**Software Design Documentation**

**VaqPack**

Graduate-to-Professional

Aid Pack

Version 1.0

November 20th, 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 PURPOSE…………………………………………………………………………...

1.2 SCOPE………………………………………………………………………………

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

**2. REFERENCES**………………………………………………………………….

**3. DECOMPOSITION DESCRIPTION**…………………………………………..

3.1 MODULE DECOMPOSITION.................................................................................

3.1.1 Module 1 Decomposition…………………………………………………………...

3.1.2 Module 2 Decomposition …………………………………………………………...

3.2 CONCURRENT PROCESS DECOMPOSITION………………………………….

3.2.1 Process 1 Decomposition ……………………………………………………………

3.2.2 Process 2 Decomposition.............................................................................................

3.3 DATA DECOMPOSITION………………………………………………………...

3.3.1 Data Entry 1 Decomposition ……..…………..………………………………………

3.3.2 Data Entry 2 Decomposition ………………………………………………………..

**4. DEPENDENCY DESCRIPTION**……………………………………………….

4.1 INTER-MODULE DEPENDENCIES.......................................................................

4.2 INTER-PROCESS DEPENDENCIES……………………………………………...

4.3 DATA DEPENDENCIES…………………………………………………………..

**5. INTERFACE DESCRIPTION**.............................................................................

5.1 MODULE INTERFACE……………………………………………………………

5.1.1 Module 1 Interface……….…………………………………………………………...

5.1.2 Module 2 Interface…..…...…………………………………………………………...

5.2 PROCESS INTERFACE……………………………………………………………

5.2.1 Process 1 Interface……………………………………………………………………

5.2.2 Process 2 Interface........................................................................................................

**6. DETAILED DESIGN**……………………………………………………………

6.1 MODULE DETAILED DESIGN...........................................................................

6.1.1 Module 1 Detail……………………………………………………………………...

6.1.2 Module 2 Detail………………………………………………………..…………...

6.2 DATA DETAILED DESIGN…………………………………...………………….

6.2.1 Data Entity 1 Detail……..……………………………………………………………

6.2.2 Data Entity 2 Detail……..............................................................................................

**1. Introduction \*(rough draft)**

**1.1 Purpose \***

This Software Design Document is made with the purpose of explicitly outlining the software architecture and high level design of the VaqPack Graduate to Professional Aid Pack application. Through the use of various architectural decompositions, the various modules, data structures, databases and interfaces will be depicted. The intention of this document is to provide developers an insight into meeting our Client’s needs efficiently and effectively, 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 VaqPack Graduate to Professional Aid Pack, or simply VaqPack. 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 clarify the development teams thought process during implementation.

(Blah blah blah)

**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. |
| 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. |
| JDO | Java Data Object; A standard way to access persistent data in databases, using plain old Java objects (POJO) to represent persistent data. |
| POJO | Plain Old Java Object; An ordinary Java object, not bound by any special restriction. |
| IDE | Integrated Development Environment; software that provides tools for the development and organization of programming code. |
| 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. |
| XSL | eXtensible Stylesheet Language; a style sheet language for XML documents. |
| 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/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/>

HTML - <http://www.w3schools.com/html/>

**3.** **Decomposition Description**

**3.1 Module Decomposition**

**3.1.1 Module 1 Decomposition**

Blah blah blah.

**3.1.2 Module 2 Decomposition**

Blah blah blah.

**3.2 Concurrent Process Decomposition**

**3.2.1 Process 1 Decomposition**

Blah blah blah.

**3.2.2 Process 2 Decomposition**

Blah blah blah.

**3.3 Data Decomposition**

**3.3.1 Data Entry 1 Decomposition**

Blah blah blah.

**3.3.2 Data Entry 2 Decomposition**

Blah blah blah.

**4.** **Dependency Description**

**4.1 Inter-module Dependencies**

Blah blah blah.

**4.2 Inter-process Dependencies**

Blah blah blah.

**4.3 Data Dependencies**

Blah blah blah.

**5.** **Interface Description**

**5.1 Module Interface**

**5.1.1 Module 1 Interface**

Blah blah blah.

**5.1.2 Module 2 Interface**

Blah blah blah.

**5.2 Process Interface**

**5.2.1 Process 1 Interface**

Blah blah blah.

**5.2.2 Process 2 Interface**

Blah blah blah.

**6.** **Detailed Design**

**6.1 Module Detailed Design**

**6.1.1 Module 1 Detail**

Blah blah blah.

**6.1.2 Module 2 Detail**

Blah blah blah.

**6.2 Data Detailed Design**

**6.2.1 Data Entity 1 Detail**

Blah blah blah.

**6.2.2 Data Entity 2 Detail**

Blah blah blah.