|  |
| --- |
| **Vaccination Culture Calculator Software Requirements Specification [Front End] Version 1.3 October 22th 2019  Ned Sherman & Mason** **Baird** |

Table of Contents

1.0 Introduction2

1.1 Purpose2

1.2 Definitions, Acronyms, & Abbreviations 2

1.3 System Overview 2

2.0 Functional Description3

2.1 Overview3

2.2 Main Panel3

2.3 Users Panel4

2.4 Insert User Panel4

2.5 Classes5

2.6 Possible Method Headings5

2.7 Error Handling6

3.0 System Requirements6

3.1 Hardware Requirements6

3.2 Software Requirements6

4.0 Miscellaneous6

4.1 Contact6

4.2 GitHub6

4.3 Citations6

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author(s)** |
| Oct 1st 2019 | 0.1 | Initial Draft | Mason Baird, Ned Sherman |
| Oct 2nd 2019 | 1.0 | Completion of all sections for current vision of the application, Table of contents | Mason Baird, Ned Sherman |
| Oct 8th 2019 | 1.1 | Revisions | Mason Baird, Ned Sherman |
| Oct 10th 2019 | 1.2 | Redesign | Mason Baird, Ned Sherman |
| Oct 22nd 2019 | 1.3 | Restructured | Mason Baird, Ned Sherman |

**1. Introduction**

The United Nations third Goal of Sustainable Development, “Good Health & Well-Being”, prompts people to vaccinate themselves and their family members. Vaccine preventable diseases are a cause of death even in countries that used to have deadly diseases under control. Tuberculosis, pertussis, diphtheria, and measles are diseases that still cause deaths in the United States [1]. In some cases these deaths are a result of the population’s inaccessibility to vaccination but there are also cases of voluntary refusal to vaccinate leading to death. If understanding of the safety and importance of vaccination becomes more widespread fewer deaths might be caused by vaccine preventable diseases.

**1.1 Purpose**

The Vaccination Culture Calculator will be a tool that shows a countries acceptance of vaccination so that users can educate themselves on the vaccination stance of the supplied country. The front end team’s purpose will be to make a GUI that is capable of taking a country as input and displaying information about the supplied country. The front end will also need to be prepared to accept new records for the Users table which is a record of all the tools past users.

**1.2 Definitions, Acronyms, and Abbreviations**

**Front End** – a software interface (such as a graphical user interface) designed to enable user-friendly interaction with a database

**Back End** - core computational logic of a website or database.

**VCC** – Vaccination Culture Calculator

**1.3 System Overview**

The completed Vaccination Culture Calculator should be able to take a country as input and output a summary of the selected country’s stance on vaccination so users can educate themselves on different countries. The VCC should also be able to add records to the Users table on the back end via JTextFields.

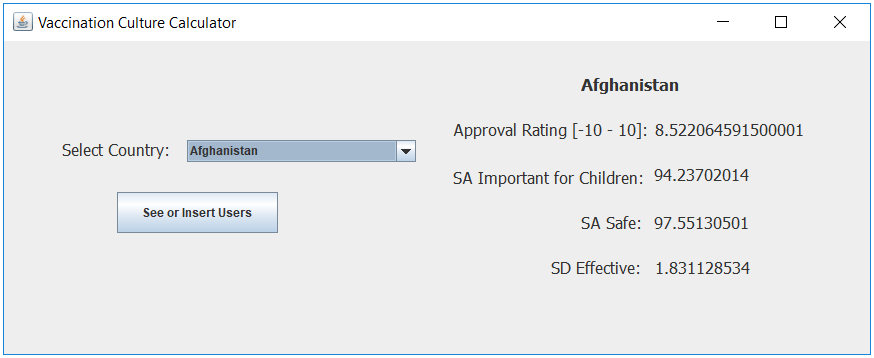
**2. Functional Description**

* 1. **Overview**

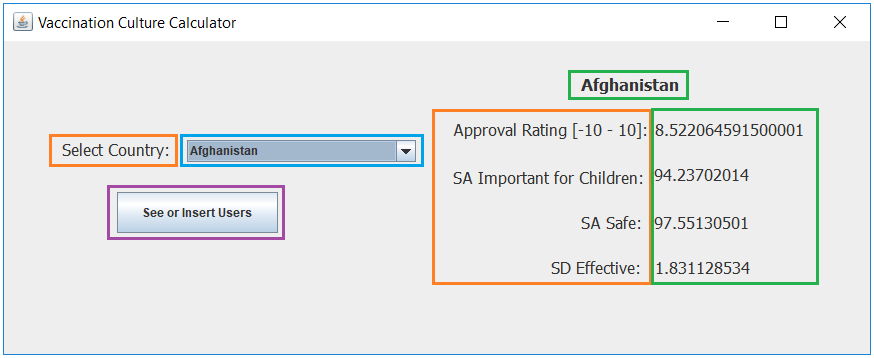
The VCC exists across 3 absolute layout JPanels that are on a single contentPane. The contentPane is set to card layout on a single JFrame. The 3 JPanels are: Main, Users, and InsertUser.

* 1. **Main Panel**

The Main Panel looks similar to the below mock up:

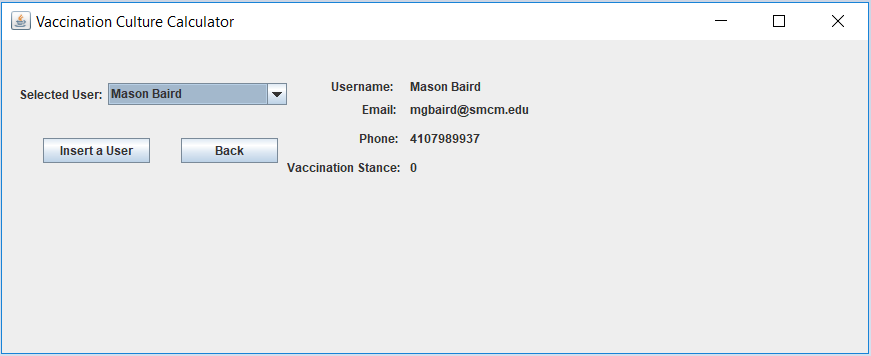


On the Main Panel there is:

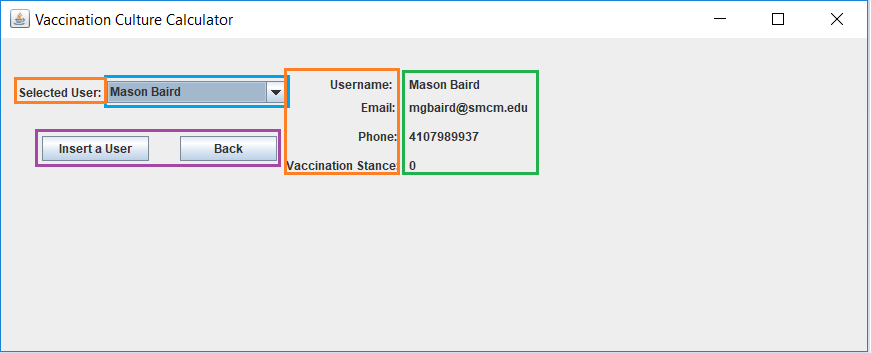


* [Blue] A JcomboBox country selection menu. The comboBox should be dynamically programmed to populate with the country data on the back end.
* [Orange] 5 static JLabels. The 5 static JLabels should be easy to implement. Just create the 5 JLabels and put them in their general location.
* [Green] 5 dynamic JLabels that should be programmed to update on the countryComboBox selection change event to reflect the newly selected countries data.
* [Purple] A Jbutton that should adjust the panels visibilities such that the Users Panel is displayed.
  1. **Users Panel**

The Users Panel should look similar to how it does in the below mock up:

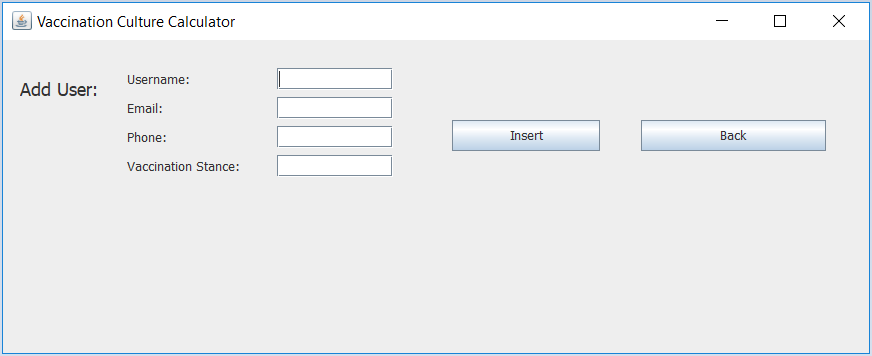


On the Users Panel there is:

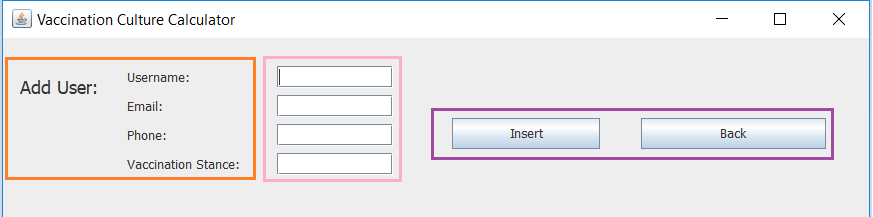


* [Blue] A JcomboBox user selection menu. The comboBox should be dynamically programmed to populate with the user data on the back end.
* [Orange] 5 static JLabels. The 6 static JLabels should be easy to implement. Just create the 5 JLabels and put them in their general location.
* [Green ] 4 dynamic JLabels that should be programed to update on the userComboBox selection change event to reflect the newly selected user’s data.
* 2 JButtons (boxed in purple). One to link to the Insert User Panel and one to return the user to the Main Panel.
  1. **Insert User Panel**

The Insert User Panel should look similar to how it does in the below mock up:



On the Insert User Panel there is:



* [Orange] 5 static JLabels as shown.
* [Pink] 4 JTextFields to take in the values to be inserted on the back end as Strings.
* [Purple] 2 JButtons. One to execute the insert statement and the second to send the user back the Users Panel.
  1. **Classes**

The entire program can be made with two classes although there is no restriction on the classes used. In my implementation example I used a *Main* class to house the UI elements and a *Queries* class to hold the methods that query the back end.

* 1. **Possible Method Headings**

To fetch data that will populate the dynamic country labels:

**public** ArrayList<String> fetchCRecord(String country){}

To fetch the countries for the country ComboBox:

**public** ArrayList<String> populateCountryCombo(){}

To fetch the Usernames for the Users ComboBox:

**public** ArrayList<String> populateUsersCombo(){}

To fetch the data that will populate the dynamic user labels:

**public** ArrayList<String> fetchURecord(String username){}

To create and send off insert statement to back end:

**public** **void** insertUserRecord(String Username, String Email, String Phone, String Stance){}

* 1. **Error Handling**

The VCC should be developed error free. Navigation between the panels should be functional. The JTextFields should be able to accept any values. All dynamic labels should update correctly. The users JComboBox should update if a new user is entered into the database.

**3. System Requirements**

The Vaccination Culture Calculator will not require any specific hardware beyond that of a standard computer. The front end development will require the machine to be equipped with the software to develop and run Java application.

**3.1 Hardware Requirements**

The application will not be hardware demanding and will not require any specific hardware. The application will be suitable to run on any machine with baseline hardware.

**3.2 Software Requirements**

The front end development will require Java and the java UI swing packages.

**4. Miscellaneous**

|  |  |  |
| --- | --- | --- |
| Name | Slack Handle | Email |
| Mason Baird | Mason Baird | [mgbaird@smcm.edu](mailto:mgbaird@smcm.edu) |
| Ned Sherman | Edmund Sherman | easherman@smcm.edu |

**4.1 Contact**

**4.2 GitHub**

The artifacts pertaining to this project are available at: <https://github.com/MasonGBaird/450MajorProject>

**4.3 Citations**

[1] Our World in Data. (2017). deaths-caused-by-vaccine-preventable-diseases Retrieved from <https://ourworldindata.org/vaccination>