

**WATI’s Technical Approach for Agile Development Pre-Qualified Vendor Pool for Digital Services**

West Advanced Technologies, Inc. (WATI), chose the Prototype B option to develop a California Express Alert System for Emergencies (**CEASE**), using Agile methodologies.

WATI, with its prior experience in developing applications using Agile-Scrum methodology for public sector agencies, has used several Agile practices during this short sprint cycle including:

* Sprint backlog;
* Continuous Integration;
* Iterative and Incremental Development;
* Refactoring;
* Timeboxing; and,
* User story / User Story mapping.

The methodologies used are fully comply with US Digital services playbook and are further detailed in the Github repository https://github.com/krishnawati/adpqproject

**The Application**

CEASE is developed to notify residents of California on alerts caused from natural disasters like Earth quake, floods, volcanoes, Tsunamis to name a few. CEASE allows California residents to establish and manage their profile and receive emergency and non-emergency notifications via email, based on the location and contact information provided in their profile. CEASE provides the authorized administrative users with the ability to publish notifications and track, and analyze and visualize related data.

The application provides Users to

* Signup and register and alternate options to register using their Gmail and Facebook profiles
* Create a user profile page with location and contact information
* Show data from Earthquakes (USGS)
  + http://sampleserver3.arcgisonline.com/arcgis/rest/services/Earthquakes/EarthquakesFromLastSeven
* To select event types they want to receive alerts for, within a specific vicinity
* Choose methods of communication
* Email interface
* SMS interface (not implemented)
* Automated phone call (not implemented)

The application has logic to polling service to get data from ESRI GIS in xml and has logic to match all the people who registered depending on the events type, location and vicinity. For the selected users the application sends out emails to all the matching registrants. The application sends alerts only for a new event and not for previously generated notified events. Additionally there is an admin screen for showing events selected for registered users and managing the users.

**The Team**

WATI has assembled a multidisciplinary and collaborative team:

* Subramanyam (Subra) Vaddi, PMP, CSM – Product Manager (Scrum Master)
* Vaibhav Kotla - Front End Web Developer
* Rohit K – DevOps Engineer
* Achaya Ramaswamy - Backend Web Developer
* Nagesh K -
* Krishna Chintalapathi – Interaction Designer/User Researcher/Usability Tester (Product Owner)

WATI assigned Subra Vaddi as the leader and gave that authority and responsibility and for development of CEASE prototype that is submitted in Github. Subra conducted daily Scrums and created new user stories and bugs based on feedback and usability testing. We held Sprint Reviews and Retrospectives and incorporated feedback into the iterative development.

**Discovery**

The Product Owner researched and studied the Netizens’ behavior on social media during the recent California Flooding and levy situation and designed the prototype by taking inputs from people who were directly impacted by the recent emergencies. The prototype incorporated items that people wished are available in a typical natural disaster situation. Our intent is to incorporate the alerts into social media as a means of mass alert system for the new generation Netizenry.

**Sprint (Design, Develop and Test)**

We used the Timebox and conducted only one sprint for the prototype. The application was developed using cooperative and iterative design by taking constant inputs to develop a customer centric prototype. For example, Users preferred easy registration process using Facebook/Gmail profile information.

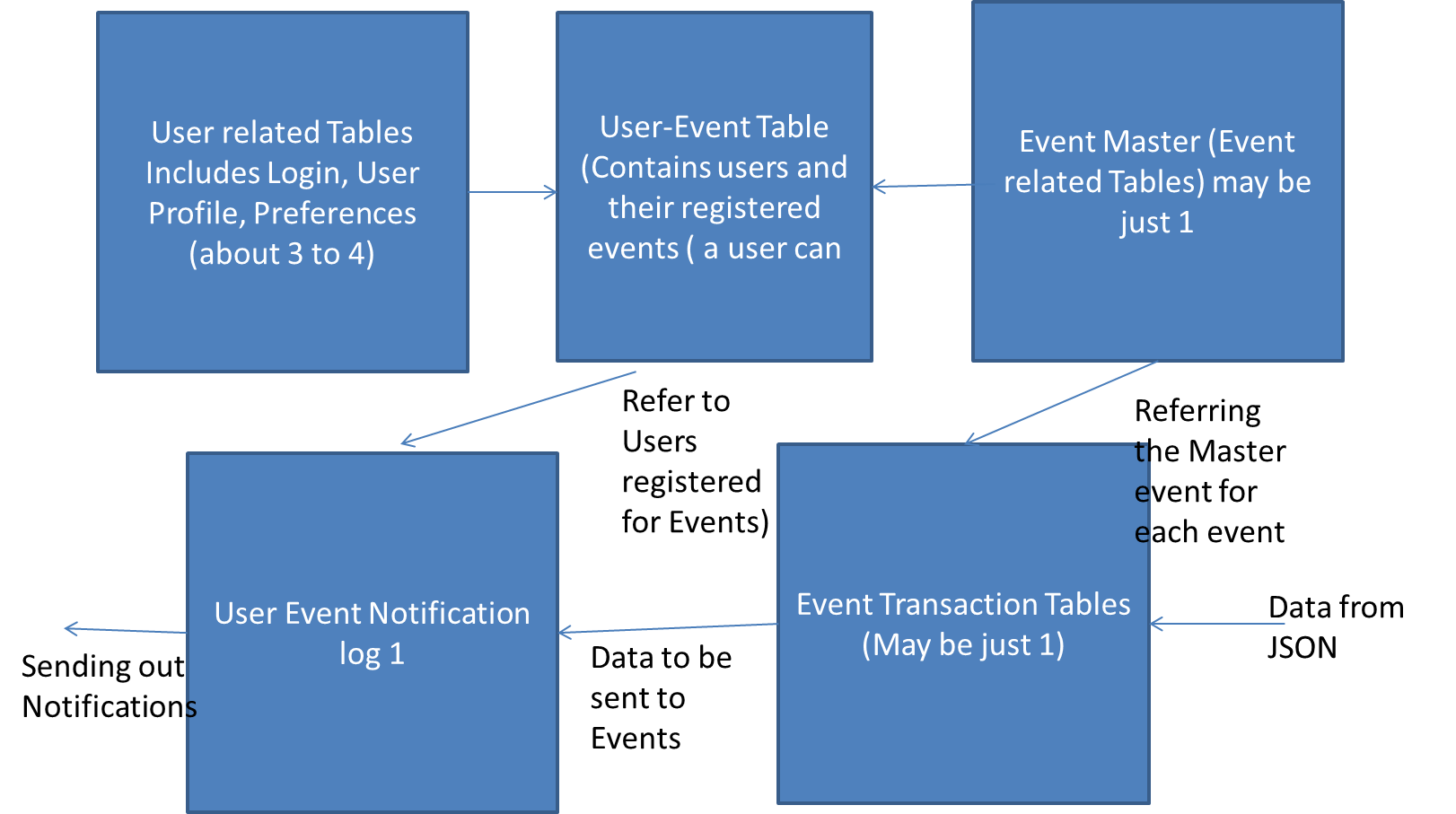
For rapid application development the team used refactoring approach by starting with a base code from <https://github.com/silverbux/laravel-angular-admin> and used GitHub to document code commits. The front end developer applying the Style Guide to create visual comps. The code was developed using Eclipse for PHP. The team ensured that the prototype complied with Section 508 of the Americans with Disabilities Act and WCAG 2.0 in selecting colors, fonts and design.

The prototype is created using all open source technologies, including

* HTML 5, PHP, and AngularJS for developing front end;
* MySQL database as storing the data; and,
* Laravel framework in PHP for developing business logic.

The prototype follows responsive design that works on multiple devices including Laptop, Tablet and Smart phones.

* The database developed in MySQL database contains the following table sets that were designed initially during the design sessions.



delivering early and often, which our engineers did by using responsive design pattern libraries to build components. Our approach was to get a minimum viable product (MVP) as quickly as possible to test it with users.

During the sprint the developers constructed PHPunit tests for the user stories and Tester executed functional tests. In addition to functional testing, we also performed accessibility testing.

The product owner and usability tester performed usability tests. Some automated tests using PHPUnit were conducted as well.

**Deployment**

The prototype is deployed on Amazon Web Services (AWS) on Linux environment onto Apache web server and the working prototype is available at : http://ec2-52-42-66-95.us-west-2.compute.amazonaws.com/#/login We deployed CEASE using our automated build pipeline. Jenkins CI server monitoring our configuration management tool, Git, for code changes. Once a code change was detected, it was compiled and tested. Jenkins then built the Docker container and deployed it to our production environment.

Like in most Agile development, we could not develop all the features that we intended to in the current Sprint (ex-Implementing SMS service). The non-developed items are moved to the top of the Sprint backlog