Tech Approach FOR myMedications FDA Drug Interaction(FDADI) Application

Explain in a line what the content of the artifact say

Volume number 2

Submitted for: GSA Agile BPA

***Revision History***

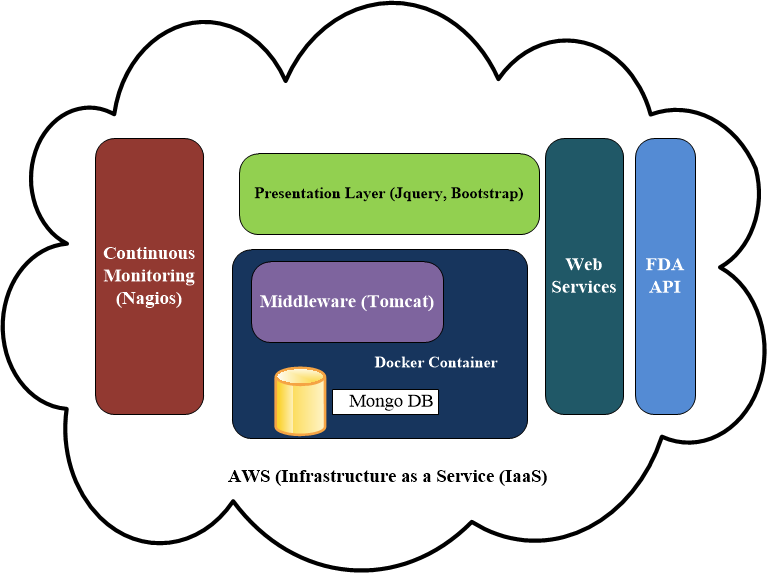
|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 6/18/15 | 1.0 | Initial Version | Gopal Kankanahalli |
| 6/19/15 | 1.1 | Updated | Gopal Kankanahalli |
| 6/30/15 | 1.2 | Updated language/formatting | Christina Paleczka |

***Intended Audience***

|  |
| --- |
| * Product Owner * Developer * Business Analysts |

***1.0 Technical Approach***

Exhibit 1 depicts a high-level technical architecture diagram of the myMedications Application( Food and Drug Administration (FDA) Drug Interaction (FDADI)) application deployed in the Amazon Web Services (AWS) cloud, which provides Infrastructure as a Service (IaaS) cloud infrastructure.



**Exhibit 1: A High-Level View of the Logical Architecture and its associated components**

The FDADI application is a three tier application with a presentation layer using jQuery, Bootstrap, CSS and HTML 5 technologies for front-end design. Bootstrap and jQuery were selected to facilitate Responsive Designs and User Centric Design. We validated the user interaction and other user interface centric questions by presenting wireframes to the Product Owner.

The Application utilizes Continuous Integration and Deployment using a variety of automated mechanisms such as Bamboo, and Continuous Integration Server, integrated with Maven Scripts with automated Static Analysis (Find Bugs), jUnit Tests, Selenium and other tests run as part of the continuous integration and deployment process. The Continuous Integration server is hosted in the AWS. Upon successful build completion, the software was automatically deployed to the AWS cloud where the Middleware infrastructure based on Apache Tomcat was deployed. Nagios Continuous Monitoring Server provides continuous monitoring of the Application, in addition to the continuous monitoring of the infrastructure which is provided as part of the Amazon AWS infrastructure.

The data for the myMedications application is obtained through RESTful web services calls to the FDA API to obtain data from <https://open.fda.gov>. The required data may be sourced from <https://api.fda.gov/drug>. The data exchange format is JSON.

***2.0 Development Concerns***

## *Decomposition*

## The application will be decomposed as a three tier application using the MVC pattern. The front end UI will use a responsive design approach using Bootstrap, jQuery, HTML5, CSS and related technologies. The middleware will be based on Spring MVC and the backend will utilize a MongoDB store.The application will be containerized using Docker and deployed on the AWS, Infrastructure as a Service (IaaS) using Maven and Git.*UI Controls*

The myMedications User Interface (UI) is built using Human Centric design ideas and responsive designs to support multiple modalities and multiple devices, such as Android, and iOS Tablets in addition to mobile phones. Primarily HTML5, CSS, jQuery, Bootstrap is used for the presentation layer design.

## *Architectural Runway Dependencies*

*There are no Dependencies identified in this epic.*

## *Third Party Frameworks*

See section on Reusable components.

## *Data Access and Storage*

We stored user data in the MongoDB. Adequate storage was provisioned to meet the estimated number of users. Encryption of data at rest has to be addressed.

## *Interfaces*

We used the RESTful API from <https://open.fda.gov>.

Are there any concerns that we need to identify here?

***3.0 Testing Concerns***

One testing concern was if the testing in Mobile emulators would be sufficient for testing. We also tested on android and iOS devices. The testing concern is that this may not be completely comprehensive since Mobile Device Management (MDM) interactions were not tested.

***4.0 Reusable Components***

(Identification of possible component reuse that is available including open source software, etc.)

|  |  |  |
| --- | --- | --- |
| Component | Functionality | License |
| Apache Tomcat | Application Server | Open Source |
| Spring MVC | Middleware framework | Open Source |
| Bootstrap | Presentation Framework | Open Source |
| Nagios | Monitoring Framework | Open Source |
| FindBugs | Static Analysis | Open Source |
| JUnit | Unit Testing | Open Source |
| Selenium | Web Front end Testing | Open Source |
| MongoDB | Database engine | Open Source |
| Docker | Container | Open Source |
| Logging SL4j | Logging | Open Source |
| Nessus Scanner | Vulnerability Scanning | Opensource |

***5.0 Security and Continuous monitoring***

Weekly Nessus Scans will identify the vulnerabilities. All vulnerabilities found will be recorded in JIRA as bugs and assigned to Sprints and tracked to completion. The team will ensure all the components are patched appropriately and are at the required patch levels with no serious vulnerabilities identified.

***6.0 Deployment Concerns***

## *Environment Dependencies*

We will deploy the application on docker containers running on AWS infrastructure.

***7.0 New Interfaces***

(

In future releases we need to address additional interfaces for data retrieval and consumption.

***8.0 PHI/PII Concerns***

Since the application stores PHI information, may have to consider multi-factor authentication and FIPS 140-2 compliant encryption for data at rest.in future releases.