SOFTWARE DEVELOPMENT PROPOSAL

Kenyon College
Software and Systems Design
MATH 348 - Spring 2020

CLIENT
City of Mount Vernon, Ohio

CREATED
2020 02 02

ABOUT US

We are a group of developers in the Kenyon College Software and Systems Design course (MATH 348), based in Gambier, Ohio, United States.

Our goal is to create a software system that fulfills your needs, while prioritizing speed, accuracy, and reliability. We are all looking forwards to working with you as we explore the latest technologies to create dynamic solutions for your problems.

Our development team consists of three developers and systems designers, Michaela Brydon, Shane Canfield, and Sejin Kim, under the guidance of Prof. Jim Skon.

GET IN TOUCH

Michaela Brydon *(software developer)* - C++14, Python, SQL, HTML/CSS/JS
brydon1@kenyon.edu

Shane Canfield *(software developer)* - C++14, Python
canfield1@kenyon.edu

Sejin Kim *(systems designer)* - C++14, Python, SQL, HTML/CSS/JS, Ubuntu/Debian and CentOS server administration
kim3@kenyon.edu

Jim Skon (*advisor*)
skonjp@kenyon.edu

PROJECT SCOPE

The Mount Vernon Department of Water needs a read-only archival access system to help facilitate the departments transition between system providers. They are currently using CMI's UtyX and will only be able to bring the past 3 years of customer data with them. The organization would like us to develop a database where one could search by address or unique account number and pull up all relevant information, specifically billing history and text file comment that are attached to each account. There needs to be standard availability as well as an emphasis on security. If there is extra time, they would like a report generation function that can return all relevant fields either as a .csv, .psv, .tsv, or a .txt document.

This project will follow the timeline outlined below and does not include any ongoing mantenance of the portal outside of what may be stated in the scope.

Brian Ball *(city engineer)*
engineer@mountvernonohio.org
(740) 393-9528

Mathias Orndorf *(utilities director)*
treatdistadmin@mountvernonohio.org

Terry Scott *(auditor)*
mtvauditor@mountvernonohio.org

Dynamic Networks (*IT consulting*)

PROPOSED SOLUTION

Technology Stack & Server Architecture

We *recommend* the following technology stack for platform development.

Server-Side Programming Language	C++14
Client-Side Programming Language	HTML5, CSS3, jQuery, Bootstrap 4.3
Database	MySQL 5.6
Web Server	Apache 2.x.x OR Nginx 1.x.x
Local Hosting	Oracle VirtualBox 6 VM running Ubuntu Server 18.04 LTS using Windows Server 2019 as the host
Cloud Storage	None recommended
Content Delivery Network (CDN)	None recommended
Session & Cache Storage	None recommended
Version Control System	Git

For security reasons, local hosting is recommended for hosting the platform, and the minimum server requirements are as mentioned below:

- LAMP Stack - *https://en.wikipedia.org/wiki/LAMP_(software_bundle)*
- Operating System - Linux x86, x86-64
- PHP version 7.0 or greater - *http://php.net*
- MySQL version 5.6 or greater - *https://www.mysql.com/*
- Apache 2.x.x (mod_rewrite module enabled, *https://httpd.apache.org/*) OR Nginx 1.x.x (*https//www.nginx.com/resources/wiki/*)
- Required PHP extensions
  - OpenSSL
  - XML
  - Ctype
  - JSON

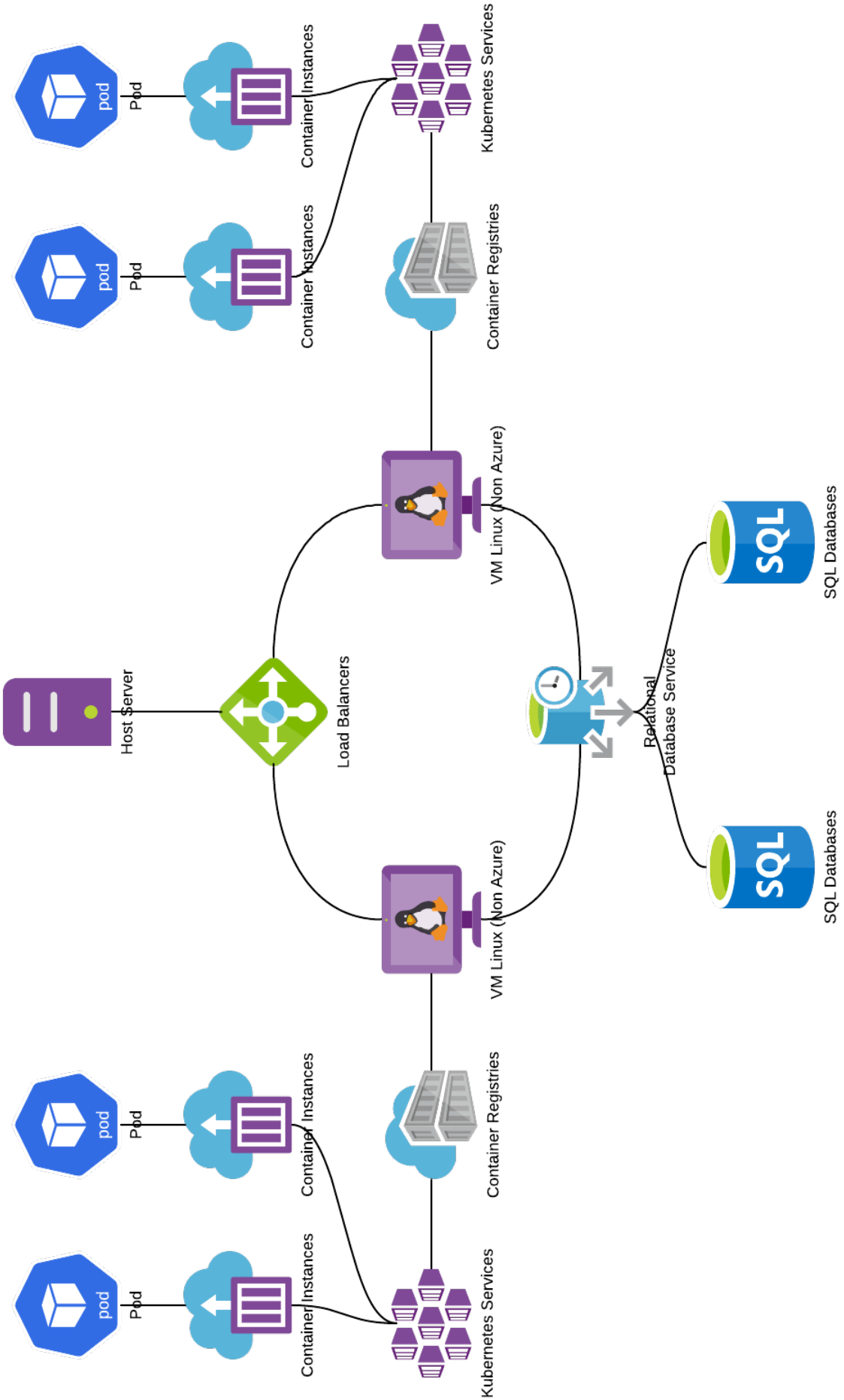
For better performance, reliability, security, and scalability, we *suggest* the following:

- Securing the website by SSL certificate

We *recommend* deploying the software in a virtual machine, using the Oracle VirtualBox hypervisor. This would allow you to run the Unix executable files on your Windows Server-based servers. We are only supporting deployments on bare metal with Windows Server 2012 or later, Ubuntu Server 16.04 LTS (xenial) or 18.04 LTS (bionic), Debian 10 (buster), and CentOS 7.

We are unable to support macOS Server deployments and will not support Windows Server 2008 or earlier.

The recommended high-level server architecture for the platform is as illustrated in the following diagram.

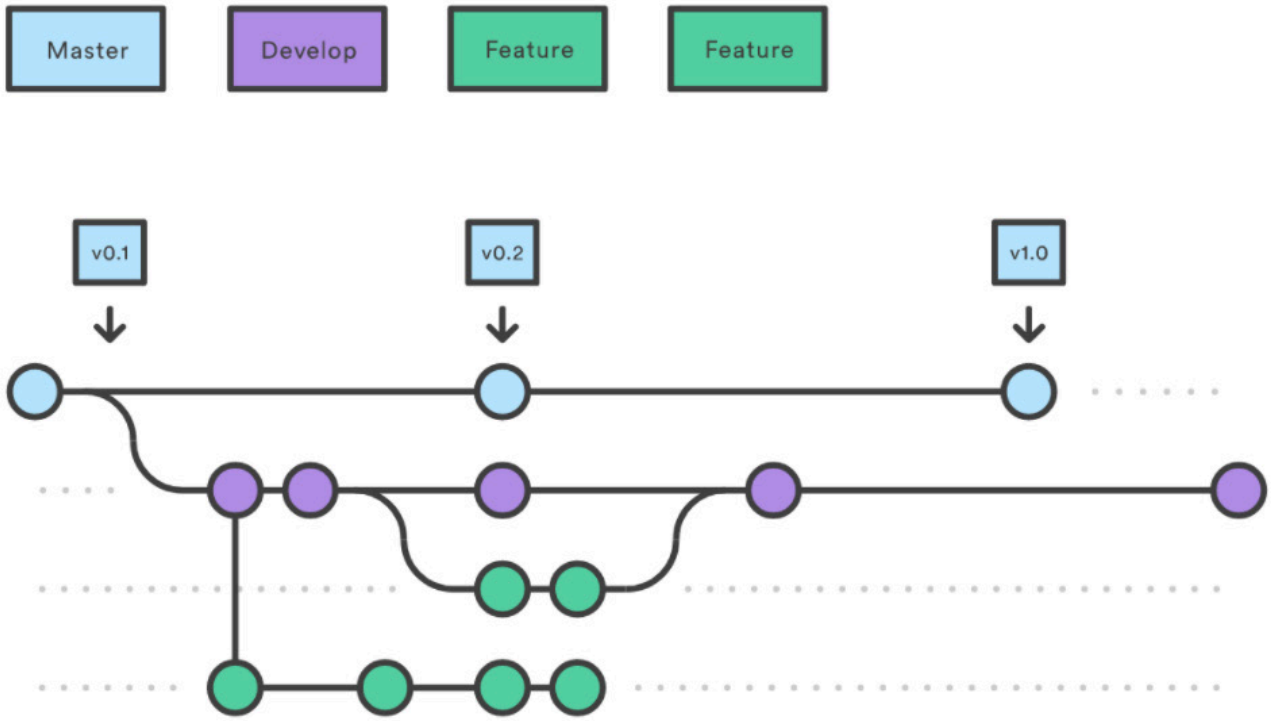

pod
Pod
Container Instances
Kubernetes Services
Container Registries
VM Linux (Non Azure)
Load Balancers
Host Server
Relational Database Service
SQL
SQL Databases

FEATURES OVERVIEW

- There will be two types of users namely:
  - Admin - Who can access and manage all aspects of the platform including all registered entries, and can edit entries
  - User - Who can view any information on the platform
- The user interface for the front-end (User area) of the platform will be responsive, but not necessarily mobile friendly, as the end product is intended for desktop use. It shall support the following resolutions:
  - 320px for common smartphones
  - 480px and below for common smartphones at the landscape orientaiton
  - 768px and 960px for tablets in their common orientations
  - 960px and upwards for desktop monitors
- The following pages will be designed:
  - Home page
  - View entries
  - Search entries

VERSION CONTROL

GitHub will be used as the online version control repository hosting service for tracking all development and coding changes. We usually follow the GitFlow workflow, which as a Master branch, hotfixes, release, develop, and feature branches.



DELIVERY

Deliverables will include the following:

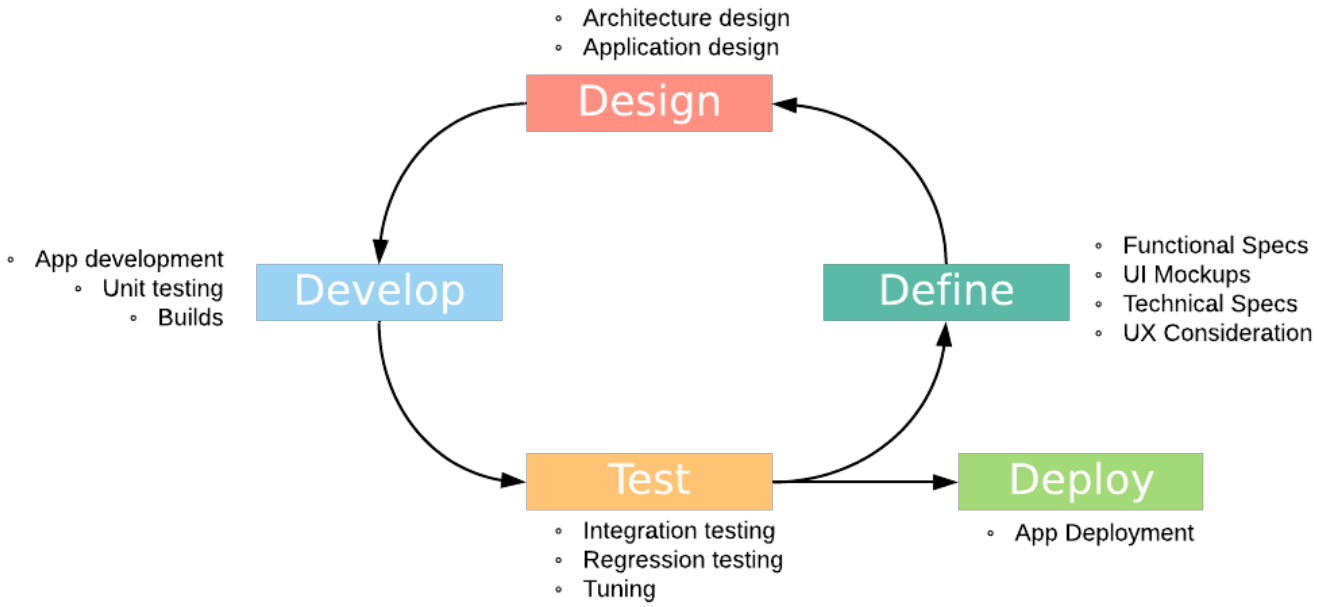
- Project plan.
- UI design mockups.
- Deployment on the requested domain.
- A backup copy of the files and database associated with the platform.
- Content/data addition and migration.

The following are excluded from the project scope:

- Logo design
- Mobile app (iOS and Android) for Admin uses and related RESTful AP/Web services

DEVELOPMENT METHOD

The project execution methodology which will be followed for this project design and development is the rapid development (cyclical) model. We find that the rapid develoment model is a great way to successfully manage projects that are relatively small in scope, with all functional and nonfunctional requirements defined in sufficient detail before any code is written.



TASK TRACKING

We use Trello for our task tracking and project management, and Slack to centralize our documentation and communications. We also manage source code on a private development server and hold our code on remote GitHub repositories, which can be found at *https://github.com/mvwater*.

CREATED
2020 02 02

Kenyon College
Software and Systems Design