

**Assignment 3**  
**Deadline – 03/03/2019 before the class (11:59)**  
**Late Submission 10 points penalty**

**TASK 1: Planning Scheduling and Peer Evaluation**

You will make a table like the one you have created for the previous week. The table should focus on **the current assignment only!!**

Assignee Name	Email	Task	Duration (hours)	Dependency	Due date	Evaluation
Jane Doe	Jane123@student.gsu.edu	Class Diagram	4 hours	None	2/2/19	75% Jane – partial contribution Alex did it
Alex Brian (coordinator)	example@gsu.edu	Technical writing	3 hours	Use cases, requirements, diagrams	2/4/19	100%
Michael Jordan	example@gsu.edu	Use Cases	6 hours	None	2/3/19	0% Michael did not do the assigned task Alex did it

**Note : the evaluation is how much the person was given by the rest of the team. For example, here Jane was given 75% for her work by her team**

**TASK2: Revise and Refine your System**

- Based on the feedback and the additional topics covered in class, you are to revise, refine, complete and include your system requirements and system molding with A3. Therefore, you will write an improved version of the Problem Statement you provided in A1. Refined Problem Statement, requirements, class diagram

**TASK3: System Modeling (Analysis)**

**1. Class Diagrams (object Modeling, or structural modeling): (Ch5, 5.3.1, 5.3.2, 5.3.3)**

- Identify objects.
- What are the associations between them?
- What is their multiplicity?
- What are the attributes of the objects?
- What operations are defined on the objects?
- Create system class diagram.

**2. Architecture modeling:**

Identify the architectural design pattern for your system (Ch6: 6.3 and all its subsections). Use MS Visio to draw your chosen architectural design pattern.

**3. Behavioral modeling (Dynamic Modeling):**

Using MS Visio, create a UML sequence diagram to illustrating in more details **two of the major** use cases of your system. One of those two must **correspond to the creation of your system**. Show appropriate lifelines, activations, and message types. You may also use loop, alternative (alt), and optional fragments if needed. You may use your class diagram to identify the objects' names and messages (methods) that you need to develop your sequence diagram.

**Important Note:**

- A sequence diagram is a visual representation of how objects in a system interact. Keep in mind that the reasons to create a sequence diagram is to:
- refine your use case diagram and uses cases (adding missed cases [functionalities]),
- refine your class diagram (adding missed methods [messages]),
- transition from the conceptual model and start thinking about the implementation, which is the most importantly.

## **TASK4: Implementation**

- **Implement the Database Design (Tables, Backend):**
  - Choose the tables related to the **two of the major use cases** of your system
  - Implement the chosen tables using the selected database management system (MySQL, MS-SQL server, Oracle, etc.)
    - Creating tables using SQL scripts makes it easy to create the necessary tables on another computer: Development Environment, Test Environment, Production Environment; Deployment

**Note:** These are some free hosting database management systems:

  - [Amazon RDS](#) (Relational Database Service) has a free usage tier *for 12 months* to run a Micro DB instance with 20GB of storage and 10 million I/Os
  - [Google Cloud Firestore / Datastore](#) is document-store with a permanent free tier of 1GB storage, 50k reads and 20k writes per day.
  - [MongoDB Atlas](#) has a free tier with a 3-node replica set and 512MB storage.
  - [Redis Labs](#) offers a 30MB Redis instance for free.
  - [Heroku Postgres](#) has a free tier limited to 10k individual rows of data.
- **Implement the Class Diagram Design (Frontend and Logic): (develop/write ode)**
  - Get an overview of the software frameworks or platforms, programming languages, host, etc.
  - Install the necessary Software
  - Start creating the **main parts (two of the major use cases)** for your application (both Frontend, logic (CODE), and GUI).
  - Create a good structure for your code.
  - Test that you are able to communicate with the Database (data go from Frontend [GUI] to Backend [Tables]).
  - A description of how to compile and run your code
  - You should not instruct us to compile and execute your code in any particular IDE.
  - We recommend writing your own build script or generating one in an IDE (ant, mvn, etc.).

- Or, if you chose to host your system in the cloud, make sure to provide the host link, user name and password in the section of “description for how to compile your code”

**Note:** These are some Web Based Frameworks and IDE tools:

#### Backend frameworks

- Spring (Java)
- ASP.NET (C#)
- Django (Python)
- Laravel (PHP)
- Express (Node.js)
- Ruby on rails

#### Frontend Javascript Frameworks

- Angular (Java Script, by Google)
- React
- Vue
- Ember
- Backbone

#### Integrated Development Environment (IDE) Software

- Visual Studio
- IntelliJ IDEA
- Xcode
- Eclipse
- WebStorm
- PhpStorm
- NetBeans
- AWS Cloud9

## TASK5: Testing

- Develop a collection of requirements based tests (functionality testing):
  - Use the following steps to develop your test cases:
    - **Identify features** (functionality) related to the **main two of the major use cases**
      - E.g: insert (int N, list A)
    - **Partition inputs** into equivalence classes: (for each feature)
      - E.g: int **N** is 5-digit integer between 10000 and 99999,
      - One possible partition is: <10000, 10000-99999, >100000)
      - E.g: **list A** is a list of length 1-10.
      - One possible partition is: empty list, list of length 1, list of length 2-10, list of length > 10
    - **Test Specification:** (for each feature)
      - E.g: insert(< 10000, empty list)
    - **Test cases:**
      - E.g: insert(50000, {})
- When you are giving test specification clearly mention the use case that the test specification is being given for.

## TASK 6 : Communication and Collaboration

### Github :

- Using the project you created for A1, identify a new To do; In Progress and Done columns for A2 Under these columns, create your own cards and have the assigned tasks and their status (in progress and done) written there as a list.
- Submit a screen shot of your project page

## Report:

### 1. Report Format:

- First page has (individual page): the project title, group name, Group number, members' names, semester, and year.
- Font size 12, Font type is times new roman, single space between lines.
- All paragraphs must Text Justified.

- Pages are numbered
  - Diagrams and tables must be centered.
2. **Report Sections:** (remember to revise and refine your system)
- Planning and Scheduling
  - Problem Statement
  - Requirements
    - User Requirements
    - System Requirements
  - System Modeling
    - Class Diagrams
    - Database specification and analysis
    - Architecture modeling
    - Behavioral modeling
  - Implementation
  - Testing
  - Appendix: Have all the screen shots and links in this section
3. **Report Submission:** you will submit the report electronically as a PDF file:
- The file you submit should be named coordinatorName Groupname.pdf
  - Only the team coordinator should submit the report
    - Individuals will only get points if they are in a group that has been submitted.
  - You (team coordinator) will also print out a copy of the report and submit it before the class..