Software Requirements Document for a Personal Address and Phone Book

Part 2: Functional Requirements

Prepared for C-S 341: Software Engineering

By group #: 3

Group Members:

Tom Gendreau
Kasi Periyasamy
David Riley

Date: Sep 20, 2001

1. About this document

This document describes the functional requirements for the software for a personal address and phone book. The product overview and assumptions for the product are given in [1].

2. Functional Requirements

Each functional requirement is given in the following format:

Index:
Name:
Purpose:
Input parameters:
Action:
Output parameters:
Exceptions:
Remarks:
Cross-references:

Index refers to a unique index assigned to this functional requirement. It will be used for cross-referencing this requirement from other requirements and also from other documents such as the design document.

Name is a descriptive name given to this functional requirement. This name does not need to be unique.

Purpose is a short description (in a line or two) of the function. It is used to quickly understand the functionality and is also used to search the required function when all the functionalities are browsed through.

Input parameters refers to a set of parameters that the given function accepts as input. These parameters are required in order to design and implement the current function. No type information will be included for parameters in this document.

Output parameters refers to a set of parameters that are output/exhibit by the current function, when implemented. No type information will be included for parameters in this document.

Action refers to a set of tasks or activities that must be performed in order to satisfy this requirement. These tasks/activities are listed in no particular order. The design will take care of sequencing the tasks.

Exceptions refers to a set of conditions, each of which indicates a situation in which the function will stop. Notice that this column only lists a set of exceptions that might occur but does not suggest any action that must be taken when the exceptions occur. These actions will be included in the design document.

Remarks includes a set of comments that explain more about the functionality. It also describes hints to the designer and implementer that are suggested by the requirements analyst.

Cross-references refers to a set of other functional requirements that are related to the current function.

2.1 Functional Requirements for the Phone Book

Index: PB.1
Name: InitializePhoneDiary
Purpose: To initialize the phone diary with empty set of entries.
Input parameters: None.
Action: Initialize the phone diary with all the fields empty.
Output parameters: None.
Exceptions: None.
Remarks: This function may be used whenever the phone diary is to be reset. The designer may choose to have another function called "Reset Phone Diary" which may be linked to this function.
Cross-references: None.

Index: PB.2
Name: InitializeAppointmentCalendar
Purpose: To initialize the appointment calendar with empty set of entries.
Input parameters: None.
Action: Initialize the appointment calendar with all the fields empty.
Output parameters: None.
Exceptions: None.
Remarks: This function may be used whenever the appointment calendar is to be reset. The designer may choose to have another function called "Reset Appointment Calendar" which may be linked to this function.
Cross-references: None.

Index: PB.3
Name: SelectPhoneDiary
Purpose: To select the phone diary.
Input parameters: None.
Action: Make the phone diary active.

Output parameters: None.
Exceptions: None.
Remarks: This function will make the phone diary active. Perhaps, it will bring the entries into main memory for access.
Cross-references: None.

Index: PB.4
Name: SelectAppointmentCalendar
Purpose: To select the appointment calendar.
Input parameters: None.
Action: Make the appointment calendar active.
Output parameters: None.
Exceptions: None.
Remarks: This function will make the appointment calendar active. Perhaps, it will bring the entries into main memory for access.
Cross-references: None.

Index: PB.5
Name: WritePhoneDiary
Purpose: To write the entries in the phone diary back onto secondary storage.
Input parameters: None.
Action: Write back all entries of the phone diary onto secondary storage.
Output parameters: None.
Exceptions: None.
Remarks: This function will create a back up copy of the phone diary.
Cross-references: None.

Index: PB.6
Name: WriteAppointmentCalendar
Purpose: To write the entries in the appointment calendar back onto secondary storage.
Input parameters: None.
Action: Write back all entries of the appointment calendar onto secondary storage.
Output parameters: None.
Exceptions: None.
Remarks: This function will create a back up copy of the appointment calendar.
Cross-references: None.

2.2 Functional Requirements for the Phone Diary

Index: Ph.1
Name: AddEntry
Purpose: To add an entry into the phone diary.
Input parameters: last name, first name, address, a set of phone numbers
Action: Ensure that the last name does not exist in the phone diary.
Ensure that the phone numbers are classified into phone or fax numbers.
Ensure that each phone number is 7 to 21 digits long.
Update the phone diary by adding all the input entries
into the phone diary.
Output parameters: None
Exceptions: Last name in the input parameter already exists in the diary.
One or more phone numbers are not classified.
One or more phone numbers are not within range (have less than 7
digits or more than 21 digits).
Phone diary is full.
Remarks: The diary may be sorted on last names so that searching for the
names would be easier. Each phone number includes the number and the
classifier that distinguishes whether it is a phone number or a fax number.
The designer must choose appropriate format for implementing this.
Cross references: None

Index: Ph.2
Name: AddPhoneNumbers
Purpose: To add one or more phone numbers to an existing last name in the phone diary.
Input parameters: last name, a set of phone numbers
Action: Ensure that the last name exists in the phone diary.
Ensure that each phone number to be added is classified into phone or fax number.
Ensure that each phone number is between 7 to 21 digits long.
Ensure that each phone number is not already associated with the last name that is given in the parameter.
Update the phone diary by adding the new phone numbers to the last name.
Output parameters: None
Exceptions: Last name in the input parameter does not exist in the diary.
One or more phone numbers are already associated with the last name.
One or more phone numbers are not within the 7 to 21 digits range.
One or more phone numbers are not classified into phone or fax.
The phone diary is full.
Remarks: The diary may be sorted on last names so that searching for the
names would be easier.
The set of phone numbers entered so far may be stored in a separate table

along with the classification so that each new phone entry can be checked for conflicting classification. For example, one user may enter a number as "phone number" and another user might enter the same number as "fax number".

Cross references: None

Index: Ph.3
Name: DeleteEntry
Purpose: To delete a phone entry.
Input parameters: last name
Action: Ensure that the last name already exists in the phone diary.
Update the phone diary by deleting the last name and all associated information (first name, address and phone numbers) from the phone diary.
Output parameters: None
Exceptions: Last name in the input parameter does not exist in the diary.
Remarks: The diary may be sorted on last names so that searching for the names would be easier.
Cross references: None

Index: Ph.4
Name: DeletePhone
Purpose: To delete a phone number that is associated with a given last name.
Input parameters: last name, phone number
Action: Ensure that the last name already exists in the phone diary.
Ensure that the phone number already exists in the diary and is associated with the last name given as the parameter.
Update the phone diary by deleting the phone number from the entry corresponding to the given last name.
Output parameters: None
Exceptions: Last name in the input parameter does not exist in the phone diary.
The given phone number is not associated with the given last name.
Remarks: The phone diary may be sorted on last names so that searching for the names would be easier.
Cross references: None

Index: Ph.5
Name: ModifyEntry
Purpose: To modify an entry in the phone diary.
Input parameters: last name, address, a set of phone numbers
Action: Ensure that the last name already exists in the phone diary.
Ensure that each phone number in the *input parameters* is in the range of 7 to 21 digits.

Ensure that each phone number in the input *parameters* is classified as phone or fax.
Update the phone diary by overwriting the new address corresponding to the last name in the phone diary. Update the phone numbers if they are new. If a number already exists or conflicts with a number that is previously stored, ignore that number.

Output parameters: None

Exceptions: Last name in the input parameter does not exist in the diary.
One or more phone numbers is out of the 7-21 digits range.
One or more phone numbers is not classified as phone or fax.

Remarks: The phone diary can be sorted on last names so that searching for the names would be easier.
If the *input parameters* contain phone numbers that are already associated with the given last name, or conflict with the phone numbers that are already associated with the given last name, this function will not generate exceptions. Rather, it simply ignores those phone numbers.
A conflict between two phone numbers occurs when they have the same number but is classified differently (one is classified as a phone and the other as a fax number).
Not all fields in the *input parameters* listed above are required to modify an entry. If not provided, those fields are assumed to be empty. Thus, this function may be used to modify address or phone numbers alone.
If only new phone numbers added, this function may make use of Ph.2.

Cross references: Ph.2

Index: Ph.6
Name: SearchEntry
Purpose: To search and retrieve information corresponding to a last name.
Input parameters: last name
Action: Ensure that the last name already exists in the phone diary.
Retrieve and display all the information associated with this last name.
Output parameters: All information (last name, first name, address and phone numbers) associated with the given last name.
Exceptions: Last name in the *input parameters* does not exist in the diary.
Remarks: The diary may be sorted on last names so that searching for the names would be easier.
Cross references: None

Index: Ph.7
Name: ListEntries
Purpose: To list all entries for a given phone number.
Input parameters: phone number
Action: Ensure that the phone number is between 7 to 21 digits.
Select and display all the entries in the phone diary in which the given

phone number appears.

Output parameters: A set of entries.

Exceptions: The given phone number is not between 7 to 21 digits.

Remarks: This function may display almost all entries in the phone diary. The designer may choose appropriate display options (one entry at a time, or a page of entries at a time and so on).
The given number may be a phone number or fax number; this function does not distinguish between the two.

Cross-references: None.

2.3 Functional Requirements for the Appointment Calendar

Index: AC.1
Name: AddApointment
Purpose: To add a new appointment into the appointment calendar.
Input parameters: date, time, appointment, last name
Action: Ensure that date and time as specified in the input are valid.
Ensure that the date is not beyond a year from the current date.
Ensure that the date is not in the past.
Ensure that there is no appointment scheduled at this date & time.
Update the appointment calendar by including the new date, time and appointment along with the last name, if the last name is specified in the input.
Output parameters: None.
Exceptions: Date and/or time is invalid.
Date is beyond a year from the current date.
Date is in the past.
Another appointment already exists at this date and time.
Appointment calendar is full.
No entry corresponding to the name exists in the phone diary.
Remarks: The phone diary may be sorted on date and time so that searching for the entries would be easier.
Cross references: None

Index: AC.2
Name: DeleteAppointment
Purpose: To delete an appointment from the appointment calendar.
Input parameters: date, time
Action: Ensure that the date and time are valid.
Ensure that the date is not beyond a year from the current date.
Ensure that the date is not in the past.
Ensure that there exists an appointment already in the appointment calendar at this date and time.
Update the appointment calendar by deleting all information at the

specified date and time.
Output parameters: None.
Exceptions: Date and/or time is invalid.
Date is beyond a year from the current date.
Date is in the past.
No appointment exists at this date and time.
Remarks: The phone diary may be sorted on date and time so that searching for the entries would be easier.
Cross references: None

Index: AC.3
Name: ModifyAppointment
Purpose: To modify an appointment in the appointment calendar.
Input parameters: date, time, new appointment, last name
Action: Ensure that date and time are valid.
Ensure the date is not beyond a year from the current date.
Ensure the date is not in the past.
Ensure that there exists an appointment already in the appointment calendar at this date and time.
Update the appointment calendar by overwriting the appointment at this date and time with the new appointment and the last name, if specified, in the input.
Output parameters: None.
Exceptions: Date and/or time is invalid.
Date is beyond a year from the current date.
Date is in the past.
No appointment exists at this date and time.
Remarks: The phone diary may be sorted on date and time so that searching for the entries would be easier.
Cross references: None

Index: AC.4
Name: SearchAppointment
Purpose: To search an appointment in the appointment calendar.
Input parameters: date, time
Action: Ensure that date and time are valid.
Ensure that the date is not beyond a year from the current date.
Ensure that the date is not in the past.
Ensure that there exists an appointment already in the appointment calendar at this date and time
Display all the information in the entry corresponding to this date & time.
Output parameters: Appointment and a last name, if present in the appointment.
Exceptions: Date and/or time is invalid.
Date is beyond a year from the current date.

Date is in the past.
No appointment exists at this date and time.

Remarks: The phone diary can be sorted on date and time so that searching for the entries would be easier.

Cross-references: None.

Index: AC.5

Name: MoveAppointment

Purpose: To move an appointment within the appointment calendar.

Input parameters: old date, old time, new date, new time

Action: Ensure that old date, new date, old time and new time are valid.
Ensure that both old date and new date are not beyond a year of the current date.
Ensure that both old date and new date are not in the past.
Ensure that there exists an appointment already in the appointment calendar at the old date and old time.
Ensure that there is no appointment at the new date and new time.
Update the appointment calendar by writing the appointment at the new date and new time and deleting the appointment at the old date and old time. If the appointment has a last name before, it should also be moved.

Output parameters: None.

Exceptions: Date and/or time is invalid.
Old date or new date is beyond a year from the current date.
Old date or new date is in the past.
No appointment exists at the old date and old time.
Another appointment already exists at the new date and new time.

Remarks: The phone diary can be sorted on date and time so that searching for the entries would be easier.
This function can be achieved by concatenating search, add and delete.

Cross references: AC.3, AC.2 and AC.1

References

1. Kasi Periyasamy, Dave Riley and Tom Gendreau, “Software Requirements Document for a Personal Address and Phone Book – Part 1: Product Overview and Assumptions”, September 2003.