Software Design Document

Consumer Behavior Simulation

Philip A Grim II

Contents

1. Introduction....4

1.1. Purpose....4
1.2. Intended Audience....4
1.3. References....4
2. System Overview....4

2.1. Purpose....4
2.2. Functionality....4
3. System Architecture....4

4. Data Design....6

4.1. Data Classes....6
4.2. Data Dictionary....7
5. Component Design....10

5.1. User Interaction Components....10
5.1.1. Generate a Population (REQ-1)....10
5.1.2. Create an Inventory (REQ-2)....11
5.1.3. Create and Execute a Simulation (REQ-3)....12
5.2. System Components....13
5.2.1. Generate a Population (REQ-1)....13
5.2.2. Execute a Simulation (REQ-3)....14
6. Human Interface Design....16

6.1. Basic Interface View....16
6.2. Add Simulation Button to Accordion Pane (REQ-3.2.1)....17
6.3. Add New Market Simulation Button (REQ-3.2.2)....18
6.4. Add Population (REQ-1.2.1) and Inventory (REQ-2.2.1) to Entities Drop-Down Menu....19
6.5. Add New Population Screen (REQ-1.2.2)....20
6.6. Population Properties Screen (REQ-1.2.3)....21
6.7. Population Screen (REQ-1.2.4)....22
6.8. New Inventory Screen (REQ-2.2.2)....23
6.9. Inventory Management Screen (REQ-2.2.3)....24
6.10. New Market Simulation Screen (REQ-3.2.3)....25
6.11. Simulation Execution Screen (REQ-3.2.4)....27
Appendix A: Requirements Matrix....28

Appendix B: Glossary....32

Figure 1: Consumer Behavior Simulation Architecture....5
Figure 2: Consumer Behavior Simulation ERD....6
Figure 3: Data Dictionary....9
Figure 4: Generate Population User Action Process Flow....10
Figure 5: Create Inventory User Action Process Flow....11
Figure 6: Create and Execute Simulation User Action Process Flow....12

Figure 7: Generate Population System Action Process Flow 14
Figure 8: Execute Simulation System Action Process Flow 15
Figure 9: Basic GUI 16
Figure 10: Main Screen with Simulation Button 17
Figure 11: Main Display Simulation Pane Open 18
Figure 12: Entity Selection drop-down menu 19
Figure 13: New Population Screen 20
Figure 14: Population Properties Screen 21
Figure 15: Population Screen 22
Figure 16: New Inventory Screen 23
Figure 17: Inventory Management Screen 24
Figure 18: New Market Simulation screen 26
Figure 19: Simulation Execution Screen 27

1. Introduction

1.1. Purpose

This document specifies the software design for the Consumer Behavior Simulation software system. This software is a testing tool that generates simulated marketing data for the purpose of providing input to various ETL, database, and analysis systems.

1.2. Intended Audience

This document is intended for reading by the users of the generated test data. This includes project managers, software developers, test engineers, and data analysts who either contribute to the development of ETL and storage software, or who develop analytics that use the test data the simulation generates.

1.3. References

- Consumer Behavior Simulation System Requirements Specification dated 11 Apr 2014

2. System Overview

2.1. Purpose

The purpose of this product is to create a simulation of consumer behavior and provide the results as input data to Big Data ETL systems and analytics. It will involve creating a data model, several random generators, user interface components, and a simulation module.

2.2. Functionality

The Consumer Behavior Simulation will allow the user to create a simulation of consumer purchasing, the output of which can be used to test analytics such as a coupon recommendation engine. To create the simulation, the user will generate a simulated population of customers. The size, gender distribution, and age distribution of the population will be able to be specified. The resulting list of customers will be able to be edited and saved using controls in the user interface. The user will also create a product inventory using the graphical user interface.

A new simulation will be created using the GUI, and configured with a Population, an Inventory, and a start and end date. An output folder will also be specified for the transaction data. The simulation will be executed through the GUI or at the command line using a saved simulation configuration. The user will able to interrupt the simulation in progress.

3. System Architecture

The software will be a single user application intended for use on a standard workstation. The software will be a pure Java implementation, and will work on any operating system that supports a standard Java Virtual Machine (JVM). The software is deployed as an executable Java Archive (jar) file accompanied by Linux shell scripts and Windows batch files to configure and execute the software either in Graphical User Interface mode or command line mode. The deployment also includes supporting libraries, default configuration files, sample data, and documentation.

The following diagram depicts the software architecture and the modules that will be created by this project.

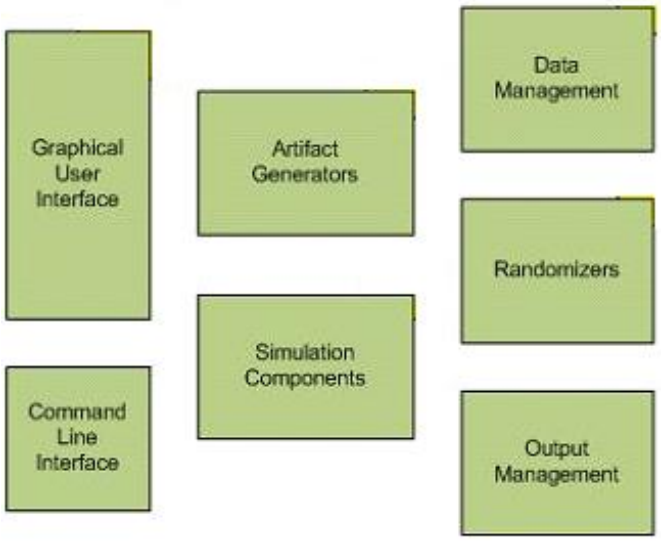

Figure 1: Consumer Behavior Simulation Architecture

4. Data Design

4.1. Data Classes

This update requires a number of data classes. The following diagram depicts the data objects and relationships.

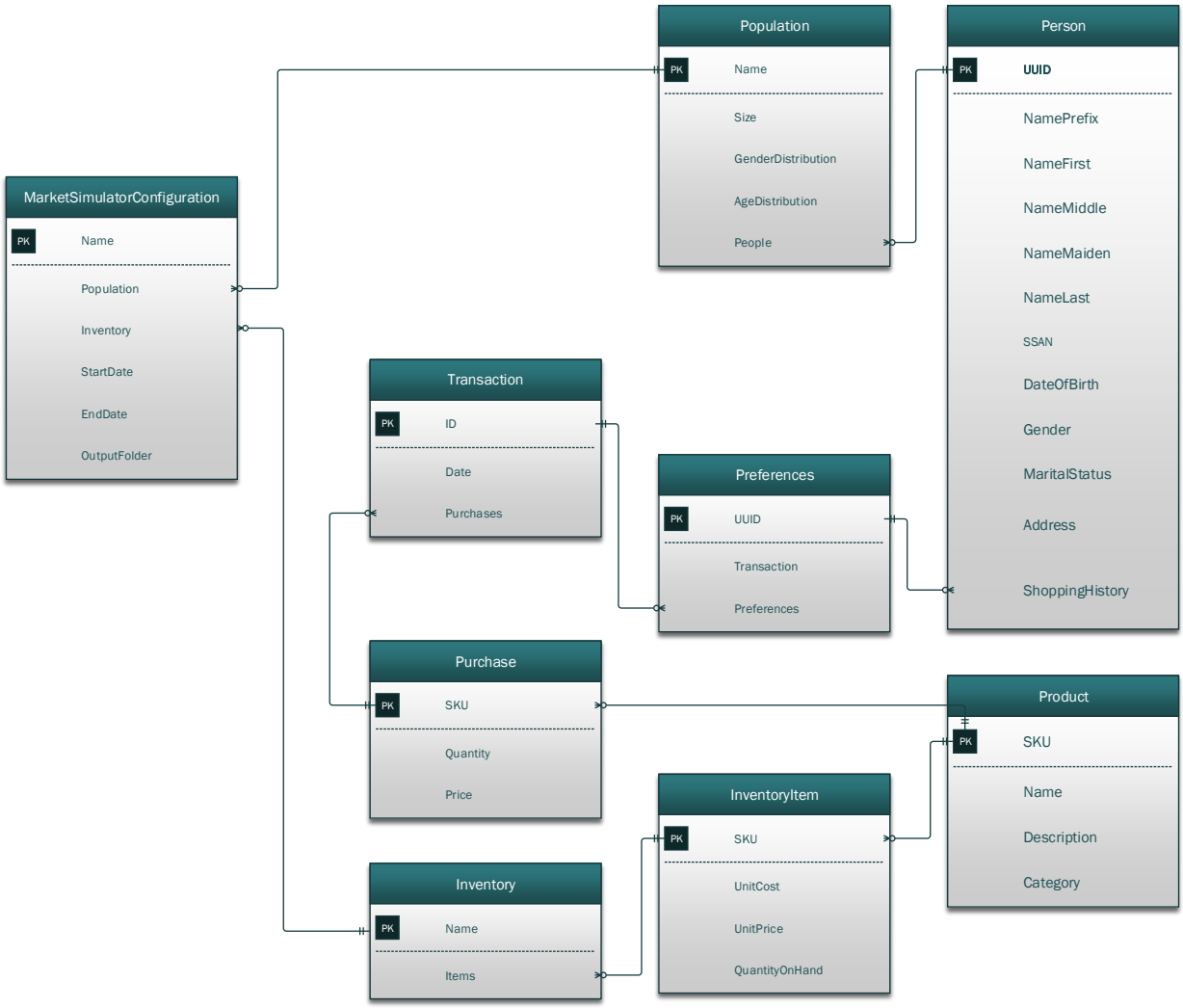

Figure 2: Consumer Behavior Simulation ERD

4.2. Data Dictionary

The following table details the data classes, attributes, and relationship of the new data classes.

Entity	Attribute	Data Type	Description	Requirement
Person		Java Class	Class for storing data pertaining to a person entity.	REQ-1.1.2
	NamePrefix	String	Person's name prefix, e.g. Mr., Mrs., Miss	REQ-1.1.2
	NameFirst	String	Person's First Name	REQ-1.1.2
	NameMiddle	String	Person's Middle Name	REQ-1.1.2
	NameMaiden	String	Person's Maiden Name	REQ-1.1.2
	NameLast	String	Person's Last Name	REQ-1.1.2
	SSAN	String	Person's Social Security Number	REQ-1.1.2
	DateOfBirth	Date	Person's Date of Birth	REQ-1.1.2
	Gender	String	Person's Gender	REQ-1.1.2
	MaritalStatus	String	Person's marital status	REQ-1.1.2
	Address	String	Person's mailing address	REQ-1.1.2
	ShoppingHistory	Preference	Contains the history and shopping preferences for the person.	REQ-1.1.2.1
Preference		Java Class	Stores transaction history and preferences for products.	REQ-1.1.4
	UUID	UUID	Unique identifier, relates to Person	
	History	List<Transaction>	Shopping history	REQ-1.1.4.1
	Preferences	HashMap<Product,Double>	Map of product to calculated preference	REQ-1.1.4.1
Population		Java Class	Container for a generated population	REQ-1.1.1
	Name	String	Unique Name	REQ-1.1.1.1
	Size	long	Size of population	REQ-1.1.1.2
	GenderDistribution	double[2]	Percentage of females and males	REQ-1.1.1.3
	AgeDistribution	double[3]	Min, max, and mean age.	REQ-1.1.1.4
	People	List<Person>	List of people	REQ-1.1.1.5

Product		Java Class	A product that can be purchased	REQ-2.1.2
	SKU	String	Stock keeping unit	REQ-2.1.2.2
	Name	String	Name of product	REQ-2.1.2.1
	Description	String	Description of product	REQ-2.1.2.4
	Category	String	Product category	REQ-2.1.2.3
InventoryItem		Java Class	Line item for an inventory list	REQ-2.1.3
	Product	Product	Reference to a product	REQ-2.1.3.1
	UnitCost	double	The wholesale cost of the product.	REQ-2.1.3.2
	UnitPrice	double	The retail price of the product	REQ-2.1.3.3
	QuantityOnHand	long	Number of the item in the inventory	REQ-2.1.3.4
Inventory		Java Class	Container for a product inventory	REQ-2.1.1
	Name	String	Unique name for the inventory	REQ-2.1.1.1
	Items	List<InventoryItem>	List of items in the inventory	REQ-2.1.1.2
Purchase		Java Class	Record of a product purchase	REQ-3.1.4
	Product	Product	The product purchased	REQ-3.1.4.1
	Quantity	long	Number purchased	REQ-3.1.4.2
	Price	double	The product price	REQ-3.1.4.3
Transaction		Java Class	A record of a shopping transaction	REQ-3.1.3
	Timestamp	Date	Time of the transaction	REQ-3.1.3.1
	ID	UUID	Unique transaction identifier	REQ-3.1.3.4
	Customer	Person	The person performing the transaction	REQ-3.1.3.2
	Purchases	List<Purchase>	The purchases that make up the transaction	REQ-3.1.3.3
Market Simulation Configuration		Java Class	Configuration object to set up a simulation	REQ-3.1.1
	Population	Population	The population used for the simulation	REQ-3.1.1.1
	Inventory	Inventory	The inventory used for the simulation	REQ-3.1.1.2
	Name	String	The name of the	REQ-3.1.1.3

			simulation run	
	StartDate	Date	The simulation clock start date	REQ-3.1.1.4
	EndDate	Date	The simulation clock end date	REQ-3.1.1.5
	OutputFolder	File	Where to write the simulation output	REQ-3.1.1.6

Figure 3: Data Dictionary

5. Component Design

This section illustrates the design and flow of the various user interactions and system processes for the Consumer Behavior Simulation functionality.

5.1. User Interaction Components

The Consumer Behavior Simulation functionality consists of three user processes: generating a population, creating an inventory, and creating and executing a simulation run. The following sections show the flow of each of these processes in detail.

5.1.1. Generate a Population (REQ-1)

The users generates a population by selecting the Population item from the New Entity drop-down menu. The New Population screen appears. The user fills in a unique name for the population and selects a data file to save the population in, then presses the Next button. The New Population Properties screen appears. The user fills in a size for the population, adjusts the gender ratio, adjusts the age distribution, and presses the Next button. The system generates a population, then the Population screen appears. The population is displayed in tabular form. The user may edit the properties of the people in the population, then press the Save button to save the changes.

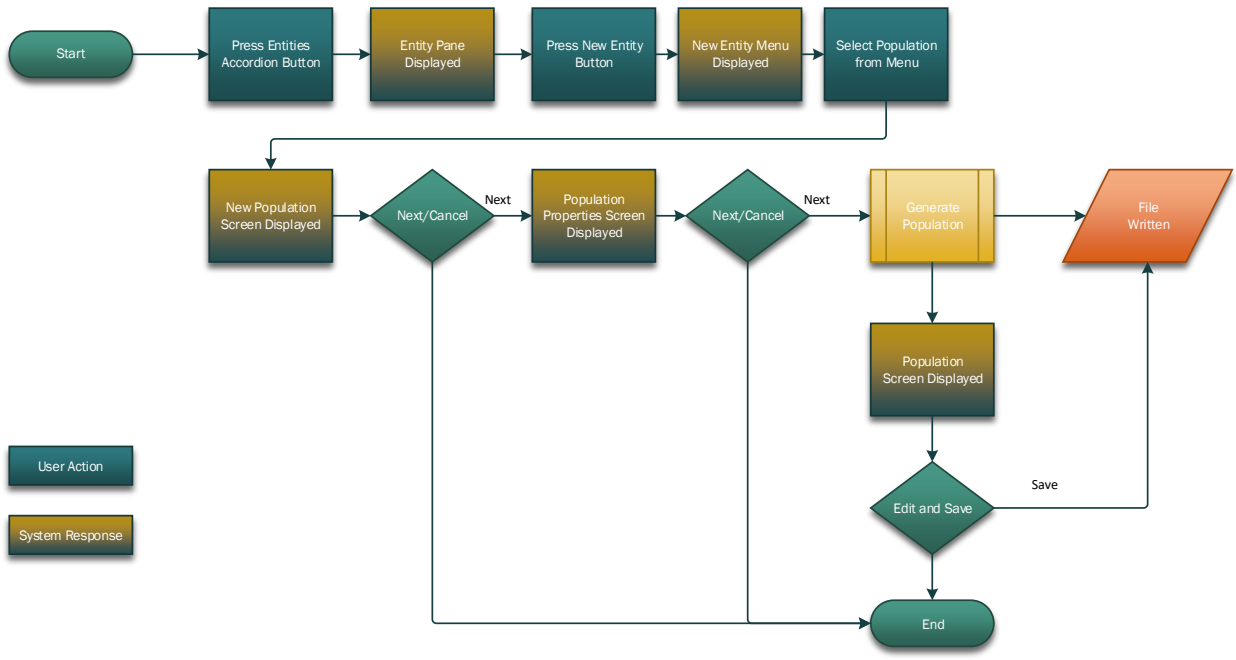

Figure 4: Generate Population User Action Process Flow

5.1.2. Create an Inventory (REQ-2)

The user creates a new inventory by selecting the Inventory item from the New Entity drop-down menu. The New Inventory screen is displayed. The user fills in the unique name of the new inventory and selects a data file to save the inventory to. The user then selects the Next button and the Inventory Management screen appears. The user creates Product records on the Products tab, then uses those products to create InventoryItem records on the Inventory tab. The user selects the Save button to save the inventory and product data.

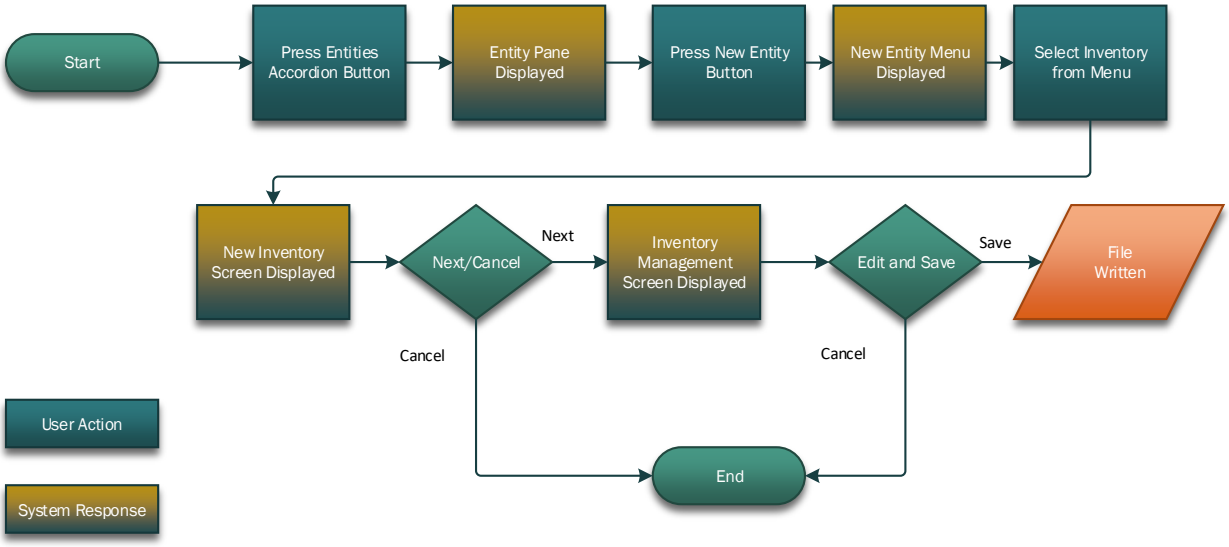

Figure 5: Create Inventory User Action Process Flow

5.1.3. Create and Execute a Simulation (REQ-3)

The user creates a new simulation by pressing the Market Simulation button in the Simulations accordion panel. The New Market Simulation screen is displayed. The users fills in a unique name for the simulation run, chooses a Population and Inventory to use for the run, sets the start date and end date, and selects an output folder for the simulation output. The user can save the simulation configuration to a file, cancel the new simulation, or execute the simulation by pressing the corresponding button. If the user presses the Run Simulation button, the system begins the simulation and displays the Simulation Execution screen. The user can observe the simulation in process, and has the option to manually terminate the simulation.

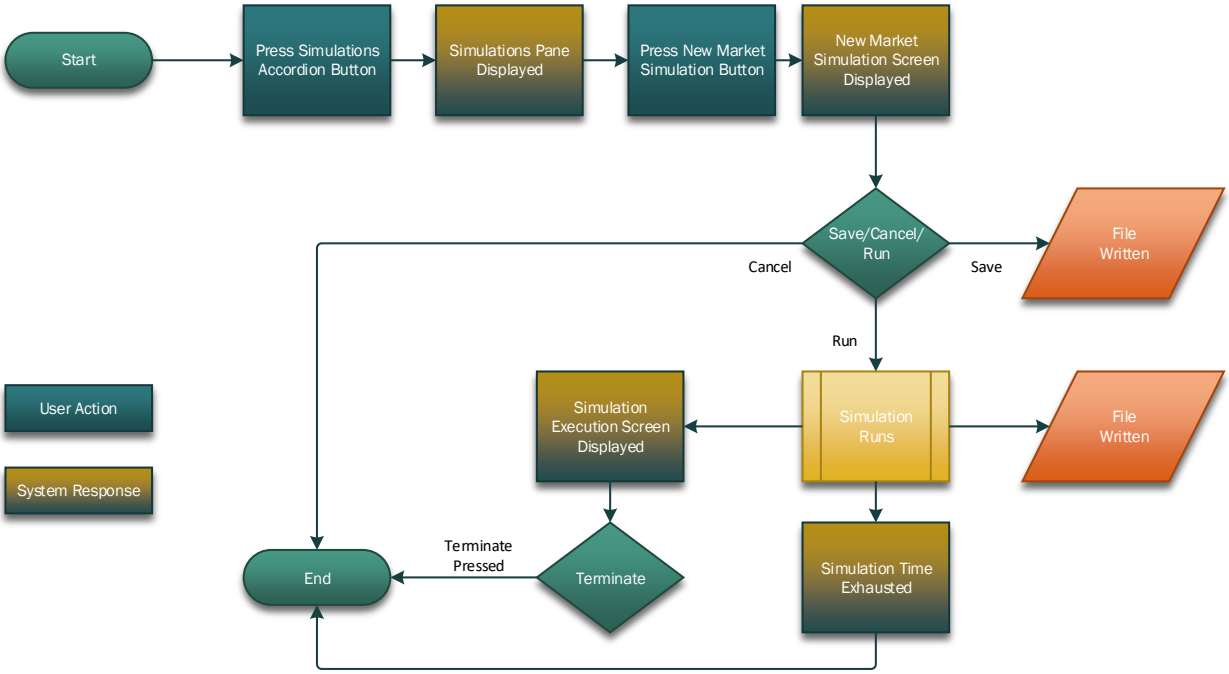

Figure 6: Create and Execute Simulation User Action Process Flow

5.2. System Components

The Consumer Behavior Simulation functionality consists of two system processes: generating a population and executing a simulation run. The following sections show the flow of each of these processes in detail.

5.2.1. Generate a Population (REQ-1)

The system generates a population in response to user input. The PopulationGenerator takes a Population object as input. The generator iterates until the number of people generated equals the size specified in the Population object. During each iteration, the generator uses the PersonRandomizer to create a Person object, supplying the gender and age distribution values specified in the population object. The generator uses the AddressRandomizer to assign an address to each generated Person object. When each Person object is complete, the generator adds it to the Population object. When the generator has finished iterating, it saves the Population to the save file specified, and returns control to the GUI. The GUI then displays the Population screen populated with the generated data.

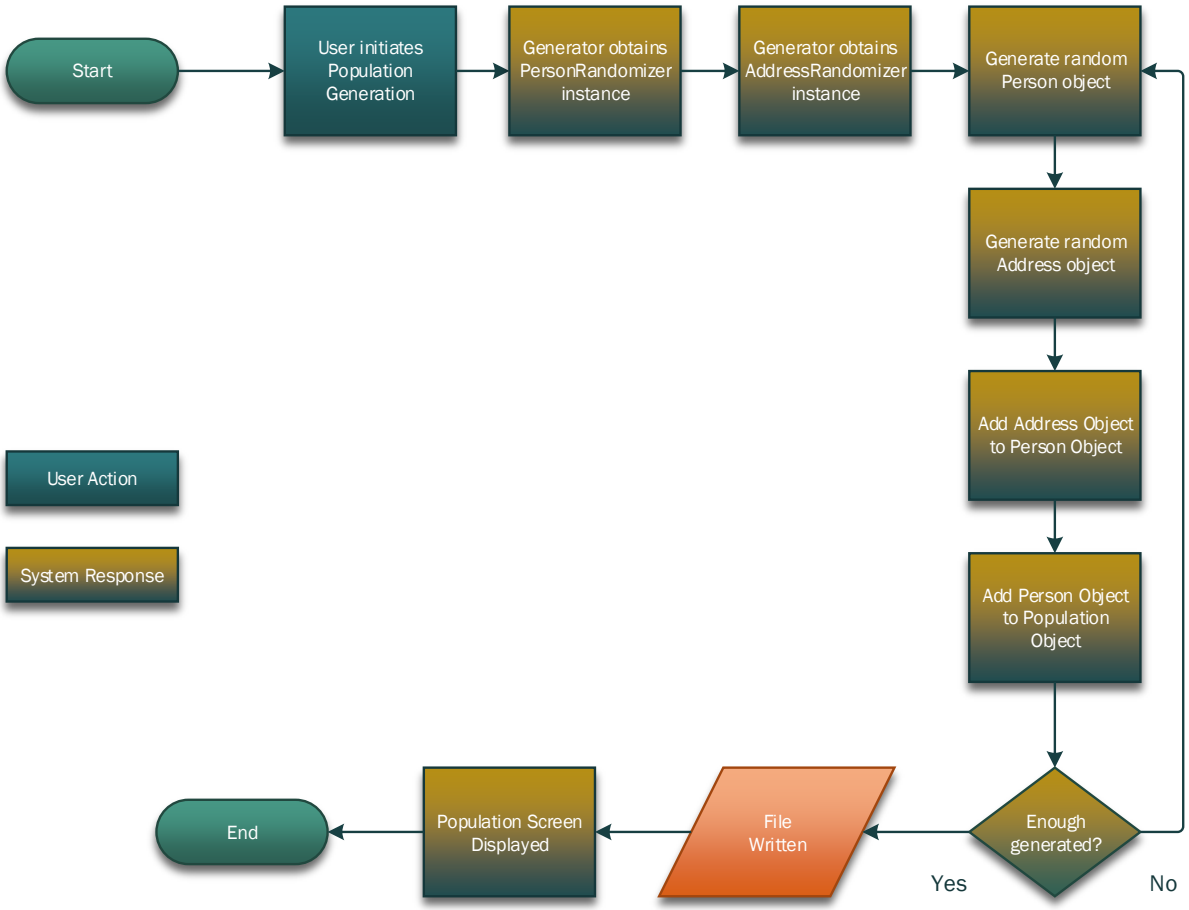

Figure 7: Generate Population System Action Process Flow

5.2.2. Execute a Simulation (REQ-3)

The Consumer Behavior Simulation executes when commanded by the user, either by pushing the Run Simulation button on the New Market Simulation screen, or by running the command-line interface with the appropriate saved configuration file. The MarketSimulation object is created with the MarketSimulationConfiguration object as input. The MarketSimulation will set the simulation clock to the user-supplied start date. The simulation will verify the existence of a Population with a non-zero size, and an Inventory with a non-zero size. The MarketSimulation will create a data file for output of generated transactions. The MarketSimulation will advance the simulation clock one day for each iteration of the simulation. The MarketSimulation will terminate when the simulation clock reaches the user-supplied stop date, unless the user commands early termination by pressing the Stop Simulation button. For each iteration, the MarketSimulation will use a Poisson probability distribution to decide which members of the population will shop during the iteration. For each shopper, the simulation will create a Transaction object, then use a Poisson probability distribution combined with

the purchase history of each shopper to decide which products the shopper will purchase during the iteration. The MarketSimulation will create a Purchase object for each purchase for the shopper and store it in the Transaction object. When purchasing is complete for the shopper, the MarketSimulation will store the Transaction object in the Person object's history property and write the Transaction to the output file. The simulation will then advance the simulation clock one day. If time remains in the simulation, and the user has not terminated the simulation, another iteration will begin. The MarketSimulation will write appropriate logging messages during execution.

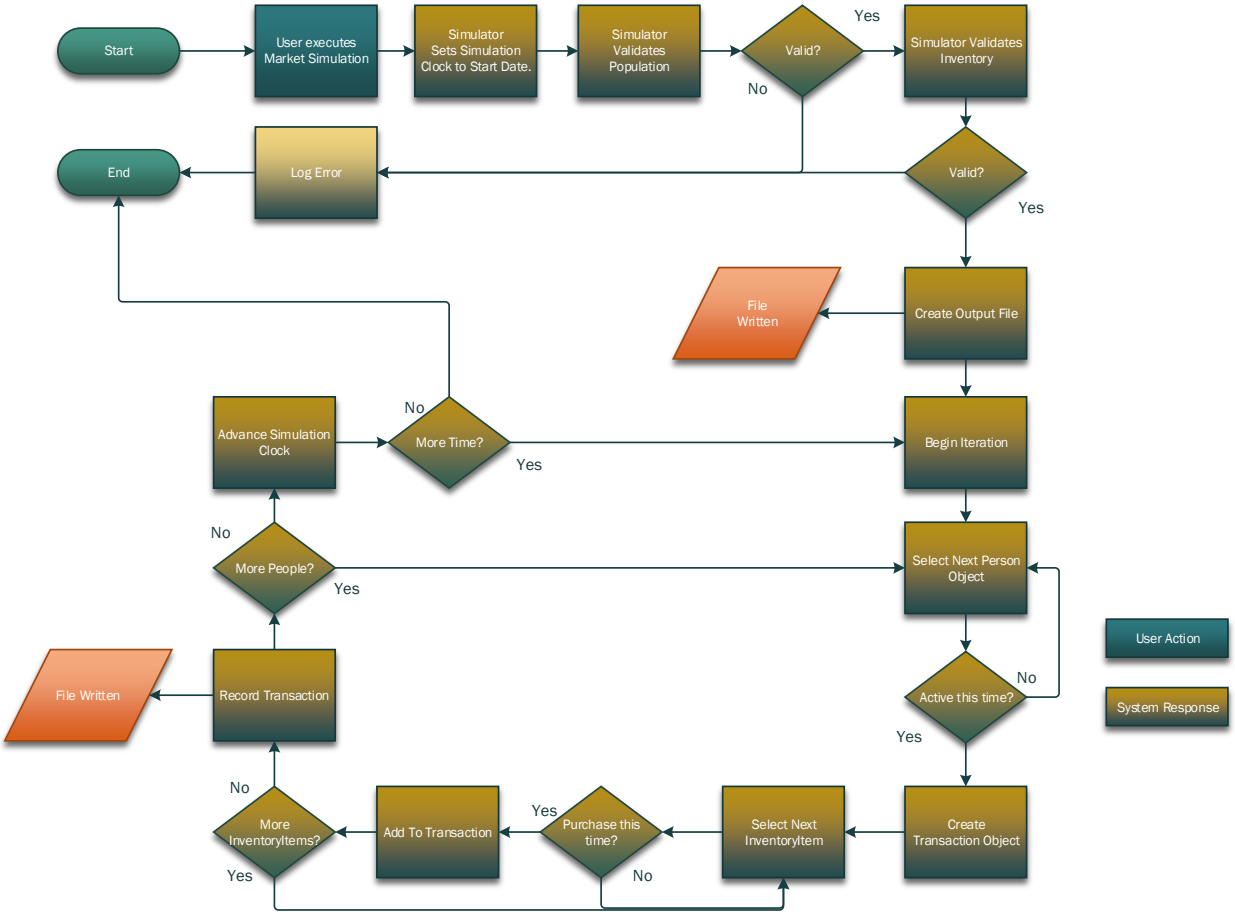

Figure 8: Execute Simulation System Action Process Flow

6. Human Interface Design

This section details the additions design of the Graphical User Interface components of the Consumer Behavior Simulation.

6.1. Basic Interface View

The following figure depicts the GUI main screen. The drawings in the following sections depict the components and controls that will be added to the basic GUI. These drawings are notional and not intended as the final look and feel of the components.

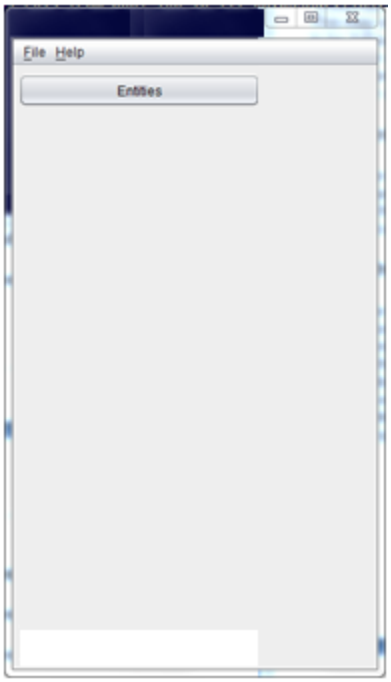

Figure 9: Basic GUI

6.2. Add Simulation Button to Accordion Pane (REQ-3.2.1)

A Simulation Button will be added to the accordion pane on the left side of the main display. Pressing the Simulation button will cause the currently open accordion panel to close and the Simulation panel to open.

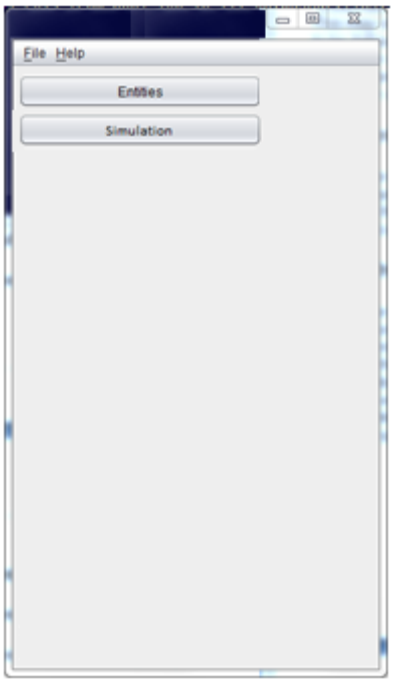

Figure 10: Main Screen with Simulation Button

6.3. Add New Market Simulation Button (REQ-3.2.2)

A New Market Simulation button will be added to the Simulation accordion pane. This button will be the same size and shape as the subordinate buttons on the other accordion panes. Pressing this button displays the New Market Simulation screen in the right pane of the main display.

Figure 11: Main Display Simulation Pane Open

6.4. Add Population (REQ-1.2.1) and Inventory (REQ-2.2.1) to Entities Drop-Down Menu

Pressing the Entities button causes a drop-down menu to be displayed prompting the user to choose an entity type to create or edit. Population and Inventory will be added to this menu.

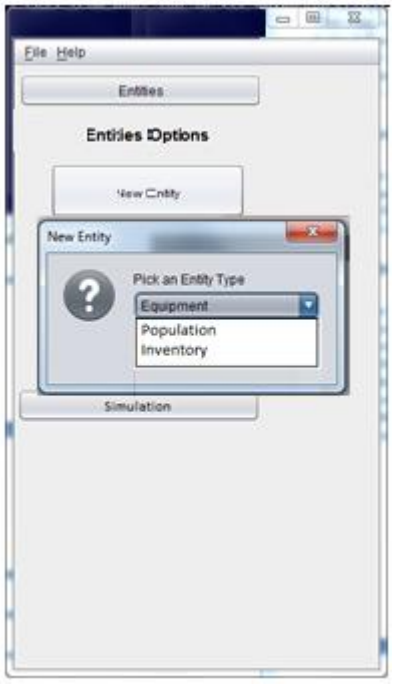

Figure 12: Entity Selection drop-down menu

6.5. Add New Population Screen (REQ-1.2.2)

When Population is selected from the Entity drop-down menu, the New Population screen is displayed in the right hand pane of the main display (REQ-1.2.2.3).

The New Population screen will contain a text field in which to enter the name of the new population (REQ-1.2.2.1), and a FileChooser field to allow the user to select or create a data file in which to save the new population (REQ-1.2.2.2).

The New Population screen will display a Next button to proceed to the Population Properties screen (REQ-1.2.2.4) which will be disabled until the name field and the save file field have been filled in. The screen will also display a Cancel button (REQ-1.2.2.5) that will close the screen and return to the main display without creating a new population.

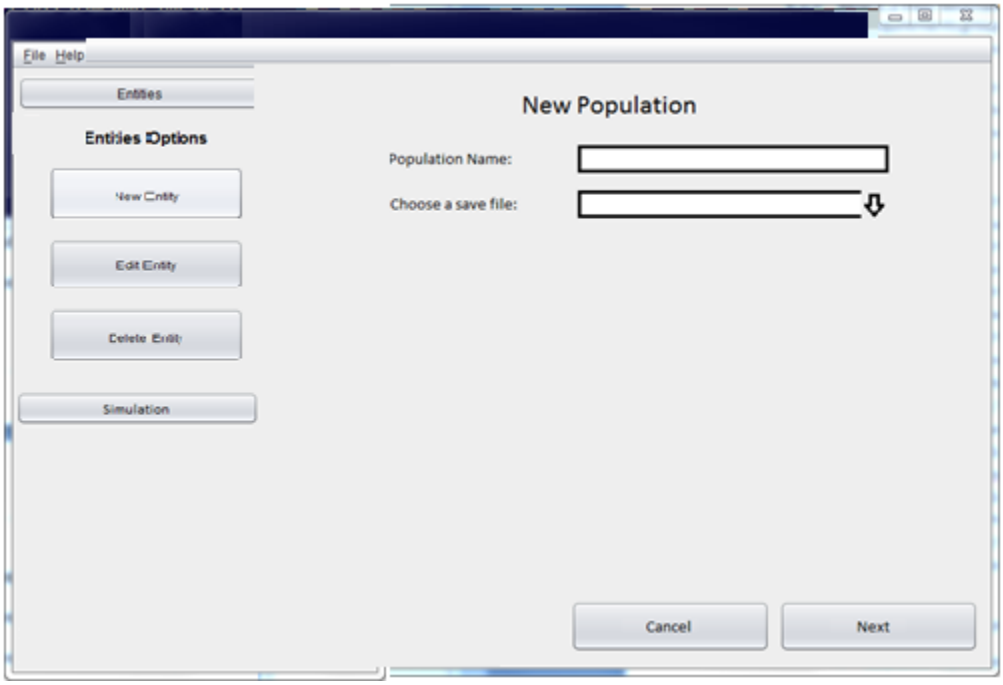

Figure 13: New Population Screen

6.6. Population Properties Screen (REQ-1.2.3)

The Population Properties screen will be displayed when the Next button is pressed on the New Population screen.

The Population properties screen will contain a text field in which the user can enter the size of the population (REQ-1.2.3.1).

The Population Properties field will also contain a slider with two linked text fields to allow the user to select the gender proportion for the population (REQ-1.2.3.2). The linked text fields will show the numeric percentages as the slider is adjusted.

The screen will also contain a multiple-control slider for adjusting the age distribution for the population (REQ-1.2.3.3). The slider will allow the adjustment of the minimum age, the mean age, and the maximum age. Linked text fields will display the resulting percentages as the sliders are adjusted.

The screen will display a Next button (REQ-1.2.3.4) to trigger the generation of the population and display the Population screen. The next button will be disabled until the size field is populated. The screen will also include a Cancel button (REQ-1.2.3.5) that will close the Population Properties screen and return to the main display.

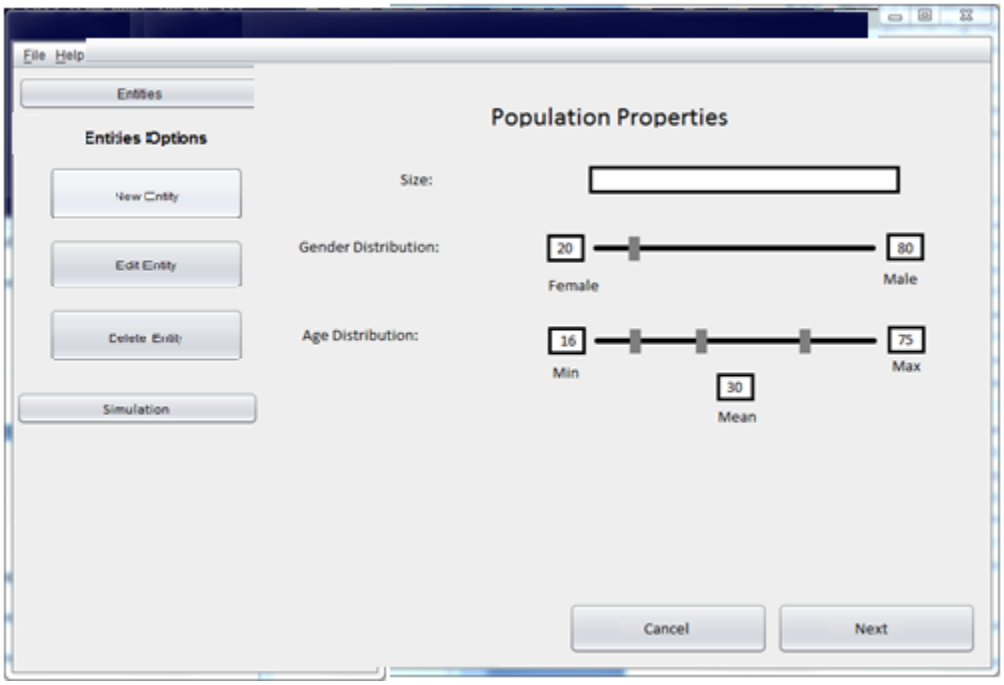

Figure 14: Population Properties Screen

6.7. Population Screen (REQ-1.2.4)

The Population screen will be displayed after the population has been generated, which is triggered by pressing the Next button on the Population Properties screen.

The Population screen displays an editable table inside a ScrollPane (REQ-1.2.4.1). This table will be filled in with generated Person entities. Values edited in the table will be reflected back into the underlying Person beans stored in the population object.

The Population screen will display a Save button (REQ-1.2.4.2) that writes the population data in the table to the data file specified on the New Population screen.

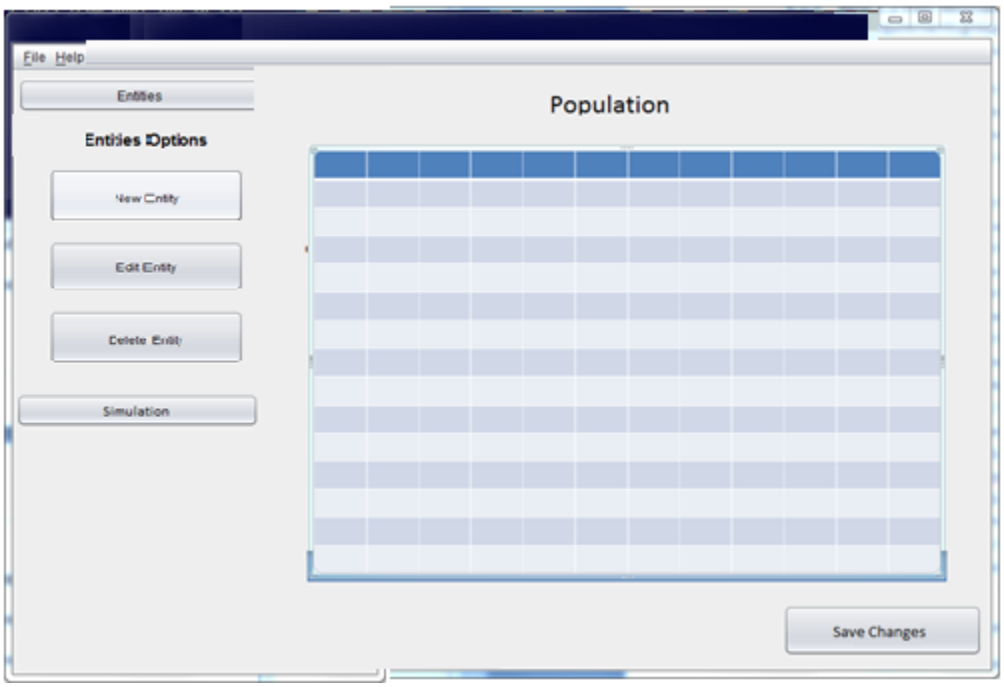

Figure 15: Population Screen

6.8. New Inventory Screen (REQ-2.2.2)

The New Inventory Screen will be displayed in the right hand pane of the main display when Inventory is selected from the New Entity drop-down menu (REQ-2.2.2.3).

The New Inventory screen will contain a text field in which to enter the name of the new inventory (REQ-2.2.2.1), and a FileChooser field to allow the user to select or create a data file in which to save the new inventory (REQ-2.2.2.2).

The New Inventory screen will display a Next button to proceed to the Inventory Management screen (REQ-2.2.2.4) which will be disabled until the name field and the save file field have been filled in. The screen will also display a Cancel button (REQ-2.2.2.5) that will close the screen and return to the main display without creating a new inventory.

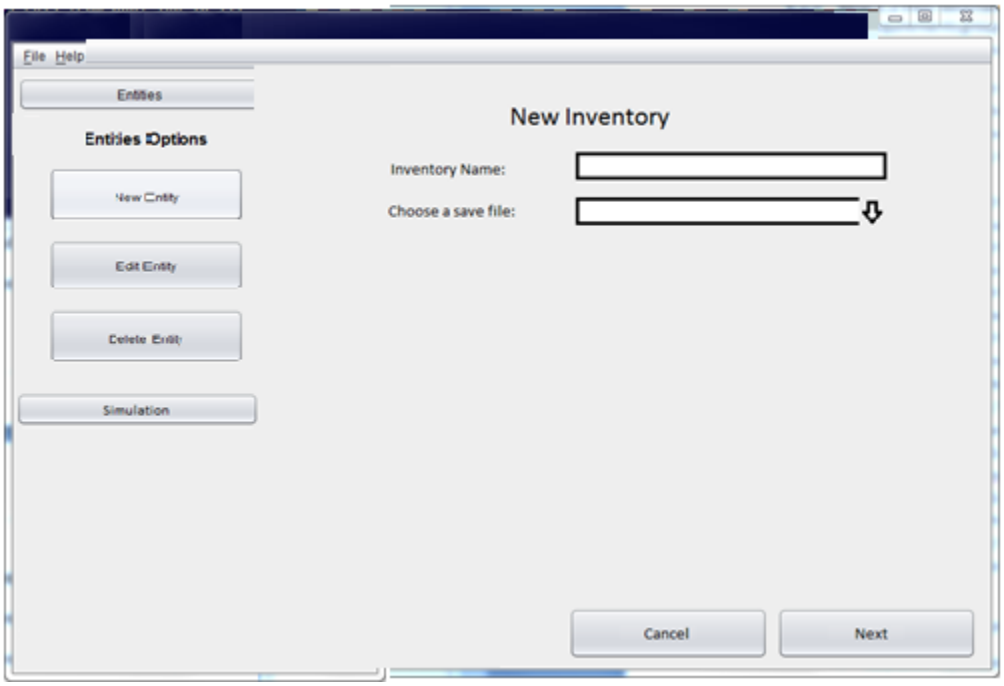

Figure 16: New Inventory Screen

6.9. Inventory Management Screen (REQ-2.2.3)

The Inventory Management screen will be displayed in the right hand pane of the main display when the Next button is pressed on the New Inventory screen. The Inventory Management screen will include a TabbedPane with two tabs, labeled Inventory and Products (REQ-2.2.3.1).

The Inventory tab will contain an editable table contained in a scroll pane (REQ-2.2.3.1.1). The table will initially be empty. InventoryItems can be created and edited in this table, and will be stored in the underlying Inventory object.

The Products table will contain an editable table contained in a scroll pane (REQ-2.2.3.1.2). The table will initially be empty. Products can be created and edited in this table, and will be stored in the underlying Inventory object.

The Inventory Management screen will display a Save button (REQ-2.2.3.2) that will save edits to the specified data file.

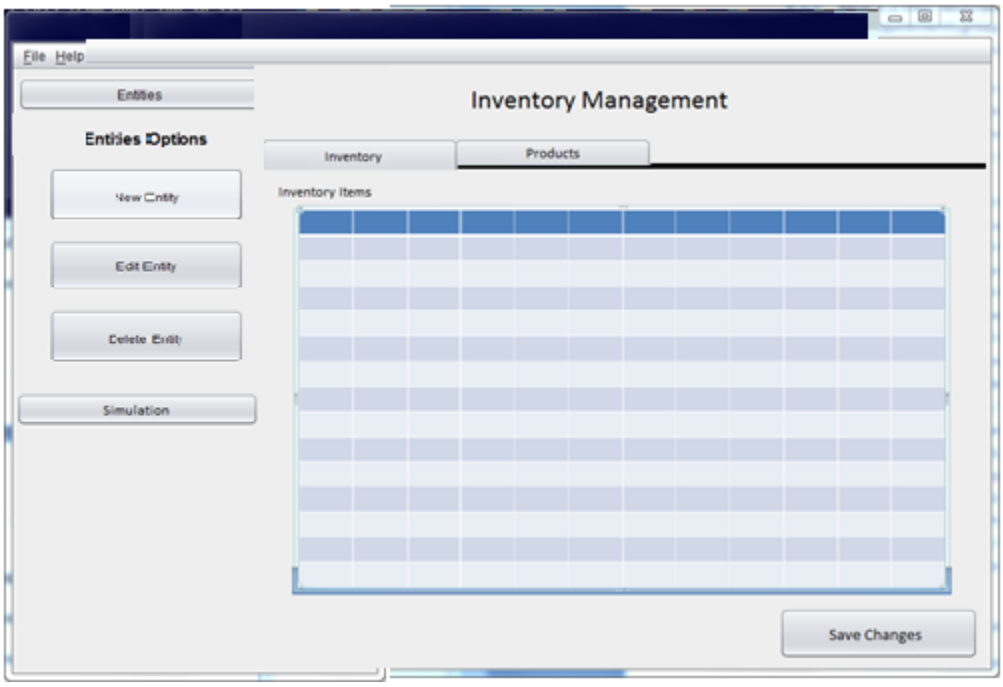

Figure 17: Inventory Management Screen

6.10. New Market Simulation Screen (REQ-3.2.3)

The New Market Simulation screen will be displayed in the right hand pane of the main display when the Market Simulation button in the Simulation accordion panel is pressed (REQ-3.2.3.1).

The New Market Simulation screen will contain a text field that will allow the user to enter a name for the new simulation run. (REQ-3.2.3.2)

The screen will display a text field with a FileChooser control that will allow the user to select a saved Population to use for this simulation run (REQ-3.2.3.3).

The screen will display a text field with a FileChooser control that will allow the user to select a saved Inventory to use for this simulation run (REQ-3.2.3.4).

The screen will display a text field with a DateChooser control that will allow the user to select the start date for this simulation run (REQ-3.2.3.5).

The screen will display a text field with a DateChooser control that will allow the user to select the end date for this simulation run (REQ-3.2.3.6).

The screen will display a text field with a FileChooser control that will allow the user to select or create an output folder for the results of this simulation run (REQ-3.2.3.7).
The screen will display a Save button that will allow the user to save this simulation configuration (REQ-3.2.3.8). The button will be disabled until all of the input fields have been filled in.

The screen will display a Cancel button that will allow the user to cancel the creation of the new simulation (REQ-3.2.3.9). This will close the New Market Simulation screen and return to the main display.

The screen will display a Run Simulation button that will allow the user to begin a simulation run with the configured parameters (REQ-3.2.3.10). This button will be disabled until all of the input fields have been filled in.

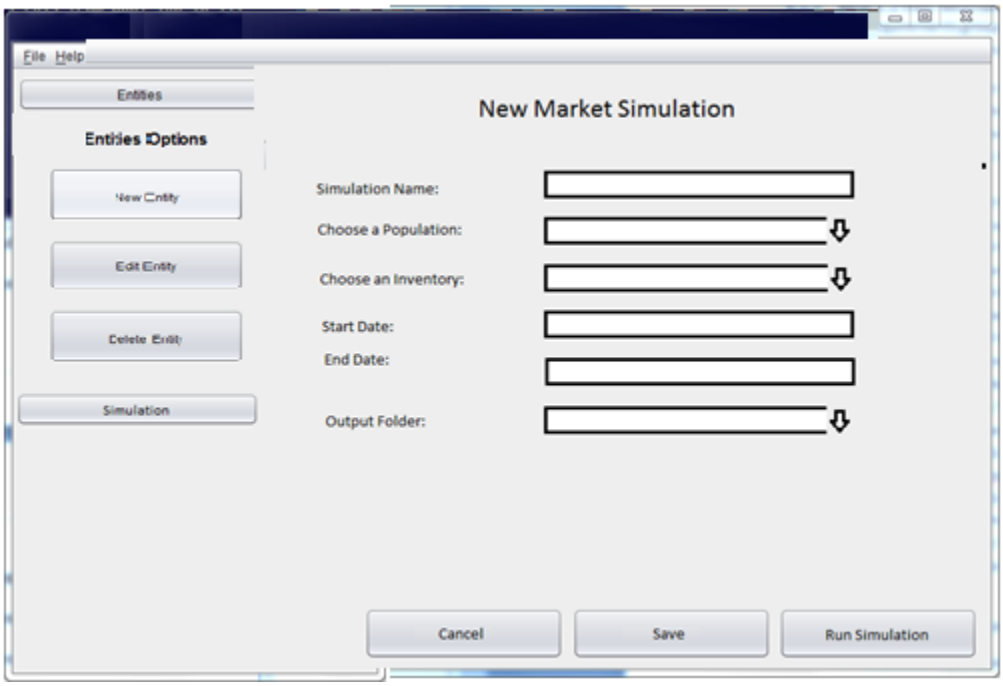

Figure 18: New Market Simulation screen

6.11. Simulation Execution Screen (REQ-3.2.4)

The Simulation Execution screen will be displayed in the right hand pane when the user has pressed the Run Simulation button on the New Market Simulation screen (REQ-3.2.4.1).

The Simulation Execution screen will display all of the configured parameters of the simulation run (REQ-3.2.4.2).

The Simulation Execution screen will display a scrolling text field that will show the logging output of the running simulation (REQ-3.2.4.3).
The screen will display the current simulation time (REQ-3.2.4.4).

The screen will display a button to allow the user to manually stop the current execution run (REQ-3.2.4.5).

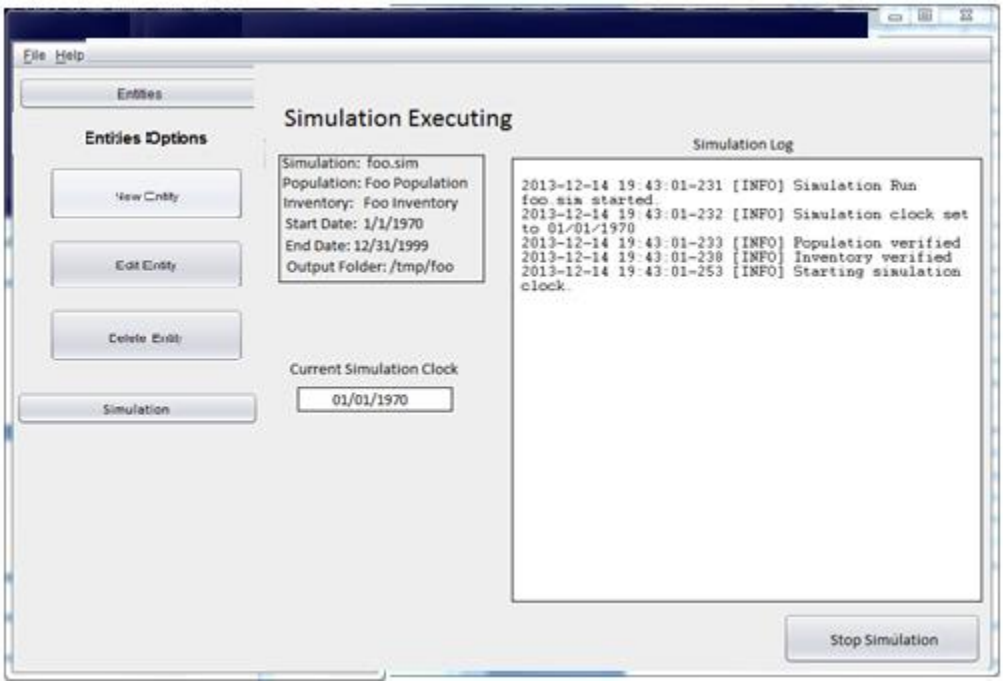

Figure 19: Simulation Execution Screen

Appendix A: Requirements Matrix

Requirement	Summary Description	Status
REQ-1	Generate Population	
REQ-1.1	Population Data Structures	
REQ-1.1.1	Population Class	
REQ-1.1.1.1	Name Property	
REQ-1.1.1.2	Size Property	
REQ-1.1.1.3	Gender Distribution Property	
REQ-1.1.1.4	Age Distribution Property	
REQ-1.1.2	Person Entity	
REQ-1.1.2.1	Add shopping history	
REQ-1.1.3	PostalAddress Class	
REQ-1.1.4	Preferences Class	
REQ-1.1.4.1	Purchase history	
REQ-1.2	Population User Interface	
REQ-1.2.1	Add Population to New Entity drop-down menu	
REQ-1.2.2	Create New Population Screen	
REQ-1.2.2.1	New Population Screen Name Field	
REQ-1.2.2.2	New Population Screen Data File Field	
REQ-1.2.2.3	New Population Screen displayed from New Entity drop-down menu	
REQ-1.2.2.4	New Population Screen Next button	
REQ-1.2.2.5	New Population Screen Cancel button	
REQ-1.2.3	Population Properties Screen	
REQ-1.2.3.1	Population Properties Screen Size Field	
REQ-1.2.3.2	Population Properties Screen Gender Distribution Field	
REQ-1.2.3.3	Population Properties Screen Age Distribution Field	
REQ-1.2.3.4	Population Properties Screen Next Button	
REQ-1.2.3.5	Population Properties Screen Cancel Button	
REQ-1.2.4	Create Population display and editing screen	
REQ-1.2.4.1	Population screen editable table	
REQ-1.2.4.2	Population screen Save Button	
REQ-1.3	Population Generator Processes	
REQ-1.3.1	Create PopulationGenerator class	
REQ-1.2.1.1	PopulationGenerator class takes Population object as input	
REQ-1.3.1.2	PopulationGenerator uses properties of Population class	
REQ-1.3.1.2.1	PopulationGenerator uses size property	
REQ-1.3.1.2.2	PopulationGenerator uses gender distribution	
REQ-1.3.1.2.3	PopulationGenerator uses age distribution	
REQ-1.3.1.3	PopulationGenerator uses existing PersonRandomizer	
REQ-1.3.1.3.1	Update PersonRandomizer Name Characteristics	
REQ-1.3.1.3.1.1	PersonRandomizer Male Person no maiden name	
REQ-1.3.1.3.1.2	PersonRandomizer Female Single Person no maiden name	

REQ-1.3.1.3.2	Update PersonRandomizer Age Characteristics	
REQ-1.3.1.3.2.1	PersonRandomizer Person under 18 must be Single	
REQ-1.3.1.4	PopulationGenerator uses existing AddressRandomizer	
REQ-1.3.1.5	PersonGenerator stores generated Person objects in provided Population object.	
REQ-2	Create and Manage Product Inventory	
REQ-2.1	Product Inventory Data Structures	
REQ-2.1.1	Create Inventory Class	
REQ-2.1.1.1	Inventory Class Name property	
REQ-2.1.1.2	Inventory Class item list	
REQ-2.1.2	Create Product Class	
REQ-2.1.2.1	Product Class Name property	
REQ-2.1.2.2	Product Class SKU property	
REQ-2.1.2.3	Product Class Category property	
REQ-2.1.2.4	Product Class Description property	
REQ-2.1.3	Create InventoryItem Class	
REQ-2.1.3.1	InventoryItem Class Product property	
REQ-2.1.3.2	InventoryItem Class Unit Cost property	
REQ-2.1.3.3	InventoryItem Class Unit Price property	
REQ-2.1.3.4	InventoryItem Class Quantity On Hand property	
REQ-2.2	Inventory User Interface	
REQ-2.2.1	Add Inventory to New Entity drop-down menu	
REQ-2.2.2	Create New Inventory screen	
REQ-2.2.2.1	New Inventory Screen Name field	
REQ-2.2.2.2	New Inventory Screen Data File field	
REQ-2.2.2.3	New Inventory Screen displayed by choosing Inventory from New Entity drop-down menu	
REQ-2.2.2.4	New Inventory Screen Next Button	
REQ-2.2.2.5	New Inventory Screen Cancel Button	
REQ-2.2.3	Create Inventory Management Screen	
REQ-2.2.3.1	Inventory Management Screen Inventory and Product Tabs	
REQ-2.2.3.1.1	Inventory tab editable table of InventoryItems	
REQ-2.2.3.1.2	Product tab editable table of Products	
REQ 2.2.3.2	Inventory Management Screen Save Button	
REQ-2.3	Inventory Management Processes	
REQ-2.3.1	Maintain referential integrity	
REQ-2.3.1.1	Verify product entered into InventoryItem exists in Product table	
REQ-2.3.1.2	Verify modifications to Products reflected in Inventory	
REQ-2.3.1.3	Verify deletion of Product causes InventoryItem referencing that Product to be deleted	
REQ-3	Generate Purchasing Activity	
REQ-3.1	Purchasing Activity Data Structures	
REQ-3.1.1	Create MarketSimulatorConfiguration class	
REQ-3.1.1.1	MarketSimulatorConfiguration Population property	

REQ-3.1.1.2	MarketSimulatorConfiguration Inventory property	
REQ-3.1.1.3	MarketSimulatorConfiguration Name property	
REQ-3.1.1.4	MarketSimulatorConfiguration Start Time property	
REQ-3.1.1.5	MarketSimulatorConfiguration End Time property	
REQ-3.1.1.6	MarketSimulatorConfiguration Output Directory property	
REQ-3.1.2	Create MarketSimulator class	
REQ-3.1.2.1	MarketSimulator takes MarketSimulatorConfiguration as input	
REQ-3.1.3	Create Transaction class	
REQ-3.1.3.1	Transaction Class Transaction Time property	
REQ-3.1.3.2	Transaction Class Customer property	
REQ-3.1.3.3	Transaction Class Purchase property	
REQ-3.1.3.4	Transaction Class Unique Identifier property	
REQ-3.1.4	Create Purchase class	
REQ-3.1.4.1	Purchase Class Product property	
REQ-3.1.4.2	Purchase Class Quantity property	
REQ 3.1.4.3	Purchase Class Price property	
REQ-3.2	Market Simulation User Interface	
REQ-3.2.1	Add Simulations button to accordion pane	
REQ-3.2.2	Add Market Simulation button to Simulations pane	
REQ-3.2.3	Create New Market Simulation Screen	
REQ-3.2.3.1	New Market Simulation Screen displayed when Market Simulation button pressed	
REQ-3.2.3.2	New Market Simulation Screen Name field	
REQ-3.2.3.3	New Market Simulation Screen Population field	
REQ-3.2.3.4	New Market Simulation Screen Inventory field	
REQ-3.2.3.5	New Market Simulation Screen Start Date field	
REQ-3.2.3.6	New Market Simulation Screen End Date field	
REQ-3.2.3.7	New Market Simulation Screen Output Directory field	
REQ-3.2.3.8	New Market Simulation Screen Save button	
REQ-3.2.3.9	New Market Simulation Screen Cancel button	
REQ-3.2.3.10	New Market Simulation Screen Execute button	
REQ-3.2.4	Create Simulation Execution Screen	
REQ-3.2.4.1	Simulation Execution Screen displayed when Execute button pressed	
REQ-3.2.4.2	Simulation Execution Screen displays Simulation configuration parameters	
REQ-3.2.4.3	Simulation Execution Screen displays logging information in scrolling window	
REQ-3.2.4.4	Simulation Execution Screen displays current simulation time	
REQ-3.2.4.5	Simulation Screen Stop Button	
REQ-3.3	Market Simulation Processes	
REQ-3.3.1	Market Simulation Started by Execute button or CLI	
REQ-3.3.1.1	Market Simulation tracks simulation time with simulation	

	clock	
REQ-3.3.1.2	Market Simulation sets simulation clock to specified start date	
REQ-3.3.1.3	Market Simulation advances simulation clock 1 day per iteration	
REQ-3.3.1.4	Market Simulation terminates when simulation reaches specified stop date	
REQ-3.3.1.5	Market Simulation creates output file in specified directory	
REQ-3.3.1.6	Market Simulation iterates until time exhausted	
REQ-3.3.1.6.1	Market Simulation uses probability distribution to select shoppers	
REQ-3.3.1.6.2	Market Simulation uses probability distribution and shopping history to select products for each shopper	
REQ-3.3.1.6.3	Market Simulation creates a Transaction object for each shopper	
REQ-3.3.1.6.4	Market Simulation creates a Purchase object for each product purchased	
REQ-3.3.1.6.5	Market Simulation stores the Transaction in the Person record	
REQ-3.3.1.6.6	Market Simulation writes the Transaction record to the output file	
REQ-3.3.1.6.7	Market Simulation writes appropriate logging messages	
REQ-3.3.1.7	Market Simulation terminates on user command	
REQ-4	Create Data Files	
REQ-4.1	Create Data Files Data Structures	
REQ-4.1.1	All object classes that must be saved must implement the standard serialization interface	
REQ-4.1.1.1	Population Class must implement the standard serialization interface	
REQ-4.1.1.2	Inventory Class must implement the standard serialization interface	
REQ-4.1.1.3	InventoryItem Class must implement the standard serialization interface	
REQ-4.1.1.4	Product Class must implement the standard serialization interface	
REQ-4.1.1.5	Transaction Class must implement the standard serialization interface	
REQ-4.1.1.6	Purchase Class must implement the standard serialization interface	
REQ-4.1.1.7	MarketSimulationConfiguration Class must implement the standard serialization interface	
REQ-4.1.2	Transaction output data must be written using the standard Formatter and Wrapper classes	
REQ-4.2	Data Output User Interface	Complete
REQ-4.3	Data Output Processes	Complete

Appendix B: Glossary

CI - Continuous Integration
CLI - Command Line Interface
CM - Configuration Management
DT - Data Tactics Corporation
ERD – Entity Relationship Diagram
GUI - Graphical User Interface
JVM - Java Virtual Machine
PMP - Project Management Plan
SCM - Source code management
SDD - Software Design Document
SRS - Software Requirements Specification
SVN - The Subversion source code management system