2016

VeriHandy System Design Document

Built for Shawn Squire

Anil Kendir – Team Facilitator

Matthew Wheeler – Requirements Leader

Justus Jackson – Implementation and Testing Leader

Dean Fleming – Design Leader

Tristan Adams – Delivery Leader

VeriHandy

System Design Document

#### Table of Contents

Page

## 1. Introduction 3

* 1. Purpose of This Document 3
  2. References 3

1. System Architecture 3

2.1 Architectural Design 3

2.2 Decomposition Description 4

3. Persistent Data Design 4

3.1 Database Descriptions 4

3.2 File Descriptions 4

1. Requirements Matrix 7

Appendix A – Agreement Between Customer and Contractor 8

Appendix B – Peer Review Sign-off 9

Appendix C – Document Contributions 10

1. **Introduction**

1.1 Purpose of This Document

This document serves as an overview of the design for the VeriHandy application, whose requirements are laid out in the System Requirements Document. Includes information about the architecture and database, as well as notes about how design will be carried out, and the relationships between parts of the architecture.

1.2 References

* VeriHandy System Design Document
* VeriHandy UI Design Document
* Laravel PHP Framework – found at <http://laravel.com/>, upon request of Shawn Squire

1. **System Architecture**

2.1 Architectural Design

VeriHandy Users

VeriHandy Server

Users use web pages to communicate with server and utilize VeriHandy

Server sends information using Laravel and displays using HTML web pages

*Figure 2.1.1 Basic Diagram showing the Client-Server Architecture in addition to some specifics of how the client-server relationship works for VeriHandy*

The VeriHandy system uses a basic Client-Server Architecture as shown in Figure 2.1.1. The current system will utilize the Laravel PHP Framework to assist this architecture and make implementing and testing the product easier.

The current architecture is responsible for communicating a variety of information between the user and the server. This information includes unique usernames, passwords, location information, contact information, and job requesting information. This can break down the design into a few main parts:

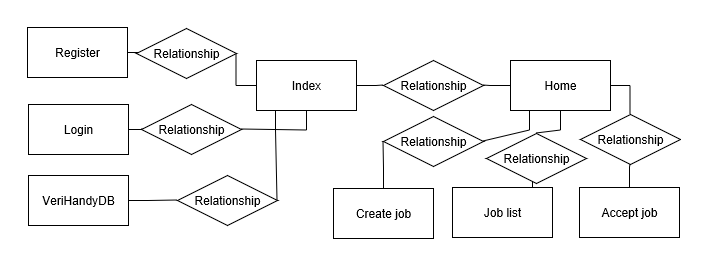
* Users
  + Interact with the server through a UI outlined in the VeriHandy UI Document
  + Information is requested by users, such as job listings and account information, from the server
  + Information is given by users, such as their ID, password, search parameters, and job information
* Server
  + Stores jobs, job information, user account information, and reviews
  + Communicates requests and receives to and from Users through the UI outlined in VeriHandy UI Document

2.2 Decomposition Description

Every component in our design is a .png image, an HTML file, a PHP file, or a .SQL file

These main components are currently broken down into acceptjob, create\_job, home, index, joblist, link, login, logout, register, review, review form, viewreview, and the VeriHandyDB

This Er diagram is incomplete but gives a fair understanding of how the files are interacting



Other – If you are using PHP or another language that cannot be adequately described using a structural decomposition diagram and/or class diagrams, create custom illustrations. Show implementation components of the system (e.g., scripts, files) and their relationships.

Regardless of the type of diagram(s) that you use, refer the reader to the diagram(s) and describe what it is intended to communicate. Give a brief description of each of the components. If you are using a pre-defined pattern (e.g., Model-View-Controller), explain this to the reader. [Two to three substantial paragraphs]

1. **Persistent Data Design**
   1. Database Descriptions

3.1.1 Jobs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| JobID | Int(11) | NO | PRI | NULL | Auto\_inc |
| CustomerID | Int(11) | NO |  | NULL |  |
| WorkerID | Int(11) | NO |  | NULL |  |
| Description | Varchar(512) | NO |  | NULL |  |
| Price | Decimal(12,2) | NO |  | NULL |  |
| Date | Date | NO |  | NULL |  |
| Time | Time | NO |  | NULL |  |
| Adress | Varchar(50) | NO |  | NULL |  |
| ZipCode | Int(11) | NO |  | NULL |  |
| Completed | Tinyint(1) | NO |  | 0 |  |

3.1.2 Reviews

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| JobID | Int(11) | NO | PRI | NULL |  |
| CustomerRating | Tinyint(4) | YES |  | NULL |  |
| CustomerReview | Varchar(256) | YES |  | NULL |  |
| WorkerRating | Tinyint(11) | YES |  | NULL |  |
| WorkerReview | Carchar(256) | YES |  | NULL |  |

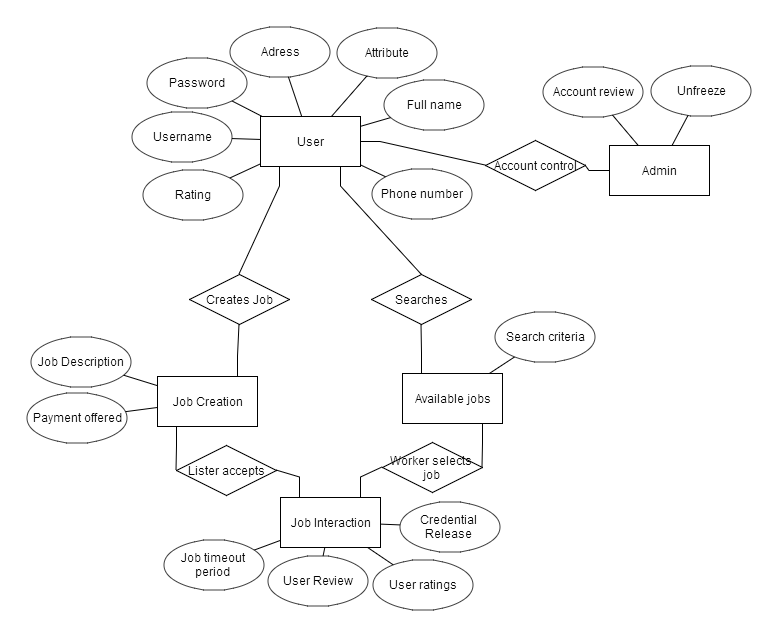
3.1.3 UserInfo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| UserID | Int(11) | NO | PRI | NULL |  |
| FirstName | Varchar(20) | NO |  | NULL |  |
| LastName | Varchar(20) | NO |  | NULL |  |
| Adress1 | Varchar(50) | NO |  | NULL |  |
| Adress2 | Varchar(50) | YES |  | NULL |  |
| City | Varchar(20) | NO |  | NULL |  |
| State | Varchar(20) | NO |  | NULL |  |
| Zipcode | Int(11) | NO |  | NULL |  |
| Email | Varchar(20) | NO |  | NULL |  |
| PhoneNumber | Int(11) | YES |  | NULL |  |

3.1.4 Users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| UserID | Int(11) | NO | PRI | NULL | Auto\_inc |
| UserName | Varchar(20) | NO |  | NULL |  |
| Password | Char(32) | NO |  | NULL |  |
| Type | Char(1) | NO |  | 0 |  |

3.1.5 ER Diagram of webpage interaction



* 1. File Descriptions

All information is stored using SQL tables as described in Section 3.

1. **Requirements Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Module | | | |
|  |  | SQL Database | HTML | CSS | PHP |
| Use Cases | 1, 2: Register/Login | Stores the User's Information | Allows user to enter information, and helps validate | Regulates page appearance | Communicates between SQL and HTML |
| 3, 4, 5, 6, 11, 12, 17: Job Related Cases | Stores both users' info as well as job info | Allows both users to enter the relevant data via fields and buttons | Regulates page appearance | Communicates between SQL and HTML |
| 7, 8, 14, 15, 16: Admin Actions | Stores info about users and admin permissions | Allows admin to fill various fields related to specific case | Regulates page appearance | Sends the info from HTML to SQL |
| 9, 10, 18: User Page Actions | Stores user info | Allows user to enter in new data and validates | Regulates page appearance | Sends changes to SQL from HTML |
| 13: Report | Stores the report info | Allows a user to enter in report which can only be seen by admins | Regulates page appearance | Sends report from HTML to SQL |

**Appendix A – Agreement Between Customer and Contractor**

In case of future changes to be made to the project/requirements, the following procedure will be used:

1. Mr. Shawn Squire and the VeriHandy Development Team, consisting of at least the Facilitator, Anil Kendir, and 2 other Team Members based on availability, will meet to discuss changes to be made to the requirements
2. Ideas, progress, and changes will be discussed between the Development Team and Mr. Shawn Squire and a new agreement will be reached.
3. The Development Team will revise this and other documents and update them as necessary.
4. A full set of signatures from Shawn Squire and the Development Team indicates that Mr. Shawn Squire and the Development Team agree on the requirements listed and design and development can proceed.

Customer Comments:

---------------------------------------------------------------------------------------------------------------------

Signatures:

Shawn Squire: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Anil Kendir: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team Member: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team Member: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team Member: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team Member: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Appendix B – Team Review Sign-off**

Signatures here indicate that all members of the team have reviewed and agreed upon the document’s content and format.

Team Comments:

---------------------------------------------------------------------------------------------------------------------

Signatures:

Anil Kendir: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Matt Wheeler: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Justus Jackson: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dean Fleming: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tristan Adams: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Appendix C – Document Contributions**

Tristan Adams

* Created file and added basic high level information for design
* Added Section 1
* Added Section 3.2
* Added the Requirements Matrix in Section 4
* Edited all sections

Anil Kendir

* Added Section 2
* Added Section 3.1
* Edited all sections