Software Requirements Specification

for

Address Book

Version 1.0

Emertxe Information Technologies (P) Ltd.

05-November-2012

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Team Emertxe	05-Nov-2012	Initial Draft	1.0

1. Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

Address Book is a small application written in C language. It keeps track of names and telephone/mobile numbers and e-mail addresses. It is a console application which uses standard I/O for adding and deleting contact names, phone numbers and e-mail addresses, searching names and associated numbers and email addresses, updating numbers and email addresses, and deleting contacts.

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

The SDLC for this product uses an incremental model with multiple phases. Requirements for phases I and II are given in this document. Within each phase, the water fall model is used with the following steps:

R – Requirements

D – Design, both high and low level.

C – Coding

T – Testing (Unit Testing + System Testing)

The documents listed above will have tags against each important requirement, design, coding or testing item. Traceability can be documented and verified for completeness of the product.

1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

This document is the requirement specifications for the Address Book. It documents all the requirements for the product. It serves as a reference for validating the same with the customer. Once accepted by the customer, it serves as a reference for high level design for the “Address Book”. The designs and requirements are tagged for traceability.

1.4 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

Phase I of the Address Book will be a small product demonstrating the use of only C in such an application. It will assume a small file-based storage which will be accessed using file operations. It will not be suitable for a large amount of data. It will not be using any optimal data structures or algorithms in the initial versions. Phase I does not include import/export feature and groups.

Later versions can incorporate fast searching (for example, using indexes) and fast addition/updates, larger amounts of data using suitable data structures and algorithms.

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

Customer Requirement Specification Document: Address_Book_CRS.odt.

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

Address Book is a new product. It is designed to support a set of functionalities and for a small amount of book data in the initial version. Later versions can extend it for larger books and more efficient operations.

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

The functionality to be implemented in the initial version are:

Phase I functionality

1. Add a Contact
    1. Add a contact giving user name [mandatory].
    2. Add phone numbers to the newly added contact[optional].
    3. Add e-mail addresses to the newly added contact[optional].

2. List All Contacts

3. Search a Contact
    1. Search a contact by name.
    2. Search contact by phone number.
    3. Search contact by e-mail address.
    4.
4. Edit a Contact
    1. Edit name.
    2. Add/edit/delete a phone number.
    3. Add/edit/delete an email address.

5. Delete a contact along with all associated phone numbers and email addresses.

6. Save the address book.

7. Exit the application.

Phase II functionality

Phase II functionality may be described in this document. However, they can be modified/refined later and implemented only after Phase I is successfully implemented, tested and deployed.

8. Export/import contacts.
9. Groups

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or

privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

There will a single person controlling all editing as well as searching options. In future versions, a server-like system can be provided for a restricted user who has limited operations such as searching only. Groups can be added giving more powerful access to data about other members in the group.

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

This product can be used on Linux systems which has C compiler and run-time support, i.e., standard C library(glibc).

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

This project is meant more for demonstration than for an end user. There is restriction to using only C programming at application level. No sophisticated data structures should be used for initial version. Simple arrays and strings can be used.

Later versions can add more to it after gaining sufficient knowledge of data structures, efficient algorithms and UNIX internals.

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

1. On-line help with -h command option.
2. README.txt Release Notes files.
3. README.html browser based help.
4. For any prompt, using help should provided context sensitive help.

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions

are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

Address Book will be invoked using command line. A list of command line and Menu options is given below:

Let ABK be the name of the executable.

The menus invoked in a hierarchical order after invocation of ABK are:

ABK main menu

1. [A]dd contact
2. [L]ist all the contacts
3. [E]dit contact name and details
4. [S]earch – Search for a contact, display details
5. [D]elete contact
6. SA[V]E
7. E[X]IT to command line.

Add contact menu

1. Add a [C]ontact name
2. Add [D]etails menu
3. E[X]IT to main menu

Add details menu

1. Add a [P]hone number menu
2. Add an [E]mail address menu
3. E[X]IT to add contact menu

Edit contact menu

1. [L]ist all contact names with details
2. Search by [C]ontact name
3. Search by [P]hone number
4. Search by [E]mail address

Choose contact to use by serial number

If successful go to modifications menu

Modifications Menu

1. Edit [C]ontact
2. Edit [P]hone numbers of the contact
3. Edit [E]mail addresses of the contact
4. E[X]IT to main menu

Edit Phone Numbers Menu

1. [A]dd phone number
2. [M]odify phone number
3. [D]elete phone number
4. E[X]it to modification menu

Edit Email Addresses Menu

1. [A]dd email address
2. [M]odify email address
3. [D]elete email address
4. E[X]it to modification menu

Choose contact prompt: Enter the serial number of contact
If valid contact, go to edit details menu, else give message and prompt again.
If response to prompt is blank string, return to edit contact menu.

Search menu

1. [L]ist all contacts
2. Search by [C]ontact name
3. Search by [P]hone number
4. Search by [E]mail address
5. E[X]IT

Delete contact menu:

1. Search for a contact
2. Choose a name (if more than 1 successfully)
3. Delete
4. Exit

Exit from application

1. [S]ave changes and exit
2. [D]iscard changes
3. [C]ancel

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

For the user, the monitor and the keyboard connected to the system will be the hardware interfaces to use this application.

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

Address Book will store all its data in a file with .ABK extension. It will use the default file unless any other file is specified at invocation time.

After Phase II, It will be able to import from and export to other similar files.

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

The product will use multiple phases for delivering the requirements. The features below are scheduled for a first release and a second release. The first release will have the basic functionalities.

Phase I Features:

These are the basic functionality of address book.

4.1 Adding a Contact

This is the feature by which contacts are added to the address book. Once a contact is added, its details, namely, zero or more phone numbers and zero or more email addresses can be associated with it.

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from from a high of 1 to a low of 9).>

Priority: 1

1. It should be possible to add the name of a contact.
2. If the contact name is already present, it should display the error and not add the contact again.
3. Once a contact has been added, it should be possible to add phone numbers and email addresses for the contact.
4. It should be possible to exit the application.
5. Before exiting, the user will be asked if the data have to be saved or not.
6. In case save is required, it will be saved in the default .ABK file.

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

The application should be invoked and should be in the main menu. Choosing the menu item code of [A] should start the interaction for adding a contact. The system will ask for the name of the contact. In case it succeeds, it should display success and move to the add details menu organized under the main menu.

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

TAG	Menu	Option Chosen	Action
RS_AB_01_01	Command line	ABK	Main menu should be displayed and add contact menu option 'A' should be selected to use this feature of the address book.
RS_AB_01_02	Add contact	[A]	Prompt for the name. Create a contact with the response name if not present and move to add details menu. Else give error and return to add contact menu.
RS_AB_01_03	Add details menu	[P]	Prompt for a phone number. If response has a valid phone number which is not present already for the last contact, add it. Else if is not valid or if it is already present, give suitable error message and prompt again. If response is a blank string, return to add details menu.
RS_AB_01_04	Add details menu	[E]	Prompt for an email address. If response has a valid email address which is not present already for the contact, add it. Else if is not valid or if it is already

			present, give suitable error message and prompt again. If response is a blank string, return to add details menu.
RS_AB_01_05	Add details menu	[X]	Return to add contact menu.
RS_AB_01_06	Add contact menu	[X]	Return to main menu.
RS_AB_01_07	ABK main menu	[X]	Ask user whether to save any changes if made. Save .ABK file if needed. Return from the program.

4.2 Deletion of Contact

By this feature a contact and all its associated phone numbers and email addresses will be deleted.

4.2.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

Priority: 9

1. It should be possible to delete a contact.
2. For this a contact name can be entered and searched.
3. If found, the user will be asked to confirm deletion. If user confirms then delete.
4. Otherwise, if not found or if user cancels it, do not delete the contact.
5. Once a contact is deleted, all associated phone numbers and email addresses should also be deleted.

4.2.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

The application should be invoked and should be in the ABK main menu. Choosing the menu item code of [D] should start the interaction for deleting a contact. The system will ask for the name of the contact. In case it finds, it should ask for confirmation from the user and after that delete the contact along with all associated phone numbers and email address. If not found it should give the user an error message and give the prompt again. If the response is a blank name input, application should go back to the main menu.

4.2.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

TAG	command line/ specific Menu of ABK	Command Name with Arguments	Semantics
RS_AB_01_08	ABK main menu	[D]	Prompt for contact name. If name is valid and found, ask confirmation and delete contact along with associated details. Else give suitable error and prompt for name again. If the response to a contact name prompt is a blank string, return to the main menu.

4.3 Search a Contact, by Name, Phone Number, E-mail Address

Using this feature the address book will be searched. If search is successful, the details can be viewed.

4.3.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

Priority: 9

1. It should be possible to search a contact:
   a) by name
   b) by phone number
   c) by email address
2. If found, the results will be displayed. Please note that multiple records are possible in case multiple contacts share phone numbers and/or email.

4.3.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

The application should be invoked and should be in the ABK main menu. Choosing the menu item code of [S] should start the interaction for searching a contact. Three menu options will then be provided: [C] for contact, [P] for phone number and [E] for email address. After the user chooses one of them and enters a value, the records which match them will be displayed. Please note that 0 or more records can be found.

4.3.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

TAG	Menu	Option Chosen	Action
RS_AB_01_09	ABK main menu	[S]	Move to search menu
RS_AB_01_10	Search menu	[C]	Prompt for contact name. Search by contact name provided. Display the single record if found and go to search details menu. If no contact found, report it and prompt again. If a response to prompt is a blank string, get back to search menu.
RS_AB_01_11	Search menu	[P]	Prompt for phone number. Search by phone number provided. Display all matching records. If no contact found, report it and prompt again. If a response to prompt is a blank string, get back to search menu.
RS_AB_01_12	Search menu	[E]	Prompt for email address. Search by email address provided. Display all matching records. If no contact found, report it and prompt again. If a response to prompt is a blank string, get back to search menu.
RS_AB_01_13	Search Menu	[X]	Back to ABK main menu

4.4 Edit Contact

This feature allows modification of contact name and add/modify/delete operations to details after searching contacts based on name, phone numbers and email addresses and choosing a single contact from them.

4.4.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

<u>Priority: 9</u>

1. It should be possible to view/modify a contact which is in the search result.
2. It should be possible to view/add/modify/delete a phone number.
3. It should be possible to view/add/modify/delete an email address.
4. If the operation is successful, give success message
5. Otherwise, give error message.
6. Stay in edit contact menu.
7. If X is chosen, go back to main menu.

4.4.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

The application should be invoked and should be in the [E] must be chosen from main menu. The edit menu will first ask for choosing the contact. Contacts can be searched using a sub-menu similar to search menu. The output of search will be 0 or more contacts. If 0 contact is found, repeat the search until user gives a blank string, in which case go back to edit menu.

If a single contact is available, go to edit menu. Here choose to modify contact name using [C], a phone number using [P] and an email address using [E]. Choosing [X] will take back to Edit contact menu.

4.4.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

TAG	command line/ Specific Menu of ABK	Option Chosen	Action
RS_AB_01_14	Edit contact	[C]	Prompt for contact name.

	menu		Search by contact name provided. Display the single record if found and go to choose contact menu. Else give error and prompt again. If a response to a prompt is a blank string, go back to edit contact menu.
RS_AB_01_15	Edit contact menu	[P]	Prompt for phone number. Search by phone number provided. Display all matching records and go to choose contact menu if multiple records are found. If single record is found, go to modifications menu. If no contact found, report it and prompt again. If a response to prompt is a blank string, get back to edit contact menu.
RS_AB_01_16	Edit contact menu	[E]	Prompt for email address. Search by email address provided. Display all matching records and go to choose contact menu if multiple records are found. If single record is found, go to modifications menu. If no contact found, report it and prompt again. If a response to prompt is a blank string, get back to edit contact menu.
RS_AB_01_17	Choose contact menu	List of contacts and their details is provided. The user should be asked to choose a serial number of a contact from this list.	If serial number provided is valid, remember the chosen contact and go to modifications menu. In case of any error, display the error and prompt for serial number again. If serial number is blank, return to edit menu
RS_AB_01_18	Modification s menu	[C]	Prompt for a new contact name. If this name is valid and does not clash with any other contact, change the name. Redisplay the chosen contact with all its details and in modification menu.

RS_AB_01_19	Modifications menu	[P]	Go to modify phone numbers menu
RS_AB_01_20	Modifications menu	[E]	Go to modify email addresses menu
RS_AB_01_21	Modifications menu	[X]	Return to choose contact menu.
RS_AB_01_22	Modify phone numbers menu	[A]	Ask for new number to add. Take in number. If valid and no conflicts for the chosen contact, add the number to the existing contact. Redisplay the modify phone numbers menu. If any error, display the message and prompt again. If response is blank string, return to modifications menu
RS_AB_01_23	Modify phone numbers menu	[D]	Ask for the serial number. If valid response given, confirm deletion and then delete the number. Re-display the modify phone numbers menu with the change. If any error, display the message and prompt again. If response is blank string, return to modifications menu.
RS_AB_01_24	Modify phone numbers menu	[E]	Prompt for a serial number. If response is fine, ask for the new phone number to replace the older value. If successful, confirm and replace. If any error, display the message and give two prompts again. If response is blank serial number or blank string, return to modifications menu.
RS_AB_01_25	Modify phone numbers menu	[X]	Return to modifications menu
RS_AB_01_26	Modify email addresss	[A]	Ask for new email address to add. Take in email address. If valid and no

	menu		conflicts for the chosen contact, add the email addrress to the existing contact. Redisplay the modify email addresss menu. If any error, display the message and prompt again. If response is blank string, return to modifications menu
RS_AB_01_27	Modify email addresss menu	[D]	Ask for the serial number. If valid response given, confirm deletion and then delete the email address. Re-display the modify email addresss menu with the change. If any error, display the message and prompt again. If response is blank string, return to modifications menu.
RS_AB_01_28	Modify email addresss menu	[E]	Prompt for a serial number. If response is fine, ask for the new email address to replace the older value. If successful, confirm and replace. If any error, display the message and give two prompts again. If response is blank serial number or blank string, return to modifications menu.
RS_AB_01_29	Modify email addresss menu	[X]	Return to modifications menu

4.5 List All the contact

Using this feature user can see / list all contact details present in the address book.

4.5.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

Priority: 9

1. It should be display all detail like Contact name , contact phone number and contact email address available in given address book.
2. If Enter key pressed then it should go to main menu.

4.5.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

The application should be invoked and should be in the [L] must be chosen from main menu. The list all menu will first display all contact detail in given address book. After displaying the details of all the user

4.5.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

TAG	command line/ specific Menu of ABK	Command Name with Arguments	Semantics
RS_AB_01_30	ABK main menu	[L]	Application should display all the contact with the details if no contact is entered means address book is empty then it should display the error message. After display the detail or error message by pressing enter key user can go to main menu.

Phase II Features:

These features will be postponed to Phase II. Hence these requirements need to be updated before starting dependent SDLC activities on Phase II.

4.6 Help Feature

4.6.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

Priority: 7

1. Using the command line as "ABK -h" should list all the commands and Menus and their expected usage.
2. The same should be invocable as help under any of ABK's nested Menus. This help will be a subset (context sensitive help) of the above, specific to the current Menu.

4.6.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

The application should be invoked and should be in the ABK Menu. ABK -h should display all the help for using the product. On ABK all the sub-Menus, the command, "hep" should also describe the help associated with the Menu.

4.6.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

4.7 Address Book Storage

This is the storage feature for the "Address Book" product.

4.7.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

Priority: 6 *[The priority is initially low, because, the developer must try to implement the other features first].*

1. The address book should save its data in a file with extension ".ABK" marked for it.
2. The file format should be easily human readable and editable.
3. It should be a CSV file which can be loaded into an eXcel sheet easily.
4. It should be possible to export the data into a new .ABX file.
5. It can import data from existing .ABK file.

6. However, if there is a name conflict, the importing should be aborted and the old .ABK should be restored.

4.7.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

The application should be invoked and should be in the ABK Menu. It will load the default .ABK file or a .ABK file specified on the command line. Before exiting, it should confirm from the user and save the file, if required.

4.7.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

TAG	Command line/ specific Menu of ABK	Option	Action
RS_AB_02_31	CL	ABK	Start the Application using the default data file.
RS_AB_02_32	CL	ABK -f <filename>.abk	Import all data from given .abk file. If failed report error and exit. If passed, move to operations loop. Use the same file for contact information for this run.
RS_AB_02_33	ABK	[I]mport <filename>.abk	Load the information in the file in addition to the existing contacts. Success if there are no conflicting contacts. Otherwise give error and rollback to situation before import. Stay on in ABK Menu.
RS_AB_02_34	ABK	[E]xport <filename>.abk	Export data to given file. Report error if it fails.

4.8 Group Feature

This feature helps in creating groups.

4.8.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.8.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

.

4.8.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.1.1 Quick Loading and Initial Response (timing to be decided)

RS_AB_02_XX:

The program must quickly respond to command line and load the .ABK file quickly.

5.1.2 Quick Response for Menus (timing to be decided)

RS_AB_02_XX:

The program must quickly respond to each command at all Menus.

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.2.1 Data File Safety

RS_AB_02_XX:

File corruption is possible if multiple users act on the same file. This should be avoided as far as possible. A locking measure can be provided by exclusive opening of file. If file does not open, it should give message and exit.

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.3.1 Address Book Information Security

RS_AB_02_XX:

The address information maintained by the end user should not be discovered by anyone without proper authentication/access.

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

1. SDLC will be using RDCT model.
2. Design should be modular.
3. At least about 4 independent modules should be developed in parallel.

4. Unit Testing should be done rigorously.
5. System Testing should be done.
6. Test Coverage should be above 95%.
7. Reviews will be done for all artifacts as per the model.

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

1. *SDLC – Software Development Life Cycle*
2. *SRS – **S**ystem **R**equirements **S**pecification*
3. *HLD – **H**igh **L**evel **D**esign (Specification Document)*
4. *ST – **S**ystem **T**esting (Specification and Test Cycles Document)*
5. *IT – **I**ntegration **T**esting (Specification and Test Cycles Document)*
6. *UT – **U**nit **T**esting (Specification and Test Cycles Document)*
7. *LLD – **L**ow **L**evel **D**esign (Specification Document)*
8. *.ABK – File extension for **A**ddress **B**oo**k** data file.*

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>