

Shared Core

Monday, January 30, 2017 9:35 AM

Every piece of code or design needs to be in Github as well as you need to keep notes and justification of your choices based on the textbook and any references (site your sources).

Application Authentication

- Each Application will need to work with OTS to enable CAS authentication (this is a plugin that allows you to use your @hawk account for access) (This will be covered in the coming weeks.)
- (PROD & TEST & DEV) This will be done via a trouble ticket to application services. You will be provided with a sandboxed portal by OTS and one of your representatives will need to work with OTS to get this configured. You will provide them your team's @hawk emails plus: (covered in the coming weeks as well)
 - hajek@iit.edu
 - trygstad@iit.edu
 - lmchughi@iit.edu
 - lenard@iit.edu
- This will handle authentication but you will need to handle account creation within the applications as well as a user profile that tracks site usage and or grants privileges to a logged in user.
- You will need to assign/create three user levels, ADMIN, USER, NON-AUTHENTICATED user. Your site must have different functionality based on what **type of User**.
- You will need to define a single site and make use of CSS to hide or expose elements.

--

 - Site needs to be Mobile/Responsive as well
- Site needs to develop at least 50 additional "users" with actual data for usage testing. Can be done with Faker.js or some comparable script to generate random but VALID data. (No test, test1, test1, testtesttest for example)
 - This means you need to identify your schemas and tables and what data goes into them first before you start coding
 - Datasets need to be created so that DEV & TEST can import this data and get to a system with running and working data.
 - Datasets need to be versioned and kept in Github

UI/UX

- Site needs an explanation of tools and design factors they decided and chose
- Site needs to have a Standards Guide (used for fonts, colors, and logos) placed in Github and updated/consistent
- Site needs to have a flow chart of operations (Visio or some other type of diagram)

Security

- Production site needs to have [Let's Encrypt](#) TLS/SSL site enabled and installed (very easy)
- Secure - all software package passwords need to be secure and software versions need to be at the latest possible (per OS) Firewalls need to be tight on every system.

Infrastructure

Team needs to choose a production stack which includes:

1. Entire TEST, DEV, and PROD systems must be scripted and deployable in an automated fashion.
2. Must provide a Build System for Continuous Delivery (we will cover this part soon)

- a. Jenkins or Snap/Go (from Thought Works) will be the software you can choose
 - b. This will provide rudimentary system tests
 - c. PHP, CSS, and or JavaScript Linters
 - d. If using compiled code - then compilers will be called at this point
- 3. Operating System - Normally the choice would be a bit larger but due to the cloud platform, I am introducing a constraint: Ubuntu 14.04.5 or CentOS 6.7 (if other needs come we can look at adding other systems)
- 4. Programming Language/Frameworks) PHP Symphony, NodeJS, Android, etc etc
- 5. Database (Some kind of database needs to be split into a sharded database, Master for Writes and Slave for Reads (application code needs to reflect this)
 - a. Database needs to be configured to receive remote connections but local firewall needs to be configured to prevent the database from being open to the world.
- 6. Presentation (Android/iOS or HTML5/CSS -- Mobile Responsive is a required feature)
- 7. (PROD) Load balancer will be deployed in production via the cloud platform Eucalyptus. (Dev will not have this)
 - a. DNS entry for ELB (Elastic Load Balancer) needs to be modified with OTS via support ticket (we will work with this when we come to production)
- 8. Need an admin page/portal feature that allows an Admin to backup and export/import datasets on a live an running system
 - a. Need an admin page/portal that allows for disabling of features (for instance an uploader) or set the site to READ-ONLY mode