



Bangalore, Madurai
15 June 2017

HONEYWELL INTERN HACKATHON

Hackathon Challenge Statements

Honeywell
THE POWER OF **CONNECTED**

Challenge - 1

Plot and Query

- Plot a list of locations you have travelled on a map service of your choice (Google Map etc.)
- Provide a few filters of your choice to view the plotted information

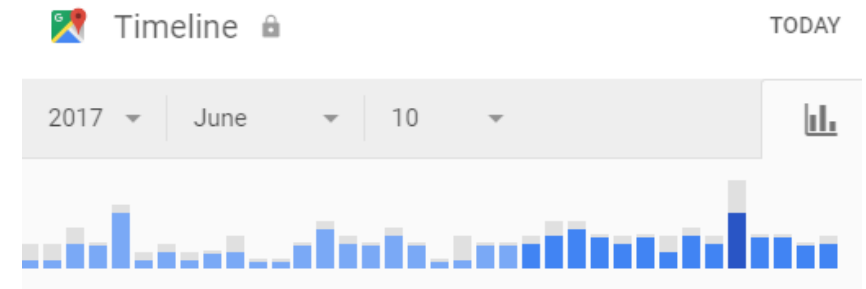
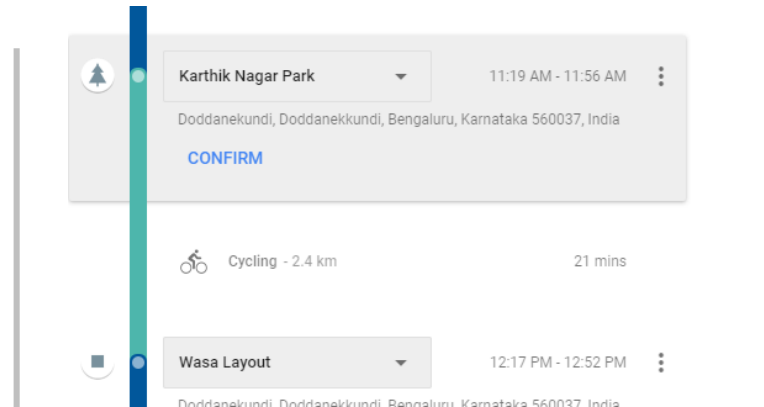
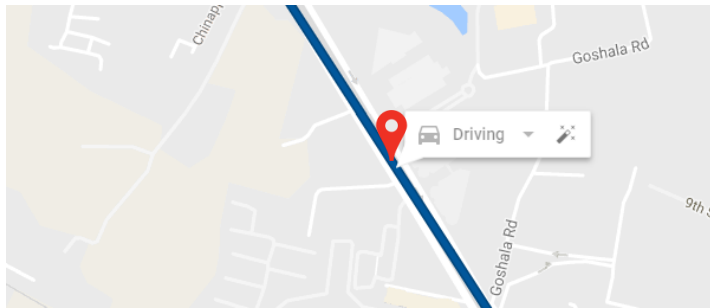
Requirements:

- As an end user, I should be able to
 - Upload my own Google Location Data (in JSON format)
 - Ex:** Sample location history by Google can be downloaded at https://takeout.google.com/settings/takeout/custom/location_history?hl=en&gl&expflags
 - View map markers (📍) at locations from the uploaded file on a map
 - Click on any map marker to view more details of my activity at that location
 - Enter a time duration (from – to) and be able to view relevant map markers

Dos and Don't s:

- Do pick a language of your choice.
- Pick an interface of your choice – web, desktop, mobile
- Feel free to assume any other detail if needed be
- Be ready to discuss design patterns, classes, interfaces, multi threading etc.
- Do not restrict yourself with the given requirements.
- Save your code on GitHub
- Be awesome !!

Inspiration:



Challenge - 2

Get me a Seat

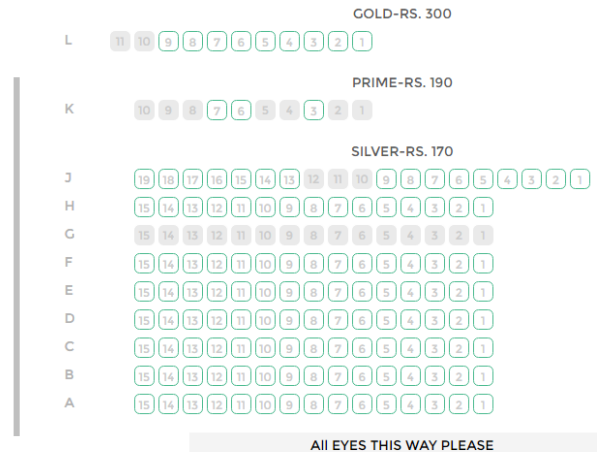
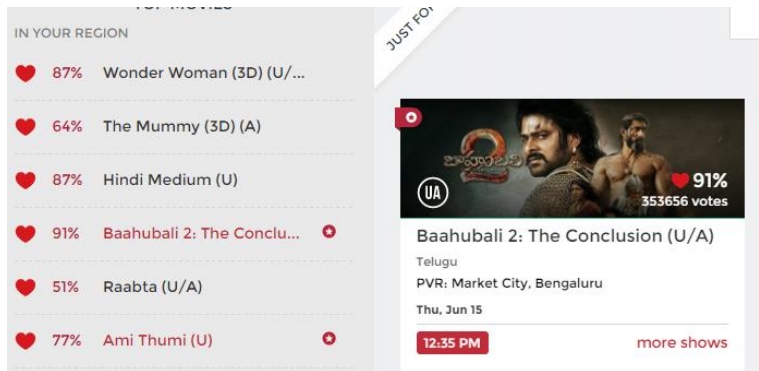
- Display a list of your favorite movies and their show times in a city of your choice
- Surge pricing for a movie ticket booking . Prices can go steeply high or steeply low.

Requirements:

- As an end user, I should be able to
 - View movies and their show times
 - Select seats by seeing the **price** at the time of booking
 - Receive notifications if the price for the user's favorite seat decreases
- System should allow concurrent users trying to book the same seat at the same time.

Ex: As the demand for such particular seat is more in this case, the price of the seat should immediately go up !

Inspiration:



Dos and Don't s:

- Do pick a language of your choice.
- Pick an interface of your choice – web, desktop, mobile
- Feel free to assume any other detail if needed be
- Be ready to discuss design patterns, classes, interfaces, multi threading etc.
- Do not restrict yourself with the given requirements.
- Save your code on GitHub
- Be awesome !!

Challenge - 3

Near Real Time Stock Ticker

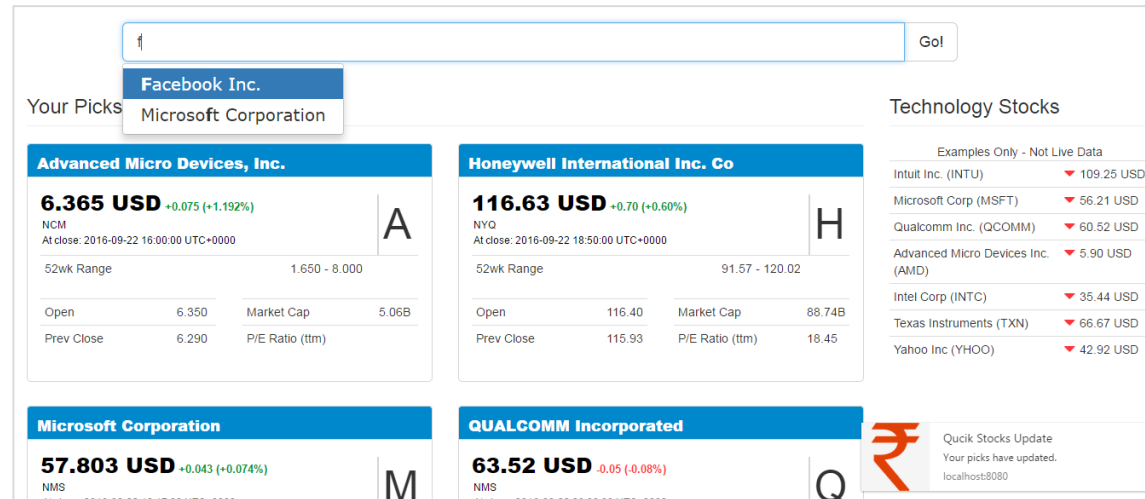
- Display a list of your favorite companies and their stock listing
- Track current price, % change from last day, 52 week high and 52 week low

Requirements:

- As an end user, I should be able to
 - Search and add view as per need, remove if not needed
 - View +ve change as **green** and -ve change as **red**
 - Updates should be asynchronous
 - Visual update should not take too much of time

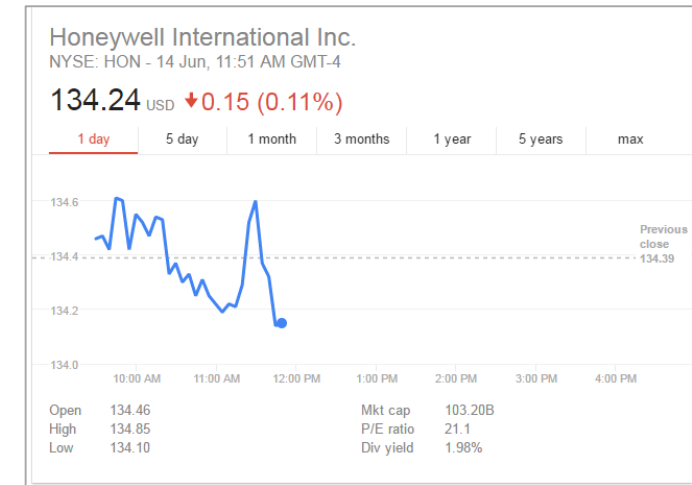
Ex: As soon as the stock changes, user should be immediately notified

Inspiration:



Dos and Don't s:

- Do pick a language of your choice.
- Pick an interface of your choice – web, desktop, mobile
- Feel free to assume any other detail if needed be
- Be ready to discuss design patterns, classes, interfaces, multi threading etc.
- Do not restrict yourself with the given requirements.
- Save your code on GitHub
- Be awesome !!



Challenge - 3

Near Real Time Stock Ticker

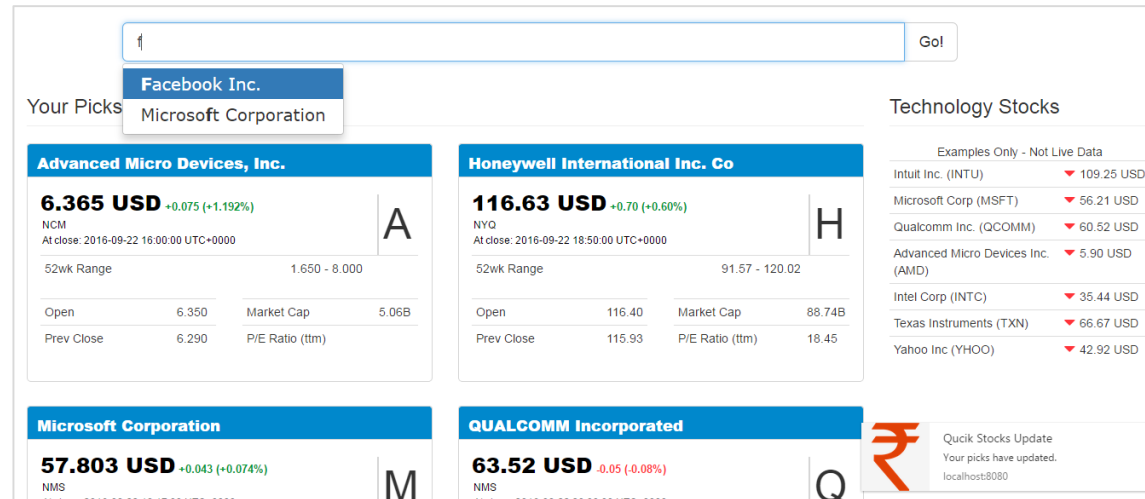
- Display a list of your favorite companies and their stock listing
- Track current price, % change from last day, 52 week high and 52 week low

Requirements:

- As an end user, I should be able to
 - Search and add view as per need, remove if not needed
 - View +ve change as **green** and -ve change as **red**
 - Updates should be asynchronous
 - Visual update should not take too much of time

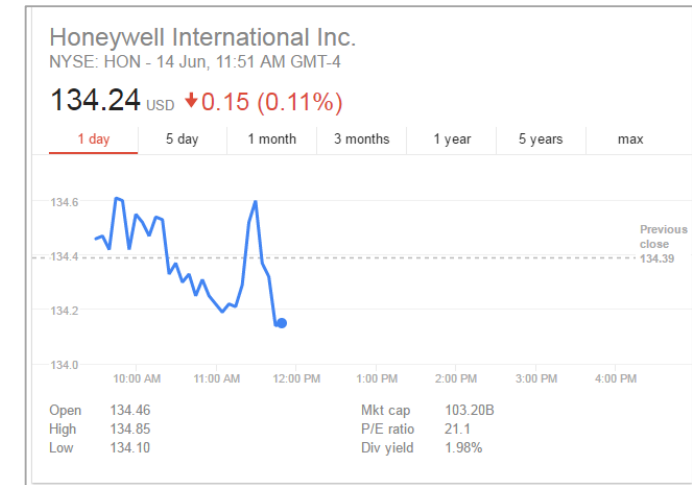
Ex: As soon as the stock changes, user should be immediately notified

Inspiration:



Dos and Don't s:

- Do pick a language of your choice.
- Pick an interface of your choice – web, desktop, mobile
- Feel free to assume any other detail if needed be
- Be ready to discuss design patterns, classes, interfaces, multi threading etc.
- Do not restrict yourself with the given requirements.
- Save your code on GitHub
- Be awesome !!



Challenge - 4

Simulate a Digital Signal Tx-Rx Chain

- Design the Tx-Rx chain to overcome errors induced by noise and errors by a channel.
- This would require error correction capabilities.

Requirements:

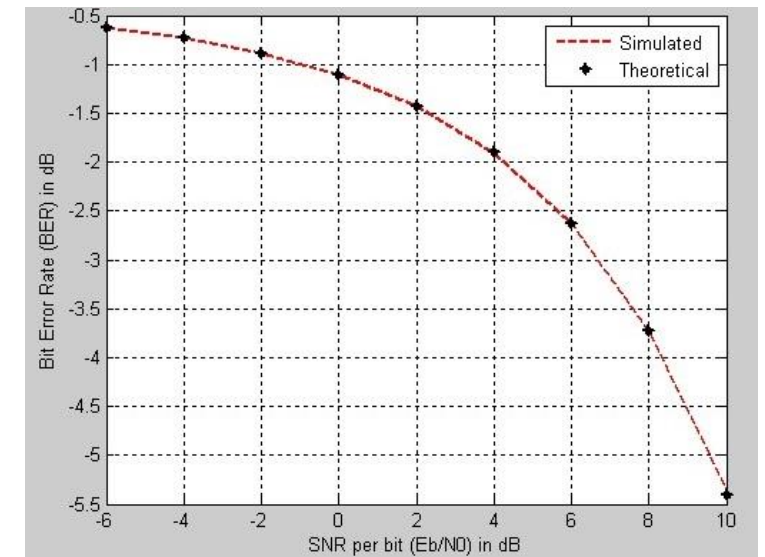
- Solve issues like energy concentration and burst errors by using **Energy Dispersal** and **Interleaver**
- Enhance data reliability by controlling the errors - introduce redundancy, a technique called **Forward Error Correction**
- The receiver needs to have the de-energy disperser, de-interleaver and decoder to get back the information.
- Incorporate the techniques mentioned above to simulate a robust Digital Signal Tx-Rx chain.
- Add noise to the signal after the Tx chain before sending it to the Rx chain.
- Check the Bit Error Rate of the received signal for different noise levels.

Inspiration:

You can generate random bits and use it as the input signal or discretize an audio signal.

Sources for audio data can be found online.

- <http://techslides.com/sample-files-for-development>
- <http://www.sample-videos.com/download-sample-audio.php>



Challenge - 5

Truly-Meetable

- Bring people together @ tech events, tech meet ups !
- Allow *tweetizens* to share their interests, activities etc.

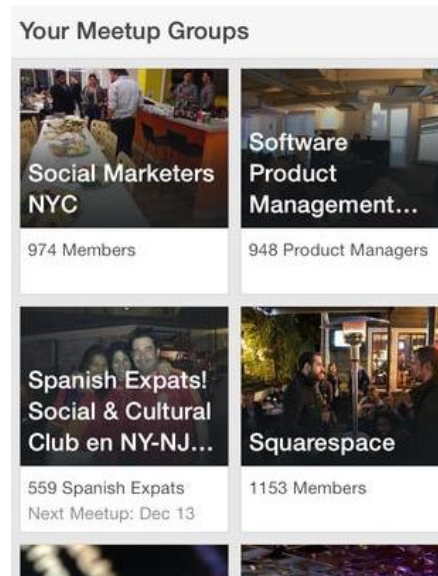
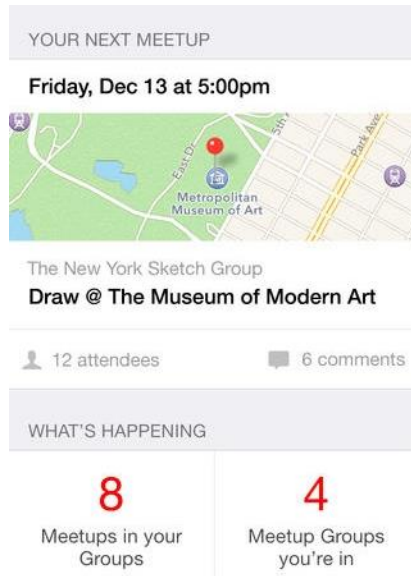
Requirements:

- As an end user, I should be able to
 - Find people around with “similar” interests
 - Share local activities and “interests” with each other
 - Share “deals” with each other (minimum 5 users in parallel)
 - Update should not take too much of time

Dos and Don't s:

- Do pick a language of your choice.
- Pick an interface of your choice – web, desktop, mobile
- Feel free to assume any other detail if needed be
- Be ready to discuss design patterns, classes, interfaces, multi threading etc.
- Do not restrict yourself with the given requirements.
- Save your code on GitHub
- Be awesome !!

Inspiration:



Challenge - 6

A lot can happen over fitness !

- Its all about fitness and fitness gatherings
- Leaderboards for fitness

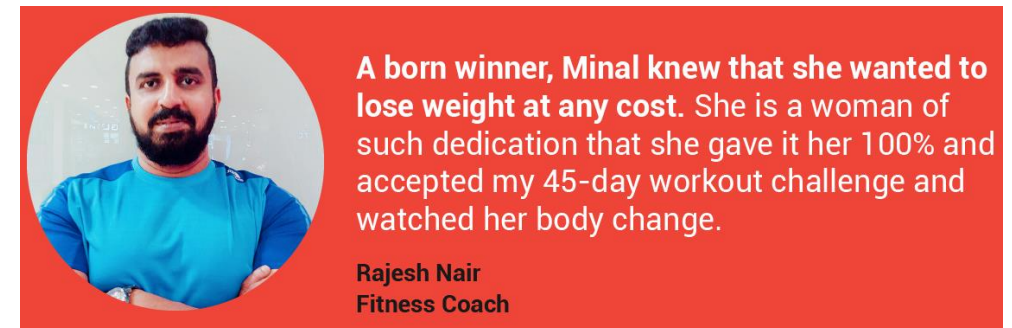
Requirements:

- As an end user, I should be able to
 - Learn and share my fitness goals with “people”
 - View and share fitness and athletic events happening around “my location”
 - Find fitness coaches around
 - Treat fitness as “fun”

Dos and Don't s:

- Do pick a language of your choice.
- Pick an interface of your choice – web, desktop, mobile
- Feel free to assume any other detail if needed be
- Be ready to discuss design patterns, classes, interfaces, multi threading etc.
- Do not restrict yourself with the given requirements.
- Save your code on GitHub
- Be awesome !!

Inspiration:



Honeywell is building a smarter, safer,
and more sustainable world

THAT'S THE POWER OF **CONNECTED**
THAT'S THE POWER OF **HONEYWELL**

Connected Aircraft • Connected Automobile • Connected Home • Connected Building
Connected Plant • Connected Supply Chain • Connected Worker

Honeywell

THE POWER OF **CONNECTED**