In British Columbia, wildfires resulted in 297 million dollars in government spending in 2015. Part of this large spending is due to the current inefficient response system to new wildfires. With our application, users are able to report new wildfires using the "report" tab on our web application, or by texting our SMS service run through IBM Watson. The users will input their phone number, name, and a description of the fire. Using Telus API's implemented on NodeRed via IBM Bluemix, we are able to quickly find users current or most recently seen location via cell phone tower triangulation. Telus send/receive text message API will then send them a text message to confirm that we have received their report. IBM Watson will interact with the users, intelligently parsing information from the user about the severity of the fire amongst other details. The TwilIO interface will provide user texting services. The government is notified in real time about all of the incoming information via TM Forum's trouble ticketing services. In the future, IBM Watson may be able to send intelligent UAVs to monitor the user’s area, providing live imaging, which can be prototyped on the mangOH board. The BC government may also be providing address data to be able to give feedback about what addresses and neighborhoods are within the "danger" radius.

Intelligently

With this definition in mind, we brainstormed ideas and created preliminary mock-ups and schematics regarding the function of our app. At first it was difficult to see how the entire process would come together, seeing as none of my team had used any of the technologies necessitated by the competition. We needed to learn to use, code, and interact with various technologies. These included REST, NodeRed, IBM BlueMix, IBM Watson, JSON, jQuery, JavaScript, HTML5, CSS3, Telus API’s, TM Forum API’s, SQL, among others. Much of the learning process occurred side by side, during design and implementation. Our preliminary mock-ups were not so different from our finished product, though we were astounded that we were able to cover such a grandiose breadth of topics and create a finished product. Near the top of the page, one can see a mock-up design/pseudo-UML.

Essentially, our schematic shows that users input their name, phone number, and a description of the fire into the website. The website parses this data, sends the phone number to a Telus API which gathers the location, and sends this location to a map page where real time user data can be seen. The data is then fed into a CloudAnt SQL database. Watson begins conversing with the user via TwilIO, after a Telus API sends them a confirmation message. All this excess data is also parsed into a JSON object which is stored in the SQL database. This data can be accessed raw via a view page.

Implementation was an extremely fast, yet rough process. Having no experience with these technologies prior, I needed to rely on my intuition, logic, and programming experience to tackle problems under high pressure and in small time frames. With many cups of coffee and 30 hours of fast paced coding with little rest – but lots of REST, my team and I were able to complete this project. Using GIT version control on IBM servers, we collaboratively took on one problem at a time. My initial focus was the implementation of the website and its interactive features. I finished a website prototype and ensured all of the logical aspects, implemented through NodeRed (and including various web programming languages) were working. I worked on the full stack, keeping in communication with event facilitators regarding their companies’ respective technologies. Seeing as my forte is logic, this was my major focus. I ensured all logical components meshed together smoothly and that data was being exchanged through our implementation correctly. This included learning new programming languages and solving many problems that I had never encountered before.

With me having the main working copy, the other four broke up into groups of one to two and worked on individual aspects. Our team included 5 people, Adam, Hamed, Ross, Sam, and I. Hamed focused on IBM Watson, with Ross implementing the Twilio service to interact with Watson. Adam worked on implementing the maps, and researching problems as we presented them to him, to increase workflow. Sam and I worked on the large flow of the entire app, implementing the back end functions and ensuring the calls were being executed successfully.

Its 2pm and we somehow have a working app. Wait, 2pm!? With two and a half hours to spare until the 4:30pm deadline!? This shocked us, yet certainly gave us a magnanimous advantage over the other groups. We tested the functions we wrote individually, with two separate use cases, then tested them together.

To test this case, we tried various different linguistic flows, injecting them into Watson and ensuring that they were all parsed correctly. Further, we algorithmically tested the SMS interface with numerous approaches and overloading to ensure it worked under stress.

The presentation went precisely as planned; everything worked! All of our features were demonstrated in real time and they all worked together perfectly. This required a small amount of luck and very intense testing. Although our coding was inherently rushed, we always ensured our logic was not flawed.

Creating this app, we were amazed that we were able to come together and create something with no prior experience in the technologies. Further, we had no prior planning or code. Somehow, we managed to win. With wildfires raging at this moment whence I am writing, I see that this app can revolutionize the way we collect wildfire data. It is also scalable, it can change the way we interact with emergency response systems in a fantastic way. The amount I learnt in those 30 hours trumps the amount one usually learns in weeks of coding class. I learnt that with the right team and the right dreams, one can implement exceptionally useful applications at fast speeds. I would like to thank all of the event organizers, judges, and all those others who were present both during and after the hackathon. It was an honor working with my team and competing with other programmers who had far more experience than I. I would like to especially thank Clifford and Stefania from the IBM team for being so exquisitely helpful at even the strangest hours of the night. I hope to continue learning and working with all the technologies I have learnt and furthering my skills.