

**E.A.S.**

**(Efficient And Simple)**

**Acceptance Test**

**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

**April 20th 2015**

Table of Contents

1. Introduction

[1.1 Product Overview and Summary 7](#_Toc417277177)

[2. Requirements Inventory 8](#_Toc417277179)

[2.1. User Case Narratives 8](#_Toc417277180)

[2.1.1. Employee 8](#_Toc417277181)

[2.1.2. Administrator 8](#_Toc417277183)

[2.2. Functional Requirements Inventory 8](#_Toc417277184)

[2.2.1. Employee 8](#_Toc417277186)

[2.2.2. Administrator 8](#_Toc417277198)

[2.3. Non-Functional Requirements Inventory 9](#_Toc417277204)

[2.4. Future Enhancements 9](#_Toc417277205)

[3. External Design Specification 10](#_Toc417277206)

[3.1. User Displays & Report Formats 10](#_Toc417277207)

[3.2. Logical Data Dictionary 10](#_Toc417277208)

[3.3. Logical Format of Data Files and Databases (ER Diagrams) 12](#_Toc417277210)

[3.4. Relational Schema 13](#_Toc417277212)

[3.5. Database Tables 14](#_Toc417277214)

[4. Architectural Design Specification 24](#_Toc417277215)

[4.1. UML Deployment Diagram 24](#_Toc417277216)

[4.1.1. Deployment Diagram Legend 24](#_Toc417277218)

[4.1.2. Deployment Diagram 25](#_Toc417277230)

[4.2. Website Map 26](#_Toc417277231)

[4.2.1. Website Map Legend 26](#_Toc417277232)

[4.2.2. Website Map: Context 27](#_Toc417277239)

[4.2.3. Website Map: Employee 28](#_Toc417277240)

[4.2.4. Website Map: Administrator 29](#_Toc417277241)

[4.3. Development & Production Environments 30](#_Toc417277242)

[4.4. Deliverables 31](#_Toc417277243)

[4.4.1. A CD-ROM (DVD or Flash Drive) with the following: 31](#_Toc417277244)

[4.4.2. Blackboard Submissions 31](#_Toc417277245)

[4.4.3. Hard copies of the Acceptance Test document 31](#_Toc417277249)

[4.4.4. Hard copies of Acceptance Test PowerPoint presentation 31](#_Toc417277250)

[4.4.5. Evaluation forms, work logs, and (for team leads only) attendance sheets 31](#_Toc417277251)

[4.5. Data Flow Diagrams (reference appendix) 31](#_Toc417277252)

[4.6. Source Code (reference deliverables CD) 31](#_Toc417277253)

[5. Test Requirements & Results 32](#_Toc417277254)

[5.1. Explanation of Test Plan/Strategy 32](#_Toc417277255)

[5.2. Test Results (reference appendix) 32](#_Toc417277256)

[6. Appendix 33](#_Toc417277257)

[6.1. Time-Line 33](#_Toc417277258)

[6.2. Glossary of Terms 34](#_Toc417277259)

[6.3. Sources of Information 35](#_Toc417277260)

[6.4. Data Flow Diagrams 36](#_Toc417277261)

[6.4.1. Data Flow Diagram Legend 36](#_Toc417277263)

[6.4.2. Data Flow Diagram: Context Diagram 37](#_Toc417277268)

[6.4.3. Data Flow Diagram: Level 0 Diagram 38](#_Toc417277269)

[6.4.4. Data Flow Diagram: Level 1 Diagrams 39](#_Toc417277270)

[6.5. Test Results 45](#_Toc417277271)

**1.** Introduction

## 1.1 Product Overview and Summary

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. Requirements Inventory

## 2.1. User Case Narratives

### 2.1.1. 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 store’s 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 customer’s purchase history. The employee can also create, add, edit, or delete donor and customer profiles.

### 2.1.2. 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.

## 2.2. 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.

### 2.2.1. 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

### 2.2.2. 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

## 2.3. Non-Functional Requirements Inventory

Non-functional requirements are requirements to E.A.S. that are not specific features in regards to functionality of the program. Instead they are either broad concepts to be implemented or descriptions of how the program will run.

* E.A.S. will be easy to use
* E.A.S. will be efficient
* E.A.S. will require little maintenance
* E.A.S. will user friendly

## 2.4. Future Enhancements

One foreseeable modification that Maroon Solutions is looking into is our idea of having a history of changes to the system. This will keep track of who made changes to store inventory or to customer profiles or donor profiles. Currently we are maintaining a three month history log, but in the future we may look into expanding the history log for longer. If there are other modifications that our clients, Grassroots Givers, would like to see we will surely do our best and implement them.

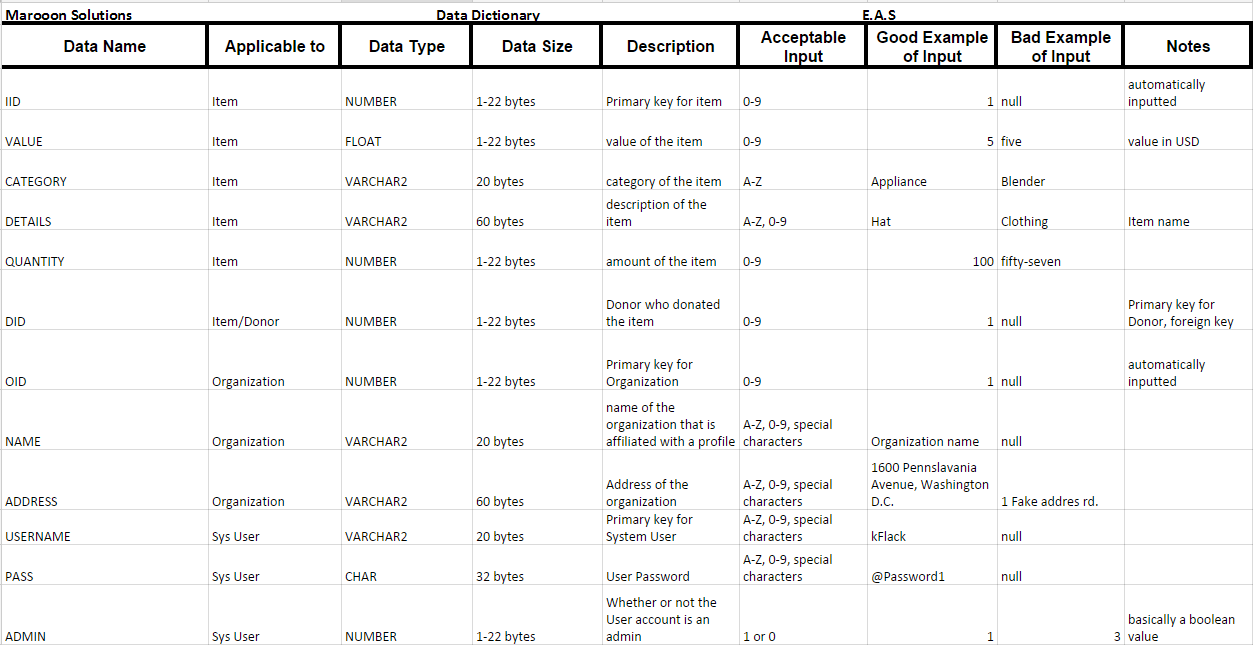
# 3. External Design Specification

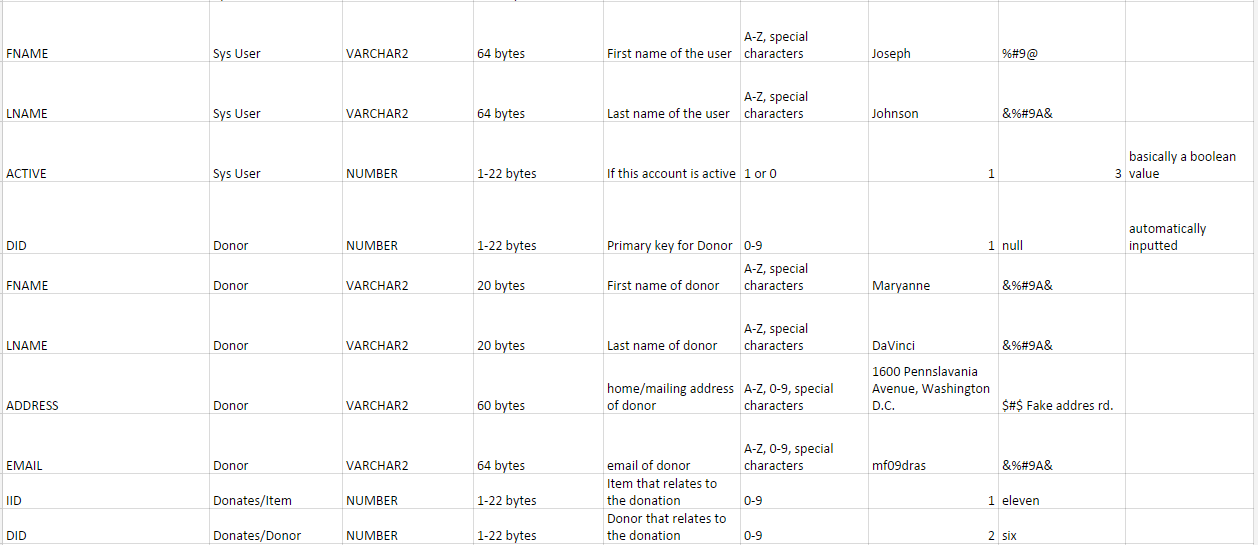
## 3.1. User Displays & Report Formats

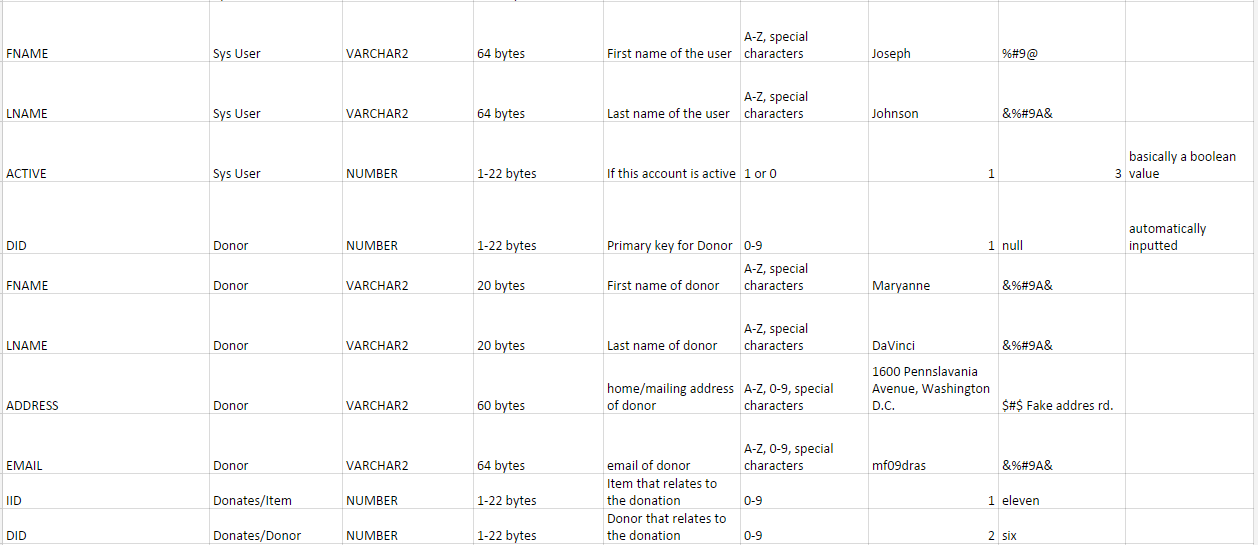
The user displays vary between two different layouts. The administrative layout consists two functions that the employee layout does not—the administrative layout has the ability to view employees in the system and edit prices of items in the system. We are reporting this information and differentiating it through our Data Dictionary, Entity-Relationship Diagram, Relational Schema, and Database tables.

## 3.2. Logical 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.

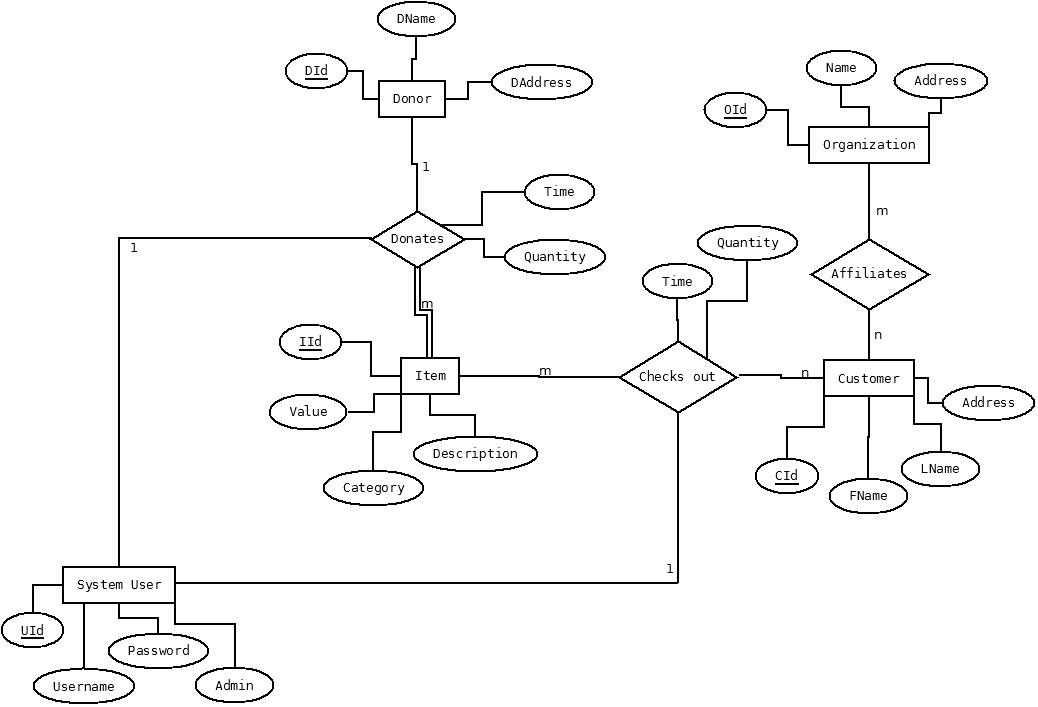






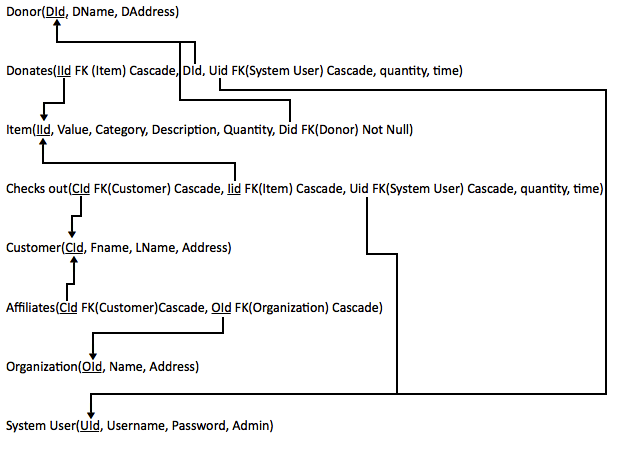
## 3.3. Logical Format of Data Files and Databases (ER Diagrams)

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.



## 3.4. Relational Schema

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



## 3.5. Database Tables

--------------------------------------------------------

-- DDL for Table AFFILIATES

--------------------------------------------------------

CREATE TABLE "PERM\_MAROON"."AFFILIATES"

( "CID" NUMBER,

"OID" NUMBER

) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Table CHECKSOUT

--------------------------------------------------------

CREATE TABLE "PERM\_MAROON"."CHECKSOUT"

( "CID" NUMBER,

"IID" NUMBER,

"USERNAME" VARCHAR2(20 BYTE),

"QUANTITY" NUMBER,

"TIME" DATE DEFAULT sysdate,

"CHECKSOUTID" NUMBER

) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Table CUSTOMER

--------------------------------------------------------

CREATE TABLE "PERM\_MAROON"."CUSTOMER"

( "CID" NUMBER,

"FNAME" VARCHAR2(20 BYTE),

"LNAME" VARCHAR2(20 BYTE),

"ADDRESS" VARCHAR2(60 BYTE)

) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Table DONATES

--------------------------------------------------------

CREATE TABLE "PERM\_MAROON"."DONATES"

( "IID" NUMBER,

"DID" NUMBER,

"USERID" VARCHAR2(20 BYTE),

"QUANTITY" VARCHAR2(20 BYTE),

"TIME" DATE DEFAULT sysdate,

"DONATESID" NUMBER

) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Table DONOR

--------------------------------------------------------

CREATE TABLE "PERM\_MAROON"."DONOR"

( "DID" NUMBER,

"FNAME" VARCHAR2(20 BYTE),

"LNAME" VARCHAR2(20 BYTE),

"ADDRESS" VARCHAR2(60 BYTE)

) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Table ITEM

--------------------------------------------------------

CREATE TABLE "PERM\_MAROON"."ITEM"

( "IID" NUMBER,

"VALUE" FLOAT(126),

"CATEGORY" VARCHAR2(20 BYTE),

"DETAILS" VARCHAR2(60 BYTE),

"QUANTITY" NUMBER,

"DID" NUMBER

) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Table ORGANIZATION

--------------------------------------------------------

CREATE TABLE "PERM\_MAROON"."ORGANIZATION"

( "OID" NUMBER,

"NAME" VARCHAR2(20 BYTE),

"ADDRESS" VARCHAR2(60 BYTE)

) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Table SYSUSER

--------------------------------------------------------

CREATE TABLE "PERM\_MAROON"."SYSUSER"

( "USERNAME" VARCHAR2(20 BYTE),

"PASS" CHAR(32 BYTE),

"ADMIN" NUMBER(1,0) DEFAULT 0,

"FNAME" VARCHAR2(64 BYTE),

"LNAME" VARCHAR2(64 BYTE)

) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Index AFFILIATES\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "PERM\_MAROON"."AFFILIATES\_PK" ON "PERM\_MAROON"."AFFILIATES" ("CID", "OID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Index CHECKSOUT\_PK1

--------------------------------------------------------

CREATE UNIQUE INDEX "PERM\_MAROON"."CHECKSOUT\_PK1" ON "PERM\_MAROON"."CHECKSOUT" ("CHECKSOUTID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Index CUSTOMER\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "PERM\_MAROON"."CUSTOMER\_PK" ON "PERM\_MAROON"."CUSTOMER" ("CID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Index DONATES\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "PERM\_MAROON"."DONATES\_PK" ON "PERM\_MAROON"."DONATES" ("DONATESID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Index DONOR\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "PERM\_MAROON"."DONOR\_PK" ON "PERM\_MAROON"."DONOR" ("DID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Index ITEM\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "PERM\_MAROON"."ITEM\_PK" ON "PERM\_MAROON"."ITEM" ("IID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Index ORGANIZATION\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "PERM\_MAROON"."ORGANIZATION\_PK" ON "PERM\_MAROON"."ORGANIZATION" ("OID")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- DDL for Index SYSUSER\_PK

--------------------------------------------------------

CREATE UNIQUE INDEX "PERM\_MAROON"."SYSUSER\_PK" ON "PERM\_MAROON"."SYSUSER" ("USERNAME")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ;

--------------------------------------------------------

-- Constraints for Table AFFILIATES

--------------------------------------------------------

ALTER TABLE "PERM\_MAROON"."AFFILIATES" ADD CONSTRAINT "AFFILIATES\_PK" PRIMARY KEY ("CID", "OID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ENABLE;

ALTER TABLE "PERM\_MAROON"."AFFILIATES" MODIFY ("CID" NOT NULL ENABLE);

ALTER TABLE "PERM\_MAROON"."AFFILIATES" MODIFY ("OID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table CHECKSOUT

--------------------------------------------------------

ALTER TABLE "PERM\_MAROON"."CHECKSOUT" ADD CONSTRAINT "CHECKSOUT\_PK" PRIMARY KEY ("CHECKSOUTID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ENABLE;

ALTER TABLE "PERM\_MAROON"."CHECKSOUT" MODIFY ("CID" NOT NULL ENABLE);

ALTER TABLE "PERM\_MAROON"."CHECKSOUT" MODIFY ("IID" NOT NULL ENABLE);

ALTER TABLE "PERM\_MAROON"."CHECKSOUT" MODIFY ("CHECKSOUTID" NOT NULL ENABLE);

ALTER TABLE "PERM\_MAROON"."CHECKSOUT" MODIFY ("USERNAME" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table CUSTOMER

--------------------------------------------------------

ALTER TABLE "PERM\_MAROON"."CUSTOMER" ADD CONSTRAINT "CUSTOMER\_PK" PRIMARY KEY ("CID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ENABLE;

ALTER TABLE "PERM\_MAROON"."CUSTOMER" MODIFY ("CID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table DONATES

--------------------------------------------------------

ALTER TABLE "PERM\_MAROON"."DONATES" ADD CONSTRAINT "DONATES\_PK" PRIMARY KEY ("DONATESID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ENABLE;

ALTER TABLE "PERM\_MAROON"."DONATES" MODIFY ("IID" NOT NULL ENABLE);

ALTER TABLE "PERM\_MAROON"."DONATES" MODIFY ("DONATESID" NOT NULL ENABLE);

ALTER TABLE "PERM\_MAROON"."DONATES" MODIFY ("DID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table DONOR

--------------------------------------------------------

ALTER TABLE "PERM\_MAROON"."DONOR" ADD CONSTRAINT "DONOR\_PK" PRIMARY KEY ("DID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ENABLE;

ALTER TABLE "PERM\_MAROON"."DONOR" MODIFY ("DID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table ITEM

--------------------------------------------------------

ALTER TABLE "PERM\_MAROON"."ITEM" ADD CONSTRAINT "ITEM\_PK" PRIMARY KEY ("IID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ENABLE;

ALTER TABLE "PERM\_MAROON"."ITEM" MODIFY ("IID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table ORGANIZATION

--------------------------------------------------------

ALTER TABLE "PERM\_MAROON"."ORGANIZATION" ADD CONSTRAINT "ORGANIZATION\_PK" PRIMARY KEY ("OID")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ENABLE;

ALTER TABLE "PERM\_MAROON"."ORGANIZATION" MODIFY ("OID" NOT NULL ENABLE);

--------------------------------------------------------

-- Constraints for Table SYSUSER

--------------------------------------------------------

ALTER TABLE "PERM\_MAROON"."SYSUSER" ADD CONSTRAINT "SYSUSER\_PK" PRIMARY KEY ("USERNAME")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT)

TABLESPACE "USERS" ENABLE;

ALTER TABLE "PERM\_MAROON"."SYSUSER" MODIFY ("USERNAME" NOT NULL ENABLE);

ALTER TABLE "PERM\_MAROON"."SYSUSER" MODIFY ("PASS" NOT NULL ENABLE);

--------------------------------------------------------

-- Ref Constraints for Table AFFILIATES

--------------------------------------------------------

ALTER TABLE "PERM\_MAROON"."AFFILIATES" ADD CONSTRAINT "AFFILIATES\_FK1" FOREIGN KEY ("CID")

REFERENCES "PERM\_MAROON"."CUSTOMER" ("CID") ON DELETE CASCADE ENABLE;

ALTER TABLE "PERM\_MAROON"."AFFILIATES" ADD CONSTRAINT "AFFILIATES\_FK2" FOREIGN KEY ("OID")

REFERENCES "PERM\_MAROON"."ORGANIZATION" ("OID") ON DELETE CASCADE ENABLE;

--------------------------------------------------------

-- Ref Constraints for Table CHECKSOUT

--------------------------------------------------------

ALTER TABLE "PERM\_MAROON"."CHECKSOUT" ADD CONSTRAINT "CHECKSOUT\_FK1" FOREIGN KEY ("CID")

REFERENCES "PERM\_MAROON"."CUSTOMER" ("CID") ON DELETE CASCADE ENABLE;

ALTER TABLE "PERM\_MAROON"."CHECKSOUT" ADD CONSTRAINT "CHECKSOUT\_FK2" FOREIGN KEY ("IID")

REFERENCES "PERM\_MAROON"."ITEM" ("IID") ON DELETE CASCADE ENABLE;

ALTER TABLE "PERM\_MAROON"."CHECKSOUT" ADD CONSTRAINT "CHECKSOUT\_FK3" FOREIGN KEY ("USERNAME")

REFERENCES "PERM\_MAROON"."SYSUSER" ("USERNAME") ON DELETE CASCADE ENABLE;

--------------------------------------------------------

-- Ref Constraints for Table DONATES

--------------------------------------------------------

ALTER TABLE "PERM\_MAROON"."DONATES" ADD CONSTRAINT "DONATES\_FK1" FOREIGN KEY ("IID")

REFERENCES "PERM\_MAROON"."ITEM" ("IID") ON DELETE CASCADE ENABLE;

--------------------------------------------------------

-- Ref Constraints for Table ITEM

--------------------------------------------------------

ALTER TABLE "PERM\_MAROON"."ITEM" ADD CONSTRAINT "ITEM\_FK1" FOREIGN KEY ("DID")

REFERENCES "PERM\_MAROON"."DONOR" ("DID") ENABLE;

--------------------------------------------------------

-- DDL for Trigger CHECKSOUT\_TRG

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "PERM\_MAROON"."CHECKSOUT\_TRG"

BEFORE INSERT ON CHECKSOUT

FOR EACH ROW

BEGIN

<<COLUMN\_SEQUENCES>>

BEGIN

IF INSERTING AND :NEW.CHECKSOUTID IS NULL THEN

SELECT CHECKSOUT\_SEQ.NEXTVAL INTO :NEW.CHECKSOUTID FROM SYS.DUAL;

END IF;

END COLUMN\_SEQUENCES;

END;

/

ALTER TRIGGER "PERM\_MAROON"."CHECKSOUT\_TRG" ENABLE;

--------------------------------------------------------

-- DDL for Trigger DONATES\_TRG

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "PERM\_MAROON"."DONATES\_TRG"

BEFORE INSERT ON DONATES

FOR EACH ROW

BEGIN

<<COLUMN\_SEQUENCES>>

BEGIN

NULL;

END COLUMN\_SEQUENCES;

END; /

ALTER TRIGGER "PERM\_MAROON"."DONATES\_TRG" ENABLE;

--------------------------------------------------------

-- DDL for Trigger DONATES\_TRG1

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "PERM\_MAROON"."DONATES\_TRG1"

BEFORE INSERT ON DONATES

FOR EACH ROW

BEGIN

<<COLUMN\_SEQUENCES>>

BEGIN

IF INSERTING AND :NEW.DONATESID IS NULL THEN

SELECT DONATES\_SEQ1.NEXTVAL INTO :NEW.DONATESID FROM SYS.DUAL;

END IF;

END COLUMN\_SEQUENCES;

END;

/

ALTER TRIGGER "PERM\_MAROON"."DONATES\_TRG1" ENABLE;

--------------------------------------------------------

-- DDL for Trigger ITEM\_TRG

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "PERM\_MAROON"."ITEM\_TRG"

BEFORE INSERT ON ITEM

FOR EACH ROW

BEGIN

<<COLUMN\_SEQUENCES>>

BEGIN

IF INSERTING AND :NEW.IID IS NULL THEN

SELECT ITEM\_SEQ.NEXTVAL INTO :NEW.IID FROM SYS.DUAL;

END IF;

END COLUMN\_SEQUENCES;

END;

/

ALTER TRIGGER "PERM\_MAROON"."ITEM\_TRG" ENABLE;

# 4. Architectural Design Specification

## 4.1. UML Deployment Diagram

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.

### 4.1.1. Deployment Diagram Legend

**System Boundary -** This is where all the interactions occur.

Represents what is within the system and outside of it.

**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** **-** Open Database Connectivity is a standard programming

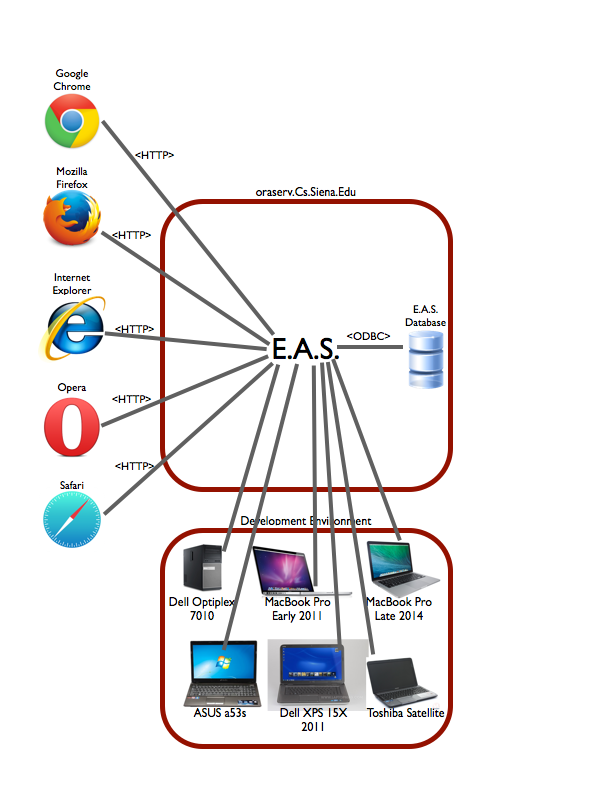
language middleware for accessing database management systems.

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

<HTTP>

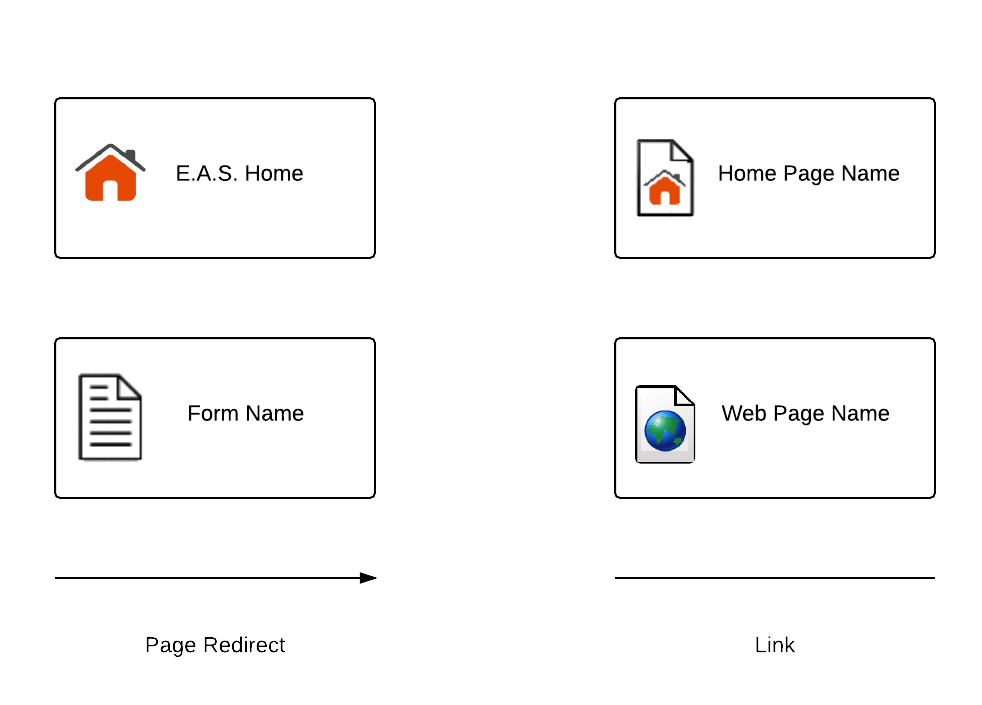
<ODBC>

### 4.1.2. Deployment Diagram



## 4.2. Website Map

### 4.2.1. 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.

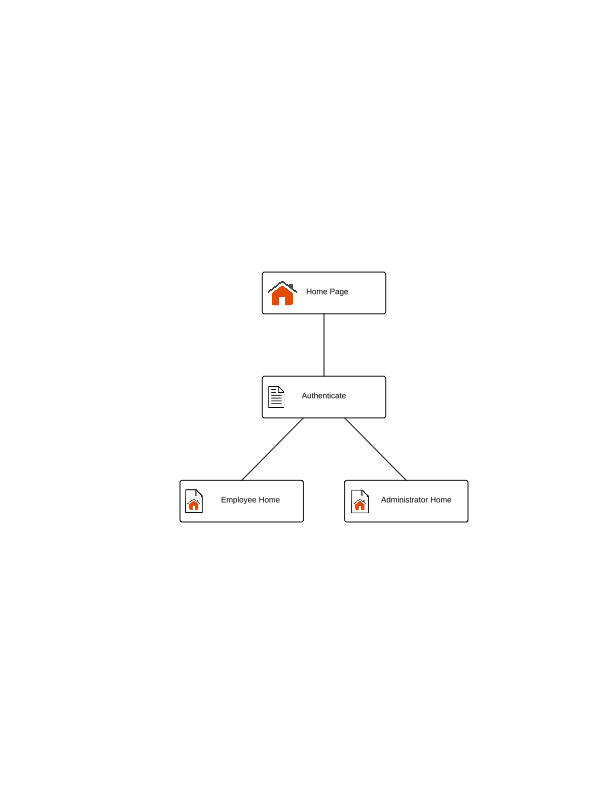
**Form** - Represents data fields that require user input.

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

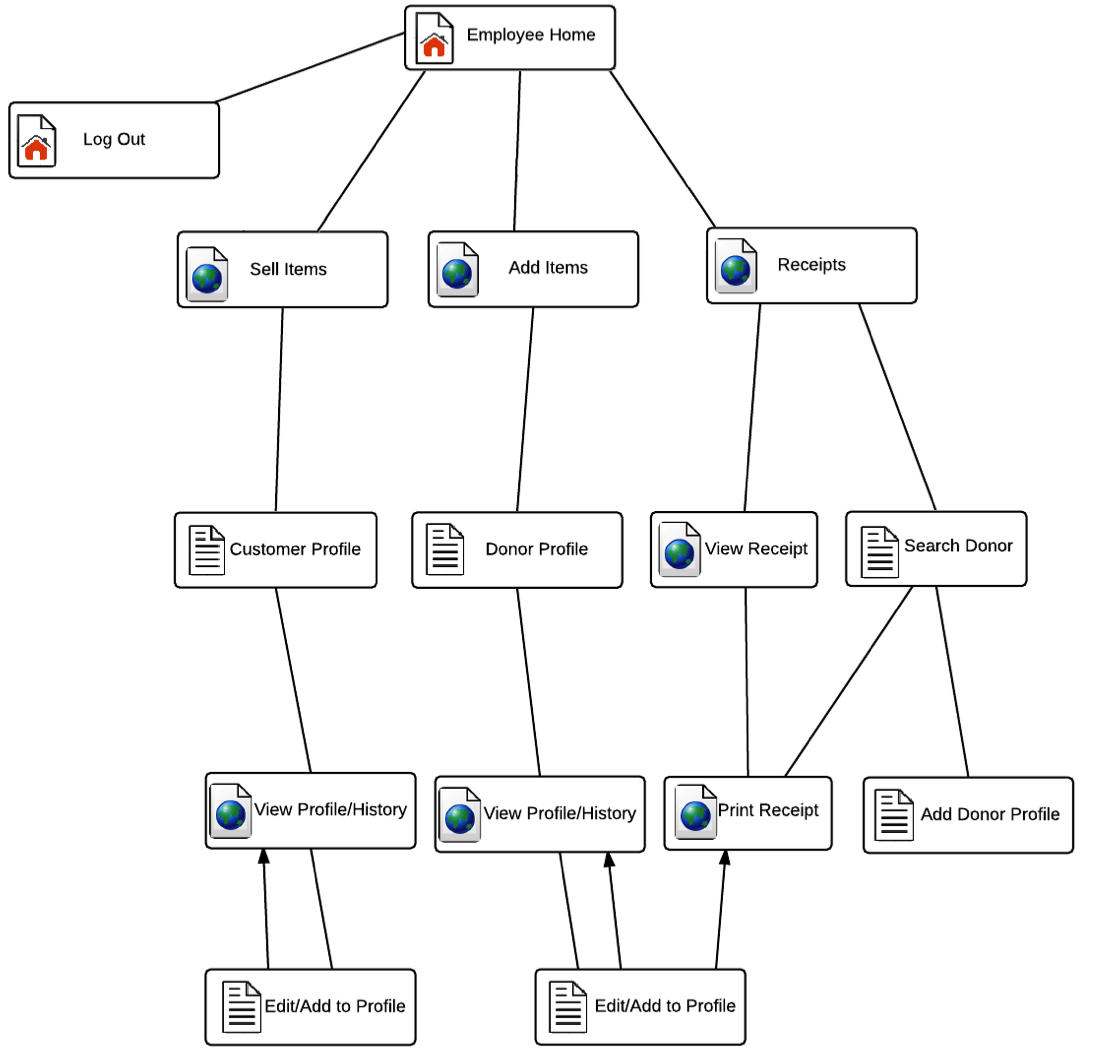
**Link** - Represents a page being accessible from another page.

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

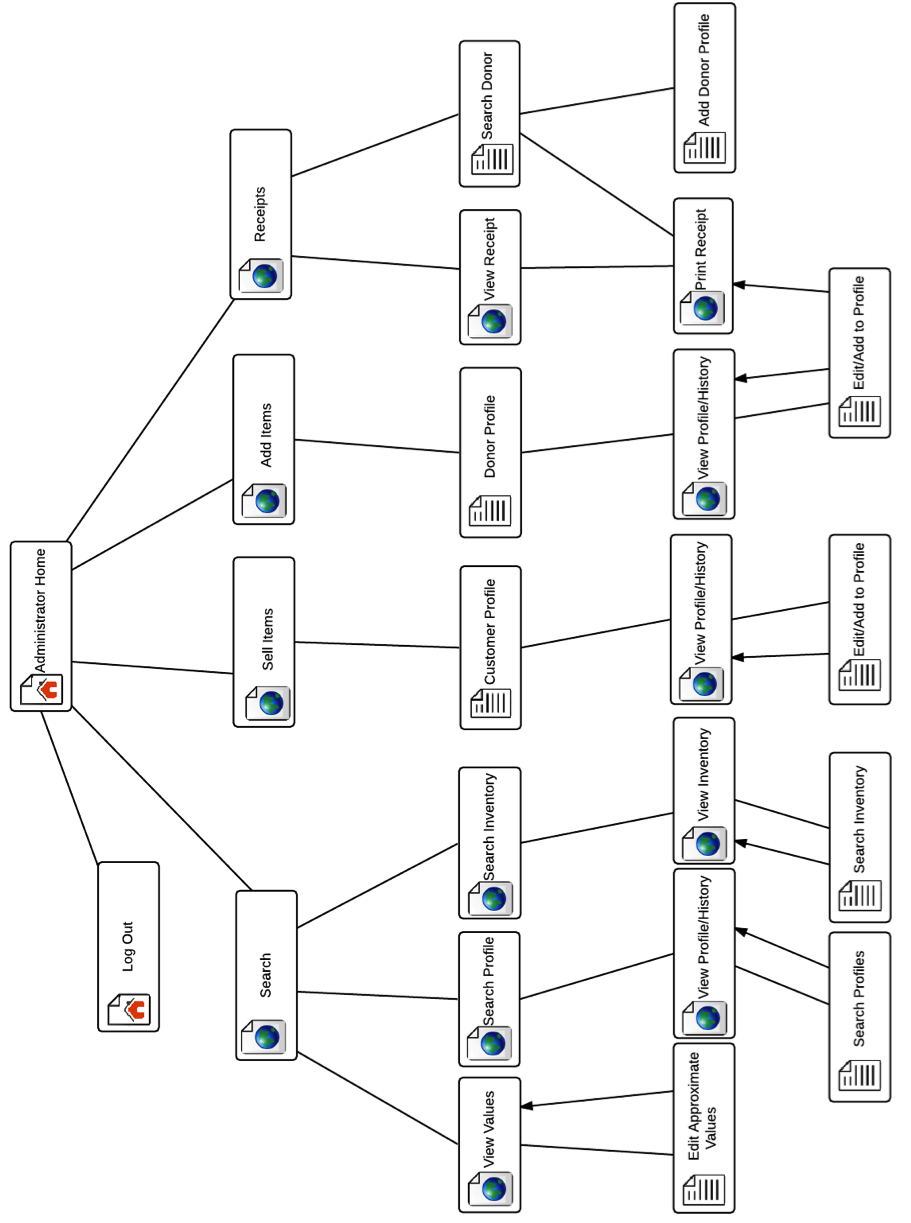
### 4.2.2. Website Map: Context



### 4.2.3. Website Map: Employee

**

### 4.2.4. Website Map: Administrator

******

## 4.3. Development & Production Environments

*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

Safari

Google Chrome

Microsoft Excel, Word, Query, PowerPoint

Coda 2

*Server:*

Hostname: [oraserv.cs.siena.edu/~perm\_maroon/](http://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 Environment (build 1.6.0 10-rc-b28)

GCC Version 4.1.2 20071124 (Red Hat 4.1.2-42)

## 4.4. Deliverables

### 4.4.1. A CD-ROM (DVD or Flash Drive) with the following:

* 1. A full copy of our files from our directory, including, of course, all website files (all folders, files, images, etc).
  2. The above files should, include all files associated with your project, including all documentation (e.g. Software Plan, etc) and PowerPoint presentations.
  3. The website files should reference all “local” URLs as relative links. One subdirectory named maroonSolutions-public\_html (for the team website files) and other subdirectory named grassrootGivers-public\_html (for the project website files).
  4. A README.TXT file that explains what files are where (this README.TXT will be at the highest level directory.
  5. Any needed usernames/passwords (and database names) and documentation related to gaining access to any database(s) that were used as part of our project will be provided.
  6. The lyrics to our team song; a copy of a sound/music file for our team song; and, an audio/video recording of our team song.  These will be placed in a subdirectory named SONG.

### 4.4.2. Blackboard Submissions

* 1. Acceptance Test document
  2. Acceptance Test presentation
  3. Music Video

### 4.4.3. Hard copies of the Acceptance Test document

* 1. 1 for your clients
  2. 1 for Academic Celebration

### 4.4.4. Hard copies of Acceptance Test PowerPoint presentation

1. 1 (or more) for your clients

### 4.4.5. Evaluation forms, work logs, and (for team leads only) attendance sheets

## 4.5. Data Flow Diagrams (reference appendix)

## 4.6. Source Code (reference deliverables CD)

Our source code can also be viewed [here](https://github.com/SienaCollegeSoftwareEngineering/2014-15-Team-Documentation---Maroon-Solutions/tree/master/GrassrootGitHub).

# 5. Test Requirements & Results

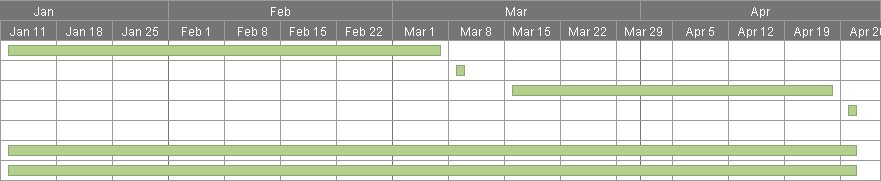
## 5.1. Explanation of Test Plan/Strategy

Maroon Solutions’ strategy was to test each unit on our several planned browsers. Various team members would each in turn go through a unit test with the various browsers, then would present this to the team. Typically, someone on the team would find another test that the original tester did not think of while testing. After testing the new tests, the tester would move on to a different unit test.

## 5.2. Test Results (reference appendix)

# 6. Appendix

## 6.1. Time-Line



## 6.2. 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

## 6.3. 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.

## 6.4. Data Flow Diagrams

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:

### 6.4.1. Data Flow Diagram Legend

**Process:** Transforms or manipulates data.

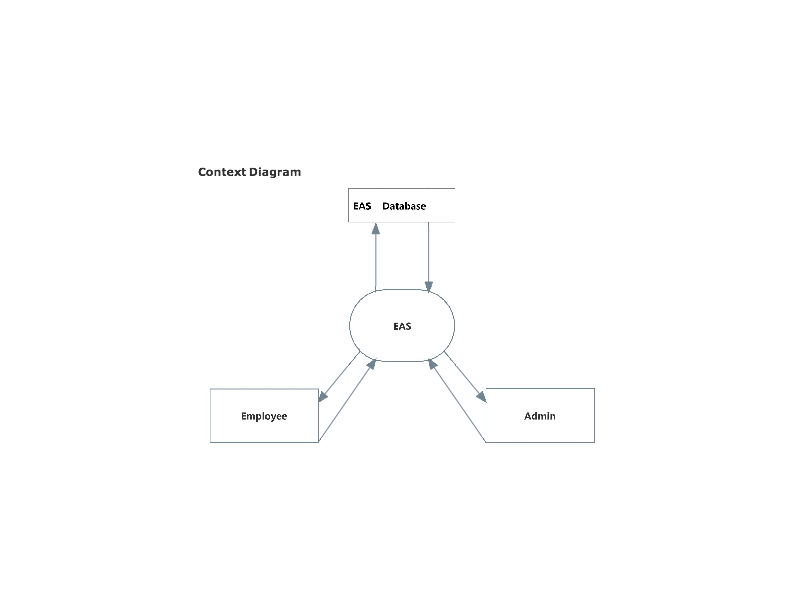
**External Entity:** Contributes data or information to the system or which receive data/information from it.

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

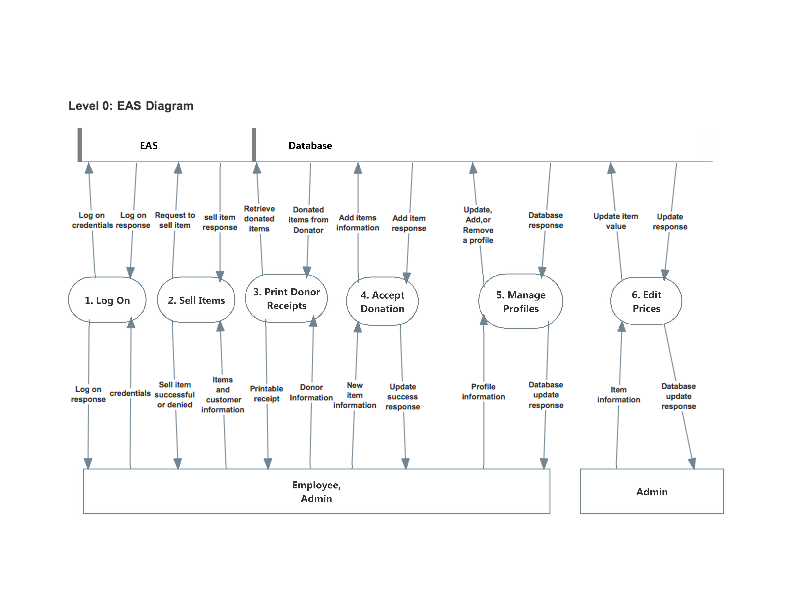
**Data Flow:** Data/information flowing to or from a process where C is the data/information.

C

### 6.4.2. Data Flow Diagram: Context Diagram

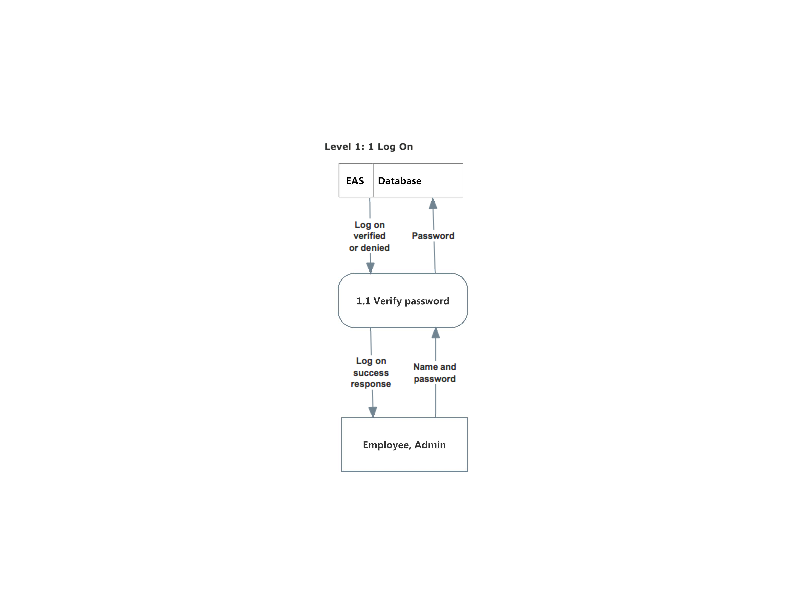


### 6.4.3. Data Flow Diagram: Level 0 Diagram

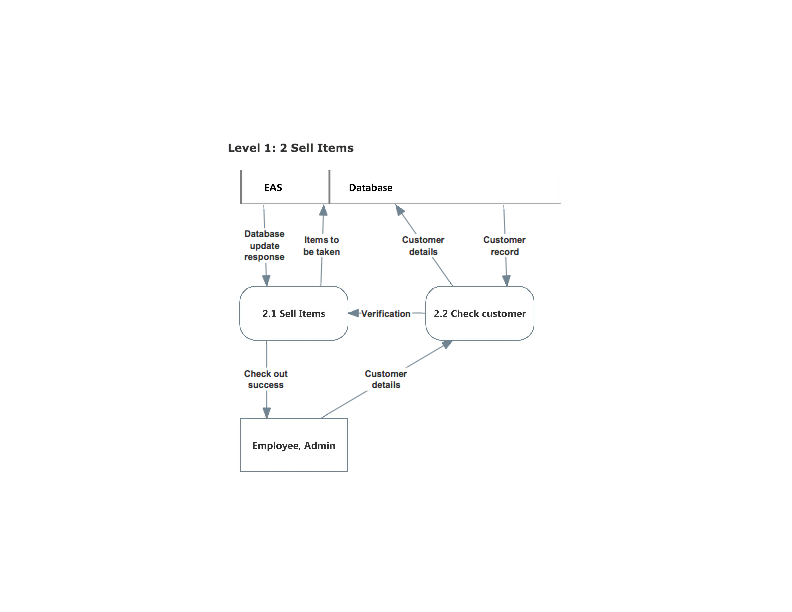
******

### 6.4.4. Data Flow Diagram: Level 1 Diagrams

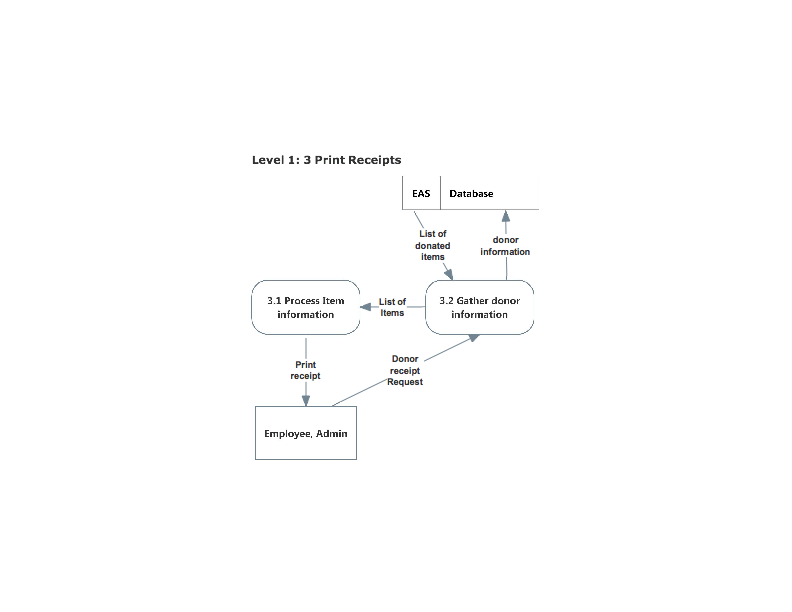
#### 6.4.4.1. Log On



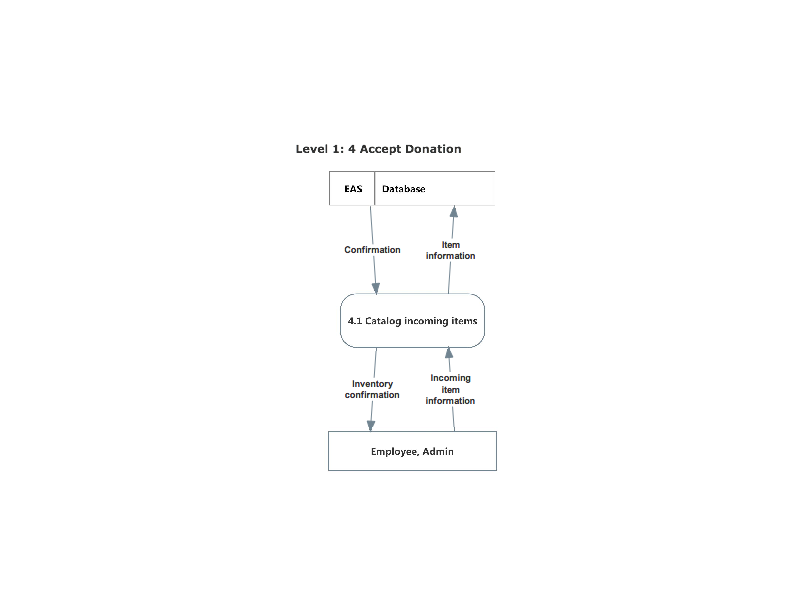
#### 6.4.4.2. Sell Items

****

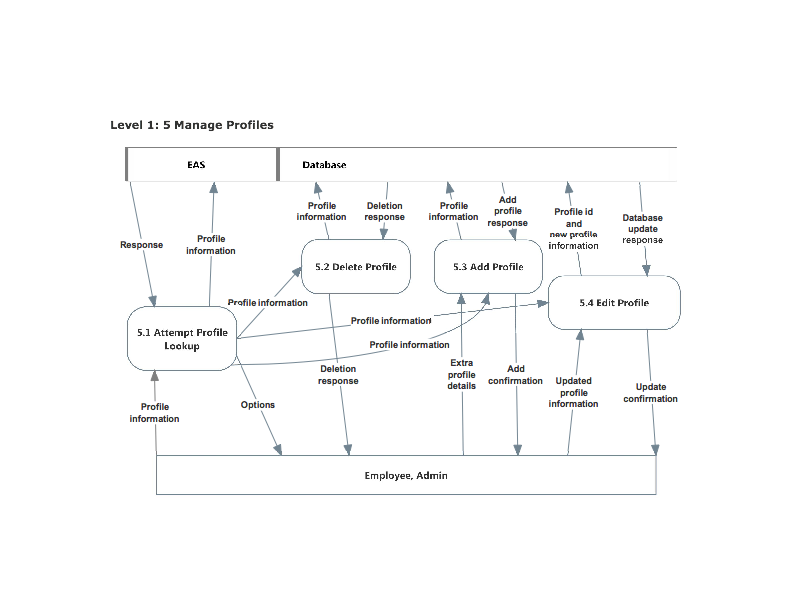
#### 6.4.4.3. Print Receipts



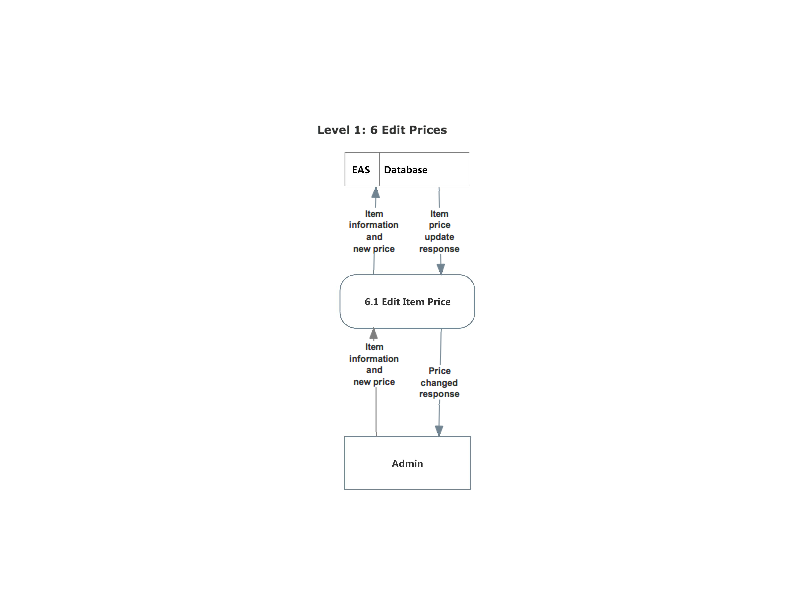
#### 6.4.4.4. Accept Donations

****

#### 6.4.4.5. Manage Profiles



#### 6.4.4.6. Edit Prices



## 6.5. Test Results

Test results and test cases can be viewed [here](https://docs.google.com/spreadsheets/d/1zET04TkNNbp4X_0dqA480Jx9nJxTbaGygAoVX26R0ek/edit?usp=sharing).