

S.no.	Name of the Program	Date of Submission	
1.	To find the problem statement of the software to be implemented	2/8/16	Nish 9/8/2016 (2)
2.	Do the feasibility study & develop SRS specification sheet for system	9/8/16	Nish 23/8/2016 (2)
3.	Do the requirement analysis using interview based techniques for system	23/8/16	Nish 30/8/2016 (2)
4.	To perform the use view analysis for the suggested system - Use case diagram & use case template	30/8/16	Nish 18/9/2016 (1)
5.	To perform function covered diag DFD - 0-1 level	30/9/16	Nish 22/10/2016 (1)
6.	To draw structural view diag for system : class & object diag.	18/10/16	Nish 22/10/2016 (2)
7.	To draw behavioural view diag State chart & activity	18/10/16	Nish 22/10/2016 (2)
8.	To perform sequence & collaboration diag	18/10/16	Nish 22/10/2016 (2)
9.	To draw component diagram of system	18/9/16	Nish 22/10/2016 (2)
10.	To develop deployment diagram	18/9/16	Nish 22/10/2016 (2)
11.	To perform various testing for suggested system	3/10/16	Nish 22/10/2016 (2)
12.	Perform estimation effort using FP Estimation for chosen system	3/10/16	Nish 25/10/2016 (2)

Aim : To find the problem statement of the software to be implemented.

Software requirement : Umbrella

Theory :

- 1) Due to the absence of electronic attendance system, it is very hard to monitor ones attendance. According to the university norms, 75% attendance is mandatory to be eligible to sit for exams. The 75% attendance is recorded manually & the students cannot see their attendance as per their need. As a result of which, students sometimes fail to manage 75% attendance in each subject & are detained.
- 2) This can be solved if there is a software, where the student can enter the class he/she attended or missed & on the basis of that, the software will provide a detailed analysis of his/her attendance, as how many classes he/she has to attend in order to maintain 75% attendance from the remaining number of classes. If the attendance falls below 75%, the software will generate a warning message so that the student can avoid getting detained in that particular subject.
- 3) The software will be pre-loaded with time-table of all the branches and their respective sections (if any).
- 4) The student (user) will add his time-table and for him/her, the software will follow that time-table.
- 5) Either the student can mark his "presence" or "missed" status for the classes, after every class or at the end of the day.
- 6) The software will also be pre-loaded with total number of lectures



each subject.

- 7) Therefore, with every entry that the student makes for the classes he/she attended, the overall data will be updated everyday.
- 8) Bar Graphs, Pie-Chart and Histograms will also be provided depicting the same data.

Nish
9/8/2016



AIM : Do the feasibility study and develop SRS specification sheet for suggested system

SOFTWARE REQUIREMENTS : Umbrella version 2.13.3

THEORY

Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives dominate the negatives, then the system is said to be feasible.

TECHNICAL FEASIBILITY

We can strongly say that it is technically feasible, since there will not be much difficulty in setting required resources for the development & maintaining the system. We will require a strong algorithm that supports the functionality of the software and to that we will have to study PHP programming language.

SCHEDULE FEASIBILITY

For the whole development of the program & its testing, it will take around 10 weeks.

ECONOMIC FEASIBILITY

Development of this application is highly economically feasible. The only thing is to be done is making an environment for the development with an effective supervision. The whole process will cost around 1,00,000. Some amount of money will also be required to take appropriate permissions & installations.

LEGAL/ETHICAL FEASIBILITY

The developed software will be legal & will not harm any one's ethics. The entire software will be first sanctioned by the higher authorities.

CULTURAL FEASIBILITY

People will not come across any cultural problem while using this software.

RESOURCE FEASIBILITY

All the resources required for the development of the proposed software are easily available. Resources such as data in the form of time-table of different branches & sections, computer systems, internet accessibility



and programmers for front-end development & one for backend development

MARKETING FEASIBILITY

This software can be used at many places with a small change in the software. So it has a huge scope in the market. Our target audience are the students of any engineering college or any other institution.

OPERATIONAL FEASIBILITY

It is operationally feasible & all the solutions are already defined for all the possible errors which might occur in near future.



NotesHub.co.in | Download Android App

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

1. INTRODUCTION

1.1 PURPOSE

The main objective of this document is to illustrate the requirements of the project "Attendance Manager". The document gives the detailed description of both the functional & non-functional requirements proposed by the user/client. The purpose of this software is to provide a friendly environment to maintain the details of attendance & to keep a track of it.

1.2 SCOPE

Attendance Manager can be used by the student of an institution or organization to keep a track of their attendance. It will help the students/employees to maintain a healthy attendance as well as help them to plan their vacation/holidays accordingly.

1.3 DEFINITIONS, ACRONYMS & ABBREVIATIONS

- ① SQL - Structured Query Language
- ② SRS - Software Requirement Specification
- ③ PHP - Hypertext Preprocessor
- ④ JAVA - platform independence.

2. REFERENCES

<http://www.google.co.in>

1.5 OVERVIEW

The rest of SRS contains details about requirements, description of software, document approvals, supporting information etc.



2. THE OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The user can be either a student or an employee of an organization. This product will be useful to them if there is no electronic or digital way of entering attendance. This software will replace the manual entering of attendance. The user can see how many classes he/she has attended, how many he/she has to attend to maintain healthy attendance. The product will also warn the user if he/she is short on attendance.

2.1.1 SYSTEM INTERFACES

System interfaces will consist of interface b/w the login page, database marking the attendance page, viewing the attendance page, time-table page.

2.1.2 THE INTERFACES

The proposed software will contain interfaces like filling student's information into login details, inserting, deleting & updating the subjects in the time-table and marking/demarking the attendance on the attendance page.

2.1.3 HARDWARE INTERFACES

Screen resolution of the software will vary from 320×480 pixels to 1080×1920 so that the software can be seen both on mobiles & desktop screens.

2.1.4 SOFTWARE INTERFACES

Any windows/ubuntu based operating software/system supporting SQL for database functioning.



Ethernet Port : very for network connection.

Power-supply: to run the computer.

2.1.6

MEMORY CONSTRAINTS

The memory required for the program is nearly 500 MB in computer and 25 MB in mobile. It will grow as the entries are made & new time-tables are inserted.

2.1.7

OPERATIONS

Operations include login, entering time-table, choosing pre-installed time-table to follow, marking and demarking the attendance corresponding to certain subjects & viewing the progress so far. All this stored in the database, though it can be recovered if at all the software crashes.

2.1.8

SITE ADAPTION REQUIREMENT

The terminals at client side will have to support the hardware interfaces specified in above section.

2.2

PRODUCT FUNCTIONS

The main function of this project is to reduce the manual handling of attendance of students in institutional form and to help the students track their attendance. The software is capable of providing the user (student) with pre-installed time-table according to his/her branch & section, allow the user to mark the classes he/she attended, provide a detailed analysis of his/her attendance in every subject & warn the student if his/her attendance is below the acceptable percentage.



- Features available for student are:
- (i) He/she can view the time-table of his/her class after logging in.
 - (ii) He/she can choose any one of the pre-installed time-table to follow or can also insert his/her own time-table.
 - (iii) He can mark "Attended", in front of the classes he/she did & can mark "Not Attended", in front of the classes he/she bunked.
 - (iv) He can check the attendance in pie-chart/bar-graph & histogram representations.
 - (v) He/she can set a limit of percentage attendance. If the attendance drops below this %, the software will warn the user.

2.4 CONSTRAINTS

- 1. The success of this software will depend upon how ~~punctual~~ punctual the user make entries.
The system will work well until:
 - a) The user make entries regularly.
 - b) The program is not corrupted.
 - c) Power is there

2.5 ASSUMPTIONS & DEPENDENCIES

- 1. The coding should be error free.
- 2. The system should be user friendly so that user can use it easily.
- 3. The information of all the entries made by the user must be stored properly & should be accessible at all times.

DEPENDENCIES

- 1. a) The specific software & hardware due to which the product will run.
- b) The admin should have proper understanding of the program.



REQUIREMENTS (APPORTIONING)

Not Required.

SPECIFIC REQUIREMENTS

3.1

External Interfaces

No such interfaces required.

3.2

FUNCTIONS

NONE

PERFORMANCE REQUIREMENT

NONE

3.4

LOGICAL DATABASE REQUIREMENTS

SQL is required for storing the data info of student, and the time-table of every branch & section. It will also store the markings & demarcations made by user corresponding to the classes he/she attended & missed respectively.

3.3.5

DESIGN CONSTRAINTS

1. Programming language should be JAVA
2. Environment should be eclipse 3.4.0 IDE
3. Database : Microsoft SQL 2008
4. Operating System : Windows/Ubuntu with atleast 10GB Hard Disk, 2 GB RAM & 1.2 GHz Processor.

3.

SOFTWARE SYSTEM ATTRIBUTES

3.6.1 RELIABILITY

The reliability is tested & the product can be trusted fully if proper care of database & software is taken.



The database is stored in the backend so the availability of some tables i.e data can be accessed by user only after successful login.

3.6.3 SECURITY

The application will be password protected & only that user can open, who has a valid account with a unique set of username & password.

3.6.4 MAINTAINABILITY

The application will be designed in a maintainable manner.

3.6.5 PORTABILITY

The application is easily portable on windows & Ubuntu OS.

3.7 ORGANIZING THE SPECIFIC REQUIREMENTS

3.7.1 SYSTEM MODE

Windows & Ubuntu OS

3.7.2 USER CLASS

Database

Login

Mark "Attended" & Mark "Not Attended" - MARK Attendance

View Attendance

View Time Table

3.7.3 OBJECTS

3.7.4 FEATURE

It will allow the user to mark/demarc attendance, view attendance.

3.7.5 STIMULUS

It will be quick in response as the user is directly responsible for the results.

3.7.6 RESPONSE

Software response will be adequate as long as good coding & hardware are being used.

3.7.8 ADDITIONAL COMMENTS

The user must have good knowledge of computers.

Nitin
23/8/2016



AIM: Do the requirement analysis using interview based techniques for the suggested system.

SOFTWARE REQUIREMENT: Umbrella

THEORY:

Developer: D

Client : C

D: Who has requirement for such a software?

C: All the student of the institution will be using this software to track their own attendance so that even they have a proper record.

D: Why was such a software required?

C: The institution does not have automatic attendance manager. The institution also lacks a platform where the teachers can upload attendance of every student regularly & the student can login with their unique set of username & password to check their attendance. Therefore, the students are unaware of their present % of attendance.

D: Approximately how many students will be using this software?

C: Based on last month's statistics, around 500 students will be using this software to track their attendance.

D: What are the basic requirements that the software must do in terms of functionalities?

C: Through the software,

(a) The students can view time-table of their branch & section.

(b) The students can mark "Attended" corresponding to the classes he/she has attended & mark "Not Attended" corresponding to the classes he/she did not attend.

(c) On every entry made by the student, the overall attendance of every subject must be updated.



- d) The student can view his/her attendance in form of pie-chart, bar graphs & histograms.
- e) The student will be allowed to set a percentage limit on the attendance of every subject (e.g 75%).
- f) If the attendance drops below this "percentage limit", the software must show a warning message, with the number of classes he/she has to attend to again maintain a healthy percentage.

D: By when do you need this software?

C: We will be needing this software before the start of next semester i.e within 3 months.

Nit
30/8/2016



EXPERIMENT - 4

Aim: To perform the user view analysis for the suggested system with use case diagram & use case template.

REQUIREMENTS: UML

THEORY:

USE CASE DESCRIPTION

1. LOGIN

(a) BRIEF DESCRIPTION

This use case documents the process of logging into "Attendance Manager" based on user privileges.

- Professor / Teacher (Upload Time-Table)
- Student (View Time-Table, Mark Attendance, View Attendance)

(b) ACTORS

- Professor / Teacher Coordinator
- Student

(c) PRE-FLows

FLOW OF EVENTS

(c-i) BASIC FLOW

- (i) This use case starts when actor wishes to log-in to the Attendance Manager System.

start

- (ii) The system requests the actor enters his/her user-id and password. For teacher-coordinator, the user-id will be their employee-id and for student their username or user-id will be their enrollment number.

- (iii) The actor enters user-id & password.

- (iv) The system validate the user-id & password & checks for his/her privileges.

N
18/10/2020

- (iv) Otherwise, if the user is "Student", he/she will be logged into the system & presented with student's menu.

(c-2) ALTERNATIVE FLOW

- (i) If the system receives an invalid user-id &/or password, an error message is displayed & the use case ends.

(d) SPECIAL REQUIREMENTS

None

(e) PRE-CONDITIONS

NONE

(f) Post-condition

If use case is successful, the user is logged into the system, otherwise the system state is unchanged.

(g) EXTENTION POINTS

None.

2 UPLOAD TIME-TABLE

2.1 BRIEF DESCRIPTION

This use case documents the process of uploading time-table of different branches & their corresponding sections based on user privileges.

• Teacher-Coordinator

2.2 ACTORS

Teacher-Coordinator

2.3 FLOW OF EVENTS

2.3.1 BASIC FLOW

This use case starts when actor wishes to upload the time-table to the Attendance Management system.

(ii) The system requests the actor to enter the class and section of the particular branch. (Here branch & class are same).

(iii) The actor enters the branch (e.g IT) and section (e.g. 5A).

(iv) The system validates the user input & then shows the existing time-table of that branch & section (if any). The system asks the actor to upload new time-table.

(v) The actor uploads the time-table.

(vi) The details are updated in the system's database

(vii) The use case ends.

2.3.2 Alternative Flow

2.3.2.1 Branch & Section combination not available.

If the system doesn't validate the branch & section combination entered by the teacher-coordinator, then an error message is flagged & the system asks the actor to enter both the choices again.

2.4 SPECIAL REQUIREMENTS

None

2.5 PRE-CONDITIONS

Teacher-Coordinator must be logged into the system.

2.6 POST-CONDITIONS

If the use case is successful, the new time-table is uploaded.

2.7 EXTENSION POINTS

None.

3. VIEW TIME TABLE

3.1 BRIEF INTRODUCTION

This use case documents the process of viewing & choosing/selecting time-table based on student priorities.

Shubham



③ FLOW OF EVENTS

3.3.1 Basic Flow

- (i) This use case starts when actor wishes to use the system to track his/her attendance & therefore wishes to select his/her time-table.
- (ii) The system requests the actor to enter the branch & section in which he/she is studying.
- (iii) The actor enters the branch & section.
- (iv) The system validates the actor's input & shows the time-table for the chosen set of branch & section.
- (v) The system accepts this time-table to follow from now onwards until unchanged.
- (vi) The use case ends.

3.3.2 Alternative Flow

3.3.2.1 Mismatch between Branch and Section.

If the system doesn't validate the branch & section combination, an error message is flagged.

④ SPECIAL REQUIREMENTS

None

⑤ PRE-CONDITIONS

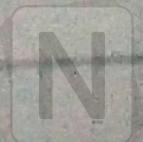
Student must be logged into the system.

⑥ POST-CONDITIONS

If use case is successful, then the student will choose his/her time-table which the system will follow.

⑦ EXTENSION POINTS

None



1. MARK ATTENDANCE

1.1 BRIEF DESCRIPTION

This use case documents the process of marking attendance corresponding to the subjects of the time-table chosen by the student.

1.2 ACTORS

Student

1.3 FLOW OF EVENTS

4.3.1 BASIC FLOW

- (i) This use case starts when actor wishes to mark his/her attendance corresponding to the class he/she attended.
- (ii) The system shows the time-table which have been chosen with the subject.
- (iii) The actor selects the subject, marks "ATTENDED" or "MISSED".
- (iv) The system asks for confirmation.
- (v) The actor provides the confirmation.
- (vi) The system updates the percentage attendance of that particular subject. It decreases it if marked "MISSED" or increases it if marked "ATTENDED".
- (vii) The use case ends.

4.3.2 ALTERNATIVE FLOW

None.

4. SPECIAL REQUIREMENTS

None.

5. PRE-CONDITIONS

Student user must be logged into the system.

6. POST-CONDITIONS

If use case is successful, then the % attendance of the subjects marked will increase or decrease.

7. EXTENSION POINTS

None.



5. VIEW ATTENDANCE

5.1 BRIEF DESCRIPTION

This use case document, it's process of viewing attendance.

5.2 ACTORS

Students

5.3 FLOW OF EVENTS

5.3.1 BASIC FLOW

- (i) The use case starts when actor wishes to view his/her attendance.
- (ii) The actor request the system to show his/her attendance.
- (iii) The system responds to actor's request & shows the attendance subject wise. It also shows the subject green in color whose attendance is above % limit, & shows the subject red in color whose attendance is below % limit.
- (iv) The use case ends.

5.3.2 Alternative Flow

None.

5.4 SPECIAL REQUIREMENTS

None

5.5 PRE-CONDITION

The student user must be logged in.

5.6 POST-CONDITION

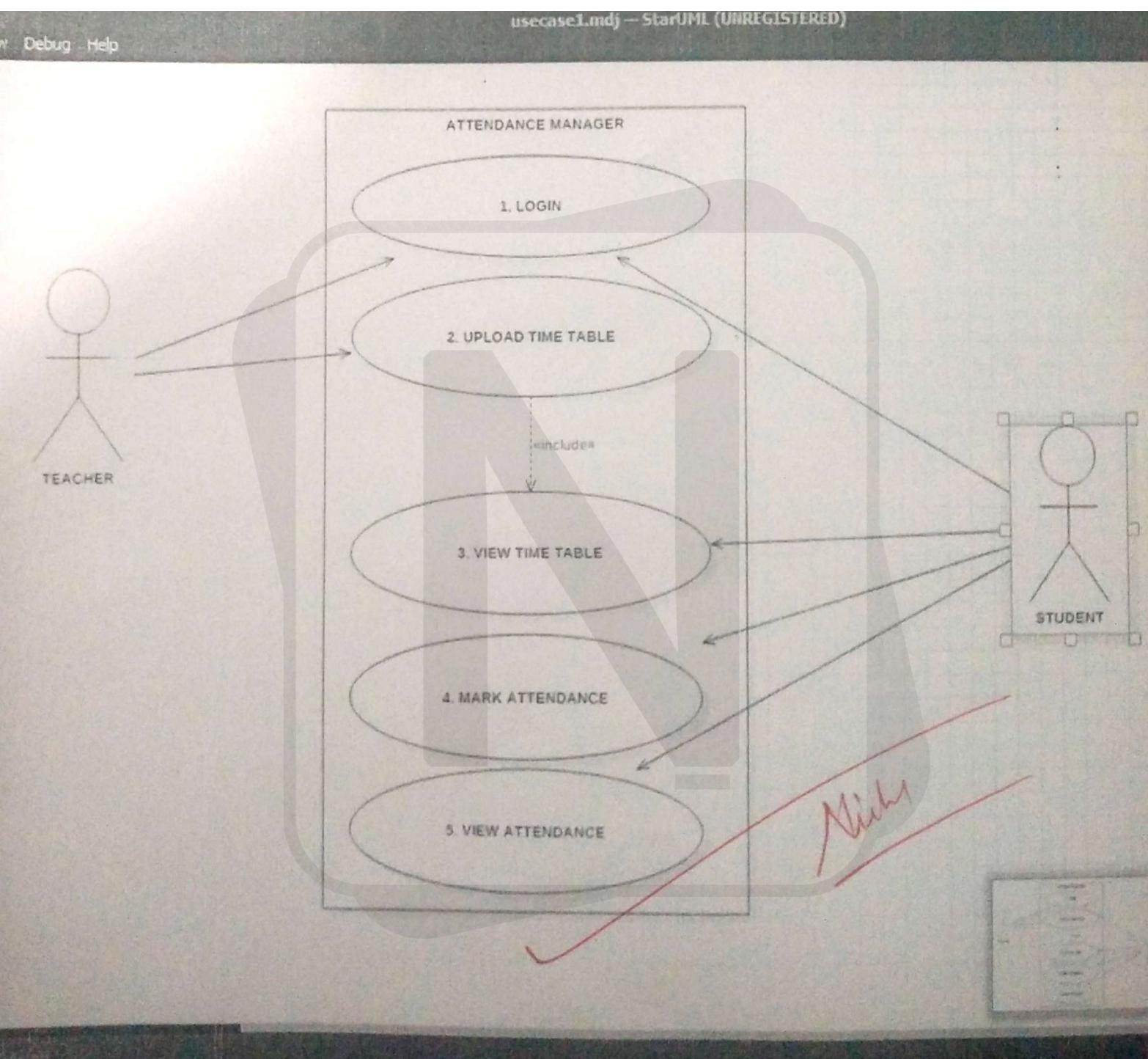
If this use case is successful, the student will view his attendance.

5.7 EXTENSION POINTS

None

Next
18/10/16





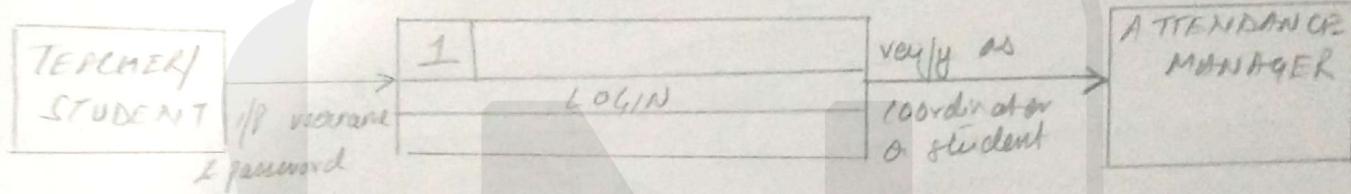
EXPERIMENT - 5

Aim: To perform the function oriented program : DataFlow Diagram (DFD)
Level 0 & Level 1 DFD.

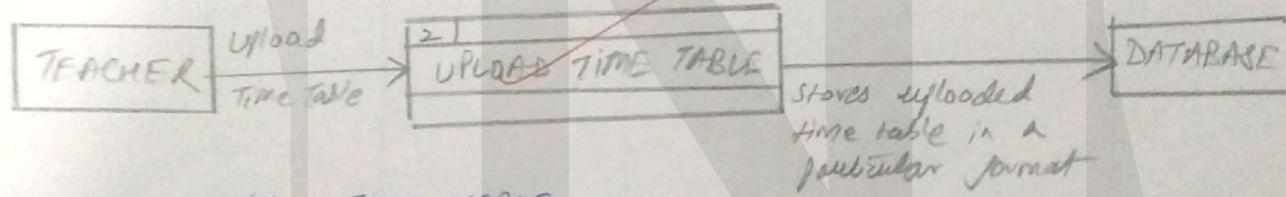
SOFTWARE REQUIRED: Umbrello Version 2.13.3

THEORY: To create DFD level-0 & level-1, first we need to identify all the major process that are going on in the whole system.

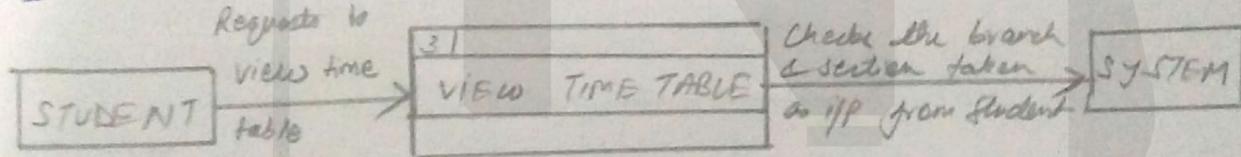
PROCESS - 1: LOGIN PROCESS



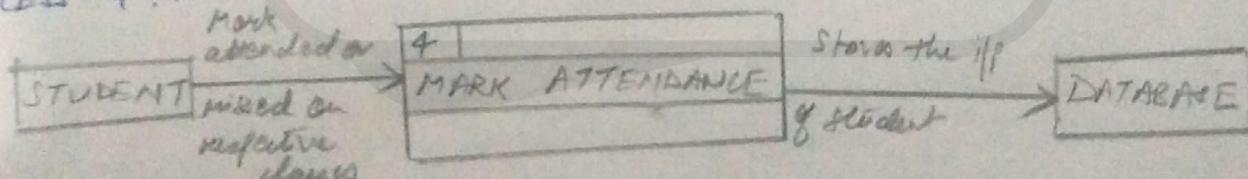
PROCESS - 2: UPLOAD TIME TABLE



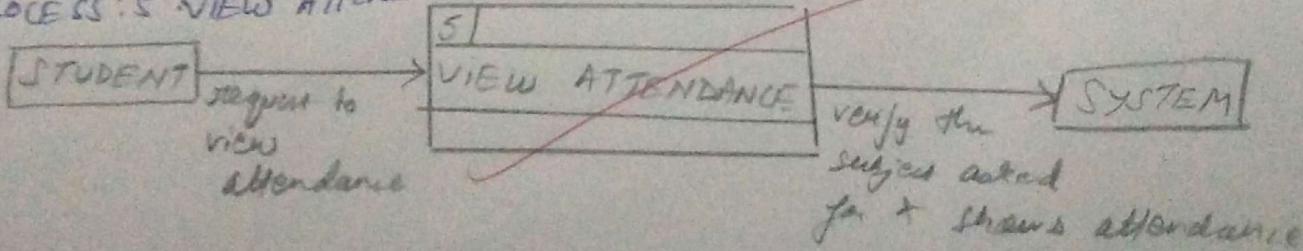
PROCESS - 3: VIEW TIME TABLE



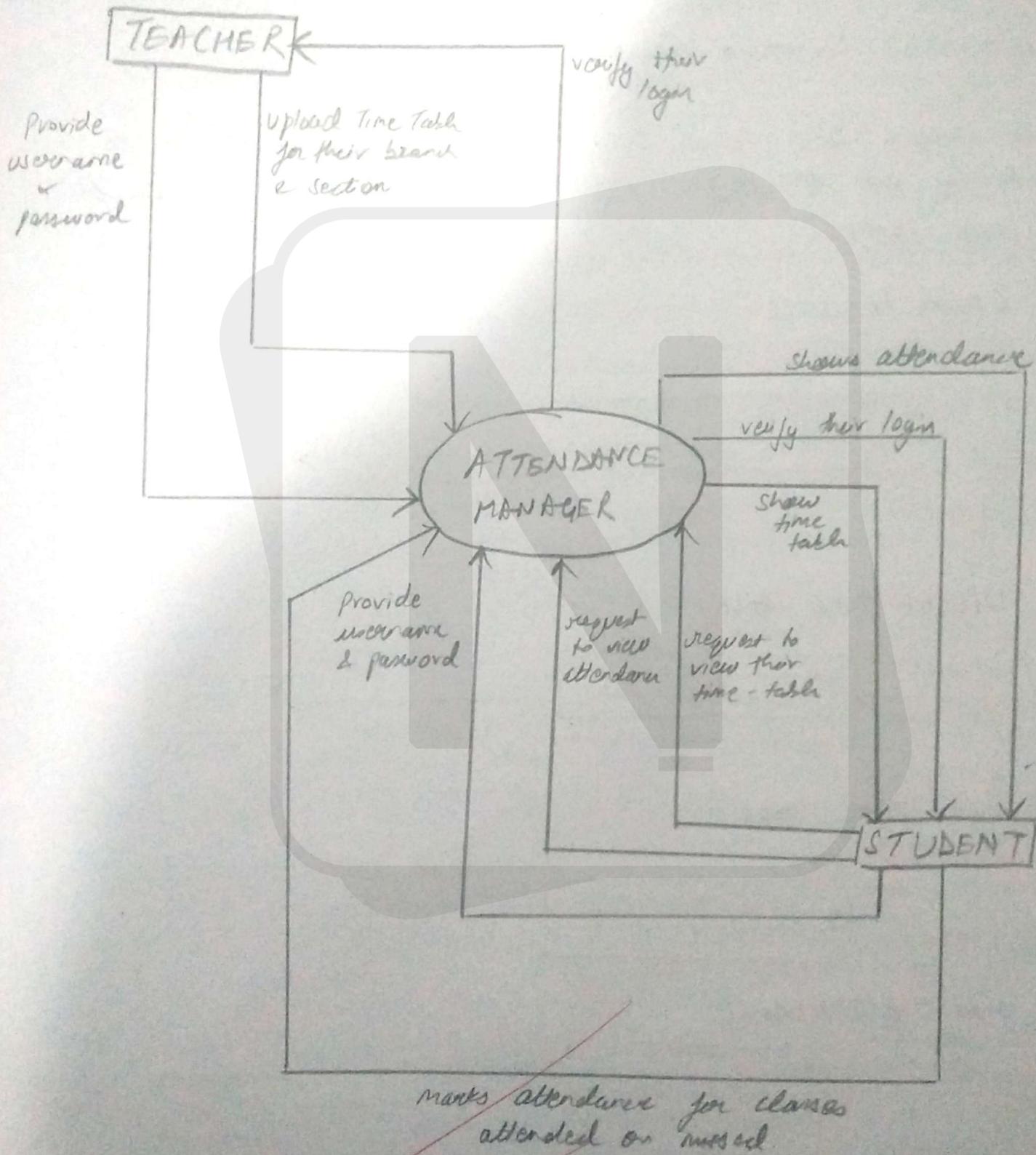
PROCESS - 4: MARK ATTENDANCE



PROCESS - 5: VIEW ATTENDANCE



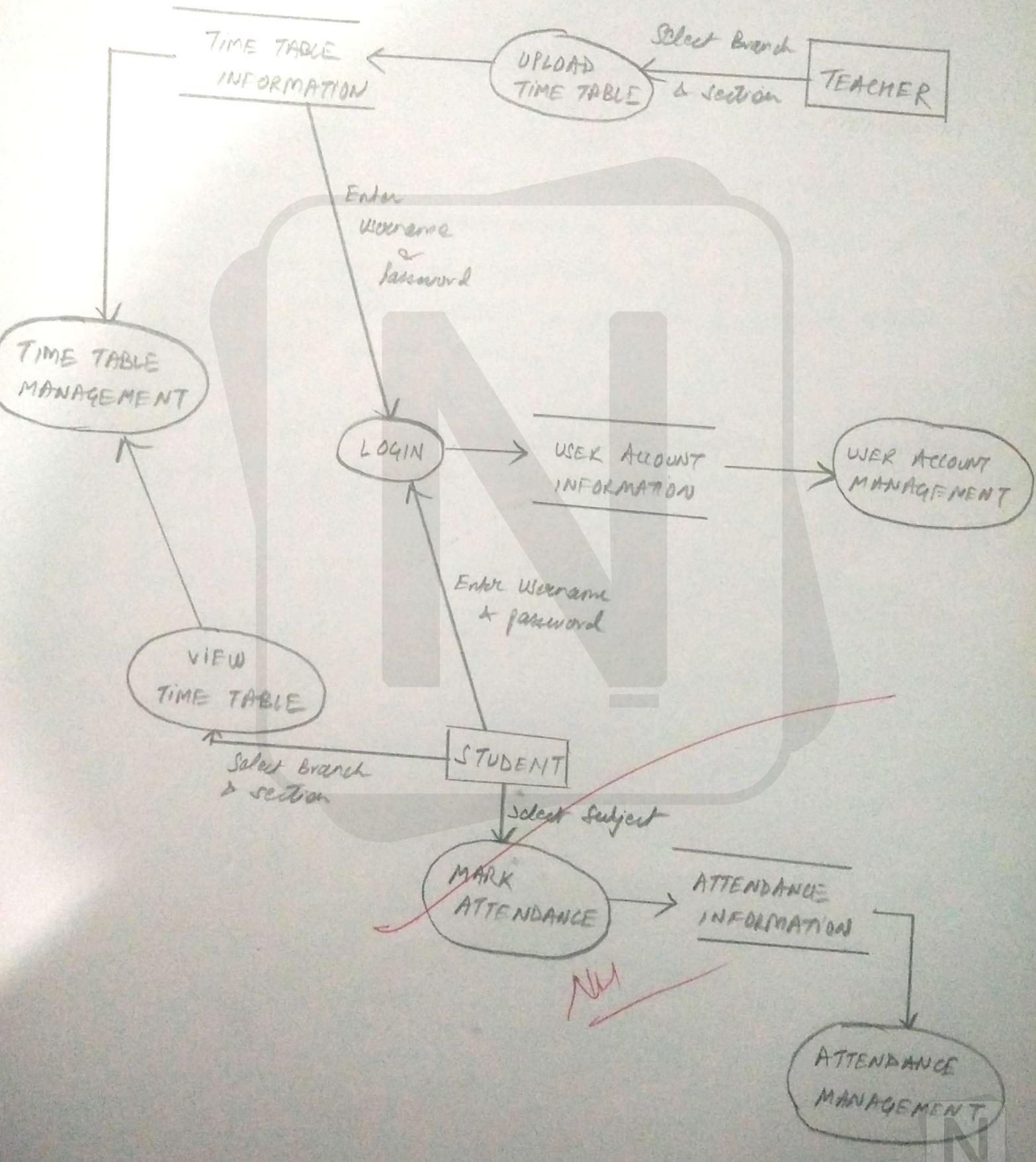
DFD - Level 0



marks attendance for classes attended or missed

NH





EXPERIMENT - 6

Aim: To draw structural view diagram for the system : class diagram & object diagram

Software Requirements: Umbrello Version 4

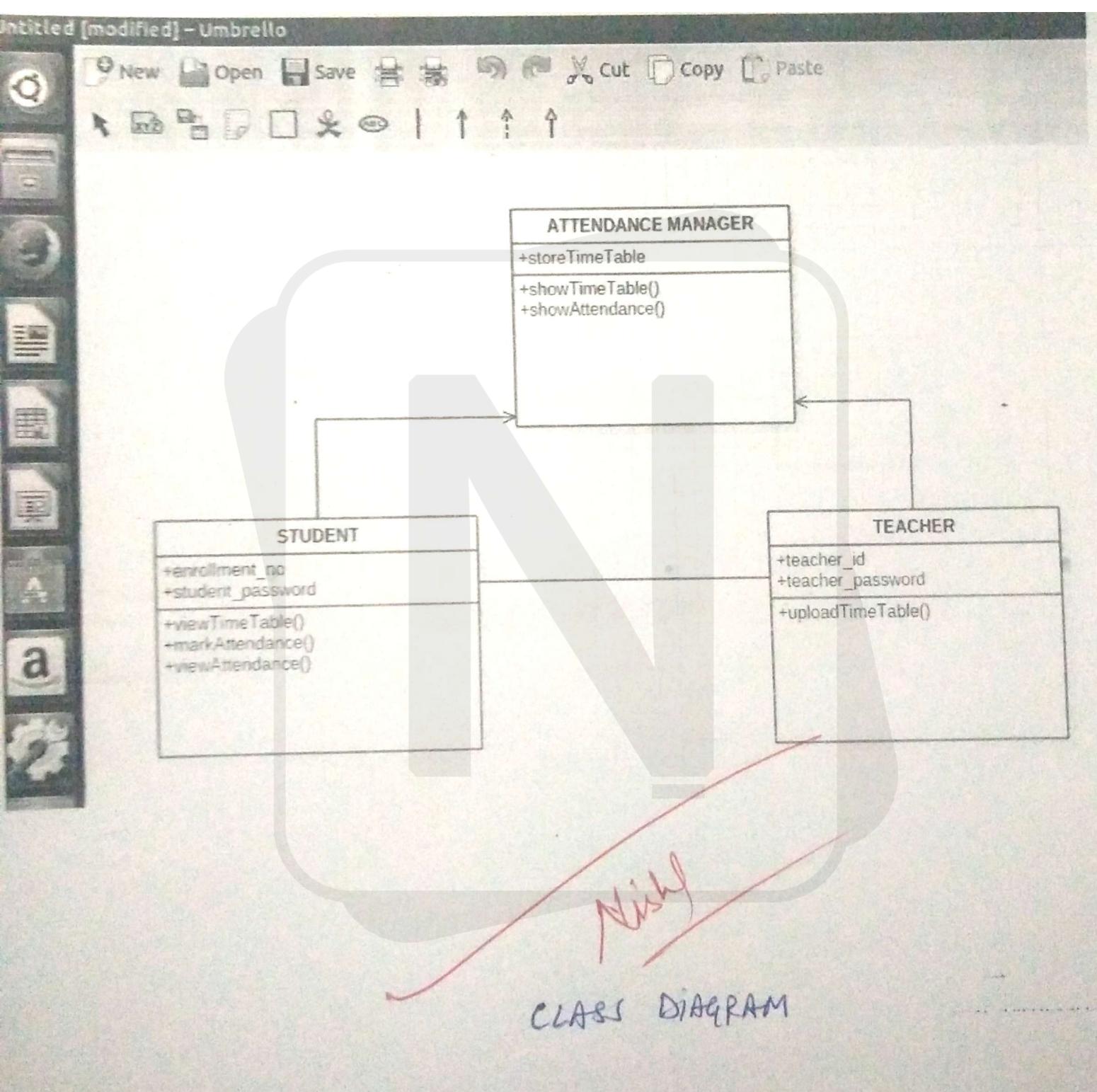
THEORY :

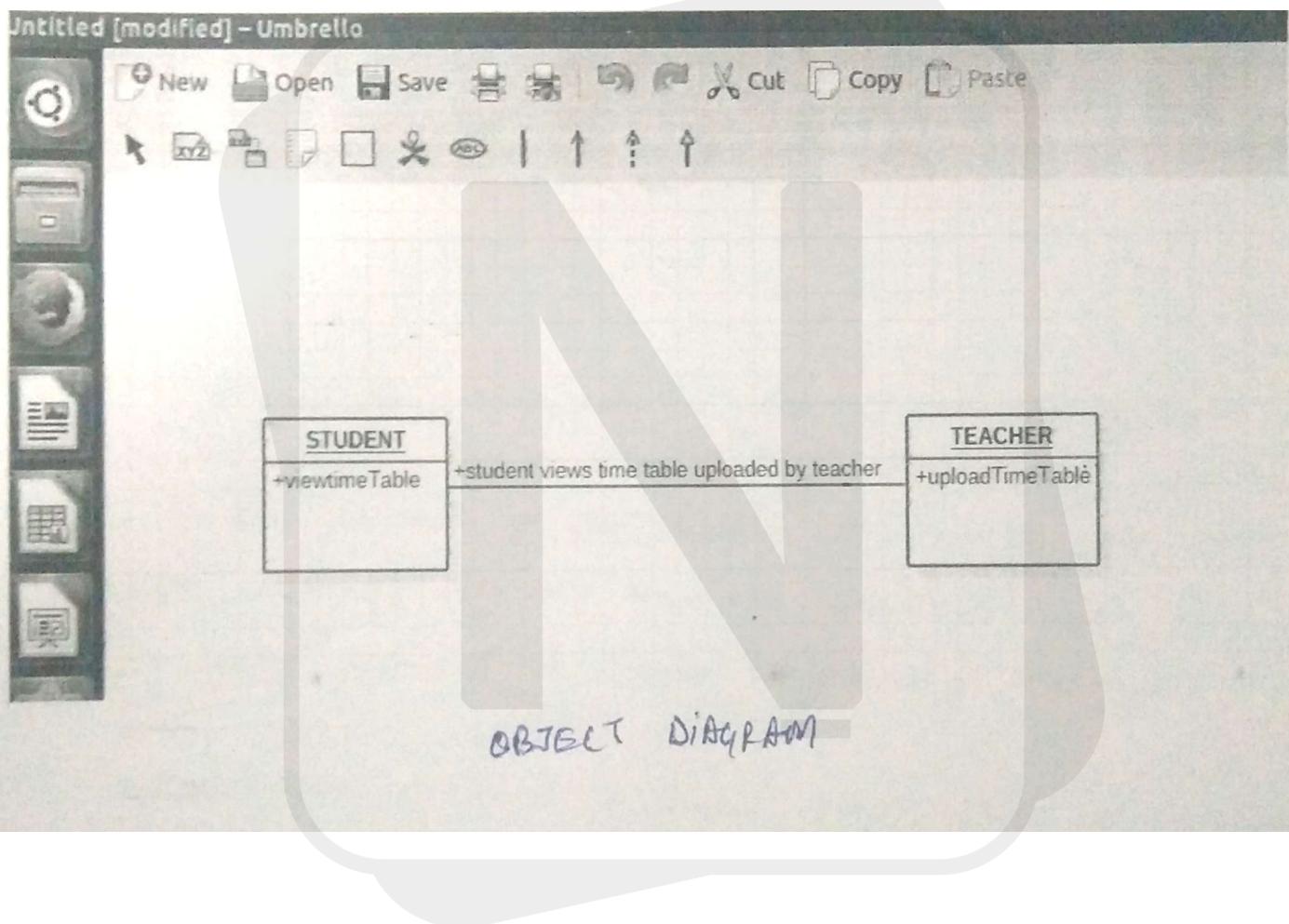
CLASS DIAGRAM: It shows the relationship amongst classes of a particular system. The class diagram is a static diagram. It represents static view of an application.
class Dg. describes the attributes & operations of class & also constraints imposed on the system.

OBJECT Diagrams: are derived from class diagram so object diagrams. They represent instance of class diagram.

NH







EXPERIMENT - 7

Aim: To draw the behavioral view diagram - state chart & activity diagram.

Software requirement: UML version 4

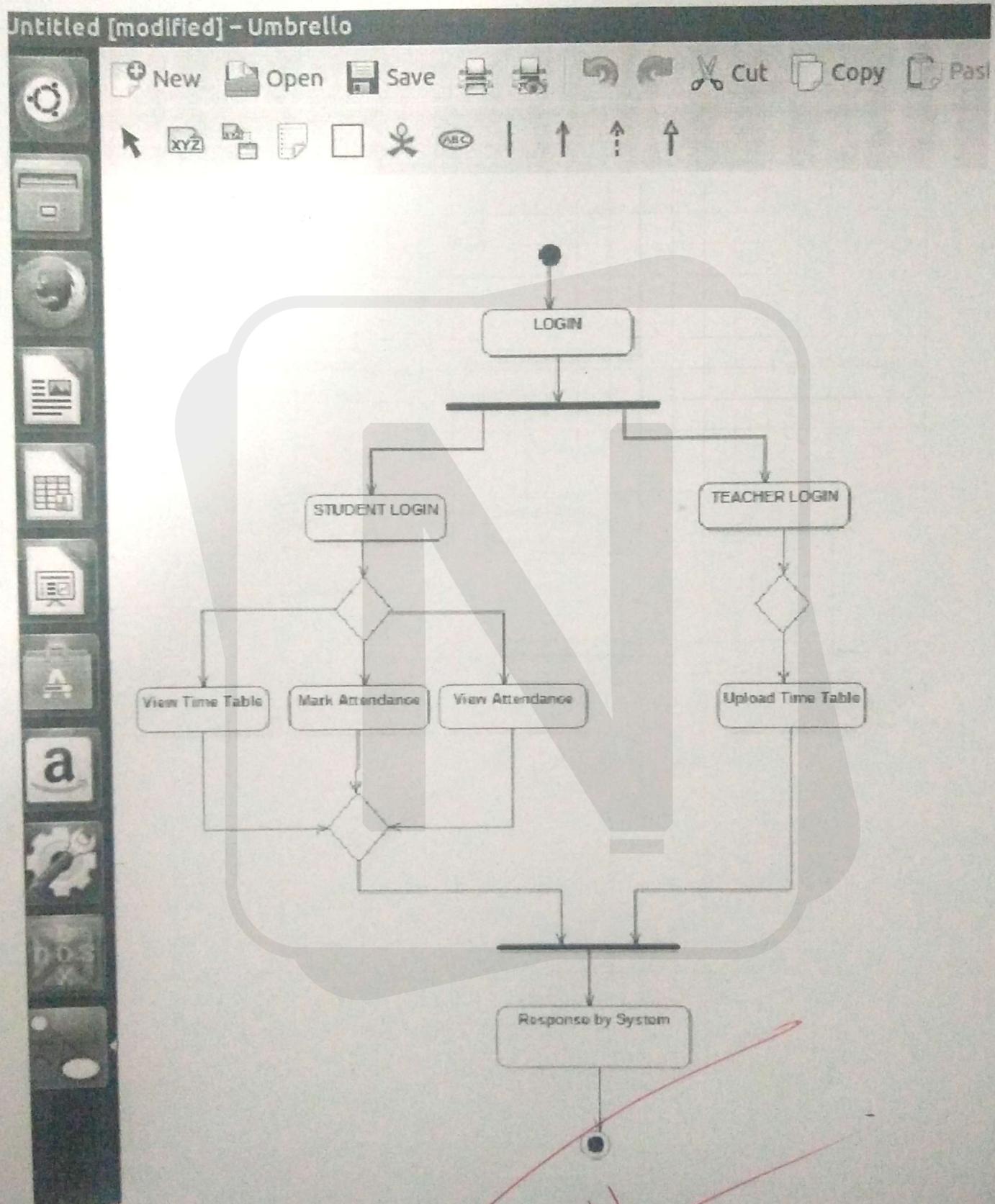
Theory:

Activity Diagram: It is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. It captures the dynamic behaviour of system. They also used the construct the executable system by using forward & reverse engineering techniques.

State Chart Diagram is used to show state space of a given class, the event that cause a transition from one state to another & actions that result from a state change.

Nish
22/10/2016

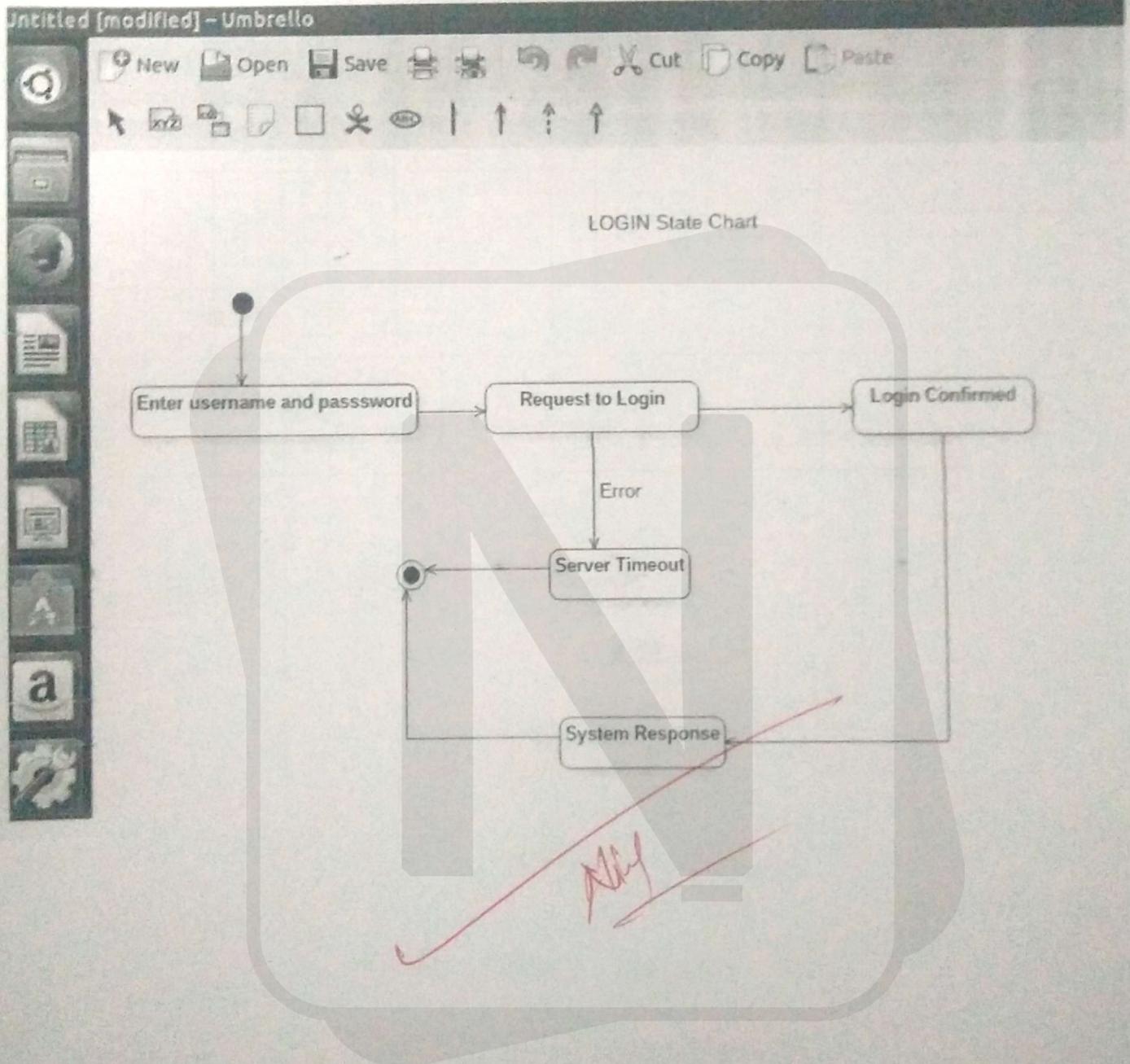


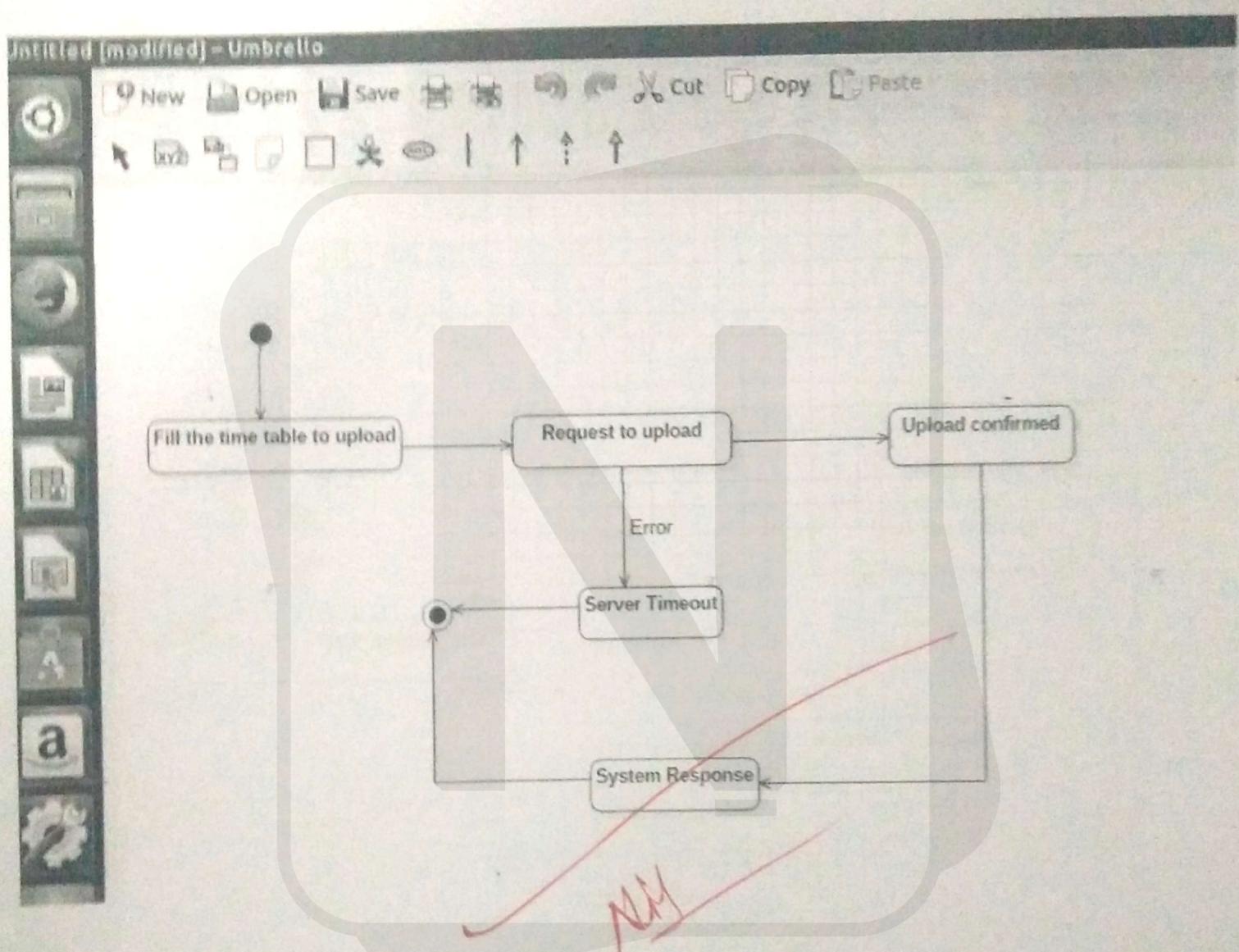


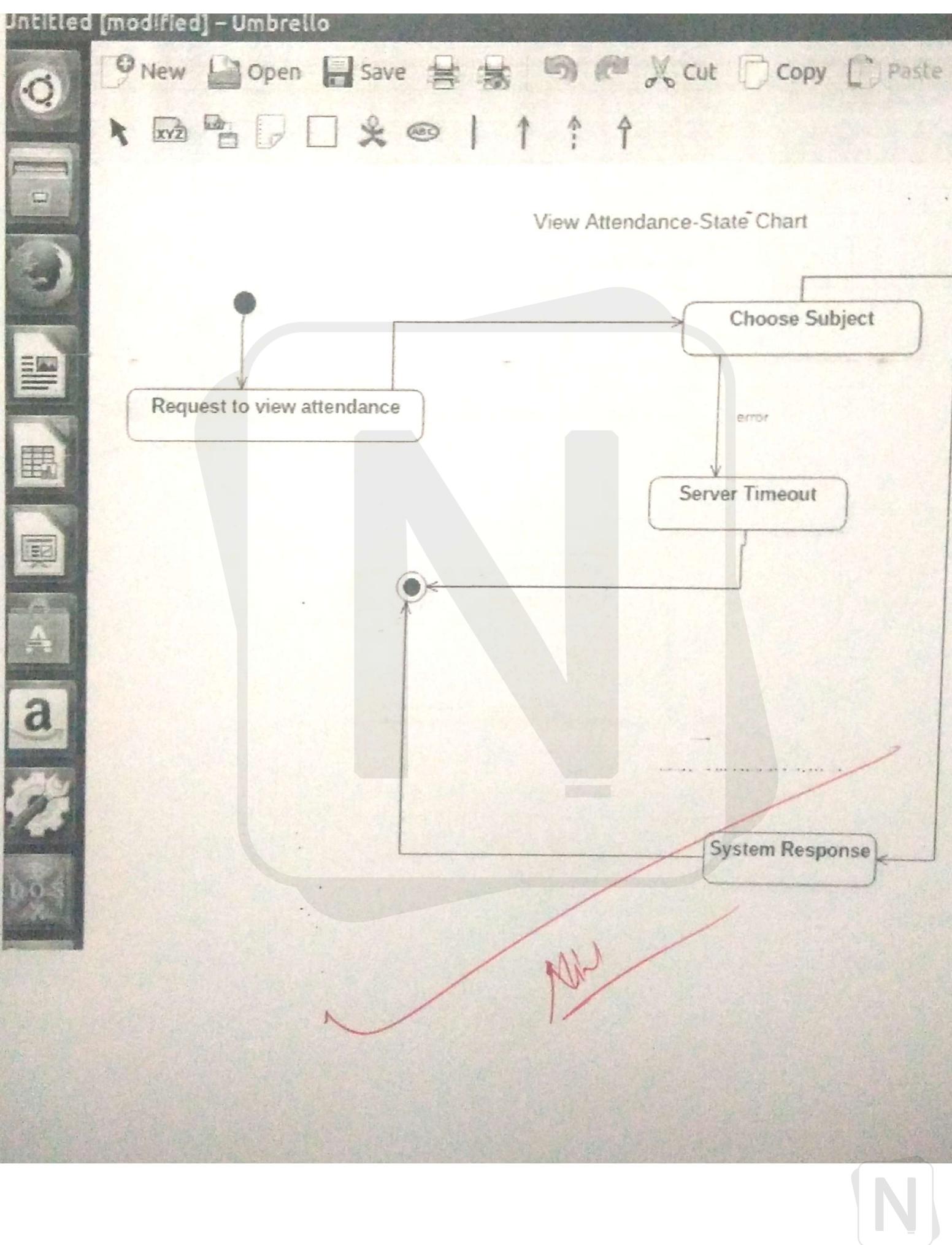
Nish

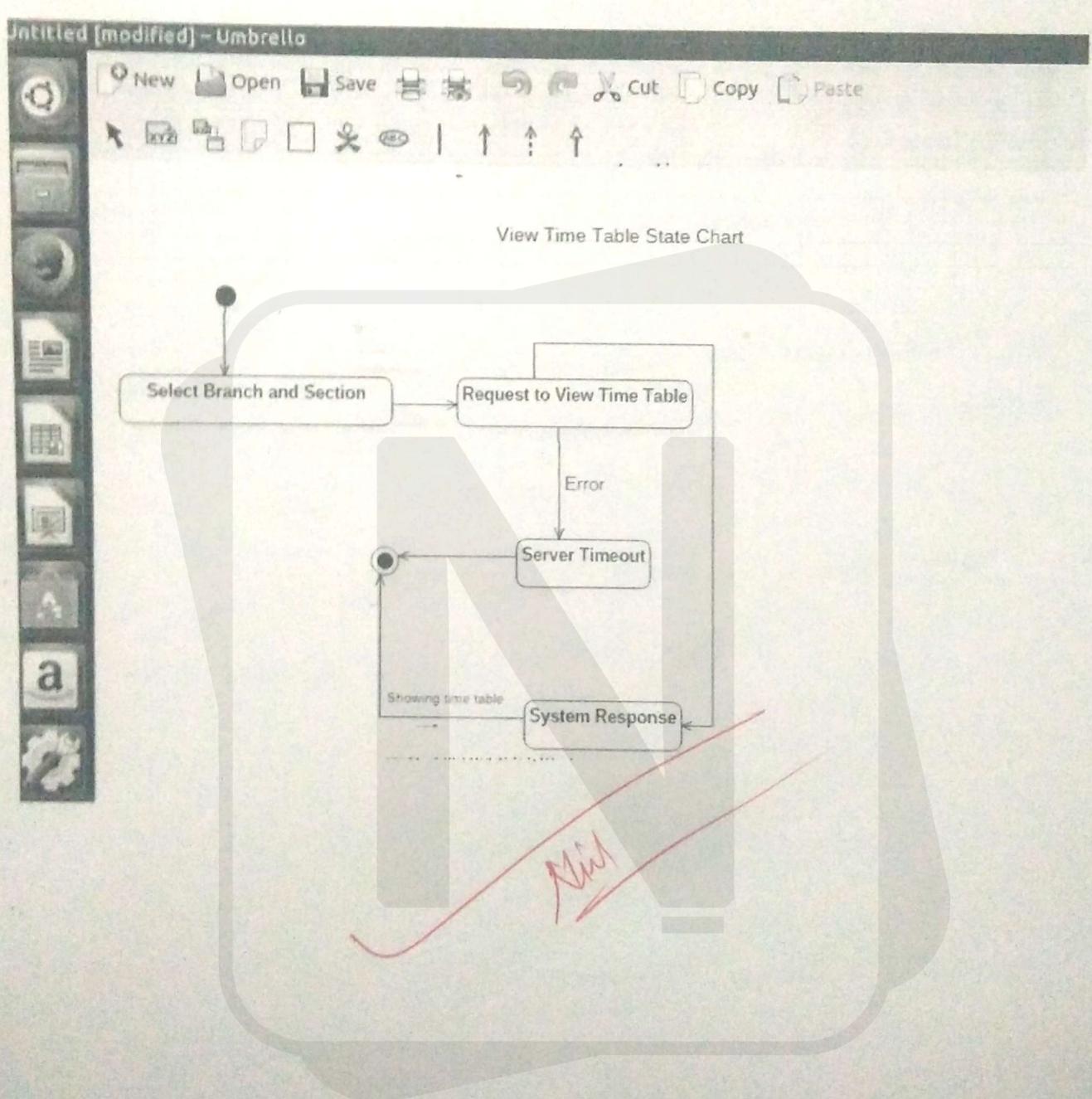
ACTIVITY Diagram











EXPERIMENT - 8

Aim: To perform the behavioral view diagram for the suggested system:
Sequence diagram, collaboration diagram.

REQUIREMENTS: Umbrello version 4

THEORY:

An interaction diagram shows an interaction, consisting of a set of objects & their relationships including the messages that may be dispatched among them. Interaction diagram addresses the dynamic view of a system.

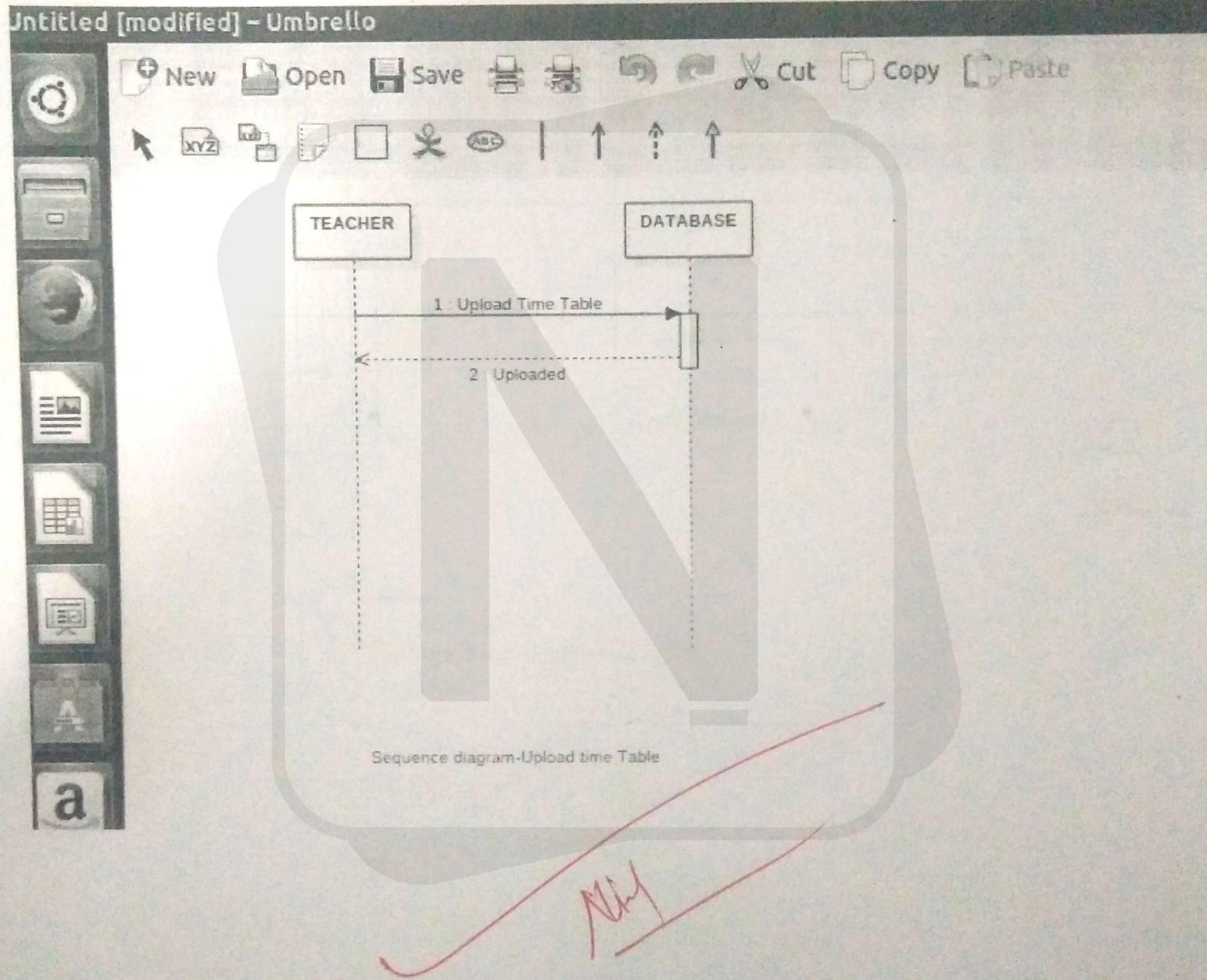
SEQUENCE Diagram: A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.

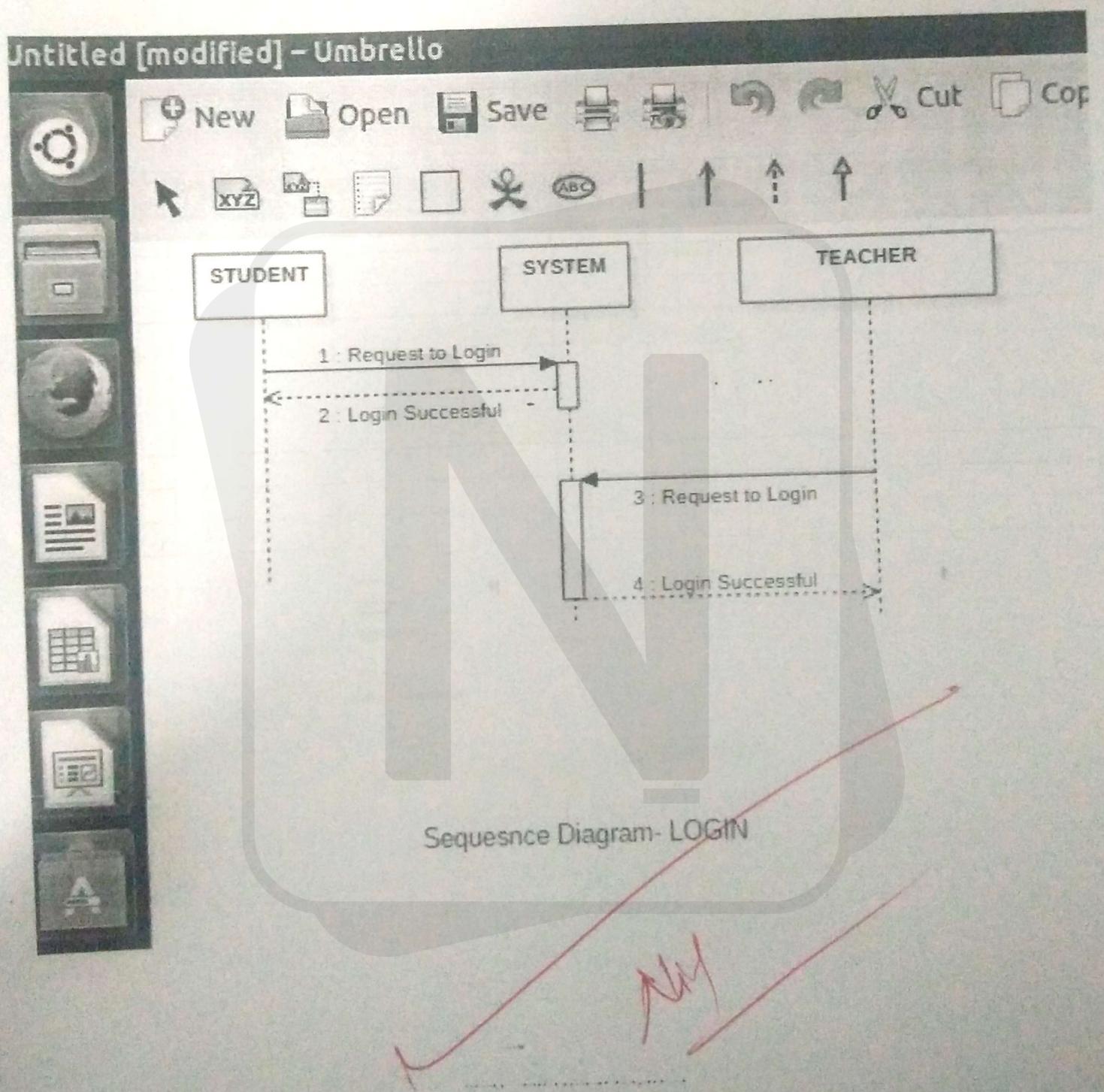
COLLABORATION: - It is an interaction diagram that emphasizes the structural organization of the objects that send & receive messages.

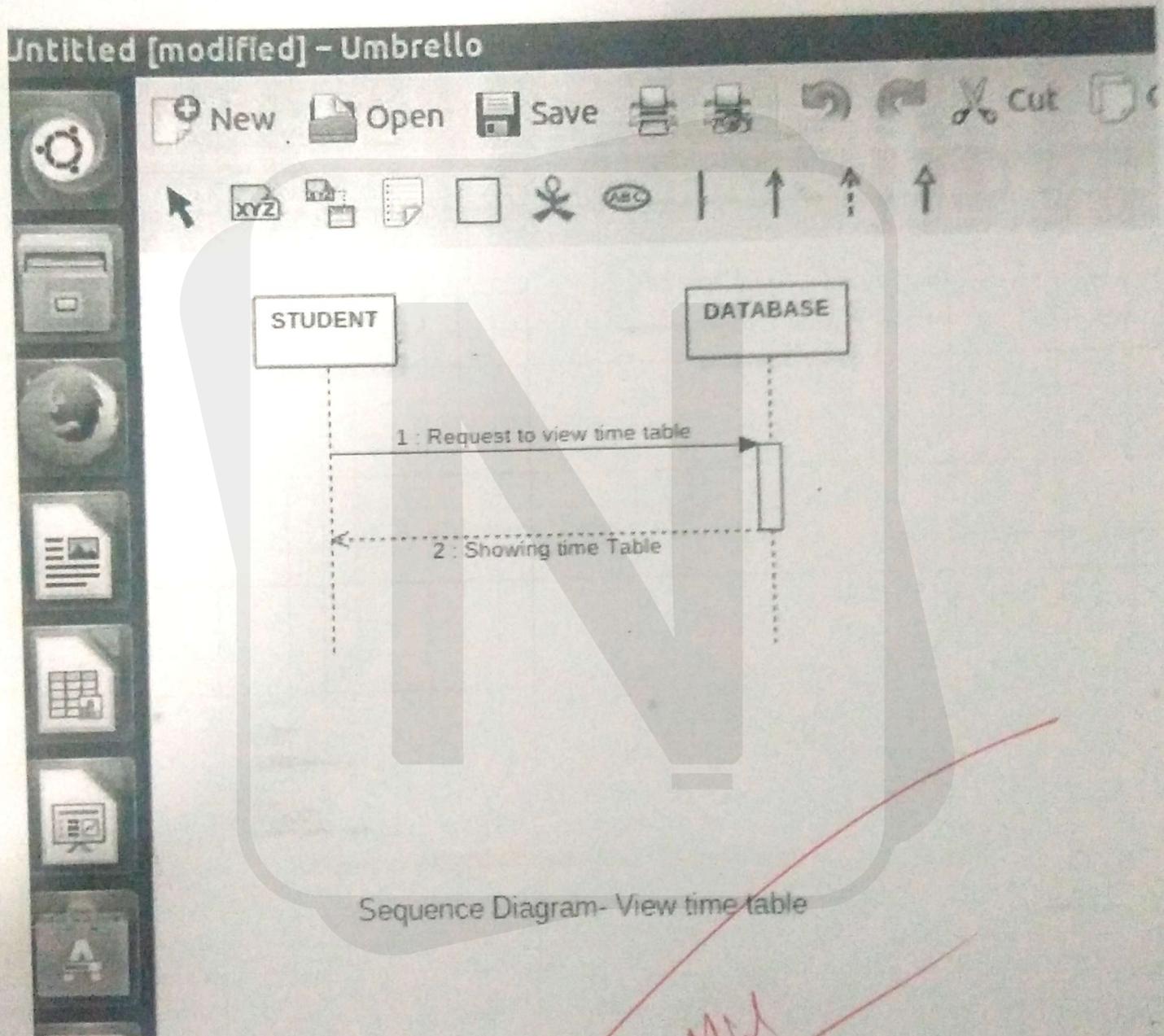
The object types used in this analysis model are entity objects, interface objects & control objects.

NM









EXPERIMENT - 9

Aim: To perform the implementation view diagram : Component diagram for the system

Software Required: Umbrello

Theory

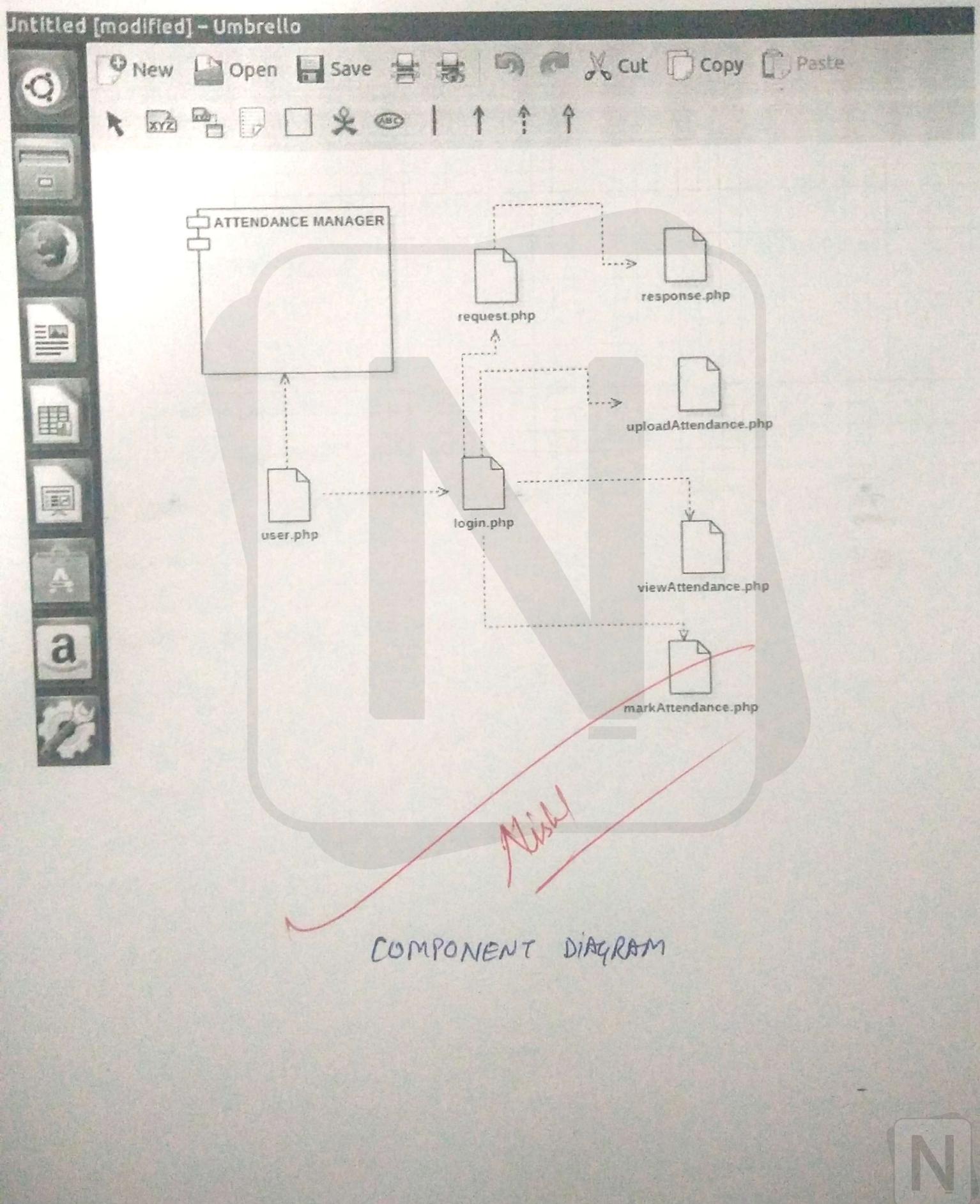
Component diagram are used to model physical aspects of the system. It does not describe functionalities of the system but describe components to make those functionalities.

Subcomponents of the system are:

USER
login
request
response
upload Attendance
view Attendance
mark Attendance

NM
22/10/2016





EXPERIMENT - 10

Aim: To perform the environmental view diagram - deployment diagram for the system.

Software Requirements: Umbraco Version 4

Theory: Deployment Diagrams are used to visualize the topology of physical components of system where software component are deployed.

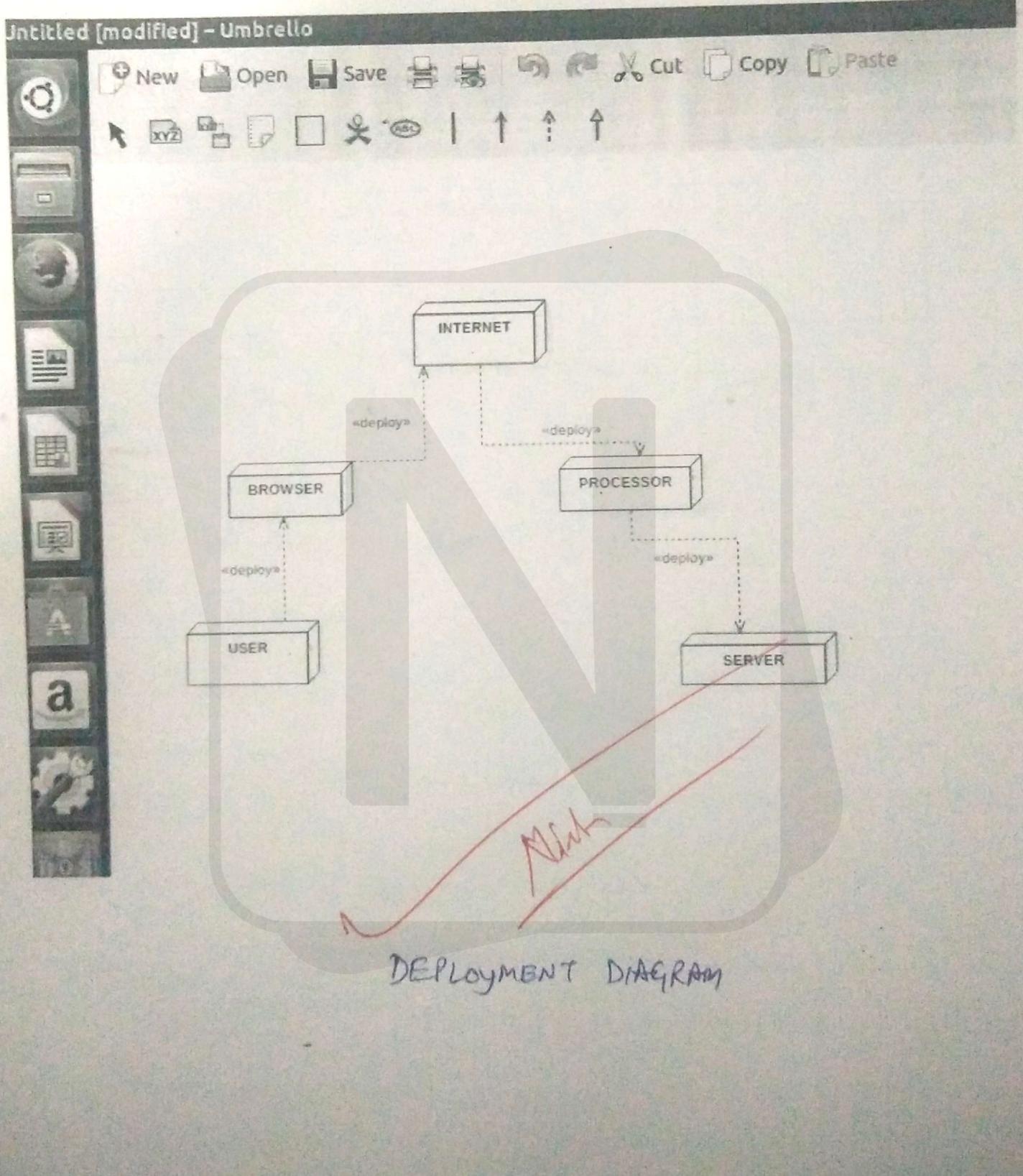
Component diagrams & deployment diagrams are closely related. Component diagram used to describe the components & deployment diagrams shows how they are deployed in hardware.

The system contains various components:

- user
- browser
- internet
- processor
- server

Nish
22/10/2016





Aim: To perform various testing for a sample code of suggested system using Boundary value analysis

Theory: Boundary Value Analysis

Experience shows that test cases that are close to boundary conditions have a higher chance of detecting an error. Boundary condition means, an input value may be on boundary or just above boundary.

According to our requirement ~~username to store as e-mail address~~
 → Email Address is valid if:-

- 1) No. of characters b/w 4 to 254 characters
- 2) Must contain character before @ & after it.

→ Password is valid iff:

- 1) No. of characters b/w 8 to 15 characters
- 2) Must contain a upper-case character, a number & a special character

Output must be valid or invalid

TEST CASE ID	I/P	OUTPUT	
	USERNAME	PASSWORD	
1.	3 Char long 4 Char long 125 Char. long 254 Char long 255 Char long	12 char long 12 char long 12 char long 12 char long 12 Char long	Invalid Valid Valid Valid Invalid
2.	125 char long 125 char long 125 char long 125 char long	7 Char long 8 Char long 15 Char long 16 Char long	Invalid Valid Valid Valid
		Nish 28/10/08	



2. Password
3. Document Credentials
4. Application Form
5. Access Feeds/Details Form

User Outputs :-

1. System Response
2. Other users response
3. error
4. Redirect

External Enquiries :-

1. Fetch Time-Table from Database
2. Modify/Update Time-Table

Internal Logical Files :-

1. Index File
2. Validation Forms
3. Login
4. Register
5. Connector to Database
6. View ~~Fetch~~ Time-Table
7. Error Pages files
8. Response files
9. Repeat files

External interface files :-

1. Connection files
2. Admin files
3. Contact files
4. Redirect to other gateways



Aim: To perform estimation & effort using FP Estimation for chosen system.

THEORY: The software has following functional units

- (1) Inputs : information entering into the system.
 - (a) Login verifier
 - (b) Operating Server
 - (c) Web / Application Server Java web server 2.0
 - (d) Database Server oracle
 - (e) Hard-disk 20GB
 - (f) RAM - 64 MB
 - (g) Developers - 4 required - 6
 - (h) Testers - 6 required - 7
 - Total - 19
- (2) Outputs : Information leaving the system
 - (a) Login
 - (b) Result
 - (c) Printer
 - (d) 5 Monitors
 - Total - 8
- (3) ENQUIRIES - request for instant access to information
 - (a) Homepage
 - (b) ^{view} Search Page
 - (c) Add Page
 - (d) Mark Page



NotesHub.co.in | Download Android App

④ Internal logical files: information held within the system.

- a) Teacher Account
- b) Time-Table information
- c) Attendance information

Total - 3

⑤ External logical files: information held by one other systems that is used by system being analyzed.

- a) Lab Resources

Total - 1

Assuming all complexity adjustments factors & weighting factors are average

Unadjusted Function Point (UFP):-

$$\text{inputs} \rightarrow \text{Total } \times 4 \text{ (Avg. weighting factors)} = 76$$

$$\text{outputs} \rightarrow \text{Total } \times 5 = 40$$

$$\text{enquiries} \rightarrow \text{Total } \times 4 = 16$$

$$\text{internal logical file} \rightarrow \text{Total } \times 10 = 20$$

$$\text{external logical file} = \text{Total } \times 7 = 7$$

$$UFP = 76 + 40 + 16 + 20 + 7$$

$$= 159$$

Complexity Adjustment factor (CAF)

$$CAF = 0.65 + 0.01 \sum F_i$$

$$= 0.65 + 0.01 \times (14 \times 3)$$

$$= 0.65 + 0.42 = 1.07$$

$$\text{Function Point (FP)} = UFP \times CAF$$

$$= 159 \times 1.07$$

$$= 170.13$$

Nisha
25/10/2016

