Tony Clayton

Prework – Assignment #2

1. Strong understanding of **REST-based APIs** is critical for maintenance of the backend + middleware.  
     
   ***Representational state transfer****(****REST****) is an architectural style used for web development. Systems and sites designed using this style aim for fast performance, reliability and the ability to scale (to grow and easily support extra users). To achieve these goals, developers work with reusable components that can be managed and updated without affecting the system as a whole while it is running.*
2. Must have a thorough understanding of **SDLC.**  
   *The****systems development life cycle****(****SDLC****), also referred to as the****application development life-cycle****, is a term used in*[*systems engineering*](https://en.wikipedia.org/wiki/Systems_engineering)*,*[*information systems*](https://en.wikipedia.org/wiki/Information_system)*and*[*software engineering*](https://en.wikipedia.org/wiki/Software_engineering)*to describe a process for planning, creating, testing, and deploying an information system.*[*[1]*](https://en.wikipedia.org/wiki/Systems_development_life_cycle#cite_note-1)*The systems development life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.*
3. Expert knowledge of NoSQL, MySql, Oracle or other **RDBMS and its performance issues.**

*A****relational database management system****(****RDBMS****) is a*[*database management system*](https://en.wikipedia.org/wiki/Database_management_system)*(DBMS) that is based on the*[*relational model*](https://en.wikipedia.org/wiki/Relational_model)*. RDBMSs have been a common choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, and other applications since the 1980s. Relational databases have often replaced legacy*[*hierarchical databases*](https://en.wikipedia.org/wiki/Hierarchical_database)*and*[*network databases*](https://en.wikipedia.org/wiki/Network_database)*because they are easier to understand and use.*

1. Experience in **Agile delivery**, especially Scrum, Kanban or Scrumban as well as DevOps.

***Agile software development****describes a set of principles for*[*software development*](https://en.wikipedia.org/wiki/Software_development)*under which requirements and solutions evolve through the collaborative effort of self-organizing*[*cross-functional teams*](https://en.wikipedia.org/wiki/Cross-functional_team)*.*[*[1]*](https://en.wikipedia.org/wiki/Agile_software_development#cite_note-Collier_2011-1)*It promotes adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages rapid and flexible response to change.*[*[2]*](https://en.wikipedia.org/wiki/Agile_software_development#cite_note-2)*These principles support the definition and continuing evolution of many*[*software development methods*](https://en.wikipedia.org/wiki/Software_development_methodologies)

1. 4 years + with HTML 5, XHTML, jQuery, **AJAX,** CSS.

***Ajax****(short for****asynchronous***[***JavaScript***](https://en.wikipedia.org/wiki/JavaScript)***and***[***XML***](https://en.wikipedia.org/wiki/XML)*) is a set of*[*web development*](https://en.wikipedia.org/wiki/Web_development)*techniques using many web technologies on the*[*client-side*](https://en.wikipedia.org/wiki/Client-side)*to create*[*asynchronous*](https://en.wikipedia.org/wiki/Asynchronous_I/O)[*Web applications*](https://en.wikipedia.org/wiki/Web_application)*. With Ajax, web applications can send data to and retrieve from a*[*server*](https://en.wikipedia.org/wiki/Web_server)*asynchronously (in the background) without interfering with the display and behavior of the existing page.*

1. Front end experience using Single-Page Applications, Bootstrap, and any of the following will be helpful **OOCSS, SMACSS, MCSS.**

*Object Oriented CSS, Scaleable and Modular Architecture CSS, and Modular CSS are various frameworks in addition to normal CSS.*

1. Knowledge of communication/ integration tools such as Node, **N-Tier Architecture, Domain Driven Design**, and TeamCity.

*In*[*software engineering*](https://en.wikipedia.org/wiki/Software_engineering)*,****multitier architecture****(often referred to as****n-tier architecture****) is a*[*client–server architecture*](https://en.wikipedia.org/wiki/Client%E2%80%93server_architecture)*in which presentation, application processing, and data management functions are physically separated.*

1. Experience with **Babel, Gulp** and other tools useful for **Automation and Compiling.**

*Compilers used for minification and copying javascript files.*

1. Familiarity with **JIRA.**

***JIRA****is a*[*proprietary*](https://en.wikipedia.org/wiki/Proprietary_software)[*issue tracking*](https://en.wikipedia.org/wiki/Issue_tracking_system)*product, developed by [Atlassian](https://en.wikipedia.org/wiki/Atlassian" \o "Atlassian). It provides*[*bug tracking*](https://en.wikipedia.org/wiki/Bug_tracking_system)*, issue tracking, and*[*project management*](https://en.wikipedia.org/wiki/Project_management)*functions.*

1. Knowledge of **MVC frameworks** (Zend, Laravel, Yii)

***Model–view–controller****(****MVC****) is a software*[*architectural pattern*](https://en.wikipedia.org/wiki/Architectural_pattern)*for implementing*[*user interfaces*](https://en.wikipedia.org/wiki/User_interface)*on computers. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. Traditionally used for desktop*[*graphical user interfaces*](https://en.wikipedia.org/wiki/Graphical_user_interface)*(GUIs), this architecture has become popular for designing*[*web applications*](https://en.wikipedia.org/wiki/Web_application)*.*