**Software Requirements**

**Specification**

**VaqPack**

Graduate-to-Professional

Aid Pack

Version 1.0

November 5th, 2015

**Lead Software Engineer:**

William Dewald

**Project Team:**

Fernando Bazan

Nathanael Carr

Erik Lopez

Raul Saavedra

Prepared for

Software Engineering

University of Texas Rio Grande Valley

Instructor: MK Quweider, Ph.D.

Fall 2015

**Table of Contents**

**1. INTRODUCTION**……………………………………………………..... **1**

1.1 PURPOSE………………………………………………………………...…. 1

1.2 SCOPE…………………………………………………………………...…. 1

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS…………...…… 1

1.4 REFERENCES……………………………………………………...………. 3

1.5 OVERVIEW………………………………………………………...………. 3

**2. GENERAL DESCRIPTION**…………………………………………..... **4**

2.1 PRODUCT PERSPECTIVE………………………………………………... 4

2.2 PRODUCT FUNCTIONS………………………………………………...… 4

2.3 USER CHARACTERISTICS………………………………………………. 4

2.4 GENERAL CONSTRAINTS……………………………………………….. 5

2.5 ASSUMPTIONS AND DEPENDENCIES…………………………………. 5

**3. SPECIFIC REQUIREMENTS**…………………………………………. **6**

3.1 EXTERNAL INTERFACE REQUIREMENTS…………………………..... 6

3.1.1 User Interfaces……………………………………………………………….. 6

3.1.2 Hardware Interfaces…………………………………………………………. 6

3.1.3 Software Interfaces………………………………………………………..… 6

3.1.4 Communications Interfaces………………………………………………….. 6

3.2 FUNCTIONAL REQUIREMENTS

3.2.1 <Functional Requirement or Feature #1>

3.2.2 <Functional Requirement or Feature #2>

3.3 USE CASES 3

3.3.1 Use Case #1

3.3.2 Use Case #2

3.4 CLASSES / OBJECTS

3.4.1 <Class / Object #1>

3.4.2 <Class / Object #2>

3.5 NON-FUNCTIONAL REQUIREMENTS

3.5.1 Performance

3.5.2 Reliability

3.5.3 Availability

3.5.4 Security

3.5.5 Maintainability

3.5.6 Portability

3.6 INVERSE REQUIREMENTS

3.7 DESIGN CONSTRAINTS 4

3.8 LOGICAL DATABASE REQUIREMENTS

3.9 OTHER REQUIREMENTS

**4. ANALYSIS MODELS**

4.1 SEQUENCE DIAGRAMS

4.3 DATA FLOW DIAGRAMS (DFD)

4.2 STATE-TRANSITION DIAGRAMS (STD)

**5. CHANGE MANAGEMENT PROCESS**

**A. APPENDICES**

**1. Introduction**

**1.1 Purpose**

The purpose of the Software Requirements Specification is to provide a detailed description of the VaqPack Graduate to Professional Aid Pack application. The intention of the SRS is to articulate the purpose and features of the application, along with its user and external interfaces, constraints, dependencies, functionality, and attributes. This artifact provides the guidelines for the design and implementation of the software, and clarifies the description of the software for the customer. Therefore, the intended audience of this document includes the client, users, and developers.

**1.2 Scope**

The software application described throughout this SRS document is the VaqPack Graduate to Professional Aid Pack, or simply VaqPack. While this free desktop application can be used within any institution, it is primarily designed for the graduating students of the University of Texas Rio Grande Valley with the purpose of aiding their progression from academia to the professional world.

Using a graphical user interface, registered users of VaqPack can generate a resume, a business card, and cover letters. The information required for the generation of these objects is collected from the user, by means of input forms within a wizard, and then stored in a pre-existing MySQL database. Users can apply themes of their choice to these objects for personal style or look-and-feel. From these objects, the user may generate PDF documents which can be sent to contacts or potential employers via email. Additionally, the user may generate an HTML file from the resume data for use as a web page. Users may retrieve stored data for the purpose of editing or augmenting information, or to send documents to contacts at any given time.

**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** |
| SRS | Software Requirement Specification |
| 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. |
| Wizard | A sequential set of prompts for input, assisting in data collection and organized such that its implementation increases ease of use. |
| Database | A structured collection of data that can be efficiently and conveniently accessed. |
| PDF | Portable Document Format; a popular electronic document file type particularly used with rich-text or styled text. |
| HTML | HyperText Markup Language; the web standard language used in the delivery of online content, interpreted and rendered by web browsers. |
| IDE | Integrated Development Environment; software that provides tools for the development and organization of programming code. |
| Git | A version control system for the development of software. |
| GitHub | A web-based Git repository used by software development teams. |
| Java Virtual Machine | Provides the necessary links allowing a java program to run on a machine using a particular operating system. |
| Java Runtime Environment | Including the Java Virtual Machine, all necessary components for a system to establish the environment in which Java programs will run. |
| SQL | Structured Query Language; the standard relational database query language |
| JDBC | Java Database Connectivity; an Java API developed by Oracle Corporation which provides methods for querying and updated a database. |

**1.4 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/jre8-downloads-2133155.html>

JavaFX - <http://docs.oracle.com/javase/8/javase-clienttechnologies.htm>

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

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

**1.5 Overview**

The remaining content of this SRS is organized in 5 sections: General Description, Specific Requirements, Analysis Models, Change Management Process, and the Appendices. The General Description section aims to make the requirements for the VaqPack application more easily understood from a high-level point of view, especially from the perspective of typical end-users. However, the Specific Requirements will define and describe the details of these requirements with the technical information needed by the developers. The Analysis Models section lists all of the models used in developing the specific requirements that are outlined in the previous section. Since the first version of VaqPack is currently being constructed, and since many requirements still need to be met, this section is subject to much change and many additions. The Change Management Process section outlines the procedures that must be followed when such changes occur throughout the development of VaqPack, including the updates to this SRS document. The Appendices include conceptual documents such as the initially provided high-level requirements and any conceptual diagrams or documents used by the developers. The documents in the Appendix may or may not be used in requirements definitions, but this is clearly specified for each document.

**2.** **General Description**

**2.1 Product Perspective**

The VaqPack product is independent in that it does not augment any existing product and is not intended to encapsulate another product. However, VaqPack must operate along with a MySQL server. Connectivity to a MySQL server is included with the software. VaqPack intends to replace an array of available paid-access products that focus on assisted resume and business card construction, ranging from desktop applications to online services.

**2.2 Product Functions**

In a general high-level point of view, the VaqPack application will perform the following functions:

* Store user login credentials and privileges in a database.
* Store collected user information in a database.
* Store a list of the user’s contacts in a database.
* Provide a system for an admin user to connect to a MySQL database server.
* Provide a system for an admin user to initialize the database on first run.
* Retrieve individual collected user information for viewing or editing.
* Retrieve individual collected user contacts for viewing or editing.
* Generate a resume object from collected user information.
* Generate a business card object from collected user information.
* Generate employer-specific cover letter objects.
* Apply themes or styles to the produced objects.
* Generate printable, distributable PDF files of the objects.
* Generate an HTML file of the produced resume.
* Store generated files in the database for fast access.
* Email the generated files to selected user contacts.
* Email the generated files to the user for personal access.
* Prompt the user with a wizard for ease in data collection.
* Provide a system to reset a forgotten user password.
* Provide a system for an admin user to migrate the database.

**2.3 User Characteristics**

While the VaqPack product is primarily designed for the graduating students of the University of Texas Rio Grande Valley, it can be used by any individual who needs to prepare and distribute the personal information that is required for employment consideration within most professions. Additionally, the VaqPack user could simply be interested in generating a printable business card for distribution to clients or for business-related networking within his or her current profession.

**2.4 General Constraints**

In a general high-level point of view, the developers of the VaqPack application will have the following constraints:

* VaqPack must be developed using the NetBeans IDE.
* VaqPack must be MySQL database-driven
* Choice of local or remote database server must be present.
* The VaqPack GUI must be JavaFX-based.
* VaqPack must be desktop-based.
* Git and GitHub must be used for version control.

**2.5 Assumptions and Dependencies**

In a general high-level point of view, the developers of the VaqPack application requirements are currently influenced by the following assumptions and dependencies:

* It is assumed that VaqPack will run on a system with an operating system that has a compatible Java Virtual Machine and up to date Java Runtime Environment.
* It is assumed that VaqPack will run on or connect to a system with an existing MySQL server.
* It is assumed, in the event of remote MySQL server connectivity, the system on which VaqPack will run has the networking capabilities to connect to said database server.

**3.** **Specific Requirements**

*This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project’s software design, implementation, and testing.*

*Each requirement in this section should be:*

* *Correct*
* *Traceable (both forward and backward to prior/future artifacts)*
* *Unambiguous*
* *Verifiable (i.e., testable)*
* *Prioritized (with respect to importance and/or stability)*
* *Complete*
* *Consistent*
* *Uniquely identifiable (usually via numbering like 3.4.5.6)*

*Attention should be paid to the carefuly organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.*

**3.1 External Interface Requirements**

**3.1.1 User Interfaces**

Like the majority of modern desktop software applications, VaqPack provides a GUI for the user to interface with all of the functionality necessary to accomplish the user’s goals in a visual manner—creating distributable, styled resumes, business cards, and cover letters.

**3.1.2 Hardware Interfaces**

VaqPack requires the screen, keyboard and mouse that is typical for desktop machines. The hardware required is that of a modern desktop computer.

**3.1.3 Software Interfaces**

VaqPack is Java program and therefore interfaces with the Java Runtime Environment and Java Virtual Machine for whichever platform the program must run. VaqPack must operate along with a MySQL server. Connectivity to a MySQL server is included with the software.

**3.1.4 Communication Interfaces**

VaqPack can connect to a remote MySQL database which may require Internet connectivity. VaqPack can also send files as email attachments which requires Internet connectivity. The product communicates with the MySQL server via a driver that is embedded in the software and uses SQL standards.

**3.2 Functional Requirements**

*This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.*

**3.2.1 Store User Login Credentials**

3.2.1.1 Introduction

VaqPack stores the user’s chosen email and password in a database.

3.2.1.2 Inputs

The user provides their email address and a password of their choosing.

3.2.1.3 Processing

VaqPack stores the email and password in the ‘user’ database table.

3.2.1.4 Outputs

Upon entering the proper credentials, the user is allowed access to their stored data.

3.2.1.5 Error Handling

VaqPack verifies the correct email and password are used to login and that valid inputs are provided at the time of account creation.

**3.2.2 Store Collected User Information**

3.2.2.1 Introduction

VaqPack provides a sequential process, or ‘wizard’, for gathering the user’s data and storing that data in a database.

3.2.2.2 Inputs

The user provides their full name, mailing address, phone number(s), career objective(s), employment history, professional skills, and references.

3.2.2.3 Processing

VaqPack stores the collected data in the ‘user\_data’ and ‘resume’ tables of the database.

3.2.2.4 Outputs

No particular output unless retrieved via database query.

3.2.2.5 Error Handling

VaqPack sanitizes inputs using software security best practices.

**3.2.3 Store User’s Contacts**

3.2.3.1 Introduction

If the user enters one or more contacts, VaqPack stores those contacts in the database for future use.

3.2.3.2 Inputs

The user provides the full name and email address for each contact they wish to add.

3.2.3.3 Processing

VaqPack stores the collected data in the ‘contact’ tables of the database.

3.2.3.4 Outputs

No particular output unless retrieved via database query.

3.2.3.5 Error Handling

VaqPack sanitizes inputs using software security best practices.

**3.2.4 Database Initialization**

3.2.4.1 Introduction

Upon launch of VaqPack, if there is no existing database, VaqPack generates the required database and associated tables.

3.2.4.2 Inputs

A stored SQL script detailing the required tables and associated attributes.

3.2.4.3 Processing

VaqPack utilizes JDBC to create a database object with a predefined admin-level account.

3.2.4.4 Outputs

A MySQL database with admin-level user and predefined tables.

3.2.4.5 Error Handling

VaqPack verifies that the required database is non-existent before a new database is initialized and ensures database integrity post initialization.

**3.2.5 Database Admin Connection**

3.2.5.1 Introduction

Upon launch of VaqPack, if there is an existing database, VaqPack connects to that database as an admin user.

3.2.5.2 Inputs

VaqPack contains a pre-defined admin-level database user.

3.2.5.3 Processing

VaqPack connects to the database as an admin.

3.2.5.4 Outputs

An admin-level database connection.

3.2.5.5 Error Handling

VaqPack notifies the user if the admin account cannot login and the reason for the connection error.

**3.2.6 Reset User Password**

3.2.6.1 Introduction

If the user forgets their password, VaqPack allows them to reset the password.

3.2.6.2 Inputs

The user answers pre-defined question(s) as an alternative method of authentication.

3.2.6.3 Processing

If user authentication passes, VaqPack allows the user to login and create a new password.

3.2.6.4 Outputs

User regains access to their stored information.

3.2.6.5 Error Handling

VaqPack notifies the user if their authentication fails and limits the number of consecutive attempts.

**3.2.7 Database Migration**

3.2.7.1 Introduction

A system which allows the VaqPack user to migrate their database to another system as needed.

3.2.7.2 Inputs

The user provides a valid login

3.2.7.3 Processing

If user authentication passes, VaqPack allows the user to login and export their data to a specified location. Alternatively, if the user is reestablishing their database, VaqPack allow import of a data file.

3.2.7.4 Outputs

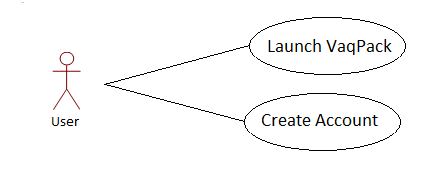
If exporting, the exported data file. If importing, a restored user database.

3.2.7.5 Error Handling

The user will be notified upon a failed export of import of the user data.

**3.3 Use Cases**

**3.3.1 Create a VaqPack Account**

****

**Goal:** User Creates an Account

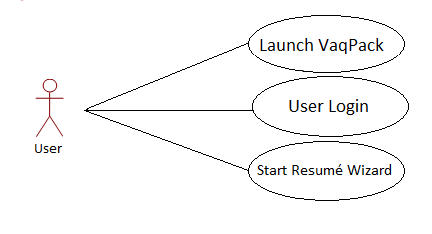
**Input:** Email/Password

**Output:** none.

**Main Scenario:** User creates an account to store user login credentials

**Pre-Condition:** VaqPack must be running.

**3.3.2 User Creates a Resume**

****

**Goal:** User Creates a Résumé

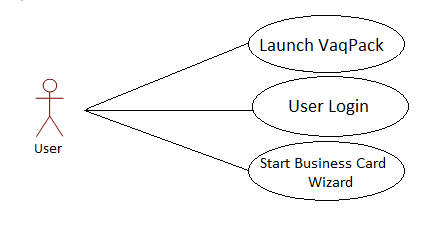
**Input:** User chooses a template & fills in all required fields in the résumé wizard.

**Output:** Resume PDF, Resume HTML,

**Main Scenario:** User wants to create a résumé.

**Pre-Condition:** VaqPack must be running, user must login.

**3.3.3 User Creates a Business Card**

****

**Goal:** User Creates a Business Card

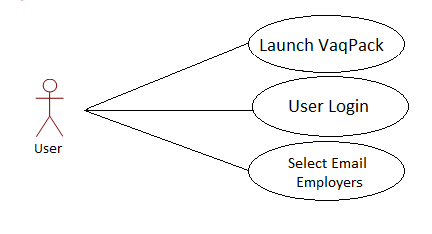
**Input:** User chooses a template & fills in all required fields in the business card wizard.

**Output:** Business Card

**Main Scenario:** User wants to create a business card.

**Pre-Condition:** VaqPack must be running & user must log in.

**3.3.4 User Emails Résumé / Business Card / Both**

****

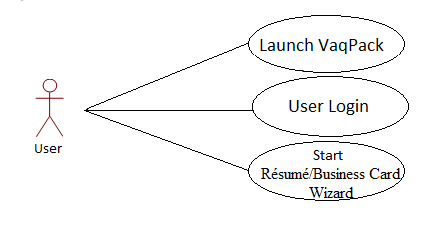
**Goal:** User Emails Potential Employers Résumé / Business Card

**Input:** User inputs destination email address and recipient information.

**Output:** Email containing Résumé and or Business Card.

**Main Scenario:** User wants to email Résumé/Business Card to potential employers.

**Pre-Condition:** VaqPack must be running, user must log in, Résumé Wizard and or Business Card Wizard must have already been completed.

**3.3.5 User Updates Résumé / Business Card / Both**

**Goal:** User Updates Résumé / Business Card

**Input:** User chooses which fields to update when running through the Wizard again.

**Output:** Résumé PDF / Résumé HTML / Business Card

**Main Scenario:** User wants to update personal information on Résumé / Business Card.

**Pre-Condition:** VaqPack must be running, user must log in, Résumé and or Business Card Wizards must be complete.

**3.4 Classes / Objects**

**3.4.1 <Class / Object #1>**

3.4.1.1 Attributes

3.4.1.2 Functions

<Reference to functional requirements and/or use cases>

**3.4.2 <Class / Object #2>**

…

**3.5 Non-Functional Requirements**

*Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).*

**3.5.1 Performance**

**3.5.2 Reliability**

**3.5.3 Availability**

**3.5.4 Security**

**3.5.5 Maintainability**

**3.5.6 Portability**

**3.6 Inverse Requirements**

*State any \*useful\* inverse requirements.*

**3.7 Design Constraints**

*Specify design constrains imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.*

**3.8 Logical Database Requirements**

*Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.*

**3.9 Other Requirements**

*Catchall section for any additional requirements.*

**4.** **Analysis Models**

*List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.*

**4.1 Sequence Diagrams**

**4.3 Data Flow Diagrams (DFD)**

**4.2 State-Transition Diagrams (STD)**

**5.** **Change Management Process**

*Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.*

**A.** **Appendices**

*Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.*

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

**A.1 Appendix 1**

**A.2 Appendix 2**