



**MANNING SCHOOL  
OF BUSINESS**

**Milestone 1,2, and 3**  
**Ahmed Siddiqui, Tom Condon, Andrew**  
**Feng**

# Table of Contents

## Milestone 1

Introduction.....	4
1.0 System Planning and Selection.....	5
1.1 Service Request Form.....	5
1.2 Scenario.....	6
1.2.1 Describing the Project Scope.....	6
1.2.2 Describing the Project Alternatives.....	6
1.2.3 Describing the Project Feasibility.....	7
1.2.3.1 Economic Feasibility.....	7
1.2.3.2 Operation Feasibility .....	7
1.2.3.3 Technical Feasibility.....	7
1.2.3.4 Schedule Feasibility .....	7
1.2.3.5 Legal and Contractual Feasibility.....	7
1.2.3.6 Political Feasibility.....	7
1.3 Dividing the Project into Manageable tasks and Gantt Chart.....	8
1.4 Estimating Tangible Cost and Benefits and Creating a Preliminary Budget..	10
1.5 Calculating ROI (%), Break-even, and Break-even Chart.....	12
1.6 Developing a Communication Plan.....	13
1.7 Determining Project Standards and Procedures.....	13
1.8 Identifying and Assessing Risk.....	13
1.9 Baseline Project Plan.....	14
1.10 Preparing a Project Scope Statement.....	15
2.0 User Requirements .....	16
3.0 System Requirements.....	16

## Milestone 2

<b>1.0 Data Flow Diagrams.....</b>	<b>18</b>
1.1. Context Diagram.....	18
1.2. Level-0 Diagram.....	19
1.3. Leveled Diagrams.....	20
1.3.1. Level 1 Data Flow Diagram.....	21
1.3.2. Level 2 Data Flow Diagram.....	25
1.3.3. Primitive Data Flow Diagram.....	25
1.3.4. Gap Analysis.....	26
<b>2.0 Entity Relationship Diagrams.....</b>	<b>27</b>
2.1. Conceptual Entity Relation Diagram.....	27
2.2. Relationships and Cardinalities.....	27
2.3. Entity Relationships.....	28

<b>3.0 Data Dictionary.....</b>	<b>29</b>
---------------------------------	-----------

## Milestone 3

### 1.0 Form and Report Design

1.1 Specification-Deliverables and Outcomes (Forms).....	33
1.2 Specification-Deliverables and Outcomes (Reports).....	48

### 2.0 Interface and Dialogue Design)

2.1 Specification-Deliverables and Outcomes).....	50
2.1.1 Website.....	50
2.1.2 Databases.....	57
2.2 Guidelines	
2.2.5 Design the Dialogue Sequence Diagram.....	65

### 3.0 Documentations

3.1 User Manual.....	66
3.2 User Training Material.....	76
3.3 System Documentation.....	77
3.3.1 Source Code.....	77
3.3.2 Technical Report.....	87
3.3.3 Maintenance Manual.....	89
3.3.4 Testing Data.....	90
3.3.5 Errors in Our System.....	95

## Introduction

University of Massachusetts Lowell (UMass Lowell) is a State University in Massachusetts, Lowell. It is a part of the UMass system of schools located around Massachusetts, namely UMass Amherst, UMass Boston, UMass Dartmouth, UMass Lowell, and UMass Medical. UMass Lowell was created in 1972 when Lowell State College and Lowell Technological Institute joined hands to work together to offer more and mixed opportunities to their students.

UMass Lowell is a well known university for its value filled education, staff, accreditations, and love of sports. It is one of the very first and few Universities to have its Business School be accredited by the Association to Advance Collegiate Schools of Business International. UMass Lowell is a University that has continued to grow as more students enroll for wide arrangements of bachelor's, master's, and doctoral degrees.

With the recent return to campus from the Covid-19 pandemic students will also resume living on campus or near campus, with that comes the issue of the dining halls and overcrowding in the dining halls. Currently the only way for students to make use of the meal plan is to walk into the dining hall, order food from the counter or grab the ready food under the warmers and eat while seated in the dining hall. Being forced to dine at the dining hall makes it hard for students who are in between classes or running late.

With the new system in place students will be able to order food at the dining hall and have it ready by the time they come in and be able to pick it up and take it wherever they go, whether it be to their next class, to their team project meeting location, to their dorm, or wherever else they may need to go if they don't have the time to spend in the dining hall itself.

## 1.1 Service Request Form

REQUESTED BY: Professor Edward T. Chen    DATE: September 16, 2021

DEPARTMENT: Dining Hall

LOCATION: Lowell, University of Massachusetts

CONTACT: Tel (978)-934-2756

### TYPE OF REQUEST

- System Enhancement

### URGENCY

- Business Losses can be tolerated until the system enhancement is completed

### PROBLEM STATEMENT

With the return of students after Covid-19, we need to improve the safety measures for students with meal plans returning to campus. There is currently no system for on the go ordering. With an on the go ordering system students will be able to avoid the crowding in the dining halls which will make complying with covid regulations easier.

### SERVICE REQUEST

Requisition for a new system with capabilities to handle transactions in web format, Student Data, Food Data, Assist the management team with adding and removing items from the menu.

IS LIAISON: Ahmed Siddiqui, Tom Condon, Andrew Feng

SPONSOR: MIST.4020

\_\_\_\_\_TO BE COMPLETED BY SYSTEM PRIORITY BOARD\_\_\_\_\_

[     ]     REQUEST APPROVED

START:

ASSIGNED TO:

[     ]     RECOMMENDED REVISION

[     ]     SUGGEST USER DEVELOPMENT

[     ]     REJECT:

### **1.2.1 Describing the Project Scope**

This project is going to design a new web based system that will help students order food for “on the go” orders using RiverHawk Dollars that come with their Student Meal Plans. The web platform will handle online orders of the available items on the menu and take in student information. The database system will allow the managers or those in the position of authority to change the menu for that allotted day or time.

In all this system is going to make busy students or students who will have a hard time grabbing lunch during their tight schedule to be able to grab a meal before heading off to class, work, or meeting in the case that they may end up having to miss a meal and not enjoy what's available that day in the dining hall.

### **1.2.2 Describing the Project Alternatives**

A possible alternative to this may be that students order at restaurants or cafes around campus. This is an alternative but it's not efficient for students as there would be increased wait times in each of these restaurants and it would require the students to utilize actual money apart from their RiverHawk Dollars or their meal plan. It would be much simpler for the students to be able to grab a fresh hot lunch from the dining hall as they leave for the main campus rather than have to run to one of the open and packed restaurants outside/around campus and grab their food. In the case that these restaurants have “on the go” ordering they are usually packed with other students and understaffed to be able to timely handle “on the go” ordering.

The issues that arise for these alternatives is that even if a restaurant or cafe were to hire more people to accommodate its influx of customers it would be hard to expand their borders/doors to accommodate the larger number of people because of the price of property near the ever expanding campus. In the option that a restaurant or cafe was to expand its business it would require it to go under construction which would cause it to close its doors for a few days which could potentially push its customers to other nearby restaurants or cafes.

For UML's “on the go” ordering the benefit is that a student can order at any of the campuses and go pick up their food. If a student for instance dorms or lives near south campus but needs to head to north campus for class or a meeting then they can order at the north or east dining halls, grab their food and go.

### **1.2.3.1 Economic Feasibility**

#### **Benefits**

- Error Reduction
- Increase speed of activity
- Increase dining hall sales
- Improve dining hall service

#### **Costs**

- Training
- System Development
- Maintenance

### **1.2.3.2 Operational Feasibility**

This system allows for the ability to avoid the overcrowding of the dining hall during peak hours and provide flexibility to customers with tight schedules. It allows the dining experience to be more efficient and stress free.

### **1.2.3.3 Technical Feasibility**

The System requires certain tech skills such as HTML and CSS as it requires a development of a user interface.

### **1.2.3.4 Schedule Feasibility**

The Gantt and Pert chart estimate that the project will be done in late March if the project is started as soon as possible. The school would like it to be completed as soon as possible so it can be used in the spring semester.

### **1.2.3.5 Legal and Contractual Feasibility**

The team must be considerate of student's sensitive information when developing the system and make sure security measures are in place.

### **1.2.3.6 Political Feasibility**

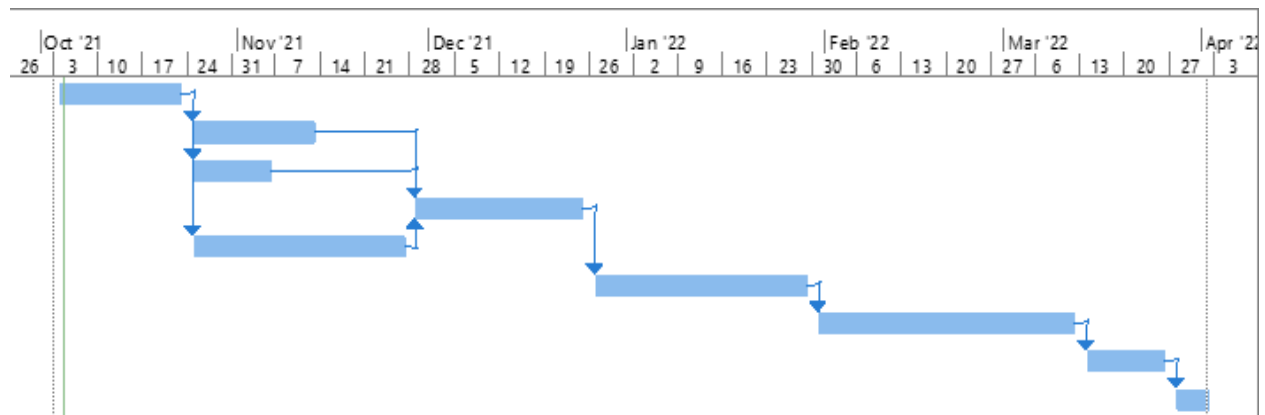
The project is determined to have limited risk associated with this.



## 1.3 Dividing the Project into Manageable tasks and Gantt Chart

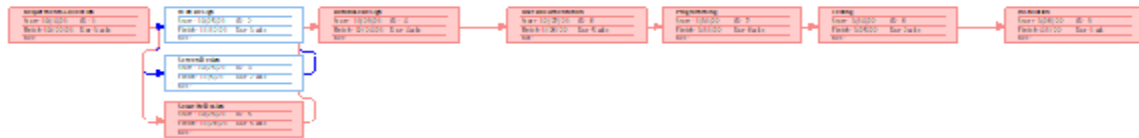
Gantt

ID	Task Name	Duration	Start	Finish	Predecessors	Finish Slack	Critical
1	Requirements Collection	3 wks	Mon 10/4/21	Fri 10/22/21		0 wks	Yes
2	Web Design	3 wks	Mon 10/25/21	Fri 11/12/21	1	2 wks	No
3	Screen Design	2 wks	Mon 10/25/21	Fri 11/5/21	1	3 wks	No
4	Database Design	4 wks	Mon 11/29/21	Fri 12/24/21	3,2,5	0 wks	Yes
5	Security Design	5 wks	Mon 10/25/21	Fri 11/26/21	1	0 wks	Yes
6	User Documentation	5 wks	Mon 12/27/21	Fri 1/28/22	4	0 wks	Yes
7	Programming	6 wks	Mon 1/31/22	Fri 3/11/22	6	0 wks	Yes
8	Testing	2 wks	Mon 3/14/22	Fri 3/25/22	7	0 wks	Yes
9	Installation	1 wk	Mon 3/28/22	Fri 4/1/22	8	0 wks	Yes



## Pert

Activity	Optimistic	Realistic	Pessimistic	Estimated Time (O+4R+P)/6	Preceding Activity
Requirements Collection	2	3	4	3	-
Web Design	2	3	4	3	1
Screen Design	1	2	3	2	1
Database Design	3	4	5	4	2,3,5
Security Design	4	5	6	5	1
User Documentation	4	5	6	5	4
Programming	5	6	7	6	6
Testing	1	2	3	2	7
Installation	1	1	1	1	8



## Activity Slack Time

Activity	TE	TI	Slack TE-TI	Critical Path
Requirements Collection	3	3	0	Yes
Web Design	6	8	2	No
Screen Design	5	8	3	No
Database Design	12	12	0	Yes
Security Design	8	8	0	Yes
User Documentation	17	17	0	Yes
Programming	23	23	0	Yes
Testing	25	25	0	Yes
Installation	26	26	0	Yes

## 1.4 Estimating Tangible Cost and Benefits and Creating a Preliminary Budget

<b>Tangible Benefits</b>		
<b>Umass Lowell system project</b>		
		Year 1 through 5
A	Decrease Wait time	15,750
B	Increase flexibility	2,000
C	Error Reduction	5,000
D	Cost Reduction	4,000
E	Inventory	2,500
F	Other	0
Total Tangible Benefits		29,250

<b>One-Time Costs Worksheet</b> <b>Umass Lowell system project</b> Year 1 through 5		
	System Development	
A	Costs	9,500
B	New Hardware	3,300
C	New Software	4,200
D	User Training	2,500
E	Station Preparation	3,500
F	Other	0
Total Tangible Benefits		23,000

<b>Recurring Cost Worksheet</b> <b>Umass Lowell system project</b> Year 1 through 5		
	Application software	
A	maintenance	6,000
B	Supplies	2,400
C	Annual training	4,050
F	Other	0
Total Tangible Benefits		12,450

## 1.5 Calculating ROI%, Break-even, and Break-Even Chart

### Economic Feasibility Analysis

Umass Lowell

On the Go online Ordering Project

		Year of Project						
		0	1	2	3	4	5	Totals
Net Economic Benefit		\$0.00	\$ 29,250	\$ 29,250	\$ 29,250	\$ 29,250	\$ 29,250	
Discount Rate	12%	1	0.892857143	0.797193878	0.711780248	0.63551808	0.56742686	
PV of Benefits		\$0.00	\$ 26,116	\$ 23,318	\$ 20,820	\$ 18,589	\$ 16,597	
NPV of all Benefits		\$0.00	\$ 26,116	\$ 49,434	\$ 70,254	\$ 88,842	\$ 105,440	<b>\$ 105,440</b>
One time Costs	\$	(23,000)						
Recurring Costs		\$0.00	\$ (12,450)	\$ (12,450)	\$ (12,450)	\$ (12,450)	\$ (12,450)	
Discount Rate	12%	1	0.892857143	0.797193878	0.711780248	0.63551808	0.56742686	
PV of Recurring Costs		\$0.00	\$ (11,116)	\$ (9,925)	\$ (8,862)	\$ (7,912)	\$ (7,064)	
NPV of All Costs	\$	(23,000)	\$ (34,116)	\$ (44,041)	\$ (52,903)	\$ (60,815)	\$ (67,879)	<b>\$ (67,879)</b>
Overall NPV								<b>\$ 37,560</b>
ROI= -(Overall NPV/NPV of All Costs)								<b>0.55</b>

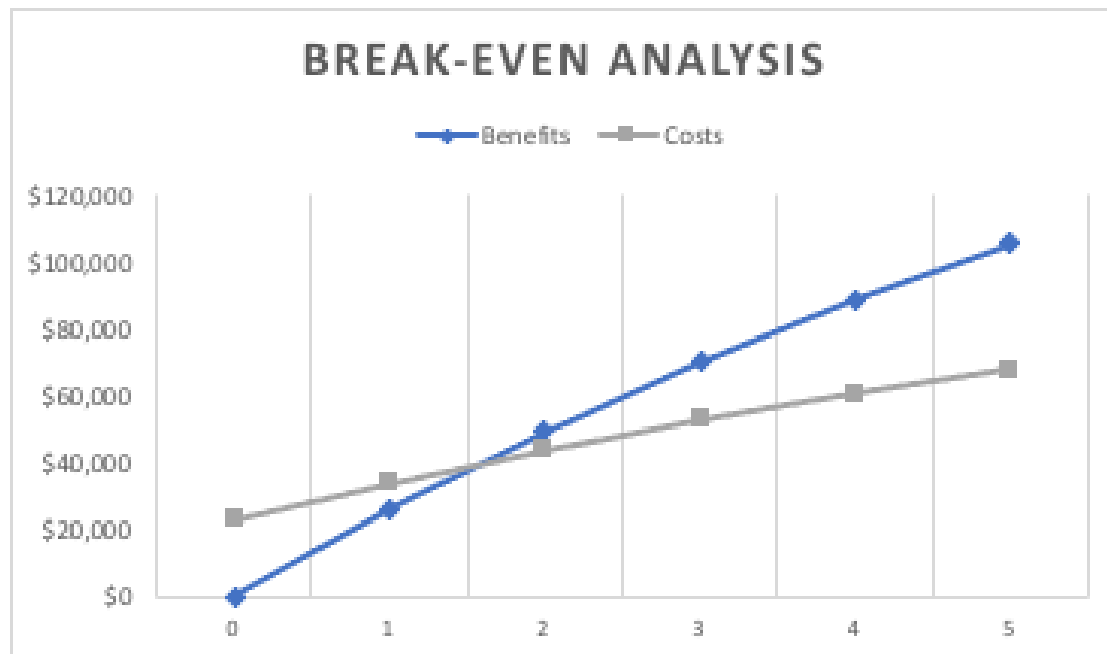
Break-Even Analysis	0	1	2	3	4	5
Yearly NPV CashFlow	\$ (23,000)	\$ 15,000	\$ 13,393	\$ 11,958	\$ 10,677	\$ 9,533
Overall NPV CashFlow	\$ (23,000)	\$ (8,000)	\$ 5,393	\$ 17,351	\$ 28,027	\$ 37,560

Project Break-Even Occurs between years 1 and 2

Use first year of positive cash flow to calculate break-even fraction 0.597

**Actual break-even occurred at 1.597 years**

**Break even occurs May 10th, 2023**



## 1.6 Developing a Communication Plan

Reason	Method	Objective	Members	Date
First Team Meeting	Face to Face	Get to know group members and understand each other's time tables	Team	Sept. 9, 2021
System Analysis	Zoom Call	Analyzing the current limitations of the system in place and planning on the upgrades eligible for the current system	Team	First Monday of Month
Design Analysis	Zoom Call	Designing the layout plan for the system implementation.	Team	Third Monday of Month
Project Analysis	Zoom Call	Discussing the project schedule and discussing any additional complaints or issues	Team	First Monday of Month
Project Roundup	Zoom Call	Rounding up and testing the system to see if functional	Team, Stakeholders	Third Monday of Month
D-Day / Project Release	Zoom Call	Project assessment and implementation	Team	First Monday of Month

## 1.7 Project Standards and procedures

To determine if the system is working up to standards, we will rely heavy on students' feedback. These feedbacks will be used to improve our system and ensure that our customers' satisfaction is high. The data collected from these feedbacks will be used to create a system that will be beneficial to both students and faculty.

## 1.8 Identifying and Assessing Risks

Our system features an online ordering service and it requires customers to input their sensitive information such as credit cards and phone numbers. Since there is sensitive information involved, there is always a risk of unauthorized users attempting to steal it. Features such as requiring a username and password will greatly decrease this threat.

## 1.9 Baseline Project Plan

### Baseline Project Plan Report

#### *Introduction*

- A. Project overview-** With the return of students onto the campus, the university is looking for a way to reduce capacity in the dining halls.
- B. Recommendation-** With an online “on the go” ordering system students can place orders for the dining hall and then walk to the dining hall to pick them up.

#### System Assessment

- C. Alternative-** Some project alternatives could be that students could order food from restaurants around campus. This would limit the amount of exposure on campus and reduce the system requirement for the university.
- D. System Description-** A new web program for the “on the go” ordering, so students can order food and pick it up at the dining halls.

#### Feasibility Assessment

##### **E. Economic Analysis-**

- Benefit- \$29,250
- Recurring Cost- \$12,450
- One Time Cost- \$23,000
- ROI- 55%
- Overall NPV- \$37,560

- F. Technical Analysis-** Employees with tech skills such as HTML and CSS as it requires a development of a user interface.

- G. Operational Analysis-** The proposed system would reduce the foot traffic in the dining hall in accordance to the covid 19 regulations.

- H. Legal and Contractual Analysis-** The team must be considerate of student’s personal information when developing the system and make sure security measures are in place.

- I. Political Analysis-** This project has no risk of political association.

- J. Schedule Analysis-** The project is estimated to be done in late March if the project is started as soon as possible. The school would like it to be completed as soon as possible so it can be used in the spring semester.

#### Management Issues

- K. Team Configuration and Management-** Ahmed, Andrew, and Tom will be tasked with completing this project. All roles and responsibilities will be split up evenly.
- L. Communication Plan-** All meetings for discussion along with the team will happen on Mondays.
- M. Project Standards and Procedures-** The milestones will be reviewed by a Project Sponsor who will be overseeing the project.

## 1.10 Preparing a Project Scope Statement

UMass Lowell Dining “on the go”

Project Scope Statement

### *General Project Information*

**Project Name:** UMass Lowell Dining “on the go”

**Sponsor:** Edward Chen

**Project Manager:** Ahmed Siddiqui, Thomas Condon, Andrew Feng

### *Problem/Opportunity Information*

UMass Lowell has over 15000 students and this opportunity provides a major convenience to the student, which in turn increases their satisfaction towards the dining experience. It also eases overcrowding in the dining hall during peak hours.

### *Project Objectives*

The system will be created to provide the students with an easy and convenient way to order food. It allows the people that have a tight schedule to have a quick bite on the fly without the hassle of waiting in line.

### *Business Benefits*

- Increase in Meal Plans Sales
- Online order and payments
- Increase customer satisfaction

### *Project Deliverables*

- Website
- Software for online ordering
- Payment system



## **2.0 User Requirements**

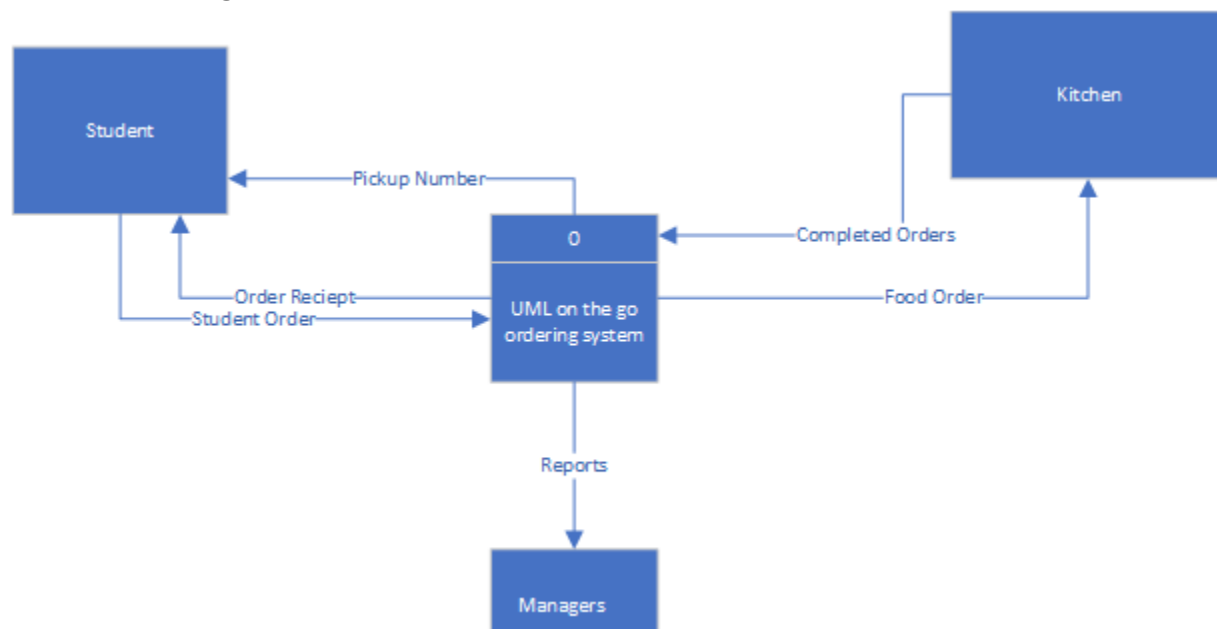
Through the system, users are able to create online orders and receive a notification when their orders are ready for pick up. A smart device such as a phone or laptop that has internet access is required in order to access the system.

## **3.0 System Requirements**

Hardware: Display, Keyboard, Memory, Processor

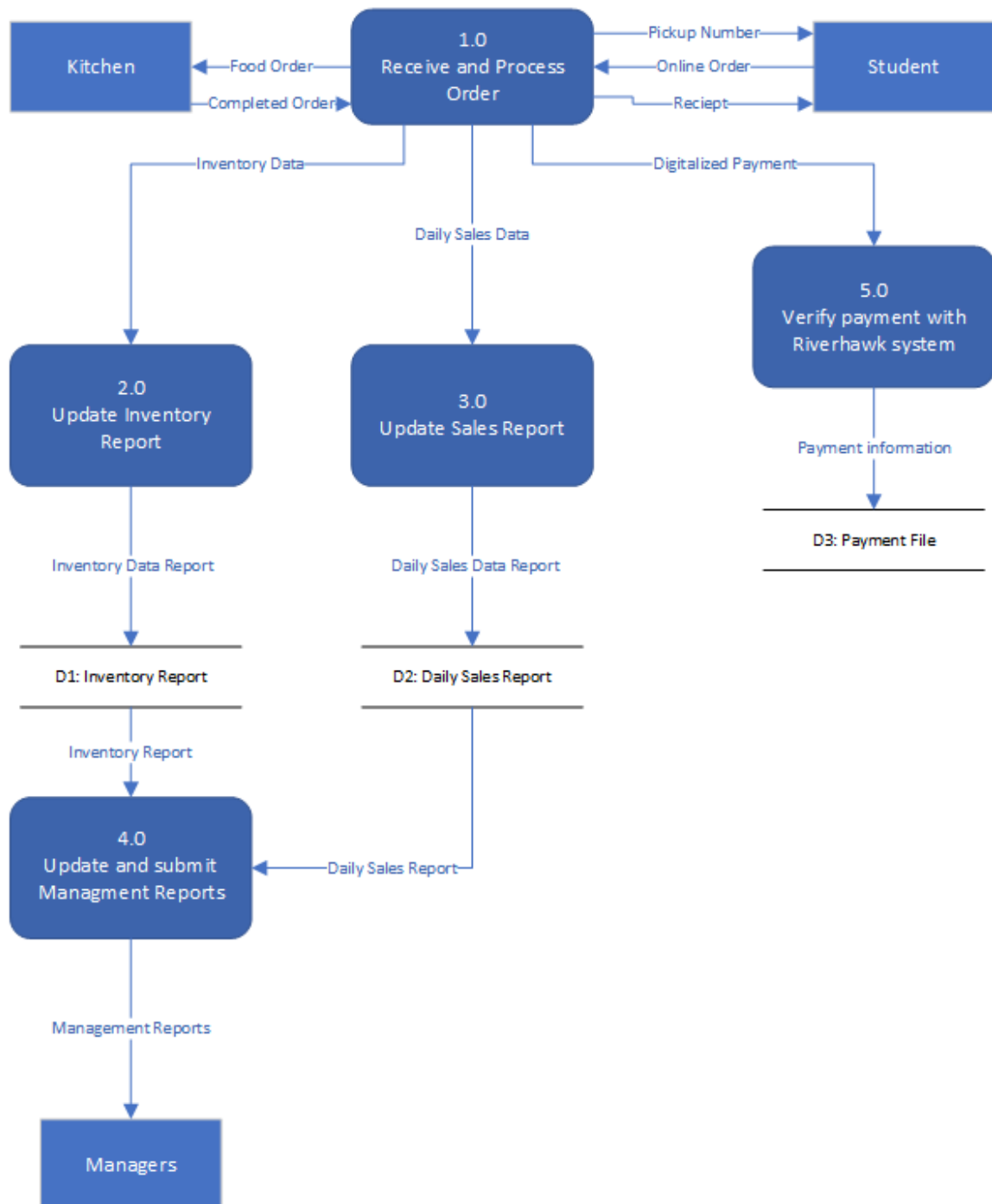
Software: System Software (Windows, MAC OS , Linux), Programming Software (HTML, CSS, Python), Application Software (Microsoft Suite, Internet Browsers)

## 1.1 Context Diagram



The Context Diagram for “ON-THE-GO Ordering” is shown above. First, the student will place an order in the online ordering system (noted as “Student Order” on the data flow). The Ordering System will then send the order information to both the Kitchen and the Managers (noted as “Food Order” and “Reports” on the data flow). The Kitchen will alert the Ordering System when the order is complete (noted as “Completed Orders” on the data flow) and let the student know when the order is ready for pick up (noted as “Pickup Number” on the data flow). Afterwards, the system will send out a receipt to the student (noted as “Order Receipt” on the data flow).

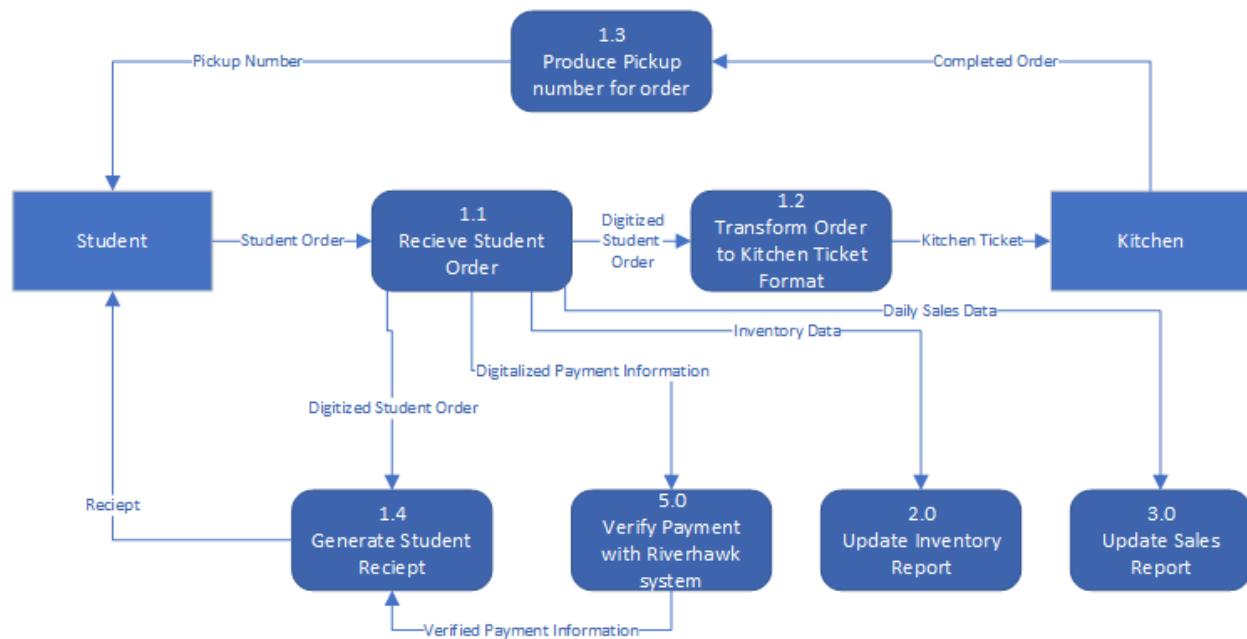
## 1.2 Level 0 Diagram



Above is the Level 0 diagram, The student will place an order, using the Riverhawk payment system. The ordering system will receive and process the order information. Afterward, the kitchen will receive the order and the system will alert the student that their order is ready for pick up. The system will also collect the daily sales data and the inventory data and file it in the daily sales report and the inventory report. This information will then be used to produce the management reports, which will go to the managers.

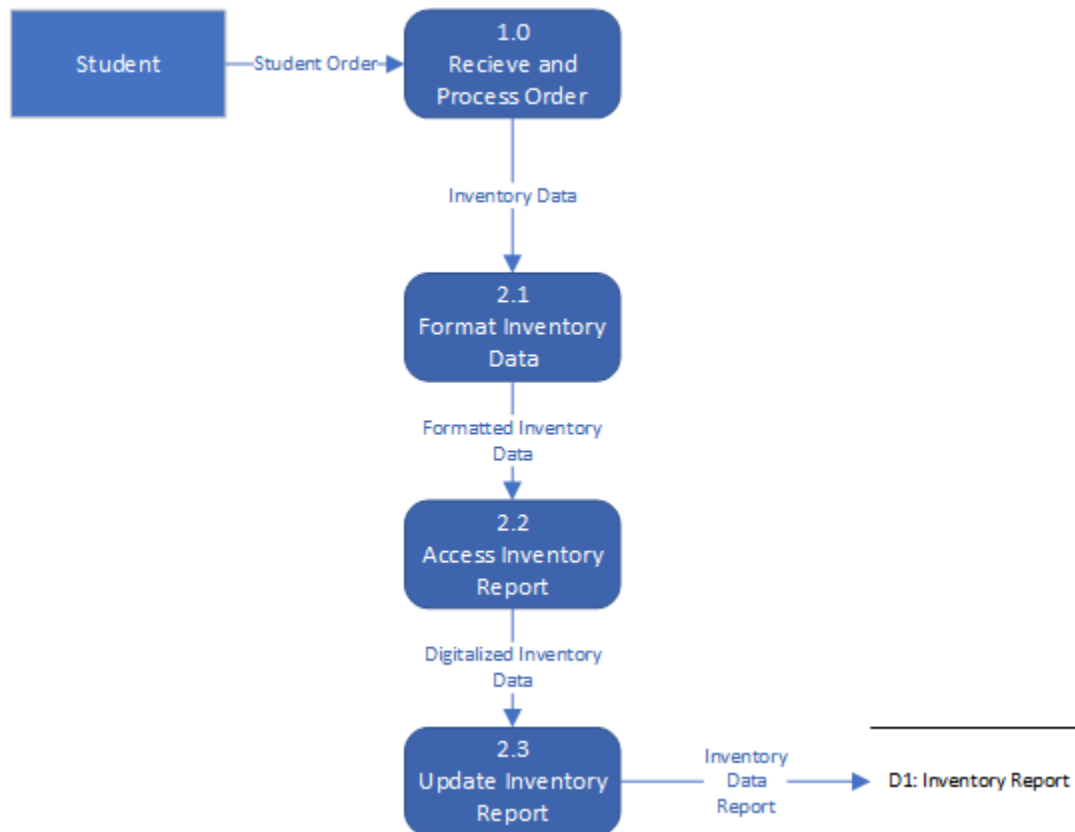
### 1.3.1 Level 1 Diagrams

#### Level 1 Diagram for Process 1.0



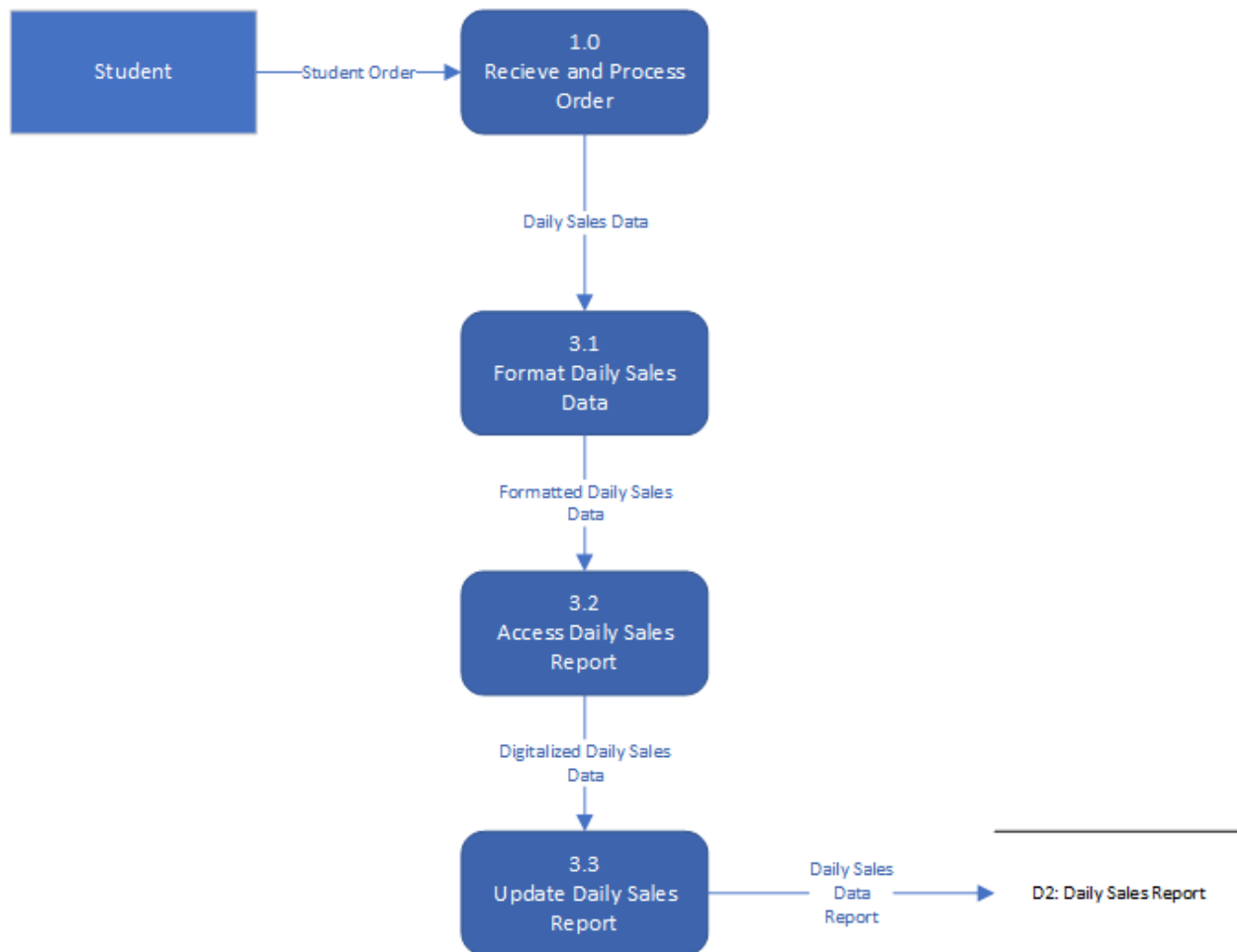
Above is the level 1 diagram for process 1.0 (Receive and Process Order). The system will take the student's order, paid using the Riverhawk payment system, and digitize it into a kitchen ticket format. The kitchen will then receive the order and begin preparing it. Once the order is finished, it will alert the student that their order is ready for pick up. The system will take the daily sales data and the inventory data and update both reports respectively as well. The system will also use the digitized student order and the digitized payment to generate a receipt, which will be sent to the student.

### Level 1 for Process 2.0



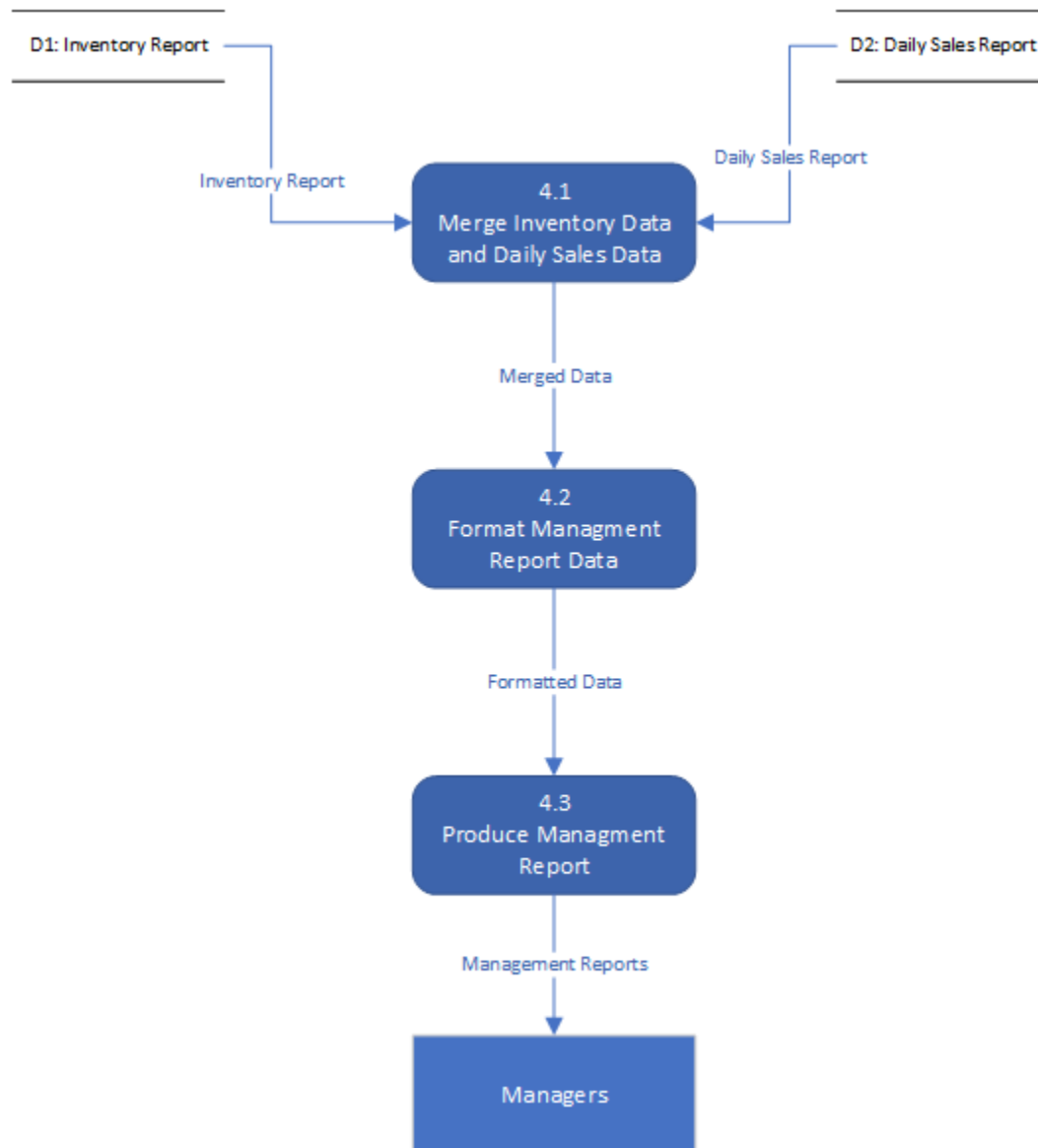
The diagram above is the level 1 diagram for process 2.0 (Update Inventory Report). Once the student's order has been received and processed, the system will format the inventory data. Afterwards, the system will access the inventory report and update the file.

### Level 1 Diagram for Process 3.0



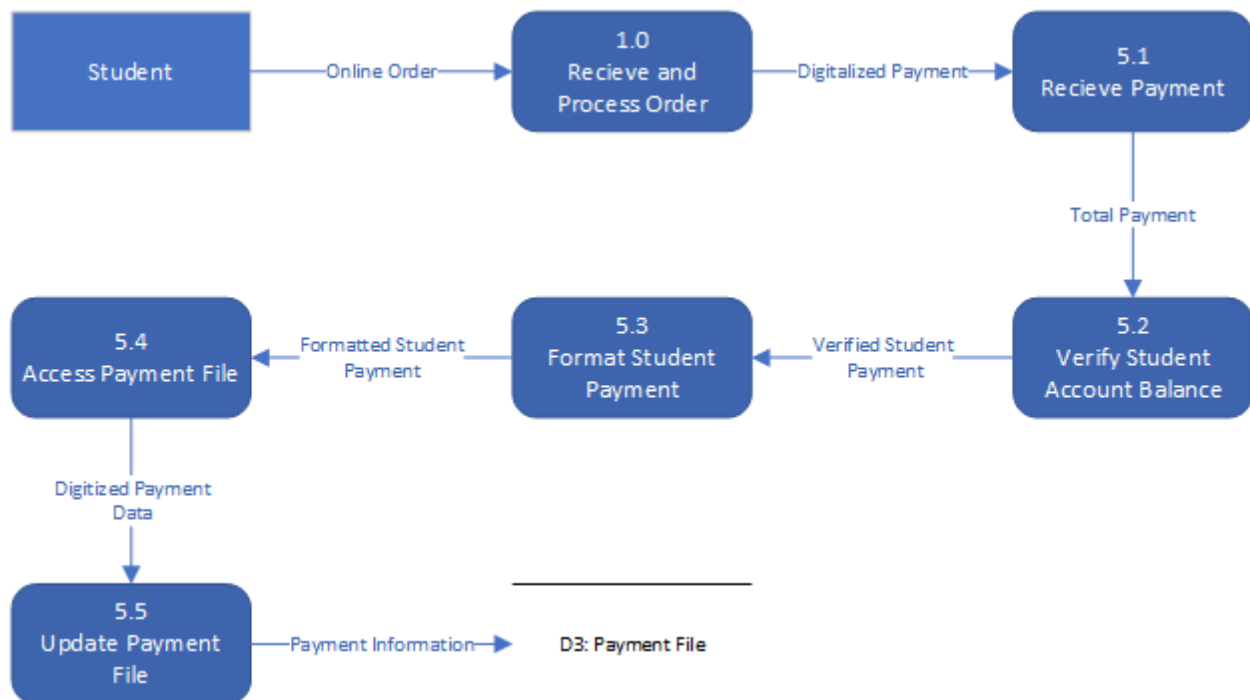
Above is the level 1 diagram for process 3.0 (Update Sales Report). Once the student's order has been received and processed, the system will format the daily sales data. Afterwards, the system will access the daily sales report and update the file.

## Level 1 Diagram for Process 4.0



Above is the level 1 diagram for process 4.0 (Update and Submit Management Report). The system will merge the inventory data and the daily sales data. Afterwards, it will format and produce the management report, which will be sent directly to the managers.

### Level 1 Diagram for Process 5.0

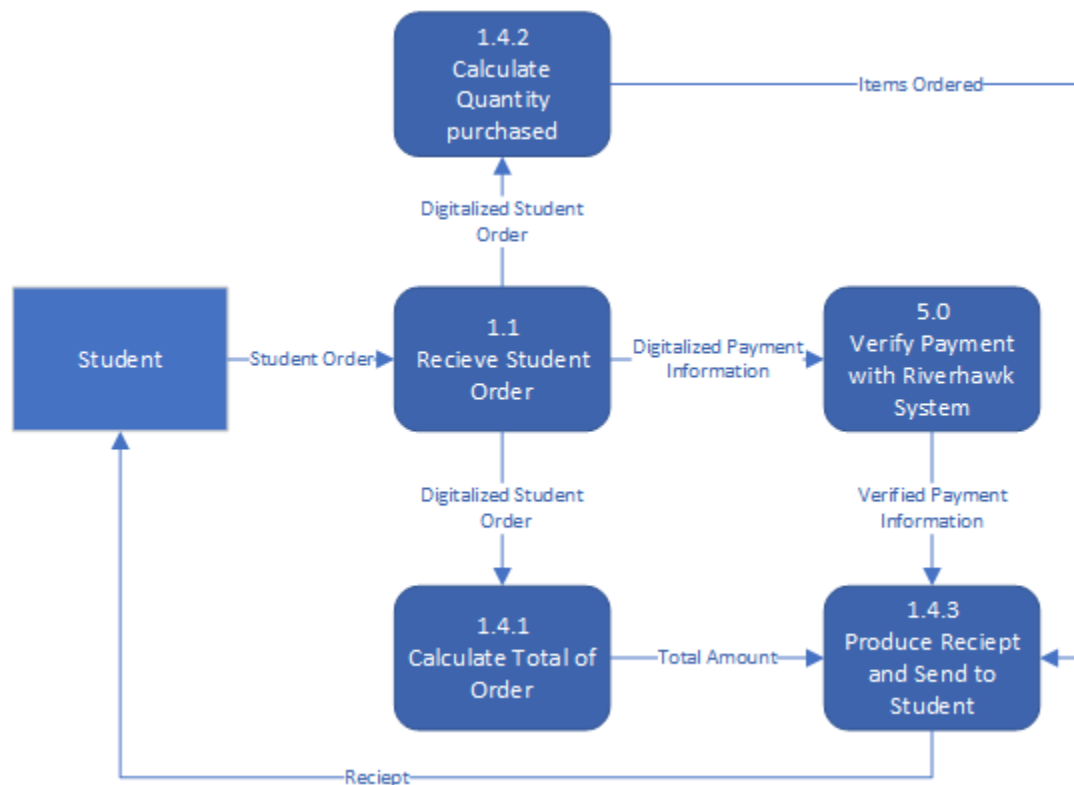


Above is the level 1 diagram for process 5.0 (Verify Payments with Riverhawk System). The system will receive the student's payment and verify the student's account balance. The system will then format the payment and access the payment file to make any updates to the file.



### 1.3.2 Level 2 Diagrams

#### Level 2 for Process 1.4



Above is the level 2 diagram for process 1.4 (Generate Student Receipt). After receiving the student's order, the system will calculate the quantity purchased and calculate the total of the order. The system will also digitalize the payment information and verify the payment with the Riverhawk system. Once the total of the order has been calculated and the payment has been verified, the system will use this information to produce a receipt. The receipt will then be sent to the student.

#### Naming Standards for Processes

The processes in the system are represented as a blue square with a rounded edge. They represent the procedure performed on the data. The process symbols include n (the number of the process) and action verb + noun (what the process does).

#### Naming Standards for Data Flows

Data flows are represented as an arrow in all Data Flow Diagrams. The arrow represents the data moving from one process to the next. All data flows are given a unique name as well and the unique name is a noun in the middle of the arrow.

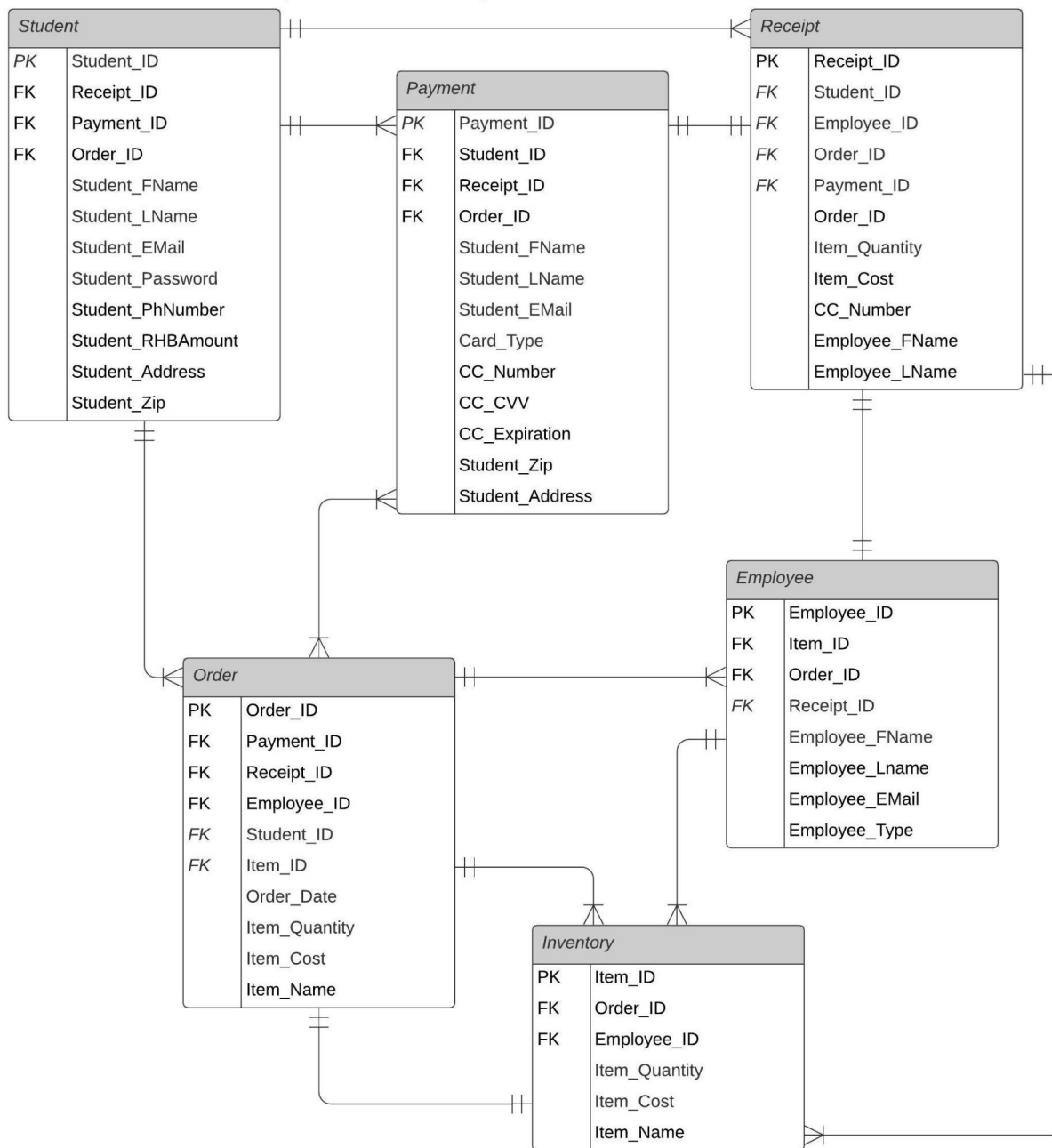
**Data Consistency for Data Flows and Data Stores**

Each Diagram includes the same data from the level 0 diagram.

**Data Miracles and Black Holes**

UML On the Go follows all data-flow diagramming rules. Processes are unable to have only outputs or only inputs. In our context diagram, “Student Order”, and “Food Order” flow into the system. “Reports”, and “Receipt” flow out of the system.

## 2.1 Conceptual Entity Relation Diagram



## 2.2 Entity Relationship Diagrams

### Cardinalities

- Student - Student\_ID (PK)
- Receipt - Receipt\_ID (PK), Student\_ID (FK), Employee\_ID (FK), Item\_ID (FK), Order\_ID (FK)
- Payment - Payment\_ID (PK), CC\_Number (FK), Student\_ID (FK)
- Employee - Employee\_ID (PK)

- Inventory - Item\_ID (PK), Employee\_ID (FK), Order\_ID (FK)
- Order - Order\_ID (PK), Payment\_ID (FK), Student\_ID (FK), Employee\_ID (FK)

Student	1 - M	Order
Student	1 - M	Receipt
Student	1 - M	Payment
Payment	1 - 1	Receipt
Payment	M - M	Order
Receipt	1 - 1	Employee
Receipt	1 - M	Inventory
Order	1 - M	Employee
Order	1 - M	Inventory
Employee	1 - M	Inventory

## 2.3 Entity Relationships

### Relationships

- Student, Order, Receipt, and Payment. These tables work together by having the Student give an order of wanted items and the order goes off on its own to get the task done. The order is then referenced to make a receipt for the customer recording and showing the items, cost, and quantity the customer bought when and with what form of payment.
- Inventory, Employee, Receipt, and Order. These tables work together after receiving the order in which it gets assigned to an employee who then must procure items from inventory to produce what the student has asked for and handle the receipt creation with the order and give the correct order with the correct receipt.

### 3.0 Data Dictionary

<b><u>Table Name</u></b>	<b><u>Principal Name</u></b>	<b><u>Data Type</u></b>	<b><u>Key</u></b>	<b><u>Notes</u></b>
Student	Student_ID	Int(8)	PK	Student's ID
	Student_FName	Char(25)		Student First Name
	Student_LName	Char(25)		Student Last Name
	Student_EMail	VarChar(35)		Student Email
	Student_Password	VarChar(35)		Student Password
	Student_PhNumber	Int(10)		Student Phone Number
	Student_RHBAmount	Int(4)		RiverHawk Bucks Amount
	Student_Address	VarChar(250)		Student Address
	Student_Zip	Int(5)		Student Zip Code
	Receipt_ID	Int(10)	FK	Receipt_ID is of Receipt
	Payment_ID	VarChar(6)	FK	Payment_ID is of Payment
	Order_ID	Int(10)	FK	Order_ID is of Order
Order	Order_ID	Int(10)	PK	Order's ID
	Order_Date	Date		Order Date
	Item_Name	Char(50)		Item_Name is of Inventory
	Item_Quantity	int(3)		Item_Quantity is of Inventory
	Item_Cost	Int(5)		Item_Cost is of Inventory
	Student_ID	Int(8)	FK	Student_Id is of Student
	Receipt_ID	Int(10)	FK	Receipt_ID is of Receipt
	Employee_ID	Int(8)	FK	Employee_ID is of Employee
	Item_ID	Int(10)	FK	Item_ID is of Inventory
	Payment_ID	VarChar(6)	FK	Payment_ID is of Payment
Payment	Payment_ID	VarChar(6)	PK	Payment's ID
	Student_ID	Int(8)	FK	Student_ID is of Student
	Receipt_ID	Int(10)	FK	Receipt_ID is of Receipt
	Order_ID	Int(10)	FK	Order_ID is of Order

	Payment_Date	Date		Payment Date
	Student_FName	Char(25)		Student First Name
	Student_LName	Char(25)		Student Last Name
	Student_Email	VarChar(35)		Student Email
	Student_Address	VarChar(250)		Student Address
	Student_Zip	Int(5)		Student Zip Code
	Card_Type	Char(10)		Card Type (Visa,AMEX,Master)
	CC_Number	Int(16)		Credit Card Number
	CC_CVV	Int(4)		Credit Card CVV
	CC_Expiration	Date		Credit Card Expiration
Receipt	Receipt_ID	Int(10)	PK	Receipt's ID
	Receipt_Date	Date		Receipt's Date
	Order_Date	Date		Order_Date is of Order
	Item_Quantity	int(3)		Item_Quantity is of Inventory
	Item_Cost	Int(5)		Item_Cost is of Inventory
	Item_Name	Char(50)		Item_Name is of Inventory
	CC_Number	Int(16)		CC_Number is of Payment
	Employee_FName	Char(25)		Employee_FName is of Employee
	Employee_LName	Char(25)		Employee_LName is of Employee
	Order_ID	Int(10)	FK	Order_ID is of Order
	Payment_ID	VarChar(6)	FK	Payment_ID is of Payment
	Student_ID	Int(8)	FK	Student_ID is of Student
	Employee_ID	Int(8)	FK	Employee_ID is of Employee
Inventory	Item_ID	Int(10)	FK	Item's ID
	Item_Quantity	int(3)		Item Amount
	Item_Cost	Int(5)		Item Cost
	Item_Name	Char(50)		Item Name

	Employee_ID	Int(8)	FK	Employee_ID is of Employee
	Receipt_ID	Int(10)	FK	Receipt_ID is of Receipt
	Order_ID	Int(10)	FK	Order_ID is of Order
Employee	Employee_ID	Int(8)	FK	Employee_ID is of Employee
	Employee_FName	Char(25)		Employee First Name
	Employee_LName	Char(25)		Employee Last Name
	Employee_EMail	VarChar(35)		Employee Email
	Item_ID	Int(10)	FK	Item_ID is of Inventory
	Receipt_ID	Int(10)	FK	Receipt_ID is of Receipt
	Order_ID	Int(10)	FK	Order_ID is of Order

<b><u>Name</u></b>	<b><u>Data Type</u></b>	<b><u>Domain/Mask</u></b>
Student_ID	Int(8)	0-9
Student_FName	Char(25)	A-Z
Student_LName	Char(25)	A-Z
Student_EMail	VarChar(35) )	A-Z, 0-9, Symbols
Student_Password	VarChar(35) )	A-Z, 0-9
Student_PhNumbe r	Int(10)	0-9
Student_RHBAmo unt	Int(4)	0-9
Student_Address	VarChar(25 0)	A-Z, 0-9, Symbols
Student_Zip	Int(5)	0-9
Order_ID	Int(10)	0-9
Order_Date	Date	MM/DD/YYYY
Item_Name	Char(50)	A-Z
Item_Quantity	int(3)	0-9

Item_Cost	Int(5)	0-9
Item_ID	Int(10)	0-9
Payment_ID	VarChar(6)	A-Z, 0-9
Payment_Date	Date	MM/DD/YYYY
Card_Type	Char(10)	A-Z
CC_Number	Int(16)	0-9
CC_CVV	Int(4)	0-9
CC_Expiration	Date	MM/DD/YYYY
Receipt_ID	Int(10)	0-9
Receipt_Date	Date	MM/DD/YYYY
Employee_ID	Int(8)	0-9
Employee_FName	Char(25)	A-Z
Employee_LName	Char(25)	A-Z
Employee_Email	VarChar(35) )	A-Z, 0-9, Symbols



## 1.0 Form and Report Design

### 1.1 Specifications - Deliverables and Outcomes (Forms)

Narrative Overview	
<b>FORM</b>	Login Page
<b>USER</b>	Student
<b>TASK</b>	Lets the Student enter in their student email and password to log into their account to order from the On-The-Go system.
<b>SYSTEM</b>	HTML
<b>ENVIRONMENT</b>	UML On-The-Go

https://Practice-UMLOTG.abulsalik.repl.co

**UML On-The-Go Login**

Student Email

Password

[Login](#)

☒ Remember me

[Forgot password?](#)


Cancel

## Testing And Usability Assessment

Consistency (1= consistent and 5 = inconsistent) : 2
Sufficiency (1 = sufficient and 5 = insufficient) : 1
Accuracy (1 = accurate and 5 = inaccurate) : 2
Usability (1 = easy to use and 5 = difficult to use) : 1

https://Practice-UMLOTG.abulsalik.repl.co

UML On-The-Go Login



UMASS  
LOWELL

Ordering

Payment Method

☐ RiverHawk Bucks

☐ [Credit Card](#)

☐ [PayPal](#)

What Would You Like To Order

Select Appitizers

☐ French Fries (\$3)

☐ Curly Fries (\$3)

☐ Sweet Potatoe Fries (\$3)

Select Entree

https://Practice-UMLOTG.abulsalik.repl.co

**Select Appitizers**

☐ French Fries (\$3)

☐ Curly Fries (\$3)

☐ Sweet Potatoe Fries (\$3)

**Select Entree**

☐ Grilled Cheese Sandwich (\$4.50)

☐ Grilled Chicken Sandwich (\$5.50)

☐ Chicken Strips (\$5.00)

☐ Chicken Patty Sandwich (\$5.50)

☐ Grilled Chicken Salad (\$7.50)

**Select Drink**

☐ Dasani Water (\$1.50)

☐ Fruit Punch Gatorade (\$1.50)

☐ Cool Blue Gatorade (\$1.50)

☐ Pepsi (\$2.25)

**Select Dining Hall**

☐ Fox Dining Commons

☐ South Campus Dining Commons

☐ Inn & Conference Center

Submit

Narrative Overview	
<b>FORM</b>	Order/Selection Page
<b>USER</b>	Student
<b>TASK</b>	Lets the Student choose what payment method they will be using, the food they wish to order, and which dining hall they would like to pick up their food at

<b>SYSTEM</b>	HTML
<b>ENVIRONMENT</b>	UML On-The-Go

<b>Testing And Usability Assessment</b>	
Consistency (1= consistent and 5 = inconsistent) :	1
Sufficiency (1 = sufficient and 5 = insufficient) :	2
Accuracy (1 = accurate and 5 = inaccurate) :	2
Usability (1 = easy to use and 5 = difficult to use) :	1

<https://Practice-UMLOTG.abulsalik.repl.co>

## Please Enter Credit Card Information

We accept:  
 -Visa  
 -MasterCard  
 -American Express

Please Fill in the information, our systems will save it for this transaction and return you to the previous page upon clicking submit.

Credit Card Number:

1234-5678-9876-5432

Expiration Date:

11/11

CVV Code:

123

Card Owner:

John Doe

Address:

123 Obvious St.

Zip Code:

01234

City:

Big Brainsville

State:

AL ▼

Submit

Narrative Overview	
FORM	Credit Card Information

<b>USER</b>	Student
<b>TASK</b>	Lets the Student enter in their credit card information if they selected the option. It will let them fill in the information and save it for that order and return them to the Order/Selection Page.
<b>SYSTEM</b>	HTML
<b>ENVIRONMENT</b>	UML On-The-Go

Testing And Usability Assessment
Consistency (1= consistent and 5 = inconsistent) : 1
Sufficiency (1 = sufficient and 5 = insufficient) : 1
Accuracy (1 = accurate and 5 = inaccurate) : 1
Usability (1 = easy to use and 5 = difficult to use) : 1

https://Practice-UMLOTG.abulsalik.repl.co

**PayPal**

Email

Enter Email

Password

Enter Password

[Login](#)

☒ Remember me

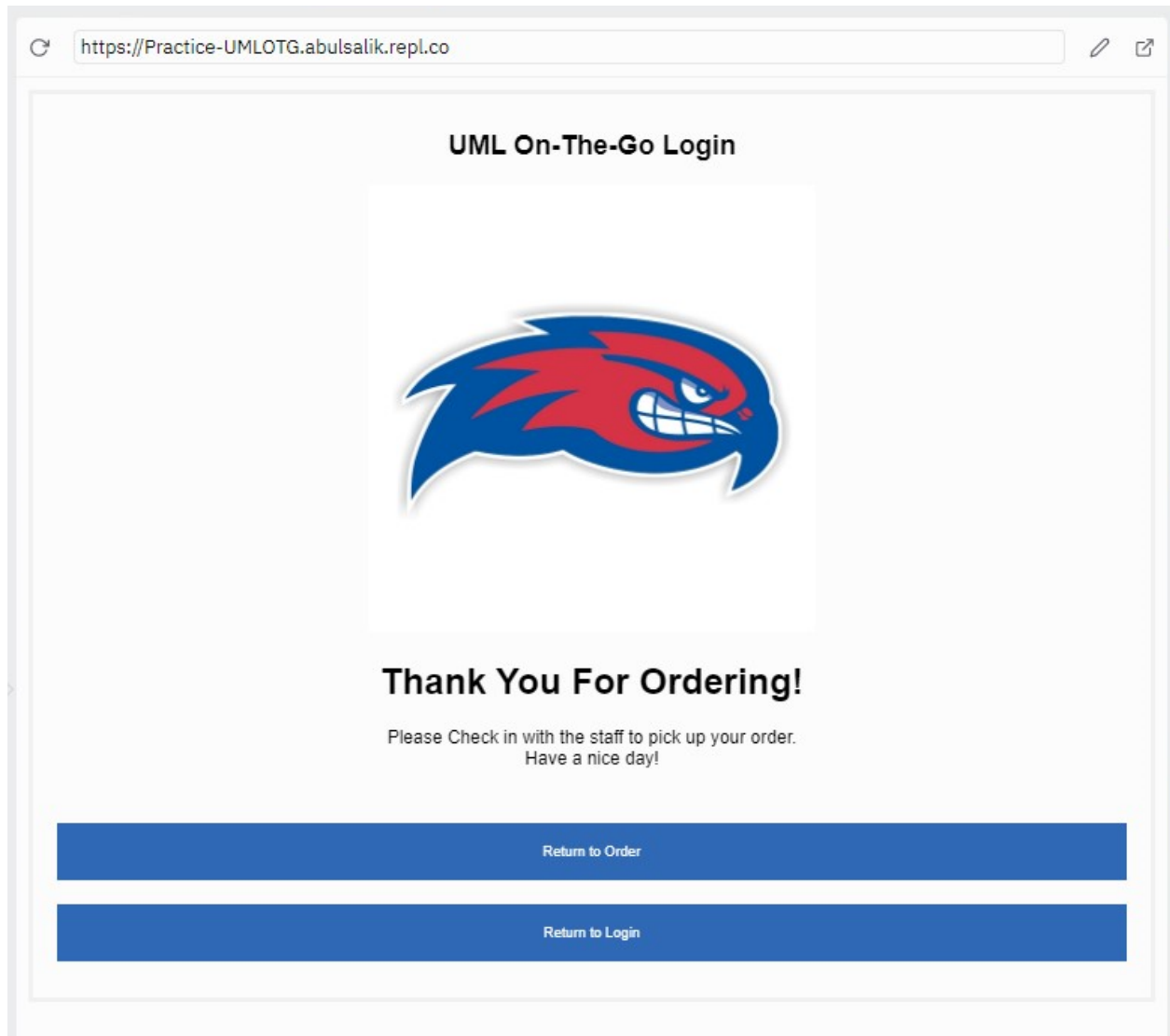
[Forgot password?](#)

Cancel

Narrative Overview	
<b>FORM</b>	PayPal
<b>USER</b>	Student
<b>TASK</b>	If the student has selected they would like to use PayPal it will redirect them to a PayPal login page and login for UML to access and charge for the purchase amount
<b>SYSTEM</b>	HTML
<b>ENVIRONMENT</b>	PayPal

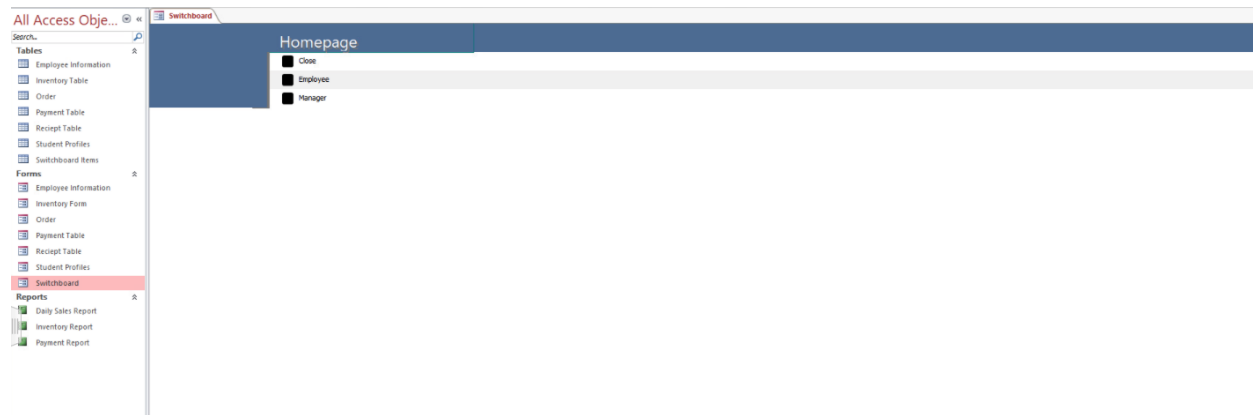
Testing And Usability Assessment
Consistency (1= consistent and 5 = inconsistent) : 1
Sufficiency (1 = sufficient and 5 = insufficient) : 2
Accuracy (1 = accurate and 5 = inaccurate) : 1
Usability (1 = easy to use and 5 = difficult to use) : 2





Narrative Overview	
<b>FORM</b>	Completed Order Page
<b>USER</b>	Student
<b>TASK</b>	Once the student has completed their order they will be shown this page and be given the choice of making a new order or returning to the login page that also logs them out of the session
<b>SYSTEM</b>	HTML
<b>ENVIRONMENT</b>	UML On-The-Go

Testing And Usability Assessment
Consistency (1= consistent and 5 = inconsistent) : 2
Sufficiency (1 = sufficient and 5 = insufficient) : 2
Accuracy (1 = accurate and 5 = inaccurate) : 1
Usability (1 = easy to use and 5 = difficult to use) : 2



Narrative Overview	
<b>FORM</b>	Homepage
<b>USER</b>	Employees and Managers
<b>TASK</b>	Employees and Managers Select the field that they are responsible for or they can close the application
<b>SYSTEM</b>	Access
<b>ENVIRONMENT</b>	UML On-The-Go

Testing And Usability Assessment
Consistency (1= consistent and 5 = inconsistent) : 1
Sufficiency (1 = sufficient and 5 = insufficient) : 1
Accuracy (1 = accurate and 5 = inaccurate) : 1
Usability (1 = easy to use and 5 = difficult to use) : 1

All Access Objects | Switchboard

Search...

**Tables**

- Employee Information
- Inventory Table
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard Items

**Forms**

- Employee Information
- Inventory Form
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard**

**Reports**

- Daily Sales Report
- Inventory Report
- Payment Report

### Employee Switchboard

- Add Student
- Edit Student
- Update Inventory
- Add On the Go Order
- Add Receipt
- Return to Homepage

All Access Objects | Switchboard | Student Profiles

Search...

**Tables**

- Employee Information
- Inventory Table
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard Items


**Forms**

- Employee Information
- Inventory Form
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard**

**Reports**

- Daily Sales Report
- Inventory Report
- Payment Report

### Student Profiles



Student\_ID

Student First Name

Student Last Name

Student Email

Student Phone Number

Student RHB Balance

Student Address

Student Zip Code

All Access Objects << Switchboard Student Profiles Inventory Table

Search...

**Tables**

- Employee Information
- Inventory Table
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard Student Profiles


**Forms**

- Employee Information
- Inventory Form
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard

**Reports**

- Daily Sales Report
- Inventory Report
- Payment Report

## Inventory



Item\_ID

Item Cost

Item Name

Item Quantity

All Access Objects << Switchboard Student Profiles Inventory Table Order

Search...

**Tables**

- Employee Information
- Inventory Table
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard Items


**Forms**

- Employee Information
- Inventory Form
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard

**Reports**

- Daily Sales Report
- Inventory Report
- Payment Report

## On the Go Order



Order\_ID

Student\_ID

Order Date

Item Name

Item Quantity

Item Cost

Total Cost

All Access Objects

Switchboard Student Profiles Inventory Table Order Receipt Table

Search...

**Tables**

- Employee Information
- Inventory Table
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Student Profiles


**Forms**

- Employee Information
- Inventory Form
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard

**Reports**

- Daily Sales Report
- Inventory Report
- Payment Report

## Receipt



Order\_ID

Order Date

Item Name

Item Quantity

Item Cost

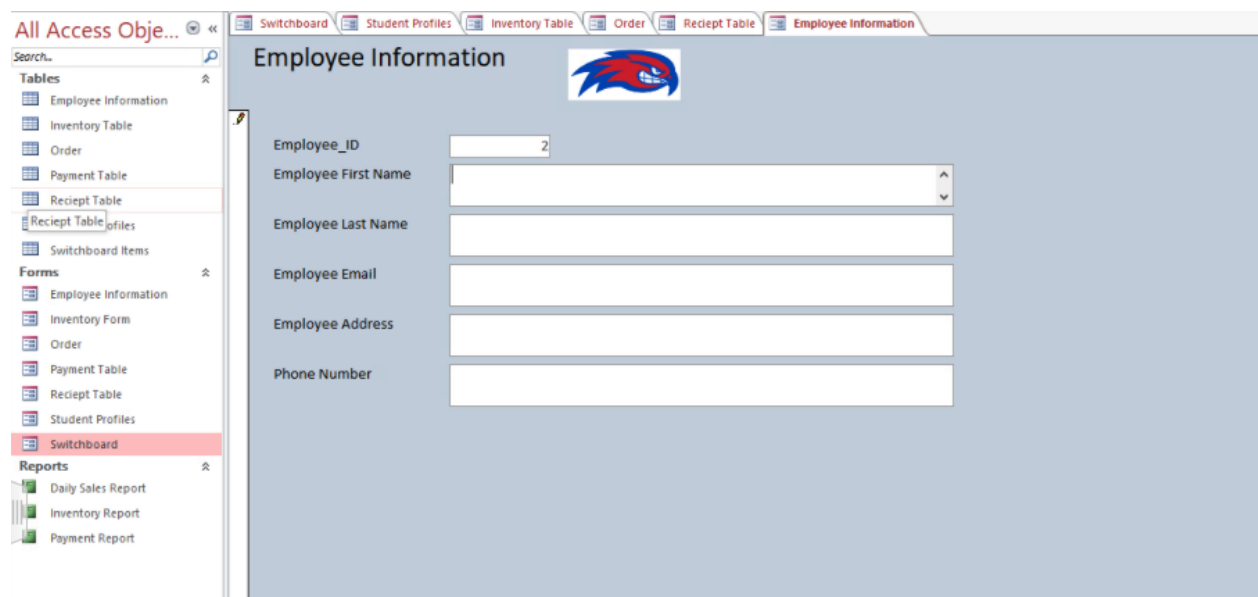
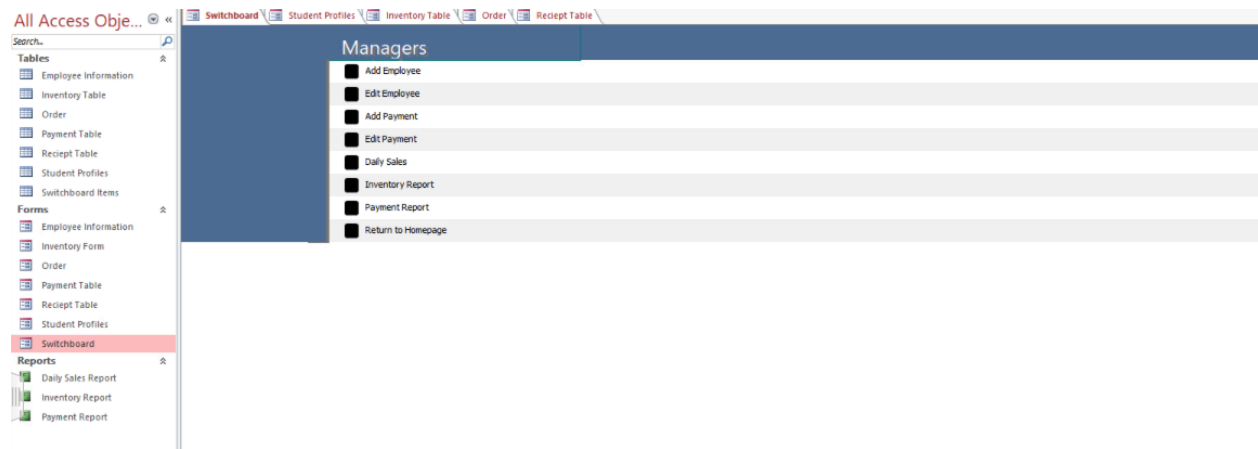
Total Cost

Employee First Name

Employee Last Name

Narrative Overview	
<b>FORM</b>	Access for Employees into the Database
<b>USER</b>	Employees
<b>TASK</b>	Employees have access to the database and are able to add and edit student records, update the inventory, add on the go orders, and add receipts
<b>SYSTEM</b>	Access
<b>ENVIRONMENT</b>	UML On-The-Go

Testing And Usability Assessment
Consistency (1= consistent and 5 = inconsistent) : 2
Sufficiency (1 = sufficient and 5 = insufficient) : 2
Accuracy (1 = accurate and 5 = inaccurate) : 3
Usability (1 = easy to use and 5 = difficult to use) : 1



All Access Objects

Search...

Tables

- Employee Information
- Inventory Table
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard Items

Forms

- Employee Information
- Inventory Form
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard

Reports

- Daily Sales Report
- Inventory Report
- Payment Report

Payments

Payment\_ID

Student\_ID

Student First Name

Student Last Name

Payment Date

Riverhawk Bucks \$p.00

Narrative Overview	
<b>FORM</b>	Access for Managers into the Database
<b>USER</b>	Managers
<b>TASK</b>	Managers have access to the database and are able to add and edit Employee Records, Add and update Payments along with access the Reports
<b>SYSTEM</b>	Access
<b>ENVIRONMENT</b>	UML On-The-Go

Testing And Usability Assessment
Consistency (1= consistent and 5 = inconsistent) : 2
Sufficiency (1 = sufficient and 5 = insufficient) : 1
Accuracy (1 = accurate and 5 = inaccurate) : 2
Usability (1 = easy to use and 5 = difficult to use) : 2

## 1.2 Specification - Deliverables and Outcomes (Reports)

All Access Objects << Switchboard Student Profiles Inventory Table **Inventory Table**

Search...

**Tables**

- Employee Information
- Inventory Table
- Order
- Inventory Table
- Receipt Table
- Student Profiles
- Switchboard Items

**Forms**

- Employee Information
- Inventory Form
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard**

**Reports**

- Daily Sales Report
- Inventory Report
- Payment Report

### Inventory Table

Item_ID	Item Cost	Item Name	Item Quantity
01	\$10.00	Frying Pans	89
02	\$3.00	Plastic Forks and Spoons	1000
03	\$5.00	Eggs	25
04	\$2.00	12oz soda	150

Wednesday, December 1, 2021 Page 1 of 1

All Access Objects << Switchboard Student Profiles Inventory Table Inventory Table **Payment Table**

Search...

**Tables**

- Employee Information
- Inventory Table
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard Items

**Forms**

- Employee Information
- Inventory Form
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard**

**Reports**

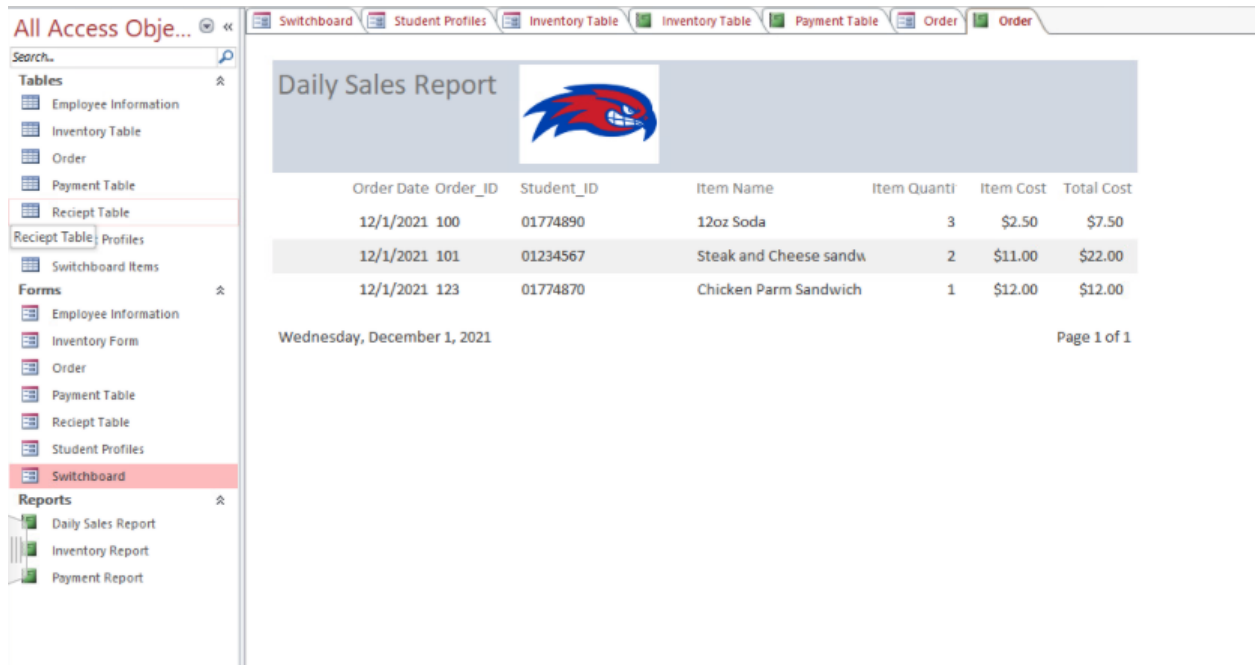
- Daily Sales Report
- Inventory Report
- Payment Report

### Payment Table

Payment Date	Payment_ID	Student_ID	Student First Name	Student Last Name	wk Bucks
					\$0.00
12/1/2021	01	02908972	Adam	Hamel	\$12.00
12/1/2021	02	09873546	Adam	Hamel	\$10.00
12/1/2021	03	01774870	Thomas	Condon	\$15.00
12/1/2021	12	01774870	Thomas	Condon	\$12.00

Wednesday, December 1, 2021 Page 1 of 1





Narrative Overview	
<b>FORM</b>	Daily Sales, Inventory, and Payment Reports
<b>USER</b>	Managers
<b>TASK</b>	Allows Managers to look at the current reports and make business decisions based on the data
<b>SYSTEM</b>	Access
<b>ENVIRONMENT</b>	UML On-The-Go

Testing And Usability Assessment
Consistency (1= consistent and 5 = inconsistent) : 2
Sufficiency (1 = sufficient and 5 = insufficient) : 1
Accuracy (1 = accurate and 5 = inaccurate) : 3
Usability (1 = easy to use and 5 = difficult to use) : 1

## **2.0 Interface and Dialog Design**

### **2.1 Specification-Deliverables and Outcomes**

#### **2.1.1 Website**

##### **Native Overview**

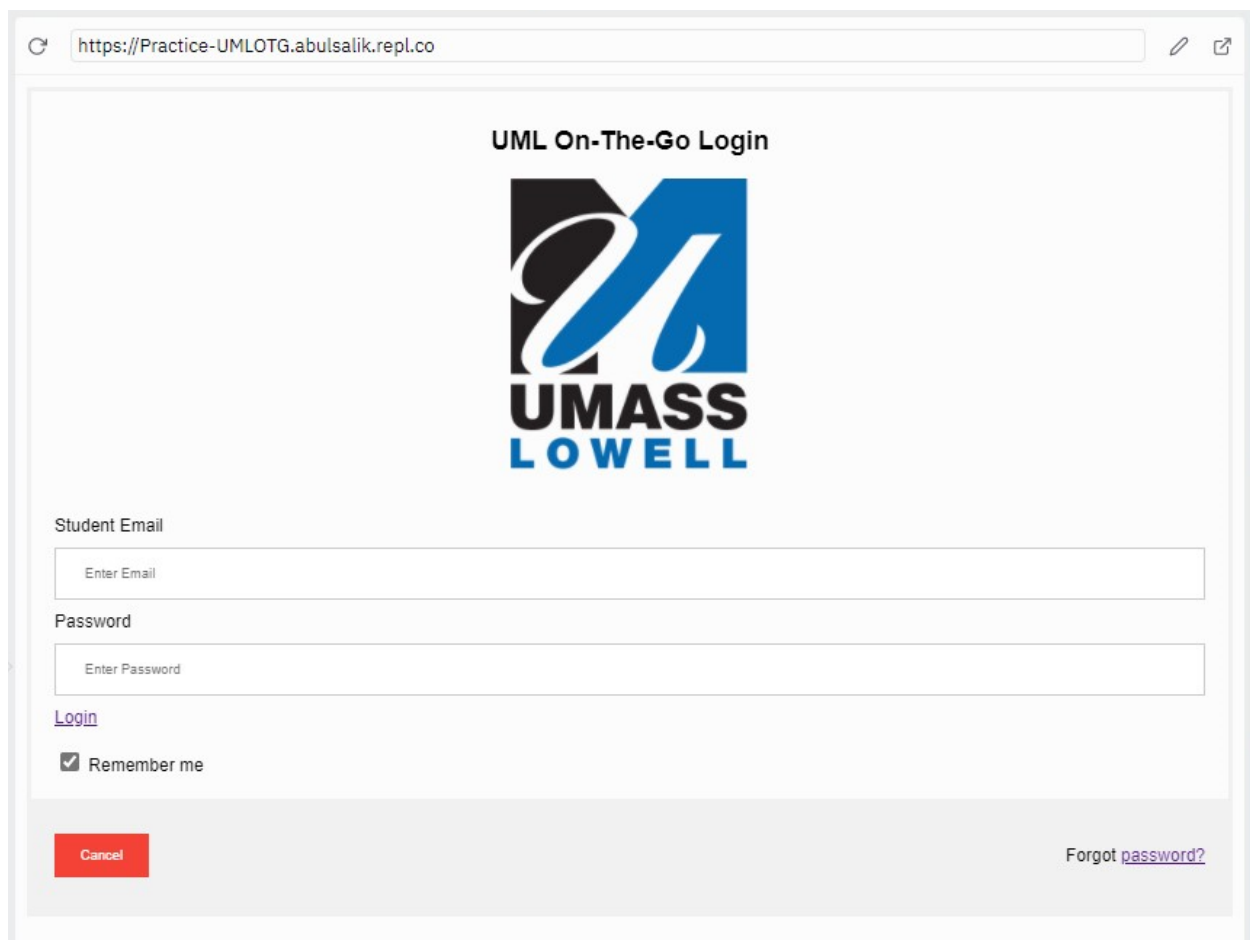
##### **Interface/Dialogue: Login**

**User Characteristics:** Student with access to an internet capable device

**Task Characteristics:** Makes Users Log into their account to access the menu and other options.

**System Characteristics:** Needs an internet application to run

**Environmental Characteristics:** Device with internet capabilities



The screenshot shows a web browser window with the URL <https://Practice-UMLOTG.abulsalik.repl.co>. The page is titled "UML On-The-Go Login" and features the UMASS Lowell logo, which consists of a stylized blue and white 'U' above the text "UMASS LOWELL". Below the logo, there are two input fields: "Student Email" with a placeholder "Enter Email" and "Password" with a placeholder "Enter Password". A "Login" link is positioned below the password field. A checkbox labeled "Remember me" is checked. At the bottom left, there is a red "Cancel" button. At the bottom right, there is a link for "Forgot password?".

##### **Testing and Usability Assessment**

**Testing Objective:** To let the user securely log into their account

**Testing Procedure:** Enter in a student email and password and 'login'

**Testing Result:** 0 Errors

**Native Overview**

**Interface/Dialogue: Menu**

**User Characteristics: Student with access to an internet capable device**


**Task Characteristics: Allows students to select a payment option, their pick-up location, and the menu items they wish to purchase in quantity.**

**System Characteristics: Needs an internet application to run**

**Environmental Characteristics: Device with internet capabilities**

https://Practice-UMLOTG.abulsalik.repl.co

UML On-The-Go Login



Ordering

Payment Method

☐ RiverHawk Bucks

☐ [Credit Card](#)

☐ [PayPal](#)

What Would You Like To Order

Select Appitizers

☐ French Fries (\$3)

☐ Curly Fries (\$3)

☐ Sweet Potatoe Fries (\$3)

Select Entree

### Testing and Usability Assessment

**Testing Objective:** Select a payment option, items and quantities up to 3 max, and pick up location

**Testing Procedure:** Click on payment options, food items and quantities, and location and click submit

**Testing Result:** 0 Errors

**Native Overview**

**Interface/Dialogue:** Credit Card

**User Characteristics:** Student with access to an internet capable device

**Task Characteristics:** Students would enter in their credit or debit card information to purchase food with.

**System Characteristics:** Needs an internet application to run

**Environmental Characteristics:** Device with internet capabilities

https://Practice-UMLOTG.abulsalik.repl.co

## Please Enter Credit Card Information

We accept:  
-Visa  
-MasterCard  
-American Express

Please Fill in the information, our systems will save it for this transaction and return you to the previous page upon clicking submit.

Credit Card Number:  
1234-5678-9876-5432

Expiration Date:  
11/11

CVV Code:  
123

Card Owner:  
John Doe

Address:  
123 Obvious St.

Zip Code:  
01234

City:  
Big Brainsville

State:  
AL ▼

Submit

### Testing and Usability Assessment

**Testing Objective:** Let students enter in their credit card information and let it be used for the current purchase they are making

**Testing Procedure:** Enter in credit card info and have it be used for their current transaction

**Testing Result:** 0 Errors

## Native Overview

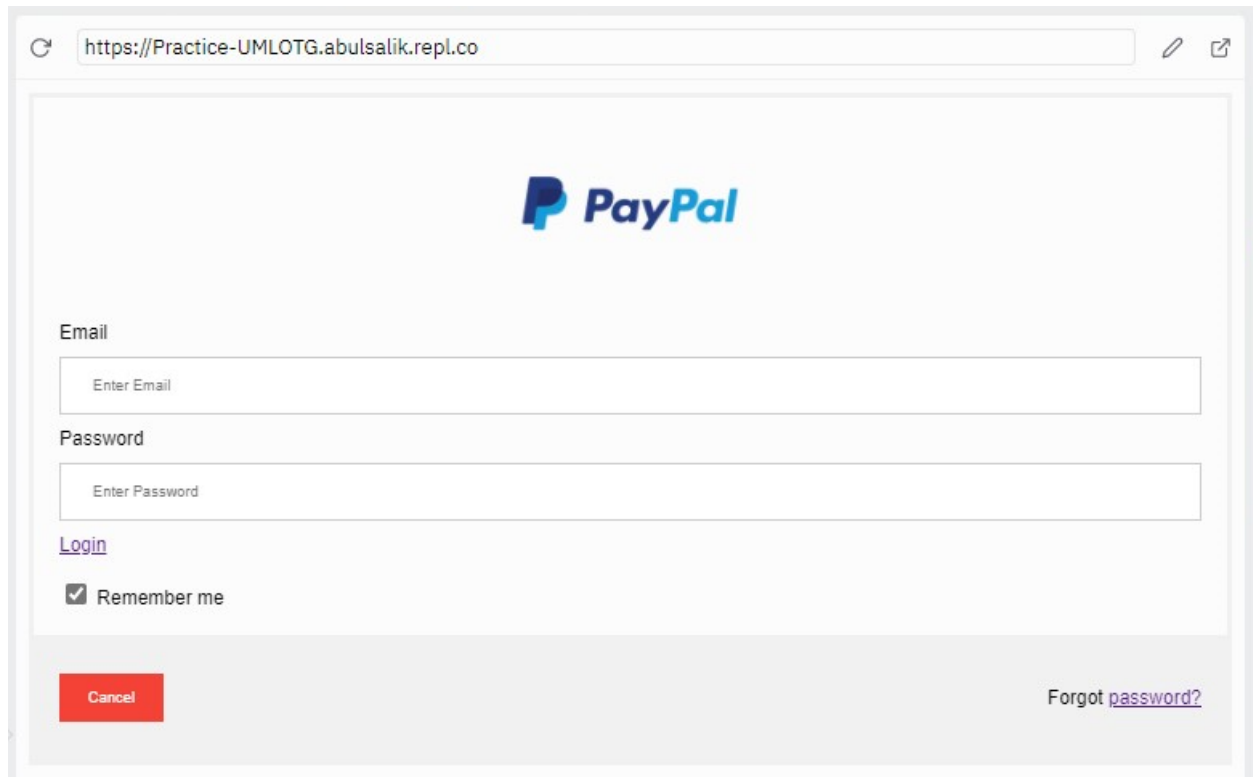
**Interface/Dialogue:** PayPal

**User Characteristics:** Student with access to an internet capable device

**Task Characteristics:** Students would log in to their PayPal Account for UML to access and charge for their purchase

**System Characteristics:** Needs an internet application to run

**Environmental Characteristics:** Device with internet capabilities

A screenshot of a web browser displaying the PayPal login page. The browser's address bar shows the URL "https://Practice-UML0TG.abulsalik.repl.co". The page features the PayPal logo at the top center. Below the logo, there are two input fields: "Email" with a placeholder "Enter Email" and "Password" with a placeholder "Enter Password". A "Login" link is positioned below the password field. A checkbox labeled "Remember me" is checked. At the bottom left, there is a red "Cancel" button. At the bottom right, there is a link that says "Forgot password?".

## Testing and Usability Assessment

**Testing Objective:** If student selects PayPal let them log in and allow UML access to charge their PayPal account

**Testing Procedure:** log into to PayPal

**Testing Result:** 0 Errors

## Native Overview

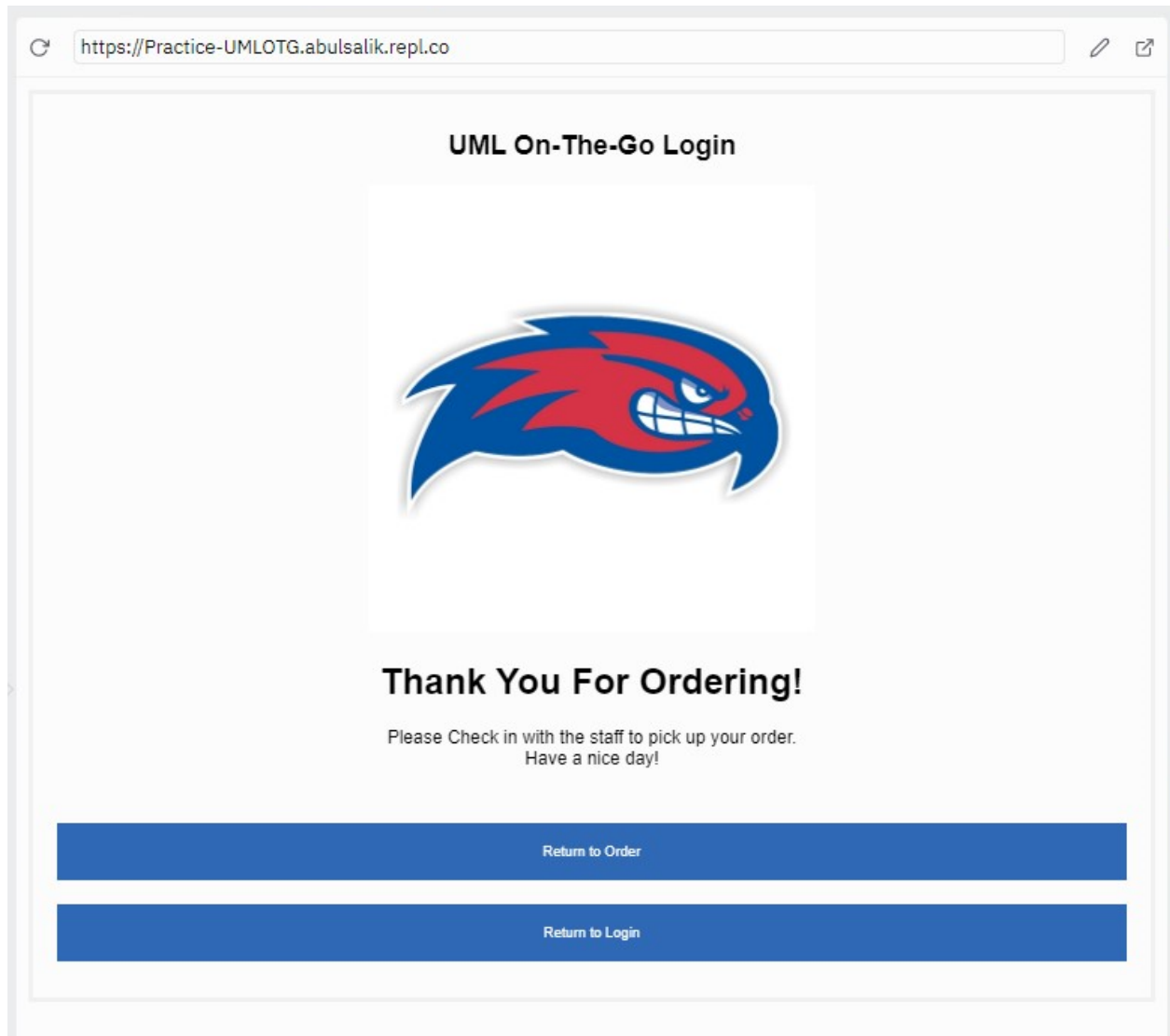
**Interface/Dialogue:** Order Complete

**User Characteristics:** Student with access to an internet capable device

**Task Characteristics:** Informs the Student of their completed Order and lets them choose to start another order or be logged out and return to the login page

**System Characteristics:** Needs an internet application to run

**Environmental Characteristics:** Device with internet capabilities



### **Testing and Usability Assessment**

**Testing Objective:** Make sure the buttons return the user to where they wish to proceed next, making a new order or logout and be returned to the login page

**Testing Procedure:** Click on either the new order button or the log out and return to login button

**Testing Result:** 0 Errors

### **2.1.2 Database**

#### **Native Overview**

**Interface/Dialogue:** Homepage

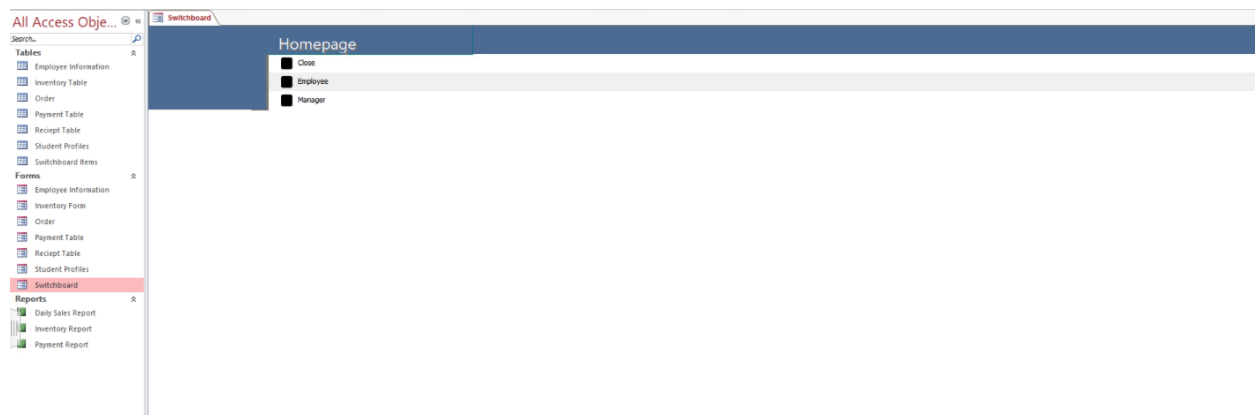
**User Characteristics:** Employees and Managers with Microsoft Access

**Task Characteristics:** Allows Employees and Managers to enter the Database



**System Characteristics: Needs Microsoft Access to run**

**Environmental Characteristics: Device with Microsoft Access installed**



**Testing and Usability Assessment**

**Testing Objective: User is brought to the Manager or Employee page depending on their title**

**Testing Procedure: select items to see if they bring us to the right page**

**Testing Result: 0 errors**

**Native Overview**

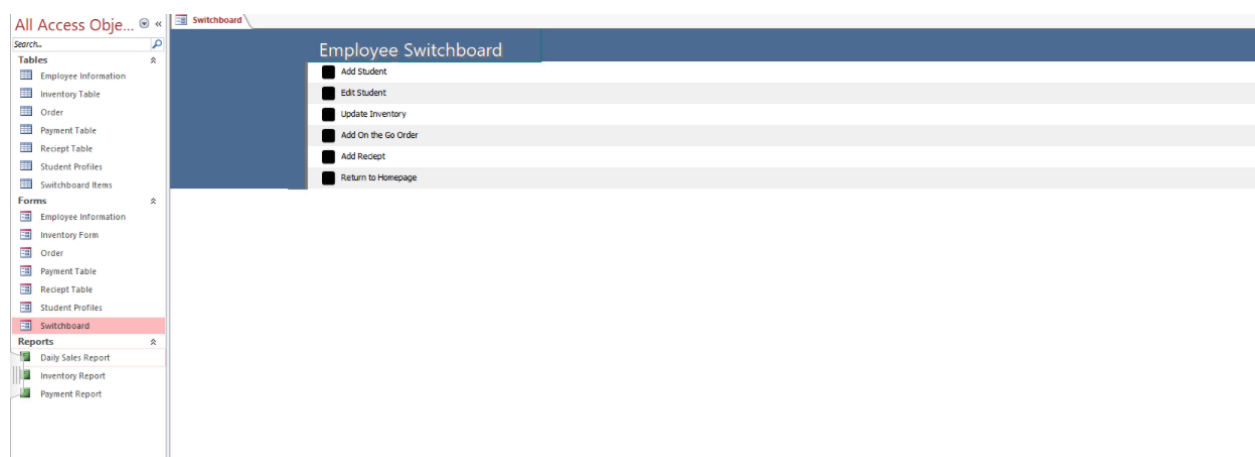
**Interface/Dialogue: Employee Page**

**User Characteristics: Employees and Managers with Microsoft Access**

**Task Characteristics: Allows Employees and Managers to enter the Database**

**System Characteristics: Needs Microsoft Access to run**

**Environmental Characteristics: Device with Microsoft Access installed**



**Testing and Usability Assessment**

**Testing Objective: User is brought to the right form that they select**

**Testing Procedure:** select items to see if they bring us to the right form

**Testing Result:** 0 errors

**Native Overview**

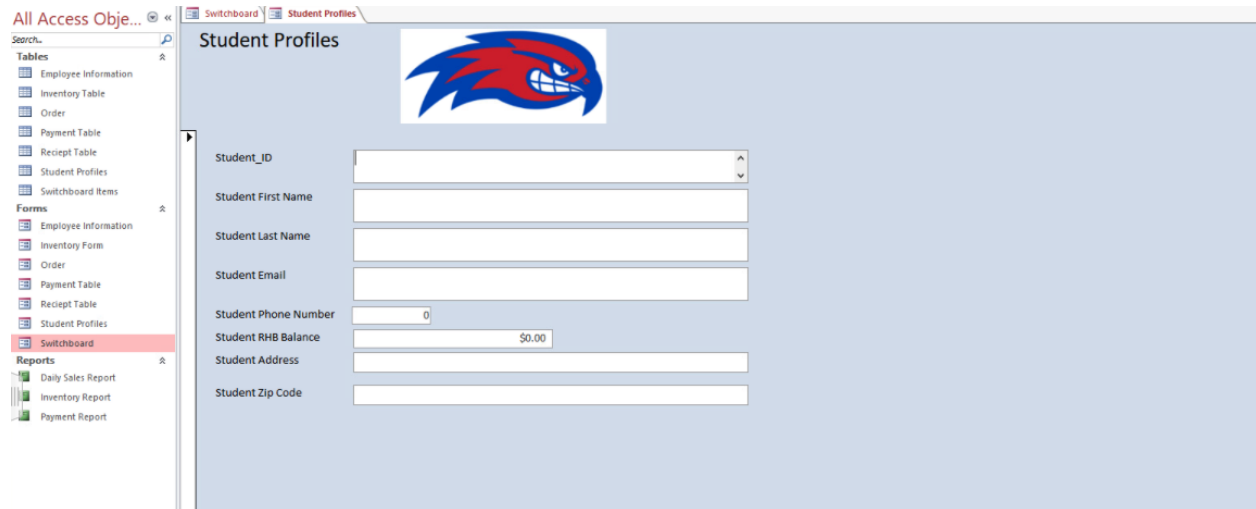
**Interface/Dialogue:** Add/Edit Students

**User Characteristics:** Employees and Managers with Microsoft Access

**Task Characteristics:** Allows Employees and Managers to input and edit the data

**System Characteristics:** Needs Microsoft Access to run

**Environmental Characteristics:** Device with Microsoft Access installed



**Testing and Usability Assessment**

**Testing Objective:** To make sure the user is brought to add student

**Testing Procedure:** select items to test if the fields are filled in correctly

**Testing Result:** 0 errors

**Native Overview**

**Interface/Dialogue:** Update Inventory

**User Characteristics:** Employees and Managers with Microsoft Access

**Task Characteristics:** Allows Employees and Managers to input and edit the data

**System Characteristics:** Needs Microsoft Access to run

**Environmental Characteristics:** Device with Microsoft Access installed

## Testing and Usability Assessment

**Testing Objective:** To make sure the user is brought to Update Inventory Form

**Testing Procedure:** select items to test if the fields are filled in correctly

**Testing Result:** 0 errors

## Native Overview

**Interface/Dialogue:** Add on the Go Orders

**User Characteristics:** Employees and Managers with Microsoft Access

**Task Characteristics:** Allows Employees and Managers to input and edit the data

**System Characteristics:** Needs Microsoft Access to run

**Environmental Characteristics:** Device with Microsoft Access installed

## Testing and Usability Assessment

**Testing Objective:** To make sure the user is brought to On the Go Order Form

**Testing Procedure:** select items to test if the fields are filled in correctly

**Testing Result:** 0 errors

## Native Overview

**Interface/Dialogue:** Add Receipt

**User Characteristics:** Employees and Managers with Microsoft Access

**Task Characteristics:** Allows Employees and Managers to input and edit the data

**System Characteristics:** Needs Microsoft Access to run

**Environmental Characteristics:** Device with Microsoft Access installed

## Testing and Usability Assessment

**Testing Objective:** To make sure the user is brought to the Receipt Form

**Testing Procedure:** select items to test if the fields are filled in correctly

**Testing Result:** 0 errors

## Native Overview

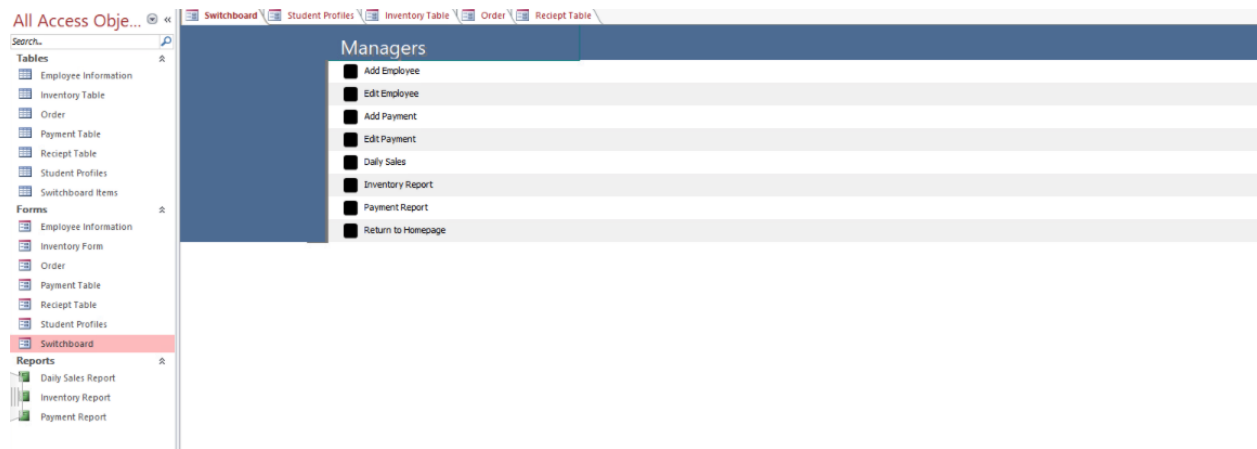
**Interface/Dialogue:** Manager Page

**User Characteristics:** Managers with Microsoft Access

**Task Characteristics:** Allows Managers to enter the Database

**System Characteristics:** Needs Microsoft Access to run

**Environmental Characteristics:** Device with Microsoft Access installed



## Testing and Usability Assessment

**Testing Objective:** User is brought to the right form when selected

**Testing Procedure:** select items to test if they are brought to the right form

**Testing Result:** 0 errors

## Native Overview

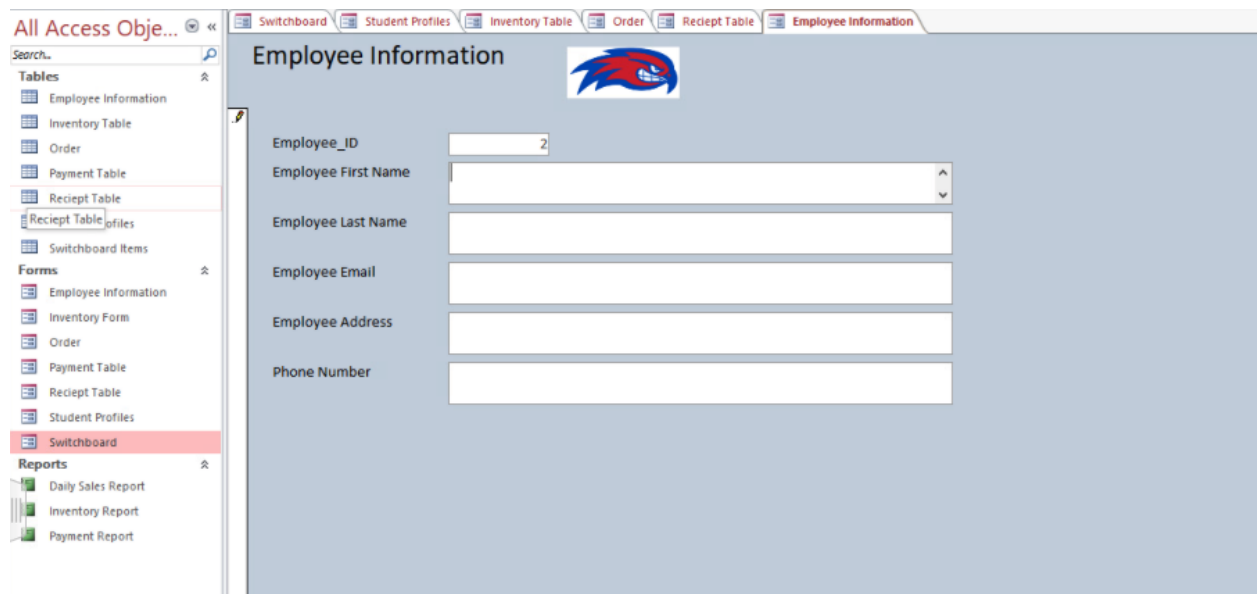
**Interface/Dialogue:** Add/edit User Employee

**Characteristics:** Managers with Microsoft Access

**Task Characteristics:** Managers to input and edit the data

**System Characteristics:** Needs Microsoft Access to run

**Environmental Characteristics:** Device with Microsoft Access installed



## Testing and Usability Assessment

**Testing Objective:** User is brought to the Employee information form

**Testing Procedure:** select items to test if the fields are filled in correctly

**Testing Result:** 0 errors

## Native Overview

**Interface/Dialogue: Add/edit Payment**

**Characteristics: Managers with Microsoft Access**

**Task Characteristics: Managers to input and edit the data**

**System Characteristics: Needs Microsoft Access to run**

**Environmental Characteristics: Device with Microsoft Access installed**

The screenshot displays the Microsoft Access application window. The title bar reads 'All Access Objects'. The left-hand navigation pane is open, showing a tree view of the database's objects. Under the 'Forms' category, the 'Switchboard' form is selected and highlighted in red. The main window area shows the 'Payments' form, which has a light blue background and a Riverhawk logo at the top. The form contains several text boxes for data entry: 'Payment\_ID', 'Student\_ID', 'Student First Name', 'Student Last Name', 'Payment Date', and 'Riverhawk Bucks'. The 'Riverhawk Bucks' field currently displays '\$0.00'. At the top of the form, there is a ribbon with tabs for 'Switchboard', 'Student Profiles', 'Inventory Table', 'Order', 'Receipt Table', 'Employee Information', and 'Payment Table'.

## Testing and Usability Assessment

**Testing Objective: User is brought to the Payments Form**

**Testing Procedure: select items to test if the fields are filled in correctly**

**Testing Result: 0 errors**

## Native Overview

**Interface/Dialogue: Inventory Report**

**Characteristics: Managers with Microsoft Access**

**Task Characteristics: Managers to input and edit the data**

**System Characteristics: Needs Microsoft Access to run**

**Environmental Characteristics: Device with Microsoft Access installed**

All Access Objects

Switchboard Student Profiles Inventory Table Inventory Table

Search...

**Tables**

- Employee Information
- Inventory Table
- Order
- Inventory Table
- Receipt Table
- Student Profiles
- Switchboard Items

**Forms**

- Employee Information
- Inventory Form
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard

**Reports**

- Daily Sales Report
- Inventory Report
- Payment Report

### Inventory Table

Item_ID	Item Cost	Item Name	Item Quantity
01	\$10.00	Frying Pans	89
02	\$3.00	Plastic Forks and Spoons	1000
03	\$5.00	Eggs	25
04	\$2.00	12oz soda	150

Wednesday, December 1, 2021

Page 1 of 1

## Testing and Usability Assessment

**Testing Objective:** User is brought to the Inventory Report and is showing the correct fields

**Testing Procedure:** select items to test if the fields are filled in correctly

**Testing Result:** 0 errors

## Native Overview

**Interface/Dialogue:** Payment Report

**Characteristics:** Managers with Microsoft Access

**Task Characteristics:** Managers to input and edit the data

**System Characteristics:** Needs Microsoft Access to run

**Environmental Characteristics:** Device with Microsoft Access installed

All Access Objects

Switchboard Student Profiles Inventory Table Inventory Table Payment Table

Search..

**Tables**

- Employee Information
- Inventory Table
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard Items

**Forms**

- Employee Information
- Inventory Form
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard

**Reports**

- Daily Sales Report
- Inventory Report
- Payment Report

**Payment Table**

Payment Date	Payment_ID	Student_ID	Student First Name	Student Last Name	iwk Bucks
					\$0.00
12/1/2021	01	02908972	Adam	Hamel	\$12.00
12/1/2021	02	09873546	Adam	Hamel	\$10.00
12/1/2021	03	01774870	Thomas	Condon	\$15.00
12/1/2021	12	01774870	Thomas	Condon	\$12.00

Wednesday, December 1, 2021

Page 1 of 1

## Testing and Usability Assessment

**Testing Objective:** User is brought to the Payment Report and is showing the correct fields

**Testing Procedure:** select items to test if the fields are filled in correctly

**Testing Result:** 0 errors

## Native Overview

**Interface/Dialogue:** Payment Report

**Characteristics:** Managers with Microsoft Access

**Task Characteristics:** Managers to input and edit the data

**System Characteristics:** Needs Microsoft Access to run

**Environmental Characteristics:** Device with Microsoft Access installed



All Access Objects

Search...

Tables

- Employee Information
- Inventory Table
- Order
- Payment Table
- Receipt Table

Receipt Table: Profiles

- Switchboard Items

Forms


- Employee Information
- Inventory Form
- Order
- Payment Table
- Receipt Table
- Student Profiles
- Switchboard

Reports

- Daily Sales Report
- Inventory Report
- Payment Report

Switchboard Student Profiles Inventory Table Inventory Table Payment Table Order Order

### Daily Sales Report



Order Date	Order_ID	Student_ID	Item Name	Item Quanti	Item Cost	Total Cost
12/1/2021	100	01774890	12oz Soda	3	\$2.50	\$7.50
12/1/2021	101	01234567	Steak and Cheese sandw	2	\$11.00	\$22.00
12/1/2021	123	01774870	Chicken Parm Sandwich	1	\$12.00	\$12.00

Wednesday, December 1, 2021

Page 1 of 1

## Testing and Usability Assessment

**Testing Objective:** User is brought to the Daily Sales Report and is showing the correct fields

**Testing Procedure:** select items to test if the fields are filled in correctly

**Testing Result:** 0 errors

## 2.2 Guidelines

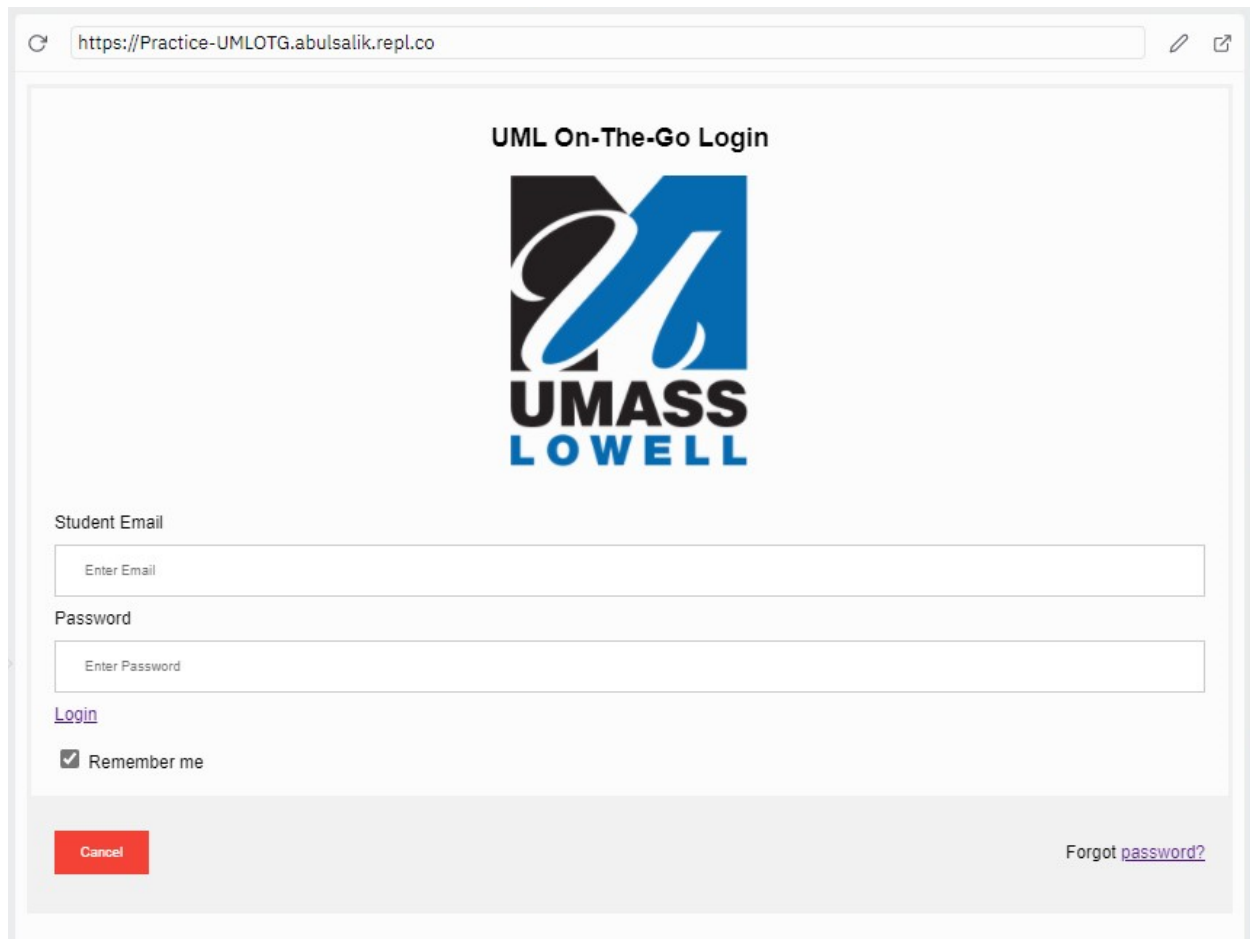
### 2.2.5 Design the Dialogue Sequence Diagram

(Needs graph thing) can be done at a later date

### **3.0 Documentations**

#### **3.1 User Manual**

Below is the step-by-step guide to use the UML On the GO Website



UML On-The-Go Login

**UMASS LOWELL**

Student Email

Enter Email

Password

Enter Password

[Login](#)

☒ Remember me


[Forgot password?](#)

Cancel

**Step 1:** The User will use an internet device to go to [www.umlonthego.com](http://www.umlonthego.com), which will bring them to the homepage of the website. This is where they will enter their username and password.

https://Practice-UMLOTG.abulsalik.repl.co

### UML On-The-Go Login



Ordering

#### Payment Method

☐ RiverHawk Bucks

☐ [Credit Card](#)

☐ [PayPal](#)

#### What Would You Like To Order

Select Appitizers

☐ French Fries (\$3)

☐ Curly Fries (\$3)

☐ Sweet Potatoe Fries (\$3)

Select Entree

**Step 2:** After the username and password has been entered, the user will be presented with various payment methods such as RiverHawk Bucks, Credit Card, and Paypal. The user will also be presented with various appetizers, entrees, and drink options and the dining hall location.

**Step 3:** If the credit card payment option is selected then the user can enter their information such as credit card number, expiration date, CVV code, and address. User can press “Submit” and go back to the payment method page

The screenshot shows a web browser window with the URL <https://Practice-UMLOTG.abulsalik.repl.co>. The page title is "Please Enter Credit Card Information". Below the title, it states "We accept: -Visa -MasterCard -American Express". A message says "Please Fill in the information, our systems will save it for this transaction and return you to the previous page upon clicking submit." The form contains the following fields:

- Credit Card Number: 1234-5678-9876-5432
- Expiration Date: 11/11
- CVV Code: 123
- Card Owner: John Doe
- Address: 123 Obvious St.
- Zip Code: 01234
- City: Big Brainsville
- State: AL (dropdown menu)

A blue "Submit" button is at the bottom of the form. A purple chat bubble icon is in the bottom right corner of the browser window.

**Step 4:** If the PayPal payment option is selected then the user can enter their email and password that's associated with their Paypal account. Users can press “Submit” and go back to the payment method page.

https://Practice-UMLOTG.abulsalik.repl.co

Email

Enter Email

Password

Enter Password

[Login](#)

☒ Remember me

[Forgot password?](#)

Cancel

**Step 5:** Once the payment method, order, location has been selected, the users can click “Submit” and be taken to a confirmation page. The user will have the options to either return to the order or return to login.

https://Practice-UMLOTG.abulsalik.repl.co

**UML On-The-Go Login**

**Thank You For Ordering!**

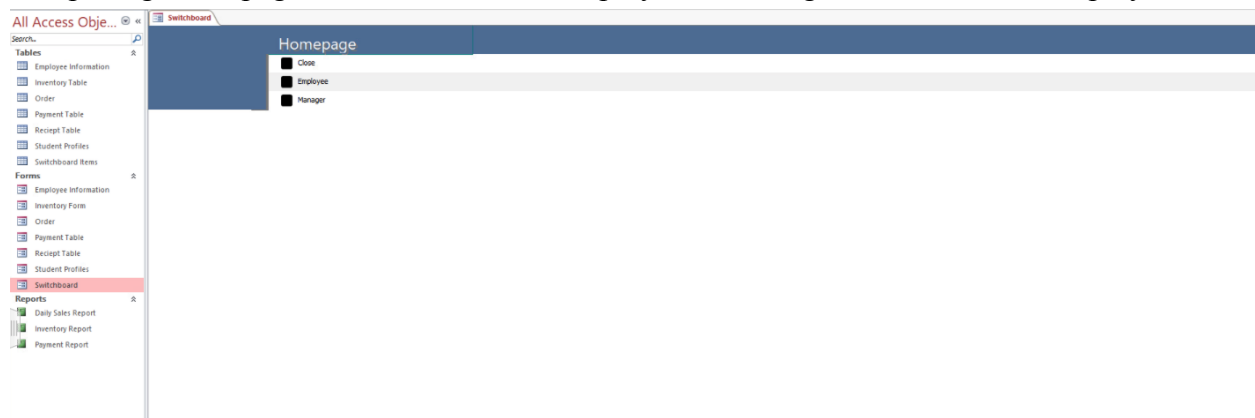
Please Check in with the staff to pick up your order.  
Have a nice day!

[Return to Order](#)

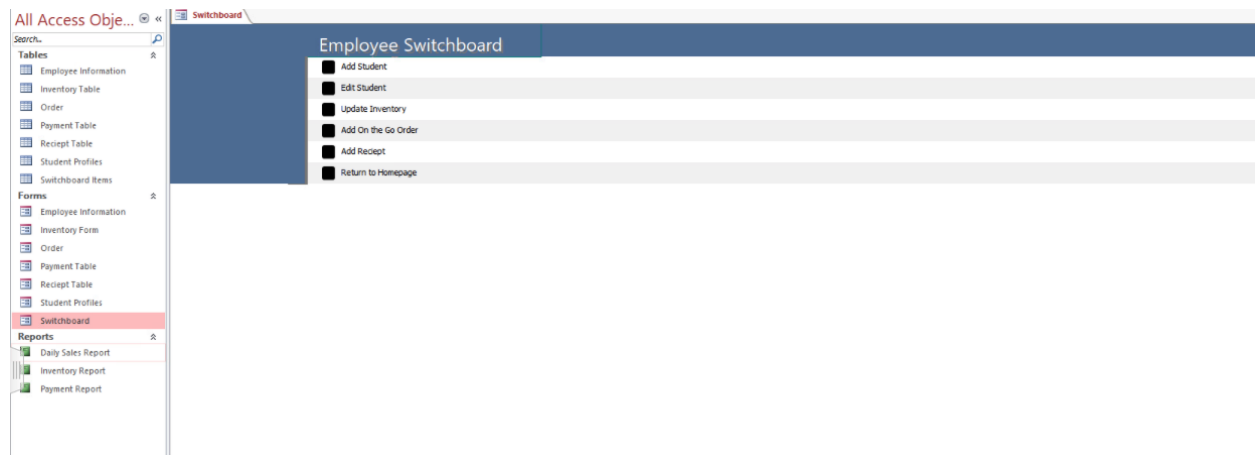
[Return to Login](#)

## Database

Step1: The user will open the UML On the Go Ordering Database. The user will be greeted to the opening Homepage with 3 selections, “Employee”, “Manager”, and “Close”. Employees



Step2: Once the User has selected their option they will be displayed on their respective screen. Employees will be given the option to Add or Edit a Student record, Update Inventory, Add on the Go Order, Add Receipt, or Return to Homepage. They will then select the option that they want. The Manager will be given the options, Add or Edit an Employee Record, Add or Edit a Payment, or Display one of the three reports (Daily Sales, Payment, or Inventory).



Step3: If the Employee Selects Add Student they will bring up the Student Profile Form and will need to fill in the fields. Once doing so they can add the student into the table. They can then save the record by clicking control and s. After that they can right click the tab for the form and select “Close”.

Step4: By selecting Edit Student the same form will show up as add Student but they can input the current information about a customer and correct any fields that may be false. They can then save the record by clicking control and s. After that they can right click the tab for the form and select “Close”.

Step5: By selecting the Update Inventory, the Employee will be displayed the form that contains the inventory items. In this form they can update the records to show the new item quantity or add another item into the inventory. They can then save the record by clicking control and s. After that they can right click the tab for the form and select “Close”.

Step6: If the Employee selects add “On the Go Order” they will be sent to the On the Go Order form. With this form the employee will fill in the respective fields that correspond to the order that the student ordered. After the order is complete the employee can save the record by clicking control and s. After that they can right click the tab for the form and select “Close”.

Step7: By selecting “Add Receipt” the receipt form will be displayed. The Employee will be prompted to fill in the fields that correspond to the correct order and receipt. After the receipt is



complete the employee can save the record by clicking control and s. After that they can right click the tab for the form and select “Close”.

The screenshot shows the 'Receipt' form in Microsoft Access. The form has a light blue background and a logo of a blue and red bird. The fields are as follows:

- Order\_ID: A dropdown menu.
- Order Date: A text box.
- Item Name: A text box.
- Item Quantity: A text box.
- Item Cost: A text box with a value of \$0.00.
- Total Cost: A text box with a value of \$0.00.
- Employee First Name: A text box.
- Employee Last Name: A text box.

The left-hand navigation pane shows a list of tables and forms. The 'Forms' section is expanded, and 'Receipt Table' is highlighted.

Step8: If the Manager selects to either Add or Edit an Employee Record they will be prompted to the Employee Information form. This form will be needed to be filled out by the Manager for new employees or an Employee who changes some of the information in the form. After the Employee record is complete the Manager can save the record by clicking control and s. After that they can right click the tab for the form and select “Close”.

The screenshot shows the 'Employee Information' form in Microsoft Access. The form has a light blue background and a logo of a blue and red bird. The fields are as follows:

- Employee\_ID: A text box with a value of 2.
- Employee First Name: A text box.
- Employee Last Name: A text box.
- Employee Email: A text box.
- Employee Address: A text box.
- Phone Number: A text box.

The left-hand navigation pane shows a list of tables and forms. The 'Forms' section is expanded, and 'Employee Information' is highlighted.

Step9: If the Manager selects to either Add or Edit a payment then they will open the Payments form. Using this form the Manager can add in a new payment for an order or make changes to an existing order. After the Employee record is complete the Manager can save the record by clicking control and s. After that they can right click the tab for the form and select “Close”.

**Payments**

Payment\_ID

Student\_ID

Student First Name

Student Last Name

Payment Date

Riverhawk Bucks

Step10: Clicking Daily Sales Report will bring up the most updated version of the daily sales report. This will show the manager the sales that were made for the day. The Manager can close the report by right clicking the tab for the report and select “Close”.

**Daily Sales Report**

Order Date	Order_ID	Student_ID	Item Name	Item Quanti	Item Cost	Total Cost
12/1/2021	100	01774890	12oz Soda	3	\$2.50	\$7.50
12/1/2021	101	01234567	Steak and Cheese sandw	2	\$11.00	\$22.00
12/1/2021	123	01774870	Chicken Parm Sandwich	1	\$12.00	\$12.00

Wednesday, December 1, 2021 Page 1 of 1

Step11: Clicking the Inventory Report button will bring up the most updated version of the Inventory Report. This shows managers the most current record of the inventory items quantity. The Manager can close the report by right clicking the tab for the report and select “Close”.

**Inventory Table**

Item_ID	Item Cost	Item Name	Item Quantity
01	\$10.00	Frying Pans	89
02	\$3.00	Plastic Forks and Spoons	1000
03	\$5.00	Eggs	25
04	\$2.00	12oz soda	150

Wednesday, December 1, 2021 Page 1 of 1

Step12: Clicking the Payments Report button will bring up the most updated version of the Inventory Report. This shows managers the most current record of the inventory items quantity. The Manager can close the report by right clicking the tab for the report and select “Close”.

**Payment Table**

Payment Date	Payment_ID	Student_ID	Student First Name	Student Last Name	wk Bucks
12/1/2021	01	02908972	Adam	Hamel	\$12.00
12/1/2021	02	09873546	Adam	Hamel	\$10.00
12/1/2021	03	01774870	Thomas	Condon	\$15.00
12/1/2021	12	01774870	Thomas	Condon	\$12.00

Wednesday, December 1, 2021 Page 1 of 1

Step13: Clicking Close in the Homepage will exit the database and send you to the menu for Microsoft Access.

### 3.2 Training Material

#### User Training Seminar Schedule

##### UML On the GO Ordering System Management Training - December 2021

Each employee and manager are required to complete a training seminar then check off the seminar on the blow agenda and provide their initials. The employees do not need to go back every session and it will be on their schedule. A report will be provided to the office at the end of each seminar.

Administrating the User Training Seminar: Andrew Feng, Ahmed Siddiqui, and Tom Condon

User Training Seminar	Date	Done? (✓)	Initials
Online Ordering System: Seminar 1 Employees and Managers 9am - 1pm “Login Screen” - Navigation on tabs on Page “Ordering Screen” - Navigation on tabs on Page <ul style="list-style-type: none"><li>• Add order - how to add order</li><li>• How to save and close reports and forms</li></ul>	December 10		
Online Ordering System: Seminar 2 Employees and Managers 9am - 1pm Continuation on “Ordering Screen” <ul style="list-style-type: none"><li>• Edit Customer - How to edit Customer information</li><li>• Confirmation screen</li></ul>	December 11		
Online Ordering System: Seminar 3 Employees and Managers 9am - 11am Login information - permission will be granted for employees to access the database	December 12		
Online Ordering System: Seminar 4 Employees and Managers 9am - 11am	December 13		
Online Ordering System: Seminar 5 Employees and Managers 9am - 1pm “Manager Screen”	December 14		

<ul style="list-style-type: none"> <li>• Add kitchen employees</li> <li>• Payment</li> <li>• Reports</li> </ul>			
Online Ordering System: Seminar 6 Employees and Managers 9am - 1pm	December 16		

### 3.3 System Documents

#### 3.3.1 Source Code

##### HTML

(Index.HTML)

```
<!DOCTYPE html>
<html>
<head>
<title>UML On-The-Go</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}
form {border: 3px solid #f1f1f1;}
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;}
button {
  background-color: #2f69b5;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;}
button:hover {
```

```

    opacity: 0.8;}
.cancelbtn {
    width: auto;
    padding: 10px 18px;
    background-color: #f44336;}
.imgcontainer {
    text-align: center;
    margin: 24px 0 12px 0;}
img.avatar {
    text-align: center;
    width: 20%;
    border-box: 30%;}
.container {
    padding: 16px;}
span.psw {
    float: right;
    padding-top: 16px;}
@media screen and (max-width: 300px) {
    span.psw {
        display: block;
        float: none;}
    .cancelbtn {
        width: 100%;}}
</style>
</head>
<body>
<form action="/action_page.php" method="post">
    <div class="imgcontainer">
        <h2>UML On-The-Go Login</h2>
        
    </div>
    <div class="container">
        <label for="uname">Student Email</label>
        <input type="text" placeholder="Enter Email" name="uname">
        <label for="psw">Password</label>
        <input type="password" placeholder="Enter Password" name="psw">
        <a style="text-align:right" href = "Page1.html">Login</a>
    <br><br>
    <label>

```

```

        <input type="checkbox" checked="checked" name="remember"> Remember me
    </label>
</div>
<div class="container" style="background-color:#f1f1f1">
    <button type="button" class="cancelbtn">Cancel</button>
    <span class="psw">Forgot <a href="#">password?</a></span>
</div>
</form>
</body>
</html>

```

### (Page1.HTML)

```

<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<style>
body {font-family: Arial, Helvetica, sans-serif;}
form {border: 3px solid #f1f1f1;}
input[type=radio], input[type=checkbox] {
    width: 10%;
    padding: 10px 50px;
    margin: 8px 0;
    display: inline-block;
    border: 5px solid #133d75;
    box-sizing: border-box;}
label {
display: inline-block;}
.imgcontainer {
    text-align: center;
    margin: 24px 0 12px 0;}
button {
    background-color: #2f69b5;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    cursor: pointer;

```

```

width: 100%;}
button:hover {
  opacity: 0.8;}
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;}
.container {
  padding: 25px;
  width: 100%;
  margin: 8px 0;
  display: inline-block;
  border: 5px solid #133d75;
  box-sizing: border-box;}
span.psw {
  float: left;
  padding-top: 16px;}
</style>
  <title>Ordering</title>
</head>
<body>
<div class="imgcontainer">
  <h2>UML On-The-Go Login</h2>
  
</div>
  <h2 style="text-align:center">Ordering</h2>
<div class="container">
  <h2 style="text-align:center;">Payment Method</h2>
<form>
  <input type="radio" id="rhbucks" name="pmethod" value="RiverHawk Bucks">
  <label for="rhbucks">RiverHawk Bucks</label><br>
  <input type="radio" id="card" name="pmethod" value="Credit Card"><a
href="Page3.html">Credit Card</a><br>
  <input type="radio" id="paypal" name="pmethod" value="PayPal"><a
href="PayPal.html">PayPal</a>
</form>
</div>
<!-- The database would be retrieved from here most likely and be used to
input that days menu items -->

```



```

<div class="container">
  <h2 style="text-align:center;">What Would You Like To Order</h2>
  <form style="text-align:left;">
    <h4>Select Appetizers</h4>
    <input type="checkbox" id="app1" name="app1" value="French Fries">
    <label for="app1" cols="1"> French Fries ($3)</label>
    <input type="number" id="quantity" placeholder="0" name="quantity"
min="0" max="3" cols="3"><br>
    <input type="checkbox" id="app2" name="app2" value="Curly Fries">
    <label for="app2"> Curly Fries ($3)</label>
    <input type="number" id="quantity" placeholder="0" name="quantity"
min="0" max="3"><br>
    <input type="checkbox" id="app3" name="app3" value="Sweet Potato
Fries">
    <label for="app3"> Sweet Potato Fries ($3) </label>
    <input type="number" id="quantity" placeholder="0" name="quantity"
min="0" max="3"><br>
  </form><form>
    <h4>Select Entree</h4>
    <input type="checkbox" id="ent1" name="ent1" value="Grilled Cheese">
    <label for="ent1"> Grilled Cheese Sandwich ($4.50)</label>
    <input type="number" id="quantity" placeholder="0" name="quantity"
min="0" max="3"><br>
    <input type="checkbox" id="ent2" name="ent2" value="Grilled Chicken
Sandwich">
    <label for="ent2"> Grilled Chicken Sandwich ($5.50)</label>
    <input type="number" id="quantity" placeholder="0" name="quantity"
min="0" max="3"><br>
    <input type="checkbox" id="ent3" name="ent3" value="Chicken Strips">
    <label for="ent3"> Chicken Strips ($5.00)</label>
    <input type="number" id="quantity" placeholder="0" name="quantity"
min="0" max="3"><br>
    <input type="checkbox" id="ent4" name="ent4" value="Chicken Patty">
    <label for="ent4"> Chicken Patty Sandwich ($5.50)</label>
    <input type="number" id="quantity" placeholder="0" name="quantity"
min="0" max="3"><br>
    <input type="checkbox" id="ent5" name="ent5" value="Grilled Chicken
Salad">
    <label for="ent5"> Grilled Chicken Salad ($7.50)</label>

```

```

        <input type="number" id="quantity" placeholder="0" name="quantity"
min="0" max="3"><br>
    </form><form>
        <h4>Select Drink</h4>
        <input type="checkbox" id="dr1" name="dr1" value="Dasani Water">
        <label for="dr1"> Dasani Water ($1.50) </label>
        <input type="number" id="quantity" placeholder="0" name="quantity"
min="0" max="3"><br>
        <input type="checkbox" id="dr2" name="dr2" value="Fruit Punch
Gatorade">
        <label for="dr2"> Fruit Punch Gatorade ($1.50) </label>
        <input type="number" id="quantity" placeholder="0" name="quantity"
min="0" max="3"><br>
        <input type="checkbox" id="dr3" name="dr3" value="Cool Blue Gatorade">
        <label for="dr3"> Cool Blue Gatorade ($1.50) </label>
        <input type="number" id="quantity" placeholder="0" name="quantity"
min="0" max="3"><br>

        <input type="checkbox" id="dr4" name="dr4" value="Pepsi">
        <label for="dr4"> Pepsi ($2.25)</label>
        <input type="number" id="quantity" placeholder="0" name="quantity"
min="0" max="3"><br>
    </form>
</div>
<div class="container">
    <form>
        <h2 style="text-align:center;">Select Dining Hall</h2>
        <input type="radio" id="wild" name="wild" value="71 Wilder">
        <label for="rhbucks">Fox Dining Commons</label><br>
        <input type="radio" id="paw" name="paw" value="100 Pawtucket">
        <label for="rhbucks">South Campus Dining Commons</label><br>
        <input type="radio" id="War" name="War" value="50 Warren">
        <label for="rhbucks">Inn & Conference Center</label><br>
    </div>
</form>
<a href="Page2.html"><button>Submit</button> </a>
</body>
</html>

```

## (Page2.HTML)

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
</style>
body {font-family: Arial, Helvetica, sans-serif;}
form {border: 3px solid #f1f1f1;}
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;}
button {
  background-color: #2f69b5;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;}
button:hover {
  opacity: 0.8;}
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;}
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;}
img.avatar {
  width: 40%;
  border-box: 50%;}
.container {
  padding: 16px;}
span.psw {
```

```

float: right;
padding-top: 16px;}
@media screen and (max-width: 300px) {
  span.psw {
    display: block;
    float: none;}
  .cancelbtn {
    width: 100%;}}
</style>
</head>
  <title>Order Complete</title>
  <form action="/action_page.php" method="post">
  <div class="imgcontainer">
    <h2>UML On-The-Go Login</h2>
    
  </div>
  <h1 style="text-align:center">Thank You For Ordering!</h1>
  <p style="text-align:center">Please Check in with the staff to pick up
your order. <br> Have a nice day!</p>

<body>
<form>
  <div class="container">
    <a href="Page1.html">
    <button type="button">Return to Order</button>
    <a href="index.html">
    <button type="button">Return to Login</button>
  </div>
</form>
</body>
</html>

```

### (Page3.HTML)

```

<!DOCTYPE html>
<html>
<head>
  <title>Credit Information</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">

```

```

<style>
body {font-family: Arial, Helvetica, sans-serif;}
form {border: 3px solid #f1f1f1;}
.container {
padding: 16px;}
span.psw {
float: right;
padding-top: 16px;}
input[type=text]{
width: 100%;
padding: 12px 20px;
margin: 8px 0;
display: inline-block;
border: 1px solid #ccc;
box-sizing: border-box;}
button {
background-color: #2f69b5;
color: white;
padding: 14px 20px;
margin: 8px 0;
border: none;
cursor: pointer;
width: 100%;}
button:hover {
opacity: 0.8;}
</style>
<h1>Please Enter Credit Card Information</h1>
<body>
    <h4>We accept:<br>-Visa<br>-MasterCard<br>-American Express</h4>
    <p>Please Fill in the information, our systems will save it for this
transaction and return you to the previous page upon clicking submit.</p>
<div class="container">
    <form action="/action_page.php">
        <label for="CNum">Credit Card Number:</label><br>
        <input type="text" id="CNum" name="CNum"
placeholder="1234-5678-9876-5432"><br><br>
        <label for="CExp">Expiration Date:    </label><br>
        <input type="text" id="CExp" name="CExp" placeholder="11/11"><br><br>
        <label for="CVV">CVV Code:</label><br>

```

```

<input type="text" id="CVV" name="CVV" placeholder="123"><br><br>
<label for="name">Card Owner:</label><br>
<input type="text" id="name" name="name" placeholder="John
Doe"><br><br>
<label for="add">Address:</label><br>
<input type="text" id="add" name="add" placeholder="123 Obvious
St."><br><br>
<label for="zip">Zip Code:</label><br>
<input type="text" id="zip" name="zip" placeholder="01234"><br><br>
<label for="city">City:</label><br>
<input type="text" id="city" name="city" placeholder="Big
Brainsville"><br><br>
<label for="stt">State:</label><br>
<select name="stt" id="stt">
  <option value="ALABAMA">AL</option>
  <option value="ALASKA">AK</option>
  <option value="ARIZONA">AZ</option>
  <option value="ARKANSAS">AR</option>
  <option value="CALIFORNIA">CA</option>
  <option value="COLORADO">AR</option>
  <option value="CONNECTICUT">CO</option>
  <option value="DELAWARE">DE</option>
  <option value="FLORIDA">FL</option>
  <option value="GEORGIA">GA</option>
  <option value="HAWAII">HI</option>
  <option value="IDAHO">ID</option>
  <option value="ILLINOIS">IL</option>
  <option value="INDIANA">IN</option>
  <option value="IOWA">IA</option>
  <option value="KANSAS">KS</option>
  <option value="KENTUCKY">KY</option>
  <option value="LOUISIANA">LA</option>
  <option value="MAINE">ME</option>
  <option value="MARYLAND">MD</option>
  <option value="MASSACHUSETTS">MA</option>
  <option value="MICHIGAN">MI</option>
  <option value="MINNESOTA">MN</option>
  <option value="MISSISSIPPI">MS</option>
  <option value="MISSOURI">MO</option>

```

```

        <option value="MONTANA">MT</option>
        <option value="NEBRASKA">NE</option>
        <option value="NEVADA">NV</option>
        <option value="NEW HAMPSHIRE">NH</option>
        <option value="NEW JERSEY">NJ</option>
        <option value="NEW MEXICO">NM</option>
        <option value="NEW YORK">NY</option>
        <option value="NORTH CAROLINA">NC</option>
        <option value="OHIO">OH</option>
        <option value="OKLAHOMA">OK</option>
        <option value="OREGON">OR</option>
        <option value="PENNSYLVANIA">PA</option>
        <option value="RHODE ISLAND">RI</option>
        <option value="SOUTH CAROLINA">SC</option>
        <option value="SOUTH DAKOTA">SD</option>
        <option value="TENNESSEE">TN</option>
        <option value="TEXAS">TX</option>
        <option value="UTAH">UT</option>
        <option value="VERMONT">VT</option>
        <option value="VIRGINIA">VA</option>
        <option value="WASHINGTON">WA</option>
        <option value="WEST VIRGINIA">WV</option>
        <option value="WISCONSIN">WI</option>
        <option value="WYOMING">WY</option>
    </select><br><br>
</div>
</form>
<a href="Page1.html"><button>Submit</button></a>
</body>
</head>
</html>

```

## PayPal.HTML

```

<!DOCTYPE html>
<html>
<head>
<title> PayPal </title>
<meta name="viewport" content="width=device-width, initial-scale=1">

```

```

<style>
body {font-family: Arial, Helvetica, sans-serif;}
form {border: 3px solid #f1f1f1;}
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;}
button {
  background-color: #2f69b5;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;}
button:hover {
  opacity: 0.8;}
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;}
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;}
img.avatar {
  text-align: center;
  width: 20%;
  border-box: 30%;}
.container {
  padding: 16px;}
span.psw {
  float: right;
  padding-top: 16px;}
@media screen and (max-width: 300px) {
  span.psw {
    display: block;

```



```

        float: none;}
    .cancelbtn {
        width: 100%;}}
</style>
</head>
<body>
<form action="/action_page.php" method="post">
    <div class="imgcontainer">
        
    </div>
    <div class="container">
        <label for="uname">Email</label>
        <input type="text" placeholder="Enter Email" name="uname">
        <label for="psw">Password</label>
        <input type="password" placeholder="Enter Password" name="psw">
        <a style="text-align:right" href = "Page1.html">Login</a>
    <br><br>
        <label>
            <input type="checkbox" checked="checked" name="remember"> Remember me
        </label>
    </div>
    <div class="container" style="background-color:#f1f1f1"><a
href="index.html">
        <button type="button" class="cancelbtn">Cancel</button></a>
        <span class="psw">Forgot <a href="#">password?</a></span>
    </div>
</form>
</body>
</html>

```

### 3.3.2 Technical Report

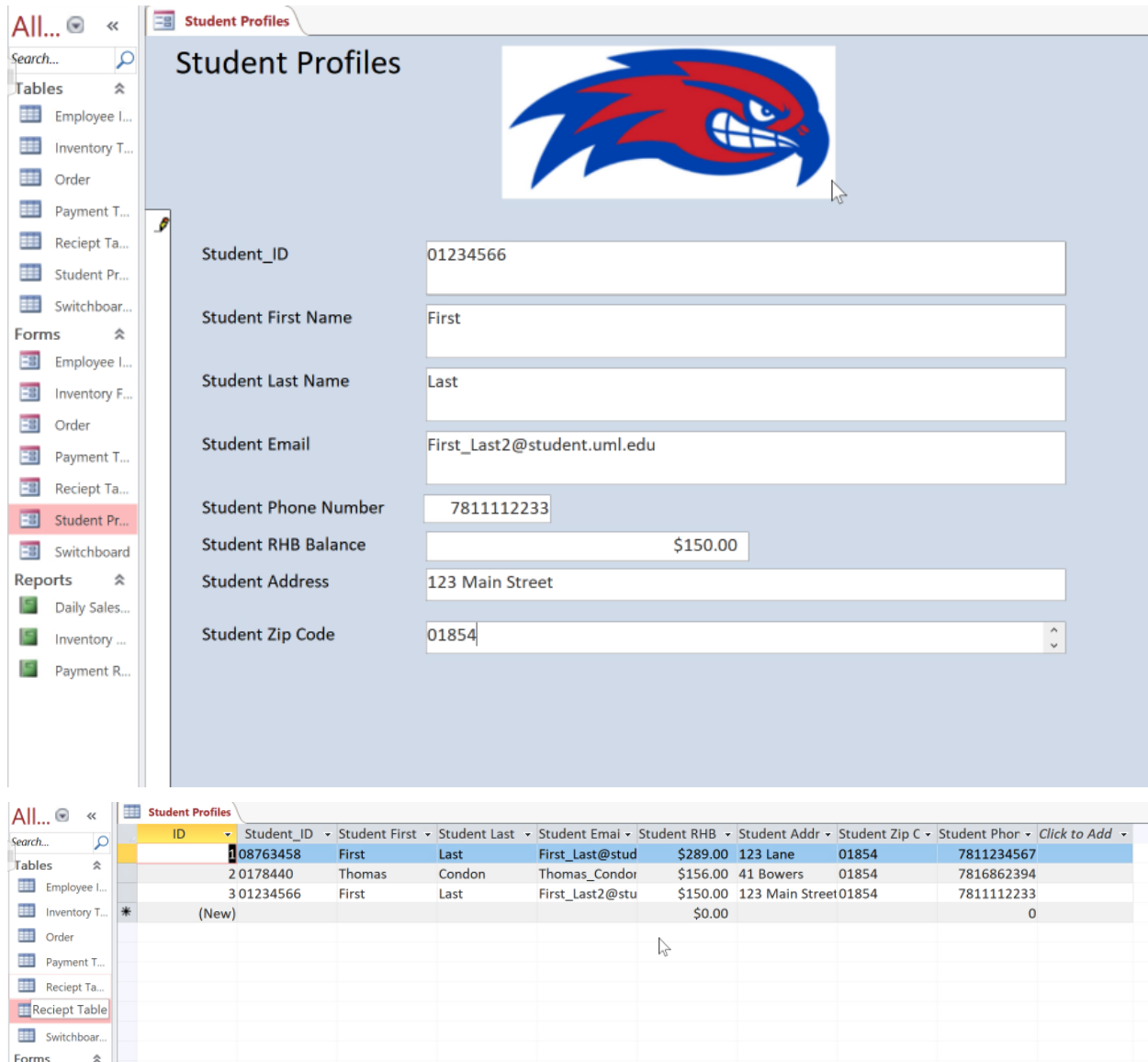
#### How We Will Address Technical Support

1. Have a troubleshoot mechanism installed upon crashes of the website database and track through logs what issues occurred when and where.
2. Update the system and software as we go and implement new technologies or sources to adapt the system to the current times or situations
3. Have maintenance done by the server side engineers to keep operations running smoothly

For a fee maintenance work can be done on the system and allow users to call in issues or submit complaints or feedback on the system for improvements

### 3.3.4 Testing Data

We tested our Database for completeness and no errors in information being created in the forms. First we have our Student form and table.



The screenshot shows a database application interface with a sidebar on the left containing a search bar, a 'Tables' list, a 'Forms' list, and a 'Reports' list. The 'Student Profiles' form is open, displaying a student's information. The form includes fields for Student\_ID, Student First Name, Student Last Name, Student Email, Student Phone Number, Student RHB Balance, Student Address, and Student Zip Code. A logo of a blue and red hawk is visible in the top right corner of the form area.

Below the form, a table view of the 'Student Profiles' table is shown. The table has columns for ID, Student\_ID, Student First, Student Last, Student Email, Student RHB, Student Addr, Student Zip C, Student Phor, and Click to Add. The table contains three rows of data, including a new entry marked with an asterisk.


ID	Student_ID	Student First	Student Last	Student Email	Student RHB	Student Addr	Student Zip C	Student Phor	Click to Add
1	08763458	First	Last	First_Last@stud	\$289.00	123 Lane	01854	7811234567	
2	0178440	Thomas	Condon	Thomas_Condor	\$156.00	41 Bowers	01854	7816862394	
3	01234566	First	Last	First_Last2@stu	\$150.00	123 Main Street	01854	7811112233	
*	(New)				\$0.00			0	

Next we tested that we could make an order with our form.

Order_ID	Student_ID	Order Date	Item Name	Item Quantity	Item Cost	Total Cost	Click to Add
123	01774870	12/1/2021	Chicken Parm Sa	1	\$12.00	\$12.00	
100	01774890	12/1/2021	12oz Soda	3	\$2.50	\$7.50	
101	01234567	12/1/2021	Steak and Chees	2	\$11.00	\$22.00	
104	01234566	12/2/2021	Chicken Cesar	1	\$9.00	\$9.00	
*				0	\$0.00	\$0.00	

91

**Receipt**



Order\_ID: 104

Order Date: 12/2/2021

Item Name: Chicken Ceasar Salad

Item Quantity: 1

Item Cost: \$9.00

Total Cost: \$9.00


Employee First Name: Tom

Employee Last Name: Condon

Order_ID	Order Date	Item Name	Item Quantity	Item Cost	Total Cost	Employee Fir	Employee Las	Click to Add
123	12/1/2021	Chicken Parm Sandwic	1	\$12.00	\$12.00	Edward	Chen	
104	12/2/2021	Chicken Ceasar Salad	1	\$9.00	\$9.00	Tom	Condon	
				\$0.00	\$0.00			

We also needed to test that users could update the inventory levels.

**Inventory**



Item\_ID: 05

Item Cost: \$3.00

Item Name: Sandwich Rolls

Item Quantity: 50

All... <<

Search...

Tables

- Employee I...
- Inventory T...
- Inventory Table
- Payment T...
- Receipt Ta...
- Student Pr...
- Switchboar...

Forms

Student Profiles Order Order Receipt Table Receipt Table Inventory Table

Item_ID	Item Name	Item Cost	Item Quantit	Click to Add
01	Frying Pans	\$10.00	89	
02	Plastic Forks and	\$3.00	1000	
03	Eggs	\$5.00	25	
04	12oz soda	\$2.00	150	
05	Sandwich Rolls	\$3.00	50	
*		\$0.00	0	

Managers are able to create an employee record so we need to test that this is correct.

All... <<

Search...

Tables

- Employee I...
- Inventory T...
- Order
- Payment T...
- Receipt Ta...
- Student Pr...
- Switchboar...

Forms

- Switchboard Items
- Inventory F...
- Order
- Payment T...
- Receipt Ta...
- Student Pr...
- Switchboard

Employee Information

Employee ID: 3

Employee First Name: First

Employee Last Name: Last

Employee Email: First\_Last@uml.edu

Employee Address: 1234 Washington St

Phone Number: 781-902-5643

All... <<

Search...

Tables

- Employee I...
- Inventory T...
- Order
- Payment T...
- Receipt Ta...
- Student Pr...
- Switchboar...


Forms

Employee Information Employee Information

Employee_ID	Employee Fir	Employee Las	Employee Em	Employee Ad	Phone Numb	Click to Add
1	Tom	Condon	Tcondon@UML	123 Park street	781-000-1212	
2						
3	First	Last	First_Last@uml.	1234 Washingtc	781-902-5643	
*	(New)					

Lastly we need to test if the managers can create a payment for our orders.

**Payments**




Payment\_ID: 13  
 Student\_ID: 01234566  
 Student First Name: Thomas  
 Student Last Name: Condon  
 Payment Date: 12/2/2021  
 Riverhawk Bucks: \$9.00

Payment_ID	Student_ID	Student First	Student Last	Payment Dat	Riverhawk Bu	Click to Add
12	01774870	Thomas	Condon	12/1/2021	\$12.00	
01	02908972	Adam	Hamel	12/1/2021	\$12.00	
02	09873546	Adam	Hamel	12/1/2021	\$10.00	
03	01774870	Thomas	Condon	12/1/2021	\$15.00	
13	01234566	Thomas	Condon	12/2/2021	\$9.00	

## Testing our Reports:

**Daily Sales Report**



Order Date	Order_ID	Student_ID	Item Name	Item Quantit	Item Cost	Total Cost
12/1/2021	100	01774890	12oz Soda	3	\$2.50	\$7.50
12/1/2021	101	01234567	Steak and Cheese sandwic	2	\$11.00	\$22.00
12/1/2021	123	01774870	Chicken Parm Sandwich	1	\$12.00	\$12.00
12/2/2021	104	01234566	Chicken Ceasar Salad	1	\$9.00	\$9.00

Thursday, December 2, 2021 Page 1 of 1

Employee Information Employee Information Payment Table Payment Table Order Inventory Table Payment Table

Search... All Access Objects

Employee I...  
Inventory T...  
Order  
Payment T...  
Reciept Ta...  
Student Pr...  
Switchboar...

Forms  
Employee I...  
Inventory F...  
Order  
Payment T...  
Reciept Ta...

### Inventory Table

Item_ID	Item Cost	Item Name	Item Quantity
01	\$10.00	Frying Pans	89
02	\$3.00	Plastic Forks and Spoons	1000
03	\$5.00	Eggs	25
04	\$2.00	12oz soda	150
05	\$3.00	Sandwich Rolls	50

Thursday, December 2, 2021 Page 1 of 1

Employee Information Employee Information Payment Table Payment Table Order Inventory Table Payment Table

Search... Tables

Employee I...  
Inventory T...  
Order  
Payment T...  
Reciept Ta...  
Student Pr...  
Switchboar...

Forms  
Employee I...  
Inventory F...  
Order  
Payment T...  
Reciept Ta...

### Payment Table

Payment Date	Payment_ID	Student_ID	Student First Name	Student Last Name	awk Bucks
12/1/2021	01	02908972	Adam	Hamel	\$12.00
12/1/2021	02	09873546	Adam	Hamel	\$10.00
12/1/2021	03	01774870	Thomas	Condon	\$15.00
12/1/2021	12	01774870	Thomas	Condon	\$12.00
12/2/2021	13	01234566	Thomas	Condon	\$9.00

Thursday, December 2, 2021 Page 1 of 1

### 3.3.5 Errors in our System

Our Database is not connected to our website currently, so the database requires employees and managers to input the data into the database. Once the connection between the database and website is made the system will work autonomously. The employees and managers will have to collect the kitchen tickets and receipts in order to have the information to input into the system. Our system is still in the testing phase and is not yet autonomous. Once more work is done to the system we can bring it to the university for implementation.