Lunar Logic Services

System Overview and Design

1. Introduction
2. System Overview
3. System Architecture
4. Data Design
5. Human Interface Design

Authored by: Jesse Wood, Casey DeDore, Logan Gosson

Introduction

The purpose of this web application is for users to easily find out more information about what services Lunar Logic provides. The web app is supposed to be interactive and fun, and allow a means for a user to “explore” what services are available.

The website will allow the user to select what services they are interested in and develop their own personal plan. Once they are satisfied with what they have selected, they will fill out a contact form that will send an email to Lunar Logic. Their contact information and the services they have selected will be sent in the email so that Lunar Logic can contact them with the appropriate information about the potential services that they may purchase.

System Overview

The system allows functionality for a user to explore what services are available. They are able to interact with a node system to reveal and obfuscate parent and child services as they go. They are able to select, or deselect, what services they desire at any time to add to their “Personal Plan.”

The system allows interaction with the graph by not only means of clicking or hovering on nodes, but also by interacting with the List of Services in the Personal Plan.

The system is web-based using ASP.NET with an MVC architecture. The system will utilize a JavaScript library known as Arbor.JS to display the node structure. The system will use Entity Framework for object relational mapping. The service data will be stored in the database and then transferred to the node structure by means of AJAX/JSON.

The system must allow for dynamic data entry, node population, and service list creation. The services that are provided could change at any time, so the system must be able to generate itself to allow for any services that are stored in the database.

The system will allow an Admin user to add new services, or change and delete them at any time.

System Architecture

Casey writes stuff here.

Data Design

Server Side

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Column Name | Data Type |
| Service | Object used to store service information and attributes of node functionality | Name  Description  ParentInclude  Selectable  ImageURL  ConnectedServices  ParentServices | STRING  STRING  BOOL  BOOL  STRING  COLLECTION  COLLECTION |

Client Side

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Column Name | Data Type |
| service | Object that server-side data model is translated into, and then eventually translated into object that arbor.js uses. | Name  Label  Desc  Selectable  Selected  Connected  Parent  Color  Shape  Imageurl | STRING  STRING  STRING  BOOL  BOOL  ARRAY  STRING  STRING  STRING  STRING |

Human Interface Design



