**BSC30920 - Mobile Applications**

**October 2020**

**Continuous Assessment Number 2 (Group (2))**

**Module Title: Mobile Applications**

**Module Code:** BSC30919

**Assessment Type**: Practical Assessment

**Student Name:** Nicholas Chinkire Chibuike-Eruba – 18630

Ana Karolina S. Dias - 20630

**Weighting: 60%**

**Git :** <https://github.com/El-Nico/joint-oroject-takeaway-app-18630-20630.git>

**Maximal Possible Mark:** 100 marks

**Submission Date:** December 19th, 2020

**Introduction**

During the 2020 Covid Pandemic, many are choosing to order food from restaurants and stay at home. To help keep Sales up, many restaurants are looking to get themselves online so that customers can still order from them from the safety and comfort of their own homes.

As we can see in the following graph, mobile devices have become increasingly popular for everyday tasks when compared to laptop/desktop devices. Therefore, we believe the best way for a restaurant to operate online is via a mobile application. This likes of Deliveroo, Just Eat & Uber for example take a % of each Sale so ideally it is best to develop your own app to maximize profits in the long run.



**1.Key Features in product**

**Viewing Menu**

User can see a Main menu is thought to Menu that the user is going to start shopping

**Adding products to the cart**

The user will be able to see the items and add them to the cart. The products are saved previous in room database.

**Removing products from the cart**

The user is going to be able to remove items before adding to the cart

**Checking out**

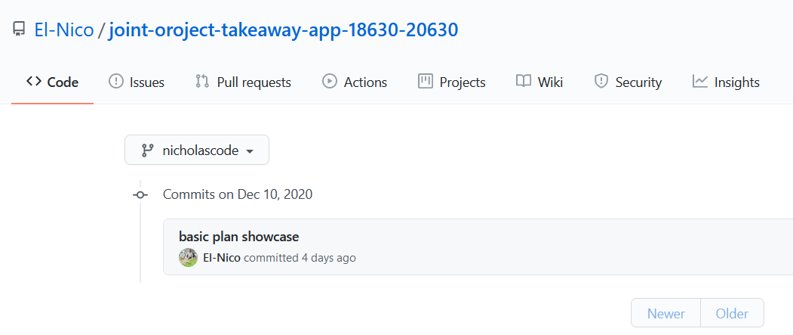
The user is able to finish the shopping and see the total amount that the shop is going to cost.

**Paying**

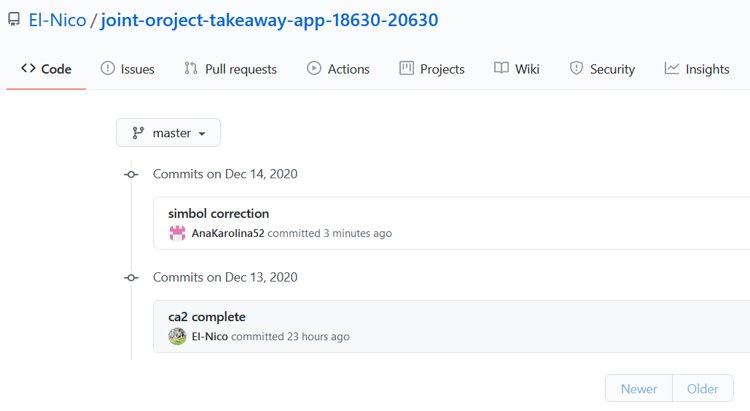
The user will pay and finish the shop. Should appears a message at the screen saying that the shop was done.

**2.GitHub snapshot**

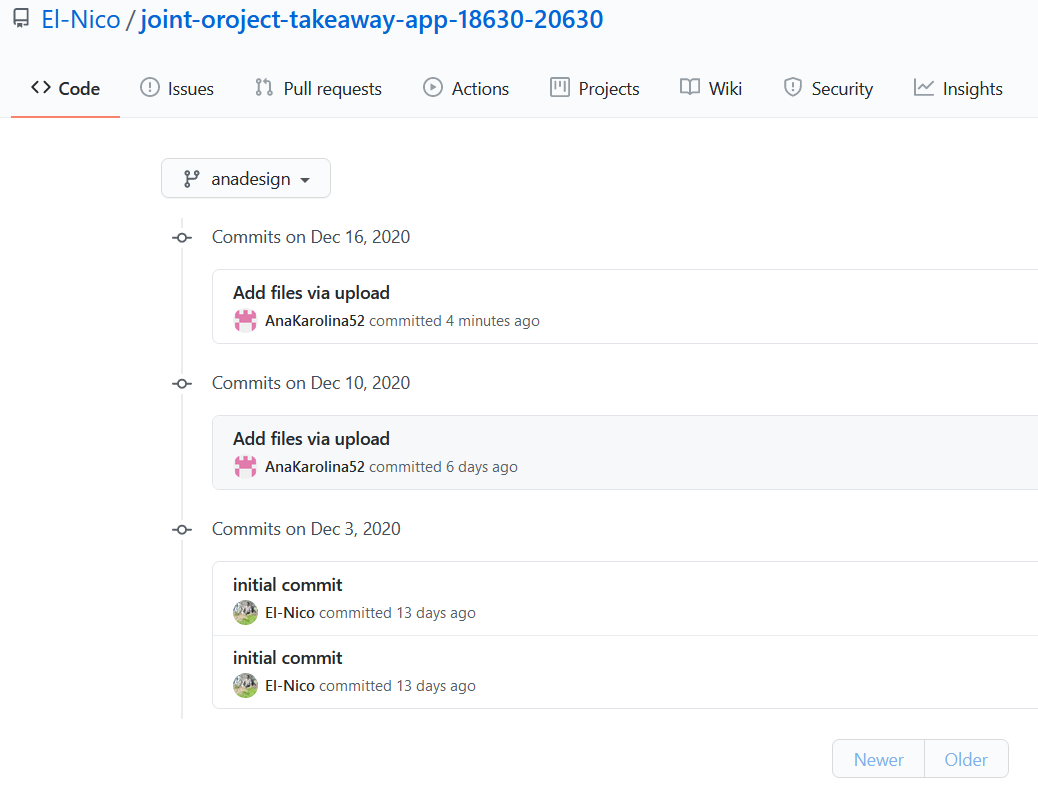
Branch: nicholascode



Branch: master



Branch: anadesign

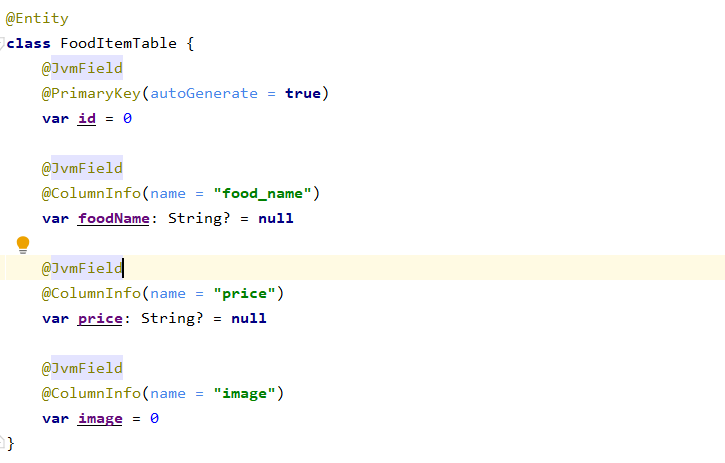


**3.Key Learning**

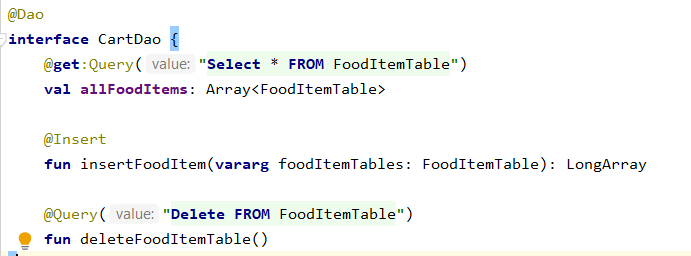
**Room Database**

Room is one of the existing libraries within the suite, it assists developers by creating an abstraction of the database layers.Layers are represented by three components:

Entity: are entities responsible for mapping the tables.



Dao (Data Access Object): are the interfaces used to access the data stored in the bank.



Database: is the representation of the abstract Database class. He will be responsible for controlling the database.



**Usability**

Even though our app is simple we still need to thing about how the user will work in the app, our plan was creating an app simple and easy to use, and that in future we could apply more complicate functions and the user still find the add easy to manipulate.

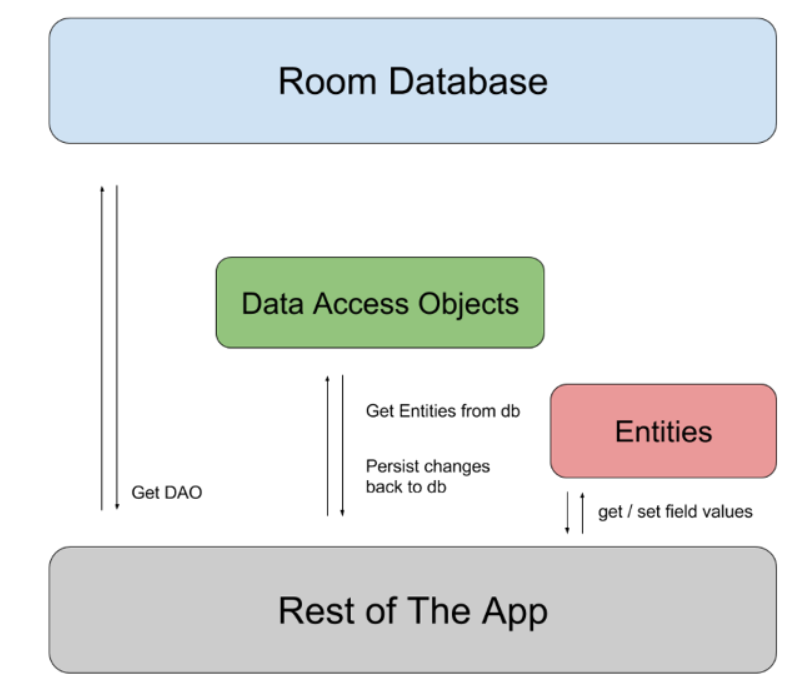
**Design**

The Design was creating thinking in a simple and cream app where the user could do their shopping without complications.

**4.Key Challenges**

The key challenge of the project was to have full comprehension of how Room works, and how to put in practice.

The image below helps us to understand how the component communicate if each other:



To start, we imported the Room library into the gradle of the app folder and synchronizedOnce you understand how the architecture works, we need to implement each component. The first one will be our FoodItemTable entity, which will represent the FoodItemTable table, in our database.

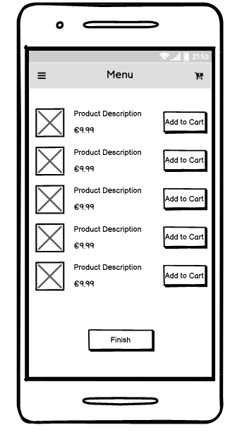
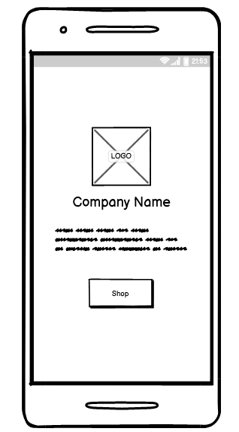
To represent an entity, the Data Class must be associated with the @Entity annotation. In addition, Room requires that there be at least one primary key in the class.

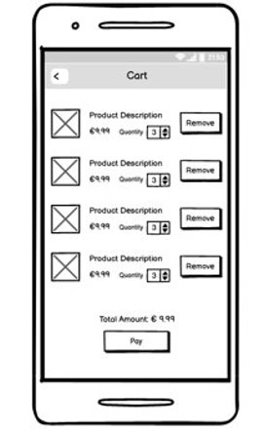
To manipulate the data from the FoodItemTable table present in the database, we will define an interface with the method signatures. This interface must be associated with the @Dao annotation and, although there are annotations like @Insert and @Delete, method signatures can contain SQL commands using the @Query annotation.

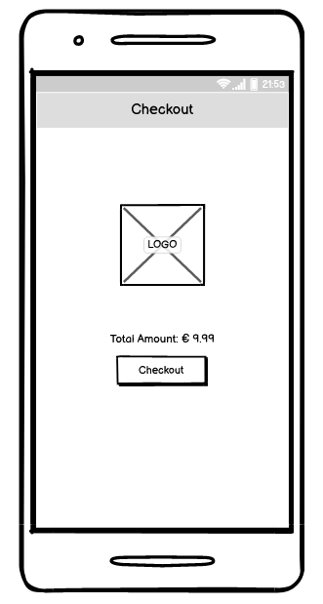
Finally, we implemented a Database class, which will serve as a connection to the database. This class should extend the RoomDatabase and use the Singleton standard so that we can make use of the same instance to access the database.With the main components already built, the next step is to create an Activity and the graphical interface to test the operation of the Room.

And that is how we went thought our challenge of how the Room library works and apply it to our code.

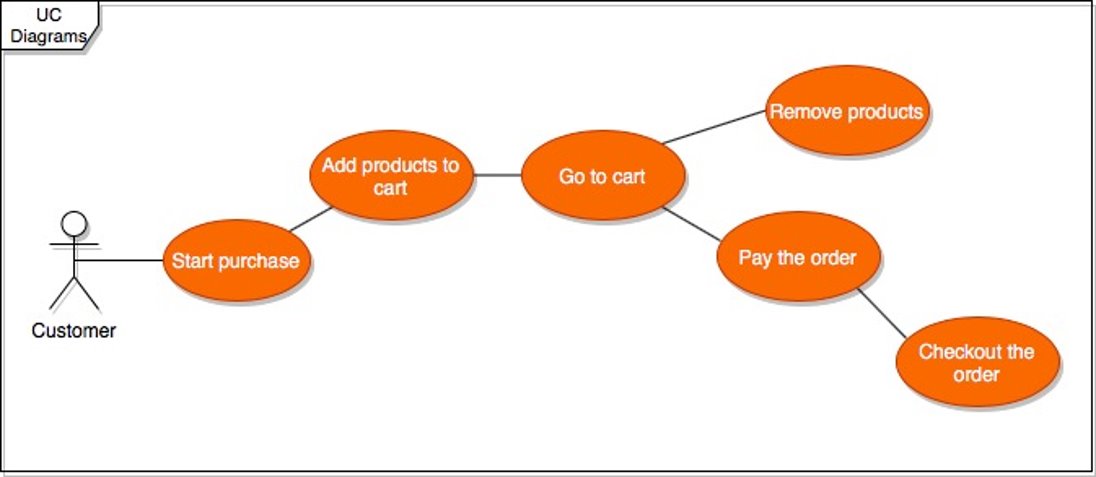
**5.Screen Prototypes**



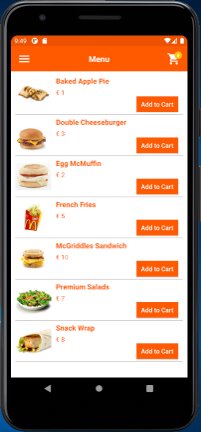
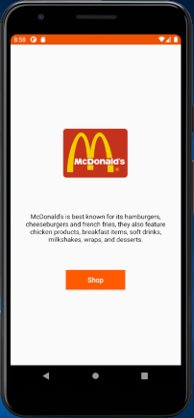


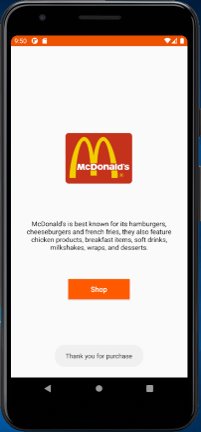
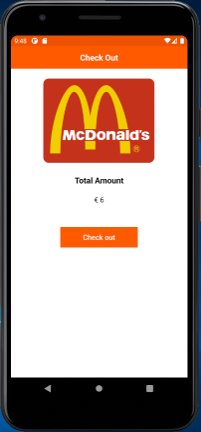
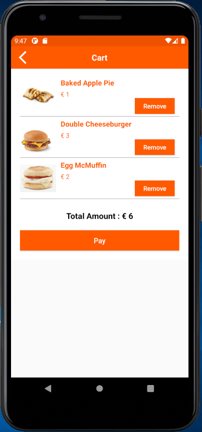


**6.UML Diagram**

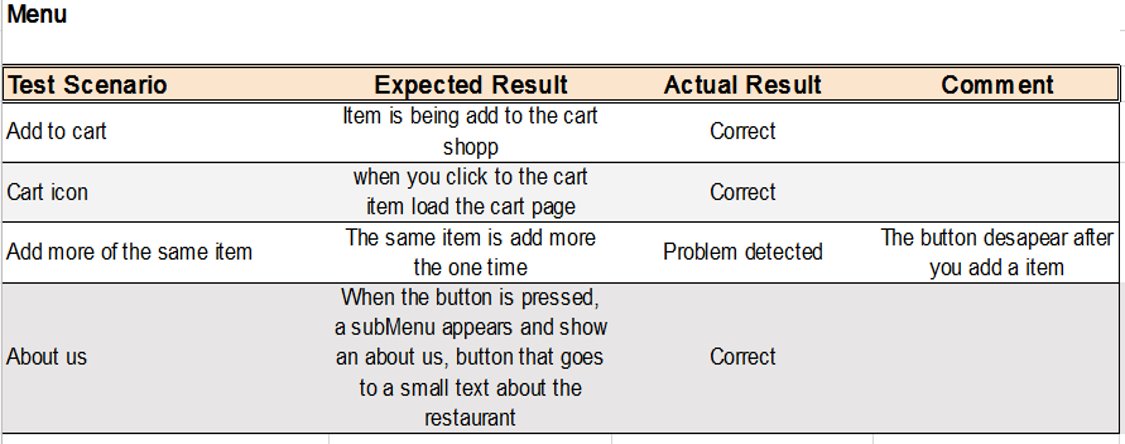


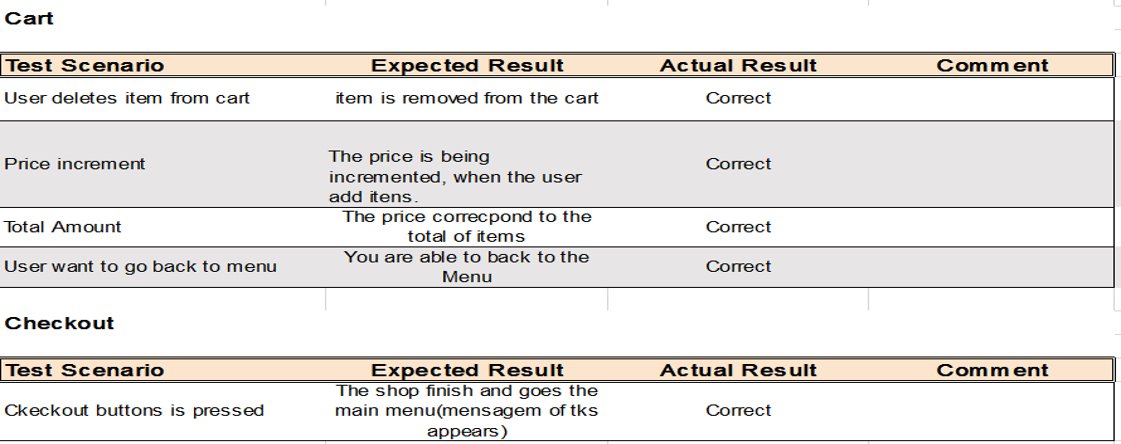
**7.System Screenshots**





**8.Testing**





**9.Group Contribution**

As the member of the group more experienced Nicholas Chinkire Chibuike-Eruba was responsible for coding and Ana Karolina was responsible for system analysis, prototype design, Testing and Documentation.