E.A.S.
(Efficient And Simple)

Detailed Design

Requested By:

Ms. Mary Partridge-Brown and Ms. Roberta Sandler
Co-Directors
Grassroot Givers
Albany, NY

Prepared By:

Maroon Solutions
Kathleen Rotondo: Assistant Developer
Mathew Banville: Data Analyst
Kyle Flack: Database Administrator
Marissa Gasparro: Team Lead
Kean Smullen: Head Developer

March 9th 2015

Table Of Contents

1. Project Summary and Overview

Our clients, Ms. Mary Partridge-Brown and Ms. Roberta Sandler, want a program that will help them during their day-to-day business operations of running their store. Maroon Solutions is going to help them by creating an inventory database, as well as a database of customers, donors, and employees' profiles. We will differentiate rights from admins and employees by giving admins more power to add employees to the system and the right to change prices of items in the inventory. The client wants to make sure that the process of adding items and customers/donors to the system is smooth and easy to enter data. The employees are able to enter data about new customers/donors regarding name, phone number, and address. When accepting donations the employee will also be able to print out a receipt for the donor for tax return purposes. This will allow Grassroots Givers to see who they are helping and where across the capital region they are helping people.

2. User Case Narratives

2.1. Employee

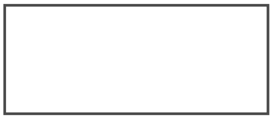
A user will go to the systems site and log on as an employee. This employee will be taking donations from a donor. If the donor is a new donor the employee can create a new donor profile. The employee will add each item into the donated inventory with the specified donor id and print a list of the items as a receipt for the donor. The employee then adds processed items to the stores inventory. The employee then can sell items to a customer. The employee can create a new customer profile and add items that were purchased by a customer to a customer's profile. The employee can search through a customers purchase history. The employee can also create, add, edit, or delete donor and customer profiles.

2.2. Administrator

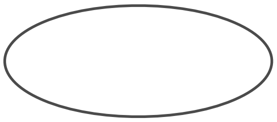
A user will go to the systems site and log on as an administrator. The administrator has all of the rights of an employee. Additionally, the administrator can create, add, delete or edit other administrator or employee accounts. The administrator can edit any inventory item values, including the appraised value of any bag not yet processed into the store inventory.

3. UML Diagrams

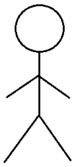
3.1. UML Use Case Legend



System Boundary: This is where all the interaction occurs. Represents what is within the system and outside of it. Scenarios go on the inside and actors go on the outside.

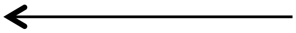


Scenarios: The actions that occur within a system and how the user interacts with the system.

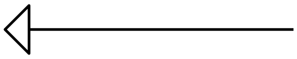


Actor: Actors interact with the system through scenarios. Actors can be human or non-human. Human actors go on the left side of the system boundary. Non-human actors go on the right side. Actor name gets displayed below the actor.

Participation Line: Shows what scenarios an actor can interact with.

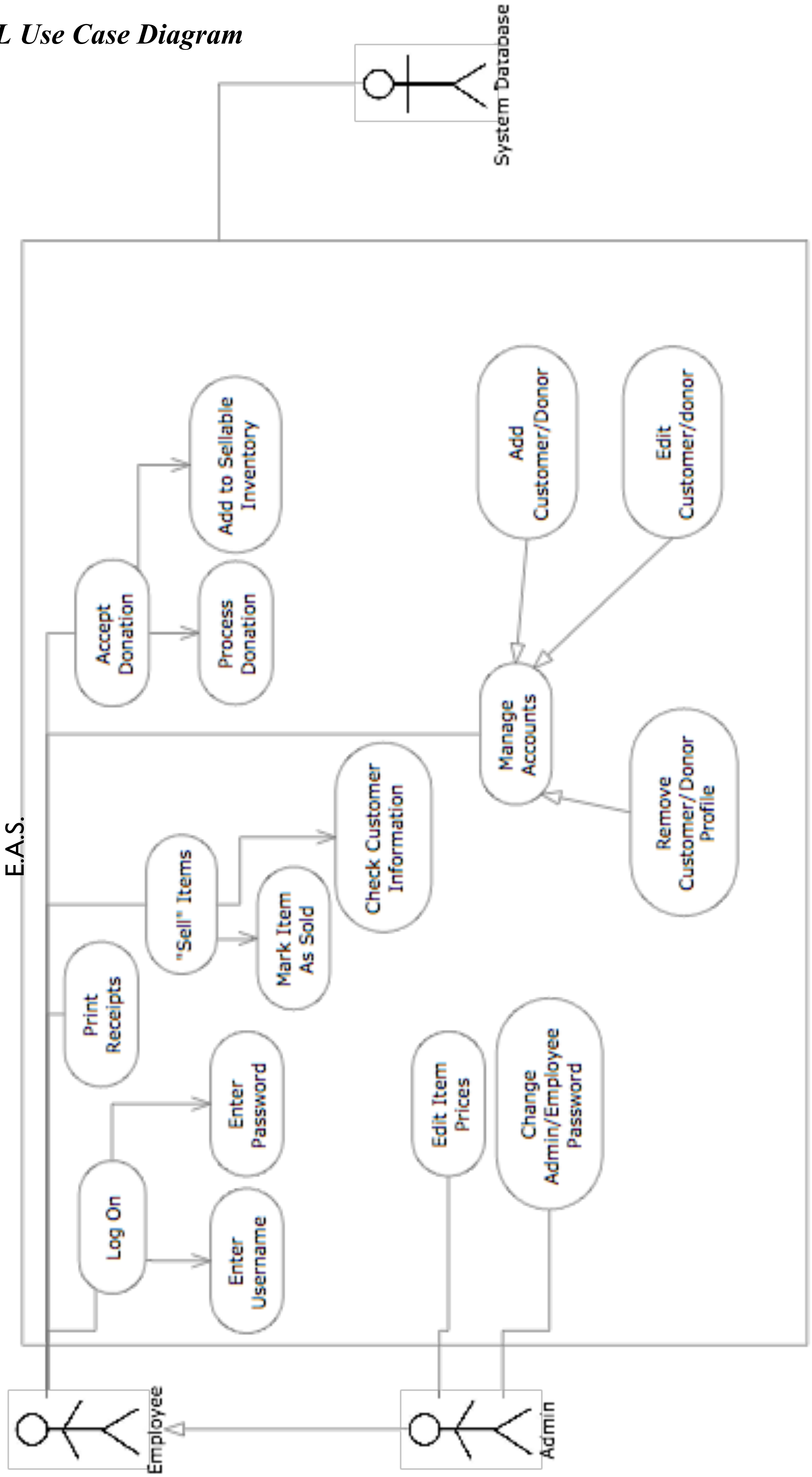


Inclusion Arrow: An arrow that points from a scenario to another scenario to show that something must be included for the scenario.



Inheritance Arrow: An arrow that points from one use to another. The use being pointed at is the parent and the other is the sub.

3.2. UML Use Case Diagram



3.3. Deployment Diagram Legend

A deployment diagram in the Unified Modeling Language (UML) models the physical deployment of devices and execution environments for a system. The E.A.S. Deployment Diagram represents the physical design of the system. The diagram shows that E.A.S. will be connected to the internet via HTTP. The database that is used will be connected to E.A.S. via ODBC.

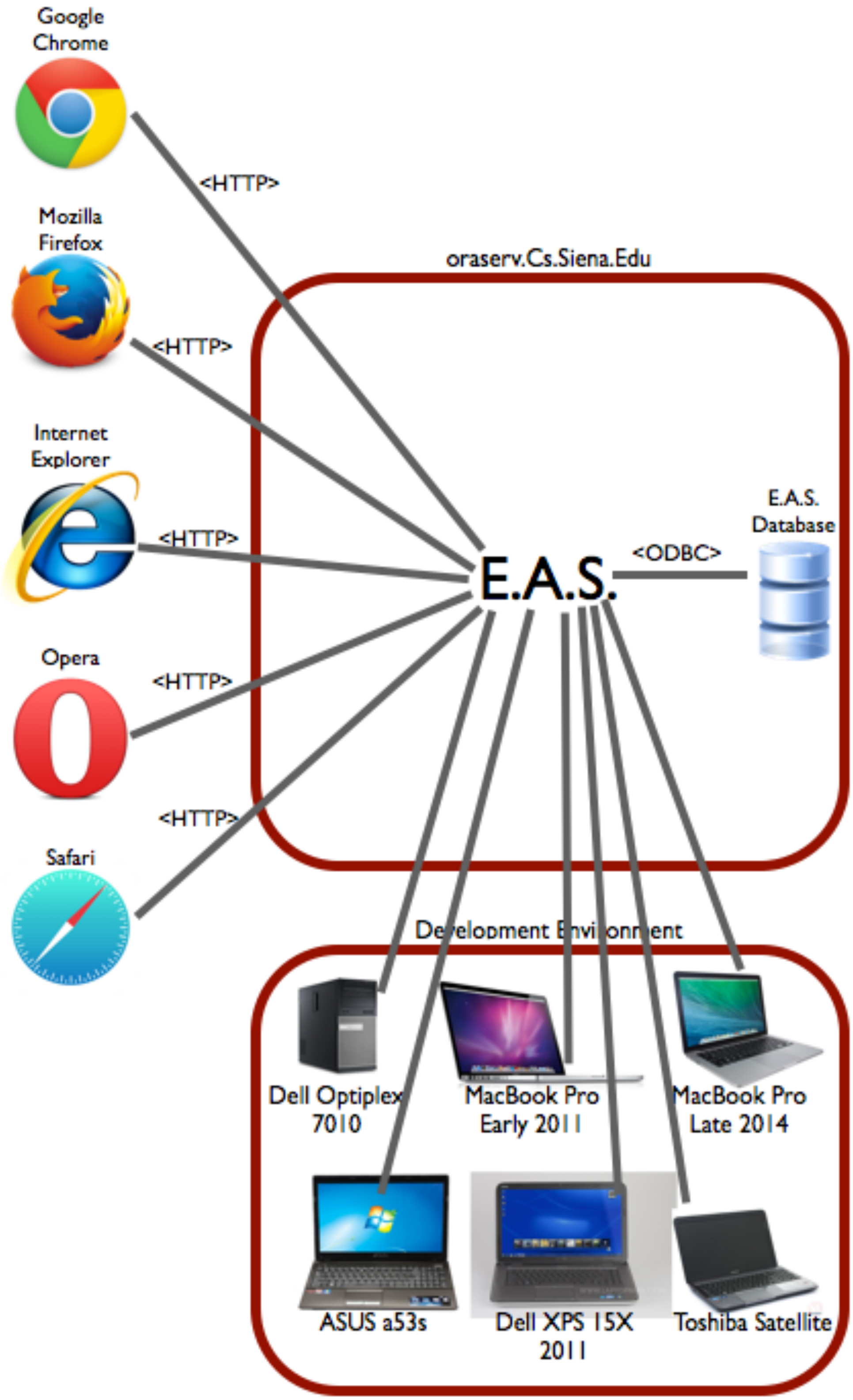
□ **System Boundary -** This is where all the interactions occur. Represents what is within the system and outside of it.

<HTTP> **HTTP -** Hypertext Transfer Protocol defines how messages are formatted and transmitted, and what actions web servers and browsers should take in response to various commands.

<ODBC **ODBC -** Open Database Connectivity is a standard programming language middleware for accessing database management systems.

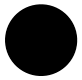
▬ **Connection -** Displays a relationship between boundaries.

3.4. Deployment Diagram

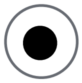


3.5. Activity Diagram Legend

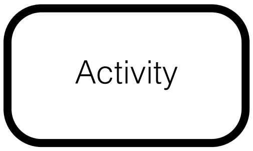
A UML Activity Diagram is a step-by-step diagram of the actions within a process. Each step will start the process, end the process, perform an action, or make a decision. The activity diagrams can also display the objects that will be needed and will be given to users of the process. Below are the nodes and other objects used in the UML Activity Diagrams for E.A.S.'s main processes.



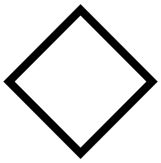
Initial Node – This is the first node in the process. The initial node is the starting point for all movement.



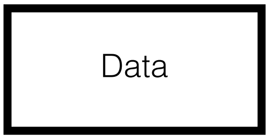
Final Node – The final node is the last node in the process. When the activity flow has reached here, the process is over.



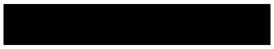
Activity Node – The activity node describes the activity or step to be done to help complete the process when the flow reaches the node.



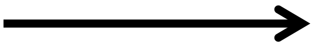
Decision Node – The decision node is used to branch the activity. A decision node is usually posed in question form, with multiple unique answers. The flow must follow one of the branches after the decision.



Data Object - Data that is used during the process. The data object can be either input or output.

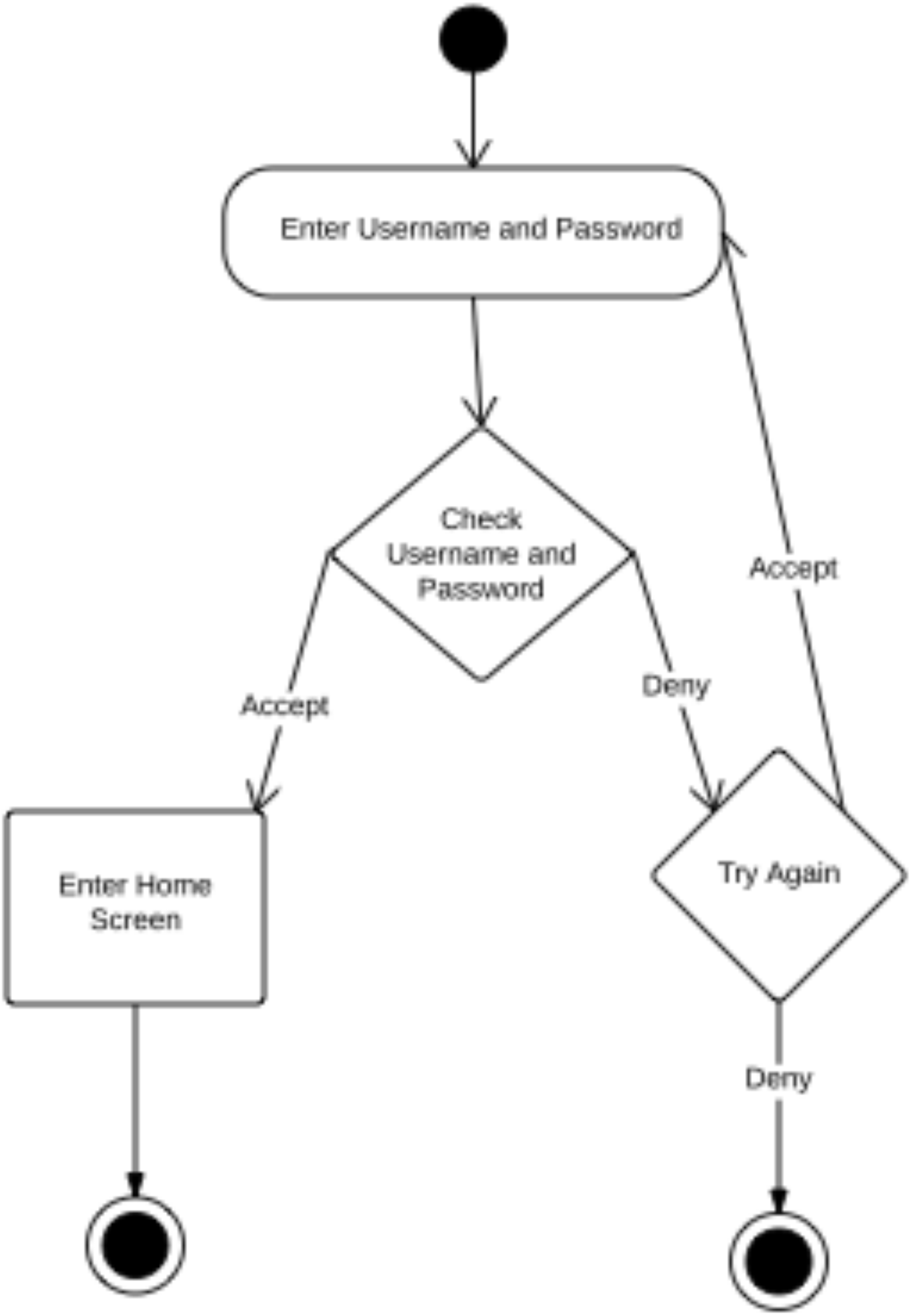


Split/Join – Can either separate activity flow to run two activities simultaneously or join them back together after simultaneous activities are completed.

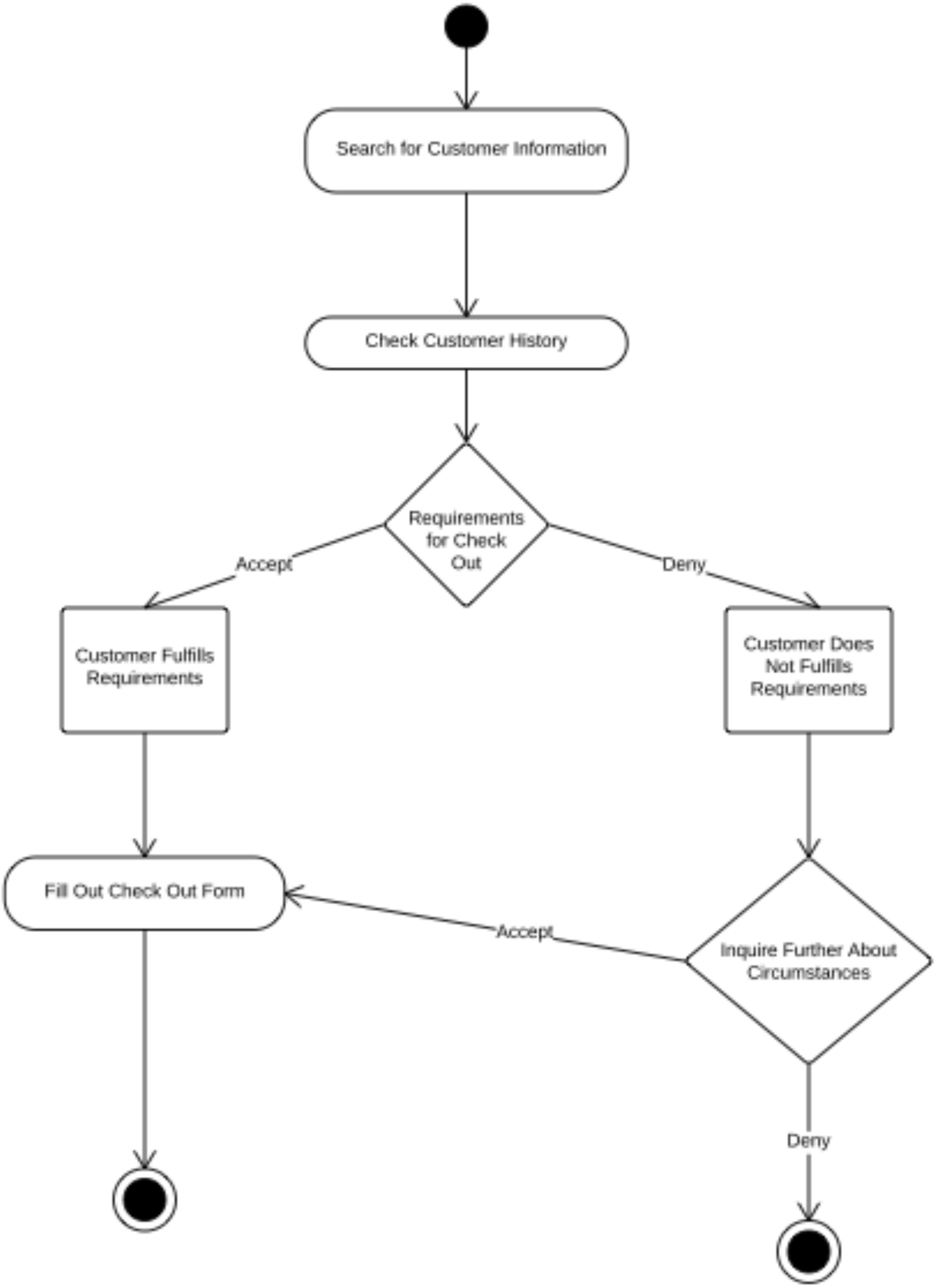


Flow – Shows the movement of action from one node to another

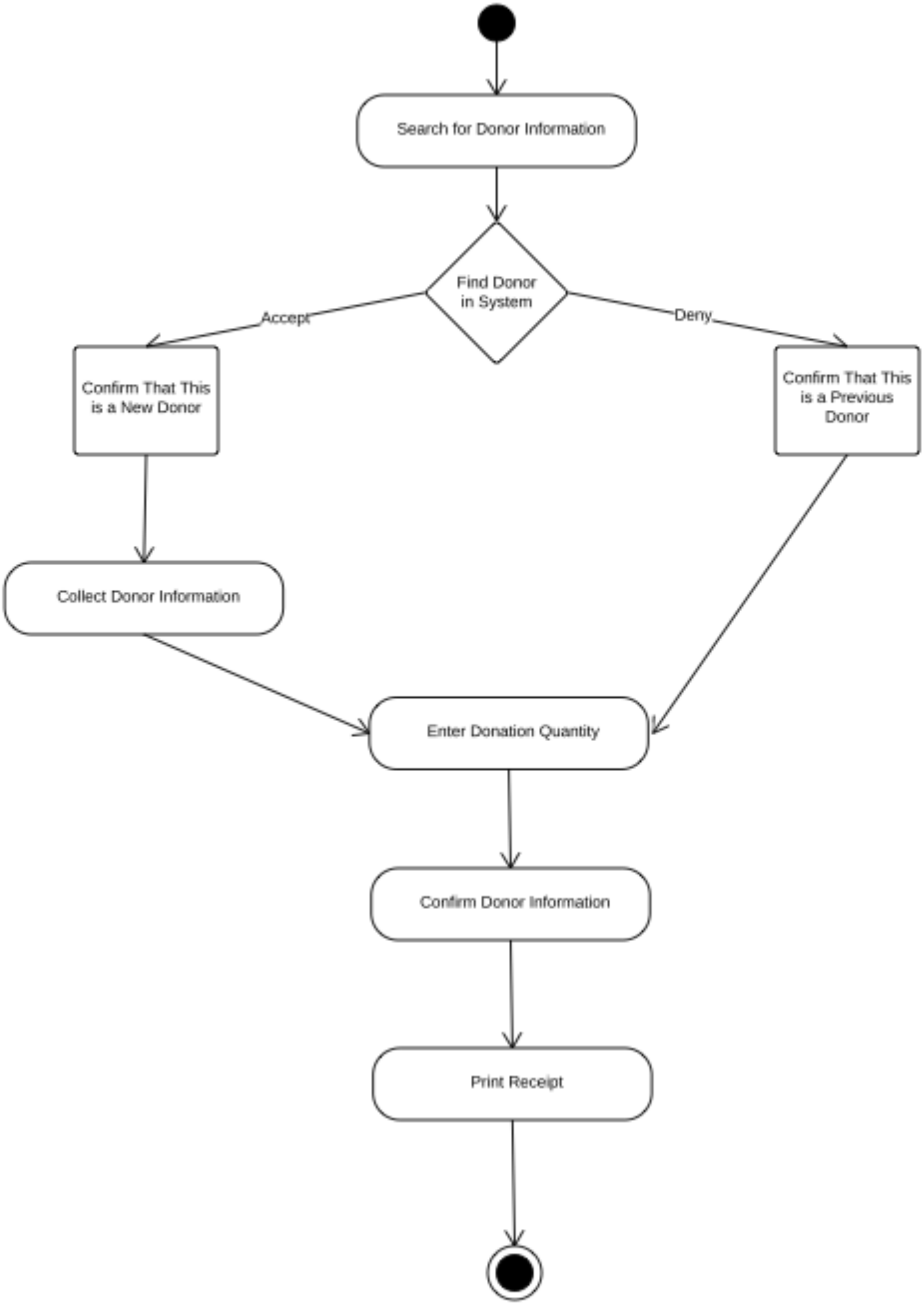
3.6. Activity Diagram: Log On



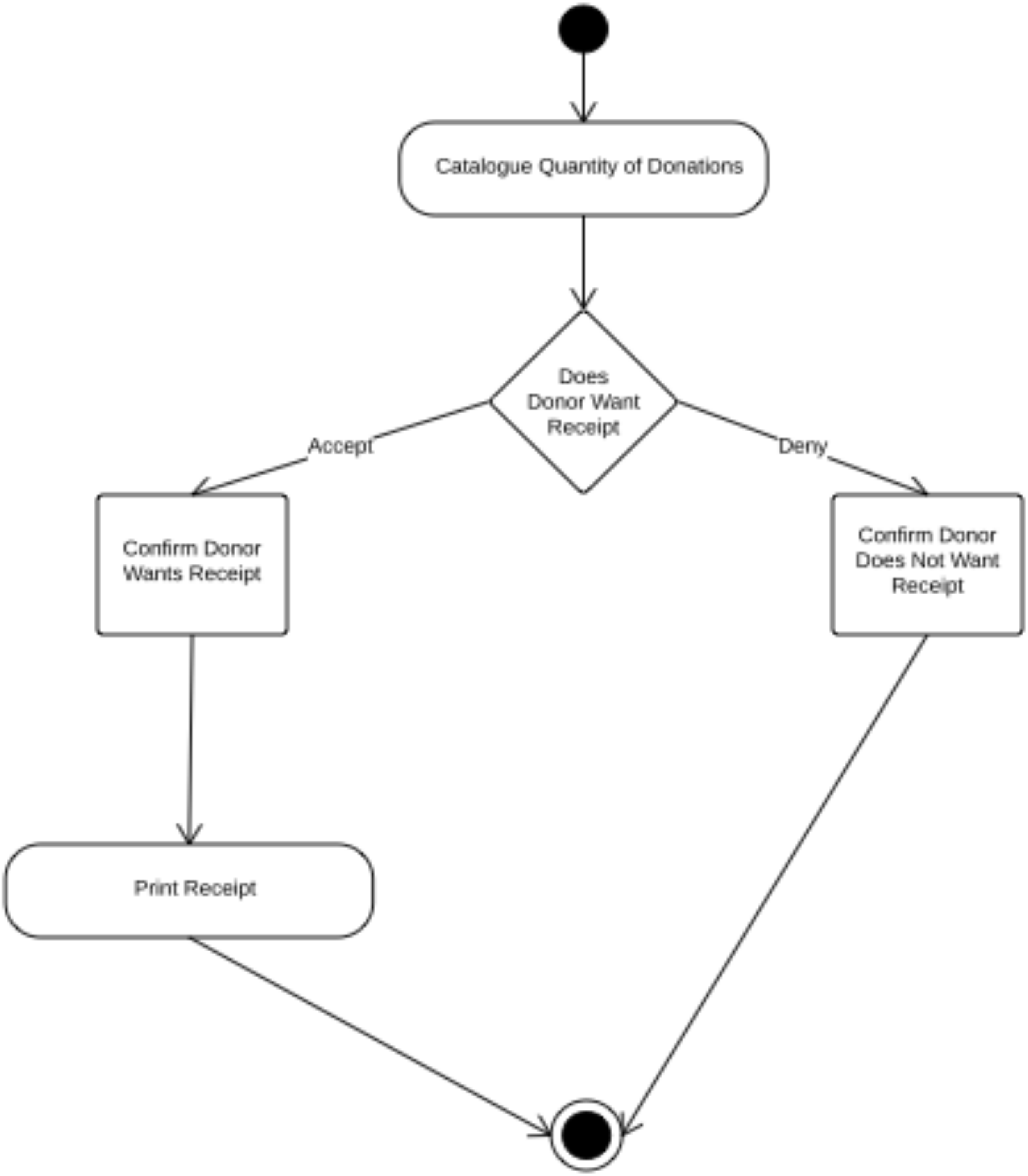
3.7. Activity Diagram: Sell Items


Search for Customer Information
Check Customer History
Requirements for Check Out
Accept
Deny
Customer Fulfills Requirements
Customer Does Not Fulfills Requirements
Fill Out Check Out Form
Accept
Inquire Further About Circumstances
Deny

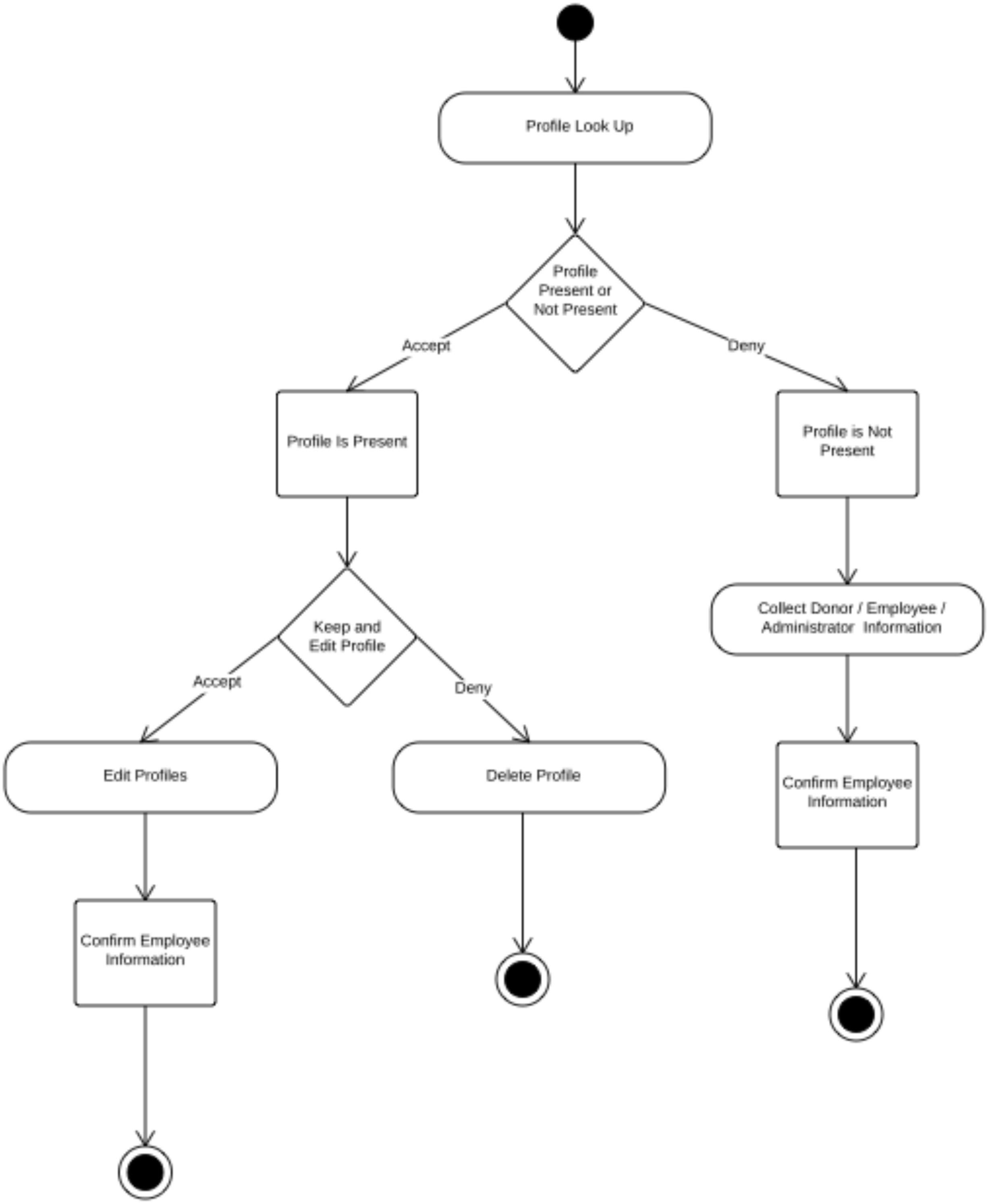
3.8. Activity Diagram: Print Receipts


Search for Donor Information
Find Donor in System
Accept
Deny
Confirm That This is a New Donor
Confirm That This is a Previous Donor
Collect Donor Information
Enter Donation Quantity
Confirm Donor Information
Print Receipt

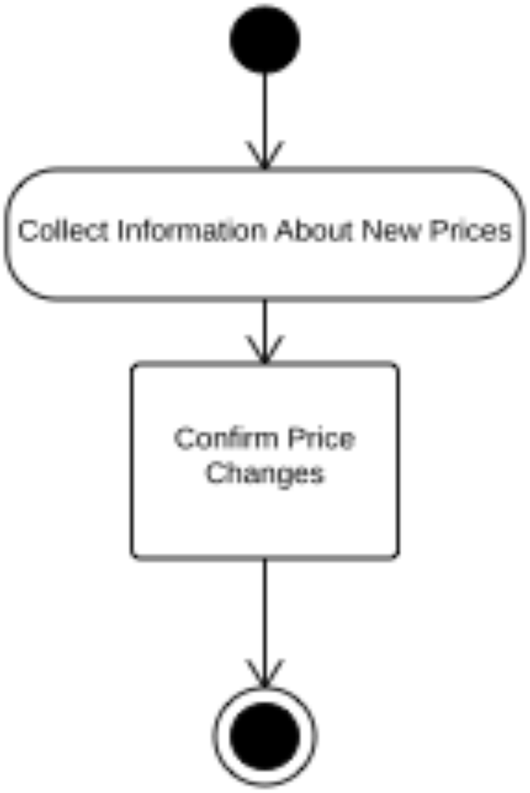
3.9. Activity Diagram: Accept Donations


Catalogue Quantity of Donations
Does Donor Want Receipt
Accept
Deny
Confirm Donor Wants Receipt
Confirm Donor Does Not Want Receipt
Print Receipt

3.10. Activity Diagram: Manage Profiles

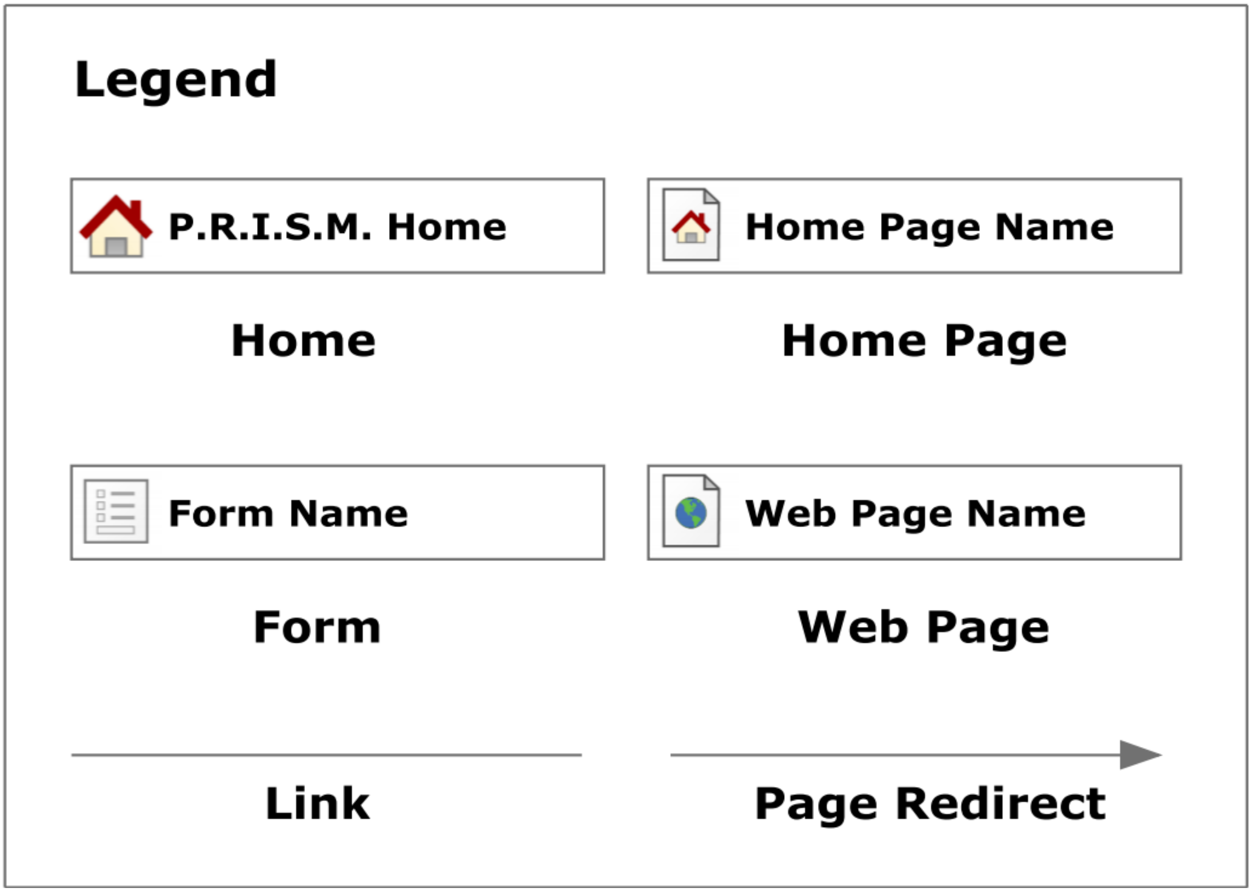

Profile Look Up
Profile Present or Not Present
Accept
Deny
Profile Is Present
Profile is Not Present
Keep and Edit Profile
Collect Donor / Employee / Administrator Information
Accept
Deny
Edit Profiles
Delete Profile
Confirm Employee Information
Confirm Employee Information

3.11. Activity Diagram: Edit Prices



3.12. Website Map Legend

The web site map shows the structure of the E.A.S. website. The map outlines the links and forms associated with each as well as the functionality of these elements. Below are the various symbols associated with our web site map.



Home - This represents the main page of E.A.S. prior to a user logging in. Various options will be presented at this time.

Home Page - Represents the user’s main page when they initially login to their account.

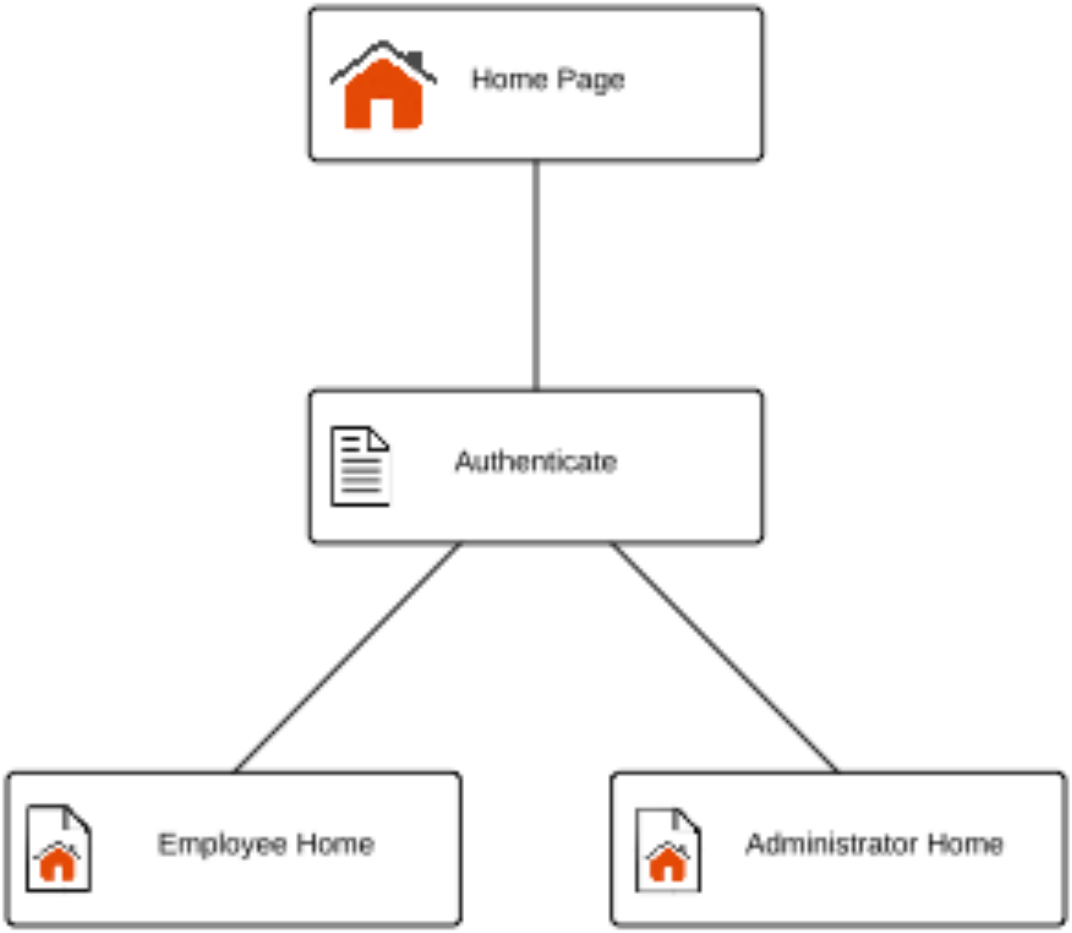
Form - Represents data fields that require user input.

Web Page - Represents a web page within the E.A.S. system.

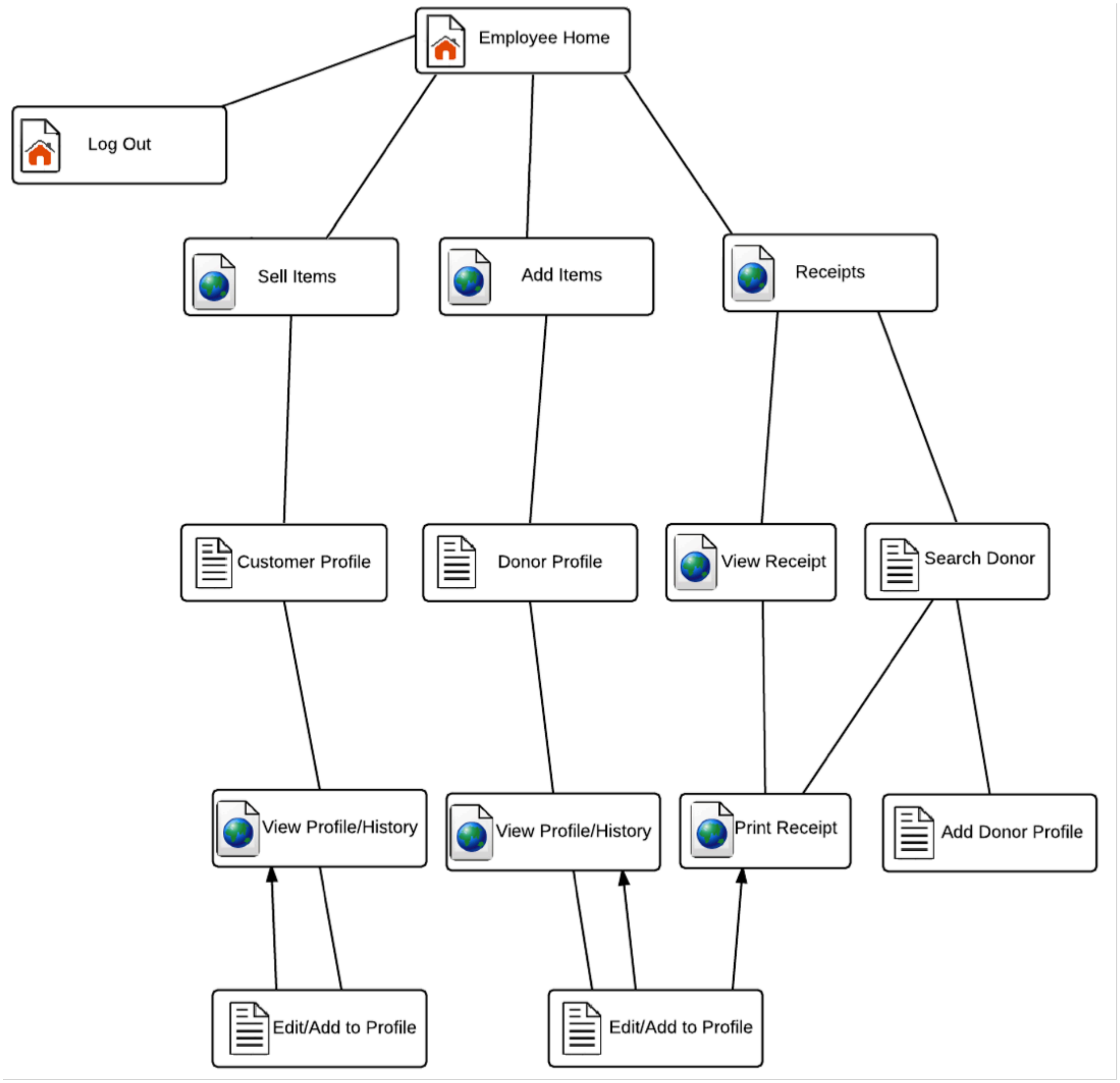
Link - Represents a page being accessible from another page.

Page Redirect - Indicates a forced reroute to a new page depending on the user’s action

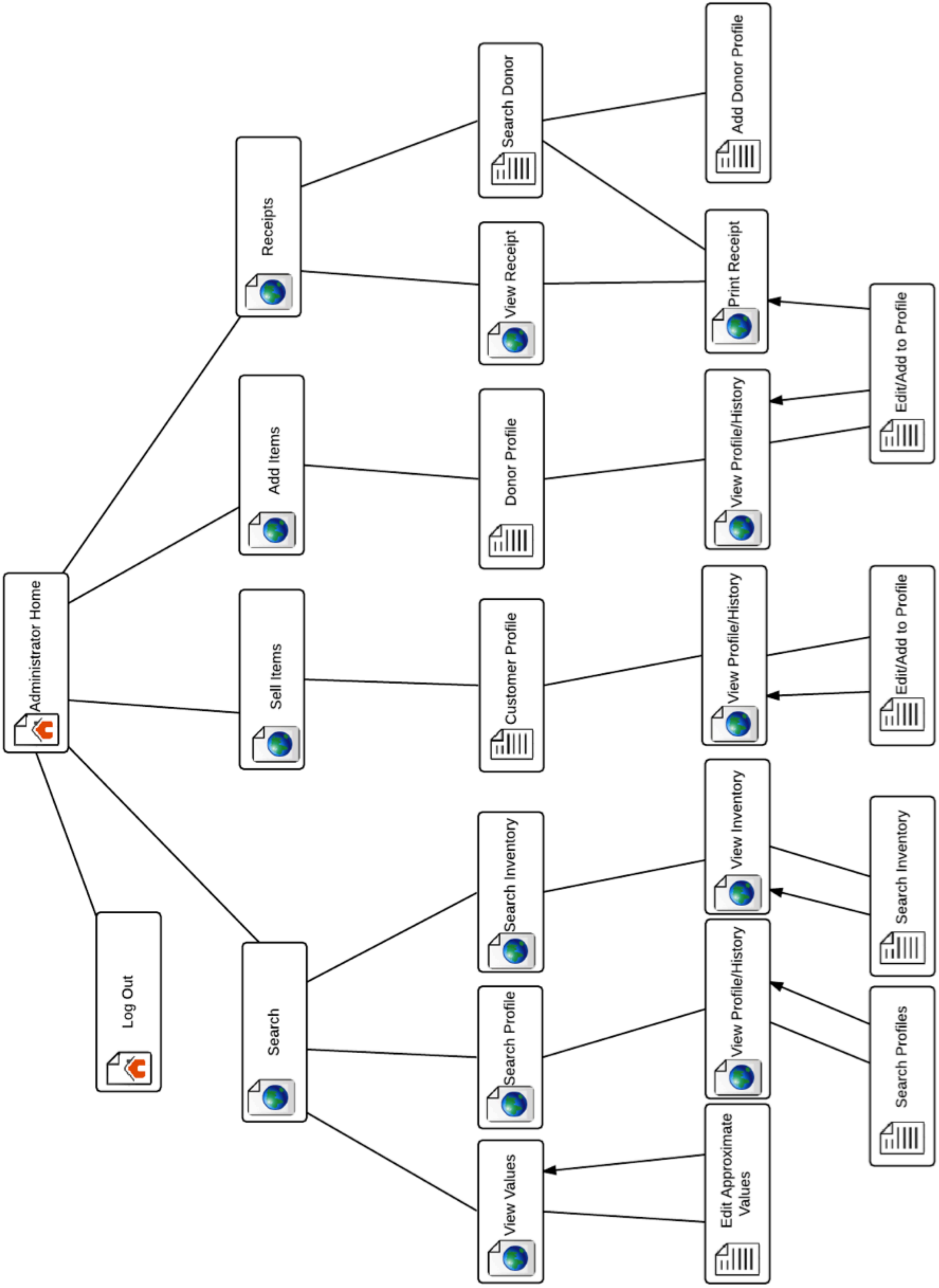
3.13. Website Map: Context



3.14. Website Map: Employee



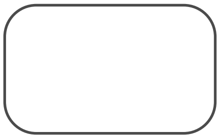
3.15. Website Map: Administrator



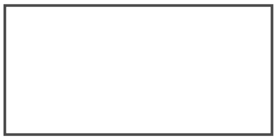
4. Data Flow Diagram

Data Flow Diagrams represent the movement of data between processes in the system as well as the movement of data between processes and external entities outside the system. The diagrams are a tool for analyzing the structure of the system and the ways in which data will be stored and retrieved by different processes. These diagrams model data flows at different levels of detail in the system. The following symbols will be used within the Data Flow Diagram:

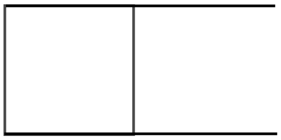
4.1. Data Flow Diagram Legend



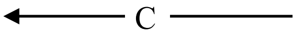
Process: Transforms or manipulates data.



External Entity: Contributes data or information to the system or which receive data/information from it.

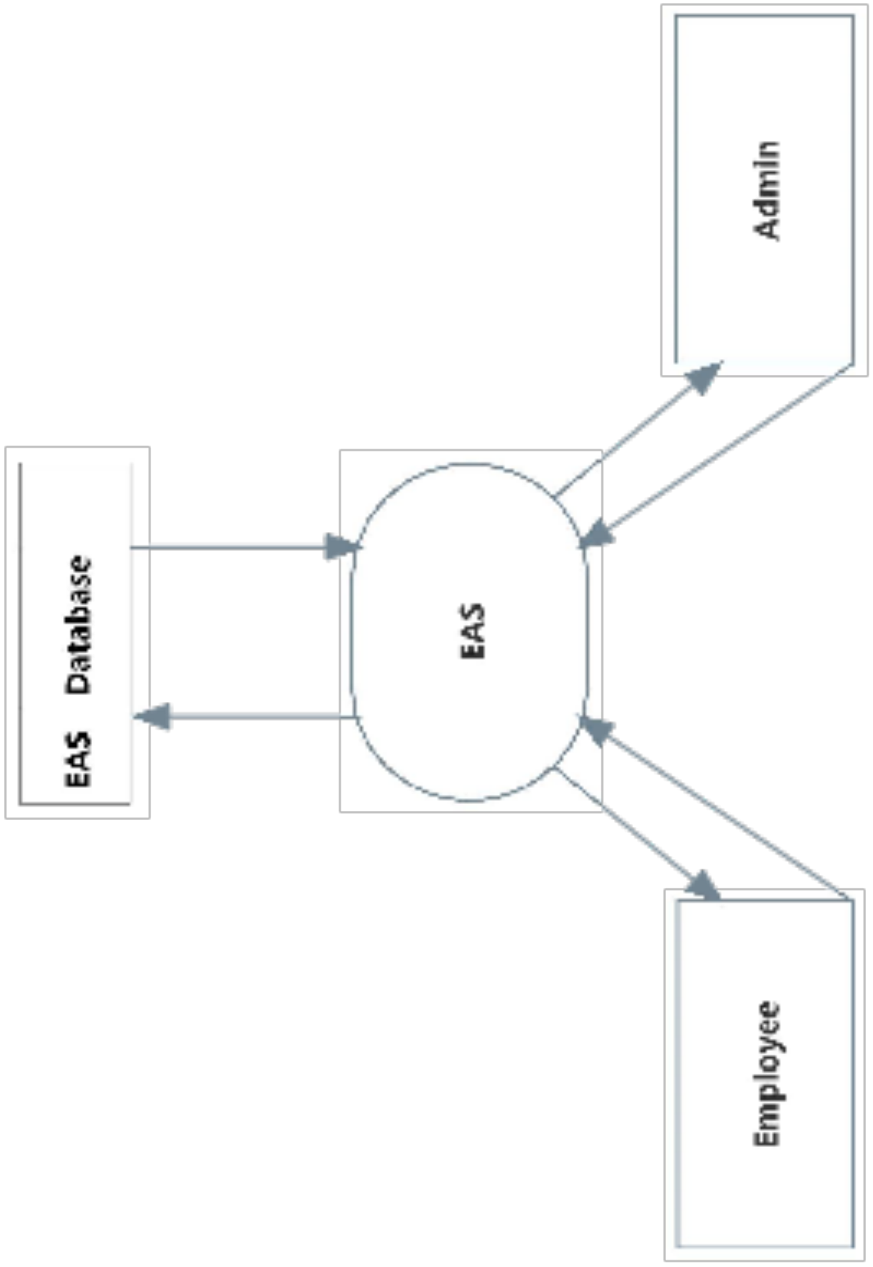


Data Store: Location where data is held temporarily or permanently.

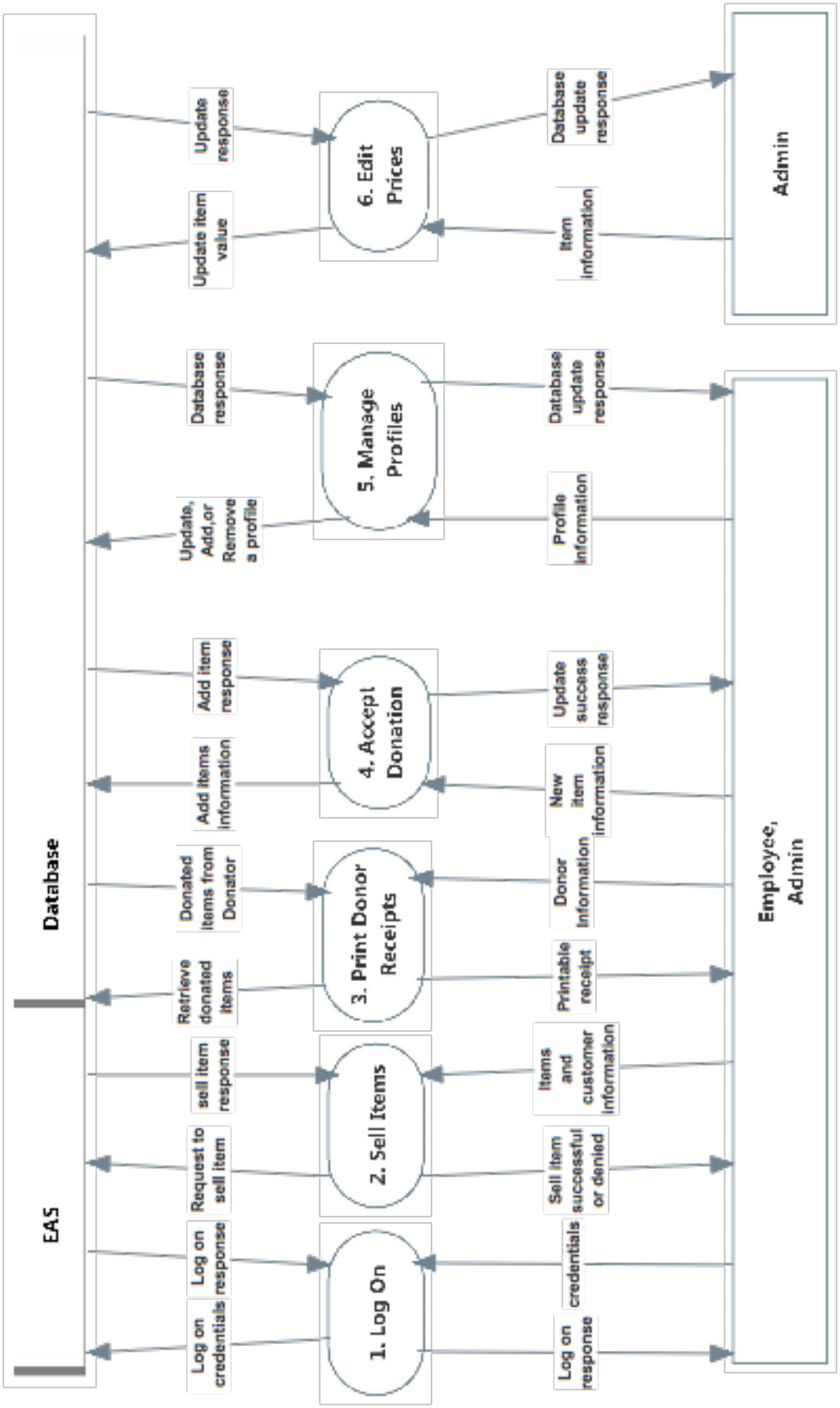


Data Flow: Data/information flowing to or from a process where C is the data/information.

4.2. Context Diagram

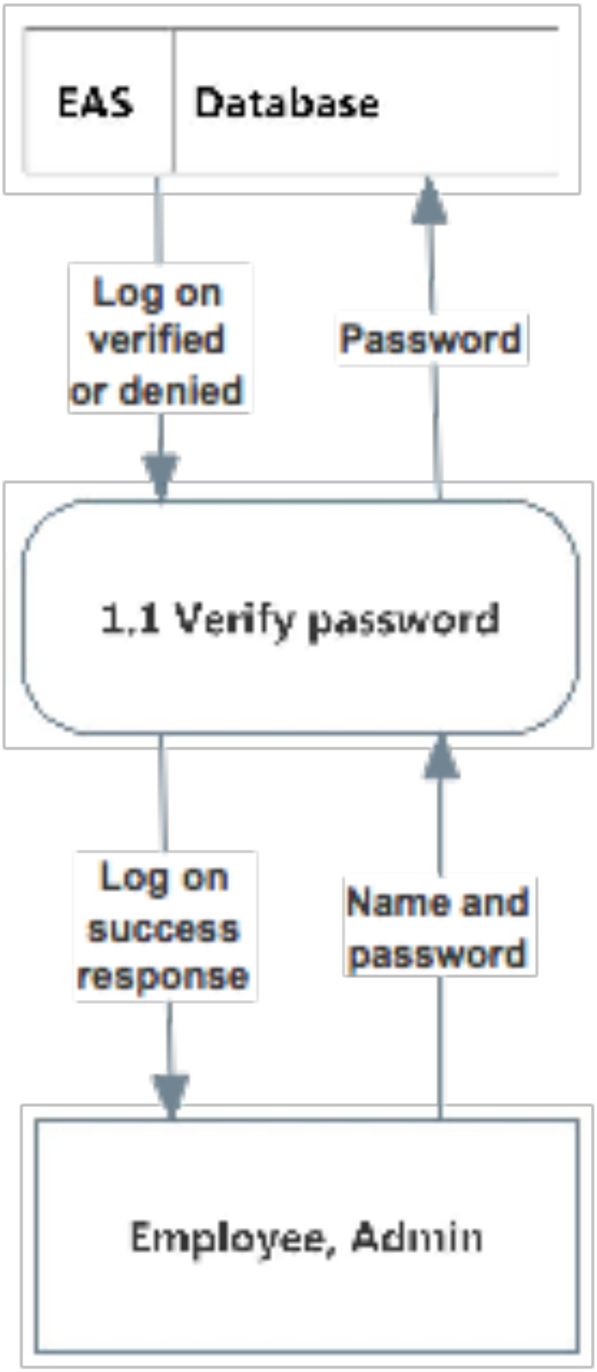


4.3. Level 0 Diagram

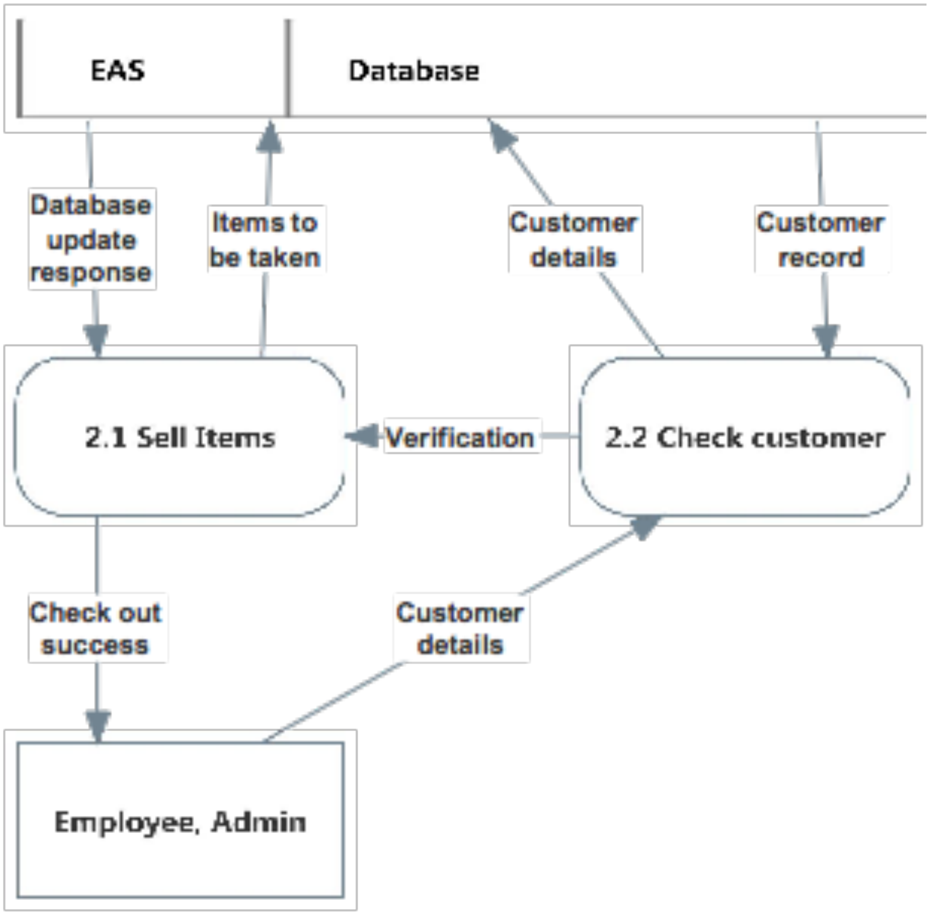


4.4. Level 1 Diagrams

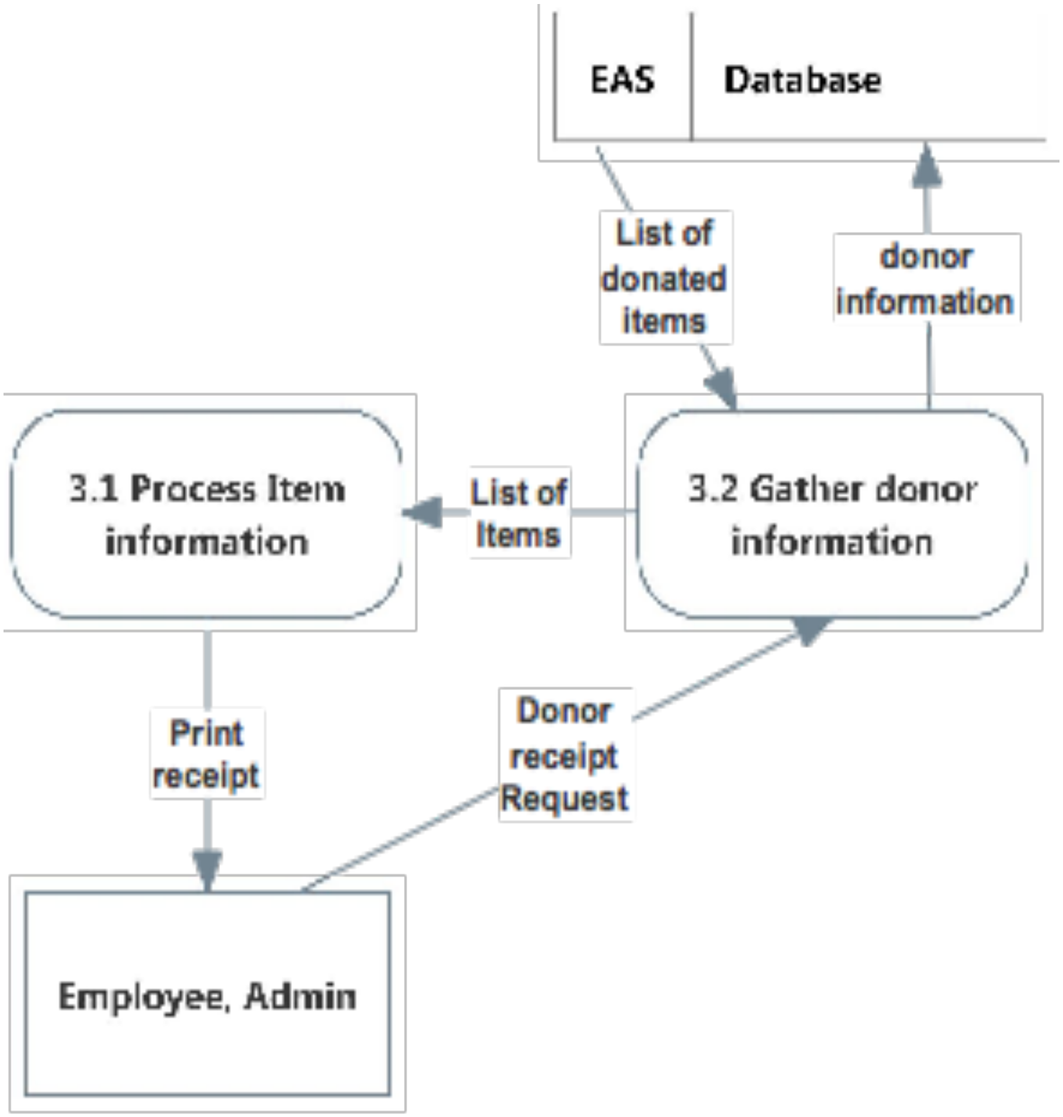
4.4.1. Log On



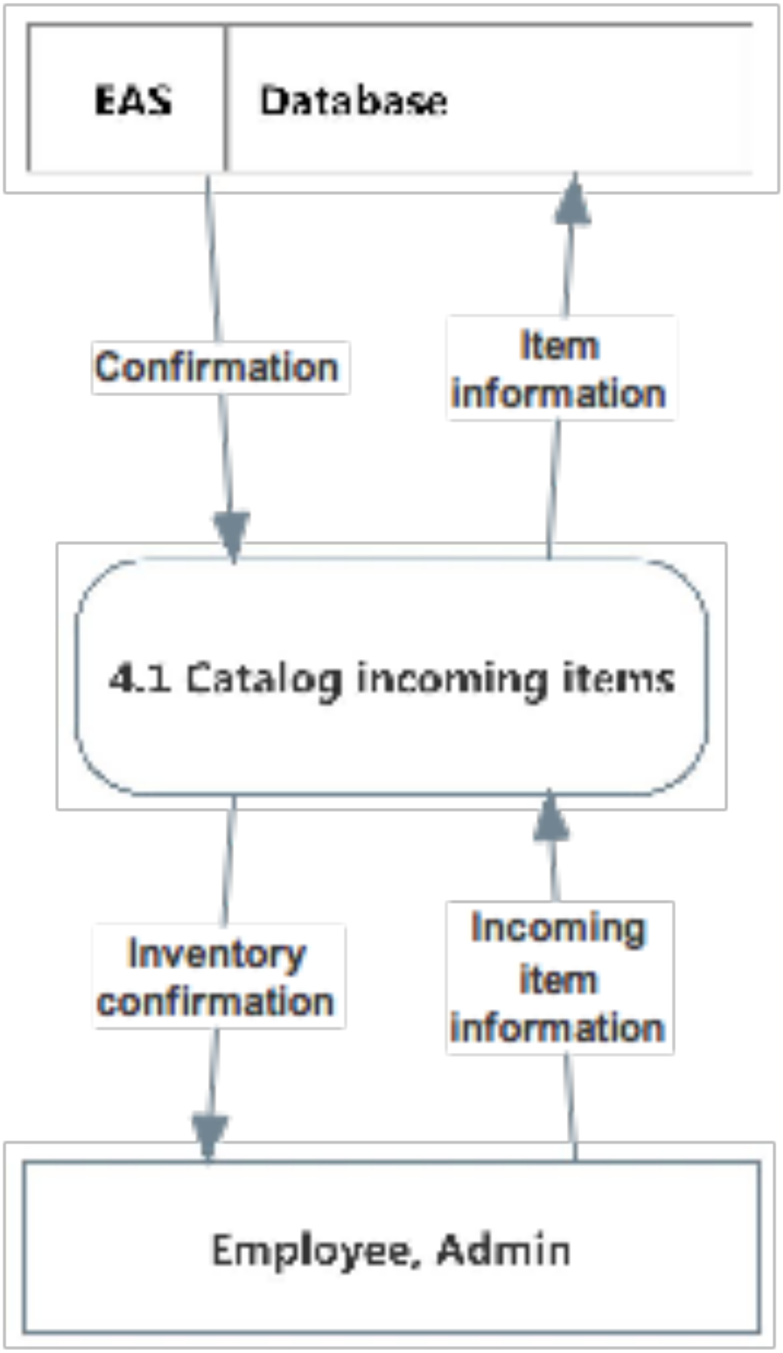
4.4.2. Sell Items



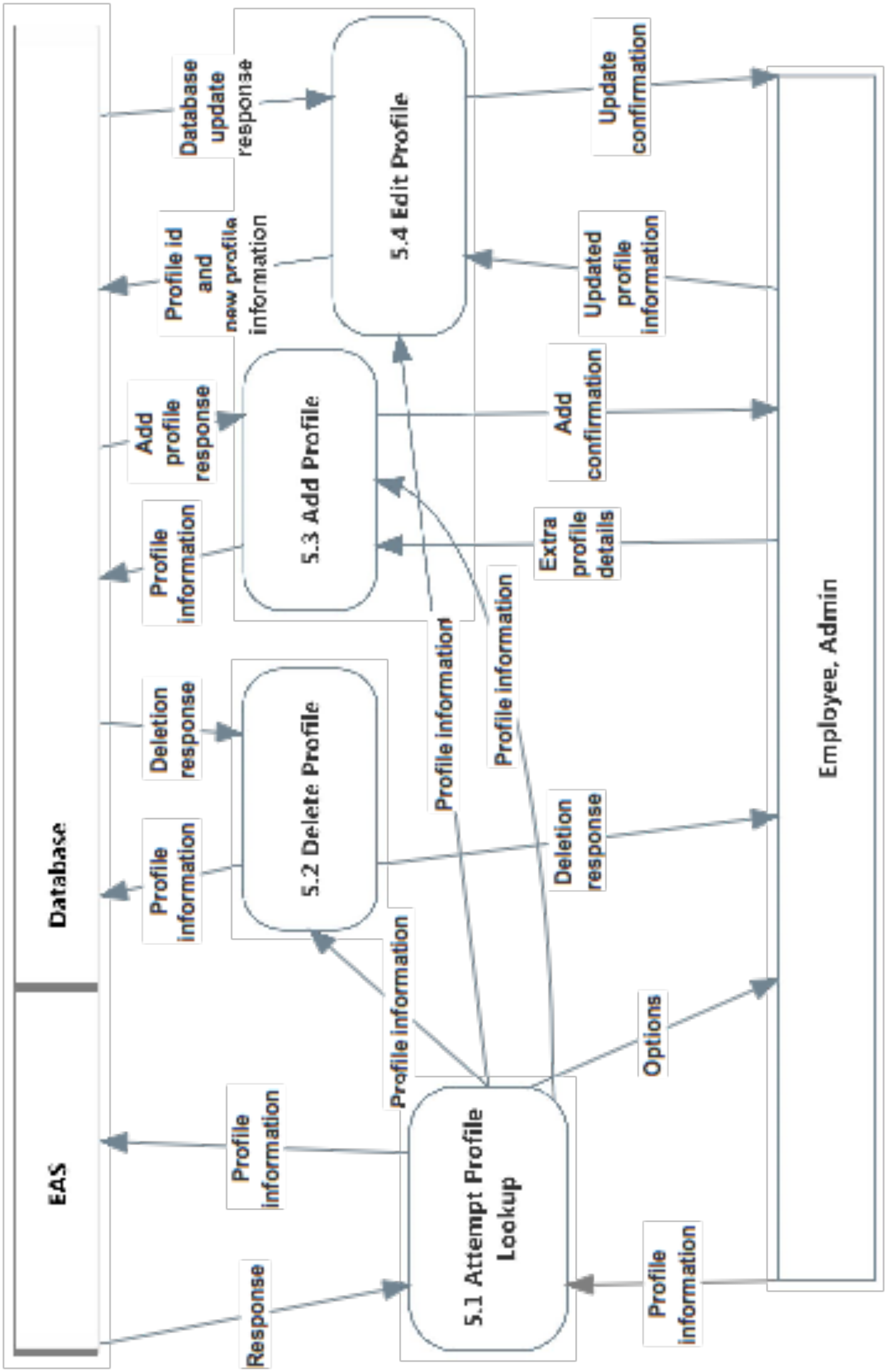
4.4.3. Print Receipts



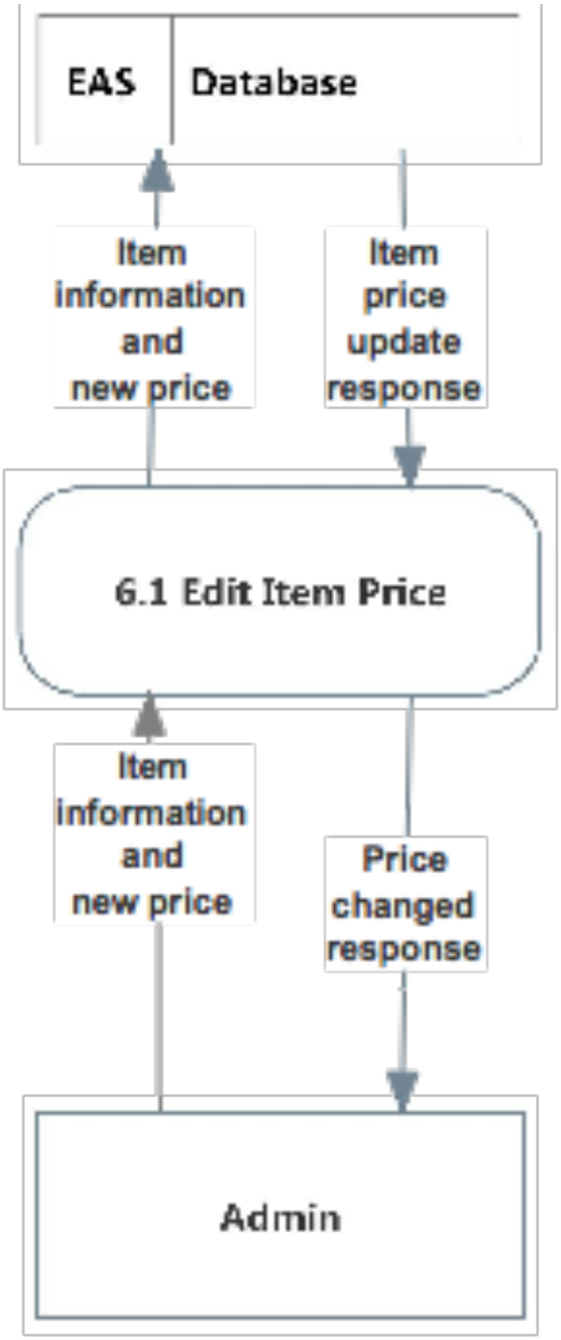
4.4.4. Accept Donations



4.4.5. Manage Profiles



4.4.6. Edit Prices


EAS
Database
Item information and new price
Item price update response
6.1 Edit Item Price
Item information and new price
Price changed response
Admin

5. Functional Requirements Inventory

The functional requirements inventory lists the functions that are necessary to the completion of E.A.S. The inventory lists all of the different abilities that each user has for interacting with the system.

5.1. Employee

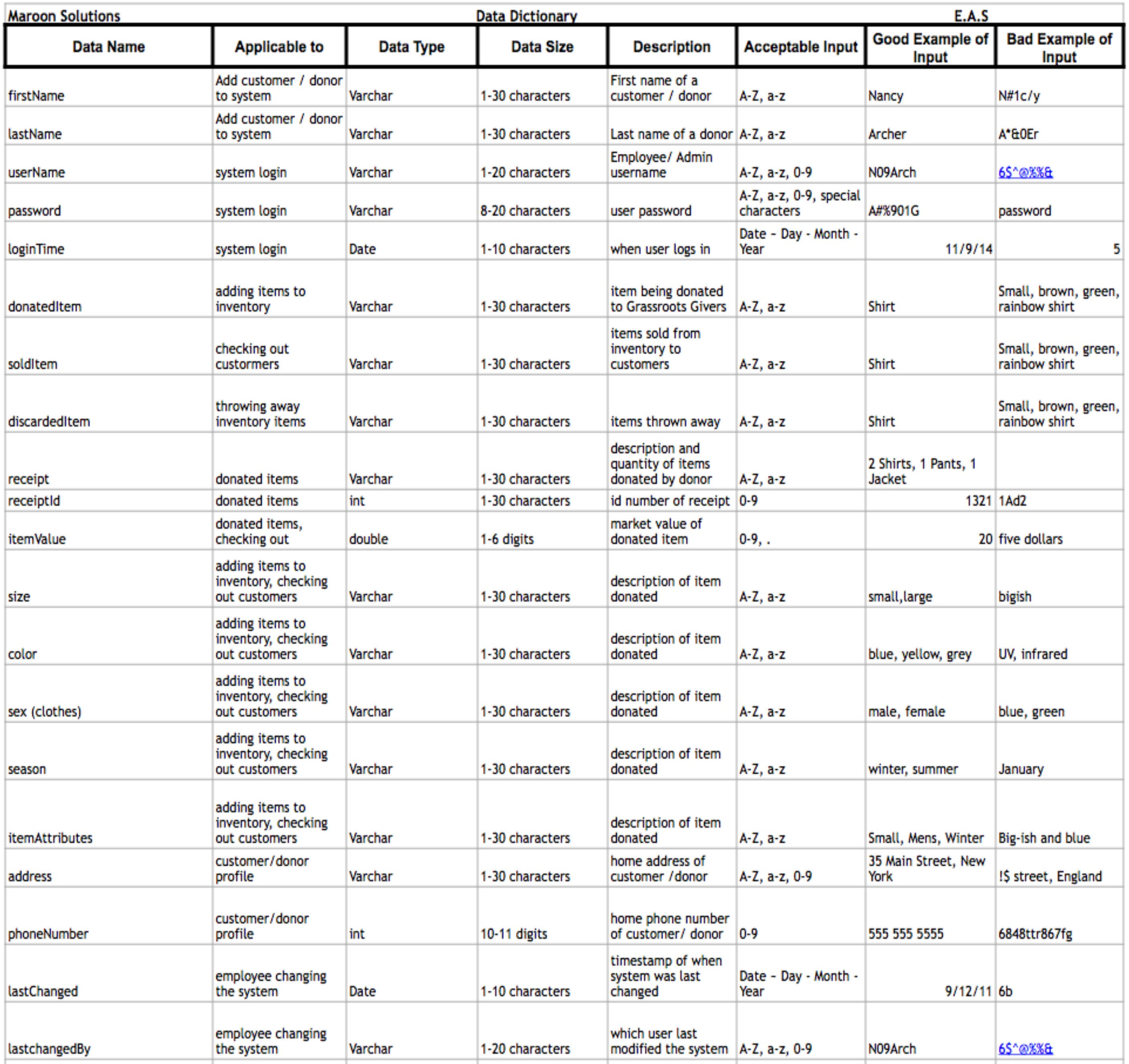
- Will be able to Log on to an employee account
  - Enter Username
  - Enter Password
- Will be able to print receipts for donors
- Will be able to accept donations
  - Process donations
  - Add donations to sellable inventory
- Will be able to sell items
  - Mark items as sold
  - Check customer information
- Will be able to add customer/donor profiles

5.2. Administrator

- Inherits all abilities of Employee
- Can manage accounts
  - Add /edit/remove employee and other admin accounts
  - Remove customer/donor profiles
- Will be able to edit prices of items

6. Logic Data Dictionary

The purpose of the data dictionary is to display the variety of information and metadata that needs to be stored in order for the system to be functional. At the current phase, the data dictionary only showcases the data names and types, and also describes the purpose of each piece of data within the system. In the next document, the Detailed Design, the data dictionary will detail how these pieces are stored within the E.A.S. database.



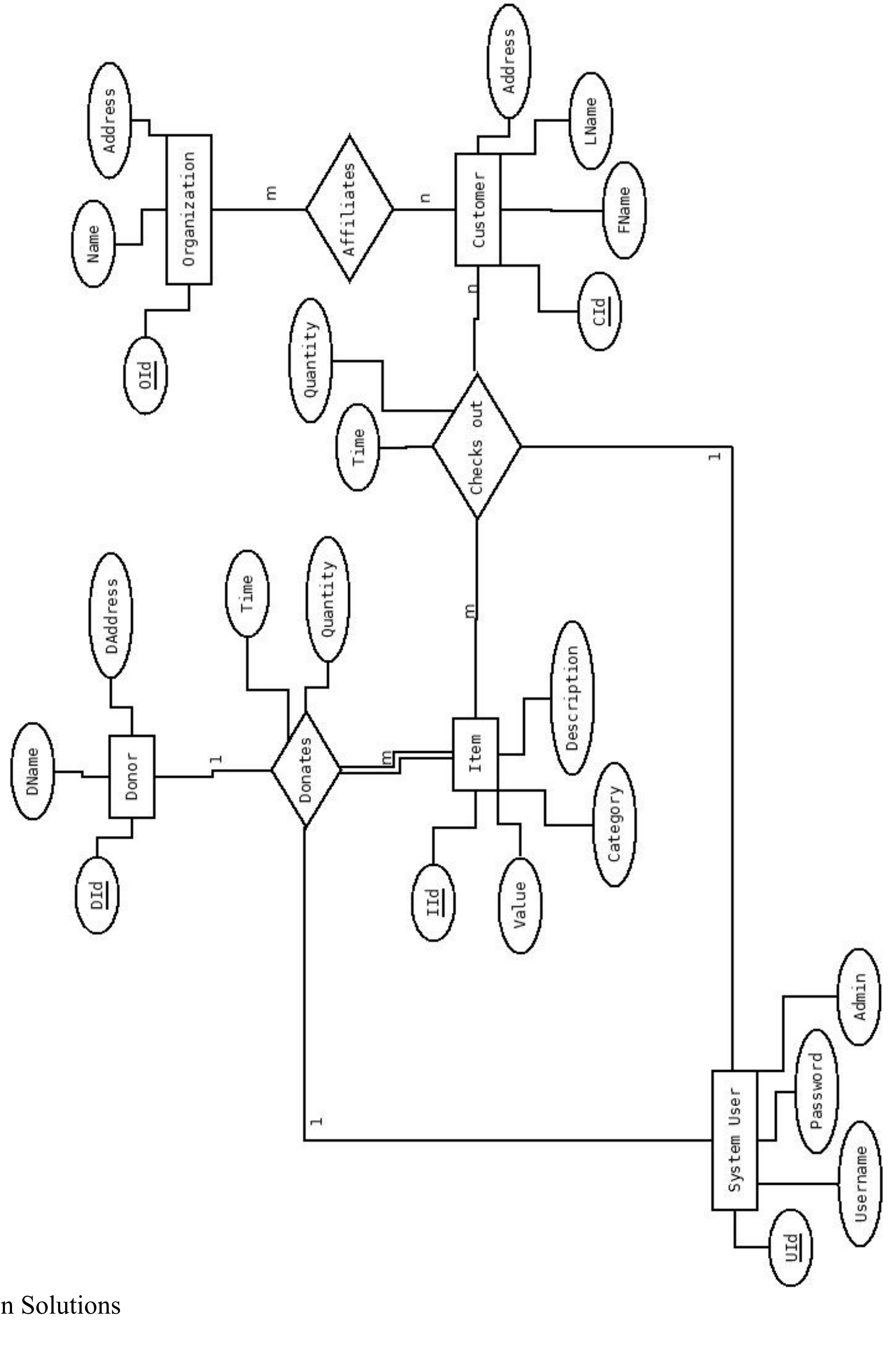
Maroon Solutions Data Dictionary E.A.S

Data Name	Applicable to	Data Type	Data Size	Description	Acceptable Input	Good Example of Input	Bad Example of Input
firstName	Add customer / donor to system	Varchar	1-30 characters	First name of a customer / donor	A-Z, a-z	Nancy	N#1c/y
lastName	Add customer / donor to system	Varchar	1-30 characters	Last name of a donor	A-Z, a-z	Archer	A*&0Er
userName	system login	Varchar	1-20 characters	Employee/ Admin username	A-Z, a-z, 0-9	N09Arch	6$^@%%&
password	system login	Varchar	8-20 characters	user password	A-Z, a-z, 0-9, special characters	A#%901G	password
loginTime	system login	Date	1-10 characters	when user logs in	Date - Day - Month - Year	11/9/14	5
donatedItem	adding items to inventory	Varchar	1-30 characters	item being donated to Grassroots Givers	A-Z, a-z	Shirt	Small, brown, green, rainbow shirt
soldItem	checking out custormers	Varchar	1-30 characters	items sold from inventory to customers	A-Z, a-z	Shirt	Small, brown, green, rainbow shirt
discardedItem	throwing away inventory items	Varchar	1-30 characters	items thrown away	A-Z, a-z	Shirt	Small, brown, green, rainbow shirt
receipt	donated items	Varchar	1-30 characters	description and quantity of items donated by donor	A-Z, a-z	2 Shirts, 1 Pants, 1 Jacket	
receiptId	donated items	int	1-30 characters	id number of receipt	0-9	1321	1Ad2
itemValue	donated items, checking out	double	1-6 digits	market value of donated item	0-9, .	20	five dollars
size	adding items to inventory, checking out customers	Varchar	1-30 characters	description of item donated	A-Z, a-z	small,large	bigish
color	adding items to inventory, checking out customers	Varchar	1-30 characters	description of item donated	A-Z, a-z	blue, yellow, grey	UV, infrared
sex (clothes)	adding items to inventory, checking out customers	Varchar	1-30 characters	description of item donated	A-Z, a-z	male, female	blue, green
season	adding items to inventory, checking out customers	Varchar	1-30 characters	description of item donated	A-Z, a-z	winter, summer	January
itemAttributes	adding items to inventory, checking out customers	Varchar	1-30 characters	description of item donated	A-Z, a-z	Small, Mens, Winter	Big-ish and blue
address	customer/donor profile	Varchar	1-30 characters	home address of customer /donor	A-Z, a-z, 0-9	35 Main Street, New York	!$ street, England
phoneNumber	customer/donor profile	int	10-11 digits	home phone number of customer/ donor	0-9	555 555 5555	6848ttr867fg
lastChanged	employee changing the system	Date	1-10 characters	timestamp of when system was last changed	Date - Day - Month - Year	9/12/11	6b
lastchangedBy	employee changing the system	Varchar	1-20 characters	which user last modified the system	A-Z, a-z, 0-9	N09Arch	6$^@%%&

7. Database Diagrams

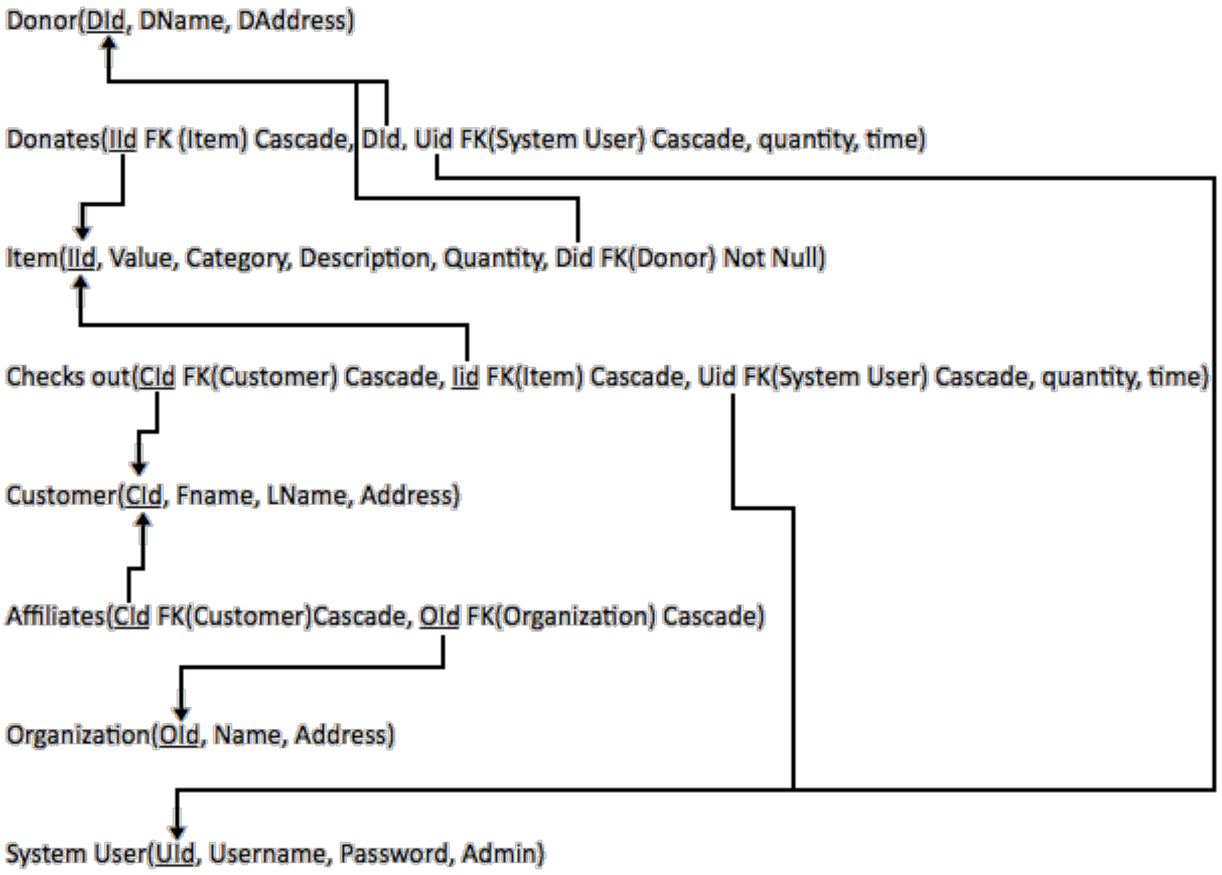
7.1. Entity-Relationship Diagram

An entity-relationship diagram (E/R Diagram) is a graphical representation of entities and their relationships to each other. It is in regard to the organization of data within our database.



7.2. Relational Database Schema

A relational database schema is the tables, columns, and relationships that make up a relational database.



Donor(DId, DName, DAddress)

Donates(IId FK (Item) Cascade, DId, Uid FK(System User) Cascade, quantity, time)

Item(IId, Value, Category, Description, Quantity, Did FK(Donor) Not Null)

Checks out(CId FK(Customer) Cascade, Iid FK(Item) Cascade, Uid FK(System User) Cascade, quantity, time)

Customer(CId, Fname, LName, Address)

Affiliates(CId FK(Customer)Cascade, OId FK(Organization) Cascade)

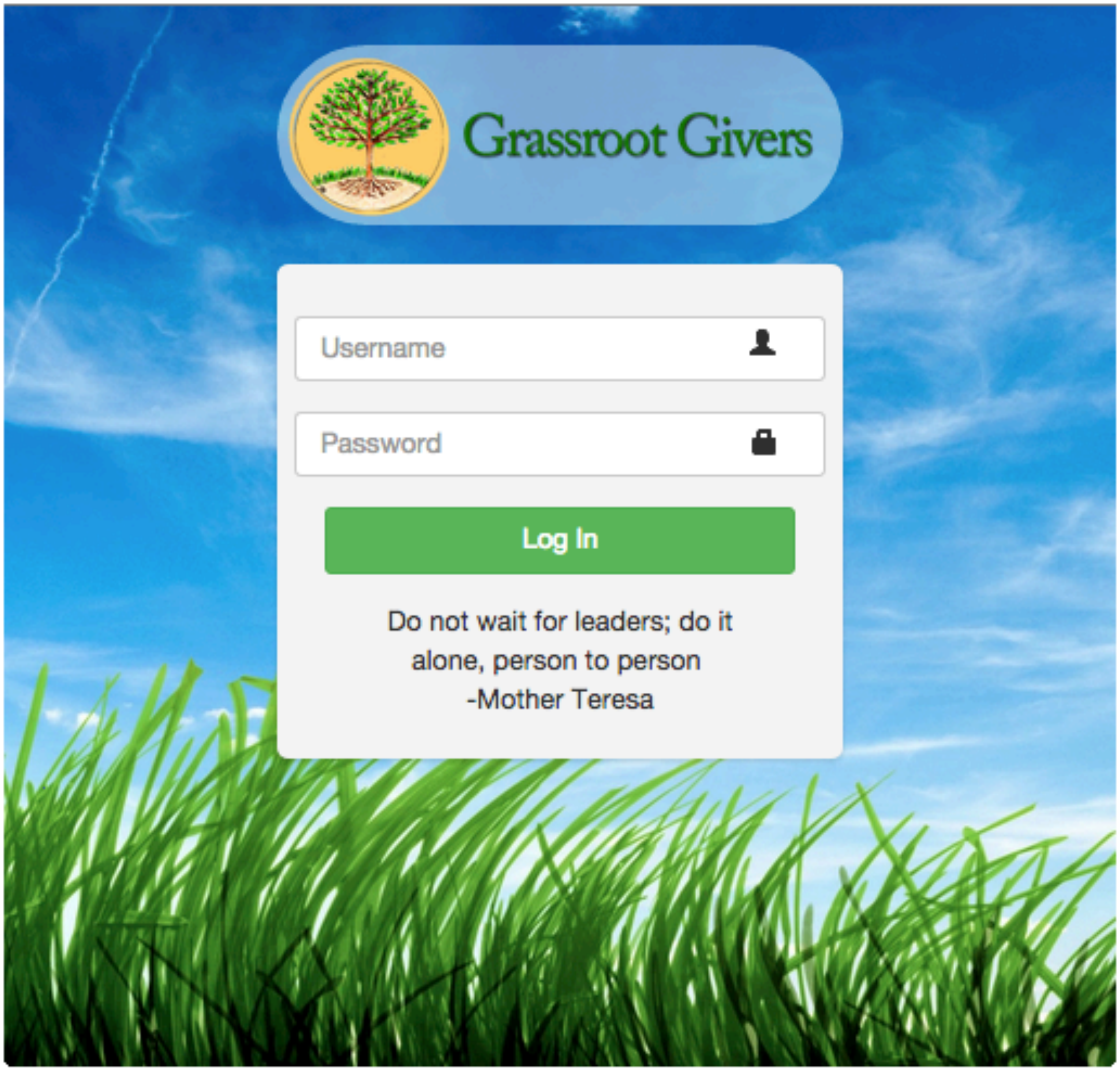
Organization(OId, Name, Address)

System User(UId, Username, Password, Admin)

8. Modules

8.1. Login Page

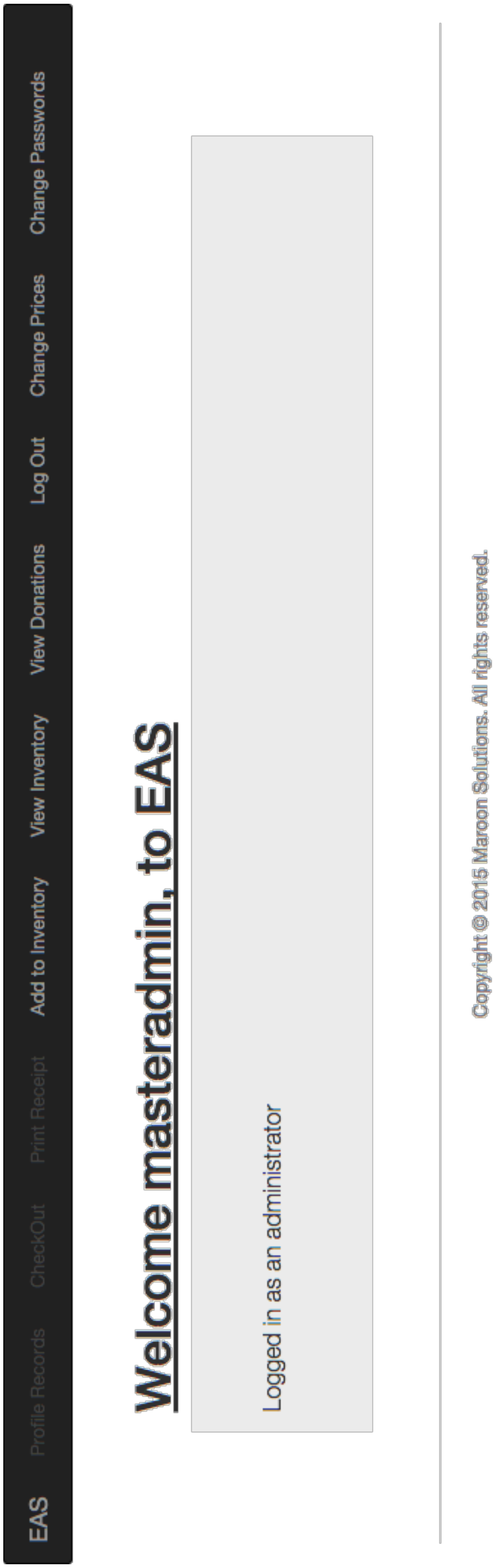
The login page is fully functional and the code can be found here: Login Page.



Do not wait for leaders; do it alone, person to person
-Mother Teresa

8.2. Home Page

The home page is fully functional and the code can be found here: Home Page.



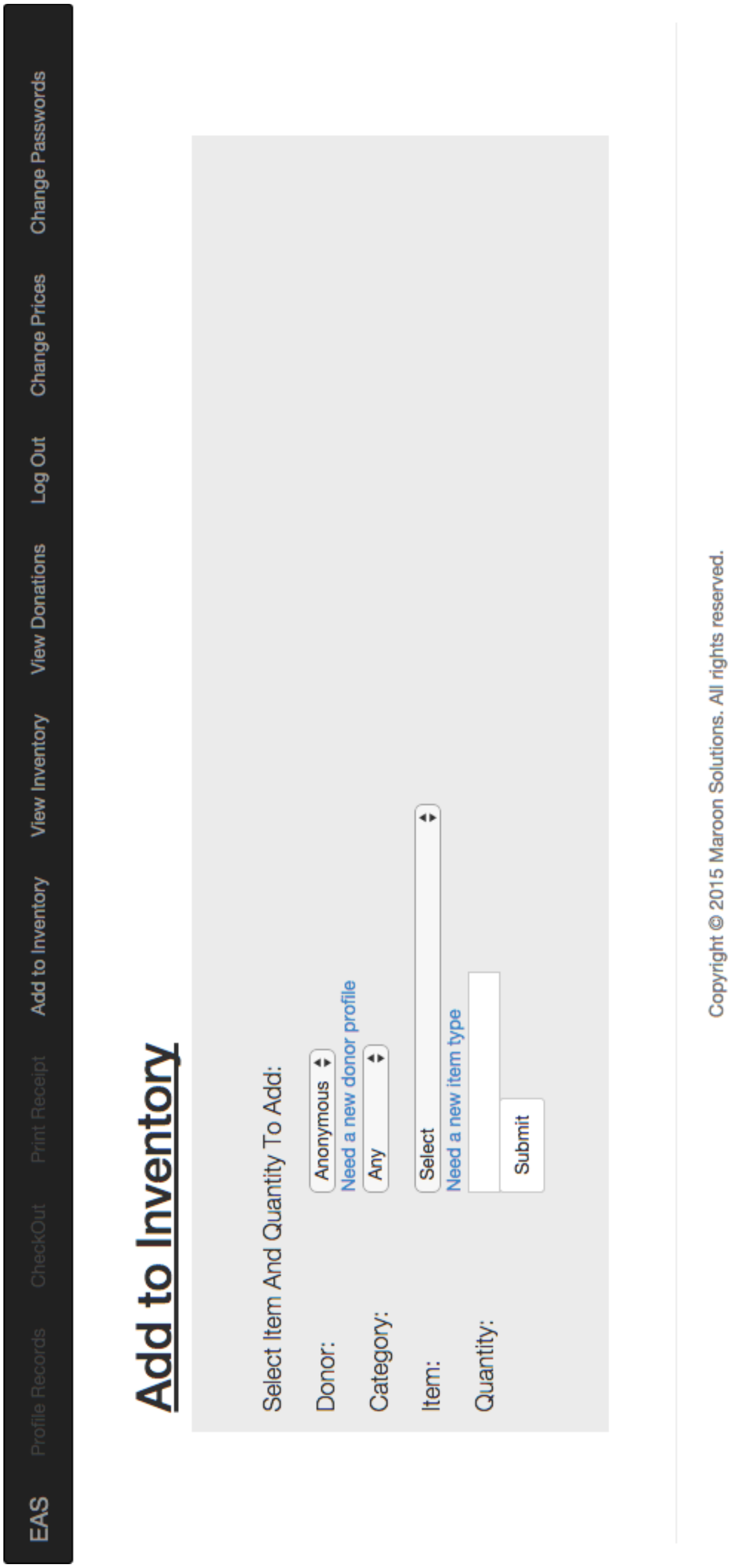
Welcome masteradmin, to EAS

Logged in as an administrator

Copyright © 2015 Maroon Solutions. All rights reserved.

8.3. Add To Inventory Page

The add items page is fully functional and the code can be found here: Add Items.



Add to Inventory

Select Item And Quantity To Add:

Donor: Anonymous

Need a new donor profile

Category: Any

Item: Select

Need a new item type

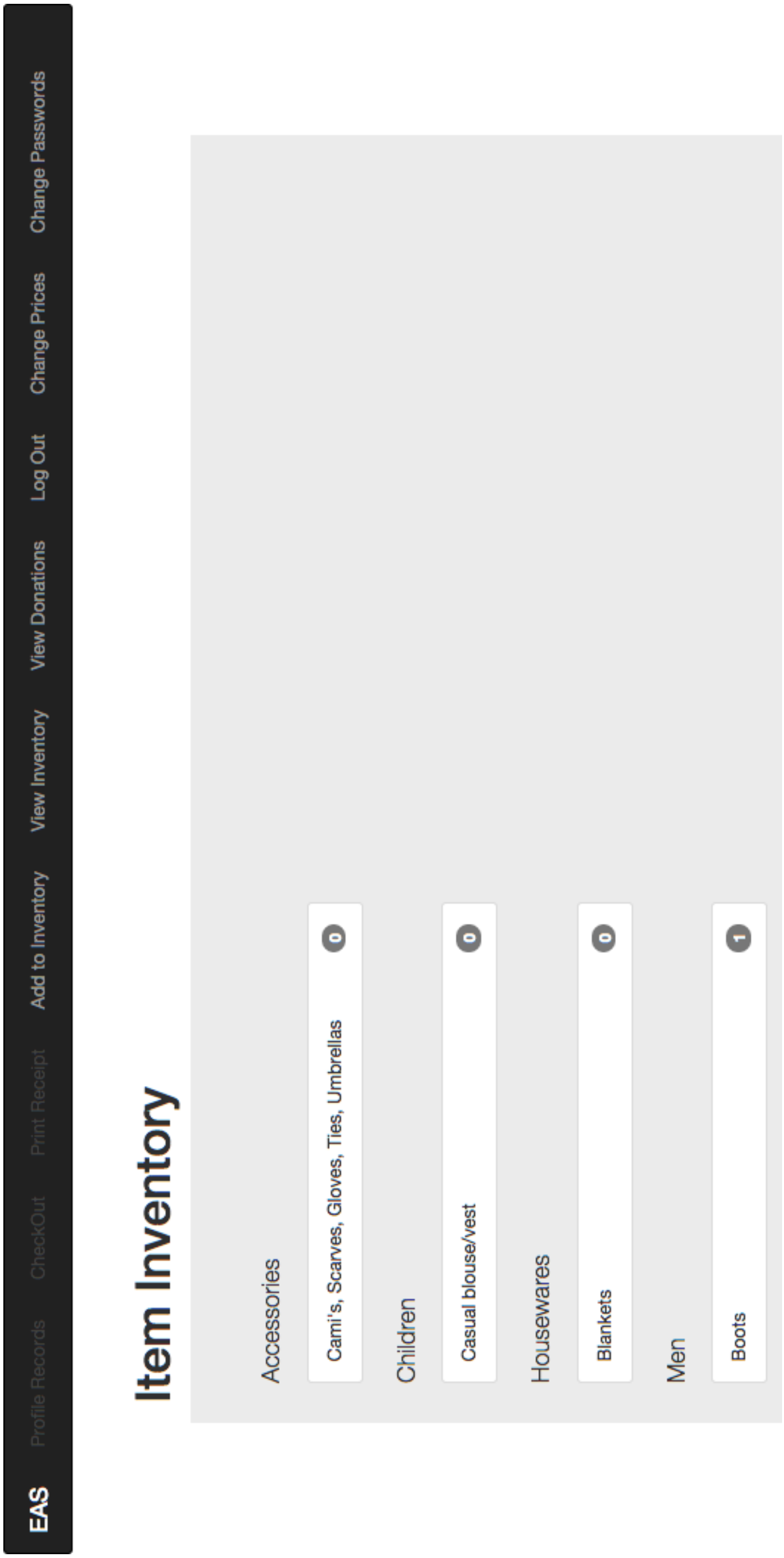
Quantity:

Submit

Copyright © 2015 Maroon Solutions. All rights reserved.

8.4. View Inventory Page

The view inventory page is fully functional and the code can be found here: View Inventory.



Item Inventory

Accessories

Cami's, Scarves, Gloves, Ties, Umbrellas 0

Children

Casual blouse/vest 0

Housewares

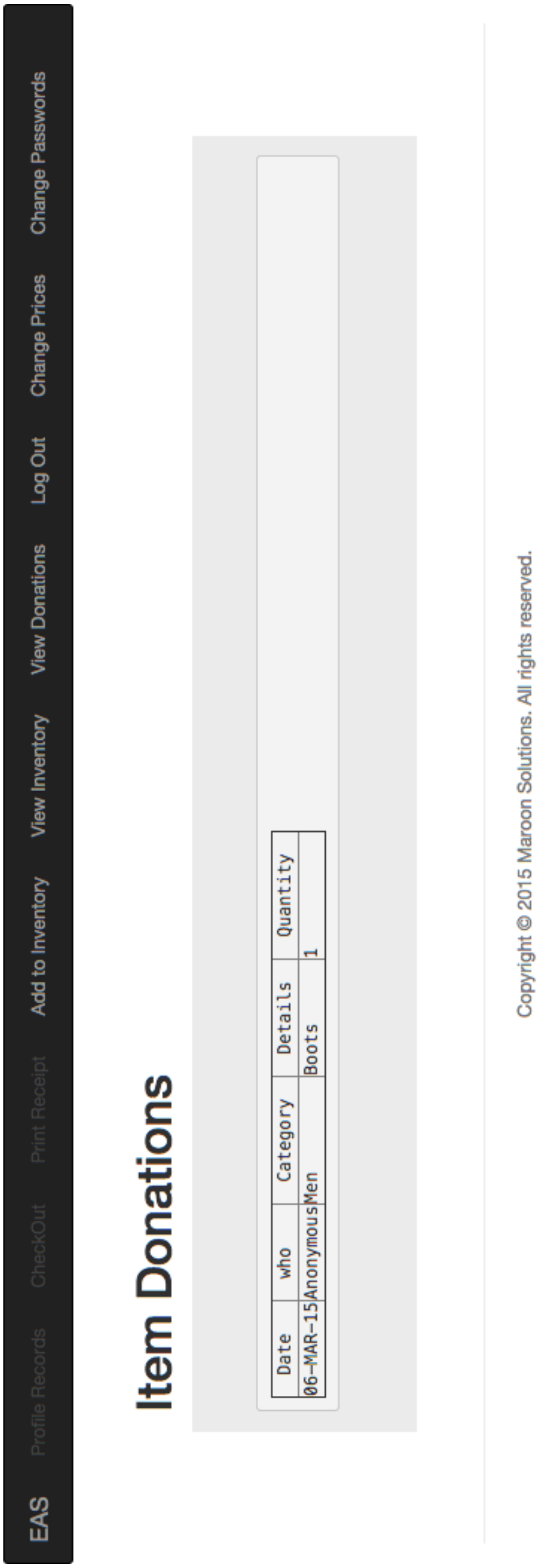
Blankets 0

Men

Boots 1

8.5. View Donations Page

The view donations page is fully functional and the code can be found here: View Donations.



Item Donations

Date	who	Category	Details	Quantity
06-MAR-15	Anonymous	Men	Boots	1

Copyright © 2015 Maroon Solutions. All rights reserved.

8.6. Managing Profiles Module

The managing profiles page will be used to search for any type of profile. If the profile is found, the user can view or edit that profile.

How it will work:
Under the managing profiles tab within E.A.S. user will be able to sort by the different types of profiles; such as customer, donor and employee profiles. We will display this by having drop down boxes for the user to select the type of profile wish to view. After that, additional options will display to further filter the users search. Once the user has found the profile they were looking for, they can select it and fully view the profiles information.

Pseudocode:

```
Dropdown(customer, donor, employee) profile
IF Customer (display customer names and further search options)
else IF Donor (display donor names and further search options)
Else IF Employee (display employee names and further search options)
Then from there the generated query will consist of the options chosen:
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers
ON Orders.CustomerID=Customers.CustomerID;
```

8.7. Print Receipts Module

This module gathers information from a chosen donor whose name is supplied. Looks up information from the date range that is selected. It creates an object that contains all useful the information that should be on the receipt and can easily be printed with that information. A receipt contains the donor's name, address, a list of the donated items, the approximate value of the donations, the date, and the total sum of the donated items.

How it works:

The user will first select the donor they want a receipt for, for the system to get their Id. Then the user will enter a date range for when the items were catalogued into the system through the donation process. Then the system will query the database for all donated items that came from this donor over the date range and gather them into a receipt format for viewing and printing.

Pseudocode:

```
procedure printReceipt(donorName, donorAddress, date){
        List<Item> items
        int totalValue
        if (donorName, donorAddress, and date supplied all){
                check for data in database
        }else{
                prompt user to select name, address, or date and display list of donors who relate
        }
        for (item : items donated by donor on date ){
                add item to items
                increment totalValue by value attribute of (item.value * item.quantity)
        }
        return new receipt(donorName, donorAddress, donations, value, date)
}
```

8.8. Edit Prices Module

For the Editing Prices tab within E.A.S., only administrators will be able to update the approximate prices of items. This is used for tax purposes and will affect how much is written on receipts. It can also be used to find out quarterly or yearly statistics for grants.

How it works:

The Edit Prices page will query the database for all item types and list them on the page. Each row will have an edit button to initialize an edit of that item's value. This button will responsively(by JavaScript) change the text box for editing, change the edit button to a submit button and create another button next to it for cancelling the change. The cancel button is for if the user has clicked on the wrong item. They can then enter new values for an item; clicking submit will initialize a connection to the database, update that item's value accordingly and finally commit the change.

Pseudocode:

```
OnPageLoad(){
        Connect database;
        Query database;
        while(Query had results){
                Generate list item with Item Id, item name and category, and add edit button;
        }
}
OnEditPressed(){
        Find Item Id where edit button was pressed;
        Enable text box with the current value to be editable;
        Hide edit button;
        Put submit button in its place;
        Put cancel button next to submit;
}
OnSubmitPressed(){
        Take the new value;
        Ajax call to update query the database;
        Alert the user of successful change or error;
}
```

8.9. Sell Items Module

Removes items from main inventory to sell to a customer. Allows user to select descriptors of an item to add through text boxes and refine the item criteria until an item name is added. Then stored in the database.

How it works:

The user will find the items the customer wishes to checkout from the list of items in the inventory. Once they find the item and fill out the quantity of that item the customer is taking the system will update the database with the proper changes to the inventory as well as log the checkout for future use.

Pseudocode:

```
procedure sellItems(){
    Item item
    //Item contains name and descriptors
    //ex: Clothing
    describe donation high level through text box
    if (description selected){
        display child text box with more descriptors
        //ex: Pants
        refine description  attribute
        if (refined selected){
            //dropdown and refined via search
            display list of items in that category
            //select from list of items in inventory
            enter name of item
            item = item(name, descriptors)
            remove (item) from database
            add information of items sold to SOLDITEMS in database
        }
    }
}
```

9. Testing Plan

9.1. Overview and Strategy

E.A.S. is a web application; therefore, E.A.S. will be tested to make sure it is functional on each major web browser. The web browsers to be tested on are Google Chrome, Internet Explorer, Mozilla Firefox, Apple Safari and mobile devises. E.A.S. can also be tested on other web browsers if time allows. E.A.S. will be tested through a number of unit tests that are determined by the functional requirements. The non-functional requirements, which cannot be measured, will be evaluated by the members of Maroon Solutions to ensure each requirement has been satisfactorily met. A brief overview of the testing follows this section. In the detailed design there will be more detail for the testing plan.

9.2. Acceptance Test

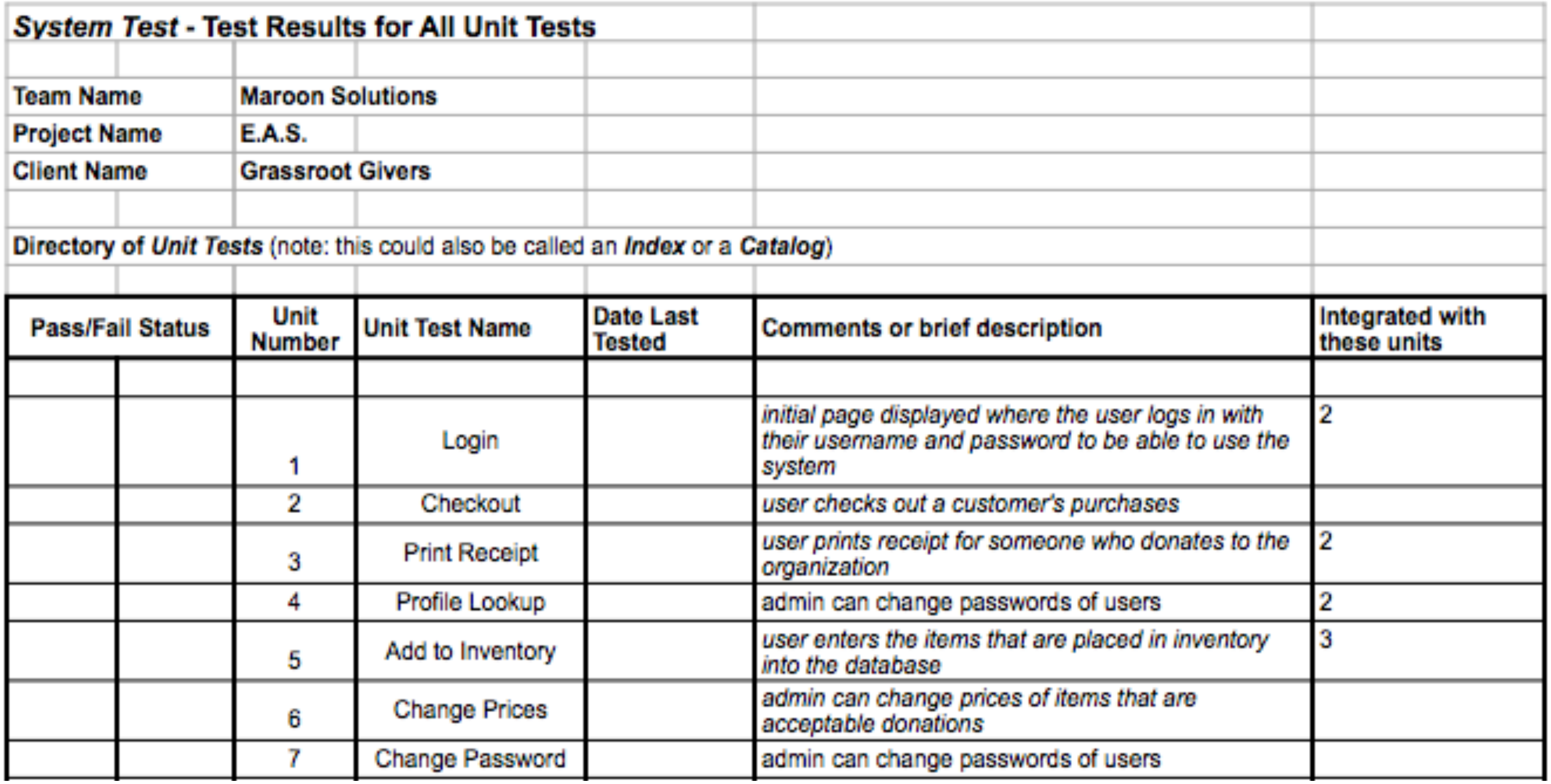
Each component of E.A.S. will have individual tests that will be performed on the system. These tests will be done in order to make sure the individual parts of the system are working properly. Once the individual tests have been completed, an acceptance test will be executed to check if the functional requirements have been met. Once the acceptance test is over, Maroon Solutions and the clients, Ms. Mary Partridge-Brown and Ms. Roberta Sandler, will decide whether all of the requirements have been sufficiently met. Maroon Solutions will design the testing plan using the functional requirements inventory.

9.3. Unit Tests

The unit tests are specific tests that will be run to ensure that the system is running properly as it is developed. The unit tests give step-by-step directions in the test cases on what to input and what to look for when a test is run. Certain tests need to be run together to ensure that all functions of the system work together properly and do not interfere with each other; these specific tests are listed in the "Integrated with these units" column.

9.3.1. Test Cases

Each Unit is made up of a series of test cases. In these test cases there are directions on what to input into the system and in what order. The test cases also contain information on what the system should look like before the test is run and how it should respond after the test is run. If all of the test cases pass in the unit test, then the unit is considered to be functioning properly.



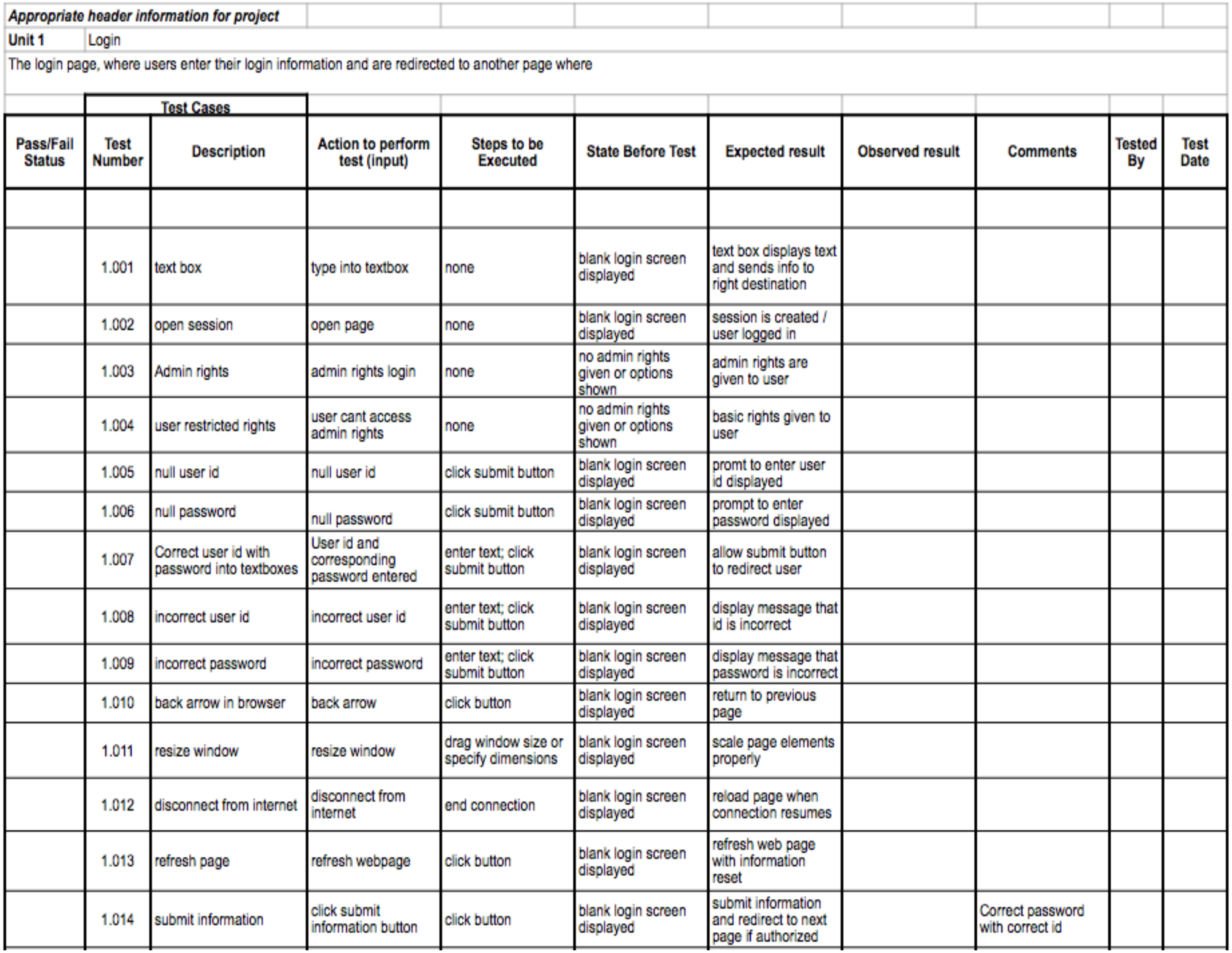
System Test - Test Results for All Unit Tests

Team Name	**Maroon Solutions**
Project Name	**E.A.S.**
Client Name	**Grassroot Givers**

Directory of _Unit Tests_ (note: this could also be called an ***Index*** or a ***Catalog***)

Pass/Fail Status	Unit Number	Unit Test Name	Date Last Tested	Comments or brief description	Integrated with these units
	1	Login		*initial page displayed where the user logs in with their username and password to be able to use the system*	2
	2	Checkout		*user checks out a customer's purchases*	
	3	Print Receipt		*user prints receipt for someone who donates to the organization*	2
	4	Profile Lookup		admin can change passwords of users	2
	5	Add to Inventory		*user enters the items that are placed in inventory into the database*	3
	6	Change Prices		*admin can change prices of items that are acceptable donations*	
	7	Change Password		admin can change passwords of users	

9.3.1.1. Log On



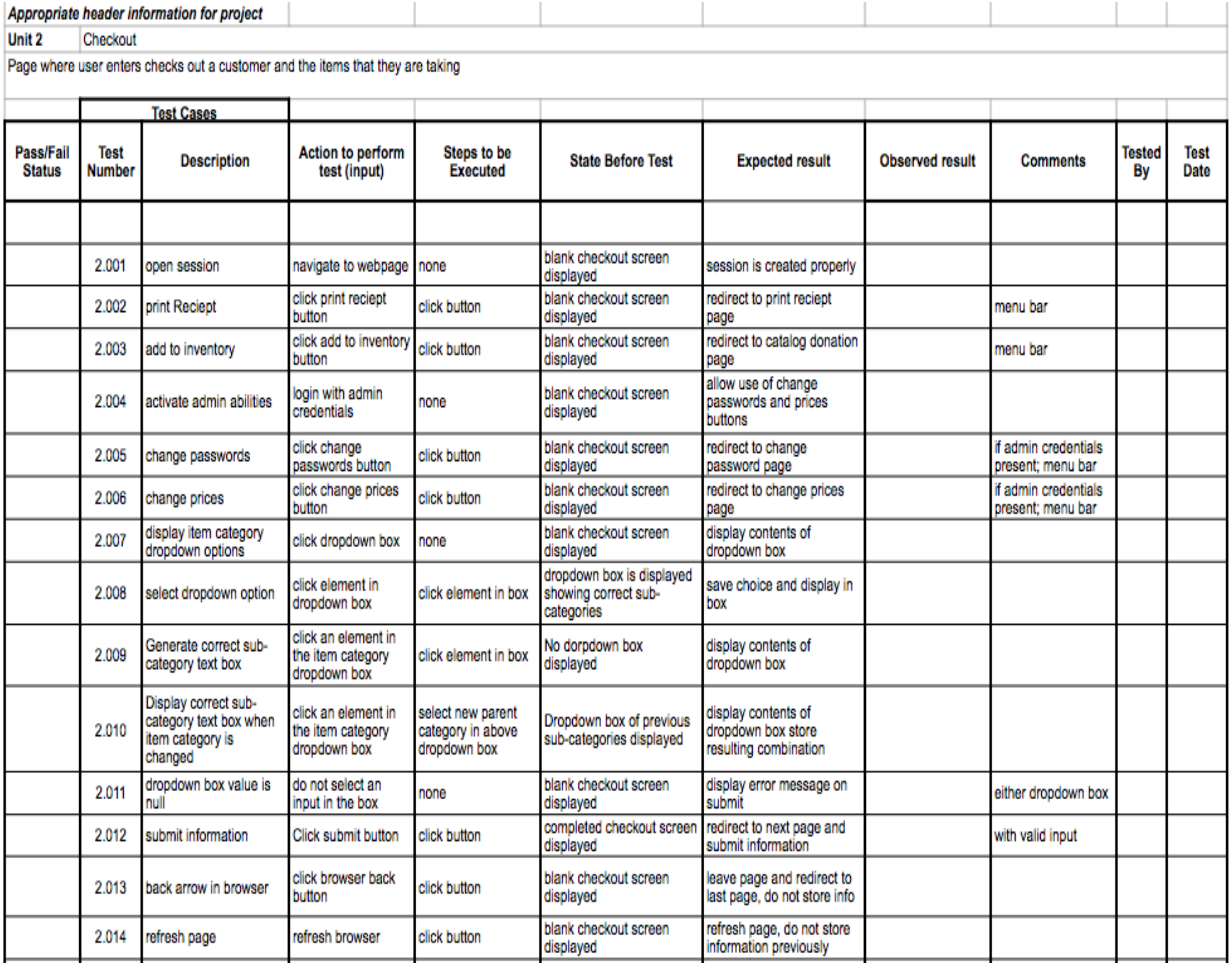
Appropriate header information for project

Unit 1 Login

The login page, where users enter their login information and are redirected to another page where

	Test Cases									
Pass/Fail Status	**Test Number**	**Description**	**Action to perform test (input)**	**Steps to be Executed**	**State Before Test**	**Expected result**	**Observed result**	**Comments**	**Tested By**	**Test Date**
	1.001	text box	type into textbox	none	blank login screen displayed	text box displays text and sends info to right destination				
	1.002	open session	open page	none	blank login screen displayed	session is created / user logged in				
	1.003	Admin rights	admin rights login	none	no admin rights given or options shown	admin rights are given to user				
	1.004	user restricted rights	user cant access admin rights	none	no admin rights given or options shown	basic rights given to user				
	1.005	null user id	null user id	click submit button	blank login screen displayed	promt to enter user id displayed				
	1.006	null password	null password	click submit button	blank login screen displayed	prompt to enter password displayed				
	1.007	Correct user id with password into textboxes	User id and corresponding password entered	enter text; click submit button	blank login screen displayed	allow submit button to redirect user				
	1.008	incorrect user id	incorrect user id	enter text; click submit button	blank login screen displayed	display message that id is incorrect				
	1.009	incorrect password	incorrect password	enter text; click submit button	blank login screen displayed	display message that password is incorrect				
	1.010	back arrow in browser	back arrow	click button	blank login screen displayed	return to previous page				
	1.011	resize window	resize window	drag window size or specify dimensions	blank login screen displayed	scale page elements properly				
	1.012	disconnect from internet	disconnect from internet	end connection	blank login screen displayed	reload page when connection resumes				
	1.013	refresh page	refresh webpage	click button	blank login screen displayed	refresh web page with information reset				
	1.014	submit information	click submit information button	click button	blank login screen displayed	submit information and redirect to next page if authorized		Correct password with correct id		

9.3.1.2. Check Out



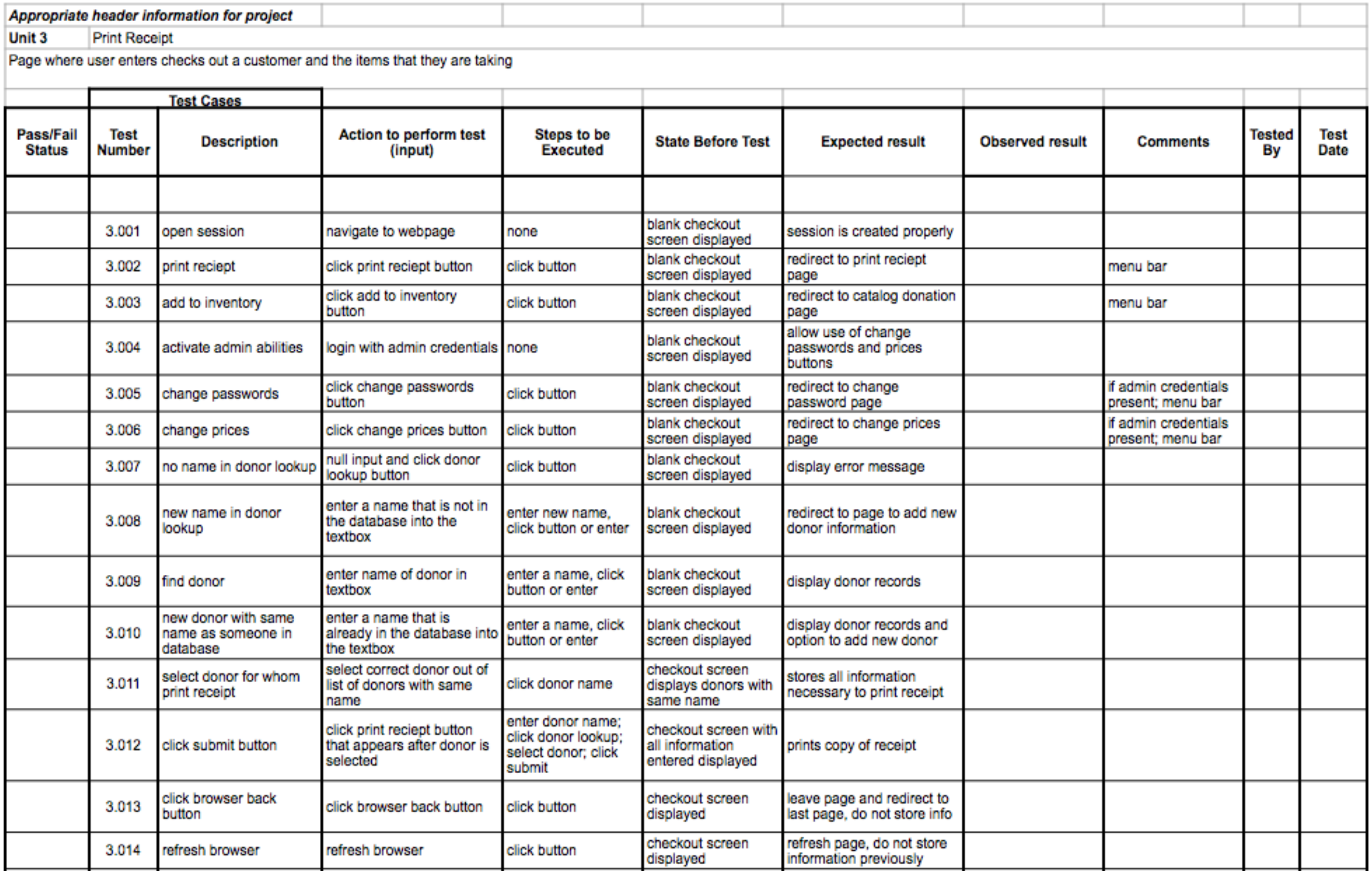
Appropriate header information for project

Unit 2 Checkout

Page where user enters checks out a customer and the items that they are taking

	Test Cases									
Pass/Fail Status	**Test Number**	**Description**	**Action to perform test (input)**	**Steps to be Executed**	**State Before Test**	**Expected result**	**Observed result**	**Comments**	**Tested By**	**Test Date**
	2.001	open session	navigate to webpage	none	blank checkout screen displayed	session is created properly				
	2.002	print Reciept	click print reciept button	click button	blank checkout screen displayed	redirect to print reciept page		menu bar		
	2.003	add to inventory	click add to inventory button	click button	blank checkout screen displayed	redirect to catalog donation page		menu bar		
	2.004	activate admin abilities	login with admin credentials	none	blank checkout screen displayed	allow use of change passwords and prices buttons				
	2.005	change passwords	click change passwords button	click button	blank checkout screen displayed	redirect to change password page		if admin credentials present; menu bar		
	2.006	change prices	click change prices button	click button	blank checkout screen displayed	redirect to change prices page		if admin credentials present; menu bar		
	2.007	display item category dropdown options	click dropdown box	none	blank checkout screen displayed	display contents of dropdown box				
	2.008	select dropdown option	click element in dropdown box	click element in box	dropdown box is displayed showing correct sub-categories	save choice and display in box				
	2.009	Generate correct sub-category text box	click an element in the item category dropdown box	click element in box	No dorpdown box displayed	display contents of dropdown box				
	2.010	Display correct sub-category text box when item category is changed	click an element in the item category dropdown box	select new parent category in above dropdown box	Dropdown box of previous sub-categories displayed	display contents of dropdown box store resulting combination				
	2.011	dropdown box value is null	do not select an input in the box	none	blank checkout screen displayed	display error message on submit		either dropdown box		
	2.012	submit information	Click submit button	click button	completed checkout screen displayed	redirect to next page and submit information		with valid input		
	2.013	back arrow in browser	click browser back button	click button	blank checkout screen displayed	leave page and redirect to last page, do not store info				
	2.014	refresh page	refresh browser	click button	blank checkout screen displayed	refresh page, do not store information previously				

9.3.1.3. Print Receipt



Appropriate header information for project

Unit 3 Print Receipt

Page where user enters checks out a customer and the items that they are taking

	Test Cases									
Pass/Fail Status	**Test Number**	**Description**	**Action to perform test (input)**	**Steps to be Executed**	**State Before Test**	**Expected result**	**Observed result**	**Comments**	**Tested By**	**Test Date**
	3.001	open session	navigate to webpage	none	blank checkout screen displayed	session is created properly				
	3.002	print reciept	click print reciept button	click button	blank checkout screen displayed	redirect to print reciept page		menu bar		
	3.003	add to inventory	click add to inventory button	click button	blank checkout screen displayed	redirect to catalog donation page		menu bar		
	3.004	activate admin abilities	login with admin credentials	none	blank checkout screen displayed	allow use of change passwords and prices buttons				
	3.005	change passwords	click change passwords button	click button	blank checkout screen displayed	redirect to change password page		if admin credentials present; menu bar		
	3.006	change prices	click change prices button	click button	blank checkout screen displayed	redirect to change prices page		if admin credentials present; menu bar		
	3.007	no name in donor lookup	null input and click donor lookup button	click button	blank checkout screen displayed	display error message				
	3.008	new name in donor lookup	enter a name that is not in the database into the textbox	enter new name, click button or enter	blank checkout screen displayed	redirect to page to add new donor information				
	3.009	find donor	enter name of donor in textbox	enter a name, click button or enter	blank checkout screen displayed	display donor records				
	3.010	new donor with same name as someone in database	enter a name that is already in the database into the textbox	enter a name, click button or enter	blank checkout screen displayed	display donor records and option to add new donor				
	3.011	select donor for whom print receipt	select correct donor out of list of donors with same name	click donor name	checkout screen displays donors with same name	stores all information necessary to print receipt				
	3.012	click submit button	click print reciept button that appears after donor is selected	enter donor name; click donor lookup; select donor; click submit	checkout screen with all information entered displayed	prints copy of receipt				
	3.013	click browser back button	click browser back button	click button	checkout screen displayed	leave page and redirect to last page, do not store info				
	3.014	refresh browser	refresh browser	click button	checkout screen displayed	refresh page, do not store information previously				

10. Development Environment and Production Environment

10.1. Development Environment

Development

SE Lab hardware/software specs - PC:

- Dell
- Windows 7
- 6GB RAM
- 499.78GB disk space, 427.84GB free
- 3.20Hz Intel® Core™ i5-3470 CPU
- Adobe AIR
- Adobe Flash Player
- Google Chrome
- Mozilla Firefox
- Microsoft IE, Office OneNote, Outlook, SQL Server, Office 2010, Visual Studio
- SmartDraw
- Oracle SQL Developer, Java SE7, Netbeans IDE 8.0
- Eclipse
- BlueJ
- WinSCP
- Audacity
- IDLE Python GUI
- Notepad++
- PUTTY

SE Lab hardware/software specs - Mac:

- iMac
- OS X Lion 10.7.5
- 4GB RAM
- 499.25GB disk space, 450.54GB free
- 2.5GHz Intel® Core™ i5
- Adobe reader
- Mozilla Firefox
- Google Chrome
- Microsoft Excel, Word, Query, Powerpoint
- XCode

Server:

- Hostname: oraserv.cs.siena.edu/~perm_maroon/
- CentOS 5.2 (final)
- Kernel: 2.6.18-92.el5
- Intel Xeon 2.66 GHz CPU
- 8 GB of Memory
- Java SE Runtime Enviornment (build 1.6.0 10-rc-b28)
- GCC Version 4.1.2 20071124 (Red Hat 4.1.2-42)
- Python 2.4.3

11. APPENDICES

11.1. Appendix A: Glossary of Terms

Actor: Actors that interact with the system through sues. Actors can be human or non human.
Adobe Dreamweaver: Tool used for web application development
Adobe Photoshop: Graphic editing application
Apache HTTP Server: Apache HyperText Transfer Protocol Server, Web server application
Apple Safari: Web browser designed by Apple
Data Stores: A component of a Data Flow Diagram that represents a location in which information or data is stored
Database: Organizes data, typically through a computer, so that the data is easily accessible
Data Flow: Data/information flowing to or from a process in a Data Flow Diagram
Data Flow Diagram: A graphical representation of the "flow" of data through an information system
Data Store: Location where data is held temporarily or permanently in a Data Flow Diagram
E.A.S.: Efficient and Simple - the software design by Maroon Solutions
Eclipse: Programming environment developed by the Eclipse Foundation
External Entities: A component of a Data Flow Diagram that represents any human or non-human user of a Software System
Functional Requirements Inventory: Defines what the system will be able to do and what is testable about the system
Gantt Chart: Bar chart typically used to project scheduling
Google Chrome: Web browser designed by Google
HTML: HyperText Markup Language, main language for creating web pages
HTTP: Hypertext Transfer Protocol, A protocol used to transfer hypertext requests and information between servers and browsers
Inclusion Arrow: An arrow that points from a scenario to another scenario to show that something must be included for the scenario
Inheritance Arrow: An arrow that points from one use to another; the use of being pointed at is the parent and the other is the sub
Internet Explorer: Web browser designed by Microsoft
Level-0 Diagram: A data flow diagram that represents a system's major processes, data flows, and data stores at a high level
Level-1 Diagram: Provides an overview of the major functional areas of the undertaking
Mozilla Firefox: Web browser designed by Mozilla Foundation and the Mozilla Corporation
Non-Functional Requirements Inventory: Requirements that are not necessarily specific features that exist in a system, but what the system is intended to do
Notepad++: Text editor specializing in syntactic highlighting of various programming languages
ODBC: Open Database Connectivity is a standard API used for connecting to database
Oracle Database: An object-relational database management system produced and marketed by Oracle Corporation
Oraserv Database: Siena College's database server
Participation Line: Shows what scenarios an actor can interact with in a UML Use Case Diagram
Process: Transforms or manipulates data in a Data Flow Diagram
Prototype: An early sample, model or release of a product built to test a concept

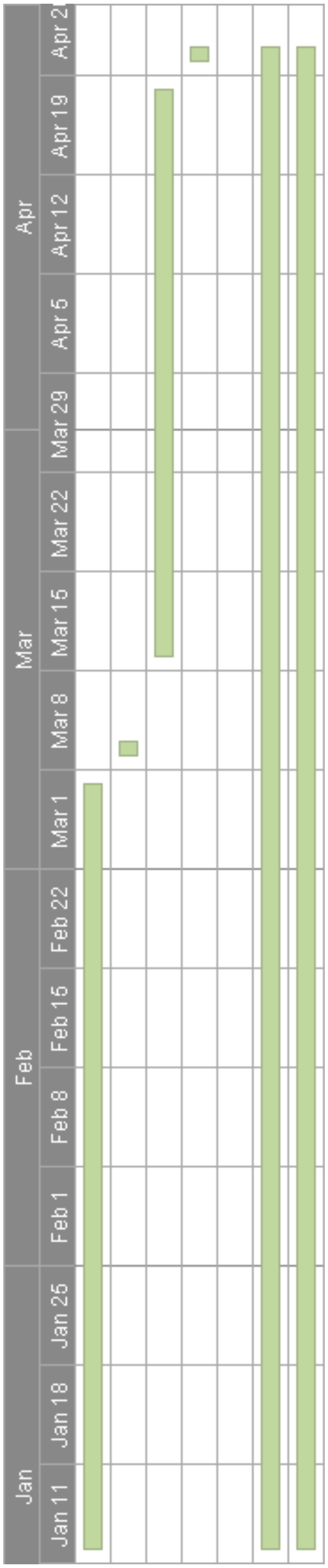
Requirements Specification: Further defining the client's problem to meet the specifications and requirements
Scenarios: The actions that occur within a system and how the user interacts with the system
SQL: Structured Query Language, language used to query databases
System Boundary: The boundary between the system and the external entities in a Data Flow Diagram
UML Use Case Diagram: A type of behavioral diagram to present a graphical overview of the functionality provided by a system
UML (Unified Modeling Language): A specification language used in software engineering
Unit Testing; A testing method where the system is broken down into units and each unit is tested
UPC (User Permission Chart): Chart that demonstrates the permissions of the different users in E.A.S.
Website Map: A list of pages of a website accessible to users

11.2. Appendix B: Sources of Information

The primary source of information necessary for E.A.S. will come from Maroon Solutions' client, Ms. Partridge-Brown and Ms. Sandler. The supervisor, Dr. Fryling, will provide extra information and help in class, while the supervisor, Dr. Lim, will give Maroon Solutions instructions through labs. Maroon Solutions will also utilize the information provided through credible sources on the World Wide Web.

Task	Duration	Start	Finish
Establish Team	1 day	9/5/14	9/5/14
Software Plan	10 days	9/6/14	9/19/14
Software Plan Due	1 day	9/19/14	9/19/01
Software Plan Presentation	1 day	9/23/14	9/23/14
Requirement Specifications	26 days	9/23/14	10/28/14
Required Document Due Date	1 day	10/28/14	10/28/14
Requirement Presentation	1 day	10/28/14	10/28/14
Preliminary Design	26 days	10/28/14	12/2/14
Preliminary Design Due Date	1 day	12/2/14	12/3/14
Preliminary Design Presentation	1 day	12/2/14	12/3/14
Team Meetings	63 days	9/5/14	12/2/14
Client Meetings	61 days	9/9/14	12/2/14

11.3. Appendix C: Timeline


Jan
Feb
Mar
Apr
Jan 11
Jan 18
Jan 25
Feb 1
Feb 8
Feb 15
Feb 22
Mar 1
Mar 8
Mar 15
Mar 22
Mar 29
Apr 5
Apr 12
Apr 19
Apr 2