SMART AND ADVANCED MANAGEMENT (SAM)

SYSTEM SPECIFICATION (DESIGN)

Prepared for:

Elain Weltz, Coordinator
5000 Foods

Prepared by:

Tai Doan
EnT Design, Inc

Table of Contents

Executive Summary

Coordinator of 5000 Foods, Ms. Elaine Weltz, has consulted EnT Design to design a computerized system that assists with the operation of a food bank and distribution service, while keeps 5000 Foods managed as a business. EnT Design has proposed a system called Smart and Advanced Management that will keep track on food and money going in and out 5000 Foods, and will provide detailed business report to aid in decision making.

In the System Proposal, EnT Design has conducted a preliminary assessment and analysis of the project, and we found SAM to be feasible and beneficial to all of the stakeholders. EnT Design and 5000 Foods are in the process of developing SAM. This document includes an introduction of the system, then gives a structural model, two infrastructure models, a security plan and preliminary user-interface designs. We will use this as a reference during the future implementation of this system to ensure that we will deliver such a system works with full functionality and fault-free.

1 Introduction

1.1 Problem Statements and Project Vision

5000 Foods is a non-profit organization, which was established to help feed the hungry in the Kirkland area. Ms. Weltz has contacted us about designing an organized, intuitive and complete computerized system to support all aspects of 5000 Foods.

EnT Design has developed Smart and Advanced Management (SAM), a system that will manage the operations of 5000 Foods in a professional business-like style. SAM is promised to solve the current problem and bring numerous additional benefits to the community.

By covering these areas: Food, Resources, Expenses and Distribution (FRED – as Ms. Weltz described in the System Request), SAM will manage 5000 Foods effectively in order to make maximum use out of minimum resources. Stakeholders and those who will benefit from 5000 Foods include 5000 Foods committees and volunteers, clients, donors, grocery stores and EnT Design developers. (For more detail, please refer to section 1.1, 1.2 and 1.4 of the System Proposal).

1.2 System Service

Smart and Advanced Management will meet following functional requirements: (Read section 4 and 5 of the System Proposal for more information)

Registration

- Users create or edit account and get ID card (See use-case 0)

Keeping track of Donation

- Donor makes donation (See use-case 1)
- Staff manages donation, record information of donated food (See use-case 2,9)
- Staff sorts out and organizes food in to boxes/bags with identical stickers (See use-case 3,9)
- Clients check-out food by barcode scanning. There is delivery service based on client's desire (See use-case 4,5)
- Staff keeps track on food by allowing to search for inventory, food location, food expiration date (See use-case 7)
- The system is able to give out information of donation/distribution, and information of clients/donors (See use-case 1,7)

Report functional requirements

- The system must be able to provide reports for all activities of 5000 Foods (including receipt for donor, food report and financial report) (See use-case 1,6,7)

Edit database

- Administrators/ Authorized Staff are able to view, manage, and modify all necessary aspects of 5000 Foods (including food information and user database) (See use-case 7,8)

1.3 Nonfunctional Requirements and Design Constraints

The following lists the major nonfunctional requirements and design constraints to consider when implementing SAM

- The hardware purchases must be minimal due to the tight budget.
- SAM must perform in variety of operating systems, allow users to get access from different PCs by scanning their ID cards.
- SAM must be easy to use for many types of users from schoolkids to seniors.
- The around environment must be maintain as required to ensure SAM's working condition.
- SAM must be easy to maintain and there are additional expenses needed for necessary maintenance/upgrade.
- EnT Design will provide documentations to help initial troubleshoot if there is any problem, as well as free tutorial sessions for new volunteer workers.
- Based on the agreement with Ms. Weltz, EnT will ensure to deliver the system by the end of 2018

1.4 System Evolution

In version 1, EnT Design will deliver the system that meets the basic requirements of 5000 Foods. We will add additional features in the future including: Pick-up donation, Monthly food package and meal for the homeless. Depend on the popularity growth of 5000 Foods, we will deliver 1 or 2 more versions, and each will take around 6 months (maximum) to be completely developed.

1.5 Document Outline

This document contains the following sections:

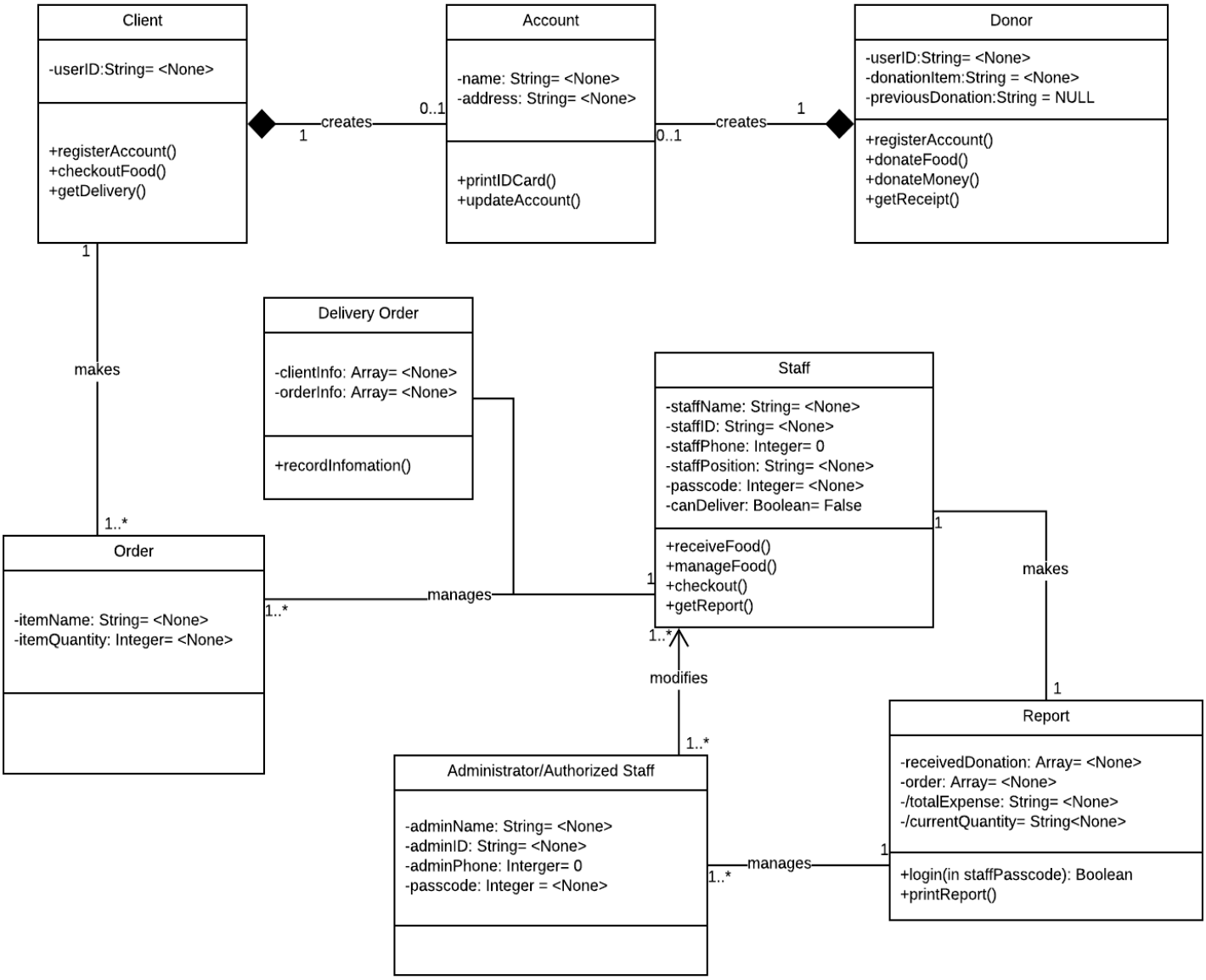
- Structure Model: A class diagram, and associated class metadata, that shows how objects and stored data will interact in SAM
- Architecture Design: Two deployment diagrams to show the physical architecture of SAM. Also included are hardware and software requirements and a system security plan.
- User-interface: Forms, screens and basic requirements and constraints for the user-interface design of SAM. Also included are layout designs for printed reports.
- Appendices: Includes bibliography and supporting documentation.

2 Structure Model

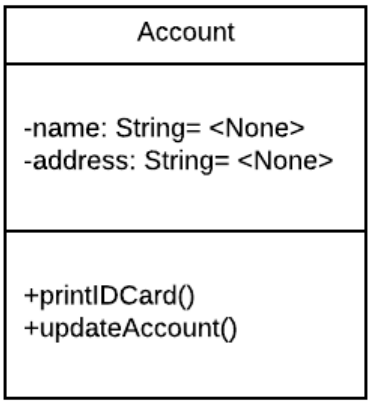
2.1 Introduction

This section shows a class diagram of the system, and the metadata, which will give more detail about attributes and operations of each of the represented classes.

2.2 Class Diagram



2.3 Metadata



Description: Represents the account that will be registered by the users

Visibility: Public

Is Abstract: No

Attributes:

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
name	Last, First name	String	No	No	Private	0..1	None
address	Address of the user (for delivery purpose)	String	No	No	Private	0..1	None

Operations:

Name	Description	Return Type	Parameters	Visibility	Scope	Query	Polymorphic
printIDCard	User prints out their ID card after registering an account	None	None	Public	Instance	Yes	No
updateAccount	Allows user to update account's information	None	None	Public	Instance	No	No

Processing Outlines:

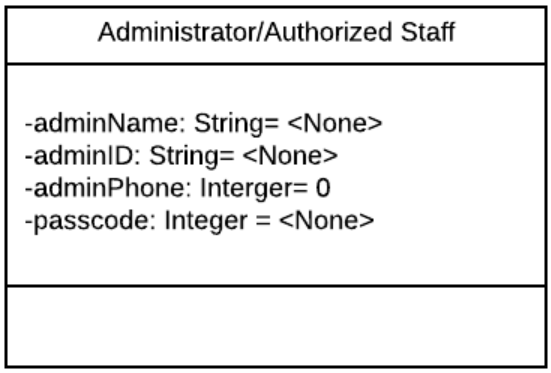
printIDCard:

while there are (name && address)

print ID Card

updateAccount:

update account information and save



Description: Represents administrator/Authorized staff of 5000 Foods

Visibility: Public

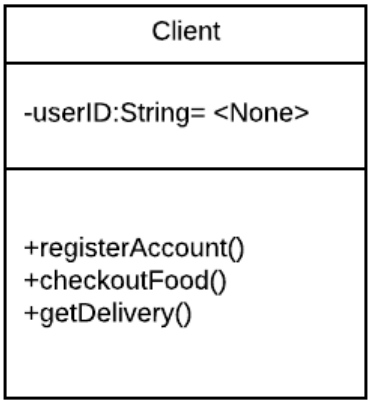
Is Abstract: No

Attributes:

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
adminName	Last, First name	String	No	Yes	Private	1	None
adminID	Identical ID of admininistrator	String	No	No	Private	1	None
adminPhone	Phone number	Integer	No	Yes	Private	1	0
passcode	Restricted passcode to get access to different departments	Integer	No	Yes	Private	1..*	None

Operations: N/A

Processing Outlines: N/A



Description: Represents the client of 5000 Foods

Visibility: Public

Is Abstract: No

Attributes:

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
userID	Identical ID of client	String	No	Yes	Private	1	None

Operations:

Name	Description	Return Type	Parameters	Visibility	Scope	Query	Polymorphic
registerAccount	Client can register for new account if has not had one	None	In name: String In address: String	Public	Instance	Yes	No
checkoutFood	Client checks out food at the counter	None	None	Public	Instance	Yes	No
getDelivery	User can get delivery if cannot come to pick up food	None	None	Public	Instance	No	No

Processing Outlines:

registerAccount:

if not Account

in name

in address

confirm to create new account

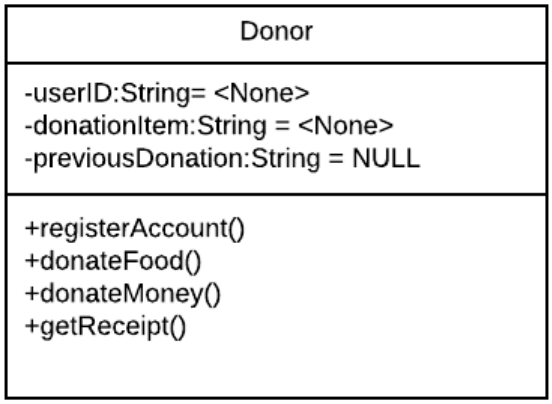
checkoutFood:

if there is food in cart

scan barcode sticker

getDelivery

Order and get delivery service if cannot come to pick up



Description: Represents the donor of 5000 Foods

Visibility: Public

Is Abstract: No

Attributes:

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
userID	Identical ID of donor	String	No	Yes	Private	1	None
donationItem	Which is donated from the donor	String	No	No	Public	1	None
previousDonation	Information of previous donations of the donor	String	No	Yes	Public	1	NULL

Operations:

Name	Description	Return Type	Parameters	Visibility	Scope	Query	Polymorphic
registerAccount	Donor can register for new account if has not had one	None	In name: String In address: String	Public	Instance	Yes	No
donateFood	Donor donates food at the check in counter	None	In donorName : String In itemName: String In itemQuantity: String	Public	Instance	Yes	No
donateMoney	Donor can donate money at the check in counter	None	In amount: float	Public	Instance	Yes	No
getReceipt	Donor gets the receipt of their donation	None	None	Public	Instance	No	No

Processing Outlines:

registerAccount:

if not Account

in name

in address

confirm to create new account

donateFood:

while the donated item is food

in donorName

in itemName

in itemQuantity

donateMoney:

while the donated item is not food

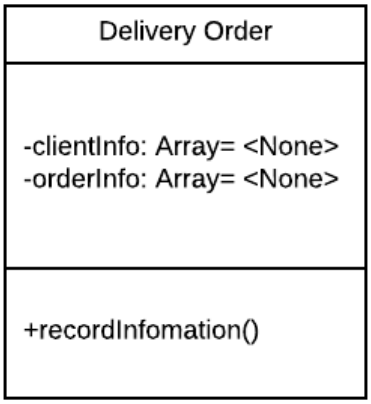
in donorName

in amount

getReceipt:

if the donation has made

get receipt from the staff



Description: Represents the order that is delivered to the client

Visibility: Public

Is Abstract: No

Attributes:

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
clientInfo	Last, First name of client	Array	No	Yes	Private	1	None
orderInfo	Information of order	Array	No	No	Private	1	None

Operations:

Name	Description	Return Type	Parameters	Visibility	Scope	Query	Polymorphic
recordInfomation	staff record information of client and order before delivering	None	None	Public	Instance	No	No

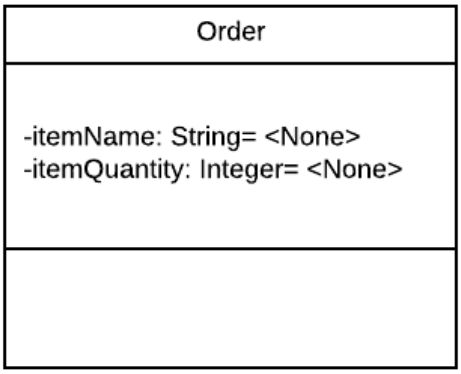
Processing Outlines

recordInformation

while there is a staff that canDeliver is true available

input clientName and orderInfo

deliver food to the client



Description: Represents the order made by client

Visibility: Public

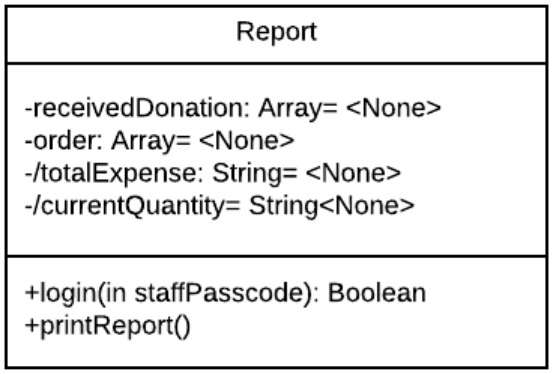
Is Abstract: No

Attributes:

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
itemName	What food is picked up by the client	String	No	Yes	Private	1..*	None
itemQuantity	Quantity of food	Integer	No	No	Private	1..*	None

Operations: N/A

Processing Outlines: N/A



if valid output report / else inform an error

Description: Represents administrator/Authorized staff of 5000 Foods

Visibility: Public

Is Abstract: No

Attributes:

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
receivedDonation	Information of donation	Array	No	Yes	Private	1	None
order	Information of taken out food	Array	No	No	Private	1	None
/totalExpense	Information of total amount of expenses	String	Yes	Yes	Private	1	None
/currentQuantity	Available quantity of food	String	Yes	Yes	Private	1	None

Operations:

Name	Description	Return Type	Parameters	Visibility	Scope	Query	Polymorphic
login	Login to get access to see the report	None	Passcode: Integer	Public	Instance	Yes	No
printReport	Print out the report	None	None	Public	Classifier	Yes	No

Processing Outlines

Login

Admin or authorized staff scan ID card

Input passcode

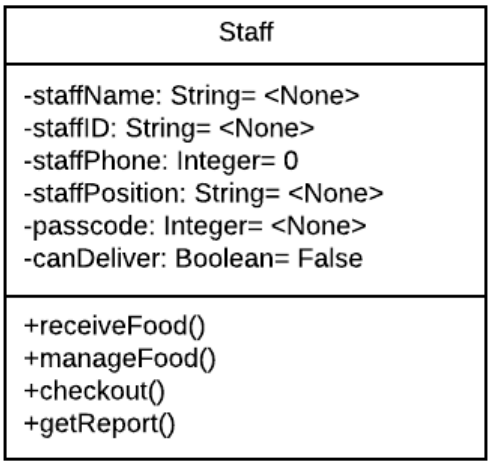
If valid get access to the report page

If not valid inform an error

printReport

while in the report page

print report



Description: Represents volunteer staff that works for 5000 Foods

Visibility: Public

Is Abstract: No

Attributes:

Name	Description	Data Type	Is Derived	Is Read Only	Visibility	Multiplicity	Default Value
staffName	Last, First name	String	No	Yes	Private	1	None
staffID	Identical ID of staff	String	No	No	Private	1	None
staffPhone	Phone number	Integer	No	Yes	Private	1	0
staffPosition	Position of staff in 5000 Foods	String	No	Yes	Private	1	None
passcode	Restricted passcode to get access to different departments	Integer	No	Yes	Private	0..*	None
canDeliver	If the staff can handle delivery service	Boolean	No	Yes	Public	1	False

Operations:

Name	Description	Return Type	Parameters	Visibility	Scope	Query	Polymorphic
receiveFood	Staff receives food donation	None	None	Public	Instance	No	No
manageFood	Staff record food information and organize food	None	In itemName: String In itemQuantity: String In itemLocation	Public	Instance	Yes	No
checkout	Staff scan item from the client	None	None	Public	Instance	No	No
getReport	Authorized staff get the report	None	In passcode: Integer	Public	Instance	Yes	No

Processing Outlines:

receiveFood:

while there is donated food

staff receives food

manageFood:

if there is food donation

in itemName

in itemQuantity

in itemLocation

checkout

scan barcode sticker to checkout

getReport

in passcode

3 Architecture Design

3.1 Introduction

The following includes infrastructure models for SAM to help clarify the physical architecture of the system, hardware and software that are necessary for operation and give a security plan.

EnT Design recommends using a third-party cloud server for data storage of SAM. All of the devices will be connected to a main server and the main server is connected to the cloud. This cloud server will store user database, food information and financial data of 5000 Foods. This model is not only efficient but also very safe and secure.

3.2 Infrastructure Model

Figure 1: Deployment Diagram – Architecture Overview

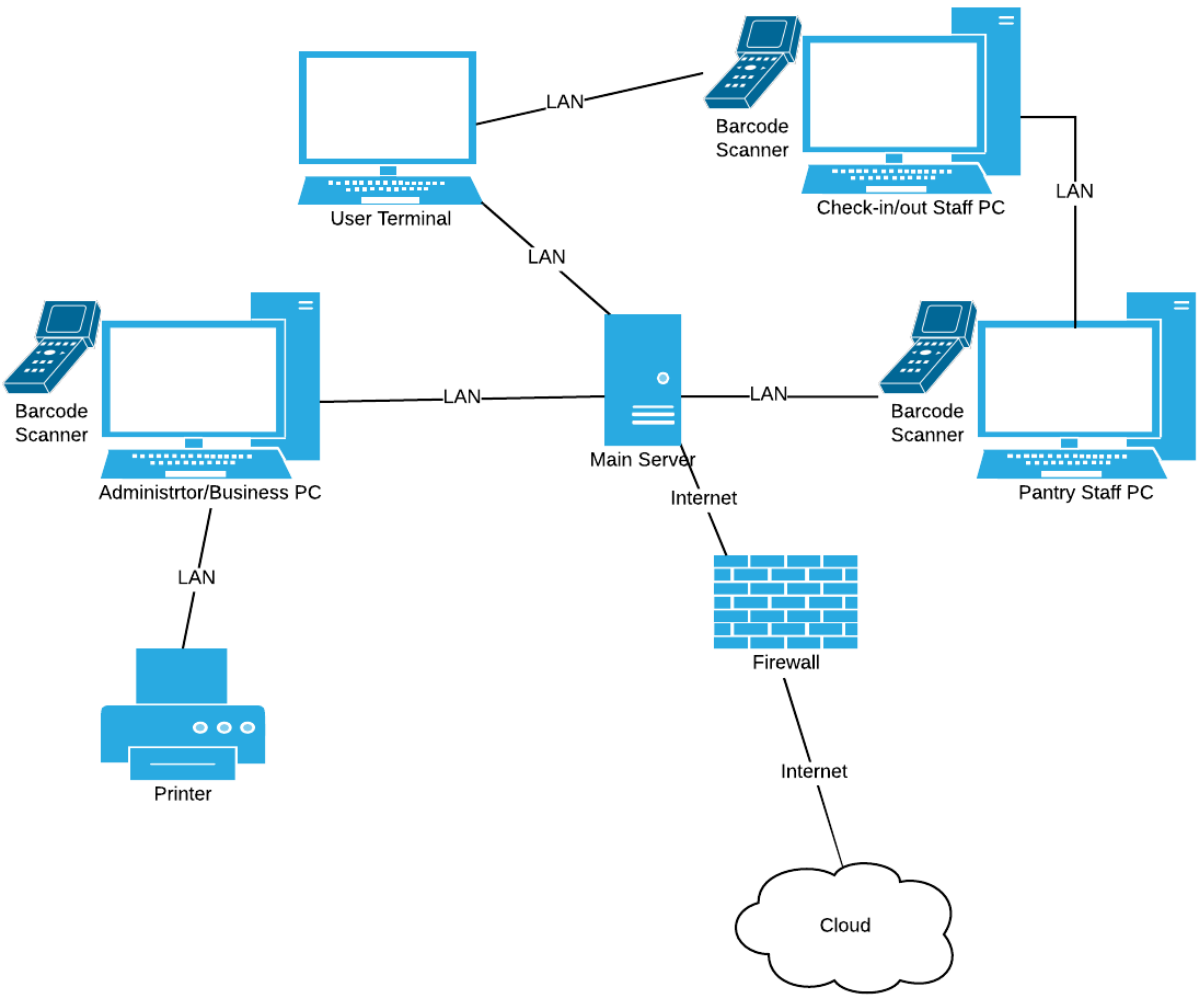
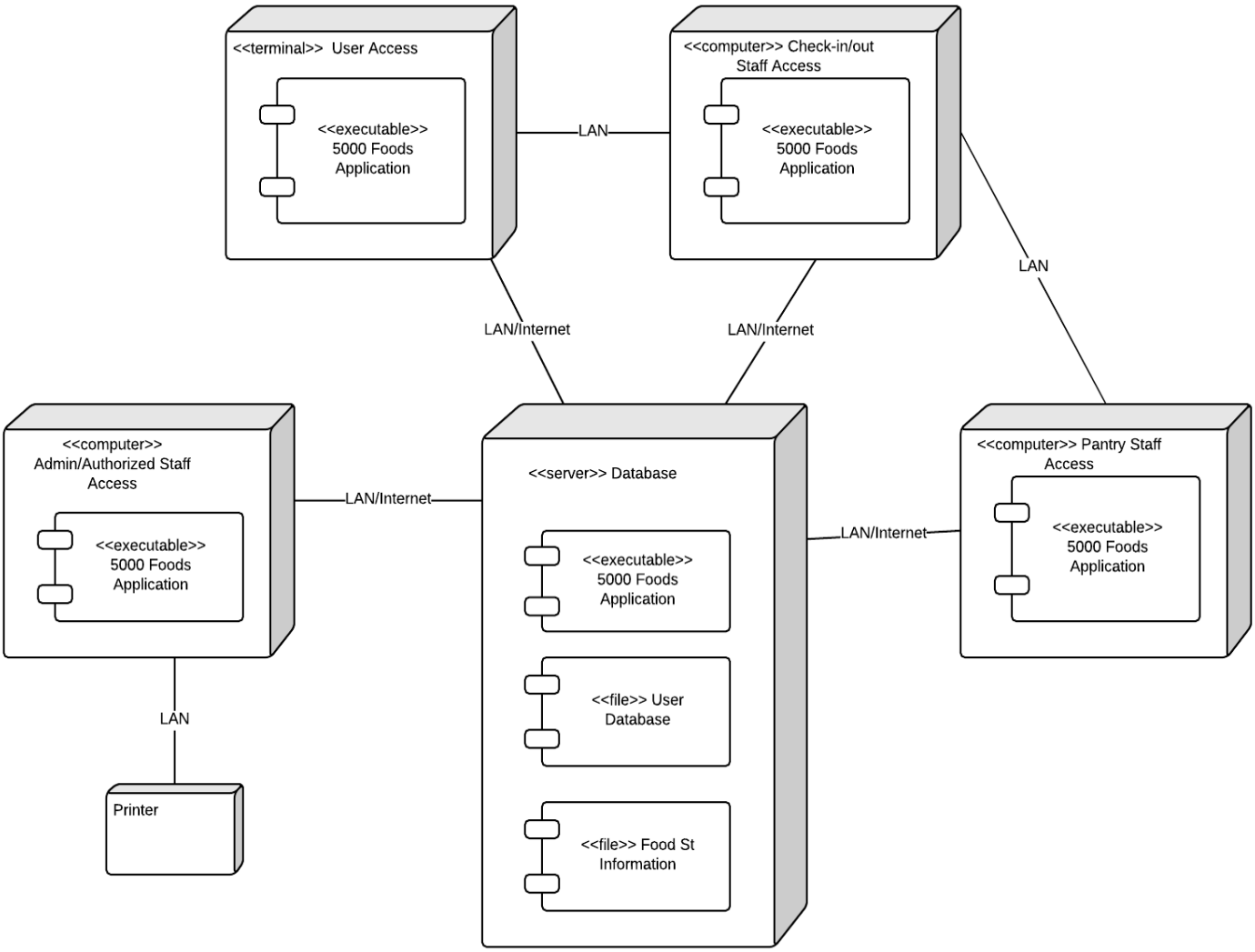


Figure 2: Deployment diagram – Nodes and Artifacts



3.3 Hardware and Software Requirements

3.3.1 Required Hardware Components

- Computers (PCs) for check-in/out staff, pantry staff, at least one at each station
- A terminal (can use a monitor) for user's registration
- Computer (PC) for Administrator/Business Staff (prefer to be brand-new)
- Printer for reports
- Delivery Staff's devices for GPS (5000 Foods is not responsible for these)
- Cloud Server Subscription
- Barcode scanners at each station

3.3.2 Required Software Components

- SAM must support the variety of operating systems, including Windows (8 or later) and MacOS
- Software supports on financial report
- Anti-virus software.
- Cloud must provide database management

3.4 Security Plan

The security plan that EnT Design has made for SAM focuses on unauthorized access and loss of important information. The staffs at 5000 Foods are all volunteer workers, so the professional skills on their job are not guaranteed. Therefore, background check and periodic training are mandatory. The cloud server provider that comes into contract with 5000 Foods must be reputable and trustworthy. All passcodes required for accesses to different departments must be secured within the authorized staff. Other threats include physical damage to, or theft of, office and kitchen computers, viruses and hackers. These threats will be mitigated by the following controls.

Security Table:

Threat / Components	Unauthorized Access		Disruption, Destruction and Disaster			
	External Intruder	**Internal Intruder**	**Fire**	**Power loss**	**Virus**	**Destruction**
Server	4,7,9	4,7,9	10	10	6,7,10	1,8,10
5000 Foods Devices	3,5,8,11	3,5,8,11	3,4,11	2,3	X	1,3,
Food Storage	3,5,8,9,11	3,5,8,9,11	4,11	2	X	1,8,11
Software	4,7,9	4,7,9	X	X	6,7,8	X
Database	4,7,9	4,7,9	10	10	6,7,10	10
People	5,8	5,6	5,8,11	8	X	8

Controls:

1. Disaster recovery and emergency plans
2. Back-up power generator
3. Insurance for hardware purchased by 5000 Foods
4. Fire alarms for office and pantry
5. Background check
6. Firewall between main server and cloud server
7. Anti-virus software
8. Training for staff
9. Strong passcode
10. Hire a trustworthy cloud server provider, and easy to back-up data.
11. Security camera and monitor

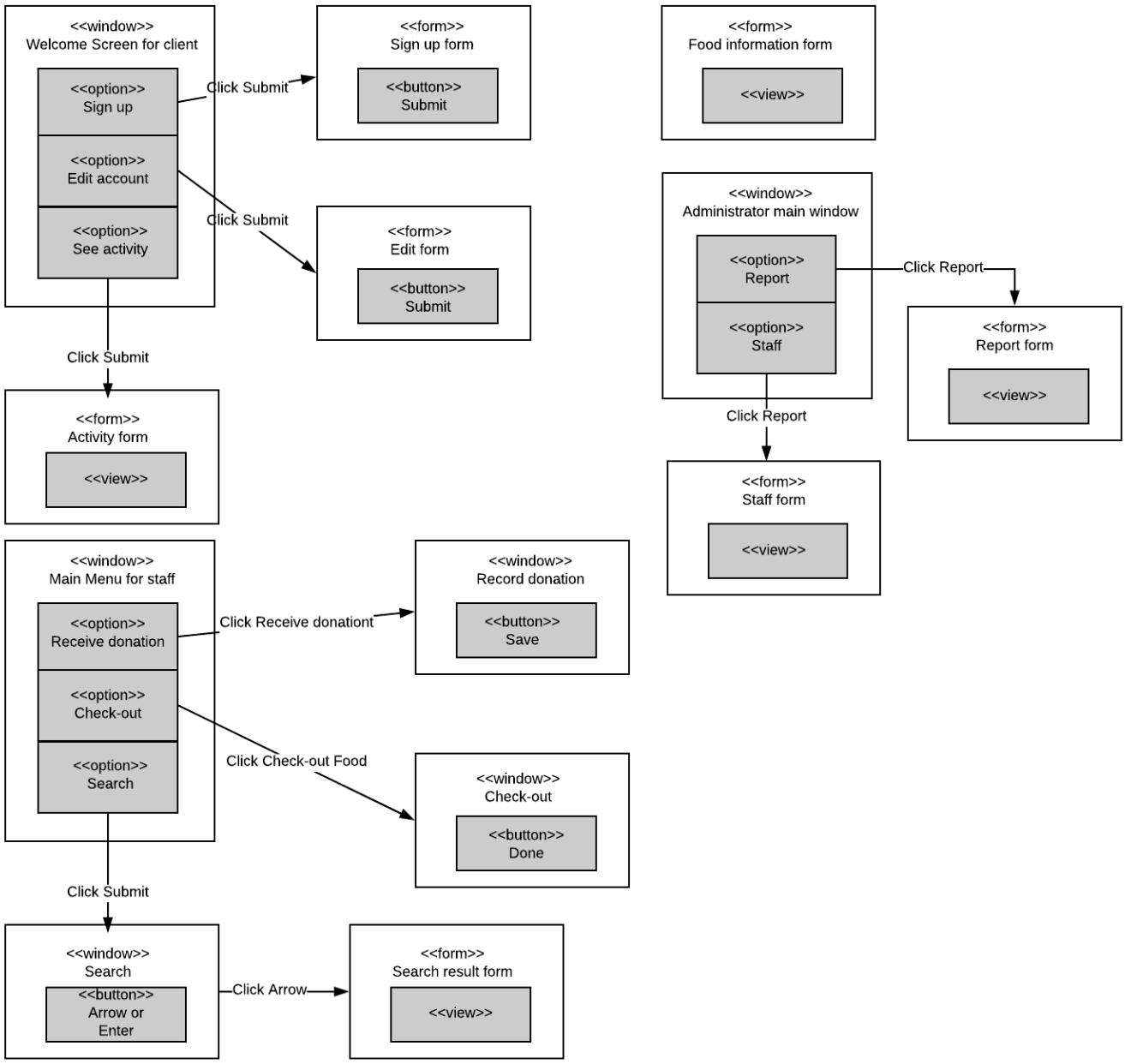
4 User Interface

4.1 User Interface Requirements and Constraints

This section includes designs for user-interaction and printed output. Forms: Screen/User-Interaction Design shows SAM's visual components and screens, while Report: "Printed Output" Design show the preliminary outlines for the business report of 5000 Foods.

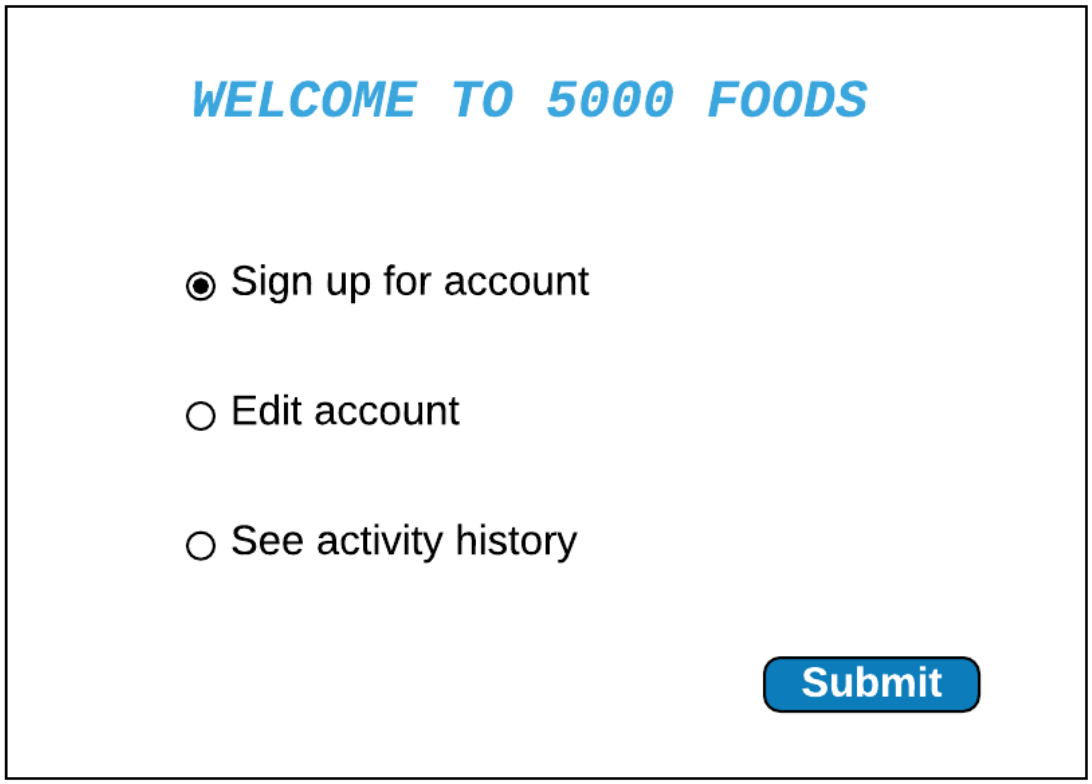
Knowing that users of 5000 Foods may or may not be tech-savvy, the system will be designed so that users are able to use, knowing that it is user-friendly and easy to learn. Our priority of this design is utility, usability. The application layouts for client, donor and for staff will look the same, but they have very different motivations for the operations. There will be a guideline at each screen so that users may follow to reach their goal, and a button to ask for help from experts.

4.2 Window Navigation Diagram

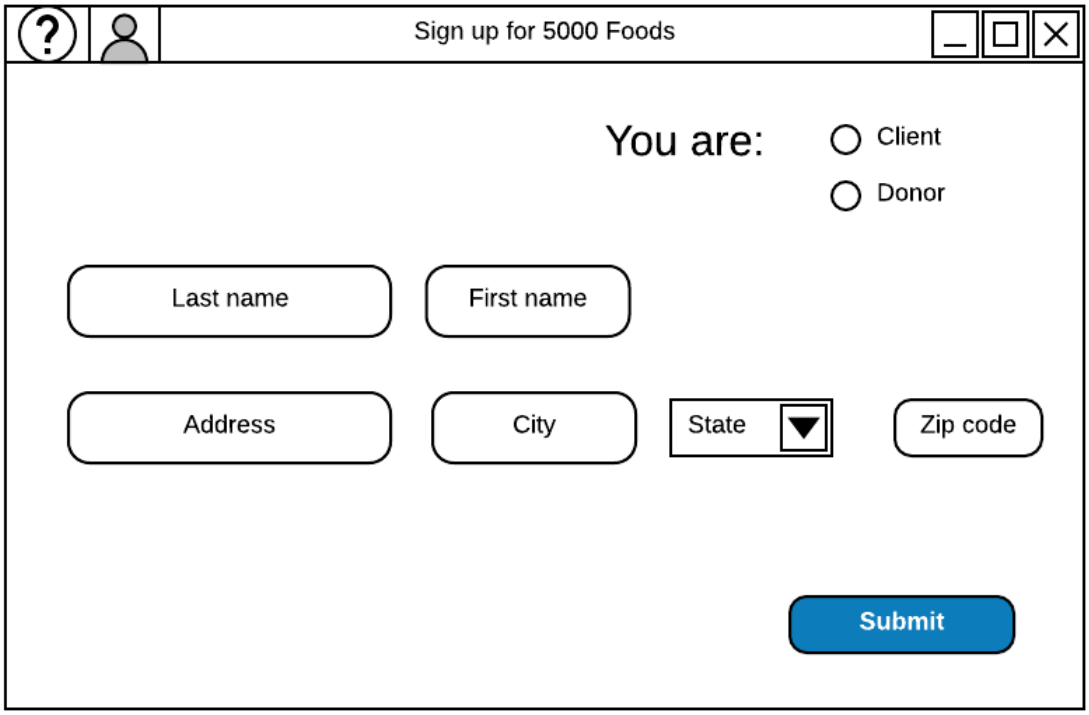


4.3 Forms: Screen/User-Interaction Design

4.3.1 User welcome screen

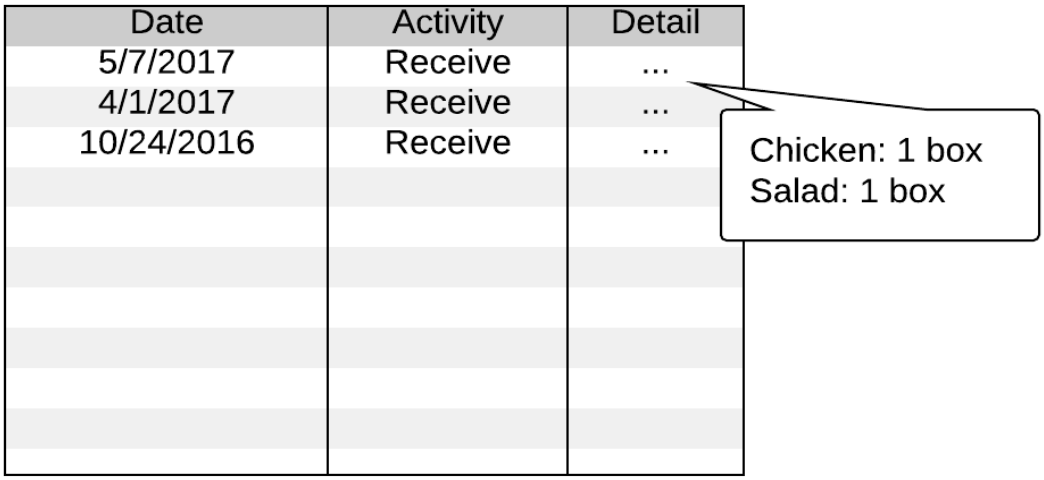


4.3.2 Registration Form



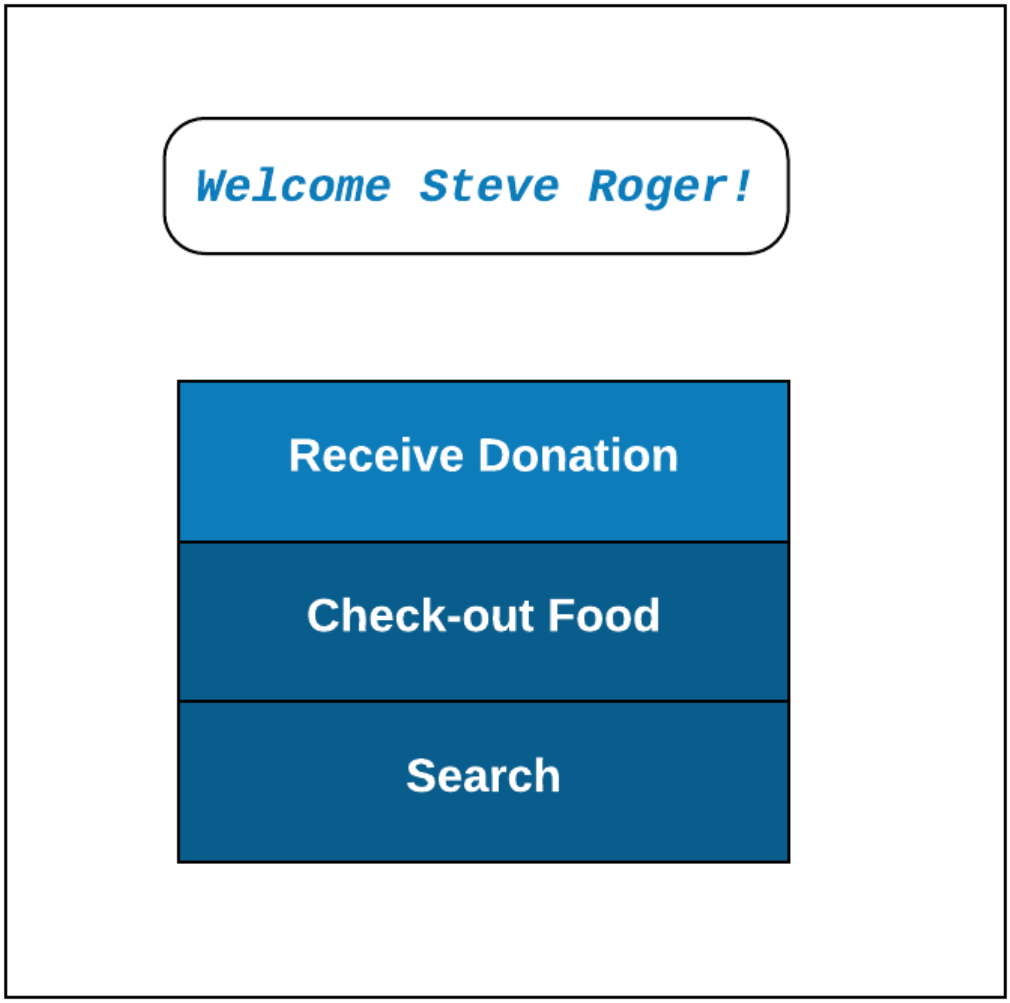
Note: The “Edit account” screen is similar to this screen

4.3.3 Account Activity screen



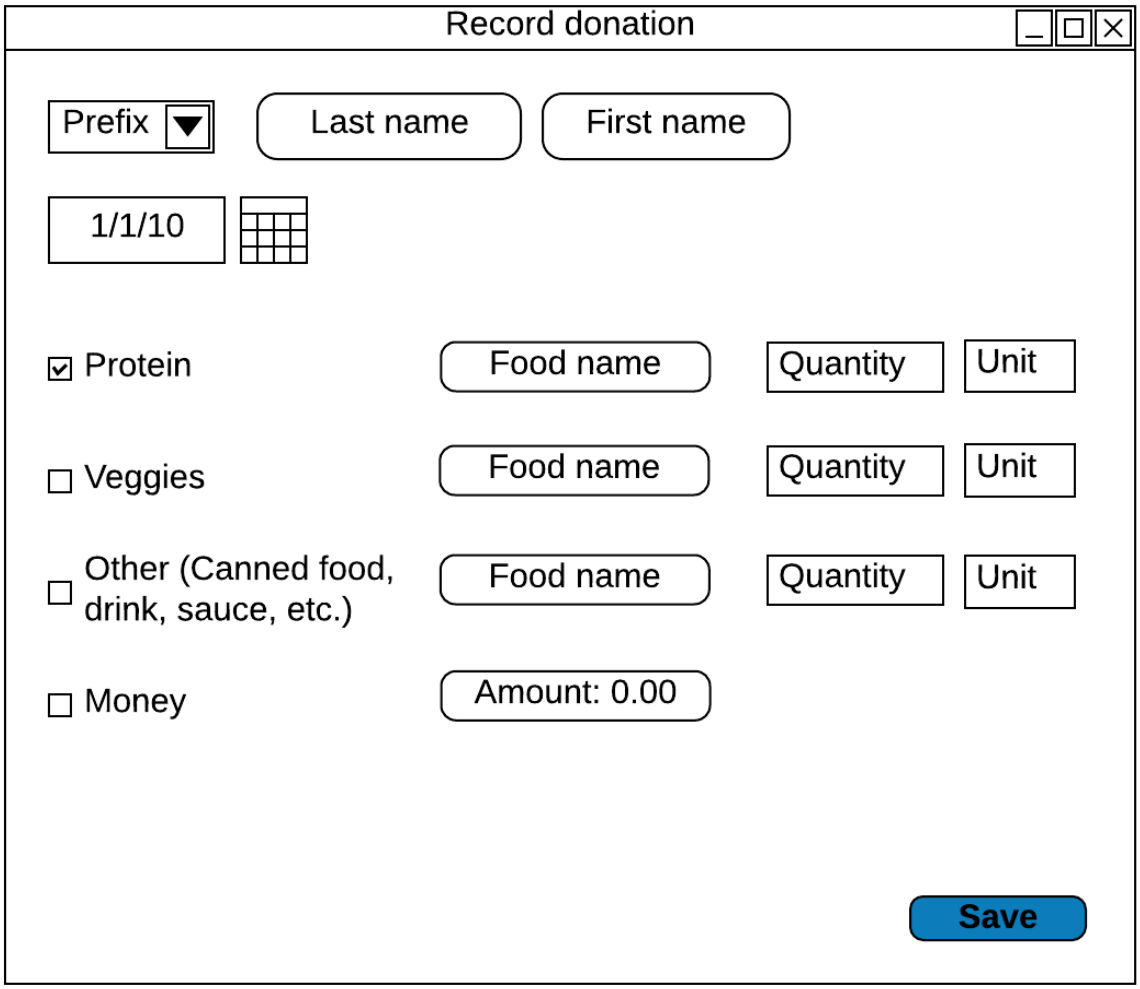
Note: To see “Detail”, click on the dots […]

4.3.4 Staff main menu



Note: Main menu will show up after staff scan the ID card. Click on an option to go to that page.

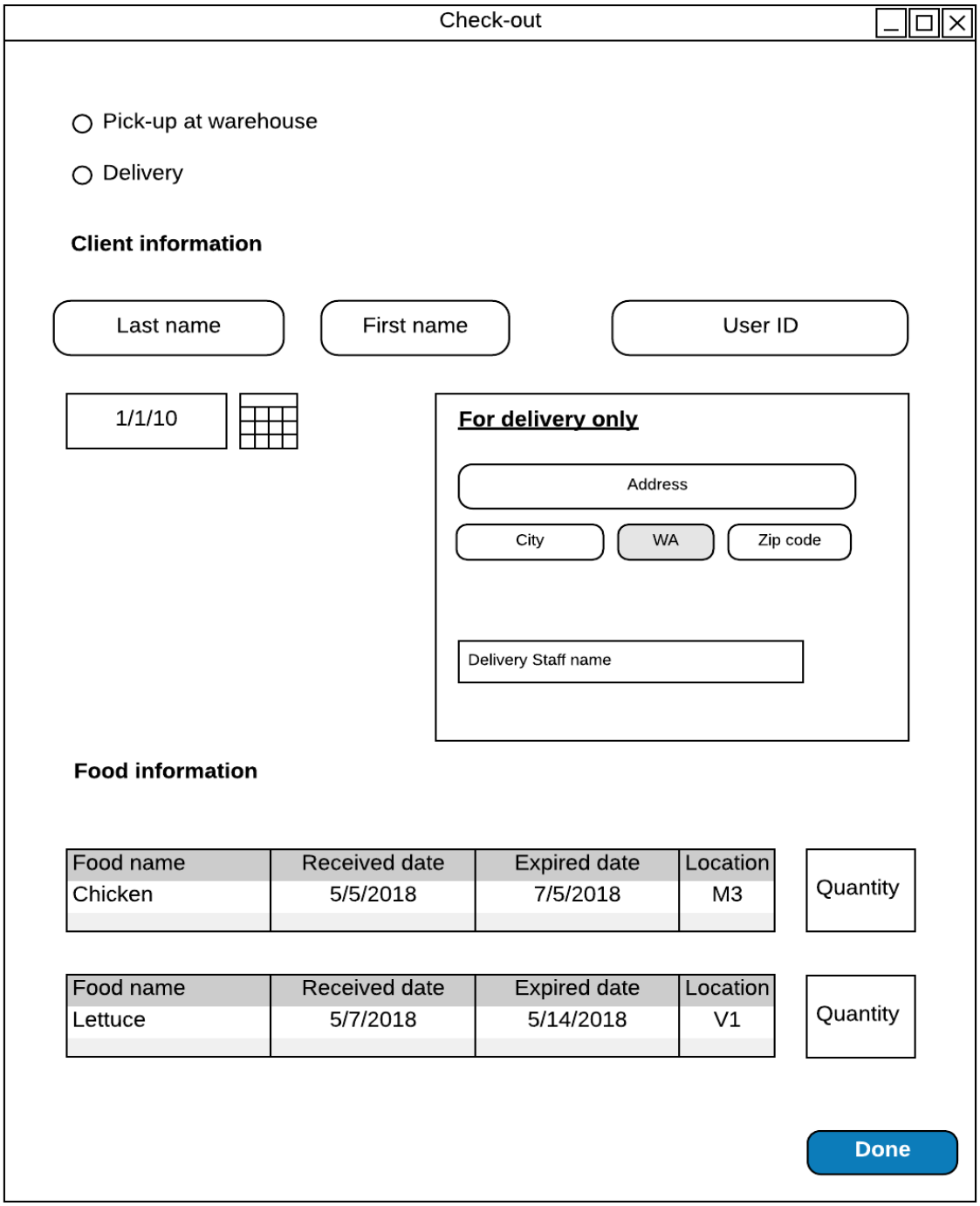
4.3.5 Check-in staff receive donation



4.3.6 Screen displayed when scan the food sticker

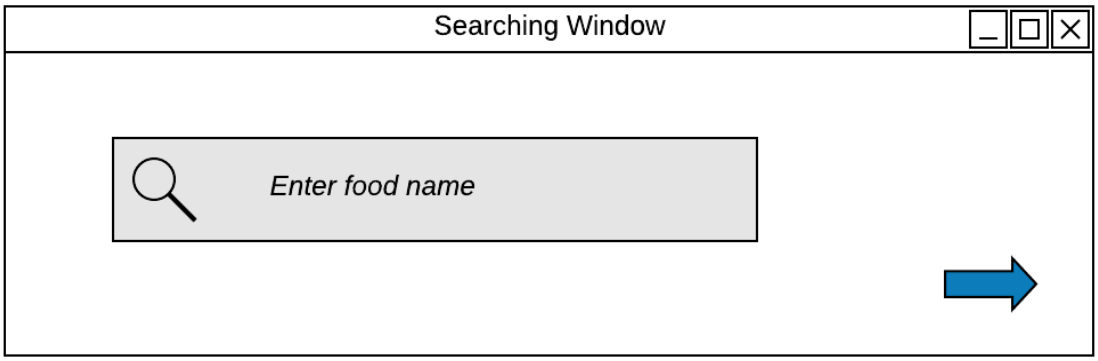
Food name	Received date	Expired date	Location
Chicken	5/5/2018	7/5/2018	M3

4.3.7 Check-out food screen



Note: Client information and Food information will automatically display when staff scan the user ID card or food sticker. If client does not have an account, staff input client information manually.

4.3.8 Search for food information

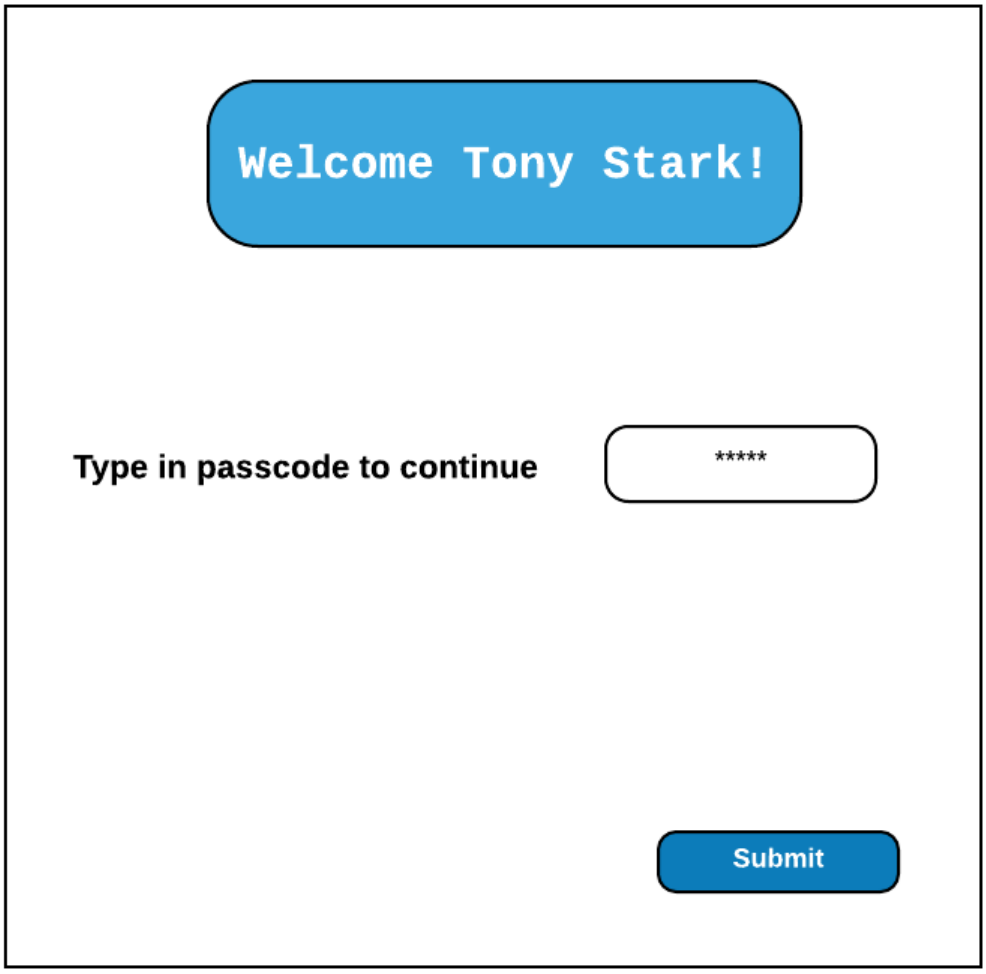


4.3.9 Search result

Click on the arrow symbol or simply hit Enter button, the screen will show as below:

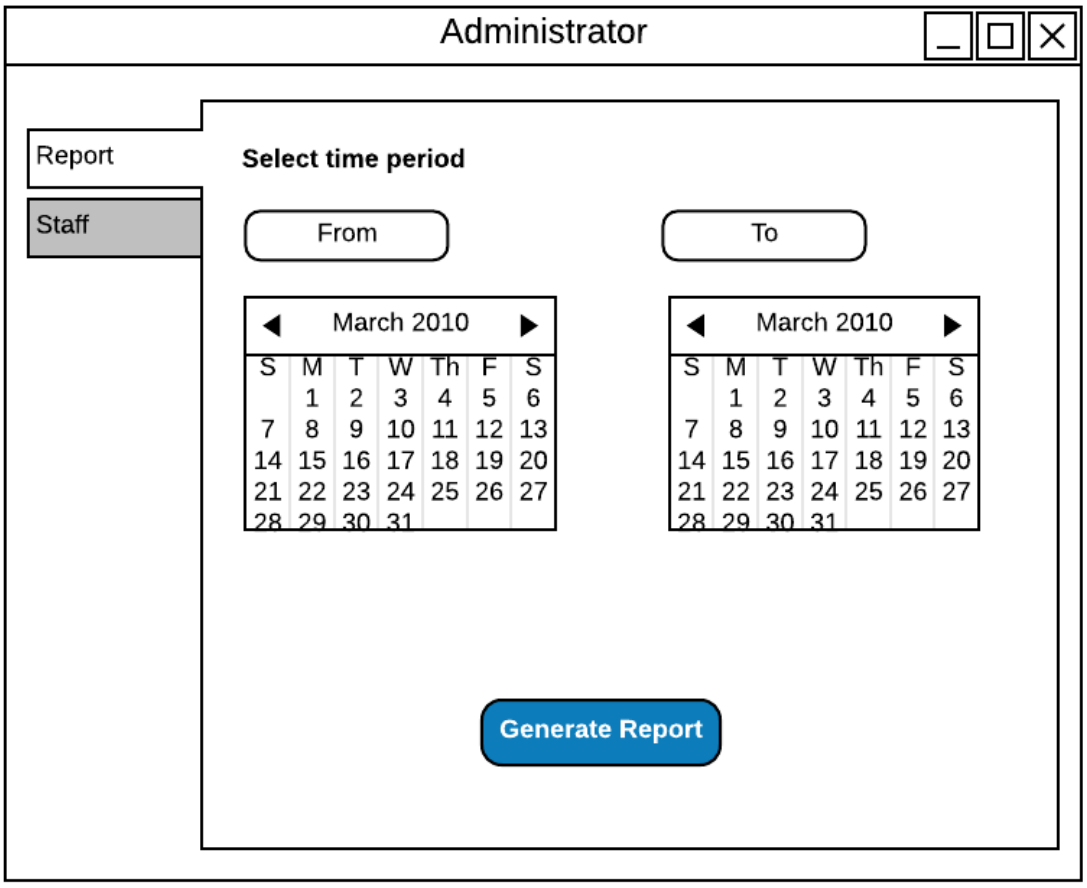
Food name	Quantity	Expired date	Location
Chicken	50 boxes	7/5/2018	M3

4.3.10 Administrator Main Window



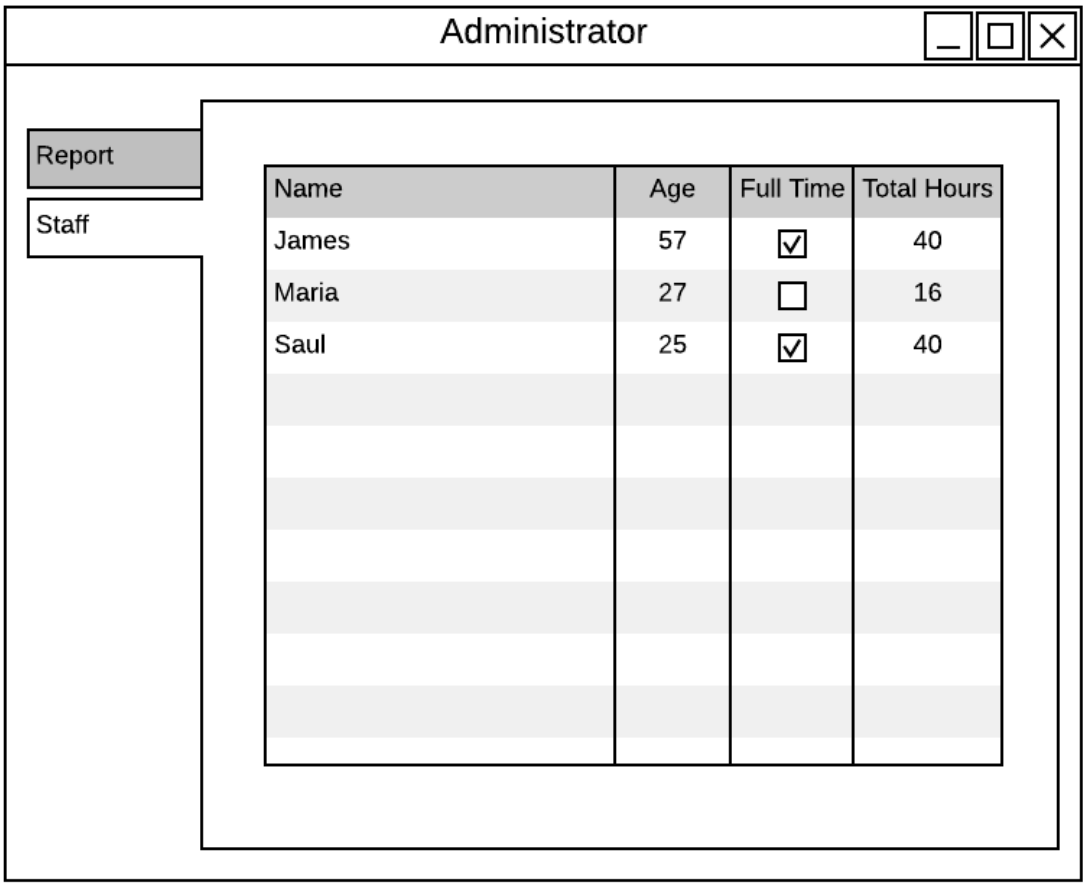
Note: This window will display after Admin scan the ID card.

4.3.11 Report Page



If user clicks “Generate Report”, a printable report is made (see section 4.4 for outline).

4.3.12 Staff page



4.4 Report: “Printed Output” Design

5000 Foods
Report
<Time Period of Report>

DONATION:

CASH TOTAL: $5000.00

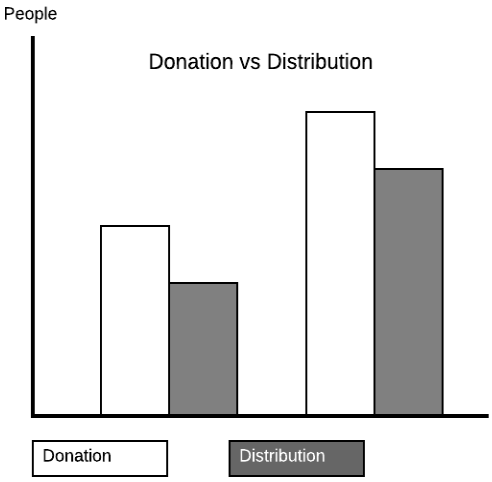
Name	Date	Item	Quantity
Maria	6/6/2018	Beef	5 lbs
Saul	6/1/2018	Broccoli	2 lbs

FOOD DISTRIBUTION

Name	Date	Item	Quantity
Unregistered Client	6/6/2018	Chicken	1 box
MATT	6/1/2018	Broccoli	2 bags

FOOD STORAGE
(Sample)

Item	Quantity	Location	Expired date
Salad	25 boxes	V1	6/19/2018



5 Appendices

Glossary

5.1 Bibliography

CSC 3150 Sample Book, CSC 3150. Seattle Pacific University, Seattle, Washington. 2016.

Dennis, Alan, Barbara Haley Wixom, and David Paul Tegarden. *Systems Analysis Design: An Object Oriented Approach with UML*, Fifth Edition. Hoboken, NJ: John Wiley & Sons, 2012. Print.

Pfeiffer, William S. *Pocket Guide to Technical Communication*. Upper Saddle River, NJ: Prentice Hall, 2011. Print.

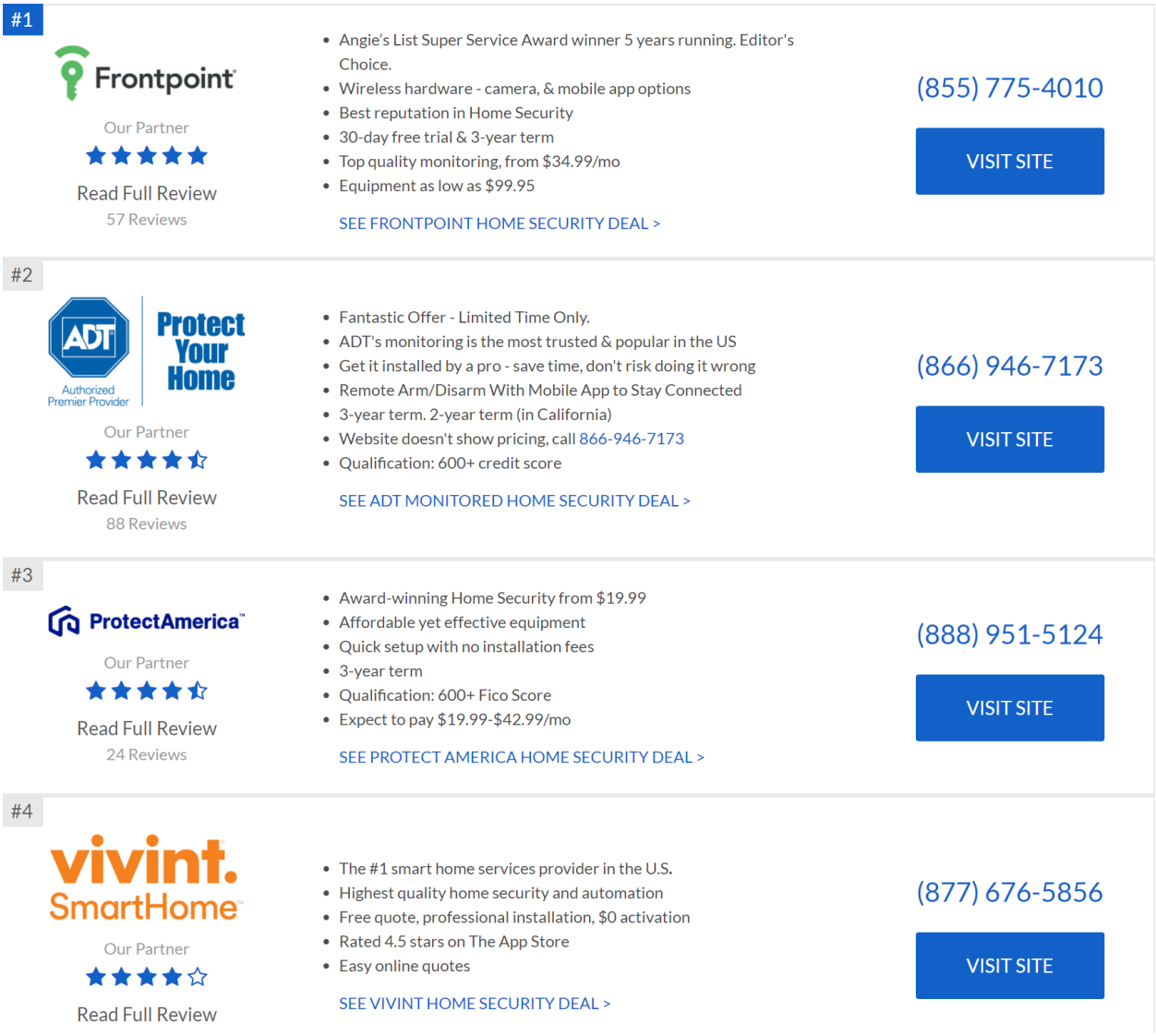
Weltz, Elaine (2016). Systems Design, various lectures [PowerPoint slides/Word Documents]. Retrieved from Professor Weltz and http://canvas.spu.edu.

Cuisine by Car System (CBCS) Proposal, Cuisine by Car Proposal - 1.pdf

Security system: https://www.consumersadvocate.org/home-security/a/best-home-security?matchtype=p&keyword=security%20systems&adpos=1t3&gclid=Cj0KCQjwjN7YBRCOARIsAFCb9368m49JVMYizyjpjPTujBvw-R0jyC0TZKHFNtn8B7yXAK3gZNTEJ0saAuq9EALw_wcB

Internet router: https://bestreviews.com/best-wireless-routers

5.2 Supporting documentation



(See citation, “Security system”)

	BEST OF THE BEST				BEST BANG FOR THE BUCK
					
	ASUS Wireless Dual Band Check Price amazon.com Check Price at 1 store	**TP-LINK** Wireless N300 Check Price amazon.com Check Price at 1 store	**Actiontec** 300 Mbps Wireless Check Price amazon.com Check Price at 1 store	**TRENDnet** Wireless AC1900 Check Price amazon.com Check Price at 1 store	**NETGEAR** Nighthawk AC1900 Check Price amazon.com Check Price at 1 store
BOTTOM LINE	A popular, reliable router that tops our list for all it has to offer: fast WiFi, reliable connectivity, easy use.	An acceptable choice if budget is your top concern, but others have more to offer.	You may not need its modem feature or have compatible internet service, but if you meet the criteria, it's a reliable choice.	With a satisfactory range and fairly reliable consistency, this router provides quality at an affordable price.	A feature-packed device that provides great value for the money.
PROS	Incredible range and speed. Suitable for online gaming and rapid file sharing.	Inexpensive and easy to set up. A consistent performer.	A router/modem combination. Pre-configured settings auto-connect for easy setup. Owners rave about the long range.	Intuitive setup wizard. Great for HD video streaming. Includes options for blocking and guest networking.	Stable connection and excellent speed. Fast setup. Functions can be controlled remotely from another device.
CONS	Expensive. Some complaints about customer support.	No USB ports. Some complaints about slower WiFi speed.	Not compatible with all internet providers, which eliminates a lot of customers.	Some complaints of dropped WiFi signals and QoS confusion.	Expensive, but its performance is worth the price.

See citation, “Internet Router”