CSCD350, Holiday Mailer: Software Requirements Specification

Steve Berg, Zach Lesperance, Tim Lynch, Steven Mather

November 7, 2014

1 Introduction

1.1 Goals and objectives

To build a program that stores addresses of friends and relatives and allows the user to send a 'Holiday Email' to people of her/his choice.

1.2 Software context

Consumer grade product that is intuitive for use by the average computer user. The software handles their 'Holiday Letters' for them electronically.

1.3 Major constraints

The software will need to be able to run minimally on a consumer grade machine so as to not noticeably impact performance. Web connection while running to allow access to the external e-mail servers.

1.4 Use-cases

Use case Add new person to database

Actor user
Basic choose to enter new person, enter first name, enter last name, enter address, enter if and email was sent, submit to database

Use case Remove person from database

Actor user
Basic choose to remove, choose who to remove, make sure, remove

Use case Display entries by last name

Actor user
Basic choose to display, choose to display by last name, display

Use case Display entries by first name

Actor user
Basic choose to display, choose to display by first name, display

Use case Display entries by last received date

Actor user
Basic choose to display, choose to display by last received date, display

Use case Display entries with last name with certain letter

Actor user
Basic choose to display, choose to display by certain letters, display

Use case send email to everyone

Actor user
Basic choose to send email to everyone, pick holiday email to send, send

Use case send email to everyone who has sent one

Actor user
Basic choose to send email to everyone who has sent you one, pick holiday email to send, send

Use case send email to specific people

Actor user
Basic choose to send email to specific people, pick people, pick holiday email to send, send

2 Data Model and Description

For this project, we will be using SQLite technology for our data object interactions.

2.1 Data Description

2.1.1 Data objects

Name: contacts
Description: A table that stores all of the user's contacts and their information
Fields:

- email (varchar(50)): The contact's email address
- firstName (varchar(50)): The contact's first name
- lastName (varchar(50)): The contact's surname
- lastReceivedDate (date): The date that the contact last sent a holiday letter to the user

2.1.2 Complete data model

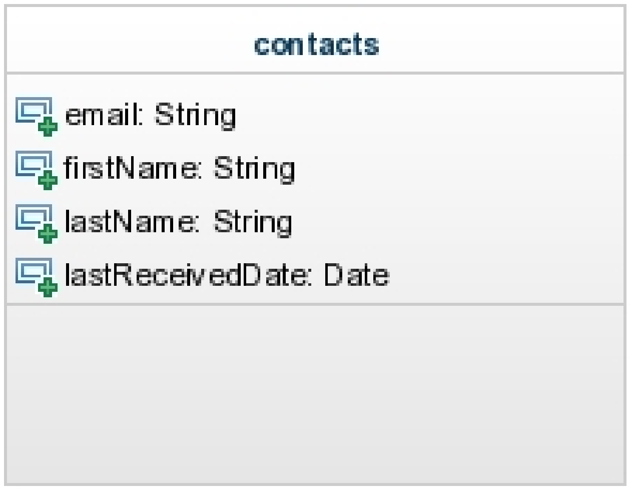

Figure 1: Entity-Relation Diagram for our Data Model

3 Functional Model and Description

The Client class is the main anchor for the program.
It contains references to two objects that handle all input from the user and all output to the user.
The client will also contain the methods needed to delegate work to the proper objects, i.e. contact creation, contact modification, contact deletion, display

The ContactFactory object will be used to create contacts and will become more complex as as the contacts become more complex.
The DBAccess object will be used for all interaction with the database.
The contact class is an object that represents tuples in the database.
The userOut and userIn objects handle and validate the interactions the program has with the user.

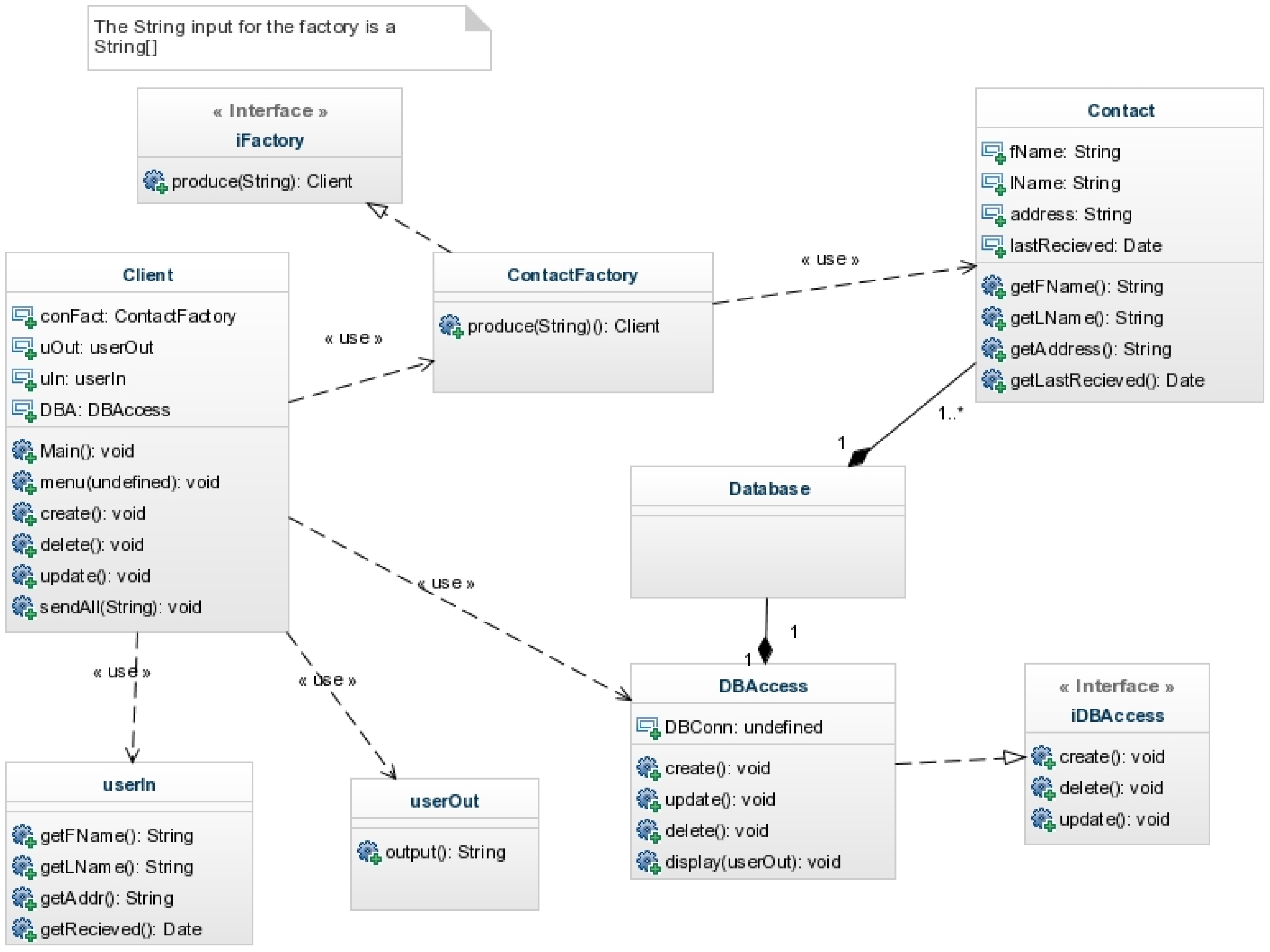

Figure 2: UML Diagram of the Software

3.1 Software Interface Description

3.1.1 Human interface

A command line menu driven interface will initially be built and if time permits and we make all the properties that we want, a graphical user interface will be build.

4 Validation Criteria

4.1 Classes of tests

We will be running JUnit tests against each iteration of our code to make sure that our program stays consistent.

JUnit will be testing all CRUD (Create Update Delete) methods, database interation, data validation, etc methods.