

---

# 2018 U.S. Information Technology Collegiate Conference Systems Analysis and Design Competition

---

**Do not put your name(s) or your school's name on anything that you submit.  
Doing so will result in disqualification of your team.  
The only identifying information you should use is your team number.**

## **PROBLEM STATEMENT: Purchase Order Processing System**

The hypothetical organization needs an automated purchase order processing system to help speed up their internal process. In the current practice, a purchase order form is needed for anyone who wish to pay outside parties using organizational accounts. There are many organizational accounts currently existing. Each account will have at least one, sometimes more personnel in charge of approving such transactions. In order to automate the whole process and put the physical forms online, the organization is seeking a new purchase order processing (POP) system. The POP system must be able to accomplish the following requirements:

1. The system will allow different people to login as different roles. There will be three main roles: PO Requestor, Department Admin, and Accounts Payable Admin.
2. The PO requestor will fill in the information regarding the vendor, itemized articles, accounts will be used to fulfill the request, and total amount of funds requested.
3. The vendor information will be pre-populated in the system through a separate form by the accounts payable admins. Information regarding the vendor include name, address, city, state, zip code, and phone number. A vendor has to be approved by accounts payable admin before it can be used on the purchase order form.
4. The PO requestor will fill in information regarding the purchase order. Each time the requestor start a new purchase order, a PO number will be automatically assigned. The purchase order information includes: department name, contact information, requestor name, account information, and item information.
5. Account information includes account description, account number, and amount will be used by that account.
6. Item information includes business purpose of itemized articles, unit, quantity, and total.
7. The requestor can also specify different processing options. They can request to pay immediately, to re-pay, or encumber only. If they request to pay immediately, an itemized invoice must be attached with the request as well.
8. The requester will also choose one of the two check disbursement options: by mail or pick-up at cashier.
9. When the requestor finishes all required fields on the purchase order form, the system will notify the accounts payable admin to review and approve the request.

10. When the accounts payable admin log in to the system, they would be able to see any pending requests, approved requests, and denied requests.
11. The system should have all security mechanisms necessary to protect the data and information inside. For example, measures need to be taken to prevent common SQL injection attacks to the database.
12. The system should also be able to provide a comprehensive report to the accounts payable admins if needed.
13. The user accounts can be set up by department administration.

**YOUR ASSIGNMENT IS** to use only **one** technique (either Object Oriented **OR** Structured/Traditional Technique) to specify how the system should operate. If you use a structured technique you must specify the flow of data inside the system. If you use an OO technique then you must specify the classes inside the system and how they are used in order to achieve the system's objectives.

### **WHAT TO TURN IN:**

If you are using the **structured/traditional approach**, then you are expected to turn in the following:

1. A Context Diagram.
2. A level 0 (zero) Dataflow Diagram.
3. A Level 1 DFD for each one of the processes that you identified in your Level 0 System DFD.
4. Process descriptions for the processes contained in your DFD's.
5. An Entity Relationship Diagram (ERD) showing the 3rd Normal Form Database that will support the system you designed.
6. Prototype with Windows Forms and/or Web Pages.

If you are using **an Object-Oriented approach**, then you are expected to turn-in the following:

1. Use-case Diagrams.
2. Use-case Descriptions.
3. Sequence and/or Activity Diagrams.
4. A Class Diagram (for objects in persistence storage).
5. State machine diagrams.
6. Prototype with Windows Forms and/or Web Pages.

For creating models, use your own business modeling software. This could include any CASE, I-CASE or other model-based development product.

The prototype must be developed based on your models. It does not have to be fully implemented; however, a system design that provides mocked up screens with window form/web page interaction will be considered in the overall grading. The screens can be created using any graphical drawing software (such as Microsoft Paint or Photoshop) or you can take screen shots from development tools (such as Microsoft Visual Studio, Access, or Eclipse). Given the time

limit of the contest, handwritten mock-ups are allowed; however, the screens created by computer software will be given better grades.

When submit your work, make sure you submit **one pdf file through the contest server**. Do not write any information about yourself or your college. **ONLY** write your **team number** on your submission. If your submission contains any personal information, your team will be disqualified.

**Contest Evaluation**

The judges will use the following categories in evaluating your team solution. The models that your team is required to develop depend on which methodology is selected.

NOTE: Competitors are expected to utilize ONE and ONLY ONE Analysis and Design approach. Using a combination of components from both the Structured/Information Engineering approach and the Object-Oriented approach should be avoided.

	<b>%</b>	<b>Structured/Info. Engineering</b>	<b>Object-Oriented Approach</b>
<b>Information Flow</b>	<b>40</b>	Decomposition, DFD's, Dependency and Process Action Diagrams.	Use Cases, Sequence and/or Activity Diagrams
<b>Information Structure</b>	<b>40</b>	Entity Relationship Diagrams (ERD's) and Data Constraints	Class Diagrams (for objects in persistence storage) and State Charts (State Machine Diagrams)
<b>Prototyping</b>	<b>20</b>	Windows, Screens and/or Web Pages	Windows, Screens and/or Web Pages
<b>Overall</b>	<b>100</b>		