CSC510 - PROGRESS REPORT

Team P - HAI Dash Board module
Tam Nguyen, Matt Farver, Sz Ting TZeng

Solution: "To Do Check List"

1. What has been done:

- A working IBM Cloudant NoSQL database has been set up and shared among members. It is a live, online database where changes will be reflected immediately in the dashboard as well as the mobile client. Json schemas were also designed.
- A working example of add, modify, display entries from Cloudant DB was also tested successfully
- Due to the fact that we do not have the resources to buy the indoor precision tracking system PoZyx to use with this project, we decided to simulate the physical position changes by modify database entries directly. A dashboard was created to show the stats of each nurse when s/he is finishing up tasks or moving between rooms. The dashboard also displays the threat map and the google trends
- A mobile client was also developed where nurses can view and keep track of their remaining tasks. An installation file (apk) was produced for Android.
- A video demo of basic functionalities were made and the link to it was posted on our github page
- An Agile driven tool chain was set up to automate versioning → build → test → deploy (for the nodeJS + cloudant part)
- A message server is set up to handle push-notifications to mobile devices

2. What is still under progress:

- Algorithm for calculating total threat score needs to be refined
- Depending on what parameters to be added or removed from total threat score equation, database json schema will be changed
- Automation of the simulator needs to be further developed. We need to see virtual nurses move and tasks list got updated accordingly leading to dynamic threat score changes by the minutes. Because there are uncertainty about what parameters to be used, how the score will be calculated and how data got randomized (we do not want to go total random because in order for the simulation to make sense, values cannot be totally random)
- More research needs to be done on how to use IBM testing console name SauceLabs and integrate SauceLab well with the tool chain.

- Build push-notifications based on missing task statistics

3.Roadblocks:

- IBM is migrating its Cloud Foundry infrastructure from DEA to Diego leading to unstable access to its cloud service control interface. For example, the database is working but one time, we couldn't get into the database control interface to make changes to the schemas. Same thing happened with our Node.JS, and mobile builder interface. At one time, we ran into 404 page
- IBM also is spinning off its cloud mobile builder to be a separate company called kinetise that will charge $10 for each time we want to download the source codes to local computer.

4. Plan for user testing / survey:

- With our current demo videos, I strongly believe that we can demonstrate to users how the app works and get their feedbacks. We crafted several scenarios to highlight the features of both the dash board and the mobile app

Solution: "HAI Threat Map"

1. What is the solution:

The threat map application is designed to help hospital staff allocate resources where they are needed the most in an effort to reduce healthcare associated infections (HAI). During our research, we discovered that a majority of hospital workers indicated that a lack of time and resources prevented them from following strict HAI protocols; with this information in mind, we wanted to develop a tool that would focus hospital staffs attention where HAI are most likely to occur. The threat map works by taking data that is already generated within the hospital to create a threat level for each patient. Each piece of data is used as a variable which is given a numeric value and aggregated to provide a threat level for each patient. Examples of the variables used to create a threat level include patient information and their reason for being at the hospital, length of stay, the number of visitors, and the last time the patient's room was cleaned as well as others variables. We gave most variables their own rating on a scale from 1 to 10, with 1 indicating low risk and 10 indicating very high risk. Other variables were given a score based on time, for example the last time a patient's room was cleaned would continue to increase by one for every hour that is remained un-clean and would be reset back to zero when the room is cleaned. Once all variables have a rating, they are aggregated together to provide an over threat level. The threat levels range from low, medium, and high, and are meant to indicate which patients are at a higher risk for contracting an HAI, this gives hospital staff an indication of where they need to focus their resources.

2. Work done, work in progress and roadblocks:

Work completed on the threat map application to date include the user interface and the database that feeds the threat map data which is needed to generate the threat levels for each patient. We have also created a web based version as well as a mobile version of our applications. We are currently in the process of generating a demo that simulates how the threat map works, this demo also includes our other applications. While we have a functioning threat map application, there are requirements that still need to be addressed. Although our database is linked to the application, we need to manually refresh the screen each time data is modified. To provide increased functionality we would like for our interface to refresh automatically once data has changed; this requirement will allow for our application to be hands free and would not require hospital staff to refresh the screen every five minutes. An issue that may not be solved within this class is our own knowledge of creating accurate threat levels. While this is not a programming problem it does create a lability for our application. From our limited research we have a basic understanding of some of the variables that contribute to HAIs. However, we do not know how these variables should be weighed against each other to provide and accurate overall threat level.

Solution: Real-time threat analytic system:

A searching trends feature embedded in website which allows users to query any trends from google in 3 years. This feature may helps users to identify how the user searching in specific region changes and therefore indicates possible breakout

1. What has been done:
   - A backup php website with incomplete draft solutions
   - Setting for android push notification on Bluemix

2. What is still under progress:
   - Formulas to define an ascending trend that might indicates a breakout
   - Build push notifications based on ascending trends of infectious diseases

3. Roadblocks:
   - IBM migrating issues to some extent interrupt the development of our project.
   - Sag is the only client of Cloudant DB for php, but to integrate cloudant and php with google trends api there’re still some issues needed to be fixed.
   - Apple requires developers to purchase developer account for push notifications feature