## PGDip in Computer Science E-Portfolio:

<https://garciar.github.io/University-of-Essex-Richard-Garcia/>

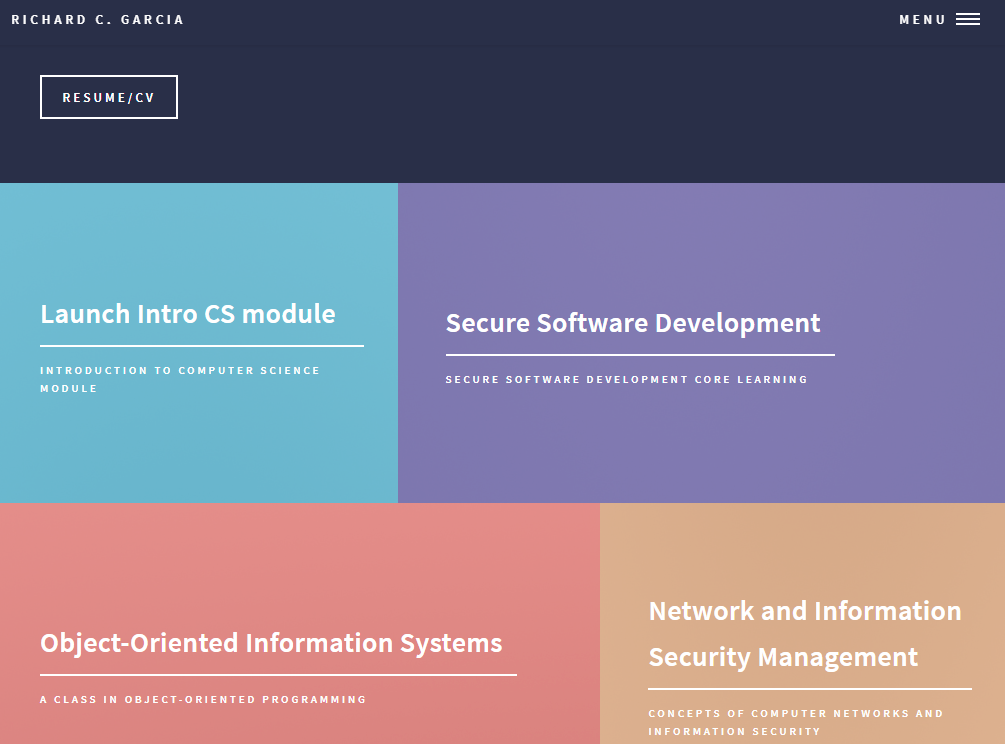


Fig 1

Main page of e-profile showing multiple Modules

## What?

Software development using the object-oriented paradigm is a major process. It is always dynamic and also allows for greater flexibility through its life cycle. In my own view, this course tries to achieve knowledge on the process through a set of units that are all polarized towards the same objective. Software development can be overwhelming when the steps are not followed resulting 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 apply it on my future projects in my professional development. My role on this was to get as much knowledge on this as possible by going through the entire unit collection which often 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 have in my understanding of the software development life cycle. Additionally, I will be able to apply this knowledge when required to develop any type of software. Through the mastery of content, will improve my efficiency, software quality, and also be able to avoid confusion and maneuver the complexities involved in software development.

To get the most out of this course, I was deeply buried in achieving the short-term goals which aggregated to the long-term goal of the entire project.

The first goal of the course was learning how to design a database. That was achieved by investigating the core elements of a database. Gribkov E. (2020) Therefore, I was supposed to develop knowledge and skills required in the application of the database design principles to my projects. Thereafter, I was to 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. Sergievskiy, M., (2017) When modeling a database, I learned there are various types of modeling a database to represent different aspects of the database. Some of the commonly applied types that I came by included; network model, document model, relational model, star schema, entity attribute value model, hierarchial database model, and the object relational model. It was also clear to me that there are just two domains in which we could characterize a model. They included the structural models and behavioral models. I used structural models to build the structure of the database design using a class diagram. The class diagram I designed was supposed to model the entities in which

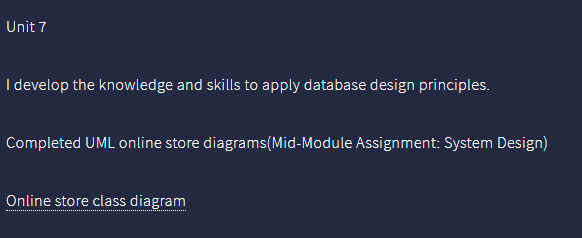


Fig 2

Fig 2 shows an excerpt of the reflaction page to download the diagram.

the database will be stored together with their attributes. The entities are objects representing real-life objects in the domain of e-commerce stores. The 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 and it required that I anticipate what data will be required and develop a requirements list from which I formulate the main objects which acted as the 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 referred to as normalization; which can be used to make any database effective. I was able to check and validate my design all the time to make sure it conformed to the rules of the third normal form (3NF) database which included; avoiding data redundancy, reduce data anomalies, ensure referential integrity, and lastly being able to simplify the data management. Moreover, the 3NF should have no transitive functional dependencies and should be within the ranges of the second normal form (2NF) database. Hence, at the 3NF, most of the data is usually presented in a form that can no longer become decomposed to advanced normal forms of normalization.

Carlos Coronel and Steven Morris (2017)

Practice makes perfect so being said, the second goal was hands-on with the database design. Therefore, 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. Dietrich Suzanne W. and Urban Susan D.(2011) 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. Hence, the entity is an object that can be distinguished from the rest of the surrounding environment. 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 the relationships between different accounts; for example a seller account and 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 learnt all the above designing prerequisites and 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. 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 and to understand the different types of queries and their uses 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 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. Implementation of a SQL database at that level was in preparation for the development of the database in my final project.

SQL is widely used in any modern information system. 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 goal 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 in a 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. Additionally, I was able to gain further knowledge by working with SQL and databases. It 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 integrated; first initializing an empty database to store data, then creating tables on the database. The table was used to store similar collections of data, and the attributes of the data formed the columns of the table while adding relations between table attributes 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.

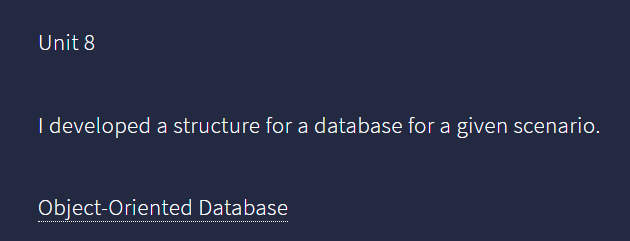


Fig 3

Fig 3 show the link to download the database I created.

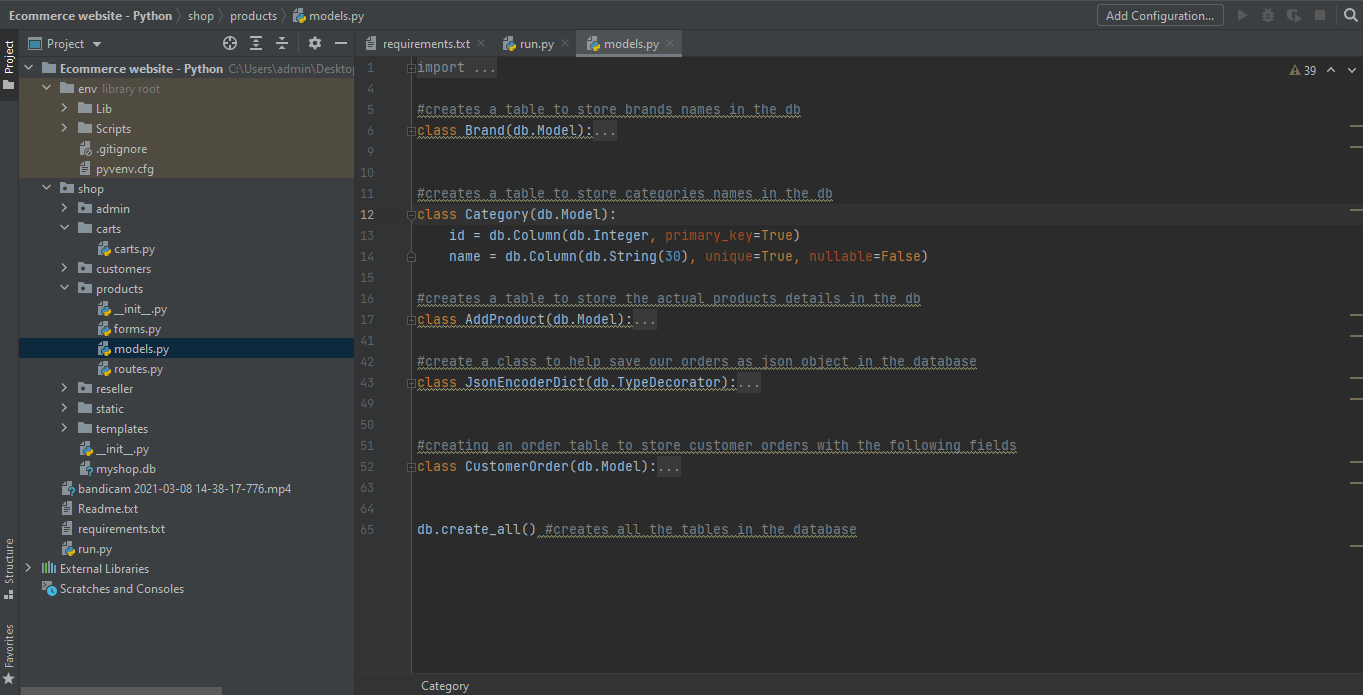
The final goal of the course was to help me to put together the knowledge acquired during the previous goals achieved and successfully implement an e-commerce website. The first step towards the achievement of that was focusing on the requirements to include in the database. I knew by developing the requirements I had formed the basis upon which our database would be designed on. Thus, I was able to anticipate the requirements for the project. Based on them I formulated an entity-relation diagram, that I used to model the database essential pillars to form a clear view of how the objects will relate to the database once implemented. Once the design was ready, I used SQLalchemy for implementation. Since I was working on a pythonic environment, inclusion of the SQLAlchemy seemed intuitive because it allowed me to create, read, update and delete from an SQL database with ease. The python-like structure queries which were also backed with the SQLAlchemy queries, translated to real SQL queries that a relational database could easily understand. Using SQLAlchemy felt cool because I was using two programming languages to archive database interactions which were camouflaged by the python environment and easy-to-learn structures. The files responsible for database interactions are referred to as models. 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.

Fig 4

Show me creating the Database relations

Once the database was 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. Dietrich Suzanne W. and Urban Susan D.(2011) That was important because I was able to anticipate the behavior of entities at certain points and call for necessary events to happen on the front-end or back-end. 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 to the other on the system. That led to a deep understanding of the inner working system components’ functioning. In the spirit of designing before implementation, I went further to create a wireframe to prototype the front-end of the e-commerce web app. Doing that, alleviates the complexities involved just like in designing using UML on the backend. I was able to spend my time on developing functionalities on the system rather than on the look and feel of the application. That is because it was already done using the prototype design that uses the wireframe. Gradually, it led to reducing the amount of time required to implement the e-commerce application.

Throughout the development cycle of the e-commerce store, I took advantage of the knowledge acquired in those previous units. That led to efficiently designing and implementing 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. Shalabh Aggarwal (2019)

## So what?

This endeavor has been fruitful because it led me down a path to be an exquisite applications developer who not only develops applications starting from coding, but also one 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 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. First, I identify the database requirements that the software application needs to accomplish; then I design the implementation of the requirements on paper using a modeling language like UML; and finally, implement the software using the modeling diagrams designed.

By the end of the implementation of the e-commerce store, it was clear that this was indeed a professional approach to software development that I lacked before. 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 knew exactly where and how a certain object should behave or look like on the front-end. That made everything else in the process 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 expresses that idea to others. Starting from the paper approach usually makes me think of the best solutions to real-life problems that software can address while in the development process. That is why I strongly feel the outline of the development processes provided in this course can settle well with me and make an impact on my software quality.

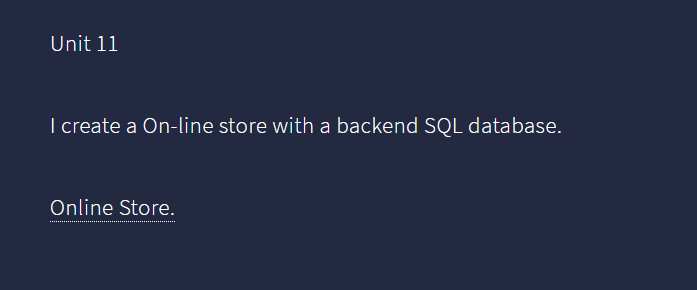


Fig 5

Here shows the link to the completed e-commerce website I developed these last 12 weeks

## Now what?

Learning the development processes from hands-on course units has given me a different perspective and approach as a developer. 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 skills obtained. I can confidently and successfully apply the concept in the development process of a flight booking system using python. I noted that it took me lesser time to develop than I would have without the current knowledge acquired in the course. The process outlined in this course has made all the difference for me as a softwared developer. It has changed my attitude on software development being tough to a smooth and sequential process with little technicalities. With such a process, the development of an application is easily manageable and less time consuming. The other advantage which I find quite fascinating as a developer is that it has led to skill improvement that enables me to subdivide my tasks in every development stage. Thus, 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 a quality software are measured as excellent when the software performs under the requirements stipulated. Having to experience the different processes hands-on has greatly impacted my psyche and morale. I feel more confident to handle any software challenge as long as I can identify the requirements needed for the software. That kind of motivation 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; one of the ways professionals evolve and move beyond rule-bound behavior, enables them to function in a world of uncertainty and see problems holistically and act appropriately.

References:

UKEssays. November 2018. Object Oriented Programming. [online]. Available from: https://www.ukessays.com/essays/information-technology/object-oriented-programming-what-is-object-information-technology-essay.php?vref=1 [Accessed 1 April 2021].

Peachy Essay. November 9 2020. How to Write a Reflection Paper - Full Reflection Essay Writing Guide [online]. Available from: <https://www.youtube.com/watch?v=O7ysHdYKU18> [Accessed 3 April 2021]

UNSW. Student Guide. December 23 2019. Reflective Writing [online]. Available from: <https://student.unsw.edu.au/reflective-writing> [Accessed 3 April 2021]

University of Edinburgh. Novenber 2 2020. Reflection Toolkit [online]. Available from: <https://www.ed.ac.uk/reflection> [Accessed 3 April 2020]

Gribkov E. (2020) SQL Database Design Basics with Example. Available from:

https://blog.devart.com/sql-database-design-basics-with-example.html [Accessed March 12 2021].

Sergievskiy, M., (2017). Description logic application for UML class diagrams optimization. *International Journal of Advanced Computer Science and Applications*, *8*(1), pp.268-272

Coronel C, & Morris S., (2017). Database Systems: Design, Implementation, and Management, 12th Edition. PP.202-244

Dietrich Suzanne W. and Urban Susan D.(2011) Fundamentals of Object Databases Object-Oriented and

Object-Relational Design: Morgan & Claypool Publishers.

Shalabh Aggarwal .(2019) Flask Framework Cookbook 2nd Edition Packit Publishing.