FPT UNIVERSITY
FPT UNIVERSITY CAN THO CAMPUS
SOFTWARE ENGINEERING

Java Web Application Development PRJ301

Project Document

Food Ordering Manager

Advisor: Huong Hoang Luong

CAN THO CITY, NOVEMBER 2022

Member list & Workload

Full name	Student ID	Work Load	Percent	Complete
Le Duy Khanh	CE160277	• Design, edit ERDs and data in database. • Support writing requirements. • Build frontend of JSPs (Add/Update/Delete food), catch user input errors. • Create a navbar and footer.	18%	100%
Tran Trung Kien	CE161053	• Create database and data in MySQL. • Build frontend of backend JSPs (FoodList). • Design and write reports. • Support writing requirements.	18%	100%
Vo Minh Dat	CE160116	• Write requirements for ERD. • Build frontend and backend of JSPs (Profile). • Support adding data to the database. • Write a report	18%	100%
Vo Hong Quan	CE160078	• Build frontend and display data from JSP's DB (Login), catch user input errors • Create Controller classes of EmployeeController • Create Filter to decentralize Admin/staff • Backend of Order and Payment	28%	100%
Huynh Chi Hai	CE160053	• Design and edit ERDs. • Build frontend of JSPs (Order/Payment). • Create Controller classes of FoodController • Create order successful page	18%	100%

Contents

1 Abstract **4**

2 Software development model **4**

3 Diagram **5**

4 Architecture, Functional and Non-functional. **7**
4.1 Architecture 7
4.2 Functional and non-functional requirements 9
4.2.1 Functional requirements 9
4.2.2 Non-functional requirements 9

5 Test model **10**
5.1 User Login Form Validation 10
5.2 Add New Employee Form Validation 11
5.3 Category Validation 12
5.4 Employee Validation 13
5.5 Foods Validation 14
5.6 Make Payment Validation 15
5.7 Order Validation 16
5.8 Table Validation 17

6 Methods for managing source code changes **18**

7 Interface **19**

1 Abstract

Nowadays, digitization is proving more and more beneficial in terms of revenue and user experience. However, restaurants are using the method of ordering food and paying by word of mouth or handwritten which has disadvantages affecting the customer experience such as unprofessional-ism, time consumption and possible mistakes. Instead of having to use the above inefficient method, employees can now use a flexible application that makes ordering and paying tasks easier, more accurate and more convenient. At the same time, customers have access to more professional service when eating at the restaurant. Most importantly, the restaurant saves on staff at the restaurant, laying the foundation for maximizing revenue.

2 Software development model

Scrum is a project management system that prioritizes collaboration, responsibility, and incremental advancement toward a clear objective. The framework starts with a straightforward directive: Begin with what is visible or understandable. After that, evaluate the results and make any required adjustments. We choose the Scrum model because of the following advantages:

- Easy to plan and task as it breaks down into specific sprints.
- One person in the team can do a lot of things both coding and testing.
- Update the current Sprint status every day to detect bugs early that can be quickly fixed. As well as easily communicating with members when there are questions and timely support.
- Saving time.
- The sequential execution process is easy to understand, easy to follow, and hassle-free.

3 Diagram

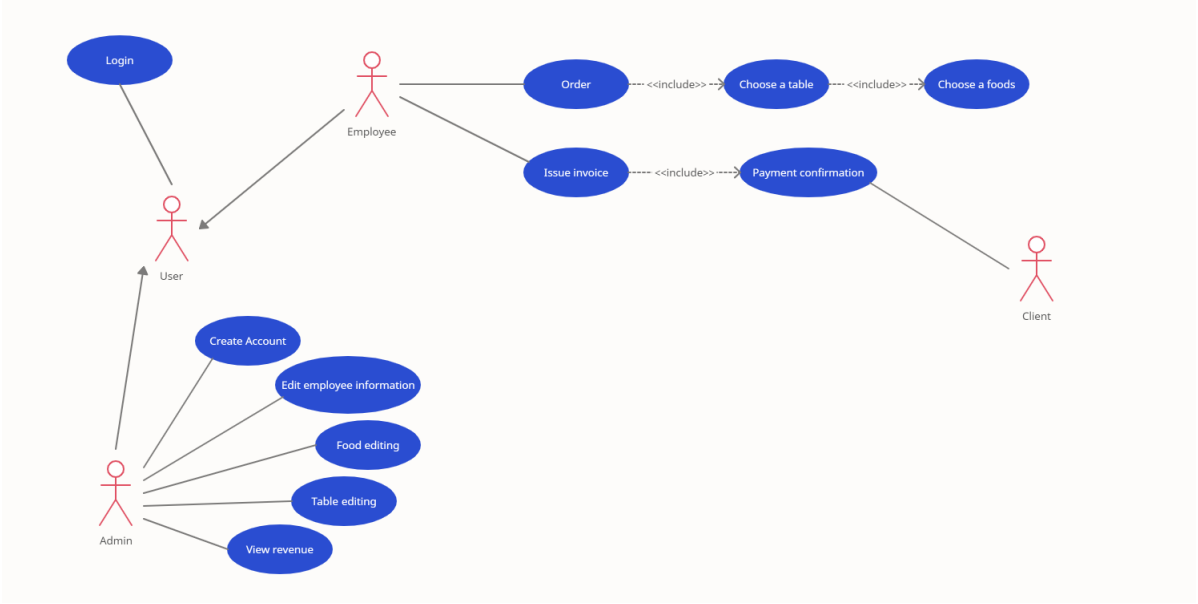

Figure 1: This is use case of

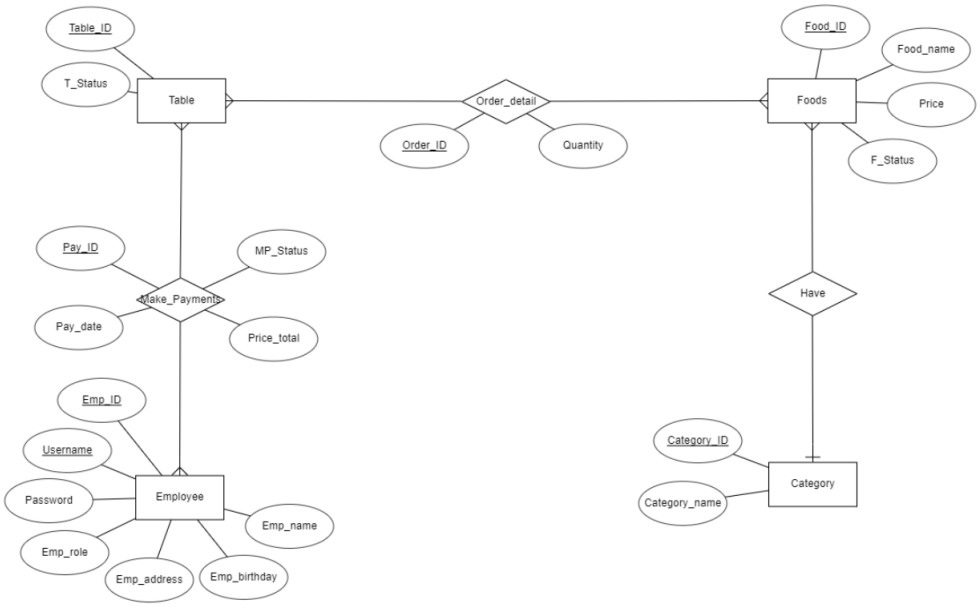

Figure 2: The erd of project

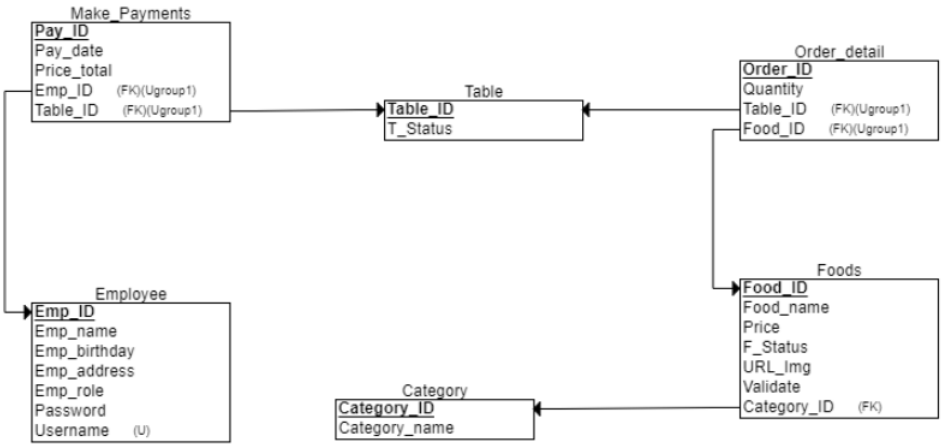

Figure 3: The relational schema

4 Architecture, Functional and Non-functional.

4.1 Architecture

Model–view–controller (MVC) is a software architectural pattern commonly used for developing user interfaces that divide the related program logic into three interconnected elements.

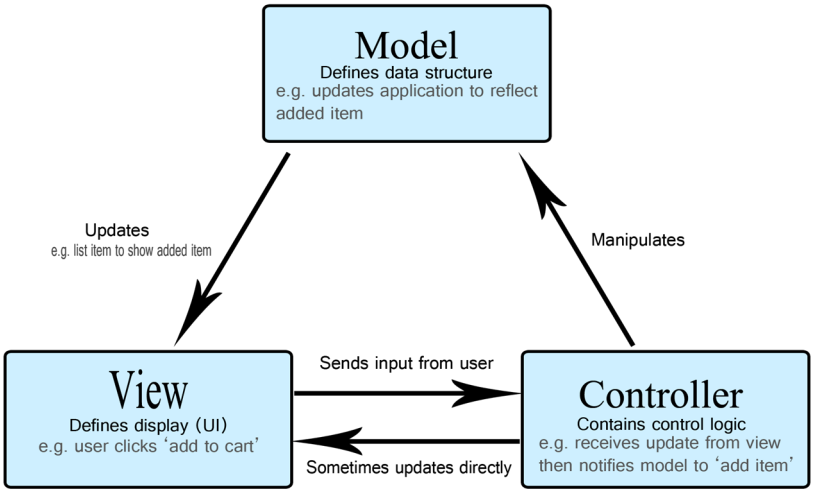

Figure 4: The MVC model

MVC architecture offers a lot of advantages for a programmer when developing applications, which include:

- Multiple developers can work with the three layers (Model, View, and Controller) simultaneously
- Offers improved scalability, that supplements the ability of the application to grow
- As components have a low dependency on each other, they are easy to maintain
- A model can be reused by multiple views which provides reusability of code
- Adoption of MVC makes an application more expressive and easier to understand
- Extending and testing of the application becomes easy

The benefit of MVC is convenience to our team. We chose it to develop this project. In the MVC design pattern, the model is the data layer which defines the business logic of the system and represents the state of the application. The model objects (figure 5) retrieve and store the state of the model in a database. Through this layer, we apply rules to data, which eventually represents the concepts our application manages. To connect the models to database we use JDBC. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database. JDBC is an API for accessing relational databases using Java. Through it we connect to MySQL a database that we use. It is about connection also to access data in that we have to use DAO (figure 6). Data Access Object (DAO) is an abstraction for accessing data, the idea is to separate the technical details of data access from the rest of the application. It can apply to any kind of data.

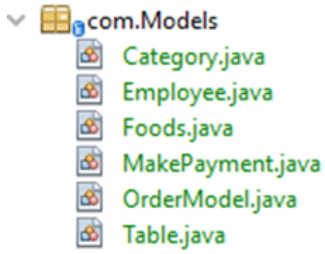

Figure 5: The models

Figure 6: The DAO

The View Layer of the MVC design pattern represents the output of the application or the user interface. It displays the data fetched from the model layer by the controller and presents the data to the user whenever asked for. It receives all the information it needs from the controller, and it doesn't need to interact with the business layer directly. We use JSP for the user interface. JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc. We use JSP for the user interface. It's easy to use, no need to recompile and redeploy the project when editing. JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic.

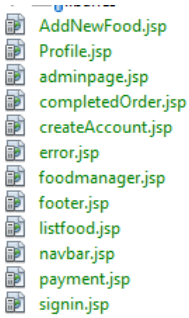

Figure 7: The JSP

The Controller is like an interface between Model and View. It receives the user requests from the view layer and processes them, including the necessary validations. The requests are then sent to model for data processing. Once they are processed, the data is again sent back to the controller and then displayed on the view.

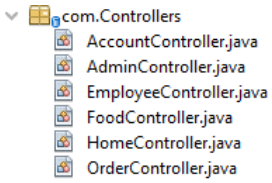

Figure 8: The Controller

4.2 Functional and non-functional requirements

4.2.1 Functional requirements

Accounts are handled via username and password. And the position of that account in the hierarchy of the system. There will be an account belonging to the admin and system operator to add employee accounts, information about dishes, categories, and orders. The system must collect and process data, and then store it in the database.

Employees who use the system will retrieve information for an order. The order is processed to have the necessary information such as the number of items, the number of people, the amount to be paid. If there is a mistake during that process, the order can be corrected through the system.

Only the administrator has the right to manage feeds in the system. The system must collect and process that data into a database. The database will store it and can be accessed through the system. The system provides where the administrator can edit the information that has been entered.

The system must process the data to analyze many aspects of the data. The system must visually display the information. //

4.2.2 Non-functional requirements

Speed: As long as the server and the internet, the speed is not an issue. It can also be said that the speed is high because the data is not too large.

Availability: Simultaneously, multiple devices can be used at the same time. The login system will recognize the username. We cannot use the same username at the same time. Therefore, many different usernames can be used.

Capacity: Depends on the storage capacity of the server. But can backup to another place so the storage capacity is quite large.

Reliability: The system stores passwords in encrypted form. All access is authenticated and only admin can add accounts for employees.

Usability: The site is highly compatible. Can be used on mobile, PC, tablet and other devices.

5 Test model

5.1 User Login Form Validation

FUNCTION NAME	DESCRIPTION	TEST STEP	EXPECTED RESULT	STATUS
USERNAME	Do not allow users to use special characters to set username	Randomly enter special characters in the username field	Block and output invalid username message	CHECKED
PASSWORD	Allow user to enter any password, MD5 encryption	Enter any string regardless of special characters	Accept the root password and encrypt the MD5	CHECKED

5.2 Add New Employee Form Validation

FUNCTION NAME	DESCRIPTION	TEST STEP	EXPECTED RESULT	STATUS
EMPLOYEE ID	Must not be the same as the old ID, with a length of no more than 50 characters, no special characters or blanks	Log in to an account with administrator rights, switch to the management page, enter the character string for the employee ID	Employee ID is valid, check and show message if ID is duplicate or invalid character	CHECKED
FULL NAME	The length must not exceed 50 characters, do not use special characters or leave blanks	Enter the character string for the employee full of name	Check if string is not valid then show message	CHECKED
USERNAME	The length must not exceed 50 characters, do not use special characters or leave blanks. Username must not be the same as the old username	Enter the character string for the employee's username	Check the same old login name, display a message if invalid	CHECKED
GET ALL CATEGORY	Select all the categories in the catalog table and display them on the website: Category_ID, Category_Name	After successful login, the list of dishes taken from the menu is displayed on the website	Full display of all dishes in the category	CHECKED
DELETE CATEGORY	Delete the list of category by Category_ID	Select and delete unwanted menu items	Delete successfully	CHECKED
ADD NEW CATEGORY	Add a new dish category, don't duplicate the old Category_ID	Create a category and name a new dish category	Create a successful category, prevent if the category ID is duplicated	CHECKED

5.3 Category Validation

FUNCTION NAME	DESCRIPTION	TEST STEP	EXPECTED RESULT	STATUS
GET ALL CATEGORY	Select all the categories in the catalog table and display them on the website: Category_ID, Category_Name	After successful login, the list of dishes taken from the menu is displayed on the website	Full display of all dishes in the category	CHECKED
DELETE CATEGORY	Delete the list of category by Category_ID	Select and delete unwanted menu items	Delete successfully	CHECKED
ADD NEW CATEGORY	Add a new dish category, don't duplicate the old Category_ID	Create a category and name a new dish category	Create a successful category, prevent if the category ID is duplicated	CHECKED

5.4 Employee Validation

FUNCTION NAME	DESCRIPTION	TEST STEP	EXPECTED RESULT	STATUS
GET ALL EMPLOYEE	Show all employee information (don't show password): Emp_ID, Emp_birthday, Emp_gender, Emp_phone, Emp_address, Emp_role, Username	Log in to the admin account, switch to the management page, the list of employees will be displayed	Show list of employees by table but do not show password	CHECKED
ADD NEW EMPLOYEE	Adding new employees to the database can only be done by the administrator. The new Emp_ID cannot be the same as the old Emp_ID	Log in to the administrator account and go to the management page, enter the employee's information and add it to the database	Add new employee successfully by admin	CHECKED
DELETE EMPLOYEE	Delete an account through the Emp_ID, check the Emp_ID exists or not	In the management page, select the trash can button symbolizing delete the account, the account will be deleted from the database	Delete successfully if ID exists	CHECKED
UPDATE EMPLOYEE	Update information for account by Emp_ID	On the management page, enter all the information you need to update	Update is successful if the ID exists, otherwise it will not update and display a message	CHECKED

5.5 Foods Validation

FUNCTION NAME	DESCRIPTION	TEST STEP	EXPECTED RESULT	STATUS
GET ALL FOOD	Get all food information: Food_ID, Food_name, Price, F_Status, URL_img, Category_ID	This function will work when you call certain functions	Display the list of dishes on the website if the dish exists in the database	CHECKED
GET FOOD IN CATEGORY	Show all dishes with Category_ID	This function will work when you call certain functions	Displays a list of dishes that are in the category you selected if the dish exists in the database	CHECKED
SET FOOD STATUS	Set food status is available or not	This function will work when you call certain functions	Set food status successful	CHECKED
DELETE FOOD	Delete food from database by Food_ID	This function will work when you call certain functions	Delete selected dish successfully by Food_ID if Food_ID already exists	CHECKED
ADD NEW FOOD	Add a food provided that the Food_ID is not the same as the old Food_ID	This function will work when you call certain functions	Delete selected dishes successfully with Food_ID	CHECKED
UPDATE FOOD	Update new Food_name, Price, F_Status, URL_img or Category_ID	This function will work when you call certain functions	Update successfully if Food_ID exists, display message if Food_ID does not exist	CHECKED
GET FOOD PRICE	Get price of food by Food_ID	This function will work when you call certain functions	Show the price of the food	CHECKED
GET FOOD NAME	Get food name by Food_ID	This function will work when you call certain functions	Display food name via Food_ID	CHECKED
GET FOOD BY ID	Get food by Food_ID	This function will work when you call certain functions	Display all information of the dish to be searched	CHECKED

FUNCTION NAME	DESCRIPTION	TEST STEP	EXPECTED RESULT	STATUS
GET FOOD STATUS	Get food status by Food_ID	This function will work when you call certain functions	Display the status of the food is available or sold out	CHECKED
GET ALL CATEGORY ID	Get all information of category form category table	This function will work when you call certain functions	Show all information of category	CHECKED

5.6 Make Payment Validation

FUNCTION NAME	DESCRIPTION	TEST STEP	EXPECTED RESULT	STATUS
GET ALL PAYMENT	Get all the invoices paid in admin management	Login as admin to view the payment list	Show all payment list	CHECKED
GET NUMBER OF PAYMENT	Count all payment	Login as admin to see the number of paylists	Show total payment amount	CHECKED
DELETE MAKE PAYMENT	Delete payment from database by Pay_ID	Select the payment section in the management section and delete it from the database	Delete payment successfully if Pay_ID exists, otherwise display a message	CHECKED
ADD NEW MAKE PAYMENT	Add payment when order is successful	Select the table and choose the item to pay, then the payment invoice will be added to the database	Show successful payment and add to payment invoice table	CHECKED
SET MAKE PAYMENT STATUS	Set payment paid or not	If you only choose the item but haven't paid yet, the invoice will not be issued to the table	Show status from unpaid to paid	CHECKED
GET PAYMENT ID	Get payment information by Pay_ID	Enter payment ID to look up information	Show payment information via certain ID lookup	CHECKED
GET TOTAL IN MONTH	Get the total money earned in 1 month	Select the month to get the total amount collected	Displays the total amount earned in the selected month	CHECKED

5.7 Order Validation

FUNCTION NAME	DESCRIPTION	TEST STEP	EXPECTED RESULT	STATUS
GET ALL ORDER	Get all information of the order: Order_ID, Quantity, Table_ID, Food_ID	Select the table and select the number of dishes, then order, the list of food in the order will appear	Show list of dishes on order page	CHECKED
GET NUMBER OF ORDER	Count order list from Order_ID	Show the number of dishes in the order	Display the number of selected dishes on the order page	CHECKED
DELETE ORDER	Delete a food order by Order_ID	Delete a certain dish from the order	Successfully deleted the selected dish	CHECKED
ADD NEW ORDER	Serialize new orders to old orders	Return to the order menu and select the quantity of the item to order	Successfully added a new item to an existing order	CHECKED
GET ORDER TO CALC PAYMENT	Calculate the order of the dish that is in Order_ID	Show the price of each dish on the same order	Display information about the price of each dish	CHECKED
GET ORDER	Get order detail from Order_ID	Get information about orders by specific ID	Displaying staff order information, table number	CHECKED
GET TOTAL PRICE	Get total price from Order_ID	Information about the total amount will be displayed when ordering	Show total payment price	CHECKED

5.8 Table Validation

FUNCTION NAME	DESCRIPTION	TEST STEP	EXPECTED RESULT	STATUS
GET ALL TABLE	Get table ID and status of empty or occupied table	Choose a table from the list to order	Show all tables in the table list	CHECKED
GET NUMBER OF TABLE	Count all table	Show the number of tables in the list	Show the number of tables in the current list	CHECKED
SET TABLE STATUS	Set status for table empty or occupied	Select dishes based on the selected table to confirm that the table is occupied	Show empty or busy table	CHECKED
DELETE TABLE	Remove the table from the database by Table_ID. This function is performed by the administrator	Enter or select the table ID to remove the table from the database	Delete table successfully by administrator. If the desk ID is not found, a message is displayed	CHECKED
ADD NEW TABLE	Add new table into database	Enter the new table ID and select the initial state of the table	Add a new table, the new table ID must not be the same as the old table ID	CHECKED
GET TABLE	Get table ID and status of empty or occupied table by Table_ID	Enter the table ID to get information	Show tables information with the table ID	CHECKED

6 Methods for managing source code changes

Figure 9: Github

Use Github to make it easier for team members to exchange code and to show off project progress. Quan will be the one responsible for overseeing the Project on Github directly. Every time there is a change, the team will be notified so that it can be updated on Github for everyone to download and see. It is also very convenient to know where the code has changed on Github so that it can be readily modified. when required. Additionally, members will be informed of issues in the current code so they may work together to find solutions.

7 Interface

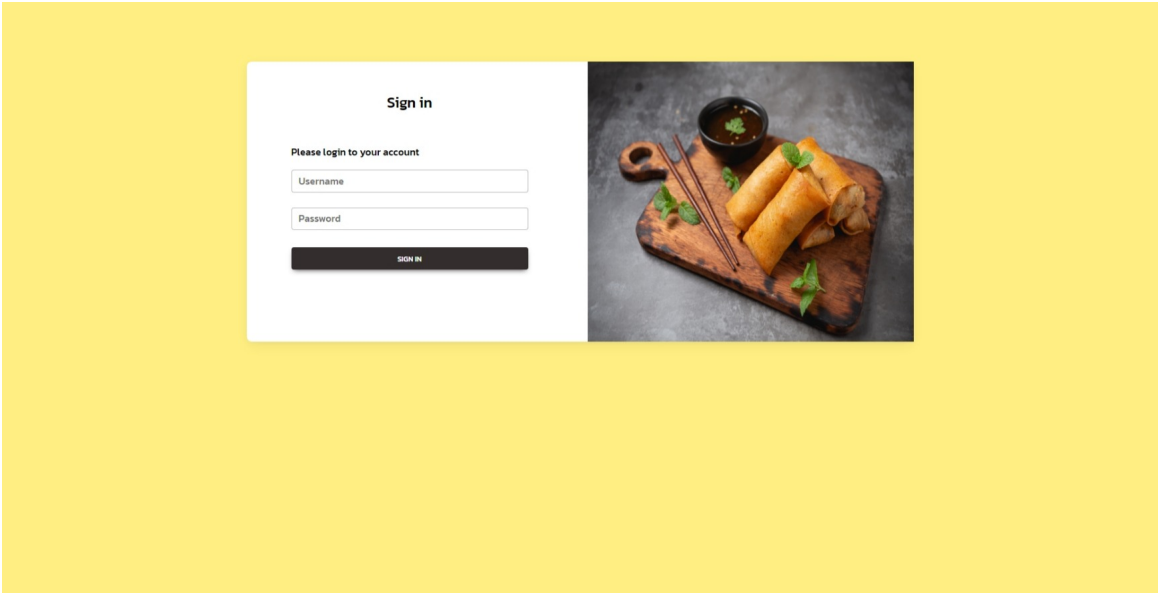

Figure 10

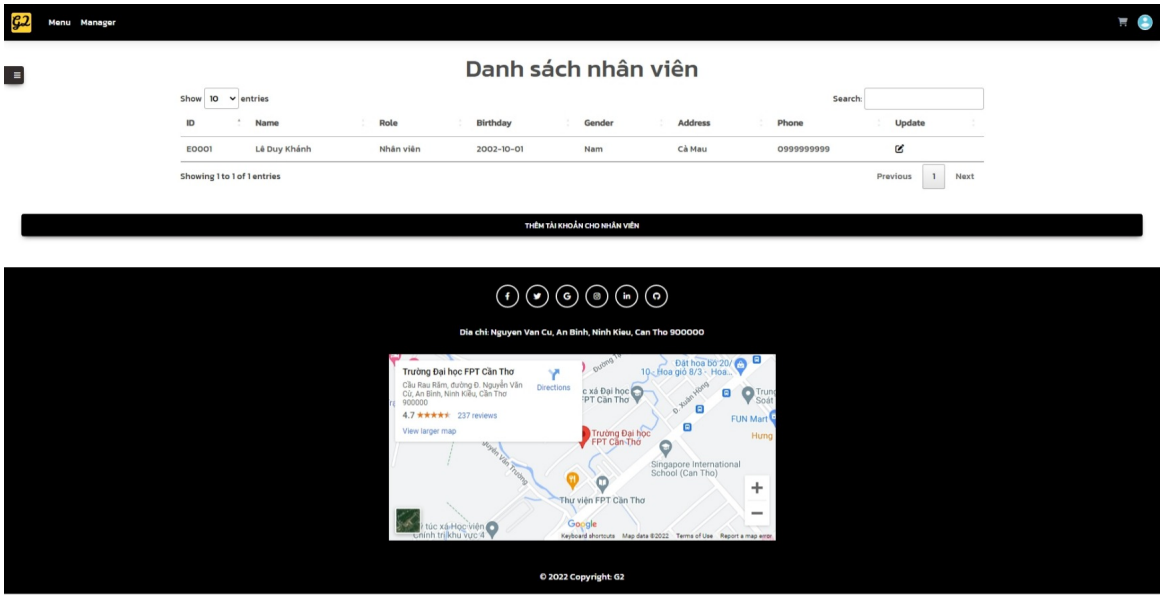

Figure 11

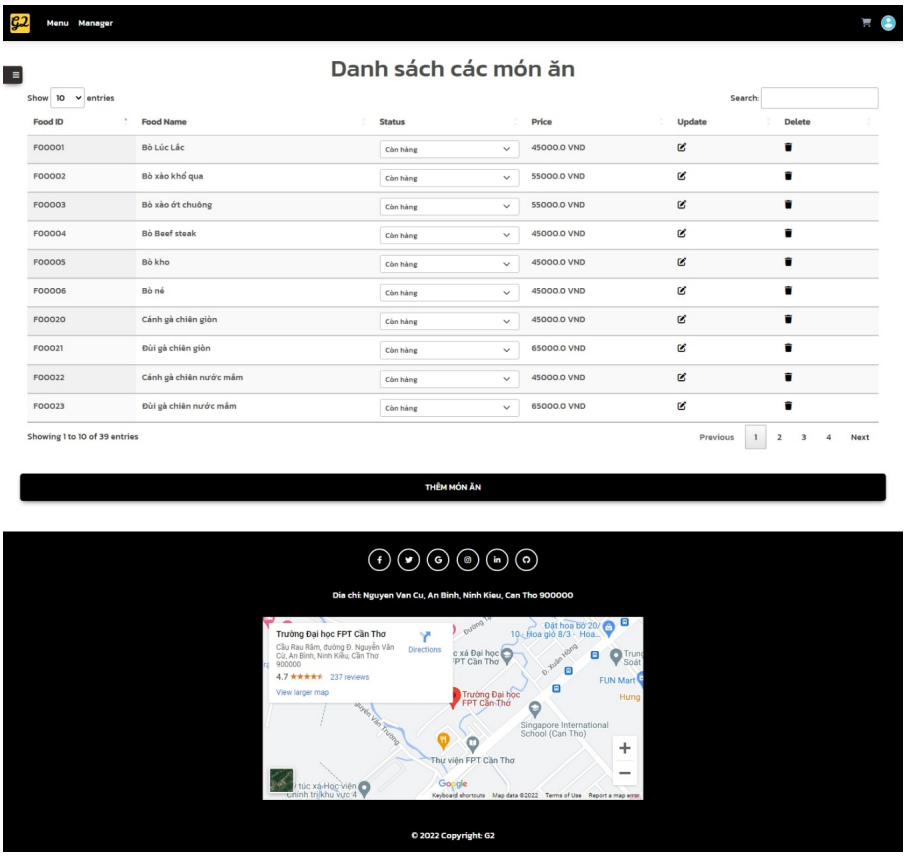

Figure 12

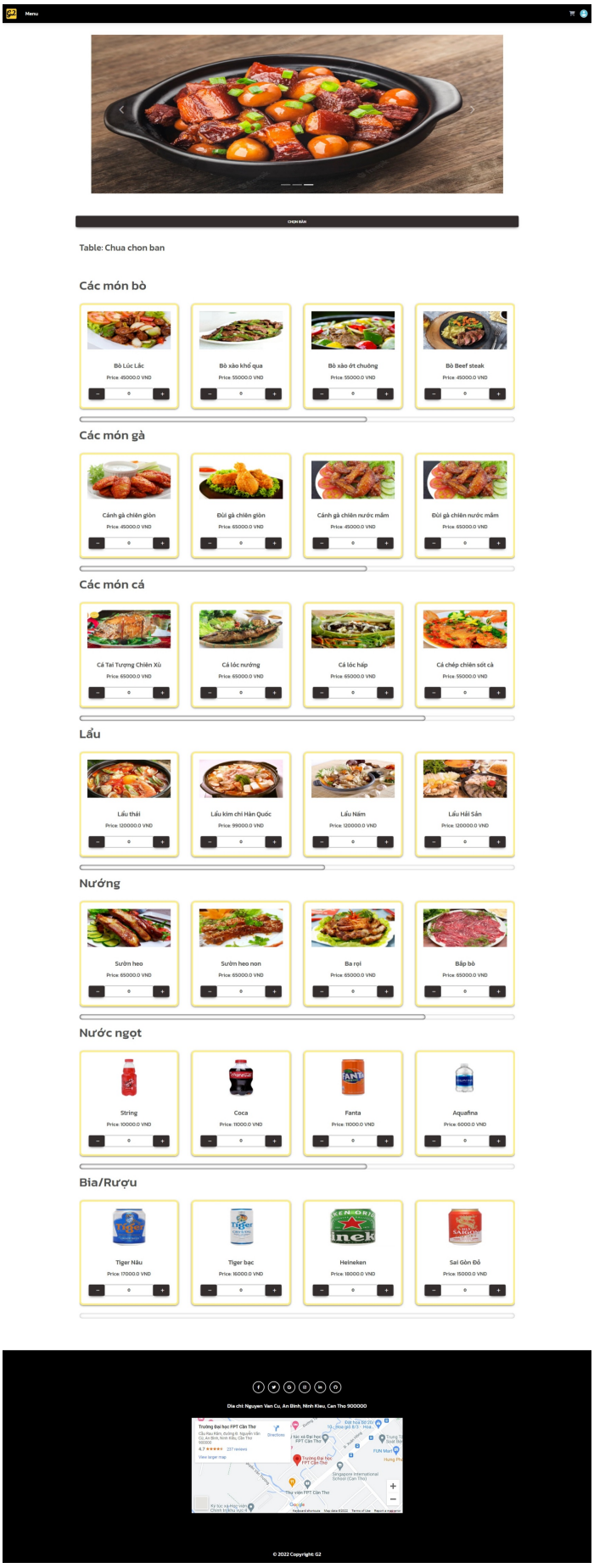
Menu
CHỌN BÀN
Table: Chua chon ban
Các món bò
Bò Lúc Lắc
Price: 45000.0 VND
Bò xào khổ qua
Price: 55000.0 VND
Bò xào ớt chuông
Price: 55000.0 VND
Bò Beef steak
Price: 45000.0 VND
Các món gà
Cánh gà chiên giòn
Price: 45000.0 VND
Đùi gà chiên giòn
Price: 65000.0 VND
Cánh gà chiên nước mắm
Price: 45000.0 VND
Đùi gà chiên nước mắm
Price: 65000.0 VND
Các món cá
Cá Tai Tượng Chiên Xù
Price: 65000.0 VND
Cá lóc nướng
Price: 65000.0 VND
Cá lóc hấp
Price: 65000.0 VND
Cá chép chiên sốt cà
Price: 55000.0 VND
Lẩu
Lẩu thái
Price: 120000.0 VND
Lẩu kim chi Hàn Quốc
Price: 99000.0 VND
Lẩu Nấm
Price: 120000.0 VND
Lẩu Hải Sản
Price: 120000.0 VND
Nướng
Sườn heo
Price: 65000.0 VND
Sườn heo non
Price: 65000.0 VND
Ba rọi
Price: 65000.0 VND
Bắp bò
Price: 65000.0 VND
Nước ngọt
String
Price: 10000.0 VND
Coca
Price: 11000.0 VND
Fanta
Price: 11000.0 VND
Aquafina
Price: 6000.0 VND
Bia/Rượu
Tiger Nâu
Price: 17000.0 VND
Tiger bạc
Price: 16000.0 VND
Heineken
Price: 18000.0 VND
Sai Gòn Đỏ
Price: 15000.0 VND
Địa chỉ: Nguyen Van Cu, An Binh, Ninh Kieu, Can Tho 900000
© 2022 Copyright: G2

Figure 13

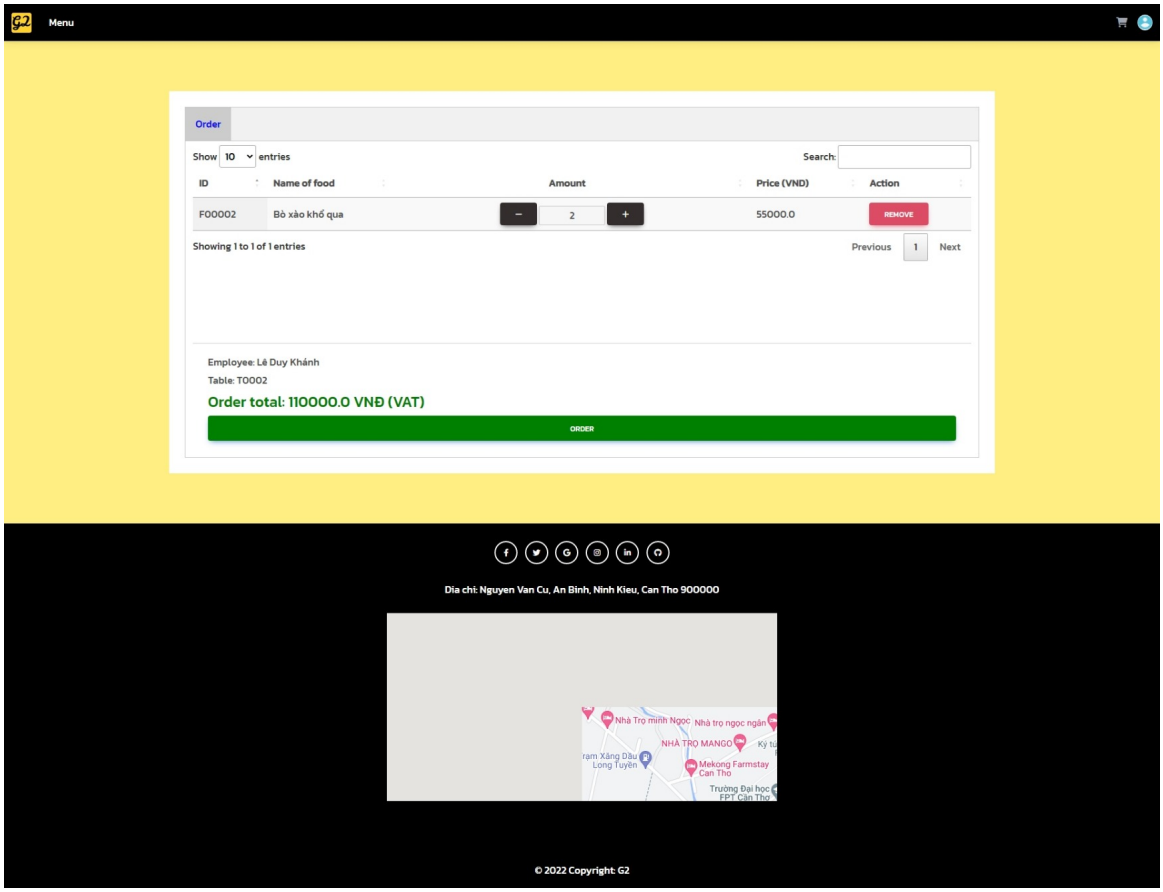

Figure 14

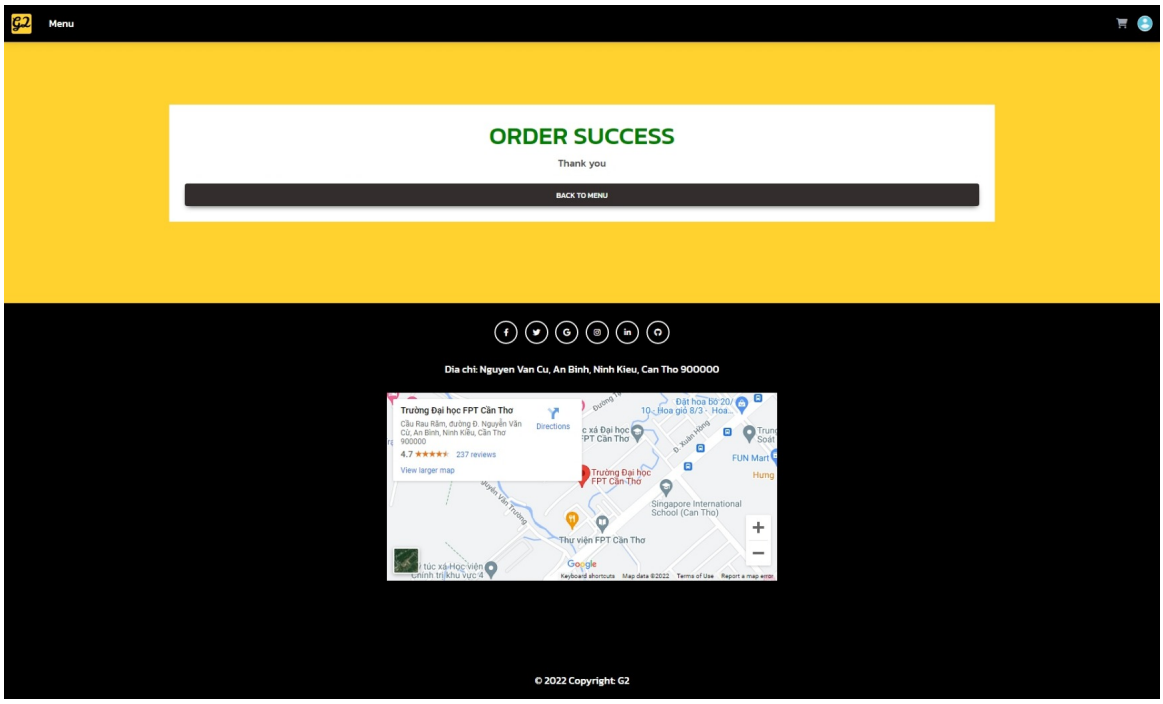

Figure 15

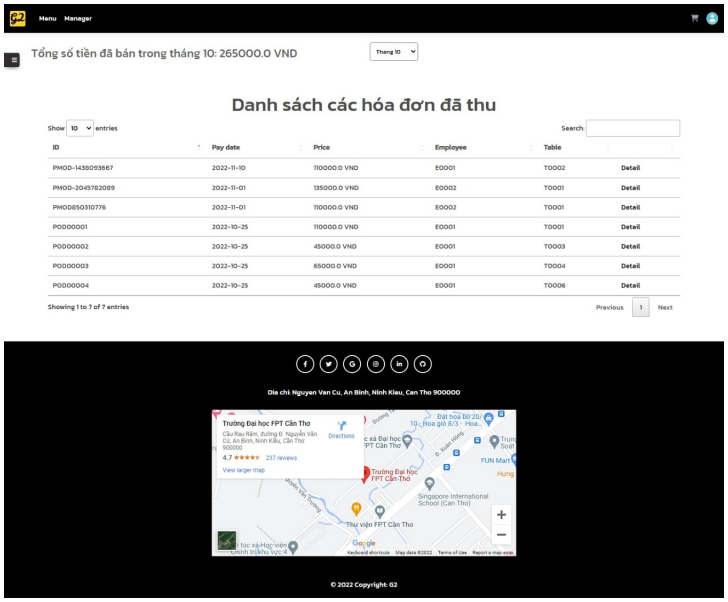

Figure 16

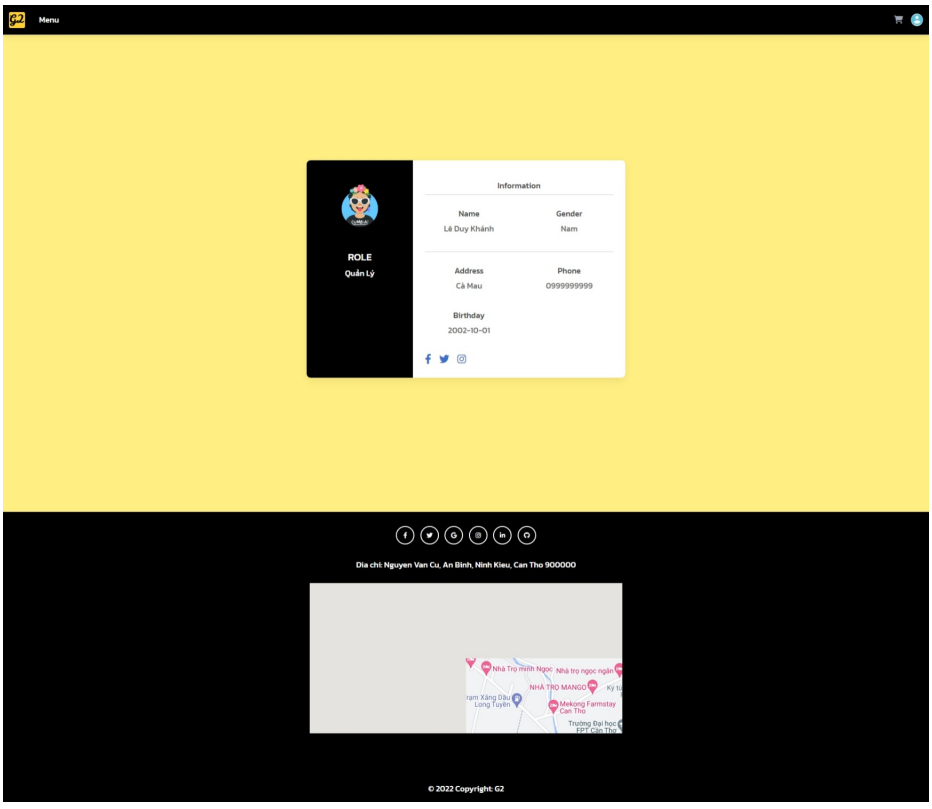

Figure 17

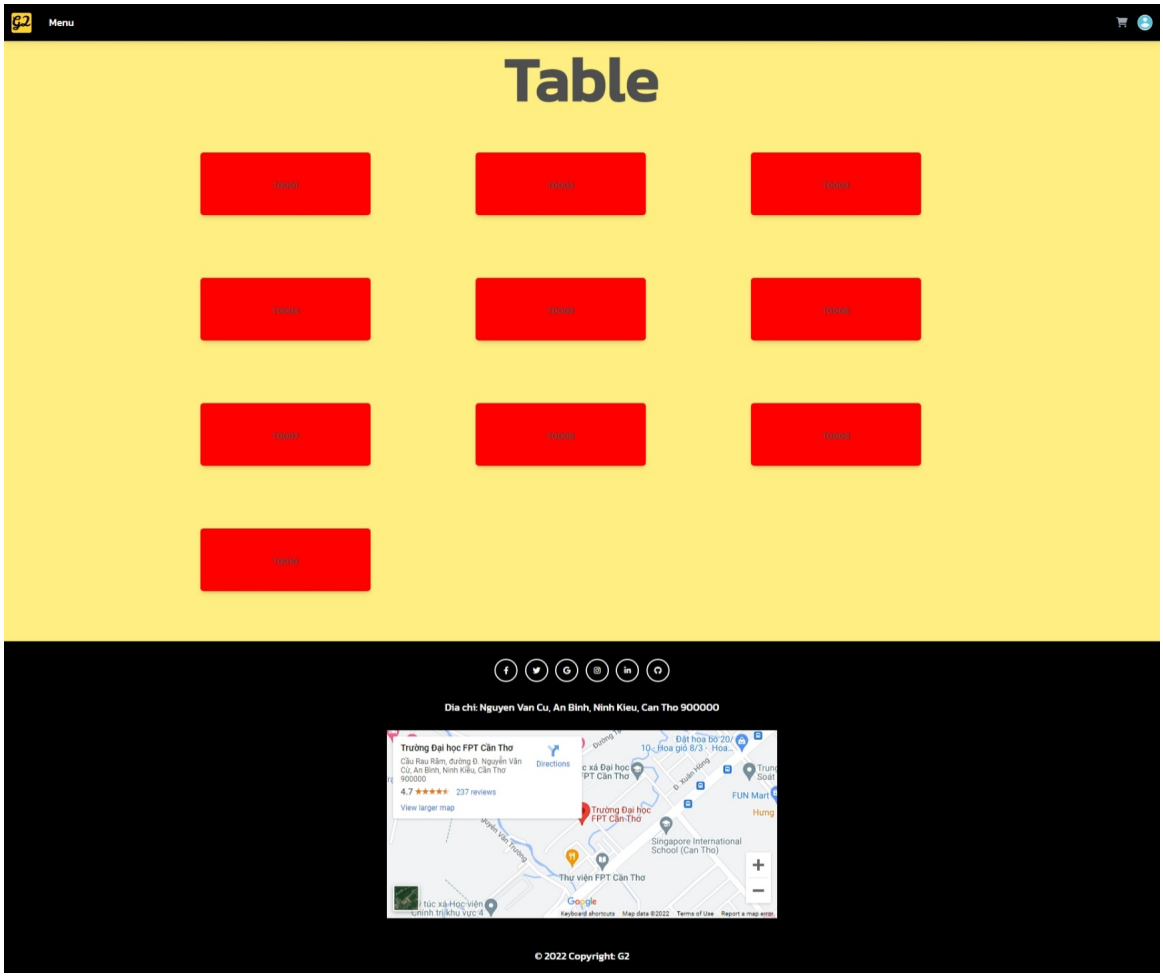

Figure 18

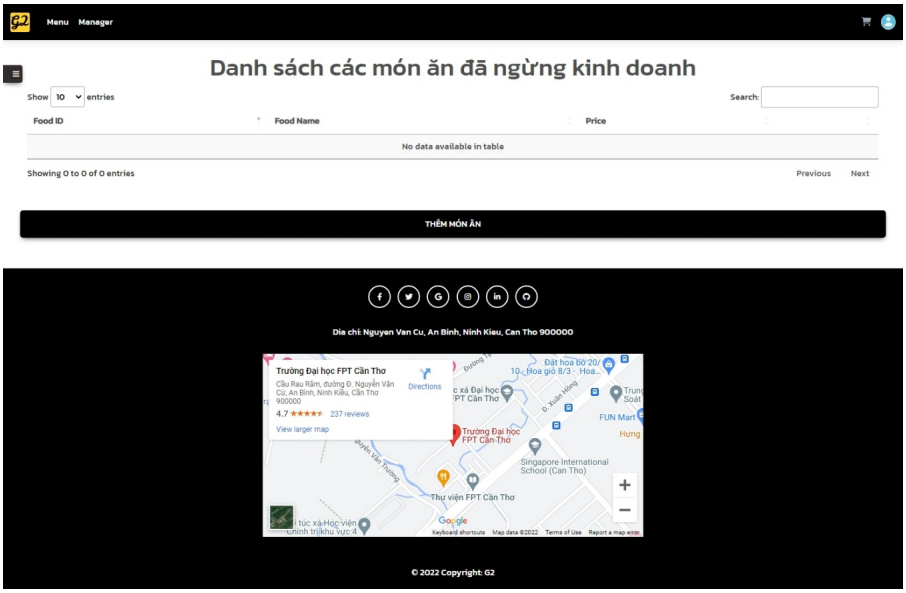

Figure 19