**Software Testing Address Book**

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

**February 8th, 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 a simple address book. This program will be critically viewed and tested. It will store people's names, their phone’s number, and their address.

**1.2 Definitions, Acronyms, and Abbreviations**

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

**1.3 Overview**

This program will be an address book, containing people’s names, their phone number, and their address. It will be designed with the intent of it being vigorous tested to the requirements.

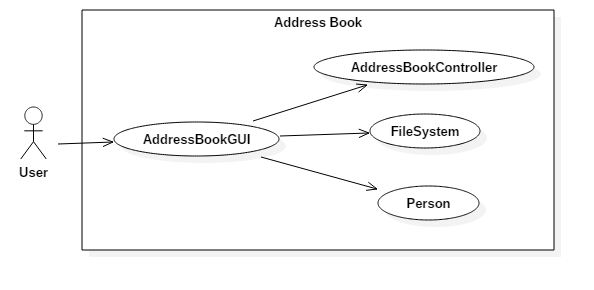
**2. Product Overview**

**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** |
| **AddressBookController** |  |
| **FileSystem** |  |
| **Person** |  |

**3. Specific Software Requirements**

|  |
| --- |
| **No: 0** |
| **Statement: The software shall hold at least 10 unique identities** |
| **Test Criteria: automated tests for putting 10 names into the system** |

|  |
| --- |
| **No: 1** |
| **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: 2** |
| **Statement: The GUI shall display the title of the current address book if such exists** |
| **Test Criteria: visual confirmation of the address book title being displayed** |

|  |
| --- |
| **No: 3** |
| **Statement: The GUI shall display the state of the “Save” menu option.** |
| **Test Criteria: creating a change to the address book and monitoring for the change in “Save” state** |

|  |
| --- |
| **No: 4** |
| **Statement: The “Save” state will prevent the user from saving when no changes have been made** |
| **Test Criteria: failing to save changes when no changes have been made** |

|  |
| --- |
| **No: 5** |
| **Statement: The GUI “Save” button will be inactive when the “Save” state demands it** |
| **Test Criteria: visual confirmation of the “Save” button becoming unclickable** |

|  |
| --- |
| **No: 6** |
| **Statement: The “Save” state will allow the user to save when changes have been made** |
| **Test Criteria: succeeding to save changes when changes have been made** |

|  |
| --- |
| **No: 7** |
| **Statement: The GUI “Save” button will be active when the “Save” state demands it** |
| **Test Criteria: visual confirmation of the “Save” button becoming clickable** |

|  |
| --- |
| **No: 8** |
| **Statement: The software shall read a stored address book from a file** |
| **Test Criteria: verifying that the current address book contains the data imported from the file** |

|  |
| --- |
| **No: 9** |
| **Statement: The software shall write to a file the current address book** |
| **Test Criteria: saving the address book to a file then performing use case No. 8** |

|  |
| --- |
| **No: 10** |
| **Statement: The software shall add, edit, and delete persons from the current address book** |
| **Test Criteria: saving the address book to a file then performing use case No. 8** |

|  |
| --- |
| **No: 11** |
| **Statement: The software shall sort entries by name and ZIP code** |
| **Test Criteria: verify the order of saved entries** |

|  |
| --- |
| **No: 12** |
| **Statement: The software shall offer the user to save the current address book when necessary** |
| **Test Criteria: succeed in attempting to trigger the saving suggestion** |

|  |
| --- |
| **No: 13** |
| **Statement: Deleting a contact. 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: 14** |
| **Statement: Adding a contact. The software shall create a new entry in the database for a new contact.** |
| **Test Criteria: Searching for the new contact and verify that the contact exist in the database.** |

|  |
| --- |
| **No: 15** |
| **Statement: Editing a contact. The software shall alter current contact information.** |
| **Test Criteria: Edit a contact and then searching again for the same contact and verify that the alteration took place.** |

**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.