Why - communicate decision of tools and processes chosen and why so that future needs or developers understand current position and options moving forward.

This living document helps tell the story of the how and why of the front end development for the product LeveUp. Front end development combines many topics and integrates with many other processes and teams. Topics have options and tradeoffs of tools and processes. Tools may also have dependencies and versioning. In some instances, decisions may be made without the need for options. Future needs or changes may lead to a change in tool, dependency, or process. This product information allows for a shorter time to market in making a change with greater confidence in understanding dependencies as well as, possibly, the whole system.

|  |  |  |
| --- | --- | --- |
| **TOPIC** | **DISCUSSION** | **OUTCOME** |
|  | Considering most well known tools/technology, patterns and practices |  |
| Client Framework | Angular  Version 2 and 4 had minimal changes, level set of all libraries included | Angular 4, supports internationalization |
| Dev App | Visual Studio 2017 - Pro costs, Community is free  Visual Studio Code - Free, needs host, used with Angular CLI has all needs for compiling, hosting and testing | Visual Studio Code with Angular CLI provides code generation, local dev host, minification and bundling, GitHub integration, provides internationalization support, app follows JavaScript application pattern and not ASP.Net web app  Detail any plugins and/or configs used in VS Code |
| Code Analysis | Automation of coding standards | TSLint is included with Angular CLI  Determine CSS and HTML linting |
| JavaScript Transpiler | In order to follow more strongly typed language with reuse of Object Oriented principles and intellisense, TypeScript will be used. | TypeScript comes with Angular CLI |
| JavaScript Libraries | jQuery is heavily used within the existing application. The direction is to remove as much as possible.  May need to review jQuery and dependency versions with current browsers if used.  Handlebars also used for client side templating engine and may be replaced completely with Angular |  |
| JavaScript Environment | Node | Node.js can be installed with NPM (Node package manager) and downloads dependent packages for the app.  Testing runners, Grunt/Gulp, Angular CLI depend on Node. |
| Front End Testing Framework | Mocha, Jasmine  Both are async | Jasmine with Karma (test runner) since QA also uses Ruby |
| Front End Integration Testing |  |  |
| CSS Build |  | Not existing for current application |
| CSS Framework | Bootstrap 3 | Preexisting with app, inline styles exist and will need to be added to local CSS |
| Devices/Browsers/Versions | Determine priority of device, browser and version to develop and test for | Targeting -  Desktop, tablet, mobile with priority accordingly.  Chrome, Safari, IE Edge/11, Firefox |
| Version Control | Company direction/standard | GitHub  Review commits to JIRA Work Type |
| Internationalization | xi18n support needed in data attributes of templates for changing languages.  Client profiles may need to store the language being used. | Angular CLI and Angular support, Angular expects separate builds per language  If one build, define strategy  Define intellisense capabilities during development |
| Ahead Of Time compilation (AOT) | Build/Transform/Transpile Angular | Angular supports AOT, Angular CLI supports with WebPack |
|  |  |  |
| Hosting Environment | Company direction/standard | AWS S3 static file delivery, no dynamic or application hosting |