Software Requirements Specification

for

Holiday Mailer

Version 1.0

Prepared by Chris Park

Team: Bits Be Flippin'
Chris Park
Nathan Pilgrim
Sami Awwad

November 5, 2014

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Chris Park	10/5/14	initial draft	1.0 draft 1

1.0 Introduction

1.1 Goals and objectives
This SRS describes the functionality requirements for version 1.0 of the Holiday Mailer (HM). This document is to be used by the project team members that will implement and verify the functionality of the software.

1.2 Project scope and features
The Holiday Mailer will allow a user to send a friendly holiday greeting email to any of the contacts available within the database maintained by the software.

1.3 Context
The Holiday Mailer is intended for use by anyone running a windows operating system, with an email address and access to the internet.

2.0 Usage Scenario

2.1 User profiles
User:
A user is any operator of the software who has added their name and email address to the program.

2.2 Use-cases
Add additional as necessary

Add Contact:
Opens a dialog allowing the user to enter the Name and Email address of a contact to add to the database.

Edit Contacts:
Opens a dialog populated by the selected list of contacts from the main form. The user may click a contact in the list to populate the text box fields with that contacts information. Information for that contact may then be edited and changes may be confirmed and added back into the database.

Remove Contacts:
Any contacts selected within the main form will be removed upon clicking the remove button.

Add User:
Opens a dialog allowing a user to enter their Name and Email address to be added to the list of users.

Send Email:
Sends a holiday email to all selected contacts in the list. Options to send to all contacts and contacts by previously received email will also be available
Sort Contacts:
While the contact list is displayed on the GUI, sorting by last name, or listing contacts by specific first letters of last name must be available.

2.3 Special Requirements
Add as available

3.0 Data Model and Description

3.1 Class/Data objects
The Holiday mailer will contain the following:

- A SQLite database for storage of contacts.
- A DatabaseManager class that creates, maintains, and allows changes to the SQLite database of contacts.
- A Contact class that holds information about a contact from the SQLite database.
- A User class that holds information about the person sending emails to contacts in the database.

3.2 Relationships
Insert UML diagram here.

4.0 Model and Description

4.1 Features
Add additional features as they become available

- The Holiday greeting will be provided as a template with contact and user information inserted where applicable.

- A user will be able to attach and remove files to an email as necessary.

4.2 Class descriptions
Flesh out detailed information about the fields and methods of each class here

Class	Fields	Methods
DatabaseManager	*Add as created*	*Add as created*

Contact	➓ First Name ➓ Last Name ➓ Email Address ➓ Previous Email Received (Boolean)	*Add as created*
User	➓ First Name ➓ Last Name ➓ Email Address	*Add as created*

4.3 User interface (GUI)

Form	Contains
Main *Add Thumbnail Image*	➓ List of Contacts ➓ Sortable by first name, last name ➓ List by last initial ➓ Email preview ➓ Context buttons: ➓ Add contact ➓ Remove contact ➓ Add user ➓ Attach file ➓ Send emails
Add Contact *Add Thumbnail Image*	➓ Field for first name ➓ Field for last name ➓ Field for email address ➓ Context Buttons ➓ Add ➓ Cancel
Add User *Add Thumbnail Image*	➓ Field for first name ➓ Field for last name ➓ Field for email address ➓ Context buttons: ➓ Add ➓ Cancel
Edit Contacts *Add Thumbnail Image*	➓ Field for first name ➓ Field for last name ➓ Field for email address ➓ List of selected contacts
Attach File *Add Thumbnail Image*	➓ Field for file path ➓ Context buttons: ➓ Browse ➓ Attach ➓ Cancel

5.0 Behavioral Model and Description
Flesh out as development progresses

6.0 Restrictions, Limitations and Constraints
Add additional constraints as development progresses

The DatabaseManager class will provide all necessary information to the User and Contact classes which will be used to display and provide information for the user to interact with inside of the GUI interface.

Any changes or additions to the contacts will be updated by the DatabaseManager via the User and Contact classes.

7.0 Validation Criteria
Add additional criteria as development progresses

7.1 Classes of tests
Tests will either be presented as mock mains or unit tests which target the specific functionality of a feature.

7.2 Expected software response
All tests are expected to pass in so far as they are relevant to a given feature. Every attempt should be made to eliminate bugs as they appear, before committing changes to the repository.

8.0 Appendices
Add as becomes available