**Assignment #2**

Back-End Development

*GROUP WORK*

*Due Date: See Instructional Plan*

**Questions:**

*Object Oriented Programming (PHP)*

1. [4 marks] Use *sessions* for authentication of username and password from the User Login page of your Project Website. Create an *authentication.php* page to verify credentials and *member.php* page containing information for authorized users only.
   * *For now you may hard code for a single username and password*
   * *Later we will allow for access by more than one user and use databases to store credentials*
2. [4 marks] In the Project, you will be dealing with a number of objects including: the elevator car, floor nodes, and a number of other sensors. Design a UML Class diagram that illustrates the public methods for each class and their interactions.
3. [4 marks] Implement the class definitions (each class in its own file) for each of these classes and create a PHP file that creates object instantiations for each class. Remember to implement the standard conventions for classes, use encapsulation as well as static properties and methods where appropriate.
4. [8 marks] Consider where you might need to use Inheritance in the Project. For example, The Elevator, three call Buttons and distance Sensor could be children of a Node class and implement various interfaces depending on input. Draw a UML diagram to illustrate the relationship between the classes / interfaces / traits you have identified. Implement (code) them and verify/test that your code works. Include this code and a screenshot of it working in your submission.
5. [8 marks] An important component of a robust program is handling the unexpected. This could range from unexpected user input that is outside an expected range (i.e. requesting the elevator to go to the 4th floor when there are only 3) to communication errors (i.e. No connection). Using exceptions can also help you to debug software when interfacing between multiple systems (e.g. CAN bus, intranet, etc.). Consider all the possible exceptions you might need to throw in the project (e.g. invalid input to node, communication error, and create new exception classes that extend from the Exception class to handle them (as demonstrated in class [slide ~15]). The exceptions can be thrown from various locations in your code and handled within the parent/calling function in a single *try … catch* block. Include this code and a screenshot of it working in your submission.

*Databases (PHP)*

1. [4 marks] In your project groups create a database for the elevator network and create several tables for the various subnetworks (CAN, internet, etc). Create a primary key that identifies each node uniquely (e.g. nodeID). Create at least one child table (i.e. for CAN network components) with at least one foreign key that references an auto incremented primary key in the parent (referenced table). Make at least one field a unique key and one field an index.
2. [8 marks] Create a function for updating any field in a given database (except the Primary key). You may use a database you have developed for the Project.
3. [4 marks] Modify your function in question 7 to use transactions. Be sure to throw exceptions when encountering errors or unexpected input.
4. [4 marks] Add functionality to your members.php page in the project to ***insert new*** content and ***display*** content from your elevatorNetwork table.
5. [8 marks] Add functionality to your members.php page in the project to ***modify*** and ***delete*** the content of an existing row in your elevatorNetwork table.

**Marking Scheme:** See Rubric in eConestoga

*PLEASE INCLUDE A SCREEN CAPTURE WHEN POSSIBLE IN YOUR SUBMISSION*