# Design

After the competition of analysis phases, we begin the design phase. During this phase we try to portray the use case of the system using the object-based diagrams or the design models. Or in this stage we try to represent our implementation and source code into a diagrammatic style so that it helps anyone to get a clear representation of that system just by looking at the diagrams.

I’ve constructed four different models for the representation which are structural modelling, behavioural modelling, database modelling and prototype design.

* 1. Structural Modelling
     1. Class Diagram
     2. Data flow Diagram

Data flow diagram is a basic diagrammatic representation of flow of data during a process in a system. Just like all other diagrams, DFD uses different symbols like rectangle, circles, labels to show input, output, storage points and routes between destinations. Below is my data flow diagram with the explanation of the notations used.

|  |  |
| --- | --- |
| Notations used | Notations description |
|  | It can be a human, system or a subsystem. |
|  | It is a business activity or function where data manipulation takes place. |
|  | It represents the storage of data that is required by the processes. |
|  | It denotes the flow of information. |

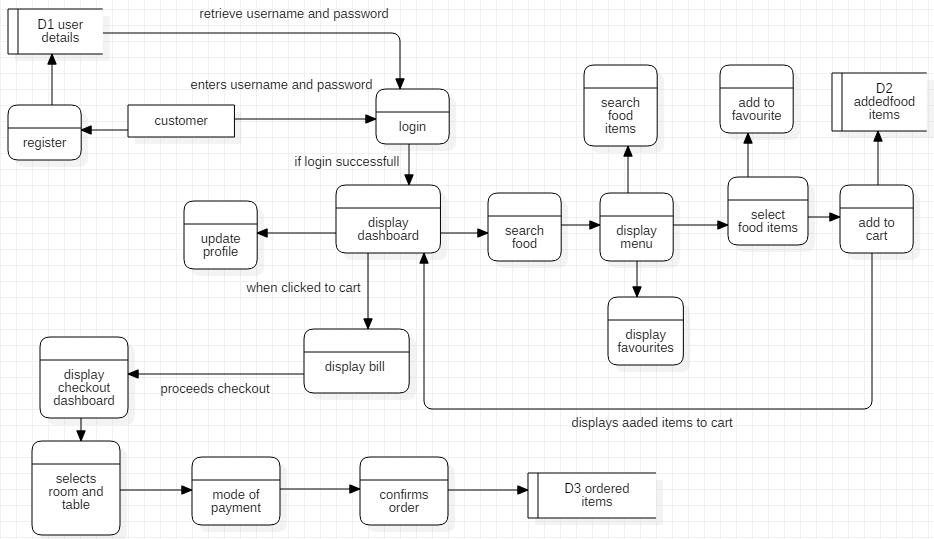


Figure 1:Data Flow Diagram

* 1. Behavioural Modelling
     1. Activity Diagram

Activity diagram tends to provide coordination between different activities on a system. Below are my activity diagrams for various operations with notations that are used in the diagram.

|  |  |
| --- | --- |
| UML Notations | Notations description |
|  | It denotes the beginning of actions or activities. |
|  | It denotes a task that is to be performed. |
|  | Split activities into different pair of parallel or same flow of activities. |
|  | Bring back the separated activities. |
|  | Used to test a condition. |
|  | Bring back activities that were separated by decision nodes. |
|  | It denotes the sequence of the action’s execution. |
|  | A way to group different activities performed by the actor. |
|  | It sends signal outside the activity and continues without waiting for any responses. |
|  | Receives the signal sent by sent signal. |
|  | It is used to denote the acceptance of time event. |
|  | Ends all the activities and processes. |

Table 1: Notations used in activity diagram

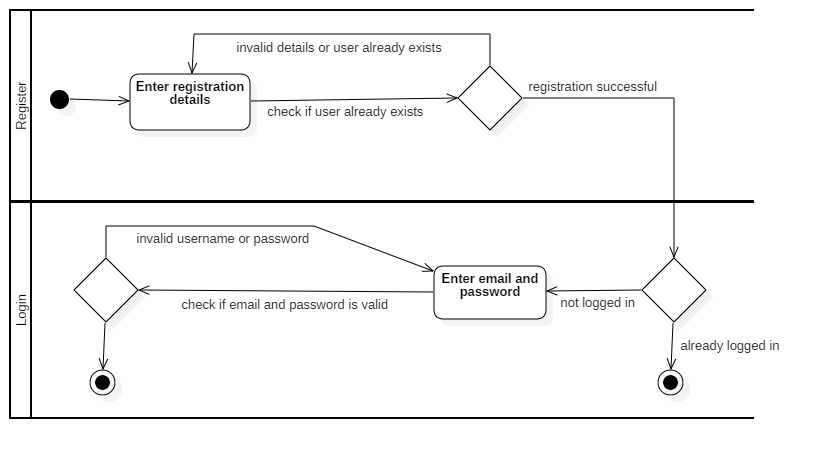
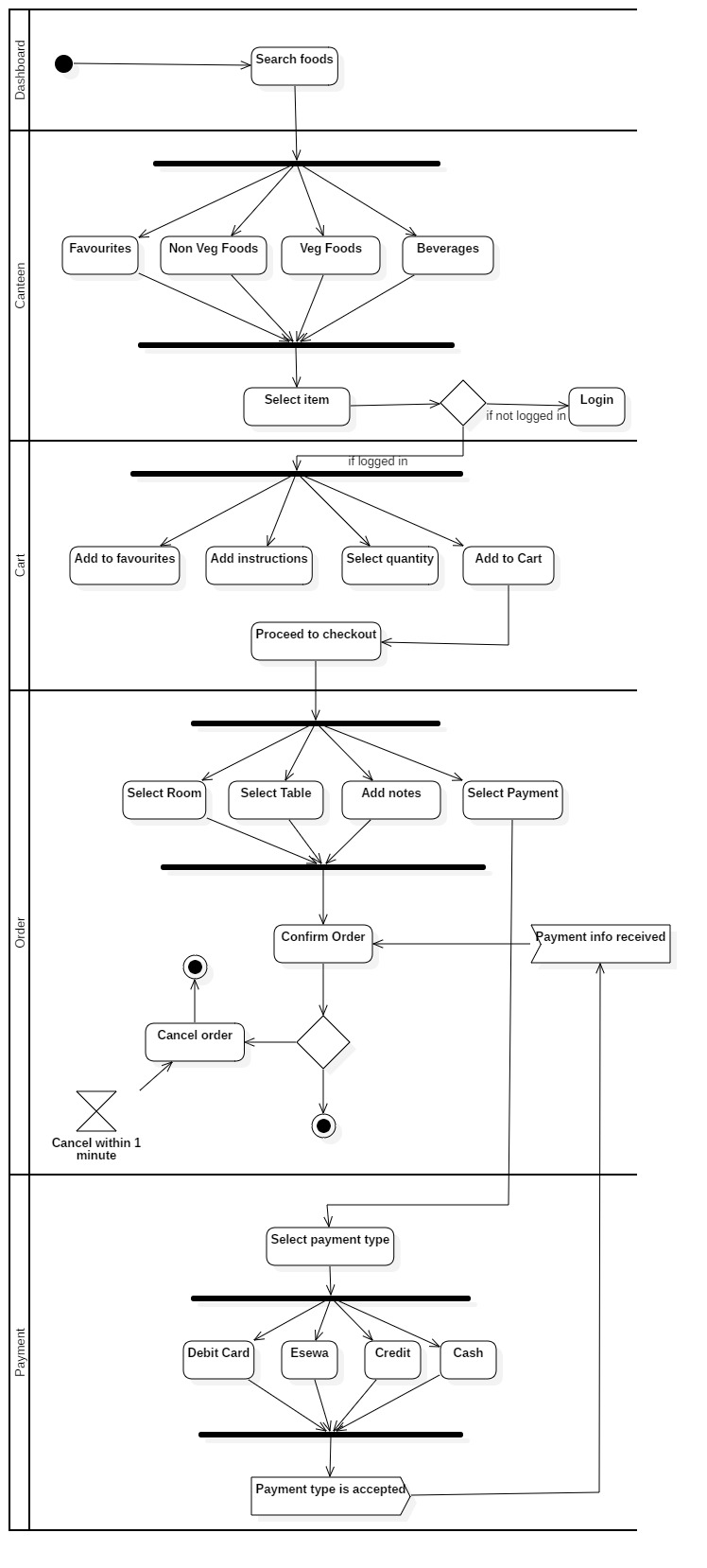


Figure 2: Activity diagram for login and registration

Following activities happen in the above activity diagram:

* During registration, user enters the registration details.
* After that, system checks if the email or the user already exists.
* If not, then the user is registered successfully.
* During login also, after user enters email and password, system checks if it is valid or not.
* If the username and passwords are correct, then the user is logged in.



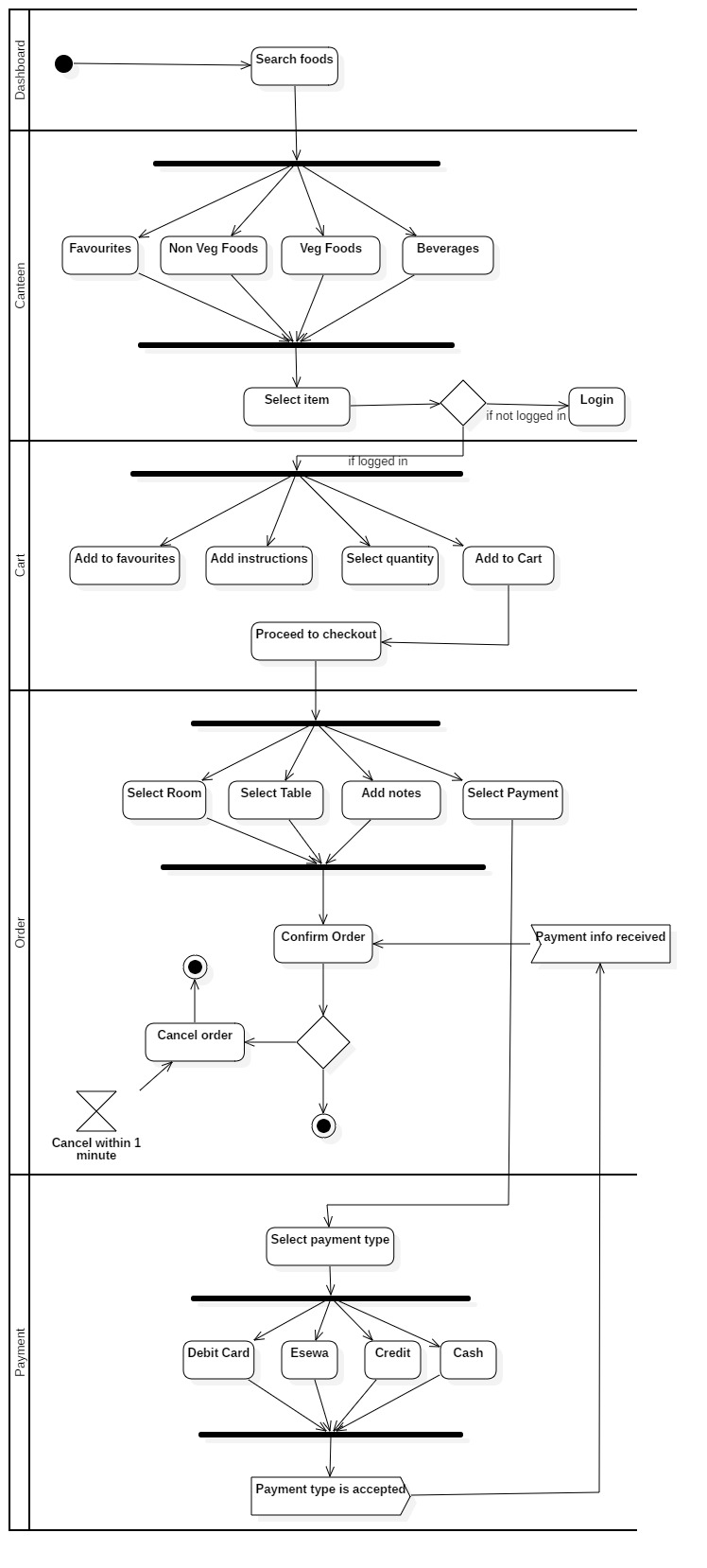


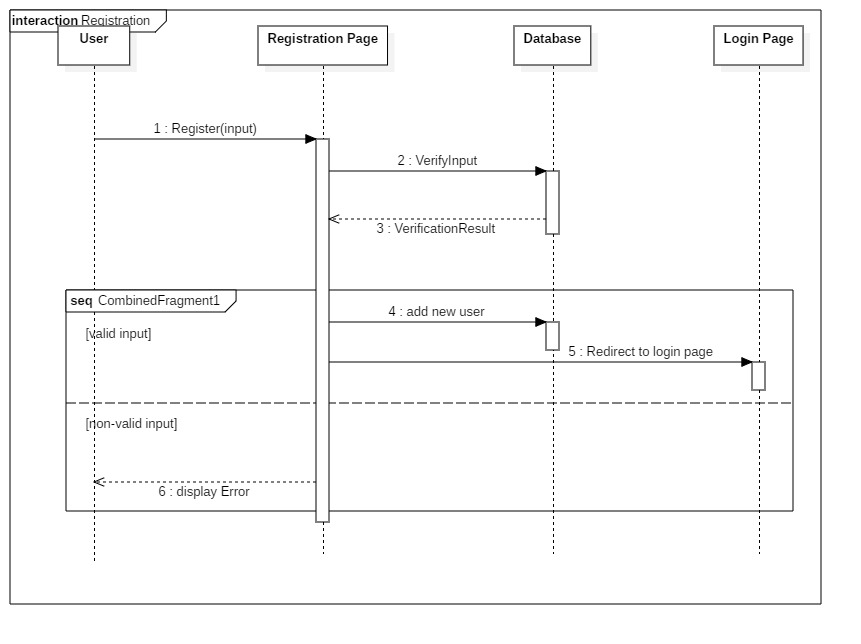
Figure 3: Activity diagram for order and payment

Following activities happen in the above activity diagram:

* At first user searches food in the landing page and gets into the canteen page.
* User can browse menu with different choices and options.
* As the user selects an item from the menu, he’ll be prompted to a login screen if he’s not logged in.
* If the user is already logged in, he’ll get different choices to add to favourites, add notes and add the item to his cart.
* From the cart, user can proceed to checkout where he’ll get different options like selecting room and table and the payment options.
* After everything is done, user can confirm the order and the user will get 1minute window to cancel the order if he likes.
  + 1. Sequence Diagram

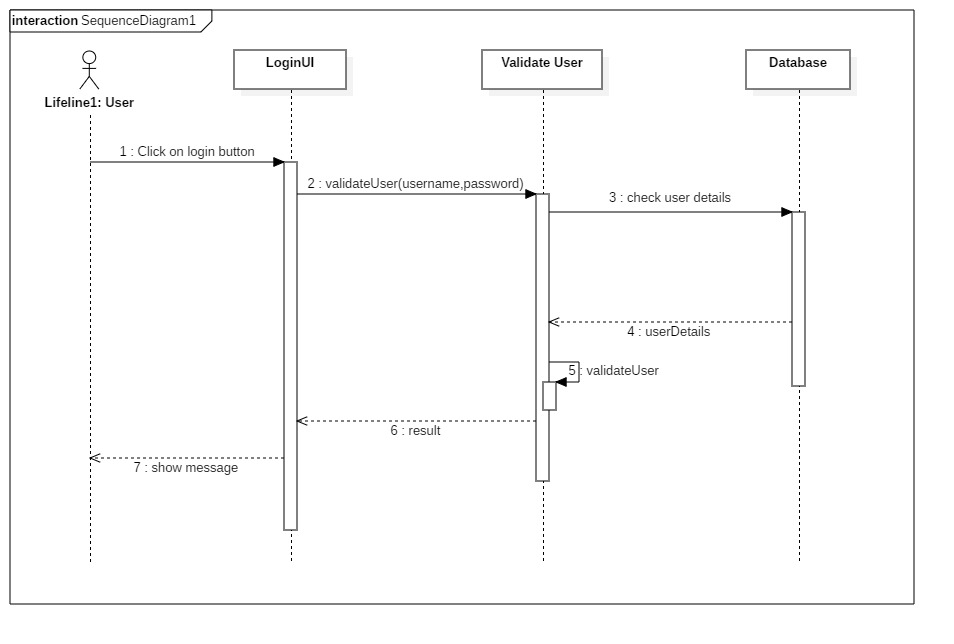
Sequence diagram shows how the process interact with each other and in what order they interact. They’re also called event diagrams sometimes. Below are my sequence diagrams for various operations with notations that are used in the diagram.

|  |  |
| --- | --- |
| UML Notations | Notations Description |
| Actor | A role played by an entity that interacts with the user. |
|  | Denotes the participation of an individual. |
|  | Denotes a communication between lifelines. |
|  | It is the reply back of the previous message to the caller. |
|  | Invocation of the message on the same lifeline. |
|  | Denotes logical groupings. |



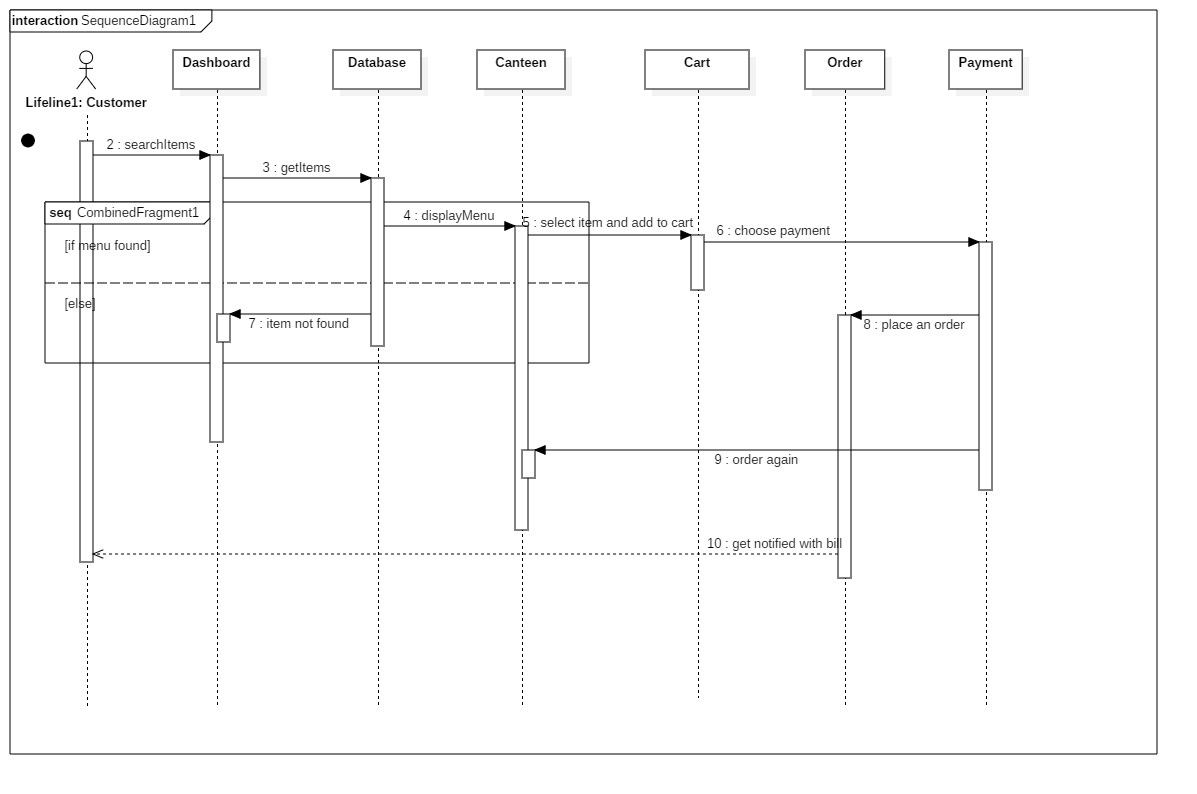
Following events happen in the above sequence diagram:

* At first, user inputs the details in registration page.
* The details are then verified with the database for existing users.
* If the given details are valid, user will be redirected to a login page otherwise the user will be redirected on the same page showing the error.



Following events happen in the above sequence diagram:

* After entering the email and password, user clicks on the login button.
* Then the email and passwords are validated by checking user details on the server.
* If the details are correct, then the user will be redirected to a dashboard otherwise it’ll display the error to the user on the LoginUI.



Following events happen in the above sequence diagram:

* On the dashboard, user searches for food which will be looked up on the database.
* If the food is found, the user will be redirected on the respective food’s page or canteen, otherwise he’ll get error message saying food not found.
* If the food is found, the user can add the food of his choice into the card and proceed for payment and checkout.
* After checking out, the user will be notified with his bill.
  1. Data Modelling
     1. Data Dictionary

Data dictionary holds information of the data in the database like data ownership, relationships of the data with other objects and other data. It holds the metadata of the database.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Customer | | |
| Description | This table will store data of all the users who register into the system. | | |
| Primary Key | CustomerID | | |
| Foreign Keys | CollegeID, RoleID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| CustomerID | integer(10) | PK |  |
| fname | varchar(25) | Not Null |  |
| lname | varchar(25) | Not Null |  |
| email | varchar(25) | Not Null |  |
| phone | varchar(25) | Not Null |  |
| CollegeID | integer(10) | FK |  |
| RoleID | integer(10) | FK |  |
| InstituteID | integer(10) | Not Null |  |
| password | varchar(50) |  | Must contain 8 characters. |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | College | | |
| Description | This table will contain all the college that the web application is associated with. | | |
| Primary Key | CollegeID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| CollegeID | integer(10) | PK |  |
| CollegeName | varchar(50) |  |  |
| CollegeAddress | varchar(50) |  |  |
| CollegePhone | varchar(25) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Canteen | | |
| Description | This canteen holds the canteen data. | | |
| Primary Key | CanteenID | | |
| Foreign Keys | CollegeID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| CanteenID | integer(10) | PK |  |
| CanteenName | varchar(50) |  |  |
| CollegeID | integer(10) | FK |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Role | | |
| Description | This table holds the data of different roles of users in the organization. | | |
| Primary Key | RoleID | | |
| Foreign Keys | CollegeID, CanteenID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| RoleID | integer(10) | PK |  |
| Role | varchar(50) |  |  |
| CollegeID | integer(10) | FK |  |
| CanteenID | integer(10) | FK |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Item | | |
| Description | All the items or food that are available in the canteen are stored in this table. | | |
| Primary Key | itemID | | |
| Foreign Keys | CanteenID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| itemID | integer(10) | PK |  |
| itemName | varchar(50) |  |  |
| price | decimal |  |  |
| CanteenID | integer(10) | FK |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Cart | | |
| Description | This table will store the items that are added to the cart. | | |
| Primary Key | cartID | | |
| Foreign Keys | CustomerID, itemID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| cartID | integer(10) | PK |  |
| CustomerID | integer(10) | FK |  |
| itemID | integer(10) | FK |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Order | | |
| Description | All the orders that are made are stored in this table. | | |
| Primary Key | OrderID | | |
| Foreign Keys | itemID, CustomerID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| OrderID | integer(10) | PK |  |
| itemID | integer(10) | FK |  |
| CustomerID | integer(10) | FK |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Room | | |
| Description | Room data will be stored in this table. | | |
| Primary Key | RoomID | | |
| Foreign Keys | CanteenID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| RoomID | integer(10) | PK |  |
| Room\_name | varchar(10) | NN |  |
| Canteen\_ID | integer(10) | FK |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Table | | |
| Description | Table data will be stored in this table | | |
| Primary Key | TableID | | |
| Foreign Keys | RoomID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| TableID | integer(10) |  |  |
| Table\_Name | varchar(10) |  |  |
| RoomID | integer(10) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Favourites | | |
| Description | This table will store all the favourite items of the customers. | | |
| Primary Key | FavouritesID | | |
| Foreign Keys | CustomerID, itemID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| FavouritesID | integer(10) | PK |  |
| CustomerID | integer(10) | FK |  |
| itemID | integer(10) | FK |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Payment | | |
| Description | All the payment data will be stored in this table. | | |
| Primary Key | PaymentID | | |
| Foreign Keys | CustomerID, OrderID, PaymentTypeID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| PaymentID | integer(10) | PK |  |
| CustomerID | integer(10) | FK |  |
| OrderID | integer(10) | FK |  |
| PaymentTypeID | integer(10) | FK |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | PaymentType | | |
| Description | Information related to payment type is stored in this table. | | |
| Primary Key | PaymentTypeID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| PaymentTypeID | integer(10) | PK |  |
| PaymentType | Varchar(50) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Credits | | |
| Description | All the customers who have due payment will be stored here. | | |
| Primary Key | CreditID | | |
| Foreign Keys | CustomerID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| CreditID | integer(10) | PK |  |
| credit\_date | date |  |  |
| CustomerID | integer(10) | Fk |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | notification | | |
| Description | This table will store the notifications that are to be sent to users on certain events. | | |
| Primary Key | NotificationID | | |
| Foreign Keys | CustomerID | | |
|  | | | |
| Field Name | Data Type | Constraints | Remarks |
| NotificationID | integer(10) | PK |  |
| CustomerID | integer(10) | FK |  |
| Notification | varchar(255) |  |  |
| noti\_date | datetime |  |  |

* + 1. ER Diagram

An ER diagram or Entity Relationship diagram is a diagram that graphically illustrates the relationship between different entities in a database. Below is my ER diagram of the database I’ve designed for my system along with the explanation of the notations used.

|  |  |
| --- | --- |
| UML Notations | Notations Description |
|  | Entity can be considered a table that holds attributes that are rows and columns. |
|  | Denotes one to one relationship. |
|  | Denotes one to many relationships. |
|  | Denotes many to many relationships. |

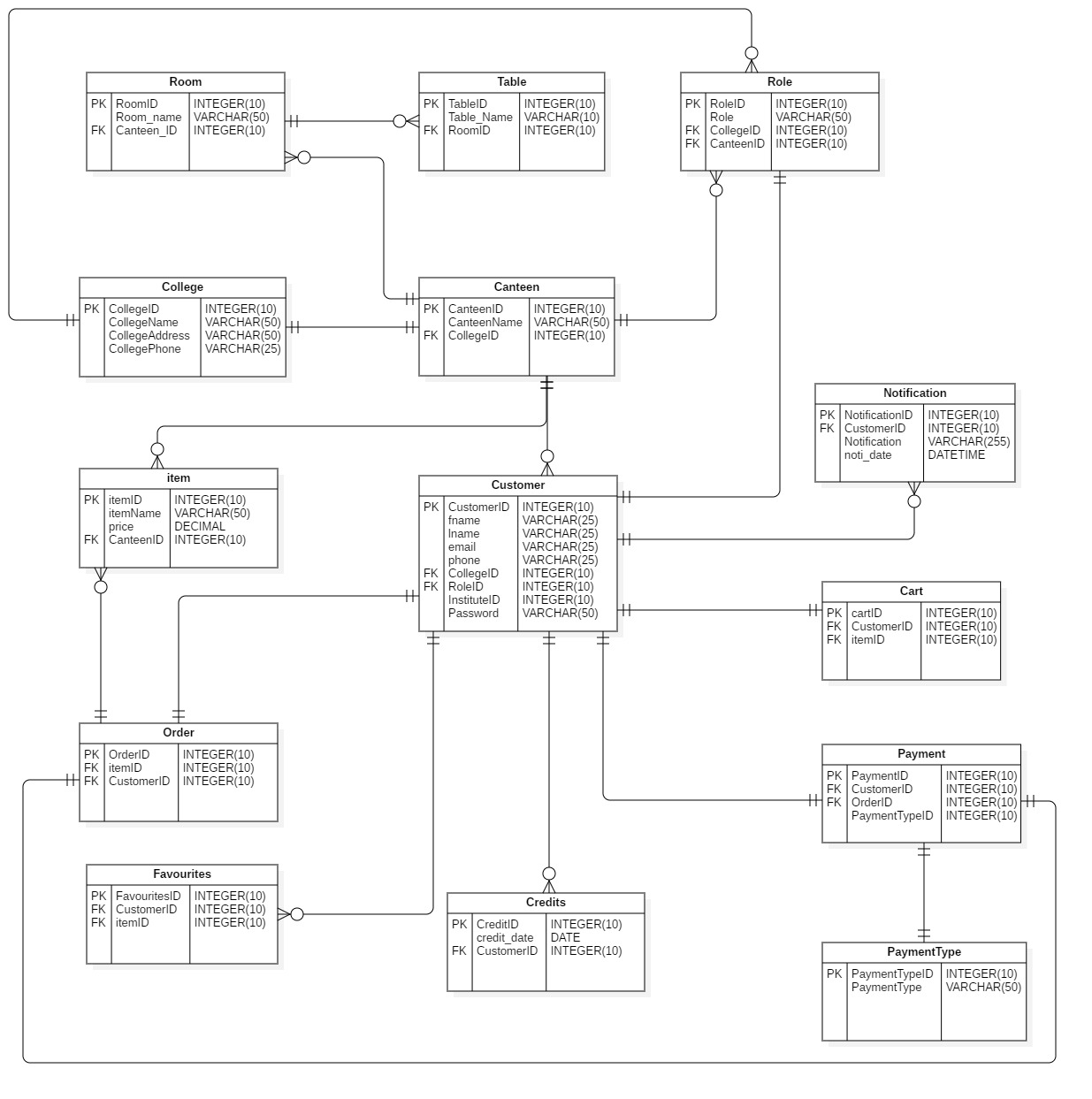


Figure 4:ER Diagram

* 1. Prototype

A prototype gives a clear representation on how the product looks so that the user can get a better understanding of the user interface. It helps the customer or the client to know exactly how the system looks. I’ve done a digital prototype for the canteen management system using the software Adobe XD. Below are the snapshots of the prototype.

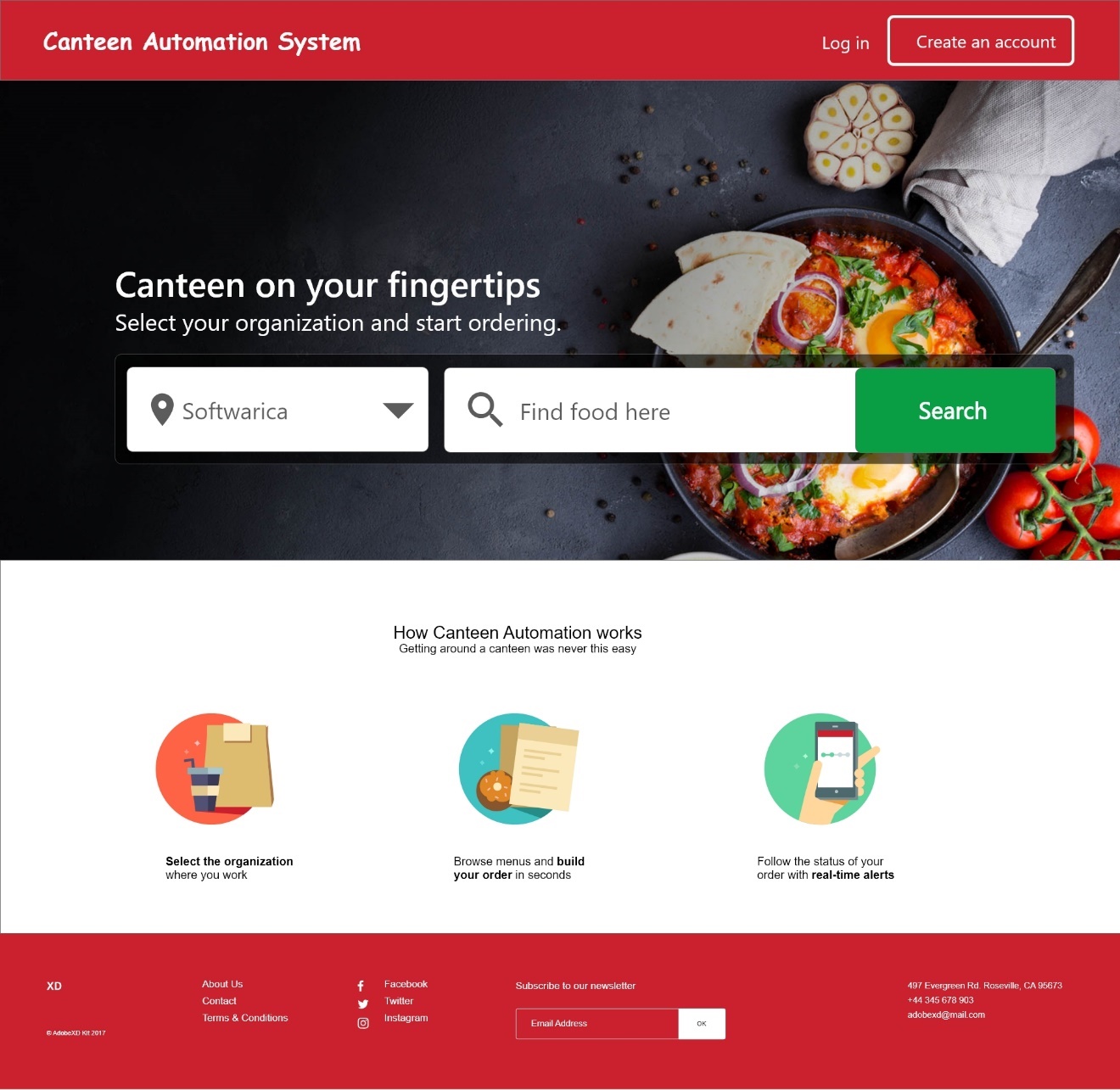


Figure 5:Landing Page

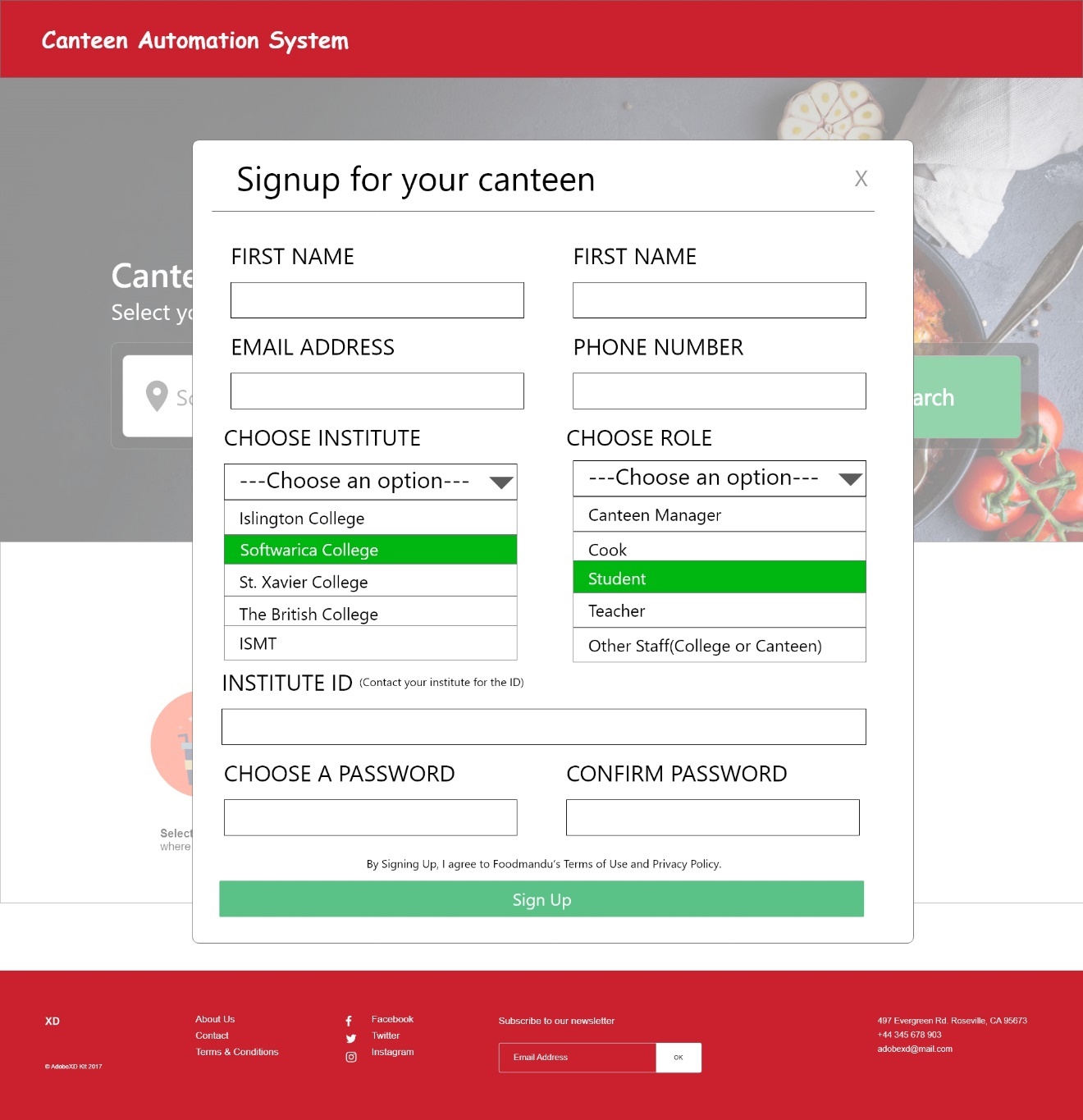


Figure 6:Registration page

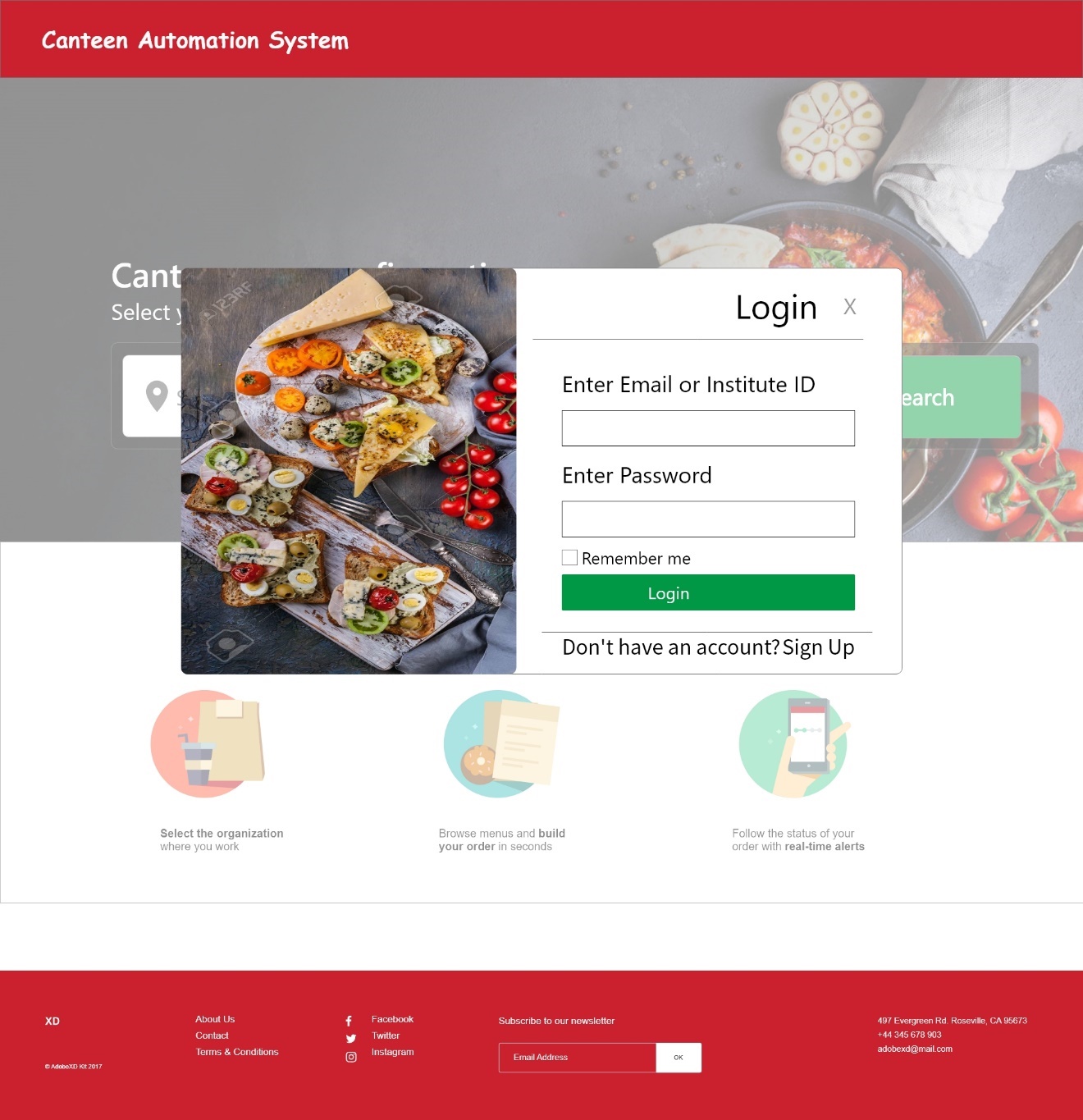


Figure 7:Login Page

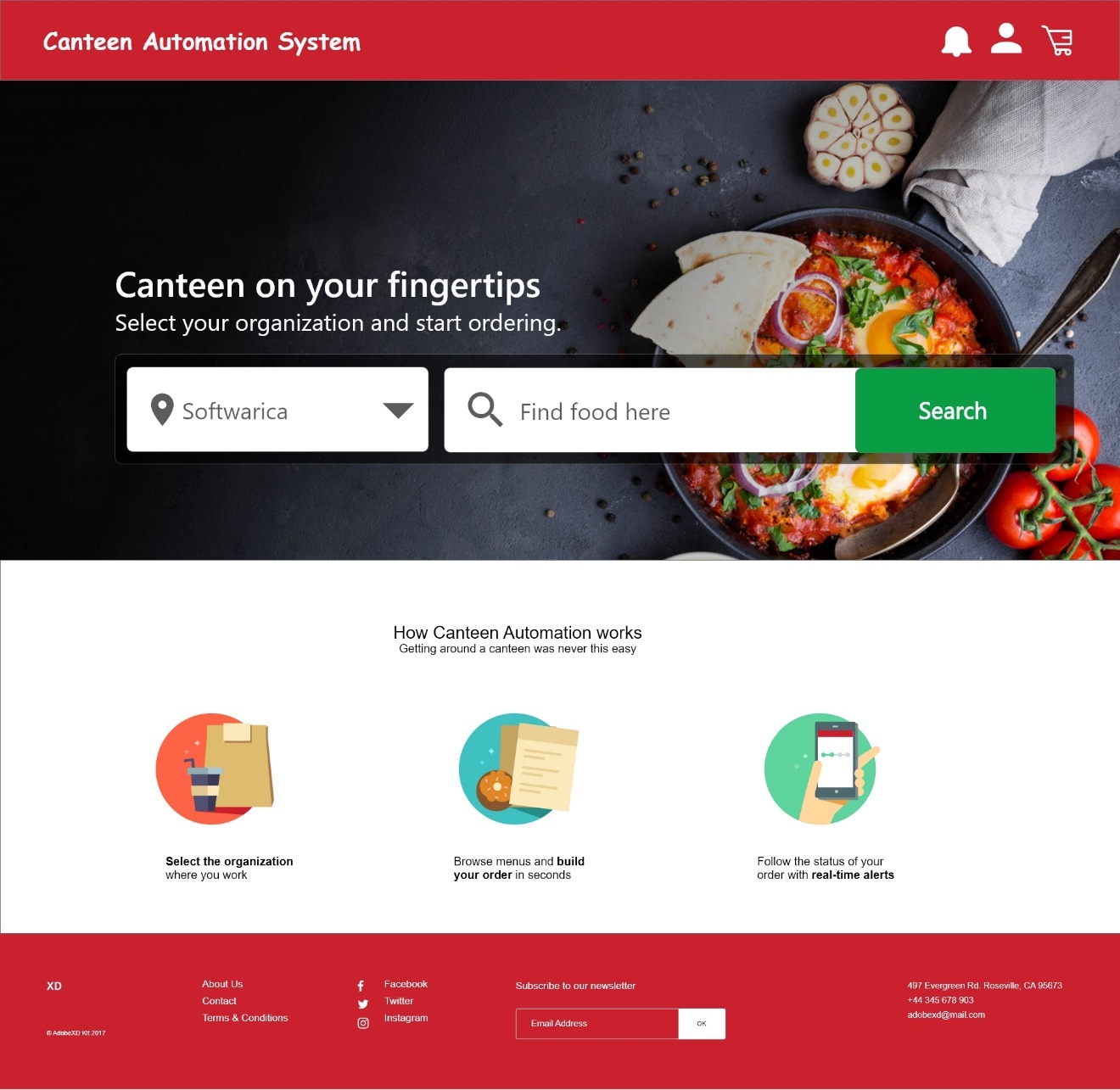


Figure 8:User profile and notifications after login

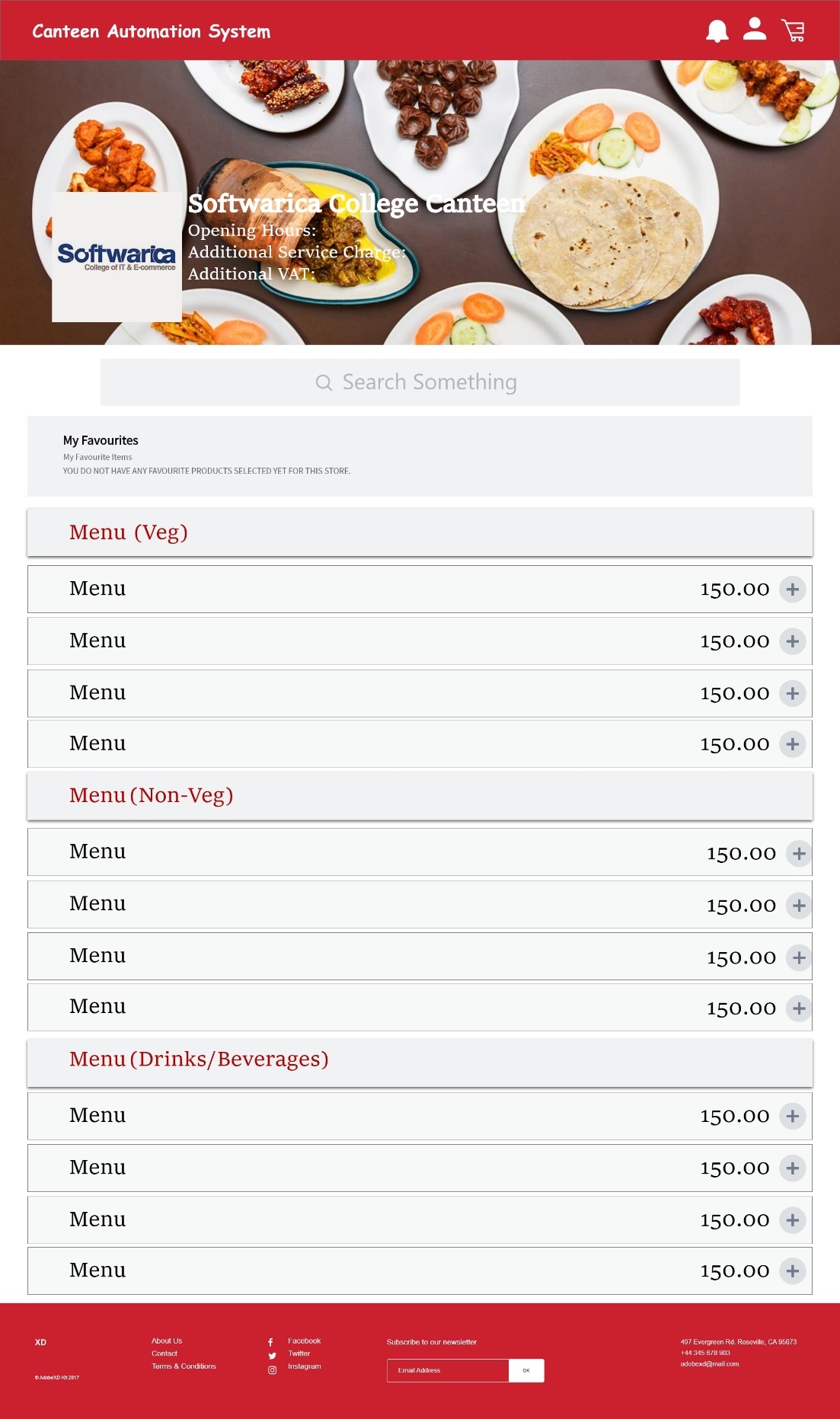


Figure 9:Canteen menu page

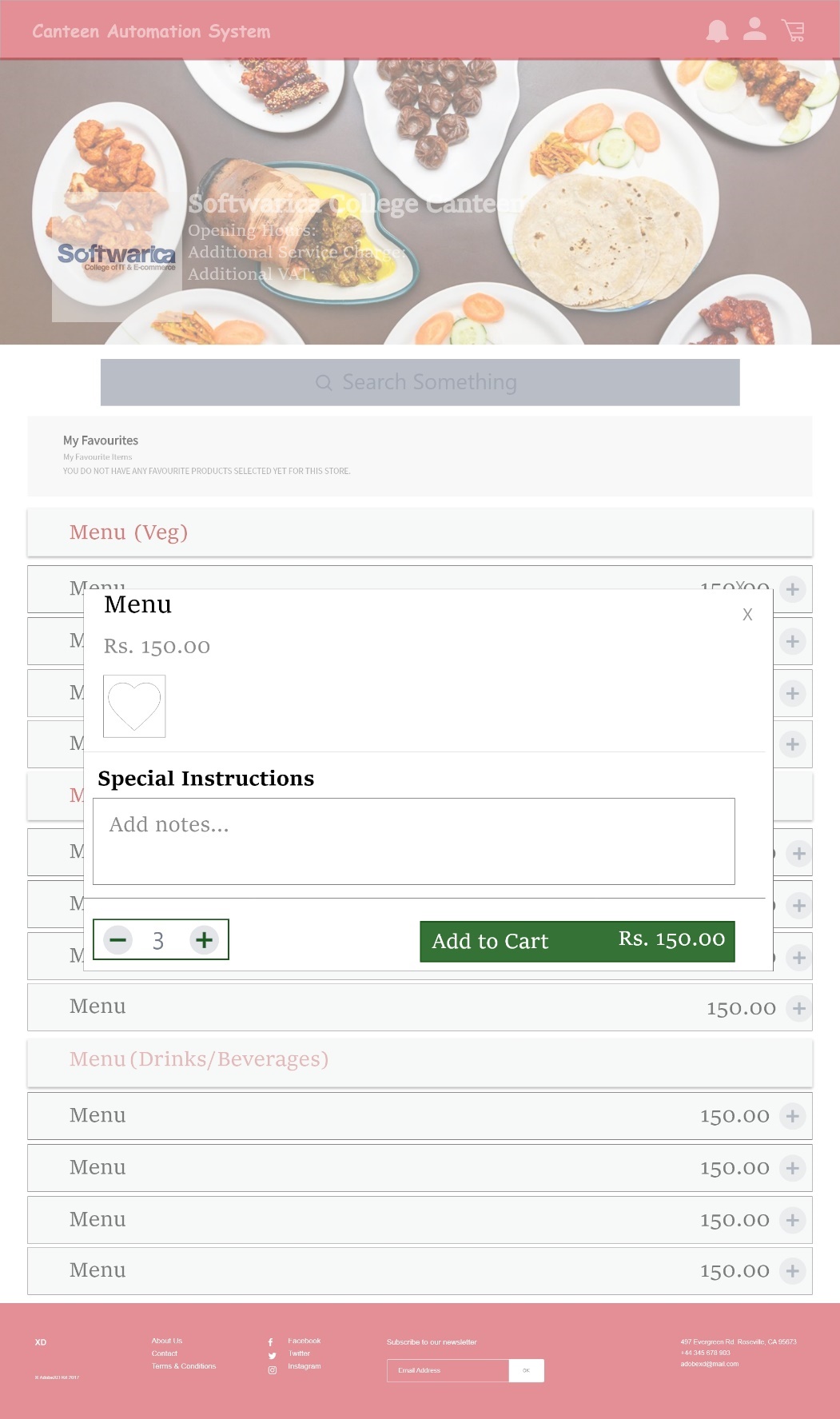


Figure 10:Selecting an item from the menu

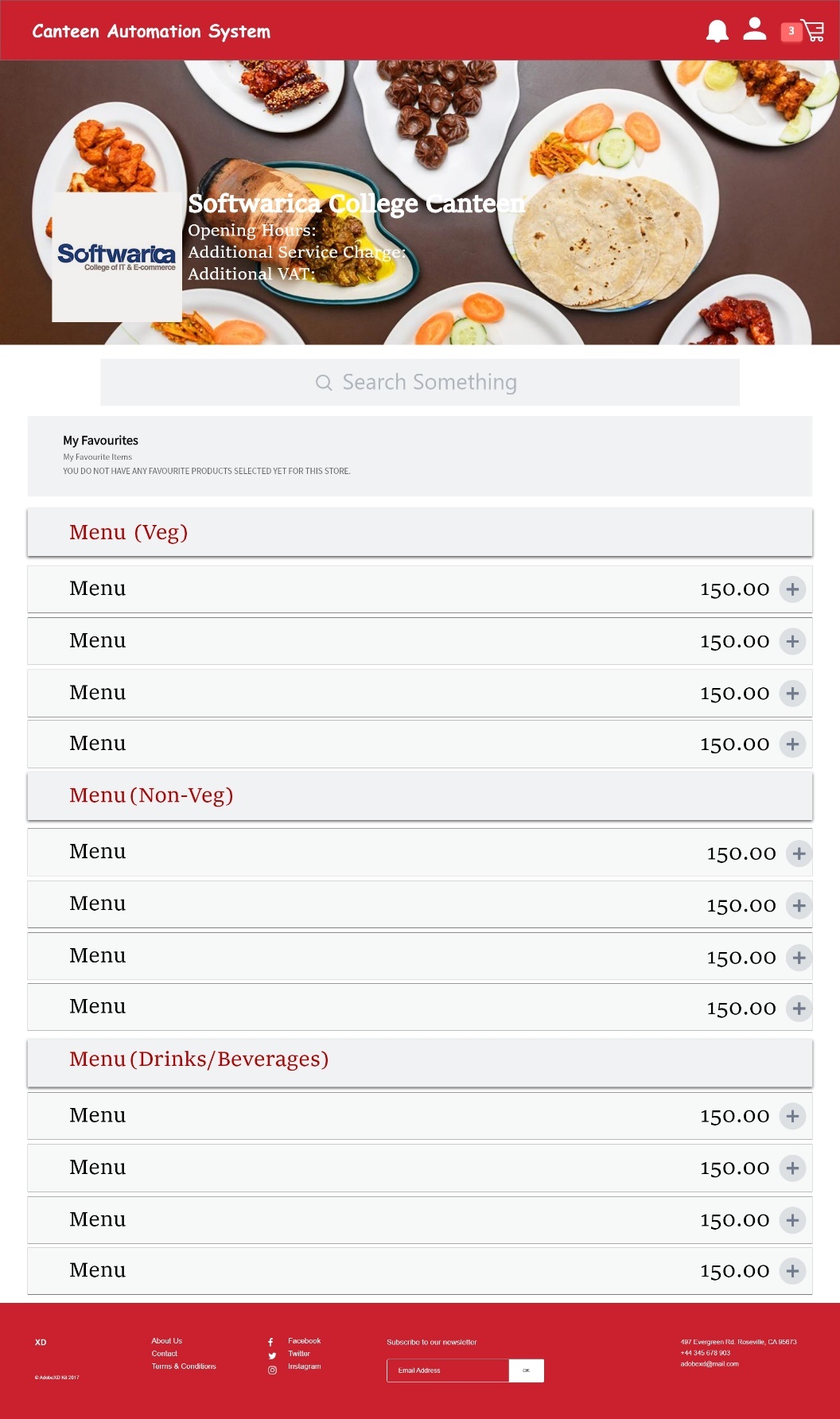


Figure 11: Cart icon after adding items into the cart

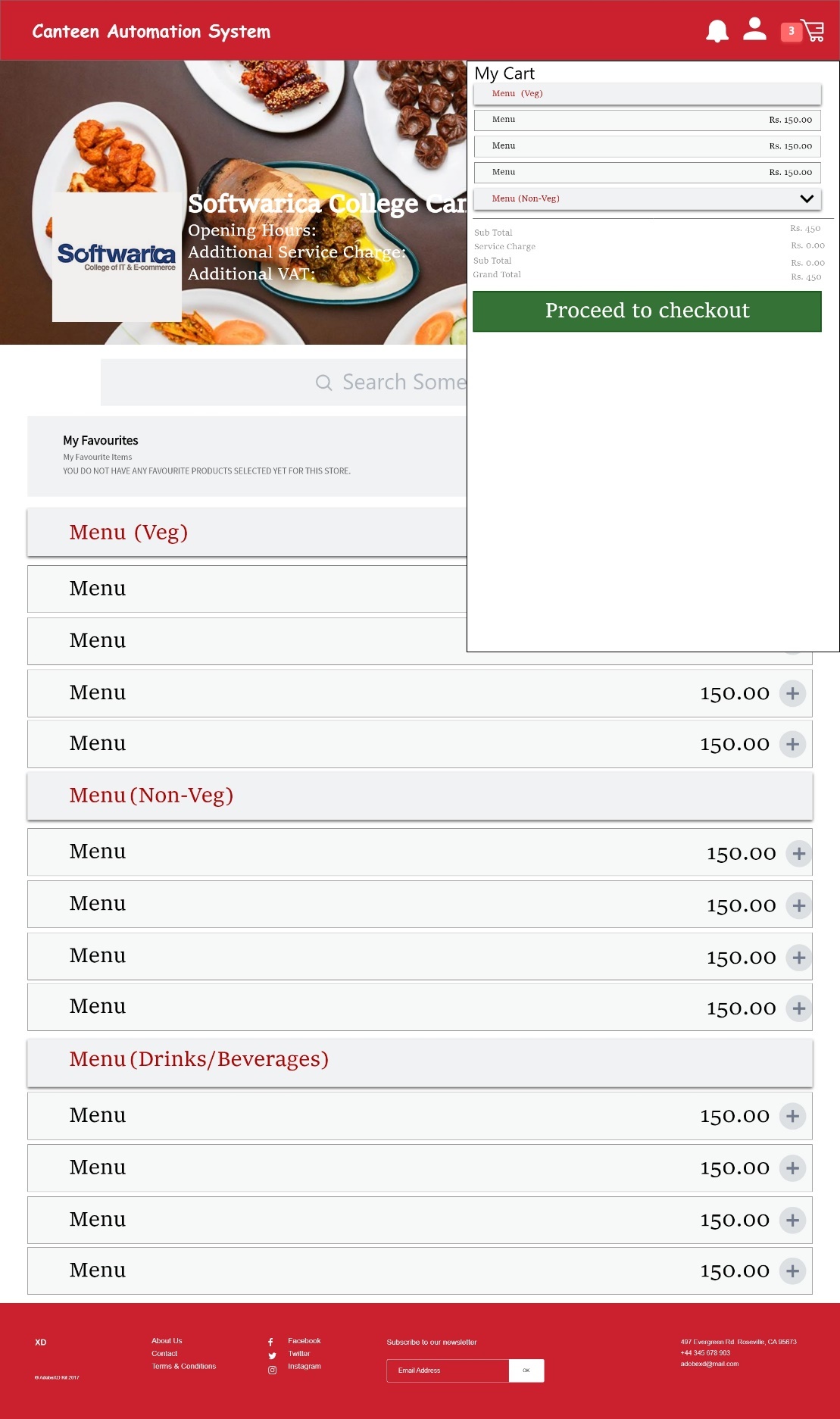


Figure 12: Cart menu

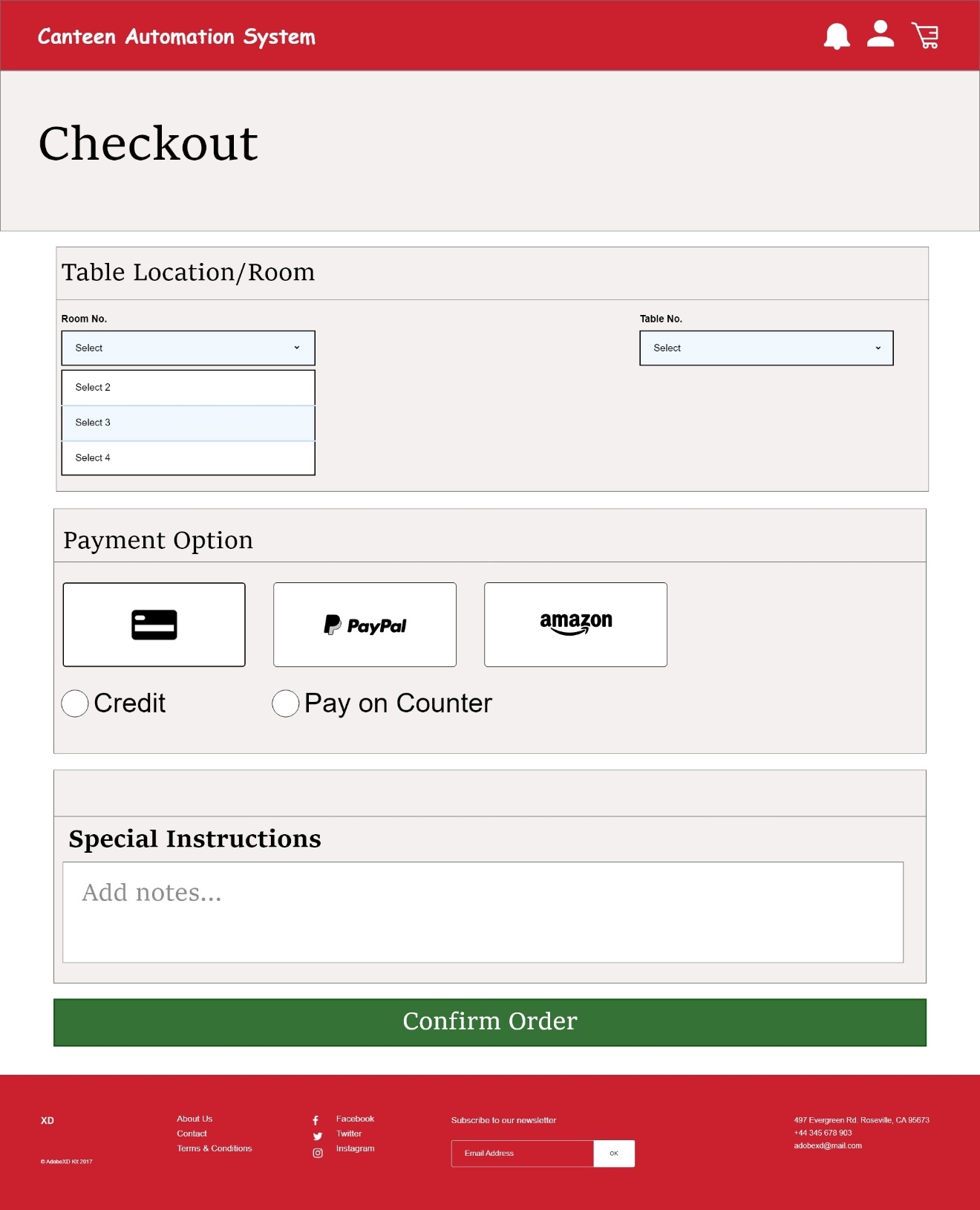


Figure 13:Checkout page