OULUN YLIOPISTO
UNIVERSITY of OULU

# PERSONALISED DISPLAYS

## Stage 3: Evaluation


AUTHORS

**Juan Camilo García**, 2418810
jgarciah@student.oulu.fi

**Haejong Dong**, 2292191
s2haejong@gmail.com

**Henri Koski**, 2190426
henri.koski@student.oulu.fi

**Yifei Zuo**, 2443937
yifei.zuo@student.oulu.fi



University of Oulu
Applied Computing Project II
18-04-2016

# Table of Contents

# Introduction

## Purpose of the project

Public displays are starting to become more popular every day, but they still have a low interaction rate with the potential users. Most of its use is based on advertisement, which, although it's a revenue for the owner of the display, doesn't provide all that it could to its potential consumers. This types of public spaces are very powerful since they give interactivity and dynamic functionalities to people around it, but the downside is either that the information shown is very generic so that it's useful for everyone that approaches, or it displays too much data so that anyone can find the information they want. This situation limits the richness of the interaction as well as the possible uses of a public display.

Our purpose with this project is to provide more relevant information in public displays through automatic personalization of the content is shows by leveraging the fact that most users carry a smartphone with internet connectivity at all times. This will be achieved by mimicking a bulletin board and transforming it into its digital version which is not only smart but interactive and dynamic. By being digital, more organized layouts could exist compared to the current ones seen on Figure 1 and 2 and much more data could be displayed at the appropriate time while consuming less physical space. The organization is key, since bulletin boards have a big flaw which is that finding content on them is hard. Updates are not as frequent and certainly less visible when they happen, since changing one of the posters doesn't immediately trigger a response from the user.


*Figure 1. Bulletin board on the Infinite Corridor at MIT (2004) [31].*

*Figure 2. A bulletin board from a Starbucks [32].*

With this, we would like to find if users are more likely to interact in public spaces if the content is tailored for them when they approach as well as the pros or cons the digital counterpart of a bulletin has versus it. This also gives us the opportunity to explore personalization for groups instead of individuals, as well as study how much information are consumers willing to give in exchange of more tailored information.

## Evaluation Strategy

## Assumptions and Constraints

- Measurements of signal strength from one side of the display are equal to the measurements on the other side of the display.
- The measurements (RSSI, time to personalise) done in the lab are a representation of in the wild performance.

# Glossary

This section provides a brief explanation of the terms relating to the system design document and the project in general. The terms are alphabetically ordered.

Hypertext Markup Language (HTML)

Is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page [1].

Cascading Style Sheets (CSS)

Is a style sheets that describe how HTML elements are to be displayed on screen, paper, or in other media [2].

JavaScript

Is a programming language commonly used in web development [3].

Go Programming Language (golang)

Is an open source programming language that developed with the purpose of easy, reliable, and simple software building [4].

Application Program Interface (API)

Is a set of routines, protocols, and tools for building software applications [5].

Wi-Fi

Is a wireless networking technology that allows computers and other devices to communicate over a wireless signal [6].

Airmong-ng

Is a script that can be used to enable monitor mode on wireless interfaces [7].

Go-socket.io

Is an implementation of socket.io in golang, which is a real-time application Framework.

It is compatible with latest implementation of socket.io in node.js, and supports room and namespace [8].

PostgreSQL

Is an open source relational database management system ( DBMS ) developed by a worldwide team of volunteers. PostgreSQL is not controlled by any corporation or other private entity and the source code is available free of charge [9].

Transport Layer Security and Secure Sockets Layer (TLS/SSL)

Both of which are frequently referred to as 'SSL', are cryptographic protocols designed to provide communications security over a computer network [10].

## Evaluation process

As shown in figure 3, we made an evaluation plan first. Then we made a system evaluation in which we tested all the features of the system and found its limits. Afterwards we improved the system using the results of the system evaluation. With it we proceeded with a user evaluation, then analysed the results and finally provided some suggestions of improvement to the system.

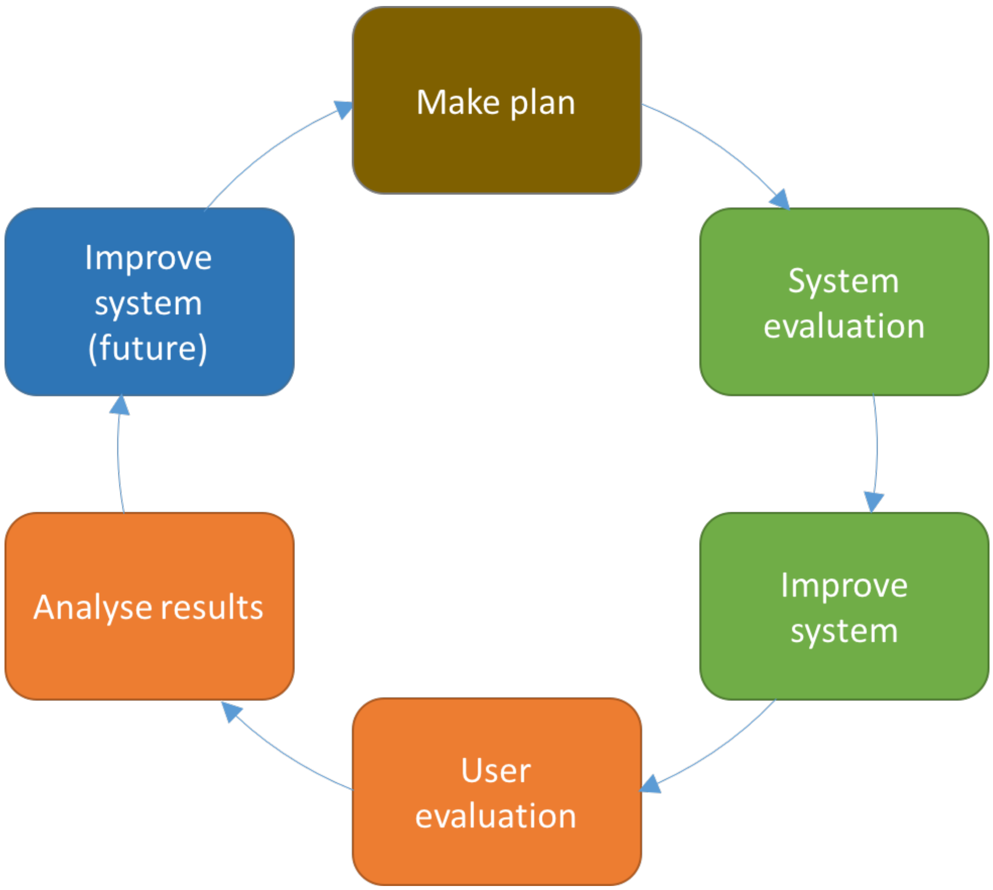


*Figure 3. Evaluation process diagram*

For the system evaluation we used predictive evaluation to identify how a typical user would act and predict usability problems. We wanted to make sure that the system worked in a real life environment as well as determine its weaknesses to address them. This was done by testing the features of the system as well as its limits obtaining its performance. Then for the user evaluation we made a mix between a "Quick and dirty" evaluation and field studies. We used qualitative techniques such as interviews and questionnaires as well as task-based usability evaluation while implementing the think-aloud protocol.

## Setup

### Test types and objectives

We did two evaluations, one for the system's performance and one for the system's usability.

## System evaluation

Using predictive evaluation, we identified the common uses of a typical user and hence identified the possible weaknesses of our system. The first possible weakness was if the user would be recognized when he walks by the public display. Secondly we would test if the public display was capable of identifying which user was the closest to the display if more than one user was present.

### *Test Procedure*

### User Detection

To test if the user would be recognized when he walked past the public display we decided to make a two-fold test. First we tested the speed of personalisation and then we measured the maximum distance of personalization. To test the speed of personalisation we placed a mobile device registered with our system at a two meters' distance from the public display and measured the time span between turning the Wi-Fi sensor on and seeing the content personalised on the screen. For this to work we first registered the device and subscribed it to several topics so that we would acknowledge when the content was personalised.

This was done at a two meters' distance arbitrarily since the Wi-Fi speeds are extremely faster than our system's detection, so the difference of speed between close distances and long distances is negligible. To have an accurate time we tested this process 20 times.

For the maximum distance we turned of Wi-Fi on the mobile device at a 10 meters' distance and waited to see if our sensing device detected it. If it did, we turned Wi-Fi off, waited for our device to stop sensing it, moved 10 meters away and turned Wi-Fi back on. We iterated this procedure until either the device wasn't sensed anymore. Figure 4 shows the setup for this experiment.

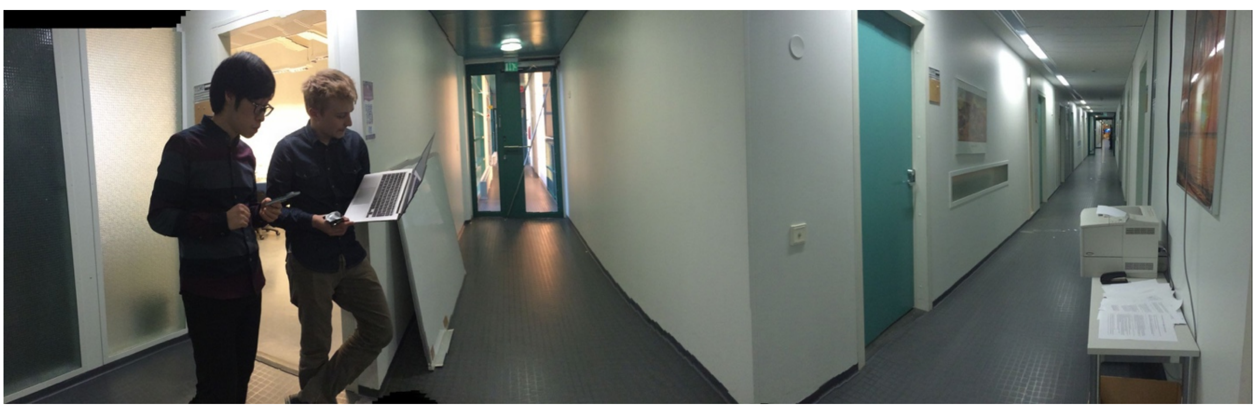

*Figure 4. Maximum distance setup*

### User Proximity Detection

To test if we could identify which was the closest user we decided to make a heat map of signal strength around the public display since this is the measurement our system uses to determine the closet user. To do this, we took measurements at a 1 meter' interval distance starting at 1 meter in front of the display. We turned on the Wi-Fi on the mobile device used and leave it still for 1 minute, then moved it 1 meter and waited again for 1 minute until we went to 8 meters. Then we did it again to one of the sides of the public display, assuming both sides would have the same results. Figure 5 shows the setup for front and side.

*Figure 5. Signal strength experiment setup*

### *Environment*

The test was carried out on a lab room. The sensing device was an Andriod device with Aircrack-ng, a network software to sniff Wi-Fi packets. Since this is not supported natively by Android, a custom kernel developed by Ruby Feinstein and Omri Ildis [11] [12] was used. This kernel only works on WiFi Chipsets BCM4329 and BCM4330. Because of this, a Nexus 7 (2012) which has Chipset BCM4330B2 was used. The OS on it was Cyanogen 10.1 and it was rooted to be able to access the monitor mode features. This sensing device was placed on the bottom of the display near ground level as shown on the following picture. (INSERT PICTURE)

The public display was connected to a Mac Mini and was using Chrome version 49.0.2623.87 (64-bit). It was connected to the internet through the wireless network PanOulu.

The mobile device used for all the testing was a Samsung Galaxy S4 mini.

### *Expected outcome*

#### User Detection

We expected to have an average below 20 seconds to personalise the display from the moment the mobile device was turned on since we were scanning every 10 seconds and the communication going from the sensing device, to the server and back to the public display should take more than 10 extra seconds. We assumed it would take longer than 10 seconds since the mobile phone takes some time to start sending broadcasts with its mac address.

For the maximum distance we were expecting the values that an average router advertises as the maximum coverage which is around 46 meters indoors.

### User Proximity Detection

In this experiment we were expecting to have many areas in which we had false positives because of our experience with the system from before hand, but we had to test it appropriately to make sure.

## User evaluation

For the user evaluation we took features of field studies as well as task-based evaluation. During the evaluation, the think-aloud protocol was used.

### *Test Procedure and Environment*

The evaluation process was carried out with one participant at a time. During the evaluation process, we provided as few instructions on how to interact with the system as possible until the very last stage or until the user was badly lost. This was done so that we could associate this result with people's behaviour in the wild environment. Our participants were at first shown to the public display and allowed to explore as they wish. In this way we would expect to observe how users may figure out the usage of the system as well as turning it into a metric that measures usability of user interface in general. We also asked the users to find some information on the public display so that they would see what the contents were. Then we moved on to the mobile application and once again let participants play around with it. Then we brought them back to the display however this time, the difference was having two systems together to interact. After some time passed we gave them a couple of tasks to perform. At around this point participants would have had noticed how the system works, in general. At the end of the evaluation, we did some Q&A, provided the exact idea of the system and how it works, and the entire process was ended by feedback and a second Q&A.

### *Participants*

| | Age group | Gender | Profession/occupational Background | Past usage in public display | Activeness in joining public events |
|---|---|---|---|---|---|
| Subject A | 21~25 | Male | Student / Computer Science | Checking traffic information | Almost none (2~3 times per year) |
| Subject B | 21~25 | Male | Student / Computer Science | Entertainment application (selfie app) | Almost none (once in many months) |
| Subject C | 21~25 | Male | Student / Computer Science | Checking nearby services | Almost none (2~3 times per year) |

*Table 1. Participants basic information*

During the evaluation phase we collected 3 participants in total. Table 1 lists the basic information of them which could help us build assumptions on the behaviours of the participants while interacting with the system. First thing to take a look at is commonality. We have three subjects with similar background, this can be noted as one of the limitations of the result of this evaluation. However more likely, these kind of evaluation settings may provide better qualitative feedback in technical related matters.

# Data and Results

## System Evaluation

Regarding the user detection, we collected 20 samples which in average resulted in a 28.6 seconds time to detect from the moment the Wi-Fi sensor of the mobile device is turned on, and a standard deviation of 5.616. The confidence interval of 95% is (25.6,31.3), which means it takes in maximum 31.3 seconds to detect the user once his Wi-Fi is turned on. The maximum distance was much more than expected, the maximum distance at which we could detect the device was 90 meters. We couldn't test in a longer distance since this was the longest distance indoors we had.

With this results we can assure that any user will see personalized content when he reaches the display, since assuming a normal walk speed of 5 kilometres per hour, traveling 80 meters would take almost 2 minutes (111 seconds). Even at twice the speed it would be the equivalent of 1 minute to reach a radius of 10 meters around the display, and it will be personalised within 30 seconds of detection of the device which happens from more than 90 meters away.

For the user proximity detection, we got 5 or more results for each scenario and first made T-tests to compare the means between the results of the measurements done in front of the screen and the side of the screen. The p-value was 0.3902, which means the means are not different and our initial assumption was true in a global scale. We then tested within groups by splitting the data by the distance from the sensing device. The results obtained are in table 2.

| Meters | P-value |
|---|---|
| 1 | 0.1607 |
| 2 | 0.1632 |
| 3 | 0.08525 |
| 4 | 0.00001982 |
| 5 | 0.04025 |
| 6 | 0.5911 |
| 7 | 0.0009098 |
| 8 | 0.07854 |

*Table 2. Statistic results of t-test tests between side and front*

To view this data properly, we merged the values from the meter's distance that weren't significantly different, as these regions behaved similarly, and then created a heat map using green as 0 RSSI and red -100 RSSI visible in figure 6. FIgure 7 contains the same heat map in a more interpretable scale. The numeric results are also available in table 3.

<table>
<tr><th>Meters</th><th>Front</th><th>Side</th></tr>
<tr><td>1</td><td colspan="2">-35.5</td></tr>
<tr><td>2</td><td colspan="2">-38.7</td></tr>
</table>

| | | |
|---|---|---|
| 3 | -41.8 | |
| 4 | -40.3 | -47.7 |
| 5 | -40.2 | -44.8 |
| 6 | -50.2 | |
| 7 | -50.2 | -55.3 |
| 8 | -55.3 | |

*Table 3. RSSI of different places around the public display*

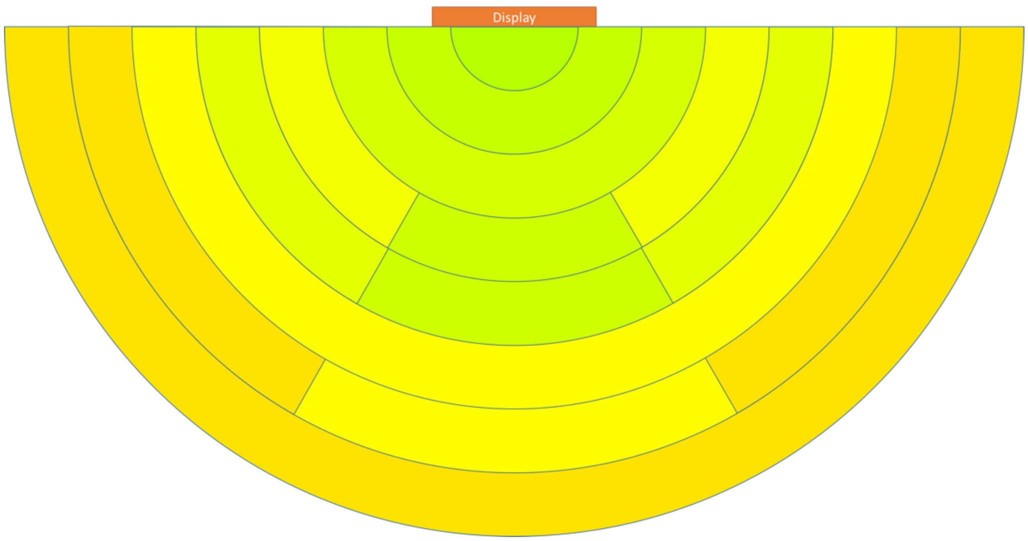


*Figure 6. Heat map of signal strength (each radius is one more meter)*

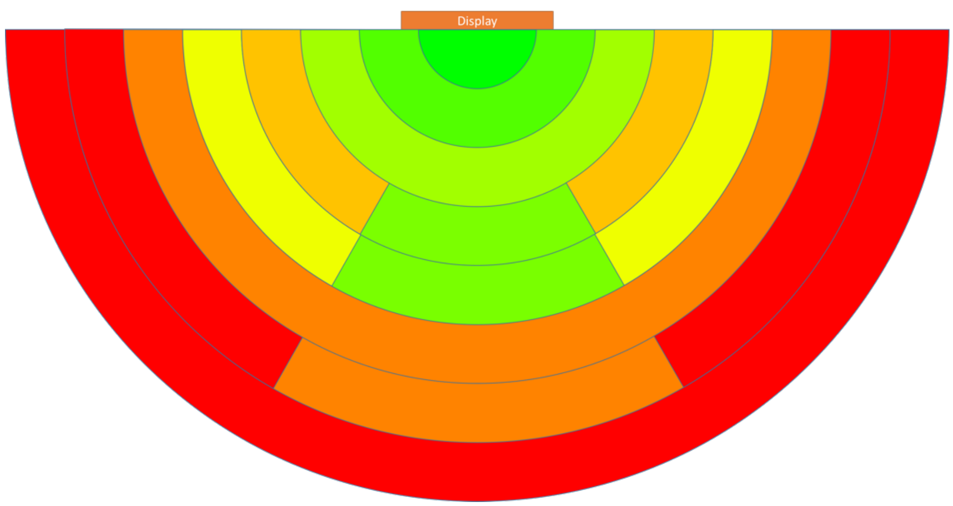


*Figure 7. Heat map with more clear colors (refer to figure 4)*

After this we decided to create boxplots which might better explain the overlaps between regions (Figure 8). We merged the regions that weren't equal according to the t-test because we believe this discrepancy is not due to a real difference but to a lack of data points.

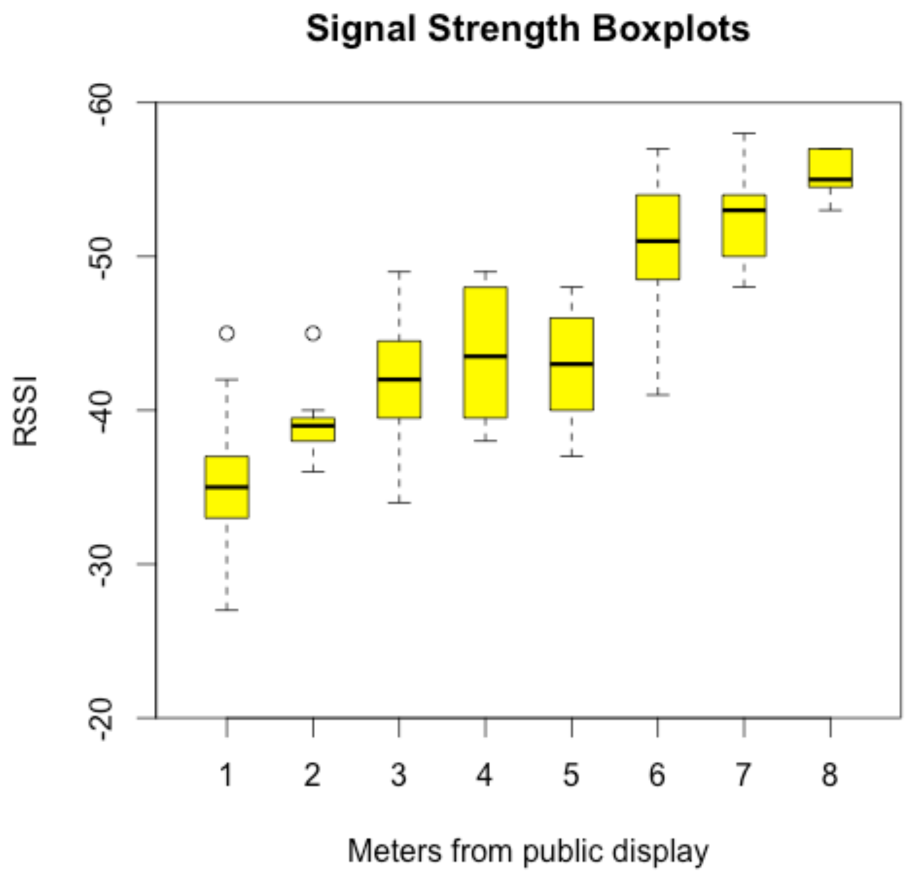


*Figure 8. Boxplots of signal strength around the public display*

As expected, some regions are hard to determine. The real problem resides in the region within 3 meters of the public display, which after observation, can contain several users, counting ones that aren't interacting with it. The detection of the closest user would have false positives specially within the close range region which is where the users would be interacting with the display.

## User Evaluation

Although we have relatively small set of data to evaluate and it is definitely not enough to generalize most of people's behaviour towards human computer interaction, we have received decent feedback in the aspect of UI design. All of participants didn't spend much time learning how both the mobile and web application systems work. One issue that was observed on the mobile application was that some of the participants complained about not being able to see all the available categories that were pre-registered for the evaluation purpose at a glance. Participants are only able to search by trying many words they are interested in, and in some cases, they typed synonym words to the registered ones, and hence they weren't displayed on the device. Similarly, some of the participants mentioned that the titles on the web application, which represent each column of the content array, appear to be confusing as they are predefined words and are not instructive enough to describe somewhat arbitrary contents which are coupled together by few common aspects. They believed the categories were too broad.

On the other side we had a severe issue while using the system. Users were not able to interact with the display by a single tap. This was assumed to be a hardware issue though, if it was a wild testing the whole system would have become useless and likely made people annoyed and never to use it again.

Another big problem we had during the experiment was that the backend server was not sending the selected contents to the mobile client. We could only make approximately one in five successful data transmission to the client which is very disappointing for such kind of system, in which this is a key feature. However, we conclude that with current technology, limited time and given commodities high succeed rate would not be one of our major expectations to achieve (as seen in the system evaluation). The issue was driven by inaccurate detection of the proximity of the mobile devices.

One of the issues mentioned every time by participants was the contents updating rate, from when they subscribe to a topic till the topic is on the public display. We believe this was merely caused by the way that we performed the evaluation. Based on the result from the system performance test the system worked fine, but since in the laboratory testing the participants had to fulfil the experiment in a relatively short period of time, the system appeared to become as a search engine where people search questions and browse information conveniently just as what the participants had to do, add categories to the interest and immediately come to the display, wait until it changes, and read the content. The ideal way of fulfilling this experiment would have been deploying the system in the wild and assess how the users interact with the system and if the personalisation engages users. Instead we did it in this way because the system was not appropriately stable to be set in such an environment.

One last thing mentioned in respect to the UI design of the web application was discovered by one participant who subscribed to multiple topics at the same time, was that we can only keep two events per category at the same time on the screen, and hence he couldn't see some of the topics he had subscribed to. Since the web UI provides only two row to be displayed in the current prototype the participant had no way to search contents that he is lately interested in. This would be solved if the perspective of the system changed into more of a bulletin board instead of a search engine, as the users wouldn't be expecting to always see content that relates to all of the topics subscribed. And when more data is available, the content would rotate so always fresh content is appearing on the display.

# Analysis

## Overview

From our system's performance point of view, detecting users is not a problem. While setting up the experiment for the user evaluation, we found out that the sensing device stopped seeing the device after some minutes of the Wi-Fi signal being turned on. For this we modified the mobile application, and now it forces Wi-Fi scans every minute when a public display is detecting the device. In this sense, once the device is detected, it will only be not detected when it's far away. This made the system more robust. Since we found out we could detect the devices from a very long distance, we decided not to personalise content as soon as the device is visible but only when the signal strength is strong enough, which after seeing the heat map data was decided to be any value greater than -50.

Although our detection was impeccable, the proximity detection to determine the closest user wasn't that accurate. For this we propose two solutions. The first one is to display a list of users around the public display on the screen, and wait for the user to select himself. The second solution is to use an NFC reader next to the public display and remove the send to phone button. In this way the visible data on the public display would be sent when the user approaches his device to the NFC tag. This NFC tag would also give a good way of attracting users to download the application of the system, since putting a device without our application would open the application store on the smartphone on the page of our application.

In general misunderstanding of the system pervaded, most of the participants used it as if it was a search engine. To prevent this from happening we would need to take enough time deploying the system in the wild and evaluate based on what users do with the system in a prolonged time span

One participant claimed when asked to type interesting keywords for him that most likely he wouldn't know what information he needs. Thus suggesting that it would be great if the system could satisfy this need by his search engine queries or other data usage of his phone. This however has still remained unsolved as it definitely requires a perfectly performing system which would then be deployed in the wild and studied with time as well as other type of algorithms that use the data from the user's smartphones.

We have debated during the system design phase about the matter of security concerns for this project, however one participant claimed that he wants to be recognized by the system and be notified that the system is capable of providing information for him. This was a discrete case in the light of security. In the same vein he also wished to highlight contents that are particularly personalized to him so that he can recognize them.

Lastly he suggested that the default contents that are displayed on the public display are distracting, but again this is because the users are seeing the system as a search engine and not as an opportunistic kind of system.

One interesting common suggestions amongst all the participants was about adding feature for bus time table. This might have to do with the context of the city where the participants are presently residing.

## Scalability and Fault Tolerance

One hypothetical problem that was considered, was the lack of computing power in our backend server. The server is a virtual machine from upcloud which has one 3GHz cpu core and 1Gb ram. To test the capacity of our server hardware we ran http load tests with Vegeta. As the Vegeta's README file states "Vegeta is a versatile HTTP load testing tool built out of a need to drill HTTP services with a constant request rate."

In our test we decided to call [https://acp.velho.xyz/categories](https://acp.velho.xyz/categories) route. The reason why we chose this route is that first it requires http request parsing, making database query, parsing database rowset to Go struct and marshalling that to json response body. The test was run with the following command: "./vegeta attack -targets route -duration 60s | ./vegeta report -output results.txt". The first part launches http attack that lasts 60 seconds to url that is defined in file called "route". The second part takes the binary output from attack command and decodes it to results.txt file. During the test cpu usage was between 3 and 7 percentage and memory usage stayed at 330Mb. The results were the following:

| | | |
|---|---|---|
| Requests | [total, rate] | 3000, 50.02 |
| Duration | [total, attack, wait] | 59.980701739s, 59.979999861s, 701.878µs |
| Latencies | [mean, 50, 95, 99, max] | 889.397µs, 708.014µs, 1.133466ms, 4.140317ms, 37.697949ms |
| Bytes In | [total, mean] | 2322000, 774.00 |
| Bytes Out | [total, mean] | 0, 0.00 |
| Success | [ratio] | 100.00% |
| Status Codes | [code:count] | 200:3000 |

So with the default 50 request per second rate our server had no problems with performance and the mean average latency was under one ms in localhost. Doubling the rps to 100 did not affect the mean average latency so we proceeded to increase the rcp value to find the limit of our server. With request rate of 200 we slightly higher latency of 957.797µs. The rate of 500 caused increase in response latency of 77.89536ms but still all requests were handled without errors. With request rate of 1000 per second we were able to cause 3 failed request in 10000 request and the mean average latency was 399.608546ms. With the rate of 5000 we were able to increase latency to 1.449949228s with the same 3 failed request.

With these results it is fair to say that even though we are running single core computer the performance won't be a problem. If we would get more users it would be easy to scale things up by adding more cores to the system because golang handles each request in own goroutine and scales to all available cores.

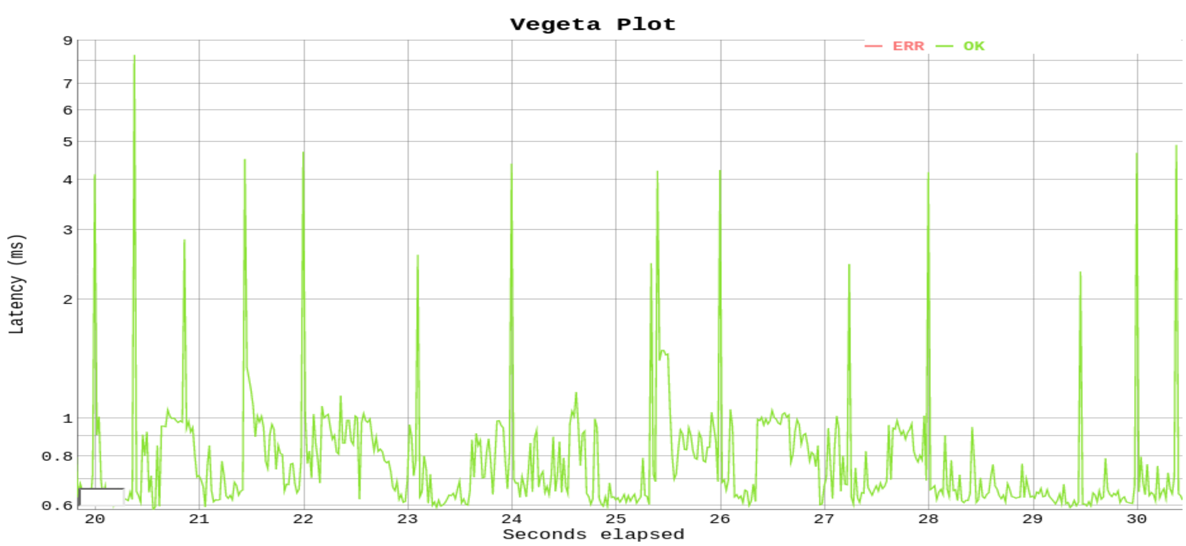

*Figure 9. Request latency in the first test with rcp of 50*

### Security

With regards to security, the connections between the admin panel, the display and the sensing devices to the server need an access token identifying the display. The connection between the mobile client and the server need the client's unique id, except when creating a new user. At the moment, the user's id is an auto incrementing id, but it will be changed to UUID v5 (standard RFC4122) [13].

All the connections are encrypted, which means the tokens can't be captured from the network. It also means, the private data sent from the mobile device to the server (personal events) is secure.

## Discussion

After all the analysis, we found the system is viable, but the design still needs some tweaking to fulfil the users needs. Some of the users expressed that if they need information they would actively search for it and if they don't need information then they wouldn't go through the hassle of thinking what would interest them and subscribing to these topics (as our system works). In this sense it may be hard to sell the idea, but if deployed in the wild and enough data is collected, maybe the personalisation can be done behind the scenes without the user's interaction (subscribing to topics). These users were also asked if they used bulletin boards, and many didn't, which may be because most of the content is in a different language. This is important to us since our system is trying to improve the space usage of the bulletin boards, as well as improve the user experience.

The next step for the system would be to improve the dynamics of sending information to the mobile device from the public display, as well as redesigning the UI to better communicate what the system does, and in this way deploy it in the wild and see how the system performs by comparing number of users approaching when the content is personalised and when it's not.

## References


[1] R. Margaret, "HTML (Hypertext Markup Language) definition," September 2005. [Online]. Available: http://searchsoa.techtarget.com/definition/HTML.

[2] w3schools, "CSS Introduction," [Online]. Available: http://www.w3schools.com/css/css_intro.asp.

[3] P. Christensson, "JavaScript Definition," 8 August 2014. [Online]. Available: http://techterms.com.

[4] golang, "The Go Programming Language," [Online]. Available: https://golang.org/.

[5] B. Vangie, "API - application program interface," [Online]. Available: http://www.webopedia.com/TERM/A/API.html.

[6] P. Christensson, "Wi-Fi Definition.," March 2014. [Online]. Available: http://techterms.com.

[7] Aircrack-ng, "Airmon-ng," [Online]. Available: http://www.aircrack-ng.org/doku.php?id=airmon-ng.

[8] GoDoc, “Package socketio,” [Online]. Available: https://godoc.org/github.com/googollee/go-socket.io.

[9] R. Margaret, “PostgreSQL,” May 2006. [Online]. Available: http://whatis.techtarget.com/definition/PostgreSQL.

[10] E. R. T. Dierks, "The Transport Layer Security (TLS) Protocol, Version 1.2", 2008.

[11] T. Bret, “Parallel Development Strategies for Software Congifuration Management,” Congluence Systems Ltd, 2004. [Online]. Available: http://www.methodsandtools.com/archive/archive.php?id=12.

[12] R. Feinstein and O. Ildis, “Wardriving from your pocket: Using Wireshark to Reverse Engineer Broadcom WiFi chipsets,” in *RECON*, Montreal, 2013.

[13] R. Feinstein, O. Ildis and Y. Ofir, “Monitor mode for Broadcom WiFi Chipsets,” 14 7 2013. [Online]. Available: http://bcmon.blogspot.fi/2013/07/monitor-mode-reloaded_14.html.

[14] Google, “Google Cloud Messaging,” [Online]. Available: https://developers.google.com/cloud-messaging/.

[15] Google, “Android OS Distribution,” [Online]. Available: http://developer.android.com/about/dashboards/index.html#Platform. [Accessed 2 2016].

[16] UpCloud, “UpCloud,” [Online]. Available: https://www.upcloud.com/.

[17] Let's Encrypt, “Let's Encrypt,” [Online]. Available: https://letsencrypt.org/.

[18] D. Huie, “gomigrate,” [Online]. Available: https://github.com/DavidHuie/gomigrate.

[19] jQuery, “jQuery,” [Online]. Available: https://jquery.com/.

[20] BULMA, “Bulma,” [Online]. Available: http://bulma.io/.

[21] julienschmidt, “httprouter,” [Online]. Available: https://github.com/julienschmidt/httprouter.

[22] “pq,” [Online]. Available: https://github.com/lib/pq.

[23] “gopkg.in/gorp.v1,” [Online]. Available: http://gopkg.in/gorp.v1.

[24] “gopkg.in/guregu/null.v3,” [Online]. Available: http://gopkg.in/guregu/null.v3.

[25] heppu, “facebook-go-sdk,” [Online]. Available: https://github.com/heppu/facebook-go-sdk.

[26] heppu, “simple-cors,” [Online]. Available: https://github.com/heppu/simple-cors.

[27] OkHttp, “OkHttp,” [Online]. Available: http://square.github.io/okhttp/.

[28] Google, “gson,” [Online]. Available: https://github.com/google/gson.

[29] GitHub, “GitHub,” [Online]. Available: https://github.com/about.

[30] “A universally Unique IDentifier (UUID) URN Namespace,” [Online]. Available: https://tools.ietf.org/html/rfc4122.

[31] “Bulletin Board,” 10 11 2015. [Online]. Available: https://en.wikipedia.org/wiki/Bulletin_board.

[32] “Bulletin Board @ 15th Ave Coffee & Tea,” 6 11 1009. [Online]. Available: https://www.flickr.com/photos/gumption/4083521070.

[33] “Usabilityfirst. introduction to user-Centered Design,” [Online]. Available: http://www.usabilityfirst.com/about-usability/introduction-to-user-centered-design/.

[34] NGINX, “Welcome to NGINX Wiki’s documentation,” [Online]. Available: https://www.nginx.com/resources/wiki/.

[35] R. Margaret, “Apache definition,” September 2005. [Online]. Available: http://searchsoa.techtarget.com/definition/Apache .

[36] P. Christensson, “Bluetooth Definition,” 2006. [Online]. Available: http://techterms.com.

[37] T. Kevin, “GATT.,” March 2014. [Online]. Available: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt .

[38] R. Margaret, “PayPal definition,” [Online]. Available: http://searchsoa.techtarget.com/definition/PayPal.

[39] R. Margaret, “Redis,” April 2013. [Online]. Available: http://whatis.techtarget.com/definition/Redis.

[40] gorillatoolkit, “Package mux,” [Online]. Available: http://www.gorillatoolkit.org/pkg/mux.

[41] P. Christensson, “HTTP Definition,” May 2015. [Online]. Available: http://techterms.com.

## Contributions

| Date | Start time | End time | Total time (hours) | Description | Yifei | Haejong | Henri | Camilo |
|---|---|---|---|---|---|---|---|---|
| 14/03/2016 | 16:00 | 20:00 | 4 | Making Evaluation Plan | X | X | X | X |
| 21/03/2016 | 16:20 | 19:46 | 3.433333333 | Lab Test | X | X | X | X |
| 22/03/2016 | 10:43 | 13:20 | 2.616666667 | Lab Test Result collecting | X | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| 1/4/2016 | 15:02 | 15:27:00 | 0.4166666667 | Questionnaire design | X | | | |
| 1/4/2016 | 16:00 | 18:00 | 2 | Questionnaire design | | X | | |
| 31/03/2016 | 16:21 | 18:24 | 2.05 | Questionnaire design | | | | X |
| 14/04/2016 | 20:00 | 22:00 | 2 | Questionnaire improvement | | X | | |
| 14/04/2016 | 19:00 | 21:00 | 2 | Questionnaire improvement | | | | X |
| 14/04/2016 | 18:58 | 19:10 | 0.2 | Questionnaire improvement | X | | | |
| 14/04/2016 | 16:00 | 21:09 | 5.15 | Wild test | X | X | | X |
| 15/04/2016 | 10:50 | 11:40 | 0.8333333333 | Wild test | X | X | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| 17/04/2016 | 10:00 | 16:00 | 6 | System stress testing | | | X | |
| 18/04/2016 | 9:58 | 11:45 | 1.783333333 | photos collecting + report writing assistance | X | | | |
| 18/04/2016 | 12:00 | 23:59 | 11.98333333 | Report writing | | | | X |
| 18/04/2016 | 20:00 | 23:00 | 3 | Report writing | | X | | |
| 18/04/2016 | 20:00 | 23:00 | 3 | Report writing | | | X | |
| | | | | | | | | |
| Total time | | | | | 18 | 20 | 16 | 29 |