**Software Design Documentation**

# VaqPack (Database)

Version 1.0

November 30, 2016

## Lead Software Engineer: Elijah Lopez

**Project Team:**

Michelle Garcia  
Jose Rodriguez  
Juan Delgado

Prepared for

Software Engineering

University of Texas Rio Grande Valley Instructor: MK Quweider, Ph.D.

Fall 2016

## Table of Contents

1. **INTRODUCTION**……………………………………………………………..**1** 1.1 PURPOSE………………………………………………………………………….1
   1. SCOPE………………………………………………………………………....…..1
   2. DEFINITIONS, ACRONYMS, AND ABBREVIATIONS……………………….1

1. **REFERENCES**………………………………………………………………**3**

1. **DETAILED DESIGN**………………………………………………………..…4

3.1 MODULE DETAILED DESIGN.................................................................................4

3.1.01: DBOperations.java....................................................................................................4

3.1.02: User.java.....................................................................................................................5

3.1.03: Utilities.java...............................................................................................................6

3.2 DATA DETAILED DESIGN……………………………...……………..…….28

* + 1. User Access Level...................................................................................................28
    2. Business Card Information....................................................................................29
    3. Business Card PDF.................................................................................................29
    4. Contact Information................................................................................................29
    5. Cover Letter Information........................................................................................29
    6. Cover Letter PDF.........................................................................................30
    7. Custom Theme.......................................................................................................30
    8. Registering User...................................................................................................30
    9. Reset Code...........................................................................................................30
    10. Resume Information............................................................................................31
    11. Resume HTML...................................................................................................31
    12. Resume PDF.........................................................................................................31
    13. User.......................................................................................................................31
    14. User Data..............................................................................................................32

### 4. USER INTERFACES BY USE CASE………………………………………33

4.01 Case01-Initial Administrator System Configuration...........................................35

4.02 Case02-Administrator System Configuration Password Mismatch....................36

4.03 Case03-User Creates New Account....................................................................37

4.04 Case04-User Login...........................................................................................38

4.05 Case05-Update Personal Information..............................................................39

4.06 Case06-Update Personal Information-Required field......................................40

4.07 Case07-Update Personal Information-Incorrect format...................................41

4.08 Case08-User Forgets Password.......................................................................42

4.09 Case09-New Password Not Strong Enough....................................................44

4.10 Case10-Inactivity Timeout……………….......................................................45

#### A. APPENDICES

A.1 Appendix 1........................................................................................................

A.2 Appendix 2........................................................................................................

### 1. Introduction

#### 1.1 Purpose

This Software Design Document is made with the purpose of explicitly outlining the software architecture and high level design of the database portion of the VaqPack application. The intention of this document is to provide developers an insight into utilizing the database to its fullest ability to serve each and every user, corresponding with the requirements set forth in the SRS. Therefore, this document is mainly intended for the developers, present company, as well as future organizations.

#### 1.2 Scope

The software application described throughout this SDD is the only the database portion of the VacPack applications. This SDD is intended for a base level system in order to provide a proof of concept for the use of building an evolutionary prototype that demonstrates the functionality specified by the corresponding SRS. This will be achieved through the use of use-case models, state models, class models and data flow models that will serve to clarify the development teams thought process during implementation.

#### 1.3 Definitions, Acronyms, and Abbreviations

The following terms, acronyms, and abbreviations are used throughout this document and are presented in the table below by order of appearance.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| SDD | Software Design Description; A written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. |
| SRS | Software Requirement Specification; A comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform. |
| VaqPack | VaqPack Graduate to Professional Aid Pack, in short. |
| GUI | Graphical User Interface; provides a visual, interactive means for a software user to manipulate the controls, commands, or features of that software. |
| Database | A structured collection of data that can be efficiently and conveniently accessed. |
| JVM | Java Virtual Machine; Provides the necessary links allowing a java program to run on a machine using a particular operating system. |
| JRE | Java Runtime Environment; Including the Java Virtual Machine, all necessary components for a system to establish the environment in which Java programs will run. |
| DBA | Directs or performs all activities related to maintaining a successful database environment. |
| SQL | Structured Query Language; the standard relational database query language. |
| JDBC | Java Database Connectivity; a Java API developed by Oracle Corporation which provides methods for querying and updated a database. |
| XML | Extensible Markup Language; a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. |
| HTML | HyperText Markup Language; the web standard language used in the delivery of online content, interpreted and rendered by web browsers. |
| PDF | Portable Document Format; a popular electronic document file type particularly used with rich-text or styled text. |

### 2. References

Git -<https://git-scm.com/>

GitHub -<https://github.com/>

Java Virtual Machine -<https://java.com/en/download/>

Java Runtime Environment - [http://www.oracle.com/technetwork/java/javase/downloads/jre8downloads-2133155.html](http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html)

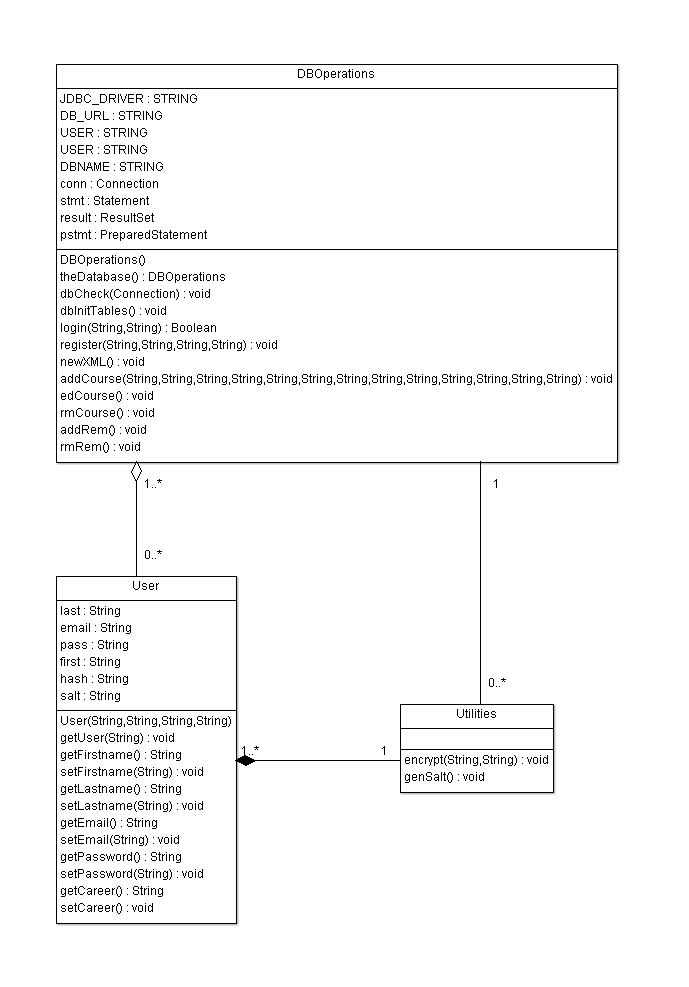
MySQL -<http://dev.mysql.com/downloads/mysql/>

NetBeans -<https://netbeans.org/>

JDBC - http://www.oracle.com/technetwork/java/javase/jdbc/index.html

### 3. Detailed Design

#### 3.1 Module Detailed Design

****

##### 3.1.01: DBOperations.java

|  |  |
| --- | --- |
| **Class Name:** DBOperations | |
| **Brief description:** The DBOperations class is responsible for building and maintaining the database. | |
| **Attributes (fields)** | **Attribute Description** |
| String JDBC\_DRIVER; | This is a variable used to identify the JDBC driver. |
| String DB\_URL; | This is a variable used to identify the database URL. |
| String USER; | This is a variable used to identify the username. |
| String PASS; | This is a variable used to identify the user’s password |
| String DBNAME; | This is a variable used to identify the database name. |
| Connection conn; | This is a declaration of a Connection datatype to be used throughout the class. |
| Statement stmt; | This is a declaration of a Statement datatype to be used throughout the class. |
| ResultSet result; | This is a declaration of a ResultSet datatype to be used throughout the class. |
| PreparedStatement pstmt; | This is a declaration of a PreparedStatement datatype to be used throughout the class. |
|  |  |
| **Methods (operations)** | **Method Description** |
| DBOperations() | This is a constructor method. |
| theDatabase() | This is a method which returns the database object, which is a singleton. |
| dbCheck(Connection) | This is a method that accepts a parameter of Connection datatype and will check if the database exists. |
| dbInit(Connection) | This is a method that accepts a parameter of Connection datatype and initializes the database for the VaqPaq application. |
| dbInitTables() | This is a method that initializes database tables. |
| login(String, String) | This is a method that accepts two String parameters for logging a user in. |
| register(String, String, String, String) | This is a method that accepts four String parameters for registering a user. |
| newXML() | This a method that extracts information from new XML files. |
| addCourse(String, String, String, String, String, String, String, String, String, String, String, String, String) | This is a method that accepts 13 String parameters for adding a new course to the database. |
| edCourse() | This is a method that edits a course. |
| rmCourse() | This is a method that removes a course. |
| addRem() | This is a method that adds a reminder. |
| rmRem() | This is a method that removes a reminder. |

##### 3.1.02: User.java

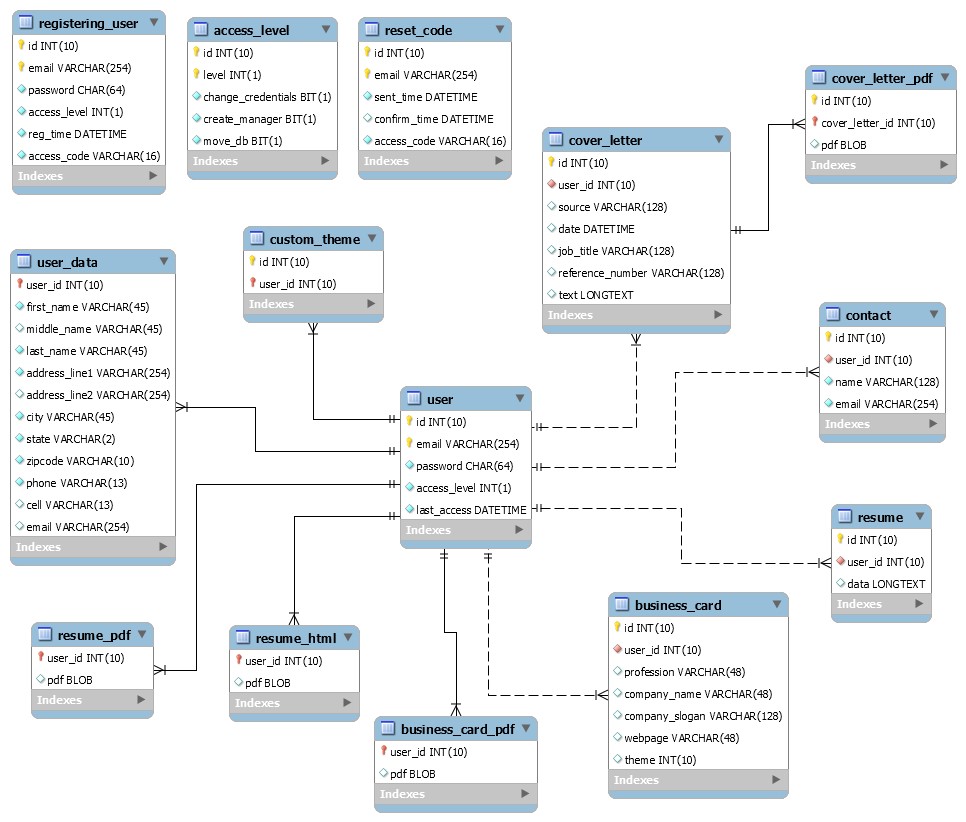
|  |  |
| --- | --- |
| **Class Name:** User | |
| **Brief description:** The User class is responsible for the creation of new users. | |
| **Attributes (fields)** | **Attribute Description** |
| No Attributes |  |
|  |  |
| **Methods (operations)** | **Method Description** |
| User(String, String, String, String) | This is a default constructor method. |
| getUser(String) | This is a method which accepts a String parameter getting Username info. |
| getFirstname() | This is a method for getting the User’s first name. |
| setFirstname(String) | This is a method that accepts a String parameter for setting the User’s first name. |
| getLastname() | This is a method for getting the User’s last name. |
| setLastname(String) | This is a method that accepts a String parameter for setting the User’s last name. |
| getEmail() | This is a method for getting the User’s email. |
| setEmail(String) | This is a method that accepts a String parameter for setting the User’s email. |
| getPassword() | This is a method for getting the User’s password. |
| setPassword(String) | This is a method that accepts a String parameter for setting the User’s password. |
| getCareer() | This is a method for getting the User’s career. |
| setCareer(String) | This is a method that accepts a String parameter for setting the User’s career. |

##### 3.1.03: Utilities.java

|  |  |
| --- | --- |
| **Class Name:** Utilities | |
| **Brief description:** The Utilities class is responsible for generating and managing the salt and encryption of passwords. | |
| **Attributes (fields)** | **Attribute Description** |
| No Attributes |  |
|  |  |
| **Methods (operations)** | **Method Description** |
| encrypt(String, String) | This method accepts two String parameters to encrypt a password. |
| genSalt() | This is a method which generates a salt to be added to a password. |

##### 

#### 3.2 Data Detailed Design



##### 3.2.01 User Access Level

|  |  |
| --- | --- |
| **Table:** | **access\_level** |
| **Description** | Used to store and define the access level for each unique user. |
| **Attributes** | **Description** |
| id | ID, integer, not null, auto-increment. |
| level | Access Level, integer, not null. |
| create\_admin | Privilege, bit, not null. |
| move\_db | Privilege, bit, not null. |

##### 3.2.02 User Information

|  |  |
| --- | --- |
| **Table:** | **Users** |
| **Description** | Used to store user information. |
| **Attributes** | **Description** |
| email | email, not null. |
| first | First name, not null. |
| last | Last name, not null. |
| password | Password, not null. |

##### 3.2.03 Reminders

|  |  |
| --- | --- |
| **Table:** | **Reminders** |
| **Description** | Used to store reminder information. |
| **Attributes** | **Description** |
| reminderName | Name of reminder, not null. |
| message | Message to be issued, not null. |
| currentTime | Time of reminder, time, not null. |

#### 4. User Interfaces by Use Case Scenarios

##### 4.1

**Use Case 01: Initial Administrator System Configuration.**

VaqPack shall allow the system administrator to configure and connect to the MySQL DB using a previously established DBA account upon initial installation. Upon creating a VaqPack Administrator account the user shall have access to the special administrator functions that are locked within VaqPack to normal users that allow for the Administrator to govern within the program.

###### A Valid Scenario

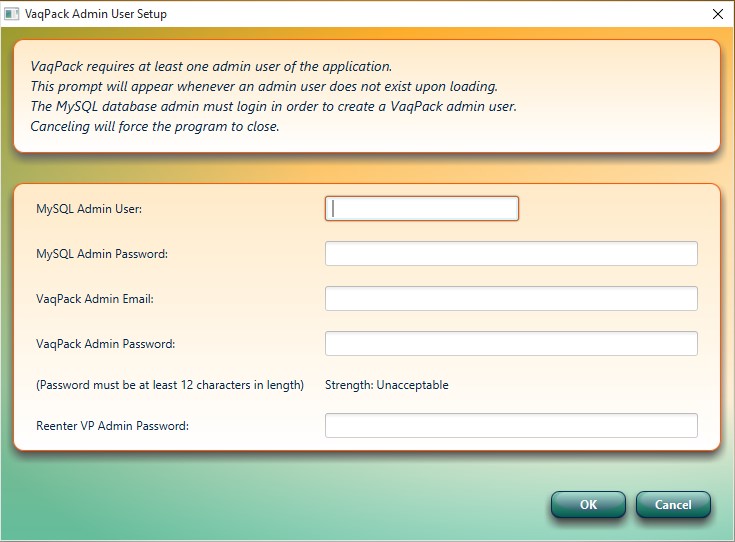
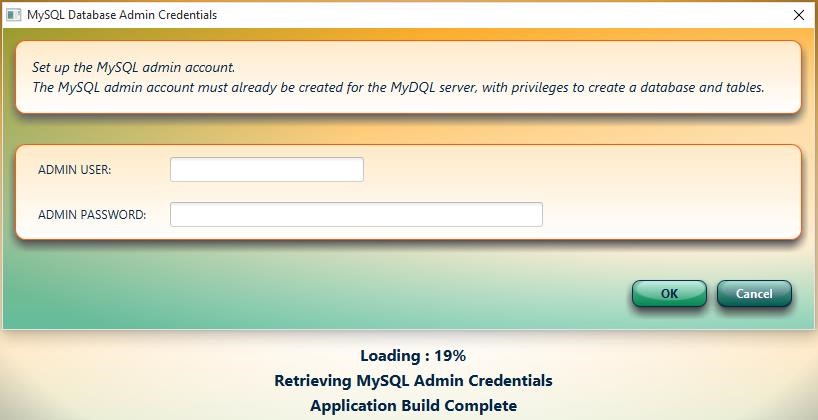
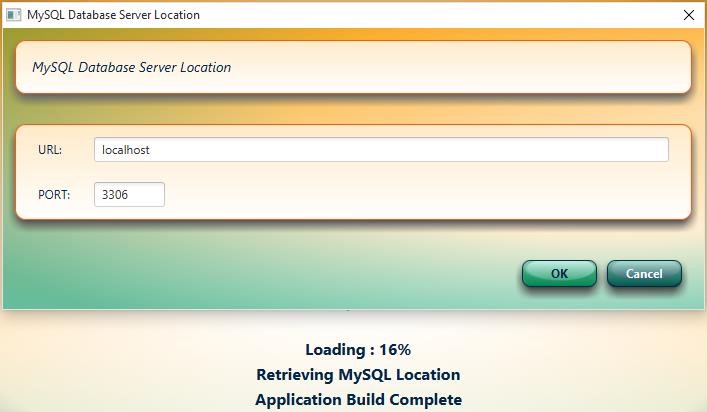
1. User runs VaqPack
2. User correctly inputs MySQL URL and PORT#
3. User successfully inputs MySQL admin username and password
4. User successfully creates initial VaqPack administrator account

###### Invalid Scenarios

1. User runs VaqPack
2. User incorrectly inputs MySQL URL or PORT#
3. VaqPack prompts error

1. User runs VaqPack
2. User correctly inputs MySQL URL and PORT#
3. User unsuccessfully inputs MySQL admin or password
4. VaqPack prompts error

##### Case 01 – Initial Administrator System Configuration

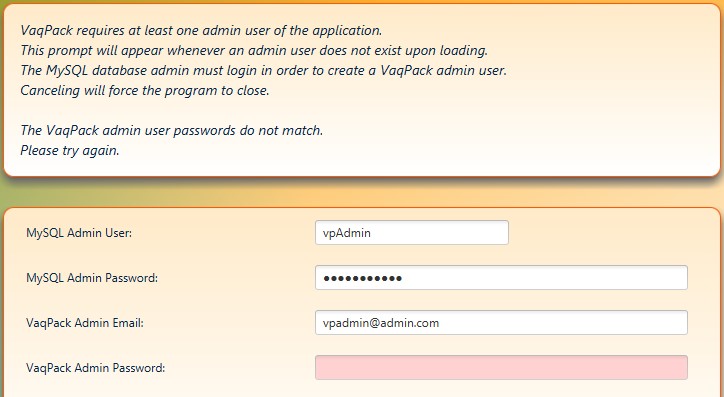


##### 4.2

###### Use Case 02: Administrator System Configuration Password Mismatch Case Scenario

1. User runs VaqPack
2. User correctly inputs MySQL URL and PORT#
3. User successfully inputs MySQL admin and password
4. User unsuccessfully inputs fields for initial VaqPack administrator account
5. VaqPack prompts error

###### Case 02



##### 4.3

**Use Case03: User Creates New Account**

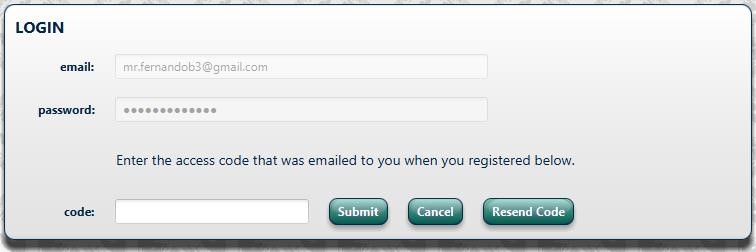
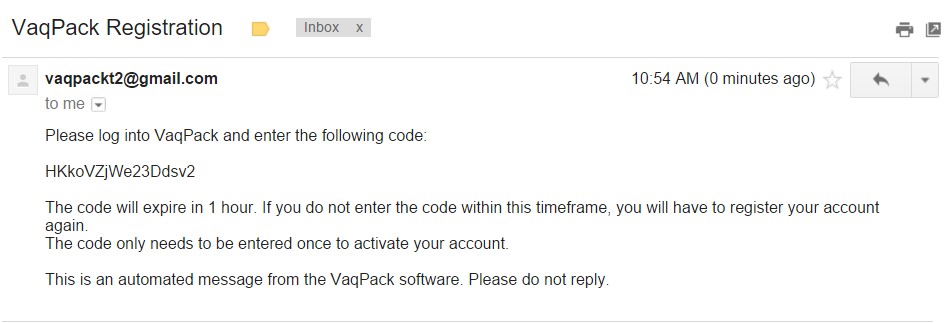
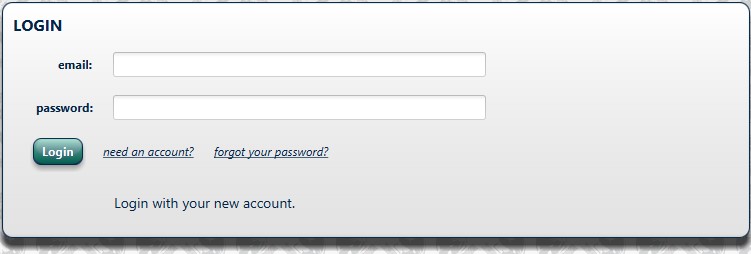
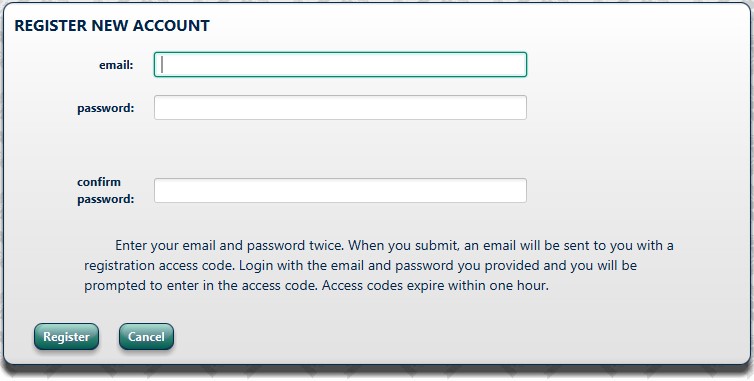
VaqPack shall allow and require users to create an account as stated in the SRS.

###### A Valid Scenario

1. User in login window, clicks on “need an account?”
2. User successfully inputs email and password
3. User retrieves VaqPack system generated code sent to the supplied email
4. User successfully inputs email password and code
5. User logs into VaqPack **An invalid Scenario**
6. User in login window, clicks on “need an account?”
7. User unsuccessfully inputs email and password
8. VaqPack prompts error

1. User in login window, clicks on “need an account?”
2. User successfully inputs email and password
3. User unsuccessfully inputs email password and code
4. VaqPack prompts error

##### Case 03 – User Creates New Account Successfully



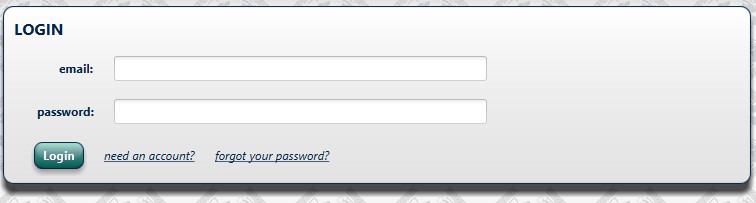
##### 4.4

###### Use Case04: User Login

VaqPack shall allow users that have created an account to log in to gain access to their personal information as stated in the SRS. **Case Scenario**

1. User in login window
2. User successfully inputs email and password
3. User logs into VaqPack

###### Case 04



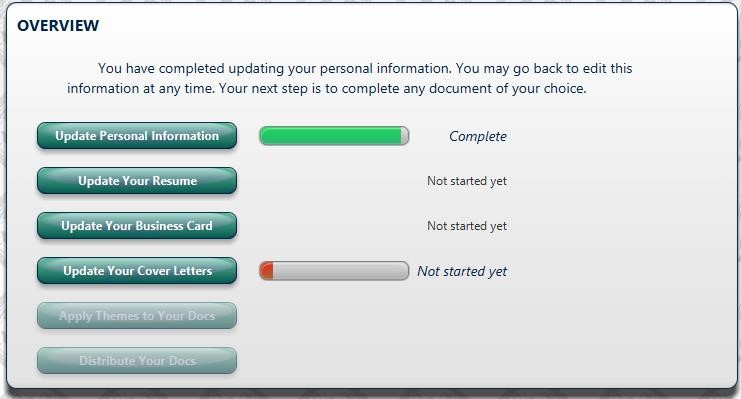
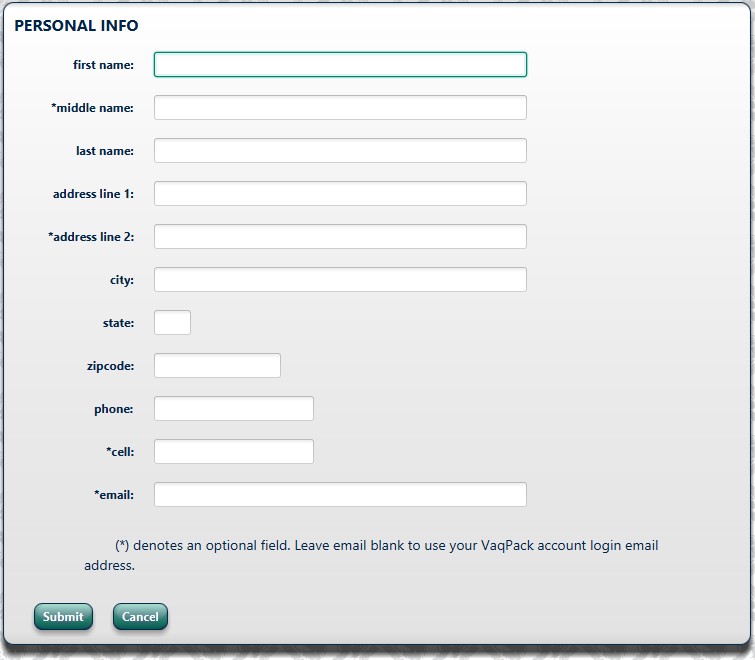
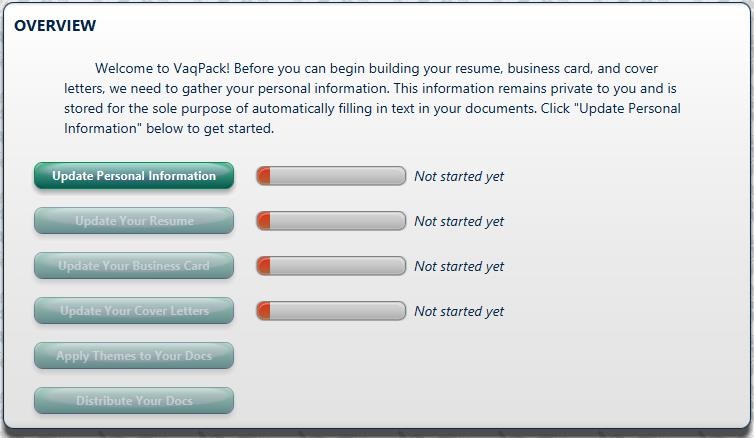
**4.5**

###### Use Case05: User Updates Personal Information Successfully

Upon successful log in, VaqPack will require users to Update Personal Information as an initial step.

###### Case Scenario

1. User is in the Overview page
2. User clicks “Update Personal Information”
3. User successfully updates personal Information
4. User gain access to additional program features via enabled buttons **Case 05**

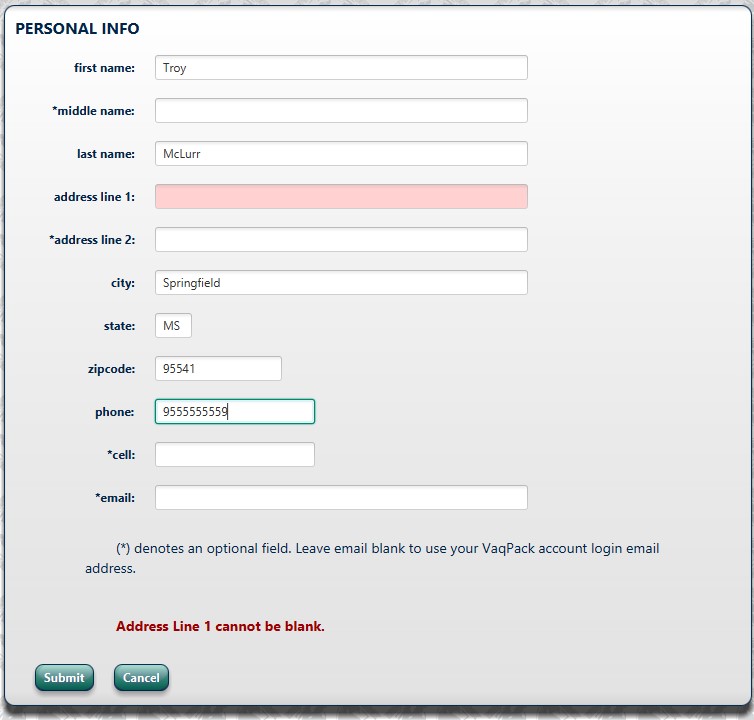


##### 4.6

###### Use Case06: User Updating Personal Information – Missing Required Field Case Scenario

1. User is in the Personal Info page
2. User leaves a required field blank
3. VaqPack highlights missing field, text displays which fields are missing

###### Case 06

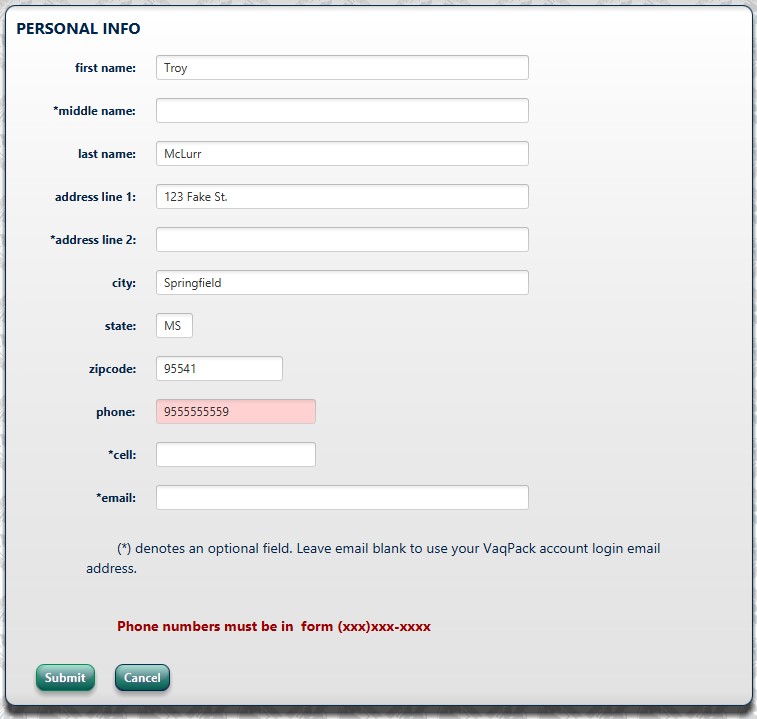


##### 4.7

**Use Case07: User Updating Personal Information – Incorrect Format Case Scenario:**

1. User is in the Personal Info page
2. User inputs a required field incorrectly
3. VaqPack highlights problem field and supplies correct format parameters

###### Case 07



##### 4.8

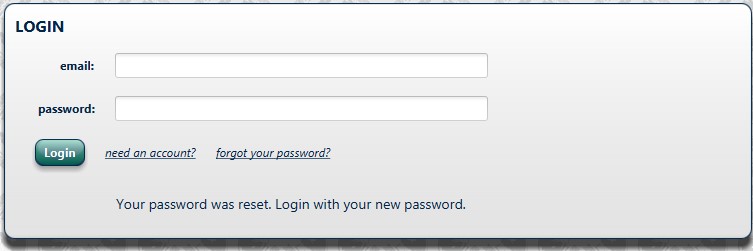
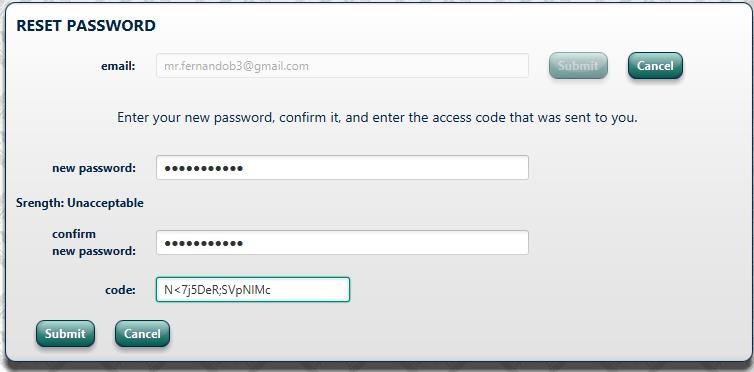
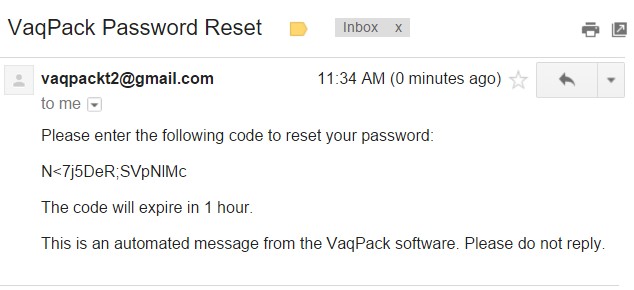
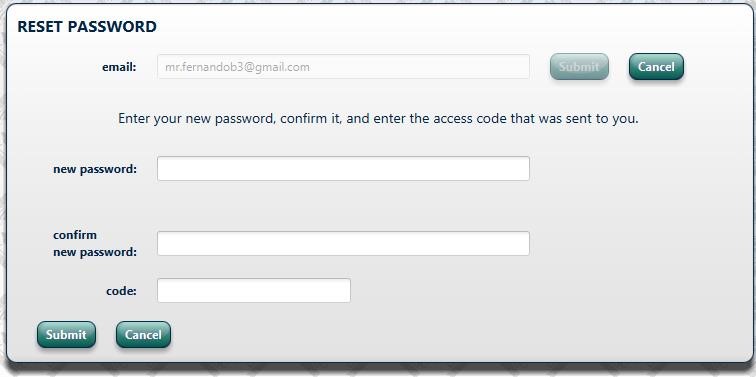
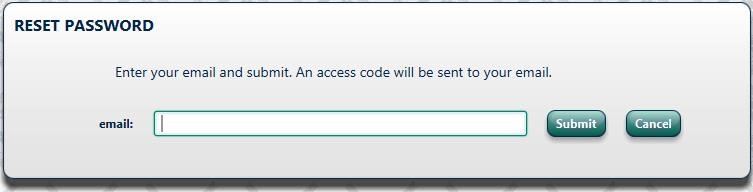
###### Use Case08: User Forgets Password

VaqPack shall allow users to reset their passwords upon completing a verification process consisting of confirming a code sent to the email supplied upon creating the account.

**Case Scenario:**

1. User is in login page
2. User clicks on “forgot your password?”
3. User successfully inputs email and clicks submit.
4. VaqPack sends email
5. User retrieves code from email
6. User updates password upon code verification

##### Case 08 – User Forgets Password



##### 4.9

###### Use Case09: Users New Password Not Strong Enough

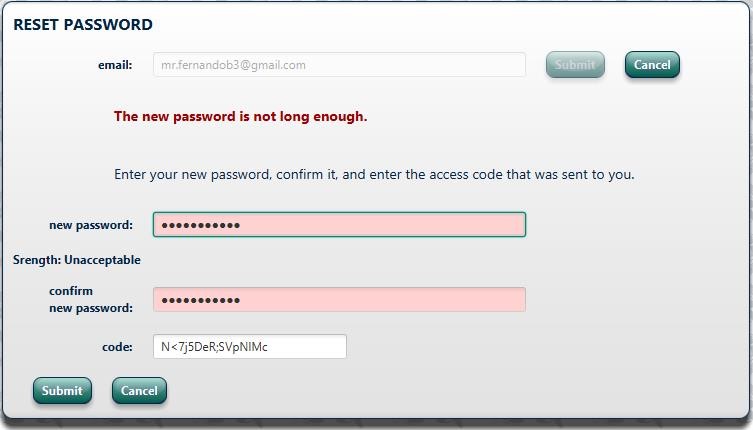
VaqPack requires a certain minimum strength/length before it accepts any users password choice.

**Case Scenario:**

1. User is in login page
2. User clicks on “forgot your password?”
3. User successfully inputs email and clicks submit
4. VaqPack sends email
5. User retrieves code from email
6. User updates password but doesn’t supply a strong/long enough password

4. VaqPack informs user to choose a stronger password

##### Case 09 – User New Password Not Strong Enough



##### 4.10

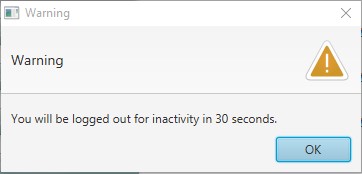
###### Use Case10: Users is inactive, automatic logout

VaqPack shall have a built in logout timer to log out a user after 5 minutes of inactivity. A warning window will popup when there is 30 seconds left. Any mouse movement or keyboard key presses will reset the timer.

**Case Scenario:**

1. User is inactive for 4minutes 30 seconds
2. VaqPack prompts warning window
3. 30 seconds later the user will automatically be logged out

##### Case 10 – User Is Inactive, automatic logout



### A. Appendices

**A.1 Appendix 1**

Required form for the approval of changes to this SDD document:

### Document Approval

The following Software Design Specification has been accepted and approved by the following:

**Signature Printed Name Title Date**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Elijah Lopez | Lead Software Engineer |  |
|  | Dr. M.K. Quweider | Instructor, CSCI-3340 |  |

#### A.2 Appendix 2

Required form must be attached to the end of this document if there are any changes after its initial completion:

### Revision History

#### Date Description Author Comments

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |