Health-Nut

Software Design Document

Name (s): Devin Akerley, Peter Paterson, Sam Sejan

Date: (05/02/2018)

TABLE OF CONTENTS

1. INTRODUCTION

2. SYSTEM OVERVIEW

3. SYSTEM ARCHITECTURE

4. DATA DESIGN

5. COMPONENT DESIGN

6. HUMAN INTERFACE DESIGN

7. REQUIREMENTS MATRIX

8. APPENDICES

Software Design Document



1. INTRODUCTION 1.1 Purpose

This Software Design Document describes the architecture and system design of ‘Health-Nut’. This document will outline the architecture, code, and overall general setup of the application.

1.2 Scope

The overall goal of this application will be to provide users with an easy to use website that will enable people to maintain a healthier lifestyle. The application will be setup to allow users to create profiles, search for healthy recipes, track workout routines, and access resources such as articles and links to other nutritional/health websites.

1.3 Overview

Each section of this document will present information about the different parts of the design process.

1.4 Reference Material

We did not use any references in our project except for tutorials sites for language learning.

1.5 Definitions and Acronyms

Cal(calories), DV(daily value), Amt(amount), Vit(vitamin), Hr(hour), Min(minute), Sec(second), g(gram), In(inch), pkg(package), gal(gallon), fl(fluid ounces), L(liter), mL(milliliter), doz(dozen), pt(pint), qt(quart), lb(pound), oz(ounce), c(cup), Tbsp(tablespoon), tsp(teaspoon), F(fahrenheit), C(celsius)

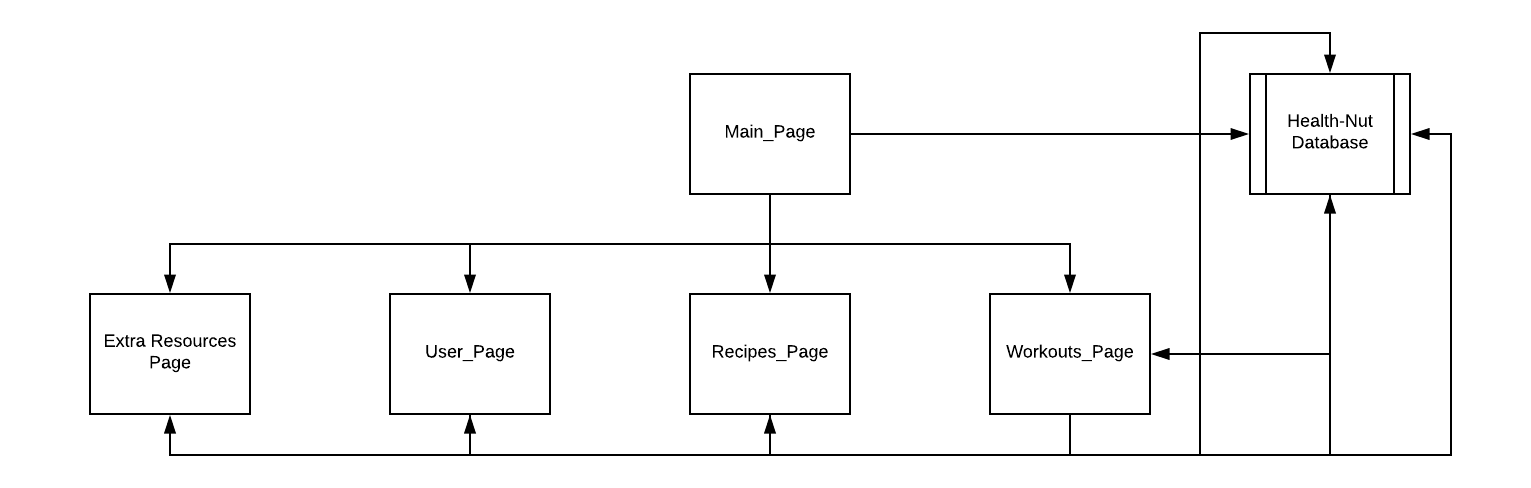
2. SYSTEM OVERVIEW

The website will be setup to have different pages which all link from a main homepage. Initially, all users will create a login, with a verified e-mail account, to access parts of the website outside of the main page. Each user will be allowed to search for recipes, exercise routines, and other information accessible on the site. Users will also be able to customize their searches and save recipes/routines that can be accessed from their user page. Users will also be able to upload recipes/routines to the website. All information will be stored in a back-end database using MySQL. The form processing to handle these requests will be created using PHP. The front-end of the website will be created using HTML and CSS for overall design, and JavaScript for any dynamic website material.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

The architecture of our system will have a structure with the following design:

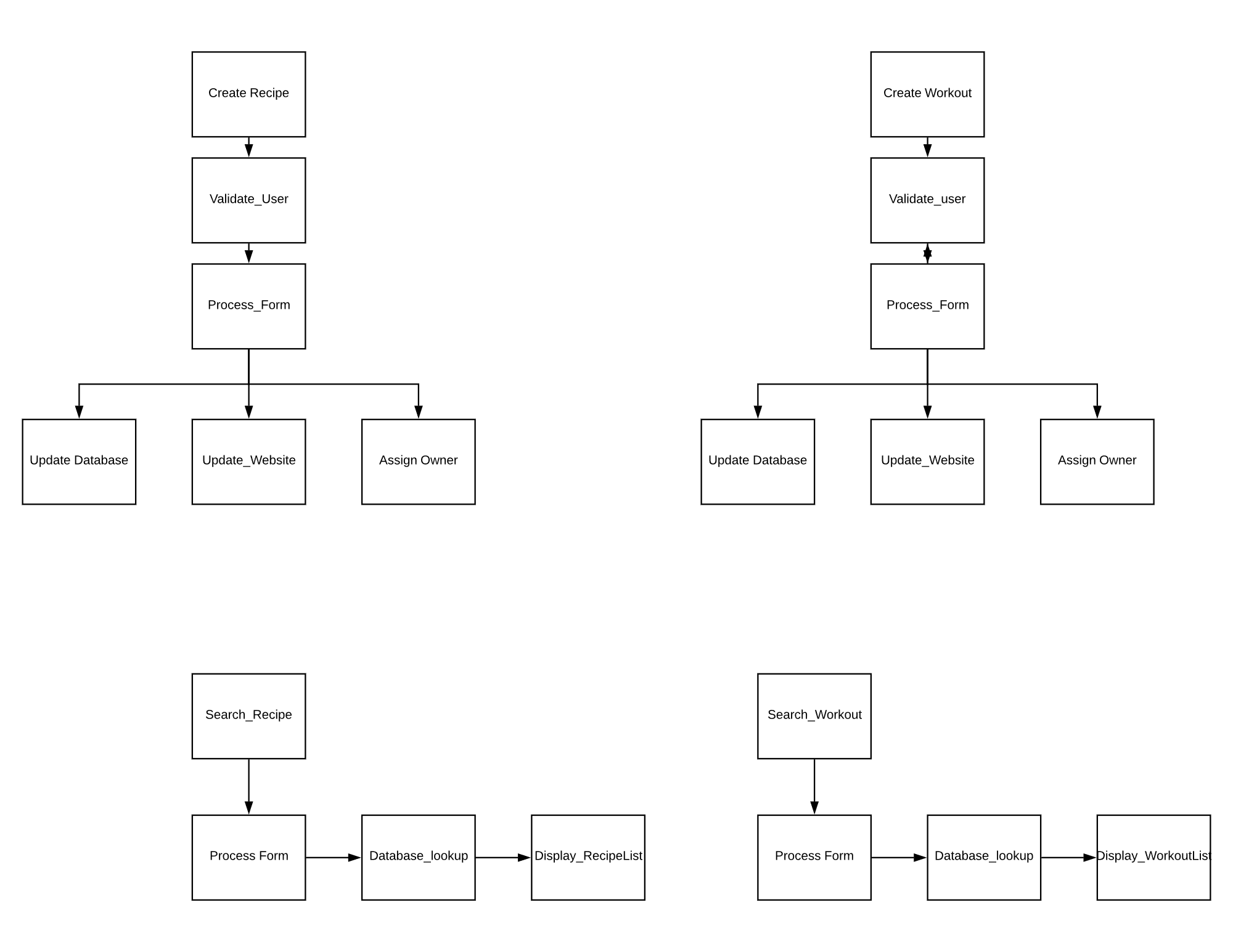


From the above figure:

The main page acts as the hub page for our website and can be used to access any other page on our site. Each page of the site can allow verified users to upload and update information that will be stored and processed using a MySQL database. All From processing will be done in PHP so that our site can protect hidden information from any unwanted users.

3.2 Decomposition Description

Chart of modules and interactions:



3.3 Design Rationale

Several factors were considered when selecting the system setup for our web application. The overall ‘multiple different pages’ setup was chosen to allow a more fluid design for the benefit of the user. Having data stored in an outside database allows information to be stored outside of the source code of the page. It will hide any information that is not meant to be accessible to all users. Using MySQL and PHP allows us to have a dynamic webpage, constantly updating with new data. The webpages will be comprised of code written in HTML, CSS, and JavaScript languages.

4. DATA DESIGN

4.1 Data Description

All information that applies to the users, recipes, and workouts will be stored in an outside database. All new data that will be uploaded to the page will be done via Form Processing using HTML and PHP which allows information to be accepted from forms. New information will then be sent to the database and the database will be updated automatically. Not all information will be accessible to all users. The users table, for example, cannot be fully accessed by any user thus providing security of personal information for all users.

4.2 Data Dictionary

Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in Section 3.2, list all the functions and function parameters. If you provided an OO description, list the objects and its attributes, methods and method parameters.

5. COMPONENT DESIGN

In this section, we take a closer look at what each component does in a more systematic way. If you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudocode. Describe any local data when necessary.

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

The user simply clicks the words at the top side of the page to get to the information that they want. They will also use search bars and links found on the page. The links and search bars will be user friendly enabling the user to find everything without any problems.

6.2 Screen Images



Image of a sample site above:

The site will have a similar design to the webpage above but will have different information and pictures. After the user logs in, the page will change and have more information and links on the right and left sides as well as the search bar at the top.

6.3 Screen Objects and Actions

The user will be able to click ‘login’ on the page which will lead to a different webpage. On that page they will have their goal on top and a place to input weight daily. The users’ saved recipes will be at the side of the page and will expand when clicked. Their saved workouts will be right under the recipe section using the same concept. On the left will be inspirational quotes that will change daily. The center of the page will have suggested meals. The top of the page will have a working search bar for specific foods, workouts, and smoothie recipes. Information on the vitamins and minerals, including their benefits to a healthy body, will be available with a link at the top of the page, so the user can learn more about the food they eat and foods to avoid.

7. REQUIREMENTS MATRIX

Provide a cross­reference that traces components and data structures to the requirements in your SRS document.

Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.