Preliminary Design

Requested by: Ms. Mary Partridge-Brown
Ms. Roberta Sandler
Co-Directors
Grassroot Givers' Community Store

SMARK Solutions

Be intuitive. Be efficient. Be SMARK.

Prepared by: Salvatore Baisley - Database Manager
Mary Ritchie - Webmaster
Anna Grant - Team Lead
Ryan Martin - Chief QA/Testing
Kathryn Egan - Lead Programmer

November 25, 2014

S.W.I.F.T. (Simple Web Inventory for Tracking)
Preliminary Design

Contents

1. Product Overview and Summary

Grassroot Givers is a non-profit organization devoted to bridging the gap between those in need and those seeking to donate. One of the ways in which this is achieved is through the Community Store within their facilities at the GWU Center in Albany, NY. The mission of this store is to create a boutique-like atmosphere so that customers can "shop with dignity". Co-directors of Grassroot Givers, Mary Partridge-Brown and Roberta Sandler would like to develop an easy to use, web-based application to supplement their everyday functions of the store. S.W.I.F.T. (Simple Web Inventory For Tracking) is a web-based application that will allow Grassroot Givers to track incoming items through the creation of donor and customer profiles, database searching, and receipts.

2. Use Case Narratives

2.1. Volunteer

The volunteer will login on to S.W.I.F.T. using a unique username and password. The volunteer will have access to a page where the volunteer will choose to either enter a new customer, look up the history of a specific customer, check out a specific customer, or take in a donation. If the volunteer wishes to enter a new customer into the database, there will be a form for creating a customer profile. The form will require the volunteer to enter the name of the customer, address of the customer, number of family members in the customer's household including each member's age, and other agencies which the customer is affiliated with, additionally the date that the profile was created will be stored in the customer's profile. The volunteer will have the ability to search the customer records by name and address to view the customer's profile. The volunteer will have the ability to search the inventory to see the quantity of the items that are in high demand. The volunteer will be able to check out customers, during which the customer's history within the past three months will be reviewed. If the customer is eligible to take the items the customer has selected, the volunteer will record and store the items in the customer's profile along with the date of the transaction, and the name of the volunteer doing the checkout. If the volunteer is taking a donation, the volunteer will record the number of bags and boxes being donated and the contents of the packages. The customer will tell the volunteer whether or not the customer desires a receipt and if so, a form will be filled by the volunteer indicating the items donated and their value. Regardless of whether the customer desires a receipt, the number of boxes and bags will still be recorded. Multiple volunteers will be able to be logged in at once.

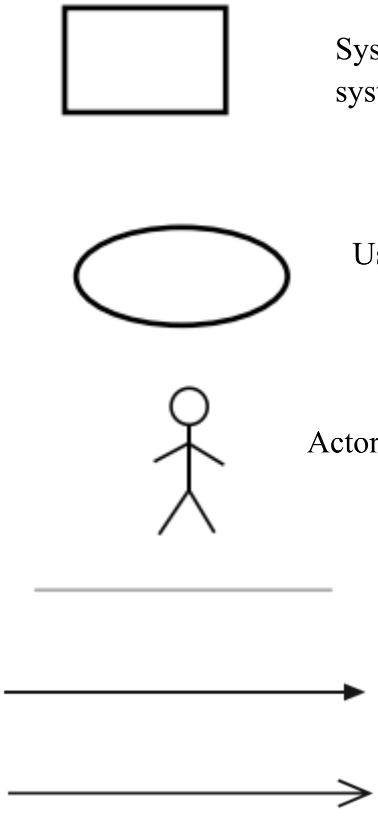
2.2. Director

The director will login on to S.W.I.F.T. using a username and password specific to being a director. The director will have access to a page where the director will choose to either enter a new customer, look up the history of a specific customer, check out a specific customer, editing and deleting customer information, adding a new volunteer, or take in a donation. If the director wishes to enter a new customer into the database, there will be a form for creating a customer profile. The form will require the director to enter the name of the customer, address of the customer, number of family members in the customer's household including each member's age, and other agencies which the customer is affiliated with, additionally the date that the profile was created will be stored in the customer's profile. The director will have the ability to search the customer records by name and address to view, edit, or delete the customer's profile. The director will have the ability to search the inventory to see the quantity of the items that are in high demand. The director will be able to check out customers, during which the customer's history within the past three months will be reviewed. If the customer is eligible to take the items the customer has selected, the director will record and store the items in the customer's profile along with the date of the transaction, and the name of the director doing the checkout. If the director is creating a new volunteer account, the director will submit a form with the volunteer's information to S.W.I.F.T.. If the director is taking a donation, the director will record the number of bags and boxes being donated and the contents of the packages. The customer will tell the director whether or not the customer desires a receipt and if so, a form will be filled by the director indicating the items donated and their value. Regardless of whether the customer desires a receipt, the number of boxes and bags will still be recorded. Multiple directors will be able to be logged in at once.

3. UML Diagrams

3.1. Use Case Diagram

3.1.1. Use Case Legend



System Boundary: where interactions between both use cases inside the system and actors outside the system are shown

Use Case: the activities that actors interact with inside the system

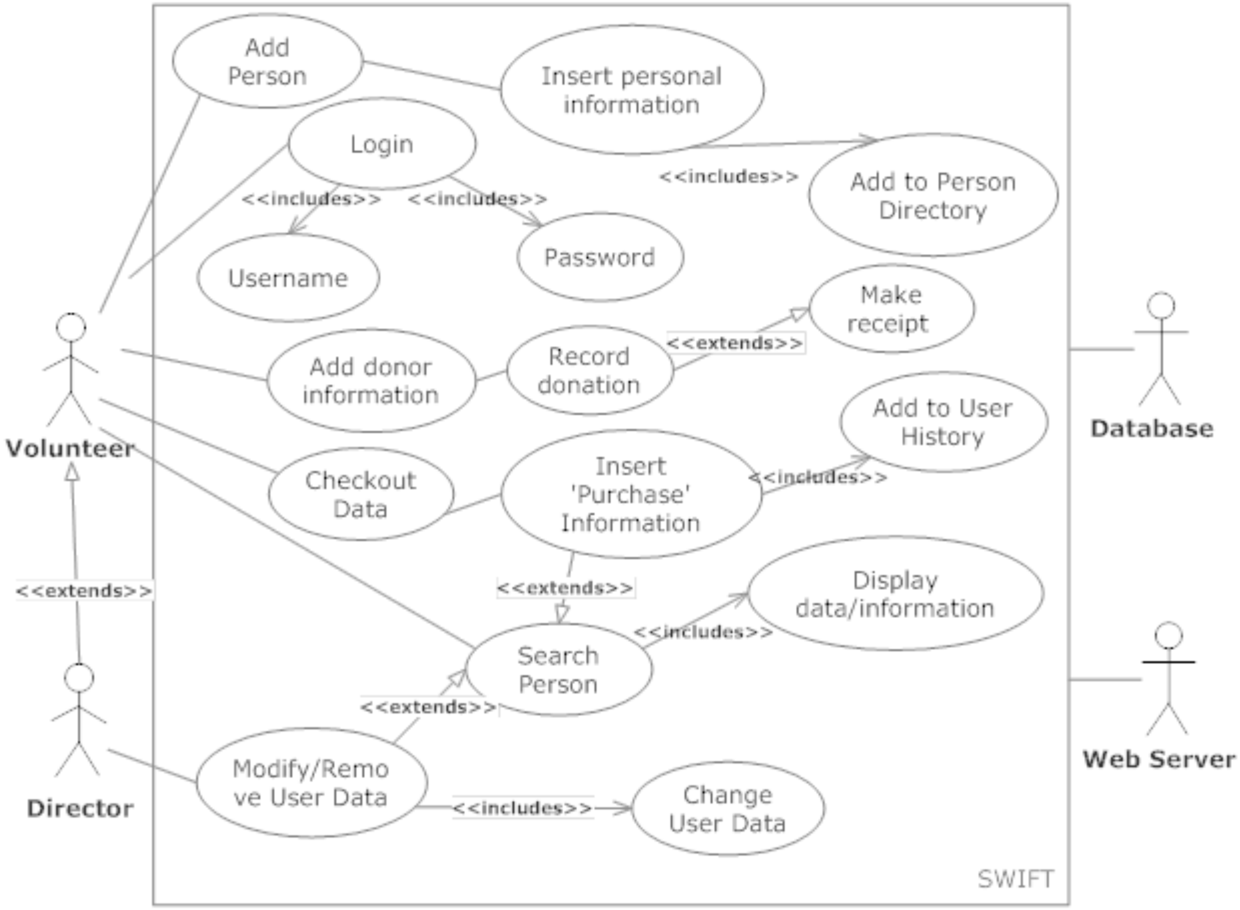
Actor: human or non human users that interact with the system.

Participation line: lines that connect use cases and actors to show what actors participate in

Extends: used to represent items that may be included in a use case

Includes: used to represent items that must be included in a use case

3.1.2. UML Use Case Diagram



3.2. Deployment Diagram

3.2.1. Deployment Legend

<HTTP> Hypertext Transfer Protocol - application protocol used in distributing, collaborating, and hypermedia information systems.

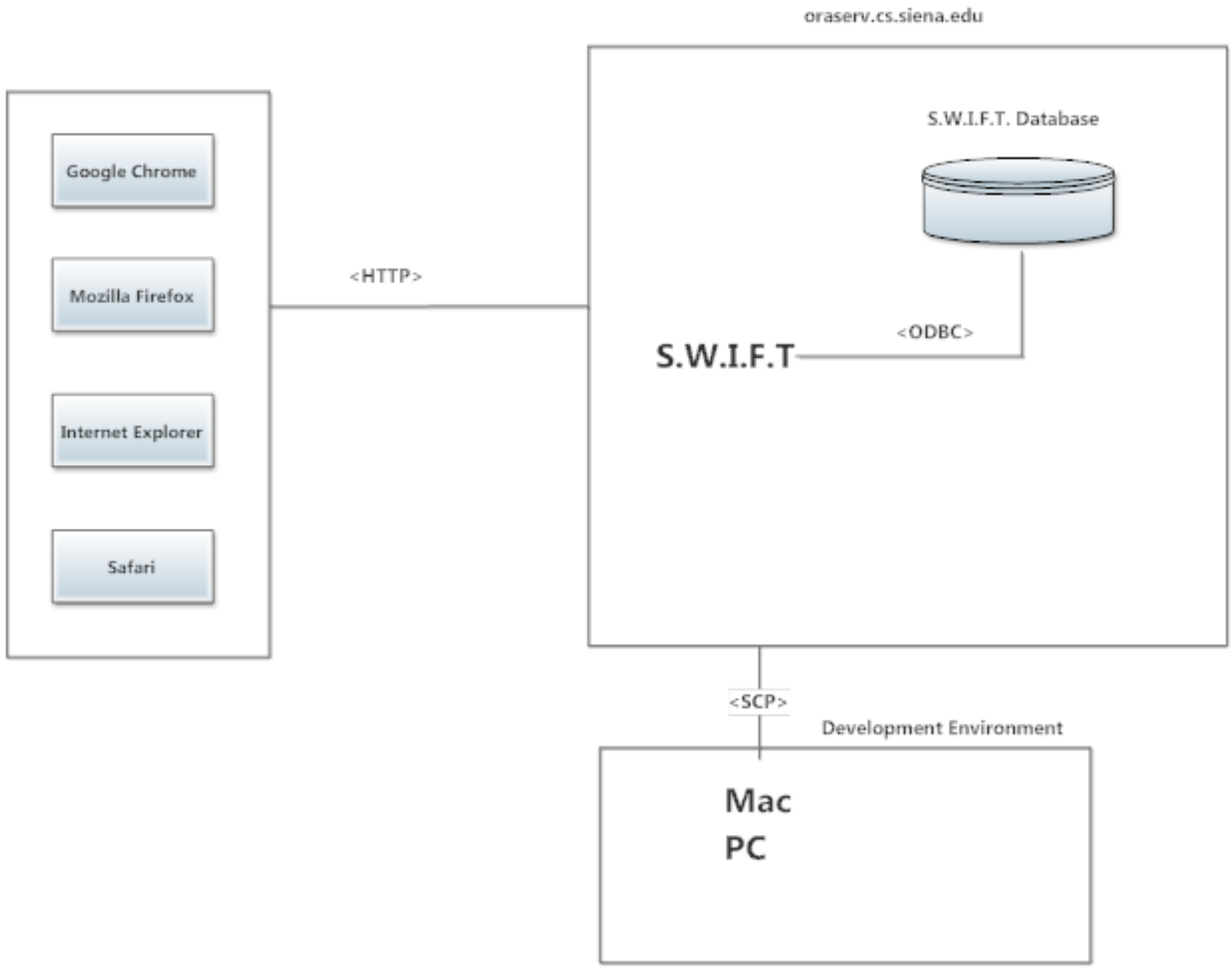
<SCP> Secure Copy - A way to securely transfer computer files between a local host and a remote host.

<ODBC> Open Database Connectivity - Standard programming language API for DBMS.

□ System Boundary - Represents the divide between the inside of the system where the interactions occur and the outside.

─ Connection - Represents a relation between system boundaries.

3.2.2. Deployment Diagram



3.3. Activity Diagrams

3.3.1. Activity Diagram Legend

● Initial Node - The starting point for the movement through the activity.

◉ Final Node - The ending point, where the process is over.

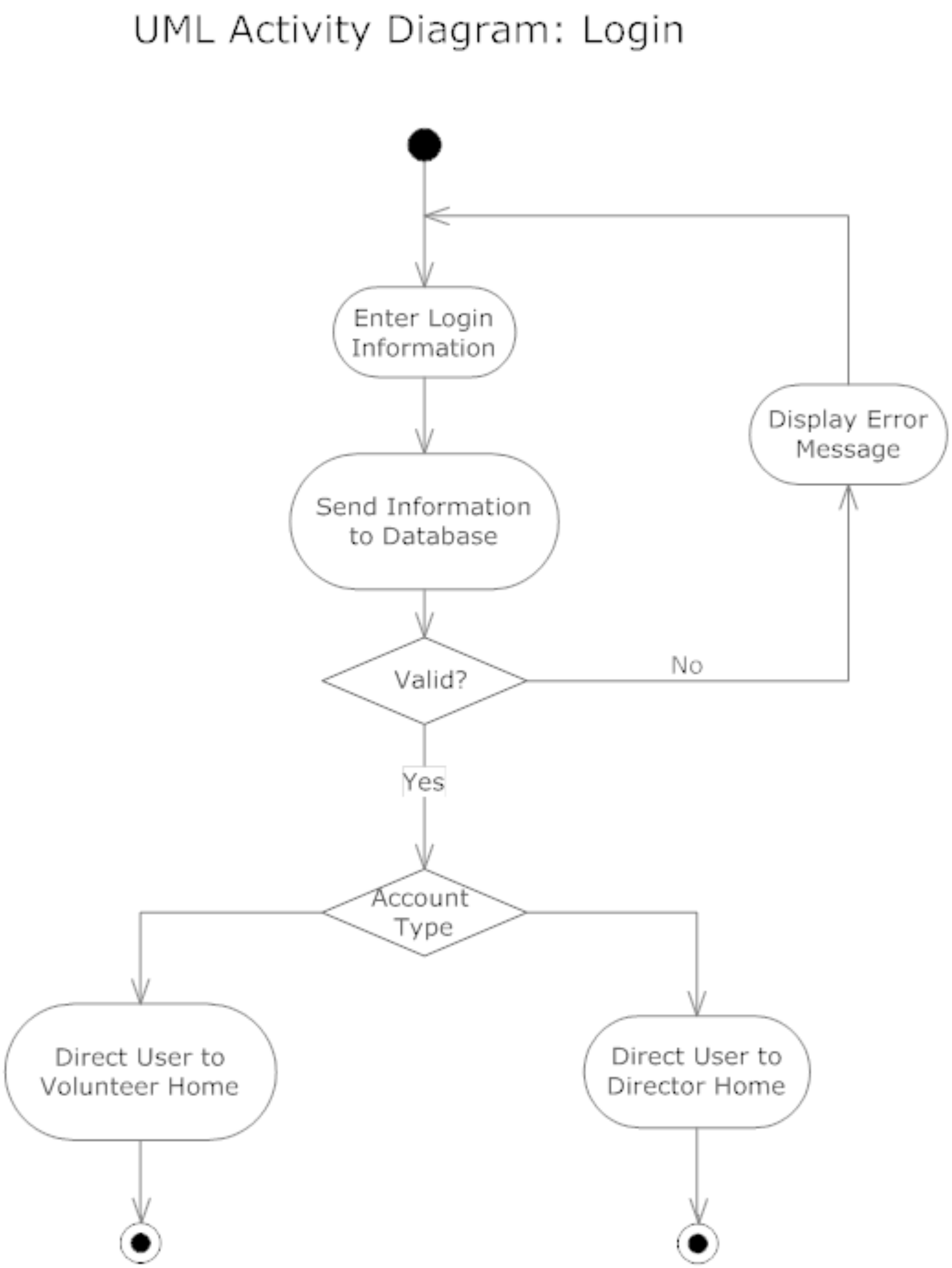
◇ Decision Node - Represents a branching in the flow of the activity. Each branch represents a unique answer.

▢ Activity Node - Describes the next step completed or that needs to be completed by the system to continue.

▬ Split/Join - Separates or joins several different flows.

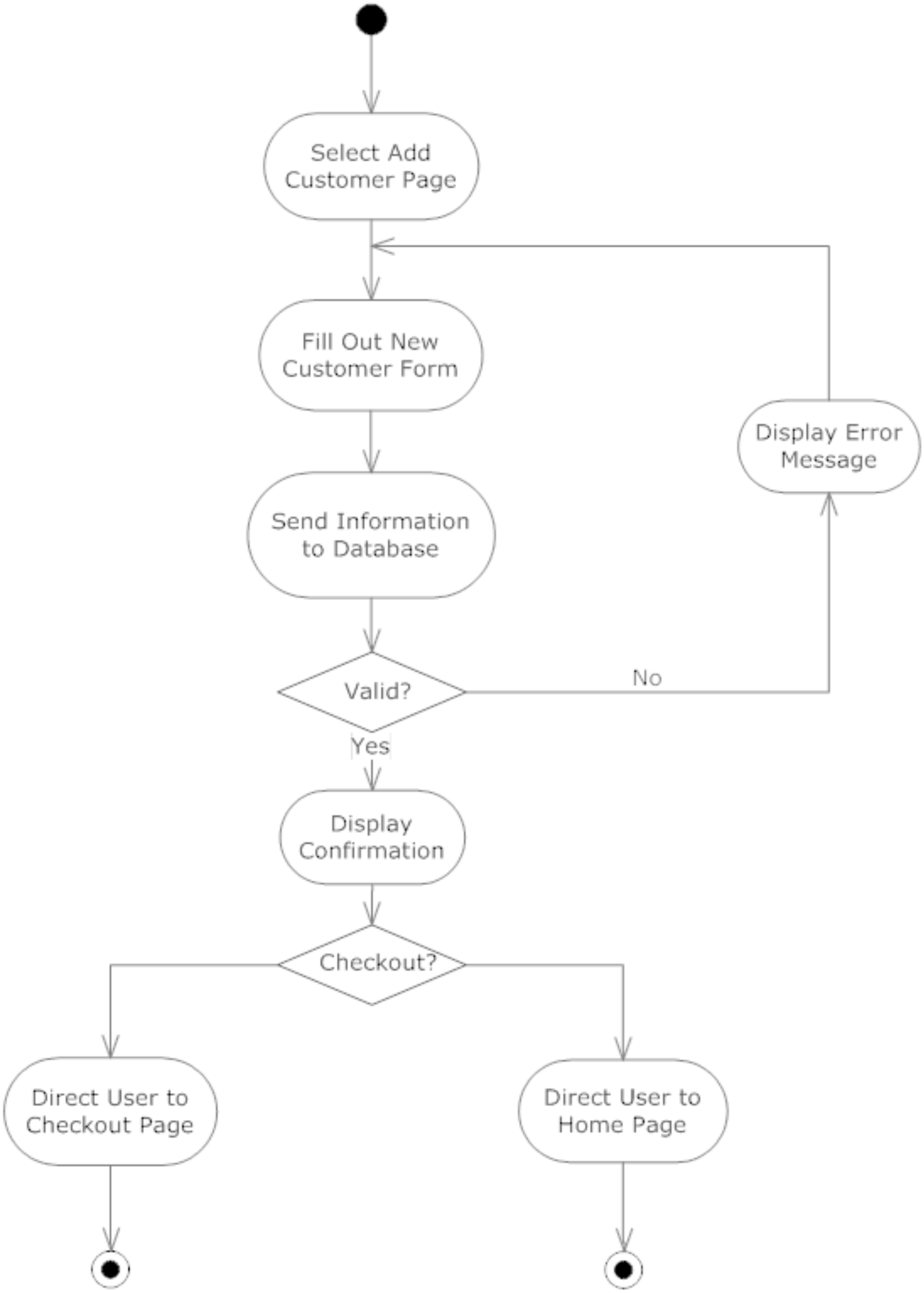
→ Flow - Demonstrates the movement of the activity.

3.3.2. Activity Diagram: Login

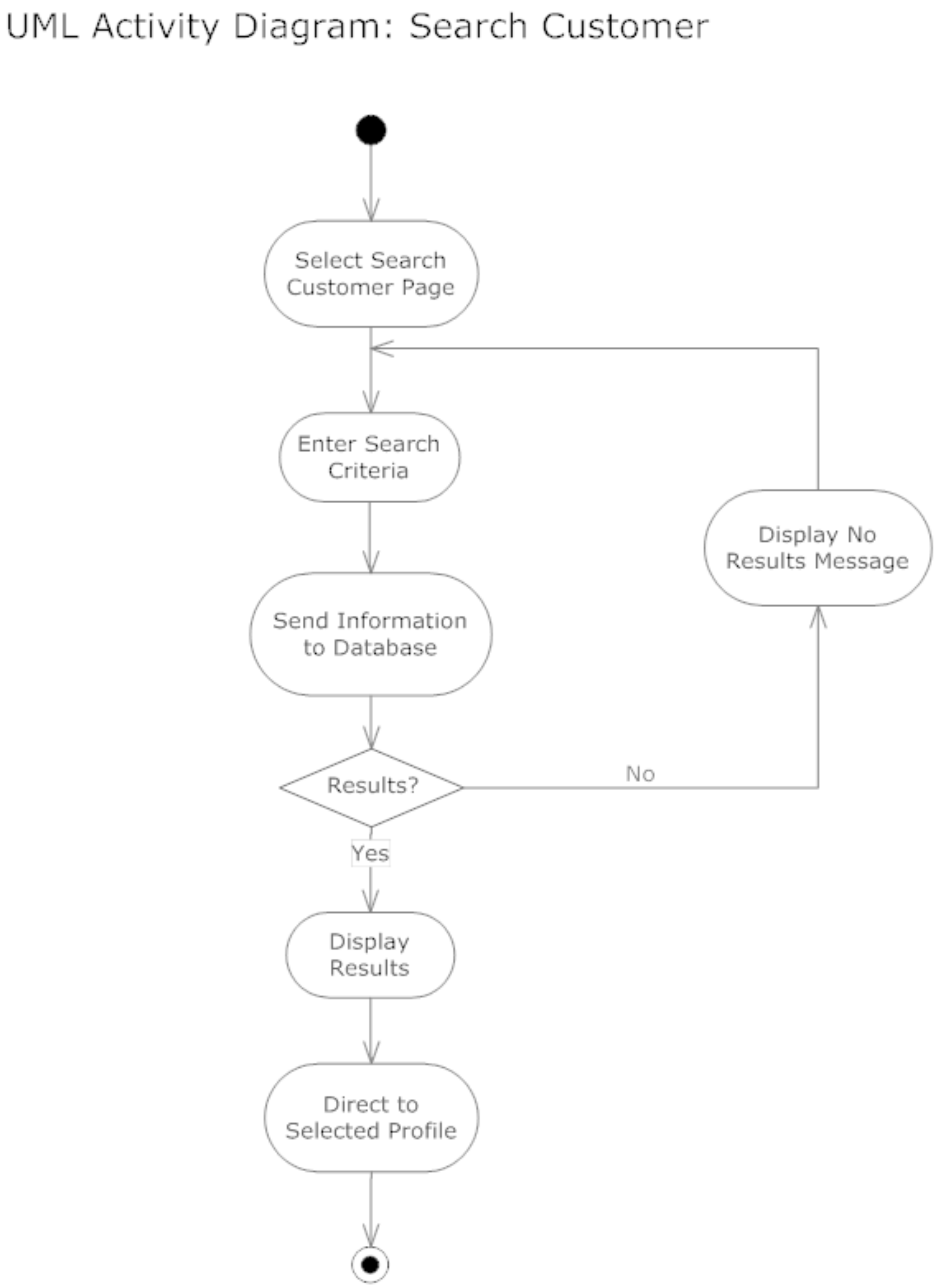

UML Activity Diagram: Login
Enter Login Information
Send Information to Database
Valid?
No
Display Error Message
Yes
Account Type
Direct User to Volunteer Home
Direct User to Director Home

3.3.3. Activity Diagram: Add Customer

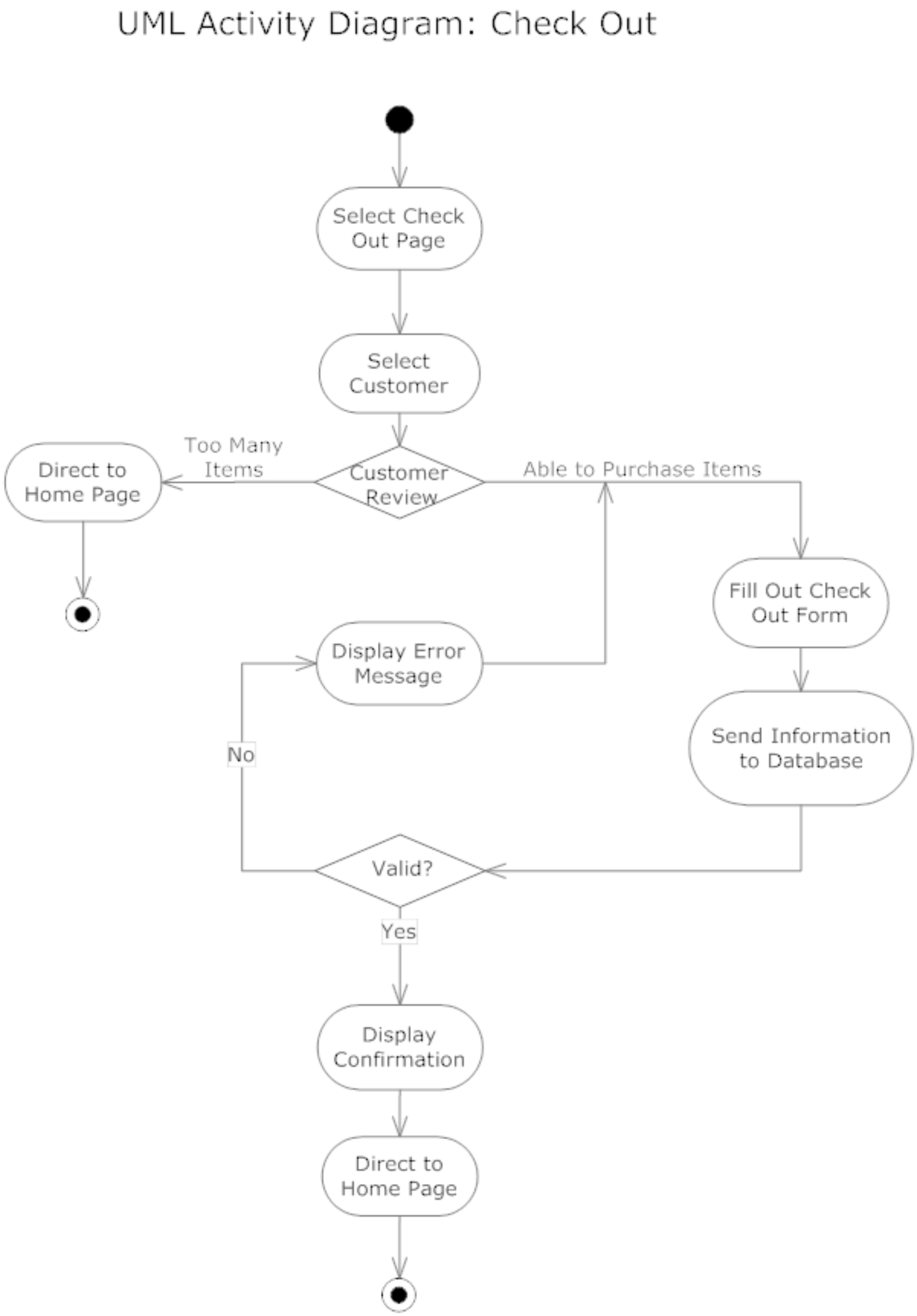
UML Activity Diagram: Add Customer



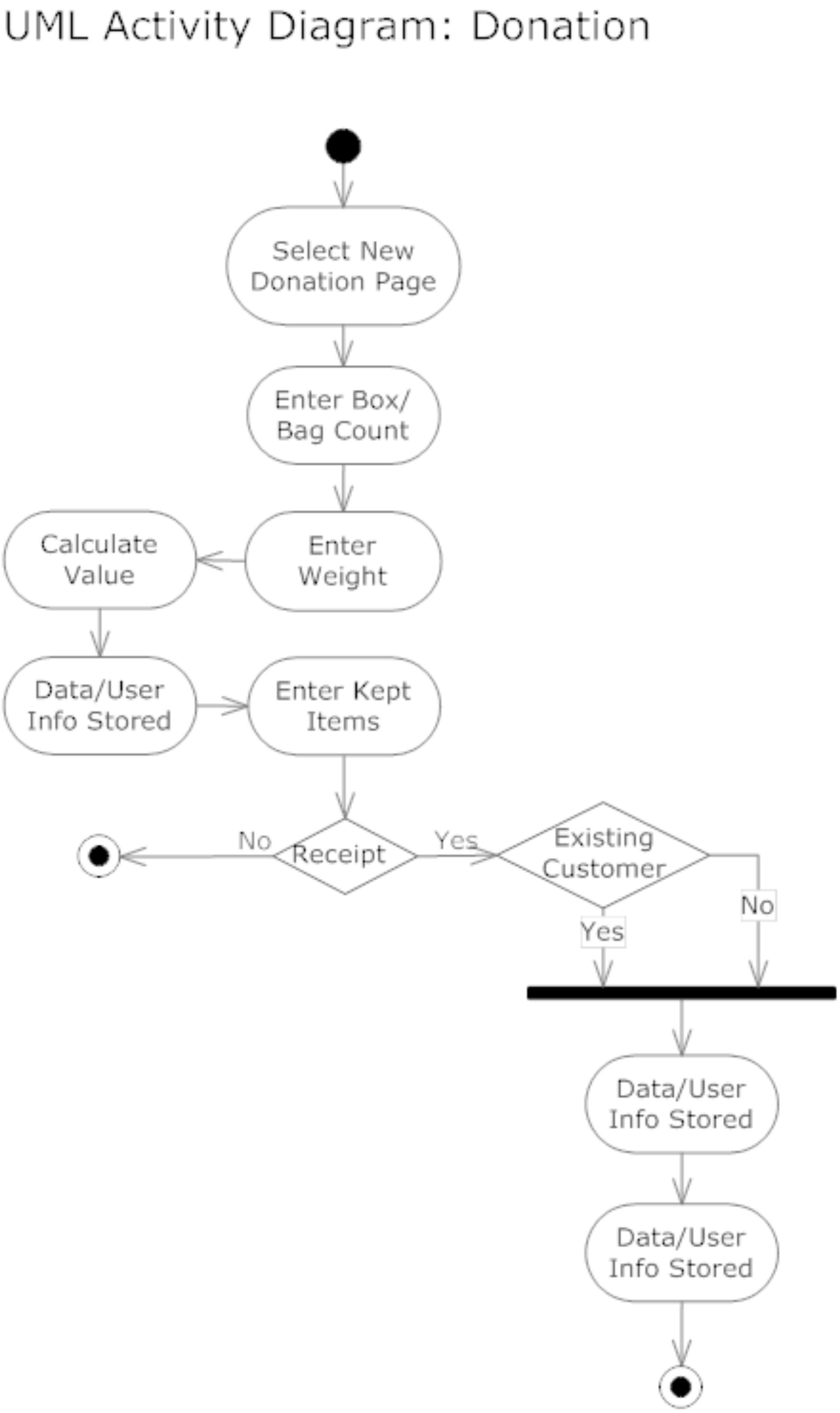
3.3.4. Activity Diagram: Search Customer


UML Activity Diagram: Search Customer
Select Search Customer Page
Enter Search Criteria
Send Information to Database
Results?
No
Display No Results Message
Yes
Display Results
Direct to Selected Profile

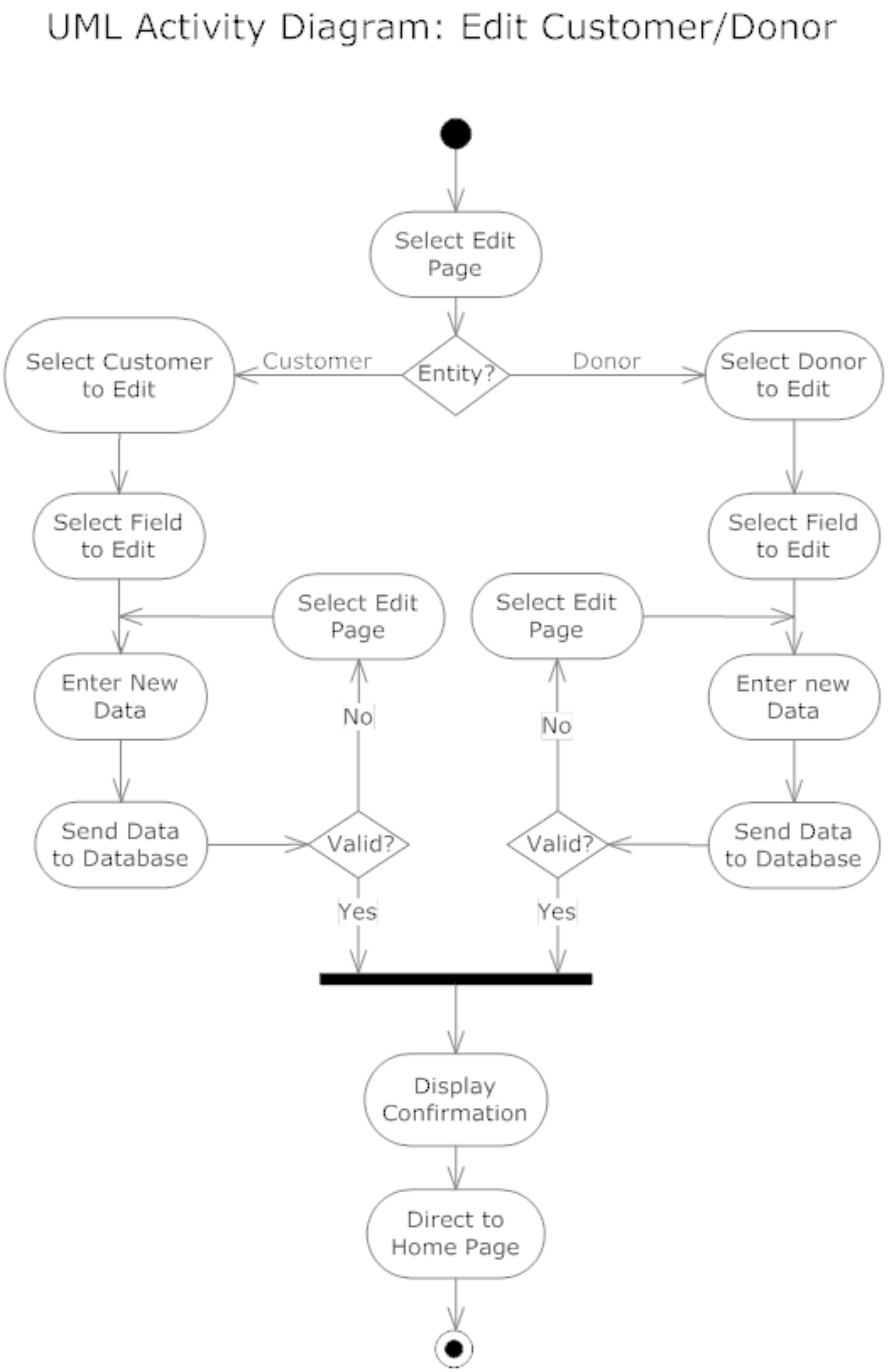
3.3.5. Activity Diagram: Check Out



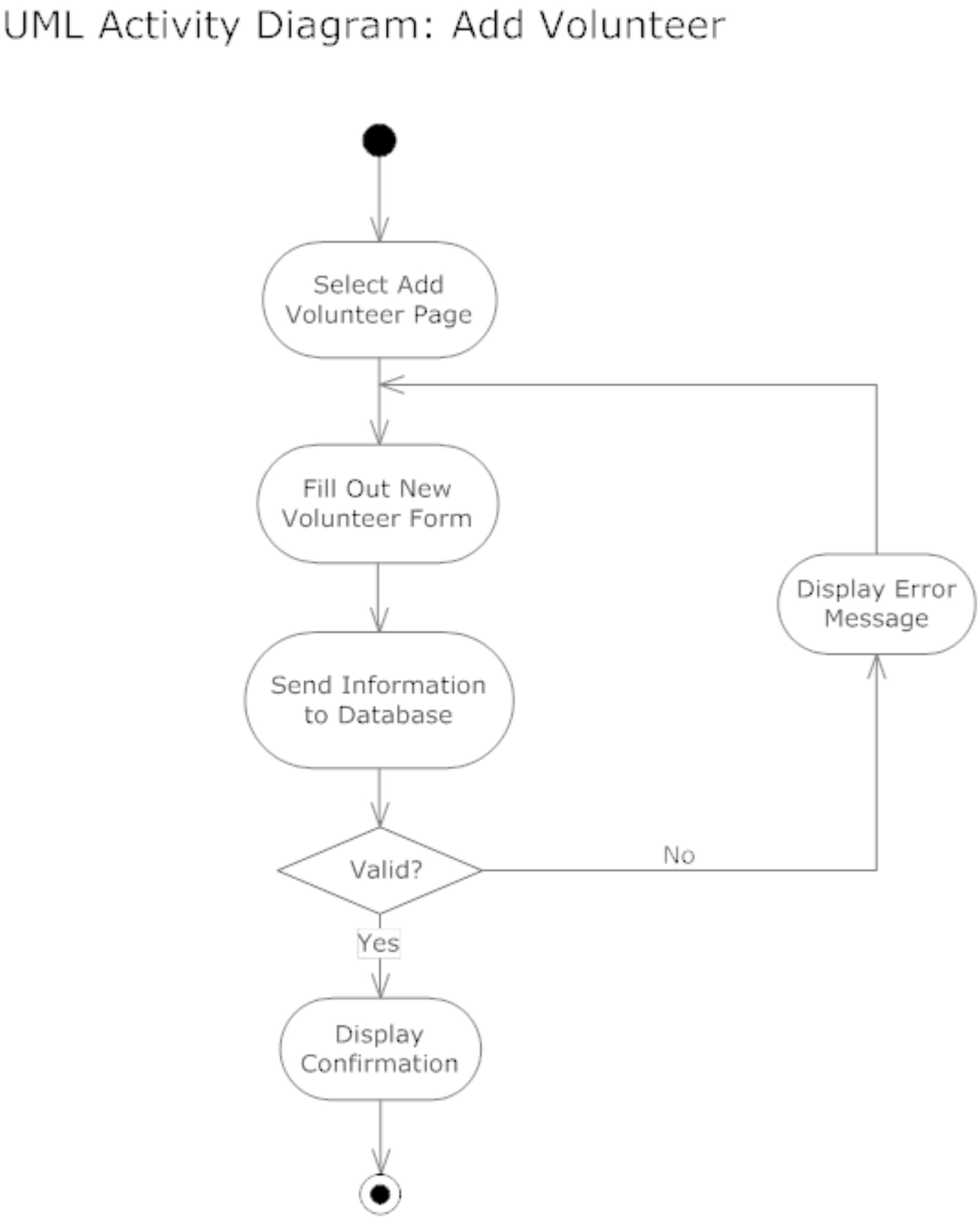
3.3.6. Activity Diagram: Record Donation



3.3.7. Activity Diagram: Edit Customer/Donor



3.3.8. Activity Diagram: Add Volunteer


UML Activity Diagram: Add Volunteer
Select Add Volunteer Page
Fill Out New Volunteer Form
Send Information to Database
Valid?
No
Yes
Display Error Message
Display Confirmation

3.4. Website Maps

3.4.1. Website Map Legend

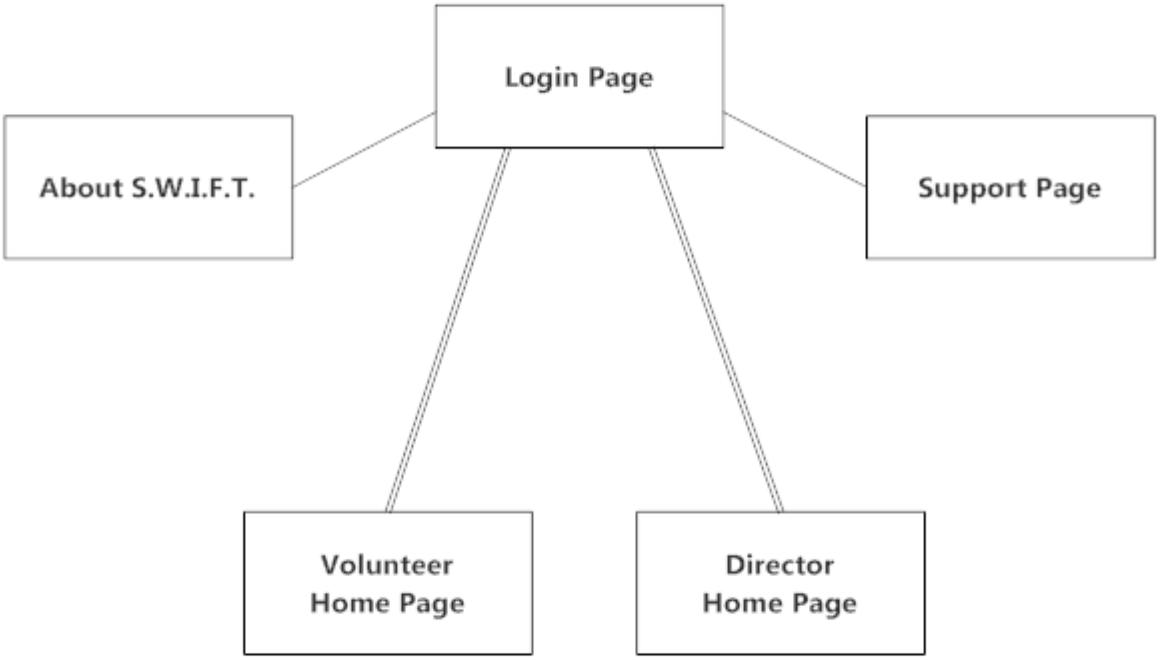
□ Web Page - Represents a web page in the system.

→ Page Redirect - Forced direct to another page.

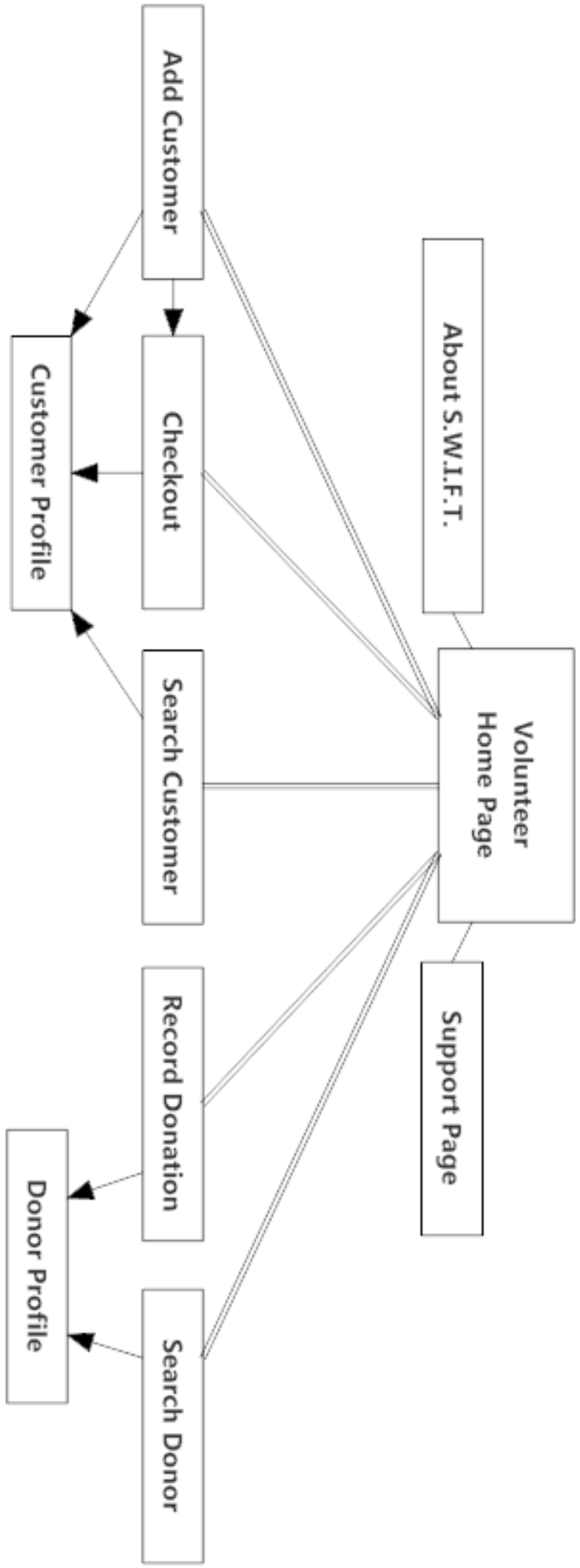
— Link - Represents the ability to access one page from the other.

═ Double Link - Ability to go back and forth between the two connected pages.

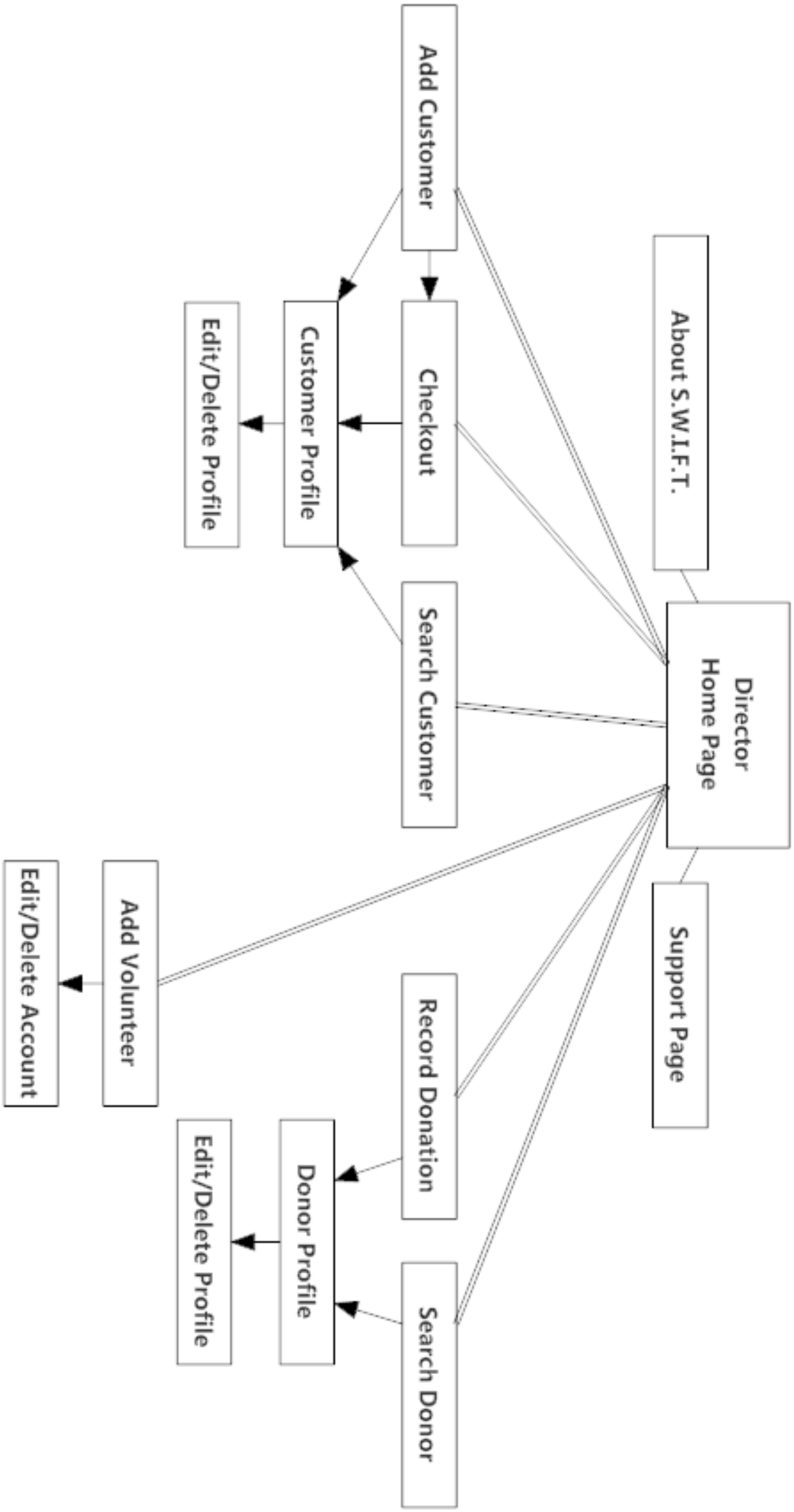
3.4.2. Website Map: Landing Page



3.4.3. Website Map: Volunteer



3.4.4. Website Map: Director



4. Data Flow Diagrams

4.1. Data Flow Legend

○ Process - Functions which are able to receive, modify, and output data.

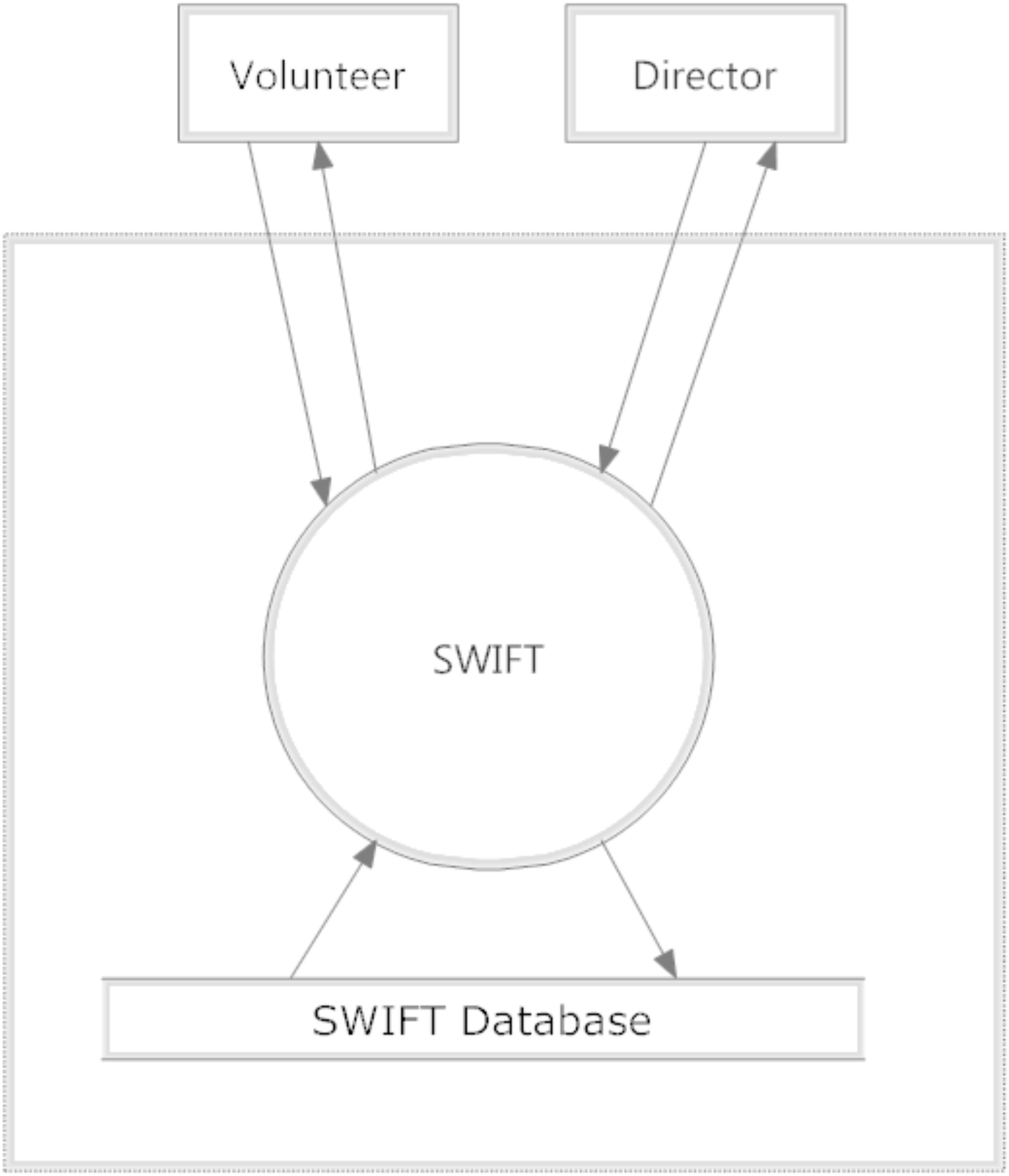
□ Entity - Contributes data to and receives data from the system.

═ Data Store - A place where data is stored permanently or temporarily.

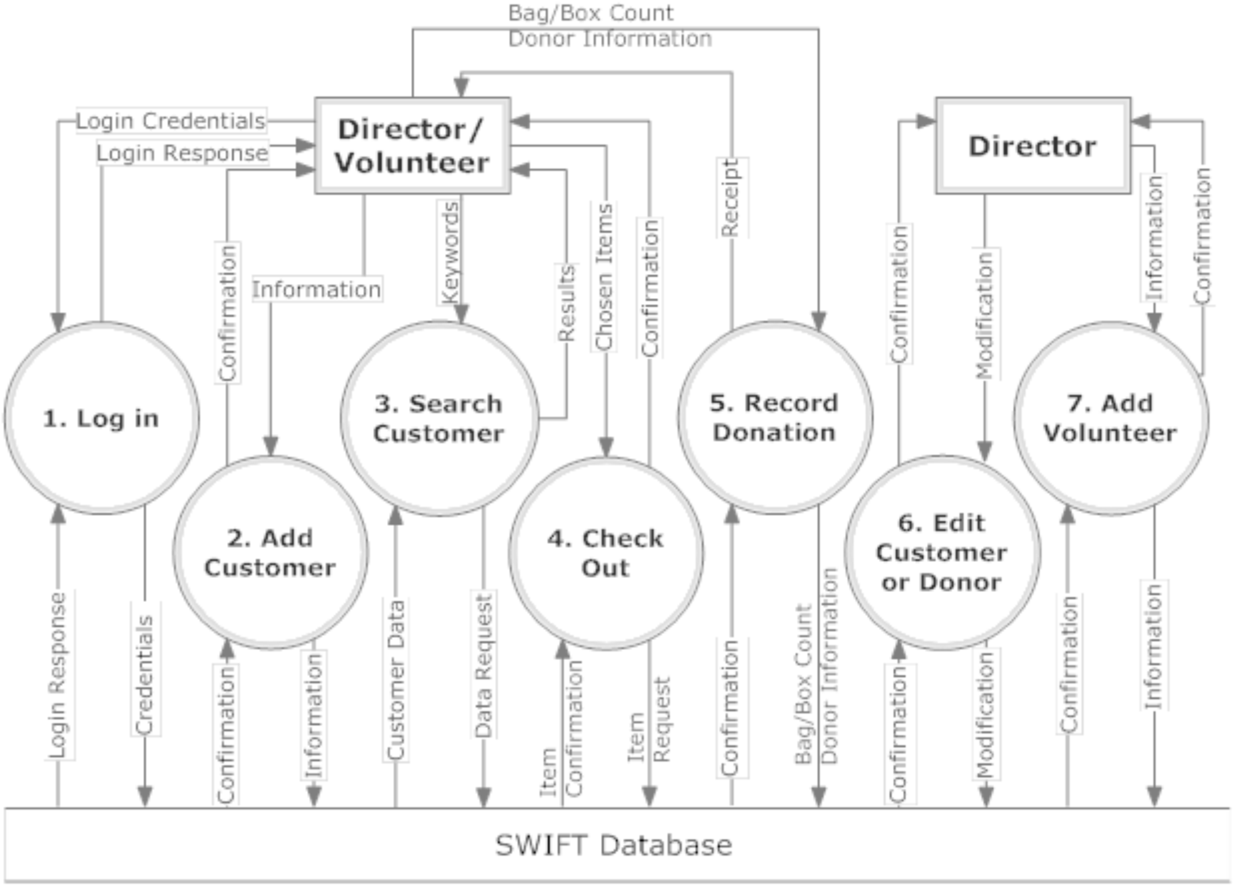
→ Data Flow - Represents the movement and direction of data.

⬚ System Boundary - represents the boundary showing the data stores and the system processes inside the system.

4.2. Context Diagram

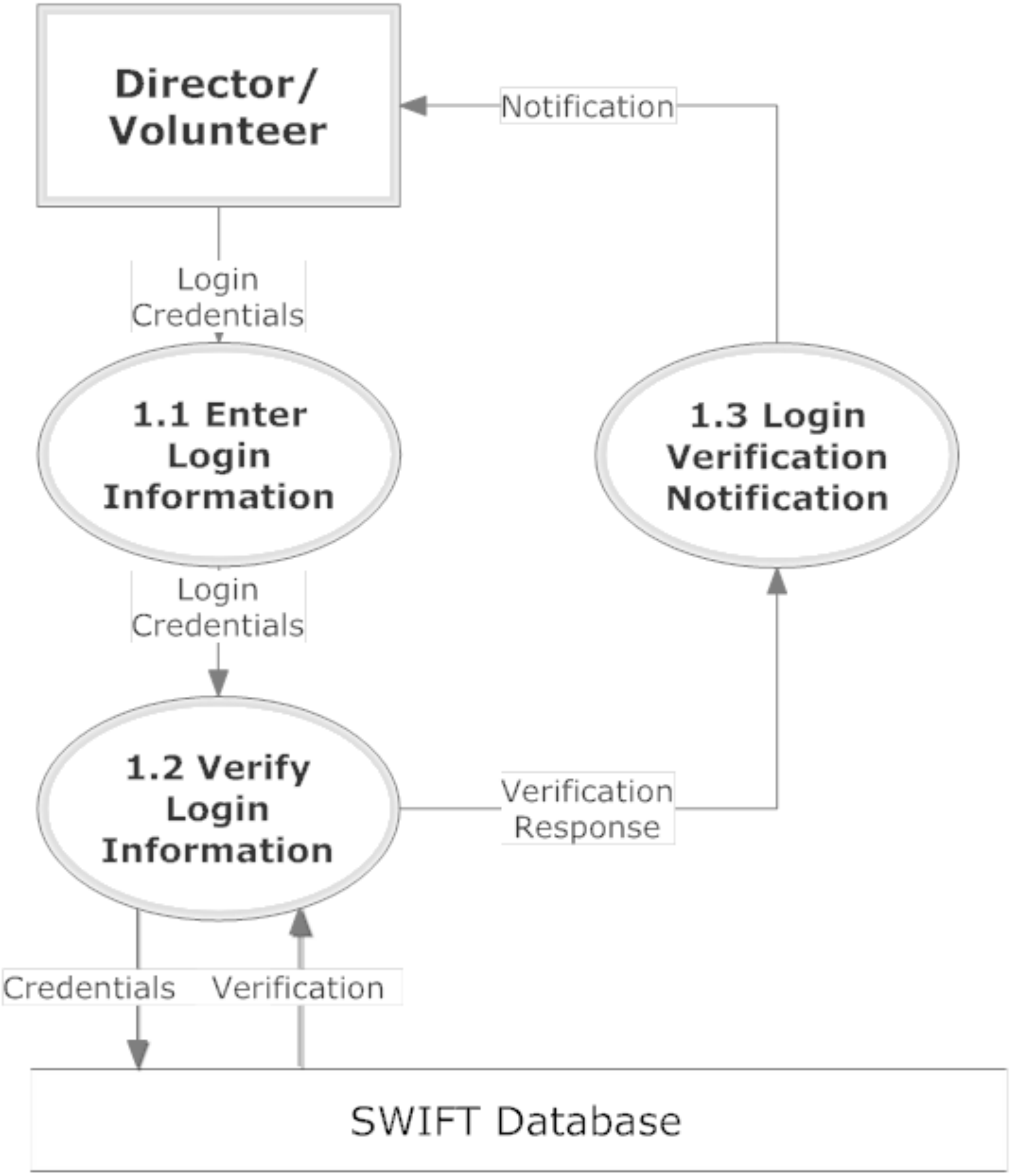


4.3. Level 0 Diagram

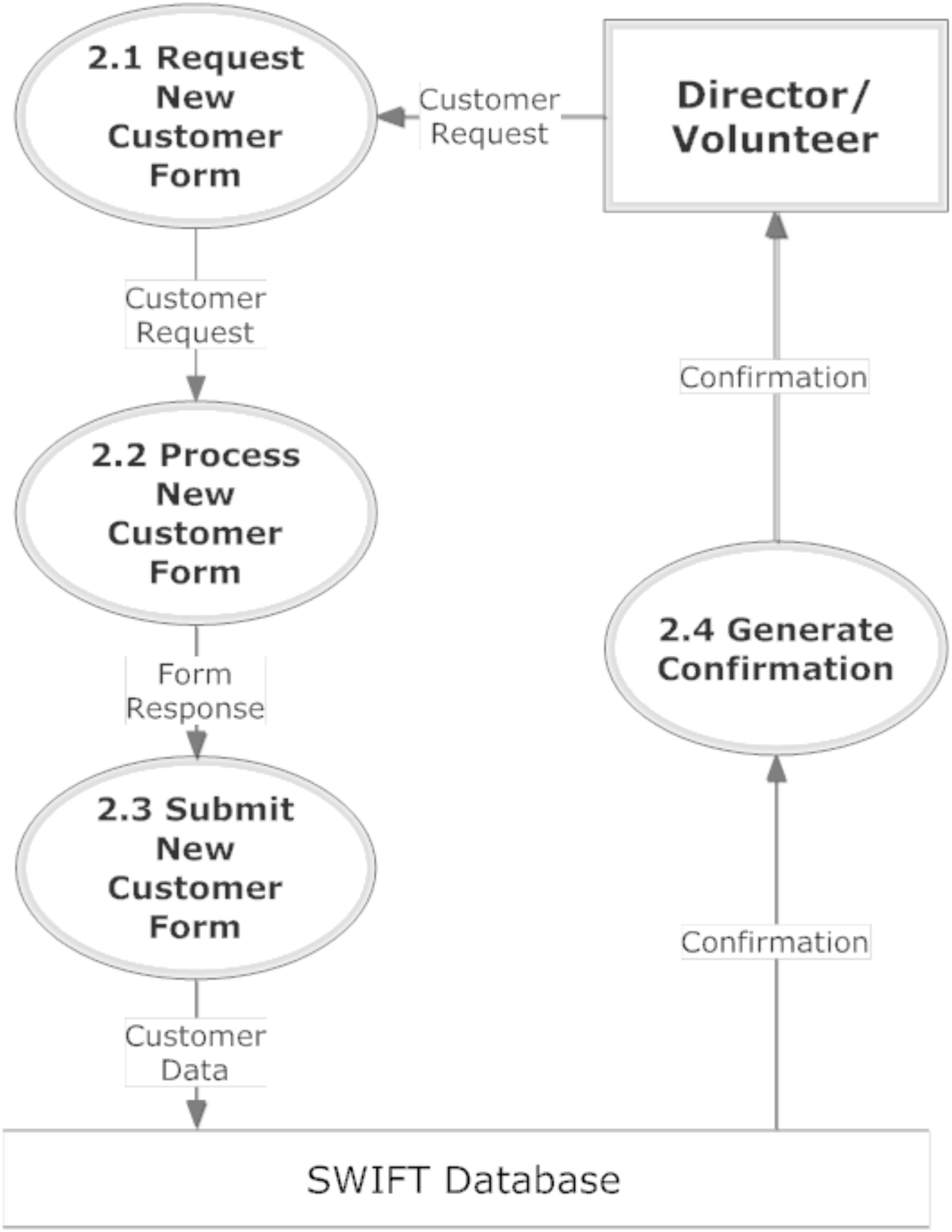


4.4. Level 1 Diagrams

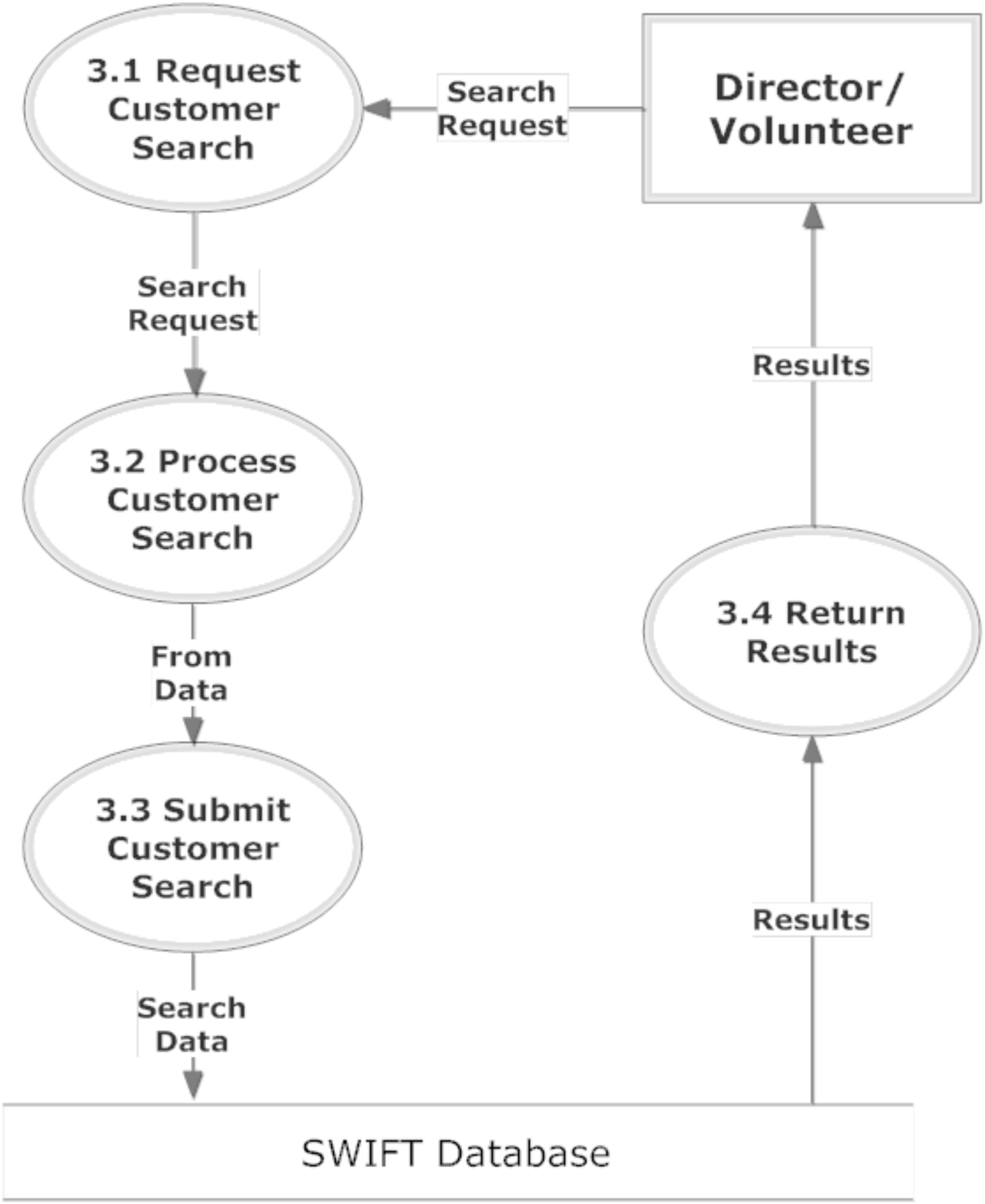
4.4.1. Level 1: Log In



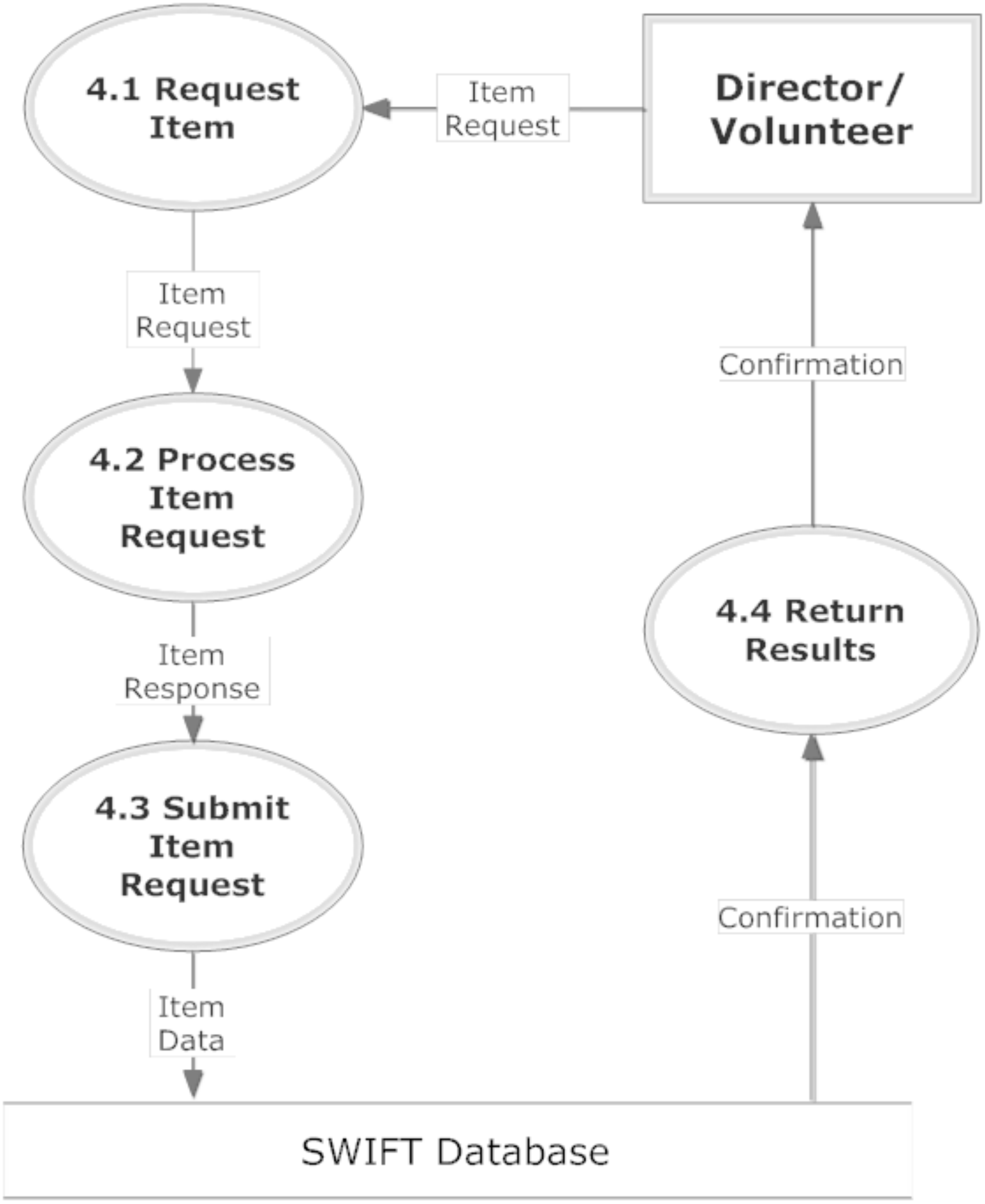
4.4.2. Level 1: Add Customer



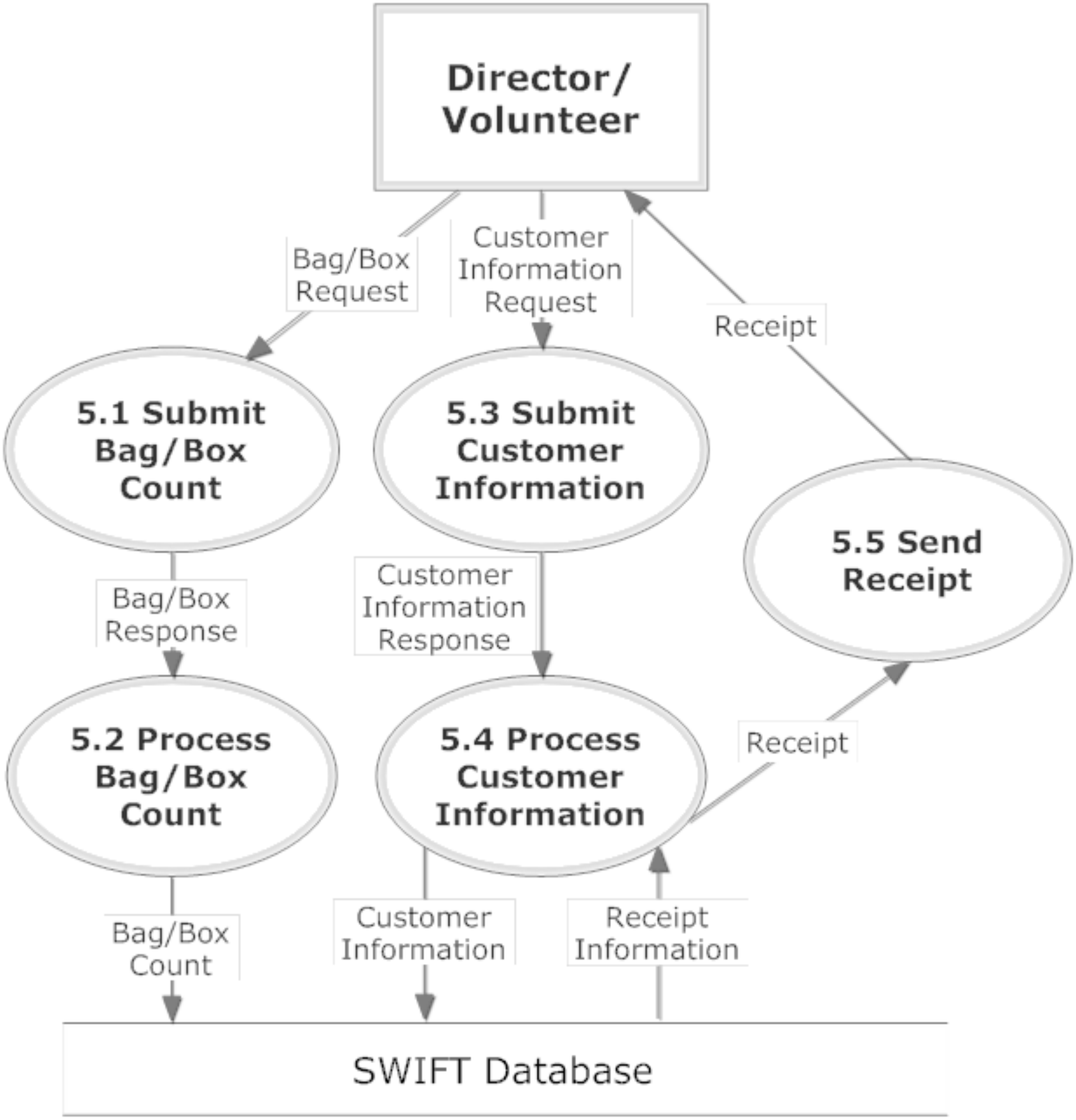
4.4.3. Level 1: Search Customer



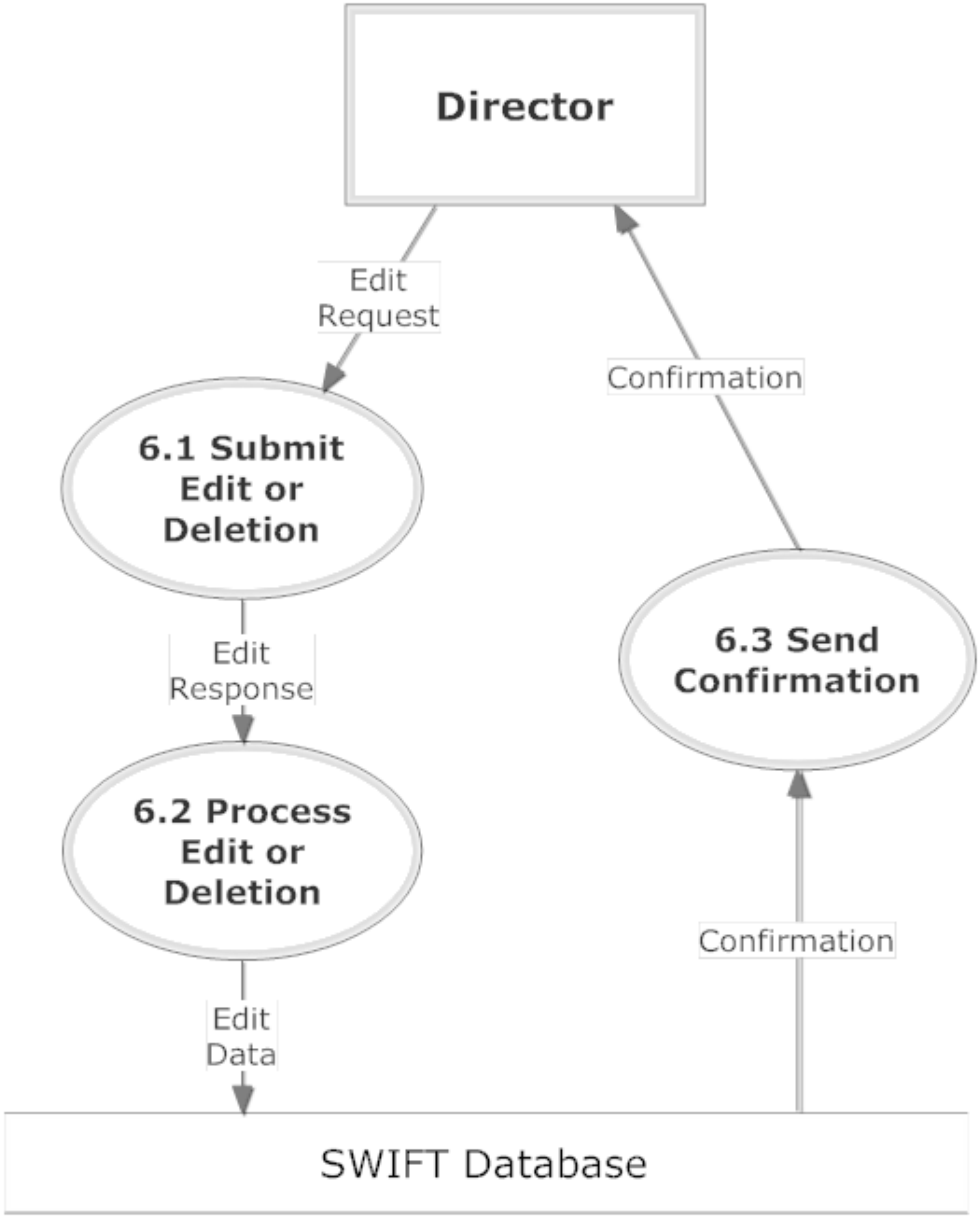
4.4.4. Level 1: Checkout



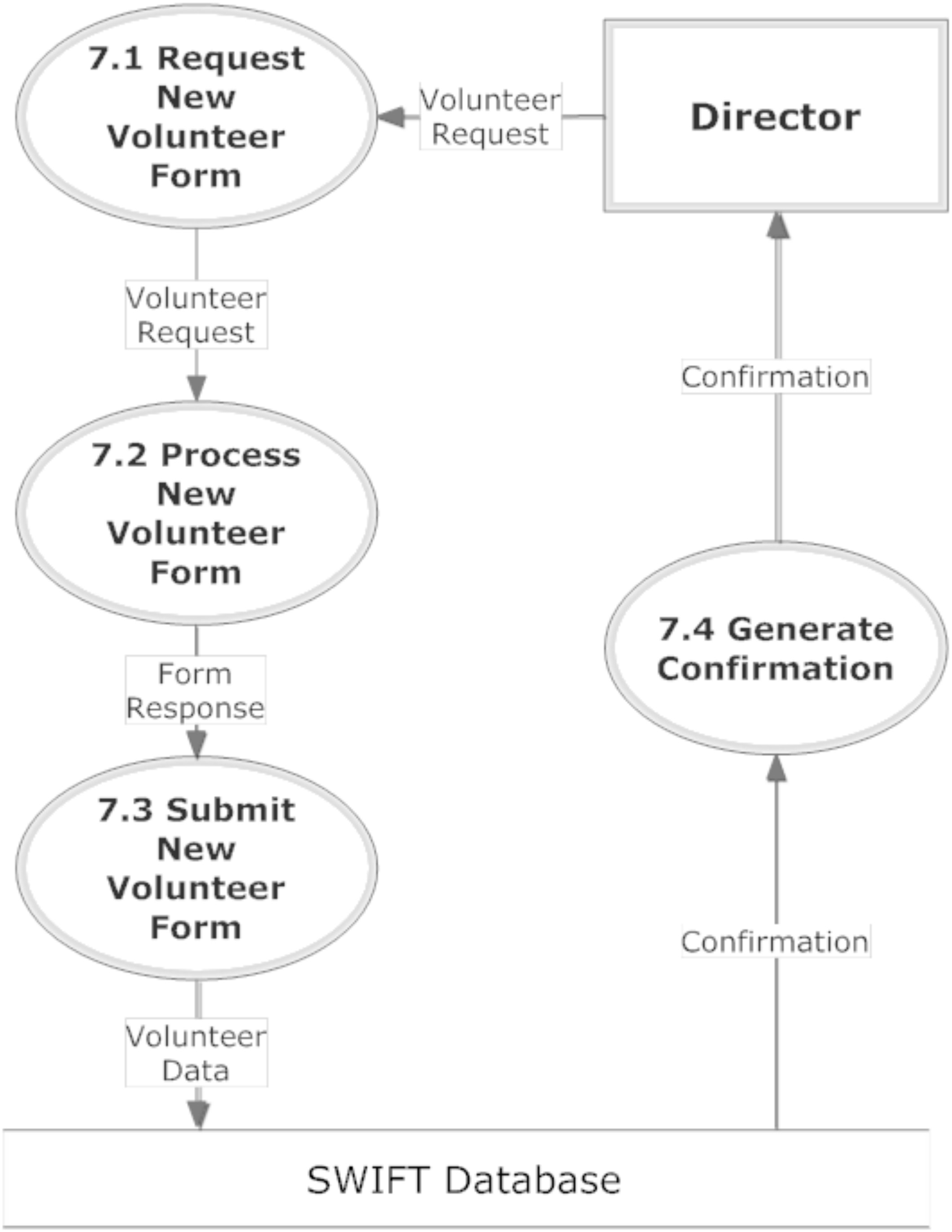
4.4.5. Level 1: Record Donation



4.4.6. Level 1: Edit Customer/Donor



4.4.7. Level 1: Add Volunteer

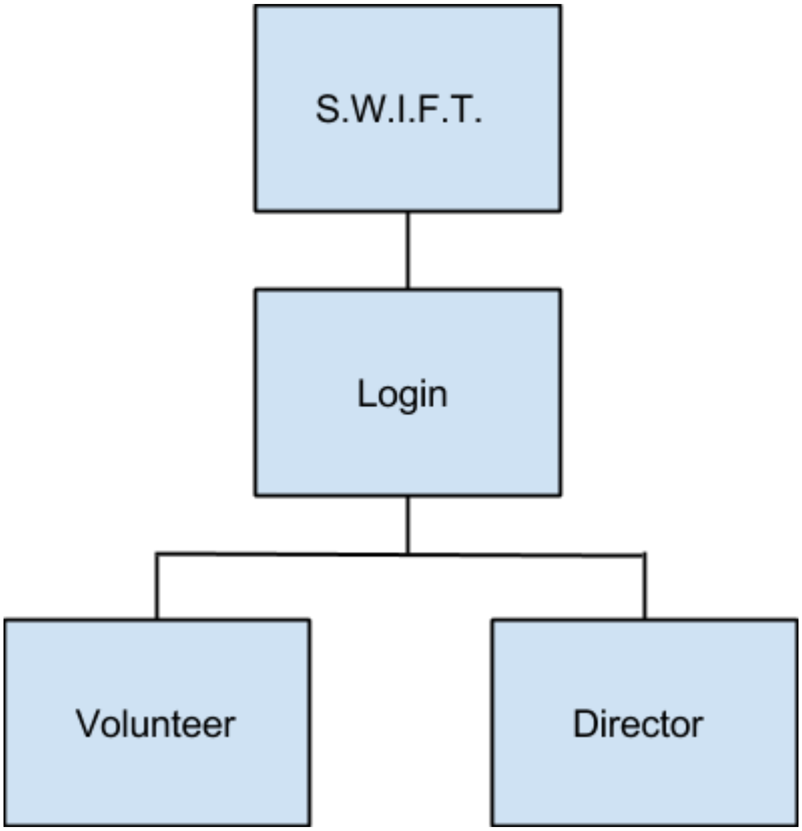

7.1 Request New Volunteer Form
Volunteer Request
Director
Volunteer Request
Confirmation
7.2 Process New Volunteer Form
7.4 Generate Confirmation
Form Response
7.3 Submit New Volunteer Form
Confirmation
Volunteer Data
SWIFT Database

5. Structure Diagrams

Structure Diagrams show the hierarchy of all the elements involved in S.W.I.F.T.

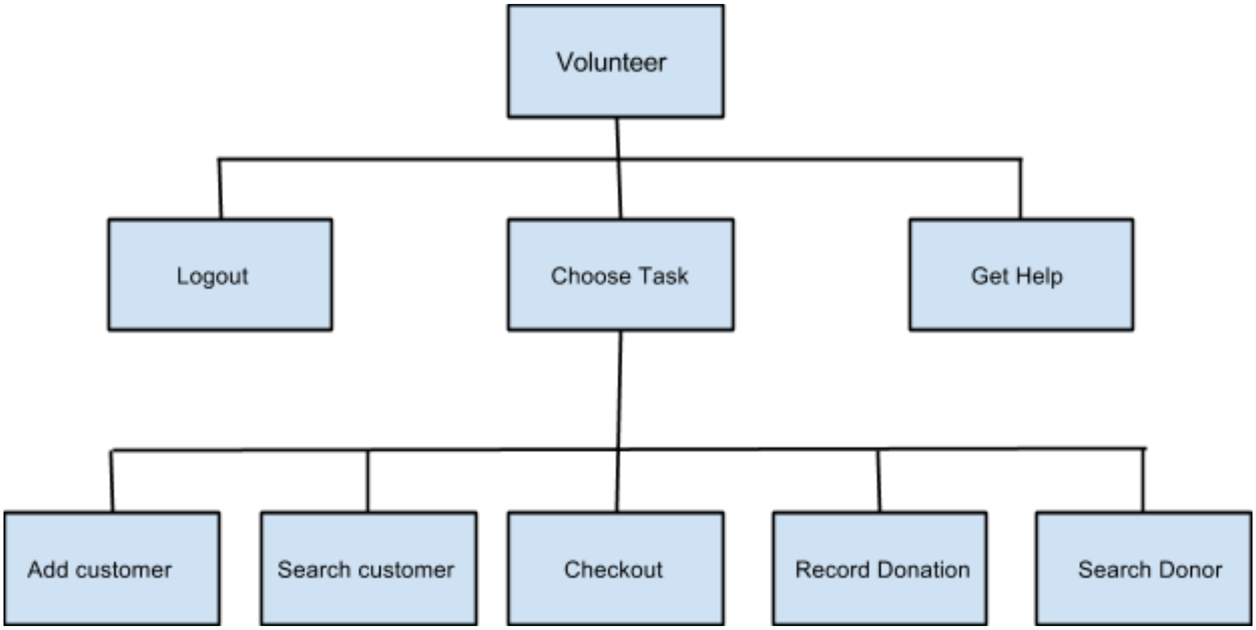
5.1. Login Structure

This diagram shows the login structure of the S.W.I.F.T system that allows access to components of the system depending on your role.



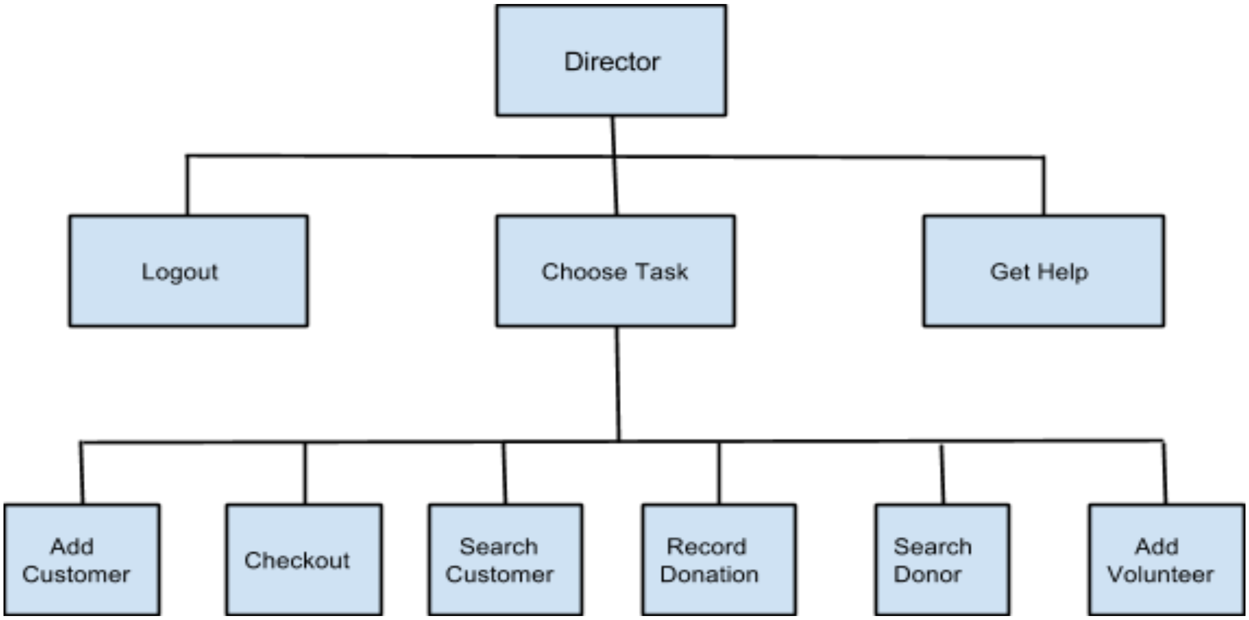
5.2. Volunteer Structure

The Volunteer Structure shows what components of S.W.I.F.T. the volunteer has access to.



5.3. Director Structure

The Director Structure shows what components of S.W.I.F.T. the director has access to.



6. Functional Requirements Inventory

The list below will provide a general outline for the users involved in the system and what they will have access to do. Since the software will be a user friendly web application, it will be able to be used on all major web browsers. The browsers that the software will be compatible with include Google Chrome, Safari, Mozilla Firefox, and Internet Explorer.

6.1. Volunteer

- Will be able to login
    - Logins will be individualized names and passwords
- Will be able to log out
- Will be able to search for a person
- Will be able to checkout items
- Will be able to create donation receipts
- Will be able to insert
    - customer information
    - purchase information
    - donor information

6.2. Director

- Will inherit all functional requirements for the volunteer
- Will be able to edit existing data on the system
- Will be able to delete data no longer wanted on the system
- Will be able to add volunteer accounts
- Will be able to delete volunteer accounts

7. Logical Data Dictionary

This logical data dictionary is used to describe the metadata that we will use in our database for S.W.I.F.T. The information we will keep track of are data name, synonym, what the data is applicable to, data type, data size, description, acceptable input, an example, and notes

Notes	Example	Acceptable Input	Description	Data Size	Data Type	Applicable to	Synonym	Data Name
	AA	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	Agencies the customers are affiliated with	1-30 Characters	Varchar	adding a customer to the system, viewing a customer in the system	customers affiliation with agencies	affiliation
Min:0 Max:99	2	0 through 9	Count of bags/boxes donated by the donor	2 Digits	Integer	adding donation to the inventory	Bag/box count of donation	box_count
	555 Anystreet Drive, Albany, NY, 12206	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	Customer's current address	1-70 Characters	Varchar	Adding Customer to system, viewing customer in system	Customer's current address	cust_address
	Joe	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	Customers first name	1-30 characters	Varchar	Adding Customer to system, searching for customer within system	Customer First Name	cust_firstname
	Smith	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	Customer's last name	1-30 Characters	Varchar	Adding Customer to system, searching for customer within system	Customer Last Name	cust_lastname
	Bob	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	Dependent's first name	1-30 Characters	Varchar	adding customer information to the system, viewing customer in the system	Dependent's first name	dep_firstname

Notes	Example	Acceptable Input	Description	Data Size	Data Type	Applicable to	Synonym	Data Name
	M	M for Male item, F for Female item	Dependent's Gender	1 Character	Char	Adding Customer to system, searching for customer within system	Dependent Gender	dep_gender
	Smith	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	Dependent's last name	1-30 Characters	Varchar	adding customer information to the system, viewing customer in the system	dependent's last name	dep_lastname
	Son	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	The relationship of the dependent to the customer	1-30 Characters	Varchar	adding customer information to the system, viewing customer in the system	relationship of dependent to customer	dep_relation
	Store	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	The director's first name	1-30 Characters	Varchar	name stamp on entered information	director's first name	director_firstname
	Director	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	The director's last name	1-30 Characters	Varchar	name stamp on entered information	director's last name	director_lastname
	donationS2	Password must be 6-12 characters long, must contain a number, and capital letter	The director's password so they can log into S.W.I.F.T.	6-12 Characters	Varchar	login and logout	director's password	director_password
	director		The director's user name so they can log into S.W.I.F.T.	1-30 Characters	Varchar	login and logout	director's username	director_username

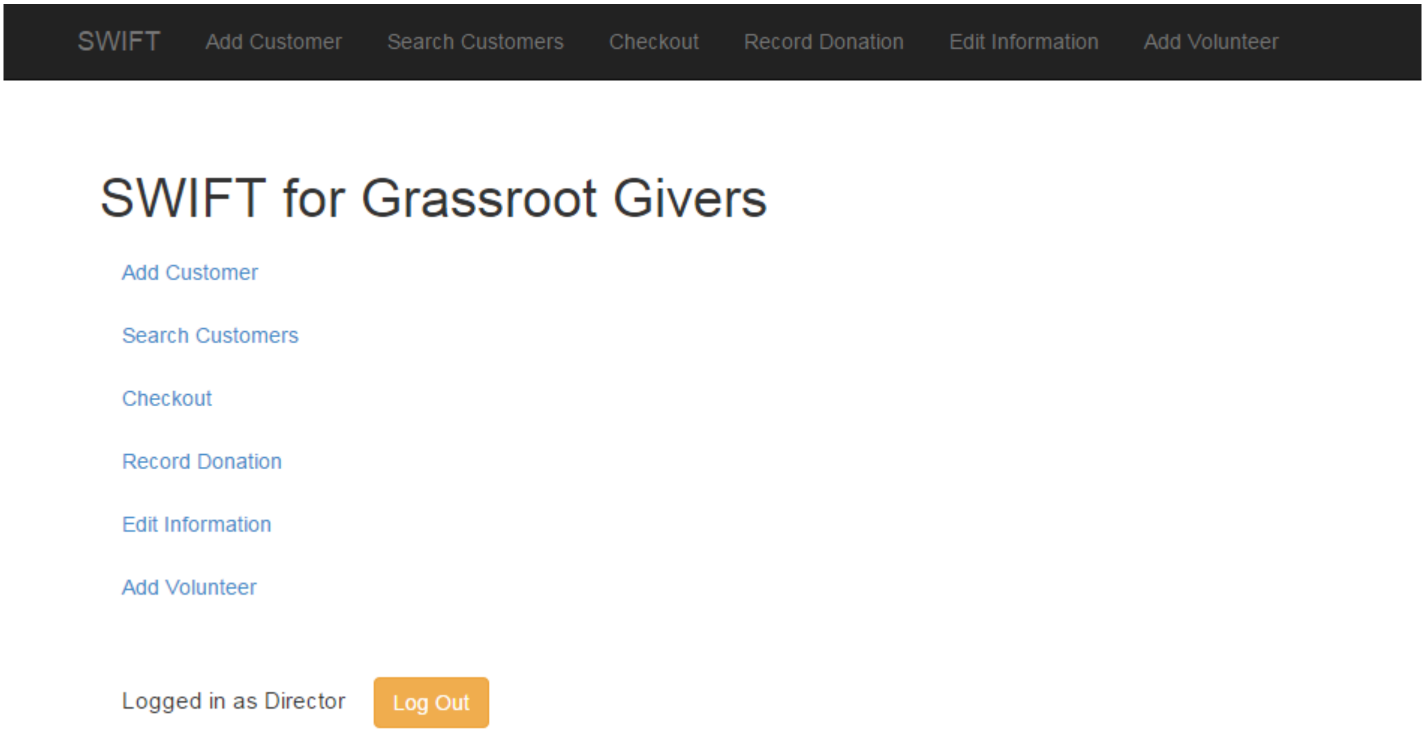
Notes	Example	Acceptable Input	Description	Data Size	Data Type	Applicable to	Synonym	Data Name
Value in Dollars and cents, must be proper decimal format	123.45	0 through 9 and '.'	The monetary value of donation	5 Digits	Float	adding donation to the inventory	Monetary value of donation	donation_value
Weight in pounds	12.55	0 through 9 and '.'	The amount the bags/boxes donated weigh	5 Digits	Float	adding donation to the inventory	Weight of bag/boxes from donation	donation_weight
	555 Anystreet Drive, Albany, NY, 12206	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	Donor's address	1-70 Characters	Varchar	adding donor to the system, viewing donor in the system, creating receipts	Donor's address	donor_address
	Fred	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	Donor's first name	1-30 Characters	Varchar	adding donor to the system, viewing donor in the system, creating receipts	Donor's first name	donor_firstname
	Johnson	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	Donor's last name	1-30 Characters	Varchar	adding donor to the system, viewing donor in the system, creating receipts	Donor's last name	donor_lastname
Standard US 10 digit phone number	518-555-5555	0 through 9 and '-'	Donor's phone number	1-15 Characters	Varchar	adding donor to the system, viewing donor in the system, creating receipts	Donor's phone number	donor_phone

Notes	Example	Acceptable Input	Description	Data Size	Data Type	Applicable to	Synonym	Data Name
Color, Gender, Article of clothing	black men's suit jacket	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	Describes the item	1-60 Characters	Varchar	adding donation to the inventory	desription of item	item_desc
	Women's pants, men's shirt	M for Male item, F for Female item	Keeps track of the gender the item is for	1 Character	Char	Adding donation to the inventory	Keeps track of the gender the item is for	item_gender
Min:0 Max:99	4	0 through 9	Count of the number of items	2 digits	Integer	adding donation to the inventory	number of items	item_quant
	14:24 11/09/2014	0 through 9, '/', and ':'	The time and date the information was input	16 Characters	Varchar	checking out the customer, creating customers and donors	timestamp for entered information	timestamp
	Ed	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	Volunteer's first name	1-30 Characters	Varchar	creating volunteer accounts, name stamp on entered information	volunteer's first name	vol_firstname
	Jones	ASCII char 32 (space), -, A-Z, a-z, ', `, ASCII char 128 to ASCII char 165	Volunteer's last name	1-30 Characters	Varchar	creating volunteer accounts, name stamp on entered information	volunteer's last name	vol_lastname
	Donations8	Password must be 6-12 characters long, must contain a number, and capital letter	The password for a volunteer so they can log into S.W.I.F.T.	6-12 Characters	Varchar	creating volunteer accounts, login and logout	volunteer's password	vol_password
Will be first initial and last name	ejones	Volunteer's first initial, followed by their last name	The user name for a volunteer so they can log into S.W.I.F.T.	1-30 Characters	Varchar	creating volunteer accounts, login and logout	volunteer's username	vol_username

8. Prototypes

This section includes some preliminary screen design ideas for a couple of the S.W.I.F.T. forms.

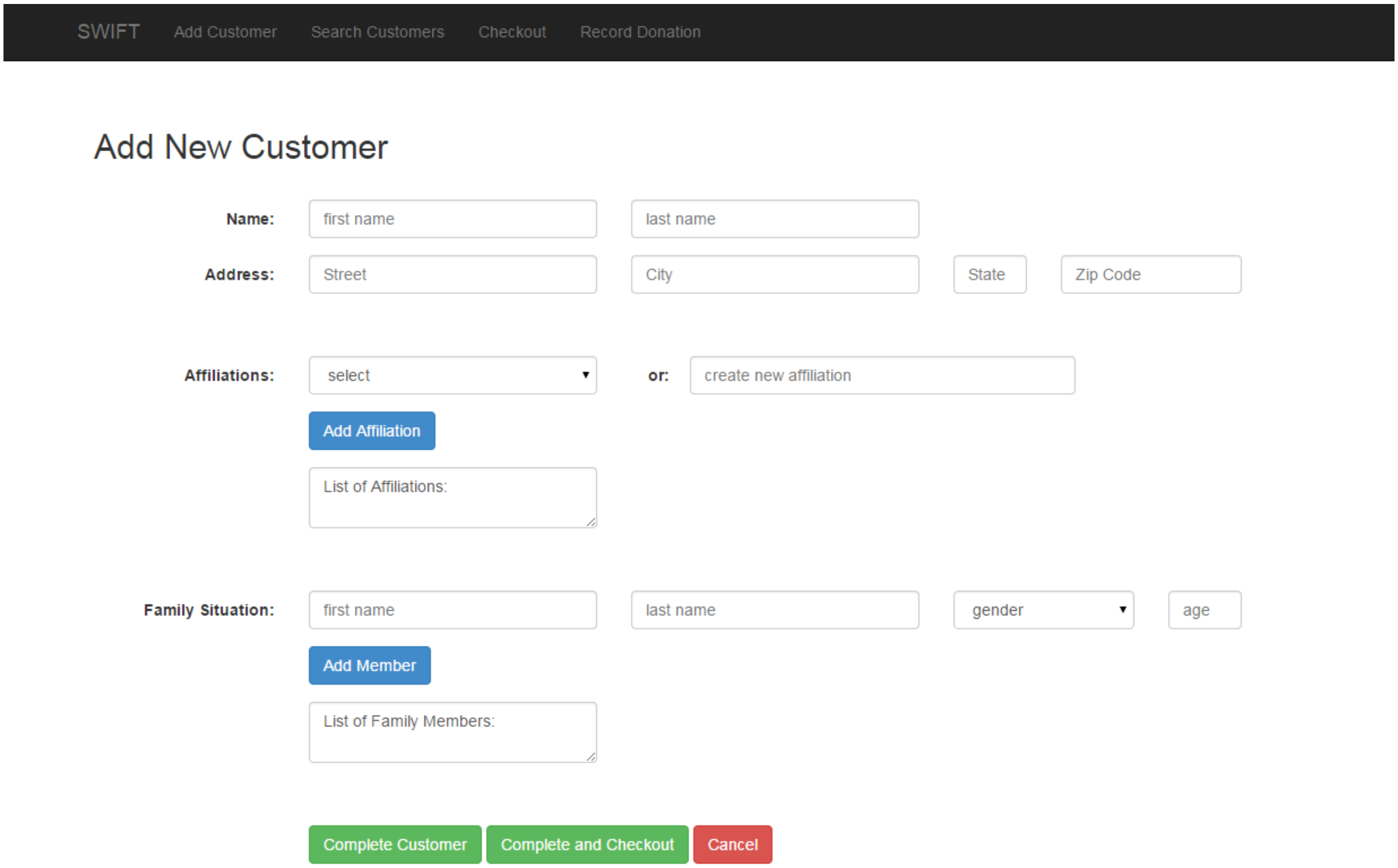
8.1. Prototype: Home



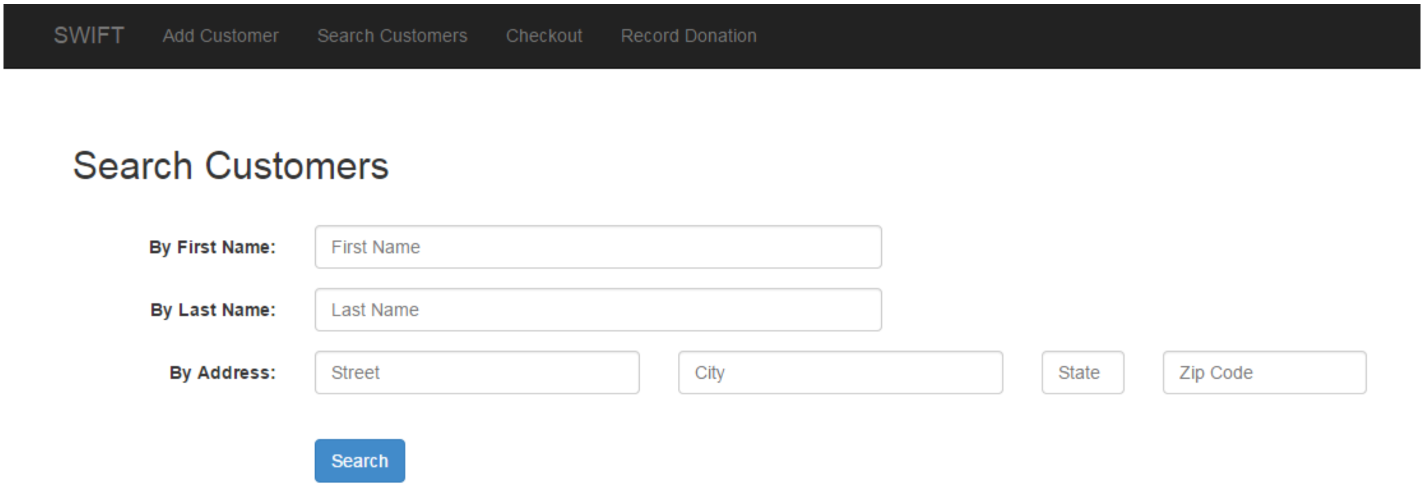
SWIFT for Grassroot Givers

Logged in as Director

8.2. Prototype: Add Customer



8.3. Prototype: Search Customer



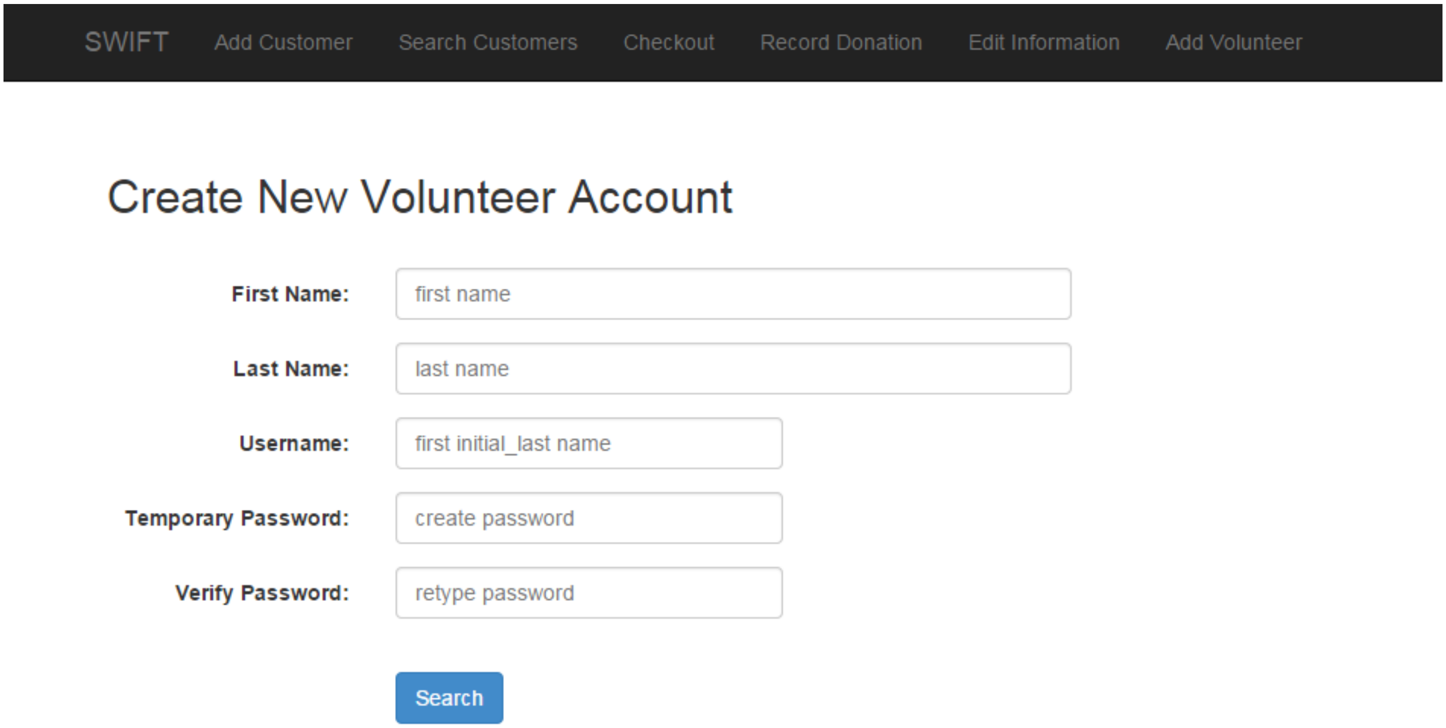
Search Customers

By First Name: First Name

By Last Name: Last Name

By Address: Street City State Zip Code

8.4. Prototype: Add Volunteer



Create New Volunteer Account

First Name: first name

Last Name: last name

Username: first initial_last name

Temporary Password: create password

Verify Password: retype password

9. Testing Plan

9.1. Overview and Strategy

SMARK Solutions is working to create an application that goes above and beyond the expectations of the clients, Dr. Fryling, and Dr. Lim. S.W.I.F.T. is a web-based application, therefore it will be tested to ensure compatibility with all major web browsers. These browsers include Apple Safari, Mozilla Firefox, Google Chrome, and Internet Explorer. Along with this capability, we will test our application by using a number of test cases designed to ensure our application correctly accepts information, and then does not accept the information that is not designed to enter the database. In this section, you will find our test cases, along with our acceptance test. Actual results are not included in this document, as they will be entered in our Detailed Design document, which will be written during the spring semester.

The best and most efficient solutions possible will be created to help solve our client's problems. Together we can be intuitive, be efficient, be SMARK.

9.2. Acceptance Test

Each of the following major functional processes will have test cases created designed to test all of the individual aspects our S.W.I.F.T. to make sure that S.W.I.F.T. works the way it is designed to. Based on the results of these test cases, SMARK Solutions will come together to decide whether or not the program is acceptable to deploy for client use.

9.3. Unit Tests

The following unit tests are tests that will be run to ensure that our application is working properly. Each test will test an individual aspect of one of the major processes to show what inputs will work and and what ones will not/should not work. Following these individual unit tests, a final test will be run to ensure each of the processes will integrate together correctly, allowing our application to do its designed job.

9.3.1. Test Cases

Each of the Unit Tests is made up of a series of test cases. In each case, there are specific guidelines in how to properly test the functionality of the process being tested. Each case also contains information about how the application should respond when each input is entered both before and after the input is processed. If the system responds how it is supposed to according to the case, then the system will be considered to be functional.

9.3.2. Unit Test: Login

Test Date	Tested By	Comments	Observed result	Expected result	State Before Test	Steps to be Executed	Action to perform test (input)	Description	Test Number	Pass/Fail Status
				failed login	login form	click login with null fields	pass null username and passwords	Null username Null password	1.001	
				failed login	login form	click login with a username but not a password	pass a username and null a password	Not Null username Null password	1.002	
				failed login	login form	click login with a username but not a password	pass a null username and a password	Null username Not Null Password	1.003	
				failed login	login form	click login with random username and password	pass a username and password that isnt a user	Not Null username Not Null password	1.004	
				failed login	login form	click login with a random username but valid password	pass invalid username with a correct password	Incorrect username	1.005	
Test	**Tested**	**Comments**	**Observed**	**Expected**	**State**	**Steps to**	**Action to**	**Description**	**Test**	**Pass/Fail**

Date	By		result	result	Before Test	be Executed	perform test (input)		Number	Status
				failed login	login form	click login with an existing username but incorrect password	pass valid username with a incorrect password	Incorrect password	1.006	
				accepted login, redirect to director home page	login form	login with director credentials	pass director username and correct password	Correct director login	1.007	
				accepted login, redirect to volunteer home page	login form	login with volunteer credentials	pass volunteer username and correct password	Correct volunteer login	1.008	

9.3.3. Unit Test: Add Customer

Test Date	Tested By	Comments	Observed result	Expected result	State Before Test	Steps to be Executed	Action to perform test (input)	Description	Test Number	Pass/Fail Status
				First Name Added to Database	Create Customer Form	Enter First Name in form	Add Valid First Name	ValidFirstName	1.001	
				Last Name Added to Database	Create Customer Form	Enter Last Name in Form	Add Valid Last Name	ValidLastName	1.002	
				Address Added to Database	Create Customer Form	Enter Address consisting of the customer's full street address, must be under 70 characters	Add Valid Address Information	ValidAddress	1.003	
				Dependent's First Name Added to Database	Create Customer Form	Enter Dependent First Name in Form	Add Dependent First Name	ValidDepFirstName	1.004	
				Dependent's Last Name Added to Database	Create Customer Form	Enter Dependent Last name in Form	Add Dependent Last Name	ValidDepLastName	1.005	
				Dependent's Relation Added to Database	Create Customer Form	Enter Dependent's Reltation to customer in Form	Add dependent Relation to customer	ValidDepRelation	1.006	
				Dependent Gender Added to Database	Create Customer Form	Enter Dependent Gender into Form	Add Valid Dependent Gender, only adds if dependents exist	ValidDepGender	1.007	

Test Date	Tested By	Comments	Observed result	Expected result	State Before Test	Steps to be Executed	Action to perform test (input)	Description	Test Number	Pass/Fail Status
				Error	Create Customer Form	Pass Null Value In	Add Invalid First name	NoFirstName	1.008	
				Error	Create Customer Form	Pass Null Value In	Add Invalid Last name	NoLastName	1.009	
				Error	Create Customer Form	Pass Null Value In	Add Invalid Address	NoAddress	1.010	

9.3.4. Unit Test: Search Customer

Test Date	Tested By	Comments	Observed result	Expected result	State Before Test	Steps to be Executed	Action to perform test (input)	Description	Test Number	Pass/Fail Status
				Return Customer's information	Search Page	Enter customer's name into search input	Enter a customer first name or last name that meets the criteria stated in the add customer test	ValidCustName	1.001	
				Error Returned	Search Page	Enter invalid customer name into search input	Enter a customer first name or last name that does NOT the criteria stated in the add customer test	InvalidCustName	1.002	
				Return Customer names that include this letter	Search Page	Enter Letter in search	Enter a single alphabetical letter in search bar	SingleLetter	1.003	

9.3.5. Unit Test: Checkout

Test Date	Tested By	Comments	Observed result	Expected result	State Before Test	Steps to be Executed	Action to perform test (input)	Description	Test Number	Pass/Fail Status
				Add Customer Name to Checkout Form, Customer's address will be added from databse to Form, Recent History will show items recently bought	Checkout Form	Enter customer's name into form	Customer is in the database and must abide by the restrictions stated in the Data Dictionary	ValidCustName	1.001	
				Item Added to Checkout Form	Checkout Form	Enter item into input on checkout form	Add an item that is currently in the database	ValidItem	1.002	
				Error	Checkout Form	Enter item into input on checkout form	Add an item that is not currently in the database	InvalidItem	1.003	
				Error	Checkout Form	Enter item into input on checkout form	Add Item when the history includes more than 8 items in the last 30 days	ItemLimitReached	1.004	

Test Date	Tested By	Comments	Observed result	Expected result	State Before Test	Steps to be Executed	Action to perform test (input)	Description	Test Number	Pass/Fail Status
				Customer's history of items bought is shown	Checkout Form	Click View Full History Item	Click View History Button	ViewHistory	1.005	
				Customer's Profile is Shown	Checkout Form	Click Full Customer Profile	Click Full Customer Profile	ViewProfile	1.006	
				Items on page will be removed from the item inventory table, Customer History is updated with new items just purchased	Checkout Form	Click Complete Checkout Button	Click Complete Checkout Button	CompleteCheckout	1.007	
				User will be brought back to main landing page	Checkout Form	Click Main Menu Button	Click Main Menu Button	MainMenu	1.008	

9.3.6. Unit Test: Record Donation

Test Date	Tested By	Comments	Observed result	Expected result	State Before Test	Steps to be Executed	Action to perform test (input)	Description	Test Number	Pass/Fail Status
				Add Donor Name to Checkout Form	New Donation Form	Enter donor name into form	Donor is in the database and must abide by the restrictions stated in the Data Dictionary	ValidDonorName	1.001	
				Item Added to Database	New Donation Form	Enter item into input on form	Add an item to the database	ValidItemName	1.002	
				Item description Added to Database	New Donation Form	Enter item description into input on form	Add an item's description	ValidItemDescp	1.003	
				Item gender added to database	New Donation Form	Enter item gender into input on form	Add an item's gender	ValidItemGender	1.004	
				Error	New Donation Form	Enter item into input on form	Add an item name that is too long	InvalidItemName	1.005	
				Error, prompt user to add new donor	New Donation Form	Enter donor into input on form	Add donor not in the system	InvalidDonor	1.006	
				Error	New Donation Form	Enter null value into input on form	Add null value as donor name	NullDonorName	1.008	
				Error	New Donation Form	Enter null value into input on form	Add null value as item name	NullItemName	1.009	
				Error	New Donation Form	Enter item description into input on form	Add an item's description that is the wrong length	InvalidItemDescp	1.010	

Test Date	Tested By	Comments	Observed result	Expected result	State Before Test	Steps to be Executed	Action to perform test (input)	Description	Test Number	Pass/Fail Status
				Error	New Donation Form	Enter null value into input on form	Add a null value as item description	NullItemDescp	1.011	
				Error	New Donation Form	Enter item gender into input on form	Add an item's gender that is not either M, F, or U	InvalidItemGender	1.012	
				Error, prompt user to add gender	New Donation Form	Enter null value into input on form	Add a null value as item gender	NullItemGender	1.013	

9.3.7. Unit Test: Edit Customer/Donor

Test Date	Tested By	Comments	Observed result	Expected result	State Before Test	Steps to be Executed	Action to perform test (input)	Description	Test Number	Pass/Fail Status
				New First Name Added to Database	Create Customer Form	Enter First Name in form	Edit Valid First Name	EditFirstName	1.001	
				New Last Name Added to Database	Create Customer Form	Enter Last Name in Form	Edit Valid Last Name	EditLastName	1.002	
				New Address Added to Database	Create Customer Form	Enter Address consisting of the customer's full street address, must be under 70 characters	Edit Valid Address Information	EditAddress	1.003	

Test Date	Tested By	Comments	Observed result	Expected result	State Before Test	Steps to be Executed	Action to perform test (input)	Description	Test Number	Pass/Fail Status
				New Dependent's First Name Added to Database	Create Customer Form	Enter Dependent First Name in Form	Edit Dependent First Name	EditDepFirstName	1.004	
				New Dependent's Last Name Added to Database	Create Customer Form	Enter Dependent Last name in Form	Edit Dependent Last Name	EditDepLastName	1.005	
				New Dependent's Relation Added to Database	Create Customer Form	Enter Dependent's Relation to customer in Form	Edit Dependent Relation to customer	EditDepRelation	1.006	
				New Dependent Gender Added to Database	Create Customer Form	Enter Dependent Gender into Form	Edit Valid Dependent Gender	EditDepGender	1.007	

9.3.8. Unit Test: Add Volunteer

Test Date	Tested By	Comments	Observed result	Expected result	State Before Test	Steps to be Executed	Action to perform test (input)	Description	Test Number	Pass/Fail Status
				First Name Added to Database	Create Volunteer Form	Enter First Name in form	Add Valid First Name	ValidFirstName	1.001	
				Last Name Added to Database	Create Volunteer Form	Enter Last Name in Form	Add Valid Last Name	ValidLastName	1.002	
				Username Added to Database	Create Volunteer Form	Enter username consisting of first initial and last name into form	Add Valid username	ValidUsername	1.003	
				Temporary Password Added to Database	Create Volunteer Form	Password containing at least one capital letter and one number	Add Valid Temporary Password	ValidPassword	1.004	
				Error	Create Volunteer Form	Pass Null Value In	Add Invalid First name	NoFirstName	1.005	
				Error	Create Volunteer Form	Pass Null Value In	Add Invalid Last name	NoLastName	1.006	
				Error	Create Volunteer Form	Pass Null Value In	Add Invalid Username	NoUsername	1.007	
				Error	Create Volunteer Form	Type in username using incorrect formatting	Add Invalid Username	WrongFormat Username	1.008	

Test Date	Tested By	Comments	Observed result	Expected result	State Before Test	Steps to be Executed	Action to perform test (input)	Description	Test Number	Pass/Fail Status
				Error	Create Volunteer Form	Pass Password that is too short	Add Invalid Password	PasswordTooShort	1.009	
				Error	Create Volunteer Form	Pass Password that is too long	Add Invalid Password	PasswordTooLong	1.010	
				Error	Create Volunteer Form	Pass password that is all letters	Add Invalid Password	AllLettersPassword	1.011	
				Error	Create Volunteer Form	Pass Password That is all Numbers	Add Invalid Password	AllNumPassword	1.012	
				Error	Create Volunteer Form	Pass Password That is all lowercase	Add Invalid Password	LowercasePassword	1.013	

10. Development and Production Environments

10.1. Development Environment

Windows Computer

Operating System: Windows 7 Enterprise (x64) Service Pack 1
Processor: Intel Core i5-3470 @ 3.20 GHz
Ram: 6GB
HDD Capacity: 499 GB

Macintosh Computer

Operating System: OS X Lion 10.7.5
Processor: Intel Core i5 @ 2.5 GHz
Ram: 4GB
HDD Capacity: 378 GB

10.2. Operating Environment

This information has yet to be determined by the client. This application will be web-based, so it will operate from an off-site server. The application is designed to be as simple and easy to operate as possible, to allow anyone to easily use it.

10.3. Maintenance

Maintaining this application involves ensuring that the information is correct and up-to-date, and making sure that any updates to the server are remain compatible with S.W.I.F.T. as server maintenance is completed by the third party that houses the servers.

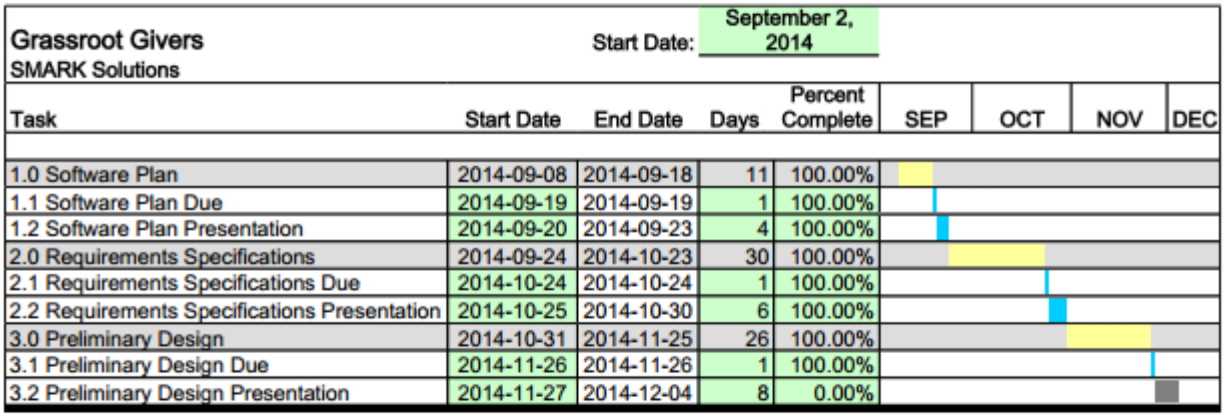
11. Appendices

11.1. Appendix C: Glossary of Terms

Gantt Chart - Bar chart typically used to project scheduling
Data Flow Diagram - A visual representation of how data moves throughout a system.
Database - An organized collection of data.
Functional Requirements - Defines what the system will be able to do and what is testable about the system.
Non-Functional Requirements - Requirements that are not necessarily specific features that exist in a system, but what the system is intended to do.
Processor - The part of the computer that handles and executes operations.
Prototype - An early sample, model or release of a product built to test a concept.
Random Access Memory (RAM) - a memory unit that allows any specific byte to be used randomly at any time.
Server - a computer or program that manages access to a resource or service in a network.
S.W.I.F.T. - Simple Web Inventory For Tracking
UML Use Case Diagram - A visual representation of the users interaction with the system in a specific instance.
Use Case Narrative - a written explanation of the course of events a user will encounter when interacting with the system

11.2. Appendix D: Timeline

Development Timeline:



Grassroot Givers
SMARK Solutions

Start Date: September 2, 2014

Task	Start Date	End Date	Days	Percent Complete	SEP	OCT	NOV	DEC
1.0 Software Plan	2014-09-08	2014-09-18	11	100.00%	🟨			
1.1 Software Plan Due	2014-09-19	2014-09-19	1	100.00%	🟦			
1.2 Software Plan Presentation	2014-09-20	2014-09-23	4	100.00%	🟦			
2.0 Requirements Specifications	2014-09-24	2014-10-23	30	100.00%	🟨	🟨		
2.1 Requirements Specifications Due	2014-10-24	2014-10-24	1	100.00%		🟦		
2.2 Requirements Specifications Presentation	2014-10-25	2014-10-30	6	100.00%		🟦		
3.0 Preliminary Design	2014-10-31	2014-11-25	26	100.00%			🟨	
3.1 Preliminary Design Due	2014-11-26	2014-11-26	1	100.00%			🟦	
3.2 Preliminary Design Presentation	2014-11-27	2014-12-04	8	0.00%			▨	▨