**ONLINE TRANSACTION PROGRAM**

Submitted to:

Sir Jessie Lagrosas

Sir Rangie Obispo

Members:

Abellanosa, Jan Gilbert W.

Camelotes, Earl Franzes Louise L.

Tejada, Randy Louis M.

March 14, 2019

Table of Contents

Contents

[1. INTRODUCTION 2](#_Toc3448165)

[1.1 Overview of the Project 2](#_Toc3448166)

[1.2 Objectives 2](#_Toc3448167)

[1.3 Scope and Limitation 2](#_Toc3448168)

[1.4 Functionalities 3](#_Toc3448169)

[2. Program design and Implementation 3](#_Toc3448170)

[2.a Pseudocode 3](#_Toc3448171)

[2.b Data Structure and Algorithm discussion with code Snippets 4](#_Toc3448172)

[3. Conclusion 5](#_Toc3448173)

[4. Acknowledgement 5](#_Toc3448174)

[5. Reference 5](#_Toc3448175)

[6. Appendices 5](#_Toc3448176)

[A. PROJECT PROPOSAL 5](#_Toc3448177)

ONLINE TRANSACTION PROGRAM

Jan Gilbert Abellanosa

BS Information Technology  
Xavier University

Ateneo de Cagayan

abellanosajangilbert@gmail.com

Earl Louise Camelotes

BS Information Technology

Xavier University   
Ateneo de Cagayan

louisecameltoes@gmail.com

Randy Louis Tejada

BS Information Technology

Xavier University  
Ateneo de Cagayan

luigitejada11@gmail.com

**ABSTRACT**

An Online Transaction is another method where you can transact through the internet when you are in your home and there are people nowadays who want have a priority where it can transact first to those who are rushing. This paper contains the result and implementation of Online Transaction Program that shows between the regular and with priority in transaction. The paper focuses more about the data and structure of transaction funds where there are people have a priority. Inside the paper you can see the overview of the project, objectives of the program, and the functionalities of each method. Also the program design and implementation will be discussed and further elaborated by this paper. The overall result of this paper was successful with a simple program and run by the console properly.

**CCS Concepts**

**CCS → Information systems → Data management systems → Database management system engines → Database transaction processing → Transaction logging**

**Keywords**

Online; Transaction; Programming; PIN-debit; funds; credit card; Payment.

# INTRODUCTION

Online transactions take many forms. In business-to-business (B2B) transactions, businesses conduct transactions with one another. [8]

Online transaction is a payment method in which the transfer of fund or money happens online over electronic fund transfer. Online transaction process (OLTP) is secure and password protected. Three steps involved in the online transaction are Registration, Placing an order, and, Payment.[7]

When you pay for goods or services with your debit card, you have an option for the payment to be processed in two different ways: as an offline transaction via a credit card processing network, or as an online transaction via an EFT system, requiring a personal identification number (PIN) to complete the process.[6]

Debit card makes it easy to use in checking account, and debit cards are accepted almost everywhere credit cards are accepted. But, if you have the option to use a credit card, it’s probably safer to use credit especially when shopping online. [4]

An online transaction using a debit card online isn’t the only way to get scam directly, despite all the hazards, nowadays people use a lot on using debit card because it is sufficient and handy. You don’t have to be afraid using debit card on online because there are a lot of websites are safe and secured.

## Overview of the Project

An online transaction, also known as a PIN-debit transaction, is a password-protected payment method that authorizes a transfer of funds over an electronic funds transfer (EFT) This project is aimed to create a simple program that emulates an online transaction wherein the user can create a user ID, input their transaction/s, and check the status of their transaction if it is next or not. The project will also allow the user to cancel their transaction at any given time. The user can also search for their transactions at any given time. The user can also put their transaction first if it is deemed urgent.

## Objectives

* **General Objective:** This project aims to make a working online transaction where it is divided into two parts the regular and with priority.
* **Specific Objectives:**
* Create a user ID, input their transaction/s, and check the status of their transaction if it is next or not.
* The project will also allow the user to cancel their transaction at any given time.
* The user can also search for their transactions at any given time and also put their transaction first in priority.

## Scope and Limitation

This program is to create a simple and a concept of online transaction system that authorize by user and have a priority when transaction happens. This program will be run from the command prompt only; the program will not be using any additional graphic interface. The other limitation if this project is we don’t have a program where the data can be saved in the file reader. There is also no automatic dequeuer, transactions will be manually removed.

## Functionalities

* Account creation
* Account creation will allow the user create a user ID name and input their transaction.
* Transaction creation
* Transaction creation will allow the user create a transaction between other users
* List of Transactions
* List of Transaction will show the list that been added in queue.
* Search specific transaction
* Search specific transaction will allow the user to see their transaction, priority or otherwise.

# Program design and Implementation

## 2.a Pseudocode

2.a.1 Node class

This project utilizes 3 classes with the Node class establishing of various key information such as the name of the client, their declared price, as well as the reference pointer to the next node. This was achieved using a linked list data structure. This class contains the getters and setters that involves the accounts and pro. This declaration is shown below:

Create Node class{

` Initialize private int called price;

Initialize private String called doto;

Initialize private node next;

Initialize private boolean highprio;

Create QNode normal prio class constructor (String, int, and next){

}

Create QNode high prio class constructor(String, int, next,and Boolean){

}

Create getter that gets Price{

returns the price;

}

Create getter that gets Name{

return doto as n;

}

Create setter that sets Name(String n){

Set doto to n;

}

Create setter method that sets Price(int p){

Set price to p;

}

Create getter method that gets next and returns next{  
}

Create setter method that sets next as nx{  
}

Create Boolean method getprio{

returns highprio;

}

2.a.2 List Class

This class contains all the methods used that involves the operations of the program such as adding transactions, removing, searching, priority transaction, and display. This implementation as pseudocode is shown below:

Create QueueL class{

Initialize private QNode called head;

Initalize private QNode called tail;

Initalize private QNode called temp;

Initalize private QNode called tamp;

Create QueueL constructor{

Set head and tail equal to null;

}

Create method boolean isEmpty(){

returns head is equal to null

}

Create method add(name and price){

temp receives new node(n,p);

If queue isEmpty(){

head and tail equals to temp;}

else

set tail to next node

tail receives next

}

Create method addhead(QNode AddThis){

temp receives AddThis;

temp.setNext(head);

head receives temp;}

Create method dequeue(){

temp receives head;

head receives head.getnext();

temp receives null;

}

Create method display(){

Temp receives head;

while(temp is not equals to null){

print temp.getName() and temp.getPrice();

temp receives temp.getnext();}

Create method boolean containsPrio(){

Set boolean x to false;

for(temp equals head and temp is not equal to null; temp receives temp.getnext())

if(temp.getprio() is equal to true){

x receives true;

break;

}

}  
return x;

}

Create method addprio(name and price){

if(queue does not containsPrio())

temp receives new Node(n,p,head,true);

addHead(temp);

}  
else

QNode curr receives head;

while(curr.getnext().getprio()){

curr = curr.getnext();

}

Temp receives new node(n,p, null, true);

Temp.setNext(curr.getnext());

curr.setNext (temp)}

Create method boolean search(String name){

Set boolean a to false;

for( temp equals head and temp is not equal to null, temp receives temp.getnext()){

if(n.equalsIgnoreCase(temp.getName()) a is equal to true;

}

Return a;

}

2.a.3 Main class

The Main class contains the main method of which the code functions on:

Print menu;

Case 1: add name and transaction

Program asks for name and price of transaction, after that it immediately queues it

pause();

Break;

Case 2: dequeue

Program dequeues first transaction

print(Transaction complete. Removing..);

pause();

break;

case 3: search transaction

program will ask if your transaction is there and user will input name

if transaction is present, program prints “your transaction is here”

else program will print transaction is non-existent;

case 4: priority

program asks for name and price of transaction and queues it immediately at the first position;

case 5: display

prints contents of queue

}

## 2.b Data Structure and Algorithm discussion with code Snippets

*2.b.1 Linked List*

Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array.[2]

public class QueueL {

private QNode head;

private QNode tail;

private QNode temp;

private QNode tamp;

public QueueL(){

head = tail = null;

}

public boolean isEmpty(){

return(head==null);

}

**Figure 1. Implementation of Linked List**

*2.b.2 Queues*

A Queue is a linear structure which follows a particular order in which the operations are performed. The order is First In First Out (FIFO). [9]

public void add(String n, int p)

{

temp = new QNode(n, p);

if(isEmpty()){

head=tail=temp;}

else{

tail.setNext(temp);

tail=tail.getnext();

}

}

**Figure 1. Implementation of Queues**

*2.b.3 Dequeue*

The dequeue is a double ended queue and data elements can be added or removed from either end.[3]

public void dequeue(){

temp=head;

head=head.getnext();

temp=null;

}

**Figure 2. Implementation of Deque**

*2.b.4 Linear Search*

Search is a process of finding a value in a list of values. In other words, searching is the process of locating given value position in a list of values.[5]

public boolean search(String n){

boolean a = false;

for(temp=head; temp!=null; temp=temp.getnext()){

if(n.equalsIgnoreCase(temp.getName())) a=true;

}

return a;

}

**Figure 3. Implementation of Linear Search**

*2.b.5 Priority Queues*

Priority Queue is more specialized data structure than Queue. Like ordinary queue, priority queue has same method but with a major difference. In Priority queue items are ordered by key value so that item with the lowest value of key is at front and item with the highest value of key is at rear or vice versa. So we're assigned priority to item based on its key value. Lower the value, higher the priority.[1]

public void addprio(String n, int p){

temp = new QNode(n,p,true);

if(!containsPrio()){

temp = new QNode(n, p);

temp.setNext(head);

head=temp;

}

else {

for(temp=head; temp!=null; temp=temp.getnext()){

if (!temp.getnext().getprio())break;

}

tamp = new QNode(n,p,temp.getnext());

temp.setNext(tamp);

}

}

**Figure 4. Implementation of Priority Queues**

# Conclusion

An Online Transaction is very handy when it comes to transactions through online means because you don’t need to make any effort to walk or go to places. You can stay at home looking at your phones and through internet. The program that implemented this project successfully works but is not yet fully complete in the likes of interface and advanced options. It is just a simple program that creates an account ID and the transaction and a priority transaction.

# Acknowledgement

This project would not be possible without the help of our subject professors, friends and classmates.

We would like to thank our advisers in our CC12 and CC13 subject Sir Jessie Lagrosas and Sir Rangie Obispo in approving our project and sharing to us their knowledge on programming that helped us in accomplishing our finished and polished code.

And also we would like to thank Mr. Daniel James Dolor for helping us about our program.

# Reference

[1] Data Structures & Algorithms Priority Queue: *https://www.tutorialspoint.com/data\_structures\_algorithms/priority\_queue.htm*. Accessed: 2019-03-10.

[2] Data Structures and Algorithms Linked List: *https://www.tutorialspoint.com/data\_structures\_algorithms/linked\_list\_algorithms.htm*. Accessed: 2019-03-10.

[3] Deque in Java: *https://www.tutorialspoint.com/deque-in-java*. Accessed: 2019-03-10.

[4] How to Pay Online With Debit or Credit Cards (Safely): *https://www.thebalance.com/can-i-use-a-debit-card-online-315325*. Accessed: 2019-03-10.

[5] Linear Search Algorithm - Data Structures: *http://btechsmartclass.com/data\_structures/linear-search.html*. Accessed: 2019-03-10.

[6] Online Transaction Definition & Example | InvestingAnswers: *https://investinganswers.com/dictionary/online-transaction*. Accessed: 2019-03-10.

[7] Online Transaction Processing - Steps, Stages, and Security: *https://www.toppr.com/guides/business-studies/emerging-modes-of-business/online-transactions-and-security-of-e-transactions/*. Accessed: 2019-03-10.

[8] Online Transactions | Encyclopedia.com: *https://www.encyclopedia.com/books/educational-magazines/online-transactions*. Accessed: 2019-03-10.

[9] Queue Data Structure - GeeksforGeeks: *https://www.geeksforgeeks.org/queue-data-structure/*. Accessed: 2019-03-10.

# Appendices

# PROJECT PROPOSAL

PROJECT PROPOSAL FOR CC12 AND CC13

ONLINE TRANSTACTION PROGRAM

FEBRUARY 5, 2019

**Overview**

1. Project Background and Description

An online transaction, also known as a PIN-debit transaction, is a password-protected payment method that authorizes a transfer of funds over an electronic funds transfer (EFT)

This project is aimed to create a simple program that emulates an online transaction wherein the user can create a user ID, input their transaction/s, and check the status of their transaction if it is next or not.

The project will also allow the user to cancel their transaction at any given time. The user can also search for their transactions at any given time. The user can also put their transaction first if it is deemed urgent.

1. Features and Functionalities

The program is designed to have the features of the following:

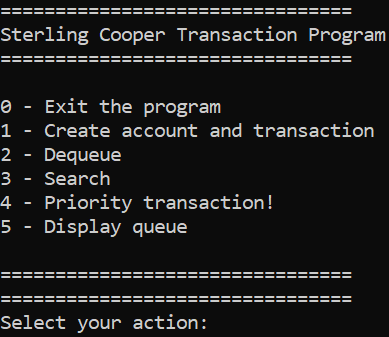
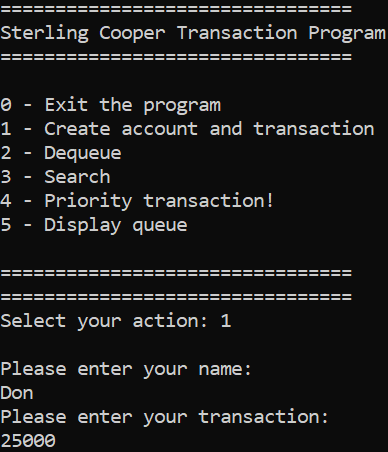
* Account Creation/Deletion
* Transaction Creation
* List of Transactions
* Search for specific transaction

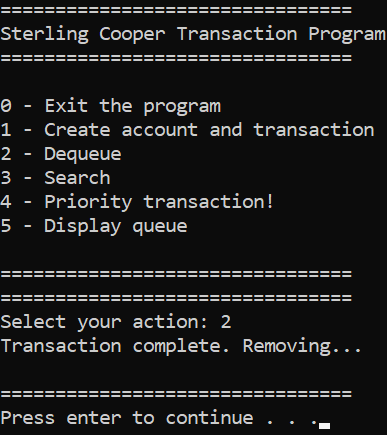
1. Possible Data Structures and Algorithms to be Used

* Queues
* Linear Search
* Priority Queue

**PHOTO DOCUMENTATION**

1. **PROGRAM SCREENSHOT** (Left to Right)

** **

**** 