



ClassDASH: Mobile Food Ordering System for the University of Victoria

DoubleFast Inc.

Coby Lam V00914020

Emile Keruzore V00877059

Pengfei Li V00831098

Shivani Ram V00972465

Zikai V00897300

November 25th, 2022

Table of Contents

1. Introduction	2
1.1 Executive Summary	2
1.2 Project Overview	6
1.3 Project Organization	12
2. Requirements	14
2.1 Functional and Non-Functional Requirements	14
2.2 Use Cases	16
UC-0: Top level	17
UC-1: ViewOrder	19
UC-2: MakePurchase	21
UC-3: Checkout	23
UC-4: RegisterClient	24
UC-5: UpdateMenu	25
2.3 Domain Models	27
I. Context Diagram	27
II. Diagram 0 for ClassDASH	28
III. Level 1 DFDs	29
3. Solution & Design	32
3.1 User Interface (UI) Models	32
4. Conclusions & Recommendations	42
5. Team Members' Contributions	42
6. Appendices	43
Glossary	43
Interview Notes	44

1. Introduction

Document Change Control

Revision Number	Date of Issue	Author(s)	Brief Description of Change
1.0	October 7th, 2022	Coby Lam, Emile Keruzore, Shivani Ram, Zikai Hao, Pengfei Li	First iteration of the UVic ClassDASH project charter submitted for review and further discussion

1.1 Executive Summary

Overview

The ClassDASH project was initiated after receipt of the Request for Proposal for a mobile food ordering solution by the University of Victoria's (UVic) Food Services. The several food outlets across campus offering a variety of foods to students, faculty, staff, and guests, were experiencing significant traffic at peak hours. This resulted in long lines and delays for hungry patrons -- many of whom could not always afford to wait in line before their next course or appointment. Therefore, a mobile ordering solution allowing patrons to submit their order ahead of time and pick up at their convenience was desired as a solution that could better serve patrons, and increase the efficiency of ordering and sales for Food Services. Consequently, upon delivery of the ClassDASH app, changes will be both experienced by the Food Service organization and their employees, as well as their hungry patrons, entailing important change management considerations. To ensure buy-in and clear lines of communication, key stakeholders were identified, including representatives from the UVic Food Services, the UVic Student Society, and the UVic Committee.

The primary goals of the project are fourfold:

- 1) To acquire at least 50% user base for students who consume one meal on campus a day (approx. 1350)
- 2) To cover at least 30% of orders for students who consume at least two meals a day through the existing food service system

- 3) To avoid food waste through the implementation of the system; minimizing the volume of abandoned orders
- 4) Clear the congestion in front of shops by diverting what would be line-up orders through the system

Milestones

The major milestones for the project include the inception, identification, and delivery of the product.

In the inception stage, completed by September 27th, 2022, the project scope is clarified, meetings with the client take place, and objectives determined and feasibility-analyses performed on potential solutions. In this phase, meeting with and receiving approval from stakeholders is a key deliverable.

Next, the project identification milestone is reached October 25th, 2022, with the completion of a series of analyses including cost-benefit, Strength-Weakness-Opportunity-Threat (SWOT), and risk assessments. A second client meeting where the project proposal is discussed takes place. Further, a key stakeholder, the App developer, is delivered the requirements and approves them.

By November 25th, 2022, the project delivery milestone will be reached. Leading up to this milestone, objectives and requirements are translated into technical criteria, allowing for design and prototyping to begin. Key deliverables in this period include describing application performance goals and the creation of a domain and UI model, along with accompanying documentation. On the business side, kickoff and quality assurance meetings are conducted with the UVic Food Services and Committee stakeholders. Soon after, performance and quality reports are generated and brought to stakeholders for approval. Iterative prototypes are then created in consultation with stakeholders, with their use case and UI needs captured. As the final product is implemented, a third client meeting will take place, wherein the final report is presented, describing the CLassDASH implementation in full detail.

Risks

Ahead of the project and as it progresses, the key risks have been identified and mitigation plans developed. These include software-integrity breaches that could result in information theft and fraud,

requiring mitigation through the use of a robust in-app security system using best practices. Further, there are important food safety considerations to ensure the risk of food borne illness is minimized, such as avoiding that food is out for significant periods of time before pick up. Some other important business risks and considerations relate to avoiding failed end user adoption through effective change management and communication processes, and forecasting the need for more food service staff and resources if the product greatly increases the volume of orders.

Authorization

This project charter formally authorizes the existence of the project, ClassDash, and provides the project manager with authority to apply organizational resources to project activities described herein. If there is a change in the project scope, the project charter will be updated and submitted for re-approval.

Pengfei Li October 7th, 2022
Executive Sponsor
Student Service Manager, University of Victoria

Leo Prosalendis October 7th, 2022
Project Sponsor
Student Representative, University of Victoria Student Society (UVSS)

Erica Christie October 7th, 2022
Project Sponsor
Food Services Manager, University of Victoria

Coby Lam October 7th, 2022
Project Lead
Lead Analyst, DoubleFast

Emile Keruzore
App Developer
Development Team, DoubleFast

October 7th, 2022

Zikai Hao
Committee Head
University of Victoria Committee

October 7th, 2022

Shivani Ram
Business Analyst
Project Team, DoubleFast

October 7th, 2022

1.2 Project Overview

Project Summary

During the first week of the fall term of 2022, the UVic campus food service department observed a massive daily population flow. The department received many complaints about the delay of campus food services facilities, including the University Center (Mystic Market) and The Student Union Building.

To optimize UVic food services and save students and staff time, we launched the project, ClassDASH, which requires the project team to provide a mobile application for our customers. Once this project is finished, students and staff on campus should order food easily from our online services.

Project Goals, Business Outcomes and Objectives

The UVic student service department performed an investigation on the current daily student flow on campus. Based on the results from the final report, roughly 2000 students rely on on-campus food services every day. About 15.7% of them consume at least two meals on campus, and 67.2% of them consume only one meal on campus. In the end, the project application will cover the majority of those people.

No.	Goals	Objectives	Business Outcomes
1	Students who consume one meal	Population of them	The system covers at least 50% of them using our application
2	Student who consumes at least two meals	Population of them	The system covers at least 30% of their meal orders
3	The system should not cause waste of food	Reducing abandoned order rate	The order abandon rate should not exceed 5% every day
4	The food provider will not get congested all day	Population of the waiting line in front of food provider shops	No more than <ul style="list-style-type: none">● 8 for the Grills● 6 for all shops in UC

Project Scope

The project ClassDASH consists of a mobile application with the functionality of ordering food on campus.

The application will provide the features below:

- Order food on campus through this application.
- Pay for their food.
- Check their food preparation status.
- Cancel orders.
- Fully integrated with the ONEcard system.

This project also includes pilot testing while the design team works on performance. While the project team tests this system, the normal UVic online system (including the ONEcard system) will not be interfered with by this project.

Scope Definition

ClassDASH is a mobile application design project which will provide UVic students, staff and guests with a convenient way to order on-campus food online. This project contains one main software design project and two environment-simulating tests. Once the project is finished, the degree of crowdedness will be half of the current situation as estimated.

Boundaries

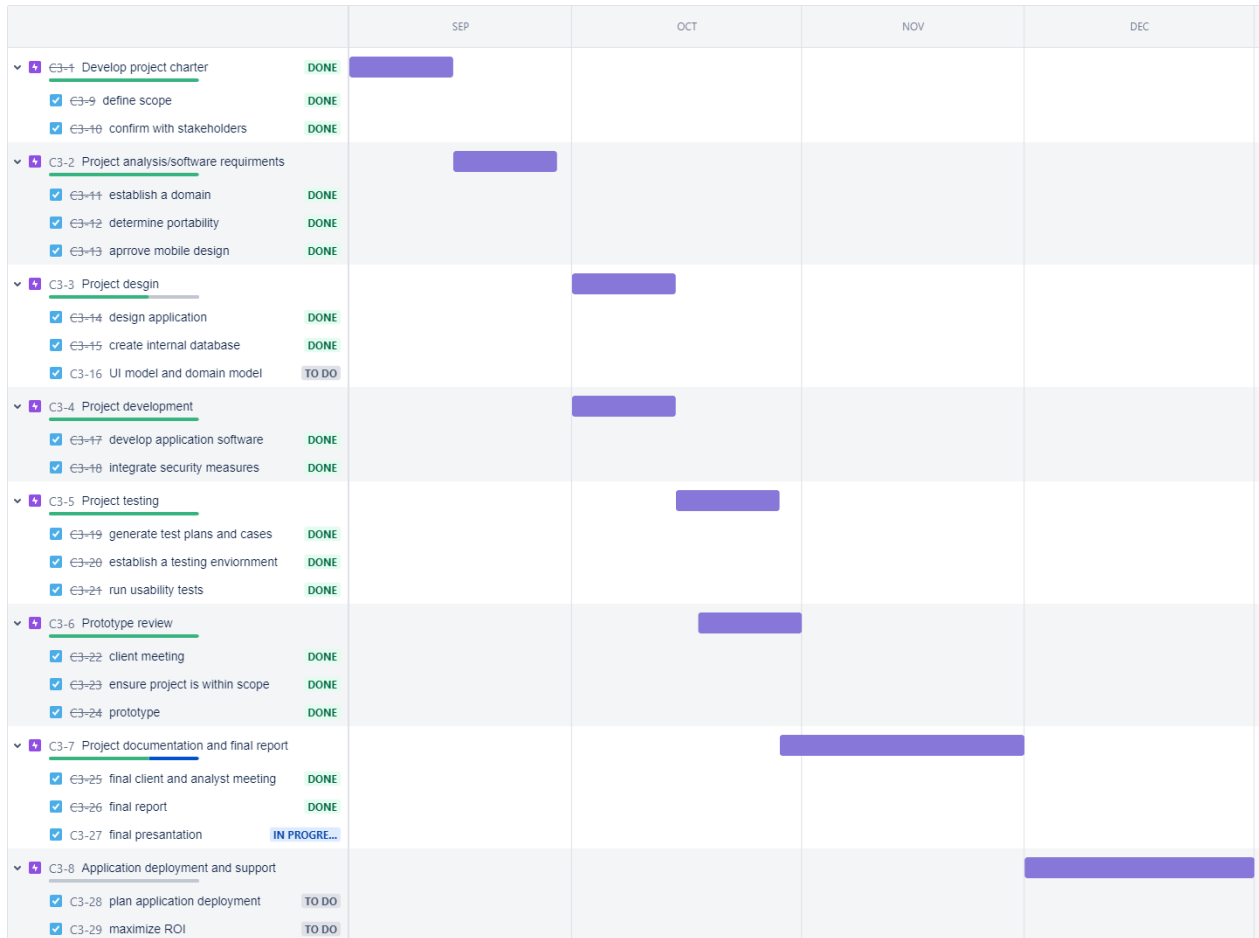
Activities In Scope	Activities Out of Scope
1. Software frame design and functionality check	1. Specific frame requirements for maintenance
2. Software interface design	2. Artworks from specific designers
3. 1 st simulation test	3. Real situation test
4. Embed system to UVic online system	4. Credit card system check
5. 2 nd simulation test	5. Organize extra volunteers
6. Application release	

There are six activities in scope for the ClassDASH project. They are listed in chronological order. The out-of-scope activities are listed, but not part of the project.

Milestones

Project Milestone	Description	Expected Date
1. Project inception stage	Clarify the scope, project objectives, solution feasibility, and first client meeting	Sep 27th, 2022
2. Project identification stage	SWOT (Strengths, weaknesses, opportunities, threats) analysis, cost-benefit analysis, risk assessment, project proposal, and second client meeting	Oct 25th, 2022
3. Project delivery stage	Translate the project objectives and requirements into technical criteria to allow detailed design, prototyping, then full implementation of the product (the ClassDash application), and third client meeting, the final presentation	Nov 25th, 2022

Gantt Chart



Deliverables

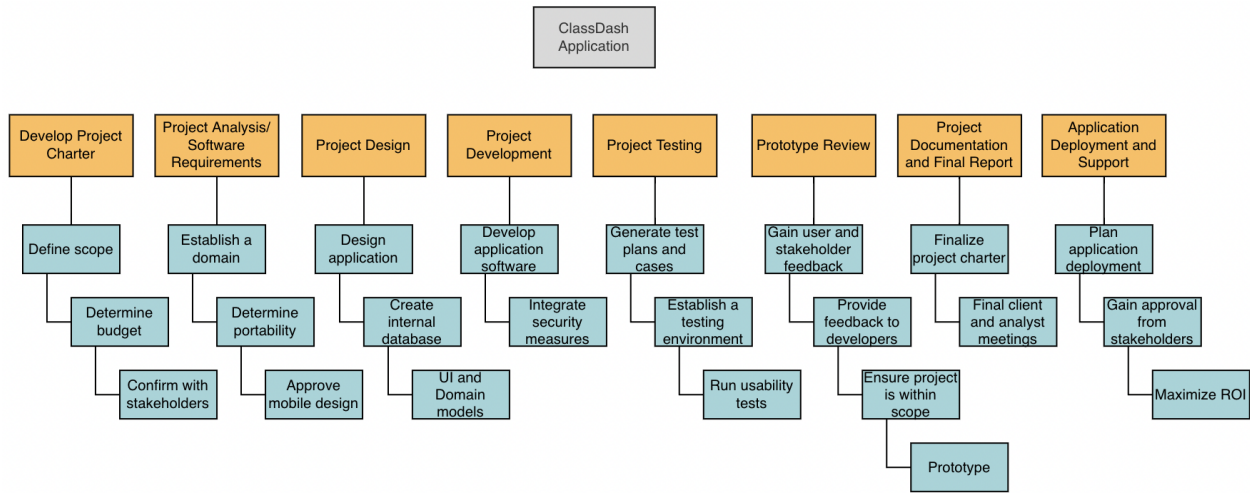
Project Deliverable 1: Project scope	
Stakeholder:	UVSS student representative, app developer, the University of Victoria Committee, the Food Service manager
Description:	The project scope includes the boundaries of the project, each people’s responsibilities, and it sets up procedures for how completed work will be approved.
Acceptance Criteria:	A project scope statement should be approved by each stakeholder.
Due Date:	Sep 27 th , 2022

Project Deliverable 2: Project analysis/software requirements and use cases section	
Stakeholder:	App developer
Description:	Project analysis/software requirements include both functional and non-functional requirements for implementing the project. Use cases section is the written description of how users will perform tasks on ClassDash
Acceptance Criteria:	List and define all the functional and non-functional requirements for the app developer, and all the requirements must be written to be objectively verified. A use cases documentation should be done
Due Date:	Oct 19 th , 2022
Project Deliverable 3: Project design: domain model and UI model	
Stakeholder:	UVSS student representative, App developer, the University of Victoria Committee, the Food Service manager
Description:	Project design includes the project lifecycle of ideas, conceptual model of the domain that incorporates both behavior and data, and UI model includes a representation of how the end user interact with ClassDash and also how the system responds
Acceptance Criteria:	A description of the ClassDash application and its performance goals should be done. A domal model and a UI model documentation should be done
Due Date:	Nov 1 st , 2022
Project Deliverable 4: Project development	
Stakeholder:	App developer, the University of Victoria Committee, the Food Service manager
Description:	Plan and allocate resources to develop the project
Acceptance Criteria:	Kickoff meetings should be done, quality assurance and allocation of resources should be done.
Due Date:	Nov 15 th , 2022
Project Deliverable 5: Project testing	

Stakeholder:	UVSS student representative, App developer, the University of Victoria Committee, the Food Service manager
Description:	Project testing includes information about actual levels of performance and quality of the project.
Acceptance Criteria:	An overall project performance report and quality report of the project should be done and approved by stakeholders.
Due Date:	Nov 18 th , 2022
Project Deliverable 6: Project UI Prototypes	
Stakeholder:	UVSS student representative, the University of Victoria Committee, the Food Service manager, App developer
Description:	Iterative analysis technique in which users are actively involved in the mocking-up of the UI for a system
Acceptance Criteria:	The textual description of requirements, use cases, domain models, and UI models that describe the client's needs should be all done
Due Date:	Nov 23 th , 2022
Project Deliverable 7: Project documentation and final report	
Stakeholder:	UVSS student representative, App developer, the University of Victoria Committee, the Food Service manager
Description:	Project documentation including the key project details and documents that are required to implement it successfully, the final report describing the outcomes of your requirements gathering, analysis, and meetings with your client team
Acceptance Criteria:	Project status report should be done, and all use cases, domain models, and UI models should be include and done in the final report
Due Date:	Nov 25 th , 2022
Project Deliverable 8: Project deployment	

Stakeholder:	UVSS student representative, App developer, the University of Victoria Committee, the Food Service manager
Description:	Project deployment is to make the application (ClassDASH) ready for delivery.
Acceptance Criteria:	A deployment plan that includes the required staff for delivering the project, a checklist of all the required tasks, and determined release strategy.
Due Date:	December 5 th ,2022

Work Breakdown Structure:



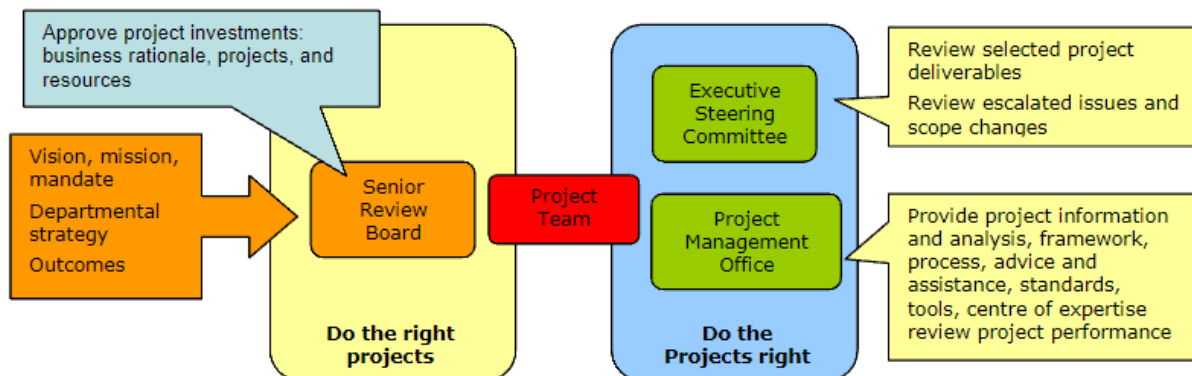
Project Risks

No.	Risk Description	Probability (H/M/L)	Impact (H/M/L)	Risk Management Plan
1	By implementing this system on a smaller scale, there is a higher chance of information theft and defrauding. There is a higher risk of security breaches if the company underestimates the need for proper security precautions within the app.	Low	High	Budget and implement a high-quality security system within the app to protect users financial information that is used to pay.

No.	Risk Description	Probability (H/M/L)	Impact (H/M/L)	Risk Management Plan
2	Food quality can be reduced significantly if left out for long periods of time. If a student orders food and picks it up an hour later, there is a higher risk of food borne illnesses.	Medium	Medium	The app can offer pick up time periods, that way the food is made shortly before the pickup period and will not be left out waiting. This also puts responsibility on the person that ordered as their food will only be served in the time slot that they chose.
3	Without adequate communication and training, the intended end users, both patrons and food service workers, may fail to adopt the app.	Low	High	Implement proper training and communication guidelines to ensure successful adoption of the system.
4	Labor costs can increase as a result of an increase in the volume of orders. By giving customers the capability of “skipping the line” they are more likely to order, resulting in the increase in need for staff and other resources.	High	Medium	This can be combated by putting a cap on mobile orders within certain time periods to limit orders to a manageable rate. Another solution would include budgeting additional staff to mitigate high influxes of orders.

1.3 Project Organization

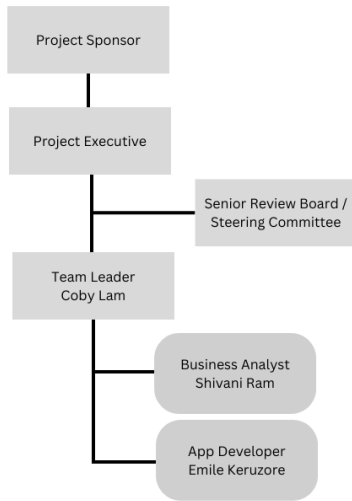
Project Governance



The project would be governed by means of UVic’s Board of Governors, Project Review Committee and our Project Team. The role of Senior Review Board will be designated to the Board of Governors, they will primarily approve of the initial project investment and provide the permission for funding. The Project

Review Committee would act as the Executive Steering Committee and the Project Management Office. They would be in charge of reviewing, escalating, maintaining the project as well as approving schedule, budget and scope changes. Lastly the Project Team is responsible for consulting clients and key stakeholders in addition to conducting internal meetings before communicating with the Project Review Committee.

3.2 Project Team Structure



3.3 Roles and Responsibilities

Project Role	Responsibilities	Assigned to
Team Leader	<ul style="list-style-type: none"> Manages functional aspects of the project Manages formal and management reviews Helps research issues and change requests Maintains the scope, estimates, and work plans Ensures the proper reporting of status for their team members 	Coby Lam
Business Analyst	<ul style="list-style-type: none"> Documents and maintains business requirements Uses analysis tools to improve project workflows Conducts research to provide systems and business support 	Shivani Ram
Project Management Office/Steering Committee	<ul style="list-style-type: none"> Ensures effective communication 	Project Review Committee

Project Role	Responsibilities	Assigned to
Senior Review Board	<ul style="list-style-type: none"> ● Approves project investments ● Reviews the business rationale, projects, and resources ● prioritizes projects based on specific criteria 	Board of Governors
App Developer	<ul style="list-style-type: none"> ● Development of software solutions to meet customer needs ● Approves system requirements delivered by project team ● Creating and implementing source code ● Testing and debugging code. 	Emile Keruzore
Project Sponsor	<ul style="list-style-type: none"> ● Has ultimate authority over and is responsible for a Program, project, or both ● Approves changes to the scope and provides whatever additional funds those changes require ● Approves budget-related deliverables ● Controls the business aspects of the project ● Assists in developing the project charter and project plans ● Executes formal reviews and management reviews ● Disposes of issues and change requests ● Makes user resources available ● Approves work products ● Assists in tracking action items and budgets ● Is responsible for the functional quality of the solution 	Project Management Office manager Scott Thompson

2. 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.*

2.1 Functional and Non-Functional Requirements

I. Functional Requirements		M	S	C	W
<i>Food Orders</i>					
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			
<i>Payment Related</i>					
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		
<i>Application Functions</i>					
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		
<i>Food Worker-side Functions</i>					
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		

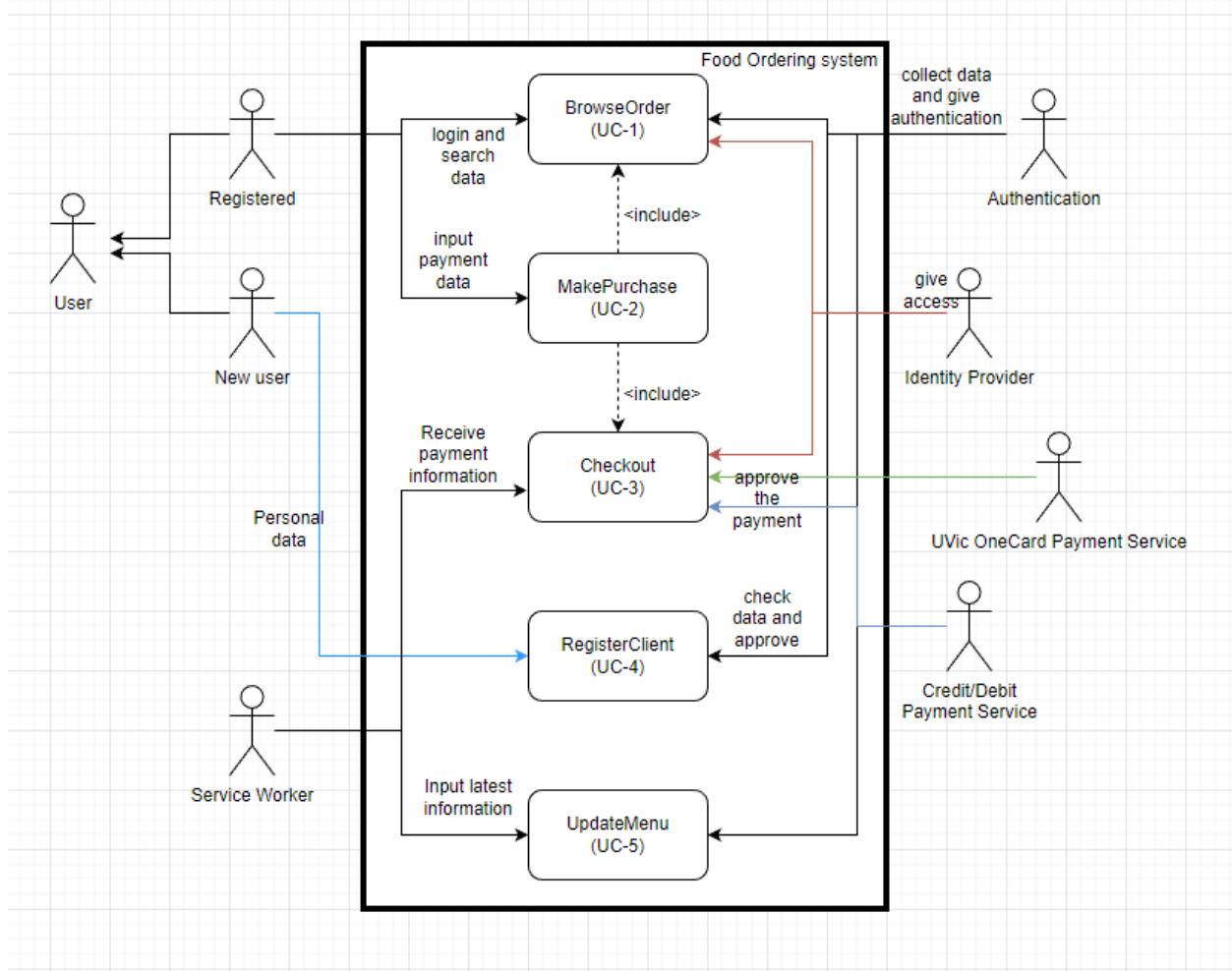
II. Non-Functional Requirements		M	S	C	W
<i>Standards</i>					

II. Non-Functional Requirements		M	S	C	W
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		
<i>Safety Related</i>					
2.1	The system should encrypted users' personal information and financial transactions in ASCII standard		X		
<i>Format Related</i>					
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			
<i>Accessibility</i>					
4.1	The system must be accessible on both Android and iOS mobile systems	X			

2.2 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

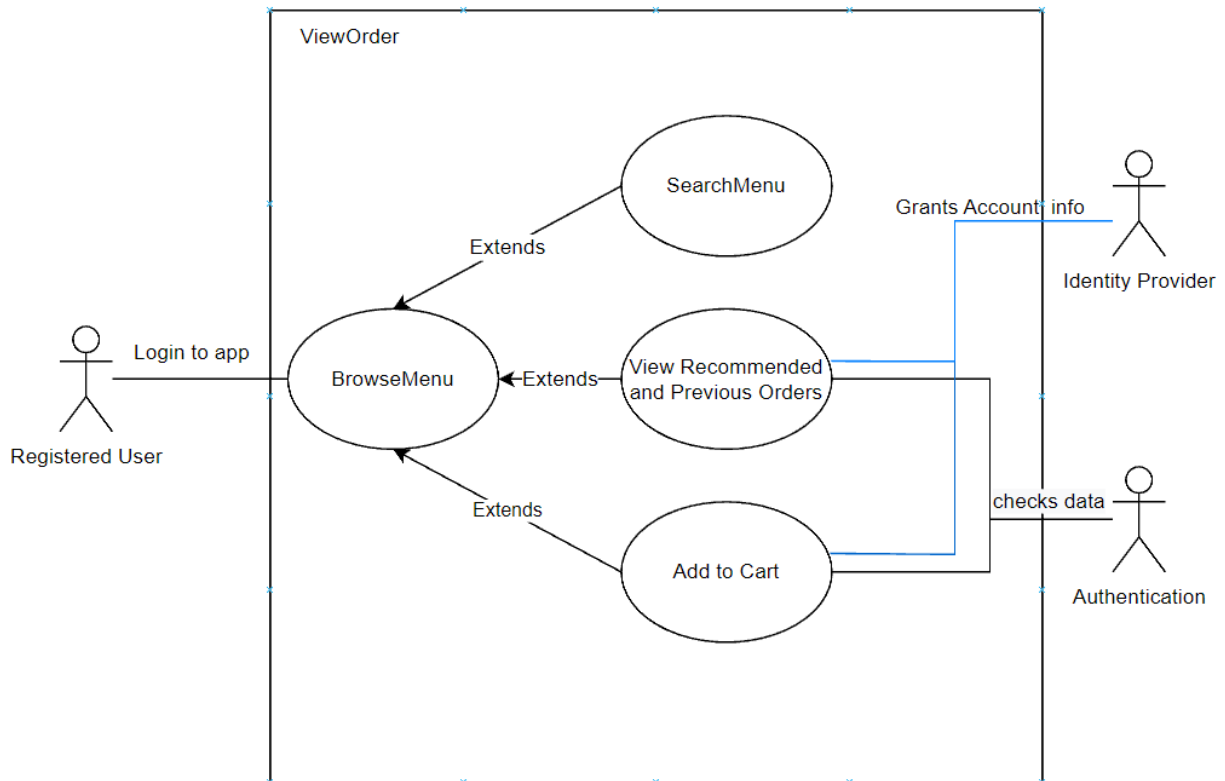
Name of Use Case:	ClassDash food ordering system		Use Case ID:	UC-0
Created by:	DoubleFast Inc.	Last Updated By:	Zikai Hao, Coby Lam	
Date Created:	19/10/2022	Last Revision Date:	28/10/2022	
Description:	The use case diagram represents the methodology used in system analysis to identify, clarify, and 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:	<ul style="list-style-type: none"> -Users (Students) must have a validate account with the system -Users can only place orders when food locations are available or open
Postconditions:	<ul style="list-style-type: none"> -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:	<ul style="list-style-type: none"> -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

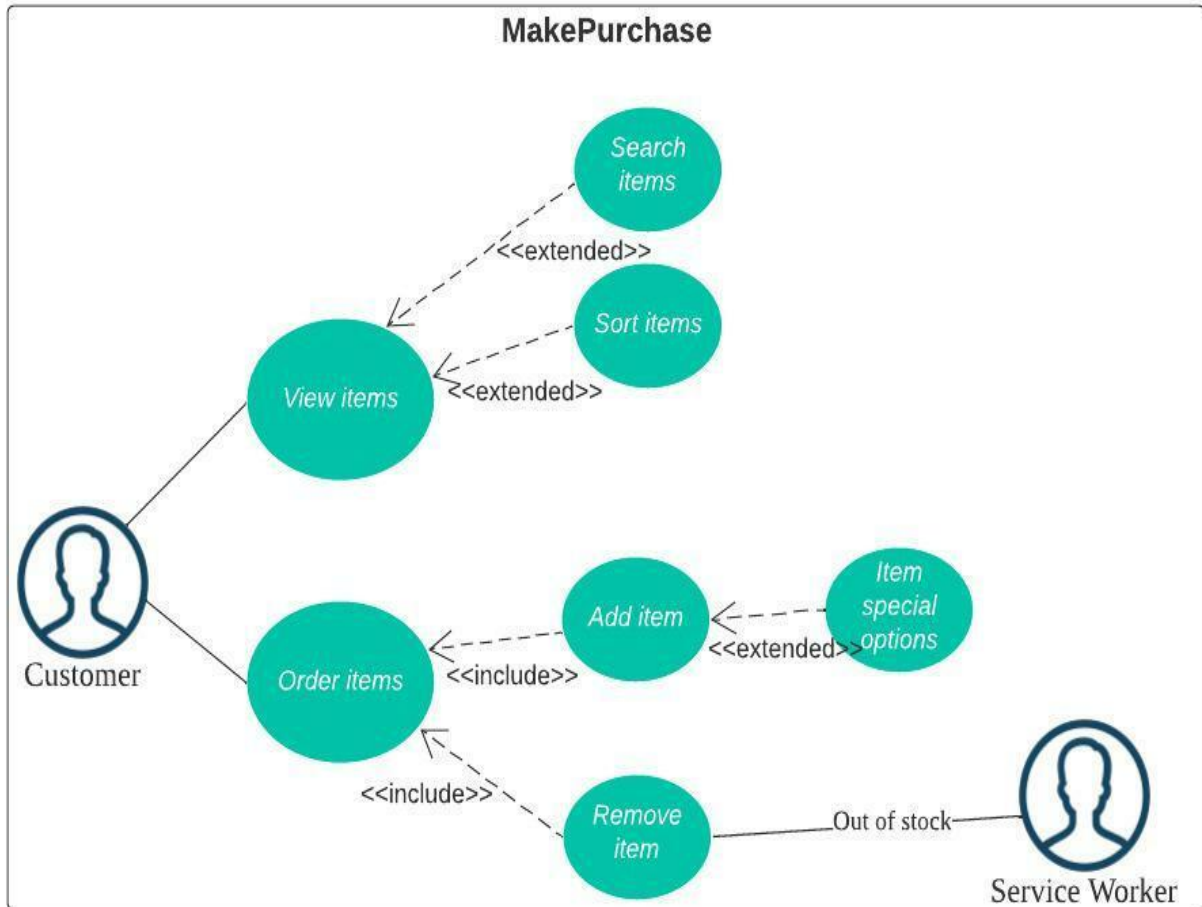


UC-1 Specification

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:	<ol style="list-style-type: none"> 1. The User opens the application and goes to the main page. 2. User finds an item they like. extension point: AddToCart <ol style="list-style-type: none"> 3. User adds the item to the cart 4. The User proceeds to checkout the application 			

Alternate Flow:	<p>Alternative flows are replacements for step 2.</p> <p>extension point: SearchMenu</p> <p>2. User specifically searches for the item they want</p> <p>or</p> <p>extension point: View Recommended and Previous Orders</p> <p>2. User views Recommended options and Previous Orders.</p> <p>3. User selects a recommended item or previous item that they ordered.</p> <p>1.Identity Provider grants Account information to 'View Recommended Orders and/or Previous Orders' extension and 'Add to Cart' extension.</p> <p>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.</p>
-----------------	--

UC-2: MakePurchase



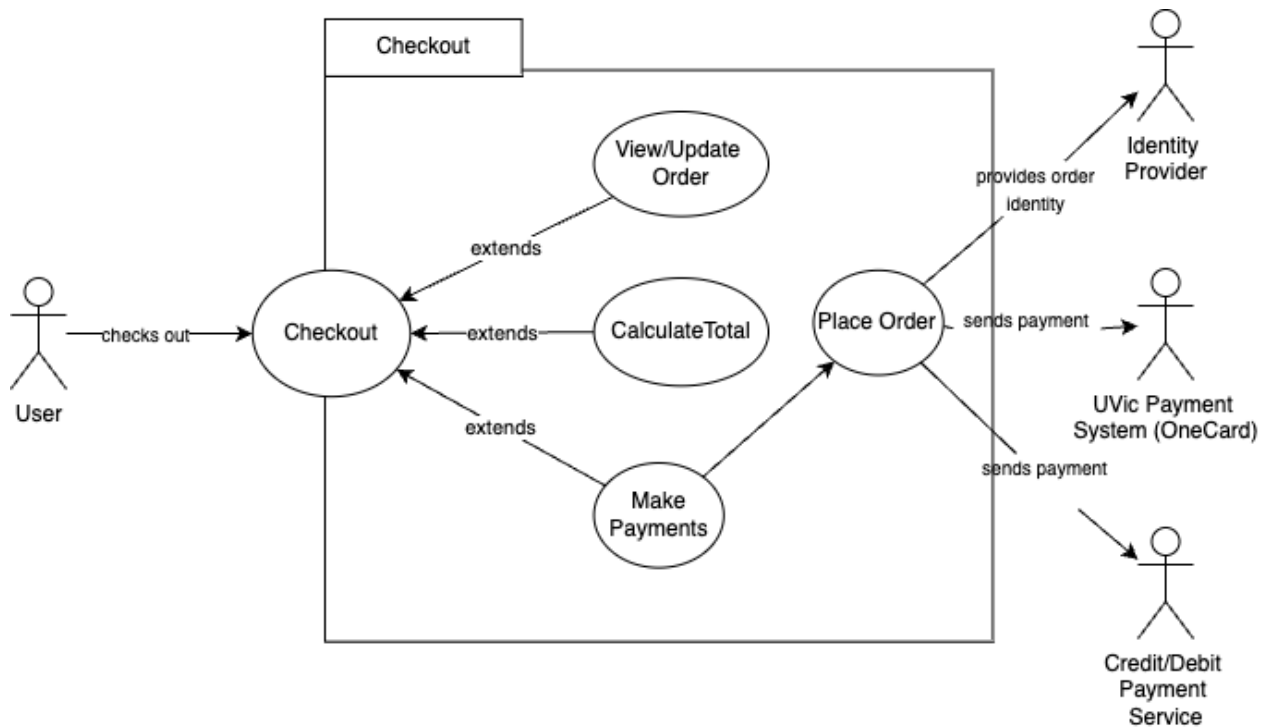
UC-2 Specifications

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:	<ol style="list-style-type: none"> 1. The registered customer wants to order food from the system. 2. While the customer cannot find the preferred food: <ol style="list-style-type: none"> 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:	<ol style="list-style-type: none"> 1. The customer adds all needed food to the cart. 2. If the customer no longer needs an item: <ol style="list-style-type: none"> 2.1 The customer removes the item from the cart. 3. Else, the customer is ready to pay for the food. 		
Alternate Flow:	<ol style="list-style-type: none"> 1. The service worker checks all the resource inventory. 2. If service worker finds a shortage of one or many kinds of materials: <ol style="list-style-type: none"> 2.1 All related dishes will be removed from the system 		

UC-3: Checkout

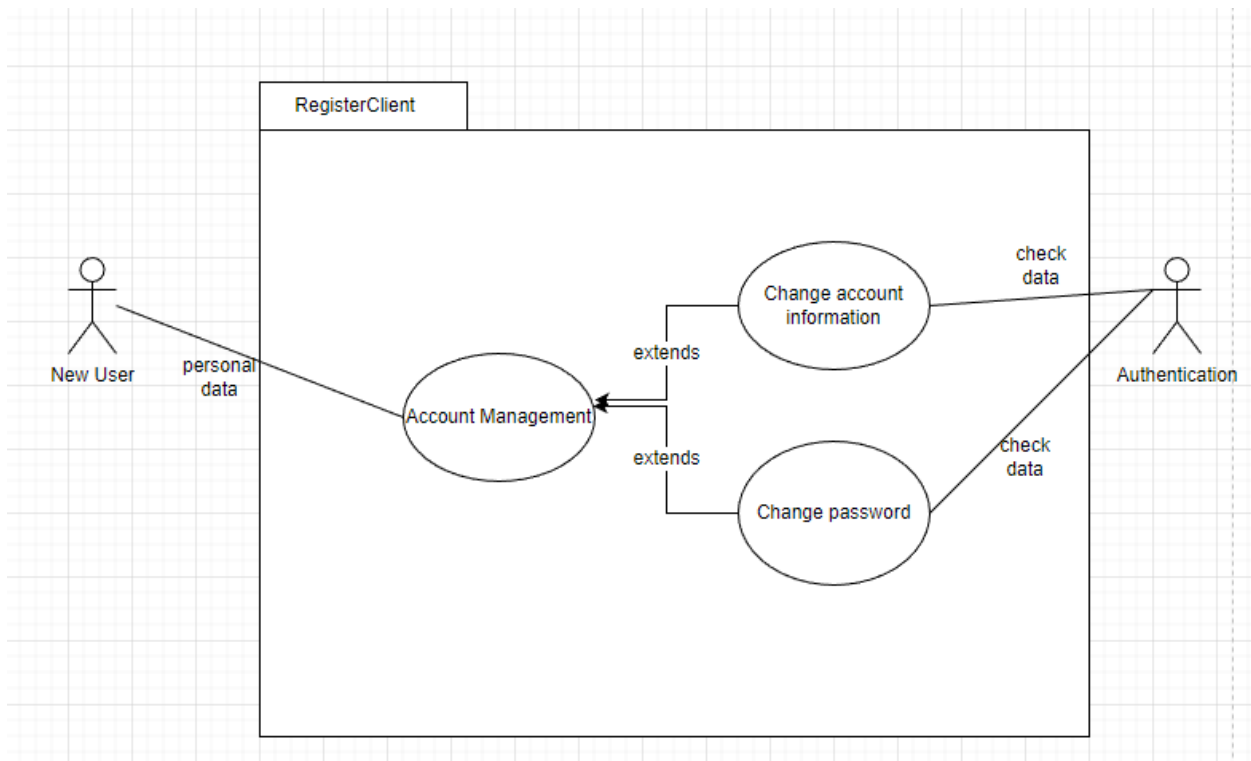


UC-3 Specifications

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:	<ol style="list-style-type: none"> 1. If the customer wants to checkout <ol style="list-style-type: none"> 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 <ol style="list-style-type: none"> 2.1 The customer can pay using OneCard 2.2 The customer can pay using Credit/Debit Services 3. While the order is placed <ol style="list-style-type: none"> 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

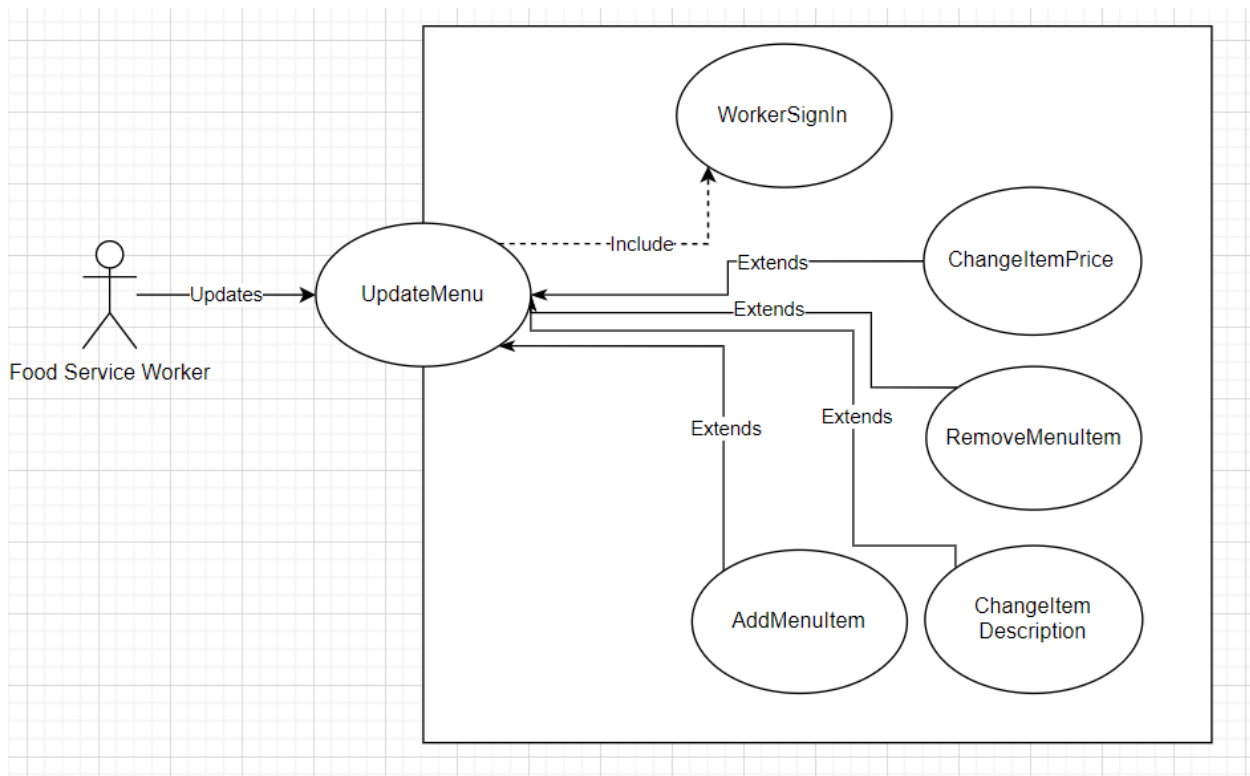


UC-4 Specifications

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:	<ol style="list-style-type: none"> 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



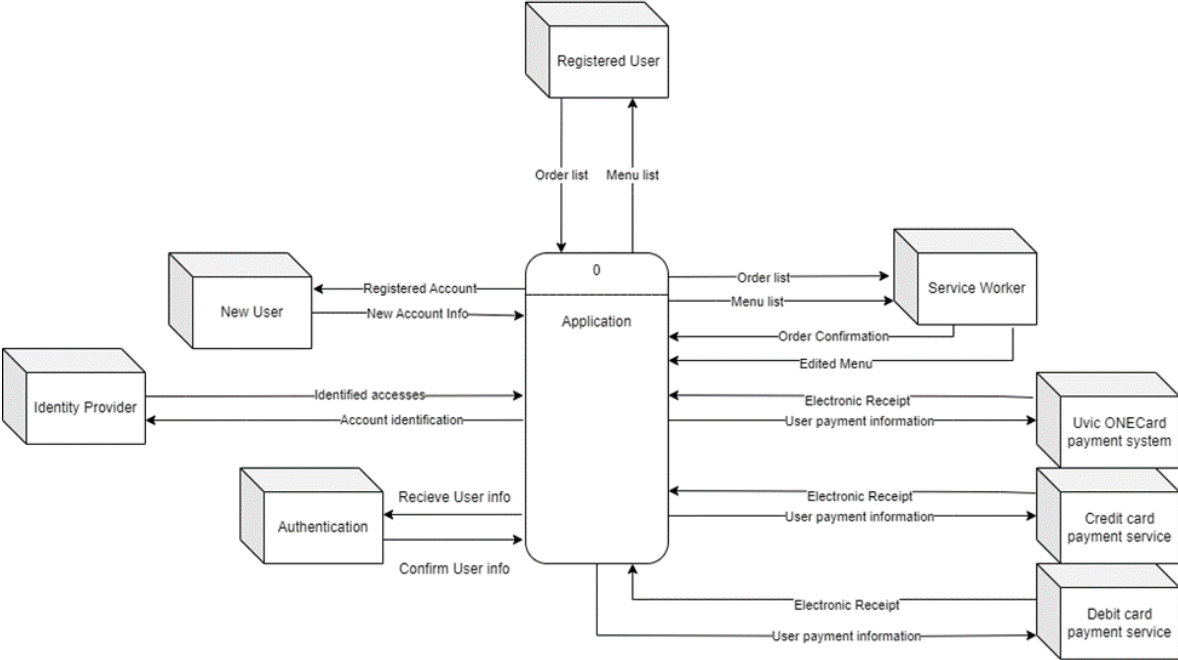
UC-5 Specifications

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:	<ol style="list-style-type: none"> 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: <ol style="list-style-type: none"> 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			

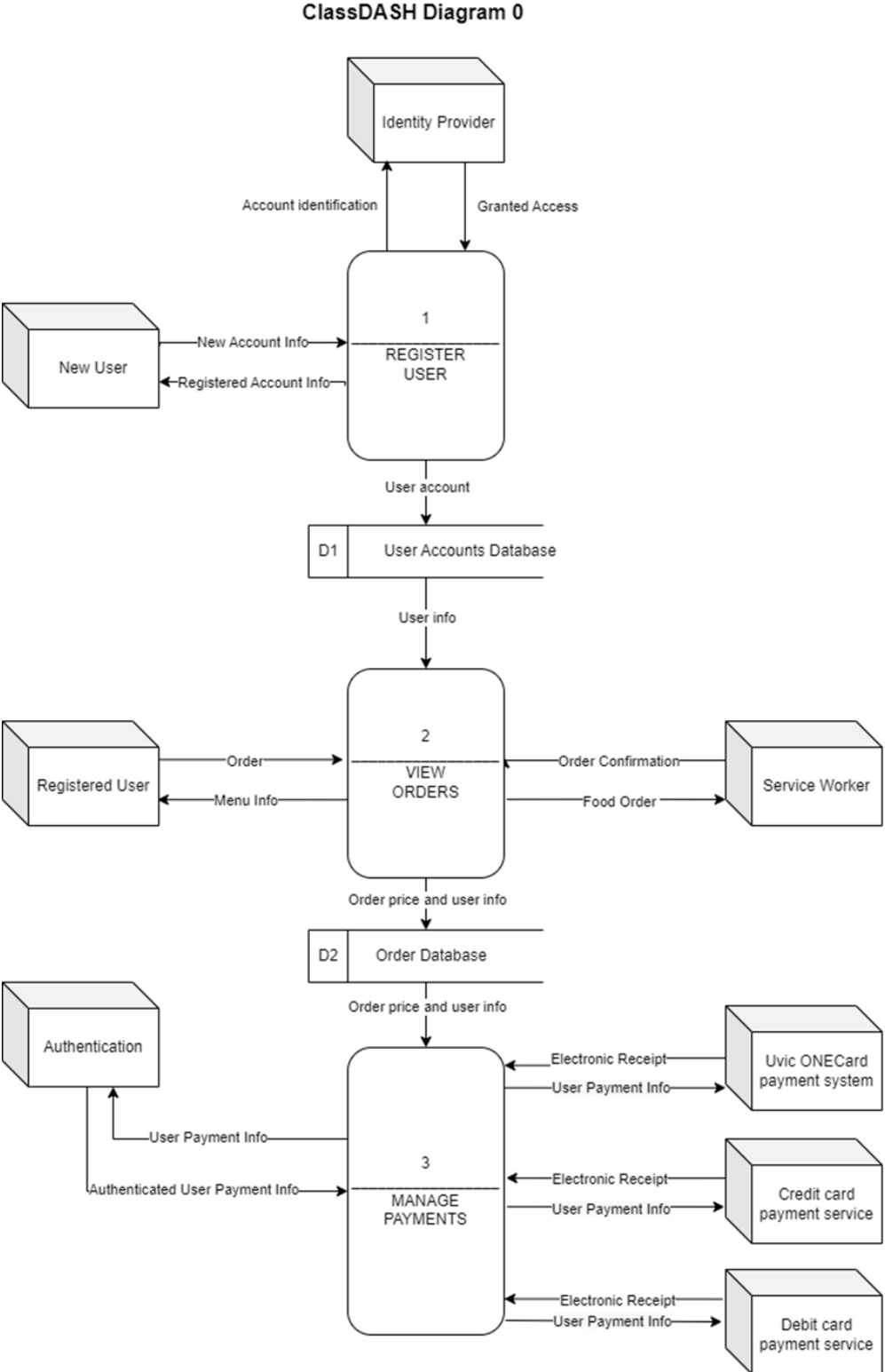
2.3 Domain Models

I. Context Diagram

ClassDASH Context Diagram



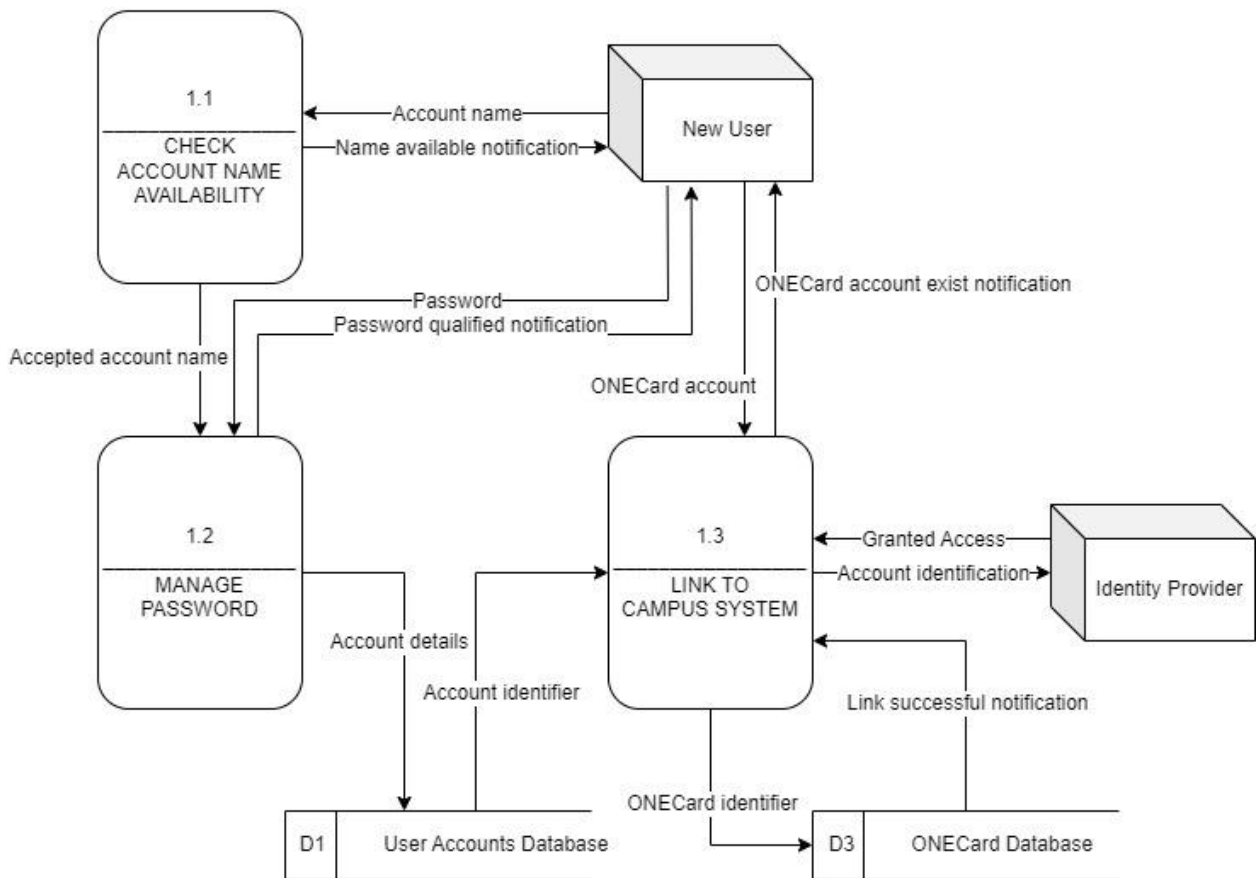
II. Diagram 0 for ClassDASH



III. Level 1 DFDs

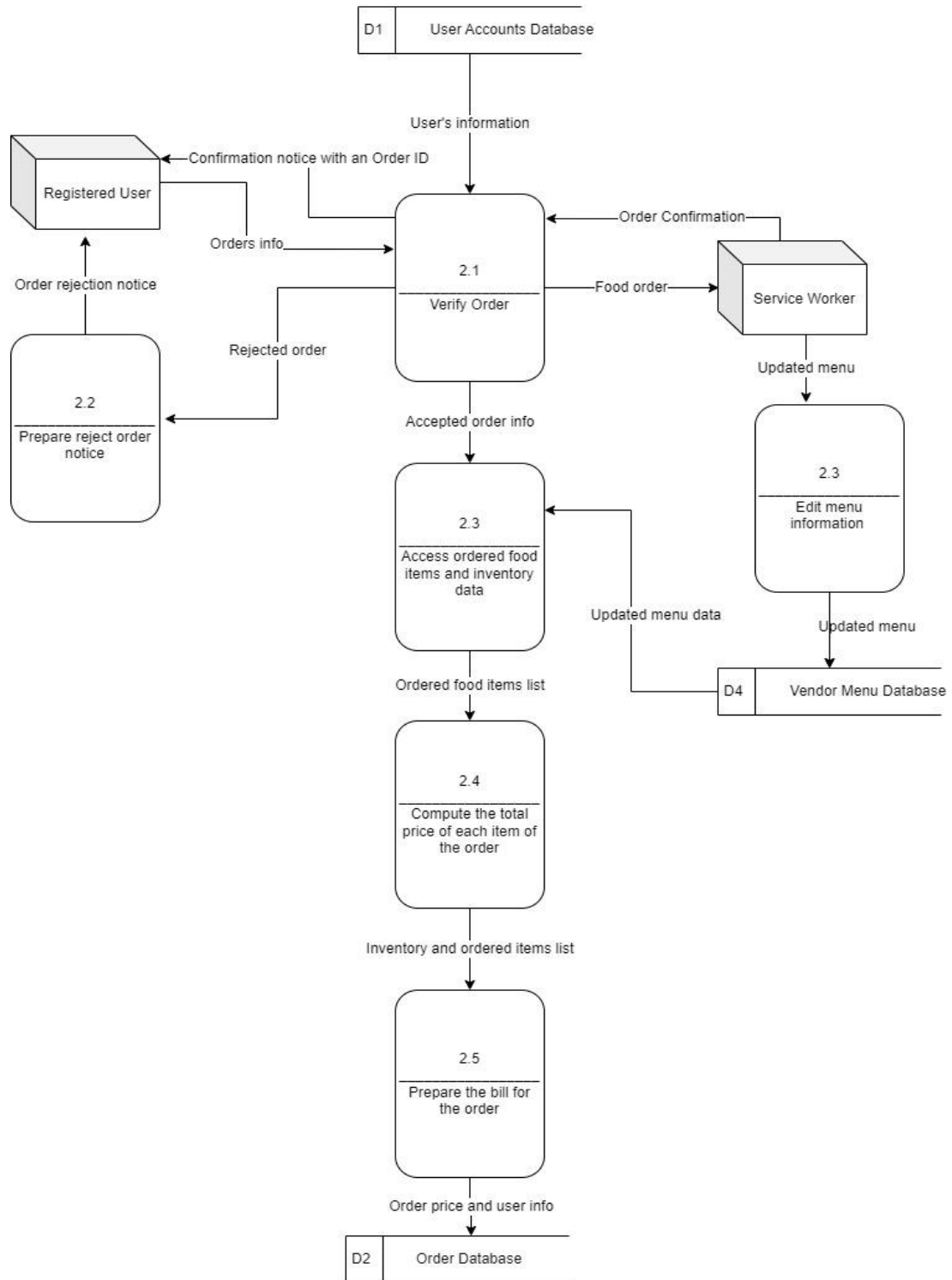
1. Register Users

ClassDASH Register User

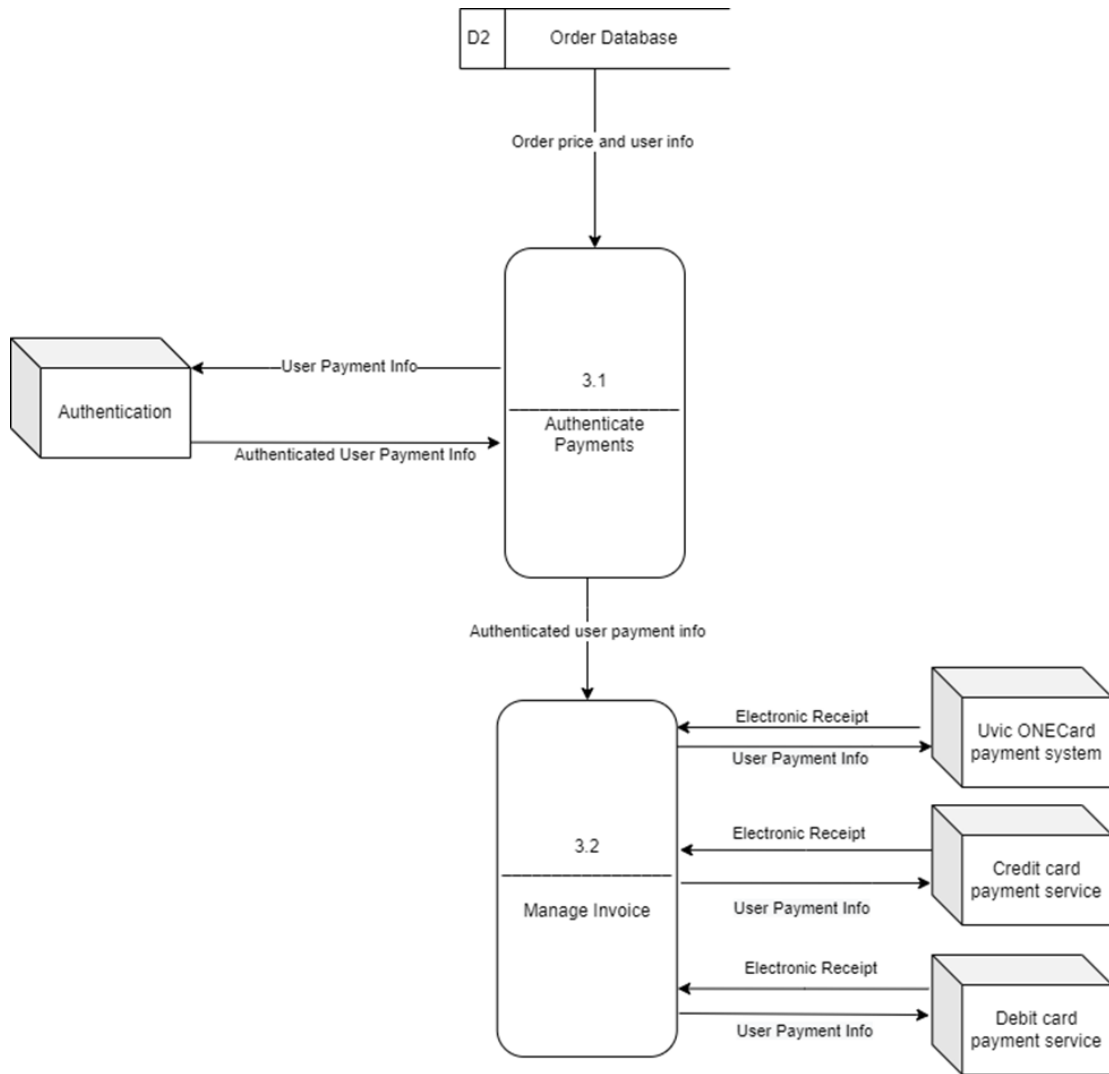


2. View Orders

ClassDASH Diagram 1 DFD for View Order



3. Manage Payments



3. Solution & Design

Next, through an iterative process, a prototype of the ClassDASH user interface was developed. With reference to existing use cases, the prototype displays the client story from logging in to picking up their order -- as well as the worker's receiving of their order.

3.1 User Interface (UI) Models

Opening the App and Logging In (Main Interface)

This UI refers to the user case 'UC-0' and the context diagram of the project ClassDASH.


The main interface only needs to provide functions for 'log in' and 'create account' for new users. Once someone logged in successfully, the main page will be replaced with 'log out' and 'browse menu' options. Based on our initial design, we do not want an unregistered users see the menu.




ClassDASH

ClassDASH







Powered by DoubleFast Inc. 

Powered by DoubleFast Inc. 

Client

User Registration & Error

These three UIs refer to UC-4 and DFD 1 Register users. of the project ClassDASH. These interfaces illustrate the account registration structure. Potential user errors that would be detected are password mismatch and invalid email entered. Once the page is correctly filled up without errors being detected and the Create Account button is pressed, it is redirected to the loading screen shown above as the third image. The red underlined cancel text brings the user back to the Create Account page if pressed.

Register as New	Register as New	Register as New
<h3>Create Account</h3> <p>Full Name John Deer</p> <p>Email JohnDeer@gmail.com</p> <p>Password 123456</p> <p>Confirm Password</p> <p>Password Mismatch</p> <p>Create Account</p> <p>Powered by DoubleFast Inc. </p>	<h3>Create Account</h3> <p>Full Name John Deer</p> <p>Email JohnDeer</p> <p>Password 123456</p> <p>Confirm Password</p> <p>Invalid Email</p> <p>Create Account</p> <p>Powered by DoubleFast Inc. </p>	 <p>Please wait the authority grant an access to you</p> <p><u>Cancel</u></p> <p>Powered by DoubleFast Inc. </p>

Login & Change Password


These two UIs refers to UC-4 'Change Password' and DFD-1 'Register Users' of the project ClassDASH. These interfaces illustrate the login structure and change password function. The red underline 'Forget password?' is the button to transfer to the second interface. We require users to confirm their new password and both new password and new password confirm must be the same.




ClassDASH ClassDASH

Account name	New password
Password	New password confirm

[Forget password?](#)

Powered by DoubleFast Inc. 

Powered by DoubleFast Inc. 

Browsing Menus & Ordering

Next, the user scrolls through the list of available vendors, noting which are open by the status on the vendor card and taps one to start an order. In the desired vendor's menu, the user chooses their order selection by tapping the plus icon. With their selections highlighted for visibility, the user may choose to tap the minus icon to remove the selection, or tap the button in the top right of the page to Checkout.

Choose a Food Vendor

Boardwalk Cafe

Coffee, espresso bar & baked goods

OPEN

Booster Juice

Juices, smoothies & specialty shots

CLOSED

Chopbox

Asian inspired noodles and curries with protein and vegetables

OPEN

Flamin' Good Grill

Burgers, fries, and BBQ favourites

OPEN

Order from Chopbox

Checkout

SIGNATURE CHOP BOXES \$8.00

TAMARIND PAD THAI +
Authentic Pad Thai with Shanghai noodles

MONGOLIAN BLACK BEAN +
Sweet & salty with ginger & sesame with Shanghai noodles

GREEN THAI CURRY +
Coconut curry flavoured with lime leaves & chilli with egg noodles

RED THAI CURRY +
Mildly spiced Thai coconut curry with egg noodles

PANANG CURRY +
Medium spiced with a hint of sweet & salty with egg noodles

RAMEN \$6.50

CLASSIC RAMEN +
Fresh noodles & vegetables in mushroom stock

ADD-ONS:

Tofu \$3.50 +

Softboiled egg \$2.00 +

Chicken \$3.50 +

Order from Chopbox

Checkout

SIGNATURE CHOP BOXES \$8.00

TAMARIND PAD THAI -
Authentic Pad Thai with Shanghai noodles

MONGOLIAN BLACK BEAN +
Sweet & salty with ginger & sesame with Shanghai noodles

GREEN THAI CURRY +
Coconut curry flavoured with lime leaves & chilli with egg noodles

RED THAI CURRY +
Mildly spiced Thai coconut curry with egg noodles

PANANG CURRY +
Medium spiced with a hint of sweet & salty with egg noodles

RAMEN \$6.50

CLASSIC RAMEN -
Fresh noodles & vegetables in mushroom stock

ADD-ONS:

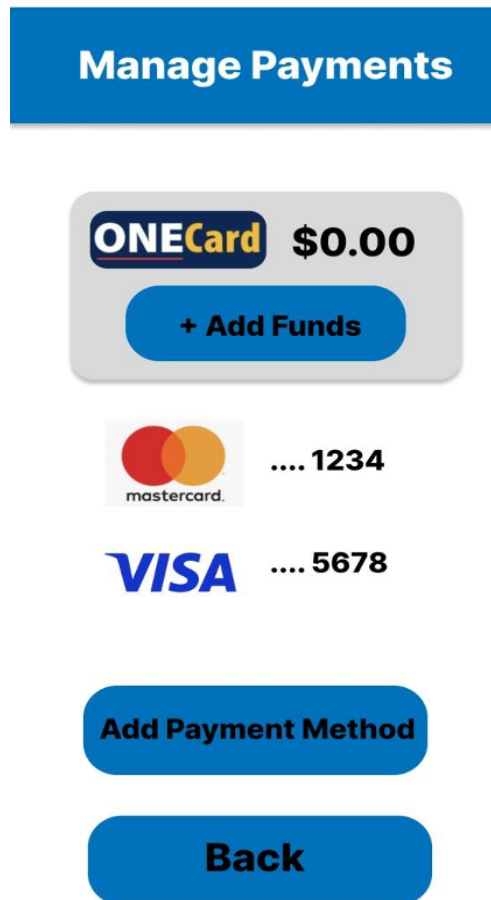
Tofu \$3.50 -

Softboiled egg \$2.00 -

Chicken \$3.50 +

Manage Payments – For a UVic Student:

This interface allows the user to manage their payments. If they are logged in as a student at the University of Victoria, they have the option to pay with their one card. The 'Add Funds' button takes them to UVic's "MyPage" where they can upload more funds to their OneCard. They can also add their choice of a Visa or Mastercard. This example contains both.



User Interface for "Checkout:

The checkout page displays a final overview of the items the user has placed. It allows them to choose a pickup time from a pre-set amount of times during the food outlet's operating hours. It gives the user a final total and shows which card they will be using to pay. After this, they can place their order.

Checkout



Chopbox

TAMARIND PAD THAI Authentic Pad Thai with Shanghai noodles	\$8.00
CLASSIC RAMEN Fresh noodles & vegetables in mushroom stock Add: Tofu Add: Softboiled egg	\$12.00

 **Friday, November 18th, 12:15pm**

Subtotal	\$20.00
Taxes and Fees	\$1.40
Total	\$21.40

VISA 5678

Place Order

Order Confirmation:

The order confirmation page gives the user their assigned order number that they need for pickup, an overview of the items they ordered, the location of the food outlet, and the pickup time they chose.

Order Confirmation

Order #: 12345678

Pickup Time: 12:15pm



Chopbox

- 1** **TAMARIND PAD THAI**
Authentic Pad Thai with Shanghai noodles

- 1** **CLASSIC RAMEN**
Fresh noodles & vegetables in mushroom stock
Add: Tofu
Add: Softboiled egg



Worker

This is the user interface for workers, once they logged in the main user interface by using their work account, this page will show and there are four options, and workers could change their status with “open” or “closed”.

ClassDASH

Current Order

Order History

Edit Menu

Edit profile

Status:

Open

Closed



Log out

Taking an Order

On this page, workers can see what's current order and there is a black bar on the right side to scroll down to the page to view more current orders, and the "back" button will lead worker back to their main user interface. Each order there is one "X" for cancel order and one "✓" mark for confirm order.

ClassDASH

Current Order

Order Number: 008971		✗	✓
Caesar salad	x1		
Cheese burger	x1		
Beef nachos	x1		
Order Number: 008973		✗	✓
Caesar salad	x2		
Cheese burger	x1		
Beef burger	x2		
Fries	x1		
Beef nachos	x1		
Order Number: 008974		✗	✓
Cheese burger	x5		

Editing a Menu

On this page, workers can edit their menu and there is a black bar on the right side to scroll down to the page to view menu, workers could categorize menu and type in their category in the white bar, they can also type in items under each category in the white bar, and on the right side of each category, there are plus buttons to add item under each category, and on the left side of each item, workers can delete item or on the right side of the item, workers can change price, add description and photo for the item but it's not required. The "back" button will lead worker back to their main user interface.

ClassDASH

Edit Menu

The screenshot displays a mobile application interface for editing a menu. It features a vertical list of menu categories, each with a plus sign (+) to its right. Under each category, there are several menu items, each with a minus sign (-) to its left and a three-dot menu icon (⋮) to its right. The categories and their items are:

- Appetizers** (+)
 - Beef nachos (⋮)
 - Spring roll (⋮)
 - Banh mi (⋮)
- Mains** (+)
 - Cheese burger (⋮)
 - Beef burger (⋮)
 - Crispy chicken burger (⋮)
- Salads** (+)
 - Caesar salad (⋮)
 - Potato salad (⋮)

4. Conclusions & Recommendations

Conclusion

In conclusion, the ClassDASH mobile application was created in order to optimize the various food outlets at UVic and save students and staff time when feeling hungry on campus. By the end of the timeline of this project, we have a prototype of ClassDASH that aims to avoid food waste on campus, clear congestion in front of UVic food outlets, and acquire at least a 50% user base for students who consume one meal on campus daily. By using lecture materials and pre-existing knowledge, the team was able to break down the project into various steps that ultimately brought it together in the end as a whole. Through the development of a project charter, functional and non-functional requirements, use cases, domain models, and UI models, ClassDash was created.

Recommendations

At the beginning of the stage, we had difficulties to develop our system, so the first recommendation is that before planning to develop an online food ordering system, it is important to understand the online food ordering system architecture first. The second recommendation is to improve the integration with food providers because there are different food providers on campus with different ordering and payment methods. For example, the dining hall in the new building is primarily a buffet. Students need to get food independently, but they can order food in university centers, so it's hard to integrate ordering ways. The third recommendation is that to be clear of requirements, when we first conduct our requirement document, some of our functional and non-functional requirements are not detailed, so it is necessary to add more details to the requirements specification. The fourth recommendation is to increase the internal communication within the group because some of our ideas are redundant or similar, so it is important to connect more with the group, brainstorm and share ideas together.

5. Team Members' Contributions

Coby Lam

- Project Charter: Project Organization
- Website
- Requirement development & client feedback
- Use Case-1: View Order
- DFD Model: Manage Payments
- UI models: Register Users

Emile Keruzore

- Project Charter Executive Summary
- Website Text
- Requirement development & client feedback
- Use Case-5: UpdateMenu
- DFD Model: Manage Payments
- UI Models: Browsing Menus & Ordering

Pengfei Li

- Project Charter: project summary, goal and project scope
- Nonfunctional Requirements
- Use Case-2: MakePurchase
- Requirement Documents
- DFD Model: Register Users
- UI Models: Register Users, Change Password

Shivani Ram

- DFD Model: View Orders
- Use Case-3: Checkout
- UI Models: Checking Out
- Requirements Document
- Project Charter - Work Breakdown Structure
- Request for Proposal

Zikai Hao

- DFD Model: View Orders
- Use Case-0: Level 0 diagram
- UI Models: Workers's Edit Menu, Take Orders & Update Menu
- Requirement Document
- Project Charter - Milestones, deliverables, and Gantt Chart
- Request For Proposal - Known Interactions and Known Constraints

6. Appendices

Glossary

Term	Definition
UVic	University of Victoria
OneCard	UVic's private, fully integrated payment and identification system
UVSS	The student union in the University of Victoria
ClassDash	The application name of the project
Acronym	Name in Full
UVic	University of Victoria

Term	Definition
UVSS	University of Victoria Students' Society
ROI	Return on Investment
UC	University Campus
SWOT	Strengths, Weaknesses, Opportunities, and Threats

Interview Notes

Client's Feedback on Requirement Specifications

- "For project scope, the main priority should be the development of a mobile application"
- "All user types (customers and workers) must be able to browse menus"
- "The ability to leave notes in regards to food allergies is a must"
- "All users regardless of payment methods log in: via their UVic account or create a guest account"
- "Supporting customer's friends picking up their order should be disregarded for the risks it presents"
- "No mapping function is needed as these already exist through UVic's website"
- "Workers must have a way to update the menus, to add new items or mark them as sold out"
- "We can limit the language to English only"
- "As the system is a mobile app, deploying to both Android and iOS is essential"