

Expt No: 03

Date : 07/04/2023

## Modeling UML Use Case Diagrams

### Aim

To draw use case diagrams for the given problems.

### Procedure to draw use case Diagram

1) Identify an actor, and write its name in the left textbox of "Table #1", click the adjacent 'add'

buttons to add this actor. Repeat this for all the possible actors.

2) Identify a use case, and write its name in the left box of "Table #2". click the adjacent 'add' button to add this use case.

Repeat this for all the possible use cases.

3) If you want to delete any actor or use case, go to the "Table #4", remove the actor or use case.

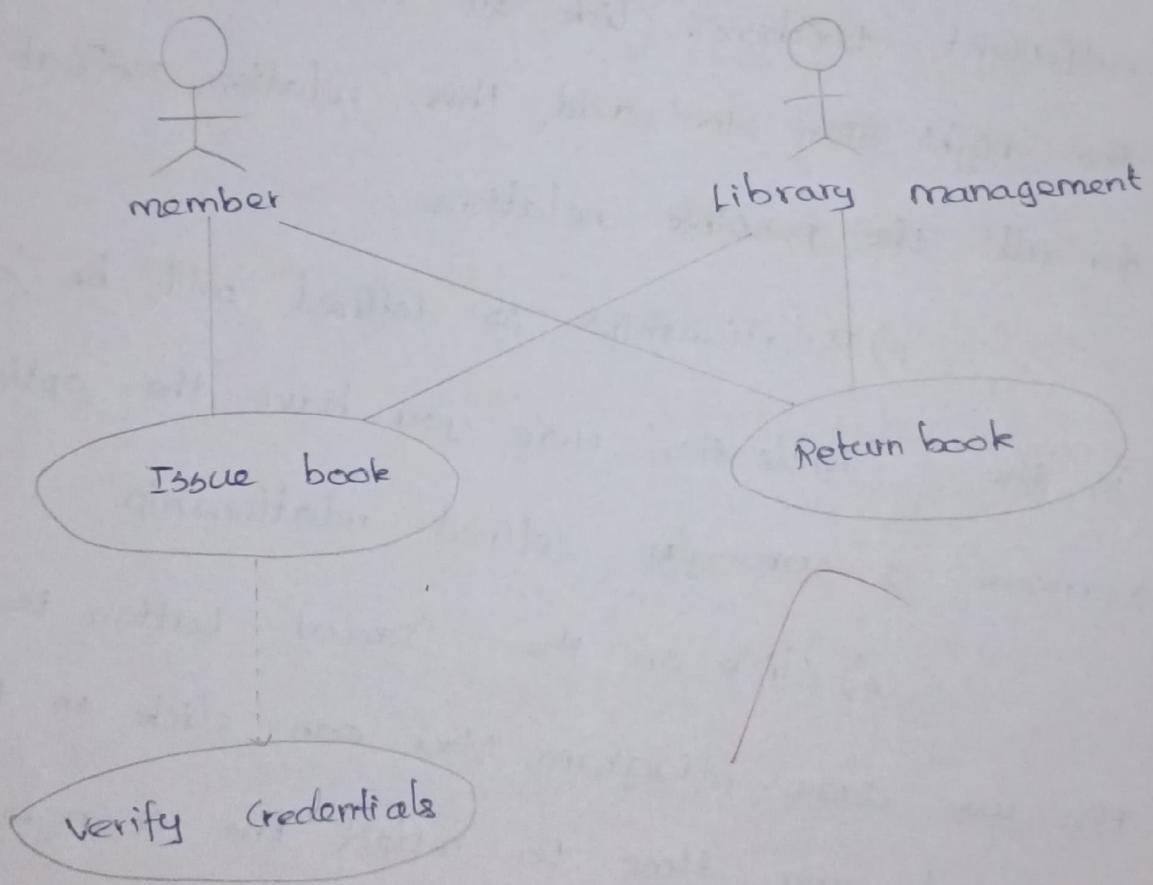
4) 'Table #3' lets you define relationships between any two components. Select the first actor/usecase from the first drop down list, the second from the third drop down list. Select one relationship from the second dropdown list. If you want to put any label to the relation, write the text for the label to the adjacent textbox. Click on the 'Add' button at the right side to add this relation. Repeat this for all the possible relations.

5) Relationships so defined will be displayed in the 'Table #5'. Here you have the option to remove a wrongly defined relationship.

6) Click on the "Draw" button to draw the use case diagram. You can click on this button at any time to reflect the changes that you have made to the actors, usecases, or relationships. The usecase diagram will be displayed at the bottom of the page.

7) To view a sample solution for the exercise, click on the 'Submit' button and then

## Use - Case Diagram



# USE CASE DIAGRAM:

## ➤ Tables and Diagram:

Table #1: Add actors

Actor	Add
<input type="text"/>	<input type="button" value="Add"/>

Table #2: Add use cases

Use Case	Add
<input type="text"/>	<input type="button" value="Add"/>

Table #3: Add relationship

From Actor / Use Case	Relationship	To Actor / Use Case	Label	Add
Issue Book	Include	Verify Credentials		<input type="button" value="Add"/>

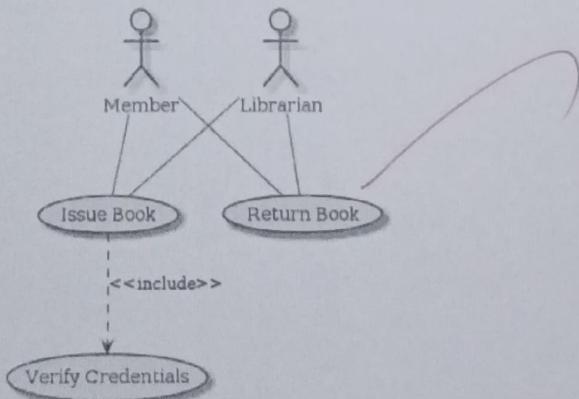
Table #4: List of actors and use cases

Actor	Use Case
<ul style="list-style-type: none"><li>Member</li><li>Librarian</li></ul>	<ul style="list-style-type: none"><li>Issue Book</li><li>Return Book</li><li>Verify Credentials</li></ul>

Table #5: List of relationships

Actor / Use Case	Relationship Type	Actor / Use Case	Label	Remove
Member	Association	Issue Book	<input type="button" value="-"/>	
Member	Association	Return Book	<input type="button" value="-"/>	
Librarian	Association	Return Book	<input type="button" value="-"/>	
Librarian	Association	Issue Book	<input type="button" value="-"/>	
Issue Book	Include	Verify Credentials	<input type="button" value="-"/>	

 Draw



Result

on the 'view solution' button

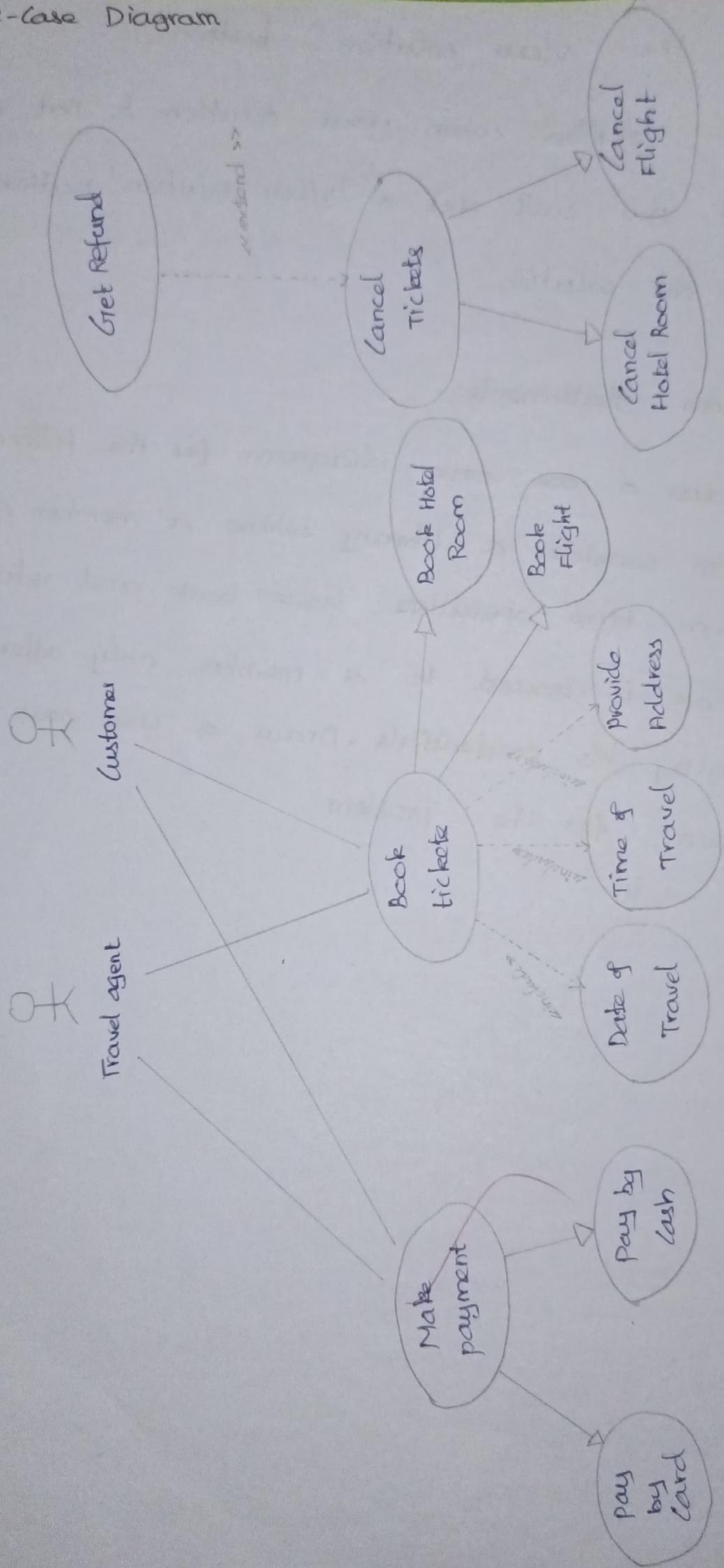
-19

8) Also when your solution is not exactly right, you will get a 'view solution' button to view the solution.

### Problem statements

i) Draw a use case diagram for the following problem consider a library where a member can perform two operations; issue book and return it. A book is issued to a member only after verifying his credentials. Draw a use case diagram for the problem

# Use-Case Diagram



# USE CASE DIAGRAM:

## ➤ Tables:

Table #1: Add actors

Actor	Add
<input type="text"/>	<input type="button" value="+ Add"/>

Table #2: Add use cases

Use Case	Add
<input type="text"/>	<input type="button" value="+ Add"/>

Table #3: Add relationship

From Actor / Use Case	Relationship	To Actor / Use Case	Label	Add
Cancel Booking	Extend	Cancel Flight	<input type="text"/>	<input type="button" value="+ Add"/>

Table #4: List of actors and use cases

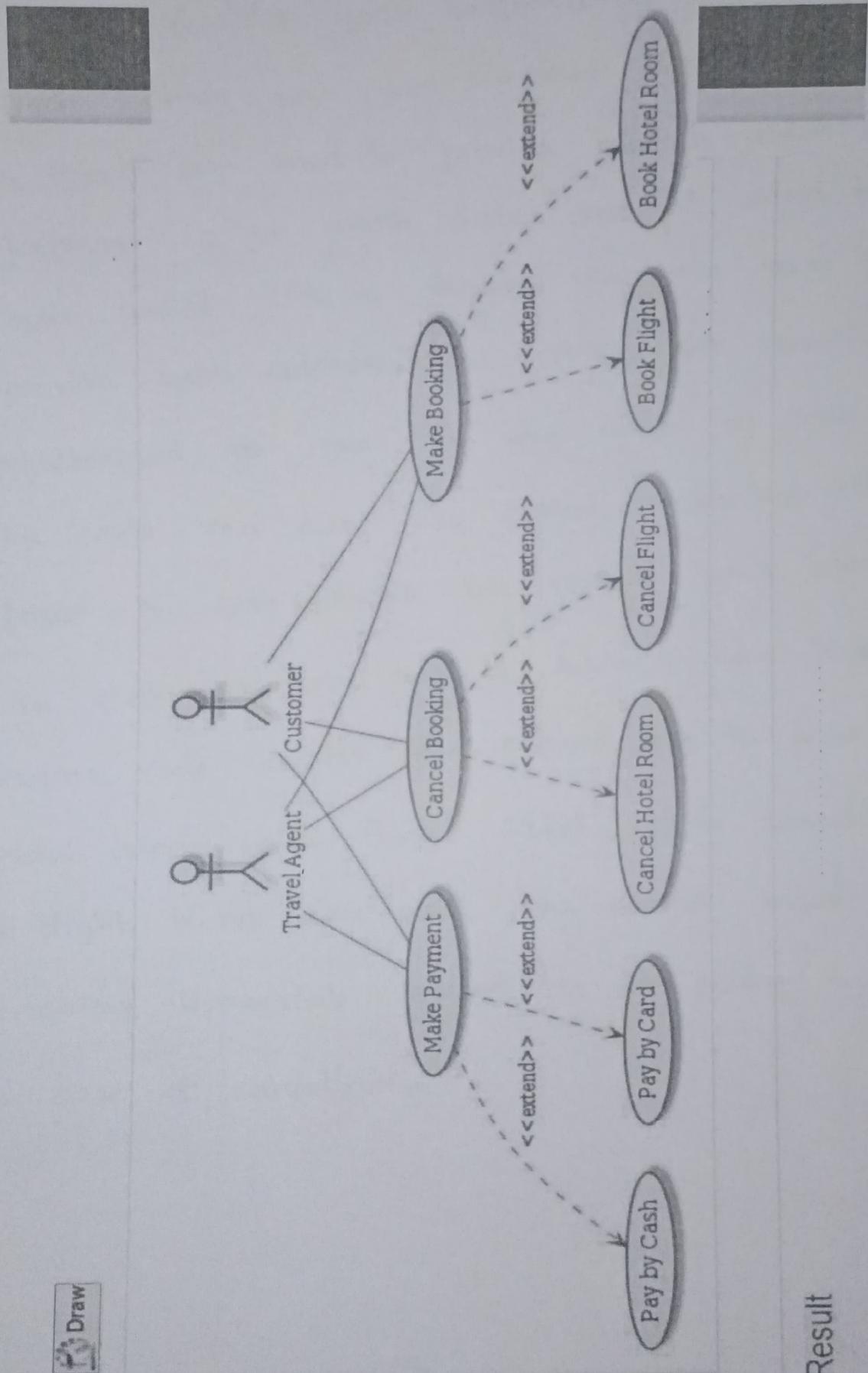
Actor	Use Case
<ul style="list-style-type: none"> <li>Travel Agent</li> <li>Customer</li> </ul>	<ul style="list-style-type: none"> <li>Make Payment</li> <li>Pay by Cash</li> <li>Pay by Card</li> <li>Make Booking</li> <li>Book Flight</li> <li>Book Hotel Room</li> <li>Cancel Booking</li> <li>Cancel Hotel Room</li> <li>Cancel Flight</li> </ul>

Table #5: List of relationships

Actor / Use Case	Relationship Type	Actor / Use Case	Label	Remove
Travel Agent	Association	Make Payment	<input type="button" value="Delete"/>	
Travel Agent	Association	Make Booking	<input type="button" value="Delete"/>	
Travel Agent	Association	Cancel Booking	<input type="button" value="Delete"/>	
Customer	Association	Cancel Booking	<input type="button" value="Delete"/>	
Customer	Association	Make Payment	<input type="button" value="Delete"/>	
Customer	Association	Make Booking	<input type="button" value="Delete"/>	
Make Payment	Extend	Pay by Cash	<input type="button" value="Delete"/>	
Make Payment	Extend	Pay by Card	<input type="button" value="Delete"/>	
Make Booking	Extend	Book Flight	<input type="button" value="Delete"/>	
Make Booking	Extend	Book Hotel Room	<input type="button" value="Delete"/>	
Cancel Booking	Extend	Cancel Hotel Room	<input type="button" value="Delete"/>	
Cancel Booking	Extend	Cancel Flight	<input type="button" value="Delete"/>	



## ➤ Use Case Diagram:



2) Draw a use case diagram for the  
following problem

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

	Total marks	obtained marks
Preparation and Viva	10	08
Observation	10	08
Design and Implementation	10	08
Output	10	07
Record	10	08
Total	50	39

~~Result~~

thus the use case diagrams have been drawn for the given problem statements in the virtual Lab.

Date: 07/04/2023

## Modeling UML Class Diagrams and Sequence Diagrams

### Aim

To draw class diagrams and sequence diagrams for the given problems in the virtual lab.

### Procedure to draw class diagram

- 1) Read the problem statement carefully and choose the workspace specified for class diagram
- 2) Identify a class and write its name in the left textbox of 'Table #1'. click the adjacent 'Add' button to add this class. Repeat this for all the possible classes.

- 3) You may add attributes to a class from 'Table #2'. select a class from the left drop down list of 'Table #2'. Identify an attribute and write its name in the next textbox of 'Table #2' and select the attribute type from the

the next drop down list. click the adjacent 'Add' button to add this attribute. Repeat this for all attributes of all classes.

4) You may add operations to a class from 'Table #3'. Select a class from the left drop down list of 'Table #3'. Identify an operation and write its name in the next textbox of 'Table #3' and select its return type from the next drop down list. click the adjacent 'Add' button to add this operation. Repeat this for all operations of all classes.

5) The list of attributes and operations along with their respective classes are shown in 'Table #6'. If you want to delete any class, attribute or operation, you may delete it from here.

6) 'Table #4' lets you define inheritance relationship between any two classes. Select one sub-class from the left drop down list, select its super-class from the next drop down list. click the adjacent 'Add' button to add this inheritance relation. Repeat this for all inheritance

relationships that you find for the system.

93

7) If you want to delete any inheritance relation, go to the 'Table #7', remove the sub-class or super-class.

8) 'Table #5' lets you define association relationships between any two classes. Select the first class name from the first drop down list, the second from the fifth (right most) drop down list. Select one association type from the second dropdown list. If you want to put any name to the association, write the text for the name to the second textbox. If you want to give role name to any one or both end of association then you write the name in first or third or in both textboxes. Click on the 'Add' button at the right side to add this association. Repeat this for all the possible associations.

9) Associations so defined will be displayed in the 'Table #8'. Here you have the option to remove a wrongly defined associations.

10) Click on the 'Draw' button to draw the

class diagram you can click on this button at any time to reflect the changes that you have made to the classes or relationships. The class diagram will be displayed at the bottom of the page.

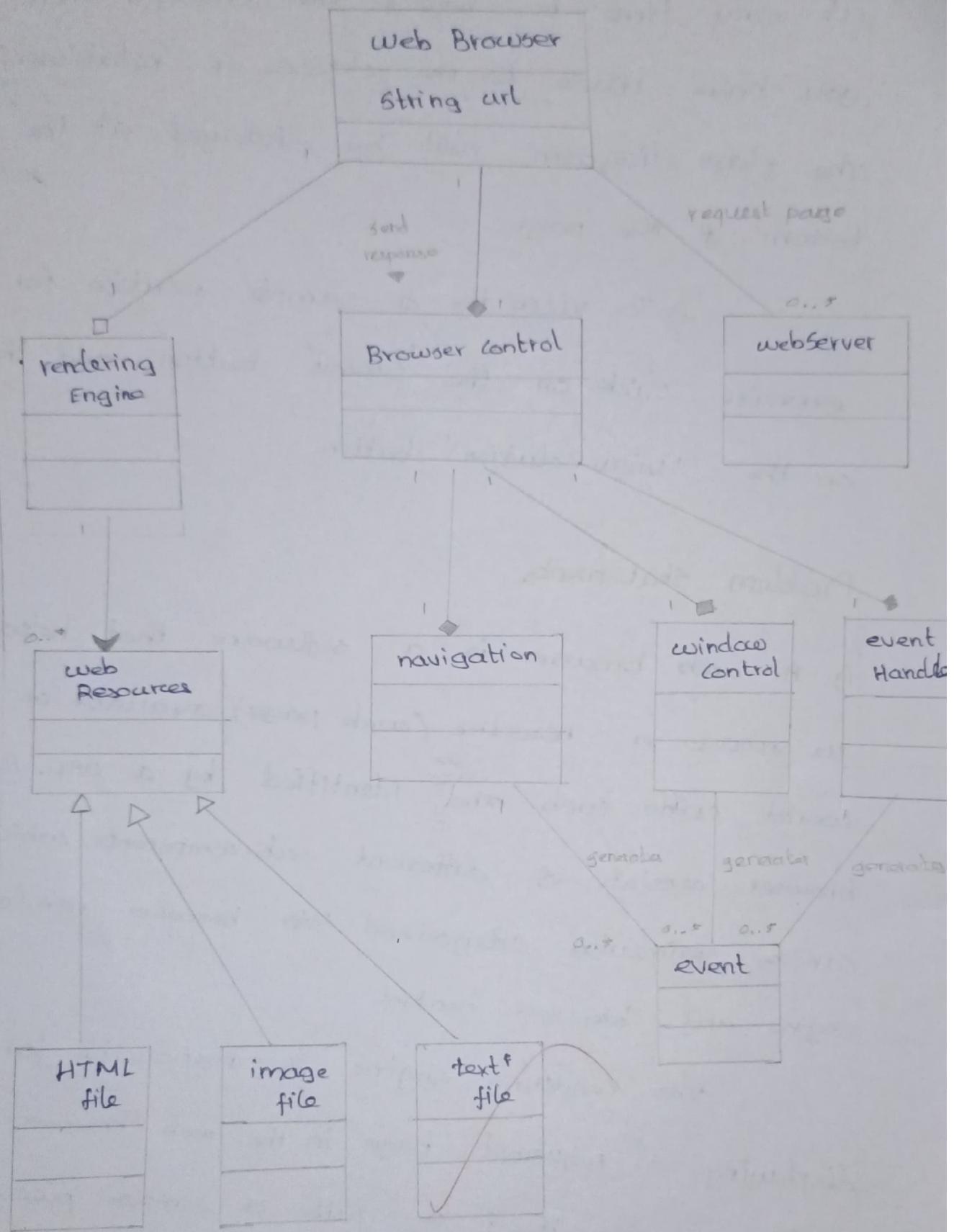
- ii) To view the a sample solution, for the exercise, click on the 'submit' button, and then on the 'view solution' button.

### Problem statements

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

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

# Class Diagram



# CLASS DIAGRAM:

## ➤ Tables:

Table #6: Classes identified for solution

Class	Attributes	Operations
webBrowser	String url	
browserRenderingEngine		
browserControl		
webResources		
HTML		
imagefile		
textfile		
event		
navigation		
windowControl		
eventHandler		
webServer		

Table #7: Generalization identified for solution

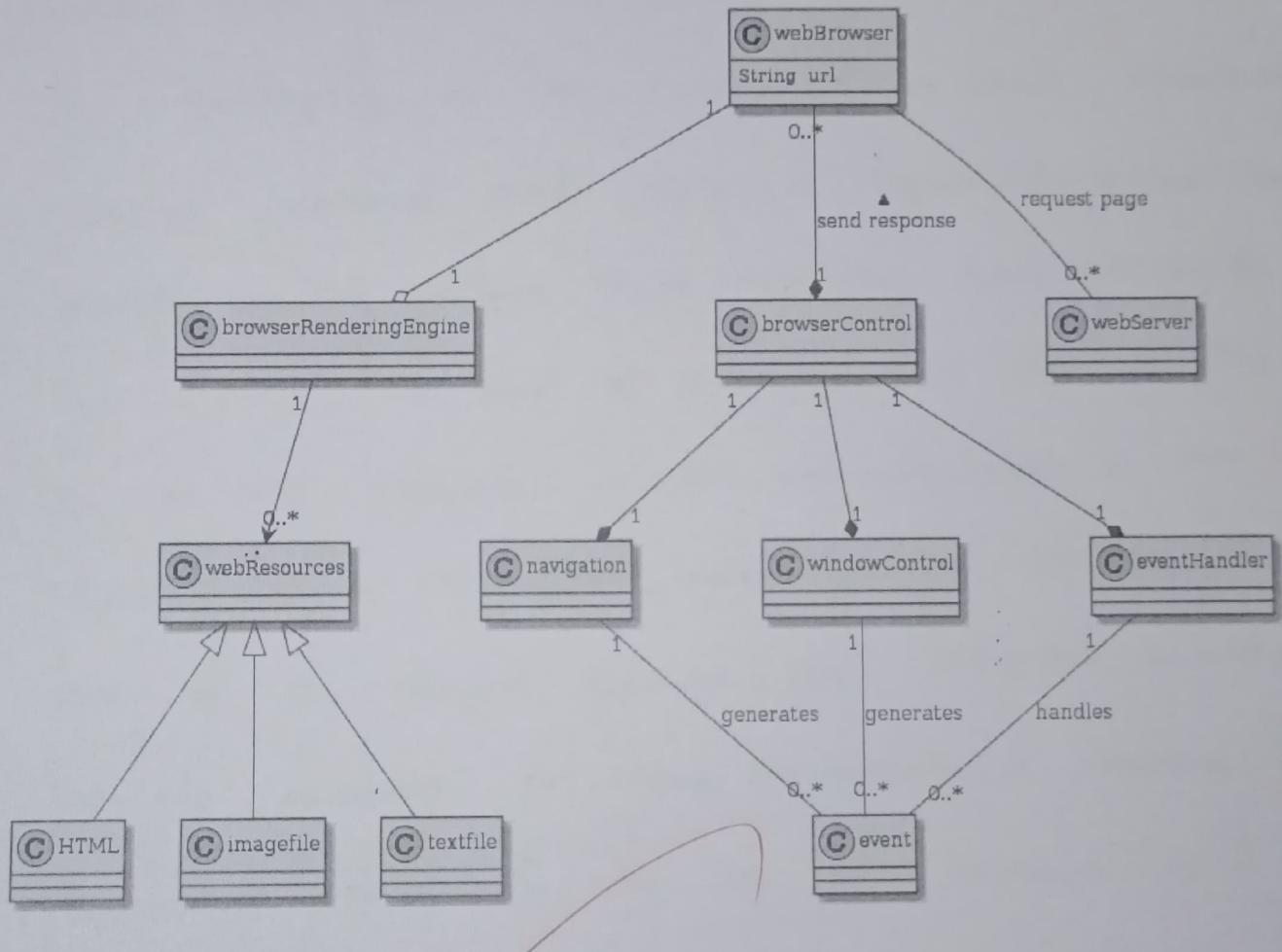
Subclass	Superclass
HTML	webResources
imagefile	webResources
textfile	webResources

Table #8: Relations between classes identified for solution

Class	Multiplicity	Relation Name	Relation Type	Multiplicity	Class	Direction
webBrowser	Not specified	request page	Simple association	0..*	webServer	Not specified
browserRenderingEngine	1		Navigation association (to)	0..*	webResources	TO
navigation	1	generates	Simple association	0..*	event	Not specified
windowControl	1	generates	Simple association	0..*	event	Not specified
eventHandler	1	handles	Simple association	0..*	event	Not specified
browserControl	1		Composition (of)	1	navigation	FROM
browserControl	1		Composition (of)	1	windowControl	FROM
browserControl	1		Composition (of)	1	eventHandler	FROM
webBrowser	1		Aggregation (of)	1	browserRenderingEngine	FROM
webBrowser	1	send response	Composition (of)	1	browserControl	TO



## ➤ Class Diagram:



limited to displaying a text and images, but can provide access to audio and video components also.

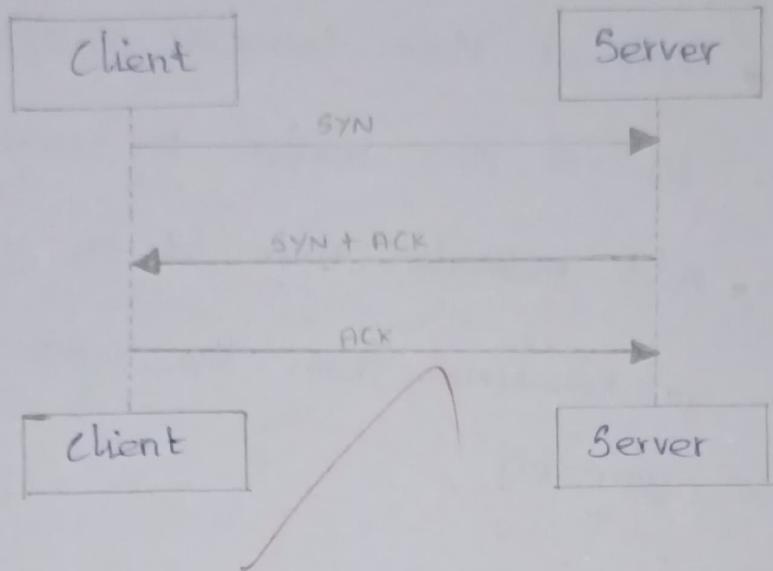
The web browser control, too, consists of several sub-components including navigation, window control, event handlers, page display. The navigation control aids users to request for web pages (resources) by specifying a URL, navigate to other resources through internal and external hyperlinks, move across pages visited earlier. Event handlers are responsible to identify the kind of activity that user is trying to do, and perform it. For example, when a user clicks on a hyperlink, event handlers identify the URL of the target resource, and delegates loading of the resource to other components. A resource that has been retrieved by the web browser is then displayed in its page display area.

Window control, in association with the rendering engine, helps in controlling various aspects of page display like changing font-size, resolution, and so on, apart from resizing or closing the window.

Represent the above problem with a class diagram  
In particular,

- \* Represent the Individual classes
- \* Represent how browser rendering engine, and browser control are related to web browser class.
- \* Add methods that let a web browser retrieve a resource from the web server, and convince yourself.

# Sequence Diagram



# SEQUENCE DIAGRAM:

## ➤ Tables and Diagram:

**Class Diagram**

**Sequence Diagram**

Table #1: Add object

Object Name (Role)	Add
<input type="text"/>	<input type="button" value="Add"/>

Table #2: Add message passed between objects

Object (Sender)	Message Type	Message	Object (Receiver)	Add
Client	Synchronous message	ACK	Server	<input type="button" value="Add"/>

### Time sequence of events

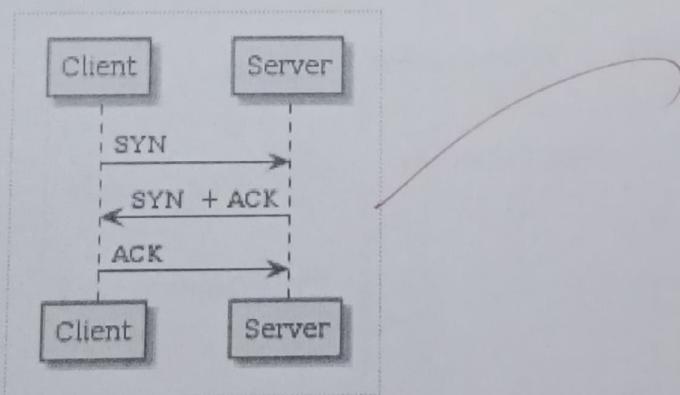
It is possible to reorder the following events by dragging them and then placing it at appropriate position.

"Client" -(SYNC) SYN-> "Server"

"Server" -(SYNC) SYN + ACK-> "Client"

"Client" -(SYNC) ACK-> "Server"

 Draw



## Procedure to draw Sequence Diagram

107

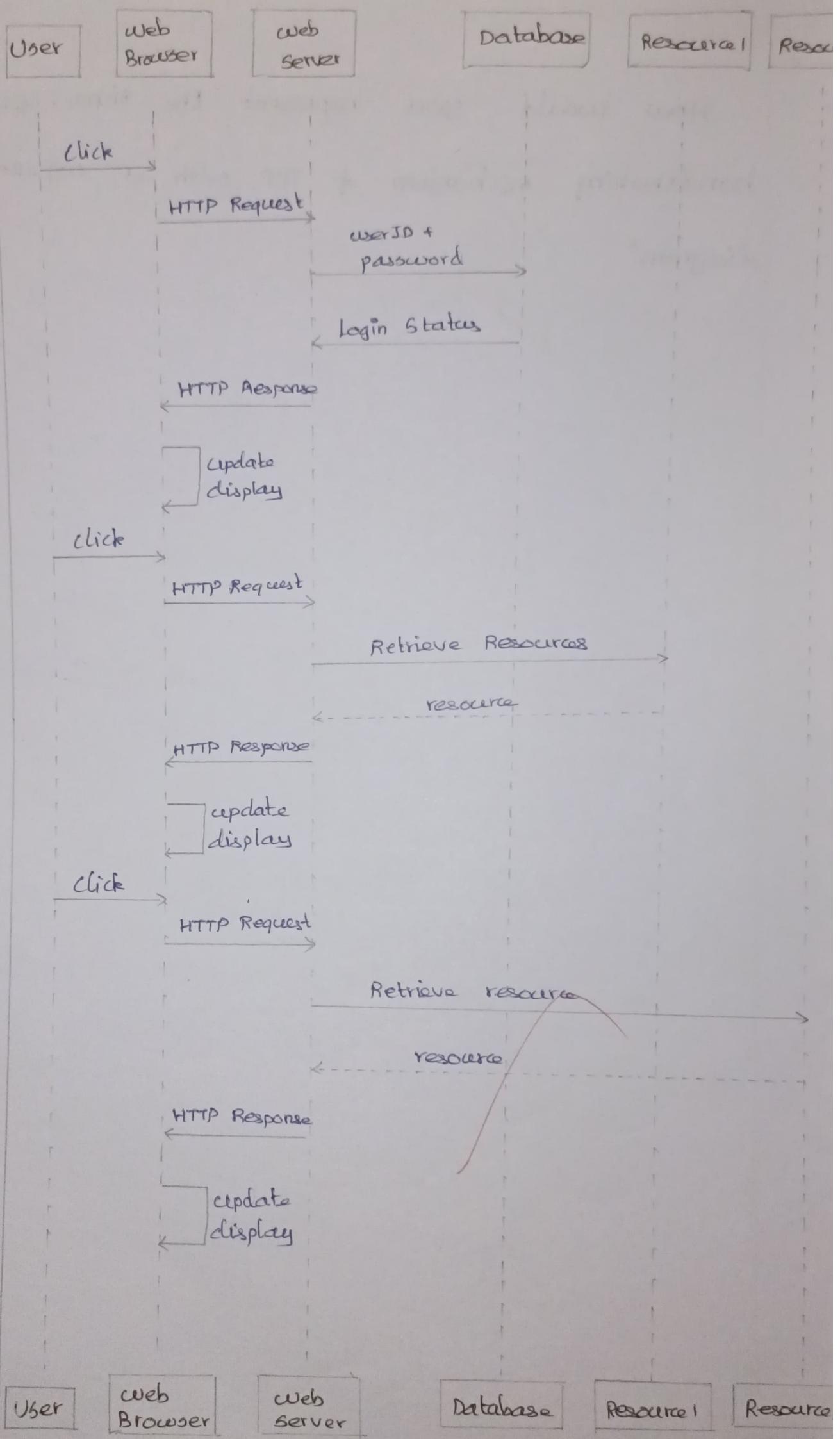
- 1) Read the problem statement carefully and choose the workspace specified for Sequence Diagram.
- 2) Identify an object that will appear on top position of the diagram and write its name in the left textbox of 'Table #1'. Click the adjacent 'Add' button to add this object. Repeat this for all the possible objects.
- 3) 'Table #2' lets you define messages between any two objects. Select the sender object from the first dropdown list, the receiver from the third dropdown list. Select one message type from the second dropdown list. Write the text for the message name to the adjacent textbox. Click on the 'Add' button at the right side to add this message passing. Repeat this for all the possible messages passed from one object to another.

3)

How would you represent the three-way handshaking mechanism of TCP with a sequence diagram?



①



# SEQUENCE DIAGRAM (*Traditional Web*):

## ➤ Tables:

Class Diagram   Sequence Diagram

Table #1: Add object

Object Name (Role)	Add
<input type="text"/>	<input type="button" value="Add"/>

Table #2: Add message passed between objects

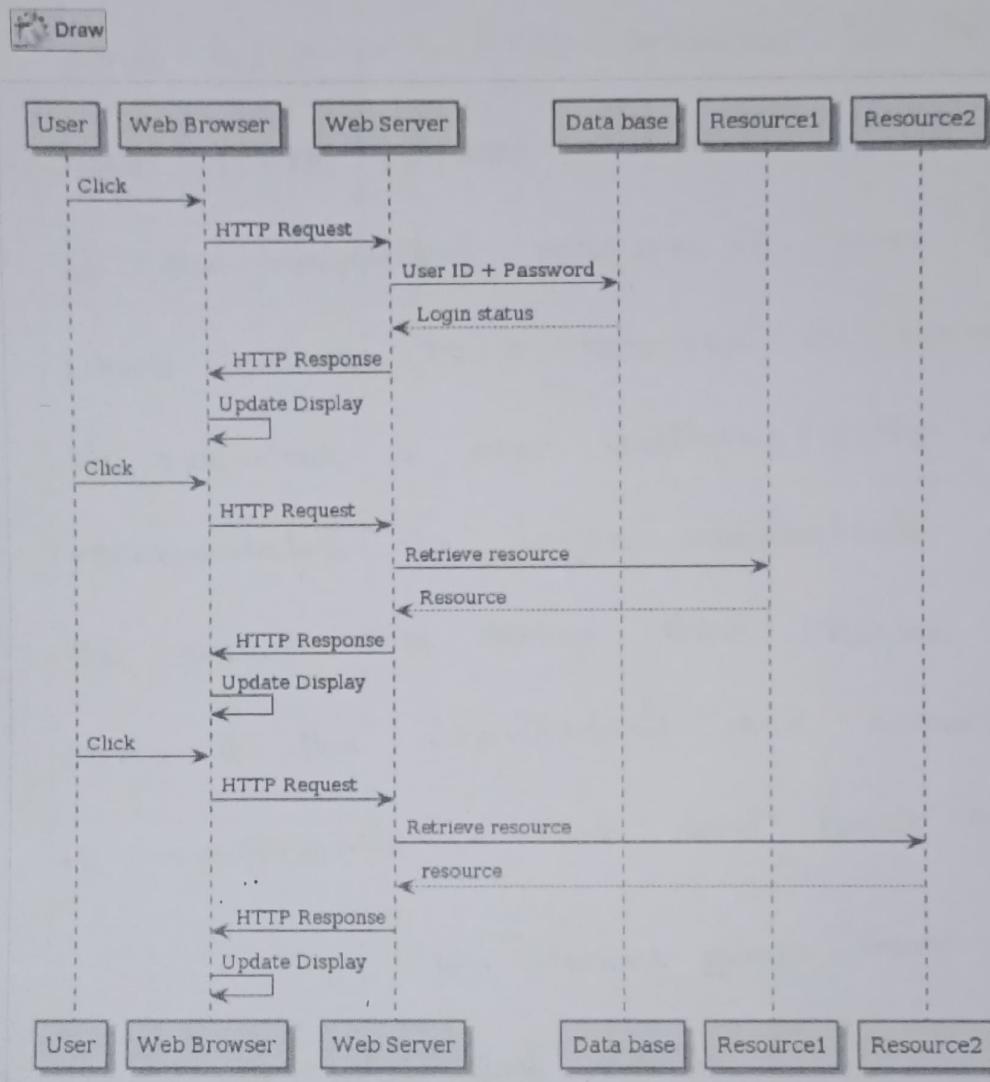
Object (Sender)	Message Type	Message	Object (Receiver)	Add
Select object	Select message type	<input type="text"/>	Select object	<input type="button" value="Add"/>

### Time sequence of events

It is possible to reorder the following events by dragging them and then placing it at appropriate position.

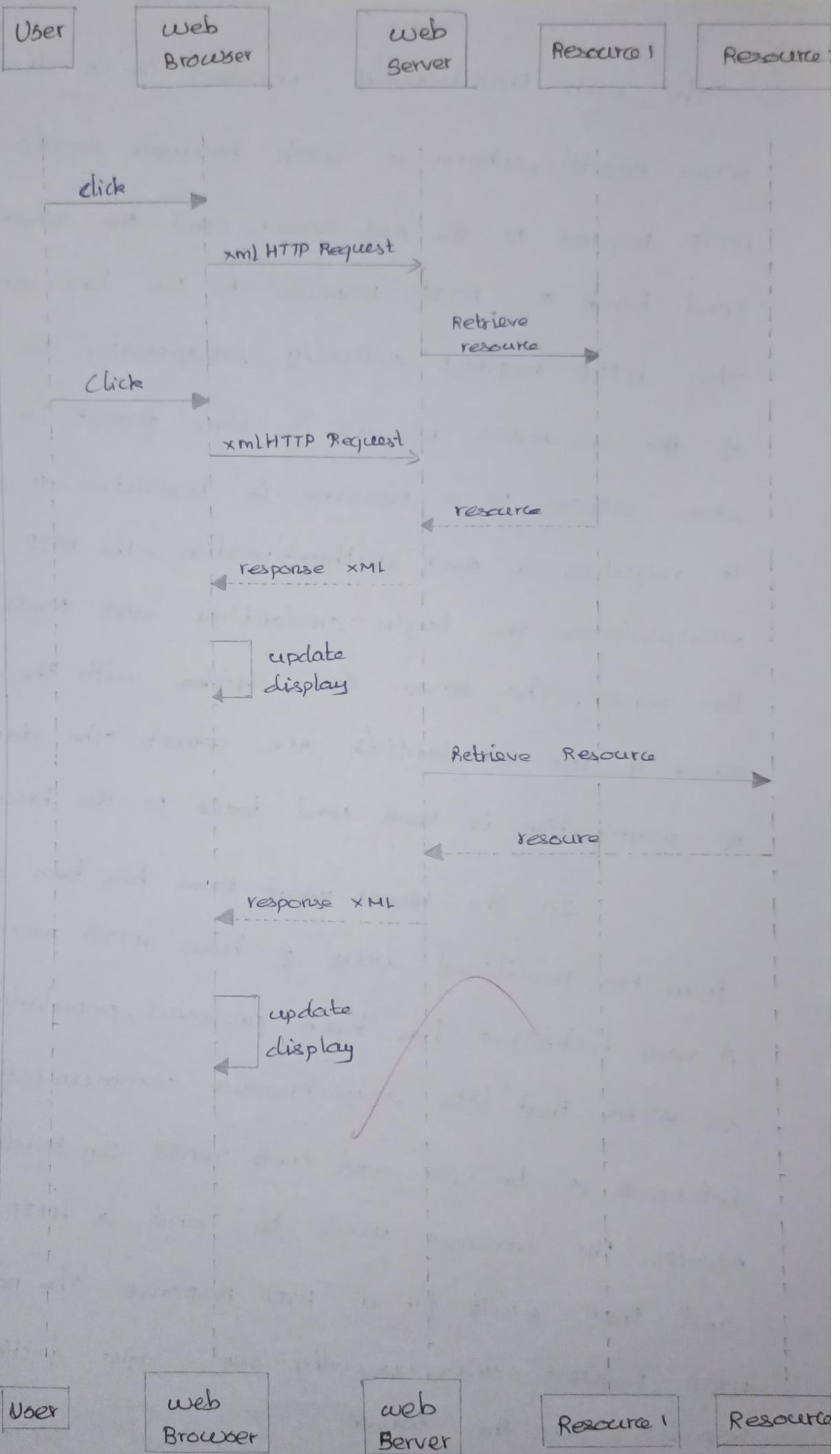
- "User" -(SYNC) Click-> "Web Browser"
- "Web Browser" -(SYNC) HTTP Request-> "Web Server"
- "Web Server" -(SYNC) User ID + Password-> "Data base"
- "Data base" -(RESP) Login status-> "Web Server"
- "Web Server" -(SYNC) HTTP Response-> "Web Browser"
- "Web Browser" -(SYNC) Update Display-> "Web Browser"
- "User" -(SYNC) Click-> "Web Browser"
- "Web Browser" -(SYNC) HTTP Request-> "Web Server"
- "Web Server" -(SYNC) Retrieve resource-> "Resource1"
- "Resource1" -(RESP) Resource-> "Web Server"
- "Web Server" -(SYNC) HTTP Response-> "Web Browser"
- "Web Browser" -(SYNC) Update Display-> "Web Browser"
- "User" -(SYNC) Click-> "Web Browser"
- "Web Browser" -(SYNC) HTTP Request-> "Web Server"
- "Web Server" -(SYNC) Retrieve resource-> "Resource2"
- "Resource2" -(RESP) resource-> "Web Server"
- "Web Server" -(SYNC) HTTP Response-> "Web Browser"
- "Web Browser" -(SYNC) Update Display-> "Web Browser"

## ➤ Sequence Diagram:



3) The web traditionally worked in a client-server model, where a web browser would send a HTTP request to the web server, and the server would send back a HTTP response to the browser. The HTTP request actually encapsulates the contents of the requested resource in some format. In cases, where access to a resource is restricted or say, it requires a user authentication, the HTTP request encapsulates the login credentials and sends to the server. The server then checks with the database if the credentials are correct. The status of verification is then sent back to the browser.

In the recent years there has been a shift from the traditional way of how HTTP works. A new technique has been proposed, popularly known as AJAX, that lets asynchronous communication between a browser and web server. In traditional model, the browser used to send a HTTP request, and then wait for a HTTP response. The next HTTP request was usually sent after getting response from the server.



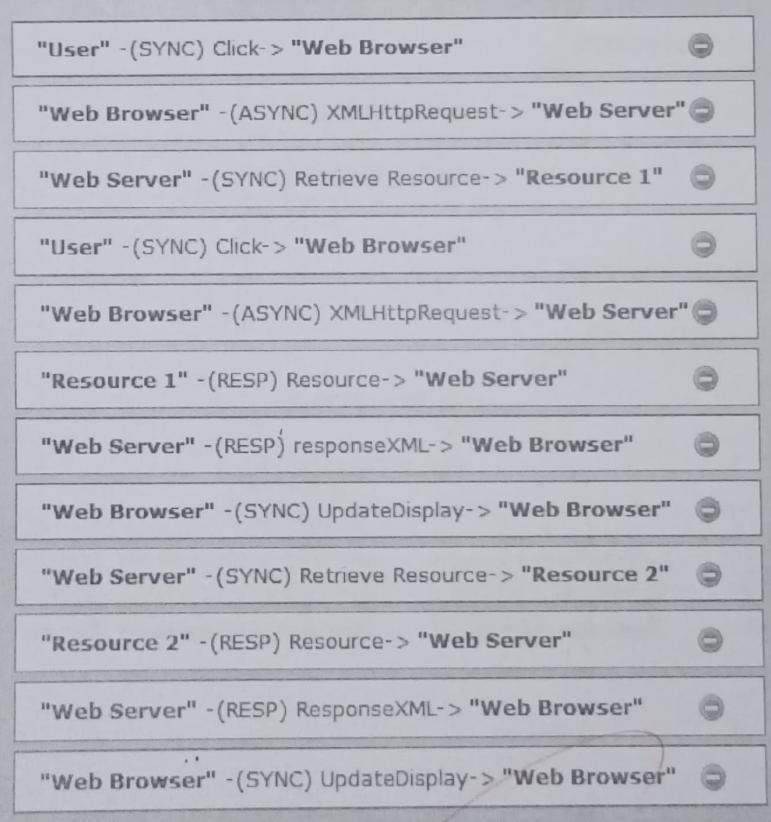
# SEQUENCE DIAGRAM (AJAX):

## ➤ Tables:

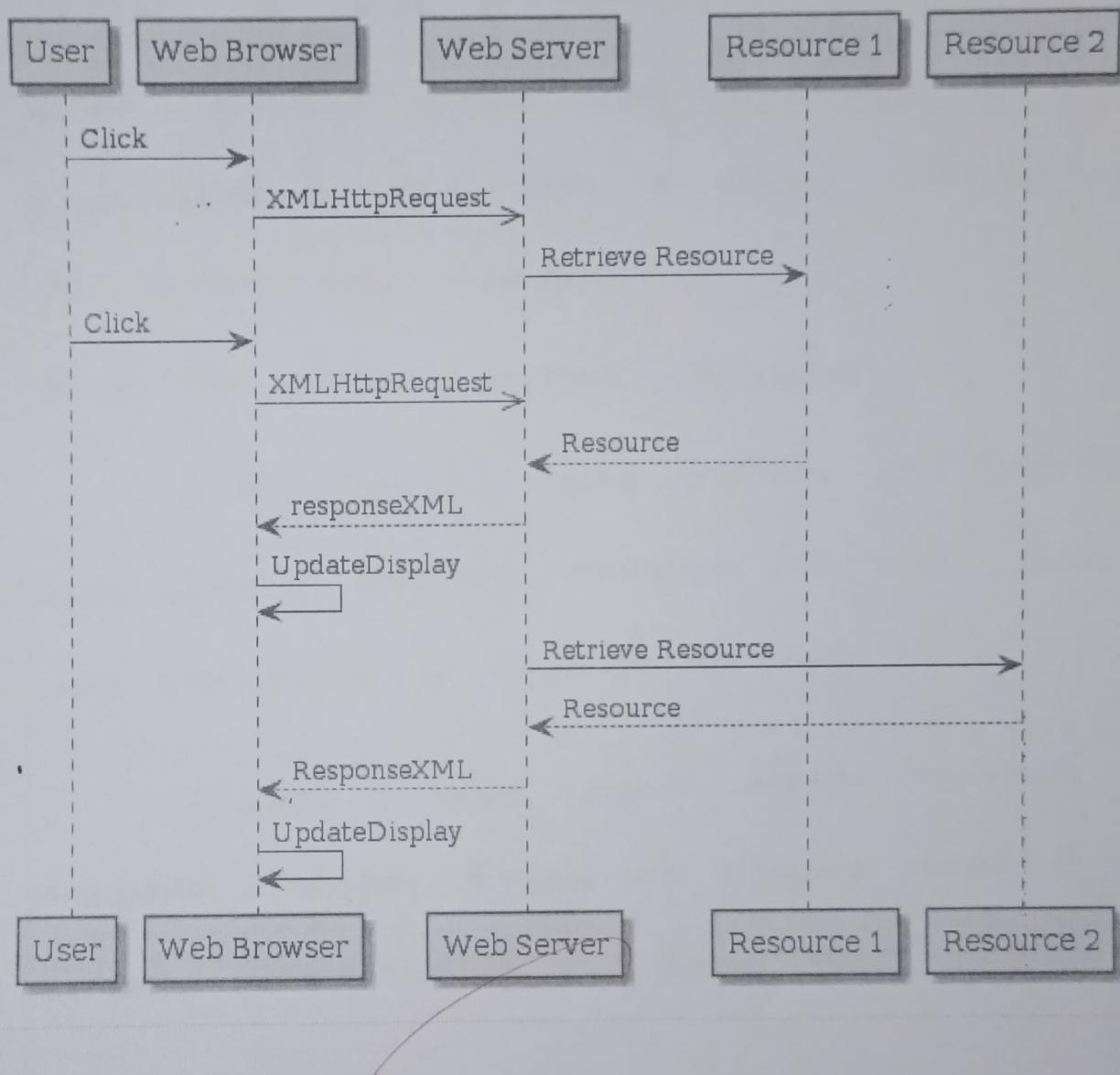
Class Diagram	Sequence Diagram										
Table #1: Add object											
<table border="1"><thead><tr><th>Object Name (Role)</th><th>Add</th></tr></thead><tbody><tr><td><input type="text"/></td><td><input type="button" value="+ Add"/></td></tr></tbody></table>		Object Name (Role)	Add	<input type="text"/>	<input type="button" value="+ Add"/>						
Object Name (Role)	Add										
<input type="text"/>	<input type="button" value="+ Add"/>										
Table #2: Add message passed between objects											
<table border="1"><thead><tr><th>Object (Sender)</th><th>Message Type</th><th>Message</th><th>Object (Receiver)</th><th>Add</th></tr></thead><tbody><tr><td>Web Browser</td><td>Synchronous message</td><td>UpdateDisplay</td><td>Web Browser</td><td><input type="button" value="+ Add"/></td></tr></tbody></table>		Object (Sender)	Message Type	Message	Object (Receiver)	Add	Web Browser	Synchronous message	UpdateDisplay	Web Browser	<input type="button" value="+ Add"/>
Object (Sender)	Message Type	Message	Object (Receiver)	Add							
Web Browser	Synchronous message	UpdateDisplay	Web Browser	<input type="button" value="+ Add"/>							

### Time sequence of events

It is possible to reorder the following events by dragging them and then placing it at appropriate position.



## ➤ Sequence Diagram:



AJAX, however, lets a browser to send multiple HTTP requests one after another, without waiting until a response is received. This approach is found to be very helpful in cases when contents of only a portion of the web page has to be updated, rather than refreshing the entire page. Web 2.0 uses AJAX in many different cases for better user experience.

From the above problem statement

- 1) How would you represent the traditional web with a sequence diagram (in both cases when user verification is required or not)
- 2) what changes would appear in your sequence diagram if you are trying model a scenario where AJAX is being used?

	Total marks	obtained marks
Preparation and Viva	10	09
observation	10	08
Design and Implementation	10	09
Output	10	08
Record	10	08
Total	50	(49)

Result

Thus the UML class and sequence diagrams have been drawn for the given problem statement in virtual lab.

Date: 18/04/2022

## STATECHART AND ACTIVITY MODELING

## Aim

To draw statechart and Activity diagrams  
for the given problems in the virtual lab

## Procedure to draw Statechart Diagram

1) Add states: Identify the possible states of the system and add them through table #1. Type in the ~~same~~ state name in the text box, and then press the 'Add' button. The new state will be shown in a new row in table #4. The new state will also get added to all the dropdown lists in the workspace, that indicates state.

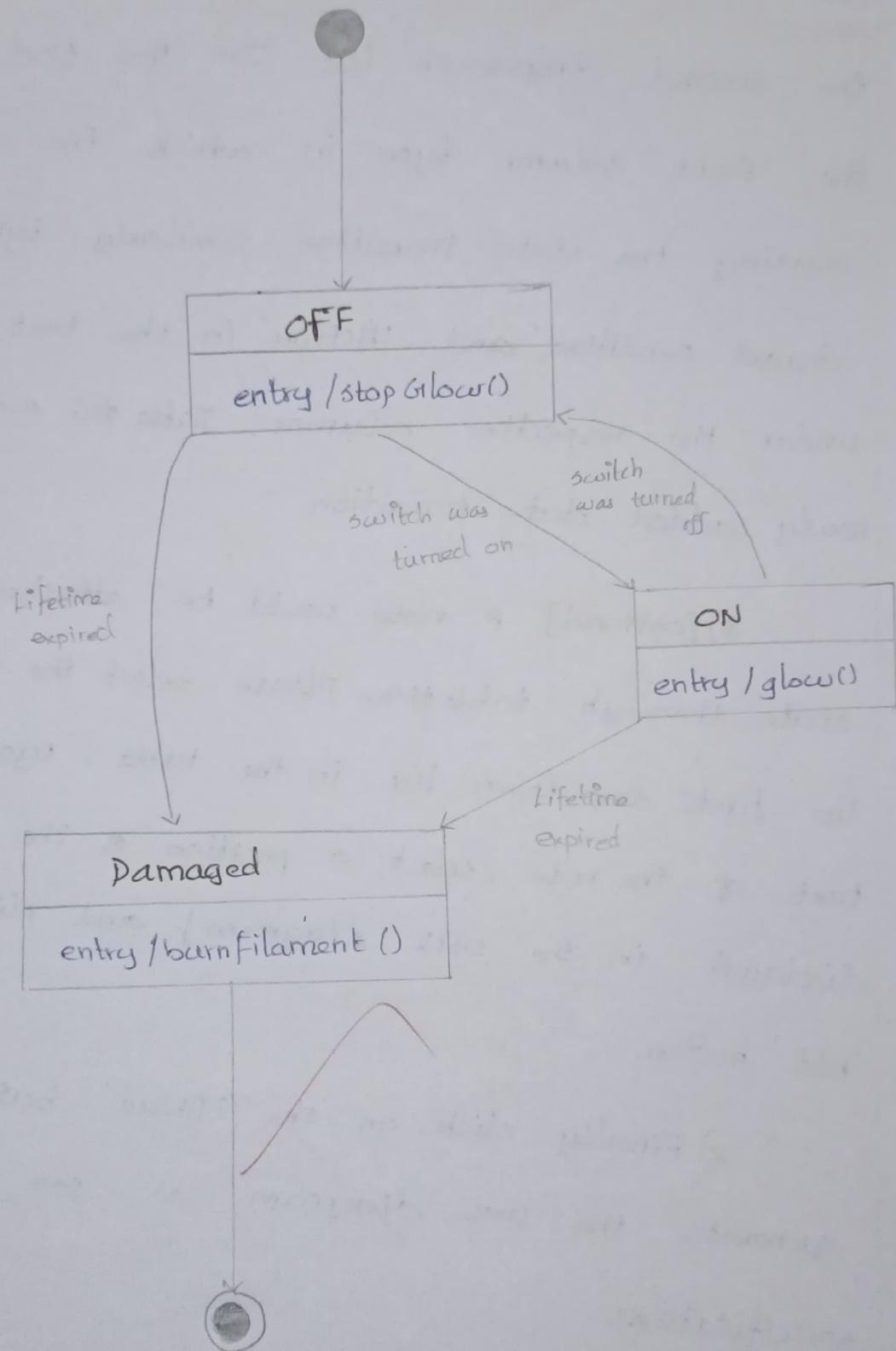
2) Internal activities of a state can be added through table #2. Select the desired state from the first dropdown list, select an action label from the second dropdown list, type in the action expression and then click on the 'Add' button. The 'Activities' column in table #4 for the selected state would get updated accordingly.

3) Add state transitions : Go to table #5 to specify state transitions. Select the current state from the first dropdown list and next state from the second dropdown list. In the text box under the 'Event' column type in which the event is causing the state transition . Similarly type in the 'Guard condition' and 'Action' in the text boxes under the respective columns. Table #6 will display the newly added state transition.

4) [Optional] A note could be attached to any state through table #3. Please select the state from the first drop down list in the table , type in the text of the note , select a position of the note (to be displayed in the UML diagram) and click on the 'Add' button.

5) Finally click on the 'Draw' button to generate the UML diagram as per your specifications

# State chart diagram



# STATE CHART DIAGRAM:

## ➤ Tables and Diagram:

Statechart Diagram   Activity Diagram

Table #1: Add states of the system

State Name	Add
<input type="text"/>	<input type="button" value="Add"/>

Table #2: Internal activities of a state

State Name	Action Label	Action Expression	Add
Damaged	Entry	<input type="text"/>	<input type="button" value="Add"/>

Table #3: Add a note [optional]

State Name	Note	Position	Add
State	<input type="text"/>	Select position	<input type="button" value="Add"/>

Table #4: List of system states

State	Activities	Note	Position	Remove
Off	entry / stopGlow()			<input type="button" value="Remove"/>
On	entry / glow()			<input type="button" value="Remove"/>
Damaged	entry / burnFilament()			<input type="button" value="Remove"/>

Table #5: Define state transitions

Current State	Next State	Event	Guard Condition	Action	Add
On	Damaged	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Add"/>

Table #6: List of state transitions

Current State	Event	Guard Condition	Action	Next State	Remove
Initial				Off	<input type="button" value="Remove"/>
Off	Switch was turned on			On	<input type="button" value="Remove"/>
On	Switch was turned off			Off	<input type="button" value="Remove"/>
Off	Lifetime expired			Damaged	<input type="button" value="Remove"/>
Damaged				Final	<input type="button" value="Remove"/>
On	Lifetime expired			Damaged	<input type="button" value="Remove"/>

Draw

```

graph TD
    Start(( )) --> Off[Off  
entry / stopGlow()]
    Off -- "Switch was turned on" --> On[On  
entry / glow()]
    Off -- "Lifetime expired" --> Damaged[Damaged  
entry / burnFilament()]
    On -- "Switch was turned off" --> Off
    On -- "Lifetime expired" --> Damaged
    Damaged --> Final((( )))
  
```

Result

## problem statement

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

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

153

procedure to draw Activity Diagram

1) In table #1, type in name of the activity in the text box and click on the 'Add' button. The new activity would get added into all the drop down lists (that indicate an activity) appearing in the subsequent tables.

2) To define a flow from one activity to another, go to table #2, select the current and next activities from the first two dropdown lists in the table, specify a (optional) guard condition, and click on the 'Add' button. The transition would appear as a new row in table #3.

3) Parallel Activities (either forking or joining) could be specified through table #4. The first row of this table allows creation of parallel activities, while the second row enables merging.

1) The third column indicates the 'Synchronization Bar' and the dropdown list under this column shows all synchronization points. A new point could be added by clicking on green coloured add

button besides this dropdown list.

135

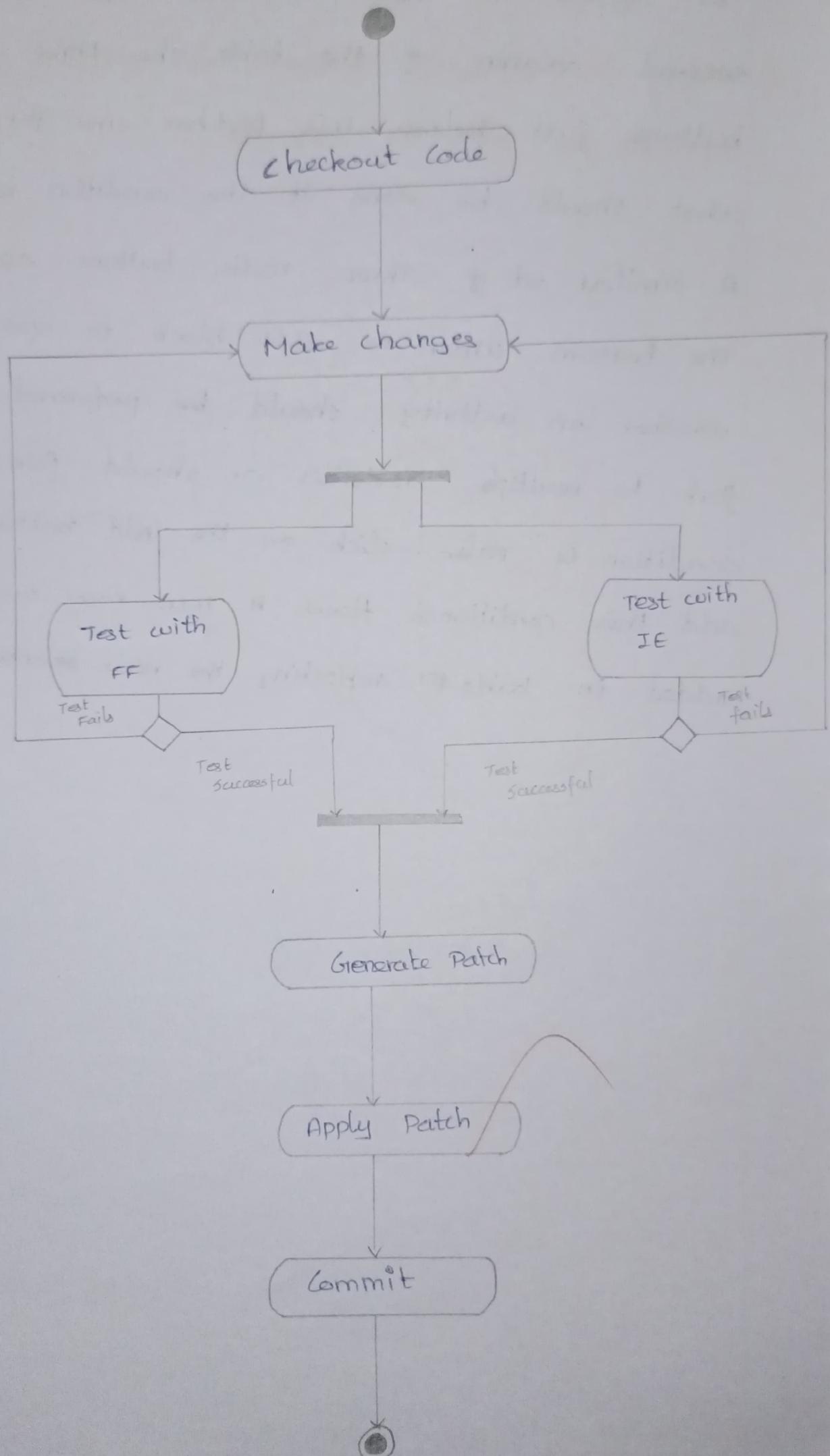
2) To create a set of parallel activities, select the current activity from the first dropdown list in the first row. Now select a synchronization point from the drop down list in the third column. Next, from the five drop down lists under 'Following Activities' column, select atleast two unique activities, then click on the 'Add' button to create the parallel activities. They would appear in a new row in table #5.

3) To join a set of parallel activities, select atleast two unique activities from the five dropdown lists in the second row of the table. Select a join point from the 'Synchronization Bar' dropdown list. Select the activity which should be performed after merging from the single dropdown list under the 'following Activity(ies)' column and then click on the 'Add' button. A new row would get inserted in 'table #5' showing those inputs.

4) Activities performed conditionally: Select the current activity in table #6 after which a decision is to be taken, the condition to check

is typed in the first text box under the second column of the table. The three radio buttons just below this textbox are for identifying what should be done if the condition is true. A similar set of three radio buttons appear at the bottom under the ELSE block to specify whether an activity should be performed, or should fork to multiple activities or should join, if the condition is false. Click on the 'Add' button to add this conditional flow. A new row would be added in table #7 reflecting the new entries.

# Activity Diagram:



# ACTIVITY DIAGRAM:

## ➤ Tables:

Table #1: Add activity

Activity Description	Add
<input type="text"/>	<input type="button" value="Add"/>

Table #2: Add simple sequential flows

Current Activity	Next Activity	Guard Condition	Add
Comit	FINAL_NODE		<input type="button" value="Add"/>

Table #3: List of simple transitions between activities

Current Activity	Next Activity	Guard Condition	Remove
INITIAL_NODE	Checkout Code		<input type="button" value="Delete"/>
Checkout Code	Make Changes		<input type="button" value="Delete"/>
Generate Patch	Apply patch		<input type="button" value="Delete"/>
Apply patch	Comit		<input type="button" value="Delete"/>
Comit	FINAL_NODE		<input type="button" value="Delete"/>

Table #4: Specify parallel workflows (simple)

Type	Current Activity(ies)	Synchronization Bar	Following Activity(ies)	Add
Create parallel activities	Make Changes	bar2	<input type="button" value="Test with FF"/> <input type="button" value="Test with IE"/> <input type="button" value="Parallel activity #3"/> <input type="button" value="Parallel activity #4"/> <input type="button" value="Parallel activity #5"/>	<input type="button" value="Add"/>
Merge parallel activities	<input type="button" value="Test with FF"/> <input type="button" value="Test with IE"/> <input type="button" value="Parallel activity #3"/> <input type="button" value="Parallel activity #4"/> <input type="button" value="Parallel activity #5"/>		Generate Patch	<input type="button" value="Add"/>

NOTE: Please specify separate synchronization bars for fork and join

Table #5: List of parallel activities (simple)

Current Activity	Fork Point	Fork To Activities	Merge Point	Merge To Activities	Remove
Make Changes	bar1	<input type="radio"/> Test with FF <input type="radio"/> Test with IE			<input type="button" value="Delete"/>
<input type="radio"/> Test with FF <input type="radio"/> Test with IE			bar2	Generate Patch	<input type="button" value="Delete"/>

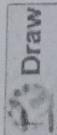
Table #6: Add conditional transitions from activity

Current Activity	Condition & Subsequent Action	Add
Test with IE	<p>IF <input type="text"/> A condition THEN</p> <p> <input type="radio"/> Activity <input type="text"/> Next activity  <input type="radio"/> Fork <input type="text"/> Synchronization bar  <input checked="" type="radio"/> Merge <input type="text"/> bar2  Guard condition: <input type="text"/> True         </p> <p>ELSE</p> <p> <input checked="" type="radio"/> Activity <input type="text"/> Make Changes  <input type="radio"/> Fork <input type="text"/> Synchronization bar  <input type="radio"/> Merge <input type="text"/> Synchronization bar  Guard condition: <input type="text"/> False         </p>	<input type="button" value="Add"/>

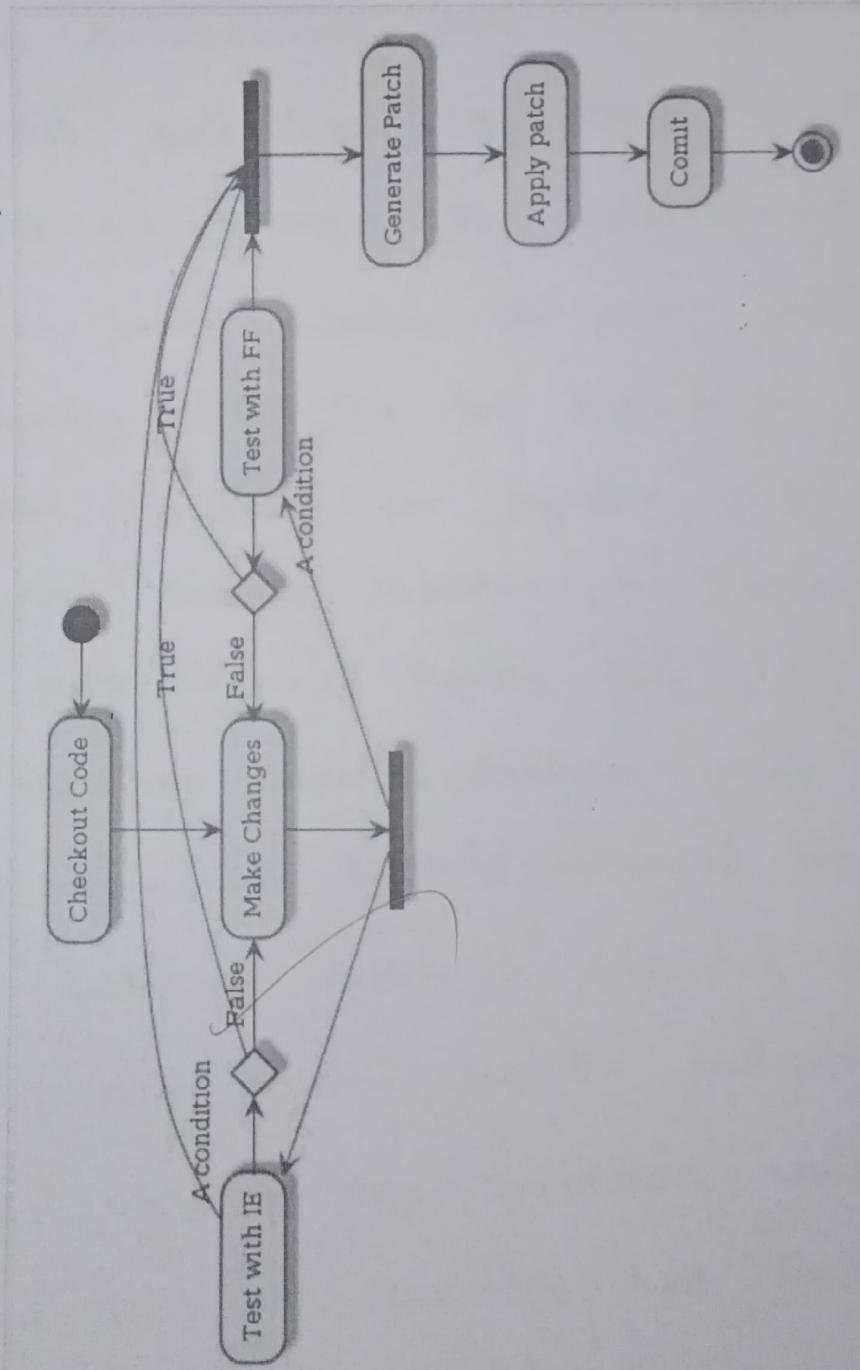
Table #7: List of conditional transitions from activity

Current Activity	Condition	If TRUE	Guard Condition	If FALSE	Guard Condition	Remove
Test with FF	A condition	==bar2==	True	"Make Changes"	False	<input type="button" value="Delete"/>
Test with IE	A condition	==bar2==	True	"Make Changes"	False	<input type="button" value="Delete"/>

## ➤ Diagram:



NOTE: Orientation of some of the arrows change randomly. Please redraw the diagram again to do so.



## Problem Statement

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

Let us consider the development activities of SE virtual labs. The process begins by checking out the code from subversion repository. Necessary modifications are then made to the checked out code (local copy). Once the developer is done, with his changes, the application has to be tested to verify whether the new functionalities are working fine. This test has to be performed with two of the more popular web browsers: Firefox and Internet Explorer, to support cross browser accessibility. If testing fails in at least one of the two browsers, developer goes back to his code, and fixes it. Only when all the browsers pass the test, a patch is generated from the local copy and applied to the production code. The local copy is then committed resulting in update of the SVN repository. Note that, if the local copy is committed before generating a

patch file, then local changes ~~would~~ get registered and one won't be further able to generate the patch file.

NOTE: For further clarification, at any point of time there exists three versions of the source code: production copy, local copy ~~in~~ in SVN repository.

	Total marks	obtained marks
preparation and viva	10	09
observation	10	08
Design and Implementation	10	08
Output	10	07
Record	10	08
Total	50	40

~~Graph~~ Result

Thus the statechart and activity diagrams have been drawn for the problem statements in the virtual lab.

Date: 18/1/2020

## Modeling Data Flow Diagrams

### Aim

To draw Data Flow Diagrams for the given problems in the virtual lab

### Procedure to draw Data Flow Diagrams

1) Identify an external entity and write it in the left textbox of 'Table #1'. Click the adjacent 'Add' button to add this external entity. Repeat this for all the possible external entity.

2) Identify a process and write its name in the left textbox of 'Table #2' and put the level of that process in the next textbox of 'Table #2'. Click the adjacent 'Add' button to add this process. Repeat this for all the possible processes.

3) Identify a data store and write its

name in the left textbox of 'Table #3'. click the adjacent 'Add' button to add this data store. Repeat this for all the possible data stores.

4) If the respective table flashes with green, then it means your entry is taken to the system and you can check it in 'Table #5'. If it flashes with red, then it means your entry is not taken to the system and you are requested to follow the messages generated by system in each other time.

5) If you want to delete any external entity, process or data store ,remove it from 'Table #5'.

6) 'Table #4' lets you define the data flows in between the components in a data flow diagram. Select the type of components from the first dropdown list, the second type from the third dropdown list (4<sup>th</sup> column in the table). Select one component from the second and the other from the fourth (5<sup>th</sup> column in the table) dropdown lists which are participating in a data flow. You

151

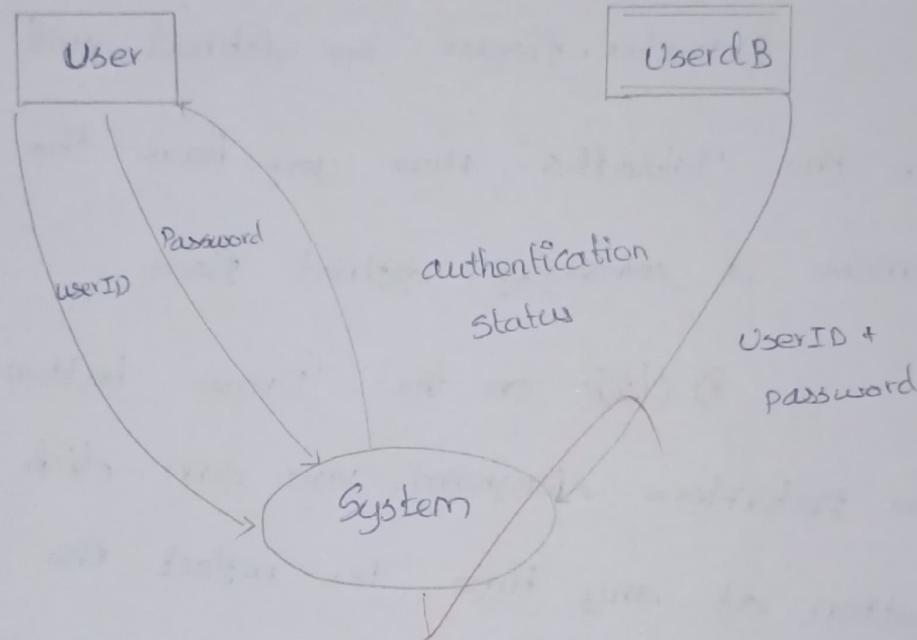
need to put the label of the data flows  
and write the text for the label to the  
adjacent textbox. click on the 'Add' button at the  
right side to add this data flow. Repeat this  
for all possible entries.

7) Data-flows so defined will be displayed  
in the 'Table #6'. Here you have the option to  
remove a wrongly defined flow.

8) click on the 'Draw' button to draw  
the Dataflow diagram. You can click on this  
button at any time to reflect the changes that  
you have made to the external entities, processes,  
data stores or dataflows. The Data Flow diagram  
will be displayed at the bottom of the page.

9) To view a sample solution for the  
exercise, click on the 'submit' button and then  
on the 'view solution' button.

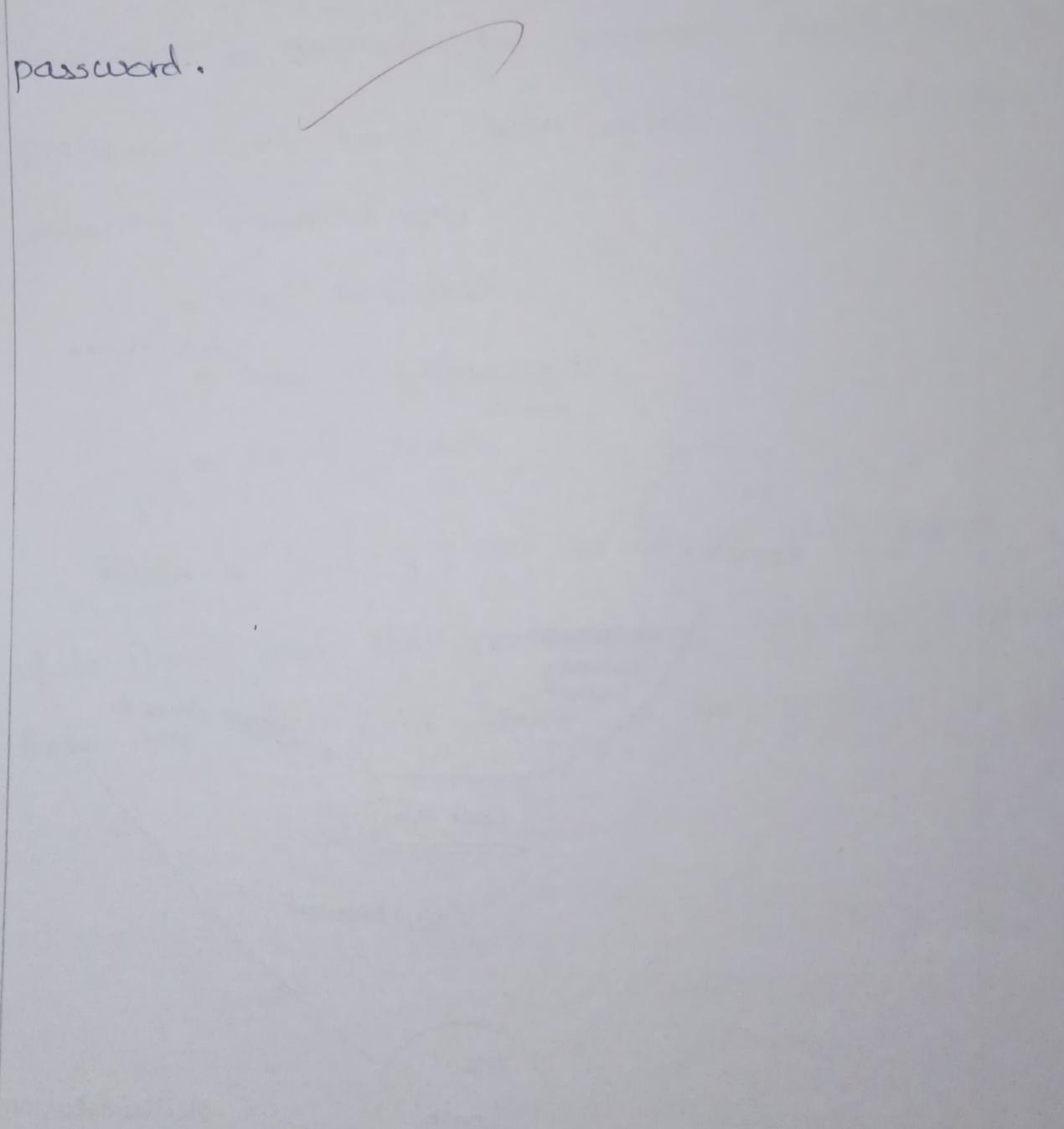
# Data Flow Diagram



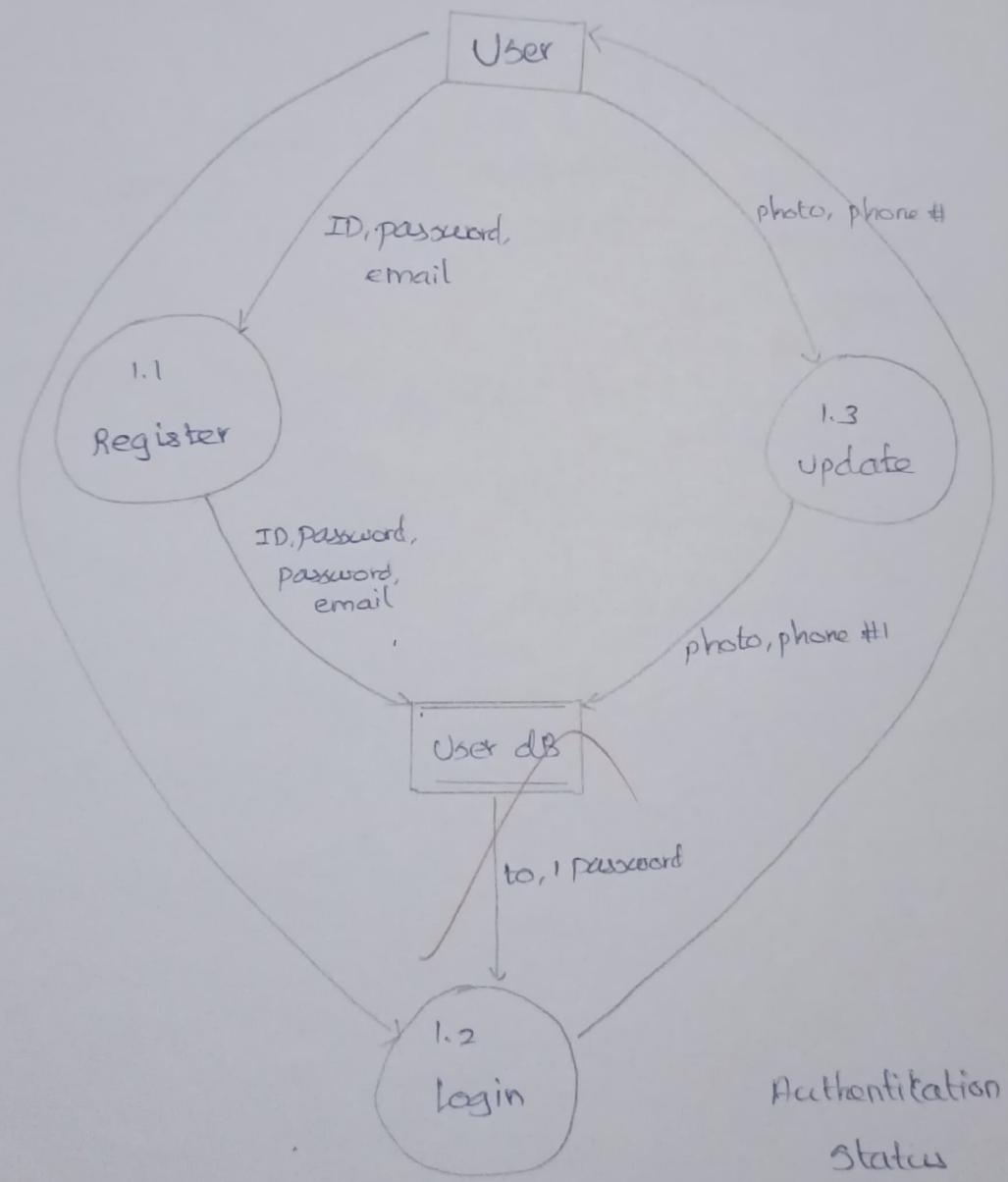
problem statement

marks 6 marks allotted

153

- i) Draw a context-level DFD to depict the typical user authentication process used by any system.
- An user gives two inputs -- username and password.
- 

# Data Flow Diagram



9) DFD for a Social Networking site

155

The Absolute Beginners Inc. is

planning to launch a revolutionary social networking site, EyeCopy. You have been entrusted with designing a DFD for the proposed application. In particular, you have been asked to show the following scenarios.

- \* User Registration
- \* User Login
- \* Profile Update

Draw a level 1 DFD to depict the above data flow and the corresponding processes. Should there be any data store in the DFD?

	Total marks	Obtained marks
Preparation and viva	10	08
observation	10	08
Design and Implementation	10	07
Output	10	06
Record	10	08
Total	50	37

~~Q~~ Result

thus the Data Flow Diagrams have been drawn for the given problem statements in the virtual lab.