Technical Stack Evaluation

Submitted To - Prof Pattrick Hill
Submitted By - Siddharth Gulati(CWID - 10468179)

1. Logging

a. What is the purpose of logging and a logging tool
   i. Logging is an essential part of any software development. Collecting, analyzing, and monitoring these logs can help developers target areas of poor performance, assess application health, improve troubleshooting, and optimize the root cause analysis of application performance errors
   ii. Manual logging is only possible for small scale applications. It involves many processes from collecting data, storing data and analyzing data.
   iii. Logging management tool can automate these tasks and will help enterprises to take help of these logs and analyze that so that they can work on improving their product

b. Tech Stack for a logging application
   i. A simple React and NodeJS application will be created so that use can have a GUI where user will be able to store and retrieve the log entries
   ii. Users data will be stored in firebase realtime data which will also help in user authentication
   iii. On the backend we will use aws timestream as it used timestamp as the primary key and will be beneficial when handling logs
   iv. Since this system will require high amount of querying we will use elasticsearch by aws to make the process of querying log entries efficient
   v. A queuing system like redis will also help as without a queuing system it becomes almost impossible to upgrade the Elasticsearch cluster because there is no way to store data during critical cluster upgrades.

c. Application Specific Questions
   i. How would you store your log entries?
      - We will use aws timestream database to store the log entries
   ii. How would you allow users to submit log entries?
      - The web application will provide the user interface to store a custom entry
   iii. How would you allow them to query log entries?

- To query the log entries we will use aws elasticsearch which can do fast querying with already fast timestream db

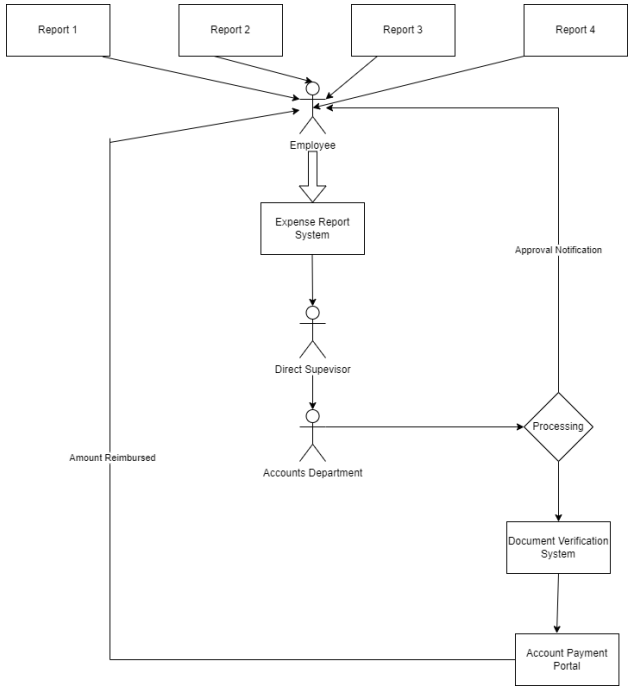
iv. How would you allow them to see their log entries?
- Custom filters to check on log entries will be added in the user interface

v. What would be your web server?
- The application will be deployed on aws EC2

2. Expense Report

a. What is the purpose of having an expense report application
   i. Expense tracking applications help companies to track and organize work expenses
   ii. This also helps employees to request for reimbursement for business related expenditure
   iii. This also comes in handy during tax returns

b. Tech Stack for Expense Report Application
   i. FrontEnd
      - Language
         a. JavaScript - To create dynamic web pages
         b. HTML - To write web pages
         c. SCSS - Add styling to web pages
      - Tool(s)
         a. VueJS - Popular and easy to learn web framework
         b. TSLint - For standardization of code
         c. Jest - Easy to use testing framework
      - Infrastructure
         a. Vercel - Reliable beginner hosting platform
   ii. Backend
      - Language
         a. JavaScript
      - Tool(s)
         a. ExpressJS - Lightweight framework for nodejs
         b. Cassandra - High performance database solution
         c. Jest/Mocha - Easy to use testing framework
         d. ChartsJS - Plug and play package to make good charts
      - Infrastructure
         a. AWS - Reliable, secure and scalable
   iii. Process Diagram



c. Application Specific Questions
    i. How would you store your expenses?
        - The application will utilize Cassandra - A NoSQL database specifically designed for storing large amounts of data. This data will include raw expense data as well as reports generated from that data
    ii. What web server would you choose, and why?
        - This application will be built on Express and will be deployed on AWS due it's low cost maintenance, scalability and reliability.
    iii. How would you handle the emails?
        - For this application a cron job will be written using node mailer along with gmail server to send out email regarding the reports
    iv. How would you handle the PDF generation?
        - This application will take help of a node package called pdfmake to generate pdf
    v. How are you going to handle all the templating for the web application?
        - We will be using VueJS which is a modern front end web framework built on JavaScript

3. A Twitter Streaming Safety Service

a. Need for Such Application

i. Twitter is a popular social media platform. Twitter cannot actually spot crime areas as criminals don't actually tweet this kind of activity. However according to a study published on techcrunch Your Tweets can help predict crime | TechCrunch there is a high correlation between the twitter activity density and crime rate. 19 out 25 times this method was effective.

b. Design of this application

i. Front end

- Ionic framework to make Cross-platform applications. Underlying technology will be ReactJS as it helps to write module based scalable applications. The benefit of this framework would be is that with single code we can have a PWA, an android application and an AppStore application
- Cordova will be used to make the application production ready

ii. Backend

- This application can be built on any rest api supported framework. We will be using expressjs for this application.
- We will also use twitter keyword targeting api to search tweets based on keywords
- Firebase will also be used for text based notifications as this is one of the most reliable service on the market and this application is heavily relied on such service
- This application will take help of redis-an in memory data structure to provide real time update on the incident report

iii. Database

- We will be using MongoDB for this application as keyword searching is fast in MongoDB. The data stored is in JSON format and will also help in integration with twitter api

c. Application Specific Questions

i. Which Twitter API do you use?

- We will be using twitter keyword targeting, POST statuses/filter | Docs | Twitter Developer Platform, GET users/lookup | Docs | Twitter Developer Platform, Filtering Tweets by location | Docs | Twitter Developer Platform, and multiple apis from twitter V2

ii. How would you build this so it's expandable beyond your local precinct?
    - This application will use the gps module from the application which is embedded in the Ionic framework. In addition to this the twitter api for location will also help add in more filters to this application
iii. What would you do to make sure that this system is constantly stable?
    - We will be using AWS which is a reliable, secure and stable environment.
    - A scheduled job for server health monitoring will be implemented
    - Load balancer will also be in place to manage the surge in the network
iv. What would be your web server technology?
    - We will be using AWS to host the application. We will be using elastic beanstalk to deploy the node js application which will have support for auto scaling in case of high traffic.
v. What databases would you use for triggers?
    - We will use AWS Lambda and AWS RDS for event triggers
vi. For the historical log of tweets?
    - We will use AWS cloudwatch logs to store and monitor historical log
vii. How would you handle the real time, streaming incident report?
    - For real time streaming incident we will use redis which is an in memory data structure and will help to provide us to share updates in a fast and efficient way
viii. How would you handle storing all the media that you have to store as well?
    - To handle and store all the large media file we will use aws s3 glacier technology because of its speed to air and ease to use

4. A Mildly Interesting Mobile Application

a. Summary of application development
   i. This application will a simple CRUD application but with efficient support of storage and geo locations
   ii. This application will be a simple NodeJS application with support of API for fast storing and retrieval of data(images)
   iii. This application will use firebase for authentication and realtime database to store users data

b. Process definition
   i. The user will first create an account or directly login through google account which will be managed by firebase authentication solution
   ii. Once the user is logged in and starts to upload mildly interesting things then we will use route 53 which is a dns web service for amazon.
   iii. We will use dynamo db with global tables to read and write data locally providing single-digit-millisecond latency. A
   iv. All the images will be stored in amazon s3 to high speed and efficient storage
   v. To provide users with fast and efficient search we will also use elastic search by aws to show geo based data
   vi.