

# CS-554 Lab 8

## Choosing Technical Stacks

### Scenario 1: Logging

Using log4js

“

```
var log4js = require('log4js');  
var logger = log4js.getLogger();  
logger.level = 'debug'; // default level is OFF - which means no logs at all.  
logger.debug("Some debug messages");
```

”

*How would you store your log entries?*

All the log can be store in file in app.js where you can find log file

*How would you allow users to submit log entries?*

log4js has 6 different log levels:

“

```
'trace': log4js.levels.TRACE,  
'debug': log4js.levels.DEBUG,  
'info': log4js.levels.INFO,  
'warn': log4js.levels.WARN,  
'error': log4js.levels.ERROR,  
'fatal': log4js.levels.FATAL
```

”

*How would you allow them to query log entries?*

Log entries are stored in different level, it's very easy to search them in log file by keywords.

*How would you allow them to see their log entries?*

Open the file log to see entries.

*What would be your web server?*

Web server would be either Apache or Nginx running express.

### Scenario 2: Expense Reports

*How would you store your expenses?*

I will use a database to store the expenses including the data fields: id, user, isReimbursed, reimbursedBy, submittedOn, paidOn, amount, time.

*What web server would you choose, and why?*

Nginx. Because Nginx has better support in highly concurrent working. Generating a PDF require high I/O, we may need high concurrency ability.

***How would you handle the emails?***

I'd like to create an email account first. Then use "Nodemailer" send email to different user. "Nodemailer" is a module for Node.js applications to allow easy as cake email sending.

***How would you handle the PDF generation?***

I'd like to use "wkhtmltopdf", because "wkhtmltopdf" can convert html to PDF.

***How are you going to handle all the templating for the web application?***

Use input the data and generate the template written in HTML, then using "wkhtmltopdf".

### **Scenario 3: A Twitter Streaming Safety Service**

***Which Twitter API do you use?***

I'd like to use the API called Tweets by Place:  
<http://twitter.com/search?q=place%3A07d9cd6afd884001>

***How would you build this so its expandable to beyond your local precinct?***

I plan to write code in different modules which is high cohesion and low coupling so that those code can be used in different area.

***What would you do to make sure that this system is constantly stable?***

There is no constantly stable system. We can reduce the risk of instability. For example, Making system consists of many projects to focus on specific topic.

***What databases would you use for triggers?***

I'd like use MongoDB with "mongo-triggers" for triggers

***For the historical log of tweets? How would you handle the real time, streaming incident report?***

I can get data from that API every minute to check if there are any new tweets based on tweets id. We then store new tweets into data base and handle new tweets.

***How would you handle storing all the media that you have to store as well?***

I can store them, and put their URL into database. I can get each media by search database.

***What web server technology would you use?***

I'd like use Apache

## Scenario 4: A Mildly Interesting Mobile Application

### *How would you handle the geospatial nature of your data?*

I can collect longitude and latitude of each image, and store all of them into elasticsearch. Then when a user request for a data, it search from elasticsearch in a range of longitude and latitude so that user can get images around them.

### *How would you store images, both for long term, cheap storage and for short term, fast retrieval?*

For long term and cheap storage, I can use different servers to store them.

For short term and fast retrieval, I can store data in a server that has, so it can retrieval data much faster than a normal machine.

### *What would you write your API in?*

login API

image API (include longitude and latitude)

interesting events API

get pictures API

interesting/not-interesting API

### *What would be your database?*

I'd like to use mongodb and elasticsearch.