

# **Final Project:**

## **Follow the instructions mentioned below:**

### **Model**

Select any one model for your project out of existing six models like water fall model, incremental model etc. Also include diagram with your selected model along with proper justification.

Write at least six existing methodologies with diagrams.  
Like water fall, incremental model etc.  
Include diagrams with all these existing methodologies.

Write the project plan with following attributes task name, starting day, closing day and number of day as well gantt chart.

Take the screen shots of project plan and gantt chart and insert into .document file.

### **DFD level 0:**

In DFD level 0 external entities interact with centralized system. There is no separate storage.

<https://www.lucidchart.com/pages/data-flow-diagram>

### **Activity diagram:**

Use the standard notations of activity diagram. Draw activity diagram for each use case with proper title. Get help from the uploaded helping material.

**Note: it's an optional**

### **ERD**

Use the standard notations of ERD.

Use rectangle for entity and oval shape notations for attributes.

There are two models used for ERD

1. Crow's feet model
2. Chen model

Get help from the uploaded helping material. Mention the relationship between entities like one to one, one to many, many to one or many to many.

<https://www.smartdraw.com/entity-relationship-diagram/>

**Sequence Diagram:**

Use the proper name of objects/classes related to title of sequence diagram.

Use “:” sign to differentiate between class and object.

Use methods to exchange information between objects and classes instead of simple text when object forward message and use simple text in return.

Use dotted line arrow for return messages

Draw the sequence diagram for each use case separately with proper title. Get help from the uploaded helping material.

Use the “:” sign to differentiate between class and objects.

Use the vertical rectangle on life line used as activation box.

Use the methods to share the information between classes/objects instead the simple text.

Use the data store class in the end.

<https://www.smartdraw.com/sequence-diagram/>

**Architecture diagram:**

Use the three tier architecture

1. Presentation layer
2. Business layer.
3. Data layer

<http://www.tonymarston.net/php-mysql/3-tier-architecture.html>

**Class diagram:**

Use the standard notations of class diagram.

Divide each class into three parts

1. Class name
2. Attributes
3. Methods

Mention the proper relationship between classes like inheritance, aggregation and composition

<https://sourcemaking.com/uml/modeling-it-systems/structural-view/class-diagram>

**Database design:**

Mention primary key and foreign key when you create table or make relationship between tables.

<http://www.datanamic.com/support/lt-dez005-introduction-db-modeling.html>

[https://www.ntu.edu.sg/home/ehchua/programming/sql/Relational\\_Database\\_Design.html](https://www.ntu.edu.sg/home/ehchua/programming/sql/Relational_Database_Design.html)

**Test cases**

Use the given template of test cases.

Template of TC:

**Test Case Title: ???**

Preconditions	
Actions	
Expected Result	
Test By:	Student VU ID/ Name
Result	Pass/Fail