**Software Testing Address Book**

**Andrew Henk, Isaac Gainey, Matt Westman**

**March 22nd, 2016**

**CEN 4072 Software Testing**

**Instructors: Dr. Ingrid Buckley**

**Software Engineering Department**

**Florida Gulf Coast University**

**Ft. Myers, FL 33965**

**Table of Contents**

**1. Introduction**

**1.1 Purpose**

**1.2 Definitions, Acronyms, and Abbreviations**

**1.3 Overview**

**2. Product Overview**

**2.1 Assumptions**

**2.2 Use Case Diagram**

**2.3 Use Case Descriptions**

**3. Specific Software Requirements**

**3.1 Functional Requirements**

**3.2 Non-functional Requirements**

**3.3 Performance Requirements**

**3.4 Design Constraints**

**1. Introduction**

**1.1 Purpose**

Software Project for Software Testing. The project will be an address book. This program will be critically viewed and tested. It will store people's names, their phone number, and their address.

**1.2 Definitions, Acronyms, and Abbreviations**

Address Book - record of the names, addresses, and telephone numbers of friends, businesses, etc.

Serialization - is the process of converting an object into a stream of bytes in order to store the object or transmit it to memory, a database, or a file.

**1.3 Overview**

This program will be an address book, containing people’s names, their phone number, and their address. The address book will serve as a collection of contacts for the user. As well as maintain the ability to edit each contact and able to save and load different address books using a graphical user interface. It will be designed with the intent of it being vigorous tested to the requirements. It will be able to edit each contact and save on a local storage drive to access later.

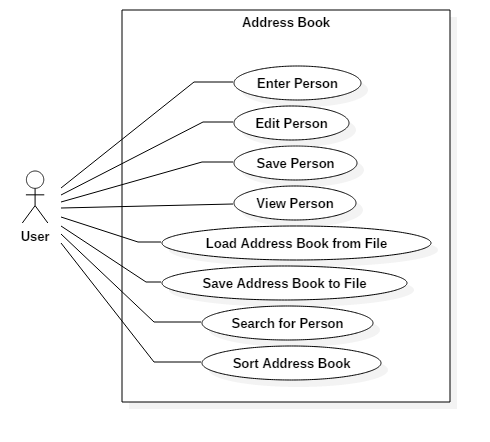
**2. Product Overview**

**2.1 Assumptions**

* Minimum Hardware:
  + Intel Pentium IV 1.4 GHz
  + 40 GB Hard Drive
  + 2 GB RAM
* Input
  + English Language is used
  + Names are no longer than 50 characters
  + Names do not contain special characters or numbers
  + Phone numbers no longer than 11 digits.

**2.2 Use Case Diagram**

Class AddressBookGUI used as a frontend for human users to interact with the program. Commands are given graphically, which are sent to the appropriate functions.



**2.3 Use Case Descriptions**

|  |  |
| --- | --- |
| **Use Case** | **Description Flow** |
| AddressBookController | * The User enters data into the GUI * The User hits ‘add a person’ button via GUI * The GUI parses the data and passes the the data to the controller * The controller creates a new person and adds to the collection. * The GUI updates with the new person in the address book * The User will view the GUI and be in aww |
| FileSystem | * The User enters the filename of address book data file * The User hits ‘load address book’ * The Gui launches the file system, parse the file name, and passes the file name * The file system searches for the file * The file system parses the data and passes to the controller * The controller enters the data into the address book and signal the the GUI * The GUI will update with new data with the address book * The User will view the GUI and be in aww |
| Person | * The User clicks on a person or types the name of a person * The GUI passes the Name to the controller * The Controller searches for the person and returns the person to the GUI * The GUI access the person and parses the person’s data to display to the User * The User will view the GUI and be in aww |

**3. Specific Software Requirements**

|  |
| --- |
| No: 1 |
| Statement: The software shall open a stored address book from a file |
| Test Criteria: verifying that the current address book contains the data imported from the file |

|  |
| --- |
| No: 2 |
| Statement: The software shall switch between address books from files. |
| Test Criteria: visual confirmation for loading an existing address book |

|  |
| --- |
| No: 3 |
| Statement: The software shall be able to create a new address book |
| Test Criteria: Check that null pointer exception from the GUI will not be created |

|  |
| --- |
| No: 4 |
| Statement: The software shall edit address books’ individual contact |
| Test Criteria: Select an individual contact and verify changes were made |

|  |
| --- |
| No: 5 |
| Statement: The software shall save address books to a local file |
| Test Criteria: |

|  |
| --- |
| No: 6 |
| Statement: The software shall delete address books |
| Test Criteria: The local file will be deleted from the storage drive |

|  |
| --- |
| No: 7 |
| Statement: The software shall hold at least 10 unique identities within an address book |
| Test Criteria: automated tests for putting 11 names into the system |

|  |
| --- |
| No: 8 |
| Statement: The software shall delete a contact |
| Test Criteria: Delete a contact and then searching for that same contact to not be found |

|  |
| --- |
| No: 9 |
| Statement: The software shall search for a contact |
| Test Criteria: confirm the table contains contact with search criteria |

|  |
| --- |
| No: 10 |
| Statement: The software shall ask for a confirmation that the contact will be removed permanently before deleting it. |
| Test Criteria: Delete a contact and then searching for that same contact |

|  |
| --- |
| No: 11 |
| Statement: The software shall add entry in an address book for a new contact. |
| Test Criteria: Searching for the new contact and verify that the contact exist in the database. |

|  |
| --- |
| No: 12 |
| Statement: The software shall edit current contact information. |
| Test Criteria: Edit a contact and then searching again for the same contact and verify that the alteration took place. |

|  |
| --- |
| No: 13 |
| Statement: The software shall sort entries by name and ZIP code |
| Test Criteria: verify the order of saved entries |

|  |
| --- |
| No: 14 |
| Statement: The GUI shall display a list of names of persons in the current address book |
| Test Criteria: visual confirmation of data entered into address book being listed |

|  |
| --- |
| No: 15 |
| Statement: The GUI shall display the title of the current loaded address book |
| Test Criteria: visual confirmation of the address book title being displayed |

|  |
| --- |
| No: 16 |
| Statement: The GUI shall display if the address book has not been saved and needs to be saved |
| Test Criteria: creating a change to the address book and monitoring for the change in “Save” state |

|  |
| --- |
| No: 17 |
| Statement: The “Save” button shall save all current contacts of the address book to a file. |
| Test Criteria: succeeding to save changes when changes have been made |

|  |
| --- |
| No: 18 |
| Statement: The software shall offer the user to save the current address book after creating or editing a contact |
| Test Criteria: succeed in attempting to trigger the saving suggestion |

**3.1 Functional Requirements**

3.1.1 - The program shall display the list of all names collected.

3.1.2 - The names display shall be in alphabetic order.

3.1.3 - After selection of an individual, all available information will be displayed to the user.

**3.2 Non-functional Requirements**

3.2.1 - The program shall be functional on all desktop’s OS.

**3.3 Performance Requirements**

3.3.1 - The machine using this program will have Java.

**3.4 Design Constraints**

3.4.1 - The program shall be written in java.

3.4.2 - The collection of names and associated data shall be stored locally.