# Essay on Object-Oriented Information Systems

## What?

Software development using the object-oriented paradigm is always a process. This process is always dynamic and also allows for greater flexibility through its life cycle. This course in my overview it's trying to achieve knowledge on this process through a set of units that are all polarized towards the same objective. Software development can be overwhelming when the due process is not followed which then leads to confusion, errors, and time wastage. The software development process cannot fall short of requirements identification, requirement analysis, designing the solution using a modeling language, implementing the design using the appropriate language, and lastly reviewing the design and the implementation to make constant adjustments if needed. The course seems to set us on this path which is a great way to handle software development challenges later in our careers.

The goal of this course will be measured on whether I understand the process immaculately to work with it on my future projects in my career development. My role on this was to get as much knowledge on this as possible by going through the entire unit collection which sometimes required a hands-on experience model to get the best of it. By going through all the units I was aware of the impact they will bring in my understanding of the software development life cycle and that I would be able to apply this knowledge when required to develop software. I was also aware my mastering this process and making it a routine, will improve my efficiency, software quality and also being able to avoid confusion and complexity involved during the development of software.

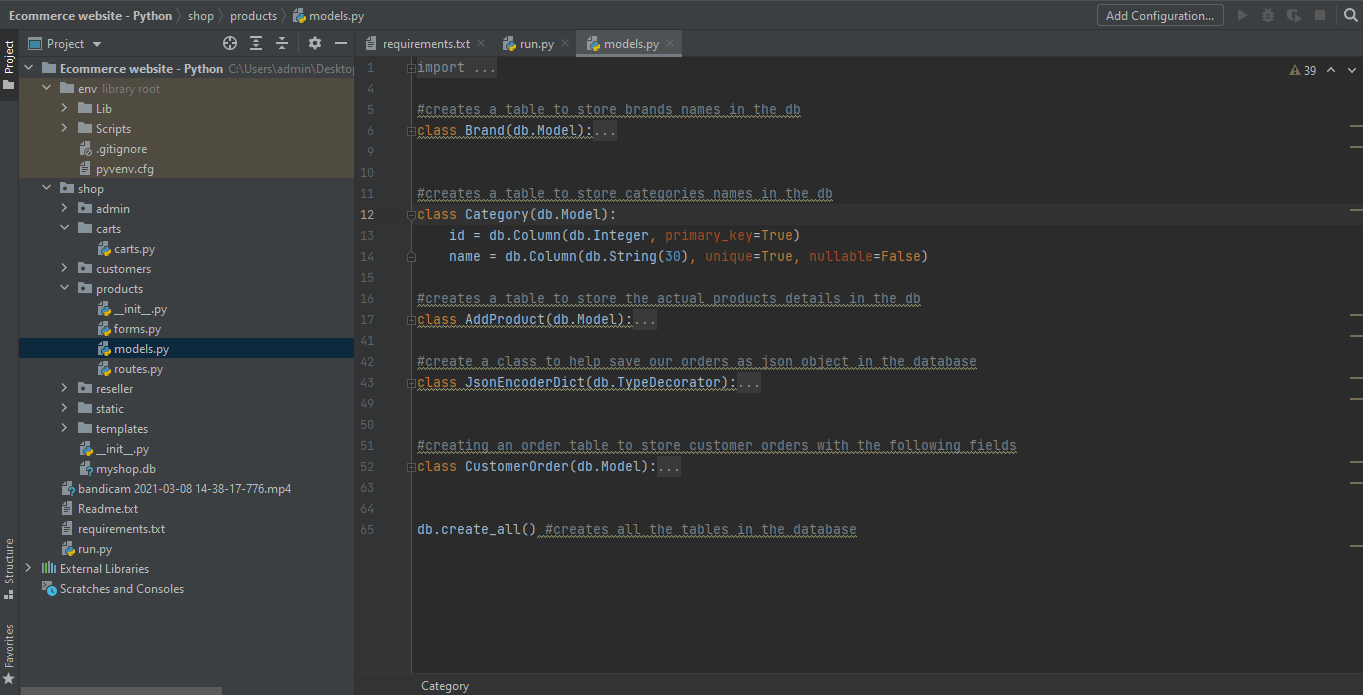
To get the most out of this course, I was deeply buried in achieving the short-term goals of this course which aggregated to the long-term goal of the course. Just to go through the process, the first goal was to learn how to design a database. This was achieved by investigating the core elements of a database for this I was supposed to develop knowledge and skills to apply the database design principles to my projects. Then learn how to design a database that can satisfy the third normal form by first learning how to model database requirements through a modeling language such as UML. When modeling a database, I learned there are various types of modeling a database to represent different aspects of the database. It was also clear to me that there are just two domains in which we could characterize a model that is structural models and behavioral models. I used structural models to model the structure of the database design using a class diagram. The class diagram I designed was supposed to model the entities the database will be stored together with their attributes. The entities are objects representing real-life objects in the domain of e-commerce stores. This blueprint would then act as the guide upon which the user interface is based for users to be able to populate the database with data.

Modeling a database involved a little bit of critical thinking for me and it required that I anticipate what data will be required and develop a requirements list of which from the list is where I formulate the main objects which acted as classes in the class diagram. This was a great opportunity to get acquitted on what the software I would be developing was all about. Modeling the database is not enough to warrant an effective database there is an important technique that I learned in this endeavor that would be normalization which can be used to make any database effective. I was able to check to validate my design all the time to make sure they conform to the rules of the third normal form database which includes avoiding data redundancy, reduce data anomalies, ensure referential integrity, and lastly being able to simplify the data management.

Practice makes perfect so being said, the second goal was hands-on with the database design which I was supposed to apply the modeling designing knowledge gained from the first goal. In doing so I was supposed to identify the key considerations when determining tables and columns and also investigate the key elements of the entity-relationship diagram. To achieve the goals outlined, I developed a database structure for a given scenario using an object-oriented paradigm. The structure of the database was an entity-relationship diagram which is also one of the techniques used in designing object-oriented databases. Entity-relationship modeling was all about modeling the various objects of the database to store in which every entity represented a real-life object like for example a product and user. Some entities relate to other entities which for example a user account, for the case of an e-commerce database may have customers and resellers. The entity relationships model is used to show these relationships between an account to that of a reseller account or a customer account. Doing this on paper gave me an idea of how the database will be implemented and how I will be able to join different tables depending on the relationships modeled on the entity-relationship diagram.

Having learned all the above designing prerequisites and have all design models on paper, the third goal was to implement a database using SQL. It was clear that we would be developing a relational database and for any relation database, SQL is the query language of choice for creating and managing databases. The goals of this topic were to be acquitted with the core concepts of SQL, get to understand the different types of queries and their uses then also without forgetting the methods to manage a database. Through the knowledge learned I was able to implement an SQL database while also creating the appropriate queries to manipulate data from the database. The queries were fundamentally about the creation of new data on the database, deleting data from the database, updating existing data from the database, and lastly reading the data from the database. On the implementation of the design models, the entities formed the table of the databases while the attributes formed the columns of the database. By implementing an SQL database at the level it was in preparation for the development of the database in my final project.

SQL has widely used in any modern information system a profound knowledge of how to work with SQL can go a long way to make sure I can develop efficient and effective databases for data storage. The fourth was aimed at further working with SQL. The learning goals for this unit were to implement a database design and while learning how to design appropriate queries to interact with the database more efficient way. By the end of this unit, I was able to create a table structure based on requirement specifications. I was able to interpret specifications and map them appropriately to form a table that can hold various attributes about the specification. Again, I was able to gain further knowledge by working with SQL and databases. This involved optimization of queries to maximize efficiency and cleaner code, which as a result led to faster database data fetching from the database. Implementing a database using SQL involved first initializing and an empty database to store data then creating tables on the database that is the table was used t store similar collections of data and the attributes of the data formed the columns of the table while adding relations between table attribute like primary keys and foreign keys. What follows is to insert data into the database and then you can read, write and delete the data through SQL queries.

The final goal of the course was to help me to put together the knowledge acquired during the previous goals achieved and successfully implement an e-commerce website. The first step to this, I critically thought of the requirements to include in the database, I knew by developing the requirements would form the basis upon which our database would be designed on. So I was able to anticipate the requirements that I needed. Based on the requirements I formulated an entity-relation diagram which I used to model the database requirements to form a clear view of how the objects will relate to the database once implemented. Once the design was ready, I used SQLalchemy to implement the database. Since I was working on a pythonic environment, working with SQLAlchemy seemed intuitive because it allowed me to create, read, update and delete from an SQL database with ease with python-like structure queries which on the backed, the SQLAlchemy queries are translated to real SQL queries that a relational database can understand. Using SQLAlchemy felt cool because it didn’t feel like I was using two programming languages to archive database interactions but rather a python environment and easy-to-learn structures. The files responsible for database interactions are known as models and a model defines classes together with class attributes which are then translated in the relational database as tables and columns respectively. See the picture below for an illustration of this.

Once I had the database set up for the online store, I used the requirements developed earlier to formulate a class diagram for the front end of the application. The classes were real objects that users will interact with like register and login which will help the user to register and login into the system respectively. I felt the need to explore one more UML diagram to model the behavior of certain state events. So I used a state machine diagram also to model states of the various objects/entities on certain inputs. This was important because I was able to anticipate the behavior of entities at certain points and call for necessary events to happen on the frontend or backend. The last important UML for the front end that I felt I needed to model was the Interaction Overview Design to visualize the flow of control from one event o the other on the system which led to a deep understanding of the inner working system components functioning. In the spirit of designing first before implementation, from the class diagram designed for the fronted, I went further to create a wireframe to prototype the front end of the e-commerce web app. By doing this, just like designing using UML on the backend, alleviates the complexities involved and I was able to spend my time on developing functionalities on the system rather than the look and feel of the application because this was already done using the prototype design using the wireframe, hence reducing the amount of time required to implement the e-commerce application.

Throughout the development cycle of the e-commerce store, I was able to take advantage of the knowledge learned in those previous units which as a result led to efficiently designing and implementation of an e-commerce website where the end product was a culmination of the processes to produce something users can interact with and store data in the database effectively and efficiently. Just to mention the e-commerce application was implemented using flask a lightweight python web framework for web development.

## So what?

This endeavor has mainly been fruitful because it led me down a path to be not only an applications developer who not only develops applications starting from coding something which I used to do before but who plans, identifies requirements, and designs the aspects of the application before it goes down to implementation. Doing this helps in alleviating the complexity by also subdividing the task into various sub-tasks which in the end achieve a highly functional software application. Going through the hands-on task from the course has given me a new perspective on how to approach software development whereby first I identify the database requirements that the software application needs to accomplish, then second design the implementation of the requirements on paper using a modeling language like UML and lastly implement the software using the modeling diagrams designed.

By the end of the implementation of the e-commerce store, it was clear to me this was indeed a professional approach to software development I was lacking before, and for this case, I needed to create a checklist containing all the processes to use in my future projects. By the time I was implementing the e-commerce store, I had laid down the key concepts of the application and I knew exactly where and how a certain object show behave or look like on the frontend and it made everything else simple to approach. I have always been a pragmatic person, and I believe the extent to which someone understands an idea is an extent to which one can express that idea to others. Starting on paper usually makes me think of the best solutions to real-life problems software can address while in the development process and that’s why I strongly feel with the outline of the development processes provided in this course I can quickly settle in very well with me and make an impact on my software quality.

## Now what?

Learning the development processes from hands-on course units has given my life as a developer a fresh start since I have been able to apply this knowledge to the development process of a couple of websites that I have been working on to sharpen the skill learned. I have confidently quite successfully in applying this for the development process of a flight booking system using python. I noted that it took me less time to develop than I would have if I did not use the process outlined in this course and it has made all the difference for me right now. It has changed my perspective from software development being hard to a more positive perspective in which with according to my experience, without this process, developing an application can be a steep mountain to climb. The other advantage which I quite find fascinating as a developer with using the above-named processes, is has led me to be able to subdivide my tasks in every development stage and I knew exactly what I was supposed to accomplish on every task and this is the key to rapid and efficient software development without quality loss. The metrics for quality software are measure to be excellent when the software performs under the requirements stipulated. Having to experience the different processes hands-on has greatly lifted my spirit and I feel more confident to handle any software challenge as long as I can identify the requirements needed for the software and that kind of feeling is what a developer needs to settle into this extensive field of software engineering which can be overwhelming without a clear strategy to approach software problems.

The processes learned in this course and their successful implementation using real-life scenarios have helped me to realize that, it is one of the ways professionals evolve and move beyond rule-bound behavior and which enables them to function in a world of uncertainty and see problems holistically and act appropriately.