Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to provide a deeper understanding into the corresponding components intended use, functionality and requirements.

1.2. Intended Audience

This project is a test of the creator’s ability to accurately implement the requested functionalities as well as properly document and test his work in the view of Asymmetrik recruiters. More generally, the component is meant for use by users of the business card optical character recognition.

1.3. Product Scope

The product parses an individual's primary contact information from the data presented allowing for quicker access to that data otherwise facilitating the removal of otherwise unnecessary data for what could be later storage.

1.4. Definitions

UML: unified modeling language
OCR: optical character recognition

1.5. References

1. Email Regex Guide: https://www.regular-expressions.info/email.html
2. Phone Number Regex Guide: https://en.wikipedia.org/wiki/E.164

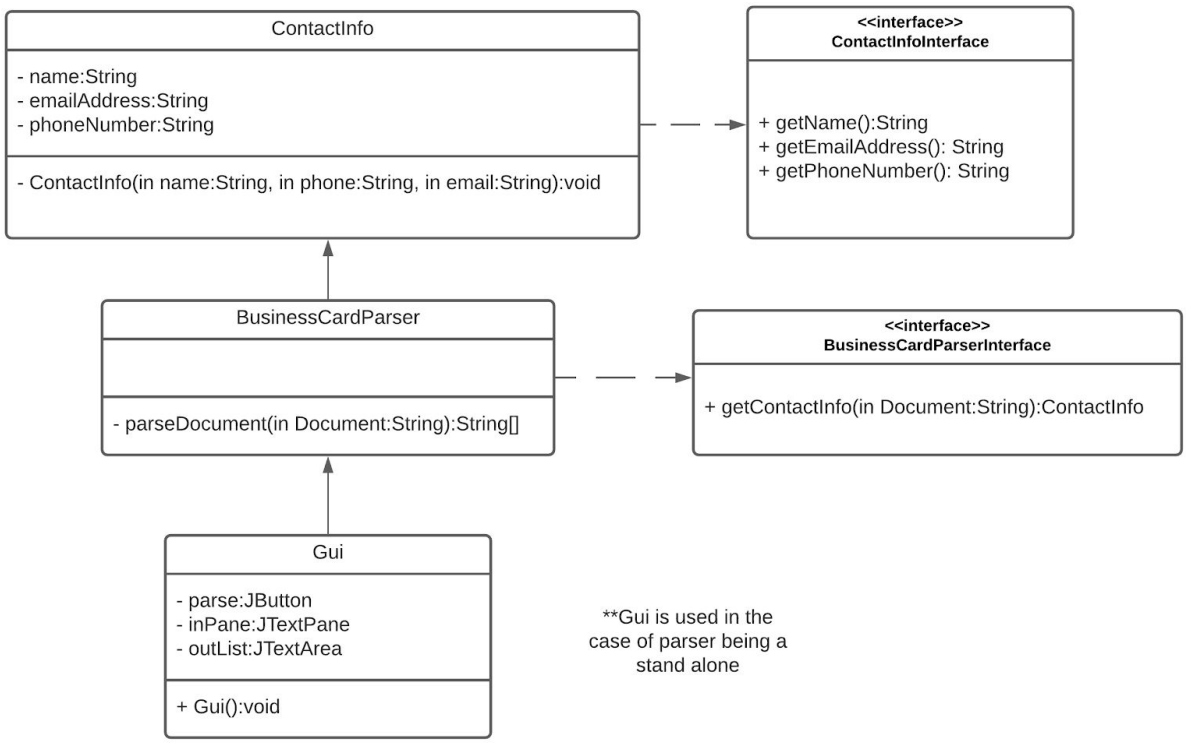
2. Overall Description

2.1. Product Perspective

The Component is a behind the scenes parser which returns the contact information of the business card subject. This contact information includes subject name, phone number and email address

2.2. Product Features

The primary feature/s of the component is shown in the UML class diagram below



2.3. User Class and Characteristics

This component is not directly interacted with by a user when used in conjunction with the business card OCR. A user of the overall system, through aid of this component, is able to retrieve the name, phone number and email address of the business cards' subject. As packaged, the component comes with a gui allowing for the input of the data as a means of using it as a standalone. The component supports one primary user who's primary action is as follows:

- Get contact info:
    - Name
    - Email
    - Phone number

2.4. Assumptions and Dependencies

This component is assumed to be a part of the larger business card OCR and is only to be used in to following application:

- Requesting scanned business card contact information

The data passed in is assumed to be business card related meaning it is expected to have the information the component is meant to parse and return i.e. a name, a phone number and a business email.

Email:

Formatted such that it consists of a partial of the card subject's name, followed by the businesses' domain where the two (person's name and business name) are distinguishable. Each potential heading is assumed to be on it's own line. The distinguishability of contact name and business name does not directly affect email retrieval. It is assumed that only one email address is provided (affects retrieval of Contact name). *(see Reference 1 for regex guide)*

Name:

Given that there is no set format for a name some rule derived from the more distinguishable email address is used. Given that the first portion of a business email consists of a partial of the contact name, comparing the two can be used as an identifier. Complications arise if another field i.e. partial name from the email also matches the business name. In this case no name is returned.

Phone:

Phone numbers are denoted by all consecutive numeric entries greater than 6 and at most length 15. To handle the possibility of fax number, the key word "fax" is used as an identifier. In the case that such key word is not provided, the last choice resembling a phone number is returned. The regex used was custom based on reading from ITU-T Recommendation E.164. It allows for the presence of non ASCII characters between digit to support multiple formats*(see Reference 2 for regex guide)*

Any field not returned is due to the field information being missing from the card or multiple fields' data entries mimicking the format of an expected field.

3. System Features

3.1. Functional Requirements

- Java Runtime Environment: JavaSE-1.8

- Apache Maven 3.6.3

4. External Interface Requirements

4.1. User Interfaces

- Back-software: Java, Apache Maven
- Optional Front-end: Java

4.2. Hardware Interfaces

- Component: depends on Business OCR
- Standalone: Computer System (i.e. mouse, keyboard, monitor, etc.)

4.3. Software Interfaces

- Operating system: Functionally on most if not all platforms supported by Java

5. Nonfunctional Requirements

5.1. Security Requirements

The component is not in charge of storing the data long term and as such the task of overarching data security is up to the system

5.2. Software Quality Attributes

- **Accuracy:** Given the proper input of data the requested contact information is always retrieved
- **Mobility:** This is a standalone component and is therefore able to perform is designed function in any Java supported system given the correct input of data
- **Extensibility:** Given more rules for parsing, the component is easily extendable
- **Usability:** The contact information is returned given that it is available. User is notified about any input errors