VaqPack

(Database)

v1.0

Software Requirements Specification

November 11,2016

Elijah Lopez
Jose Rodriguez
Juan Delgado
Michelle Marie Garcia

VaqPack (Database)

Instructor: MK Quweider, Ph.D.
Fall 2016

Revision History

Date	Description	Author	Comments
11/12/16	V1.0 (local-based)	Team Black	Non-internet based

Document Approval

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

Signature	Printed Name	Title	Date
	Elijah Lopez	Point of Contact	
	Dr. M.K. Quweider	Instructor, CSCI-3340	

VaqPack (Database)

Table of Contents

VaqPack (Database)

1. Introduction

1.1 Purpose

The purpose of the Software Requirements Specification is to provide a detailed description of the database portion of the VaqPack Software. The intention of the SRS is to articulate the purpose and features of the database, along with its associated directories, files, 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 database portion of the VaqPack application is primarily designed for use with the VaqPack software. It is intended to be used by VaqPack as a method to store and retrieve data in the form of XML files, pdfs, and htmls. The database portion of the VaqPack portion also includes the creation and maintenance of physical directories on the user's computer. User credentials will be saved to the database for the purpose of validating the user in order to retrieved his/her stored profile. The user's profile consists of his/her selected courses, e-mail, first name, last name, and password. This database is only intended to function on a local level and will not employ an internet based connection. In addition, this database is intended to work on computers that have MySQL software installed.

1.3 Definitions, Acronyms, and Abbreviations

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.
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.
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.
mySQL	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/
- Java Virtual Machine - https://java.com/en/download/
- Java Runtime Environment - http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
  MySQL - http://dev.mysql.com/downloads/mysql/
  JDBC - http://www.oracle.com/technetwork/java/javase/jdbc/index.html

2. General Description

2.1 Product Perspective

The database portion of the VaqPack software is intended to be used as a method for initializing a database, storing and retrieving different forms of data, creating physical directories, storing user credentials, storing user profiles and validating user credentials.

2.2 Product Functions

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

- Initialize a database with the appropriate tables.
- Create Directories
- Store user credentials
- Validate user login credentials
- Store a users profile (Where a profile is a users course selections/schedule)
- Store and retrieve program generated files.
- Generate a admin account for “housekeeping” operations
- Database Migration.

2.3 User Characteristics

The database portion of the VaqPack software is not intended to be used directly by the user. Instead, the database will be used by the VaqPack program itself. However, access to housekeeping functions will be granted only to the administrator account who will be able to make changes to the database.

2.4 General Constraints

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

- As of now (11-12-16), the database will be local based only and not utilize a internet based connection
- The database (where database includes tables and directories) is designed to work with the VaqPack software and will not work with other software programs.

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 the database will be created on a computer using a current version of mySQL.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

A user of the VaqPack software will not be able to directly access the database. However the GUI portion of the VaqPack software will interact with the database via buttons such as “submit”, “add”, ”remove”, “update” and “login”. These buttons will be used to add/delete the appropriate user data, register a new account or log into a existing user account. A separate admin account however will have the ability to preform direct “housekeeping” operations on the database (where housekeeping operations are actions such as initiating a database, migrating a database and or dropping tables). Wither or not these admin abilities will be realized via a GUI is still under consideration as of now (11-13-2016).

3.1.2 Hardware Interfaces

A computer capable of running a current JVM machine is assumed. In addition a computer capable of installing and running a current version of MySQL is assumed as well.

3.1.3 Software Interfaces

The JDBC API will be the main interface between the MySQL database and the VapPack java classes. The JDBC API provides functions that allow for removal, addition and changes to a mySQL database.

3.1.4 Communications Interfaces

At this point (11-12-16) no internet based communication is expected. All communications between the database and the main VaqPack software will be done via local host.

3.2 Functional Requirements

3.2.1 Database Initialization

- The main VaqPack program will start. At each start the VaqPack program will call the database portion of Vaqpack to preform the following operations:

    1. Check if there is a existing database. If there is do nothing. If there is not a existing database, create a database with the appropriate tables.

3.2.1.2 Inputs

- "CREATE" and "ADD" queries to mySQL via JDBC in order to create a database and add appropriate tables

3.2.1.3 Processing

- mySQL processes queries.

3.2.1.4 Outputs

- A working relational database with appropriate tables.

3.2.1.5 Error Handling

- Connection Error- Failure to connect with mySQL. Message "check mySQL installation"

3.2.2.Store User Credentials

- The main VaqPack program will start. After database verification VaqPack will prompt the user for their login credentials or to register a new account. Registering a new account will be handled in the following way.

    1. Accept the users credentials.
    2. Sanitize user credentials.
    3. Check current database for duplicate entry
    4. If no duplicates, Hash encrypt user password
    5. Insert credentials in "user" table via JDBC query

3.2.2.2 Inputs

- First name, last name, email, password

3.2.2.3 Processing

1. Sanitize user credentials via prepared statements.
2. Has password via Java hash function.
3. Insert credentials in user table via JDBC and mySQL.

3.2.2.4 Outputs

- A "Success" Message upon a successful registration.

3.2.2.5 Error Handling

- Connection Error- Failure to connect with mySQL. Message “check mySQL installation”
- Duplicate Entry-Message “already exist”

3.3 Use Cases

3.3.1 Use Case #1

- User creates a new Course and wants to save it to their profile

Inputs:

- newly generated xml file.

Processing:

1. Accept xml file.
2. Store as blob in Courses Table

Outputs:

- None

Error handling

- Connection Error-Failure to connect with mySQL. Message “check mySQL installation”
- Duplicate Entry-Do not store, message “Duplicate”

3.3.2 Use Case #2

- User wants to log in.

Inputs:

- email, password.

Processing:

1. Accept Password and email.
2. Sanitize input.
3. Hash Password inputted password.
4. Search database for a match (inputted credentials vs credentials currently in User table)
5. If found, allow access to VapPack. If not Found , Message “Error”.

Outputs:

- Error Message on Failure to find matching credentials
- Success Message on Success on matching credentials

Error handling:

- Connection Error-Failure to connect with mySQL. Message “check mySQL installation”

3.4 Classes / Objects

3.4.1 DB.java

Attributes:

- Class DB primary function is checking for the exist of a database and initiating a database with the appropriate tables in the event that a database does not exist.

Functions:

1. dbCheck()
   - Checks for the existence of a database of a given name.

2. DbInit()
   - Create a database of a given name and calls upon dbCreateTables() to populate the created database with the appropriate tables.
   -
3. dbCreateTables()
   - Creates tables of predefined names and predefined columns.

3.4.2 DBOperations.java

Attributes:

- Class Dboperations provides a collection of functions that will be used to query the database

Functions:

1. dbCheckUser()
   - Searches for a given user in the User table

2. dbRegister()
   - Inserts a given User into the User table.

3.4.3 User.java

Attributes:

- Class user will be used as a “middle man” object between the GUI and Database Query. The user class will be used as a container object that will be passed to the database to search for matches in the case of user login. The user class will be used as a container object that will be passed to the database to store user credentials in the case of registration.

Functions:

1. User(String Firstname, Lastname, password, email)
   - Constructor with full parameters used to register a user.

2. User (String email, String password)
   - Constructor with only 2 parameters used to search for a user in the database

3.4.3 Util.java

Attributes:

- A utilities class that will be used to hash passwords, generate and verify salts

Functions:

1. hash(String password)
   - accepts a string. Calls genSalt() to get salt. Combines string and salt then hashes and returns a hashed version of original password.

2. genSalt()

   - generates a salt, call veriSalt to make sure salt is unique.

3. veriSalt()
   - verifies that a newly generates salt is unique among salts in current database.

3.5 Non-Functional Requirements

3.5.1 Performance

- The time it takes to lookup a User in the database will be reduced by utilizing hashing functions.

3.5.2 Reliability

- The information stored in the database will be persistent and can only be removed completely by a admin level account.

3.5.3 Availability

- That database will always be available so long as mySQL remains installed

3.5.4 Security

- User passwords will be hashed in order to prevent unauthorized access to a users account.

3.5.5 Maintainability

- A admin account will be used for housekeeping

3.5.6 Portability

- The database will be able to be exported via a migration feature.

VaqPack (Database)

3.8 Logical Database Requirements

Type of Database:

- Relational

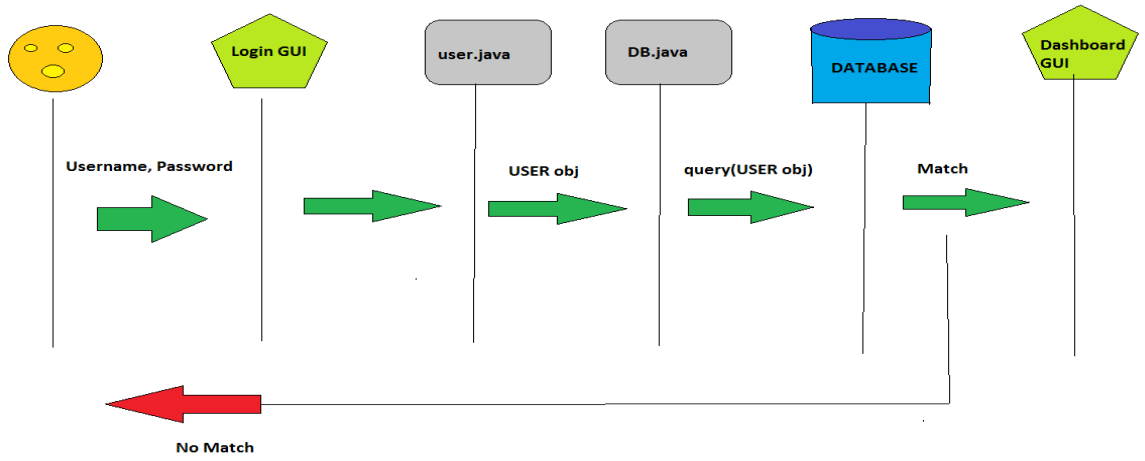
Data Formats:

- xml, pdf, html,clob/blob

4. Analysis Models

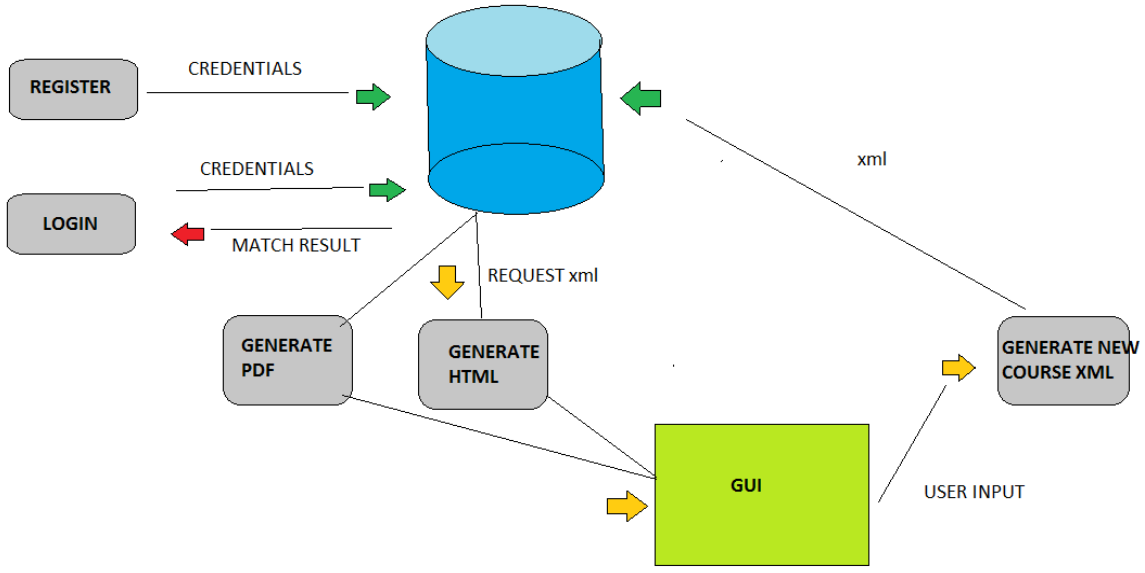
4.1 Sequence Diagrams

Case Where User attempts to login. Login credentials are passed to user class where a object of user class is created. That object is then passed to DB class which uses the info from the obj to query the database looking for a match. On a match the user is allowed to proceed to the Dashboard. On no match return to user with a error message.



4.3 Data Flow Diagrams (DFD)

A Diagram demonstration the database-driven nature of the VaqPack Software.



5. Change Management Process

As the rest of the others groups need and request changes to the database design.

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