Software Design Document for Upgrade of Tenant and Owner Portals to Housing Authority Website

Version 1 to be approved

Prepared by Juan Rojas
Simon Bach
Michael Schleicher
Juan Orozco
Ravin Bhakta

Los Angeles County, Housing Authority

March 22, 2018
Table of Contents

Revision History

1. Introduction
   1.1. Purpose
   1.2. Document Conventions
   1.3. Intended Audience and Reading Suggestions
   1.4. System Overview

2. Design Considerations
   2.1. Assumptions and dependencies
   2.2. General Constraints
   2.3. Goals and Guidelines
   2.4. Development Methods

3. Architectural Strategies

4. System Architecture
   4.1. ...

5. Policies and Tactics
   5.1. Specific Products Used
   5.2. Requirements traceability
   5.3. Testing the software

6. Detailed System Design
   6.1 Main Module
      6.1.1 Responsibilities
      6.1.2 Constraints
      6.1.3 Composition
      6.1.4 Uses/Interactions
   6.2 Login Module
      6.2.1 Responsibilities
      6.2.2 Constraints
      6.2.3 Composition
      6.2.4 Uses/Interactions
   6.3 Registration Module
      6.3.1 Responsibilities
      6.3.2 Constraints
      6.3.3 Composition
      6.3.4 Uses/Interactions

7. Detailed Lower level Component Design
   7.1 Bootstrap
      7.1.1 Classification
      7.1.2 Processing Narrative
      7.1.3 Interface Description
      7.1.4 Processing Detail

8. User Interface
   8.1. Overview of User Interface
   8.2. Screen Frameworks or Images
   8.3. User Interface Flow Model
9. Database Design.................................................................pg 17
10. Requirements Validation and Verification........................pg 19
11. Glossary..............................................................................pg 22
12. References..............................................................................pg 22
## Revision History

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Reason For Changes</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Draft</td>
<td>9/8/2018</td>
<td>Plan and fill out some material</td>
<td>0.1</td>
</tr>
<tr>
<td>Secondary Draft</td>
<td>12/7/2018</td>
<td>Filled out more material</td>
<td>0.5</td>
</tr>
<tr>
<td>Third Draft</td>
<td>3/1/2019</td>
<td>Third version of draft</td>
<td>1.0</td>
</tr>
<tr>
<td>Final Draft</td>
<td>3/26/2019</td>
<td>Final draft</td>
<td>2.0</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Purpose
The purpose of this Software Design Document, SDD for short, is to outline the Design for the Portal rewrite of the County of Los Angeles Housing Authority portal. The scope of this project only includes the housing authority portal which both tenants and owners use. This portal will connect to an already existing website owned by the county.

1.2 Document Conventions
Every requirement will have its priority unless explicitly stated.

1.3 Intended Audience and Reading Suggestions
The intended audience for this SDD the students creating the project as well as the sponsors, in this case, the county of Los Angeles.

1.4 System Overview
The housing authority portal allows both tenants and owners to submit applications for affordable housing as well as check their information and a list of what needs to be done. The purpose of the redesign of the portal is to make it modern not only aesthetically but also functionally by using current web building practices.
2. Design Considerations

2.1 Assumptions and Dependencies

Assumptions and Dependencies regarding the software and its use:

- ASP.NET
- Entity Framework
- Microsoft SQL
- Web API

2.2 General Constraints

The software must be compatible with Los Angeles County web system. The Los Angeles County currently uses C# with ASP.NET therefore the current project must also adhere to these constraints.

2.3 Goals and Guidelines

The software must be intuitive for the community since it will be used for Housing Section 8 of Los Angeles County.

2.4 Development Methods

The formal development method used is Entity Framework - Database First Approach. The database will be created first, and the Model will be created from the database. The software related to the portal will be separated as frontend. Web API server will run in a separate project as backend.
3. Architectural Strategies

- Reuse of existing software components to implement various parts/features of the system
  - Rebuilding Database
  - Database migration
- Future plans for extending or enhancing the software
  - Laserfiche
- User interface paradigms (or system input and output models)
  - AJAX
    - Vue
  - restful API
  - model based form
- Hardware and/or software interface paradigms
  - SQL Server Management Studio
- Error detection and recovery
- Memory management policies
- External databases and/or data storage management and persistence
- Distributed data or control over a network
  - Load Balancers
- Generalized approaches to control
- Concurrency and synchronization
- Communication mechanisms
- Management of other resources
4. System Architecture

The **TOPHA** architecture is summarized in the Context Diagram (DFD Level 0). The Context Diagram displays the structure of the software modules.

The diagram is a straightforward diagram of how the website will work. The user will be able to access the website through any browser on a computer and phone as well. The user will enter the appropriate URL then will be redirected to the landing page (Main Controller). From the landing page the user has access to many different action such as logging in (login controller), registering for a new account (Registration Controller), or looking at the available forms (Form Controller). All these controller talk back to the Main Controller in the form of a WEB API, which is responsible for storing and retrieving information from the database.
5. Policies and Tactics

The main responsibility of this project is to update the Housing Authority’s website for tenants and owners. The design of the system has already been created thereby eradicating the need to create it. Taking a modern approach will difficult as the old system variables counteract with the new site that is being created.

5.1 Choice of which specific products used

- Microsoft Visual Studio 2017
- Microsoft SQL

5.2 Plans for ensuring requirements traceability

In order to ensure requirements traceability is by designing the portal to be intuitive for all users. Our goal is to make the portal to be more intuitive and improve performance from the current portal.

5.3 Plans for testing the software

- API
- Integration test

5.4 Code sharing and testing environment

The L.A. County provided all students with VMWare accounts that are directly connected to the Housing Authority as well as access to a testing environment.

5.5 Decoupling frontend from backend

In order to maintain code easier, we separated the project into two parts. The backend contains the Web APIs which are responsible with storing and receiving data from the database. The frontend contains the HTML pages with VUE js.
6. Detailed System Design

- Login Module
- Registration Module
- Main Module

6.1 Main Module

6.1.1 Responsibilities
The primary responsibility of the Main Module is to both store and retrieve data to/from the database. It also serves as a web page where the user can access all other pages that are part of the site.

6.1.2 Constraints
All controllers must be written in C# in ASP.NET with the entity framework.

6.1.3 Composition
The Main Controller contains all WEB APIs for the application. APIs included are, login, register, forgot password, forgot username, confirm account.

6.1.4 Uses/Interactions
This web page can be accessed by any user through a web browser.

6.2 Login Module

6.2.1 Responsibilities
The primary responsibility of the Login Controller is to allow the user to log in to their account with basic information such as email/username and password. This controller also contains links to help the user reset their password and have their username sent to their email incase they forgot it.

6.2.2 Constraints
All controllers must be written in C# in ASP.NET with the entity framework.
6.2.3 Composition
The Login Controller is just composed of a simple HTML page with javascript.

6.2.4 Uses/Interactions
This web page can be accessed by any user through a web browser.

6.3 Registration Module

6.3.1 Responsibilities
The primary responsibility of the Registration Module is to allow the user to register for an new account. In order to register the user must enter various required fields such as an email, username, phone, social security number, tenant code, and password. In order to prevent bots from creating accounts, a captcha has been implemented, where the user must correctly respond to in order to complete their registration. Once the user completes their registration, an email is sent to them where the must click on the link provided to verify their account.

6.2.2 Constraints
All controllers must be written in C# in ASP.NET with the entity framework.

6.2.3 Composition
The Registration Module is just composed of a simple HTML page with javascript.

6.2.4 Uses/Interactions
This web page can be accessed by any user through a web browser.
7. Detailed Lower level Component Design

7.1 Bootstrap

7.1.1 Classification
Bootstrap is a HTML, CSS, JS library

7.1.2 Processing Narrative
The purpose of Bootstrap is to style web pages and add functionality

7.1.3 Interface Description
Bootstrap is included in the front end of the project, along with all the views.

7.1.4 Processing Detail
Bootstrap is used in all HTML pages
8. User Interface

8.1 Overview of User Interface
The user will be able to access the website with any browser through computer or cellphone. On arrival the first page the user will see will be the landing page. On the landing page, the user has access to all the pages part of the website through the use of a navbar. In the landing page the user can log in or register. The registration page guides the user in creating an account and showing required fields along with messages that let the user know what is needed to advance. The user also has options to reset their password or obtain their username in case they forgot it. The user must enter their email and the user will immediately get an email from the County with instructions to reset their password or obtain their username. After the user logs in, they have access to their personal information and are allowed to fill out forms. Below are some screenshots of a registration page along with a sample form but are not in its finalized state.
8.2 Screen Frameworks or Images

Login

User Name

Password

Enter the text you see above:

Login

Forgot your password?

Registration

Tenant

User Information

Tenant ID

First Name

Last 4-digit of SSN

Last Name

Account Information

Username

Email

Confirm Email

Password

Confirm Password
8.3 User Interface Flow Model
On top of the webpage, a navbar contains links to all other pages that are part of the system.
9. Database Design

For this project, we had to convert data stored in an XML string stored in one Varchar(max) column. The goal was to convert the XML elements to a relational table. This would allow for redundant data to be eliminated. One example of the Request Voucher Extension form is shown below. One recurring problem faced was the formatting of data types. Since the database was initially in XML, data type formatting does not exist. In most cases there existed a couple of fields per form where the user was allowed to enter any text they liked. When migrating the database this caused problems because

The Request Voucher Extension form has a series of checkboxes that the user is instructed to select all that apply. This means that each form can have many options, and each option can be selected in many forms. To represent this in a database a many-to-many relationship is needed.

First, we created a new table of all the options.

- Finding suitable housing for family size
- Locating a unit affordable for your family
- Finding vacant units in the areas you wish to live
- Locating landlords and property owners willing to accept Section 8 vouchers
- Credit problems/evictions
- Transportation to view potential units and meet with landlords
- Lack of money to pay for credit checks and/or Lack of move-in money (deposit and 1st month rent)
- Health issues (reasons related to reasonable accommodations)
- Other:
<table>
<thead>
<tr>
<th>ReqVoucherExtReasonCodeD</th>
<th>ReqVoucherExtReasonCodeDesc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Credit problems/evictions</td>
</tr>
<tr>
<td>3</td>
<td>Finding suitable housing for family size</td>
</tr>
<tr>
<td>4</td>
<td>Finding vacant units in the areas you wish to live</td>
</tr>
<tr>
<td>5</td>
<td>Health issues</td>
</tr>
<tr>
<td>6</td>
<td>Health issues (reasons related to reasonable accom...</td>
</tr>
<tr>
<td>7</td>
<td>Lack of money to pay for credit checks</td>
</tr>
<tr>
<td>8</td>
<td>Lack of money to pay for credit checks and/or Lack...</td>
</tr>
<tr>
<td>9</td>
<td>Lack of move-in money (deposit and first month re...</td>
</tr>
<tr>
<td>10</td>
<td>Locating a unit affordable for your family</td>
</tr>
<tr>
<td>11</td>
<td>Locating landlords and property owners willing to...</td>
</tr>
<tr>
<td>12</td>
<td>Other:</td>
</tr>
<tr>
<td>13</td>
<td>Transportation to view potential units and meet wi...</td>
</tr>
</tbody>
</table>

Then we need an extra table that maps from this option table to the forms. So one column is the description and each description has a unique id number that pertains to it.
## 10. Requirements Validation and Verification

**Requirements Related to:**

1. **Registration Page**

<table>
<thead>
<tr>
<th>Requirement No.</th>
<th>Requirement Description</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.1-1</td>
<td>The Registration Page shall allow users to enter a Username.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-2</td>
<td>The Registration Page shall allow users to enter an Email address.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-3</td>
<td>The Registration Page shall allow users to set a Password.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-3.1</td>
<td>The Registration Page shall require users to repeat the Password.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-3.2</td>
<td>The Registration Page shall require the Passwords to match.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-4</td>
<td>The Registration Page shall allow users to enter a Full Name.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-5</td>
<td>The Registration Page shall allow users to enter a Phone Number.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-6</td>
<td>The Registration Page shall allow users to enter a Tax ID or SSN.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-7</td>
<td>The Registration Page shall allow users to select between Owner, Tenant, and Staff.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-7.1</td>
<td>The Registration Page shall allow users to enter an ID based on the option selected in 4.1.1-7.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-8</td>
<td>The Registration Page shall allow users to select between different departments.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-9</td>
<td>The Registration Page shall enforce a captcha before submission.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-10</td>
<td>The Registration Page shall check all inputs for wrong data format.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-11</td>
<td>The Registration Page shall store the user information to the database.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.1-12</td>
<td>The Registration Page shall display error messages to users for wrong data.</td>
<td>Yes</td>
</tr>
<tr>
<td>Requirement No.</td>
<td>Requirement Description</td>
<td>Completed</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>4.1.2-1</td>
<td>The Login Page shall allow users to enter a Username.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.2-2</td>
<td>The Login Page shall allow users to set a Password.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.2-2.1</td>
<td>The Login Page shall match the user’s Password set in 4.1.1-3 before granting access.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.2-3</td>
<td>The Login Page shall enforce a captcha before submission.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.2-4</td>
<td>The Login Page shall be mobile friendly.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Requirements Related to:

2. Login Page
Requirements Related to:

3. **Form Pages**

<table>
<thead>
<tr>
<th>Requirement No.</th>
<th>Requirement Description</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.3-1</td>
<td>Form Pages shall display all forms in a tabular view.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.3-2</td>
<td>Form Pages shall display instructional information in an expanding view.</td>
<td>Yes</td>
</tr>
<tr>
<td>4.1.3-3</td>
<td>Form Pages shall check inputs for wrong data input.</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>4.1.3-4</td>
<td>Form Pages shall store user information to the database.</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>4.1.3-5</td>
<td>Form Pages shall be mobile friendly.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
11. Definitions, Acronyms, and Abbreviations

<table>
<thead>
<tr>
<th>TOPHA</th>
<th>Tenant and Owner Portals to Housing Authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>

12. References

   https://csns.calstatela.edu/site/f18/cs4961-4/item/6817646