Virtual Fridge

System Specification

Electronically assisted food storage system

Prepared For:

Elaine Weltz, Entrepreneur

Homestyle Solutions

Prepared By:

Naveen Janarthanan, Consultant

Estar System Development

Table of Contents

Executive Summary

Elaine Weltz of Homestyle Solutions approached Estar System Design to create a Virtual Fridge System. It will be available across mobile phones, tablets and laptops. The software will be used to store information of food that is currently in the user's fridge. It will store information such as the expiration date and the quantity of the item. Additionally the software will be able to purchase new items through the application, when an item of food is running low or it out.

Now that Estar System Development has completed the analysis for the software system, was have created this System Specification to assist in the design and creation of Cuisine by Car. This document will be detailing more of the internal and technical aspects of the system, as well as providing an overview for some of the user interfaces for the system. Virtual Fridge will be relatively easy to implement, and will provide great return in the future.

1.0 Introduction

1.1 Problem Statement and Project Vision

Homestyle Solutions came to us with an idea for a Virtual Fridge, because they realized that food was being thrown out because people were forgetting about the expiration date. Homestyle Solutions, also came to the conclusion that people find it difficult to remember what is still in the fridge. The Virtual Fridge is intended to be used as a tool to assist in grocery shopping and to track what items the user currently has in their fridge. We plan to build an application that will serve exactly this purpose. **More information on this can be found on page 4 of the System Proposal.**

Stakeholders:

- Families: Want to not throughout food because they forgot the expiration date and the be more efficient while shopping
- Grocery Stores: Will be advertise through the software, shopping will be more efficient at their stores
- Elaine Weltz of Homestyle Solutions: Would like to see Virtual Fridge be used across the nation, it would also benefit her personally.
- Estar System Development: Would like to develop Virtual Fridge to increase profit.
- Famers: Less food will be wasted, so they won't have to as high of yield every year

1.2 System Services

The following is a summary of the services the CSCI will provide. Please refer to the Requirements Definition and Requirements Model sections on pages 13-16 of the System Proposal for more details.

- Allow users to create an account. (Use Case 1, page 13)
- Allow users to create a shopping list. (Use Case 2, page 13)
- Allow users to add items to the shopping list. (Use Case 3, page 13)
- Allow users to remove items from the shopping list. (Use Case 4, page 13)
- Allow users to make payments to grocery store vendors. (Use case 5, page 13)
- Allow users to use the camera for various activities within the application (Use Case 6, page 13)
- Allow the admin to edit the database for grocery store items. (Use case 7, page 13)
- Allow the user to suggest to the admin to add item to the database. (Use case 8, page 13)
- Allow the user to add items to the fridge (Use case 9, page 13)
- Allow the user to remove items from the fridge (Use case 10, page 13)

1.3 Nonfunctional Requirements and Design Constraints

The following is a summary of Virtual Fridge's major constraints and major non-functional requirements. For additional details, refer to the Constraints section on page 5 and the Nonfunctional Requirements on page 14 of the System Proposal.

- The system will have to work across a variety of different devices that will be of different ages
- The application must be user-friendly and easy to use as clients will vary by age and experience with technology.
- User information such as login and passwords must be safely encrypted to ensure that no unauthorized person have access to it
- The software will have to work across Android, Apple and Windows phones and table, as well as Laptops.

1.4 System Evolution

The initial version of Virtual Fridge will include only the functionalities listed in the System Services section above. Refer to pages 13-16 of the System Proposal for more information. In future releases. Estar System Solutions will be incorporating additional features to Virtual Fridge to improve the functionalities of the system. We hope to allow add a pressure sensor that would sense when certain items are running low. Also we would like to add maps of grocery stores, so users can quickly locate items when they enter. For more details about the System Evolution, please see page 27 of the System Proposal.

1.5 Document Outline

This section contains a general outline for the rest of the System Specification. This document contains:

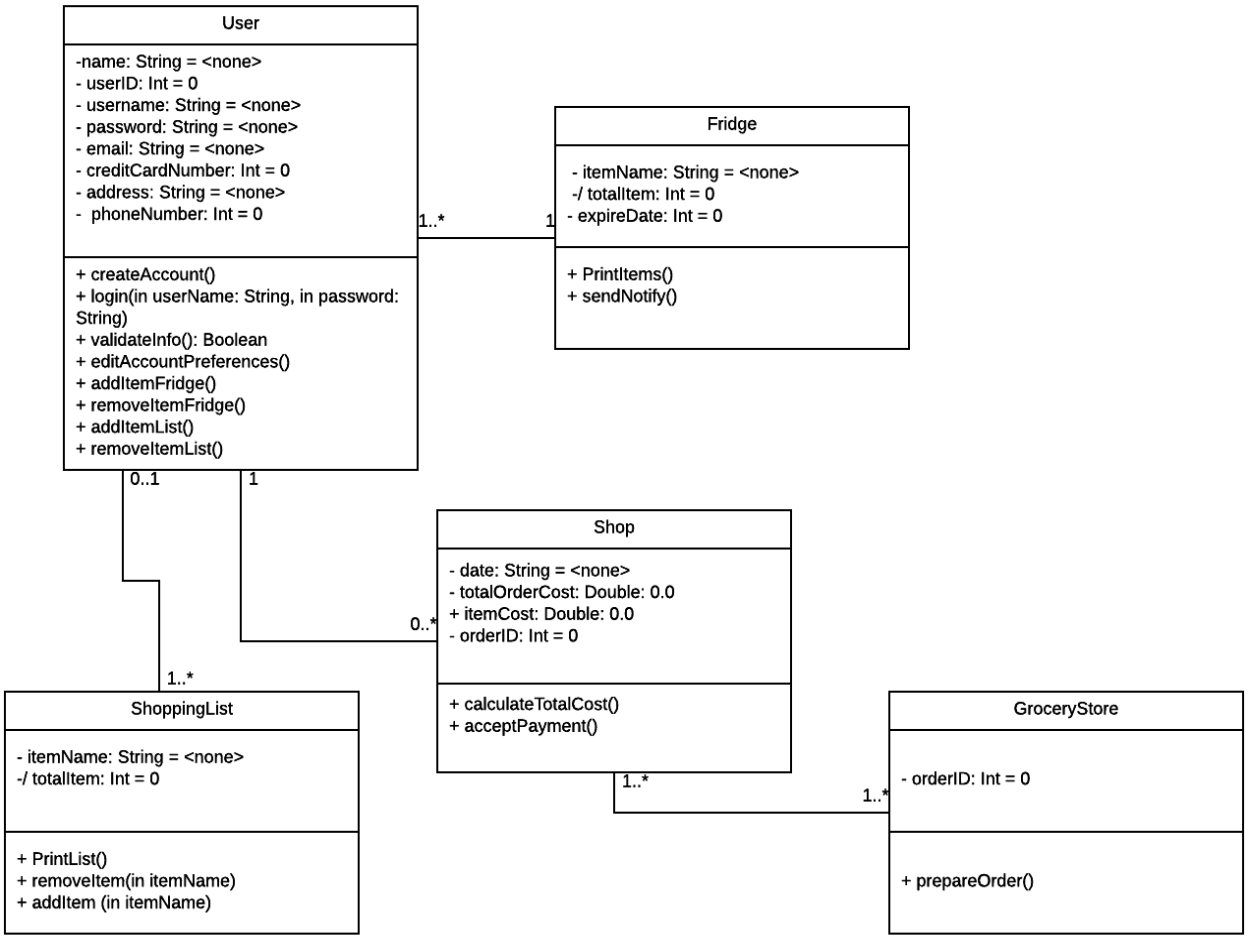
- Structural Model: A class diagram describing the software components necessary for CBC, as well as metadata to describe functions within the classes
- Architecture Design: This section will include an infrastructure model of the physical system architecture, the hardware and software requirements for the system, as well as a security plan
- User-Interface: This section covers the user-interface requirements and constraints, screen/user-interaction design, and a printed output design
- Appendices: Included in this section is a bibliography

2.0 Structural Model

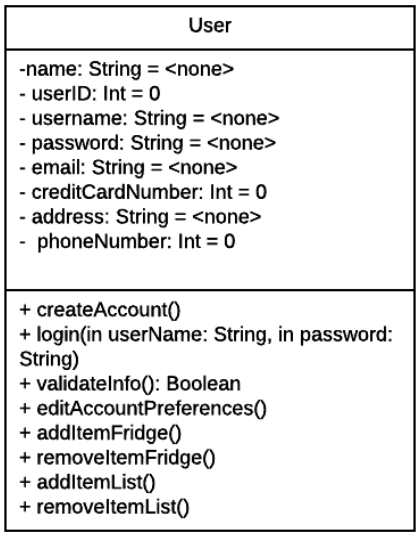
2.1 Introduction

This section contains a UML class diagram displaying the classes of Virtual Fridge, as well as the attributes and operations for each class. This section also contains detailed metadata to show each class in greater detail. The metadata will display the class model, followed by the class description, attributes, operations, and the documentation for each of the operations.

2.2 Class Diagram



2.3 Metadata



Description: Represents a user of Cuisine by Car
Visibility: Public
Is Abstract: No

Attributes:

Name	Description	Data Type	Is Derived	Read Only	Visibility	Multiplicity	Default Value
name	User's name	String	No	No	Private	1	<None>
userID	ID number of the user	Int	No	Yes	Private	1	<None>
username	Username used by user to login	String	No	No	Private	1	<None>
password	Password used by user to login	String	No	No	Private	1	<None>
Email	Email of the user	String	No	No	Private	1	<None>
creditCardNumber	Credit card for user payment	String	No	No	Private	1..*	0
Address	Address of user	String	No	No	Private	1	<None>
phoneNumber	Phone number of user	Int	No	No	Private	1..*	0

Operations:

Name	Description	Return Type	Parameters	Visibility	Scope	Is Query	Is Polymorphic
createAccount	Users register for an account	<None>	<None>	Public	Instance	No	No
Login	User logs into the system	<None>	tryName Direction: in String tryPassword Direction: in String Default: None	Public	Instance	Yes	No
validateInfo	Validate login information	Boolean	<None>	Public	Instance	Yes	No
editAccountPreferances	Allow users to edit their account information	<None>	<None>	Public	Instance	No	No
addItemFridge()	Allow users to add an item to the fridge	<None>	<None>	Public	Instance	Yes	No

removeItemFridge()	Allows users to remove an item from the fridge	<None>	<None>	Public	Instance	Yes	No
addItemList()	Allows users to add item to shopping list	<None>	<None>	Public	Instance	Yes	Yes
removeItemList()	Allows user to remove item from shopping list	<None>	<None>	Public	Instance	Yes	Yes

Processing outlines:

createAccount:

IF username not found of user click on “Register New User” button
Create new user account

login:

Validate username and password
IF valid
Allow access
ELSE
Re-prompt user to re-enter userName and password

validateInfo:

Validate userName and password to make sure login credentials were entered correctly

editAccountPreferences:

Allow user to edit their account preference and settings

addItemFridge:

Allow user to add an item the fridge

removeItemFridge:

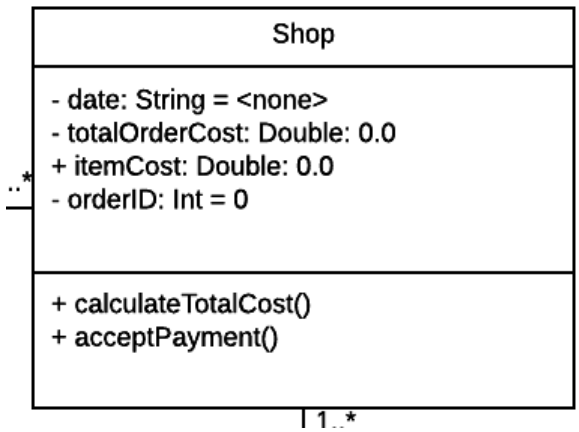
Allow user to remove and item from the fridge

addItemList:

Allow user to add item to the shopping list

removeItemList:

Allow user to add item to the shopping list



Description: Represents order placed by user
Visibility: Public
Is Abstract: No

Attributes:

Name	Description	Data Type	Is Derived	Read Only	Visibility	Multiplicity	Default Value
date	Date order placed	String	No	Yes	Private	1	<String>
orderID	Order number of user	Int	No	Yes	Private	1..*	0
totalOrderCost	Total cost of all items	Double	No	Yes	Private	1	0.0
ItemCost	Cost of an item	Double	No	No	Private	1	0.0

Operations:

Name	Description	Return Type	Parameters	Visibility	Scope	Is Query	Is Polymorphic
calculateTotalCost	Calculate total cost of order	Double	<None>	Public	Instance	Yes	No
acceptPayment	Allow user to pay for order	<None>	<None>	Public	Instance	Yes	No

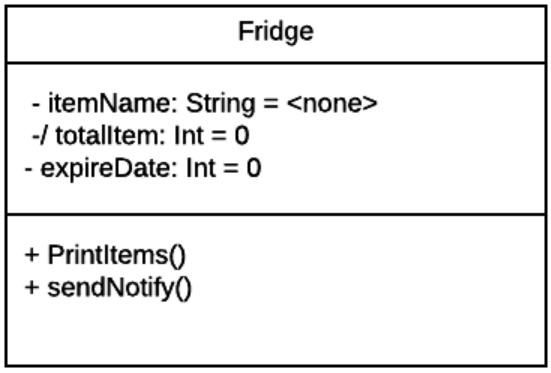
Processing Outlines:

calculateTotalCost:

Add up all of the items order by the user and return totalOrderCost.

acceptPayment:

Accepts user payment for totalOrderCost.



Description: Represents virtual fridge and items inside the fridge
Visibility: Public
Is Abstract: No

Attributes:

Name	Description	Data Type	Is Derived	Read Only	Visibility	Multiplicity	Default Value
itemName	Name of item	String	No	Yes	Private	1	<none>
totalItem	Number of items in the fridge	Int	Yes	Yes	Private	1	0
expireDate	Expiration dates of items	Int	No	Yes	Private	1	0

Operations:

Name	Description	Return Type	Parameters	Visibility	Scope	Is Query	Is Polymorphic
PrintItems	Outputs the items in the fridge	String	<None>	Public	Instance	Yes	No
sendNotifty	Sends notification when item is going to expire	<None>	<None>	Public	Instance	Yes	No

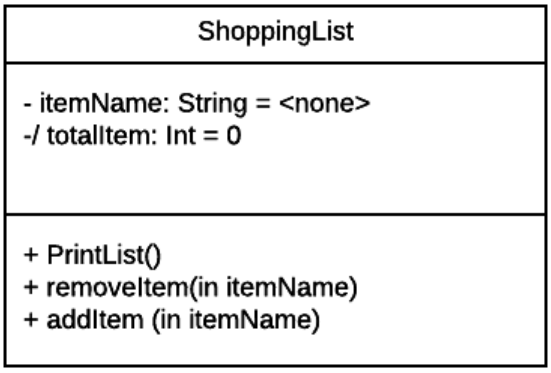
Processing Outlines:

PrintItems:

Displays all the items that are currently in the fridge

sendNotify:

Sends notification to the user when an item is going to expire



Description: Represents a shopping list of items the user wishes buy
Visibility: Public
Is Abstract: No

Attributes:

Name	Description	Data Type	Is Derived	Read Only	Visibility	Multiplicity	Default Value
itemName	Name of item	String	No	Yes	Private	1	<none>
totalItem	Number of items in the fridge	Int	Yes	Yes	Private	1	0

Operations:

Name	Description	Return Type	Parameters	Visibility	Scope	Is Query	Is Polymorphic
PrintList	Outputs the current Items on the list	String	<None>	Public	Instance	Yes	No
removeItem	Removes an item from the list	<None>	itemName	Public	Instance	Yes	No
addItem	Adds an item to the list	<None	itemName	Public	Instance	Yes	No

Processing Outlines:

PrintList

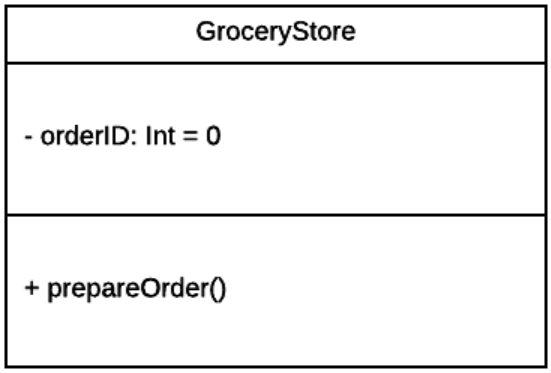
Displays all the items that are currently on the list

addItem:

Adds an item to the list

removeItem:

Removes an item from the list



Description: Represents vendors receiving and prepping the order
Visibility: Public
Is Abstract: No

Attributes:

Name	Description	Data Type	Is Derived	Read Only	Visibility	Multiplicity	Default Value
orderID	Order number of the user	Int	No	Yes	Private	1..*	<none>

Operations:

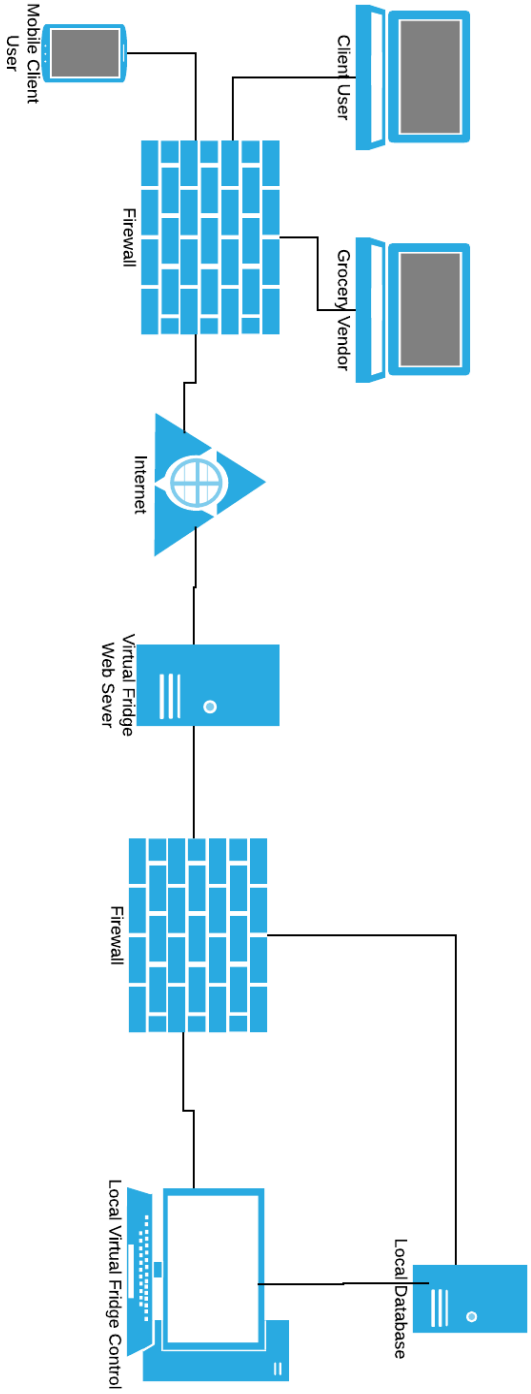
Name	Description	Return Type	Parameters	Visibility	Scope	Is Query	Is Polymorphic
prepareOrder	Grocery store readies the items that are ordered	<None>	<None>	Public	Instance	Yes	No

3.0 Architecture Design

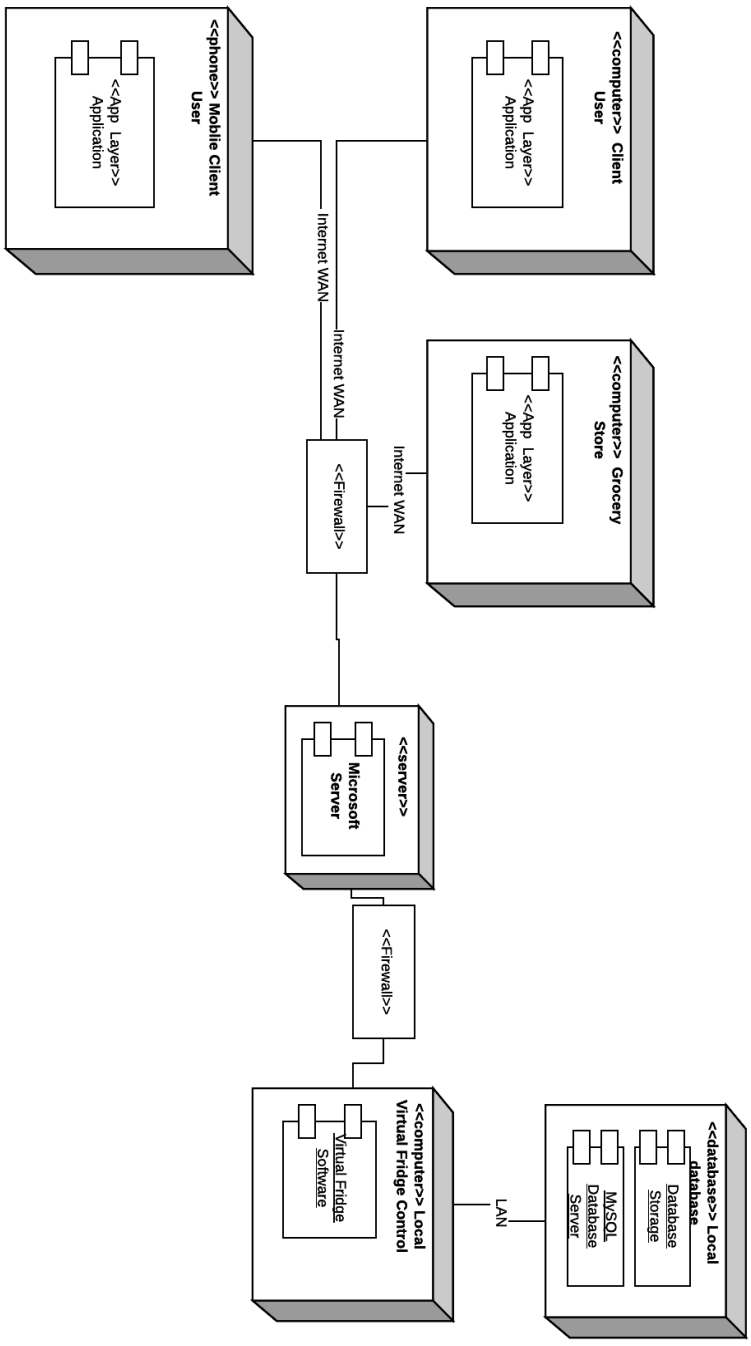
3.1 Introduction

The Architecture Design section is divided into three sections. The first section contains the infrastructure model to provide a graphical representation of the major components of the physical system architecture. The second section describes the hardware and software requirements needed to support the application. The third section is a security plan designed to describe any potential security threats or stated security requirements. Virutal Fridge will be a three-tier client-server application system where the application logic mainly resides within a web server and a local database. The user interface logic is handled by the client's computer or phone. Virtual Fridge will require the purchase of two servers to run.

3.2 Infrastructure Model



Nodes and Artifacts



3.2 Hardware and Software Requirements

The following are a list of hardware and software requirements needed to implement Virtual Fridge. The required hardware components describe what hardware will be required to implement the system's architecture. The required software components describe what software will be required to implement this system

Required Hardware Components

- **Virtual Fridge Server**
  - This server runs Windows Server 2016. The server will host the web interface for Virtual Fridge that users connect to. Ms. Weltz and Virtual Fridge will have the opportunity to purchase this server new or paid for through a web hosting service.
- **Local Cuisine by Car Computer**
  - This machine will be located within the Virtual Fridge premise. It will primarily be used by the owner and management of Virtual. This computer will run the local Virtual Fridge application that communicates with the local database. We recommend buying a new computer with at least Windows 10 for the best efficiency. However, we do have several older existing computers available for use as well.

Required Software Components

- **Windows Server 2016**
  - The system will run on the Windows Web Server to handle hosting the Virtual Fridge web interface and application. If a third party service is desired, purchase of this software is no longer necessary.
- **MySQL Database Server Software**
  - This database will run on the local database onsite. We recommend the most up to date version of this software to ensure maximum efficiency.
- **Windows 10 (or newer)**
  - A version of the Windows operating system to run on the Virtual Fridge computer. Any version of the Windows operating system would work, but we simply recommend the most up-to date version for maximum efficiency and security features.
- **Oracle Database Security Software**
  - To protect valuable information stored in the database and web server, we highly recommend the purchase of Oracle's Security Software. This software helps to ensure data privacy and protect against threats.

3.3 Security Plan

Threats	Disruption, Destruction, Disaster					Unauthorized Access		
Components	Fire	Flood	Power Loss	Circuit Failure	Virus	External Intruder	Internal Intruder	Eavesdrop
Servers	1,2	1	3	1,4	5,6	6,7,9,10	6,8,10	
User Devices	1,2	1	3	1,4	5,7	9,10	6,8,10	
Network Devices	1,2	1	3	1,4	5,7	9,10	5,6,9	
Network Software	1,2	1	3	1,4	5,7	9,10	5, 10	
People	1,2	1	3	1,4	5,7	9,10	5,6,8,9,10	

Controls

1. Disaster recovery plan
2. Fire system in host computer room; sprinklers in rest of building
3. Uninterruptable Power Supply (UPS) on all major network servers
4. Extra backbone fiber cable laid in different conduits between major servers
5. Virus checking software present on the network
6. Extensive user training on password security and reminders in monthly newsletter
7. Application layer firewall
8. Admin authorization to install software/make any system changes
9. All database encrypted
10. Strong password software

4.0 User-Interface

4.1 User-Interface Requirements and Constraints

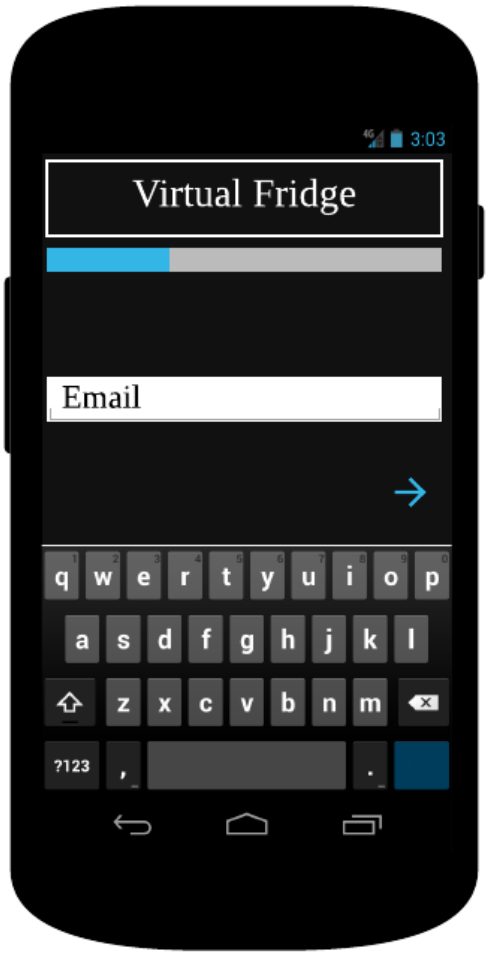
This section will cover the user-interfaces for the Virtual Fridge system. It will provide a general guideline for how the user-interface should be designed. The drawings are purely sketches and does not provide an accurate representation of color, font, style, or graphics. These interfaces show the functionality that Virtual Fridge provides for the users. Since we have designed this application with the mind that most users will not be tech-savvy, the interface should be designed with ease of use in mind. Buttons, text fields, user input, and other components of the interface should be clearly labeled. The following section contains several examples of what the interface for Virtual Fridge may look like. Although there are no printed output, printable sections are mentioned.

4.2 Forms: Screen/User-Interaction Design

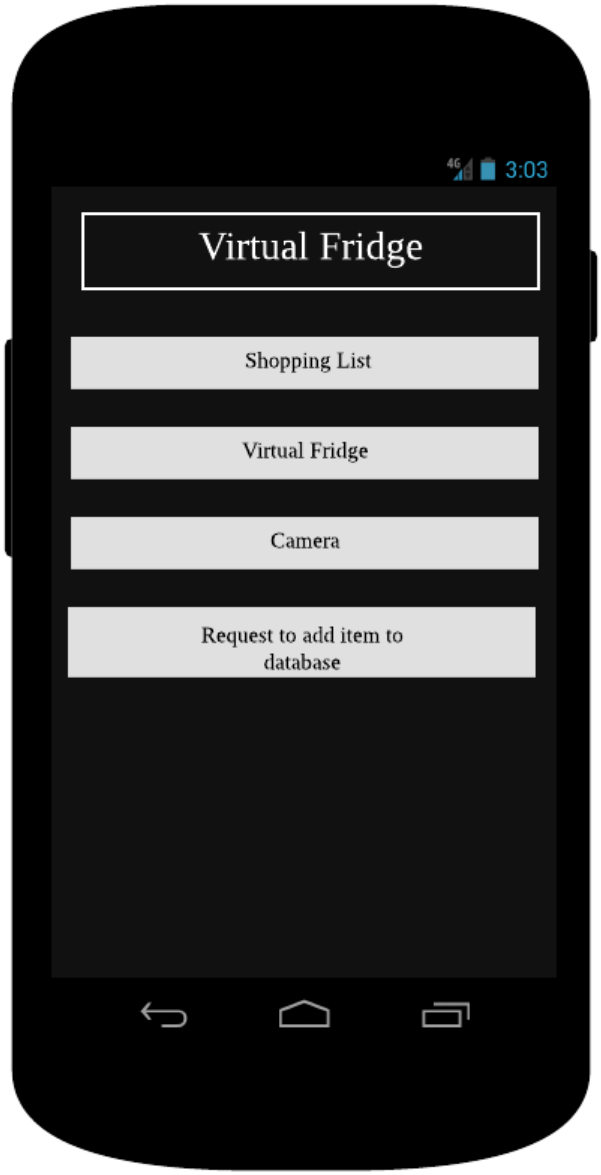
Welcome Screen: This is the home page for the Virtual Fridge application. Basic navigations include Login and Register.

Login Screen: This will be the login screen for Virtual Fridge. Here, the user can sign in or register for an account. The PC version will have the same options, except it will be formatted to a PC browser.

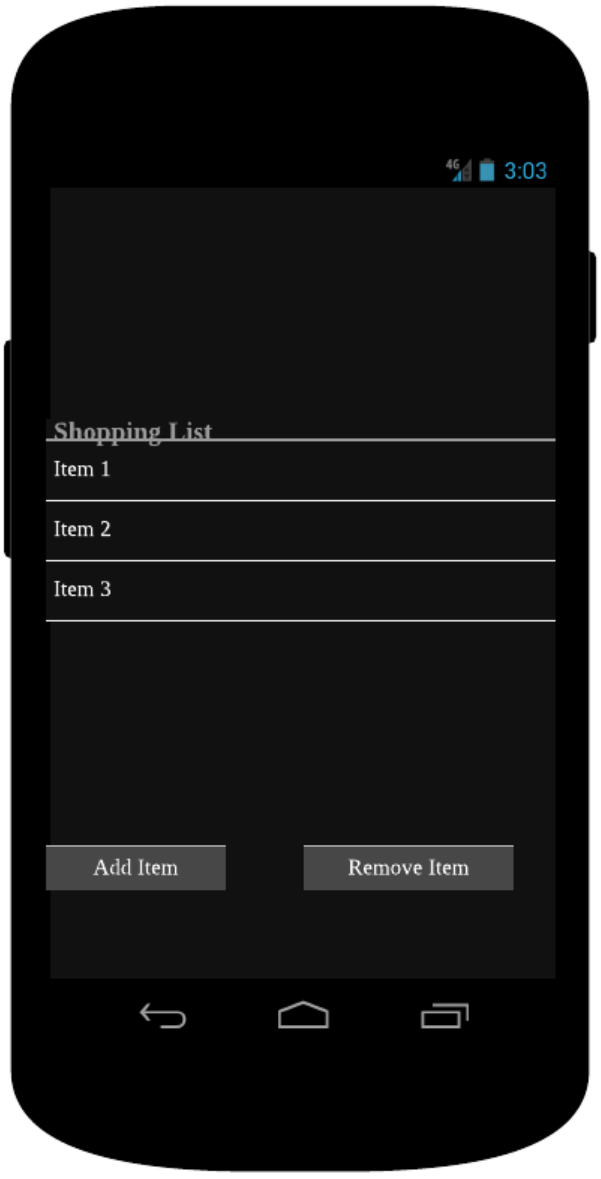
Registration Screen: This will be the page for new user to create an account. Users enter their personal information here. The first screen will prompt the user to enter their name, then they will click next and enter other information, such as email, phone, username, etc, one by one. This is so the keyboard and the text can fit on the screen. Also there will be a progress bar on top showing how much is left to register.



Page after login: Will give the user choices to go to shopping list, go to the items in the virtual fridge, take a picture of an item.



Page after clicking shopping list and Virtual Fridge: Will show list of items as well as buttons to add and remove items. Virtual Fridge will be similar except the expiration dates of the items will be shown.



Bibliography

CSC 3150 Sample Book, CSC 3150. Seattle Pacific University, Seattle, Washington. Spring 2017

Dennis, Alan, Barbara Haley Wixom, and David Tegarden. *System Analysis and Design*. Student Value Edition ed. Hoboken, NJ: John Wiley, 2015. Print.

Lucidchart. November 2016. *Lucid Software Inc.* 2017.

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

Weltz, Elaine. CSC 3150 Systems Design (Spring 2017), multiple lectures (PowerPoint slides, PDF files, Word Documents). Retrieved from Professor Weltz and Seattle Pacific University Canvas.