Requirements, Use Cases and Domain Models

Team Number:	1
Project Name:	ClassDASH
Analysts:	Zikai Hao, Pengfei Li, Shivani Ram
Clients:	Coby Lam, Emile Keruzore

Requirements

The following table lists ClassDASH's functional and nonfunctional requirements which define the scope of the project. Requirements are prioritized using the MoSCoW method wherein requirements are categorized as M - Must have, S - Should have, C - Could have, W - Won't have.

Functional requirements are organized into five categories: *Food Orders, Payment Related, Application Functions, and Food Worker-side Functions.*

Non-functional requirements are organized by *Standards, Safety-Related, Format-Related, and Accessibility.*

I. Functional Requirements		**M**	**S**	**C**	**W**
Food Orders					
1.1	All customers can order food via the mobile application	X			
1.2	Customers and workers should be able to have a browsing section with all the foods available and time restrictions on the menu	X			
1.3	Customers must be able to use a search bar to find specific items	X			
1.4	Customers must be able to view recommended orders based off of prior orders	X			
1.5	Customers can leave a note in regards to any food allergies, preferences or customize options for a selected item(eg: burger, no cheese, no lettuce, extra sauce)	X			
1.6	The Customer will be able to review their order before confirming	X			
Payment Related					
2.1	All Customers can pay for their food using their ONECard, credit card, or debit card	X			
2.2	Users will create a system account to log in/manage their account and payments		X		
Application Functions					

I. Functional Requirements		**M**	**S**	**C**	**W**
3.3	An order through the app will have a unique identifier to ensure the right Customer receives their order (order ID)	X			
3.4	The application should mark an order as having been processed and remove it from the list of active orders		X		
3.5	The application should actively update: presenting up-to-date menus to customers and retrieving new orders from the database for workers		X		
3.6	The application should alert Workers when a packed order has been sitting too long and poses a risk		X		
Food Worker-side Functions					
4.1	The system will have a food service interface that displays the orders (and the order ID) in a queue for Workers	X			
4.2	Workers must be able to create service-side accounts to manage their menus (mark items sold out/new items/new descriptions or prices)	X			
4.3	Workers can cancel the food order via the worker-facing side of the application if the food is not yet under preparation	X			
4.4	The app should pass basic security tests when transmitting information to and from application and databases		X		

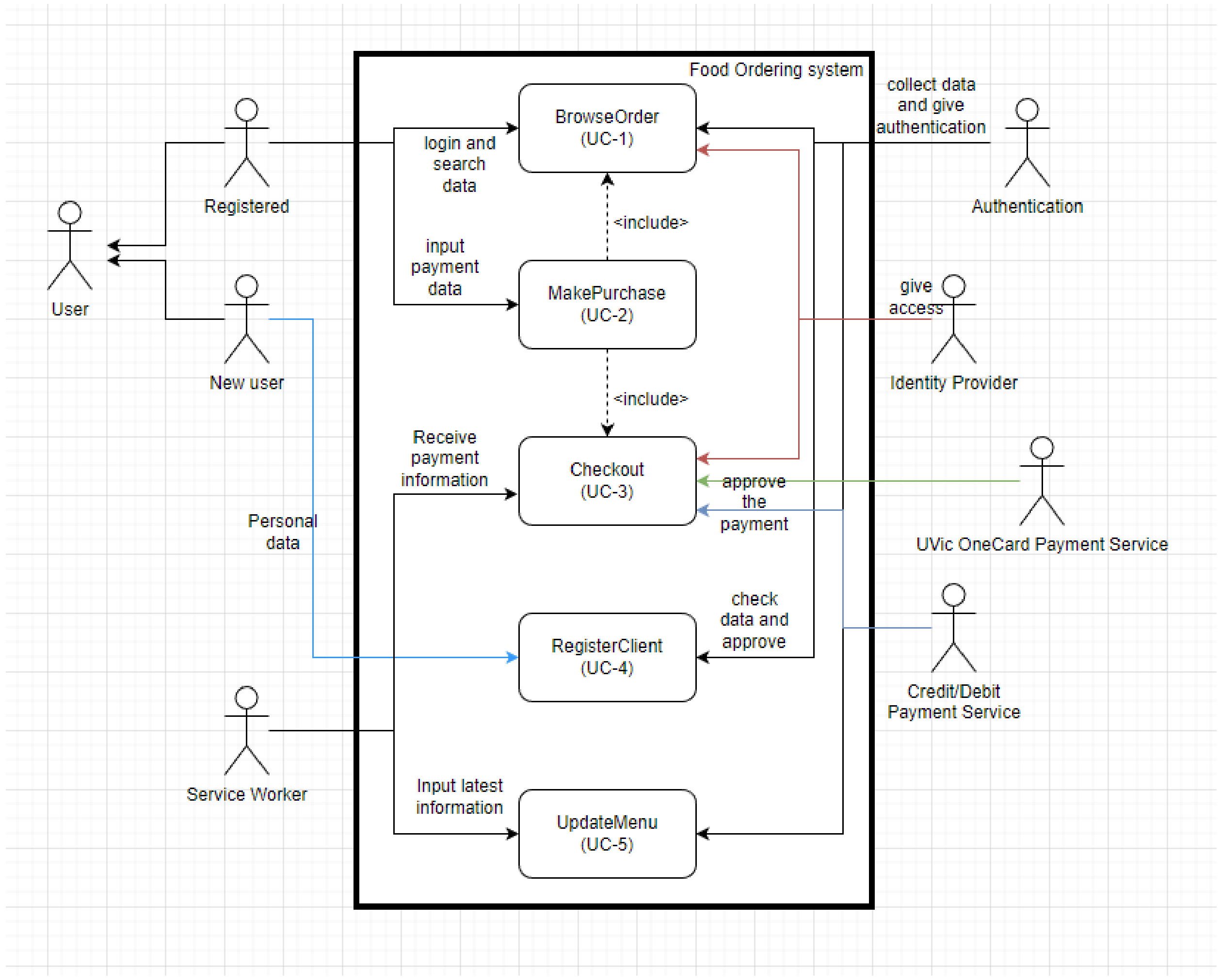
II. Non-Functional Requirements		**M**	**S**	**C**	**W**
Standards					
1.1	The application should have a minimum buffering time to ensure smooth user interactions		X		
1.2	The application must be clean and simple looking design for easy navigation	X			
1.3	The system should offer a faster ordering process than standing in line at peak hours		X		
Safety Related					
2.1	The system should encrypted users' personal information and financial transactions in ASCII standard		X		
Format Related					
3.2	The System produces standard electronic receipts for users after order submissions			X	
3.3	The food pick-up confirmation should be based on a QR-code system		X		
3.4	The format of time counting for food preparation will be a 24-hour system			X	
3.5	All currencies in the payment processes must be calculated in Canadian dollar	X			
Accessibility					

II. Non-Functional Requirements		M	S	C	W
4.1	The system must be accessible on both Android and iOS mobile systems	X			

Use Cases

The following depicts the top-level system diagram that highlights the main actors and the actions available to them. The main actors include Users, both new and previously-registered, and Food Service Workers.

UC-0: Top level



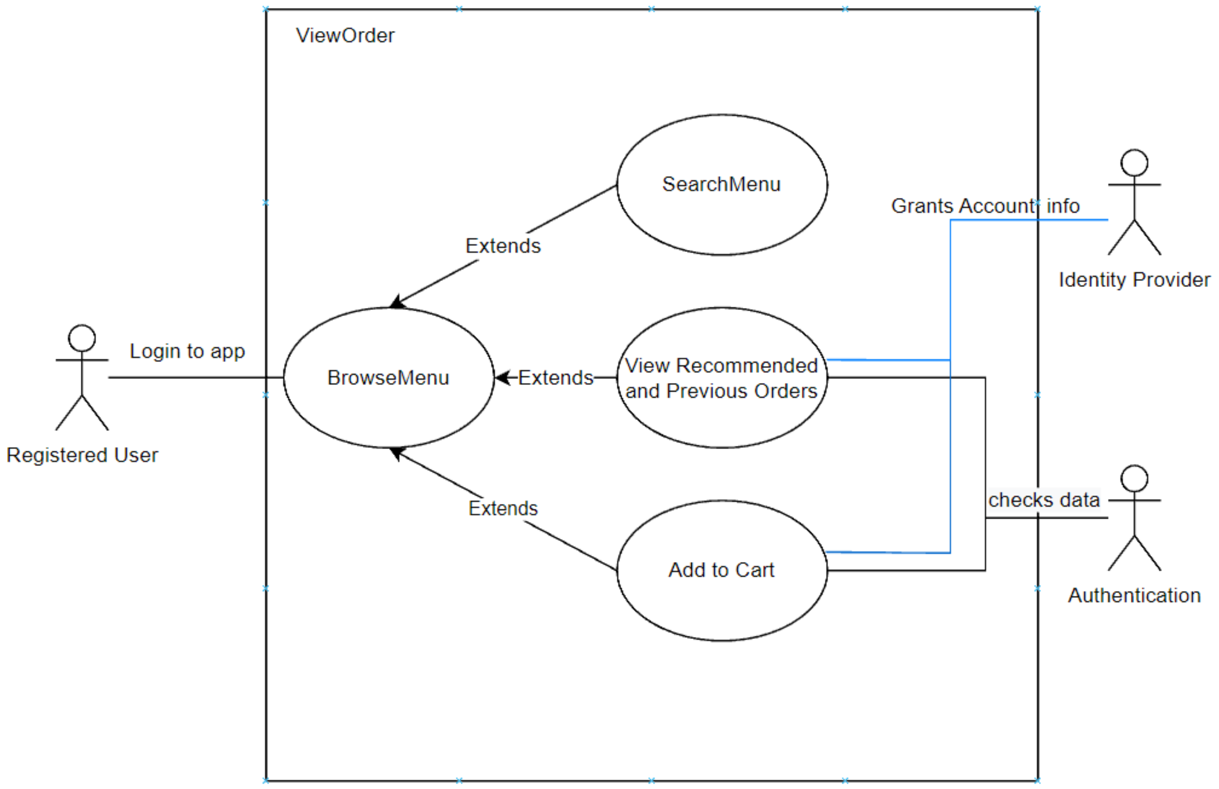
UC-0 Specifications

Name of Use Case:	ClassDash food ordering system	**Use Case ID:**	UC-0
Created by:	DoubleFast Inc.	**Last Updated By:**	Zikai Hao, Coby Lam
Date Created:	19/10/2022	**Last Revision Date:**	28/10/2022
Description:	The use case diagram represents the methodology used in system analysis to identify, clarify, and organize system requirements of the ClassDash.		
Actors:	Registered User, New User, Service Worker, Authentication,Identity Provider, Credit/Debit Payment service, Uvic OneCard payment service		
Preconditions:	-Users (Students) must have a validate account with the system -Users can only place orders when food locations are available or open		
Postconditions:	-The food providers will not be able to check users' information after the orders are completed -Each order will save to up 3 months for later customer services		
Main Flow:	-When users login to the system they can view the information of food providers -If the users want to place order, they need to choose payment type -For both payment type (UVic OneCare or credit card), the payment services will receive the information and let the food providers collect payment -After the food providers collect the payment, they can start preparing food and create receipt -While food providers start preparing the foods, users(students) could track the process of the order		
Alternate Flow:	N/A		

Users, after registering an account, will look to use the application to browse a vendor's menu before creating their order and completing their purchase, allowing them to pick up their order when ready. Workers receive orders as they are submitted and tracked by the system and assemble them for User pickup. In addition, Workers navigate their vendor menu and update it as needed.

Next, each use case outlined in the top-level diagram is exploded, revealing the flows and relationships within.

UC-1: ViewOrder

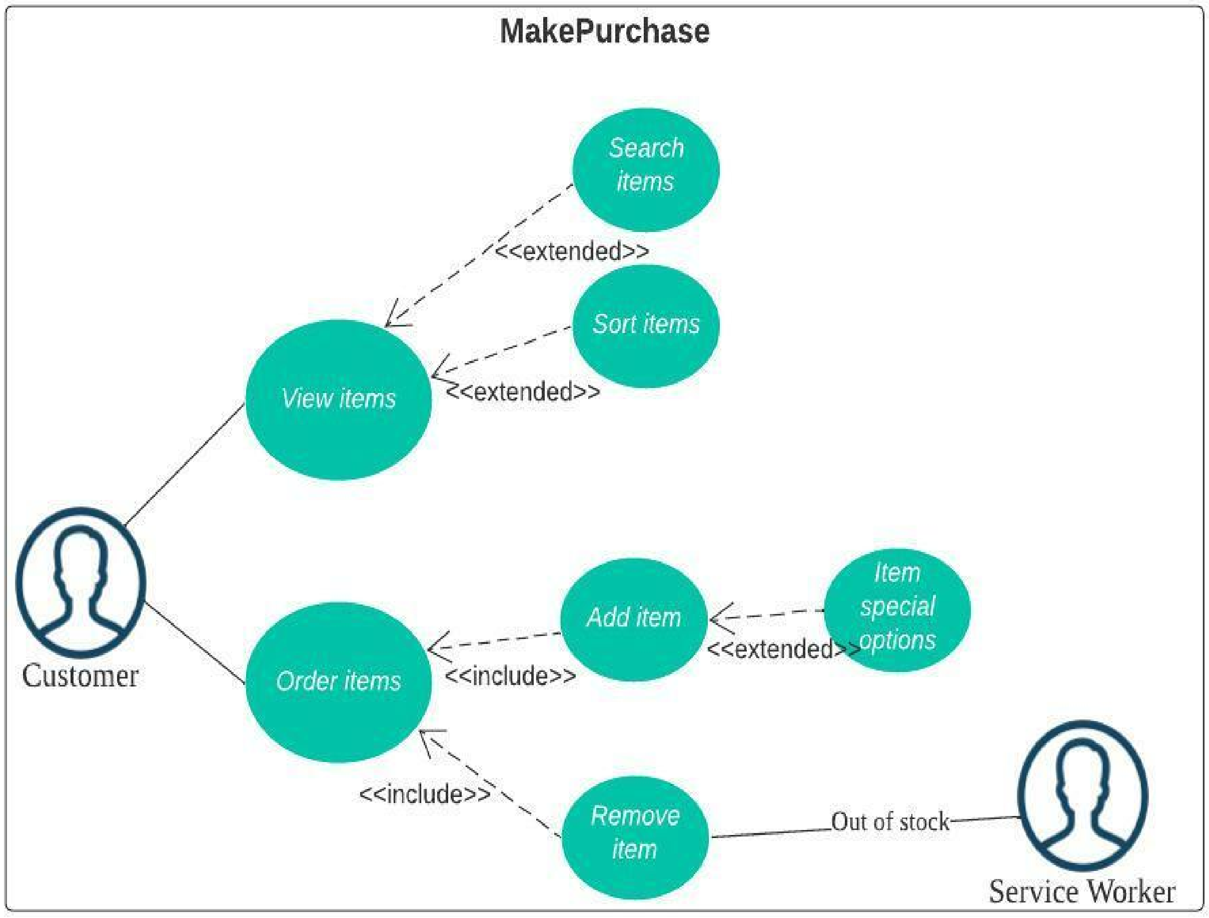


UC-1 Specification

Name of Use Case:	View Order	**Use Case ID:**	UC - 1
Created by:	DoubleFast Inc.	**Last Updated By:**	Coby Lam
Date Created:	21/10/2022	**Last Revision Date:**	28/10/2022
Description:	The User can view and interact with the menu		
Actors:	Registered User		
Preconditions:	The registered user has to be logged in.		
Postconditions:	The User is finished with Viewing potential orders and proceeds to checkout options		

Main Flow:	1. The User opens the application and goes to the main page. 2. User finds an item they like. extension point: AddToCart 3. User adds the item to the cart 4. The User proceeds to checkout the application
Alternate Flow:	Alternative flows are replacements for step 2. extension point: SearchMenu 2. User specifically searches for the item they want or extension point: View Recommended and Previous Orders 2. User views Recommended options and Previous Orders. 3. User selects a recommended item or previous item that they ordered. 1.Identity Provider grants Account information to ‘View Recommended Orders and/or Previous Orders’ extension and ‘Add to Cart’ extension. 2. Authentication checks data provided from user’s account when ‘View Recommended Orders and/or Previous Orders’ extension and ‘Add to Cart’ extension are called.

UC-2: MakePurchase



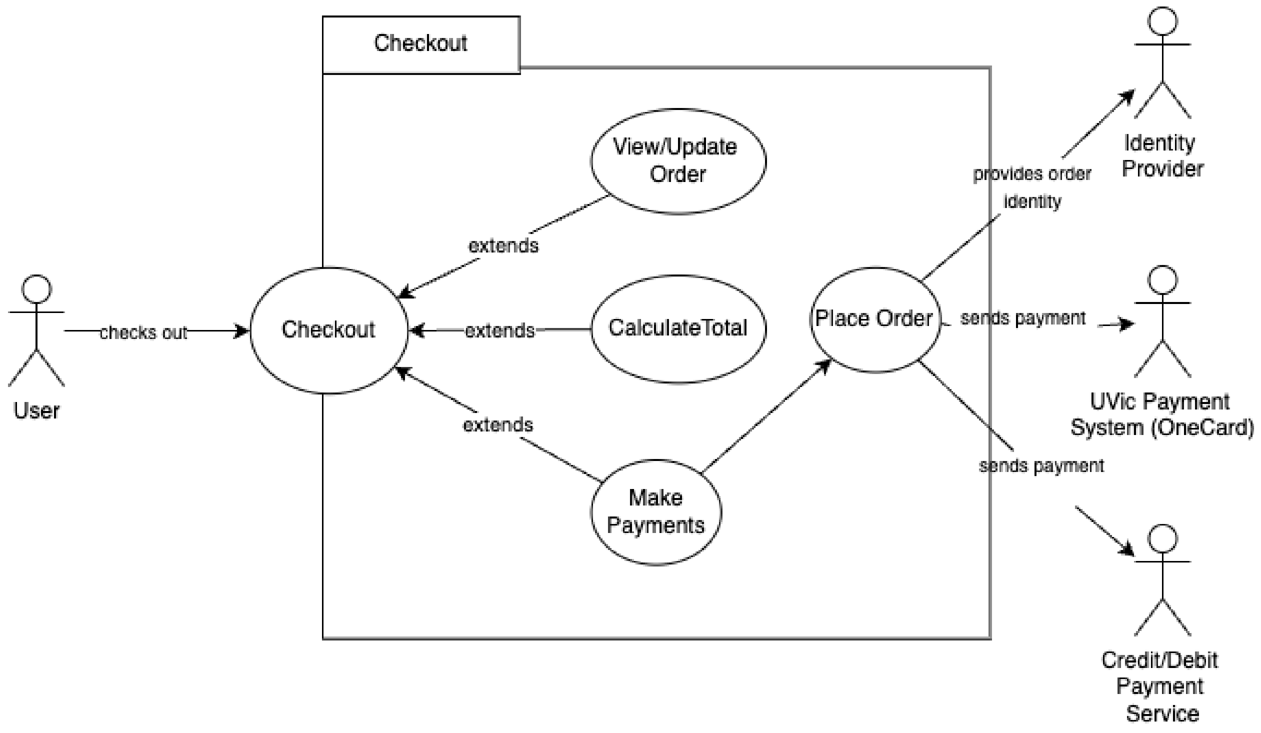
UC-2 Specifications

Name of Use Case:	ViewItems		**Use Case ID:**	UC – 2.1
Created by:	DoubleFast Inc.	**Last Updated By:**	Pengfei Li	
Date Created:	<21/10/2022>	**Last Revision Date:**	<28/10/2022>	
Description:	This use case scenario illustrates how users view items in the system.			
Actors:	Customers(Registered User)			

Preconditions:	All customers are registered and successfully logged in to the ClassDash system.
Postconditions:	Customers are ready to order what they want.
Main Flow:	1. The registered customer wants to order food from the system. 2. While the customer cannot find the preferred food: 2.1 The customer sorts the menu for a better view. 2.2 The customer searches the item name. 3. The customer found the needed food.
Alternate Flow:	N/A

Name of Use Case:	OrderItems		**Use Case ID:**	UC – 2.2
Created by:	DoubleFast Inc.	**Last Updated By:**	Pengfei Li	
Date Created:	<21/10/2022>	**Last Revision Date:**	<28/10/2022>	
Description:	This use case scenario illustrates how users order food in the system.			
Actors:	Customers(Registered User) / Service Workers			
Preconditions:	Customers found what they wanted to eat successfully in the ClassDash system.			
Postconditions:	The customers will be ready to pay for their food.			
Main Flow:	1. The customer adds all needed food to the cart. 2. If the customer no longer needs an item: 2.1 The customer removes the item from the cart. 3. Else, the customer is ready to pay for the food.			
Alternate Flow:	1. The service worker checks all the resource inventory. 2. If service worker finds a shortage of one or many kinds of materials: 2.1 All related dishes will be removed from the system			

UC-3: Checkout

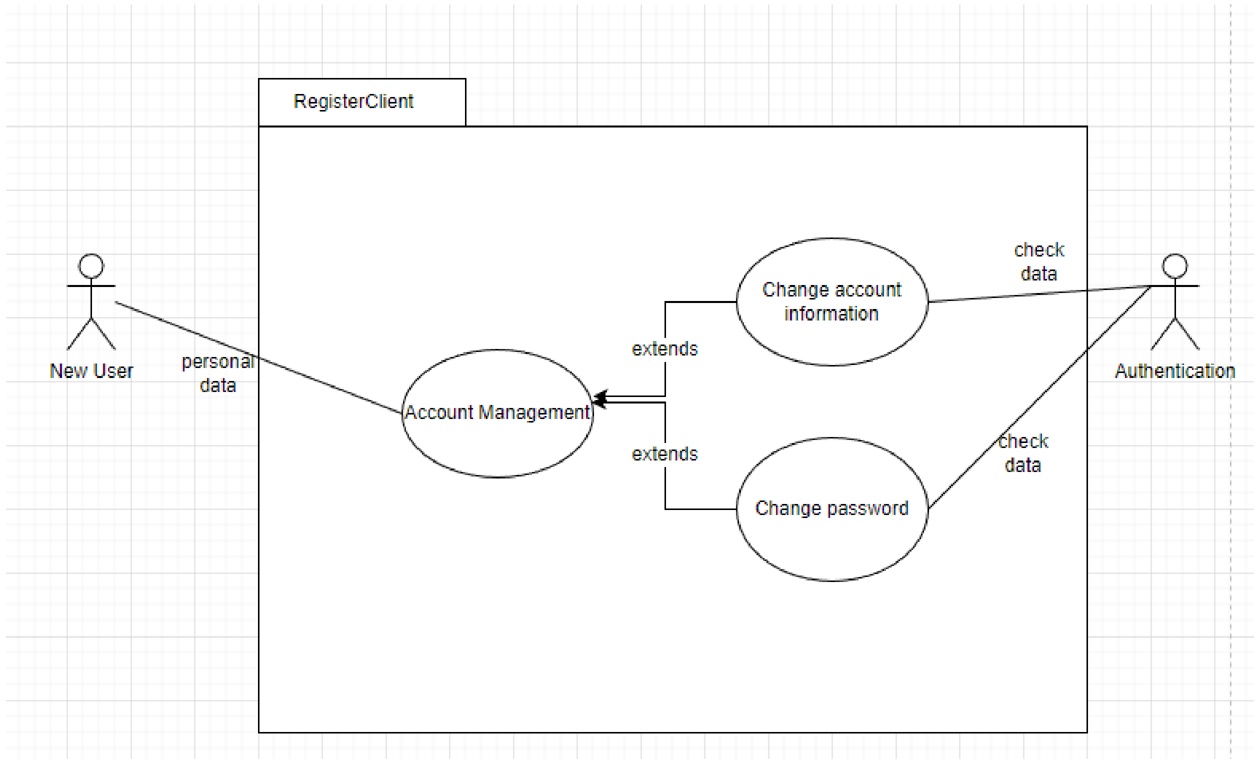


UC-3 Specifications

Name of Use Case:	Checkout	**Use Case ID:**	UC – 3
Created by:	DoubleFast Inc.	**Last Updated By:**	Shivani Ram
Date Created:	21/10/2022	**Last Revision Date:**	28/10/2022
Description:	The goal of this use case is to walk through the steps taken when a user wants to check out after placing an order at UVic food services.		
Actors:	User, Identity Provider, OneCard, Credit/Debit Payment Services		
Preconditions:	Payment by OneCard or Debit/Credit cannot be done until the user proceeds to the Payment page		
Postconditions:	Once the Payment is completed, the Payment is sent to OneCard or Credit//Debit services and the order is placed. When the order is placed, an identity is assigned to the order that is linked to the customer's authentication		

Main Flow:	1. If the customer wants to checkout 1.1 The customer can see a final confirmation of their order 1.2 The customer can see their grand total 1.3 The customer proceeds to the payment screen 2. While accessing the payment screen 2.1 The customer can pay using OneCard 2.2 The customer can pay using Credit/Debit Services 3. While the order is placed 3.1 Payment is sent to OneCard 3.2 Payment is sent to Credit/Debit Payment Services 3.3 Identity is assigned from Authentication
Alternate Flow:	N/A

UC-4: RegisterClient

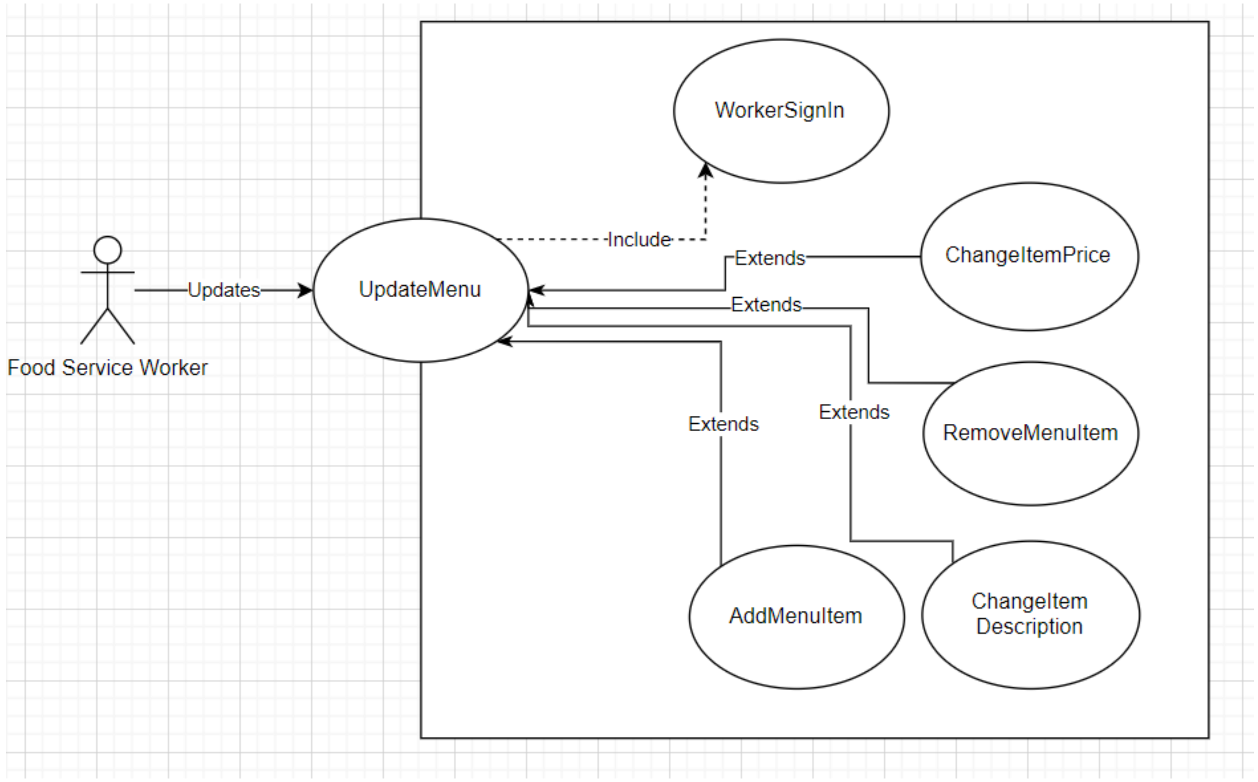


UC-4 Specifications

Name of Use Case:	RegisterClient	**Use Case ID:**	UC - 4
Created by:	DoubleFast Inc.	**Last Updated By:**	Zikai Hao, Pengfei Li

Date Created:	<22/10/2022>	**Last Revision Date:**	<28/10/2022>
Description:	The customer is about to register an account in the ClassDash system.		
Actors:	Customers / Authentication		
Preconditions:	The customer wants to order food from the ClassDash system and he or she must be a new user.		
Postconditions:	The customer registered an account in the system and is about to view menus later.		
Main Flow:	1. If the users create an account and while they access into their account management, they can input their personal information 2. In the account management, users can always change their original password later		
Alternate Flow:	N/A		

UC-5: UpdateMenu

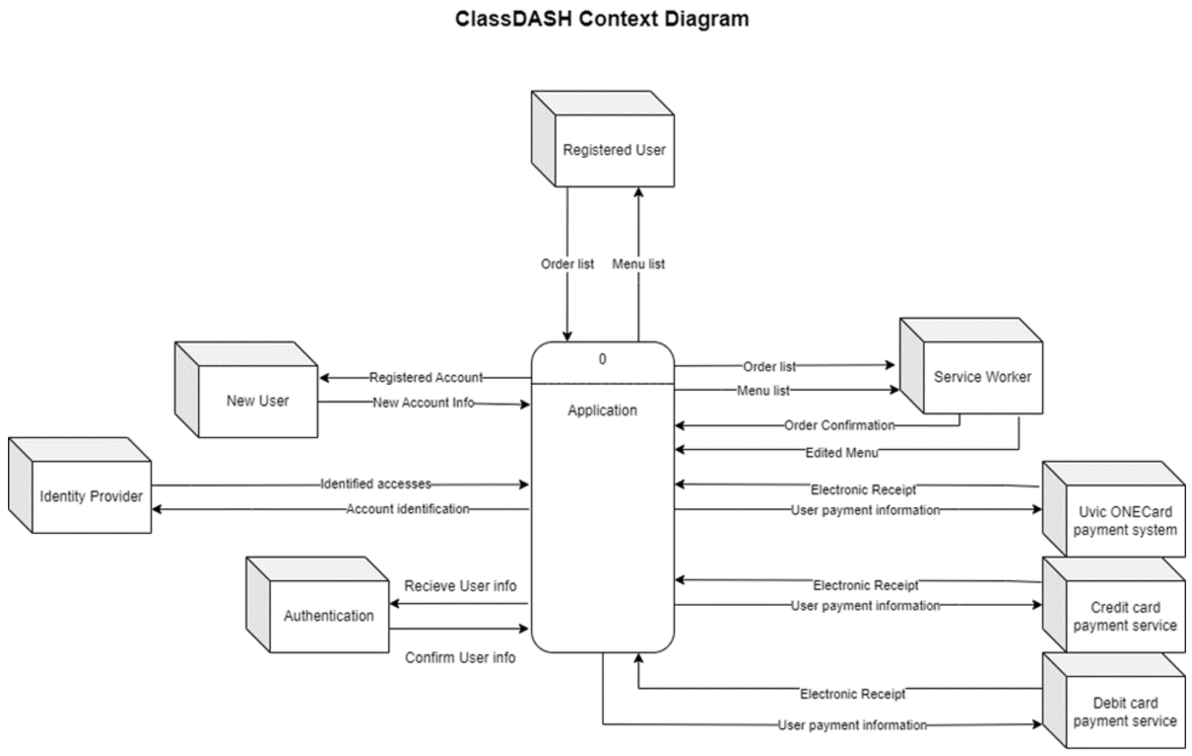


UC-5 Specifications

Name of Use Case:	UpdateMenu		**Use Case ID:**	UC - 5
Created by:	DoubleFast Inc.	**Last Updated By:**	Emile Keruzore	
Date Created:	21/10/2022	**Last Revision Date:**	28/10/2022	
Description:	The use case outlines the process of updating a vendor's menu items: adding or removing items, changing the price, or changing the associated menu description for an item			
Actors:	Food Service Worker			
Preconditions:	The user interacting must be a registered worker successfully signed-in using authentication			
Postconditions:	The relevant vendor menu has been updated			
Main Flow:	1. IF the user is a Worker and WHILE they are successfully signed in THEN: 2. The Worker may browse their vendor menu, locate the item in question and: 2.1 Change the item's price 2.2 Remove the item from the menu (If it is no longer available or out of stock) 2.3 Change the item's description (The associated describing the item and/or the display image used) 3. The Worker may add a new item to the menu with accompanying description and image if desired			
Alternate Flow:	N/A			

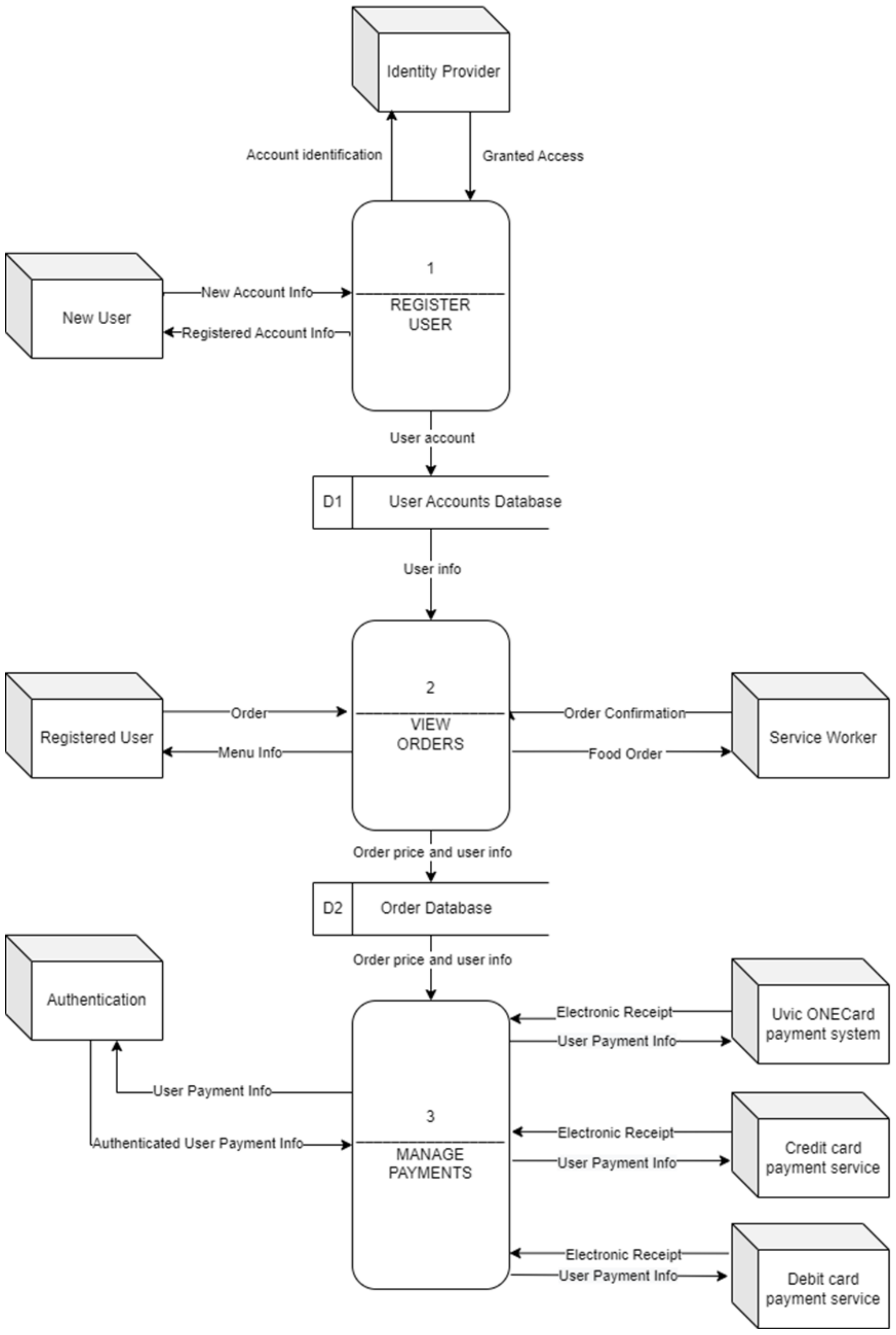
Domain Models

I. Context Diagram



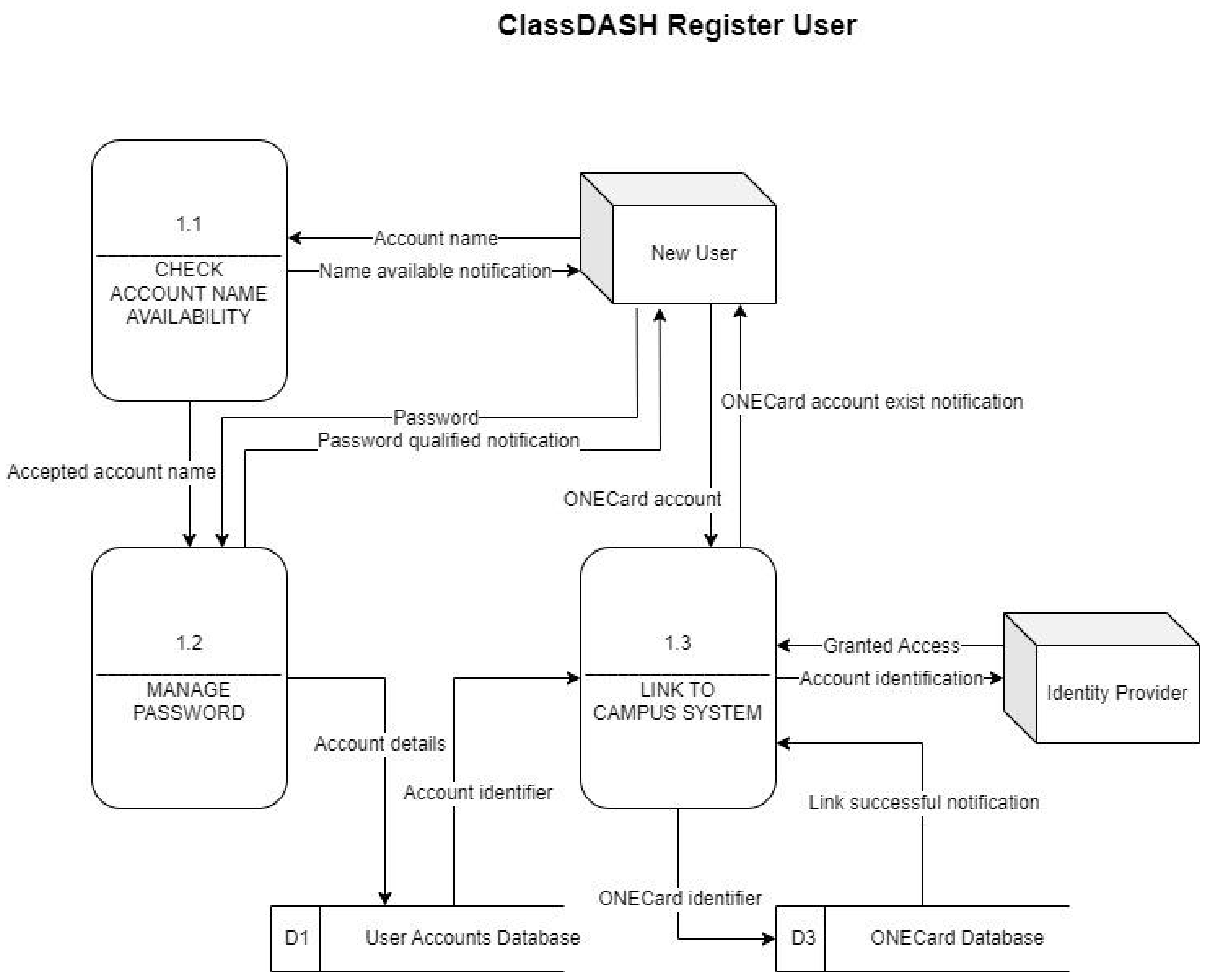
II. Diagram 0 for ClassDASH

ClassDASH Diagram 0



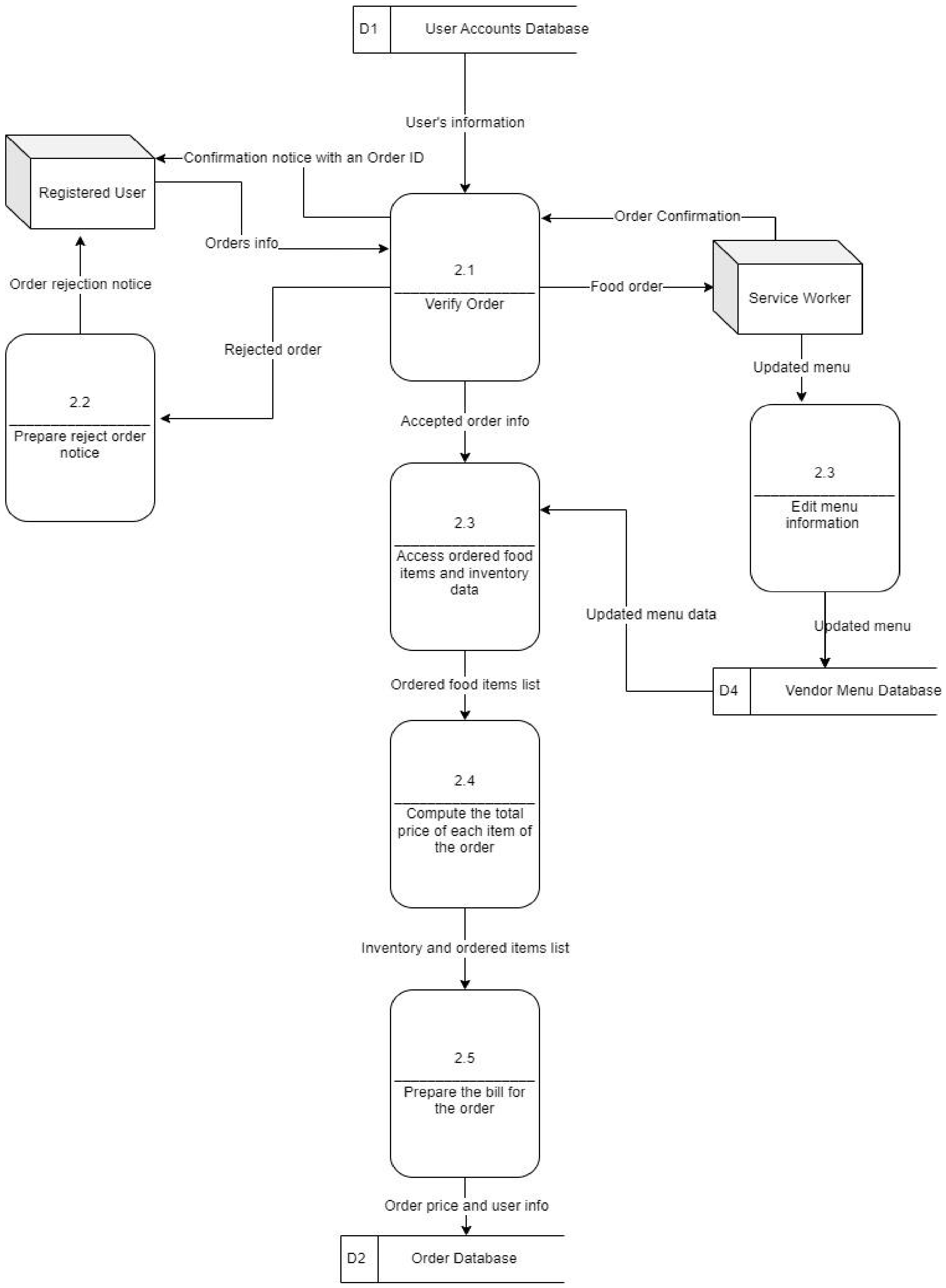
III. Level 1 DFDs:

1. Register Users

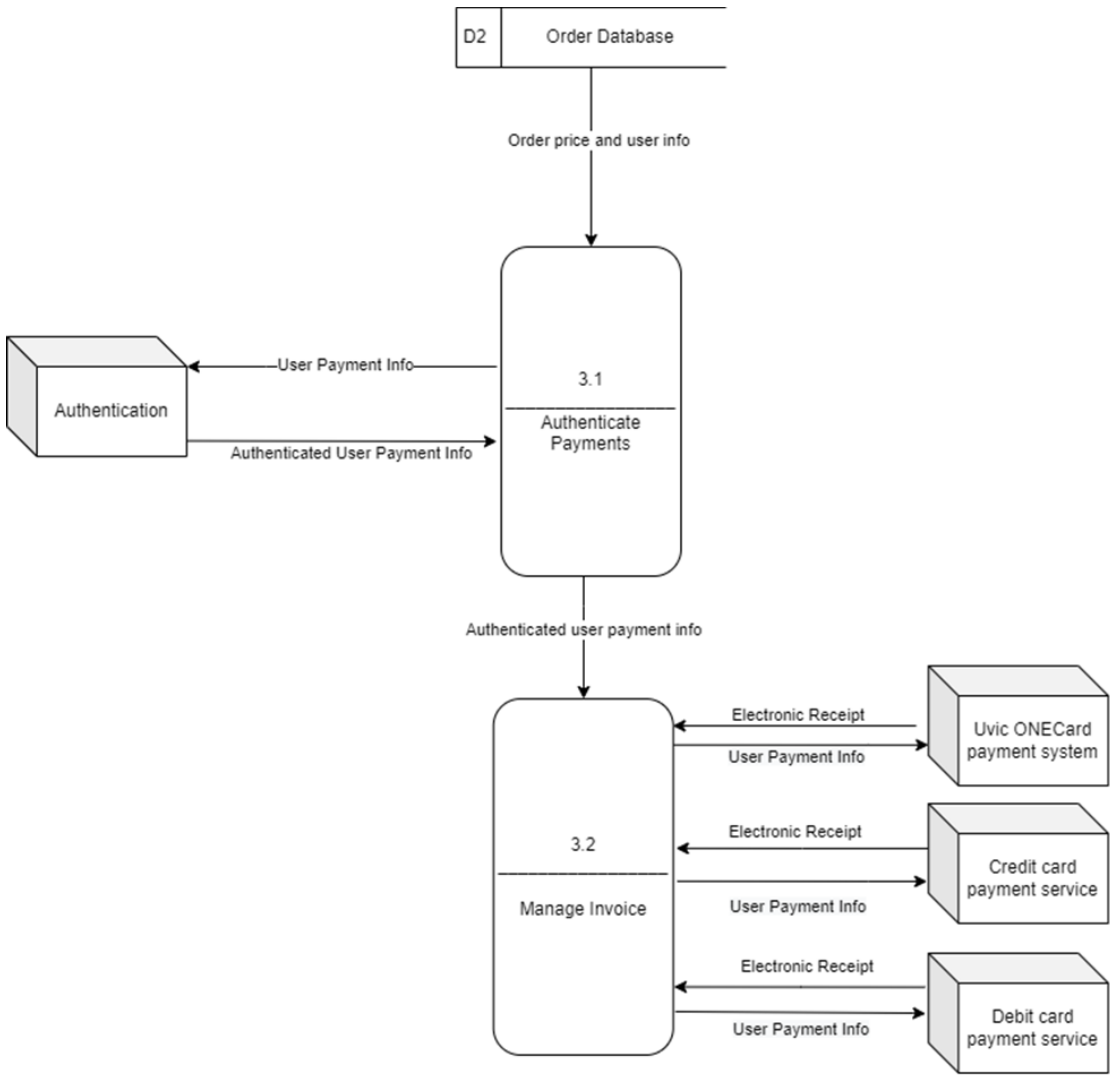


2. View Orders

ClassDASH Diagram 1 DFD for View Order



3. Manage Payments


D2
Order Database
Order price and user info
User Payment Info
Authentication
Authenticated User Payment Info
3.1
Authenticate Payments
Authenticated user payment info
3.2
Manage Invoice
Electronic Receipt
User Payment Info
Uvic ONECard payment system
Electronic Receipt
User Payment Info
Credit card payment service
Electronic Receipt
User Payment Info
Debit card payment service