

CASE TOOLS AND SOFTWARE TESTING METHODOLOGY LABORATORY MANUAL

**B.TECH
(III YEAR – II SEM)
(2018-19)**

DEPARTMENT OF INFORMATION TECHNOLOGY



**MALLA REDDY COLLEGE OF
ENGINEERING & TECHNOLOGY**

(Autonomous Institution – UGC, Govt. of India)

Recognized under 2(f) and 12 (B) of UGC ACT 1956

Affiliated to JNTUH, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2015 Certified
Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad – 500100, Telangana State, India

DEPARTMENT OF INFORMATION TECHNOLOGY

Vision

- To acknowledge quality education and instill high patterns of discipline making the students technologically superior and ethically strong which involves the improvement in the quality of life in human race.

Mission

- To achieve and impart holistic technical education using the best of infrastructure, outstanding technical and teaching expertise to establish the students into competent and confident engineers.
- Evolving the center of excellence through creative and innovative teaching learning practices for promoting academic achievement to produce internationally accepted competitive and world class professionals.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1 – ANALYTICAL SKILLS

1. To facilitate the graduates with the ability to visualize, gather information, articulate, analyze, solve complex problems, and make decisions. These are essential to address the challenges of complex and computation intensive problems increasing their productivity.

PEO2 – TECHNICAL SKILLS

2. To facilitate the graduates with the technical skills that prepare them for immediate employment and pursue certification providing a deeper understanding of the technology in advanced areas of computer science and related fields, thus encouraging to pursue higher education and research based on their interest.

PEO3 – SOFT SKILLS

3. To facilitate the graduates with the soft skills that include fulfilling the mission, setting goals, showing self-confidence by communicating effectively, having a positive attitude, get involved in team-work, being a leader, managing their career and their life.

PEO4 – PROFESSIONAL ETHICS

To facilitate the graduates with the knowledge of professional and ethical responsibilities by paying attention to grooming, being conservative with style, following dress codes, safety codes, and adapting themselves to technological advancements.

PROGRAM SPECIFIC OUTCOMES (PSOs)

After the completion of the course, B. Tech Computer Science and Engineering, the graduates will have the following Program Specific Outcomes:

1. **Fundamentals and critical knowledge of the Computer System:-** Able to Understand the working principles of the computer System and its components , Apply the knowledge to build, asses, and analyze the software and hardware aspects of it .
2. **The comprehensive and Applicative knowledge of Software Development:** Comprehensive skills of Programming Languages, Software process models, methodologies, and able to plan, develop, test, analyze, and manage the software and hardware intensive systems in heterogeneous platforms individually or working in teams.
3. **Applications of Computing Domain & Research:** Able to use the professional, managerial, interdisciplinary skill set, and domain specific tools in development processes, identify the research gaps, and provide innovative solutions to them.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
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.
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.
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.
5. **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.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
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.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
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.
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 multi disciplinary environments.
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.



MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

Maisammaguda, Dhulapally Post, Via Hakimpet, Secunderabad – 500100

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GENERAL LABORATORY INSTRUCTIONS

1. Students are advised to come to the laboratory at least 5 minutes before (to the starting time), those who come after 5 minutes will not be allowed into the lab.
2. Plan your task properly much before to the commencement, come prepared to the lab with the synopsis / program / experiment details.
3. Student should enter into the laboratory with:
 - a. Laboratory observation notes with all the details (Problem statement, Aim, Algorithm, Procedure, Program, Expected Output, etc.,) filled in for the lab session.
 - b. Laboratory Record updated up to the last session experiments and other utensils (if any) needed in the lab.
 - c. Proper Dress code and Identity card.
4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty.
5. Execute your task in the laboratory, and record the results / output in the lab observation note book, and get certified by the concerned faculty.
6. All the students should be polite and cooperative with the laboratory staff, must maintain the discipline and decency in the laboratory.
7. Computer labs are established with sophisticated and high end branded systems, which should be utilized properly.
8. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions. Misuse of the equipment, misbehaviors with the staff and systems etc., will attract severe punishment.
9. Students must take the permission of the faculty in case of any urgency to go out ; if anybody found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.
10. Students should LOG OFF/ SHUT DOWN the computer system before he/she leaves the lab after completing the task (experiment) in all aspects. He/she must ensure the system / seat is kept properly.

Head of the Department

Principal

Course Educational Objectives:

The objectives of this laboratory are:

1. Understand how UML supports the entire OOAD process.
2. Become familiar with all phases of OOAD.
3. Be able to understand the essential characteristics of tools used for Designing a model.
4. Understand different software testing tools and their features
5. Manage the project from beginning to end
6. Define, formulate and analyze a problem
7. To learn how to write software testing documents, and communicate with engineers in various forms. To gain the techniques and skills on how to use modern software testing tools to support software testing projects.

Course Outcomes:

Upon the completion of practical course Case tools & Software testing Lab, the student will be able to attain the following things:

1. Able to understand the history, cost of using and building CASE tools.
2. Ability to construct and evaluate hybrid CASE tools by integrating existing tools.
3. Understand the myths and facts of software testing.
4. Analyze and design test cases using black box testing technique which includes decision tables domain testing and transition testing.
5. Analyze and design test cases for a white box testing technique which includes path testing, data flow graphs and matrix representation for a given problem.
6. Execute how to run test script wizard and Execute how to do performance testing using testing tools including Winrunner and JMeter respectively.
7. Demonstrate the importance of testing and its role in need of software development

CASE TOOLS & STM LIST OF PROGRAMS

Sl.No	Name of the Experiment	Page No
1.	Introduction to UML	1
2.	Class Diagram for ATM	14
3.	Use Case Diagram for ATM	16
4.	Sequence Diagram for ATM	19
5.	Collaboration Diagram for ATM	21
6.	State chart Diagram for ATM	25
7.	Activity Diagram for ATM	27
8.	Component Diagram for ATM	31
9.	Deployment Diagram for ATM	32
10	Write a programs in C language in demonstration the working of the following constructs i) do..while ii) while..do iii) if...else iv)switch v) for	34
11	A program for written in C language for Matrix Multiplication fails introspect the causes for its failure and write down the possible reasons for its failure	39
12	Take ATM system and study its system specifications and report various bugs.	41
13	Write the test cases for banking application	45
14	Create test plan document for library management system	50
15	Study of testing tool (e.g. winrunner)	53
16	Study of web testing tool (e.g. selenium)	61
17	Study of bug tracking tool (e.g. bugzilla)	76
18	Study of any test management tool (e.g. test director)	92
19	a)Study of any open source testing tool (e.g. test link) b)Test Facebook Manually (beyond the syllabus)	97
20	Take a mini project and execute it during SDLC create the various UML diagrams required designing and all testing documents like test plan, TCD etc.,	118

Introduction

CASE tools known as Computer-aided software engineering tools is a kind of component- based development which allows its users to rapidly develop information systems. The main goal of case technology is the automation of the entire information systems development life cycle process using a set of integrated software tools, such as modelling, methodology and automatic code generation. Component based manufacturing has several advantages over custom development. The main advantages are the availability of high quality, defect free products at low cost and at a faster time. The prefabricated components are customized as per the requirements of the customers. The components used are pre-built, ready-tested and add value and differentiation by rapid customization to the targeted customers. However the products we get from case tools are only a skeleton of the final product required and a lot of programming must be done by hand to get a fully finished, good product. Characteristics of CASE: Some of the characteristics of case tools that make it better than customized development are;

It is a graphic oriented tool.

It supports decomposition of process. Some typical CASE tools are:
Unified Modeling Language

Data modeling tools, and

Source code generation tools

UNIFIED MODELING LANGUAGE

Introduction

The unified modeling language (UML) is a standard language for writing software blue prints of the system.

Definition:

The UML is a language for

- Visualizing
- Specifying
- Constructing
- Documenting

the artifacts of a software system.

- UML is a language that provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication.
- Vocabulary and rules of a language tell us how to create and real well formed models, but they don't tell you what model you should create and when should create them.

Building Blocks of the UML:

The vocabulary of the UML encompasses three kinds of building blocks:

1. Things
2. Relationships
3. Diagrams

Things are abstractions that are first-class citizens in a model; Relationships tie these things together;

Diagrams group interesting collections of things.

Things

Things are the most important building blocks of UML. There are four kinds of things in the UML.

1. Structural things
2. Behavioral things
3. Grouping things
4. Annotational things

1) **STRUCTURAL THINGS:**

Structural things are the nouns of the UML models.

These are static parts of the model, representing elements that are either conceptual or physical.

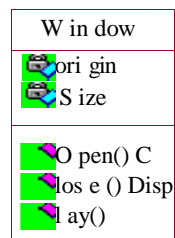
There are seven kinds of Structural things.

1. Class
2. Interface
3. Collaboration
4. Use case
5. Active class
6. Component
7. Node

Class:

A class is a description of a set of objects that shares the common attributes, operations, relationships, and semantics. A class implements one or more interfaces.

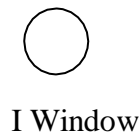
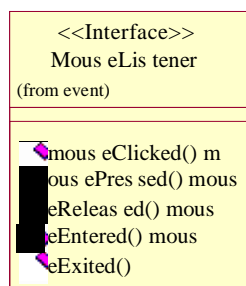
Graphically, a class is represented as a rectangle, usually including its name, attributes and operations, as shown below.



Interface:

An interface is a collection of operations that specify a service of a class or component. An interface describes the externally visible behavior of that element.

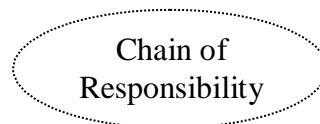
Graphically the interface is rendered as a circle together with its name.



Collaboration:

Collaboration defines an interaction and is a society of roles and other elements that work together to provide some cooperative behavior that's bigger than the sum of all the elements.

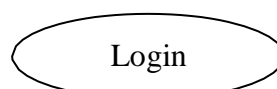
Graphically, collaboration is rendered as an ellipse with dashed lines, usually including only its name as shown below.



UseCase:

Use case is a description of a set of sequence of actions performed by a system for a specific goal for the system.

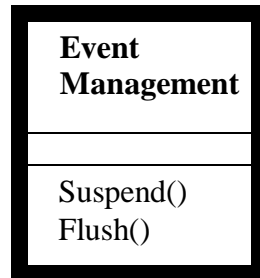
Graphically, Use Case is rendered as an ellipse with dashed lines, usually including only its name as shown below.



ActiveClass:

An active class is a class whose objects own one or more processes or threads and therefore can initiate control activity.

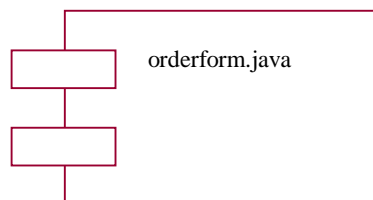
Graphically, an active class is rendered just like a class, but with heavy lines usually including its name, attributes and operations as shown below.



Component:

Component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces.

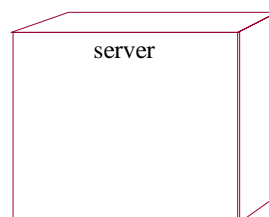
Graphically, a component is rendered as a rectangle with tabs, usually including only its name, as shown below.



Node:

A Node is a physical element that exists at run time and represents a computational resource, generally having at least some memory and often, processing capability.

Graphically, a node is rendered as a cube, usually including only its name, as shown below.



2) BEHAVIORAL THINGS:

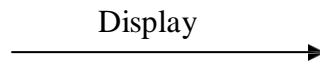
Behavioral things are the dynamic parts of UML models.

These are the verbs of a model, representing behavior over time and space.

1) *Interaction:*

An interaction is a behavior that consists of a set of messages exchanged among a set of objects(elements) within a particular context to accomplish a specific task.

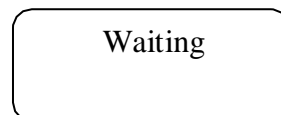
Graphically, a message is rendered as a direct line, almost always including the name of its operation, as shown below.



2) *State Machine:*

A state machine is a behavior that specifies the sequence of states of an object in its life cycle. It defines the sequence of states an object goes through in response to events.

Graphically, a state is rendered as a rounded rectangle usually including its name and its sub-states, if any, as shown below.



3) **GROUPING THINGS:**

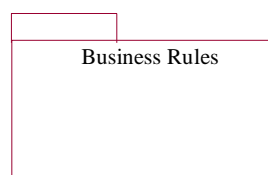
Grouping things are the organizational parts of the UML models. These are the boxes into which a model can be decomposed.

There is one primary kind of grouping thing with “package”.

Package:

A package is a general-purpose mechanism for organizing elements into groups.

Package is the only one grouping thing available for gathering structural and behavioral things.



Package

4) **ANNOTATIONAL THINGS:**

Annotational things are the explanatory parts of the UML models.

Annotational things can be defined as a mechanism to capture remarks, descriptions, and comments of UML model elements.

Note:

A note is simply a symbol for rendering constraints and comments attached to an element or a collection of elements.

Graphically a note is represented as a rectangle with dog-eared corner together, with a textual or graphical comment, as shown below.



Note

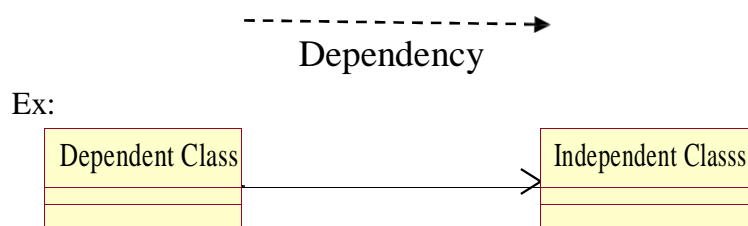
RELATIONSHIPS IN THE UML:

Relationship is another most important building block of UML. It shows how elements are associated with each other and this association describes the functionality of an application. There are four kinds of relationships in the UML:

1. Dependency
2. Association
3. Generalization
4. Realization

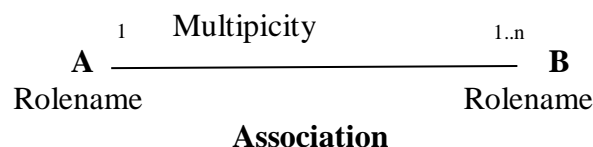
Dependency

Dependency is a relationship between two things in which change in one element also affects the other one.



Association:

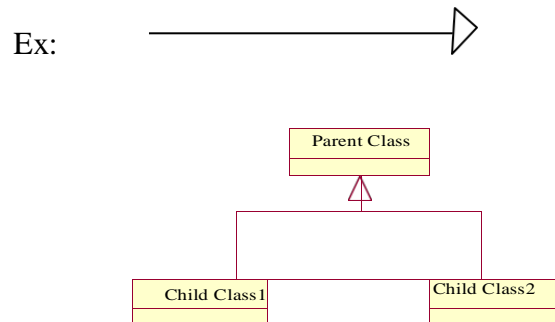
Association is basically a set of links that connects elements of an UML model. It also describes how many objects are taking part in that relationship.



Ex:

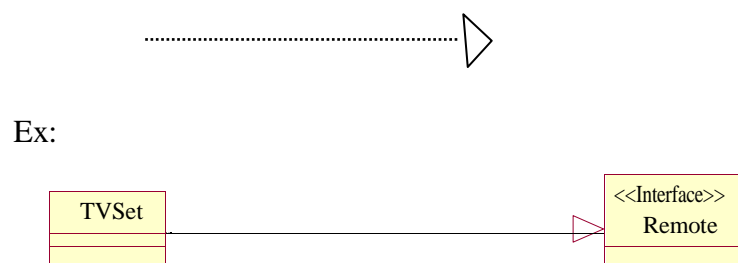
Generalization:

Generalization can be defined as a relationship which connects a specialized element with a generalized element. It basically describes inheritance relationship in the world of objects.



Realization:

Realization 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 implements them. This relationship exists in case of interfaces.



DIAGRAMS IN UML:

All the elements, relationships are used to make a complete UML diagram and the diagram represents a system.

The visual effect of the UML diagram is the most important part of the entire process.

Each UML diagram is designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction.

UML diagrams are the ultimate output of the entire system.

A diagram is the graphical presentation of a set of elements ,most often rendered as a connected graph of vertices(things) arcs (relationships).

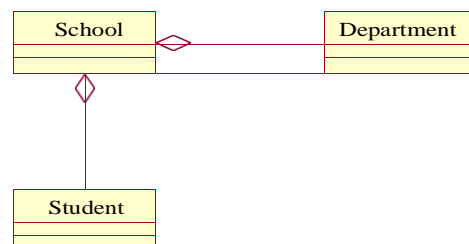
UML includes the following nine diagrams:

- 1) Class diagram
- 2) Object diagram
- 3) Use case diagram
- 4) Sequence diagram
- 5) Collaboration diagram
- 6) Activity diagram
- 7) State chart diagram
- 8) Deployment diagram
- 9) Component diagram

1. ClassDiagram

Class diagram is a diagram that shows a set of classes, interfaces, and collaborations and their relationships. Class diagrams address the static design view or the static process view of the system.

Graphically it is represented as follows:-



2. ObjectDiagram

Object diagram shows a set of objects and their relationships. These diagram the static design view or static process view of a system.

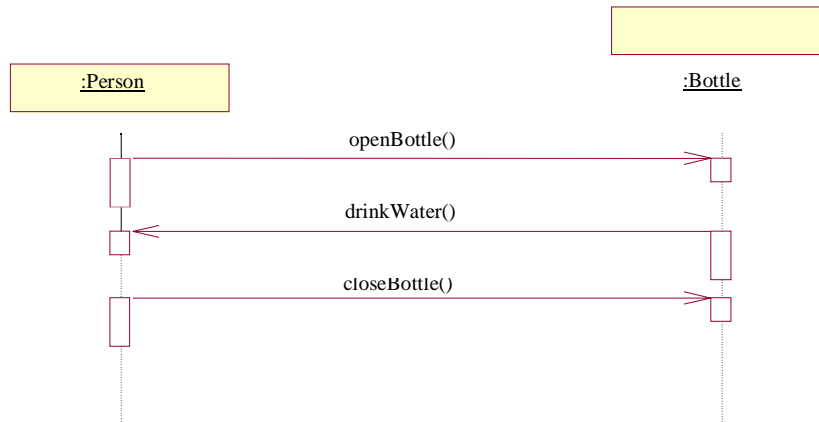
3. UsecaseDiagram

Use Case diagram shows a set of use cases and actors (a special kind of class) and their relationships. These diagrams address the static use case view of a system. Graphically it is represented as follows:-



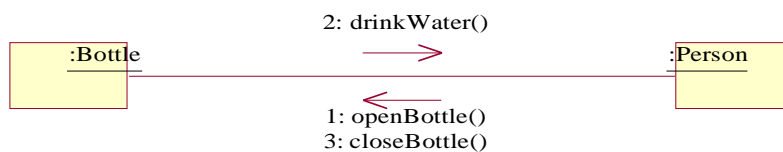
4. SequenceDiagram

Sequence diagram are interaction diagrams. This diagram emphasizes the time-ordering of messages. These diagrams address the dynamic view of a system. Sequence Diagram displays the time sequence of the objects participating in the interaction. This consists of the vertical dimension (time) and horizontal dimension (different objects). Graphically it is represented as follows:-



5 Collaboration Diagram

Collaboration diagram are also interaction diagrams. These diagrams emphasizes the structural organization of the objects that send and receive messages. These diagrams address the dynamic view of a system. Collaboration Diagram displays an interaction organized around the objects and their links to one another. Numbers are used to show the sequence of messages. Graphically it is represented as follows:-



6. Statechart Diagram

State chart diagram shows a state machine, consisting of states, transitions, events and activities. These diagrams address the dynamic view of the system. State Chart diagram displays the sequences of states that an object of an interaction goes through during its life in response to received stimuli, together with its responses and actions.

7 .Activity Diagram

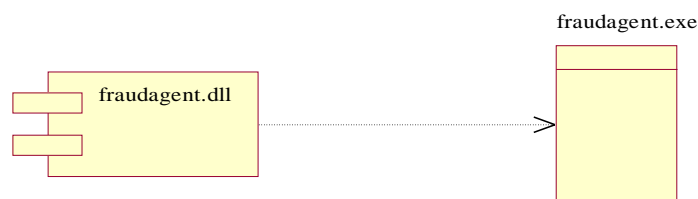
Activity diagram is a special kind of a state chart diagram that shows the flow from activity to activity within a system. These diagrams address dynamic view of a system. Activity Diagram displays a special

state diagram where most of the states are action states and most of the transitions are triggered by completion of the actions in the source states. Graphically it is represented as follows:-

8. ComponentDiagram

Component diagram shows the organizations and dependencies among a set of components. These diagrams address the static implementation of view of a system. Component Diagram displays the high level packaged structure of the code itself. Dependencies among components are shown, including source code components, binary code components, and executable components. Some components exist at compile time, at link time, at run times well as at more

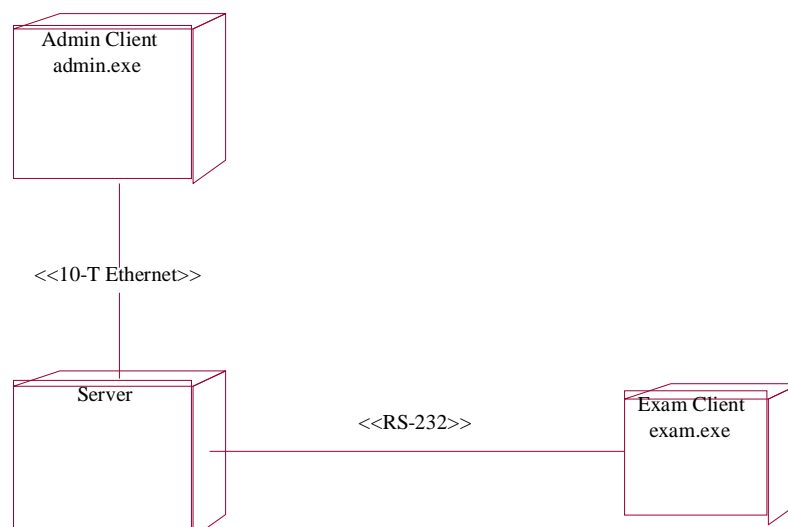
than one time. Graphically it is represented as follows:-



9. Deployment Diagram

Deployment diagram shows the configuration of run-time processing nodes and the components that live on them. These diagrams address the static deployment view of architecture. Deployment Diagram displays the configuration of run-time processing elements and the software components, processes, and objects that live on them. Software component instances represent run-time manifestations of code.

Graphically it is represented as follows:-



Experiment 1

ATM System

Automatic Teller Machine (ATM)

Description of ATM System

The software to be designed will control a simulated automated teller machine (ATM) having a magnetic stripe reader for reading an ATM card, a customer console (keyboard and display) for interaction with the customer, a slot for depositing envelopes, a dispenser for cash, a printer for printing customer receipts, and a key-operated switch to allow an operator to start or stop the machine. The ATM will communicate with the bank's computer over an appropriate communication link. (The software on the latter is not part of the requirements for this problem.)

The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) – both of which will be sent to the bank for validation as part of each transaction. The customer will then be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned – except as noted below.

The ATM must be able to provide the following services to the customer:

1. A customer must be able to make a cash withdrawal from any suitable account linked to the card. Approval must be obtained from the bank before cash is dispensed.
2. A customer must be able to make a deposit to any account linked to the card, consisting of cash and/or checks in an envelope. The customer will enter the amount of the deposit into the ATM, subject to manual verification when the envelope is removed from the machine by an operator. Approval must be obtained from the bank before physically accepting the envelope.
3. A customer must be able to make a transfer of money between anytwo accounts linked to the card.
4. A customer must be able to make a balance inquiry of any account linked to thecard.
5. A customer must be able to abort a transaction in progress bypressing the Cancel key instead of responding to a request from the machine.

The ATM will communicate each transaction to the bank and obtain verification that it was allowed by the bank. Ordinarily, a transaction will be considered complete by the bank once it has been approved. In the case of a deposit, a second message will be sent to the bank indicating that the customer has deposited the envelope. (If the customer fails to deposit the envelope within the timeout period, or presses cancel instead, no second message will be sent to the bank and the deposit will not be credited to the customer.)

If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed. If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back.

If a transaction fails for any reason other than an invalid PIN, the ATM will display an explanation of the problem, and will then ask the customer whether he/she wants to do another transaction.

The ATM will provide the customer with a printed receipt for each successful transaction, showing the date, time, machine location, type of transaction, account(s), amount, and ending and available balance(s) of the affected account ("to" account for transfers).

The ATM will have a key-operated switch that will allow an operator to start and stop the servicing of customers. After turning the switch to the "on" position, the operator will be required to verify and enter the total cash on hand. The machine can only be turned off when it is not servicing a customer. When the switch is moved to the "off" position, the machine will shut down, so that the operator may remove deposit envelopes and reload the machine with cash, blank receipts, etc.

Objectives

The objective of this software is similar to ATM software installed in ATM center. It should first validate the pin in the ATM card. Then the type of transaction is enquired and the information from the customer is validated. If it is a withdrawal the amount is asked. After the money is delivered the transaction just made is updated in the database where the customer's information is stored.

Scope

The scope of the project is to design an ATM system that will help in completely automatic banking this software is going to be designed for withdrawal and deposit of money and register the transaction in the database where the customer's information is stored.

A) Name of the experiment: Class diagram for ATMSystem

1. AIM: To design and implement ATM system through Class Diagram

Purpose:

The purpose of the class diagram is to model the static view of an application. The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction. The UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application but class diagram is a bit different. So it is the most popular UML diagram in the coder community. So the purpose of the class diagram can be summarized as:

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

Contents:

Class diagrams commonly contain the following things

- Classes
- Interfaces
- Collaborations
- Dependency, generalization and association relationships

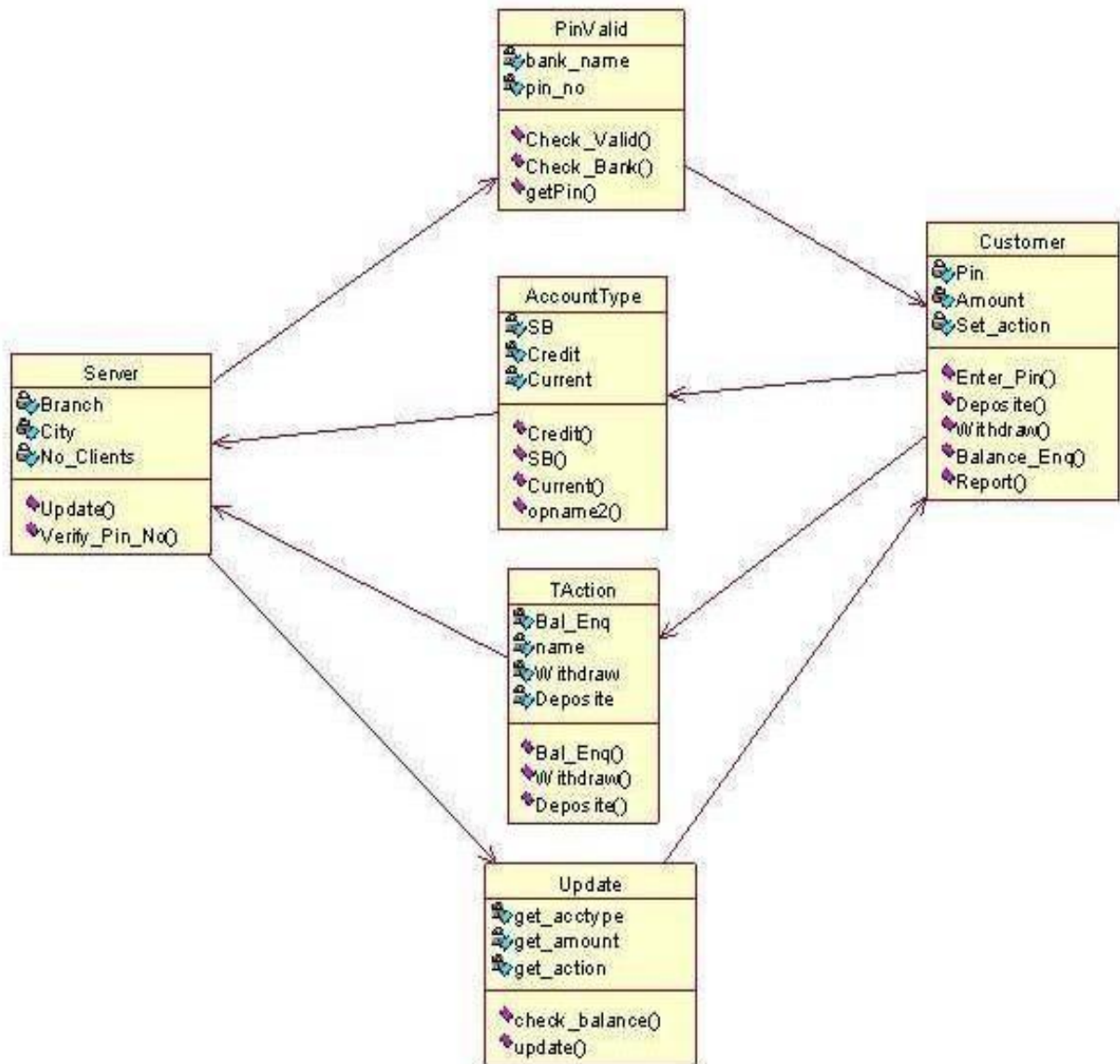
Procedure:-

Step1: First Classes are created.

Step2: Named as PinValid, Account Type, Transaction, Update, Server, Customer classes are created.

Step3: Appropriate relationships are provided between them as association.

DIAGRAM:



Inferences:

1. understand the concept of classes
2. identify classes and attributes and operations for a class
3. model the class diagram for the system

Applications:

Online transaction
Online banking

B) NAME OF EXPERIMENT: Use case diagram for ATMSystem.

AIM: To design and implement ATM System through Use case Diagram.

Purpose:

The purpose of use case diagram is to capture the dynamic aspect of a system. Because other four diagrams (activity, sequence, collaboration and State chart) are also having the same purpose. So we will look into some specific purpose which will distinguish it from other four diagrams. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

So in brief, the purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

Procedure:

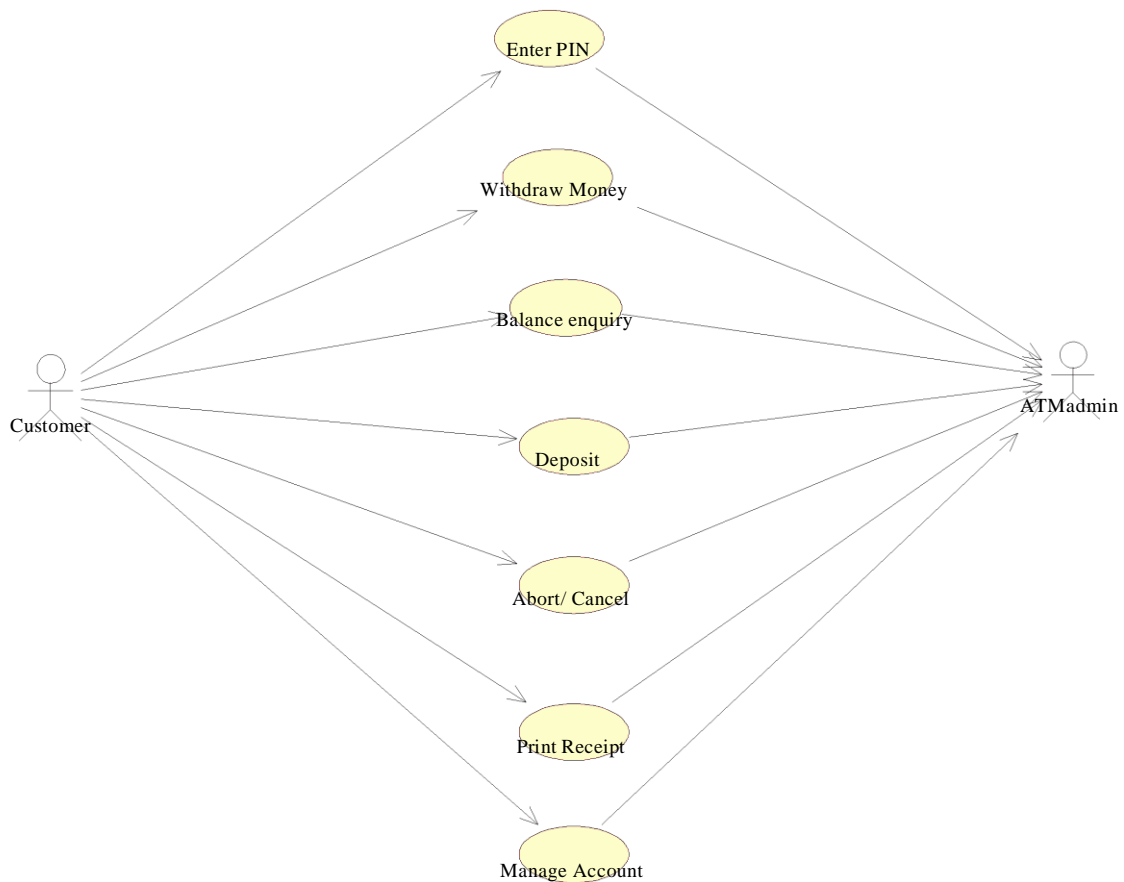
Step1: First an Actor is Created and named as User/Customer.

Step2: Secondly a system is created for ATM.

Step3: A use case Enter PIN, Withdraw money is created and connected with user as association relationship.

Step4: Similarly various use cases like Deposit money, Balance Enquiry, Manage Account etc are created and appropriate relationships are associated with each of them.

DIAGRAM:



Withdrawal UseCase

A withdrawal transaction asks the customer to choose a type of account to withdraw from (e.g. checking) from a menu of possible accounts, and to choose an amount from a menu of possible amounts. The system verifies that it has sufficient money on hand to satisfy the request before sending the transaction to the bank. (If not, the customer is informed and asked to enter a different amount.) If the transaction is approved by the bank, the appropriate amount of cash is dispensed by the machine before it issues a receipt. A withdrawal transaction can be cancelled by the customer pressing the Cancel key any time prior to choosing the amount.

Deposit UseCase

A deposit transaction asks the customer to choose a type of account to deposit to (e.g. checking) from a menu of possible accounts, and to type in amount on the keyboard. The transaction is initially sent to the bank to verify that the ATM can accept a deposit from this customer to this account. If the

transaction is approved, the machine accepts an envelope from the customer containing cash and/or checks before it issues a receipt. Once the envelope has been received, a second message is sent to the bank, to confirm that the bank can credit the customer's account – contingent on manual verification of the deposit envelope contents by an operator later.

A deposit transaction can be cancelled by the customer pressing the Cancel key any time prior to inserting the envelope containing the deposit. The transaction is automatically cancelled if the customer fails to insert the envelope containing the deposit within a reasonable period of time after being asked to do so.

InquiryUseCase

An inquiry transaction asks the customer to choose a type of account to inquire about from a menu of possible accounts. No further action is required once the transaction is approved by the bank before printing the receipt. An inquiry transaction can be cancelled by the customer pressing the Cancel key any time prior to choosing the account to inquire about.

ValidateUserUsecase:

This use case is for validate the user i.e. check the pin number, when the bank reports that the customer's transaction is disapproved due to an invalid PIN. The customer is required to re-enter the PIN and the original request is sent to the bank again. If the bank now approves the transaction, or disapproves it for some other reason, the original use case is continued; otherwise the process of re-entering the PIN is repeated. Once the PIN is successfully re-entered

If the customer fails three times to enter the correct PIN, the card is permanently retained, a screen is displayed informing the customer of this and suggesting he/she contact the bank, and the entire customer session is aborted.

PrintBillusecase

This usecase is for printing corresponding bill after transactions (withdraw or deposit, or balance enquiry, transfer) are completed.

ManageAccount

This use case is for updating corresponding user accounts after transactions (withdraw or deposit or transfer) are completed.

RESULT:

Inferences:

1. Identification of use cases.
2. Identification of actors.

INTERACTIONDIAGRAMS

We have two types of interaction diagrams in UML. One is sequence diagram and the other is a collaboration diagram. The sequence diagram captures the time sequence of message flow from one object to another and the collaboration diagram describes the organization of objects in a system taking part in the message flow.

So the following things are to be identified clearly before drawing the interaction diagram:

1. Objects taking part in the interaction.
2. Message flows among the objects.
3. The sequence in which the messages are flowing.
4. Object organization.

Purpose:

1. To capture dynamic behavior of a system.
2. To describe the message flow in the system.
3. To describe structural organization of the objects.
4. To describe interaction among objects.

Contents of a Sequence Diagram

Objects
Focus of control
Messages
Life line

Contents of a Collaboration Diagram

Objects
Links
Messages

C) NAME OF EXPERIMENT: Sequence diagram for ATMSystem.

AIM: To design and implement ATM System through Sequence Diagram.

Procedure:-

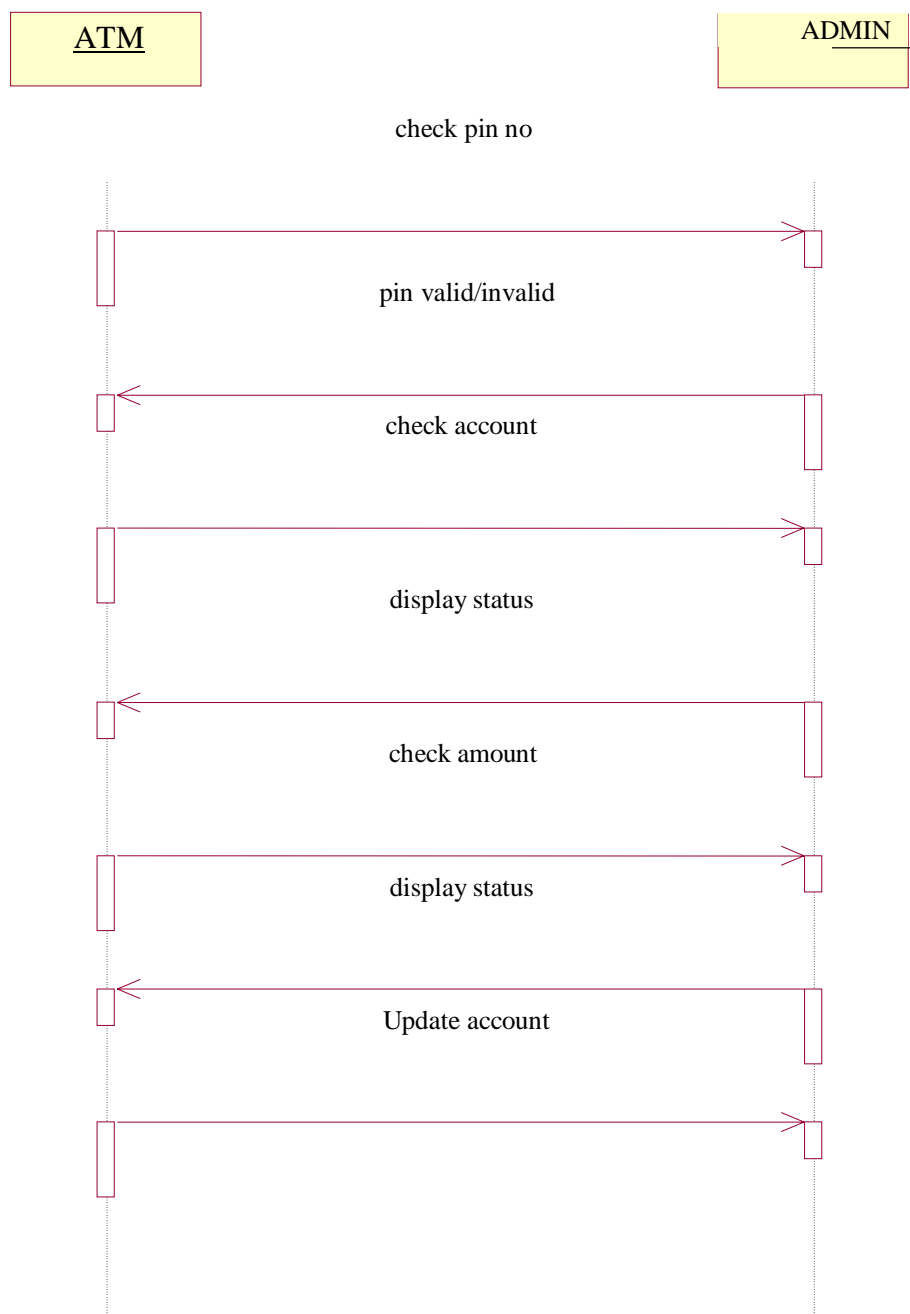
Step1: First An actor is created and named as user.

Step2: Secondly an object is created for Atm.

Step3: Timelines and lifelines are created automatically for them.

Step4: In sequence diagram interaction is done through time ordering of messages. So appropriate messages are passed between user and ATM is as shown in the figure.

DIAGRAM:



D) NAME OF EXPERIMENT: collaboration for ATMSystem.

AIM: To design and implement ATM System through Collaboration diagram.

Procedure:-

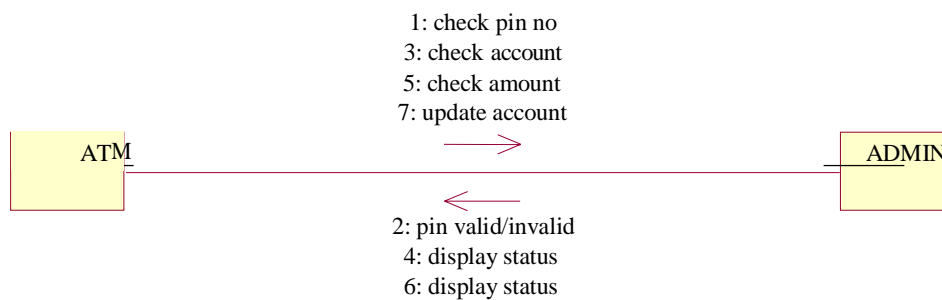
Step1: First an actor is created and named as user.

Step2: Secondly an object is created for ATM.

Step3: In collaboration diagram interaction is done through organization.

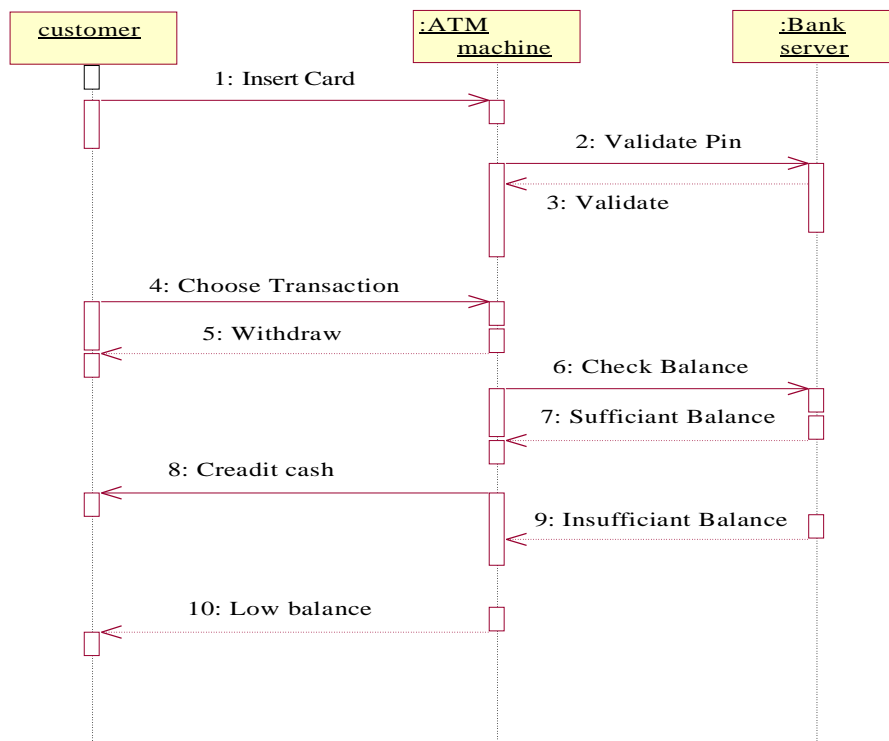
Step4: So appropriate messages are passed between user and ATM as shown in the figure.

DIAGRAM:

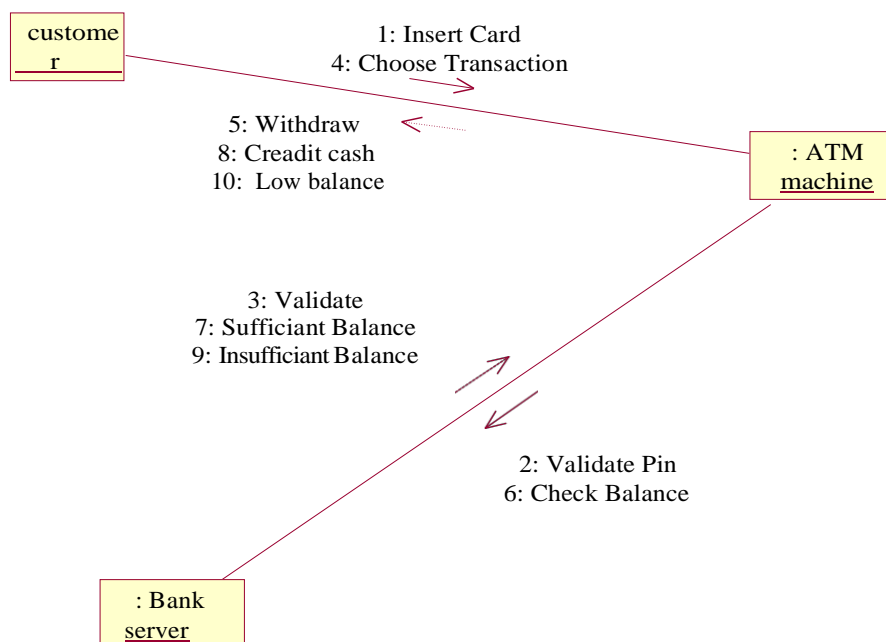


WITHDRAWUseCase:

SEQUENCE DIAGRAM

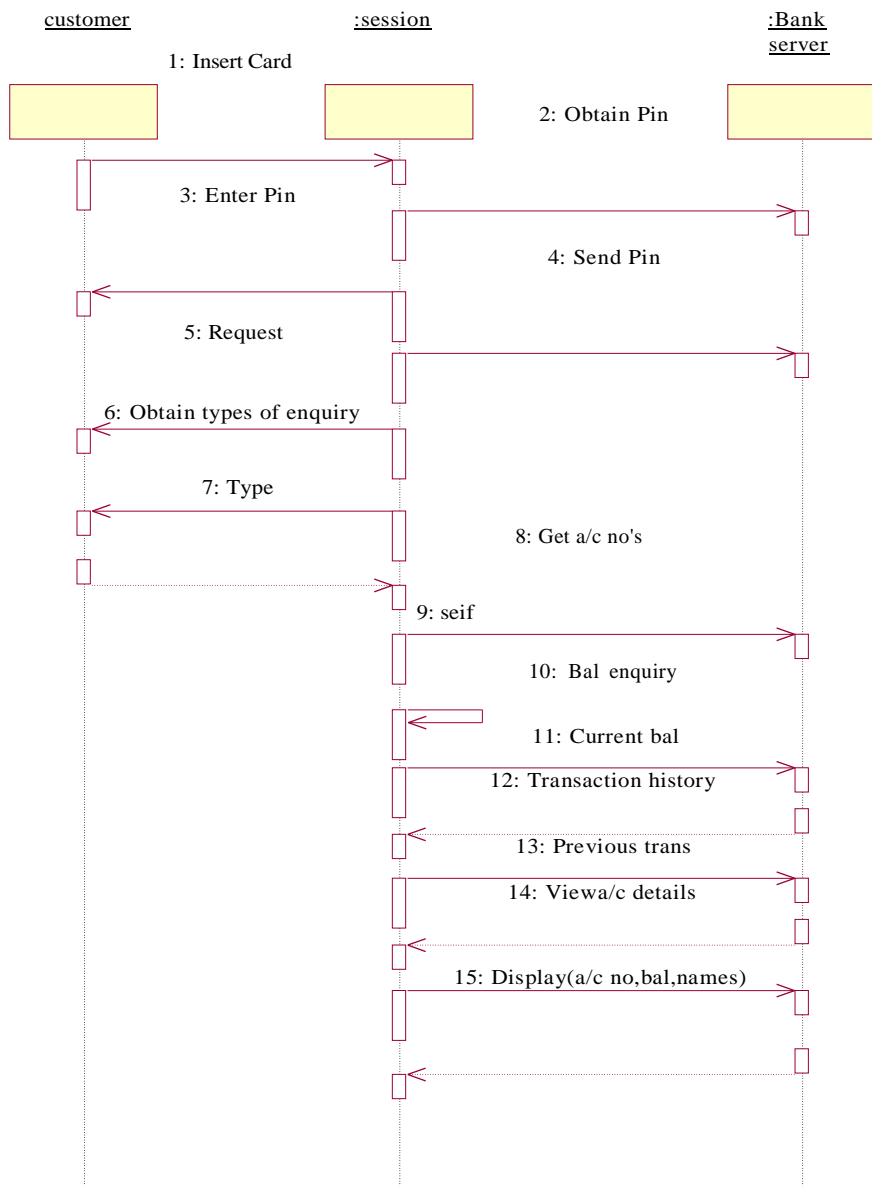


COLLABORATION DIAGRAM

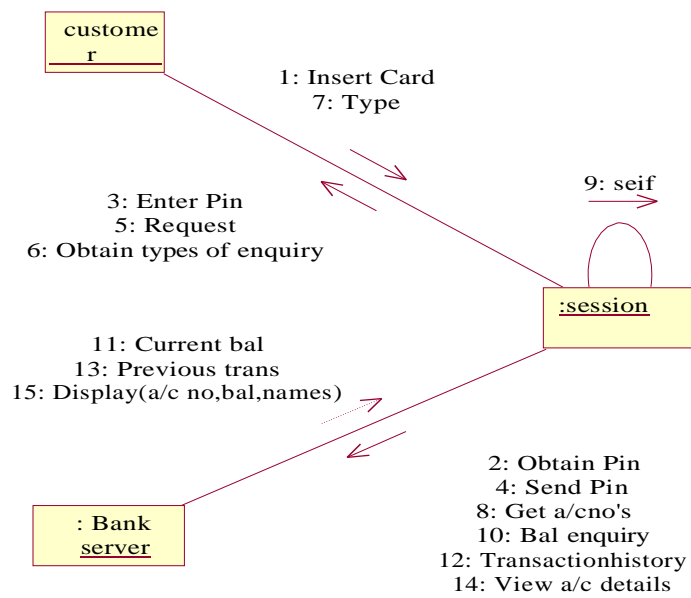


ENQUIRY UseCase:

SEQUENCE DIAGRAM:

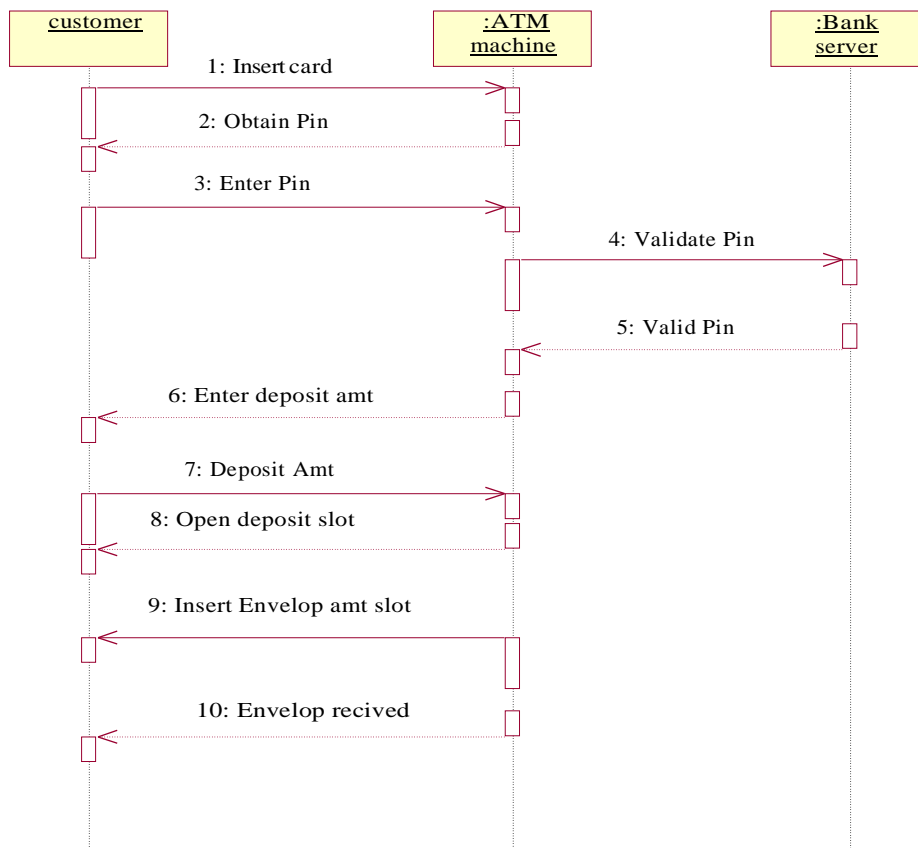


COLLABARATION DIAGRAM:

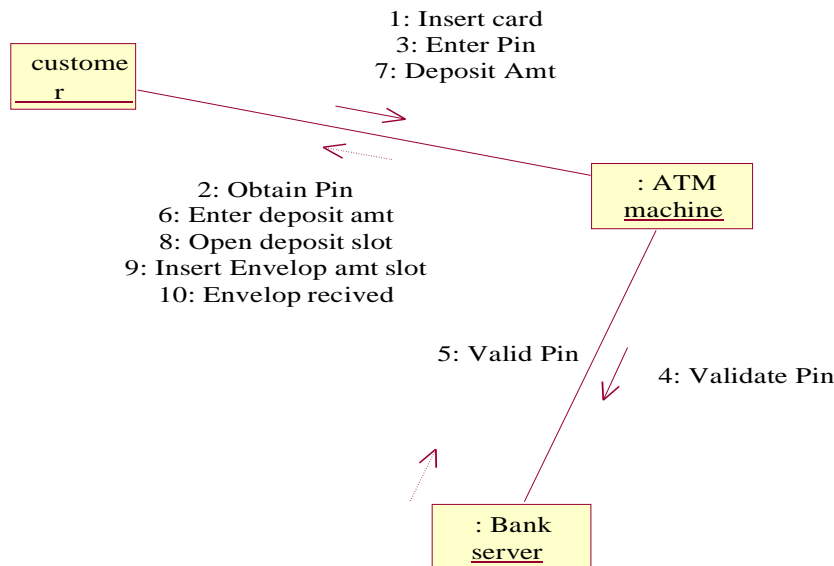


DEPOSITUseCase:

SEQUENCE DIAGRAM:



COLLABARATION DIAGRAM:



STATECHARTDiagram

State Chart diagram is used to model dynamic nature of a system. They define different states of an object during its lifetime. And these states are changed by events. State chart diagram describes the flow of control from one state to another state.

States are defined as a condition in which an object exists and it changes when some event is triggered. But the main purpose is to model reactive system.

Contents

- Simply state and composite states
- Transitions, including events and actions

E) NAME OF EXPERIMENT: State chart diagram for ATMSystem.

AIM: To design and implement ATM System through State Chart diagram.

Purpose:

Following are the main purposes of using State chart diagrams:

1. To model dynamic aspect of a system.
2. To model life time of a reactive system.

3. To describe different states of an object during its life time.
4. Defines a state machine to model states of an object.

Procedure:-

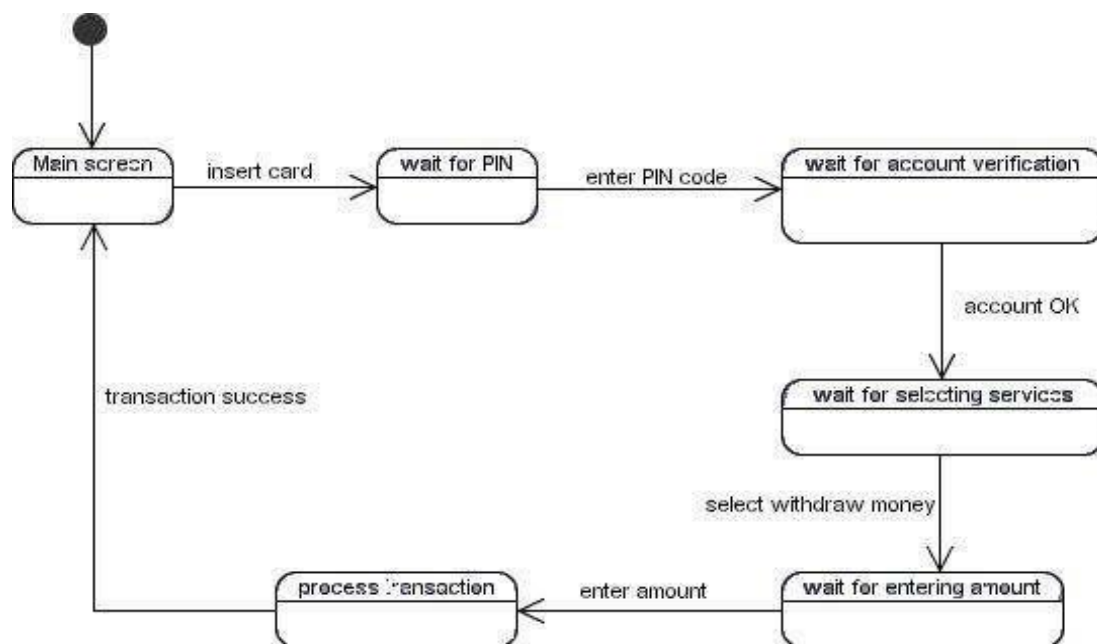
Step1: First after initial state control undergoes transition to ATM screen.

Step2: After inserting card it goes to the state wait for pin.

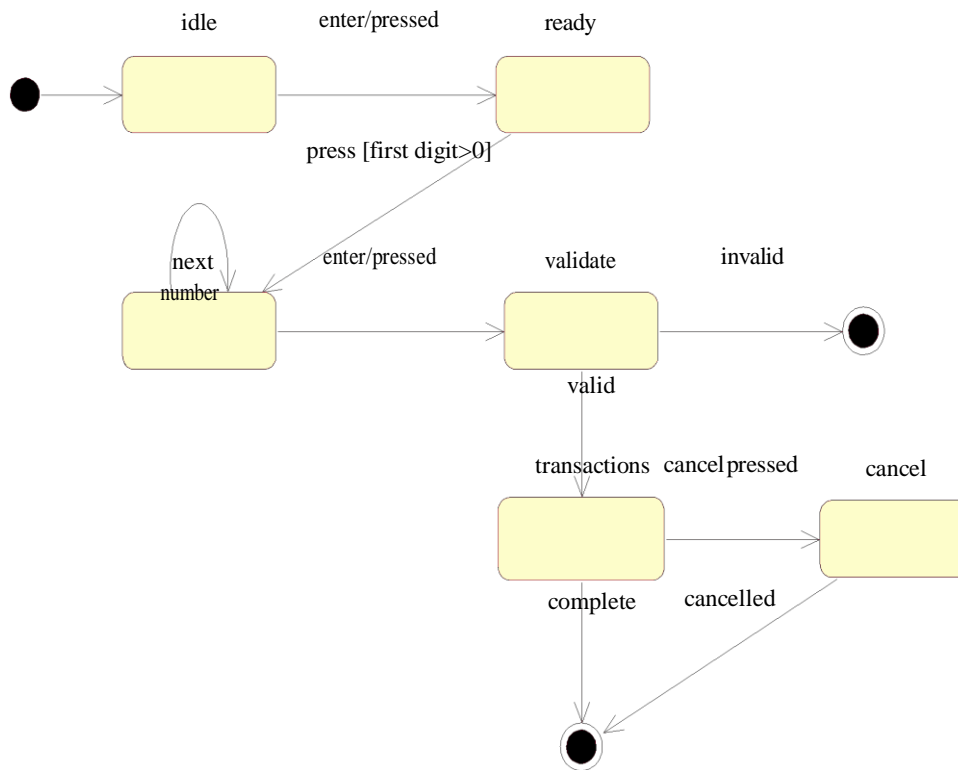
Step3: After entering pin it goes to the state account verification.

Step4: In this way it undergoes transitions to various states and finally reaches the ATM screen state as shown in the fig.

DIAGRAM:



STATE CHART FOR ATM



F) NAME OF EXPERIMENT: Activity diagram for ATM System.

AIM: To design and implement ATM System through Activity Diagram.

THEORY: An activity diagram shows the flow from activity to activity. An activity is an ongoing non atomic execution within a state machine. Activities ultimately results in some action, which is made up of executable atomic computations. We can use these diagrams to model the dynamic aspects of a system.

Activity diagram is basically a flow chart to represent the flow from one activity to another. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow by using elements like fork, join etc.

Contents

Initial/Final State, Activity, Fork & Join, Branch, Swim lanes

Fork

A fork represents the splitting of a single flow of control into two or more concurrent Flow of control. A fork may have one incoming transition and two or more outgoing transitions, each of which represents an independent flow of control. Below fork the activities associated with each of these path continues in parallel.

Join

A join represents the synchronization of two or more concurrent flows of control. A join may have two or more incoming transition and one outgoing transition. Above the join the activities associated with each of these paths continues in parallel.

Branching

A branch specifies alternate paths takes based on some Boolean expression Branch is represented by diamond Branch may have one incoming transition and two or more outgoing one on each outgoing transition, you place a Boolean expression shouldn't overlap but they should cover all possibilities.

Swimlane:

Swimlanes are useful when we model workflows of business processes to partition the activity states on an activity diagram into groups. Each group representing the business organization responsible for those activities, these groups are called Swimlanes .

Procedure:-

Step1: First initial state is created.

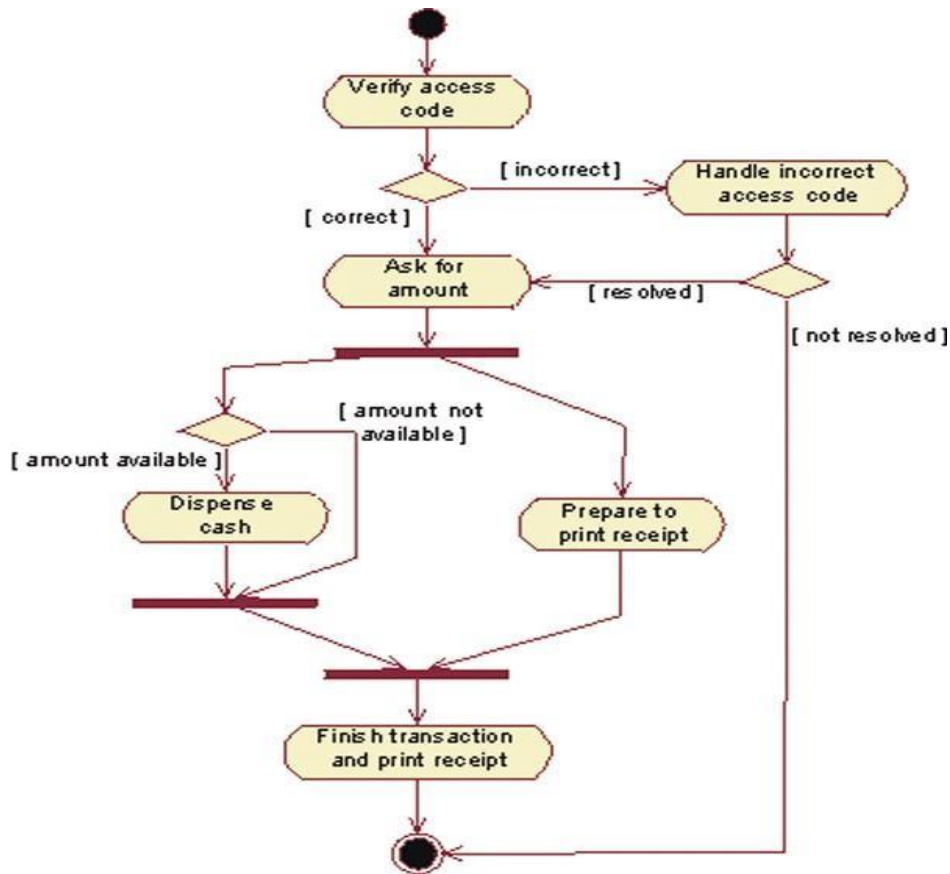
Step2: After that it goes to the action state insert card.

Step3: Next it undergoes transition to the state enter pin

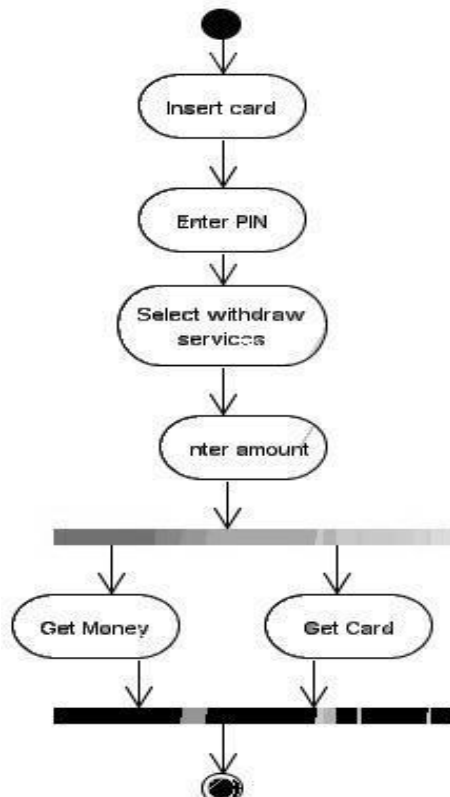
Step4: In this way it undergoes transitions to the various states.

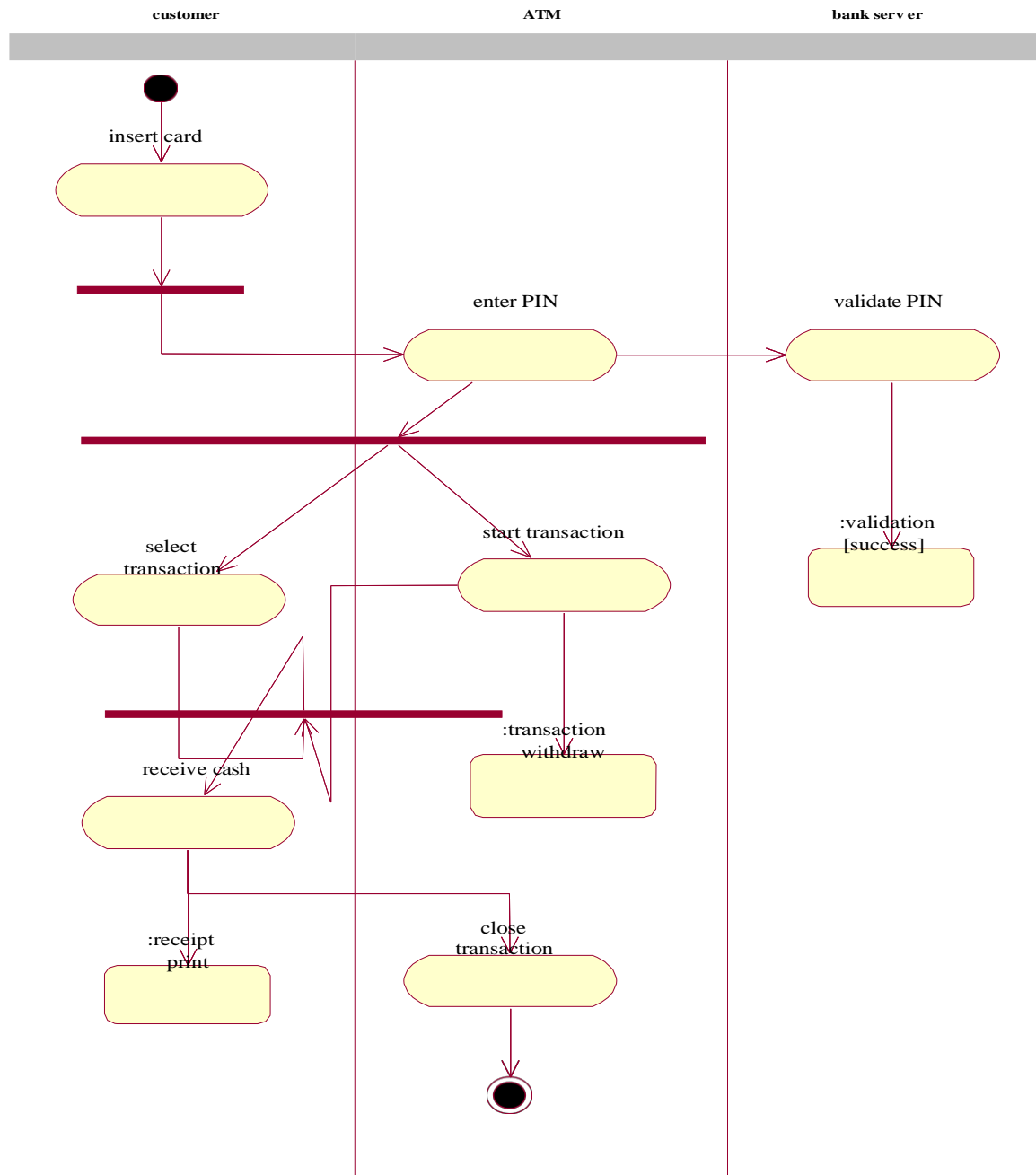
Step5: Use forking and joining wherever necessary.

DIAGRAM: Activity diagram for Transactions:



Activity diagram for Withdraw:





Inferences:

1. Identify the action states of the objects .
2. Understand the transitions and events for various objects.

G) NAME OF EXPERIMENT: Component diagram for ATMSystem.

AIM: To design and implement Component diagram for ATM System.

THEORY:

Component diagrams are used to model physical aspects of a system. Physical aspects are the elements like executables, libraries, files, documents etc which resides in a node. So component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Purpose:

Component diagrams can be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams are used to represent the whole.

Before drawing a component diagram the following artifacts are to be identified clearly:

- Files used in the system.
- Libraries and other artifacts relevant to the application.
- Relationships among the artifacts.
- Now after identifying the artifacts the following points needs to be followed:
 - Use a meaningful name to identify the component for which the diagram is to be drawn.
 - Prepare a mental layout before producing using tools.
 - Use notes for clarifying important points.

Contents

Components, Interfaces, Relationships

Procedure:-

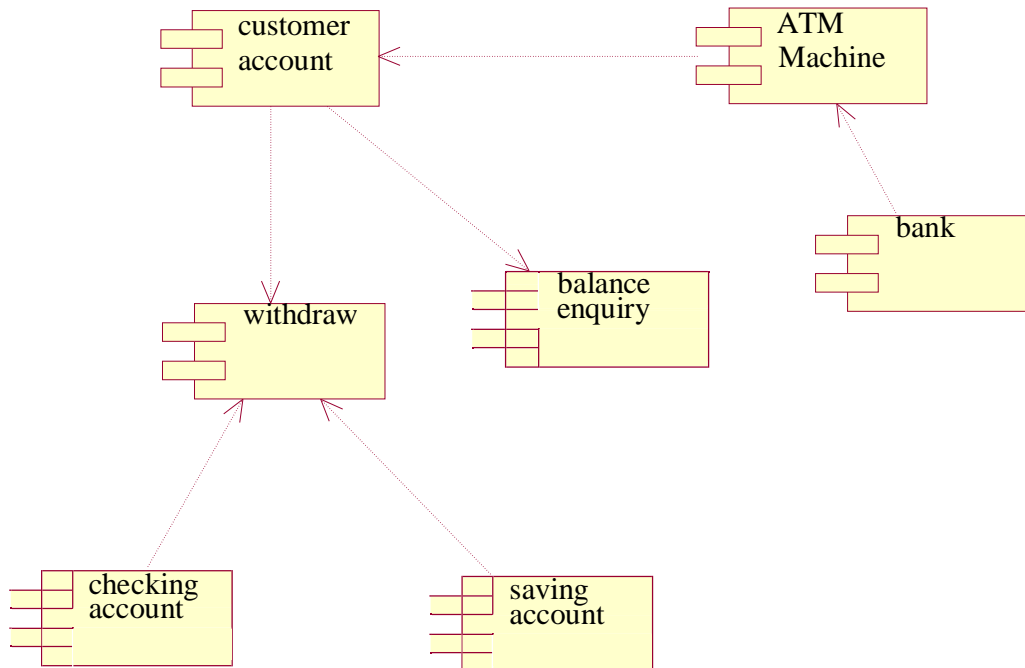
Step1: First user component is created.

Step2: ATM system package is created.

Step3: In it various components such as withdraw money, deposit money, check balance, transfer money etc. are created.

Step4: Association relationship is established between user and other components.

DIAGRAM:



H) NAME OF EXPERIMENT: Deployment diagram for ATMSystem.

AIM: To design and implement ATM System through Deployment diagram.

Purpose:

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

Contents: Nodes, Dependency & Association relationships

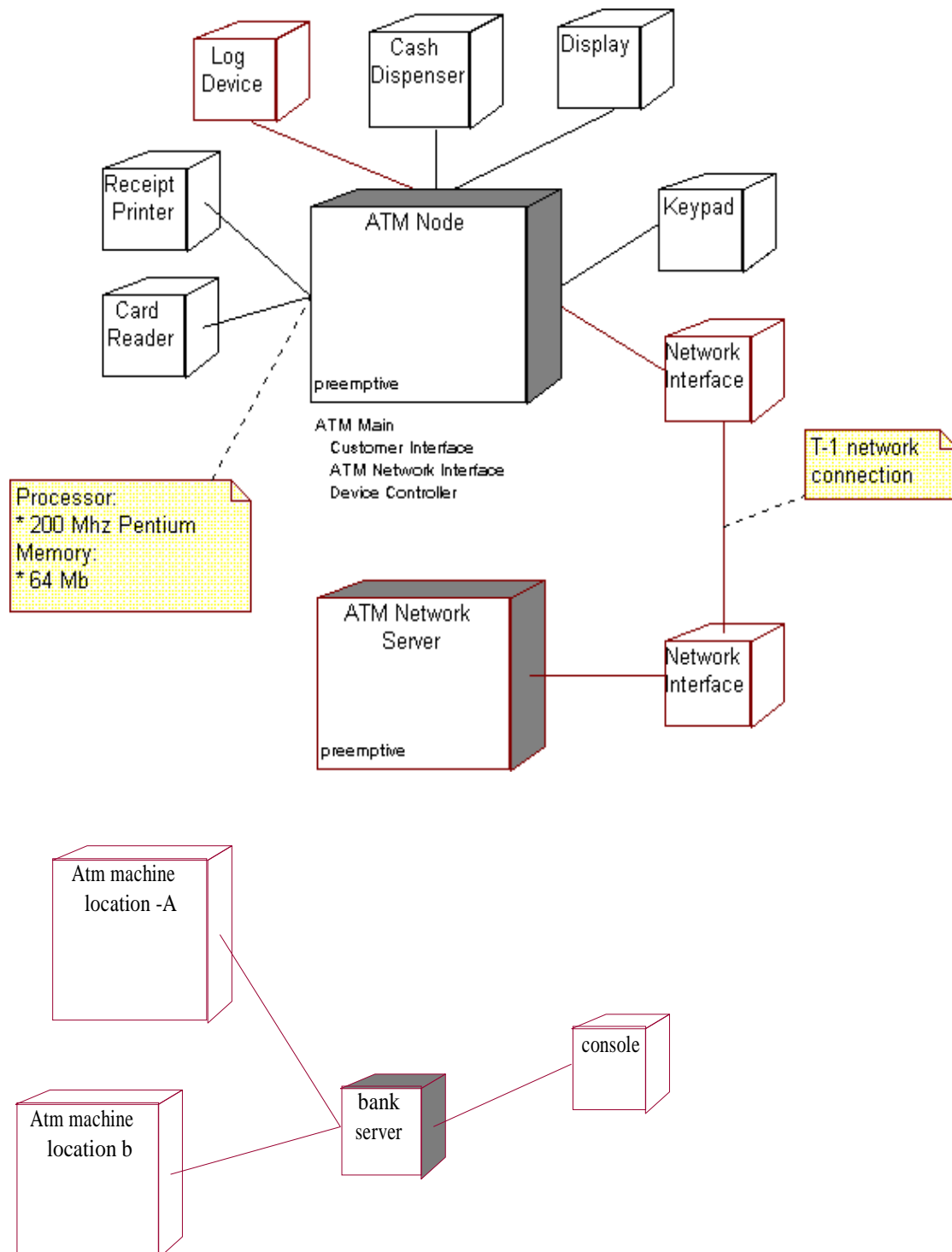
Procedure:-

Step1: First user node is created

Step2: various nodes withdraw money, deposit money, and check balance, transfer money etc. are created.

Step4: Association relationship is established between user and other nodes.

Step5: Dependency is established between deposit money and check balance.



SOFTWARE TESTING

EXPERIMENT: 1

NAME OF THE EXPERIMENT: Write program in „C,, language to demonstrate the working of the following constructs

i.)do..while:

Syntax:

Iteration-statement:

do statement while (expression) ;

Example:

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    int i = 10;
```

```
    do{
```

```
        printf("Hello %d\n", i );
```

```
        i = i -1;
```

```
        if( i == 6 )
```

```
        {
```

```
            break;
```

```
        }
```

```
    }while ( i > 0 );
```

```
}
```

ii.) do..while

```
while(condition)
{
    Loop body
    Increment or decrement;
}
```

Example:

```
#include<stdio.h>

int main()

{
    int counter, howmuch;

    scanf("%d", &howmuch);
    counter = 0;
    while ( counter < howmuch)
    {
        counter++;
        printf("%d\n", counter);
    }
    return 0;
}
```

iii) if...else

syntax:

```
if( condition 1 )
    statement1;
else if( condition 2 )
    statement2;
else if( condition 3 )
    statement3;
else
    statement4;
```

example:

```

#include<stdio.h>
int main(){
    int x,y;
    printf("Enter value for x :");
    scanf("%d",&x);
    printf("Enter value for y :");
    scanf("%d",&y);
    if ( x > y ){
        printf("X is large number - %d\n",x);
    }
    else{
        printf("Y is large number - %d\n",y);
    }
    return 0;
}

```

iv) switch

syntax:

```

switch(int/char const)
{
    Case const 1:stmt1;
    Break;
    Case const 2:stmt2;
    Break;
}
default: stmt n;
Break;
}

```

Example:

```
#include <stdio.h>
```

```

int main() {
    int color = 1;
    printf("Please choose a color(1: red,2: green,3: blue):\n");
    scanf("%d", &color);

    switch (color)
    {

```

```

    case 1:
        printf("you chose red color\n");
        break;
    case 2:
        printf("you chose green color\n");
        break;
    case 3:
        printf("you chose blue color\n");
        break;
    default:
        printf("you did not choose any color\n");
}
return 0;
}

```

v) for **Syntax:**

```

for (initialization; condition; increment/decrement)
{
    //body of the loop
}

```

Example:

```
#include <stdio.h>
```

```

int main()
{
    int x;
    /* The loop goes while x < 10, and x increases by one every loop*/ for ( x = 0; x
    < 10; x++ )
    {

        /* Keep in mind that the loop condition checks the
        conditional statement before it loops again. consequently,
        when x equals 10 the loop breaks. x is updated before the
        condition is checked. */

        printf( "%d\n", x );
    }
    getchar();
}

```

Viva questions:

1. How to find entered number is EVEN or ODD without using conditional statement (not using if.. else, while, do... while....., for ...)
2. Write a function to swap any two numbers?
3. How can we find out prime numbers from 1 to 50?

EXPERIMENT: 2

NAME OF THE EXPERIMENT:

Write a C program that uses functions to perform the following:

- ii) Multiplication of Two Matrices*/

```
#include<stdio.h>
#include<conio.h>
Void main()
{
    int a[10][10], b[10][10], m[10][10], i, j, p, q, r, s, k; clrscr ();
    printf(—enter the size of A Matrix\n);
    scanf(—%d %d\n, &p, &q);
    printf(—Enter the size of B Matrix \n);
    scanf(—%d %d —, &r, &s);
    if(q==r)
    {
        printf(—Enter the elements of matrix A:\n\n);
        for(i=0; i<p; i++)
        {
            for(j=0; j<q; j++)
            {
                scanf(—%d\n, &a[i][j]);
            }
        }
        printf(—Enter the elements of matrix b:\n\n); for(i=0; i<r; i++)
        {
            for(j=0; j<s; j++)
            {
                scanf(—%d\n, &b[i][j]);
            }
        }
        for (i=0; i<p; i++)
        {
            for(j=0; j<s; j++)
            {
                m[i][j]=0;
                for(k=0; k<q; k++)
                {
                    m[i][j] =m[i][j] +a[i][k] * b[k][j];
                }
            }
        }

        Printf(—matrix multiplication is:\n\n);
        for(i=0; i<p; i++)
        {
            for(j=0; j<s; j++)
            {
                Printf(—%d\t\n, m[i][j]);
            }
        }
    }
}
```

```

        `Printf(—\n\);
    }
}

else
printf(—matrix multiplication is not possible\);
getch();
}

```

FAILURE CASES:

output:

1. Enter the size of a: 2 3

Enter the size of b: 2 3

Matrix multiplication is not possible.

Reason to fail: to do multiplication of matrices the number of columns in matrix —a[] should be equal to number of rows in matrix —bl.

2. Enter the size of a: p q

Enter the size of b: q s

Matrix multiplication is not possible.

Reason to fail: to do multiplication of matrices the number of columns in matrix —all should be equal to number of rows in matrix —bl, and rows & columns should be integer values.

3. Enter the size of a: 1.5 2

Enter the size of b: 2 3

Matrix multiplication is not possible.

Reason to fail: to do multiplication of matrices the number of columns in matrix —all should be equal to number of rows in matrix —bl, and rows & columns should be integer values.

4. Enter the size of a: 350 480 Enter

the size of b: 480 620

Matrix multiplication is not possible.

Reason to fail: size of buffer will be not be sufficient to handle this multiplication.

5. Enter the size of a: -1 -2

Enter the size of b: -2 3

Matrix multiplication is not possible.

Reason to fail: to do multiplication of matrices the number of columns in matrix —all should be equal to number of rows in matrix —bl, and rows & columns should be positive integer values.

viva questions:

1. syntax for multiplication
2. syntax for matrix multiplication
3. what the logic for matrix multiplication?

EXPERIMENT: 3

NAME OF THE EXPERIMENT: ATM system specifications and report the various bugs

Purpose:

This document describes the software requirements and specification (SRS) for an automated teller machine (ATM) network. The document is intended for the customer and the developer (designers, testers, maintainers). The reader is assumed to have basic knowledge of banking accounts and account services. Knowledge and understanding of Unified Modeling Language (UML) diagrams is also required.

Scope:

The software supports a computerized banking network called _Bank24,. The network enables customers to complete simple bank account services via automated teller machines (ATMs) that may be located off premise and that need not be owned and operated by the customer,,s bank. The ATM identifies a customer by a cash card and password. It collects information about a simple account transaction (e.g., deposit, withdrawal, transfer, bill payment), communicates the transaction information to the customer,,s bank, and dispenses cash to the customer. The banks provide their own software for their own computers. The _Bank24,, software requires appropriate record keeping and security provisions. The software must handle concurrent accesses to the same account correctly.

Intended Audience:

The intended audience of this SRS consists of:

- ☐ Software designers
- ☐ Systems engineers
- ☐ Software developers
- ☐ Software testers
- ☐ Customers

The actors of the system are:

1. User
2. ATM Machine
3. Bank

Product Perspective:

An automated teller machine (ATM) is a computerized telecommunications device that provides the customers of a financial institution with access to financial transactions in a public space without the need for a human clerk or bank teller. On most modern ATMs, the customer is identified by inserting a plastic ATM card with a magnetic stripe or a plastic smartcard with a chip, that contains a unique card number and some security information, such as an expiration date or CVC (CVV). Security is provided by the customer entering a personal identification number (PIN).

Product functions:

Using an ATM, customers can access their bank accounts in order to make cash withdrawals (or credit

card cash advances) and check their account balances.

The functions of the system are:

1. Login
2. Get Balance Information
3. Withdraw Cash
4. Transfer Funds

Operating Environments:

The hardware,software and technology used should have following specifications:

- ☐ Ability to read the ATM card.
- ☐ Ability to count the currency notes.
- ☐ Touch screen for convenience.
- ☐ Keypad(in case touchpad fails)
- ☐ Continuous power supply.

- ☐ Ability to connect to bank's network.
- ☐ Ability to validate user.

Design/implementation constraints:

Login:

Validate Bank Card

- ☐ Validate for Card Expiration Date
- ☐ Validate that the card's expiration date is later than today's date
- ☐ If card is expired, prompt error message "Card is expired"

Validate for Stolen or Lost Card

- ☐ Validate that the card is not reported lost or stolen
- ☐ If card is lost, prompt error message, "Card has been reported lost"
- ☐ If card is stolen, prompt error message, "Card has been reported stolen"

Validate for Disabled Card

- ☐ Validate that the card is not disabled
- ☐ If card is disabled, prompt error message, "Card has been disabled as of <expiration date>"

Validate for Locked Account

Validate that the account is not locked

- ☐ If account is locked, prompt error message "Account is locked"

Validate PIN

- ☐ Validate that the password is not blank
- ☐ If PIN is blank, prompt error message "Please provide PIN"
- ☐ Validate that the password entered matches the password on file
- ☐ If password does not match, prompt error message "Password is Incorrect"

Lock Account

- ☐ If number of consecutive unsuccessful logins exceeds three attempts, lock account

Maintain Consecutive Unsuccessful Login Counter

Increment Login Counter

For every consecutive Login attempt, increment login counter by 1.

Reset login counter to 0 after login is successful.

Get Balance Information

Withdraw Cash

Transfer Funds

Assumptions and Dependencies:

- ☐ Hardware never fails
- ☐ ATM casing is impenetrable
- ☐ Limited number of transactions per day (sufficient paper for receipts)
- ☐ Limited amount of money withdrawn per day (sufficient money)

External Interface Requirements

User interfaces

The customer user interface should be intuitive, such that 99.9% of all new ATM users are able to

complete their banking transactions without any assistance.

Hardware interfaces

The hardware should have following specifications:

- ☐ Ability to read the ATM card
- ☐ Ability to count the currency notes
- ☐ Touch screen for convenience
- ☐ Keypad (in case touchpad fails)
- ☐ Continuous power supply
- ☐ Ability to connect to bank's network
- ☐ Ability to take input from user
- ☐ Ability to validate user

Software interfaces

The software interfaces are specific to the target banking software systems. At present, two known banking systems will participate in the ATM network.

- ☐ State Bank
- ☐ Indian Overseas Bank

Safety requirements:

Must be safe kept in physical aspects, say in a cabin

- ☐ Must be bolted to floor to prevent any kind of theft
- ☐ Must have an emergency phone outside the cabin
- ☐ There must be an emergency phone just outside the cabin
- ☐ The cabin door must have an ATM card swipe slot
- ☐ The cabin door will always be locked, which will open only when user swipes his/her ATM card in the slot & is validated as genuine

Security requirements:

- ☐ Users accessibility is censured in all the ways
- ☐ Users are advised to change their PIN on first use
- ☐ Users are advised not to tell their PIN to anyone
- ☐ The maximum number of attempts to enter PIN will be three

Some of the possible Bugs on ATM machine?

1. Successful insertion of ATM card
2. Unsuccessful operation due to insert card in wrong angle
3. Unsuccessful operation due to invalid account Ex: other bank card or time expired card
4. successful entry of PIN number
5. un successful operation due to enter wrong PIN number 3times
6. successful selection of language
7. successful selection of account type
8. unsuccessful operation due to invalid account type
9. successful selection of withdraw operation
10. successful selection of amount to be withdrawl
11. successful withdraw operation

12. unsuccessful withdraw operation due to wrong denominations
13. unsuccessful withdraw operation due to amount is greater than day limit
14. unsuccessful withdraw operation due to lack of money in ATM
15. unsuccessful withdraw operation due to amount is greater than possible balance
16. unsuccessful withdraw operation due to transactions is greater than day limit
17. unsuccessful withdraw operation due to click cancel after insert card
18. unsuccessful withdraw operation due to click cancel after insert card & pin number
19. unsuccessful withdraw operation due to click cancel after insert card , pin number & language
20. unsuccessful withdraw operation due to click cancel after insert card , pin number , language & account type
21. unsuccessful withdrawal operation due to click cancel after insert card , pin number , language , account type & withdrawal operation
22. unsuccessful withdrawal operation due to click cancel after insert card , pin number , language , account type , withdrawal operation & amount to be withdraw

EXPERIMENT: 4

NAME OF THE EXPERIMENT: Test cases for banking applications

Banking applications are considered to be one of the most complex applications in today's software development and testing industry. **What makes Banking application so complex?** What approach should be followed in order to test the complex workflows involved? In this article we will be highlighting different stages and techniques involved in testing Banking applications. **The characteristics of a Banking application are as follows:**

- ☐ Multi tier functionality to support thousands of concurrent user sessions
- ☐ Large scale Integration , typically a banking application integrates with numerous other applications such as Bill Pay utility and Trading accounts
- ☐ Complex Business workflows
- ☐ Real Time and Batch processing
- ☐ High rate of Transactions per seconds
- ☐ Secure Transactions
- ☐ Robust Reporting section to keep track of day to day transactions
- ☐ Strong Auditing to troubleshoot customer issues
- ☐ Massive storage system
- ☐ Disaster Management.

The above listed ten points are the **most important characteristics of a Banking application.**

Banking applications have multiple tiers involved in performing an operation. For Example, a **banking application may have:**

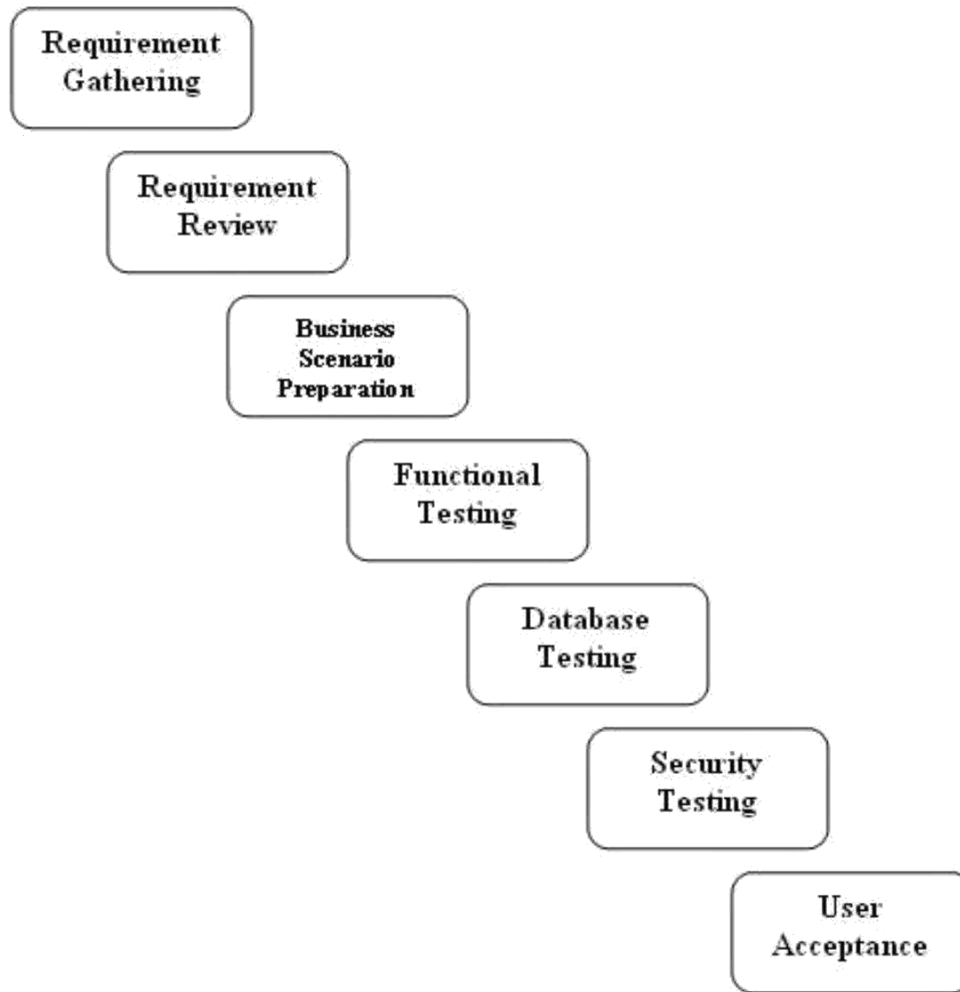
1. Web Server to interact with end users via Browser
2. Middle Tier to validate the input and output for web server
3. Data Base to store data and procedures
4. Transaction Processor which could be a large capacity Mainframe or any other Legacy system to carry out Trillions of transactions per second.

If we talk about testing banking applications it requires an **end to end testing methodology involving multiple software testing techniques to ensure:**

- ☐ Total coverage of all banking workflows and Business Requirements
- ☐ Functional aspect of the application
- ☐ Security aspect of the application
- ☐ Data Integrity
- ☐ Concurrency
- ☐ User Experience

Typical stages involved in testing Banking Applications are shown in below workflow which we will be discussing individually.

Banking Application Testing Workflow



1) Requirement **Gathering**:

Requirement gathering phase involves documentation of requirements either as Functional Specifications or Use Cases. Requirements are gathered as per customer needs and documented by Banking Experts or Business Analyst. To write requirements on more than one subject experts are involved as banking itself has multiple sub domains and one full fledged banking application will be the integration of all. For Example: A banking application may have separate modules for Transfers, Credit Cards, Reports, Loan Accounts, Bill Payments, Trading Etc.

2) Requirement **Review**:

The deliverable of Requirement Gathering is reviewed by all the stakeholders such as QA Engineers, Development leads and Peer Business Analysts. They cross check that neither existing business workflows nor new workflows are violated.

3) Business Scenario **Preparations**:

In this stage QA Engineers derive Business Scenarios from the requirement documents (Functions Specs or Use Cases); Business Scenarios are derived in such a way that all Business Requirements are covered. Business Scenarios are high level scenarios without any detailed steps, further these Business Scenarios are reviewed by Business Analyst to ensure all of Business Requirements are met and its easier for BAs to review high level scenarios than reviewing low level detailed Test Cases.

4) Functional Testing:

In this stage functional testing is performed and the usual software testing activities are performed such as:

Test Case Preparation:

In this stage Test Cases are derived from Business Scenarios, one Business Scenario leads to several positive test cases and negative test cases. Generally tools used during this stage are Microsoft Excel, Test Director or Quality Center.

Test Case Review:

Reviews by peer QA Engineers

Test Case Execution:

Test Case Execution could be either manual or automatic involving tools like QC, QTP or any other.

5) Database Testing:

Banking Application involves complex transaction which are performed both at UI level and Database level, Therefore Database testing is as important as functional testing. Database in itself is an entirely separate layer hence it is carried out by database specialists and it uses techniques like

- ☐ Data loading
- ☐ Database Migration
- ☐ Testing DB Schema and Data types
- ☐ Rules Testing
- ☐ Testing Stored Procedures and Functions
- ☐ Testing Triggers
- ☐ Data Integrity

6) Security Testing:

Security Testing is usually the last stage in the testing cycle as completing functional and non functional are entry criteria to commence Security testing. Security testing is one of the major stages in the entire Application testing cycle as this stage ensures that application complies with Federal and Industry standards. Security testing cycle makes sure the application does not have any web vulnerability which may expose sensitive data to an intruder or an attacker and complies with standards like OWASP.

In this stage the major task involves in the whole application scan which is carried out using tools like IBM Appscan or HP WebInspect (2 Most popular tools).

Once the Scan is complete the Scan Report is published out of which False Positives are filtered out and rest of the vulnerability are reported to Development team for fixing depending on the Severity.

Other **Manual tools for Security Testing** used are: Paros Proxy, Http Watch, Burp Suite, Fortify tools Etc.

Apart from the above stages there might be different stages involved like Integration Testing and Performance Testing.

In today,,s scenario **majority of Banking Projects are using:** Agile/Scrum, RUP and Continuous Integration methodologies, and Tools packages like Microsoft,,s VSTS and Rational Tools.

As we mentioned RUP above, RUP stands for Rational Unified Process, which is an iterative software development methodology introduced by IBM which comprises of four phases in which development and testing activities are carried out.

Four phases are:

- i) Inception
- ii) Collaboration
- iii) Construction and
- iv) Transition

RUP widely involves IBM Rational tools.

In this article we discussed **how complex a Banking application could be** and what are the **typical phases involved in testing the application**. Apart from that we also discussed current trends followed by IT industries including software development methodologies and tools.

Test cases for opening bank account

1. Input parameters checking -

Name

-Date of Birth -

Photo -Address

Proof -Identity

proof

-Introducers (if applicable) -

PAN card

-Initial deposit

-Whether checkbook / ATM card / Online banking facilities are needed or not -Customer

signature

Type of account -

Savings account -Salary

account -Joint account -

Current account -

Secondary account -RD

account

-Account for a company

Test cases

-Checking mandatory input parameters -Checking

optional input parameters -Check whether able to
create account entity.

-Check whether you are able to deposit an amount in the newly created account (and thus updating the balance)

-Check whether you are able to withdraw an amount in the newly created account (after deposit) (and thus updating the balance)

-Check whether company name and its pan number and other details are provided in case of salary account

-Check whether primary account number is provided in case of secondary account

-Check whether company details are provided in cases of company's current account

-Check whether proofs for joint account is provided in case of joint account

-Check whether you are able deposit an account in the name of either of the person in an joint

account.

-Check whether you are able withdraw an account in the name of either of the person in an joint account.

-Check whether you are able to maintain zero balance in salary account

-Check whether you are not able to maintain zero balance (or mini balance) in non-salary account.

viva questions

1. Can you explain boundary value analysis?
2. Can you explain equivalence partitioning?
3. Can you explain random/monkey testing?
4. What are semi-random test cases?
5. What is negative and positive testing?
6. How did you define severity ratings in your project?

EXPERIMENT: 5

NAME OF THE EXPERIMENT: Test plan document for library application

The Library Management System is an online application for assisting a librarian in managing a book library in a University. The system would provide basic set of features to add/update clients, add/update books, search for books, and manage check-in / checkout processes. Our test group tested the system based on the requirement specification.

INTRODUCTION

This test report is the result for testing in the LMS. It mainly focuses on two problems: what we will test and how we will test.

Result

GUI test

Pass criteria: librarians could use this GUI to interface with the backend library database without any difficulties

Result: pass

Database test

Pass criteria: Results of all basic and advanced operations are normal (refer to section 4)

Result: pass

Basic function test

Add a student

Pass criteria:

- ☐ Each customer/student should have following attributes: Student ID/SSN (unique), Name, Address and Phone number.

Result: pass

- ☐ The retrieved customer information by viewing customer detail should contain the four attributes.

Result: pass

Update/delete student

Pass criteria:

- ☐ The record would be selected using the student ID Result: pass
- ☐ Updates can be made on full. Items only: Name, Address, Phone number Result: pass
- ☐ The record can be deleted if there are no books issued by user.

Result: Partially pass. When no books issued by user, he can be deleted. But when there are books Issued by this user, he was also deleted. It is wrong.

- ☐ The updated values would be reflected if the same customer's ID/SSN is called for.

Result: pass

- ☐ If customer were deleted, it would not appear in further search queries.

Result: pass

Add a book

Pass criteria:

- ☐ Each book shall have following attributes: Call Number, ISBN, Title, Author name. Result: pass
- ☐ The retrieved book information should contain the four attributes.

Result: pass

Update/delete book

Pass criteria:

- ☐ The book item can be retrieved using the call number Result: did not pass. Can not retrieve using the call number
- ☐ The data items which can be updated are: ISBN, Title, Author name Result: pass
- ☐ The book can be deleted only if no user has issued it.

Result: partially pass. When no user has issued it, pass. When there are user having issued it, did not pass

- ☐ The updated values would be reflected if the same call number is called for Result: pass
- ☐ If book were deleted, it would not appear in further search queries. Result: pass

Search for book

Pass criteria:

- ☐ The product shall let Librarian query books,, detail information by their ISBN number or Author or Title.

Result: pass

- ☐ The search results would produce a list of books, which match the search parameters with following Details: Call number, ISBN number, Title, Author Result: pass
- ☐ The display would also provide the number of copies which is available for issue Result: pass
- ☐ The display shall provide a means to select one or more rows to a user-list Result: pass
- ☐ A detailed view of each book should provide information about check-in/check out status, with the borrower,,s information.

Result: pass

- ☐ The search display will be restricted to 20 results per page and there would be means to navigate from sets of search results.

Result: pass

- ☐ The user can perform multiple searches before finally selecting a set of books for check in or checkout. These should be stored across searches.

Result: pass

- ☐ A book may have more than one copy. But every copy with the same ISBN number should have same detail information.

Result: pass

- ☐ The borrower,,s list should agree with the data in students,, account

Result: pass

Check-in book

Pass criteria:

- ☐ Librarians can check in a book using its call number

Result: pass

- ☐ The check-in can be initiated from a previous search operation where user has selected a set of books.

Result: pass

- ☐ The return date would automatically reflect the current system date. Result: did not pass.

- ☐ Any late fees would be computed as difference between due date and return date at rate of 10 cents a day.

Result: did not pass

- ☐ A book, which has been checked in once, should not be checked in again Result: pass

Check-out book

Pass criteria:

- ☐ Librarians can check out a book using its call number Result: pass

- ☐ The checkout can be initiated from a previous search operation where user has selected a set of books.

Result: pass

- ☐ The student ID who is issuing the book would be entered

Result: pass

- ☐ The issue date would automatically reflect the current system date. Result: did not pass

- ☐ The due date would automatically be stamped as 5 days from current date. Result: did not pass
- ☐ A book, which has been checked out once, should not be checked out again Result: pass
- ☐ A student who has books due should not be allowed to check out any books Result: did not pass
- ☐ The max. No of books that can be issued to a customer would be 10. The system should not allow checkout of books beyond this limit.
Result: pass

[View book detail](#)

Pass criteria:

- ☐ This view would display details about a selected book from search operation Result: pass
- ☐ The details to be displayed are: Call number, IBN, Title, Author, Issue status (In library or checked out), If book is checked out it would display, User ID & Name, Checkout date, Due date Result: for checkout date and due date, did not pass
- ☐ Books checked in should not display user summary
Result: pass
- ☐ Books checked out should display correct user details. Result:
pass

View student detail

Pass criteria:

- ☐ Librarians can select a user record for detailed view Result:
pass
- ☐ The detail view should show:
 - a. User name, ID, Address & Phone number Result:
pass
 - b. The books issued by user with issue date, due date, call number, title Result: did not pass
 - c. Late fees & Fines summary and total
Result: did not pass
- ☐ The display should match existing user profile Result:
pass
- ☐ The books checked out should have their statuses marked Result: pass
- ☐ The book search query should show the user id correctly. Result: pass

Network test

Pass criteria: Results of operations (ping, ftp and ODBC connectivity check) are normal

Result: did not test this item, because no enough machines and no available envirenment.

Viva questions

1. How to create a test plan document for Library Management System?
2. what is object repository
3. How many test cases can u write
write 5 if software failed in customer environment what we called a)error b)fault c)defect d)failure
- 4.What test plan should contains?
5. What is test strategy?
6. Define test Plan?What is the difference between Master Test Plan and Test Plan?

EXPERIMENT: 6

NAME OF THE EXPERIMENT: Study of Any Testing Tool(WinRunner)

WinRunner is a program that is responsible for the automated testing of software. WinRunner is a Mercury Interactive,,s enterprise functional testing tool for Microsoft windows applications.

Testing an Application using WinRunner

After installing the WinRunner on your computer, invoke the WinRunner application:

Start -> Programs ->WinRunner ->WinRunner

The opening screen of the WinRunner application is displayed, prompting you to select one of the three options:

New Test: To create a new test script
Open Test: To open an existing test script
Quick Preview: To view the quick preview of WinRunner



Recording Test Cases

There are two modes of recording:

Context Sensitive mode: This mode of recording is used when the location of the GUI controls (i.e. X and Y coordinates) or the mouse positions are not necessary. Analog mode: This mode of recording is used when the mouse positions, the location of the controls in the application, also play an important role in testing the application. This mode of recording has to be used to validate bitmaps, testing the signature etc.

The procedure for recording a test case is as follows:

Step 1: Open a new document: File -> New (or) Select "New Test" from the WinRunner's Welcome screen.

Step 2: Open (run) the application to be tested.

Step 3: Start recording a test case. Create ->Record - Context Sensitive (or) click on the toolbar's "Record" button once, to record in Context Sensitive mode.

Step 4: Select the application to be tested by clicking on the application's title bar.

Step 5: Perform all the actions to be recorded.

Step 6: Once all required actions are recorded, stop the recording. Create -> Stop (or) Click on the toolbar's "Stop" button to stop the recording WinRunner generates the script for the recorded actions.

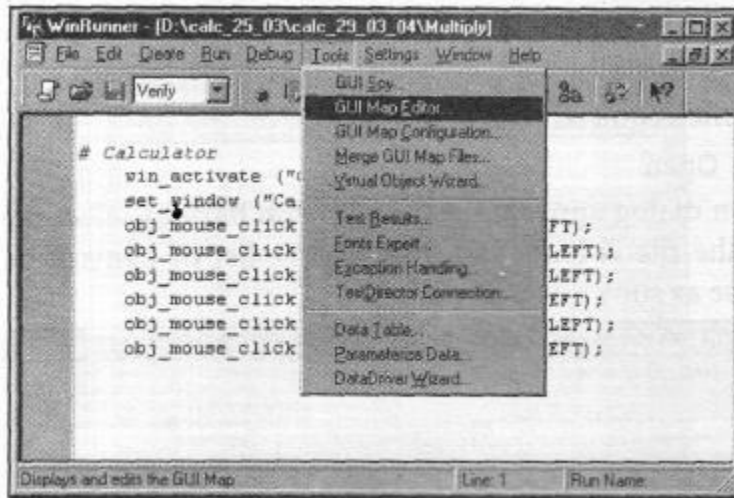
There are two modes for generating the test cases: "Global GUI map file mode" and "GUI map file per test mode". By default, it is in "Global GUI map file mode".

In Global GUI map file mode, you have to explicitly save the information learnt by WinRunner. WinRunner saves it in a file with extension "gui". When you have to run a test, you need to load the corresponding GUI map file; otherwise it will not be able to recognize the objects in the test case and displays an error message

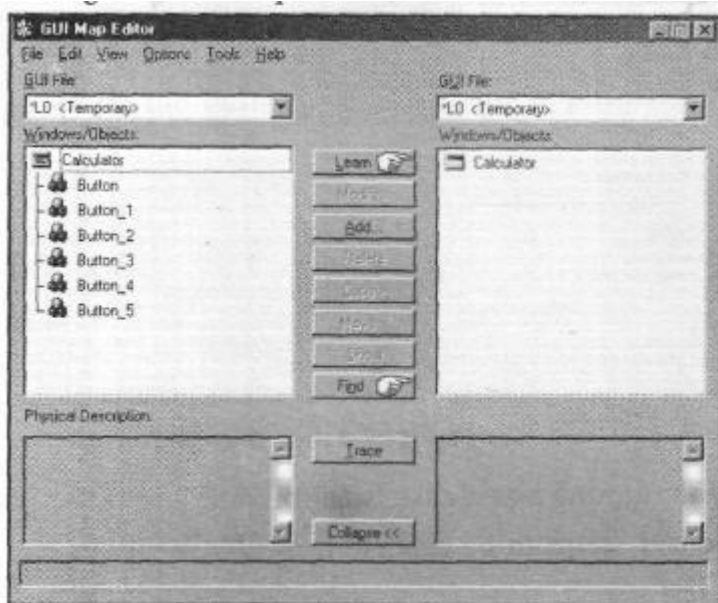
The procedure for saving the GUI map file in Global GUI map file mode is as follows:

Step 1: Record a test case by following the preceding procedure.

Step 2: Open the GUI Map Editor window as shown in Fig. Tools -> GUI Map Editor



Step 3: On selecting the GUI Map Editor. The screen as shown in figure is displayed



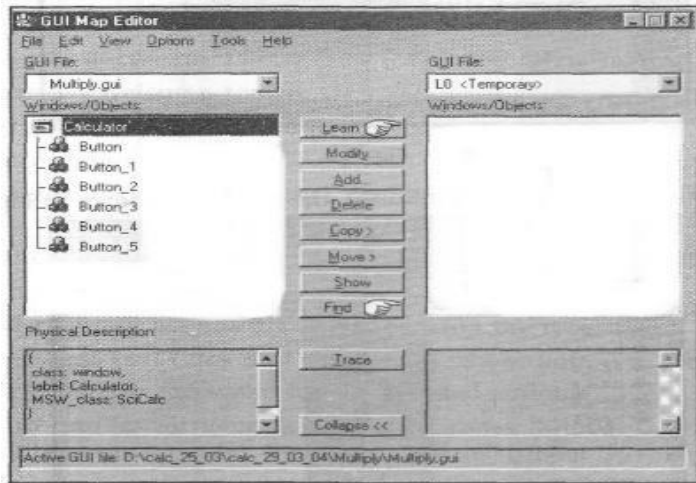
Step 4: Save the GUI Map file. File -> Save As A File dialog appears and you need to enter the filename.

Step 5: Close the GUI Map Editor window.

The procedure for loading the GUI map file is as follows:

Step 1: Open the GUI Map Editor. Tools -> GUI Map Editor

Step 2: Close all the opened GUI Map files File -> Close all.

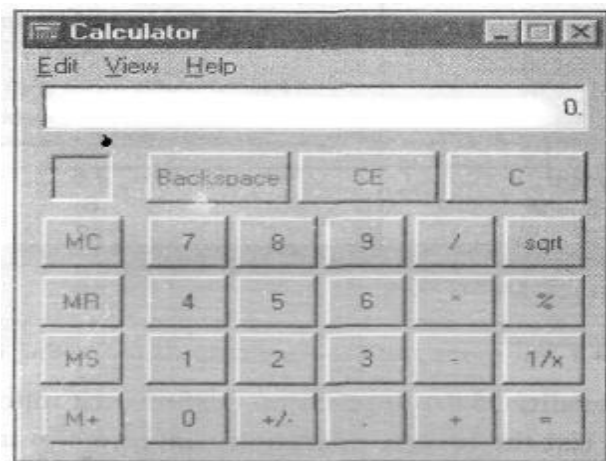


The procedure for running a test case is as follows:

Step 1: Open the test script to be executed.

Step 2: Run the test Run -> Run from top (or) press F5 to run the test.

WinRunner executes the generated script and displays the results in the Test Results window. We will now illustrate using WinRunner to test the "Standard Calculator" application available on your Windows system. You can invoke the calculator application from the desktop Start -> Programs -> Accessories -> Calculator. The GUI of the "Calculator" application is shown in Fig.



The symbols on the buttons of Calculator application represent the following functions:

+	:	To perform addition
-	:	To perform subtraction
*	:	To perform multiplication
/	:	To perform division
.	:	Decimal point
sqrt	:	To find square root of a number
%	:	To find percent
1/x	:	To find inverse of a number
MC	:	To clear the memory
MR	:	To recall from memory
MS	:	To save in the memory

M+	:	To add to the memory
C	:	To clear the current calculation
CE	:	To clear the displayed number
+/-	:	To give sign to a number (positive or negative)
Backspace:		To remove left most digit

To test the complete functionality of the application, we need to generate test cases in such a way that all the buttons are made use of. We need to generate some test cases which will give correct output and also some test cases which will give error messages. Table gives such test cases and the expected output for each test case.

Test Cases and the Expected Output for Testing the Calculator

Test Case	Expected Output
4 1/x	0.25
- 6 sqrt	Err: "Invalid input for function"
4 C	Clears the Display
1.2 * 3	3.6
5 / 2.0	2.5
7 + 8 - 9	6
600 * 2 %	12
2, MS, C, MR	2
MC, 2, M+, 3, M+, C, MR	5

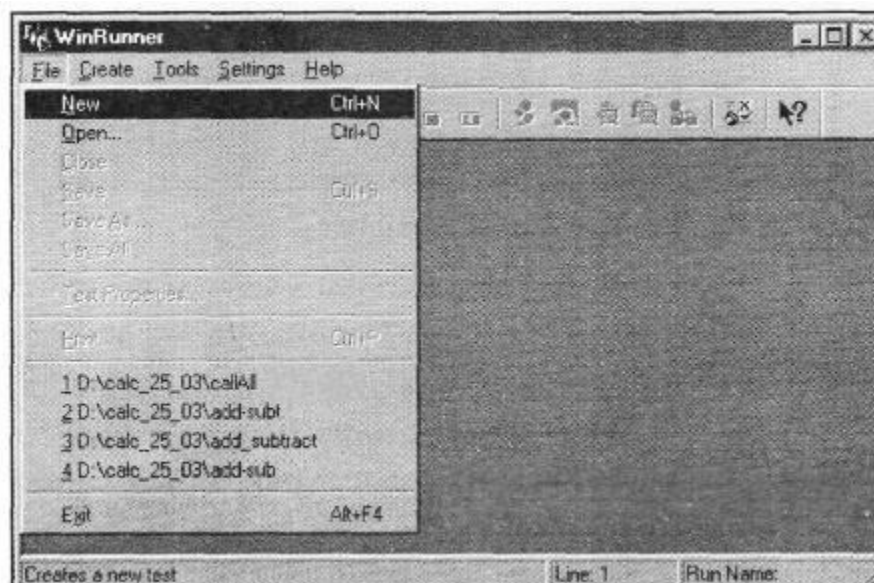
To test the functionality of the application perform the following steps:

Test Case #1: To test the Inverse operation (inverse of 4 using 1/x button)

Step 1: Open WinRunner application.

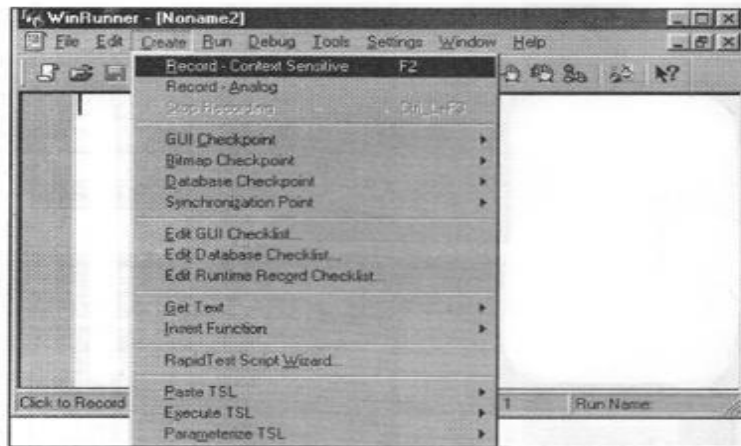
Step 2: Open Calculator application.

Step 3: Create a new document as shown in Figure. File -> New or Click Q (New) on tool bar or press Ctrl+N



Step 4: Start recording Create -> Record-Context Sensitive (or) press F2 (or) Click # on the toolbar

Click the (Record-Context Sensitive) button on the toolbar of WinRunner as shown in Figure or Select "Record - Context sensitive" option from the "Create" menu as shown in Figure.

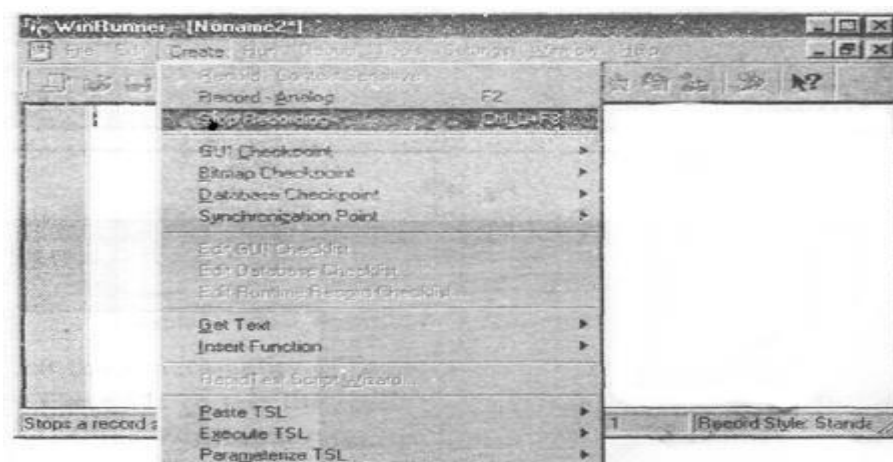


Step 5: Select the Calculator application and start recording the actions. a Click "4" on the Calculator

Click the "1/x" button on the Calculator to find the inverse of 4. The result, 0.25 will be displayed on the Calculator.

Step 6: Stop the Recording process.

Create -> Stop Recording (or) Click (Stop) on toolbar Click (Stop Recording) button on the toolbar of WinRunner as shown in Figure or Select the "Stop Recording" option from the "Create" menu as shown in Figure.

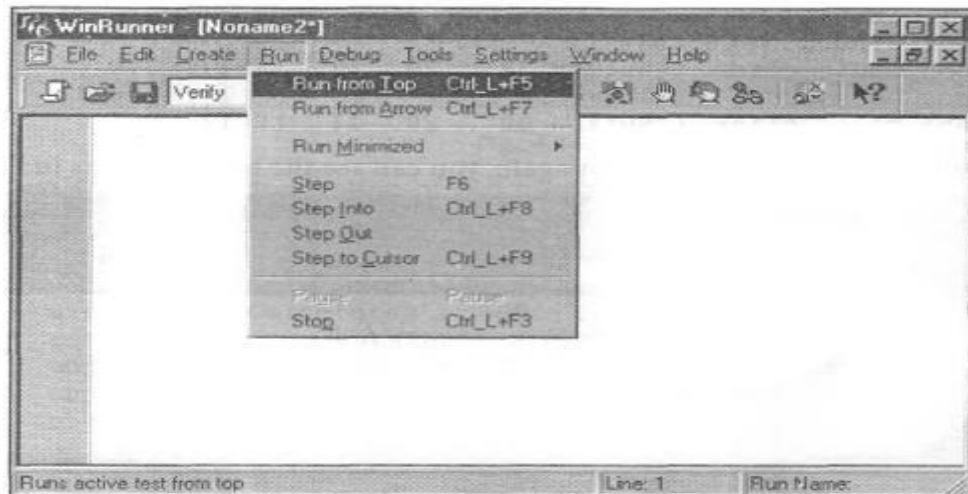


Step 8: Save the file as "inverse" in the selected folder.

File -> Save In the "Save" dialog box that appears, save the test script with name "inverse".

Step 9: Run the test script generated by WinRunner. Run -> Run from Top or press F5 or Click (Run

from Top) on the toolbar Click the (Run from Top) button on the toolbar of WinRunner as shown in Figure or select the "Run from Top" option from the "Run" menu as shown in Figure.



Step 10: After executing the TSL statements, WinRunner generates test results as shown in Figure. The Results column indicates whether the test has "Passed" or "Failed". The test results also give useful information such as the name of the test case, the line numbers in the test script and the time taken for executing the test case.

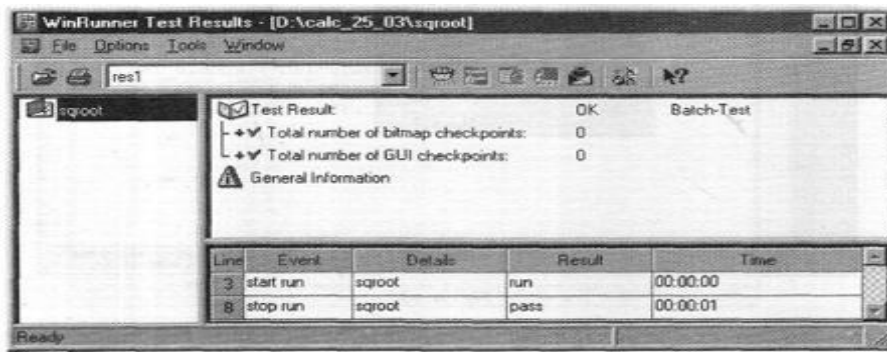
 A screenshot of the WinRunner test results window. The title bar reads 'D:\scale_25_03\inverse'. The window is divided into two main sections. The top section shows 'Test Result: OK' and 'Batch-Test'. Below this, it says 'Total number of bitmap checkpoints: 0' and 'Total number of GUI checkpoints: 0'. The bottom section is a table with the following data:

Line	Event	Details	Result	Time
3	start run	inverse	run	00:00:00
7	stop run	inverse	pass	00:00:01

You can use the same procedure explained above for recording the test case. The following test script will be generated:

```
#Calculator
winactivate("Calculator");
set_window("Calculator",1);
obj_mouse_click ("Button_38",20,12,LEFT);
objmousedrag ("Button_35",10,15,11,14,LEFT);
obj_mouse_click ("Button_60",20,11,LEFT);
```

When you run the test script again, you can see the test results, as in Figure.



Calling the Test Cases using "call" Function

The "call" function can be used to execute a series of test cases without any user interaction. The syntax of call function is: call for example, call test1();

Create a new document Write the following test script Save the file as "callAll" Execute the test case All the preceding test cases can be combined into one file as follows:

```
call inverseO; call sqrootO;
call clearO;
call MultiplayO;
call divideO;
call add_subtract()
call PercentO;
call msjnrO;
call maddmrO;
call mclearO;
call backspaceO;
```



When you execute this test script, all the earlier test cases are executed in one shot. The test results screen will be as shown in Figure. As you can see from the table, the "Details" column gives the various test cases executed. The "Result" column shows whether the test has passed or failed. The "Time" column gives the time taken to execute the test case.

When you have to retest the application using the same test cases, you can run the script in unattended mode. You can save the script in a file and run the script at specified time.

This feature of WinRunner is extremely useful for regression testing. When you are developing the software, you need to run the same set of test cases many times. So, you can run the application once,

generate the test script and then keep doing the regression testing. Obviously, the productivity of the test engineers will be very high when this tool is used.

Viva questions

1. Explain WinRunner testing process?
2. What is contained in the GUI map?
3. How does WinRunner evaluate test results?
4. What is the use of GUI map?
5. What happens when GUI map file get loaded?

EXPERIMENT: 7

NAME OF THE EXPERIMENT: Study of any web testing tool (e.g. Selenium)

What is Selenium?

JavaScript framework that runs in your web browser Works anywhere JavaScript is supported Hooks for many other languages Java, Ruby, Python Can simulate a user navigating through pages and then assert for specific marks on the pages All you need to really know is HTML to start using it right away

Selenium IDE

Selenium **I**ntegrated **D**evelopment **E**nvironment (IDE) is a Firefox plugin that lets testers to record their actions as they follow the workflow that they need to test.

It provides a Graphical User Interface for recording user actions using Firefox which is used to learn and use Selenium, but it can only be used with Firefox browser as other browsers are not supported.

However, the recorded scripts can be converted into various programming languages supported by Selenium and the scripts can be executed on other browsers as well.

Selenium- IDEDownload

Step 1 – Launch Firefox and navigate to the following URL - <http://seleniumhq.org/download/>.

Under the Selenium IDE section, click on the link that shows the current version number as shown below.



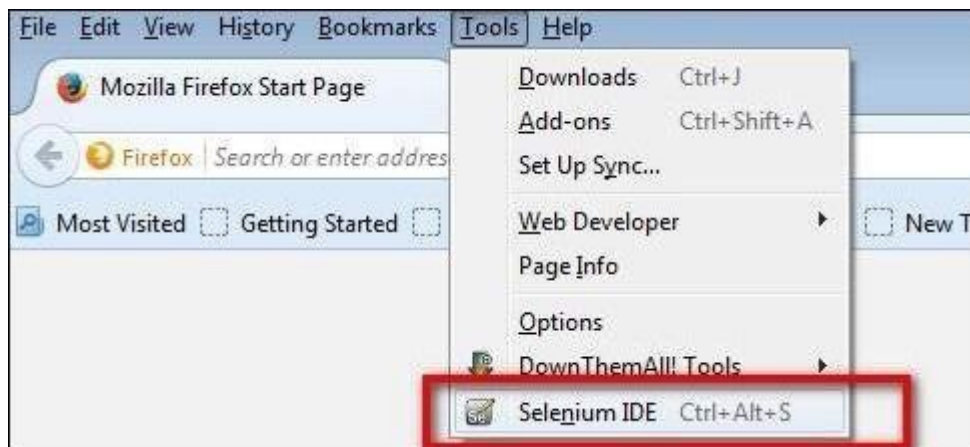
Step 2 – Firefox add-ons notifier pops up with allow and disallow options. User has to allow the installation.



Step 3 – The add-ons installer warns the user about untrusted add-ons. Click 'Install Now'.



Step 4 – The Selenium IDE can now be accessed by navigating to Tools >>Selenium IDE.



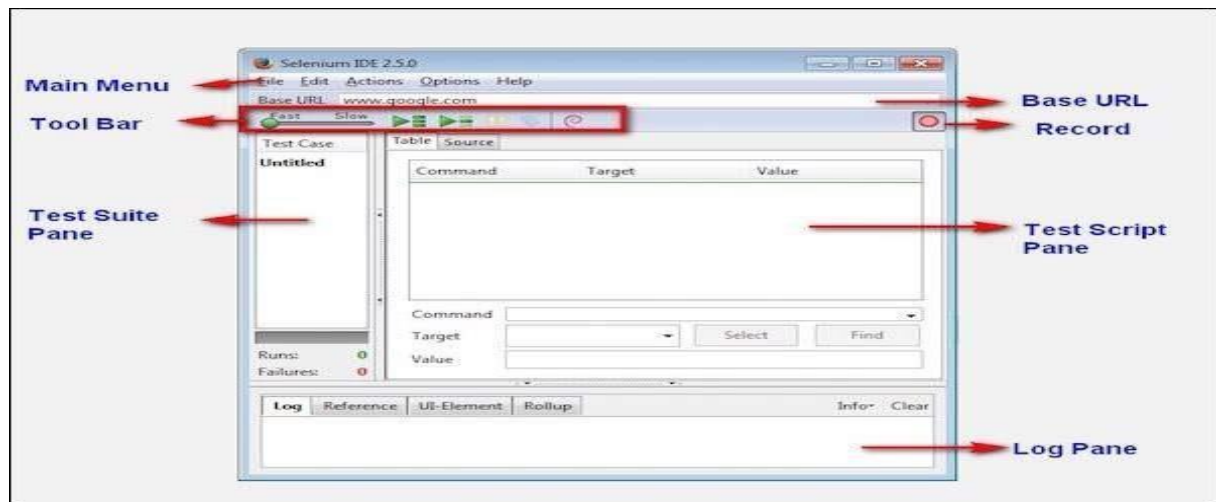
Step 5 – The Selenium IDE can also be accessed directly from the quick access menu bar as shown below.



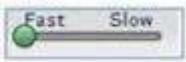







Selenium IDE Features

This section deals with the features available in Selenium IDE.

The following image shows the features of Selenium IDE with the help of a simple tool-tip.



The features of the record tool bar are explained below.

Control	Control Name	Description
	Speed Control	This helps in controlling the speed of the test case runs.
	Run All	Executes the entire test suite that contains multiple test cases.
	Run	Executes the currently selected test.
	Pause/Resume	Allows user to pause or resume the script execution. Enabled only during the execution.
	Step	Helps user to debug the test by executing only one step of a test case at a time.
	Test Runner Mode	Allows user to execute the test case in a browser loaded with the selenium Core. It is an obsolete functionality that likely to be deprecated.
	Apply Rollup Rules	This features allows repetitive sequences of selenium commands to be grouped into a single action.
	Record	This features helps user to Records the user's browser actions.

Creating Selenium IDE Tests

This section deals with how to create IDE tests using recording feature.

The following steps are involved in creating Selenium tests using IDE –

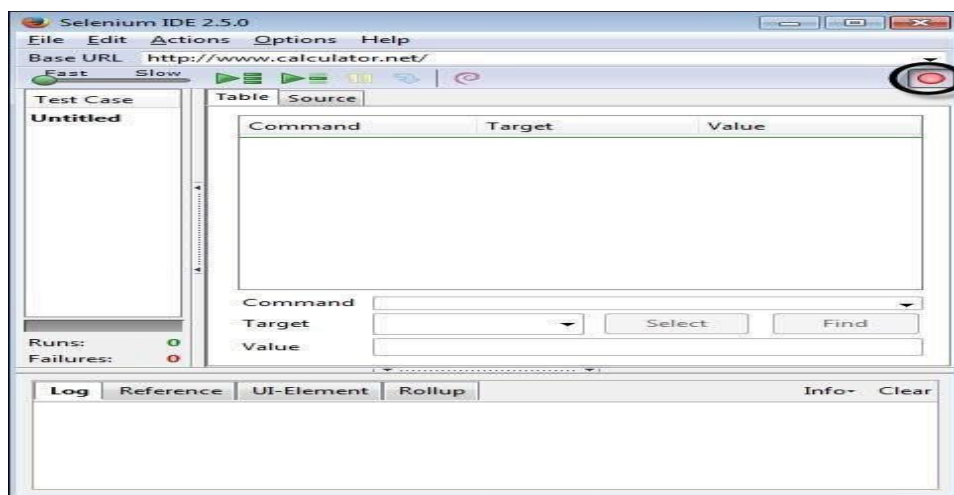
- Recording and adding commands in a test
- Saving the recorded test
- Saving the test suite
- Executing the recorded test

Recording and Adding Commands in a Test

We will use www.ncalculators.com to demonstrate the features of Selenium.

Step 1 – Launch the Firefox browser and navigate to the website - <https://www.ncalculators.com/>

Step 2 – Open Selenium IDE from the Tools menu and press the record button that is on the top-right corner.



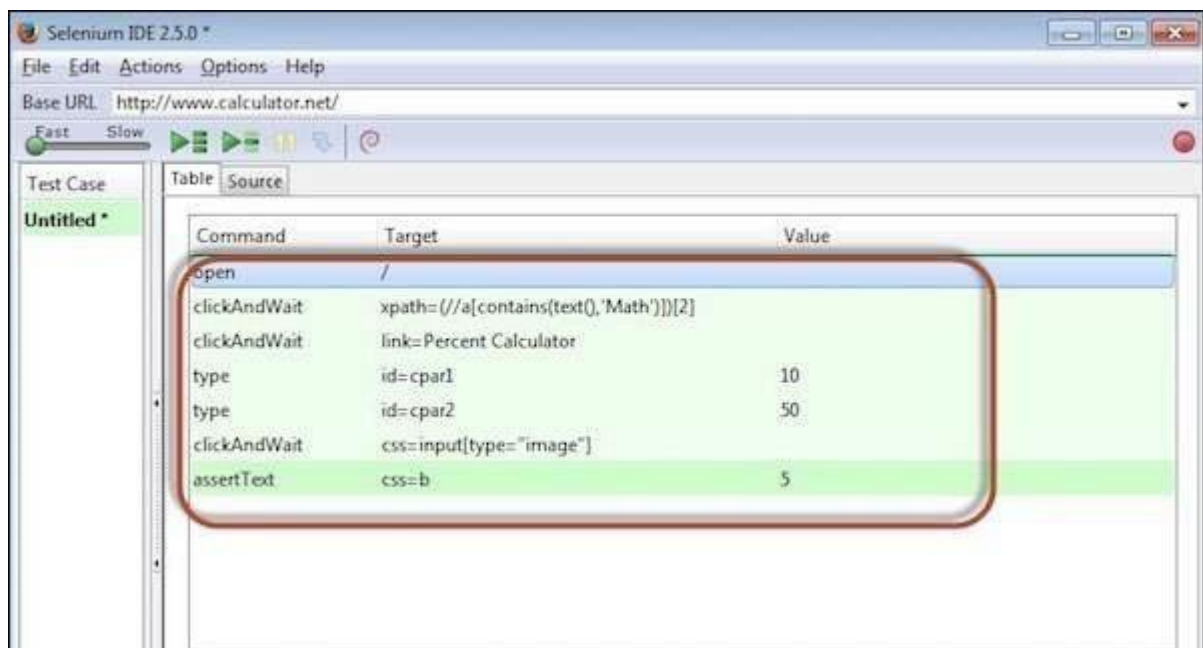
Step 3 – Navigate to "Math Calculator" >> "Percent Calculator" >> enter "10" as number1 and 50 as number2 and click "calculate".



Step 4 – User can then insert a checkpoint by right clicking on the webelement and select "Show all available commands" >> select "assert text css=b 5"



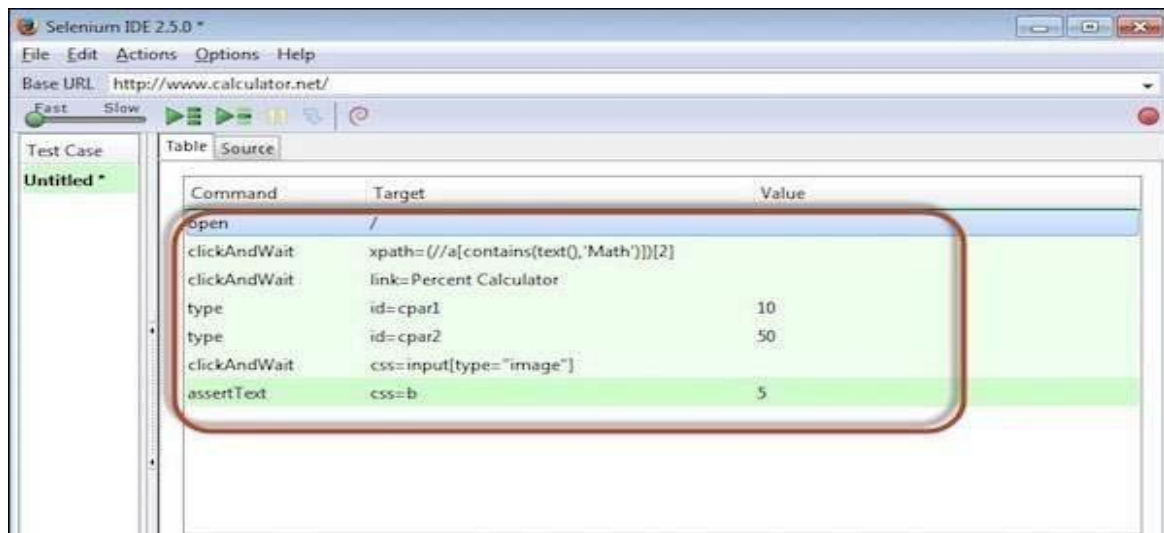
Step 5 – The recorded script is generated and the script is displayed as shown below.



Saving the Recorded Test

Step 1 – Save the Test Case by navigating to "File" >> "Save Test" and save the file in the location of your choice. The file is saved as .HTML as default.

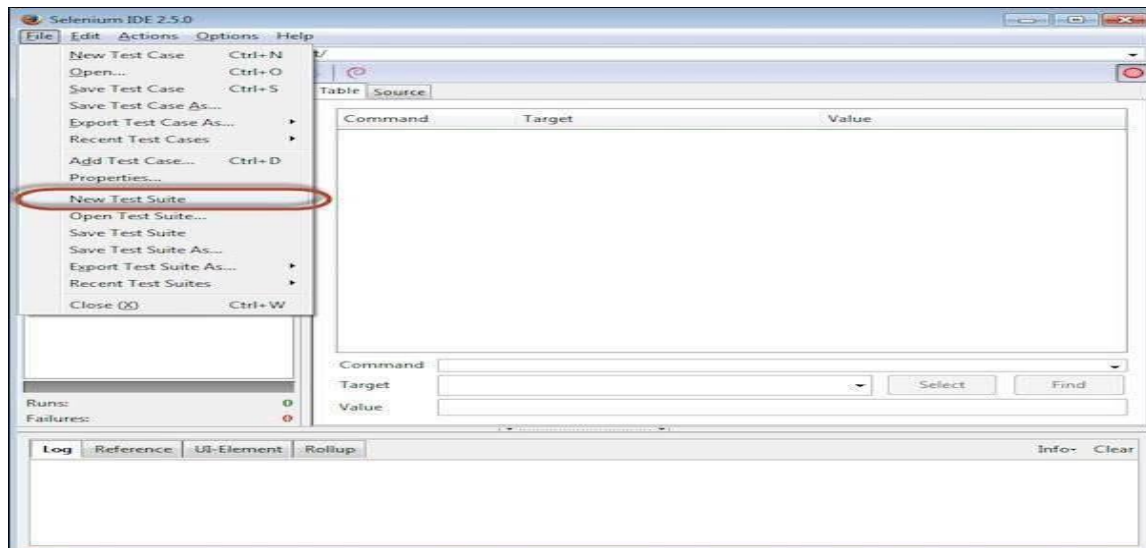
The test can also be saved with an extension htm, shtml, and xhtml.



Saving the Test Suite

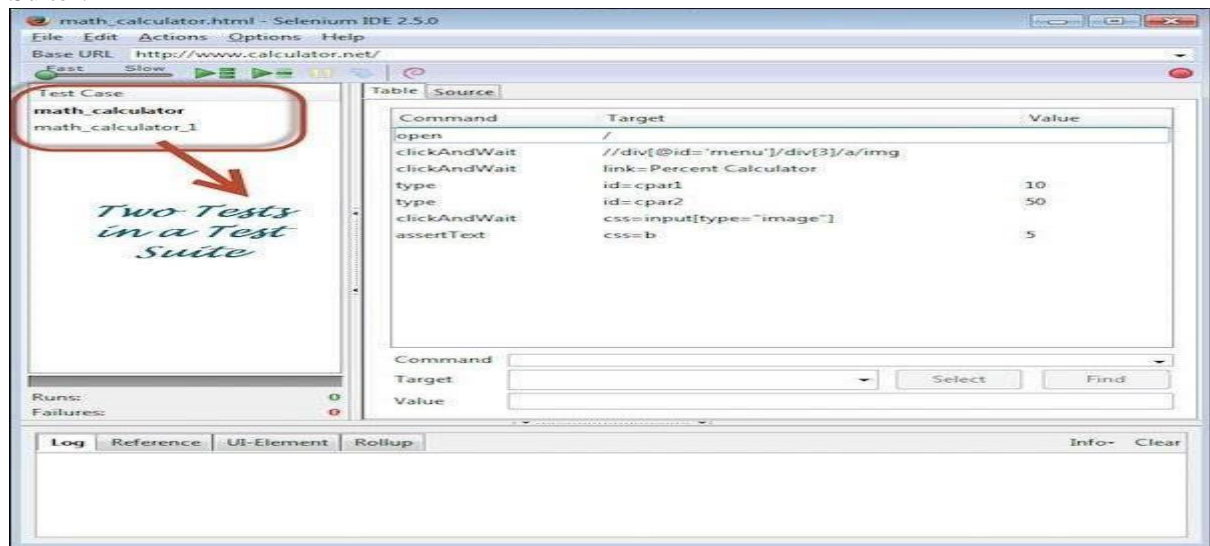
A test suite is a collection of tests that can be executed as a single entity.

Step 1 – Create a test suite by navigating to "File" >> "New Test Suite" as shown below.



Step 2 – The tests can be recorded one by one by choosing the option "New Test Case" from the "File" Menu.

Step 3 – The individual tests are saved with a name along with saving a "Test Suite".



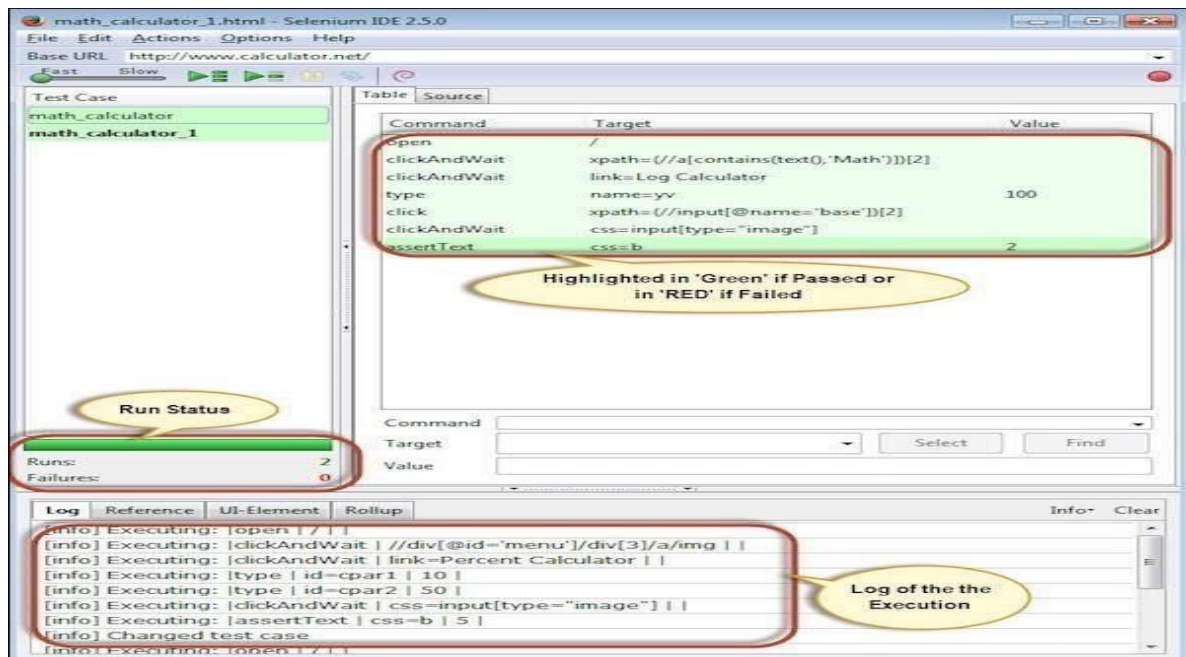
Executing the Recorded Test

The recorded scripts can then be executed either by clicking "Play entire suite" or "Play current test" button in the toolbar.

Step 1 – The Run status can be seen in the status pane that displays the number of tests passed and failed.

Step 2 – Once a step is executed, the user can see the result in the "Log" Pane.

Step 3 – After executing each step, the background of the test step turns "Green" if passed and "Red" if failed as shown below.



Selenium IDE Script Debugging

This section deals with debugging the Selenium IDE script.

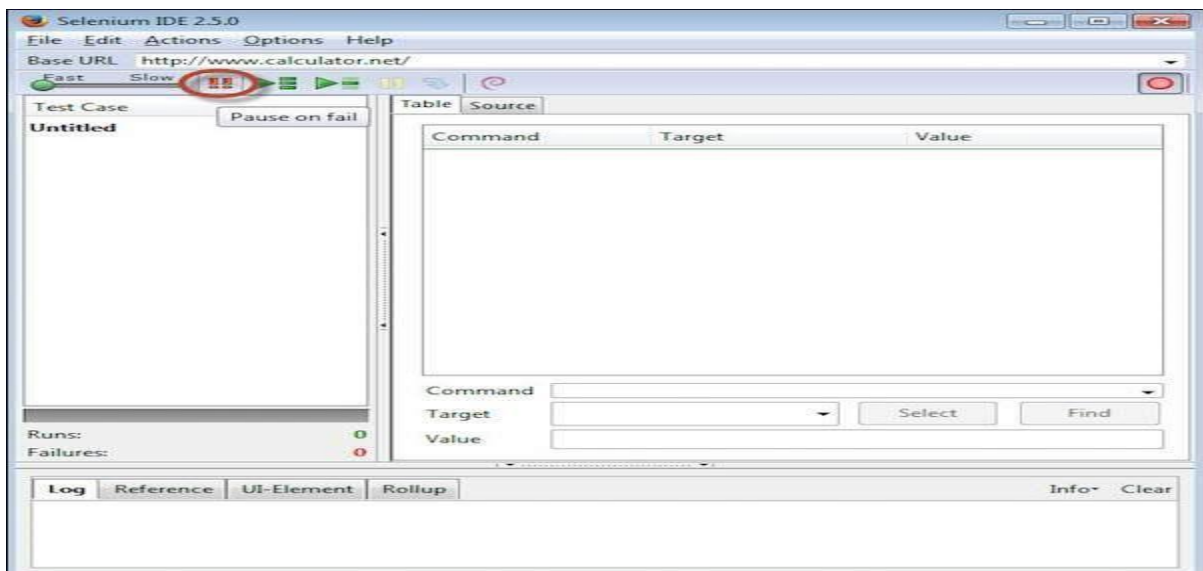
Debugging is the process of finding and fixing errors in the test script. It is a common step in any script development. To make the process more robust, we can make use a plugin "Power Debugger"

for Selenium IDE.

Step 1 – To install Power Debugger for Selenium IDE, navigate to <https://addons.mozilla.org/en-US/firefox/addon/power-debugger-selenium-ide/> and click "Add to Firefox" as shown below.



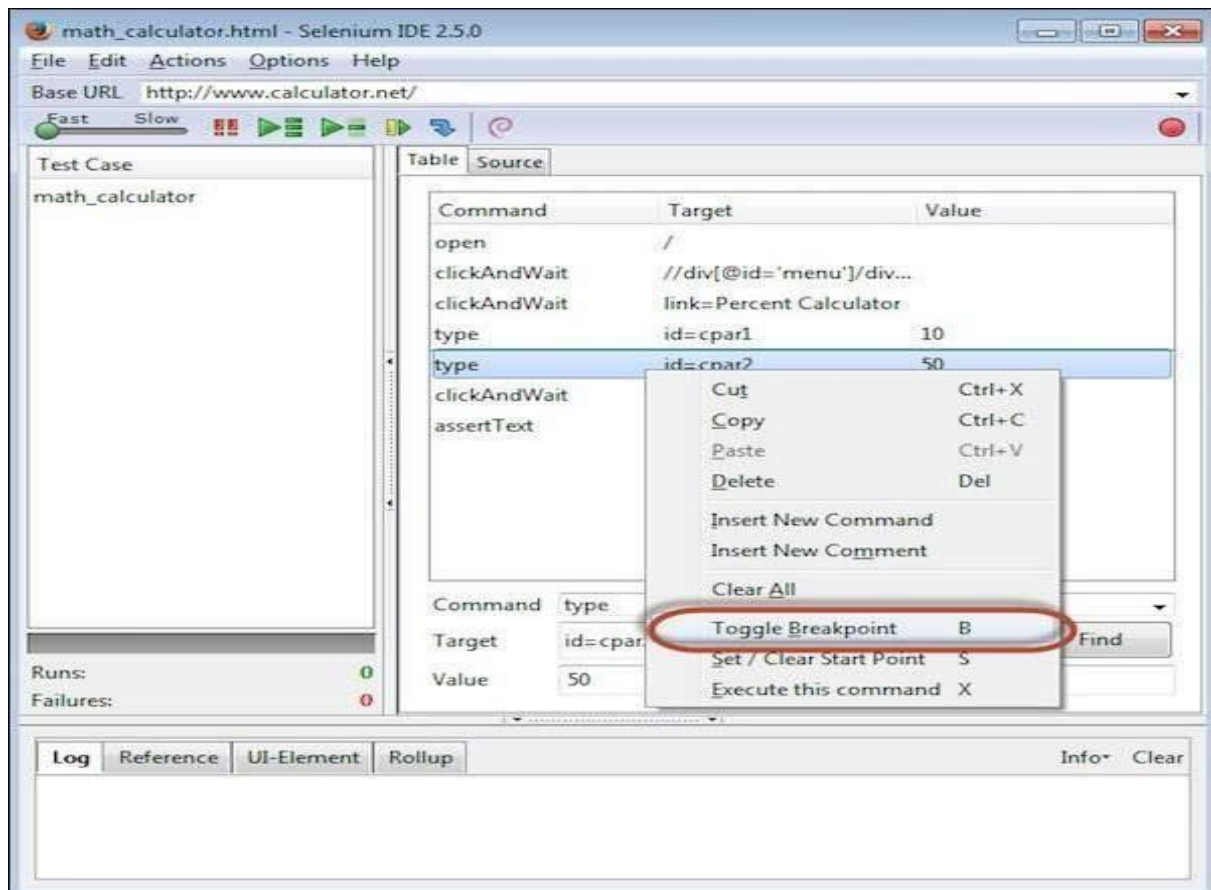
Step 2 – Now launch 'Selenium IDE' and you will notice a new icon, "Pause on Fail" on recording toolbar as shown below. Click it to turn it ON. Upon clicking again, it would be turned "OFF".



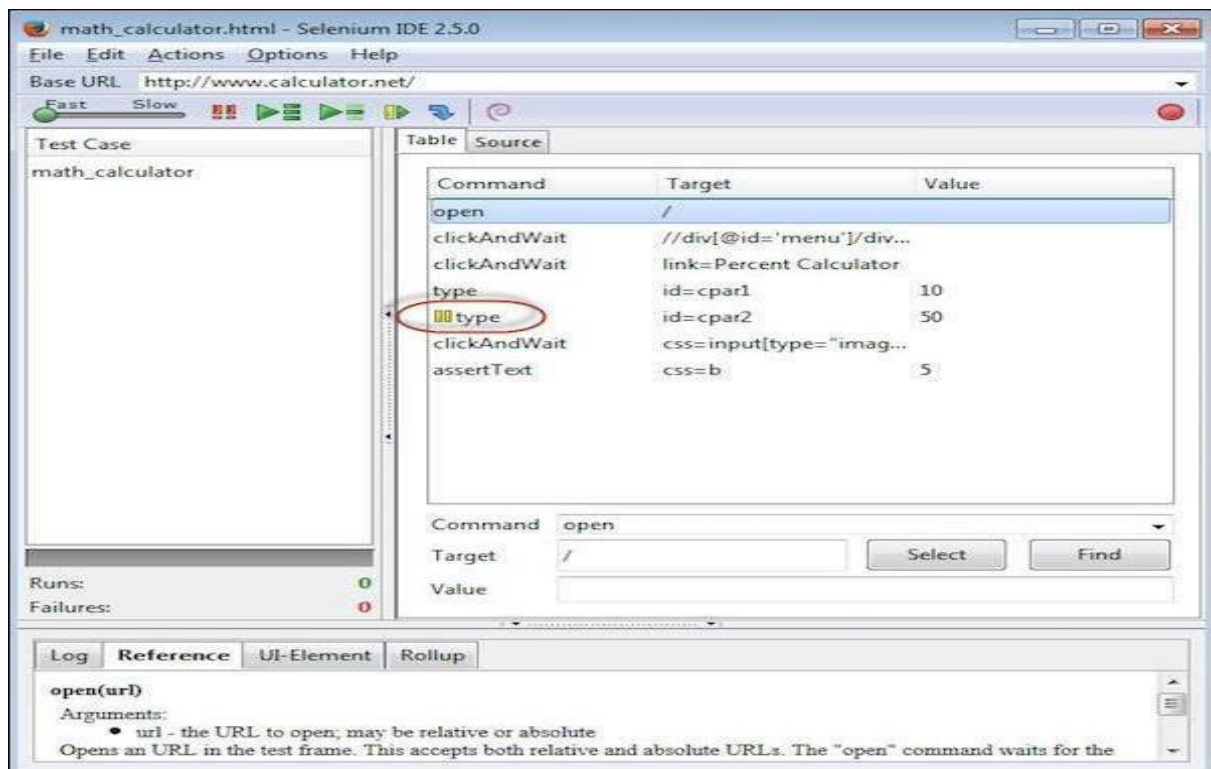
Step 3 – Users can turn "pause on fail" on or off any time even when the test is running.

Step 4 – Once the test case pauses due to a failed step, you can use the resume/step buttons to continue the test execution. The execution will **NOT** be paused if the failure is on the last command of any test case.

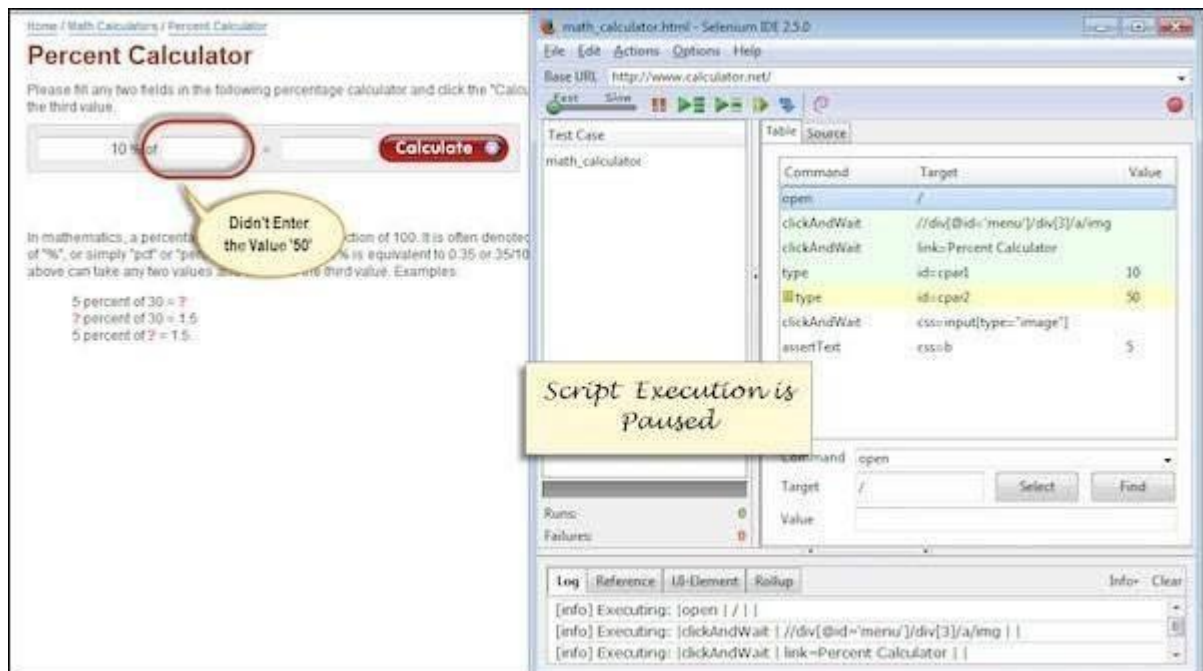
Step 5 – We can also use breakpoints to understand what exactly happens during the step. To insert a breakpoint on a particular step, "Right Click" and select "Toggle Breakpoint" from the context-sensitive menu.



Step 6 – Upon inserting the breakpoint, the particular step is displayed with a pause icon as shown below.



Step 7 – When we execute the script, the script execution is paused where the breakpoint is inserted. This will help the user to evaluate the value/presence of an element when the execution is in progress.

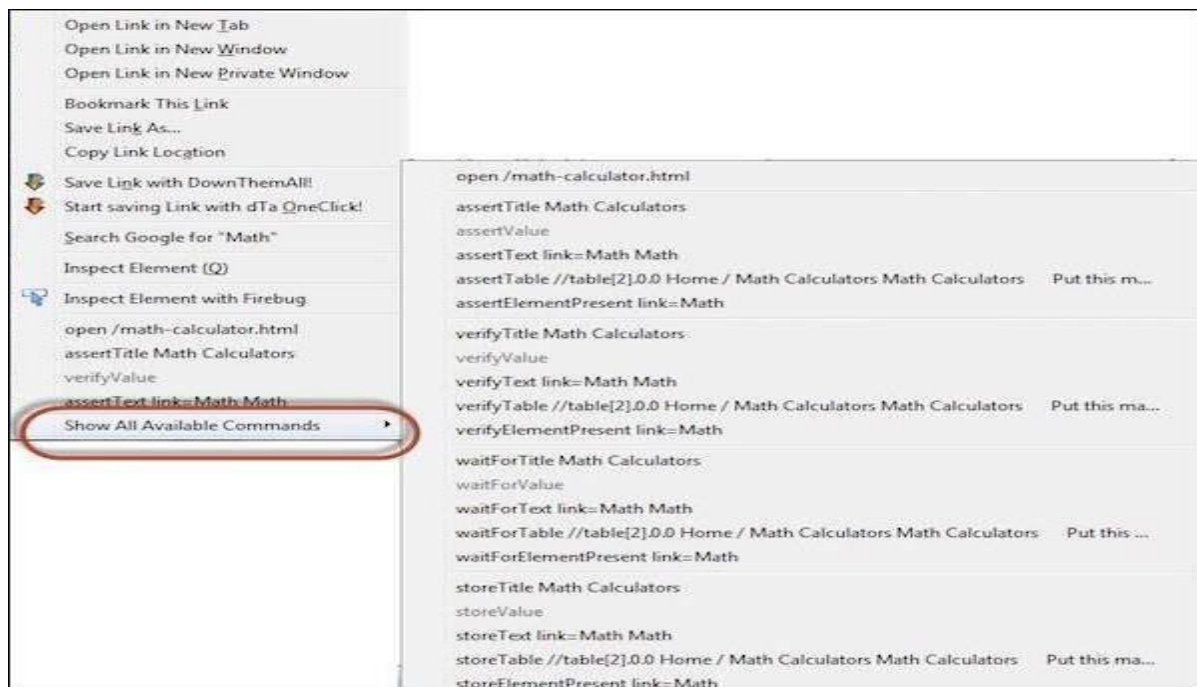


Inserting Verification Points

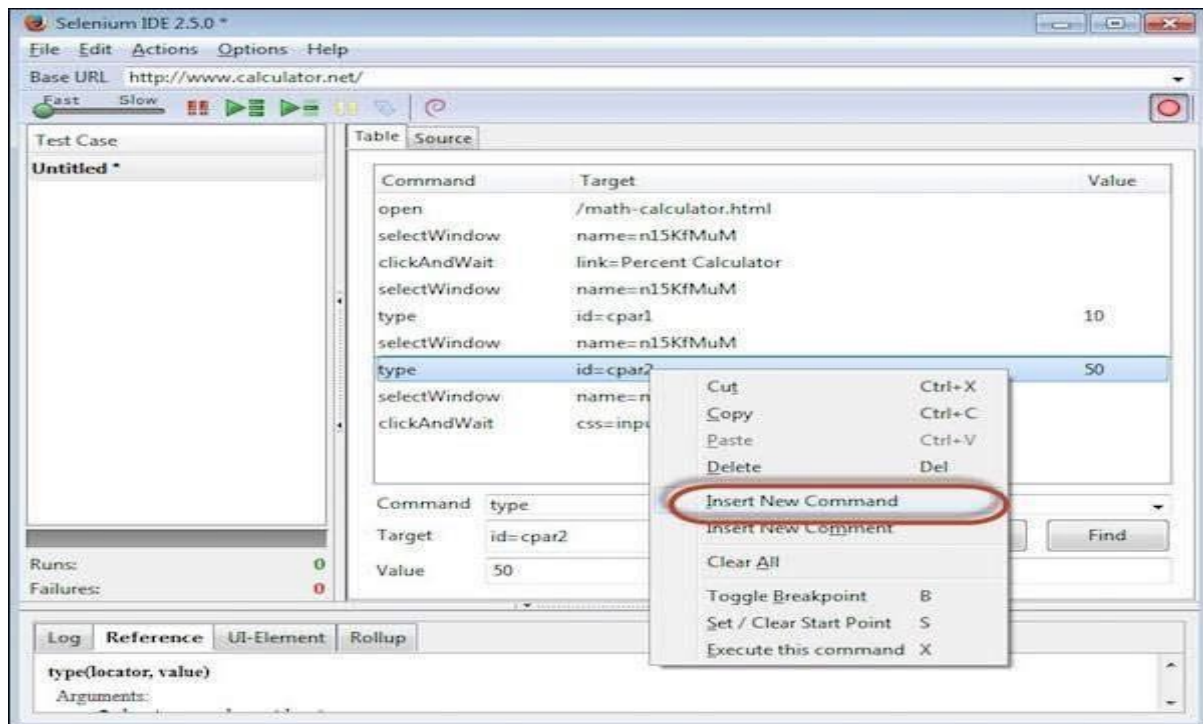
This section describes how to insert verification points in Selenium IDE.

The test cases that we develop also need to check the properties of a web page. It requires assert and verify commands. There are two ways to insert verification points into the script.

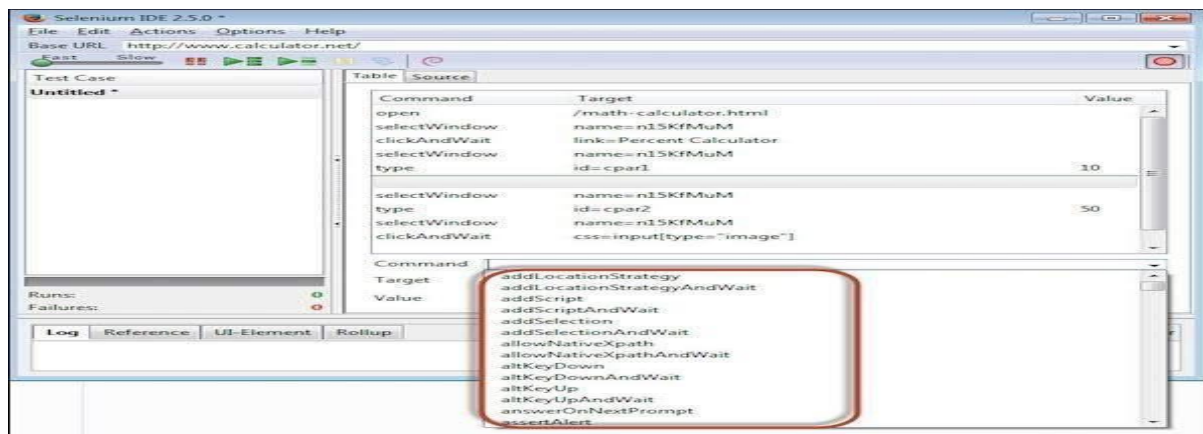
To insert a verification point in recording mode, "Right click" on the element and choose "Show all Available Commands" as shown below.



We can also insert a command by performing a "Right-Click" and choosing "Insert New Command".



After inserting a new command, click 'Command' dropdown and select appropriate verification point from the available list of commands as shown below.



Given below are the mostly used verification commands that help us check if a particular step has passed or failed.

- verifyElementPresent
- assertElementPresent
- verifyElementNotPresent
- assertElementNotPresent
- verifyText
- assertText
- verifyAttribute
- assertAttribute
- verifyChecked
- assertChecked

- `verifyAlert`
- `assertAlert`
- `verifyTitle`
- `assertTitle`

Synchronization Points

During script execution, the application might respond based on server load, hence it is required for the application and script to be in sync. Given below are few a commands that we can use to ensure that the script and application are in sync.

- `waitForAlertNotPresent`
- `waitForAlertPresent`
- `waitForElementPresent`
- `waitForElementNotPresent`
- `waitForTextPresent`
- `waitForTextNotPresent`
- `waitForPageToLoad`
- `waitForFrameToLoad`

Selenium Pattern Matching

- This section deals with how to work with regular expressions using IDE.

Like locators, patterns are a type of parameter frequently used by Selenium. It allows users to describe patterns with the help of special characters. Many a time, the text that we would like to verify are dynamic; in that case, pattern matching is very useful.

Pattern matching is used with all the verification point commands - `verifyTextPresent`, `verifyTitle`, `verifyAlert`, `assertConfirmation`, `verifyText`, and `verifyPrompt`.

There are three ways to define a pattern –

- globbing
- regular expressions, and
- exact patterns.

Globbing

Most techies who have used file matching patterns in Linux or Windows while searching for a certain file type like `*.doc` or `*.jpg`. would be familiar with term "globbing"

Globbing in Selenium supports only three special characters: `*`, `?`, and `[]`.

- `*` – matches any number of characters.
- `?` – matches a single character.

- `[]` – called a character class, lets you match any single character found within the brackets.
`[0-9]` matches any digit.

To specify a glob in a Selenium command, prefix the pattern with the keyword 'glob:'. For example, if you would like to search for the texts "tax year 2013" or "tax year 2014", then you can use the glob "tax year *" as shown below.

However the usage of "glob:" is optional while specifying a text pattern because globbing patterns are the default in Selenium.

Command	Target	Value
clickAndWait	link = search	
verifyTextPresent	glob: tax year *	

Exact Patterns

Patterns with the prefix 'exact:' will match the given text as it is. Let us say, the user wants an exact match with the value string, i.e., without the glob operator doing its work, one can use the 'exact' pattern as shown below. In this example the operator '*' will work as a normal character rather than a pattern-matching wildcard character.

Command	Target	Value
clickAndWait	link = search	
verifyValue	exact: *.doc	

Regular Expressions

Regular expressions are the most useful among the pattern matching techniques available. Selenium supports the complete set of regular expression patterns that Javascript supports. Hence the users are no longer limited by *, ? and [] globbing patterns.

To use RegEx patterns, we need to prefix with either "regexp:" or "regexpi:". The prefix "regexpi" is case-insensitive. The glob: and the exact: patterns are the subsets of the Regular Expression patterns. Everything that is done with glob: or exact: can be accomplished with the help of RegExp.

Example

For example, the following will test if an input field with the id 'name' contains the string 'tax year', 'Tax Year', or 'tax Year'.

Command	Target	Value
clickAndWait	link = search	
verifyValue	id = name	regexp:[Tt]ax ([Yy]ear)

Selenium User Extensions

The Java script that allows users to customize or add new functionality.

It is easy to extend Selenium IDE by adding customized actions, assertions, and locator-strategies. It is done with the help of JavaScript by adding methods to the Selenium object prototype. On startup, Selenium will automatically look through the methods on these prototypes, using name patterns to recognize which ones are actions, assertions, and locators.

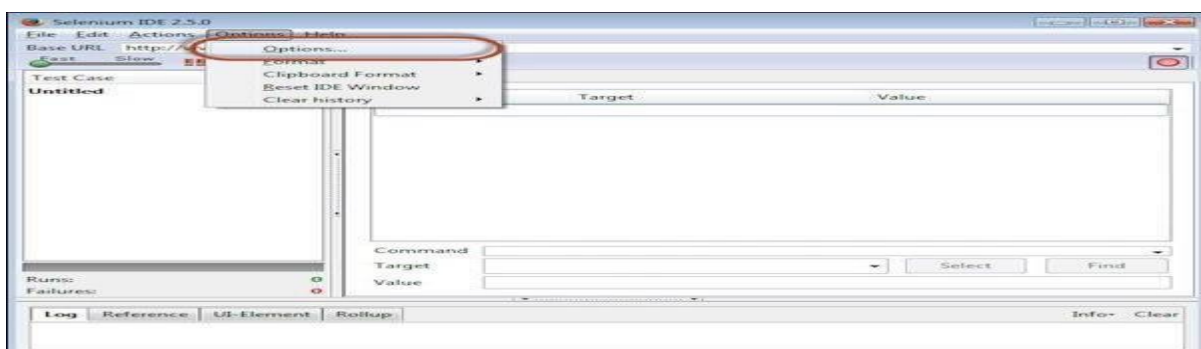
Let us add a 'while' Loop in Selenium IDE with the help of JavaScript.

Step 1 – To add the js file, first navigate

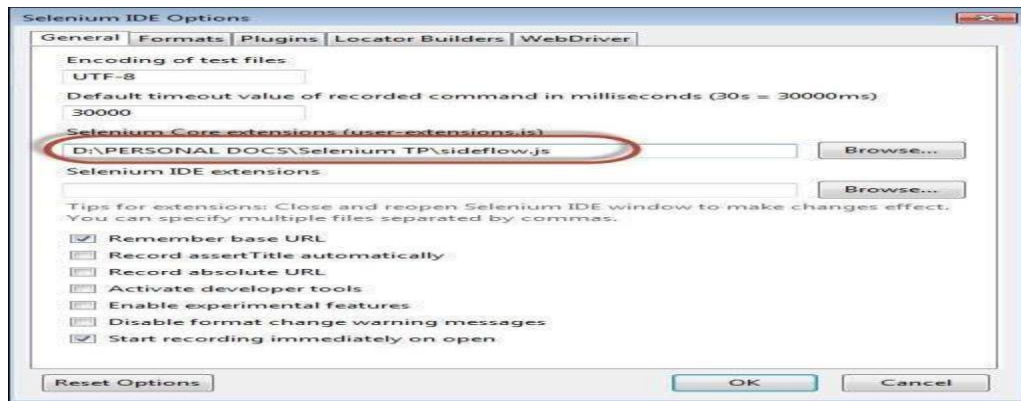
to <https://github.com/darrenderidder/sideflow/blob/master/sideflow.js> and copy the script and place save it as 'sideflow.js' in your local folder as shown below.



Step 2 – Now launch 'Selenium IDE' and navigate to "Options" >> "Options" as shown below.



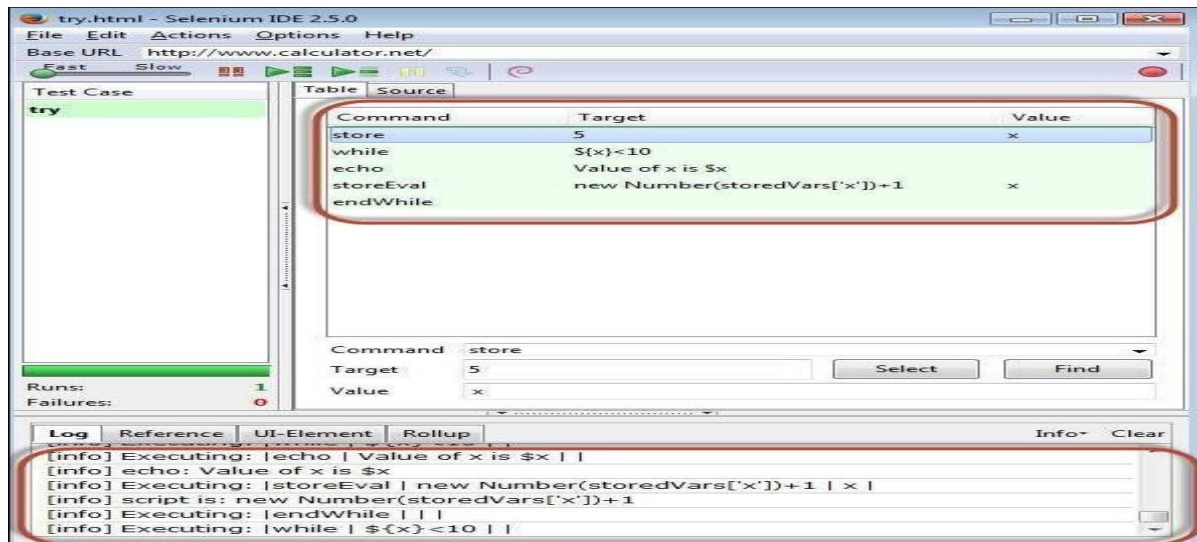
Step 3 – Click the 'Browse' button under 'Selenium Core Extensions' area and point to the js file that we have saved in Step 1.



Step 4 – Restart Selenium IDE.

Step 5 – Now you will have access to a few more commands such as "Label", "While" etc.

Step 6 – Now we will be able to create a While loop within Selenium IDE and it will execute as shown below.



Viva questions:

1. What is Selenium?
2. What are the different Selenium components?
3. What are the testing types that can be supported by Selenium?
4. What are the limitations of Selenium?
5. What is Selenese?

EXPERIMENT: 8

NAME OF THE EXPERIMENT: Study of Any Bug Tracking Tool (Bugzilla)

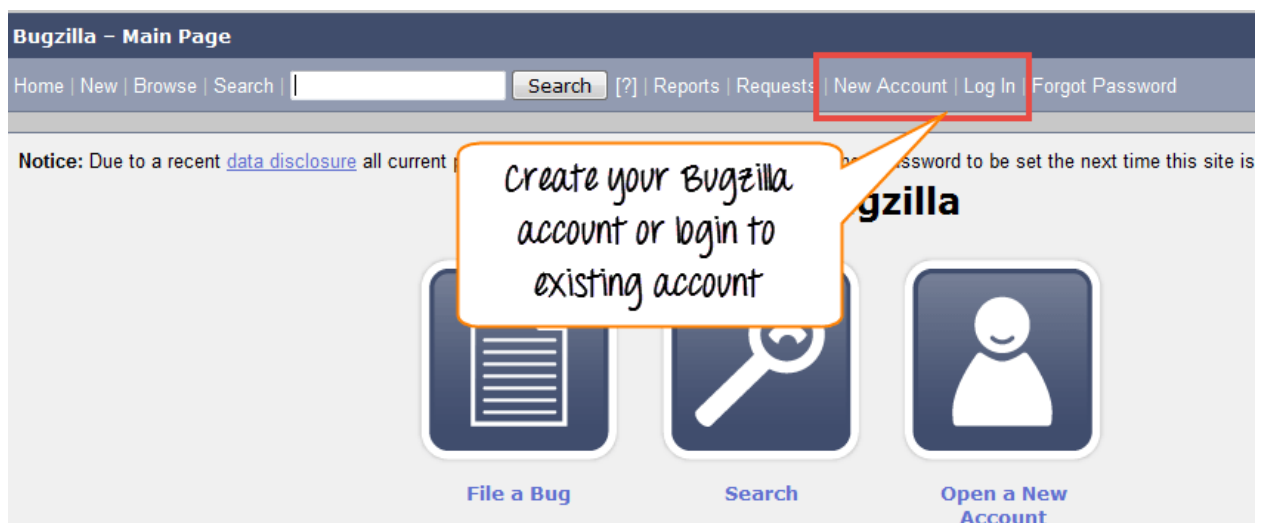
Introduction

Bugzilla is a popular bug-tracking system that allows teammates and developers to track outstanding bug, Assign, track, and resolve error fixes and conflict resolutions using the Bugzilla database Communicate with teammates, Submit and review patches and Manage quality assurance (QA).

Bugzilla is one example of a class of programs called "Defect Tracking Systems", or, more commonly, "Bug-Tracking Systems". Defect Tracking Systems allow individual or groups of developers to keep track of outstanding bugs in their product effectively.

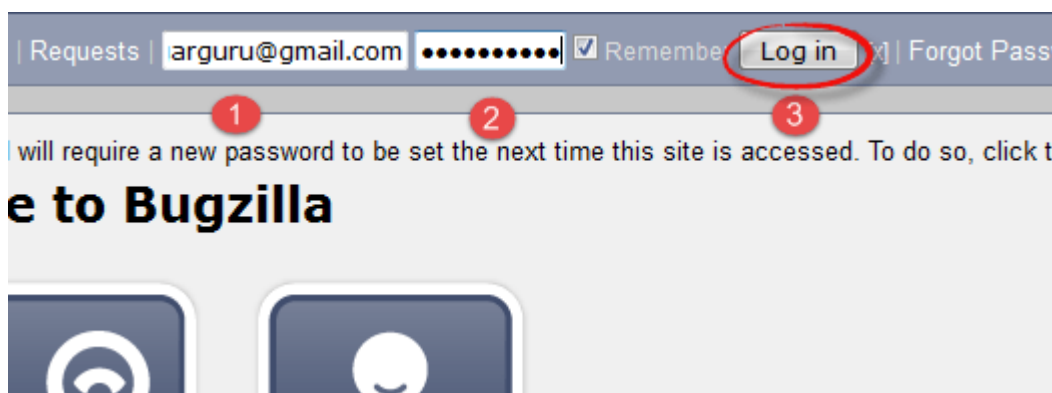
How to log-in to Bugzilla

Step 1) Use the following [link](#) for your handons. To create an account in Bugzilla or to login into the existing account go to **New Account or Log in** option in the main menu.



Step 2) Now, enter your personal details to log into Bugzilla

1. User ID
2. Password
3. And then click on "Log in"

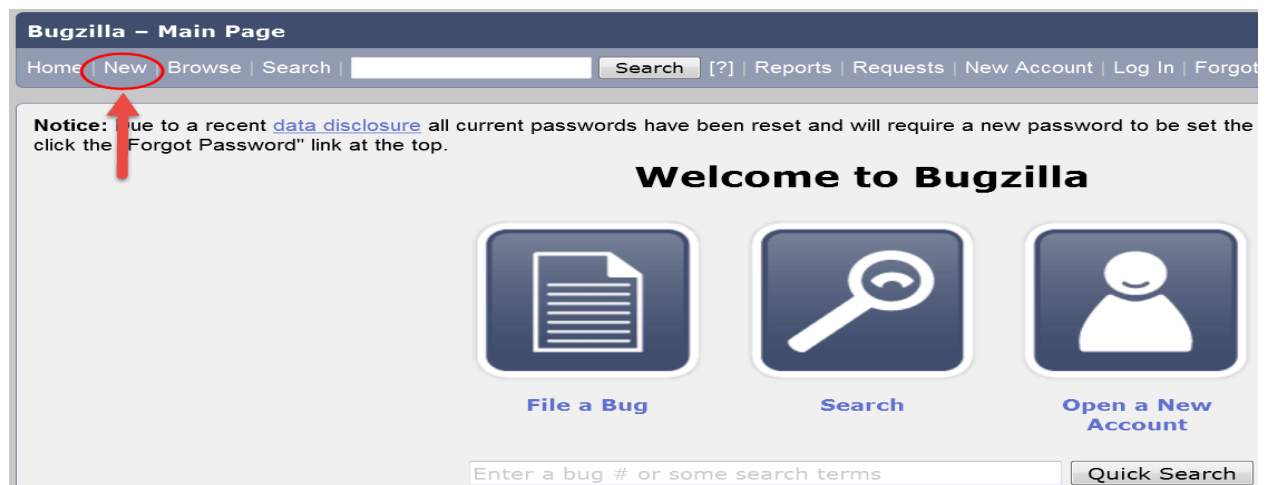


Step 3) You are successfully logged into Bugzilla system



Creating a Bug-report in Bugzilla

Step 1) To create a new bug in Bugzilla, visit the home-page of Bugzilla and click on **NEW** tab from the main menu



Step 2) In the next window

1. Enter Product
2. Enter Component
3. Give Component description
4. Select version,
5. Select severity
6. Select Hardware
7. Select OS
8. Enter Summary
9. Enter Description
10. Attach Attachment
11. Submit

NOTE: The above fields will vary as per your customization of Bugzilla

Home | New | Browse | Search | Search [?] | Reports | My Requests | Preferences | Help | Log

Notice: Due to a recent [data disclosure](#) all current passwords have been reset and will require a new password to be set. Before reporting a bug, please read the [bug writing guidelines](#), please look at the list of [most frequently reported bugs](#).

[Show Advanced Fields](#) (* = Required Field)

1 * **Product:** Sam's Widget **Reporter:** jeegarguru@gmail.com

2 **Component:** Widget Gears **Component Description:** Gears for Sam's widgets

4 **Version:** unspecified 5 **Severity:** normal

6 **Hardware:** PC 7 **OS:** Windows NT

We've made a guess at your operating system and platform. check them and make any corrections if necessary.

8 * **Summary:**

9 **Description:**

10 **Attachment:** Add an attachment

11 **Submit Bug**

NOTE: The mandatory fields are marked with *.

In our case field's

- Summary
- Description

Are mandatory

If you do not fill them you will get a screen like below

1 * **Summary:** Gears for sams widget twisted

You must enter a Summary for this bug.

Possible Duplicates:	Bug ID	Summary	Status	
	7776	when using the Widget Gears, the mV signal unexpectedly goes to 0	CONFIRMED	Add Me to the CC List
	7777	Widget Gears causes wrong mV signal to appear	CONFIRMED	Add Me to the CC List
	12431	Widget Gears cannot start	IN_PROGRESS	Add Me to the CC List
	12480	The Gear of sams widgets failed its validation	CONFIRMED	Add Me to the CC List
	15407	Sams Widget came pipe	CONFIRMED	Add Me to the CC List
	21019	Gears are bound up	CONFIRMED	Add Me to the CC List
	21841	Widget gears are stuck	CONFIRMED	Add Me to the CC List

Put your description over here

2 **Description:** The widget gears are twisted at the end and not showing correct signal

Step 4) Bug is created ID# 26320 is assigned to our Bug. You can also add additional information to the assigned bug like URL, keywords, whiteboard, tags, etc. This extra-information is helpful to give more detail about the Bug you have created.

1. Large text box
2. URL
3. Whiteboard
4. Keywords
5. Tags
6. Depends on
7. Blocks
8. Attachments

First Last Prev Next This bug is not in your last search results.

Bug 26320 - Gears for sams widget twisted (edit)

Status: CONFIRMED (edit)

Product: Sam's Widget
 Component: Widget Gears
 Version: unspecified
 Hardware: PC Windows NT
 Importance: P2 normal
 et Milestone: ---
 Assigned To: sam folk-williams (edit) (take)
 QA Contact: (edit) (take)

Reported: 2015-01-07 02:50 PST by James
 Modified: 2015-01-07 03:10 PST (History)
 CC List: 1 user including you (edit)

See Also: (add)

Large text box: 1

A multiple-select box: Always Appears
 Also Always Appears
 Third Value, Always

Drop Down List: ---
 Date Time:
 Bug ID Field:

Flags: None yet set (set flags)

Orig. Est.:	Current Est.:	Hours Worked:	Hours Left:	%Complete:	Gain:	Deadline:
0.0	0.0	0.0 + 0	0.0	0	0.0	2015-01-09

Summarize time (including time for bugs blocking this bug)

Attachments 8
 Add an attachment (proposed patch, testcase, etc.)

Bug ID number is assigned to newly created bug

Step 5) In the same window if you scroll down further. You can select deadline date and also status of the bug. **Deadline in Bugzilla usually gives the time-limit to resolve the bug in given time frame.**

Orig. Est.:	Current Est.:	Hours Worked:	Hours Left:	%Complete:	Gain:	Deadline:
0.0	0.0	0.0 + 0	0.0	0	0.0	<input type="text"/>

Summarize time (including time for bugs blocking this bug)

Attachments
 Add an attachment (proposed patch, testcase, etc.)

Additional Comments:

Status: CONFIRMED
 CONFIRMED
 IN_PROGRESS
 RESOLVED

James 2015-01-07 02:50:31 PST

Description [reply] [-]

Collapse All Comments
 Expand All Comments

The widget gears are twisted at the end and not showing correct signal

Save Changes

select deadline for your bug-report

You can select bug status over here

Create Graphical Reports

Graphical reports are one way to view the current state of the bug database. You can run reports either through an HTML table or graphical line/pie/bar-chart-based one. The idea behind graphical report in Bugzilla is to define a set of bugs using the standard search interface and then choosing some aspect of that set to plot on the horizontal and vertical axes. You can also get a 3-dimensional report by choosing the option of "Multiple Pages".

Reports are helpful in many ways, for instance if you want to know which component has the largest number of bad bugs reported against it. In order to represent that in the graph, you can select severity on X-axis and component on Y-axis, and then click on generate report. It will generate a report with crucial information.

Bugzilla - Generate Graphical Report

Home | New | Browse | Search | [?] | Reports | My Requests | Preferences | Help | Log out jeegarguru@gmail.com

Notice: Due to a recent [data disclosure](#) all current passwords have been reset and will require a new password to be set the next time this site is accessed. To do so, click the "Forgot Password" link at the top.

Choose one or more fields as your axes, and then refine your set of bugs using the rest of the form.

1 **Vertical Axis:** Severity

2 **Plot Data Sets:**
☒ Individually
☐ Stacked

3 **Horizontal Axis:** Component

4 **Multiple Images:** <none>

5 **Format:**
☐ Line Graph
☒ Bar Chart
☐ Pie Chart

6 **Summary:** contains all of the strings

12 **Generate Report**

7 **Classification:** Unclassified, Widgets, Mercury

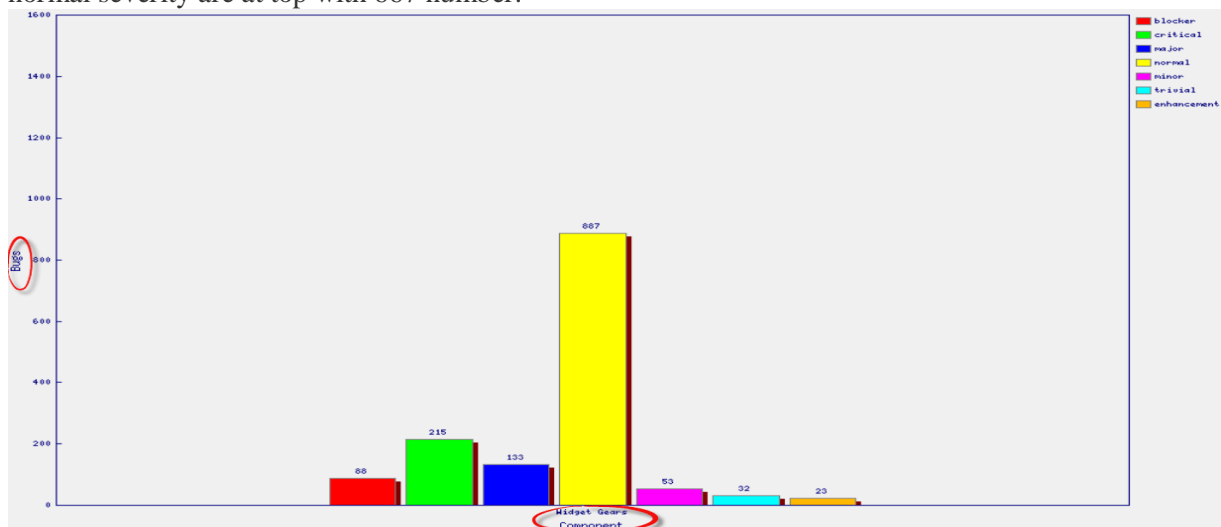
8 **Product:** Sam's Widget

9 **Component:** Widget Gears

10 **Status:** UNCONFIRMED, CONFIRMED, IN PROGRESS, RESOLVED, VERIFIED

11 **Resolution:** FIXED, INVALID, WONTFIX, LATER, REMIND, DUPLICATE

The graph below shows the Bar chart representation for the Bugs severity in component "**Widget Gears**". In the graph below, the most severe bug or blockers in components are 88 while bugs with normal severity are at top with 667 number.



Likewise, we will also see the line graph for **%complete Vs Deadline**

Step 1) To view your report in a graphical presentation,

- Click on Report from Main Menu
- Click on the Graphical reports from the given option

Bugzilla – Reporting and Charting Kitchen

Home | New | Browse | Search | Search [?] **Reports** My Requests | Preferences | Help

Notice: Due to a recent [data disclosure](#) all current passwords have been reset and will require a new password to be set the next time this site is accessed. To do so, click the "Forgot Password" link at the top.

Bugzilla allows you to view and track the state of the bug database in all manner of exciting ways.

Current State

- [Search](#) - list sets of bugs.
- [Tabular reports](#) - tables of bug counts in 1, 2 or 3 dimensions, as HTML or CSV.
- [Graphical reports](#) - line graphs, bar and pie charts.
- [Duplicates](#) - list of most frequently reported bugs.

Change Over Time

- [Old Charts](#) - plot the status and/or resolution of bugs against time, for each product in your database.
- [New Charts](#) - plot any arbitrary search against time. Far more powerful.

Step 2) Let's create a graph of % Complete Vs Deadline

In here on the vertical axis we chose **% Complete** and on our horizontal axis we chose **Deadline**. This will give the graph of amount of work done in percentage against the set-deadline.

Now, set various option to present reports graphically

1. Vertical Axis
2. Horizontal Axis
3. Multiple Images
4. Format- Line graph, Bar chart or Pie chart
5. Plot data set
6. Classify your bug
7. Classify your product
8. Classify your component
9. Classify bug status
10. Select resolution
11. Click on generate report

Bugzilla – Generate Graphical Report

Home | New | Browse | Search | Search [?] Reports | My Requests | Preferences | Help | Log out jeegarguru@gmail.com

Notice: Due to a recent [data disclosure](#) all current passwords have been reset and will require a new password to be set the next time this site is accessed. To do so, click the "Forgot Password" link at the top.

Choose one or more fields as your axes, and then refine your set of bugs using the rest of the form.

1 **Vertical Axis:** %Complete

5 **Plot Data Sets:** ☒ Individually ☐ Stacked

3 **Multiple Images:** Severity

4 **Format:** ☒ Line Graph ☐ Bar Chart ☐ Pie Chart

2 **Horizontal Axis:** Deadline

Vertical labels: ☐

Summary: contains all of the strings

6 **Classification:** Unclassified Widgets Mercury

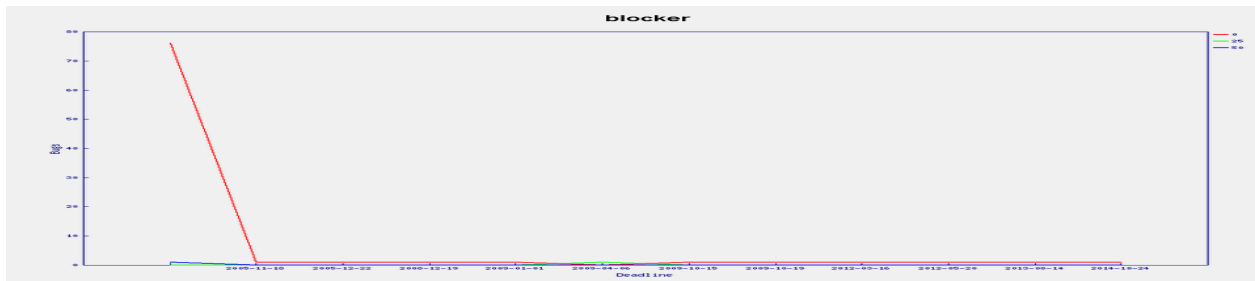
7 **Product:** Sam's Widget

8 **Component:** Widget Gears

9 **Status:** UNCONFIRMED CONFIRMED IN_PROGRESS RESOLVED VERIFIED

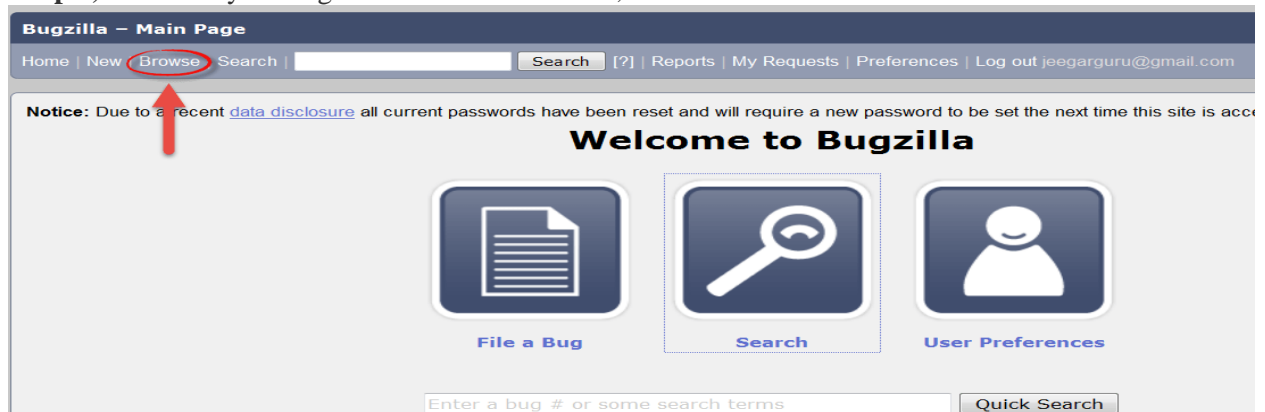
10 **Resolution:** --- FIXED INVALID WONTFIX LATER REMIND SUBDUCE

The image of the graph will appear somewhat like this



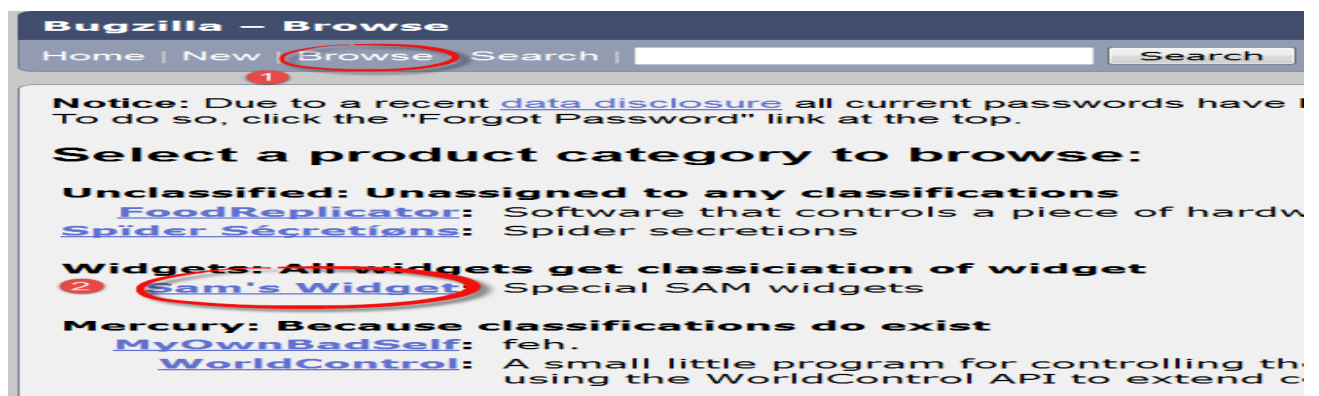
Browse Function

Step 1) To locate your bug we use browse function, click on **Browse** button from the main menu.



Step 2) As soon as you click on browse button a window will open saying "Select a product category to browse" as shown below, we browse the bug according to the category.

- After clicking the browse button
- Select the product "Sam's Widget" as such you have created a bug inside it



Step 3) It opens another window, in this click on component "**widget gears**". Bugzilla Components are sub-sections of a product. For instance, here our product is **SAM'S WIDGET** whose component is **WIDGET GEARS**.

Bugzilla – Components for Sam's Widget

Home | New | Browse | Search | Search [?] | Reports | My Requests

Notice: Due to a recent [data disclosure](#) all current passwords have been reset and will require a new password. Please click on the "Forgot Password" link at the top.

Select a component that component: **Widget Gears** **Default Assignee** **Default QA Contact**

Widget Gears **Sam's Widget**
Special SAM widgets

Default Assignee **Default QA Contact**
Sam folk-williams
Gears for Sam's widgets

Step 4) when you click on the component, it will open another window. All the Bugs created under particular category will be listed over-here. From that Bug-list, choose your Bug#ID to see more details about the bug.

Wed Jan 7 2015 20:59:10 PST

[Hide Search Description](#)

Resolution: --- Component: Widget Gears Product: Sam's Widget

This result was limited to 500 bugs. [See all search results for this query.](#)

ID	Product	Comp	Assignee	Status	Resolution	Summary	Changed
1256	Sam's Wi	Widget G	justdave@syndicomm.com	CONF	---	summary	2014-10-23
4219	Sam's Wi	Widget G	justdave@syndicomm.com	CONF	---	sdadasad	2012-06-13
4742	Sam's Wi	Widget G	stdave@syndicomm.com	CONF	---	just a test	2009-01-26
5509	Sam's Wi	Widget G	stdave@syndicomm.com	CONF	---	Test Bug	2014-05-04
2566	Sam's Wi	Widget G	ndfill@gavinsharp.com	CONF	---	test	2009-11-04
6504	Sam's Wi	Widget G	mabst45@gmail.com	CONF	---	Won't run	2010-09-21
3010	Sam's Wi	Widget G	mickesnow@yahoo.com.mx	CONF	---	buhigs	2012-03-15
24741	Sam's Wi	Widget G	neha.malik028@gmail.com	CONF	---	cancel button not working	2014-10-16

It will open another window, where information about your bug can be seen more in detail. In the same window, you can also change the assignee, QA contact or CC list.

Bug 1256 - [summary](#) ([edit](#)) [Save Changes](#)

Status: CONFIRMED ([edit](#))

Product: Sam's Widget

Component: Widget Gears

Version: unspecified

Hardware: HP Linux

Importance: P1 normal

Target Milestone: ---

Assigned To: [Dave Miller](#) ([edit](#)) ([take](#))

QA Contact: ([edit](#)) ([take](#))

URL:

Whiteboard:

Reported: 2003-05-27 13:28 PDT by [Jason McCallum](#)

Modified: 2014-10-23 08:09 PDT ([History](#))

CC List: ☒ Add me to CC list
6 users ([edit](#))

See Also: ([add](#))

Large text box:

free text:

A multiple-select box: Always Appears
Also Always Appears
Third Value, Always

How to use Simple search option in BugZilla

Bugzilla provides two ways of searching bugs, they are **Simple Search** and **Advance Search** methods.

Step 1) We will first learn the "Simple Search" method. Click on search button from the main menu and then follow these steps

1. Click on "Simple Search" button
2. Choose the status of the Bug – choose Open if you are looking the bug in Open status and closed for bug in closed status
3. Choose your category and component, and you can also put keywords related to your bug
4. Click on the search

Bugzilla – Simple Search

Home | New | Browse | **Search** | Search | [?] | Reports | My Requests | Preferences | Log out jeegarguru@gmail.com

Notice: Due to a recent [data disclosure](#) all current passwords have been reset and will require a new password to be set the next time this site is accessed. To do so, click the "Forgot Password" link at the top.

Simple Search | **Advanced Search**

Find a specific bug by entering words that describe it. Bugzilla will search bug descriptions and comments for those words and return a list of matching bugs sorted by relevance.

For example, if the bug you are looking for is a crash in the SSL flash search for "crash secure SSL flash".

Status: **Open** | Closed | All

Product: Sam's Widget

Words: Widgets gears

Search

Unclassified
All
FoodReplicator
Spider Sécretions

Widgets
Sam's Widget
Mercury
MyOwnBadSelf
WorldControl

Step 2) Here we will search for both option **open** and **closed** status, first we have selected closed status for bug and clicked search button.

Simple Search

Find a specific bug by entering words that describe it. Bugzilla will search for matching bugs sorted by relevance.

For example, if the bug you are looking for is a crash in the SSL flash search for "crash secure SSL flash".

Status: **Closed** | Open | All

Product: All

Words: widget gears

Search

Choose the keywords for the bug you looking for

For closed status, it fetched 12 bugs.

12 bugs found.

ID	Assignee	Resolution	Changed
Comp	Status	Summary	
12412	Widget G sam.folkwilliams@gmail.com	RESO FIXE The widget stole my spacecraft!!	2011-08-27
2998	WeatherC peter.rutherford@cnh.com	VERI FIXE Hardlinks not created and the world is thence seriously out of control	2010-11-23
15235	WeatherC tara@bluemartini.com	RESO DUPL weather widget failed rain test	2011-06-14
4297	Widget G justdave@syndicomm.com	RESO FIXE transation 01 - fails	2010-03-31
6312	Widget G sam.folkwilliams@gmail.com	RESO FIXE lack of cogs in widget	2008-02-05

Step 3) Likewise we have searched for Open status as well, and it has fetched 37 bugs related to our queries.

37 bugs found.				
ID	Product	Comp	Assignee	Status
12431	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	IN_P
12561	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF
7359	Sam's Wi	Widget G	raydevereaux@bc.com	IN_P
8907	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF

Also, at the bottom of the screen you have various options like how you want to see your bug - **an XML format, in Long format or just Time summary**. Apart from that you can also use other option like **send mail to bug assignee, change several bugs at once or change column of the screen**, etc.

The screenshot shows the bottom of the bug tracking interface. On the left, there are three buttons: 'Long Format', 'XML', and 'Time Summary'. A callout points to these buttons, stating: 'Use these options to view your bug in Long format, XML format or just time summary'. In the center, there are several links: 'CSV', 'Feed', 'iCalendar', 'Change Columns', 'Change Several Bugs at Once', and 'Send Mail to Bug Assignees'. A callout points to these links, stating: 'You can use these options for sending mail to assignees, change several bugs at once, change columns and so on.' On the right, there is an 'Edit Search' link and a 'Remember search' checkbox.

In next step, we will demonstrate one of this function **change column of the screen**, through which we will learn how to add or remove the column to the existing column.

How to add or remove column to default search screen

Step 1) Click on the **Change Column** as shown in above screen-shot. It will open a new window where you have to follow these steps.

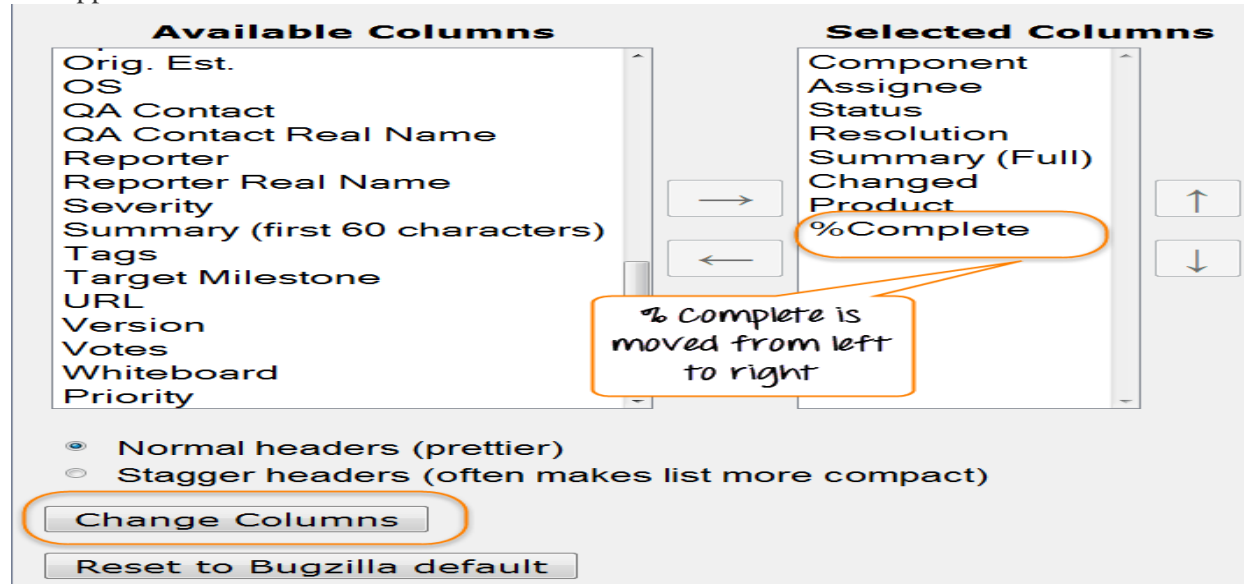
- Select any given option from the column you want to appear in the main screen - here we have selected **% complete**
- Click on the **arrow button**, it will move % complete column from ☐ **Available Column** to the **Selected column**

These steps will move the selected column from left to right.

The screenshot shows a dialog box titled 'Select the columns you wish to appear in your bug lists. Note that this'. It has two main sections: 'Available Columns' on the left and 'Selected Columns' on the right. In the 'Available Columns' list, '%Complete' is selected and highlighted with a red circle and a red arrow pointing to it. In the 'Selected Columns' list, 'Component', 'Assignee', 'Status', 'Resolution', 'Summary (Full)', 'Changed', and 'Product' are listed. A red arrow points from the '%Complete' in the 'Available Columns' list to the 'Selected Columns' list. A callout points to this arrow, stating: 'You can select + move any given option to the selected columns'. There are also up and down arrow buttons on the right side of the 'Selected Columns' list.

The % complete is moved from left to right as shown below, and once we click on **change column** it

will appear in the main screen



Before- Search result screen before using "Change Column" option-

- There is no % complete column appears in search screen result as shown below

ID	Product	Comp	Assignee	Status	Resolution	Summary	Changed
12431	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	IN_P	---	Widget Gears cannot start	2014-12-02
12561	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF	---	Gear is broke	2014-12-08
7359	Sam's Wi	Widget G	raydevereaux@bc.com	IN_P	---	funny	2011-05-01
8907	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF	---	Monkey Wren gears	2009-07-23

After- Search result screen after using "Change Column" option

- You can see **% complete** column added to the extreme right in the existing column in the main screen, which was not their previously.

37 bugs found.							
ID	Comp	Assignee	Status	Resolution	Summary	Changed	%Complete
12431	Widget G	sam.folkwilliams@gmail.com	IN_P	---	Widget Gears cannot start	2014-12-02	0 %
12561	Widget G	sam.folkwilliams@gmail.com	CONF	---	Gear is broke	2014-12-08	0 %
7359	Widget G	raydevereaux@bc.com	IN_P	---	funny	2011-05-01	0 %
8907	Widget G	sam.folkwilliams@gmail.com	CONF	---	Monkey Wren gears	2009-07-23	0 %

How to use Advance Search in BugZilla

Step 1) After Simple search we will look into **Advanced Search** option for that you have to follow the following steps.

- Click on advanced search option
- Select option for summary, how you want to search
- Enter the keyword for your bug- for example, **Widget gears twisted**

4. Select the category of your Bug under classification, here we selected Widget
5. Choose your product under which your Bug was created- Sam's Widget
6. Component- Widget gears
7. Status- Confirmed
8. Resolution

The screenshot shows the 'Advanced Search' form in Bugzilla. It includes a 'Summary' dropdown menu with search criteria like 'contains all of the strings'. Below this are several filter sections: 'Classification' (with 'Widgets' selected), 'Product' (with 'Sam's Widget' selected), 'Component' (with 'Widget Gears' selected), 'Status' (with 'CONFIRMED' selected), and 'Resolution' (with 'FIXED' selected). A callout bubble points to the 'Component' field with the text 'Here you can enter your Bug keywords'.

Step 2) Once you select all the option, click on search button. It will detect the bug you created

This screenshot shows the search results page. At the top, the search criteria are displayed: 'Summary: contains all of the strings' and 'widget gears twisted'. Below this, the filters are shown: 'Classification: Widgets', 'Product: Sam's Widget', 'Component: Widget Gears', 'Status: CONFIRMED', and 'Resolution: FIXED'. A 'Search' button is circled in red.

The advance search will find your bug, and it will appear on the screen like this

Wed Jan 7 2015 23:43:39 PST

[XD 萬歲](#)

[Hide Search Description](#)

Summary: widget gears twisted Resolution: --- Classification: Widgets Status: CONFIRMED Component: Widget Gears
 Product: Sam's Widget

ID	Assignee	Resolution	Changed
Comp	Status	Summary	%Complete
26320 Widget G	sam.folkwilliams@gmail.com	CONF --- Gears for sams widget twisted	Wed 03:10 0 %
Totals			0 %

One bug found.

Long Format XML CSV Feed iCalendar Change Columns Edit Search

Time Summary

Your bug appears on screen

How to use preferences in BugZilla

Preferences in Bugzilla is used to customize the default setting made by Bugzilla as per our requirement. There are mainly five preferences available

- General Preferences
- E-mail Preferences
- Saved Searches
- Account Information
- Permissions

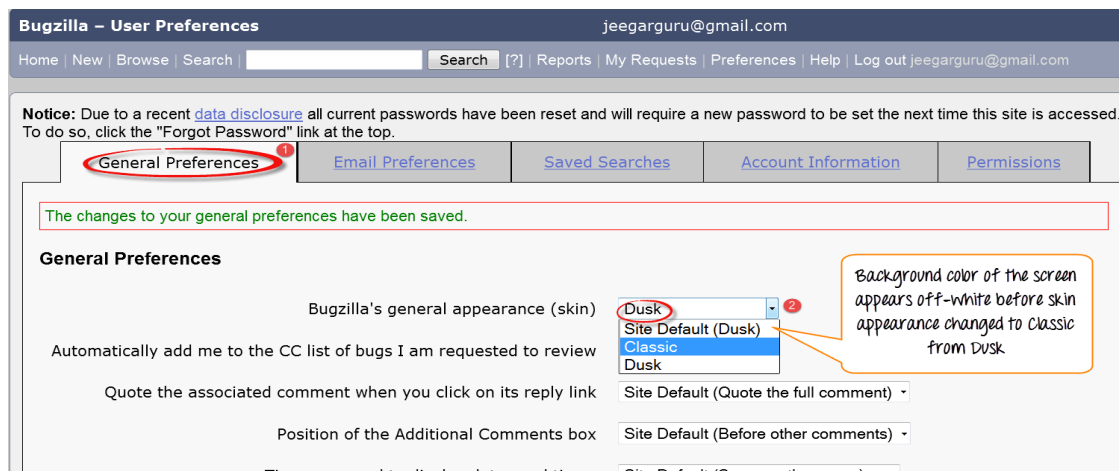
General Preferences

For **general preferences**, you have various option like **changing Bugzilla general appearance, position of the additional comment box, automatically add me to cc**, etc. Here we will see how to change the general appearance of the Bugzilla.

There are many changes you can do which are self-explanatory, and you can choose the option as per your requirement.

Step 1)

- To set the background Skin of Bugzilla
- Go to Bugzilla general preference (Skin)
- Select the option you want to see as a change and submit the change (Dusk ☐ Classic)
- A message will appear on the window saying changes have been saved, as soon as you submit the changes



After the skin preference is changed to Classic from Dusk, the back-ground color of the screen appears white

Notice: Due to a recent [data disclosure](#) all current passwords have been reset and will require a new password accessed. To do so, click the "Forgot Password" link at the top.

[General Preferences](#)
[Email Preferences](#)
[Saved Searches](#)

The changes to your general preferences have been saved.

Background color changes to white, after skin color changed from Dusk to Classic

General Preferences

Bugzilla's general appearance (skin)

Classic

Automatically add me to the CC list of bugs I am requested to review

Site Default (On)

Quote the associated comment when you click on its reply link

Site Default (Quote the full comment)

Position of the Additional Comments box

Site Default (Before other comments)

Timezone used to display dates and times

Site Default (Same as the server)

Likewise, for other default settings changes can be done.

E-mail preferences

E-mail preferences enable you to decide how to receive the message and from whom to receive the messages.

Step 1) To set the e-mail preferences

1. Click on e-mail services
2. Enable or disable the mail to avoid receiving notification about changes to a bug
3. Receiving mail when someone asks to set a flag or when someone sets a flag you asked for
4. When and from whom you want to receive mail and under which condition. After marking your option at the end, submit the changes.

General Preferences	Email Preferences	Saved Searches	Account Information	Permissions																			
<p>Email Preferences</p> <p>If you want to change your e-mail address jeegarguru@gmail.com to which Bugzilla sends notifications, click the "Account Information" tab above.</p> <p>If you don't like getting a notification for "trivial" changes to bugs, you can use the settings below to filter some or all notifications.</p> <p> <input type="button" value="Enable All Mail"/> <input type="button" value="Disable All Mail"/> </p> <p>Global options:</p> <p> <input checked="" type="checkbox"/> Email me when someone asks me to set a flag <input checked="" type="checkbox"/> Email me when someone sets a flag I asked for </p> <p>Field/recipient specific options:</p> <table border="1"> <thead> <tr> <th colspan="4">When my relationship to this bug is:</th> <th rowspan="2">I want to receive mail when:</th> </tr> <tr> <th>Assignee</th> <th>Reporter</th> <th>CCed</th> <th>QA Contact</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td>I'm added to or removed from this capacity</td> </tr> <tr> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td>A new bug is created</td> </tr> </tbody> </table>					When my relationship to this bug is:				I want to receive mail when:	Assignee	Reporter	CCed	QA Contact	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	I'm added to or removed from this capacity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	A new bug is created
When my relationship to this bug is:				I want to receive mail when:																			
Assignee	Reporter	CCed	QA Contact																				
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	I'm added to or removed from this capacity																			
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	A new bug is created																			

Saved Searches Preference

Saved searches preference gives you the freedom to decide whether to share your bug or not to share.

Step 1) Click on saved searches, it will open window with the option like **editbugs**, **don't share**, **canconfirm**, etc. Choose the option as per your need.

General Preferences | Email Preferences | **Saved Searches** | Account Information | Permissions

The changes to your saved searches have been saved.

Saved Searches

Your saved searches are as follows:

Search	Run	Edit	Forget	Show in Footer	Share With a Group
My Bugs	Run			<input checked="" type="checkbox"/>	—
Widget gears	Run	Edit	Forget	<input checked="" type="checkbox"/>	editbugs ▾

You may use these searches saved and shared by others:

Step 2) We can run our bug from "Saved Searches".

- Go to Saved Searches under preference
- Click on the "**Run**" button

General Preferences | Email Preferences | **Saved Searches**

Saved Searches

Your saved searches are as follows:

Search	Run	Edit	Forget	Show in Footer	Share With a Group
My Bugs	Run			<input checked="" type="checkbox"/>	—
Widget gears	Run	Edit	Forget	<input checked="" type="checkbox"/>	canconfirm ▾

We will run our search or queries from here

As soon as you run your search from Saved Searches it opens your bug as shown below

Resolution: --- Assignee: jeegarguru@gmail.com Reporter: jeegarguru@gmail.com

ID	Product	Comp	Assignee	Status	Resolution	Summary	Changed
26330	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF	---	Widget gears is not responsive	Wed 22:06
26320	Sam's Wi	Widget G	sam.folkwilliams@gmail.com	CONF	---	Gears for sams widget twisted	Wed 03:10









2 bugs found.

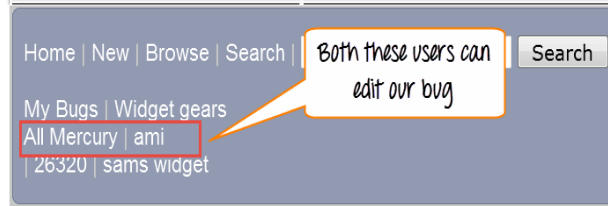
Long Format | XML | Time Summary

CSV | Feed | iCalendar | Change Columns | Change Several Bugs at Once | Edit Search

Remember search as

Step 3) In the same window we can also choose specific users with whom we want to share the search by marking or unmarking the checkbox against the users

Search	Shared By	Shared To	Run	Edit	Show in Footer
123	Amrita <osmosys.dici@gmail.com>	editbugs	Run	Edit	 
All Mercury	Ben Schultz (SofTechnics) <ben.schultz@softechnics.com>	editbugs	Run	Edit	 
ami	Ami <ami_nahmani@walla.com>	editbugs	Run	Edit	 
Bug1	Aruna <aruna_trimurti@yahoo.com>	editbugs	Run	Edit	 



viva questions

1. What is Bugzilla?
2. What are Bugzilla,,s features?
3. What are Bugzilla Components?
4. How do I change my user name in Bugzilla?
5. What is about the Bug List page?
6. How to Write a Useful Bug Report with Bugzilla

EXPERIMENT: 9

NAME OF THE EXPERIMENT: Study of Any Test Management Tool (Test Director)

Test Director is a global test management solution which provides communication, organization, documentation and structure to the testing project.

Test Director

Software Automated Tool TestDirector simplifies test management by helping you organize and manage all phases of the software testing process, including planning, creating tests, executing tests, and tracking defects.

PLANNING TESTS:

Divide your application into test subjects and build a project.

1. Define your testing goals.

Examine your application, system environment, and testing resources to determine what and how you want to test.

2. Define test subjects.





















Define test subjects by dividing your application into modules or functions to be tested. Build a test plan tree that represents the hierarchical relationship of the subjects.

3. Define tests.

Determine the tests you want to create and add a description of each test to the test plan tree.

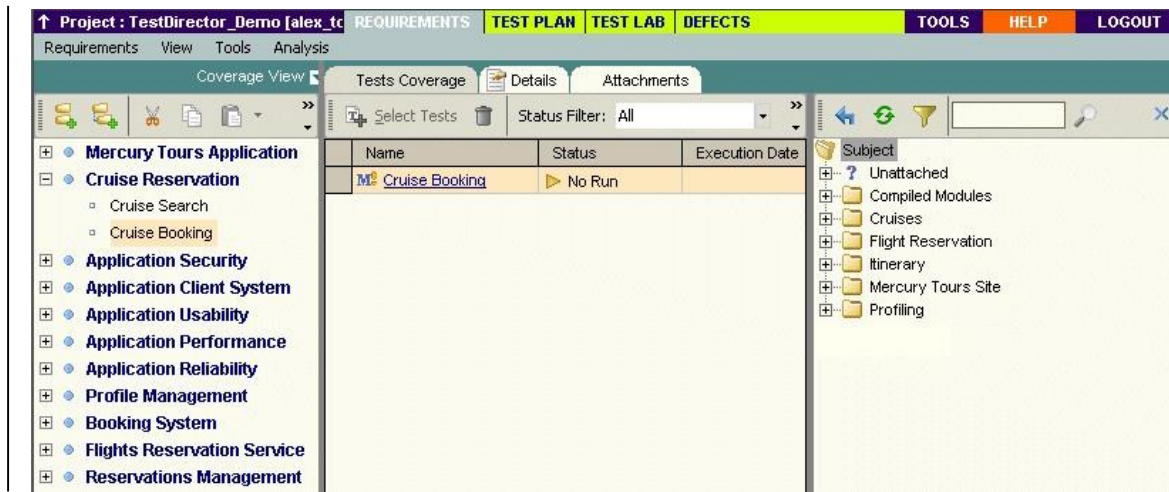
4. Design test steps.

Break down each test into steps describing the operations to be performed and the points you want to check. Define the expected outcome of each step.

Details		Design Steps		Test Script		Attachments		Reqs Coverage											
																			
Step Name		Description								Expected Result									
Display the Cruise Special page.		Click the Cruises button.								The Cruise Special page opens.									
Display the Cruise Reservation page.		Click the Now Accepting Reservations button.								The Cruise Reservation page opens.									
Book the cruise.		Enter passenger name, credit card information, and address. Click OK.								The Cruise Confirmation page opens.									
Print cruise confirmation.		Click the Print button.								A confirmation page is printed.									
Log off.		Click the Sign-Off button.								Returns to the Sign-On page.									

6. Analyze the test plan.

Generate reports and graphs to help you analyze your test plan. Determine whether the tests in the project will enable you to successfully meet your goals.



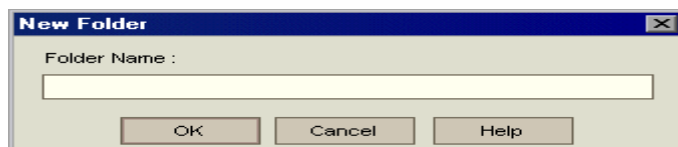
RUNNING TESTS:

Create test sets and perform test runs.

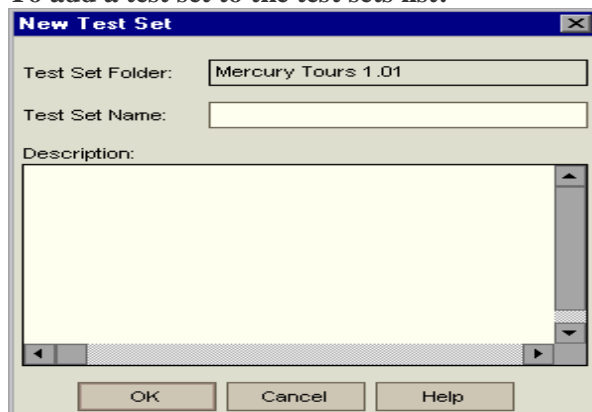
1. Create test sets.

Create test sets by selecting tests from the project. A test set is a group of tests you execute to meet a specific testing goal.

To define a test set:



To add a test set to the test sets list:

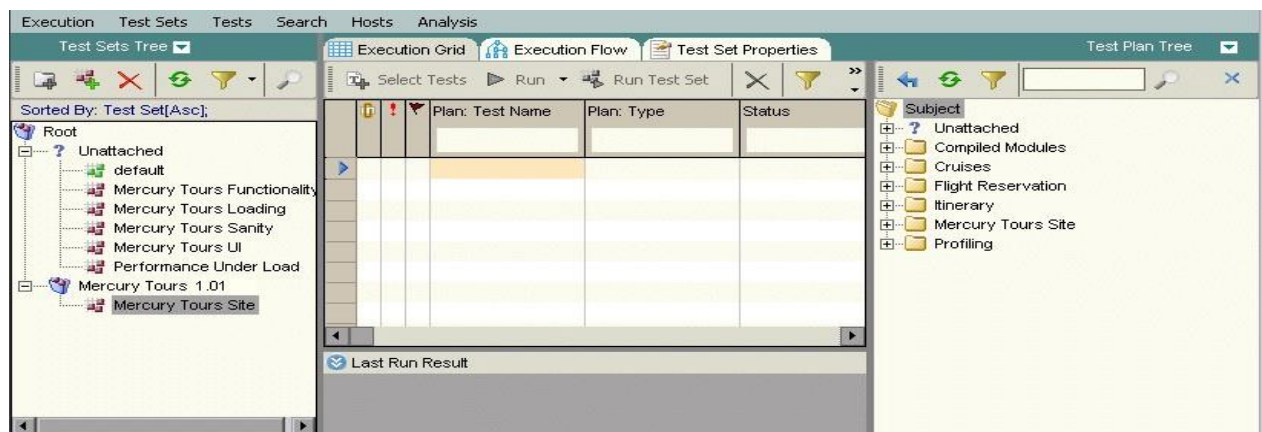


Define the test set details:

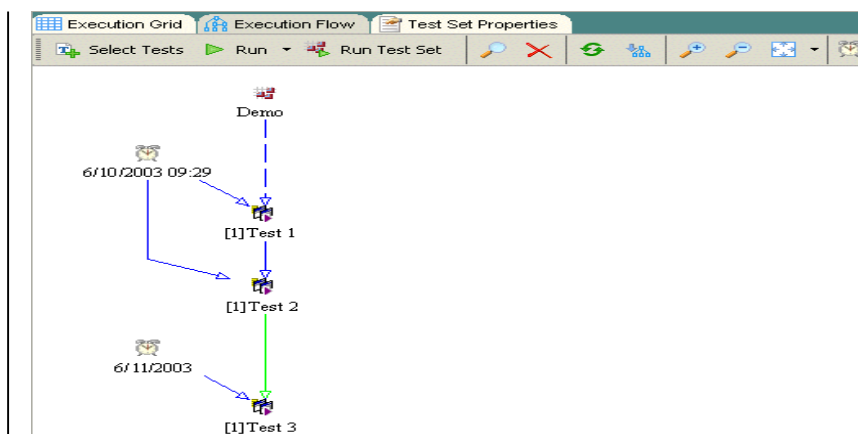
The screenshot shows the 'Test Set Properties' dialog box with the following details:

- Test Set Information:**
 - Description: This test set includes tests that verify the functionality of the Mercury Tours site.
 - Close Date: (empty dropdown)
 - Open Date: 6/3/2003
 - Status: Open
- User Defined Fields:**
 - Level: (empty dropdown)
 - Version: (empty dropdown with ellipsis)
 - Project: (empty dropdown)

Adding tests to a test set:



Scheduling Test Runs:



Schedule test execution and assign tasks to testers. Run the manual and/or automated tests in the test sets.

Running Tests Manually:

Manual Runner Test Set: <Mercury Tours Site> Test: <[1]Cruise Booking>

Exec Steps End of Run Cancel

Run Details

Run Name: Run_6-3_17-34-55 [Operating System Info](#)

Tester: cecil_td

Status: Not Completed

Exec Date: 6/3/2003

Exec Time: 5:34:55 PM

Browser: Product Version: Language:

Test Details

Name: Cruise Booking [More...](#)

Run the tests:

Execution of <Mercury Tours Site>

Run Settings

Run All Run Stop

☒ Run All Tests Locally ☒ Enable Log

Test Name	Run on Host	Status
[1]Number Of Passengers	LANDLOPER	✓ Passed

View a summary of test results in the execution grid:

Execution Grid Execution Flow Test Set Properties

Select Tests Run Run Test Set

Plan: Test Name	Plan: Type	Status	Planned Host Name	Responsible Tester
[1]Cancel All Reservations	WR-AUTOMATED	No Run		
[1]Airline Preference	WR-AUTOMATED	No Run		
[1]Number Of Passengers	QUICKTEST_TEST	✓ Passed		
[1]Cruise Booking	MANUAL	✗ Failed		
[1]Cruise Search	WR-AUTOMATED	No Run		

Last Run Result

Launch Report

Step Name	Status	Description:
Start Test		Start Test Number Of Passengers
Start Global Iteration		
Start Action		
Replay Error	✓ Passed	

Expected: Actual:

4. Analyze the testing progress.

Generate reports and graphs to help you determine the progress of test execution

TRACKING DEFECTS:

1. Report defects detected in your application and track how repairs are progressing.
2. Report defects detected in the software. Each new defect is added to the defect database.
3. Track defects.

Review all new defects reported to the database and decide which ones should be repaired. Test a new version of the application after the defects are corrected.

Updating Defects:

Defect: 44 | Unable to reserve a cruise from the Cruise page

Page 1 | Page 2

* Category: Defect
Project: Mercury Tours (HTB)
* Subject: Cruises
* Reproducible: Y
* Detected By: cecil_td
Assigned To: james_td

* Status: New
* Detected in Version: Version 1.01
* Detected on Date: 6/5/2003
Regression: N
* Severity: 2-Medium
Priority: 3-High

Planned
Planned Closing Version: ...
Estimated Fix Time: (Hours)

Actual
Closed in Version: ...
Actual Fix Time: (Hours)
Closing Date: ...

OK Cancel

4. Analyze defect tracking.

Generate reports and graphs to help you analyze the progress of defect repairs, and to help you determine when to release the application.

- a) end a defect report to the TestDirector database.
- b) Review the defect and assign it to a member of the development team.
- c) Repair the open defect.
- d) Test a new build of the application after the defect is corrected. If the defect does not reoccur, change the status of the defect.
- e) Generate reports and graphs to help you analyze the progress of the defects in your TestDirector project.

f) Reporting a New Defect

You can report a new defect at any stage of the testing process by adding a defect record to the project database. Each defect is tracked through four stages: New, Open, Fixed, and Closed. When you initially report a defect to the project database, you assign it the status New.

Viva questions

1. What is test director?
2. What are the modules of test director?
3. What is the use of test director?
4. What are the various reports in test director?

EXPERIMENT: 10

NAME OF THE EXPERIMENT: Study of any open source testing tool (TestLink)

Testlink is an open source test management tool. It enables creation and organization of test cases and helps manage into test plan.

Login to TestLink

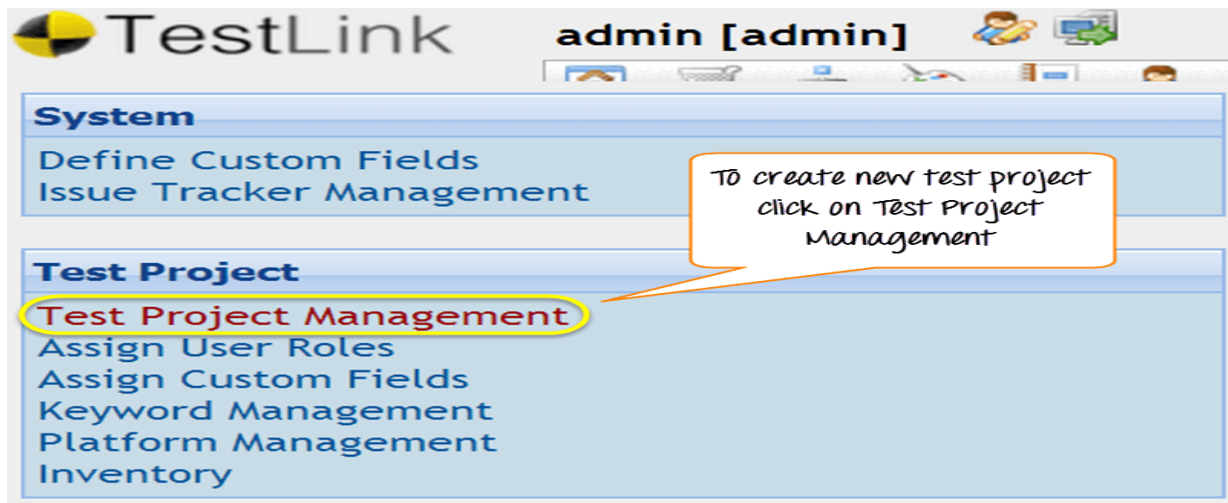
Step 1 : Open the Testlink home-page and enter the login details

1. Enter the userID – admin
2. Enter the password
3. Click on the login tab

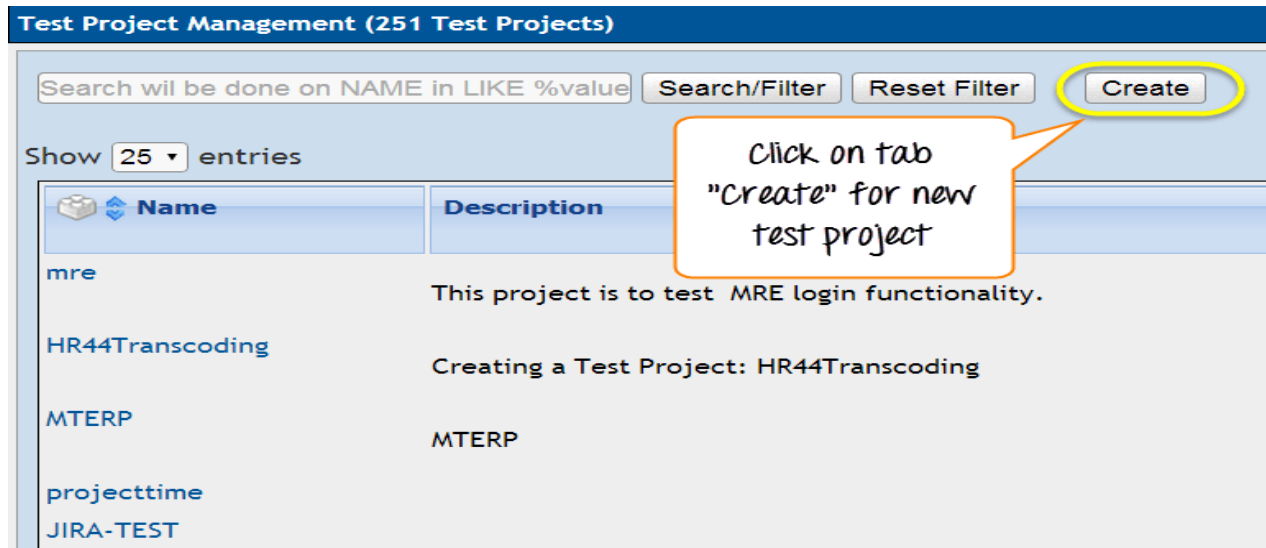


Creating a Test Project

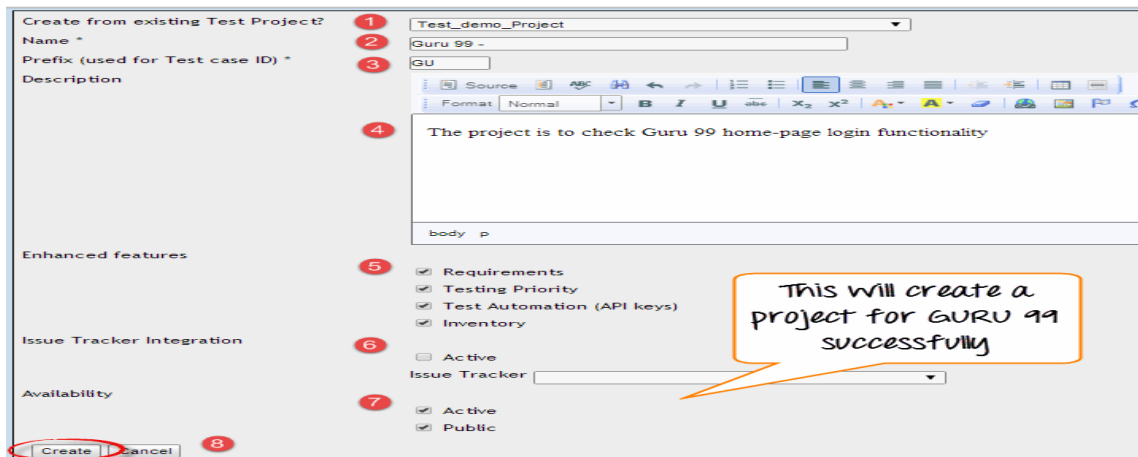
Step 1: In the main window click on Test Project Management, it will open another window



Step 2: Click on tab "create" to create a new project.



Step 3: Enter all the required fields in the window like category for test project, name of the project, prefix, description, etc. After filling all necessary details, click on tab "Create" at the end of the window.



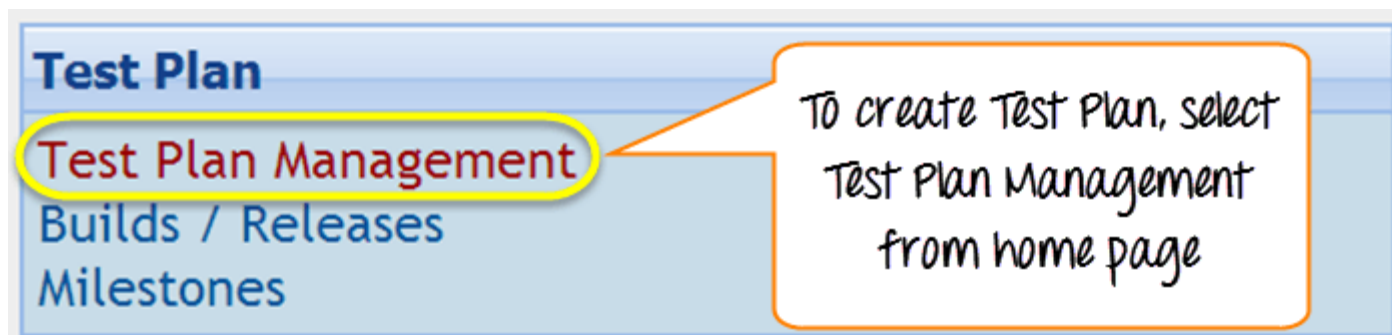
This will create your project "Guru99" successfully.



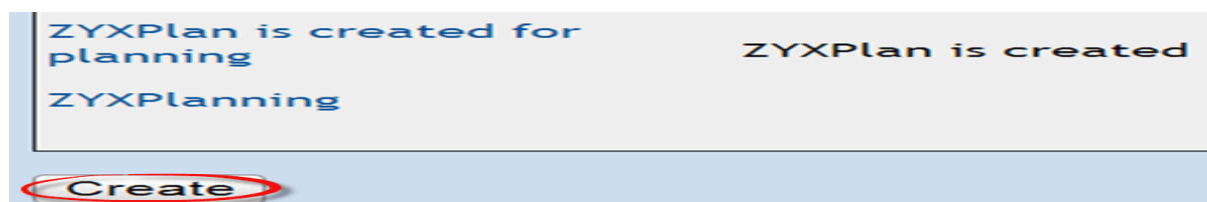
Creating a Test Plan

Test plan holds the complete information like scope of Software testing, milestone, test suites and test cases. Once you have created a Test Project, next step is to create Test plan.

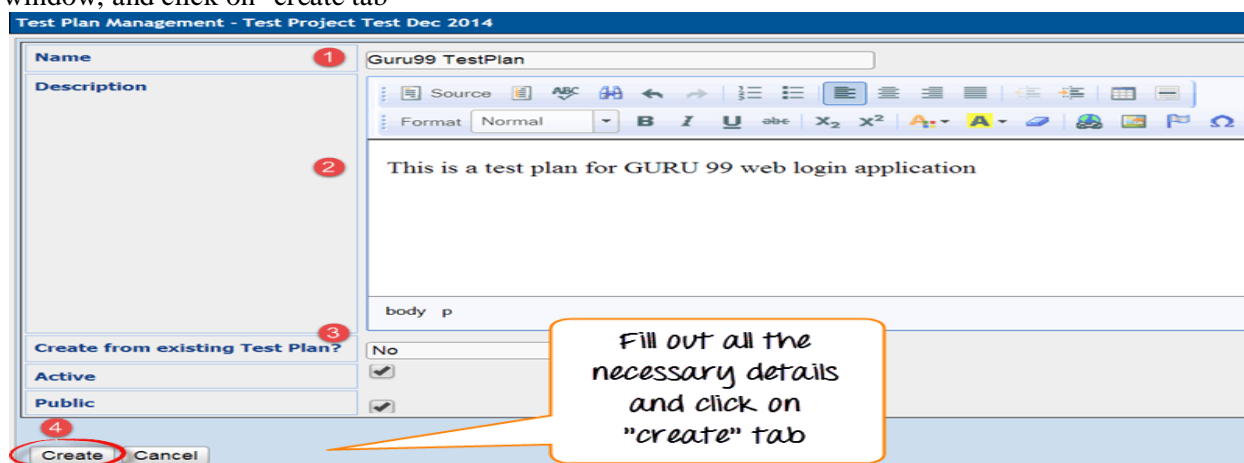
Step 1: From the home-page, click on Test Plan Management from home-page



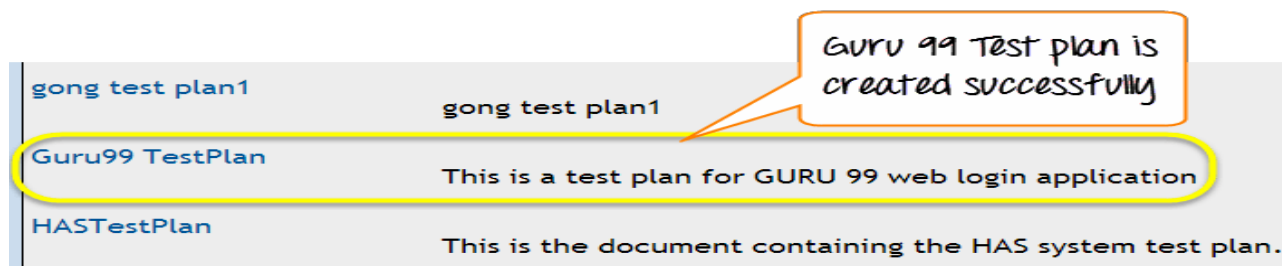
Step 2: It will open another page, at the bottom of the page click on a tab "Create"



Step 3: Fill out all the necessary information like name, description, create from existing test plan, etc. in the open window, and click on "create tab"



Step 4: Guru 99 Test Plan is created successfully



Build Creation

Build is a specific release of software

Step 1: Click on Builds/Releases under Test Plan from the home page

Test Plan

Test Plan Management Builds / Releases Milestones

From home-page select
Build/Release for creating specific
release for your software

Step 2: In the next window, fill all necessary details for software release and click on create to save your release

1. Enter the title name
2. Enter the description about the software release
3. Mark the check-box for status- Active
4. Mark the check-box for status- Open
5. Choose the data of release
6. Click on create button

Create a new Build

Title: 1 Guru 99- sample build

Description: 2 This is a sample build for Guru 99 web application

Active: 3 ☒

Open: 4 ☒

Release date: 15/01/2015 5

Copy tester assignments: ☐

Create 6 Cancel

Once you have a release the software it, will appear like this

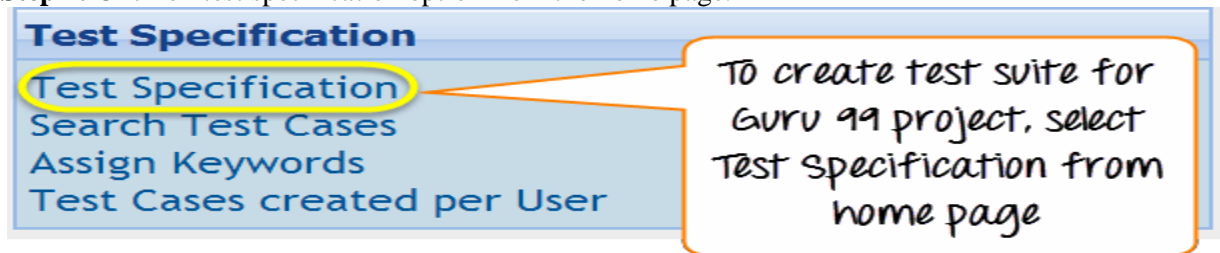
Title	Description
Guru 99- sample build	This is a sample build for Guru 99 web application
Test Build	Just for testing
samplebuild	this is sample build


Sample build for
Guru 99 is created
successfully

Creating Test suite

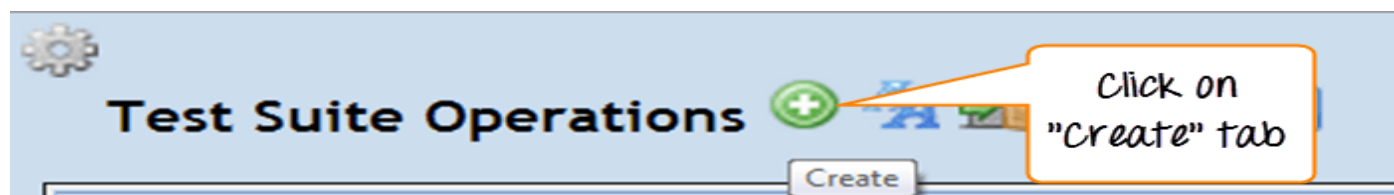
Test suite is a collection of test cases which may be testing or validating the same component. Following steps will explain how to create test suite for your project.

Step 1: Click on test specification option from the home page.



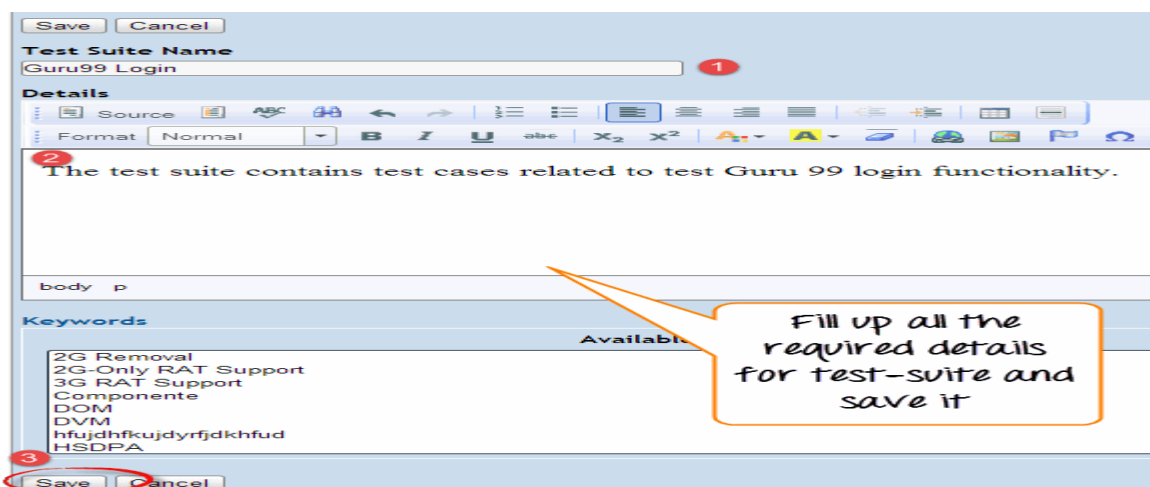
Step 2: On the right-hand side of the panel, click on the setting icon . It will display a series of test operation.

Step 3: Click on the "create" tab for the test suite

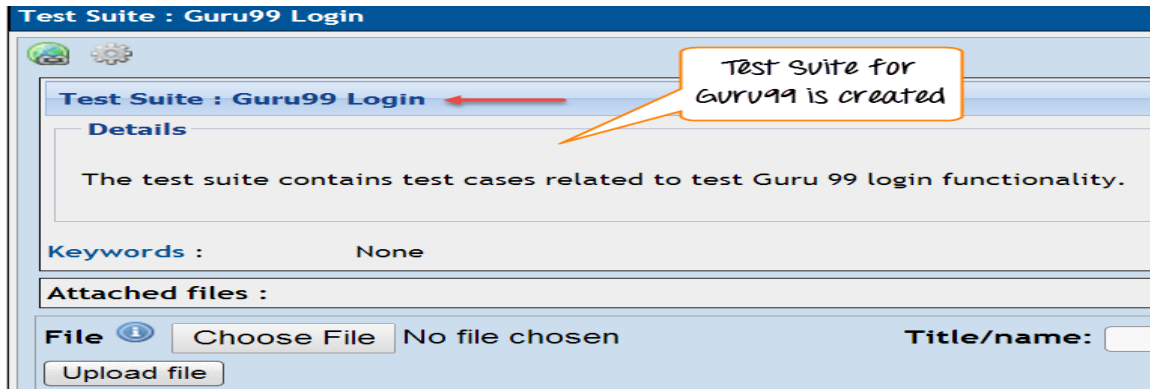


Step 4: Fill-up all the details for test-suite and click on save it tab.

1. Enter the test suite name
2. Enter the details about your test suite
3. Click on save button to save the details of test-suite



You can see test suite for Guru 99 is created

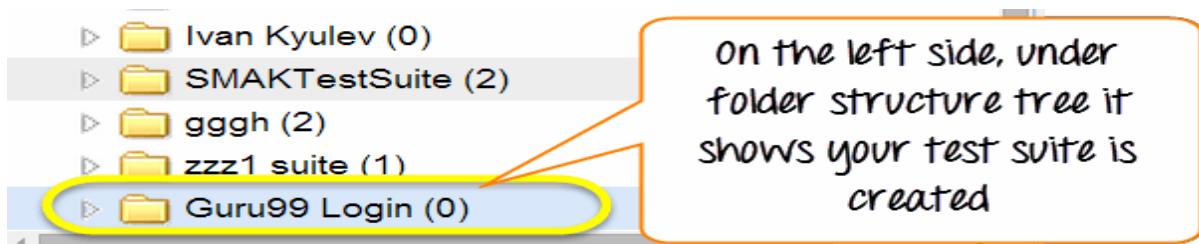


Your test suite appears on the left side of the panel under folder structure tree

Creating a Testcase

Testcase holds a sequence of test steps to test a specific scenario with expected result. Below steps will explain how to create a test-case along with test steps.

Step 1: Click on the test suite folder on the left side of the panel under folder tree structure



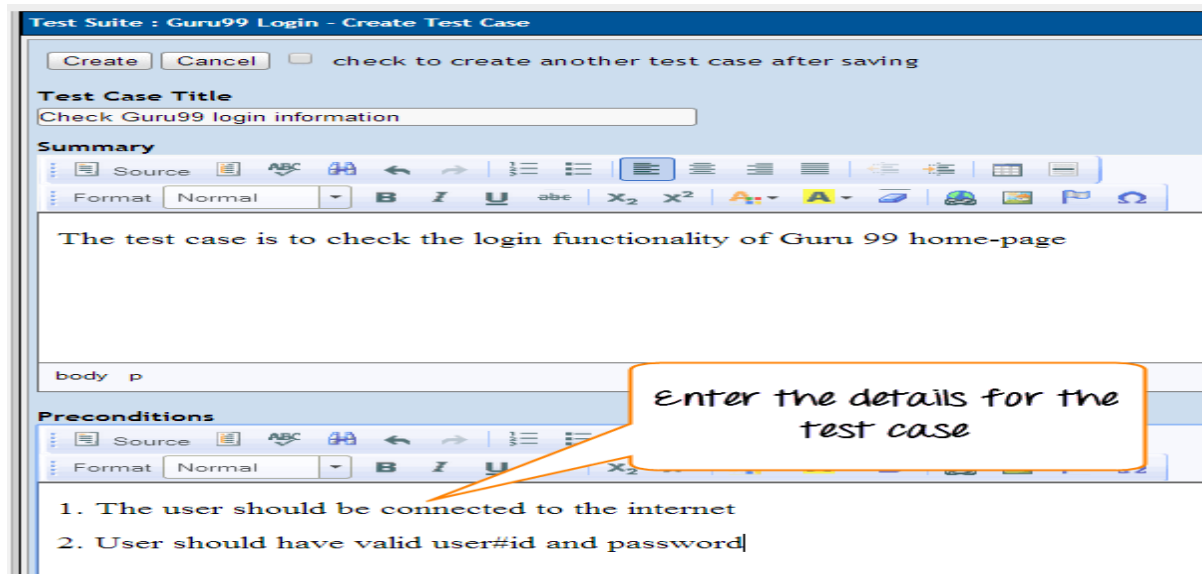
Step 2: Click on the setting icon in the right side panel. List of test case operations will be displayed on the right side panel



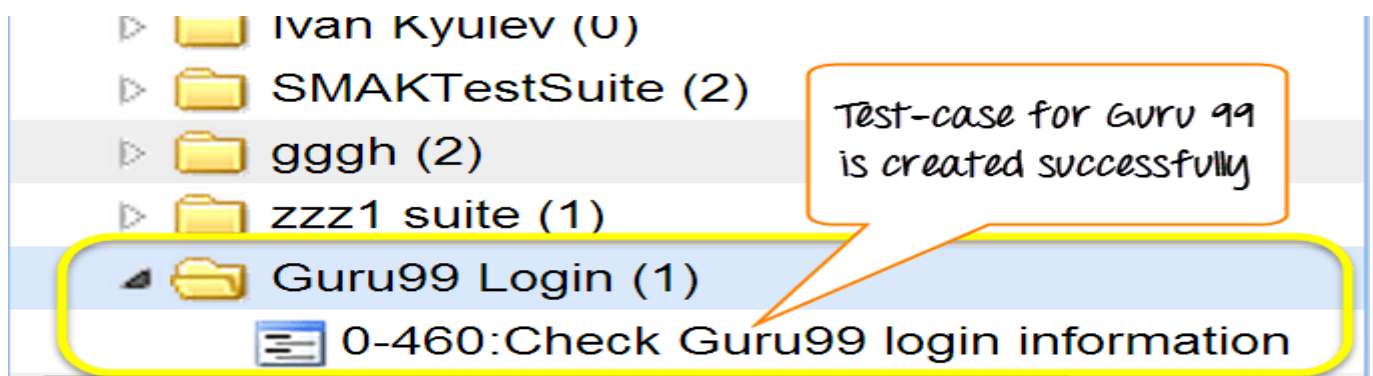
Step 3: New window will open, to create test cases click on create button in test-case operations



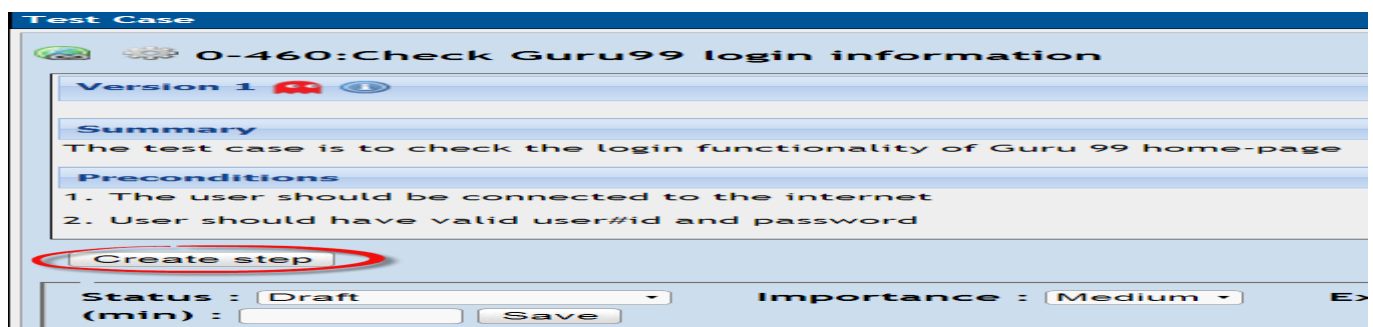
Step 4: Enter the details in the test case specification page



Step 5: After entering the details, click on "create" button to save the details. The test-case for Guru99 is created successfully

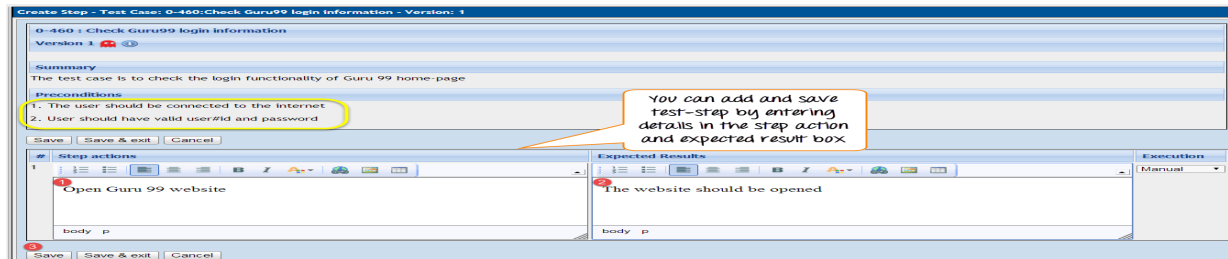


Step 6: Click on test-case from the folder as shown above, it will open a window. Click on "create steps" button in test case. It will open a test case step editor













Step 7) It will open another window on the same page, in that window you have to enter the following details

1. Enter the step-action for your test case
2. Enter the details about the step action
3. Click save it and add another step action OR click save and exit tab if there is no more test step to add



Step 8) Once you save and exit the test step, it will appear like this

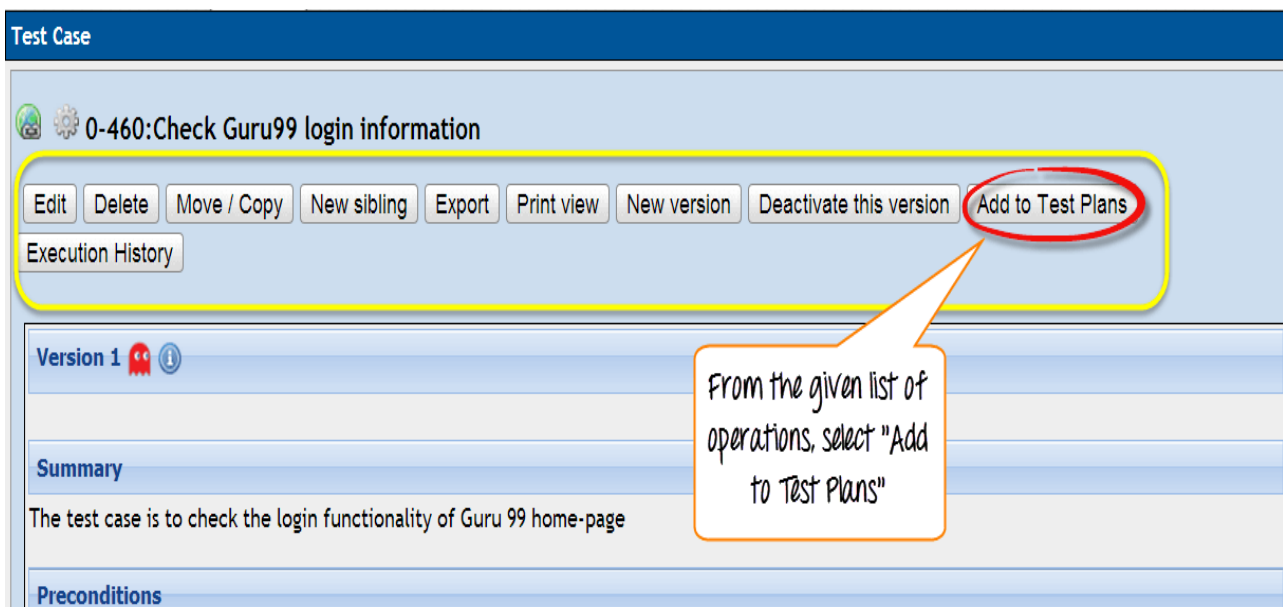
Version 1  				
Summary				
The test case is to check the login functionality of Guru 99 home-page				
Preconditions				
1. The user should be connected to the internet				
2. User should have valid user#id and password				
#	Step actions	Expected Results	Execution	
1	Open Guru 99 website	The website should be opened	Manual	 
2	Enter username in the username textbox	Text-box should accept the entered data	Manual	 
3	Enter password in the password text-box	Text-box should accept the entered data	Manual	 
4	Click on "sign in" button	Login should success and navigate to the home page	Manual	 

Assigning test case to test plan

For test case to get execute, it should be assign to test plan. Here we will see how we can assign a test-case to test plan.

Step 1) Click on the setting icon on the test panel. It will show the list of operations.

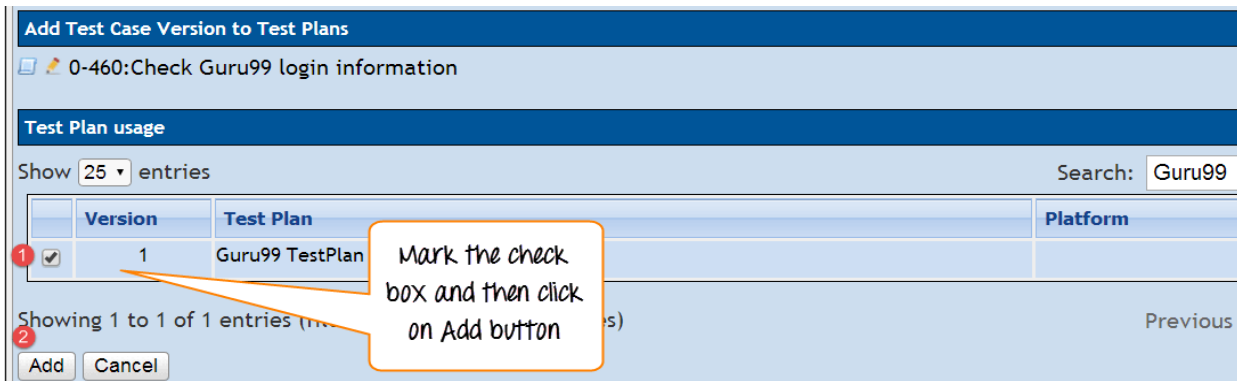
Step 2) Click on "Add to Test Plans"



Step 3)

New window will open, search your project "Guru99"

1. Mark the check box against your test plan
2. Click on add button



This will

add your test case to your Test Plan.

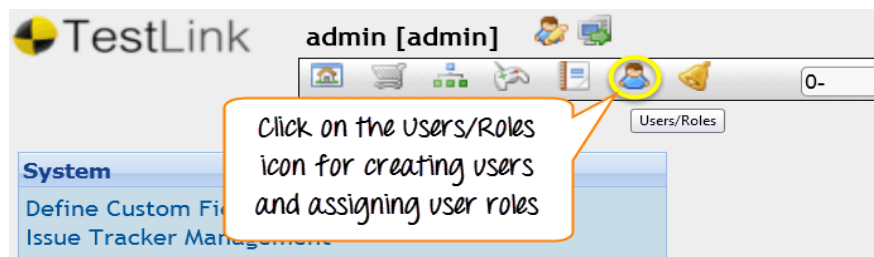
Creating Users and Assigning Roles in TestLink

Testlink provides User management and authorization features.

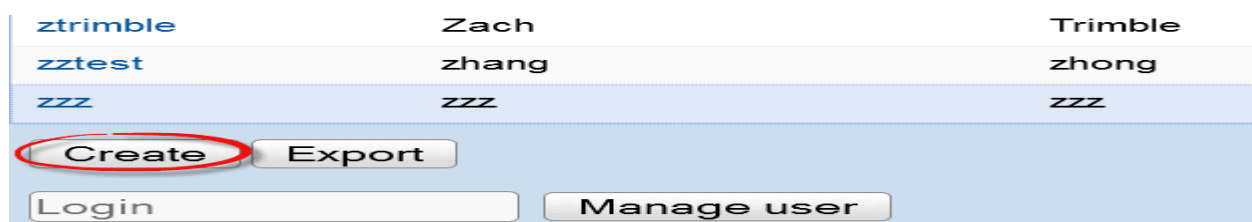
Below is list of default roles in Testlink and their rights -

Role	Test Cases	Test Metrics
Guest	View	View
Tester	Execute	View
Senior Tester	Edit & Execute	View
Leader & Admin	Edit & Execute	Edit & Execute

Step 1: From the Testlinks home-page, click on users/roles icon from the navigation bar



Step 2: Click Create



Step 3: Fill out all the users details and click the "Save" button

User Management - Account Settings

New User | View Users | View roles | Assign Test Project roles | Assign Test Plan roles

User details

Login	Jamesguru
First Name	Jamesguru
Last Name	Bryan
Password	•••••
Email	jamesguru@gmail.com
Role	Senior Test Engineer
Locale	English (wide/UK)
Authentication method	Default(DB)
Active	<input checked="" type="checkbox"/>

Fill out all the details about the user and click on save button

Save **Cancel**

Here in the list we can see the users have been created

jaofgc	João		joaofgc@gmail.com	leader
james	james		abc@163.com	leader
Jamesguru	Jamesguru	Bryan	jamesguru@gmail.com	Senior Test Engineer
jan	jan	sivaprakasam	jan.sivaprakasam@cancer.or...	leader

Here in the list we can see the users we have just created

Step 4: Allotting test project role to the user,

1. Click on "Assign Test Project Roles" tab
2. Choose the project name
3. Select the users role from the drop down

User Management - Assign roles

View Users | View roles | **Assign Test Project roles** | Assign Test Plan roles

Test Project **Guru 99 -** **Change**

Set roles to **Senior Test Engineer** **Do**

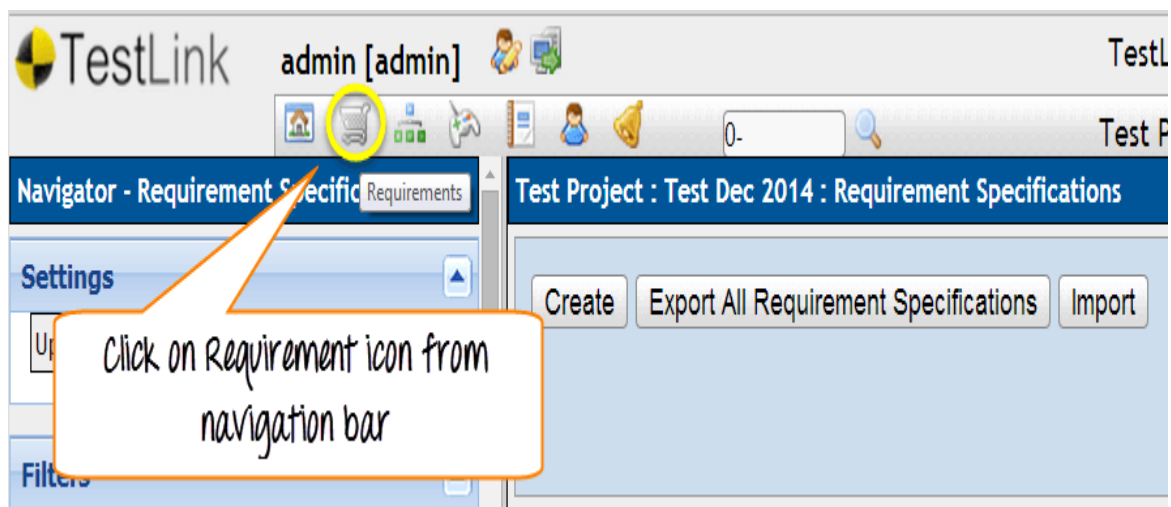
Show 25 entries

User	Test Project Role (Guru 99 -)
Jamesguru (Jamesguru Bryan)	<inherited> Senior Test Engineer

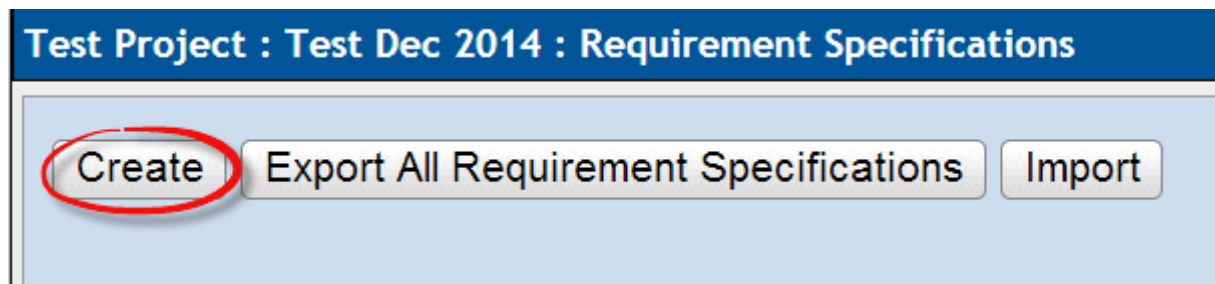
We assigned our test project "Guru99" to senior test engineer- Jamesguru

Writing Requirements:

Step 1: From the navigation bar select the "Requirements Link", it opens the Requirement page.



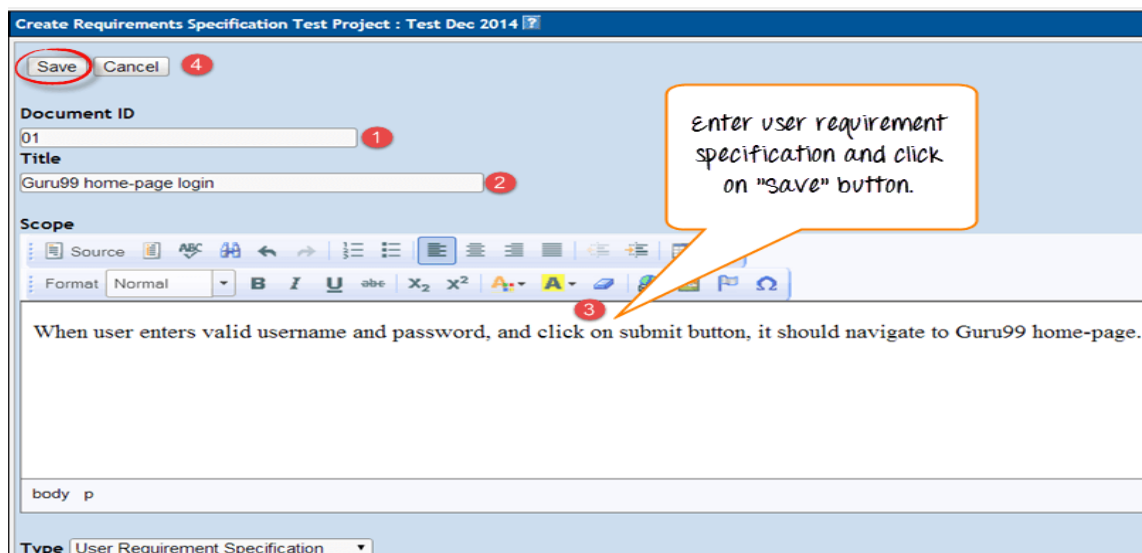
Step 2: From the requirement page, on the right side of the panel click on "create" button



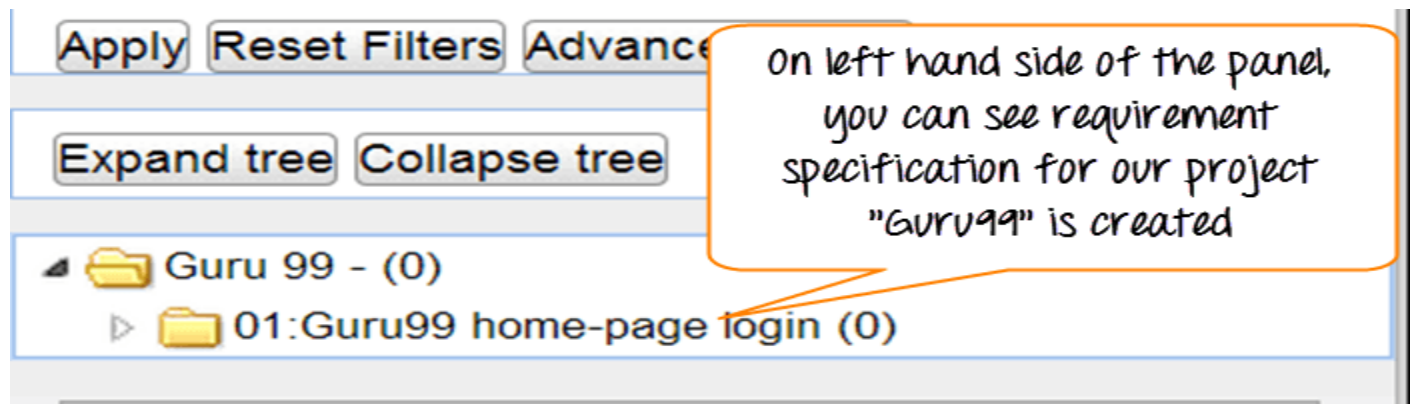
Step 3: A new window will open, enter all the details like

1. Document ID
2. Title name
3. Requirement description
4. And Click "Save" button

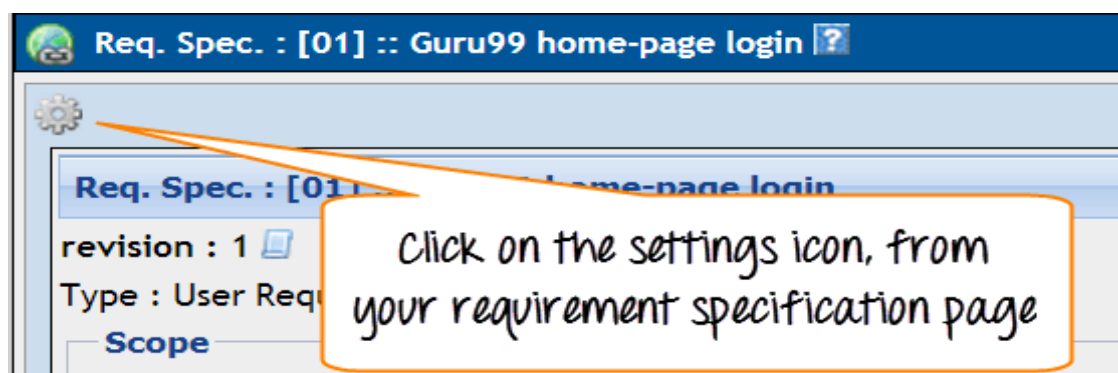
For the type, you can choose the option from the drop-down- here we chose **"User Requirement Specification"**



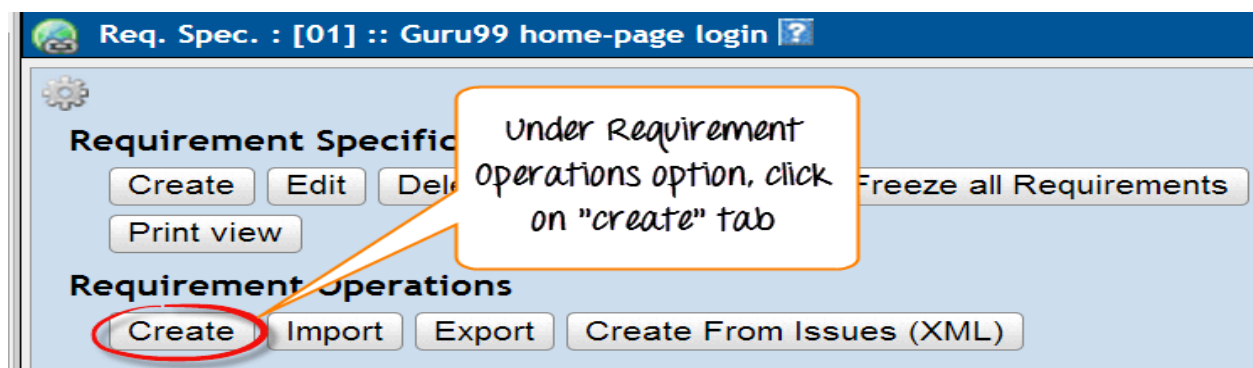
Step 4: It should create Requirement specification and displayed on the left side panel under project "Guru99".



Step 5: Select the setting button from requirements specification home-page. It will open another window.



Step 5: Click "Create" tab under Requirement Operations.



Step 6: Fill out all the specified details and click the "Save" button

1. Enter the document ID
2. Enter the title name
3. Enter the description
4. Enter the status-whether it's in draft, rework, review, not testable, etc. Here we chose valid
5. Enter the type – user interface, non-functional, informational, feature, etc. Here we chose use case
6. Enter the number of test cases needed
7. Enter "Save" button at the end

Save Cancel 7

☐ check to create another requirement after saving

Document ID
02 1

Title
Guru99 home-page login 2

Scope

Format Normal B I U abc X₂ X² A₁ A₂

Password text-box should accept the entered data in encrypted format. 3

When user login with valid user name and password it should navigate to the Guru99 home-page.

body p

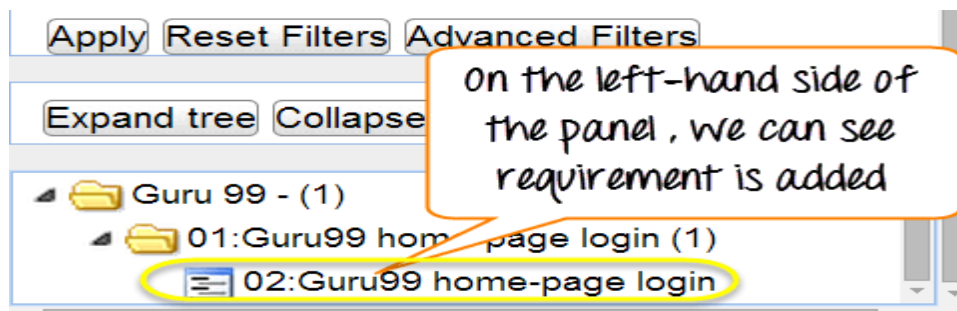
Status Valid 4

Type Use Case 5

Number of test cases needed 1 6

Note: To add more requirements you can mark the check-box and click save button

On the left side of the panel, we can see that requirement is added.



Assigning requirement to test-cases

In Testlink, Requirement can be connected to test cases. It is very crucial feature in order to track test coverage based on requirements. In test reports, you can verify which **requirements are not covered and act on them to append** in test suites for maximum test coverage

Step 1: From test specification section open any single test case and click on requirement icon

Test Case

GU-1:Check Guru99 login information

Version 1

Summary

Preconditions

1. The user should be connected to the internet

2. User should have valid user#id and password

#	Step actions	Expected Results
1	Open Guru 99 website	The website should be opened
2	Enter username in the username textbox	Text-box should accept the entered data
3	Enter password in the password text-box	Text-box should accept the entered data
4	Click on "sign in" button	Login should success and navigate to the home page

Create step Resequence Steps

Status : Draft Import Requirements

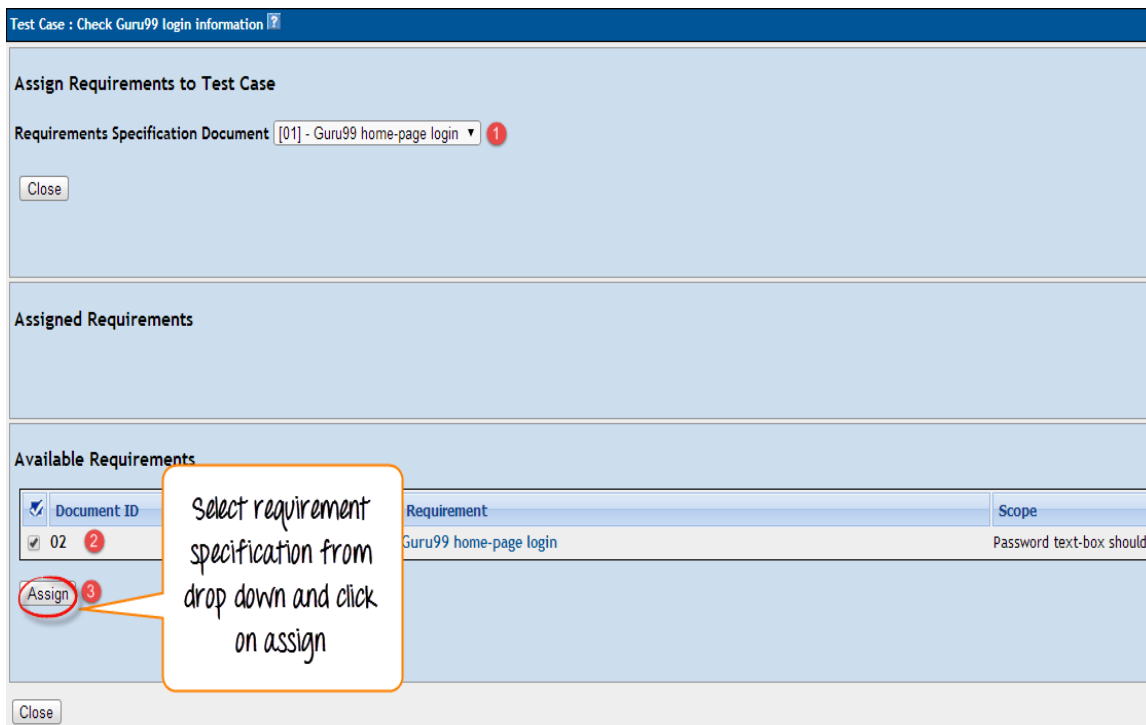
Keywords: None

Requirements : None

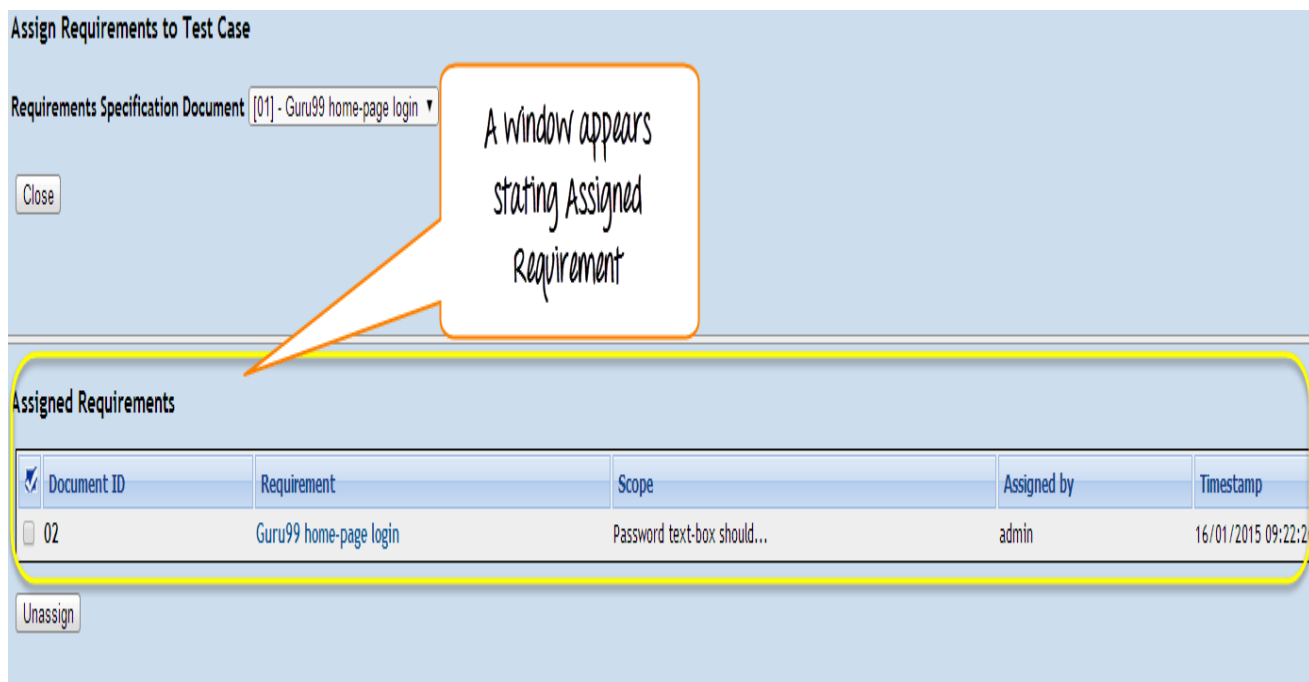
From the test specification section, click on requirements icon

Step 2: To assign requirements specification to test case you have to follow the following steps

1. Scroll the drop down box to select the requirements specification
2. Mark the requirement check box
3. Click on "assign" tab



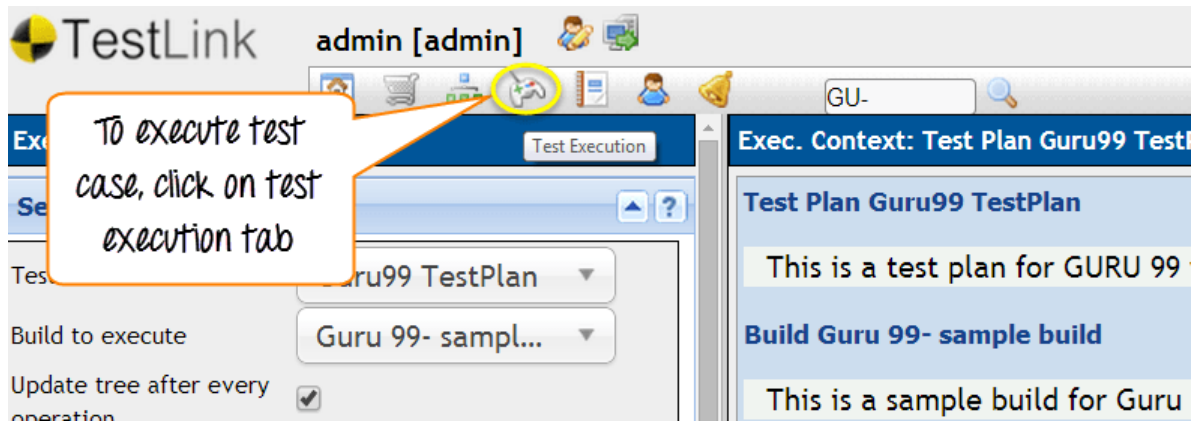
After clicking on "assign" tab, a window will appear stating "Assigned Requirement."



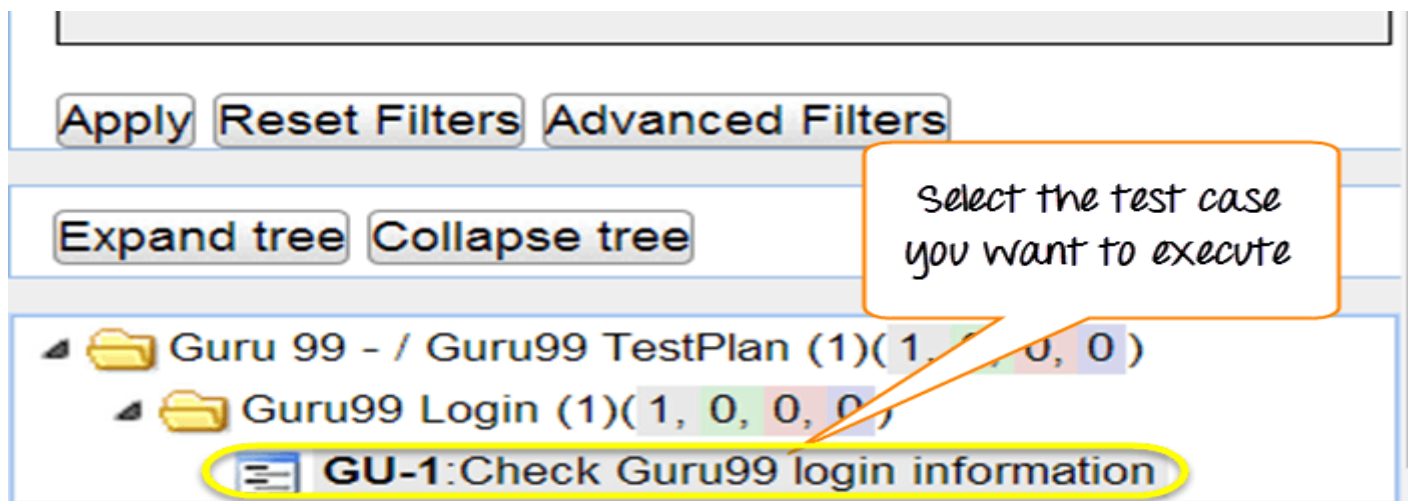
Executing a test case

In TestLink, we can run a test case and change execution status of a test case. Status of a test-case **can be set** to "blocked" "Passed", or "failed". Initially, **it will be in "not run"** status but once you **have** updated it, it cannot be altered to "not run" status again.

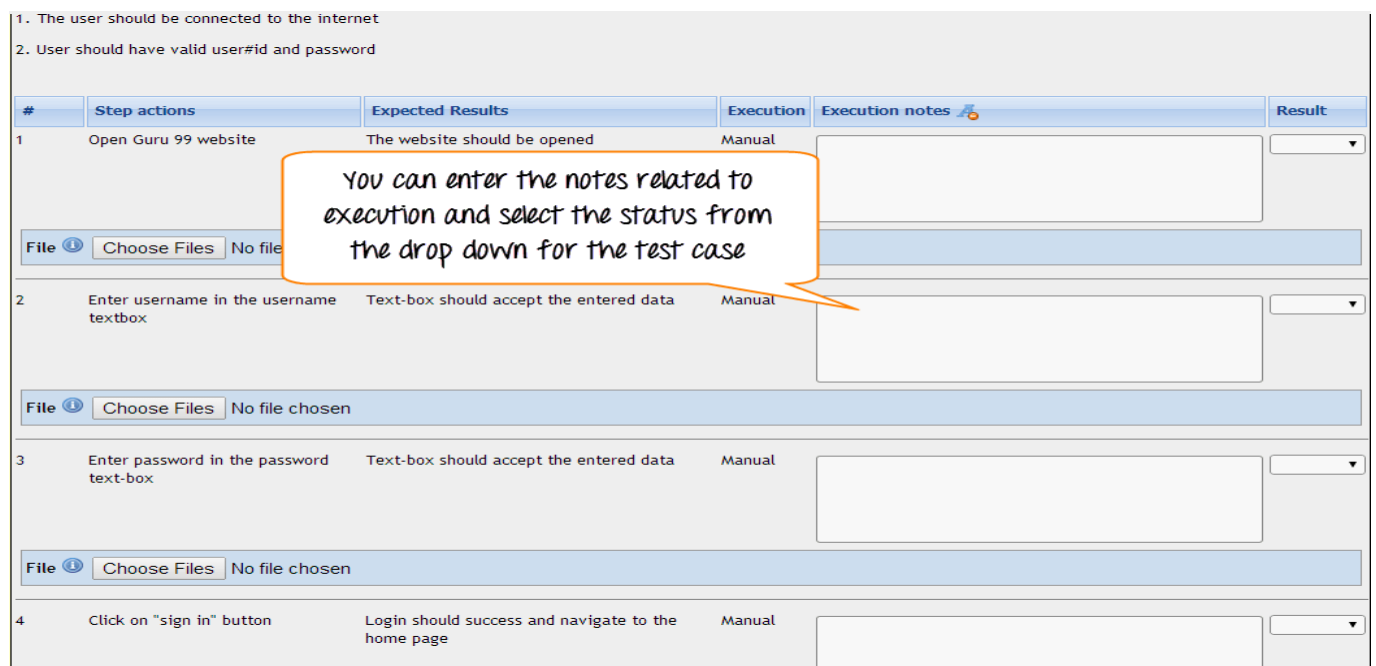
Step 1: From the navigation bar click on the "Test Execution" link. It will direct you to the Test Execution Panel.



Step 2: Pick the Test case you want to run from the left side panel



Step 3: Once you have selected the test cases, it will open a window.



Step 4: Follow the following steps

1. Enter the notes related to test case executed
2. Select its status

The screenshot shows a test case execution form. At the top, there is a 'File' section with a 'Choose Files' button and a 'No file chosen' status. Below this, there is a table with columns for 'Step', 'Action', 'Expected Result', 'Test Type', and 'Notes'. The first row is numbered '2' and contains the following data: 'Enter username in the username textbox', 'Text-box should accept the entered data', 'Manual', and a text area with the note 'When user name is entered in the username, it accepts the entered data'. To the right of the text area is a dropdown menu showing 'Passed'. There are two red circles with numbers '1' and '2' around the dropdown. A speech bubble points to the dropdown with the text 'Note about the test case and its status is selected from drop-down'.

Step 5: On the same page, you have to fill similar detail about the execution of test-case. Fill the details, select the status and then click on "save execution".

The screenshot shows the 'Test Results on Build Guru 99- sample build' page. It has a navigation bar with 'Print', 'Show complete execution history', and 'Import Results (XML)'. Below the navigation bar, there is a tree view showing the test plan structure: 'Test Plan Guru99 TestPlan', 'Build Guru 99- sample build', and 'Test Suite : Guru99 Login/'. A speech bubble points to the 'Test Suite : Guru99 Login/' item with the text 'A window will appear showing latest execution of your test case'. Below the tree view, there is a section for 'Latest execution (any build)' with a green bar showing 'Date : 16/01/2015 10:35:06 - Tested by : admin - Build : Guru 99- sample build - Status : Passed'. Below this, there is a section for 'Latest execution (current build) - Build : Guru 99- sample build' with a table showing the execution details.

Date	Build	Tested by	Status	Exec (min)	Version	Run mode
16/01/2015 10:35:06	Guru 99- sample build	admin	Passed		1	

Below the table, there is a 'Notes' section with a text area containing the note 'Test executed successfully in Firefox 12, in windows 8 platform'.

Generating Test Reports

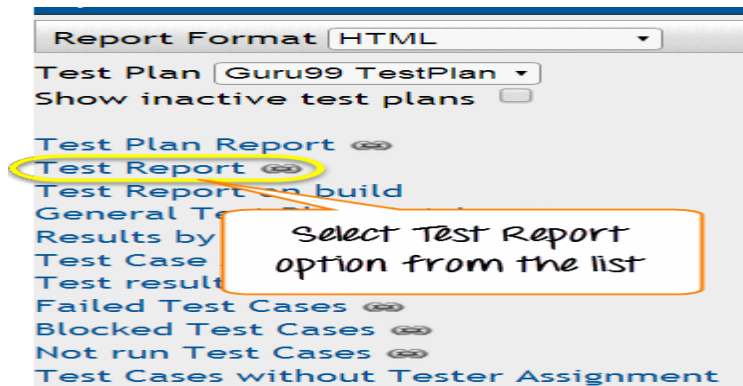
Test link supports various test report formats like

- HTML
- MS Word
- MS excel
- OpenOffice Writer
- OpenOffice calc

Step 1: From the navigation bar, click on Test Reports option

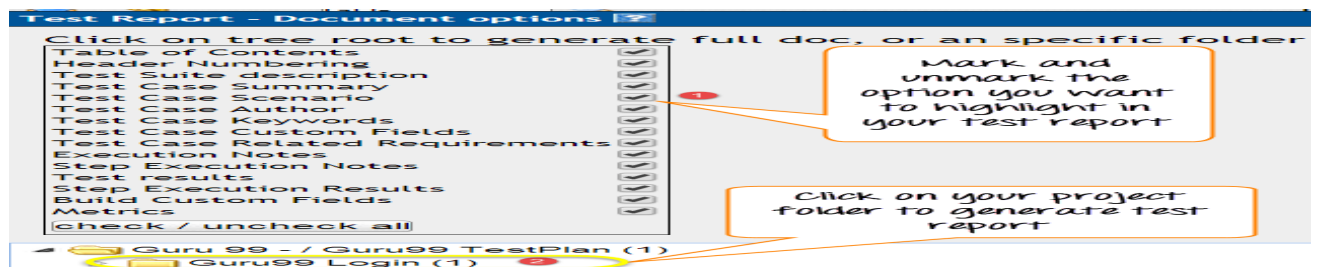


Step 2: From the left side panel, select "Test Report" link



Step 3: To generate a report follow the following steps

1. Mark and unmark the option you want to highlight in your test report
2. click on your project folder



The test report will look like this

1.1. Test Suite : Guru99 Login

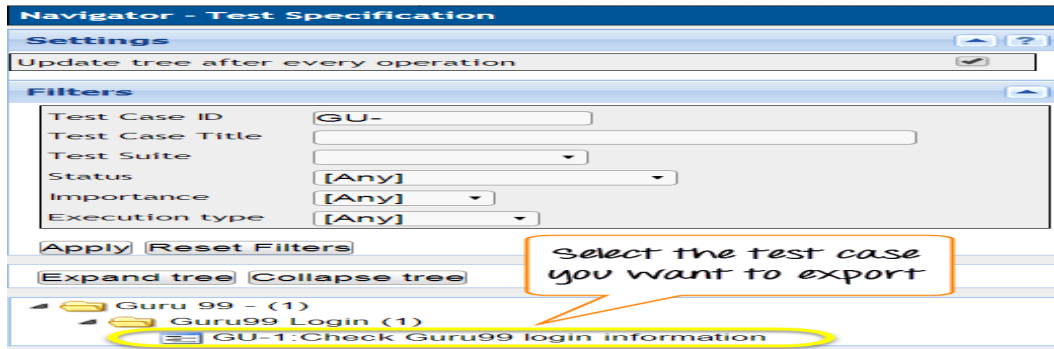
The test suite contains test cases related to test Guru 99 login functionality


Test Case GU-1: Check Guru99 login information				
Author: admin				
Preconditions:				
1. The user should be connected to the internet				
2. User should have valid user#id and password				
#	Step actions:	Expected Results:	Execution notes:	Execution Status:
1	Open Guru 99 website	The website should be opened		
2	Enter username in the username textbox	Text-box should accept the entered data	When user name is entered in the username, it accepts the entered data	Passed
3	Enter password in the password text-box	Text-box should accept the entered data		
4	Click on "sign in" button	Login should success and navigate to the home page		
Execution type: Manual				

Export Test case/ Test Suite

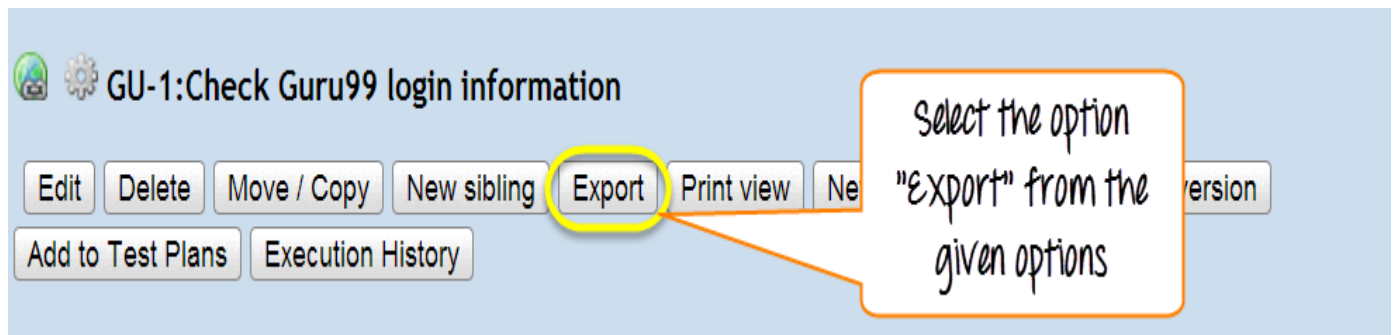
Testlink provides the features to export test projects/test suites in your Testlink and then you can import them into another Testlink project on different server or system. In order to do that you have to follow the following step

Step 1: Choose the test case you want to export in the Test specification page

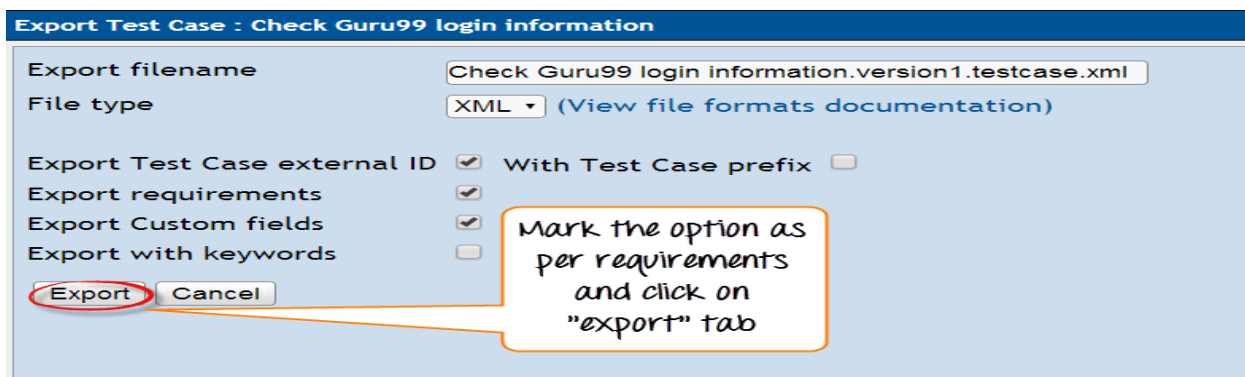


Step 2: Now on the right-hand side of the panel click on the  setting icon, it will display all the operations that can be performed on the test case.

Step 3: Click the "export" button



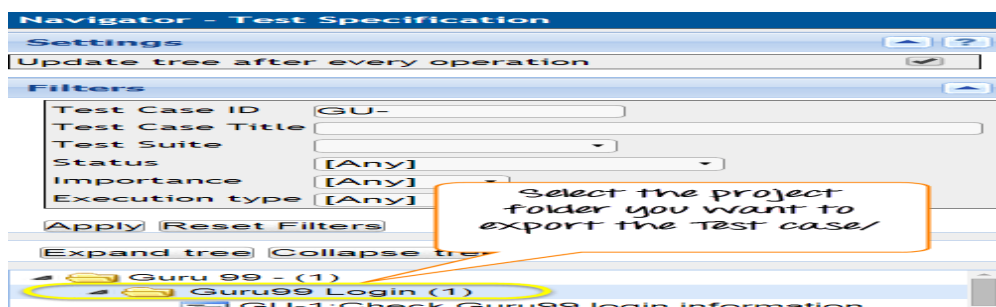
Step 4: It will open another window, mark the option as per requirement and click on the export tab




Following XML is generated

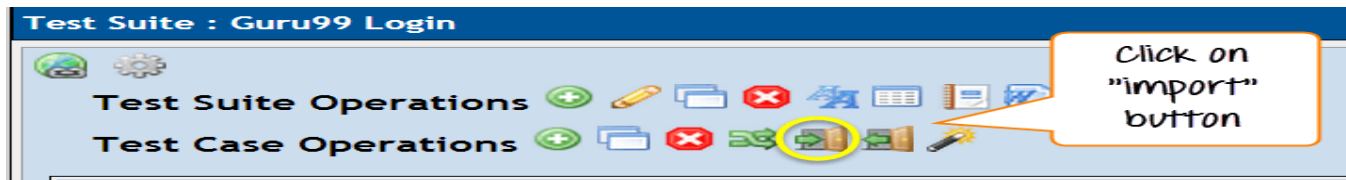
Importing Test case/ Test suite

Step 1: Select the Test suite folder inside which you want to import the test case



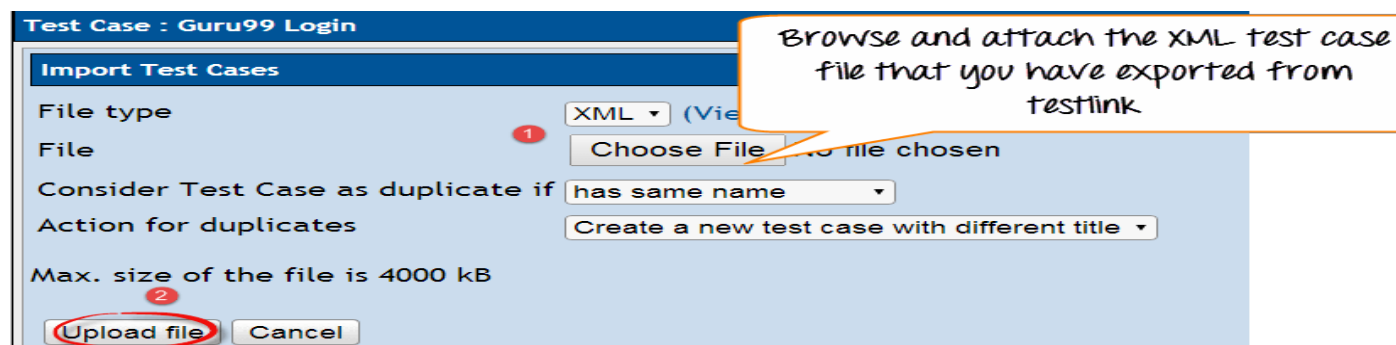
Step 2: Click on the setting icon  on the right hand-side of the panel, it will display all the operations that can be executed on the test suite/test case

Step 3: Click on the import button in the test case operations list as

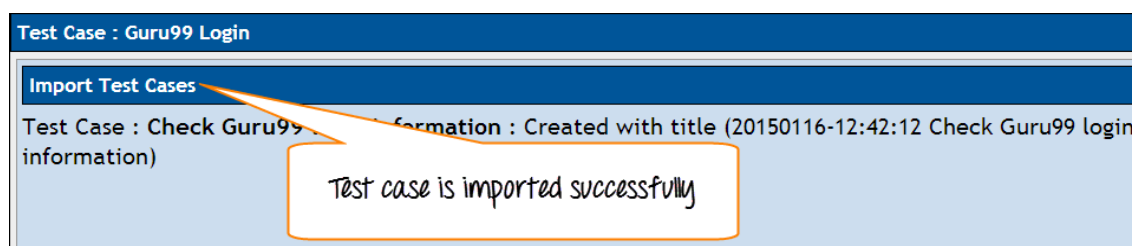


Step 4: Browse and attach the xml test case file that you have exported from test link and click on upload button.

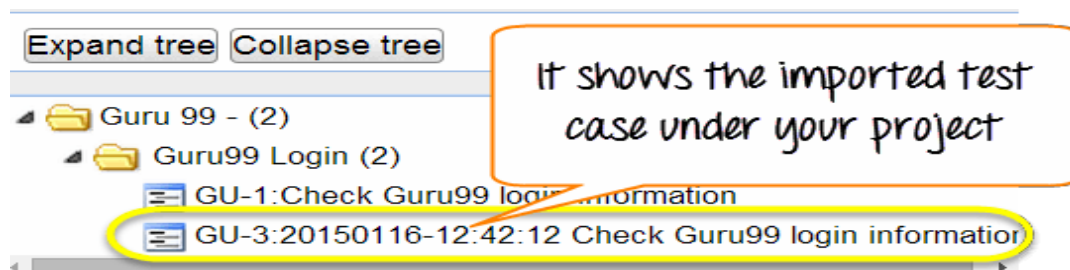
1. Use the browse option to attach the XML test case file that you have exported from testlink
2. Click on upload file



When you upload a file, it will open window stating import test cases



Step 5: Test case will be uploaded and displayed on the right-hand side of the panel



11 b) Test Facebook Manually (beyond the syllabus)

11 b) Test Facebook Manually (beyond the syllabus)

Week 12: Take a mini project and execute it during SDLC create the various UML diagrams required designing and all testing documents like test plan,TCD etc.,