



Team 9

Andrew Morton

John Pizzitola

Johnathon Phillips

Project Summary

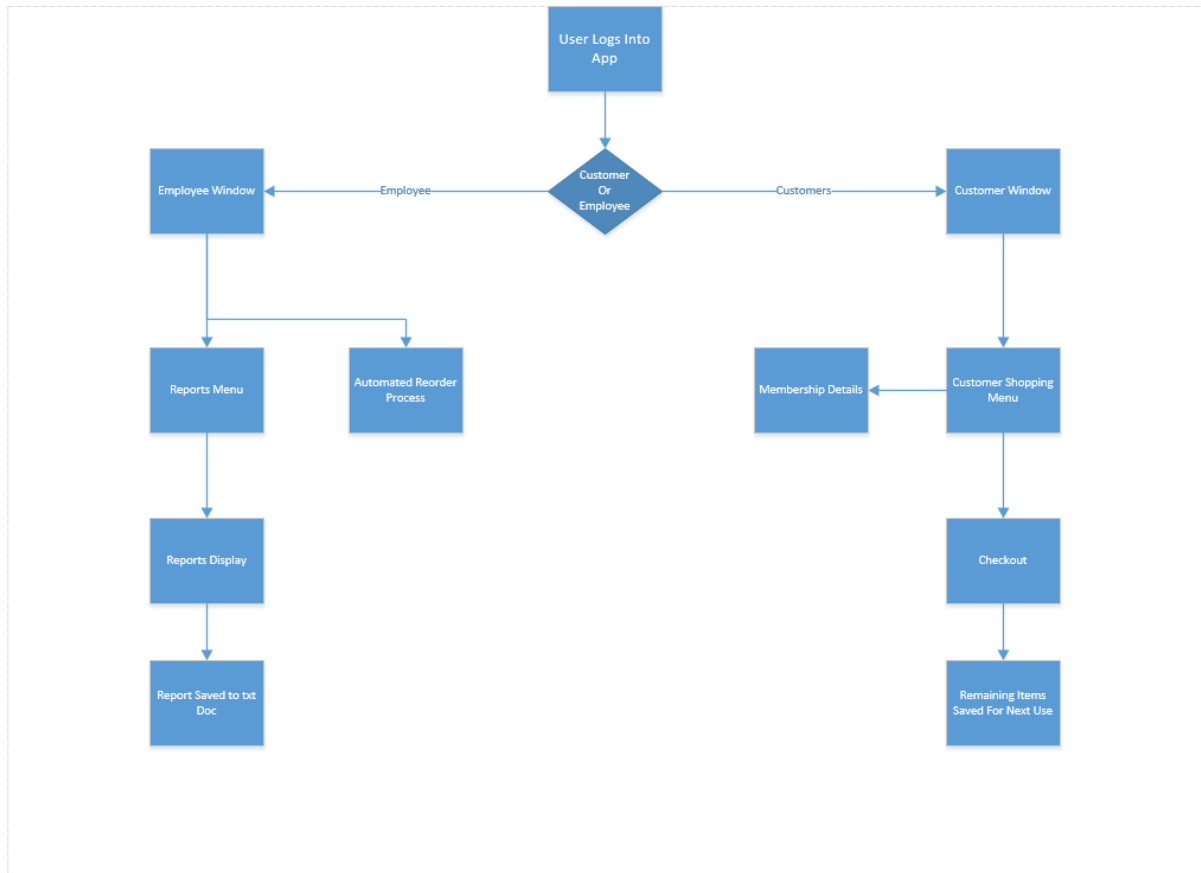
For our project we will design an inventory management system, and point of sale application for AntLife. This application will allow the user to automatically reorder products when the stock levels get low, allow a user to create a wish list and buy items from it, and see if the items are currently in stock. The wish list will also retain items between visits to AntLife. The application will also allow users to sign up for a membership with AntLife, and determine when their current membership tier and will show the user their progress towards bronze silver and gold. This application will also encrypt any credit card data or passwords to ensure user security.

Requirements and Assumptions

- Determine if items have expired
- Automatically reorder items when stock is low
- Allow reductions in inventory for shrinkage and expiration
- Use a csv file to add new items
- Allow use of USD and CAD as currency
- Allow for federal and local tax rates
- Save user logins, and create new users
- Allow addition of new employees
- Verify DOB for age restricted items
- Print Receipt
- Allow for returns
- Generate a listing of current members
- Generate a listing of expired members
- Generate a listing of expired products to be removed from the store
- Generate a listing of most sold products
- Generate a listing of products that have not been sold in the past 12 months

Business Process Overview

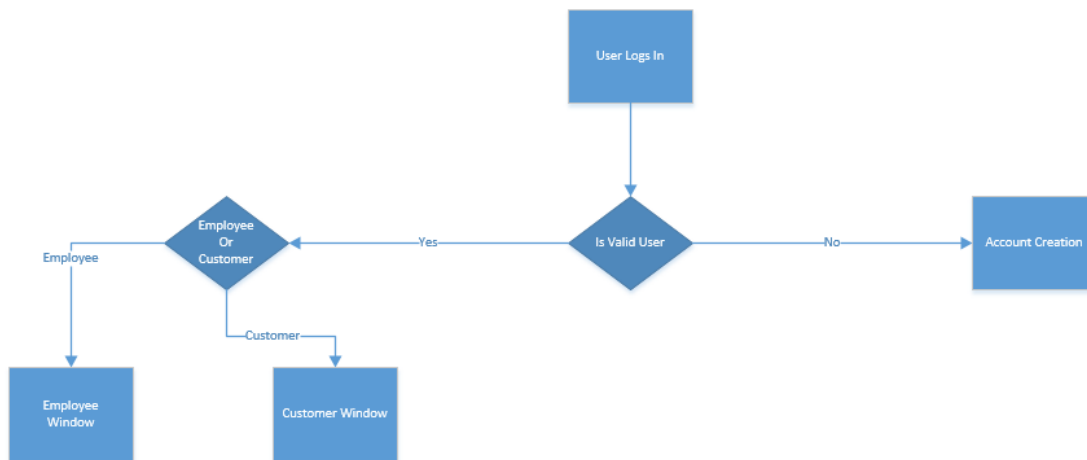
High Level



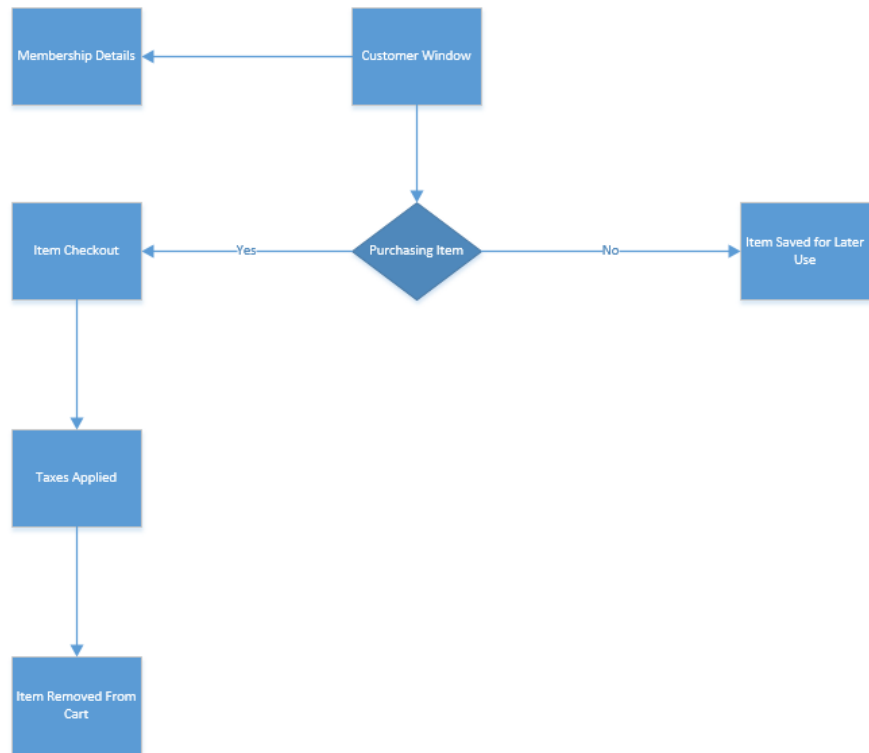
Low Level

1. Low Level

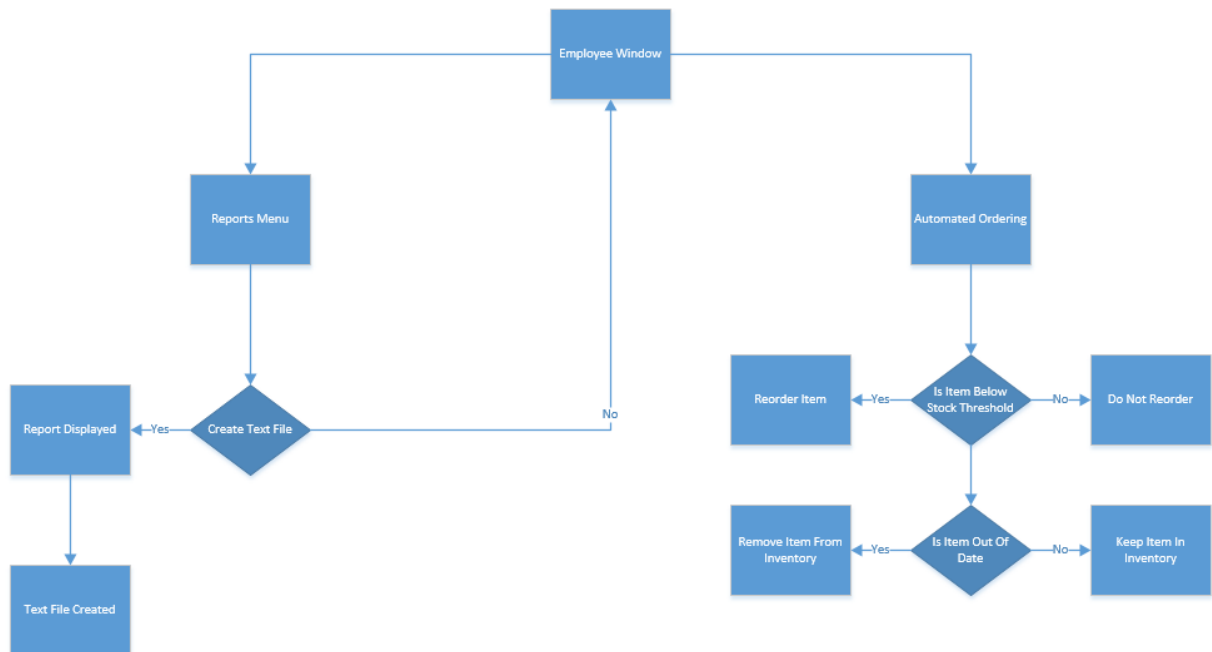
+



2. Low Level Customer



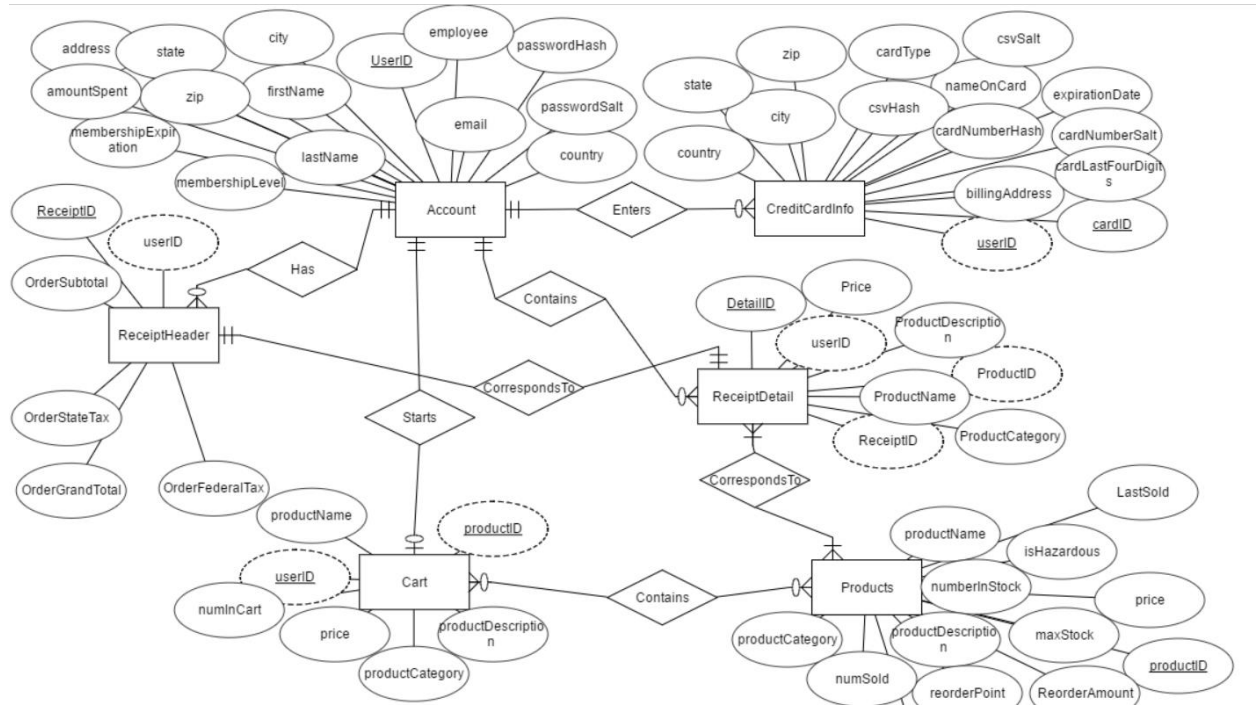
3. Low Level Employee



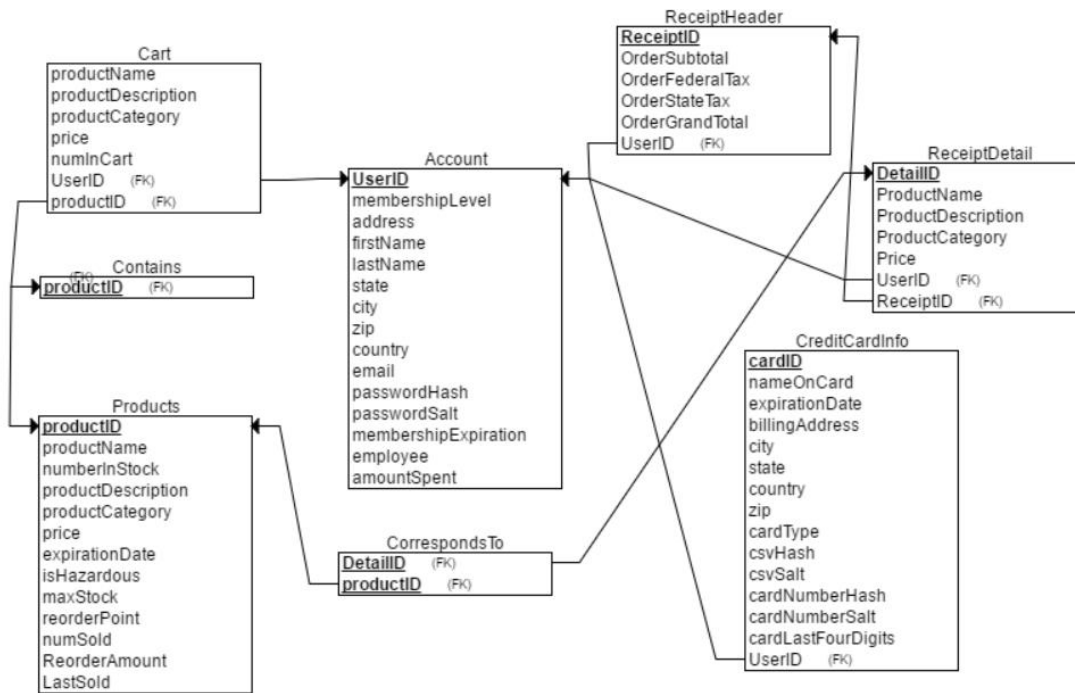
Data Design

Data Model

ERD



Relational Schema



Data Dictionary

Data	SQL Type	C# Type
UserID	Int	Int
firstName	VarChar(25)	String
Employee	VarChar (3) (yes or no)	Bool
MembershipExpiration	Date	DateTime
lastName	VarChar(25)	String
Address	VarChar(100)	String
city	Varchar(25)	String
Zip	VarChar(7)	String
State	Varchar(30)	String
Country	Varchar(50)	String
Email	Varchar(100)	String
passwordHash	Varchar(256)	string
passwordSalt	Varchar(256)	String
membershipLevel	Varchar(15)	String
amountSpent	Float	Double
cardID	Int	Int
nameOnCard	Varchar(100)	String
expirationDate	Date	Datetime
billingAddress	Varchar(100)	String
cardType	Varchar(30)	String
csvHash	Varchar (256)	String
csvSalt	Varchar(256)	String
cardNumberHash	Varchar(256)	String
cardNumberSalt	Varchar(256)	String
cardLastFourDigits	VarChar(4)	String
productDescription	Varchar(255)	String
ProductCategory	Varchar()	
ProductID	Int	Int
productName	VarChar(100)	String
Price	Varchar(20)	Double
expirationDate	Date	Datetime
isHazardous	Varchar(3)	Bool
numberInStock	Int	Int
maxStock	Int	Int
reorderPoint	Int	Int
Numsold	Int	Int
reorderAmount	Int	Int
NumSold	Int	Int

reorderAmount	Int	Int
lastSold	Date	Datetime
receiptID	Int	Int
orderSubTotal	Float	Double
orderFederalTotal	Float	Double
orderStateTotal	Float	Double
orderGrandTotal	Float	Double
detailID	Int	int
Stock	Int	Int
Expiration Date	Date	DateTime

Report Examples

Current Members

File Edit Format View Help

Current Members

MemberID	Member Name	Membership Level	Date Joined
0001	John Doe	Bronze	11/07/2016

Expired Members

File Edit Format View Help

Expired Members

Member ID	Member Name	Membership Level	Date Expired
0003	Dale Gribble	Gold	10/23/2016

Expired Products

File Edit Format View Help

Expired Products

Product ID	Product Name	Date Expired	Number Expired
0003	Milk	11/23/2015	5

Highest Sales

File Edit Format View Help

Most Sold Products

Product ID	Product Name	Number Sold
0007	Milk	5

Not Sold in One Year

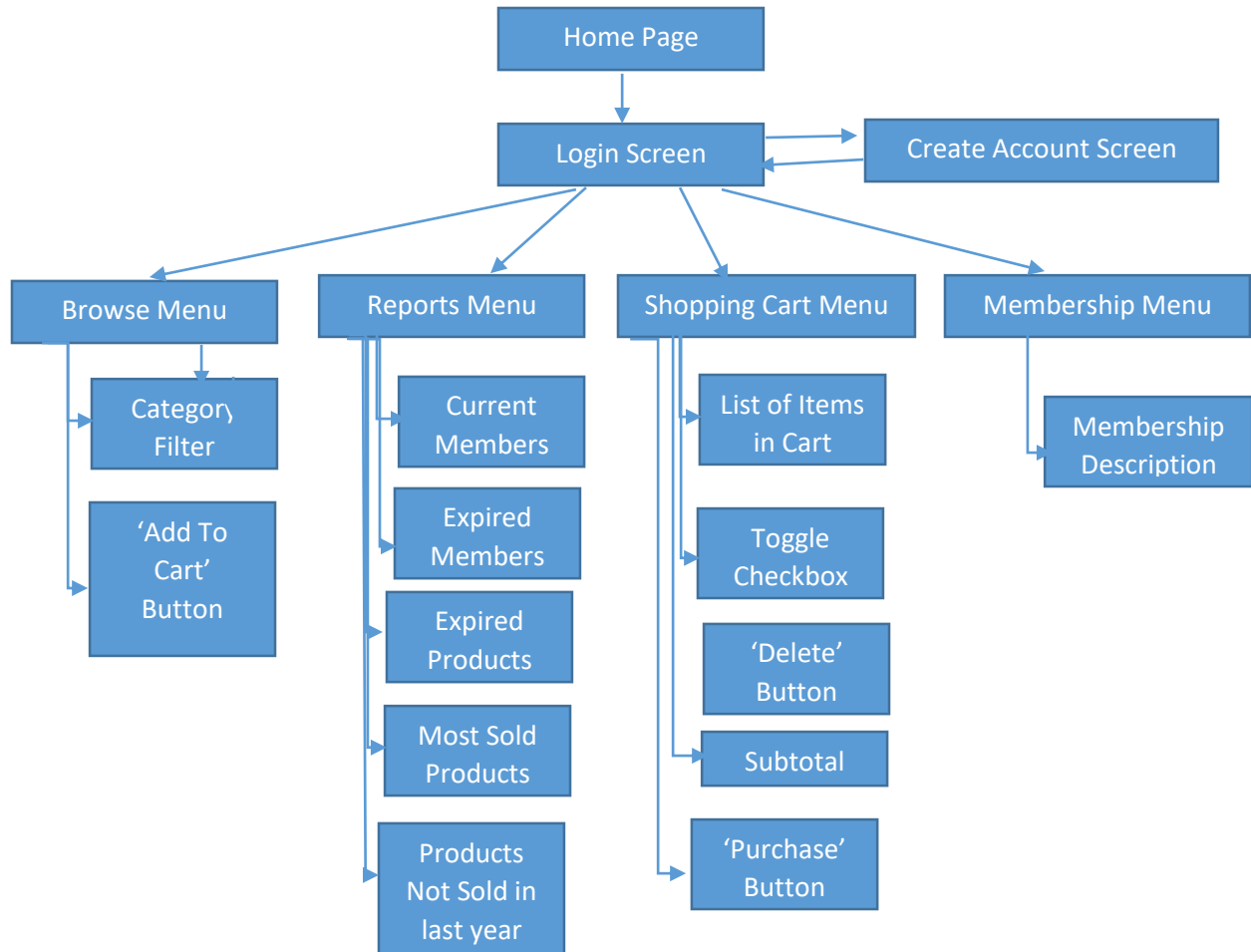
File Edit Format View Help

Not Sold In Past 12 Months

Product ID	Product Name	
0009	Bop-It	

Application Design

Sitemap



Wireframe Designs

Login Page

Welcome to AntLife!

Sign in

AntLife

Email

Password

[Forgot your password?](#)

Not an AntLife rewards member? [Create an Account.](#)

New Account Creation

Welcome to AntLife!

Create Account
AntLife

First Name

Last Name

Address

City

State

Zip

Current Email

Current Email(re-type)

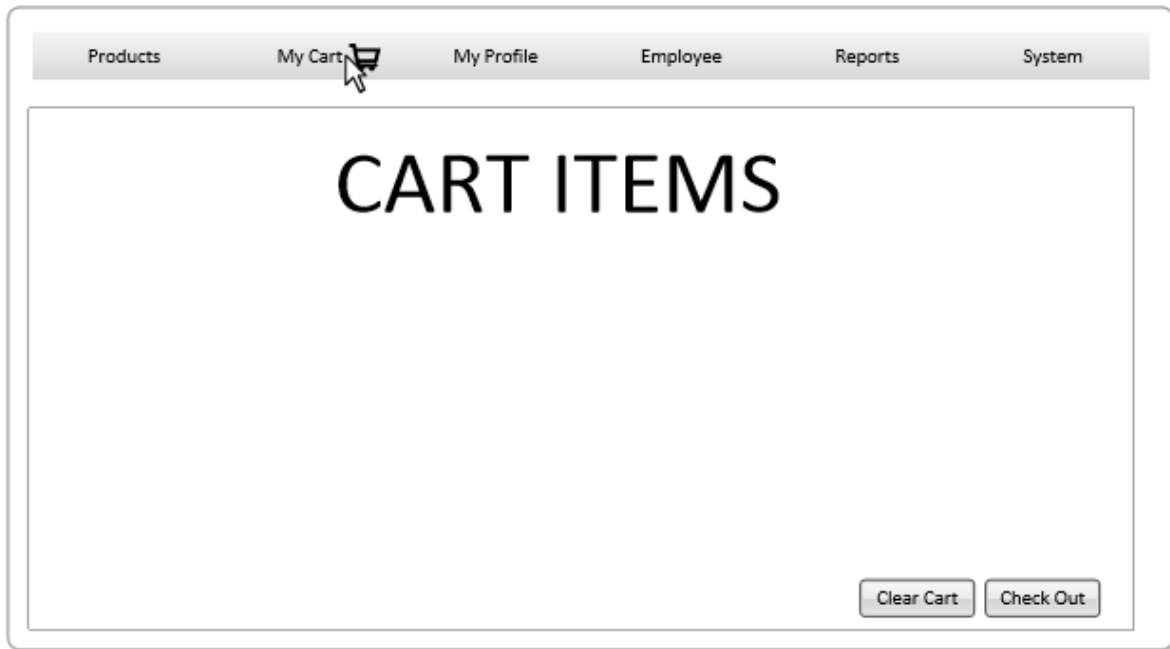
Desired Password

Desired Password(re-type)

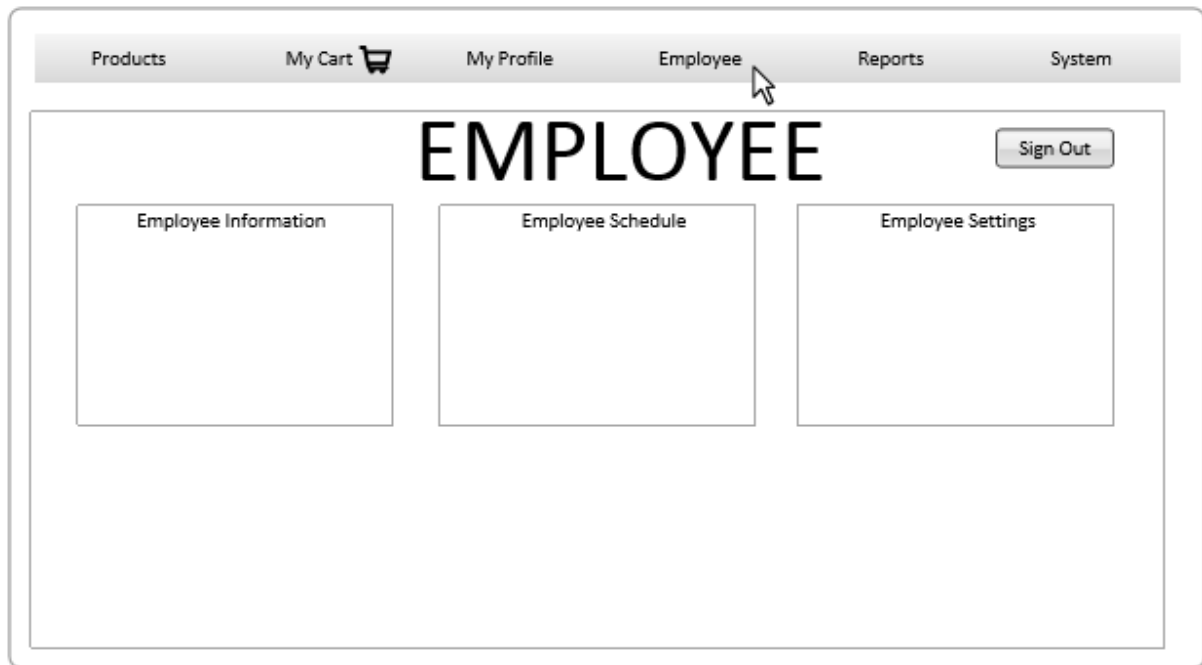
Cancel

Create Account

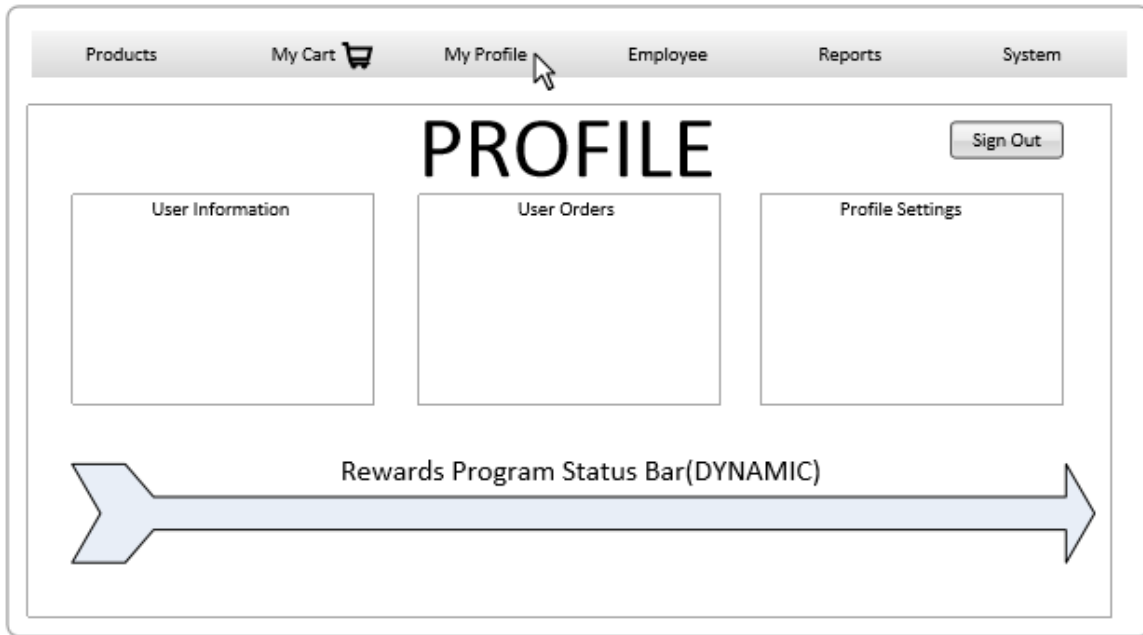
Cart Menu



Employee Menu



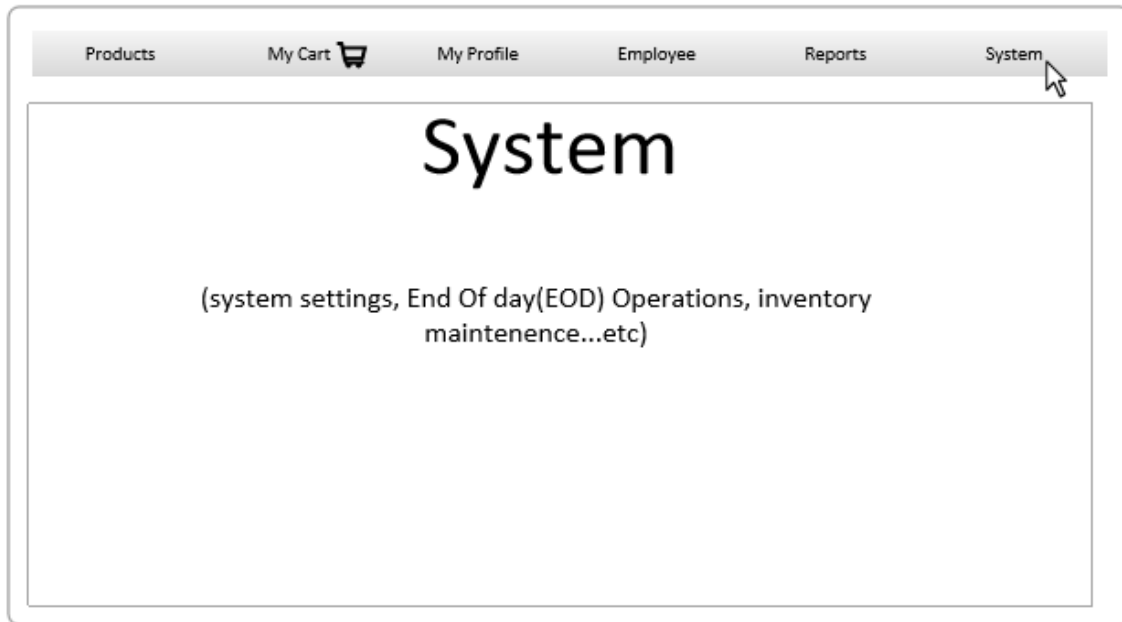
Profile



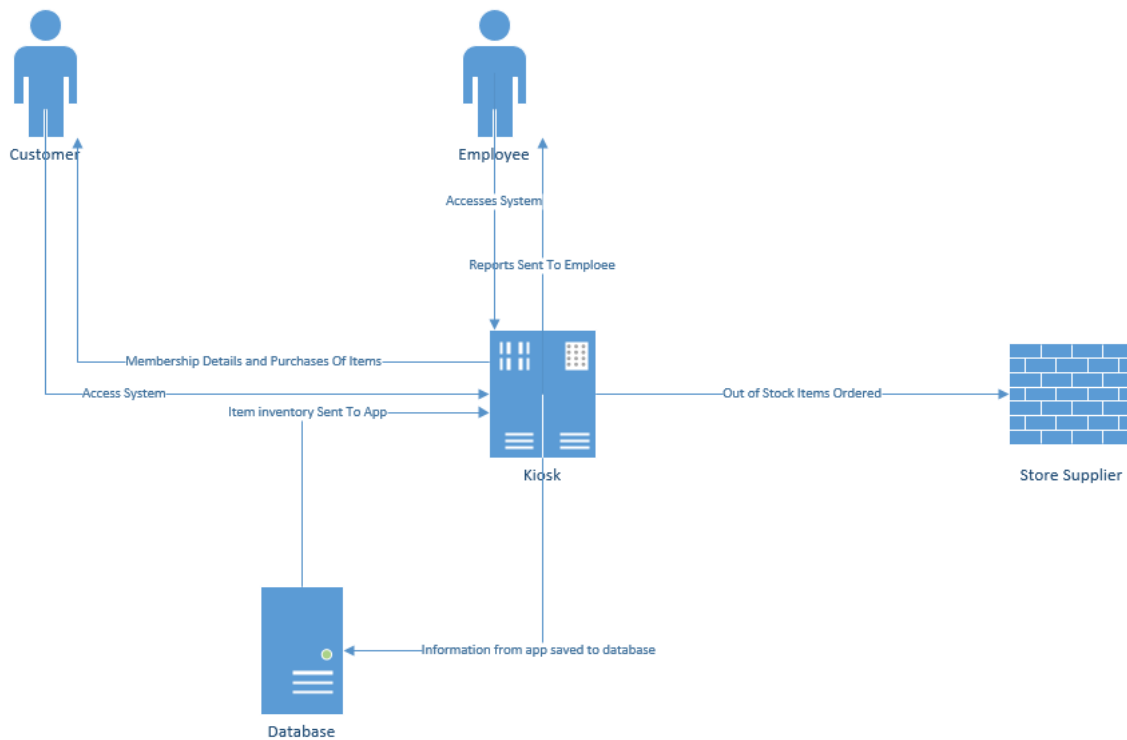
Reports Menu



System Menu

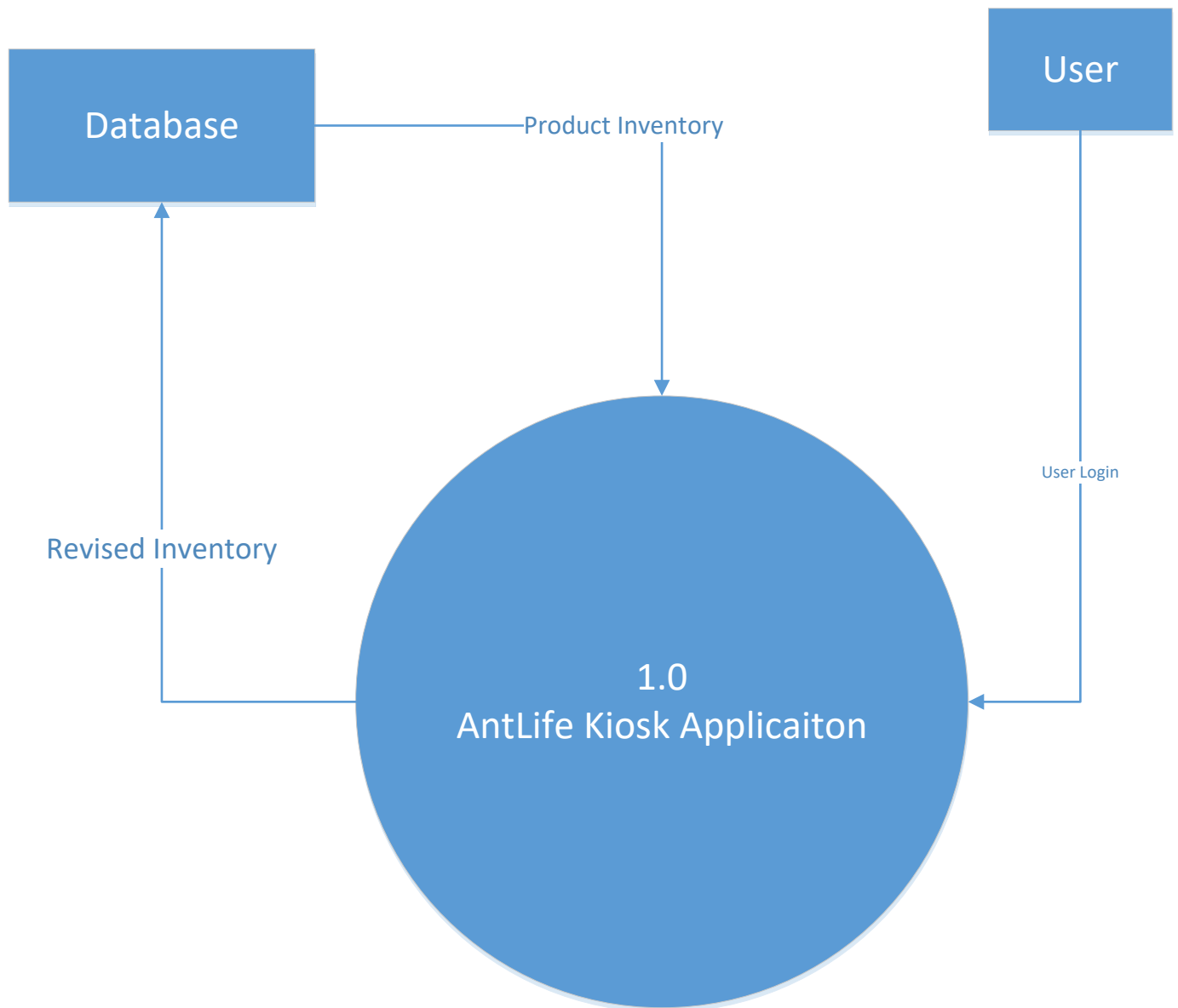


Context Diagram

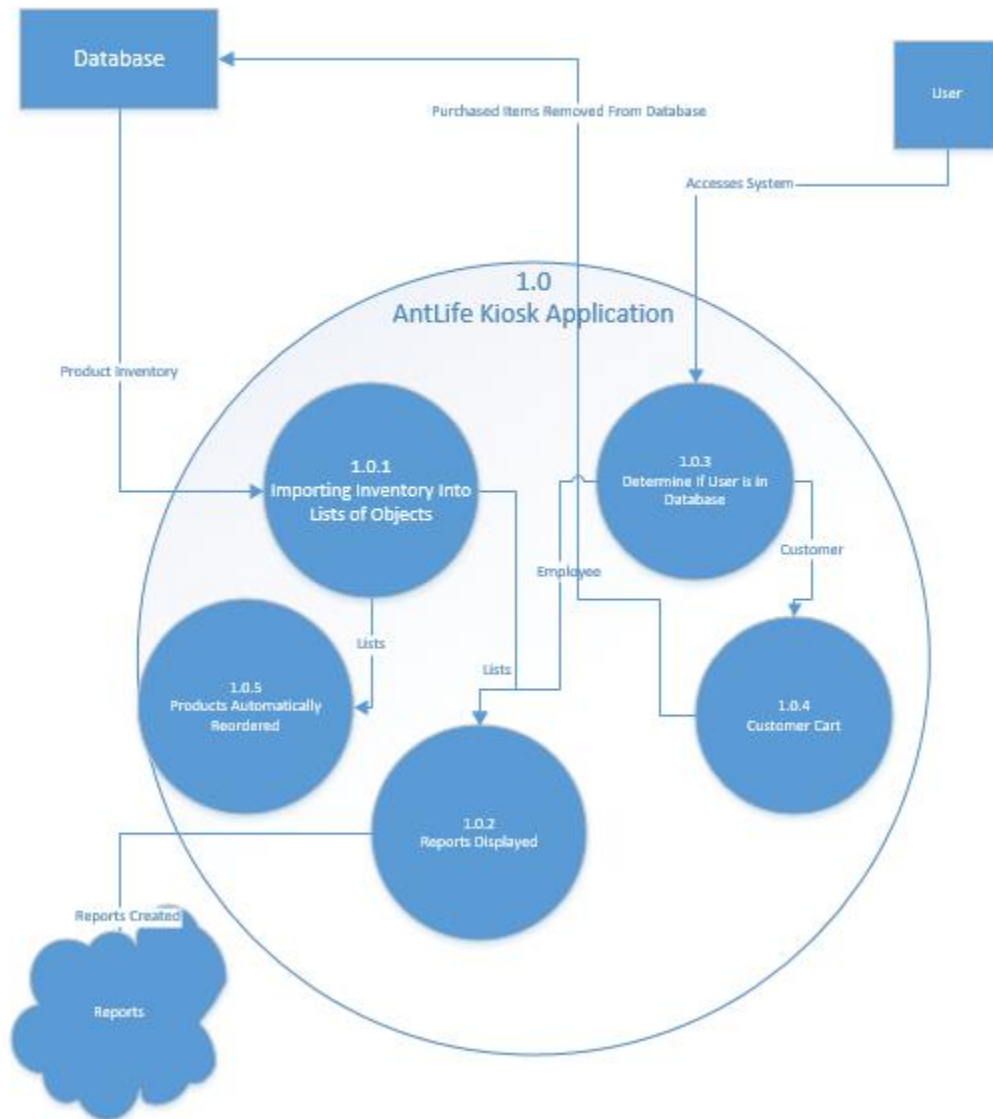


Data Flow Diagrams

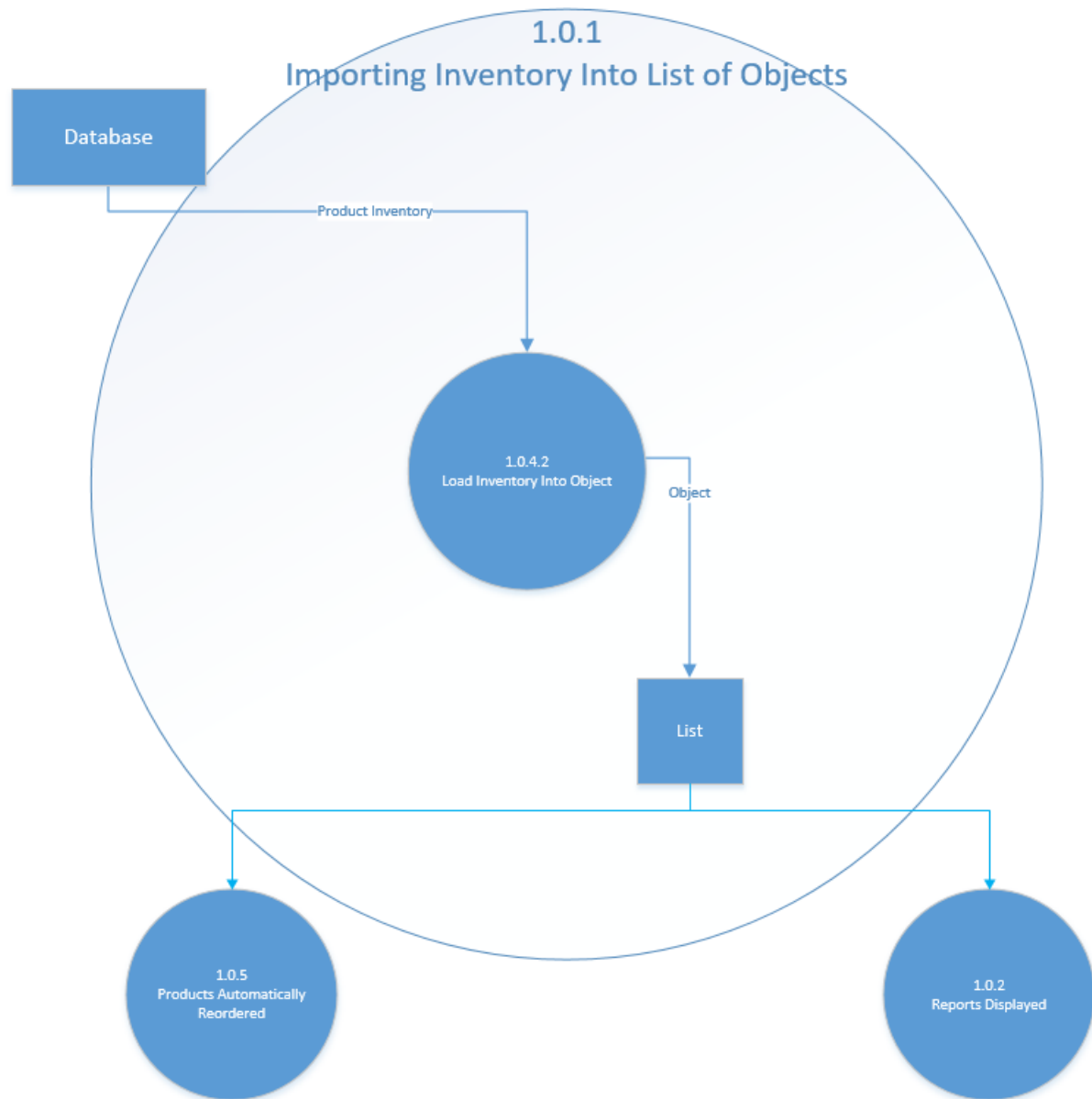
DFD 0



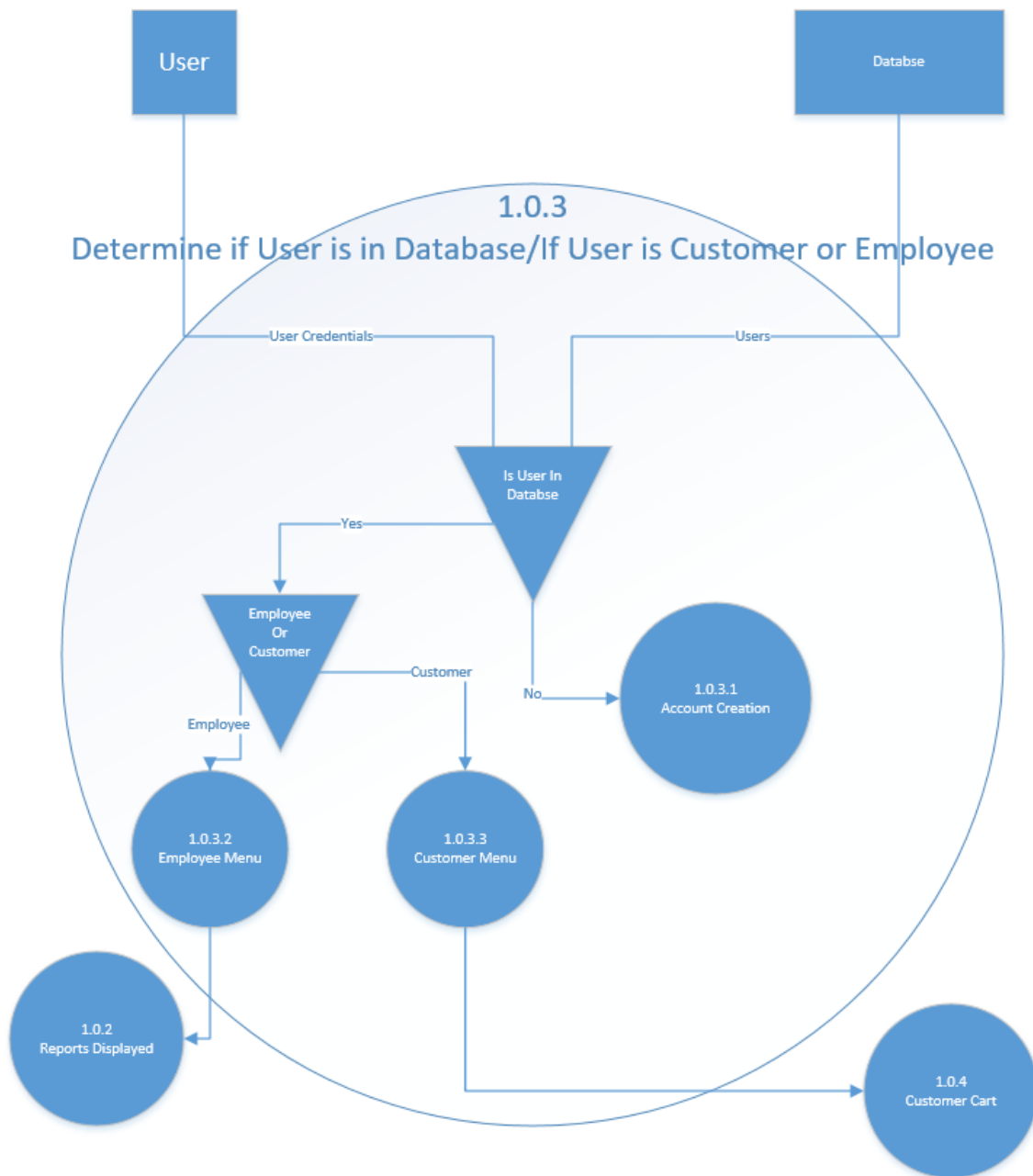
DFD 1.0



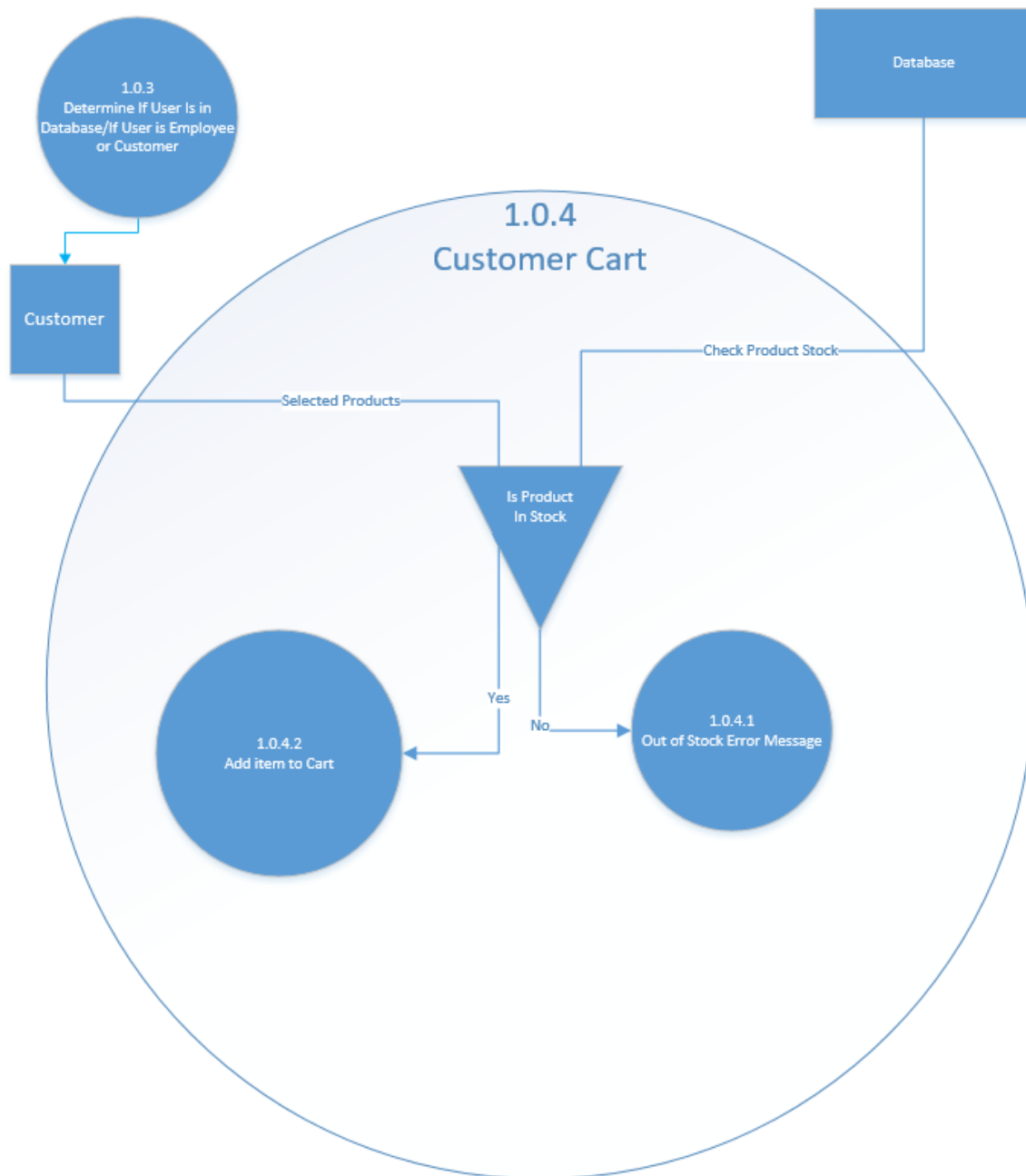
DFD 1.0.1



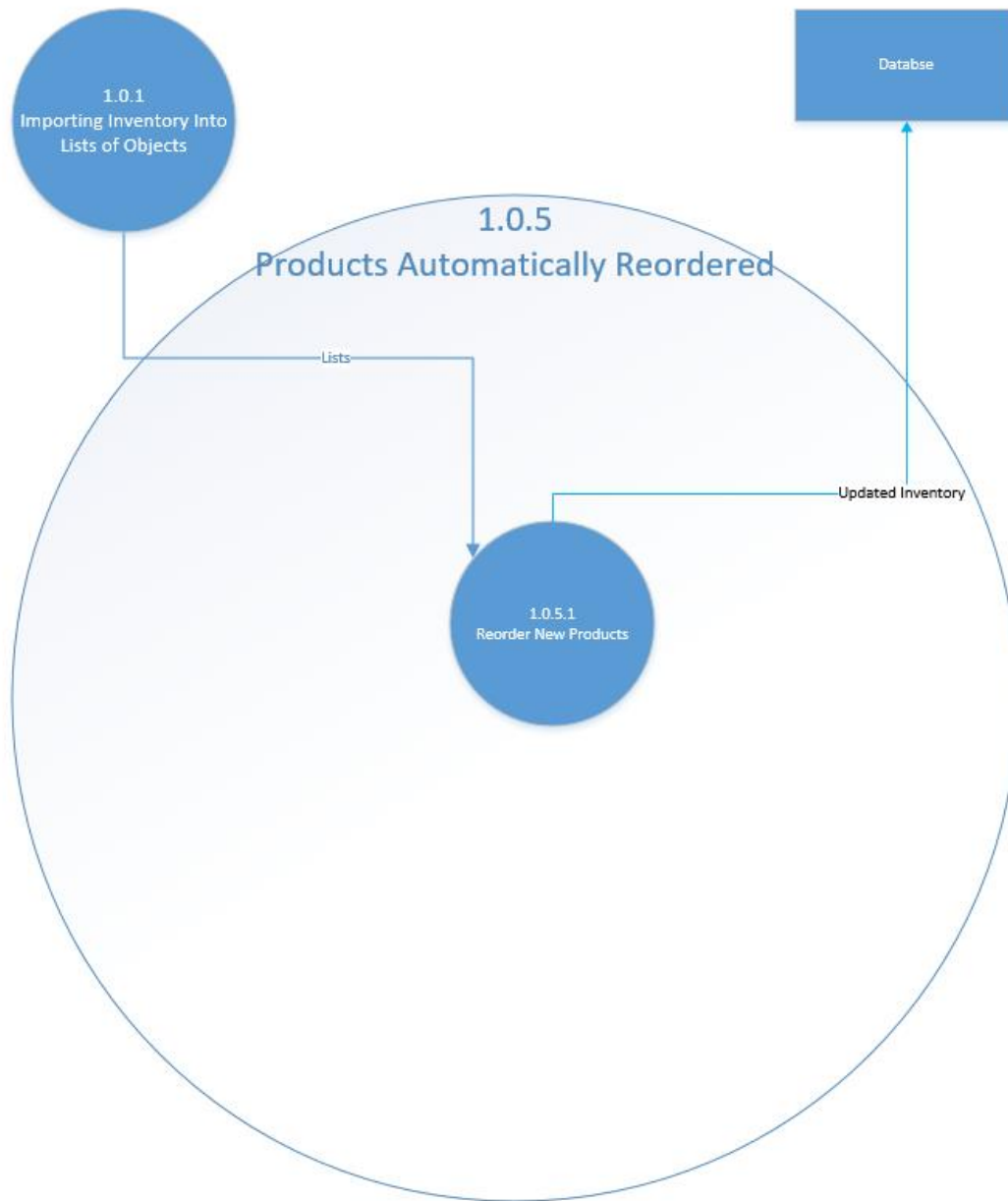
1.0.3



1.0.4



1.0.5



Class UML Diagram

Account
-firstName (String) -lastName (String) -employee (Bool) -address (String) -UserID (Int) -Zip (String) -city (String) -MembershipLevel (String) -email (String) -passwordSalt (String) -passwordHash (String) -expDate (String) -state (String) -amountSpent (double) -country (String)
+get; set; ToString()

Product
-productName (String) -Stock (Int) -productDesc (String) -ProductID (Int) -category (String) -ExpirationDate (DateTiime) -price (double) -numStock (int) -isHazardous (bool) -numSold (int) -lastSold (DateTiime) -detailID (int) -amountOrdered (int) -reorderAmount (int) -receiptID (int) -numInCart (int)
+get; set; ToString()

globalClass
-BackColor (string) -Font (string)

Pseudocode Snippets

Retrieve Data

```
List <Product>  
List <User>  
GetDataFromDB();  
List.Add(Data)
```

Automatic Reordering

```
If Stock level < 3  
    Reorder()
```

User Authentication

```
Iterate through user list  
    If User is not in list throw new account error  
    If User is in list but password hashes do not match throw password error  
    If Valid user is Employee True Send to Employee Page  
    Else Send to Customer Page
```

User Data Encryption

```
Convert plaintext password into byte array  
Create a randomly generated salt to append to the password  
Hash the new byte array through sha 256  
Save the password hash and salt hash to the database
```