Garlin Gilchrist II
ME 490 Independent Study Project Summary

AWARE Project Background

In a market economy, industrial production of consumer goods and services is driven by patterns of decision-making in the marketplace; however, individuals make these decisions with little or no ability to rationally gauge the ultimate social and environmental influence of their choices. This project involves the development of a handheld device concept and prototype to provide information to consumers about products and their manufacturers in a quick and simple way, customized, on-demand, and at the point of purchase. The idea is to see how much people might change their purchasing habits if they knew something about the social and environmental impact of different products. The completed prototype will be taken to Washington DC in May for an EPA design competition.

Objectives

It is my goal to create a functioning PDA application that will generally function as follows:

a. take as input UPC barcode from a given product
b. query a database of information that contains information about the social and environmental aspects of the product and is producer(s)
c. display this information on the handheld device in a simple, user-friendly interface that can provide information at various levels of detail in a way that is understandable, practical, and customizable

As a stretch goal, I would also like to explore designing an efficient and effective database that can contain the desired product information, which may be stored either in a web server or possibly on the devices themselves.

I will produce an application for a Windows Powered Pocket PC handheld device. The code will most likely be written using tools such as Microsoft Visual Studio and the C# programming language.

My Specific Contributions

Design/Development Tools Used

The AWARE software was designed, developed, and tested using Microsoft eMbedded Visual Basic 3.0 (eVB), the Socket Communications Software Development Kit (SDK), and Microsoft SQL Server CE 3.0 (SQL CE). I used eVB to design the user interface and write the code behind the interface. The Socket was used to integrate the software module with our Socket Barcode Scanner. The database was design with SQL CE.

Software Goals

The software description can be broken down into three specific areas: user interface, functionality, and database structure.

The goals of the user interface were simplicity and flexibility. The goal was to have the interface be simple enough for an inexperienced user, both of a PDA and the AWARE software project, to be able to pick up the device and quickly understand what they could do to get the information they desired. As a result, screens were laid out in such a way that primary functionality was easily visible and accessible using large buttons and large fonts.

The flexibility of the software is found in that we display different levels of information. Depending on the user, they may care about levels of details in our product/manufacturer rating system. We give users the ability browse these different levels of data.

The database is structured in a way such that independent data is stored in an efficient matter, leading to easily and quickly accessible information. The tables of the database are described in Appendix III.

Functionality

The AWARE software gives the user the ability to use a scanner to accept a barcode as input, and then queries the database to return our stored information on that product and/or its manufacturer. The information we have stored in the database is concerning the environmental impacts of our products, as well as the social and environmental impacts of the product's manufacturer. We translate this rating into a color-coded system that describes the product and manufacturer. These colors are then displayed to the user:

Green	Above Average
Yellow	Average
Orange	Below Average
Red	Poor

We use a picture of a person to represent the social impact, and a flower to show the environmental impact, with each picture being of the appropriate color.

Selecting a specific product takes you to a product information page that shows a more detailed view of the environmental ratings for that particular product. These details were compiled based upon research conducted by another AWARE project team member, Katie Kerfoot, and are described in her report.

Selecting a specific manufacturer takes the user to a manufacturer information page that shows the specific issues that help/hurt that manufacturer's social and environmental ratings.

We also have the ability to scan multiple products so that we can compare both product and manufacturer social and environmental ratings.

Each screen gives users the ability to scan in a new product by pressing a large scan button at the bottom. An alternative method to scan is to press the dedicated hardware button to initiate a scan. Upon success, the scanner sounds a confirmation beep. Once a complete barcode has been scanned, that barcode is then parsed to extract the manufacturer ID (UCCID). This is used to identify manufacturers for queries in our database.

The database was created from data stored in a series of *.xml files containing all of our information on the products, manufacturers, categories, and issues. Different screens have different queries that are used to return the desired information. Here is a description of those queries:

- Return all of the categories that a given manufacturer is a member of
- Return the name, social, and environmental ratings of a manufacturer that is in a given category
- Return product information of a product with a given barcode number
- Return all of the details of the issues for a given manufacturer

These queries are used to generate the information to be displayed on screen to the user.

Phases of Design/Development Process

Research

Before beginning my development of the AWARE prototype, it was necessary for me to research issues relevant to project. With direction from my advisors, I was able to find information on different organizations who housed the types of data that we would be distributing via our application. I also learned about the technology behind barcodes, how they were constructed, and that they really meant. Finally, I did research on how a Pocket PC device can be used to scan barcodes. I examined existing applications for ideas on how our prototype should look and feel, and also came across many tips and insights on the best ways to approach programming an application on a mobile device.

Design

The first step of my design process was to gather requirements for this software prototype. These requirements were collected in 3 primary ways: talking to the project advisors about their vision, personally conducted informal surveys with family and friends, and my own brainstorming. This led to the drafting of my Software Requirements Specification (SRS), which is in the appendix of this document. An SRS contains no technical details, but gives direction to the design and development processes in terms of what we wanted to accomplish on a high level.

After refining this document at our weekly meetings, I next created a Software Design Specification (SDS), which is in the appendix of this document. This is a technical document that describes the software in much more detail than the

SRS. This helped guide how I actually went about writing the code for the AWARE software module. It most importantly describes how the different pieces of the software are to interact with one another: the barcode scanner, the PDA application, and the information database.

Development

My development process focused on building functioning, independent modules that could then be integrated with the assurance that individual components functioned properly. My method can be characterized as “modular programming” or “rapid prototyping.” Priority was given to building functionality before focusing on the “attractiveness” of the program. This functionality primarily included proper interaction between all of the application’s screens and integration of the barcode scanner into the application. At each weekly meeting, I showed the current prototype and received feedback on what was good and what needed to be changed and/or added.

Testing

The “Rapid Prototyping” programming model is test-driven. In order to move on to the next prototype, or to build on top of an existing one, functionality of the current module must be tested and verified. As such, the code has been tested and verified since the beginning of development. Due to this more efficient model, there is less need for an extended “testing” period somewhere near the end of the design process. However, as a result, the development process is longer.

Learning Experiences

This project allowed me two major learning experiences: a chance to look at the software design process from conception of the idea to functional prototype, and gaining familiarity with how products and manufacturers are compared on social and environmental responsibility.

This was a great experience for me to completely and independently design, develop and test software for a mobile device. It was also great for me to learn how to work with the barcode scanner peripheral. This experience gave me my first chance to use a 3rd party SDK to write code for a removable device.

Finally, I was completely unfamiliar with the sorts of information that was gathered on our different products and manufacturers. Sifting through this information gave me the opportunity to understand the different metrics used to compare products and manufacturers.

Appendices

I. Software Requirements Specification
II. Software Design Specification
III. Database Structure
IV. XML Files

V. Software Code
VI. Screen Shots

Appendix I: Software Requirements Specification

AWARE Project
Requirements Doc
Last update: 16 Feb 2005

Purpose

The product to be developed is a system to personalize information about products bought in a supermarket. The vast amount of information available in the modern world and the increasing number of medical or environmental concerns of shoppers have made shopping increasingly confusing with many complex decisions to be made. The product developed here aims to give fast, direct access to information tailored for and by the individual. The sub-product chosen for development is called ‘Aware.’ It is a module for handheld PDA, and incorporates software enabling the user to set a profile and to scan barcodes of products.

Requirements

Primary Functionality

Software Module

The key functions of the Aware software module are as follows:

- Allow user to create profiles
- Accept barcode information
- Display customizable product information

The user shall be able to input barcode information, ultimately a numeric value, into the [the device and] Aware software module by one of two methods:

1. Barcode scan
2. Alphanumeric entry of UPC barcode number

Regardless of the chosen method of data entry, the software module shall retrieve the desired product information from the external database, and relevant information on it shall then be delivered.

The user shall be able to use the PDA interface to create a profile containing his or her preferences with regards to different characteristics of products, possibly including (but in no way limited to):

- Emission information
- Whether the product was animal tested
- Whether the product was produced in a sweat shop
- Whether the product is organic
- Etc.

The specific characteristics will be dictated by the information available from the database. Whatever the categories, the user shall have the ability to choose which categories are available for display.

Database

The database shall contain information about products, product manufacturers, and product characteristics which will be used to measure the rating of a given product. The data should include, but not be limited to:

- Manufacturer
  - Name
  - Location
  - Description
  - Rating
- Product
  - Name
  - Description
  - Manufacturer
  - Ingredients/components
  - Rating
- Characteristics (e.g. ResponsibleShopper.org)
  - Energy consumption
  - Packaging
  - Recycling
  - Nuclear energy
  - Etc.

This database will be queried appropriately to acquire the desired information on the product in question.

Secondary Functionality

Software Module & Database

The user shall be able to browse a list of products by category. These categories can include (but are in no way limited to):

- Dairy
- Meat
- Clothing
- Outerwear
- Tools
- Etc.

The user shall be able to select products from the various categories and display information about the chosen product. The information displayed shall be customizable, as is the information displayed after barcode input.

Attributes and Non-functional Requirements

- **Data integrity**

  In the event of an error accessing the external product information database, the Aware software module will not partially transmit the possibly incomplete data. The Aware software module will only accept complete sets of data from the external database. In the error case, appropriate error messages will be displayed.

- **Quality**
  The Aware software module shall not crash or in any other way disrupt the normal operation of its host device at a rate higher than 1%.
- **Performance**
  Performance is a key consideration for this product. It is our goal to provide data access that is very fast. More concretely, we want the time from which the user scans a barcode to the time product information is displayed on the device to be no longer than 2-3 seconds.
- **Usability**
  Many of the usability concerns with the Aware software module will fall under the category of how quickly accessible the commands and data are. “Quickly accessible” can be measured the amount of input needed to access a given piece of information. Input here means:
  - Barcode scan
  - Screen tap
  - Pressing a button
  - Entry of alphanumeric input

Technology specification
The prototype for the Aware software module will be written for Windows Pocket PC powered handheld devices that are outfitted with barcode scanners (built-in or as an external peripheral).

Interaction with other systems
The Aware software module shall interact primarily interact with some database containing relevant product information. This database may be internal to the device, or a remote information store.

Appendix II: Software Design Specification

AWARE Project
Specification Doc
Last update: 16 Feb 2005

Contents

1. Introduction

1.1. The purpose of this document is to fully specify the Aware software module.

1.2. This document will cover the architecture, graphical user interface, and code design. It will not define specific lines of code to be written, nor will it define methods or plans of how the module is to be tested.

1.3. This document is intended for the developer of this and subsequent Aware software modules.

1.4. References & Related Documents

1.4.1. Aware Software Requirements Document

1.5. Important terms

1.5.1. PDA = “Personal Digital Assistant,” which is essentially a handheld computer.

1.6. Overview of document

1.6.1. The rest of this SDS is broken down into the system architecture description, which consists of an overview of the modules and components as well as the structure and relationships of the modules. The detailed description of each component includes the following headings: identification, type, purpose, function, subordinates, dependencies, interfaces, resources, processing, and data.

2. System Overview, Structure, and Relationships

2.1. The software can be broken down into the following modules: a PDA interface module, Barcode Scanner module, and a Database module, which are all described later in this document. The basic use case will involve a user using their PDA equipped with a barcode scanner to scan the UPC barcode of their desired product. The PDA application will then generate a query to the product database based on that UPC barcode input. The query will return to the PDA application the desired information, which is then displayed for the user.

2.2. …

3. Design Considerations

3.1. Assumptions and Dependencies

3.1.1. Related Software

3.1.1.1. We will write these software modules to run on the Windows Mobile Operating System. The current version is Windows Mobile 2003, Second Edition.

3.1.2. Related Hardware

3.1.2.1. We will be running our software on Windows Mobile 2003-compatible Pocket PC devices

3.1.2.2. The barcode scanner we will use is the SDIO In-Hand Scan Card by Socket.

3.2. General Constraints

3.2.1. Interface/protocol requirements

3.2.1.1. …

3.2.2. Memory capacity

3.2.2.1. …

3.2.3. Performance requirements

3.2.3.1. …

3.3. Goals and Guidelines

3.3.1. A major design goal is for the PDA application user interface to be as simple and uncluttered as possible. This is to ensure quick access to only the necessary information that the users specify that they desire.

3.4. Development Methods

3.4.1. The development method has yet to be finalized. The candidates are Test Driven Development and Rapid Prototyping.

4. Architectural Strategies

4.1. Platform

4.1.1. We made a choice between using the Windows Mobile Platform and the Palm OS platform which each had pros and cons.

4.1.1.1. Palm OS

4.1.1.1.1. Less expensive hardware

4.1.1.1.2. More prevalent in consumer market

4.1.1.2. Windows Mobile

4.1.1.2.1. More prevalent in business market

4.1.1.2.2. Familiar interface (similar to desktop computers)

4.1.1.2.3. Developer familiarity with platform

4.1.2. The decision was made to go with the Windows Mobile Platform based largely on developer familiarity with the platform.

4.2. Data Storage and Management

4.2.1. For the prototype, we will create a small, local database of product information that will be queried by the PDA application.

4.2.2. Ultimately, a remote database will be used: either one we design, build, and populate ourselves, or an already existing database (e.g. www.ResponsibleShopper.org)

4.3. Communication Mechanisms

4.3.1. …

4.4. Future Plans for Extending Software

4.4.1. …

5. System Architecture

5.1. This system has three major modules, each described below. The selection of this particular breakdown is based on a logical abstraction on how data will flow throughout the software module as a whole. A high level summary of their interaction can be seen in Figure 1.

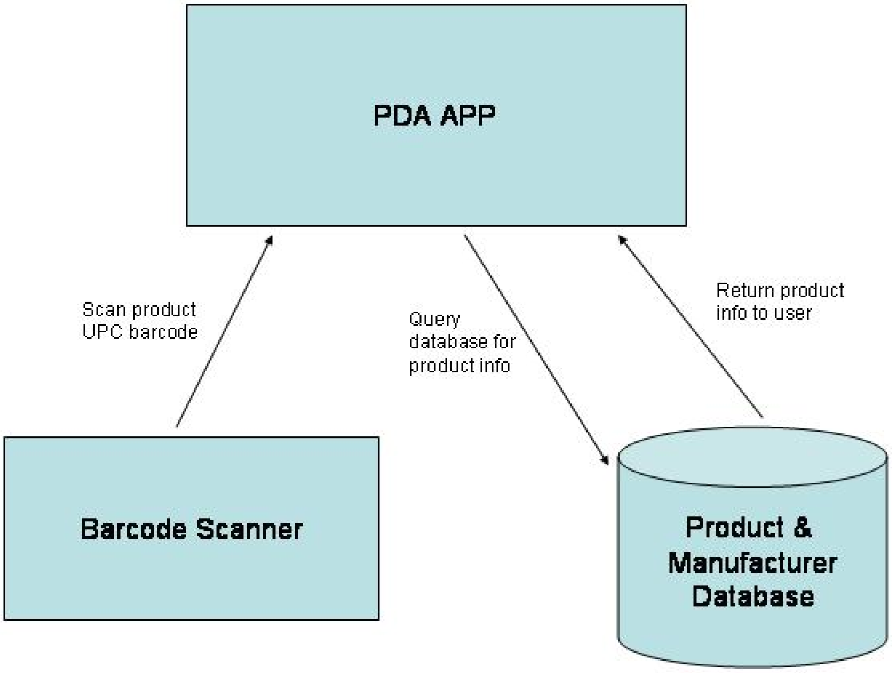
PDA APP
Scan product UPC barcode
Query database for product info
Return product info to user
Barcode Scanner
Product & Manufacturer Database

Figure 1: High Level Interaction between Software Modules

5.2. PDA Application

5.2.1. Major Responsibilities

5.2.1.1. The PDA application is the graphical front end of the Aware software module; it is the part that the user will interact with directly.

5.2.1.2. The PDA acts as a connecting "hub" between the barcode scanner and the product & manufacturer database.

5.2.2. Collaboration with other modules

5.2.2.1. The PDA application will receive input from the user and from the barcode scanner.

5.2.2.2. The PDA application will build queries to be executed against the database

5.2.2.3. The PDA application will receive data from the database and display it to the user.

5.3. Barcode Scanner

5.3.1. Major Responsibilities

5.3.1.1. This is the input device used to gather product identification information

5.3.2. Collaboration with other modules

5.3.2.1. UPC Barcode data input into the scanner will be exported to the PDA application.

5.4. Product & Manufacturer Database

5.4.1. Major Responsibilities

5.4.1.1. All relevant product and manufacturer information is stored in this local/remote database.

5.4.1.2. Collaboration with other modules

5.4.1.2.1. Queries will be accepted from the PDA application to the database.

5.4.1.2.2. Query results will be returned to the PDA application.

6. Detailed System Design

6.1. PDA Application

6.1.1. Classification

6.1.1.1. This is the user interface and main connecting hub of the module, and it will be encapsulated into a class

6.1.2. Definition

6.1.2.1. The specific purpose of this component is to accept input from the user or the barcode scanner, query data from the database, and format and display data to the user.

6.1.3. Responsibilities

6.1.3.1. The responsibility of the PDA application is to provide an easy-to-use interface for the barcode scanner and product information database to the user.

6.1.3.2. The PDA application plays the role of manager over the barcode scanner and database.

6.1.4. Constraints

6.1.4.1. …

6.1.5. Subcomponent(s)

6.1.5.1. …

6.1.6. Uses/Interactions

6.1.6.1. Input to this component will be taken from the user, the barcode scanner, and the database

6.1.6.2. Output from this component will go to the database and to the user

6.1.7. Resources

6.1.7.1. …

6.1.8. Processing

6.1.8.1. …

6.1.9. Interfaces/Exports

6.1.9.1. PDAApp Object

6.1.9.1.1. Encapsulate PDA application operations

6.1.9.1.2. Properties

6.1.9.1.2.1. BarcodeScanner barcodeScanner

6.1.9.1.2.1.1. Description

6.1.9.1.2.1.1.1. An object that handles the barcode scanner interaction

6.1.9.1.2.1.1.2. Described in more detail below

6.1.9.1.2.1.2. Initial Value

6.1.9.1.2.1.2.1. NULL

6.1.9.1.2.2. PMDB pmDB

6.1.9.1.2.2.1. Description

6.1.9.1.2.2.1.1. An object that handles the database interaction

6.1.9.1.2.2.1.2. Ddescribed in more detail below.

6.1.9.1.2.2.2. Initial Value

6.1.9.1.2.2.2.1. NULL

6.1.9.1.2.3. …

6.1.9.1.2.3.1. Description

6.1.9.1.2.3.2. Initial Value

6.1.9.1.3. Methods

6.1.9.1.3.1. string scanBarcode (BarcodeScanner barcodeScanner)

6.1.9.1.3.1.1. Description

6.1.9.1.3.1.1.1. This function will use the barcode scanner to scan for input

6.1.9.1.3.1.2. Parameters

6.1.9.1.3.1.2.1. BarcodeScanner barcodeScanner

6.1.9.1.3.1.3. Return Value

6.1.9.1.3.1.3.1. A string representing the barcode data as read in by the scanner

6.1.9.1.3.2. string cleanupBarcodeString (string inputBarcodeString)

6.1.9.1.3.2.1. Description

6.1.9.1.3.2.1.1. A helper function that will transform the barcode data into a useful format

6.1.9.1.3.2.2. Parameters

6.1.9.1.3.2.2.1. string inputBarcodeString

6.1.9.1.3.2.3. Return Value

6.1.9.1.3.2.3.1. A string representing a the barcode input in usable form

6.1.9.1.3.3. string queryDB (PMDB pmDB)

6.1.9.1.3.3.1. Description

6.1.9.1.3.3.2. Parameters

6.1.9.1.3.3.2.1. PMDB pmDB

6.1.9.1.3.3.3. Return Value

6.1.9.1.3.4. string parseQueryResult (string queryResult)

6.1.9.1.3.4.1. Description

6.1.9.1.3.4.2. Parameters

6.1.9.1.3.4.2.1. string queryResult

6.1.9.1.3.4.3. Return Value

6.1.9.1.3.5. …

6.1.9.1.3.5.1. Description

6.1.9.1.3.5.2. Parameters

6.1.9.1.3.5.3. Return Value

6.1.9.1.3.6. …

6.2. Barcode Scanner

6.2.1. Classification

6.2.1.1. This is an input device that will be encapsulated into an object

6.2.2. Definition

6.2.2.1. The specific purpose of this component is to provide the information by which we will identify individual products.

6.2.3. Responsibilities

6.2.3.1. The responsibility and role of the scanner module is to provide identification data.

6.2.4. Constraints

6.2.4.1. …

6.2.5. Subcomponent(s)

6.2.5.1. This component has not subcomponents.

6.2.6. Uses/Interactions

6.2.6.1. Output from this component will be taken and used to build a query to the database by the PDA application.

6.2.7. Resources

6.2.7.1. Barcode scanner hardware (e.g. SDIO In-Hand Scan Card by Socket)

6.2.8. Processing

6.2.8.1. …

6.2.9. Interfaces/Exports

6.2.9.1. BarcodeScanner Object

6.2.9.1.1. Description

6.2.9.1.1.1. A class that will encapsulate all of the barcode scanner functionality

6.2.9.1.2. Properties

6.2.9.1.2.1. string barcodeData

6.2.9.1.2.1.1. Initial Value

6.2.9.1.2.1.1.1. NULL

6.2.9.1.2.1.2. Decription

6.2.9.1.2.1.2.1. A string containing the result of a call to the getBarcode () method.

6.2.9.1.3. Methods

6.2.9.1.3.1. string getBarcode ()

6.2.9.1.3.1.1. Description

6.2.9.1.3.1.1.1. A function used to accept the input from the barcode scanner.

6.2.9.1.3.1.2. Parameters

6.2.9.1.3.1.2.1. None

6.2.9.1.3.1.3. Return Value

6.2.9.1.3.1.3.1. A string containing the barcode data as read by the scanner

6.3. Product & Manufacturer Database

6.3.1. Classification

6.3.1.1. This is a database that will be encapsulated into an object that will inherit from the GeneralDB object.

6.3.2. Definition

6.3.2.1. The specific purpose of this component is to hold information concerning the environmental and social impacts of products and product manufacturers.

6.3.3. Responsibilities

6.3.3.1. The database needs to persistently available to be queried by the PDA application for product/manufacturer information.

6.3.3.2. The database’s role is to provide read-only information.

6.3.4. Constraints

6.3.4.1. The database should to be accessible only by the PDA application.

6.3.5. Subcomponent(s)

6.3.5.1. GeneralDB Object

6.3.5.1.1. Description

6.3.5.1.1.1. A general class providing basic database manipulation functionality.

6.3.5.1.2. Properties

6.3.5.1.2.1. string name

6.3.5.1.2.1.1. Description

6.3.5.1.2.1.1.1. The name of the current database to be worked with

6.3.5.1.2.1.2. Initial Value

6.3.5.1.2.1.2.1. NULL

6.3.5.1.3. Methods

6.3.5.1.3.1. void open (string databaseName)

6.3.5.1.3.1.1. Description

6.3.5.1.3.1.1.1. Opens a database with the name databaseName

6.3.5.1.3.1.1.2. Sets name property to databaseName

6.3.5.1.3.1.2. Parameters

6.3.5.1.3.1.2.1. string databaseName

6.3.5.1.3.1.3. Return Value

6.3.5.1.3.1.3.1. none

6.3.5.1.3.2. void close (string databaseName)

6.3.5.1.3.2.1. Description

6.3.5.1.3.2.1.1. Closes a database with the name databaseName

6.3.5.1.3.2.2. Parameters

6.3.5.1.3.2.2.1. string databaseName

6.3.5.1.3.2.3. Return Value

6.3.5.1.3.2.3.1. none

6.3.5.1.3.3. string query (string databaseName, string queryString)

6.3.5.1.3.3.1. Description

6.3.5.1.3.3.1.1. Submit a query queryString to the database named databaseName

6.3.5.1.3.3.2. Parameters

6.3.5.1.3.3.2.1. string databaseName

6.3.5.1.3.3.2.2. string queryString

6.3.5.1.3.3.3. Return Value

6.3.5.1.3.3.3.1. A string containing the result of the query queryString

6.3.6. Uses/Interactions

6.3.6.1. Input to this component will be taken from the PDA application in the form of a query to the database.

6.3.6.2. Output from this component will be taken and used to display information to the user by the PDA application.

6.3.7. Resources

6.3.7.1. …

6.3.8. Processing

6.3.8.1. The database will accept SQL queries to perform its responsibilities.

6.3.9. Interfaces/Exports

6.3.9.1. PMDB Object

6.3.9.1.1. Inherits functionality from the GeneralDB object

6.3.9.1.2. Properties

6.3.9.1.2.1. List tables

6.3.9.1.2.1.1. Description

6.3.9.1.2.1.1.1. List containing the tables in the Product & Manufacturer Database

6.3.9.1.2.1.2. Initial Value

6.3.9.1.2.1.2.1. TBD

6.3.9.1.2.2. …

6.3.9.1.2.2.1. Description

6.3.9.1.2.2.2. Initial Value

6.3.9.1.3. Methods

6.3.9.1.3.1. …

6.3.9.1.3.1.1. Description

6.3.9.1.3.1.2. Parameters

6.3.9.1.3.1.3. Return Value

6.4.

7. Policies and Tactics

7.1. *policy/tactic-1 name or description*

7.2. *policy/tactic-2 name or description*

7.3. ...

8. Glossary

9. Bibliography

*Format from: http://www.construx.com/survivalguide/desspec.htm and example from http://www.cs.utexas.edu/users/s2s/latest/proplog1/doc/SDS.html#modules.

Appendix III: Database Structure

Manufacturer table

```
' CREATE TABLE Manufacturer
   '   (
   '   ManID int IDENTITY PRIMARY KEY,
   '   Name nvarchar(50),
   '   Location nvarchar(50),
   '   Description nvarchar(100),
   '   UCCID bigint,
   '   category1 nvarchar(50),
   '   category2 nvarchar(50),
   '   category3 nvarchar(50),
   '   category4 nvarchar(50),
   '   category5 nvarchar(50),
   '   category6 nvarchar(50),
   '   category7 nvarchar(50),
   '   category8 nvarchar(50),
   '   category9 nvarchar(50),
   '   issue1 nvarchar(50),
   '   issue2 nvarchar(50),
   '   issue3 nvarchar(50),
   '   issue4 nvarchar(50),
   '   SocialRating float,
   '   EnvironmentalRating float
   '   );
```

Product table

```
' CREATE TABLE Product
   '   (
   '   ProdID int IDENTITY PRIMARY KEY,
   '   Name nvarchar(50),
   '   Description nvarchar(100),
   '   Brand nvarchar(50),
   '   ModelNumber bigint
   '   Manufacturer nvarchar(50),
   '   UPCA bigint,
   '   UPCE bigint,
   '   EnvironmentalRating float
   '   );
```

Issues table

```
' CREATE TABLE Issues
   '   (
   '   IssueID int IDENTITY PRIMARY KEY,
   '   Name nvarchar(50),
   '   Manufacturer nvarchar(25),
   '   Description nvarchar(1000),
   '   SocialOrEnviromental integer,
   '   CONSTRAINT i_thiskey PRIMARY KEY (Name, Manufacturer),
   '   CONSTRAINT i_manufacturerkey FOREIGN KEY (Manufacturer)
REFERENCES Manufacturer (Name) ON DELETE CASCADE ON UPDATE
CASCADE
   '   );
```