Play 1: Understand what people need	
Checklist	What we did
Early in the project, spend time with current and prospective users of the service	On the second day of the project we conducted User Surveys with nineteen participants from a broad demographic spectrum.
Use a range of qualitative and quantitative research methods to determine people’s goals, needs, and behaviors; be thoughtful about the time spent	Our User Surveys were conducted using a quantitative method which asked participants to select the title of our prototype and image for our Sign In page. We then conducted several rounds of usability testing of wireframes, mockups and our prototype using qualitative methods.
Test prototypes of solutions with real people, in the field if possible	We conducted usability testing with a variety of real users including State employee, Store Owner and simulated users. We received valuable feedback that we incorporated into User Stories and implemented.
Document the findings about user goals, needs, behaviors, and preferences	During our human centered design phase, we elicited feedback while conducting our wireframe testing to understand users goals, needs and behaviors. User stories were updated based this feedback.
Share findings with the team and agency leadership	We conducted daily Scrum meetings and feedback was captured in our Usability Testing documentation and posted to the GitHub repository.
Create a prioritized list of tasks the user is trying to accomplish, also known as "user stories"	Our user stories were created based on early end user feedback and a prioritized product backlog was managed in an agile development tool, Xplanner. Prioritizes were continually evaluated based on user feedback.
As the digital service is being built, regularly test it with potential users to ensure it meets people’s needs	Early wireframe and functioning prototypes were tested with potential users along with usability testing of each iteration.
Key Questions	
Who are your primary users?	We identified users to be any california resident or non-resident with internet access who wants to be informed of emergency and non-emergency notifications within the state of California
What user needs will this service address?	Notify users on up-to-date emergency and non-emergency situations that are actively occuring within the state of California
Why does the user want or need this service?	Users want to know about events that can affect their lives as soon as possible, wherever they may be e.g. work, shopping, school.
Which people will have the most difficulty with the service?	People who have little to no computer experience
Which research methods were used?	Surveys of general public, smalll focus groups, One-on-one interviews
What were the key findings?	Wanted a fast-acting alert system for areas in California that would affect their lives and those close to them.
How were the findings documented?	Findings were formally captured and shared with team members.

How often are you testing with real people?	Testing occurred every 3-4 days for this engagement.
Play 2: Address the whole experience, from start to finish	
Checklist	
Understand the different points at which people will interact with the service – both online and in person	We gathered information from users describing how they would potentially interact with this service online.
Identify pain points in the current way users interact with the service, and prioritize these according to user needs	Cuurently users have no way to receive emergency and non-emergency notifications with a personalized feel
Design the digital parts of the service so that they are integrated with the offline touch points people use to interact with the service	N/A to the prototype; however, in our planning and design we typically consider offline customer touch points, taking opportunities like including url's or QR codes in marketing material or using analysis of web site usage data to target customer outreach campaigns.
Develop metrics that will measure how well the service is a meeting user needs at each step of the service	During user testing sessions we measured easability of navigation, comprehension of content, and usability (needs identified by the users).
Key Questions	
What are the different ways (both online and offline) that people currently accomplish the task the digital service is designed to help with?	Most users are currently using websites such as news channels or Google to obtain California alerts.
Where are user pain points in the current way people accomplish the task?	Sources of information can be unclear, hard to find, or not specific to what the user wants to see.
Where does this specific project fit into the larger way people currently obtain the service being offered?	This will act as a confident and more direct source of personalized alerts from the state of California
What metrics will best indicate how well the service is working for its users?	Administrative users will have access to numerous types of charts that include how many users are creating accounts, actively using the service or not using the service; types of alerts that are being sent, etc.
Play 3: Make it simple and intuitive	
Checklist	
Use a simple and flexible design style guide for the service. Use the U.S. Web Design Standards as a default	We implemented a simple and responsive Design Style Guide adapted from the US Web Design Standards, as well as adhering to the goals and principles of Material Design.
Use the design style guide consistently for related digital services	The style guide is used consistently when designing application user interfaces (UI), and the UI is audited from adherence to the style guidelines.

Give users clear information about where they are in each step of the process	The application provides clear headings and subheadings, error messaging, and navigation so users know where they are in the create account and update information processes.
Follow accessibility best practices to ensure all people can use the service	We used Chrome accessibility on a continual basis to ensure WCAG 2.0 compliance.
Provide users with a way to exit and return later to complete the process	Users are able to create an account then set their locations and preferences at a later time. They are able to sign out and sign back in to make any desired changes to account and/or notification preferrences.
Use language that is familiar to the user and easy to understand	We used language that is familiar to users and received input regarding "geo location" being confusing. We responded with clearer language and clarification in the FAQs.
Use language and design consistently throughout the service, including online and offline touch points	We consistently used language and design throughout the service including online and offline email, text and push notifications.
Key Questions	
What primary tasks are the user trying to accomplish?	The user most importantly wants to have the ability to personalize their account in order to receive specific alerts that fit their interests
Is the language as plain and universal as possible?	Our Usability Testing verified we used universal and plain language.
What languages is your service offered in?	Our service is offered in both English and Spanish.
If a user needs help while using the service, how do they go about getting it?	We have a 'Contact Us' link in the footer whcich displays on every page. The user can send a message directly to the administration support or they have the option call to receive immediate service.
How does the service's design visually relate to other government services?	We incorporated the California State Seal on every page and named our application "Cal Notify".
Play 4: Build the service using agile and iterative practices	
Checklist	
Ship a functioning "minimum viable product" (MVP) that solves a core user need as soon as possible, no longer than three months from the beginning of the project, using a "beta" or "test" period if needed	We focused on delivering a MVP and moved additional low priority User Stories to the icebox.
Run usability tests frequently to see how well the service works and identify improvements that should be made	We ran usability tests every few days within each iteration.
Ensure the individuals building the service communicate closely using techniques such as launch meetings, war rooms, daily standups, and team chat tools	We conducted daily Scrum meetings and many design collaboration meetings to incorporate user feedback.
Keep delivery teams small and focused; limit organizational layers that separate these teams from the business owners	Our team comprised three developers, one delivery manager, one user researcher, one product manager, one writer and designer.

Release features and improvements multiple times each month	We conducted two sprints within the month.
Create a prioritized list of features and bugs, also known as the “feature backlog” and “bug backlog”	We created a prioritized list of features using User Stories and bug backlog list.
Use a source code version control system	We used Gitub for our source code repository.
Give the entire project team access to the issue tracker and version control system	The entire team was given access to Alfresco a version control system which included our issue tracking and project docmentation.
Use code reviews to ensure quality	Code reviews were conducted amongst developers.
Key Questions	
How long did it take to ship the MVP? If it hasn’t shipped yet, when will it?	MVP was delivered after two sprints.
How long does it take for a production deployment?	Our automated continuous integration tool completes a deployment within a few minutes.
How many days or weeks are in each iteration/sprint?	We conducted two weekly sprints.
Which version control system is being used?	GitHub is being used for code version control and Alfresco for document version control.
How are bugs tracked and tickets issued? What tool is used?	We used Bugzilla to track bugs and task developers.
How is the feature backlog managed? What tool is used?	We used Xplanner to manage our User Story backlog, tasks and burn down for this project.
How often do you review and reprioritize the feature and bug backlog?	User Stories and bugs were reviewed daily and reprioritized as needed.
How do you collect user feedback during development? How is that feedback used to improve the service?	We collected user feedback during development by involving users in each iteration through Usability Testing.
At each stage of usability testing, which gaps were identified in addressing user needs?	Users identified the need to filter on notifications online without creating an account and this was implemented and verified effective with users.
Play 5: Structure budgets and contracts to support delivery	
Checklist	
Budget includes research, discovery, and prototype activities	We created and managed an internal budget for this project.
Contract is structures to request frequent deliverables, not multi month milestones	N/A
Contract is structured to hold vendors accountable to deliverables	N/A
Contract gives the government delivery team enough flexibilty to adjust feature prioritization and delivery schedule as the project evolves	N/A
Contract ensures open source solutions are evaluated when technology choices are made	N/A

Contract specifies that software and data generated by third parties remains under our control, and can be reused and released to the public as appropriate and in accordance with the law	N/A
Contract allows us to use tools, services, and hosting from vendors with a variety of pricing models, including fixed fees and variable models like "pay-for-what-you-use" services	N/A
Contract specifies a warranty period where defects uncovered by the public are addressed by the vendor at no additional cost to the government	N/A
Contract includes a transition of servcies period and transition-out plan	N/A
Key Questions	
What is the scope of the project? What are the key deliverables?	N/A
What are the milestones? How frequent are they?	N/A
What are the performance metrics defined in the contract (e.g., response time, system uptime, time period to address priority issues)?	N/A
Play 6: Assign one leader and hold that person accountable	
Checklist	
A product owner has been identified	Caroline Brown-Pierce is identified as the product owner
All stakeholders agree that the product owner has the authority to assign tasks and make decisions about features and technical implementation details	Stakeholders understood the product owner had the authority to assign tasks and make decisions about features and technical implementation details
The product owner has a product management background with technical experience to assess alternatives and weigh tradeoffs	Caroline has over 12 years of technical and product management experience implementing and managing product development projects.
The product owner has a work plan that includes budget estimates and identifies funding sources	The product owner developed and maintained a work plan.
The product owner has a strong relationship with the contracting officer	N/A
Key Questions	
Who is the product owner?	Caroline Brown-Pierce
What organizational changes have been made to ensure the product owner has sufficient authority over and support for the project?	The product owner was given full authority to make decisions on features, priorities and task assignments.

What does it take for the product owner to add or remove a feature from the service?	The product owner had complete authority to add or remove features which was based on user feedback.
Play 7: Bring in experienced teams	
Checklist	
Member(s) of the team have experience building popular, high-traffic digital services	Many of our team members have experience building high traffic digital services.
Member(s) of the team have experience designing mobile and web applications	Our Visual Designer and Front and Backend Web Developers are experienced in designing responsive and modern mobile applications.
Member(s) of the team have experience using automated testing frameworks	Our team is experienced using both front end and back end automation tools. On this project we used Junit and Karma.
Member(s) of the team have experience with modern development and operations (DevOps) techniques like continuous integration and continuous deployment	Our team has implemented many applications with continuous integration and deployment techniques.
Member(s) of the team have experience securing digital services	Our team has security experts.
A federal contracting officer is on the internal team if a third party will be used for development work	N/A
A federal budget officer is on the internal team or is a partner	N/A
The appropriate privacy, civil liberties, and/or legal advisors for the department or agency is a partner	N/A
Play 8: Choose a modern technology stack	
Checklist	
Choose software frameworks that are commonly used by private-sector companies creating similar services	In our development we use frameworks that are commonly used by private sector software developers, such as Maven, Angular2 and Jersey.
Whenever possible, ensure that software can be deployed on a variety of commodity hardware types	We used Docker that will deploy on a variety of hardware platforms.
Ensure that each project has clear, understandable instructions for setting up a local development environment, and that team members can be quickly added or removed from projects	We have developed clear and understandable insturctions for setting up local development environment using Docker.
Consider open source software solutions at every layer of the stack	Our application layer stack is open source.
Key Questions	
What is your development stack and why did you choose it?	Client: CSS, Angular2, Bootstrap. Server: Tomcat, Jersey, GSON, Java. Single page web applications require no extra queries to the server to download pages, they provide improved user experience and performance is improved because computing is distributed and not all done on the server side.

Which databases are you using and why did you choose them?	Postgresql. This is feature, rich, reliable database, powerful spatial extensions.
How long does it take for a new team member to start developing?	Team members started developing on day one.
Play 9: Deploy in a flexible hosting environment	
Checklist	
Resources are provisioned on demand	We are using ECS from Amazon. Resource utilization triggers the resizing of resources
Resources scale based on real-time user demand	We are using ECS from Amazon. Resource utilization triggers the resizing of resources
Resources are provisioned through an API	All Amazon resources are reachable via API
Resources are available in multiple regions	Multiple regions are available
We only pay for resources we use	The auto scaler scales down during periods of low utilization.
Static assets are served through a content delivery network	All static assets are hosted on cloudfront
Application is hosted on commodity hardware	Application runs on ECS based on EC2 instance
Key Questions	
Where is your service hosted?	We are primarily in the Oregon zone
What hardware does your service use to run?	Amazon Web Services
What is the demand or usage pattern for your service?	Auto scalibility based on usage and need
What happens to your service when it experiences a surge in traffic or load?	The service automatically scales up
How much capacity is available in your hosting environment?	Amazons capacity
How long does it take you to provision a new resource, like an application server?	Scripts will be able to automatically deploy a new server in less than five minutes
How have you designed your service to scale based on demand?	The ECS service has been setup to scale on demand
How are you paying for your hosting infrastructure (e.g., by the minute, hourly, daily, monthly, fixed)?	We are chared on a utilization basis not on a time basis
Is your service hosted in multiple regions, availability zones, or data centers?	For the purposes of the prototype it is not. But could easily be setup to be such.
In the event of a catastrophic disaster to a datacenter, how long will it take to have the service operational?	Our deployment scripts can bring the service back to normal operation in less than five minutes.
What would be the impact of a prolonged downtime window?	Prolonged downtime would mean failure to meet contractual service level agreements, which would incur financial penalties. This is why we architect our systems to be fault tolerant and highly available.
What data redundancy do you have built into the system, and what would be the impact of a catastrophic data loss?	Automatic database snapshots once per day

How often do you need to contact a person from your hosting provider to get resources or to fix an issue?	This has not occoured
Play 10: Automate testing and deployments	
Checklist	
Create automated tests that verify all user-facing functionality	We used Junit and Selenium to verify functionality.
Create unit and integration tests to verify modules and components	We used Junit and Selenium for unit testing and integration testing
Run tests automatically as part of the build process	Tests run automatically as part of the Maven build. Maven is a framework for building, testing, packaging and deploying software projects.
Perform deployments automatically with deployment scripts, continuous delivery services, or similar techniques	We use AWS CodePipeline service to continuously and automatically test, build and package updates to the GitHub repository, and to deploy them to AWS Elastic Beanstalk test and production environments.
Conduct load and performance tests at regular intervals, including before public launch	In the actual production product we would conduct load and performance testing. We used AWS Cloud Monitoring to provide response information.
Key Questions	
What percentage of the code base is covered by automated tests?	Automated testing is conducted throughout the development of the code base.
How long does it take to build, test, and deploy a typical bug fix?	For a minor fix it can take only minutes.
How long does it take to build, test, and deploy a new feature into production?	1 sprint
How frequently are builds created? Each commit; Many times each day	Builds are created and committed several times a day.
What test tools are used?	Junit for unit test, Selenium for functional tests and Chrome for accessibility tests.
Which deployment automation or continuous integration tools are used?	Git, GiHub, and the following AWS services: CodePipeline, CodeBuild, CodeDeploy, Elastic Beanstalk
What is the estimated maximum number of concurrent users who will want to use the system	Ten thousand
How many simultaneous users could the system handle, according to the most recent capacity test?	Automatically scales on need
How does the service perform when you exceed the expected target usage volume? Does it degrade gracefully or catastrophically?	Automatically scales on need

What is your scaling strategy when demand increases suddenly	Automatically scales on need
Play 11: Manage security and privacy through reusable processes	
Checklist	
Contact the appropriate privacy or legal officer of the department or agency to determine whether a System of Records Notice (SORN), Privacy Impact Assessment, or other review should be conducted	As part of this RFI, we were not provided with a privacy or legal officer contact. However, these types of security and privacy deliberations are a standard part of our communications plan.
Determine, in consultation with a records officer, what data is collected and why, how it is used or shared, how it is stored and secured, and how long it is kept	As part of this RFI, we were not provided with a records officer contact. However, these types of records management deliberations are a standard part of our communications plan.
Determine, in consultation with a privacy specialist, whether and how users are notified about how personal information is collected and used, including whether a privacy policy is needed and where it should appear, and how users will be notified in the event of a security breach	Although not germain to this RFI, in practice, we would collaborate with an agency privacy specialist and agree to comply with agency privacy policy (if required) with respect to agency owned data, and we would document where it should appear and how users would be notified in the event of a security breach in our communications plan.
Consider whether the user should be able to access, delete, or remove the information from the service.	The user will have access to create and edit their account information. They will also be able to delete their account and information in the production version.
“Pre-certify” the hosting infrastructure used for the project using FedRAMP	We deploy applications in Amazon Web Services (AWS) environments. AWS infrastructure is FedRAMP compliant and pre-certified.
Use deployment scripts to ensure configuration of production environment remains consistent and controllable	We use deployment scripts in a continous integration process
Key Questions	
Does the service collect personal information from the user? How is the user notified of this collection?	Users will manually enter personal information when creating an account for this service therefore they are aware of the information that is being collected.
Does it collect more information than necessary? Could the data be used in ways an average user wouldn't expect?	The information collected is the minimum needed for the user to utilize the service.
How does a user access, correct, delete, or remove personal information?	Users will have access to create and edit their account information. They will also be able to delete their account and information in the production version.
Will any of the personal information stored in the system be shared with other services, people, or partners?	No
How and how often is the service tested for security vulnerabillities?	N/A - this is a prototype
How can someone from the public report a security issue?	There is a contact page.

Play 12: Use data to drive decisions	
Checklist	
Monitor system-level resource utilization in real time	We use AWS Cloud Watch to monitor resource utilization.
Monitor system performance in real-time (e.g. response time, latency, throughput, and error rates)	We use AWS Cloud Watch to monitor performance, latency etc.
Ensure monitoring can measure median, 95th percentile, and 98th percentile performance	We use AWS Cloud Watch to measure median performance .
Create automated alerts based on this monitoring	We use AWS Cloud Watch to monitor to create automated alerts.
Track concurrent users in real-time, and monitor user behaviors in the aggregate to determine how well the service meets user needs	We use Google Analytics to track site usage metrics, such as, Average Time on Page, Traffic Sources, Return Visit, Churn Rate. This was not applicable for the prototype app, but could easily be integrated with a production version of Cal Notify.
Publish metrics internally	Analytics dashboards are continuously update so that metrics can be visualized using web browsers and mobile devices.
Publish metrics externally	External metrics are not provided
Use an experimentation tool that supports multivariate testing in production	We use open source Selenium to script a wide variety of applications test. However, as a practice, we do not run tests against application is a production environment.
Key Questions	
What are the key metrics for services?	Request/Response Times, Usage, Active Users, Session time, Session Interval, Churn Rate. The prototype demonstrates the tracking and visual of usage statistics.
How are these metrics performed over the life of service?	Web statistics are gathered continuously through Google Analytics collection APIs.
Which monitoring system tools are in place?	Google Analytics.
what is the targeted average response time for your service? What percentage of requests take more then 1 second, 2 seconds, 4, seconds 8 seconds?	This varys by application, but, in general, the targeted response time for all requests is 1 second of less.
What is the average response time and percentile breakdown (percent taking more than 1s, 2s, 4s, and 8s) for the top 10 transactions?	We have not performed these tests for the prototype.
What is the volume of each of your service's top 10 transactions? What is the percentage of transactions started vs. completed?	We have not performed these tests for the prototype.
What is your service's monthly uptime target?	99.99%
What is your service's monthly uptime percentage, including scheduled maintenance?	N/A - this is a prototype

How does your team receive automated alerts when incidents occur?	Alerts are received by email and SMS.
How does your team respond to incidents? What is your post-mortem process?	A ticket is opened immediately upon report to track the incident. Action is taken within 2 hrs of the report. Dev team meets regularly to discuss recent incidents and how they can be prevented in the future. Bugs reports are create as needed and added to the sprint backlog for planning and prioritization.
Which tools are in place to measure user behavior?	AWS Kinesis to stream user behavior facts. AWS Redshift for manage data warehouse, and Quicksight for data visualization and analytics. This is not applicable to the prototype.
What tools or technologies are used for A/B testing?	Behavior experiments using Google Analytics. This is not applicable to the prototype.
How do you measure customer satisfaction?	By analyzing KPIs like, Usage, Active users, Session time, Session interval and Churn Rate.
Play 13: Default to open	
Checklist	
Offer users a mechanism to report bugs and issues, and be responsive to these reports	Bugs and issues are tracked via helpdesk ticketing system and bug report database.
Provide datasets to the public, in their entirety, through bulk downloads and API (applications programming interfaces)	N/A - this is a prototype
Ensure that data from the service is explicitly in the public domain, and that rights are waived globally via an international public domain dedication, such as the “Creative Commons Zero” waiver	N/A - this is a prototype
Catalog data in the agency's enterprise datainventory and add any public datasets to the agency's public data listing	N/A - this is a prototype
Ensure that we maintain the rights to all data developed by third parties in a manner that is releasable and reusable at no cost to the public	N/A - this is a prototype. However, an Agency would maintain the right to all data developed by third party. The data would be provided at no cost to the public in a format of the agency's specification.
Ensure that we maintain contractural rights to all custom software developed by third parties in a manner that is publishable and reusable at no cost	Agency will maintain contractural rights to all custom software, and it will be publishable and reusable at no cost.
When appropriate, create an API for third parties and internal users to interact with the service directly	N/A - this is a prototype
When appropriate, publish source code of projects or components online	Prototype source code is published online.

When appropriate, share your development process and progress publicly	Prototype development process and progress is publicly available.
Key Questions	
How are you collecting user feedback for bugs and issues?	Bugs and issues are tracked via helpdesk ticketing system and bug report database.
If there is an API what capabilities does it provide? Who uses it? How is it documented?	The prototype application uses a REST web API to connect single page web application to back end services and data. It is documented using Swagger. There are no external users of the API.
If the codebase has not been released under an open source license, explain why?	Code base is released under an open source license.
What components are made available to the public as open source?	All components are open source.
What datasets are made available to the public	N/A.