**What if Website**

**System Design Document**

##### Team Name

##### DePaul Slackers

##### Team

##### Arturo Alvarez

##### Sunny Syed

##### Austin Chase

##### Yongshang Tang

##### Tabish Ansari

##### Michael Weiss

# REVISION HISTORY

|  |  |  |
| --- | --- | --- |
| **Revision Number** | **Date** | **Comment** |
| 1.0 | 05/01/2017 | System Design Document Creation |
|  |  |  |
|  |  |  |
|  |  |  |

Table of Contents

[**REVISION HISTORY 2**](#_1hmsyys)

[**TABLE OF CONTENTS 3**](#_41mghml)

[**DOCUMENT OVERVIEW 4**](#_2grqrue)

[Scope 4](#_vx1227)

[Audience 4](#_3fwokq0)

[**SYSTEM OVERVIEW 4**](#_2u6wntf)

[Description 5](#_19c6y18)

[System Architecture 5](#_3tbugp1)

[*Software Architecture 5*](#_17dp8vu)

[*Hardware Architectures 5*](#_3rdcrjn)

[**HARDWARE DESIGN 5**](#_28h4qwu)

[Hardware Components 5](#_nmf14n)

[*Computer Systems 5*](#_35nkun2)

[*Peripherals 6*](#_1ksv4uv)

[*Networks 6*](#_44sinio)

[*Project Specific hardware items (e.g. Sensors, Transducers, Robotics, Enclosure Design) 6*](#_2jxsxqh)

[Hardware Integration 6](#_37m2jsg)

[*Logical Design 6*](#_3j2qqm3)

[*Physical Design 6*](#_1y810tw)

[*Recovery Design 6*](#_4i7ojhp)

[**SOFTWARE DESIGN 6**](#_1mrcu09)

[Software Packages 7](#_46r0co2)

[*{Software Module #1…N} 7*](#_3whwml4)

[Software Integration 7](#_2lwamvv)

[**DATA / DATABASE / FILES 7**](#_111kx3o)

[Data Flow Diagrams 7](#_3l18frh)

[Database Design 7](#_206ipza)

[Files 8](#_4k668n3)

[Registry / System Parameters 8](#_2zbgiuw)

[**SYSTEM INTERFACES 8**](#_1egqt2p)

[{XYZ Interface} 8](#_3ygebqi)

[**SYSTEM PERFORMANCE 8**](#_2dlolyb)

[**GLOSSARY / TERMINOLOGY 8**](#_sqyw64)

# DOCUMENT OVERVIEW

This is a technical blueprint for the project.

This document has been developed by the DePaul Slackers team for the ‘What If Website’ for DePaul University’s Masters in Computer Science and Masters in Information Systems. This document was developed from previous project documents (RAD, Use Cases and Project Scope) and is intended to satisfy all the customer requirements, objectives and expectations.

## SCOPE

Ubuntu Server 16.04

Python 2.7

Web2py 2.14.6

SQLite 3.0

Nginx- 1.13.0

Bootstrap 3.3.7

Git and Cron for continuous code integration

## AUDIENCE

University of DePaul (current or prospective) Graduate students seeking a Masters in Computer Science of Information Systems.

# SYSTEM OVERVIEW

This section deals with a summary of the overall system design aspects.

## DESCRIPTION

The purpose of the ‘What If’ website is to provide Graduate DePaul University Students seeking a Master’s Degree in either Computer Science of Information Systems with the different variation in options as it pertains to what classes they can take. What makes the website unique however, is the capability to provide students with different scenarios and variations of classes they can take to graduate quicker or to lengthen the process, thus providing the student with a clear picture as to what path and classes to take.

## SYSTEM ARCHITECTURE

This section includes high level overview of system including references to the items covered in System Architecture Document – SAD, and interfaces to other items such as hardware, peripherals and systems integration. If the hardware design is following architectural standards and buses , these are to be included here.

### Software Architecture

Ubuntu Server

Web2py

SQLite

Python

Nginx

Git

HTML

CSS

Bootstrap

Javascript

### Hardware Architectures

T2 micro instance

# HARDWARE DESIGN

In the following sections provide detailed discussion on the design and integration aspects for each hardware component. Discuss the hardware design criteria and approach including at least the internal system hardware components, customization, environmental requirements, target location, physical dimensions, configuration parameters, integration requirements, and other potential design information.

## HARDWARE COMPONENTS.

T2 Micro instance hosted and provided by AWS

Single core 1 GB of Ram 30 GB hard drive

Hosted in US-East to provide optimal connection and reliability

### Computer Systems

Ubuntu Server 16.04

### Peripherals

N/A

### Networks

N/A

# SOFTWARE DESIGN

Software and integration related detailed design aspects to be included here that are not already in the SAD. Discuss all internal software components, including COTS and their configuration. Provide detailed design for all software components being built including software integration.

## SOFTWARE PACKAGES

List and define all packages / modules. "Module" is synonymous with programs (libraries, executables, scripts, etc.)

### {Software Module #1…N}

This section may be expanded if appropriate for the project. There are multiple levels of design, both above this level (such as package dependencies) and below (such as class diagrams and associated code or code design aspects). This is where more detail could be provided; If some of this information has already been captured within the Software Architecture Document, only references should be included here.

Repeat for each module. Rename appropriately. Describe each module including purpose summary of functions, language / implementation approach, execution location, data definitions (inputs / outputs, references to data defined elsewhere, parameters), references to interfaces defined elsewhere, relationship to other modules, relationship to different types of users, error handling, diagrams, control, list of source files

## SOFTWARE INTEGRATION

Describe linkages that capture interactions among packages including COTS. Also include configuration and customization. For example, discuss middleware tools for OR Mapping here.

# DATA / DATABASE / FILES

## DATA FLOW DIAGRAM

## 

## DATABASE DESIGN

List and describe tables, fields, and entity relationships (also known as data dictionary and logical/ physical database design), schema, query language, key and indices, data management functions.

# SYSTEM INTERFACES

Define all external interactions between this system and other systems. Provide definition of the software and hardware interfaces between this system and other systems.

## {XYZ INTERFACE}

Include subsystems interfaces (for development, test and production). Repeat as needed. Name each section after the interface. Include description, hardware interrupts, triggering event, message protocol (or file format) and handshaking, record definitions /data definitions, timing restrictions/frequency, queuing/buffering, error identification /handling /recovery, priority, flow-control, data transfer rate, security and capacity /volume. Include additional software interface design aspects, such as the protocols or other interfaces (e.g., EAI) related to design as applicable.

# SYSTEM PERFORMANCE

Include all capacity and sizing calculations. Show how to calculate file and database sizes, system limits, and expected response times. Include reference to performance related executable architectures from System Architecture Document.

# GLOSSARY / TERMINOLOGY