Liam Cain

Matt Marakovits

Paul McCusker

Zack Meath

Matt Musich

IT Requirements Plan

**Server Platform**

* Physical System Requirements
  + 1 TB, RAID 1 iSCSI LUN
  + 2xL5630 (4 core, 8 threads)
* Virtual System Requirements
  + OS: CentOS 7.1
  + 2x Images
    - Web Server
      * 1 vCPU
      * 4 GB of RAM
      * 60 GB HDD
      * Node.js
    - Database Server
      * 4 vCPUs
      * 8 GB of RAM
      * 120 GB HDD
      * PostgreSQL
* Connectivity
  + 2x IP Addresses (one per server)
  + Would need connection to Marist LDAP\CAS System, Marist SQL and Marist TSM Server

**Reliability**

* Service needs to be up with the rest of the Marist systems (except for scheduled downtime)
* Service also needs to be responsive (as fast as the user’s internet can allow). There should be no need to be waiting for pages to load.

**Recoverability**

* Things will be backed up to Marist’s TSM server. TSM backups are completed once a day and are securely in Donnelly.
* Backups are accessible via the TSM client and can easily be restored by those who have the admin password to TSM.
* Any sort of temporary files are not of great importance, nor is the operating system. The database and Node application are what primarily need to be backed up.

**Security and Privacy**

* Database
  + Most people should NOT have access to the backend database. Once hitting production, only the Marist DBAs should have direct access to the database with an application specific database password to be used by the service.
  + Database access will be the primary goal of the web app, however, admissions will be able to update class listings given that they have the correct access. This will not be directly in the database itself, but part of the web app.
* Account information
  + User Data
    - Users should be able to create an account saved locally with this system. Ultimately, anyone who already has an existing Marist account should also be able to log into the site (like jobs.marist.edu is set up)
    - Having an account would allow users to recall saved course comparison. These would have to be saved in the database and accessible to the user and perhaps the admissions administrators.
  + FERPA considerations
    - FERPA considerations would be as per the Marist recommendation. This doesn’t change for this application.
  + Admin access controls
    - Controls such as adding and deleting users would be done within the web app itself. Ultimately as stated before, it would be integrated with the Marist CAS, so a LDAP group could be used instead.

**Maintenance**

* Database maintenance would be scheduled during normal IT outages due to the fact that the database ultimately would be hosted on the Marist SQL server. Information Technology is normally very good about announcing these outages far in advance and scheduling them during low periods of traffic.
* Updates to course information wouldn’t need scheduled downtime, if done correctly, it could be done live and the results seen immediately.
* Since this system currently does not have any direct ties to other admissions systems, it would not be unavailable if another one of admissions systems were to go down for whatever reason.