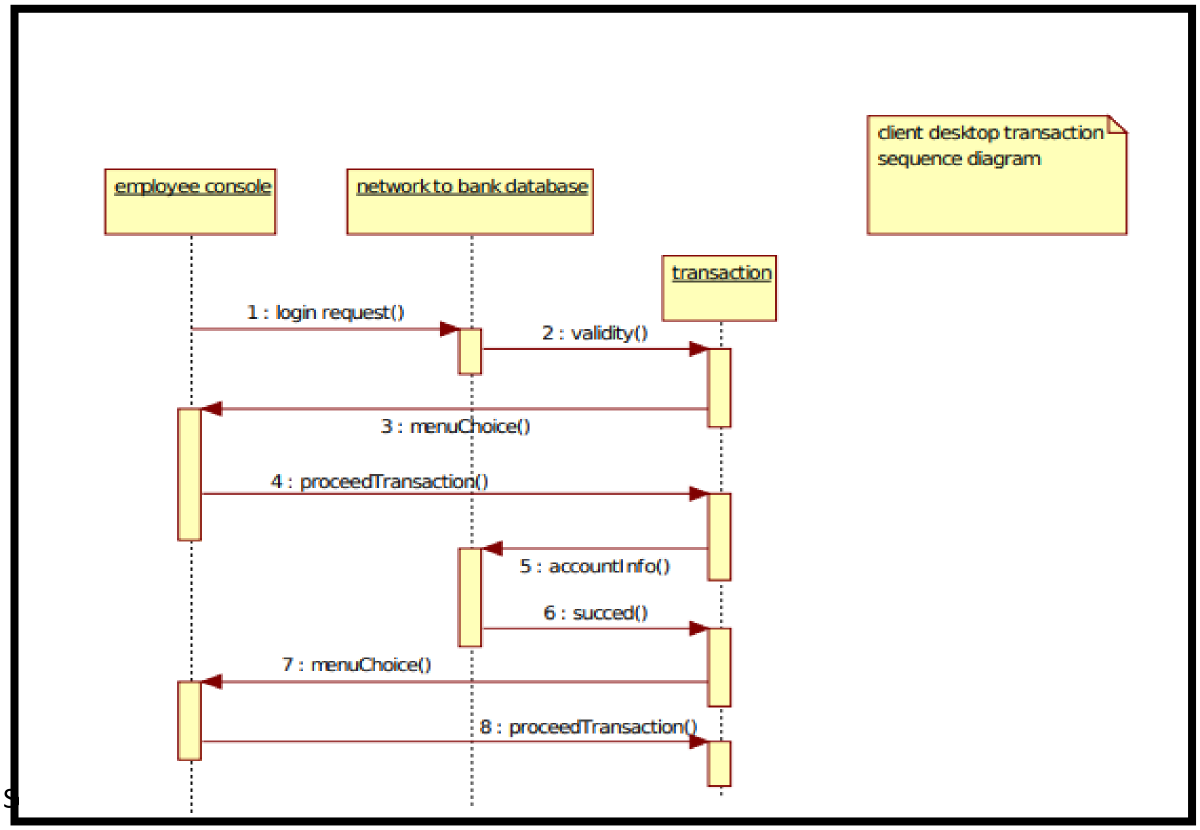
1. Right click on the model

2. Select Add Diagram – Sequence diagram

Online Transaction Sequence Diagram:

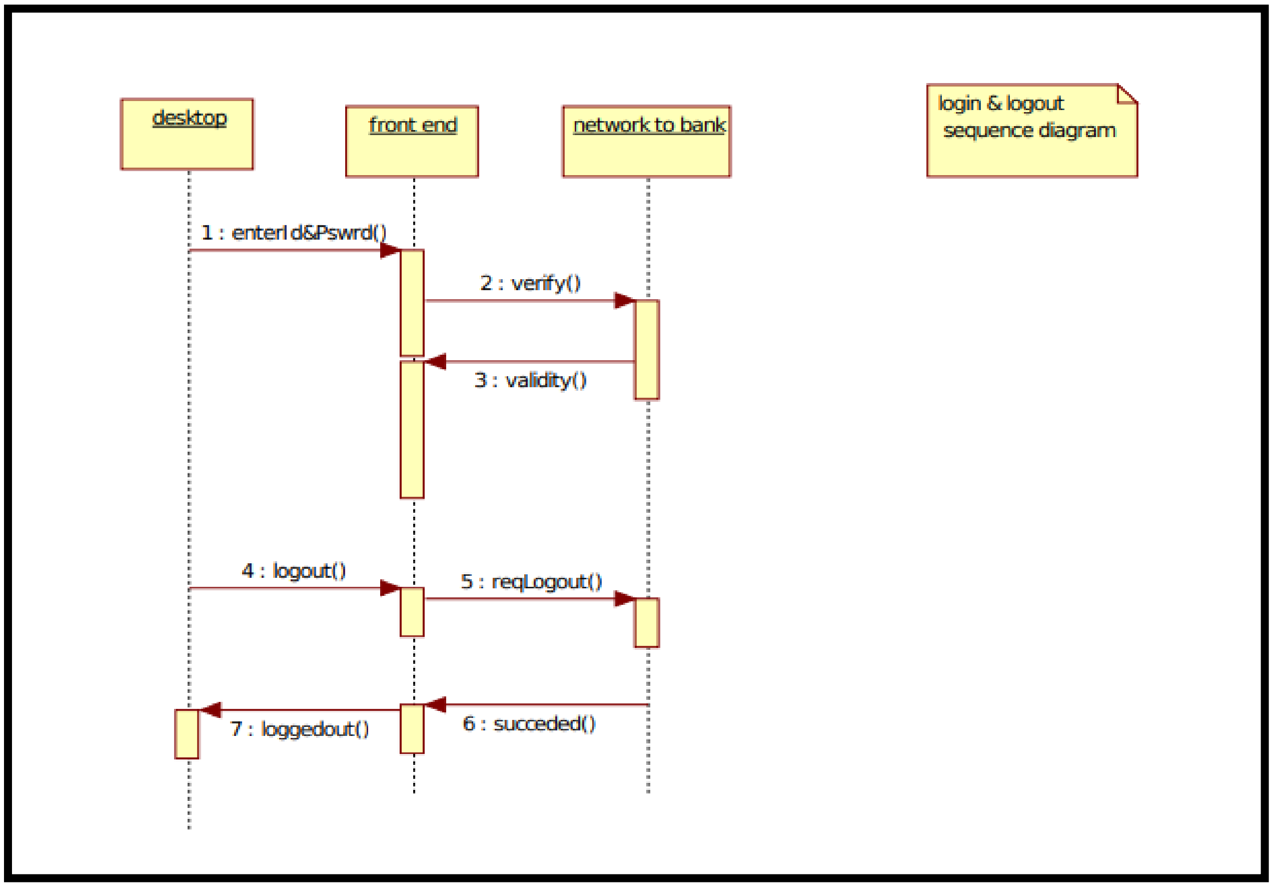


Client desktop Transaction Sequence Diagram:

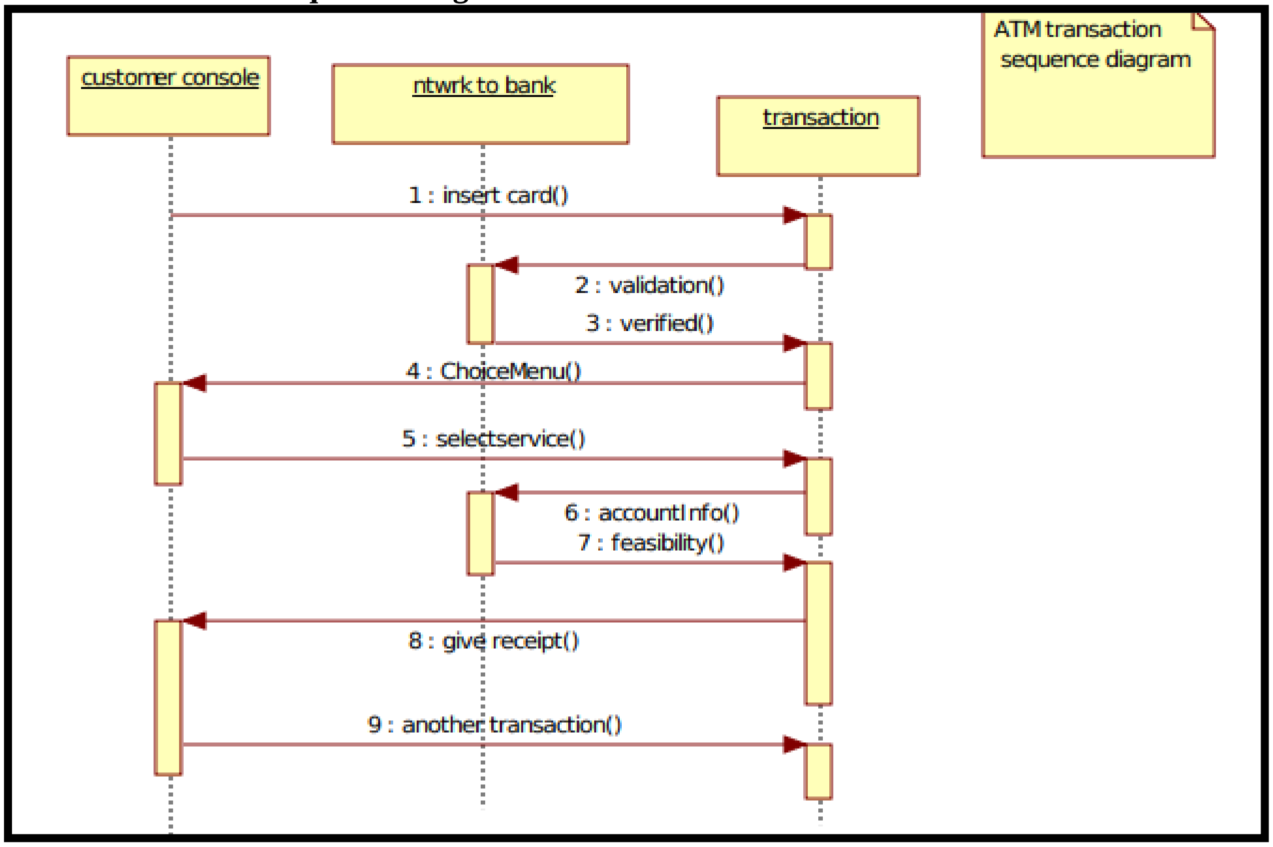


S

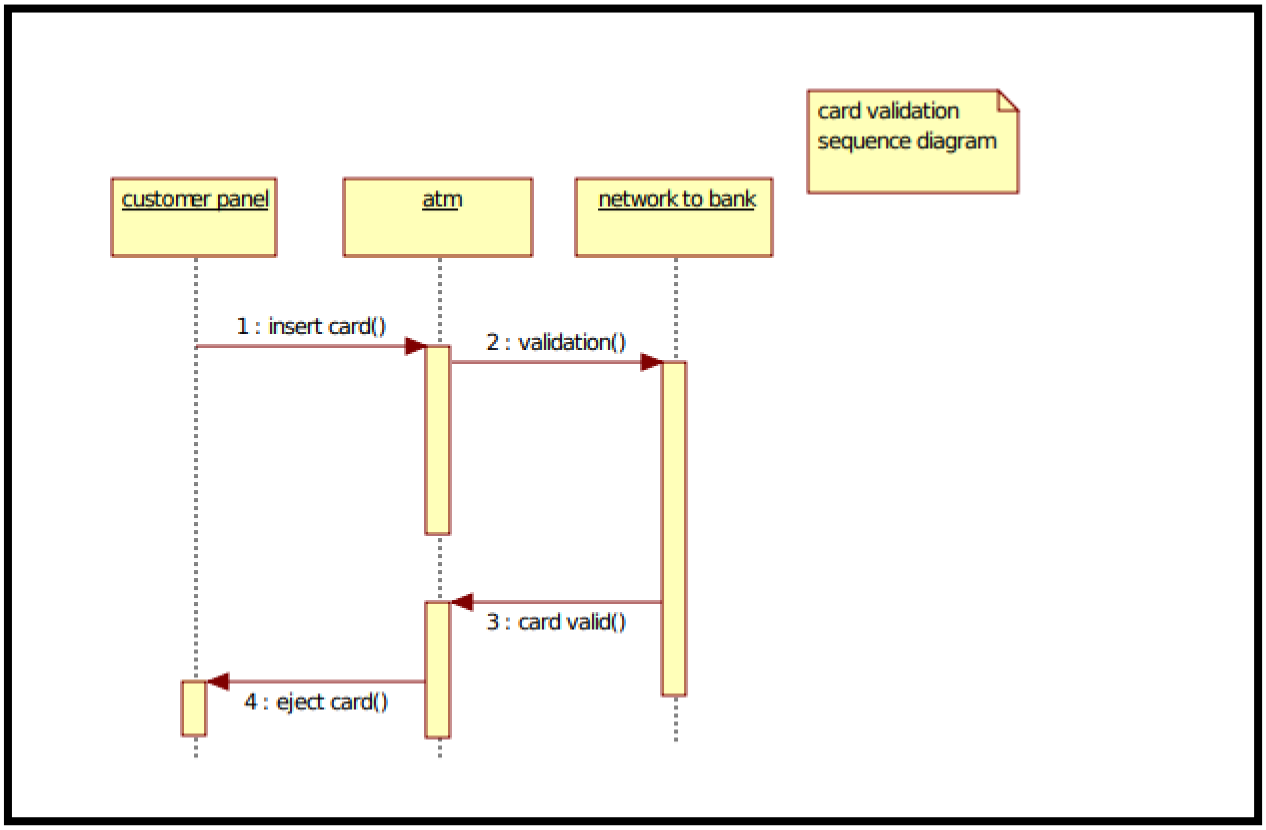
Login & Logout Sequence Diagram:



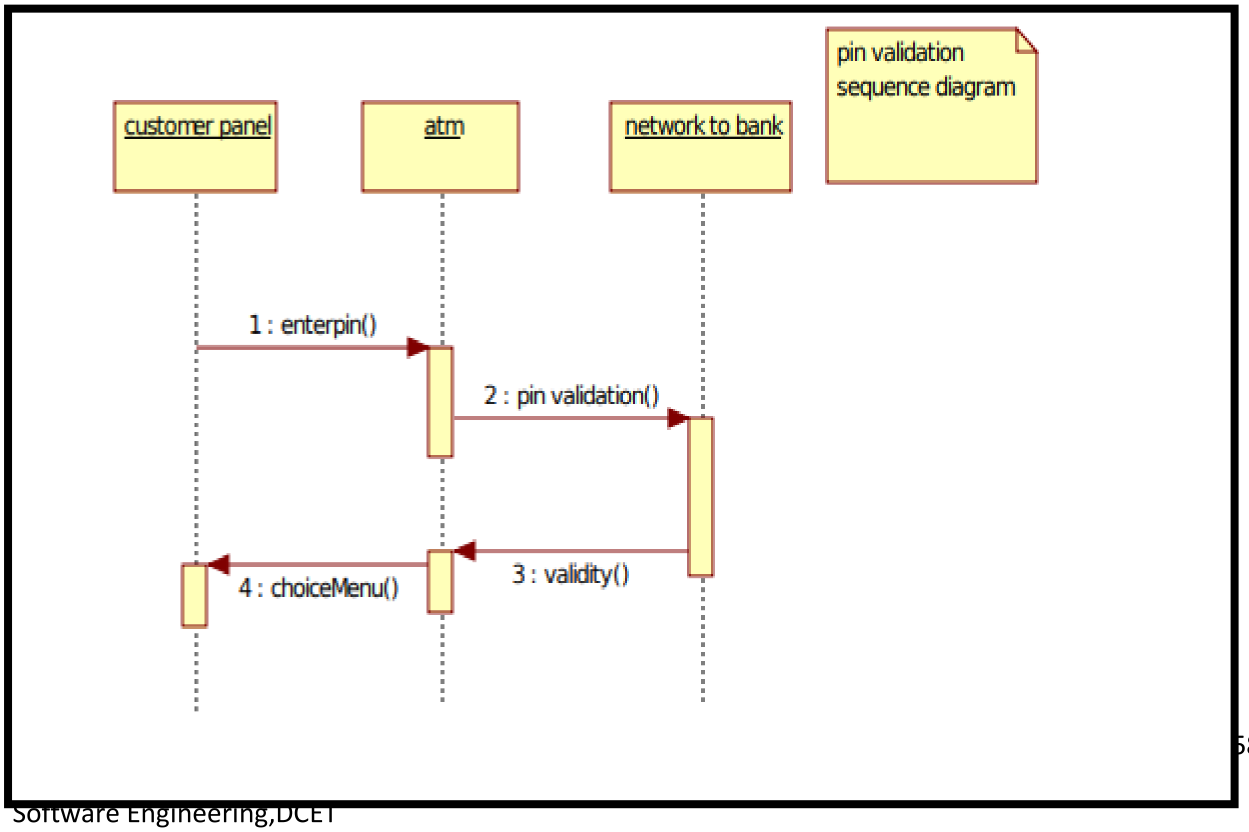
ATM Transaction Sequence Diagram:



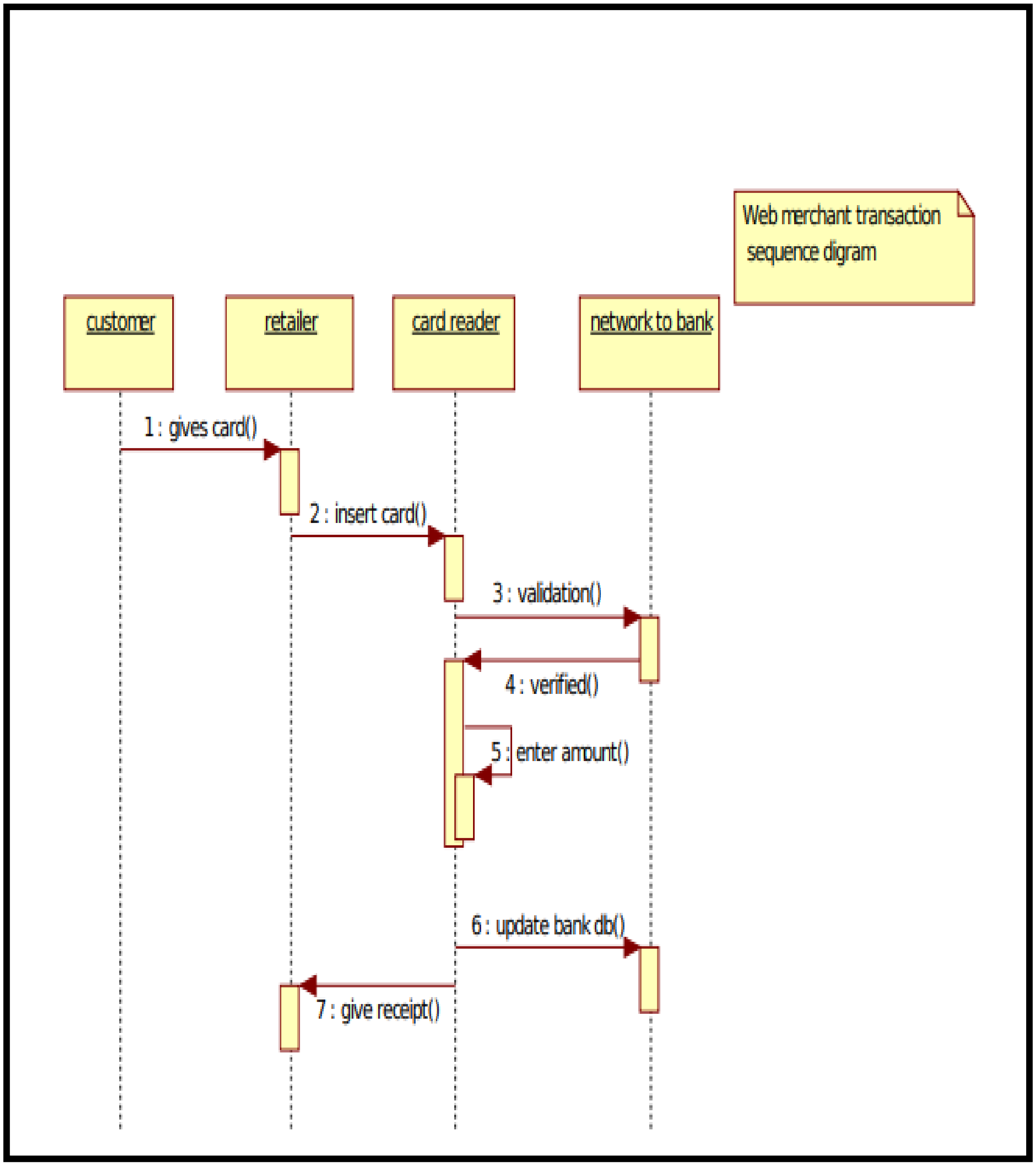
Card Validation Sequence Diagram:



Pin Validation Sequence Diagram



Web merchant Transaction Sequence Diagram

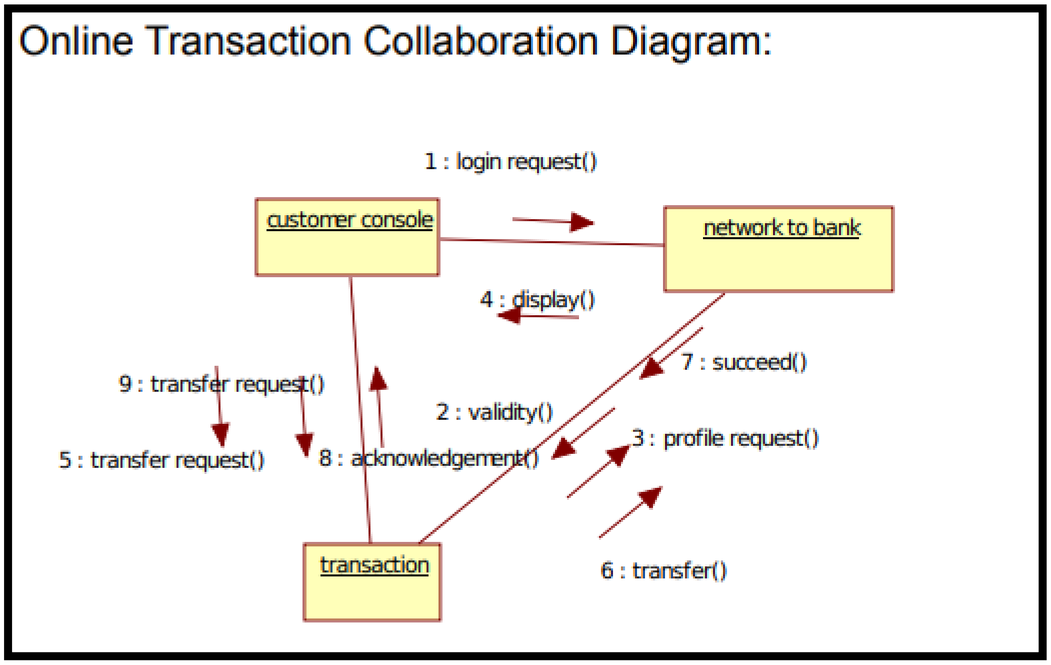


Collaboration diagram : A collaboration diagram is an interaction diagram that shows the order of messages that implement an operation or a transaction. Collaboration diagrams show objects, their links, and their messages. They can also contain simple class instances and class utility instances. Each collaboration diagram provides a view of the interactions or structural relationships that occur between objects and object-like entities in the current model. Collaboration diagrams contain icons representing objects. 1. Right click on the model
2. Select Add Diagram – Communication diagram
Sequence and collaboration diagrams are semantically equivalent as both show the interaction among objects. From one diagram we can generate another diagram. To generate a collaboration diagram from sequence diagram, right click on sequence diagram, select - Add Diagram - communication diagram . Similarly, a sequence diagram can be generated from collaboration diagram.

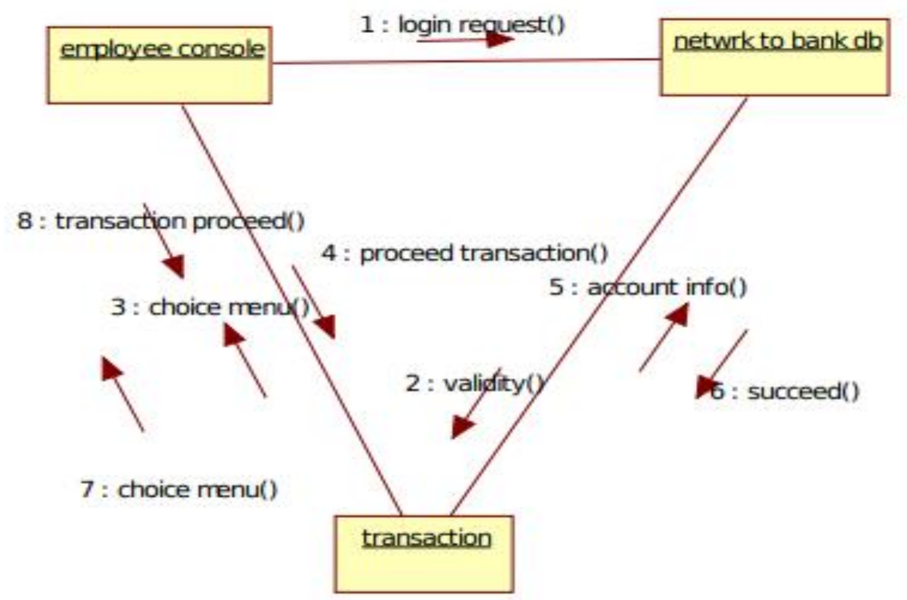
Pseudo code

1. Right click on the model

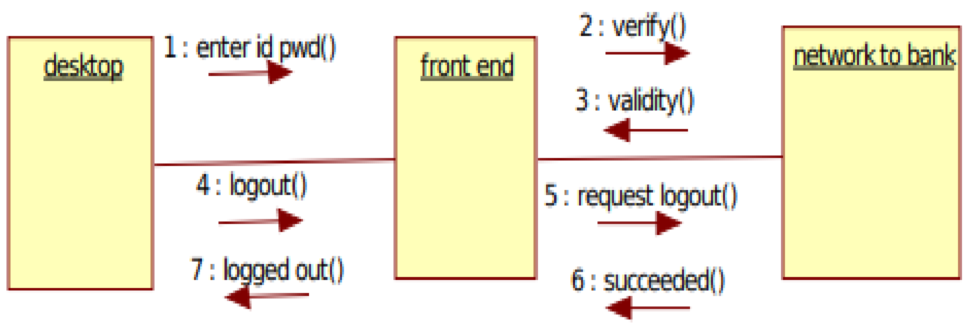
2. Select Add Diagram – Communication diagram



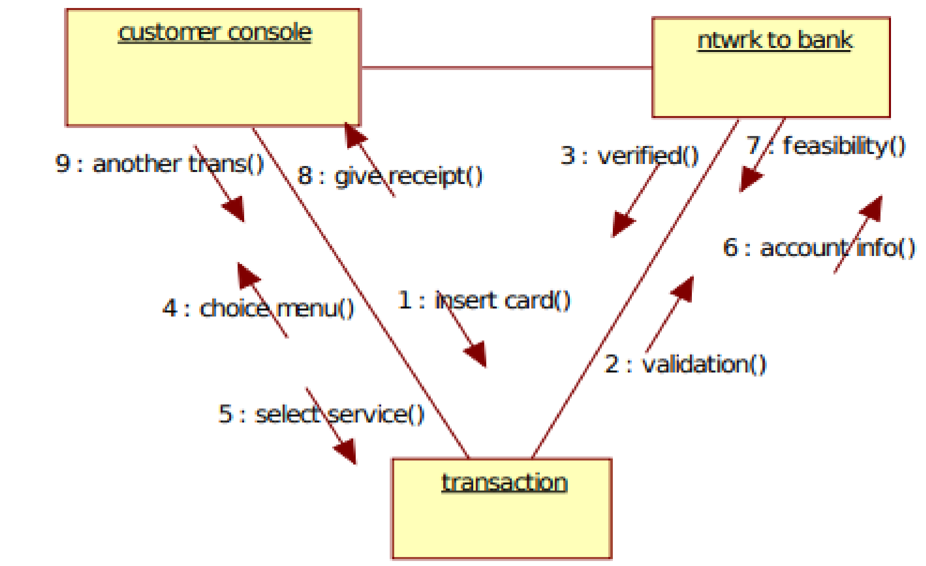
Client desktop Transaction Collaboration Diagram:



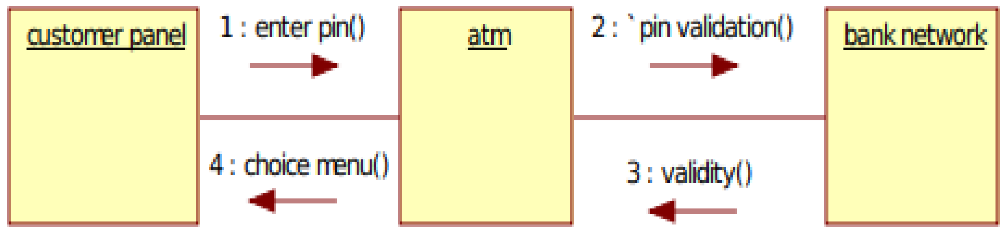
Login & Logout Collaboration Diagram:



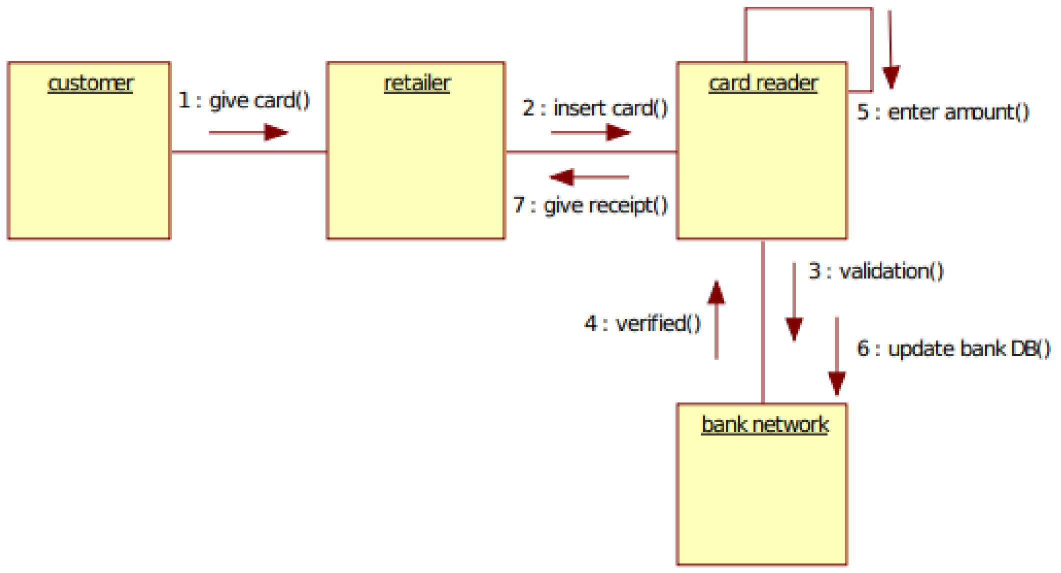
ATM Transaction Collaboration Diagram:



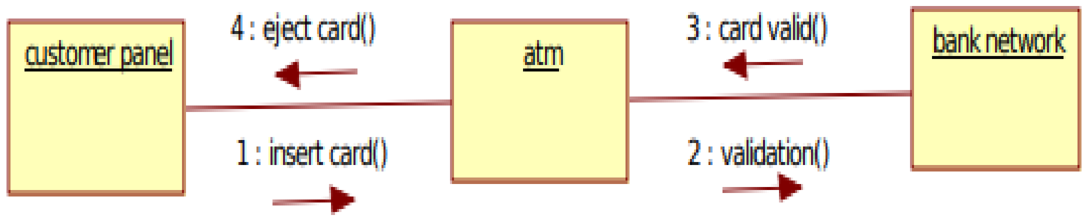
Pin Validation Collaboration Diagram:



Web merchant Transaction Collaboration Diagram:



Card Validation Collaboration Diagram:



STATE MACHINE DIAGRAM

State Machine diagrams model the dynamic behavior of individual classes or any other kind of object. They show the sequence of states that an object goes through the events that cause a transition from one state to another and the actions that result from a state change. A state Machine diagram is typically used to model the discrete stages of an objects lifetime. A state Machine diagram typically contains one start state and multiple end states.

State :- A state represents a condition or situation during the life of an object during which it satisfies some condition or waits for an event. Each state represents a cumulative history of its behavior. States can be shared between state machines. Transitions cannot be shared.

Naming: The name of the state must be unique to its enclosing class,within the state

Actions: Actions on states can occur at one of four times On entry

On exit Do On event

Start state:- A start state (also called an "initial state") explicitly shows the beginning of the execution of the state machine on the state Machine diagram or beginning of the workflow on an activity diagram. Normally, one outgoing transition can be placed from the start state.

However, multiple transitions may be placed on start state, if at least one of them is labeled with a condition. No incoming transitions are allowed.

The start state icon is a small, filled circle that may contain the name (Begin process).

● Begin process

End state:- An end state represents a final or terminal state on an activity or state Machine diagram..

Transitions can only occur into an end state.

The end state icon is a filled circle inside a slightly larger unfilled circle that may contain the name (End process).

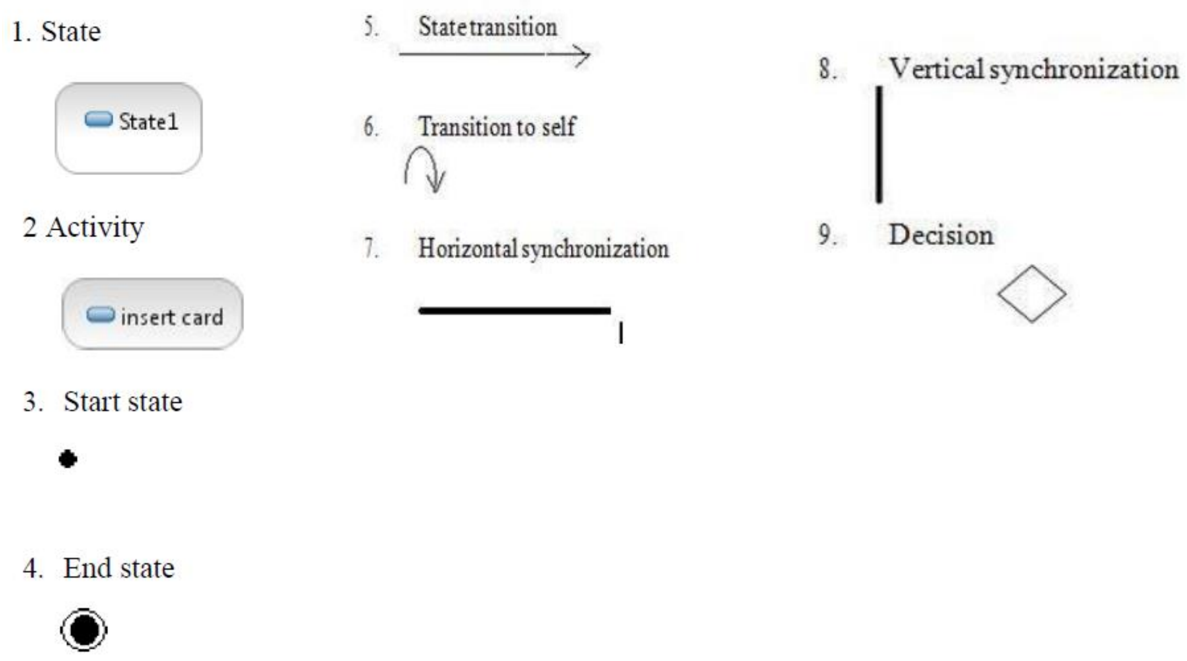
◉ End process

State transition:- A state transition indicates that an action in the source state will perform certain specified actions and enter the destination state when a specified event occurs or when certain conditions are satisfied. A state transition is a relationship between two states, two activities or between an activity or a state. The icon for a state transition is a line with an arrow head pointing toward the destination state or activity.

We can show one or more state transitions from a state as long as each transition is unique. Transitions originating from a state cannot have the same event, unless there are conditions on the event.

Naming: We should label each state transition with the name of at least one event that causes the state transition. We do not have to use unique labels for the state transitions because the same event can cause a transition to many different states or activities.

Tools:

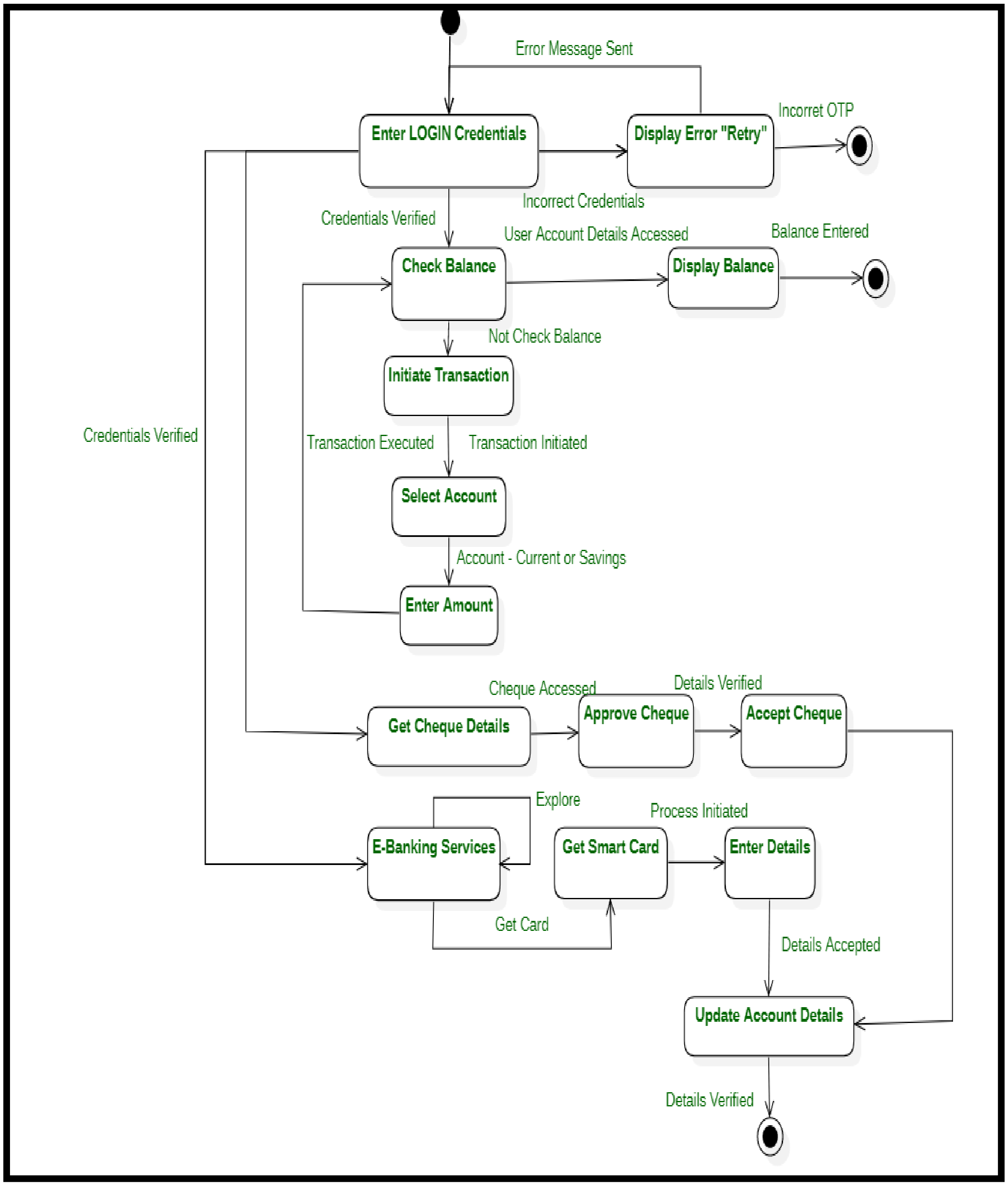


Pseudo code

1. Right click on the model

2. Select Add Diagram – State Machine diagram

State-Machine Diagram for Banking Application



COMPONENT DIAGRAM

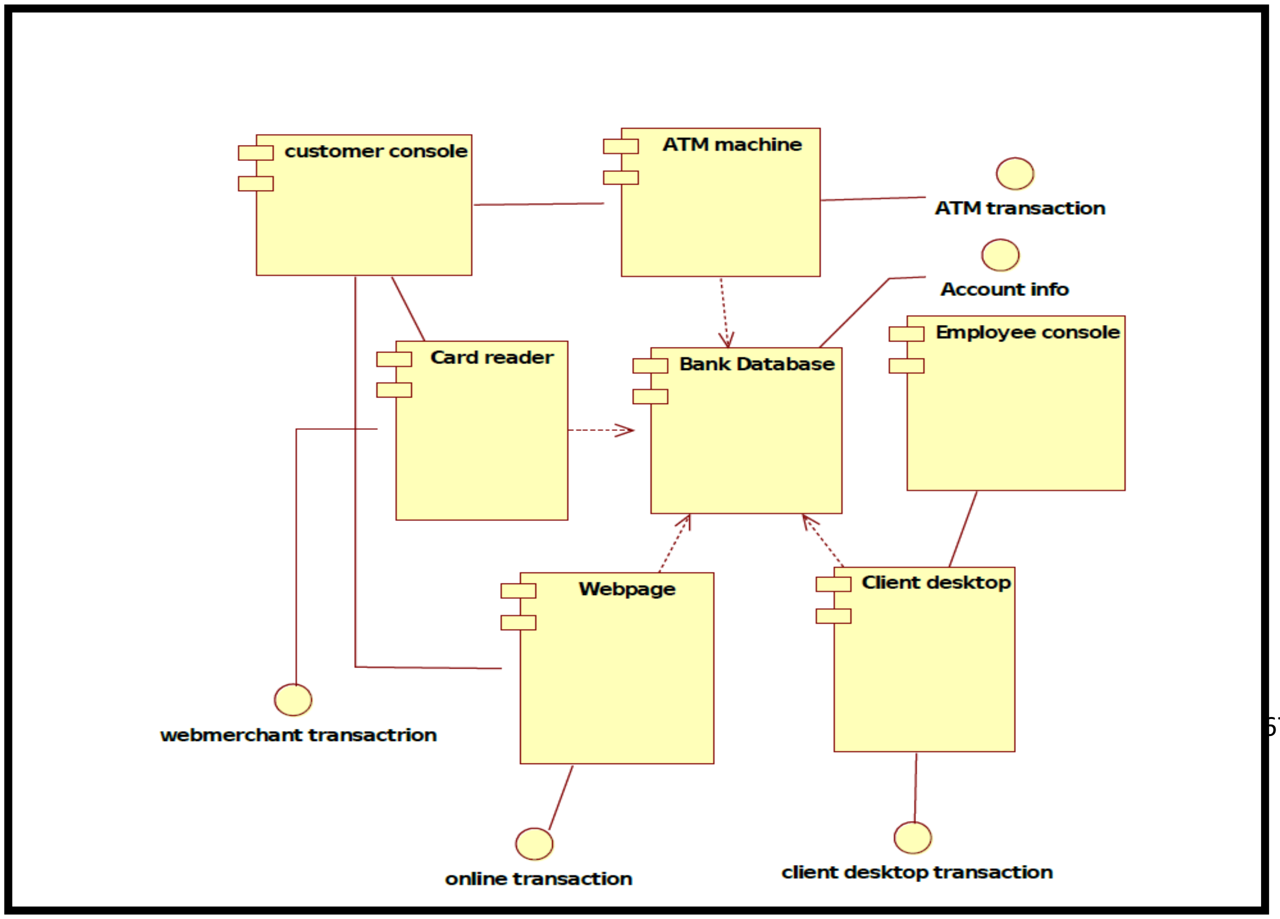
Component diagrams provide a physical view of the current model. A component diagram shows the organizations and dependencies among software components, including source code component, binary code component, and executable component. These diagrams also show the externally visible behavior of the component by displaying the interfaces of the components.
Component diagrams contain:

- Component package
- Components
- Interfaces
- Dependency relationship

Pseudo code

1. Right click on the model
2. Select Add Diagram–Component diagram

Component Diagram for Banking Application



DEPLOYMENT DIAGRAM

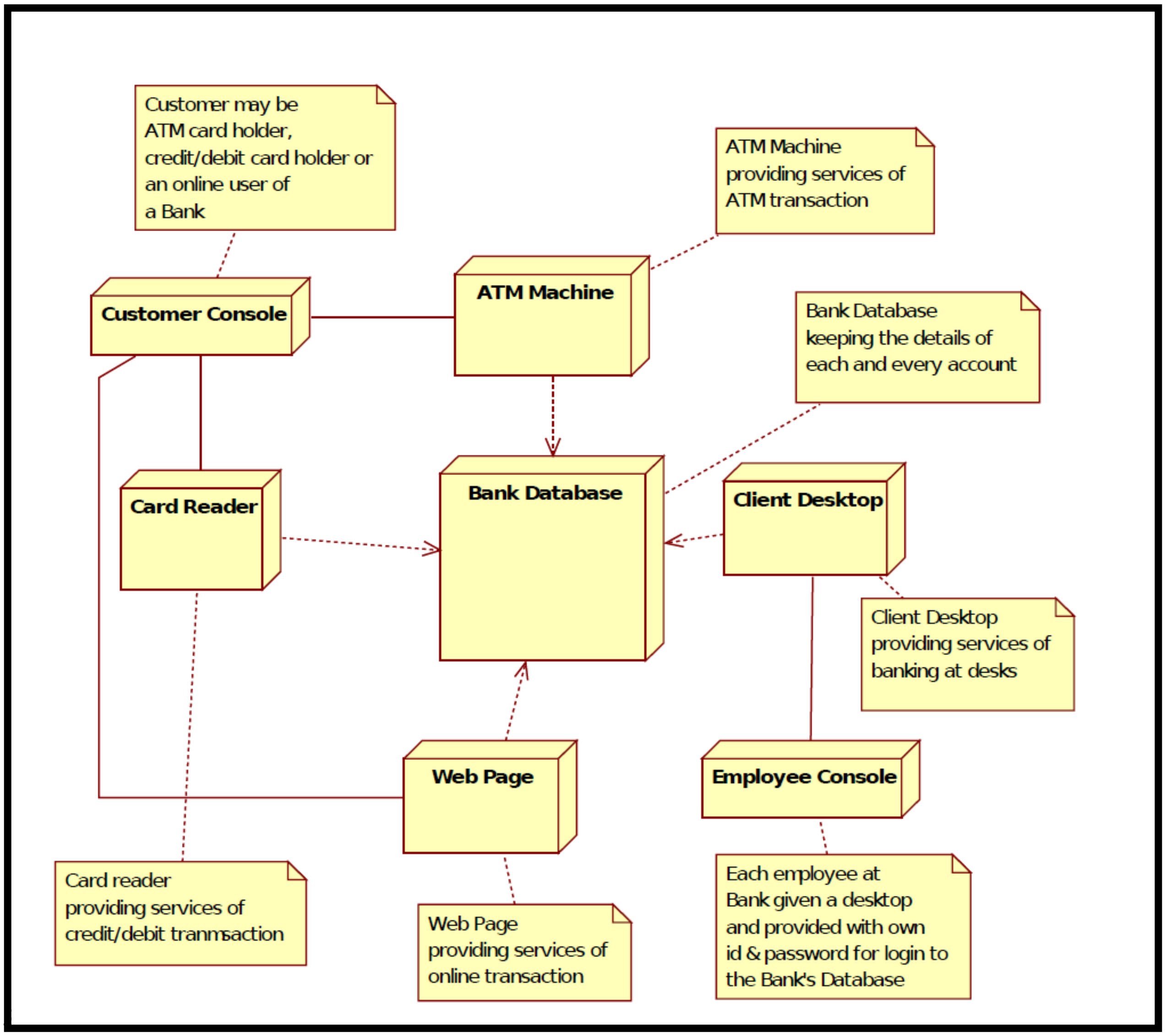
A deployment diagram shows processors, devices and connections. Each model contains a single deployment diagram which shows the connections between nodes.

Node:- A node is a hardware component capable of executing programs. Each node must have a name, there are no constraints on the node name because nodes denote hardware rather than software entities.

Pseudo code:

- Right click on the model
- Select Add Diagram– Deployment diagram

Deployment Diagram for Banking Application


Customer may be ATM card holder, credit/debit card holder or an online user of a Bank
ATM Machine providing services of ATM transaction
Customer Console
ATM Machine
Bank Database keeping the details of each and every account
Card Reader
Bank Database
Client Desktop
Client Desktop providing services of banking at desks
Web Page
Employee Console
Card reader providing services of credit/debit tranmsaction
Web Page providing services of online transaction
Each employee at Bank given a desktop and provided with own id & password for login to the Bank's Database

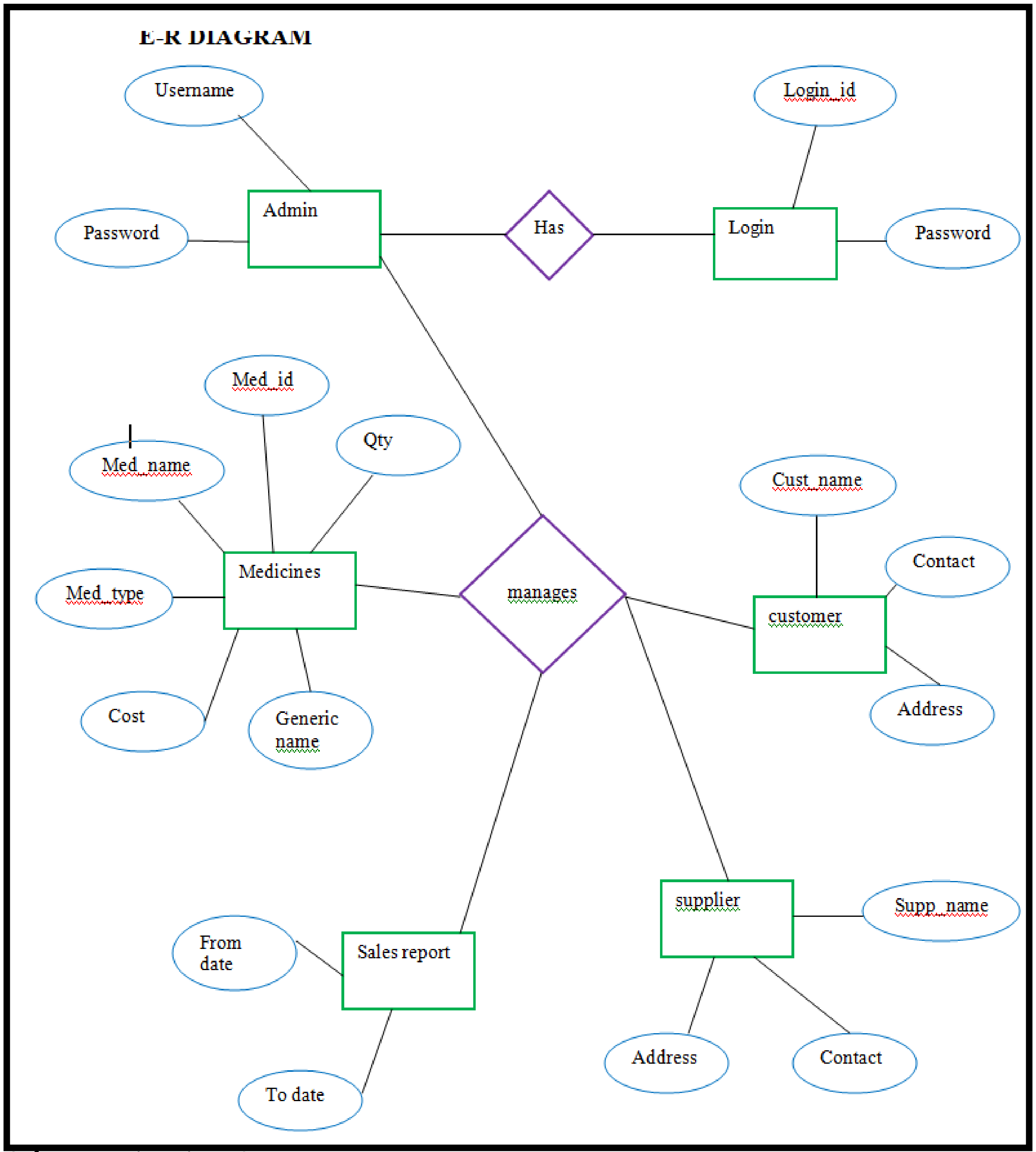
BUILDING AN E-R DIAGRAM:

•Level 1—Model All Data Objects (Entities) And Their "Connections" To One Another

•Level 2—Model All Entities And Relationships

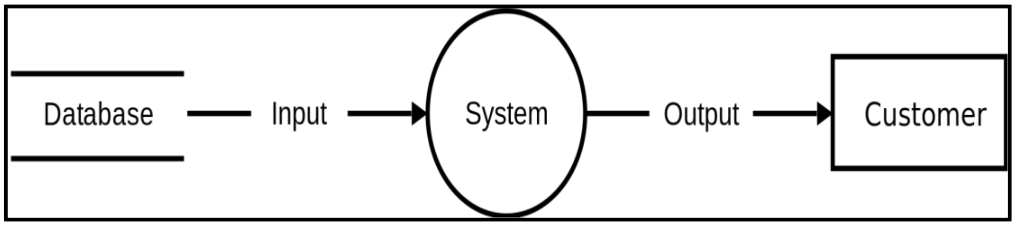
•Level 3—Model All Entities, Relationships, And The Attributes That Provide Further Depth

Follow the below given Sample as reference



DATA FLOW DIAGRAMS

A **data flow diagram** (DFD) is a way of representing a flow of a data of a processor a system (usually an information_system) The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.



For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which sub divides this process into sub-processes.

The data-flow diagram is part of the structured-analysis modeling tools. When using UML, the activity diagram typically takes over the role of the data-flow diagram. A special form of data-flow plan is a site-oriented data-flow plan.
DFD consists of processes, flows, warehouses, and terminators. There are several ways to view these DFD components.[4]

Process

The process (function, transformation) is part of a system that transforms inputs to outputs. The symbol of a process is a circle, an oval, a rectangle or a rectangle with rounded corners (according to the type of notation). The process is named in one word, a short sentence, or a phrase that is clearly to express its essence.[2]

Data Flow

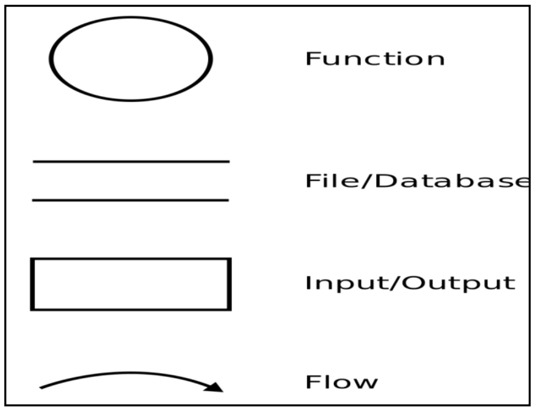
Data flow (flow, dataflow) shows the transfer of information (sometimes also material) from one part of the system to another. The symbol of the flow is the arrow. The flow should have a name that determines what information (or what material)is being moved. Exceptions are flows where it is clear what information is transferred through the entities that are linked to these flows. Material shifts are modeled in systems that are not merely informative. Flow should only transmit one type of information (material). The arrow shows the flow direction (it can also be bi-directional if the information to/from the entity is logically dependent - e.g. question and answer).Flows link processes, warehouses and terminators.[2]

Warehouse

The warehouse (data store, data store, file, database) is used to store data for later use. The symbol of the store is two horizontal lines, the other way of view is shown in the DFD Notation. The name of the warehouse is a plural noun (e.g. orders) - it derives from the input and output streams of the warehouse. The warehouse does not have to be just a data file, for example, a folder with documents, a filing cabinet, and optical discs. Therefore, viewing the ware house in DFD is independent of implementation. The flow from the warehouse usually represents the reading of the data stored in the warehouse, and the flow to the warehouse usually expresses data entry or updating (sometimes also deleting data). Warehouse is represented by two parallel lines between which the memory name is located (It can be modeled as a UML buffer node).[2]

Terminator

The Terminator is an external entity that communicates with the system and stands outside of the system. It can be, for example, various organizations (eg a bank), groups of people (e.g. customers), authorities (e.g. a tax office) or a department (e.g. a human-resources department) of the same organization, which does not belong to the model system. The terminator may be an other system with which the modeled system communicates



Sample Data Flow Diagrams for Pharmacy management system

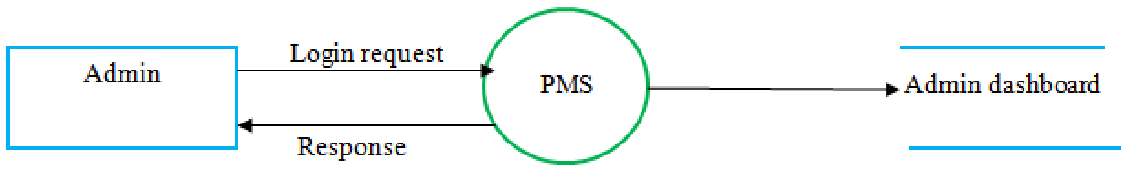


Fig: 3.6.1.2 Zero Level DFD for Admin Login

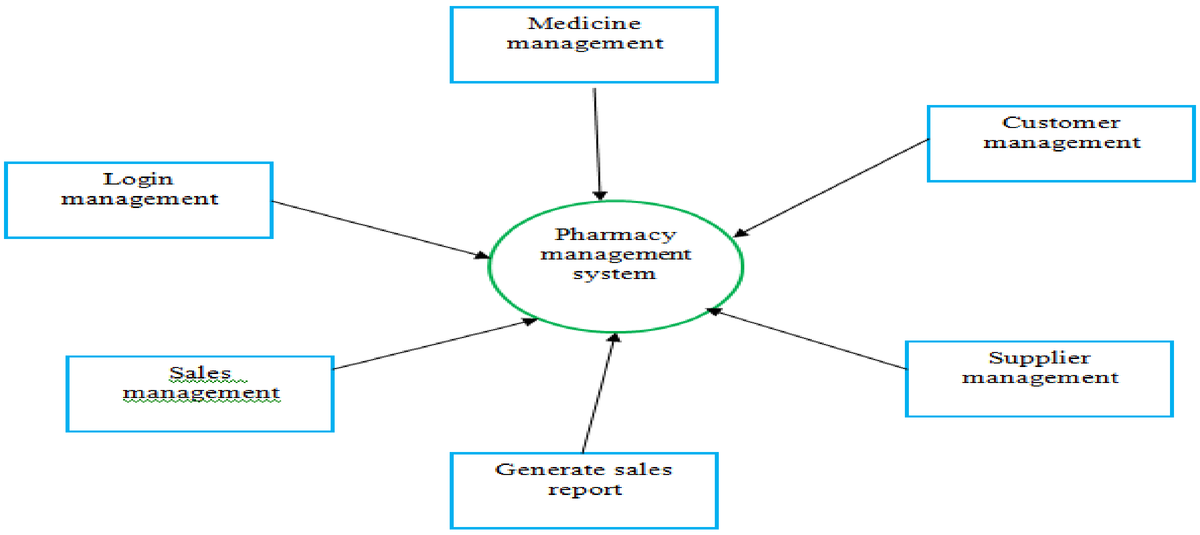


Fig: 3.6.1.1 Zero Level DFD

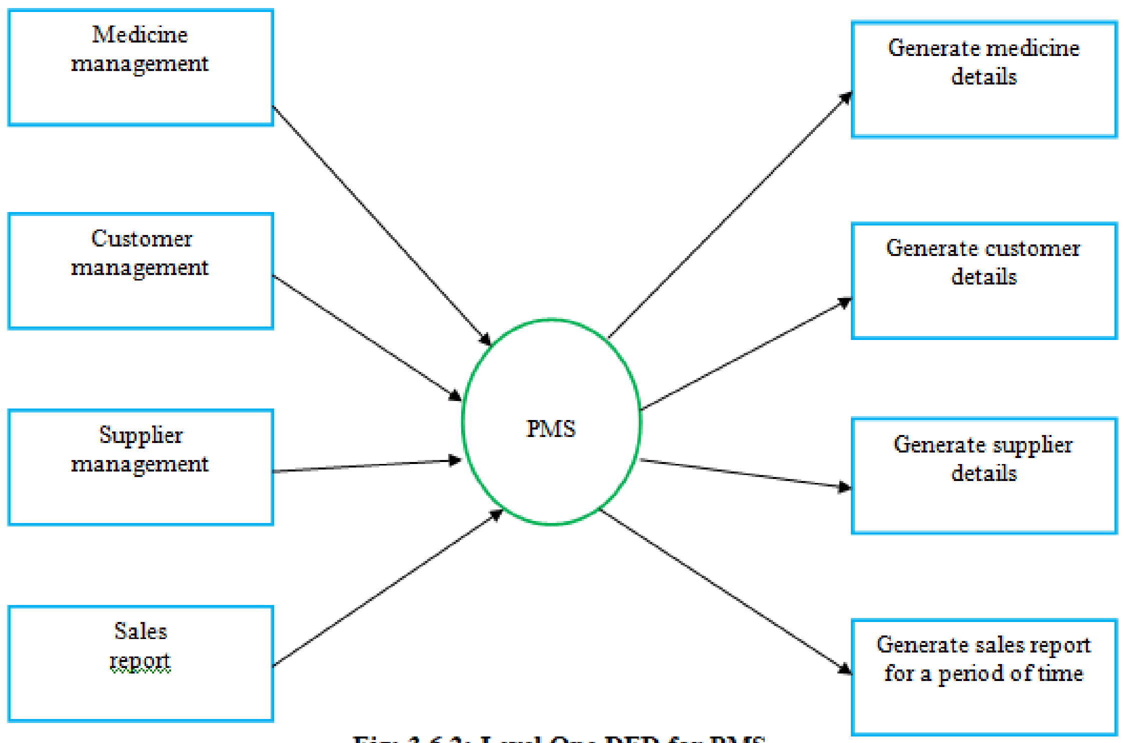

Medicine management
Customer management
Supplier management
Sales report
PMS
Generate medicine details
Generate customer details
Generate supplier details
Generate sales report for a period of time

Fig: 3.6.2: Level One DFD for PMS

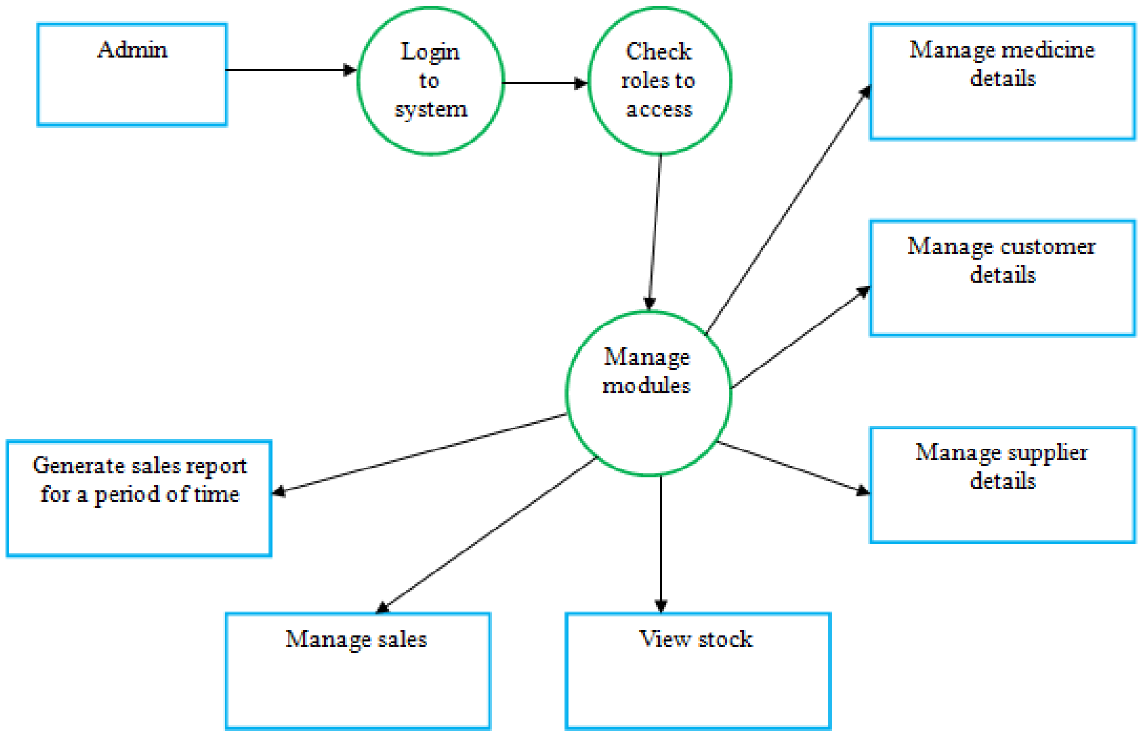


Fig: 3.6.3 level 2 DFD for admin module in PMS

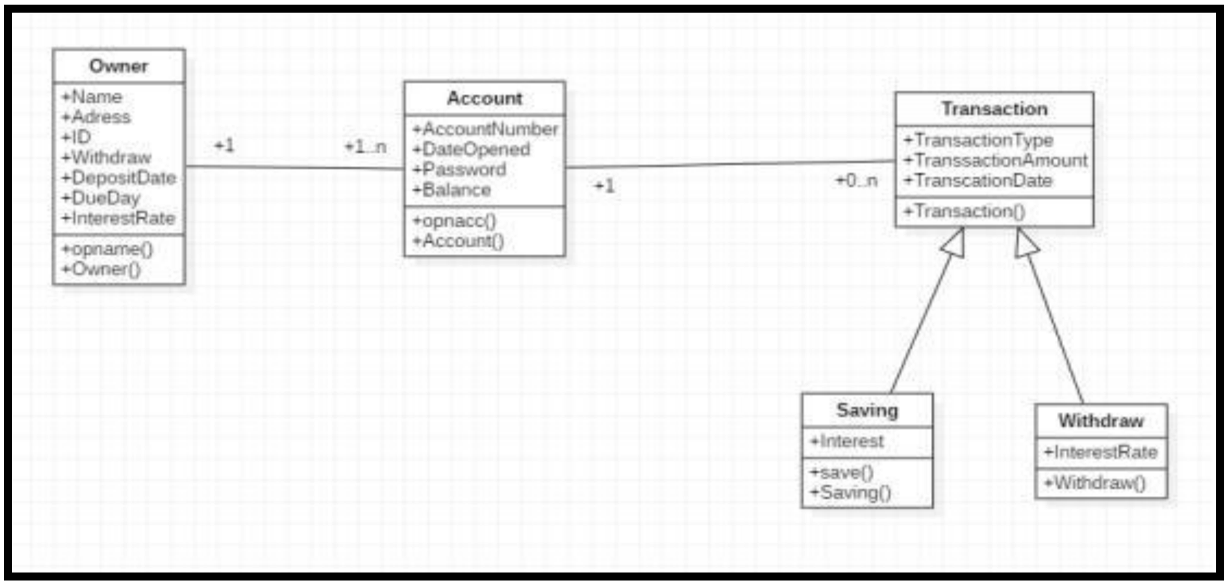
FORWARD & REVERSE ENGINEERING OF A BASIC DIAGRAM

StarUml----forward engineering operation steps

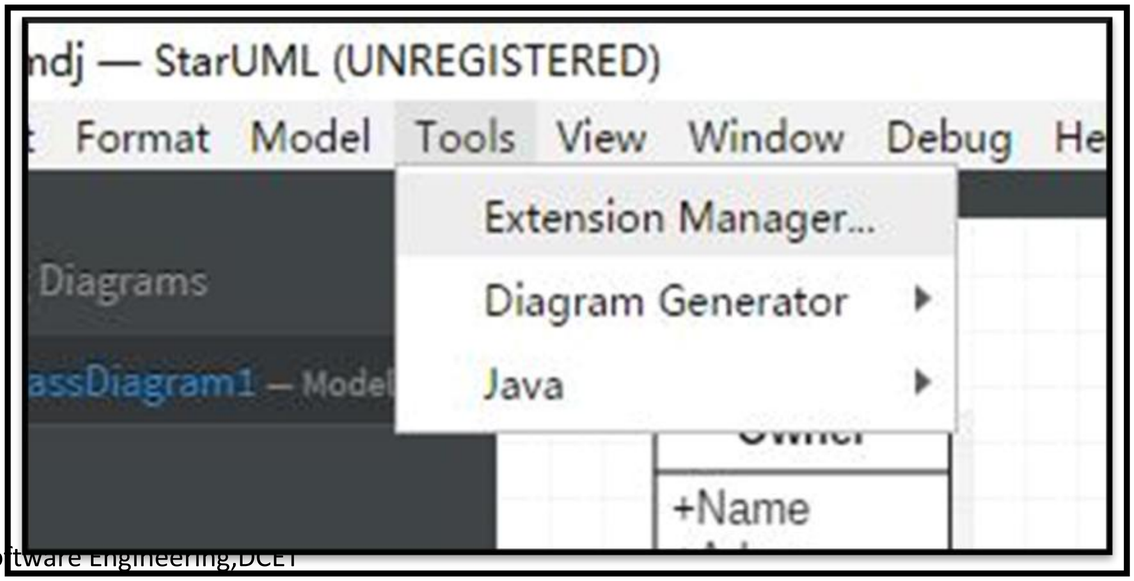
Steps to use StarUml to construct forward engineering:

Step 1: Create a new Logial View through Rose.

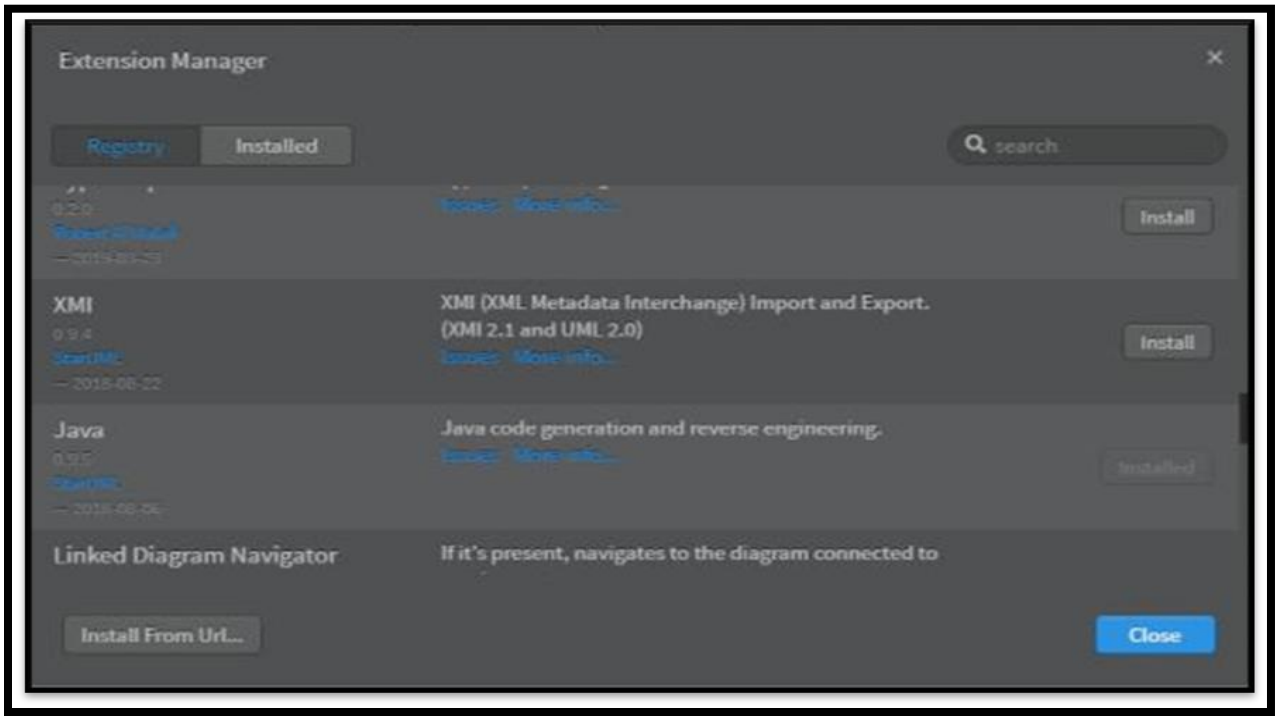
Step 2: Build a class relationship diagram, add: package, class relationship, structure, comments, etc.



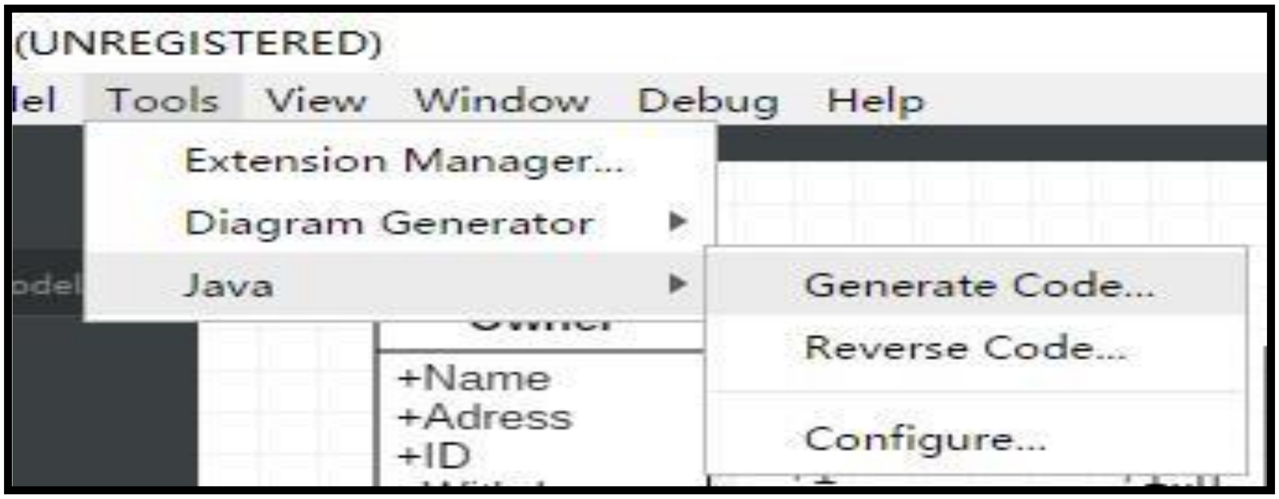
Step 3: Click the Tools drop-down menu in the upper menu bar—Extension Manager

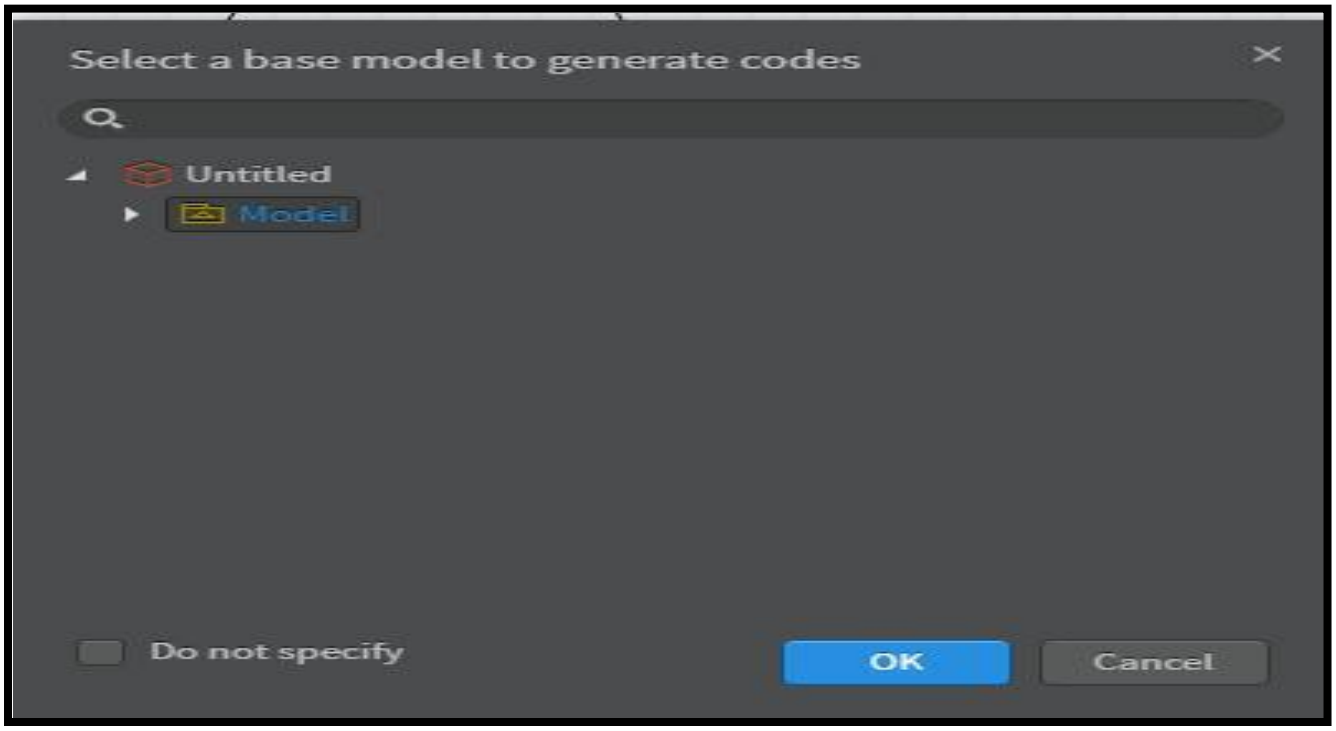


Step 4: Find Java, click install, the same for other languages


Extension Manager
Registry
Installed
search
Install
XMI
XMI (XML Metadata Interchange) Import and Export. (XMI 2.1 and UML 2.0)
Install
Java
Java code generation and reverse engineering.
Installed
Linked Diagram Navigator
If it's present, navigates to the diagram connected to
Install From Url...
Close

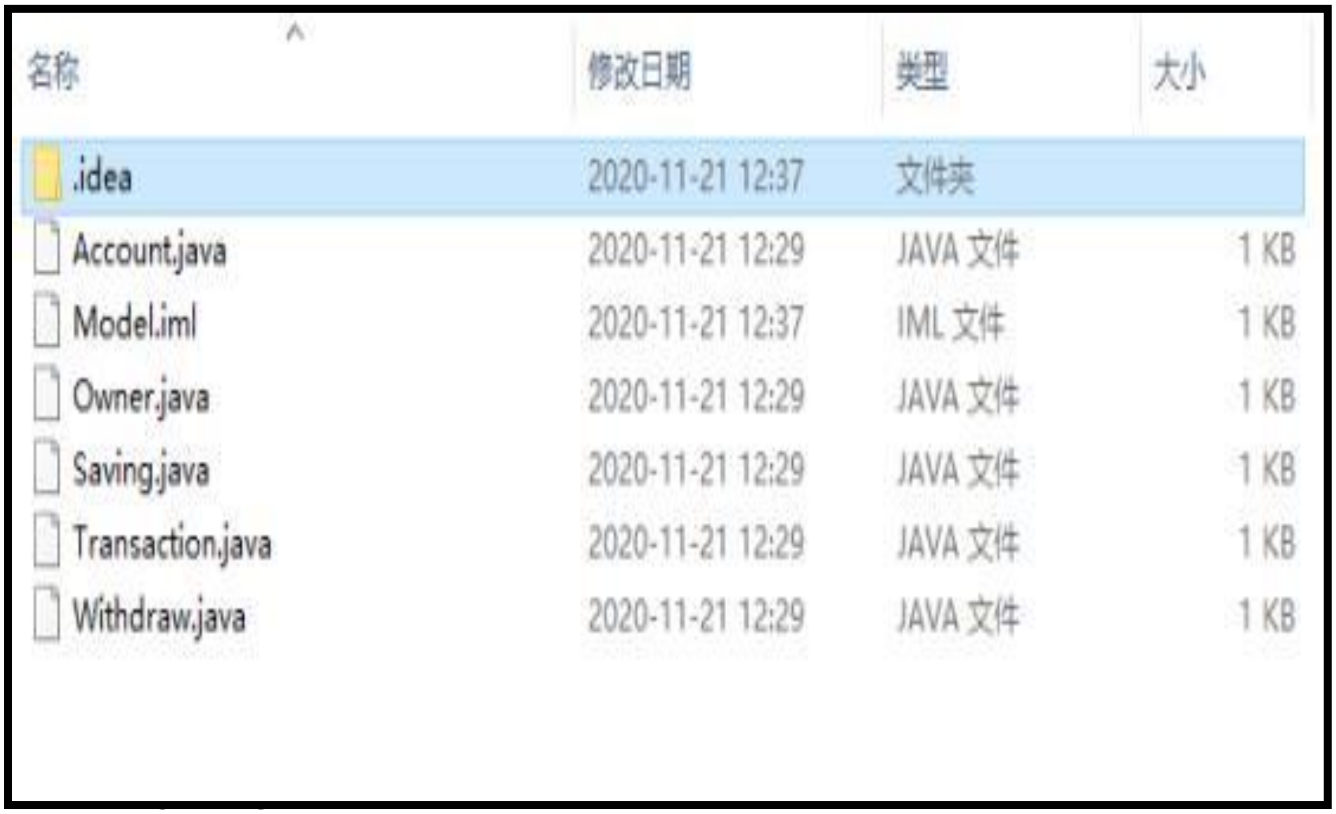
Step 5: Click Tools—Java—Generate Code





Step 6: Select the model you want to perform forward and click OK, and then the Java file is generated Click OK to the model of the direction, and then a Java file is generate

Below are the java code files generated (Forward Engineering



名称	修改日期	类型	大小
.idea	2020-11-21 12:37	文件夹	
Account.java	2020-11-21 12:29	JAVA 文件	1 KB
Model.iml	2020-11-21 12:37	IML 文件	1 KB
Owner.java	2020-11-21 12:29	JAVA 文件	1 KB
Saving.java	2020-11-21 12:29	JAVA 文件	1 KB
Transaction.java	2020-11-21 12:29	JAVA 文件	1 KB
Withdraw.java	2020-11-21 12:29	JAVA 文件	1 KB

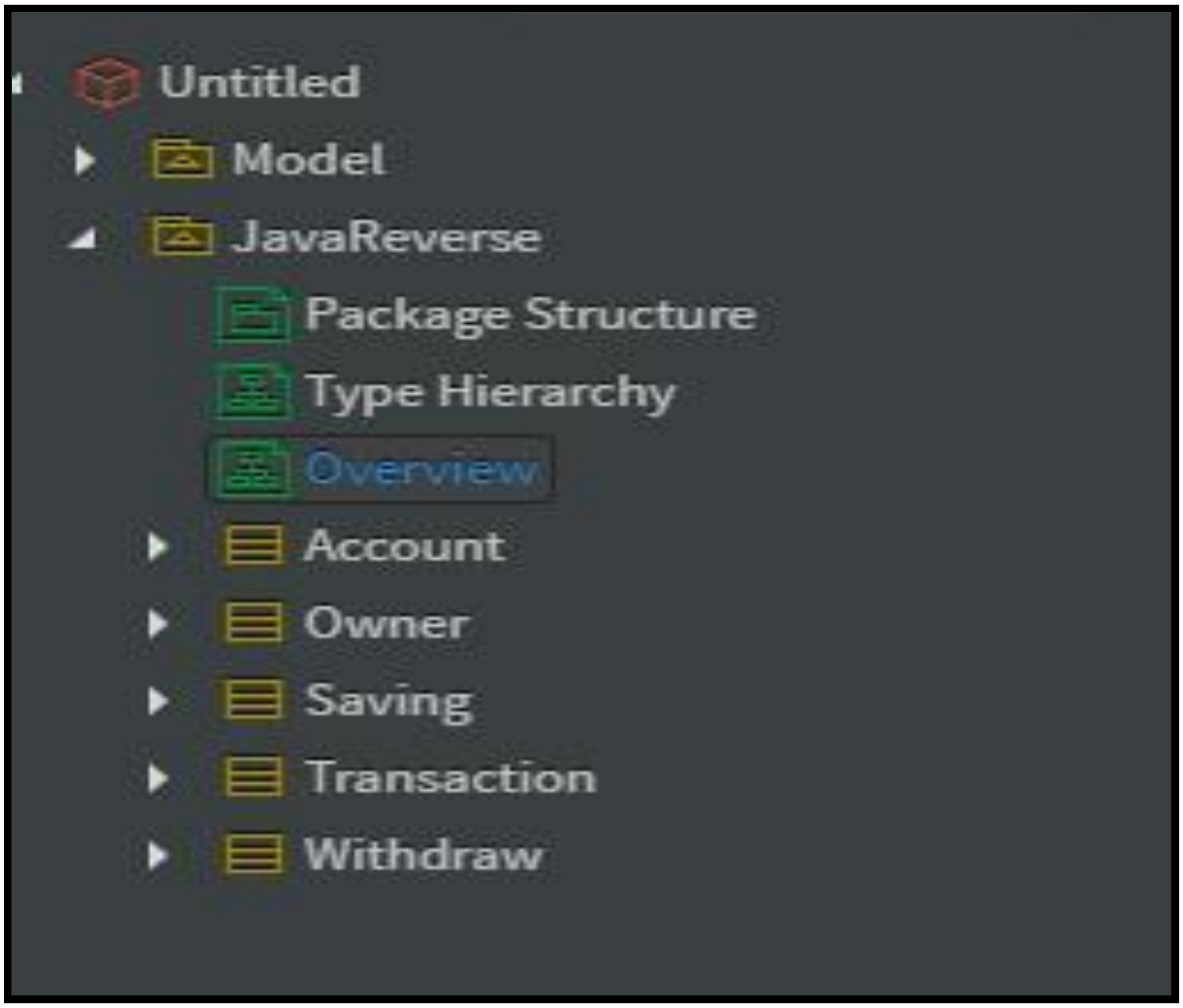
StarUml-Reverse Engineering Operation Steps

Steps to build reverse engineering using StarUml:

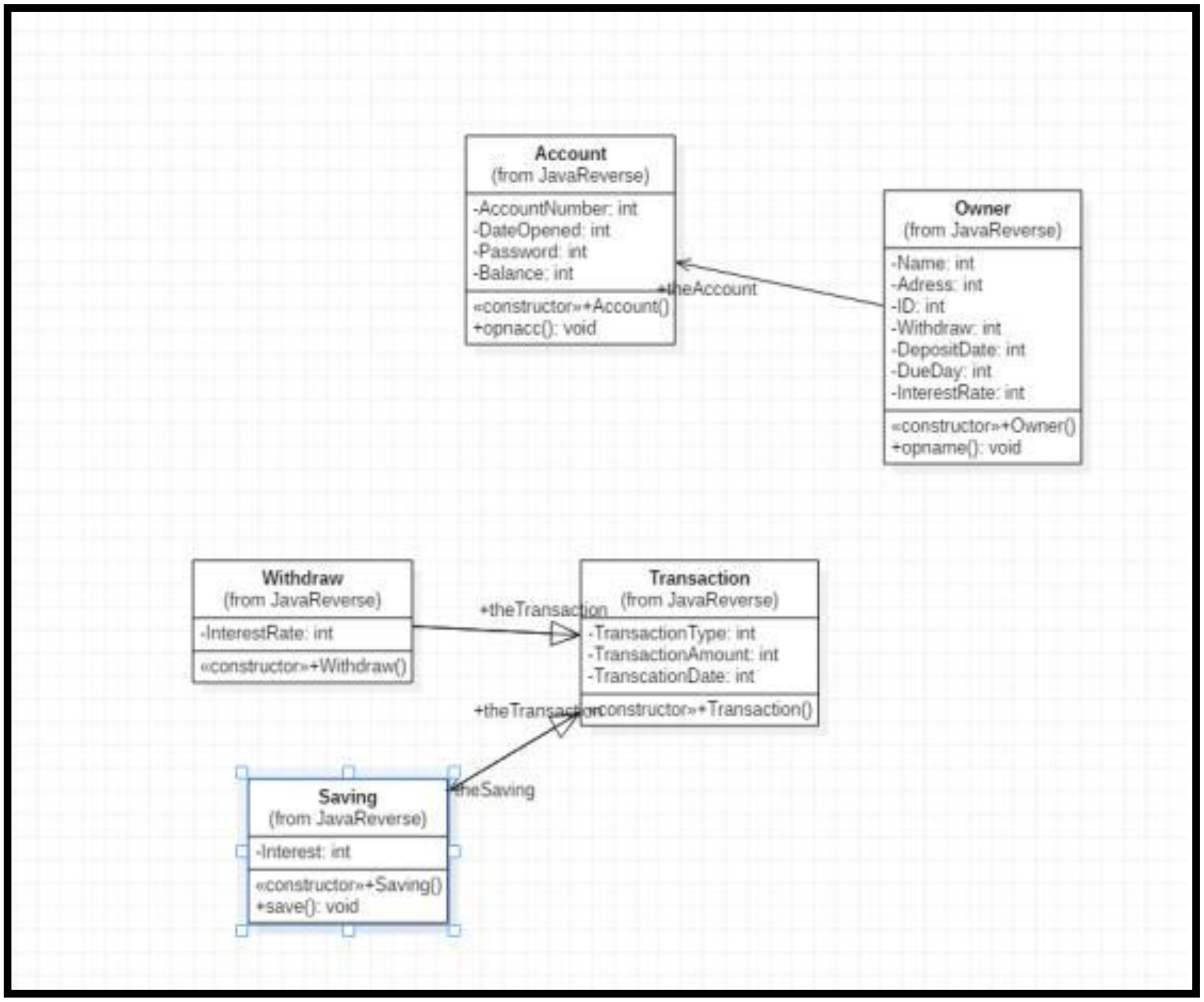
step one: Create a new Logical View through StarUml.

Step two: Click on the Tools drop-down menu Java—Reverse Code in the upper menu bar and select the folder where the code is located.

Step three: The class in the code appears in the tree structure on the left



Step Four: Drag and drop the generated UML primitives to the workspace, as shown in the figure below



Above is the diagram of the given code generated (Reverse Engineering)

Step Five: Save the mdj file

For Complete detail development of case study Code your project in any language-based IDE and Deploy it

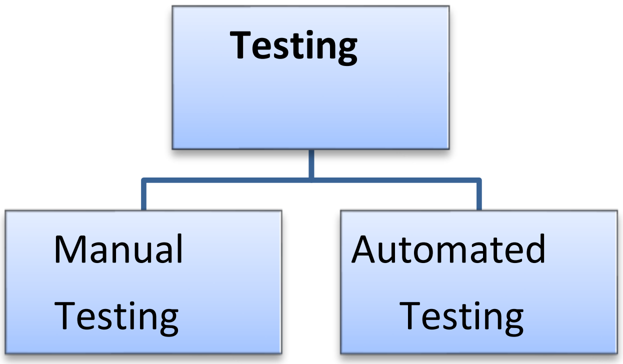
SOFTWARE TESTING

Testing Description

The purpose of testing is to assess product quality. It helps to strengthen and stabilize the architecture early in the development cycle. We can verify through testing, the various interactions, integration of components and the requirements which were implemented. It provides timely feedback to resolve the quality issues, in a timely and cost effective manner.

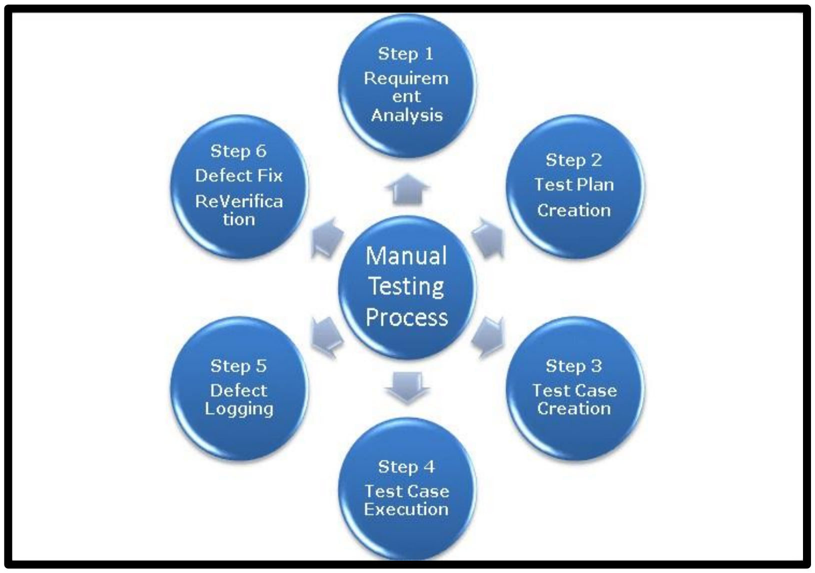
The test workflow involves the following:

- Verifying the interactions of components.
- Verifying the proper integration of components.
- Verifying that all requirements have been implemented correctly.
- Identifying and ensuring that all discovered defects are addressed before the software is deployed.



Manual Testing is a process of finding out the defects or bugs in a software program. In this method, the tester plays an important role of end-user and verifies that all the features of the application are working correctly. The tester manually executes test cases without using any automation tools. The tester prepares a test plan document which describes the detailed and systematic approach to testing of software applications. Test cases are planned to cover almost 100% of the software application. As manual testing involves complete test cases it is a time-consuming test.

The differences between actual and desired results are treated as defects. The defects are then fixed by the developer of software application. The tester retests the defects to ensure that defects are fixed. The goal of Manual testing is to ensure that application is defect & error-free and is working fine to provide good quality work to customers



Procedure of Manual Testing

1. Requirement Analysis
2. Test Plan Creation
3. Test case Creation
4. Test case Execution
5. Defect Logging
6. Defect Fix & Re-Verification

Check Login Functionality there many possible test cases are:

- Test Case 1: Check results on entering valid User Id & Password
- Test Case 2: Check results on entering Invalid User ID & Password
- Test Case 3: Check response when a User ID is Empty & Login Button is pressed, and many more

This is nothing but a Test Case.

Writing Test Cases in Manual Testing

Test Case for the scenario: Check Login Functionality

Step 1) A simple test case to explain the scenario would be

Test Case	Tester Name	Test Case Description
1	XYZ	Check response when valid username and password is entered
2	ABC	Check response redirect profile page after login successfully
3	RST	Check credentials error

Step 2) In order to execute the test case, you would need Test Data. Adding it below

Test Case	Test Case Description	Test Data
1	Check response when valid username and password is entered	Username: admin Password: admin
2	Check response redirect profile page after login successfully	Login successfully or something went wrong
3	Check credentials error	Username: admin Password: admin Invalid credentials

Identifying test data can be time-consuming and may sometimes require creating test data afresh. The reason it needs to be documented.

Step 3) In order to execute a test case, a tester needs to perform a specific set of actions on the AUT. This is documented as below:

Test Case	Test Case Description	Test Steps	Test Data

1	Check response when valid username and password is entered	Enter Username Enter Password Click Sign in	Username: admin Password: admin
2	Check response redirect profile page after login successfully	Enter Username Enter Password Successfully login and redirective to profile page Something went wrong	Login successfully or something went wrong
3	Check credentials error	Enter Username Enter Password If wrong then invalid credentials	Username: admin Password: admin Invalid credentials

Many times the Test Steps are not simple as above, hence they need documentation. Also, the author of the test case may leave the organization or go on a vacation or is sick and off duty or is very busy with other critical tasks. A recently hire may be asked to execute the test case. Documented steps will help him and also facilitate reviews by other stakeholders.

Step 4) The goal of test cases in software testing is to check behaviour of the AUT for an expected result. This needs to be documented as below

Test Case #	Test Case Description	Test Data	Expected Result
1	Check response when valid email and password is entered	Username: admin Password: admin	Login should be successful

2	Check response redirect profile page after login successfully	Login successfully or something went wrong	Login should be successful or else something error
3	Check credentials error	Username: admin Password: admin Invalid credentials	Login Unsuccessful

During test execution time, the tester will check expected results against actual results and assign a pass or fail status

Test Case #	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Check response when valid email and password is entered	Username: admin Password: admin	Login should be successful	Login was successful	Pass
3	Check credentials error	Username: admin Password: admin Invalid credentials	Login Unsuccessful	Login was unsuccessful	Fail

Step 5) That apart your test case -may have a field like, Pre - Condition which specifies things that must in place before the test can run. For our test case, a pre-condition would be to have a browser installed to have access to the site under test. A test case may also include Post - Conditions which specifies anything that applies after the test case completes. For our test case, a post condition would be time & date of login is stored in the database.

AUTOMATED TESTING USING JMETER

TESTING A SAMPLE WEB PAGE OF MAILING APPLICATION USING JMETER:

JMeter is an open-source, Java-based functional and load testing tool which was initially developed for web but has since extended to support all major protocols, like FTP, SMTP, JMS, and JDBC.

The basics of JMeter and you can configure and run load tests using JMeter.

Objective

When running a prolonged load test, you might want to receive the test reports automatically via email upon test completion, or you might be running scheduled tests and want to get test reports by email. Whatever the case, our goal is to configure JMeter to send the generated test reports automatically by email.

There are two ways you can achieve this using JMeter. The first method doesn't require any external plugins but depends on JMeter's configuration to overcome the inherent problem (described below) while sending test reports via SMTP sampler.

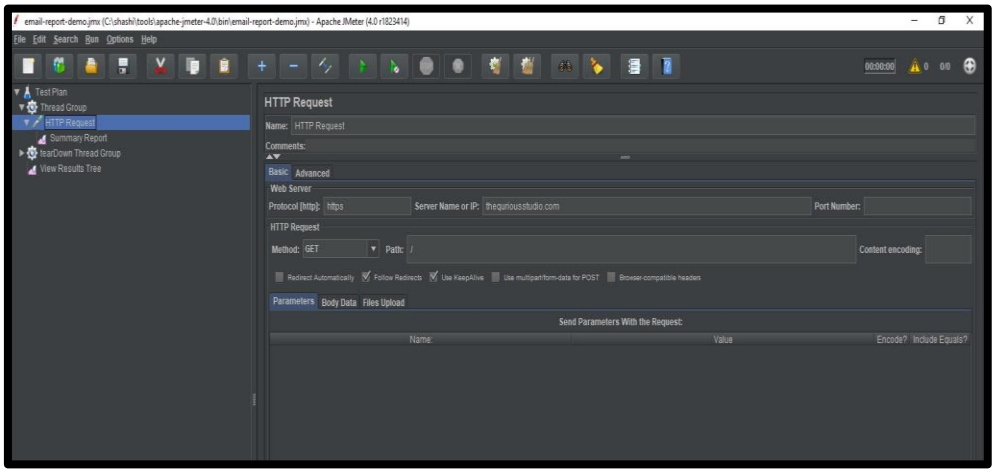
The second method requires you to install JMeter Standard Plugins (via Plugins Manager or via external jars) but provides more flexibility in the reporting section.

Prerequisites

The test has been performed with Apache JMeter v4.0, which requires JDK 8. The test script consists of a basic HTTP sampler with SMTP Sampler and "Summary Report" listener for the first method and "jp@gc Flexible File Writer" for the second method.

Using SMTP Sampler and Default "Summary Report" Listener Add a Thread Group to the Test Plan.

Add an HTTP Sampler and set up the required parameters, like Server Name, Method, Path, etc.


email-report-demo.jmx (C:\shashi\tools\apache-jmeter-4.0\bin\email-report-demo.jmx) - Apache JMeter (4.0 r1823414)

HTTP Request

Name: HTTP Request

Comments:

Web Server

Protocol [http]: https Server Name or IP: thequriousstudio.com Port Number:

HTTP Request

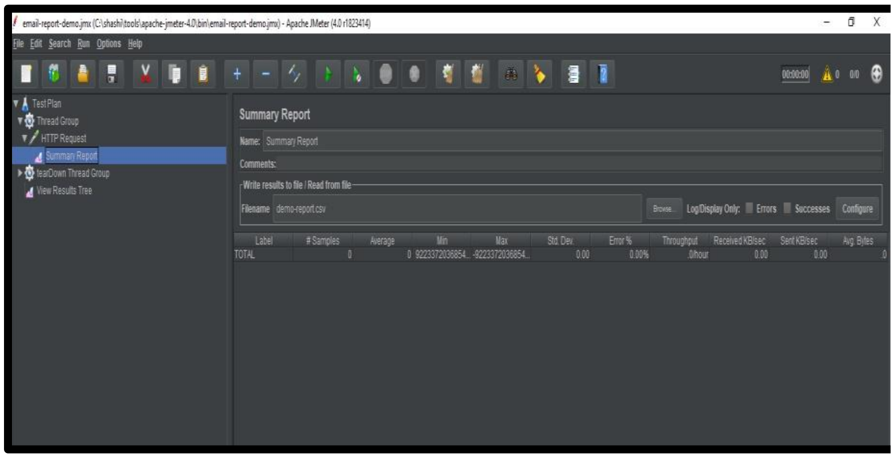
Method: GET Path: / Content encoding:

Redirect Automatically ✓ Follow Redirects ✓ Use KeepAlive Use multipart/form-data for POST Browser-compatible headers

Send Parameters With the Request:

Name:	Value	Encode?	Include Equals?

Add the "Summary Report" listener to the Thread Group. Configure the File Name parameter to define the file where reports will be saved (the default file path is JMeter's


email-report-demo.jmx (C:\shashi\tools\apache-jmeter-4.0\bin\email-report-demo.jmx) - Apache JMeter (4.0 r1823414)

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename demo-report.csv Log/Display Only: Errors Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
TOTAL	0	0	9223372036854...	-9223372036854...	0.00	0.00%	.0/hour	0.00	0.00	0

bin directory). The File

Name parameter is important as it will be referenced by the SMTP sampler.

Add a tearDown Thread group to the Test Plan. The tearDown thread group will run only after the test has been execute, so it will run after the first Thread Group is completed.

Add the SMTP Sampler to the Tear-Down Thread group and configure the mail server parameters. A demo configuration using Gmail. The SMTP server would use the following parameter values:

- **Server**- smtp.googlemail.com
- **Port**- 587

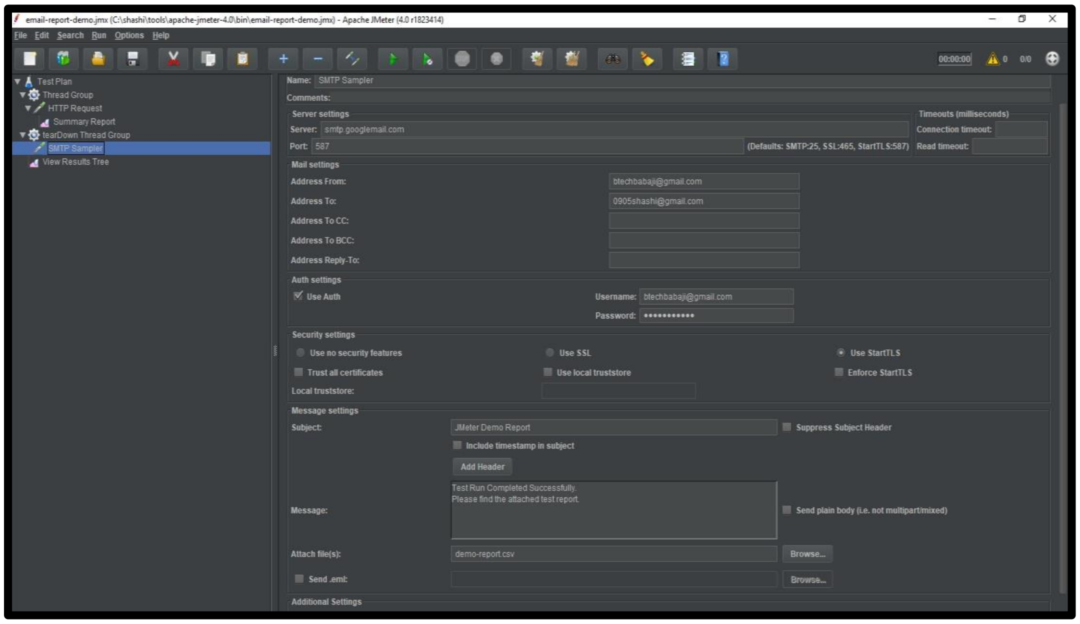
  o**Address From** - sender@gmail.com
  o**Address To** - recipient@gmal.com o**Auth Settings**

  - ✦ **username** - sender@gmail.com
  - ✦ **password** - sender's Gmail password o **Security Settings** - Use StartTLSo **Message Settings** -
  - ✦ **Subject** - Email Subject/Title to be used
  - ✦ **Message** - Email body goes here
  - ✦ **Attach File(s)** - enter the filename used in the Summary Report Listener in Step 3

When using Gmail as an SMTP server, you might need to configure Gmail's settings to allow "LessSecured" apps to sign in; otherwise, Google may block JMeter from sending any emails.

Now your test plan is almost complete, but if you run the test now, you will get an empty file in the mail without any data. This is due to the fact that the report file is written only after the test script has finished execution (which includes our tearDown Thread Group). When the SMTP Sampler is executed, it gets only an empty file without any data. The reason JMeter does this is to provide better performance — instead of executing write operations on a per-line basis, it is done all at once after test execution.

However, this property of JMeter can easily be controlled by modifying the



Name: SMTP Sampler
Comments:

Server settings
Server: smtp.googlemail.com
Port: 587 (Defaults: SMTP:25, SSL:465, StartTLS:587)

Timeouts (milliseconds)
Connection timeout:
Read timeout:

Mail settings
Address From: btechbabaji@gmail.com
Address To: 0905shashi@gmail.com
Address To CC:
Address To BCC:
Address Reply-To:

Auth settings
[✓] Use Auth
Username: btechbabaji@gmail.com
Password: ••••••••••••

Security settings
Use no security features — Use SSL — [✓ Use StartTLS]
Trust all certificates — Use local truststore — Enforce StartTLS
Local truststore:

Message settings
Subject: JMeter Demo Report
Suppress Subject Header
Include timestamp in subject
Message: Test Run Completed Successfully.
Please find the attached test report.
Send plain body (i.e. not multipart/mixed)
Attach file(s): demo-report.csv
Send .eml:

"user.properties" (or "jmeter.properties") configuration file located in JMeter's bin directory.

Find and modify (or add, if not present) the following section in the file, setting the autoflush parameter to *true*:

```
# AutoFlush on each line written in XML or CSV output

# Setting this to true will result in less test results data loss in case of Crash
# but with impact on performances, particularly for intensive tests (low or no pauses)

# Since JMeter 2.10, this is false by default jmeter.save.saveservice.autoflush=true
```

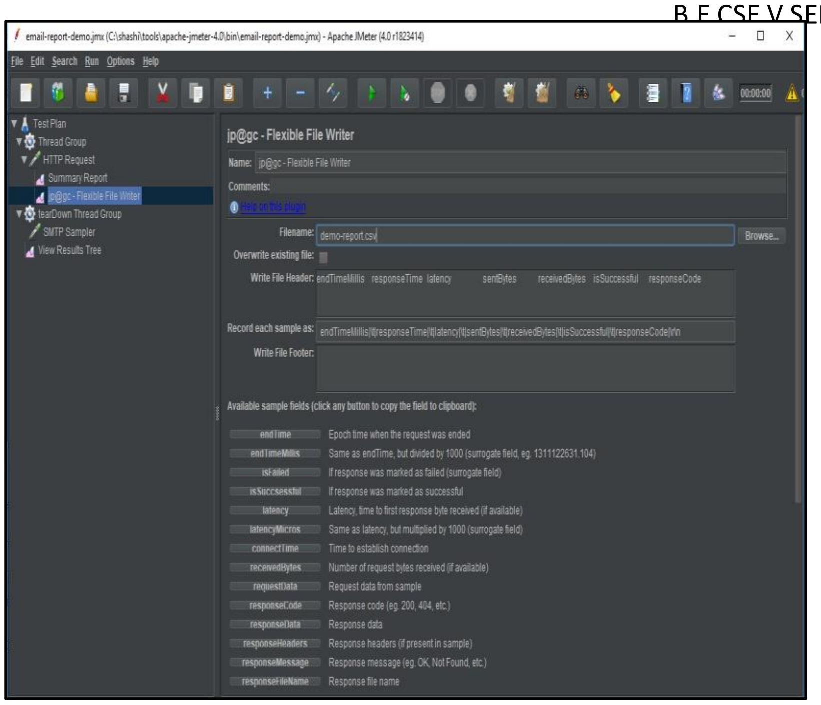
Now, restart the JMeter and run the test again. You should be able to get the test report on mail with complete data.

Using SMTP Sampler With jp@gc Flexible File Writer

The jp@gc standard plugin set for JMeter can be used to extend its capabilities, including features like writing test results in flexible formats, enabling server monitoring, and recording performance metrics.

The standard plugin set can be installed via Plugins Manager or by directly downloading the jar and putting it into the /lib/ext directory. However, Plugins Manager provides easy-to-use plugin management capabilities via UI. To install Plugins Manager,download its jar.

Once installed, you can access the Plugins Manager via File Menu in JMeter (under the "options" tab). To install the jp@gc Standard Plugin Set (which includes Flexible File Writer), go to the "Available Plugins" tab and mark the checkbox for "jp@gc Standard Set." Click the button in the bottom-right corner to "Apply Changes and Restart JMeter."


email-report-demo.jmx (C:\shashi\tools\apache-jmeter-4.0\bin\email-report-demo.jmx) - Apache JMeter (4.0 r1823414)
File Edit Search Run Options Help
00:00:00
Test Plan
Thread Group
HTTP Request
Summary Report
jp@gc - Flexible File Writer
tearDown Thread Group
SMTP Sampler
View Results Tree
Help on this plugin
Browse...

jp@gc - Flexible File Writer

Name: jp@gc - Flexible File Writer

Comments:

Filename: demo-report.csv

Overwrite existing file:

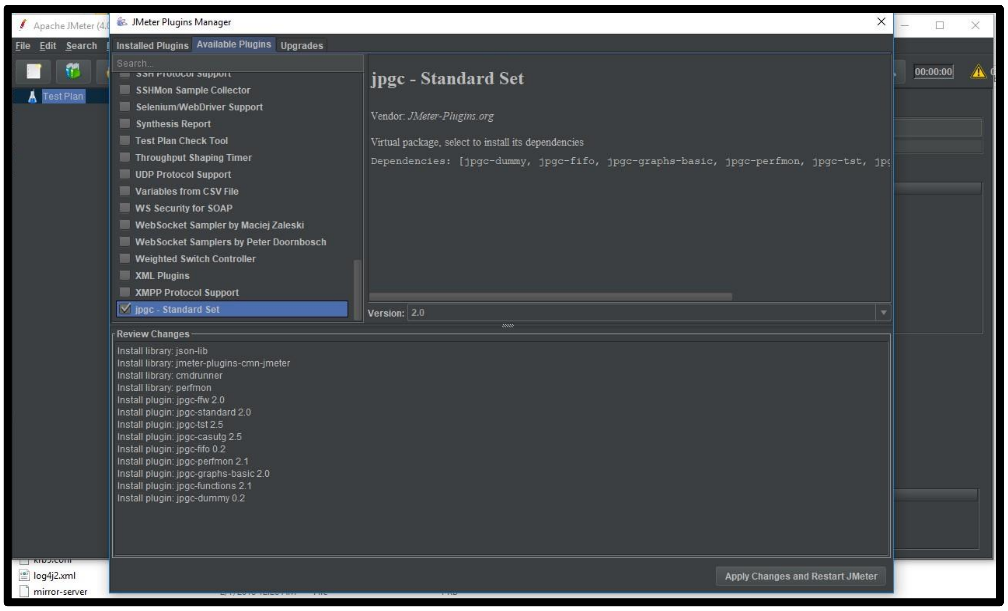
Write File Header: endTimeMillis responseTime latency sentBytes receivedBytes isSuccessful responseCode

Record each sample as: endTimeMillis|\t|responseTime|\t|latency|\t|sentBytes|\t|receivedBytes|\t|isSuccessful|\t|responseCode|\r\n

Write File Footer:

Available sample fields (click any button to copy the field to clipboard):

endTime	Epoch time when the request was ended
endTimeMillis	Same as endTime, but divided by 1000 (surrogate field, eg. 1311122631.104)
isFailed	If response was marked as failed (surrogate field)
isSuccsessful	If response was marked as successful
latency	Latency, time to first response byte received (if available)
latencyMicros	Same as latency, but multiplied by 1000 (surrogate field)
connectTime	Time to establish connection
receivedBytes	Number of request bytes received (if available)
requestData	Request data from sample
responseCode	Response code (eg. 200, 404, etc.)
responseData	Response data
responseHeaders	Response headers (if present in sample)
responseMessage	Response message (eg. OK, Not Found, etc.)
responseFileName	Response file name



SSH Protocol Support
SSHMon Sample Collector
Selenium/WebDriver Support
Synthesis Report
Test Plan Check Tool
Throughput Shaping Timer
UDP Protocol Support
Variables from CSV File
WS Security for SOAP
WebSocket Sampler by Maciej Zaleski
WebSocket Samplers by Peter Doornbosch
Weighted Switch Controller
XML Plugins
XMPP Protocol Support
[✓ jpgc - Standard Set]

jpgc - Standard Set

Vendor: *JMeter-Plugins.org*

Virtual package, select to install its dependencies

```
Dependencies: [jpgc-dummy, jpgc-fifo, jpgc-graphs-basic, jpgc-perfmon, jpgc-tst, jpg
```

Version: 2.0

Review Changes

```
Install library: json-lib
Install library: jmeter-plugins-cmn-jmeter
Install library: cmdrunner
Install library: perfmon
Install plugin: jpgc-ffw 2.0
Install plugin: jpgc-standard 2.0
Install plugin: jpgc-tst 2.5
Install plugin: jpgc-casutg 2.5
Install plugin: jpgc-fifo 0.2
Install plugin: jpgc-perfmon 2.1
Install plugin: jpgc-graphs-basic 2.0
Install plugin: jpgc-functions 2.1
Install plugin: jpgc-dummy 0.2
```

After restarting JMeter, you can find Flexible File Writer in the list of Listeners. Now, replace the

"Summary Report" listener in our test script with the "Flexible File Writer."

When you use Flexible File Writer, you don't have to modify the configuration file to enable Autoflush (done in Step 6) as it keeps writing the report file during test execution.

Notice the flexibility with the Flexible File Writer Listener. You can select the parameters to be included in the report by just selecting them from the list. You can also specify the format in which the data is written in the file.

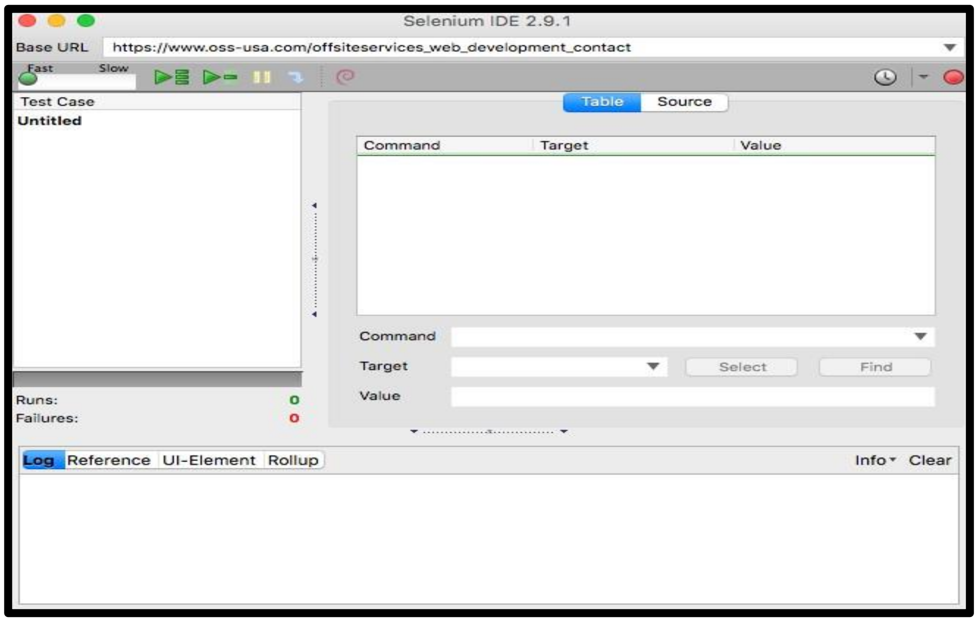
You can read more about the usage and features of Flexible File Writer.

The choice between the two methods above depends solely on your test requirements. If you only need a basic summary report, you can go with the inbuilt Summary Report Listener.

In case you need more flexibilities or control over the reports and want to explore more new features, then try out the jp@gc set of plugins.

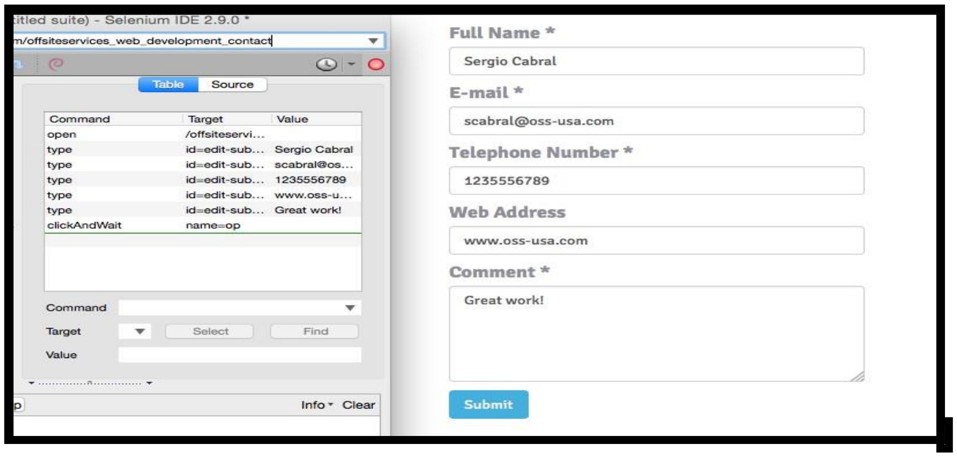
Creating A Test Case In Selenium

1. Start Selenium IDE from Firefox's Tool Bar.
2. Add "Base URL" to specify web page on which the test is going to be performed. This URL can be changed to perform the same test case on a different URL.



3. Once you've added the Base URL you can press the "Record" button (red round button top right) and start performing a manual test to the site (filling inputs, clicking links, etc.).

4. As you perform your manual test, Selenium will populate its commands table with all your interactions on the page.



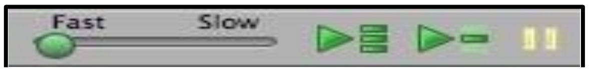
itled suite) - Selenium IDE 2.9.0 *

m/offsiteservices_web_development_contact

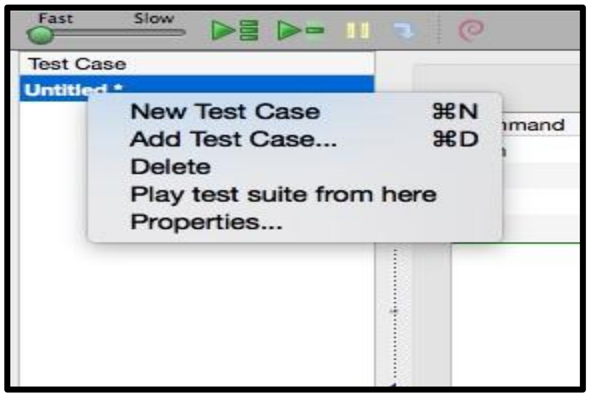
Command	Target	Value
open	/offsiteservi...	
type	id=edit-sub...	Sergio Cabral
type	id=edit-sub...	scabral@os...
type	id=edit-sub...	1235556789
type	id=edit-sub...	www.oss-u...
type	id=edit-sub...	Great work!
clickAndWait	name=op	

Full Name *	Sergio Cabral
E-mail *	scabral@oss-usa.com
Telephone Number *	1235556789
Web Address	www.oss-usa.com
Comment *	Great work!

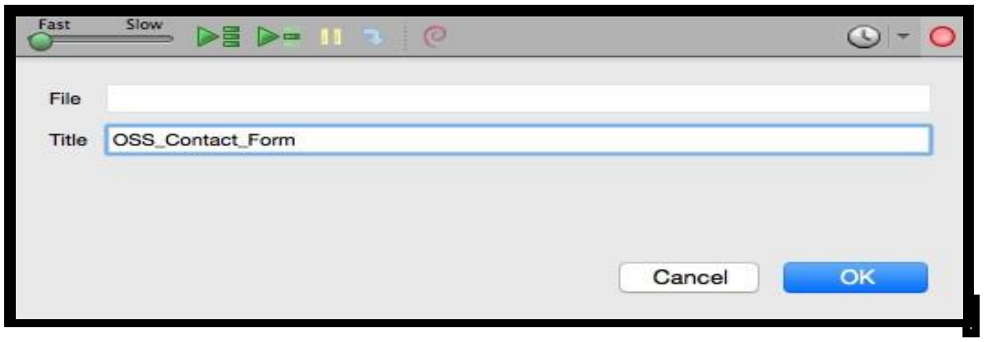
5. After you've finished recording your test, press the "Record" button again to stop the recording process.
6. Once your test is done recording, you can use the "Play" buttons to play either and entire Test Suite or current Test Case.



7. Right click on the test case's title, at the moment "Untitled *" and click on "Properties".



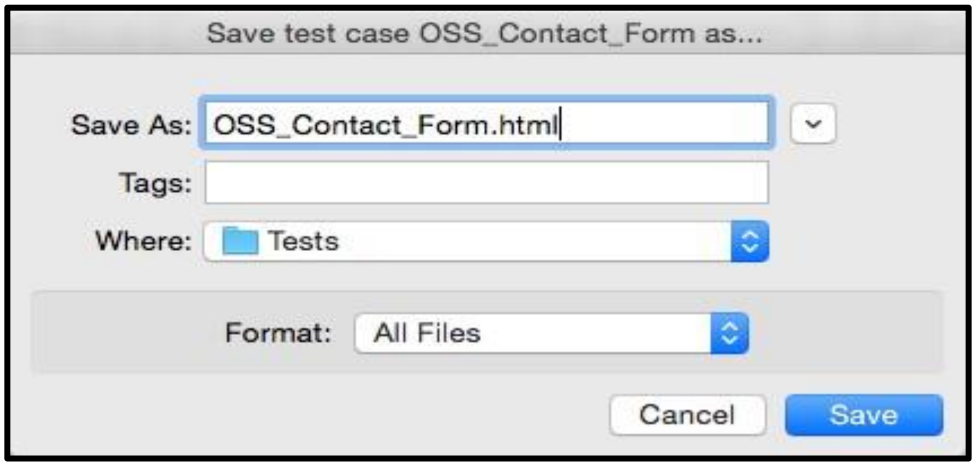
8. Here you can change the Test Case's Title to something more descriptive as you will probably have many test cases in a single Test Suite.



File:

Title: OSS_Contact_Form

9. Once your test case has a title, you can proceed to save it. Click Firefox's menu File> Save Test Case As … and add a name to the file. Test Cases could be saved as HTML files.



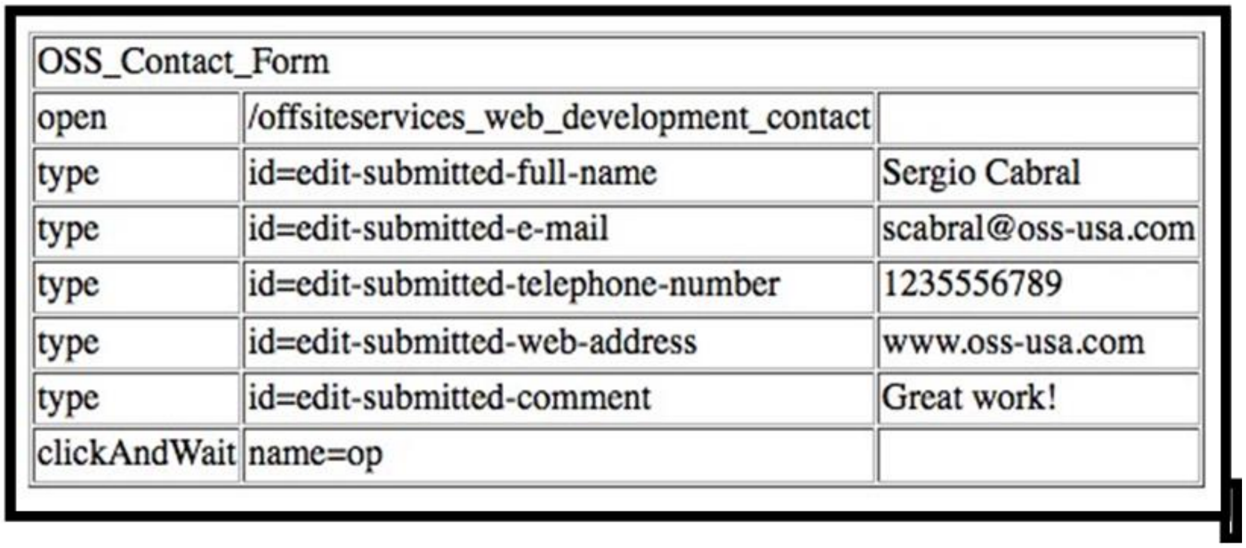
Save As: OSS_Contact_Form.html

Tags:

Where: Tests

Format: All Files

10. If you save a Test Case as an HTML file you can later double click on it and see the table on any browser.



OSS_Contact_Form		
open	/offsiteservices_web_development_contact	
type	id=edit-submitted-full-name	Sergio Cabral
type	id=edit-submitted-e-mail	scabral@oss-usa.com
type	id=edit-submitted-telephone-number	1235556789
type	id=edit-submitted-web-address	www.oss-usa.com
type	id=edit-submitted-comment	Great work!
clickAndWait	name=op	

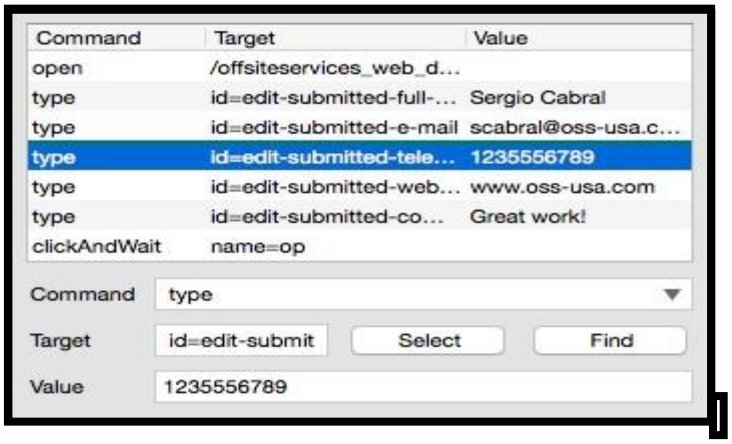
Editing an existing Test Case it's simple:

1. Start Selenium IDE from Firefox's Tool Bar.
2. Click Firefox's menu File> Open … and browse to the Test Case file that you want to work with.

Once loaded there are several things you can do to a Test Case:

- You can "Edit" a command
- You can "Add" new commands
- You can "Delete" commands **Editing a Command**

1. Click on a command from the Commands Table to select it.
2. Right below the commands table you'll see the selected command's information



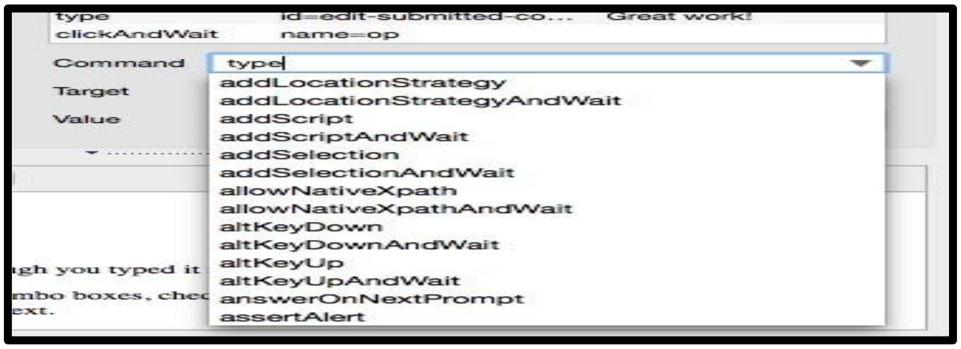
Command	Target	Value
open	/offsiteservices_web_d...	
type	id=edit-submitted-full-...	Sergio Cabral
type	id=edit-submitted-e-mail	scabral@oss-usa.c...
type	id=edit-submitted-tele...	1235556789
type	id=edit-submitted-web...	www.oss-usa.com
type	id=edit-submitted-co...	Great work!
clickAndWait	name=op	

Command: type

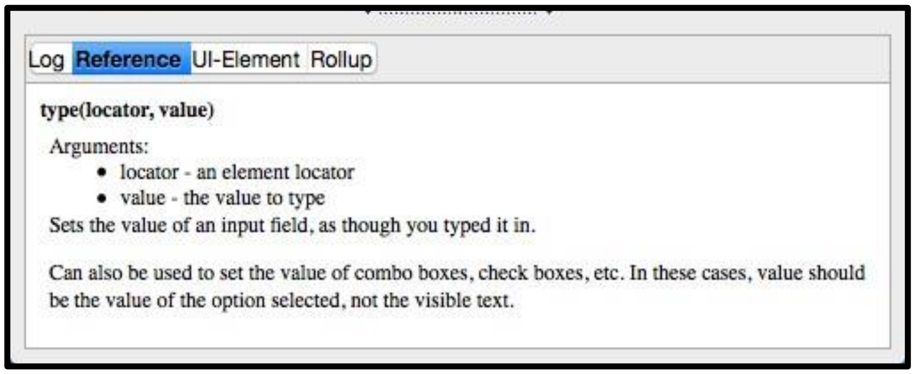
Target: id=edit-submit

Value: 1235556789

3. Here you can change the command executed selecting from the drop down menu. You CAN Find an Entire list of Selenium's commands in theSelenium Reference("broken link").



4. Once a command is selected you can see a short reference at the bottom of the app, where you'll find the command's arguments and description.



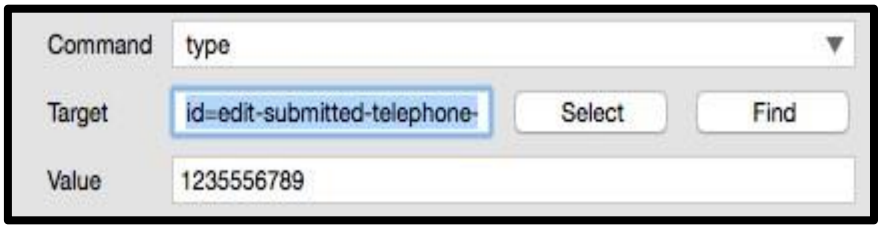
type(locator, value)

Arguments:
- locator - an element locator
- value - the value to type

Sets the value of an input field, as though you typed it in.

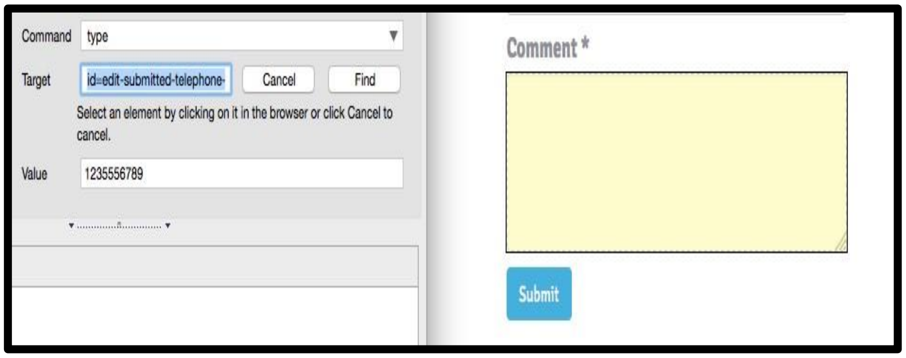
Can also be used to set the value of combo boxes, check boxes, etc. In these cases, value should be the value of the option selected, not the visible text.

5. You can also edit the target of the command by changing the target's id.



Command	type		
Target	id=edit-submitted-telephone-	Select	Find
Value	1235556789		

6. If you do not know the target's id you can click on "Select" and then click on the website's element you want to apply the command to, this will get you its id.



Command: type

Target: id=edit-submitted-telephone-

Select an element by clicking on it in the browser or click Cancel to cancel.

Value: 1235556789

7. And finally you can change the value used with the selected command by changing the "Value" field.
8. After you've finished editing commands you can proceed to save your test case as previously explained.

Adding a new command

1. Right click anywhere on the command's table and select the "Insert New Command" option.
2. An empty new command will be added to the table.
3. Edit it as previously explained.
4. After edited, drag and drop the command to its required position.
5. After you've finished adding commands you can proceed to save your test case as previously explained.

Deleting command

1. Go to your commands table and right click on the command you want to eliminate.
2. Select "Delete" from the menu.
3. After you've finished deleting commands you can proceed to save your test case as previously explained.

Conclusion of SDLC:

Software is developed following SDLC phases with Forward & Reverse Engineering, it can be implemented to fulfill all the client requirements. The system interface is very user friendly, and the overall system has been successfully tested. It has a broad future scope as new features can be incorporated in the present proposed system.

This system is designed using Unified Modeling Language concepts. The interfaces designed for the system is very user friendly and attractive. It has successfully implemented as per the client requirement.

The system has successfully passed the both testing at the development site and is under the testing phase in the presence of the client.

Viva Questions

1 What is Software Engineering?

2 What is the difference between program and software?

3 Write out the reasons for the Failure of Water Fall Model.

4 What are the characteristics of the software?

5 Define the terms :

+Agility b) Agile Team

6 What are the various categories of software?

7 What are the challenges in software?

8 Define software process

9 What are the fundamental activities of a software process?

10 What are the umbrella activities of a software process?

11 What are the merits of incremental model?

12 List the task regions in the Spiral model.

13 What are the drawbacks of spiral model?

14 What is System Engineering?

15 List the process maturity levels in SEIs CMM.

16 What is an effectors process?

17 Define the computer based system.

18 What does Verification represent?

19 What does Validation represent?

20 What is the difference between the “Known Risks” and Predictable Risks”?

21 What are the steps followed in testing?

22 Explain about the incremental model.

23 Explain in detail about the software process.

24 Explain in detail about the life cycle process.

25 Explain Spiral model and win-win spiral model in detail?

26 Name the Evolutionary process Models.

27 What are the Objectives of Requirement Analysis?

28 What is requirement engineering?

29 What are the various types of traceability in software engineering?

30 Define software prototyping.

31 What are the Requirements Engineering Process Functions?

32 What are the benefits of prototyping?

33 What are the prototyping approaches in software process?

34 What are the Difficulties in Elicitation?

35 What are the advantages of evolutionary prototyping?

36 What are the various Rapid prototyping techniques?

37 What is the use of User Interface prototyping?

38 What is System Modeling?

39 What are the characteristics of SRS?

40 What are the objectives of Analysis modeling?

41 What is data modeling?.What is a data object?

42 What is cardinality in data modeling?

43 What does modality in data modeling indicates?

44 What is ERD?

45 What is DFD?

46 What does Level0 DFD represent?

47 What is a state transition diagram?

48 Explain in detail about Functional Modeling.

49 Explain in detail about Structural Modeling.

50 Explain in detail about data modeling.

51 Explain about rapid prototyping techniques.

52 Explain the prototyping approaches in software process.

53 What are the elements of Analysis model?

54 What are the elements of design model?

55 How the Architecture Design can be represented?

56 Define design process. List the principles of a software design.

57 What is the benefit of modular design?

58 What is a cohesive module?

59 What are the different types of Cohesion?

60 What is coupling?

61 What are the various types of coupling?

62 What are the common activities in design process?

63 What are the benefits of horizontal partitioning?

64 What is vertical partitioning?

65 What are the advantages of vertical partitioning?

66 What are the various elements of data design?

67 List the guidelines for data design.

68 Name the commonly used architectural styles.

69 Explain in detail the design concepts.

70 Explain the design principles.

71 Explain the design steps of the transform mapping.

72 What are the testing principles the software engineer must apply while performing the software testing?

73 Define White Box Testing?

74 What are the two levels of testing?

75 What are the various testing activities?

76 Write short note on black box testing.

77 What is equivalence partitioning?

78 What is Regression Testing?

79 What is a boundary value analysis?

80 What are the reasons behind to perform white box testing?

81 What is cyclomatic complexity?

82 How to compute the cyclomatic complexity?

83 Distinguish between verification and validation.

84 What are the various testing strategies for conventional software?

85 Write about drivers and stubs.

86 What are the approaches of integration testing?

87 What are the advantages and disadvantages of big-bang?

88 What are the benefits of smoke testing?

89 What are the conditions exists after performing validation testing?

90 Distinguish between alpha and beta testing.

91 What are the various types of system testing?

92 Explain the types of software testing.

93 Explain in detail about Black box testing.

94 Explain about the software testing strategies.

95 What are the advantages and disadvantages of size measure?

96 Write short note on the various estimation techniques.

97 What is the Objective of Formal Technical Reviews?

98 What is COCOMO model?

99 Give the procedure of the Delphi method.

100 What is the purpose of timeline chart?

101 What is EVA?

102 What are the metrics computed during error tracking activity?

103 Why software change occurs?

104 Write about software change strategies.

105 Define CASE Tools.

106 What is software maintenance?

107 Define maintenance.

108 What are the types of software maintenance?

109 What is architectural evolution?

110 How the CASE tools are classified.

111 Explain about software cost estimation.

- Tell us the difference between activity and sequence diagrams?
- Name some of the roles that are played by the packages, modules and wrappers?
- What are the elements which are utilized in the state chart diagram?
- How does a simple use case look like?
- Can you explain 'Extend' and 'Include' in use cases?

- Can you explain class diagrams?
- How do we represent private, public and protected in class diagrams?
- what does associations in a class diagram mean?
- Can you explain aggregation and composition in class diagrams?
- What are composite structure diagram and reflexive association in class diagrams?
- Can you explain business entity and service class?
- Can you explain System entity and service class?
- Can you explain generalization and specialization?
- How do we represent an abstract class and interface UML?
- How do we achieve generalization and specialization?
- Can you explain object diagrams in UML?
- Can you explain sequence diagrams?

What is meant by Object Oriented?

Object Oriented means we organize the software as a collection of discrete objects that incorporate both data structure and behavior.

What is a class?

A class is a set of objects that share a common structure and a common behavior.

What is a formal class or abstract class?

Formal or abstract classes have no instances but define the common behaviors that can be inherited by more specific classes.

What is the need of an Object diagram?

An object diagram is used to show the existence of objects and their relationships in the logical design of a system.

What is state of an object?

The state of an object encompasses all of the properties of the object plus the current values of each of these properties.

What is unified modeling language?

Unified modeling language is a language for specifying, constructing, visualizing and documenting the software system and its components.

Write any two advantages of modeling?

The main reason for modeling is the reduction of complexity. The cost of the modeling analysis is much lower than the cost of similar experimentation conducted with real time.

Define Static model?

It can be viewed as a snapshot of a system's parameters at rest or a specific point in time. They are needed to represent the structural or static aspect of a system.

Define Dynamic model?

It can be viewed as a collection of procedures or behaviors that taken together reflect the behavior of a system over time. Dynamic modeling is the most useful during the design and implementation phases of the system development.

What is an association? Give one example.

An association is the relationship between the classes.

Ex person and company are the classes, works-for is the association name. *Works_for*

What is a qualifier? Give one example.

A qualifier is an association attribute. The qualifier rectangle is part of the association path, not part of the class.

What is a use case?

Use cases are scenarios for understanding system requirements. A use case is an interaction between users and a system.

Name the three types of relationships in a use case diagram.

Communication, Uses, extends.

Write the two types of Implementation diagram?

Component diagram, deployment diagram.

What is an activity?

An activity is a set of operations that is executing during the entire period an object is in a state.

What are the components of state chart diagrams?

There are 3 important components in the state chart diagram. They are,

Action: It triggers from one to another.

Event: Triggers the action

Guard: It is the condition that says which action to be triggered.

Explain communication diagram?

Communication diagrams and sequence diagram are similar to each other, but the difference is to show in a different way. In sequence diagram we have a time and sequence order, but in communication diagram, we pay more attention on the interaction messages between the objects.

Mention the different kinds of modeling diagram?

- Use case diagram
- Class Diagram
- Object Diagram
- Sequence Diagram
- State chart Diagram
- Communication Diagram
- Activity Diagram
- Component diagram
- Deployment Diagram
- Composite Structure Diagram
- Artifact Diagram
- Timing Diagram
- Interaction Overview Diagram
- Package Diagram

What is 'Extend' and 'Include' in the use case?

- Include relationship represents an invocation of one use case by the other. It's similar to calling another function from one function.
- Extend relationship signifies that the extending use case will work exactly like the base use case only that some new steps will inserted in the extended use case.

What is UML?

Answer: UML stands for the Unified Modeling Language.

It is a graphical language for 1) visualizing, 2) constructing, and 3) documenting *the artifacts* of a system.

It allows you to create a blue print of all the aspects of the system, before actually physically implementing the system.

What are the advantages of creating a model?

Answer: Modeling is a proven and *well-accepted engineering technique* which helps build a model. Model is a simplification of reality; it is a blueprint of the actual system that needs to be built.

Model helps to visualize the system.

Model helps to specify the structural and behavior of the system.

Model helps make templates for constructing the system.

Model helps document the system.

What are the different views that are considered when building an object-oriented software system?

Answer: Normally there are 5 views.

- Use Case view - This view exposes the requirements of a system.
- Design View - Capturing the vocabulary.
- Process View - modeling the distribution of the systems processes and threads.
- Implementation view - addressing the physical implementation of the system.
- Deployment view - focus on the modeling the components required for deploying the system.

What are the major three types of modeling used?

Answer:

The 3 Major types of modeling are

- architectural,
- behavioral, and
- structural.

How would you define Architecture?

Answer:

Architecture is not only taking care of the structural and behavioral aspect of a software system but also taking into account the software usage, functionality, performance, reuse, economic and technology constraints.

What is SDLC (Software Development Life Cycle)?

Answer: SDLC is a system including processes that are

- Use case driven,
- Architecture centric,
- Iterative, and
- Incremental.

What is the Life Cycle divided into?

Answer: This Life cycle is divided into phases.

Each Phase is a time span between two milestones.

The milestones are

- Inception,
- Elaboration,
- Construction, and
- Transition.

What are the Process Workflows that evolve through these phases?

Answer:

The Process Workflows that evolve through these phases are

- Requirement gathering,
- Analysis
- Design,
- Implementation,
- Testing,

What are Relationships?

Answer: There are different kinds of relationships:

- Dependencies,
- Generalization,
- Realization and
- Association.

Dependencies are relationships between two entities.

A change in specification of one thing may affect another thing.

Most commonly it is used to show that one class uses another class as an argument in the signature of the operation.

Generalization is relationships specified in the class subclass scenario, it is shown when one entity inherits from other.

Associations are *structural relationships* that are:

- a room has walls,
- Person works for a company.

Aggregation is a type of association where there is a has a relationship.
As in the following examples: A room has walls, or if there are two classes room and walls then the relationship is called a association and further defined as an aggregation.

How are the diagrams divided?

Answer: The diagrams are divided into static diagrams and dynamic diagrams.

List the various Static Diagrams (Also called Structural Diagram)

Answer: The following diagrams are static diagrams.

- Class diagram,
- Object diagram,
- Component Diagram,
- Deployment diagram.
- Artifact diagram
- Composite structure diagram

List the various Dynamic Diagrams (Also called Behavioral Diagrams):

Answer: The following diagrams are dynamic diagrams.

- Use Case Diagram,
- Sequence Diagram,
- Communication Diagram,
- Activity diagram,
- Statechart diagram.

What are Messages?

Answer:

A message is the specification of a communication, when a message is passed that results in action that is in turn an executable statement.

What is an Use Case?

Answer:A use case specifies the behavior of a system or a part of a system.

Use cases are used to capture the behavior that need to be developed.
It involves the interaction of actors and the system.

What are the different views that are considered when building an object-oriented software system?

Normally there are 5 views. Use Case view - This view exposes the requirements of a system. Design View - Capturing the vocabulary. Process View - modeling the distribution of the systems processes and threads. Implementation view - addressing the physical implementation of the system. Deployment view - focus on the modeling the components required for deploying the system.

Explain about composition?

Composition has a stronger variant and is represented by a filled diamond. It has a strong life cycle dependency. When a container is destroyed elements or classes present in the container are also destroyed although you will have the facility to remove the classes from the relationship.

What is UML?

UML is Unified Modeling Language. It is a graphical language for visualizing specifying constructing and documenting the artifacts of the system. It allows you to create a blue print of all the aspects of the system, before actually physically implementing the system.

What is modeling? What are the advantages of creating a model?

Modeling is a proven and well-accepted engineering technique which helps build a model. Model is a simplification of reality; it is a blueprint of the actual system that needs to be built. Model helps to visualize the system. Model helps to specify the structural and behavior of the system. Model helps make templates for constructing the system. Model helps document the system.

What are diagrams? Diagrams are graphical representation of a set of elements most often shown made of things and associations.

What are the major three types of modeling used? Major three types of modeling are structural, behavioral, and architectural.

What is Architecture?

Architecture is not only taking care of the structural and behavioral aspect of a software system but also taking into account the software usage, functionality, performance, reuse, economic and technology constraints.

What is SDLC?

SDLC is Software Development Life Cycle. SDLC of a system included processes that are Use case driven, Architecture centric and Iterative and Incremental. This Life cycle is divided into phases. Phase is a time span between two milestones. The milestones are Inception, Elaboration, Construction, and Transition. Process Workflows that evolve through these phase are Business Modeling, Requirement gathering, Analysis and Design, Implementation, Testing, Deployment. Supporting Workflows are Configuration and change management, Project management.

What are Relationships?

There are different kinds of relationships: Dependencies, Generalization, and Association. Dependencies are relations ships between two entities that that a change in specification of one thing may affect another thing. Most commonly it is used to show that one class uses another class as an argument in the signature of the operation. Generalization is relationships specified in the class subclass scenario, it is shown when one entity inherits from other. Associations are structural relationships that are: a room has walls, Person works for a company. Aggregation is a type of association where there is a has a relationship, That is a room has walls, Ãlso if there are two classes room and walls then the relationship is called a association and further defined as an aggregation.

Explain about dependency?

This form of relationship exists when a change to a certain element changes the definition and structure of the other element as well. This is indicated by a pointing arrow from the dependent side to the independent side. This form of relationship can exist between classes and inheritance.

Explain about aggregation?

Aggregation gives a much more detail than association. In aggregation you can name it and it can have same adornments. It may not be involved with more than two classes. It can have a collection of classes but its classes are not dependent on the life cycle. It`s contents are not destroyed even when its classes are destroyed.

Explain about realization and its relationships?

Realization is represented as a dashed line with empty arrow pointing towards the supplier on the diagram editor. It forms a relationship between interfaces, packages and components. It shows the relationships offered by the interface.

What are diagrams?

Diagrams are graphical representation of a set of elements most often shown made of things and associations.

What are the different views in UML?

Use Case view - Presents the requirements of a system.

Design View - Capturing the vocabulary.

Process View - Modeling the systems processes and threads.

Implementation view - Addressing the physical implementation of the system.

Deployment view - Model the components required for deploying the system.

Explain the types of diagrams in UML.

Use Case Diagram

Use Case Diagram describes “HOW” the system works. It identifies the primary elements and processes that form the system. It shows “actors” and their “roles”

Class Diagram

This diagram explores detail design of the system. The class diagram is designed using Use Case diagram. We can identify all “Nouns” in use cases as classes and “verbs” as method of the classes.

Object diagram

This diagram represents the state of classes in the system and their relationships or associations at a specific point of time.

State Diagram

This diagram represents different states that objects in the system undergo during their life cycle.

Sequence diagram

This diagram is used to explore logic of complex operations, function or procedure. In this diagram, sequence of the interactions between the objects is represented step by step.

Communication diagram

This diagram groups together the interaction between different objects.

Activity diagram

Activity diagram gives detail view of the business logic.

Deployment diagram

It shows deployment view of the system. It shows how hardware and software works together to run system.

Name the types of attributes.

Single value attribute, Multiplicity or multivalue attributes, Reference to another object or instance connection.

Write the syntax for presenting the attribute that was suggested by UML.

visibility name : type_expression = initial _value

Where visibility is one of the following

+ public visibility

# protected visibility

- private visibility

type_expression - type of an attribute

Initial_value is a language dependent expression for the initial value of a newly created object.

Write the syntax for presenting the operation that was suggested by UML

visibility name : (parameter_list): return _type_expression

Where visibility is one of the following

+ public visibility

# protected visibility

- private visibility

parameter- is a list of parameters.

Return_type_expression: is a language _dependent specification of the Implementation of the value returned by the method.

What is the need of an Interaction diagram?

An Interaction diagram is used to trace the exception of a scenario in the same context of an object diagram.

What is the need of a Class diagram?

A class diagram is used to show the existence of classes and their relationships in the logical view of a system.

What is Behavior of an object?

Behavior is how an object acts and reacts in terms of its state changes and message passing.

Define forward engineering and revere engineering.

Forward engineering means creating a relational schema from an existing object model.

Reverse engineering means creating an object model from an existing relational database layout (schema).

What are the various components in sequence diagrams?

Actor: Actor represents an external user / end user who interact with the system.

Object: Object is represented by one of components of the system.

Unit: A unit is a subsystem, or a sub component or other entity within the system.

Separator: Separator represents a boundary among sub systems, components or units.

Partition: Represents different header elements in the subsystem.

viva for oosd:

What are the building blocks?

What are different structural things, relationships?

Definitions of all diagrams.

Remember-->

Aggregation is a part of association.

Visibility has 3 things--public private and protected

There are only 2 interaction diagrams sequence and communication diagrams

What is artifact?

What is instance?

What is qualification?

Difference between action state and activity state

What are different Things in Uml?

Draw UML diagrams in their respective view as follow:

USECASE VIEW-> usecase diagram,activity diagram, swimlane diagram, state chart diagram.

LOGICAL VIEW-> class diagram, sequence diagram.

COMPONENT-> component diagram.

DEPLOYMENT-> deployment diagram.

****