# CS M117 Project Report
# Fall 2017
# Discussion 1B

# Team Neighborhood:
# Family Watch


Brian Tehrani - 604715464 - btehrani@g.ucla.edu
Lewis Hong - 904645894 - smlewishong@gmail.com
Ivan Manan - 504620372 - ivanmanan@ucla.edu
Yong Bum Yoo - 304846923 - yongy0221@gmail.com

**Abstract**

A web application is designed for parents to track the location of their lost children through google maps. The parent will be continuously updated on the location of each child logged into the website and track their path. The technology used to implement the Family Watch software are Wireless LAN (WLAN) and GPS with a background code of Node.js and React. In the current build, everyone will know of each person's location. Improvements can be made on the current build, such automatic alerts and updates when children are away from their parent within a set distance.

**Introduction**

The motivation for the Family Watch web application is to protect civilian life while arming parent enforcement with the technology to effectively locate childs. A local neighborhood parent station will act as the host and the child's will act as users. Parents can request the location of their child at any time, and the child's coordinates will appear on the front page. The child's location will be indicated by latitude, longitude, and pin marker on a google map. A GPS history of each child will be recorded by the server in order to keep track of every escaped child's previous locations to see if they caused any neighborhood damage. Each person will have a login account to the Family Watch website and connected through 802.11b WLAN.

Ensuring that our development team was working in unison, a Github page was used to git and pull all source code for each member. The four group members were split into two groups of two, where the first group worked on the front end and the second group worked on the back end. The front end group focused on the HTML/CSS, and React, and maps integration. The back end group focused on the type of connection used, Node.js, database, and security necessary to ensure proper client protection. Collaboration between group members was maintained through Github, iMessenger from Facebook, and Google Drive. Debugging was conducted on the entire code by each member of their respective groups to ensure an equal workload.

# Technical Background

## Front End

### HTML/CSS

With a development time of about five weeks, our development team must consider the quickest way to implement the Family Watch application. A web application sounds the most promising compared to an iOS or Android application because a web application offers quick debugging, implementation, and access for other users. Additionally, our development team felt most confident when handling and designing with HTML/CSS.

### React

For the UI, our development team decided to use React along with HTML/CSS to efficiently implement view pages on Node.js framework. Family Watch has to keep track of location histories of users and show them on the Google Maps in real-time. React allows view pages to change without reloading which makes the UI of Family Watch smooth and better. In addition, React allowed our development team to implement Google Maps API which is the most important part of our front end.

### Google Maps

Creating an original map using React and Node.js would take up a lot of development time away from implementing the other features of Family Watch. Incorporating a pre-built map application and UI would speed up the development time while making Family Watch more user friendly. Google Maps is used to ensure a reliable and quick way to see the exact location of the missing child as well as their surrounding location. The API of Google Maps allows our development team to add features, such as tracing the child's start to finish location and placing arrows to find the closest distance for the parent to reach their lost child.

## Back End

### Node.js

For the back end of Family Watch, our development team decided to use Node.js framework to build the server side of our application. With Node.js, our server gets the GPS locations of the user and saves the history in our MySQL database. In addition, our server connects with our MySQL database of user informations and GPS location histories. In order to pass data among the server, the view pages, and the database, HTTP GET and POST methods were used with Node.js. Then, the data passed to client side are rendered within React components with the Google Map.

### 802.11b WLAN

WLAN is prefered for Family Watch for it's quick implementation and wide area of coverage. The ability for WLAN to send smaller packets reduces the chance of packet corruption by collision or noise, while also requiring less overhead if the packets require retransmission [1]. WLAN 802.11b uses an unregulated radio frequency of 2.4 GHz, while supporting bandwidths up to 11 Mbps. However, due to 802.11b's unregulated frequency interference can occur from appliances that use a similar 2.4 GHz wireless frequency, such as microwave ovens and cordless phones [4]. The event of interference can be ignored due to the likelihood of not experiencing datagram loss from data in Experiment 2 of Computer Networks [1]. Bluetooth could only operate at close ranges because over longer ranges the data rate significantly decreases since the packets can't traverse longer wireless distances.

The type of Internet Protocol (IP) traffic important to the family watch application is User Datagram Protocol (UDP). The benefit of UDP's connectionless protocol is fast and efficient transmission of data packets of small size. Additionally, "UDP's stateless nature is also useful for servers that answer small queries from huge numbers of clients" [5]. UDP implementation will allow reception of datagrams on any wireless service. This is especially beneficial when the lost child is away from an 802.11b wireless connection and must rely on cellular 4G or LTE to connect with their parent. The downsides of UDP

often summarizes to the lack of error correction and flow control over the datagrams being sent. Since the datagrams send all at once from the connection source, fear of loss of connection when the child is in an isolated area may occur. Additionally, erroneous datagrams are discarded rather than checked adding more worry to the IP design of Family Watch. These flaws presented may sway the IP design model to focus on Transmission Control Protocol (TCP). The benefits of UDP outweigh the flaws presented as UDP constantly streams small packets in order for the connection to be established without regard in noticing a receiver such as TCP. Finally, a TCP IP design will show overall slower connection speeds if multiple families have lost children in the same area due to greater overhead and congestion control [5]. Data tables of data rate speeds and interference between UDP and TCP are given in tables 1 and 2 in the reported results section of this report.

**GPS**

Incorporating additional GPS coordinates improves the consistency, area, and accuracy of our integrated google maps UI. Most of the coordinates were taken around the UCLA campus and work fairly accurately. The benefits of GPS technology allow for an around the world coverage, increasing the likelihood of finding a lost child in remote locations. The aid of GPS becomes more apparent when Family Watch is used with mobile devices, as such devices already have dedicated GPS location services, thus making GPS integration smoother. Improvements in Linux assisted GPS commands allow android devices to obtain their time to first fix connection for faster satellite connection [6].

**MySQL**

Implementing the database requires MySQL. The database will house the information of the parent and child login information as well as the location history of the child. The login will require authorization once the account is created. To login back to the website, the user credentials will be entered, once entered the authorization will ping

the server which holds the record of the user. Once the authorization is received, the user can enter the Family Watch website. The built in protection MySQL offers gives both our development team and users confidence that information will remain secure. As users increase overtime, MySQL offers virtually unlimited scalability and performance when configuring accounts stored in the database. Ease of usability and automatic database management features of MySQL offer our development team the necessary features to create and maintain Family Watch within the development time [3].

**Firewall**

In order to establish secure and private connections between a server and client, a firewall must be enabled. For private connections, a packet filtering firewall [2] would be used where the server defined how often to ping the client. Firewalls enable proper filtering of channels if surrounding interference becomes too great for web applications to handle. This ensures that connections are reliable and secure. However, since the server had to host a web application within a Wireless Location Area Network, for simplicity sake the firewall was shutdown in order to successfully host the web application for clients. Despite the security risk of a disabled firewall, the server was successfully able to connect to all the clients.

**Report Results**

| Table 1: TCP and UDP Signal Throughput vs. Signal Strength | | | | | | |
|---|---|---|---|---|---|---|
| Location | Approx Distance from Access Point | Signal Strength (dBm) | Noise Power (dBm) | SNR (dB) | UDP Data Rate (Kbps) | TCP Data Rate (Kbps) |
| a. Wall across 3704 | 30 | -40 | -83 | 43 | 10400 | 8580 |
| b. Next to 3428 Suite Main Entry | 60 | -58 | -82 | 24 | 6540 | 4240 |
| c. Next to 3424 Entry | 90 | -62 | -87 | 25 | 2950 | 249 |

| Table 2: Noise and Throughput in presence of Microwave Oven | | | | |
|---|---|---|---|---|
| Microwave Oven Level | Signal Strength (dBm) | Noise Power (dBm) | UDP Data Rate | TCP Data Rate |
| 5 | -27 | -83 | 23.9 * 10^3 | 15.7 * 10^3 |
| 4 | -33 | -80 | 21.4 * 10^3 | 14.1*10^3 |
| 3 | -22 | -80 | 12.9 * 10^3 | 2730 |
| 2 | -23 | -80 | 16.9*10^3 | 2.44*10^3 |
| 1 | -22 | -81 | 12.1*10^3 | 8.16*10^3 |
| 0 | -28 | -80 | 22.0 * 10^3 | 15.7 * 10^3 |

| User | Time Record | Longitude | Latitude |
|---|---|---|---|
| Ivan | 2017-12-04 13:11:23 | -118.449753 | 36.068707 |
| Lewis | 2017-12-04 14:04:33 | -115.000000 | 36.000000 |
| Brian | 2017-12-03 10:11:23 | -118.224000 | 35.314542 |
| Yongbum | 2017-12-03 4:11:23 | -116.000000 | 32.000000 |

Sample GPS Data Table used for the MySQL setup.

**Implementation**

The expected functionality of the Family Watch web application, given five weeks of development time, works through the following steps:

1. A parent and child create an account on Family Watch.
2. If the child is lost, the parent can click the checkbox of the child's name on the right hand side of the Family Watch page to see the child's exact waypoint with spacial locations.
    a. Both the parent and child logged into the webpage can see each other's location.
3. The location history of the child will be readily available every time the parent pings the child.

**Challenges**

A few challenges slowed the development time when creating Family Watch. The original plan was to implement a Police Watch web application where police acting as host can have a one way knowledge of the location of escaped prisoners, which act as the users. This implementation had to change because both the host and the user know of each other's location. The design model changed to Family Watch as it is important for both the parent and child to know of each other's location. Additionally, in order for a cellular connection to work as a user, the user and the host must be connected to the same WLAN.

**Conclusion**

Although the Family Watch web application is operational, there is always room for future improvement. The current build only allows those with a wireless connection to notice each person's location. In the event that a child does not have a computer or phone, future devices such as locator chips on wristbands can be given for children to wear. For this implementation to work a more powerful Ad-Hoc network must be used. Additionally, phone alerts and automatic updates can be implemented for the Parent in the event the child leaves the parent over a certain distance. The updates can be received through the phone or on the Family

Watch website and once clicked the wristband will blink indicating the child that the parent is looking for them.

**Appendices**


Figure 1: Front Page of Family Watch

```
mysql> select * from User;
+---------+----------+----------+-------+
| User_ID | Username | Password | Host  |
+---------+----------+----------+-------+
|       1 | ivan     | password | true  |
|       2 | lewis    | password | false |
|       3 | brian    | password | false |
|       4 | yb       | password | false |
+---------+----------+----------+-------+
4 rows in set (0.00 sec)

mysql>
```

Figure 2: Database and login information

```
mysql> select * from GPS;
+------+---------------------+-------------+-----------+
| hid  | Time_record         | Longitude   | Latitude  |
+------+---------------------+-------------+-----------+
|    2 | 2017-12-04 13:11:23 | -118.449753 | 34.068707 |
|    3 | 2017-12-04 13:13:19 | -115.000000 | 34.000000 |
|    2 | 2017-12-04 13:21:18 | -118.449760 | 34.068710 |
|    2 | 2017-12-04 13:45:55 | -118.449760 | 34.068710 |
|    2 | 2017-12-04 14:01:13 | -118.449760 | 34.068710 |
|    2 | 2017-12-04 14:01:13 | -118.449760 | 34.068710 |
|    1 | 2017-12-04 14:07:11 | -118.449753 | 34.068752 |
|    2 | 2017-12-04 14:07:22 | -118.449760 | 34.068710 |
|    1 | 2017-12-04 14:07:41 | -118.449738 | 34.068752 |
|    1 | 2017-12-04 14:08:11 | -118.449745 | 34.068760 |
|    1 | 2017-12-04 14:08:41 | -118.449768 | 34.068764 |
+------+---------------------+-------------+-----------+
11 rows in set (0.00 sec)

mysql>
```

Figure 3: Sample GPS coordinates

**Responsibility Assignment**

We will have everyone doubled-up on certain portions of the project. The reason being is for easier debugging and in case one person's portion ends up being too easy and fast to complete, that person will have more things to do.

| | |
|---|---|
| Brian Tehrani | Front-end + Figure out GPS/Google Maps API's |
| Lewis Hong | Front-end + Figure out GPS/Google Maps API's |
| Ivan Manan | Back-end + Figure out WiFi, server setup, and database |
| Yong Bum Yoo | Back-end + Figure out WiFi, server setup, and database |

Presentation: https://youtu.be/g36jiHVBFH8

Demonstration: https://youtu.be/kG9u_v0YBbw

**References**


[1] M. Gerla and R. Dzhanidze, *Computer Networks: The Physical Layer*, Fall 2017. Los Angeles: UCLA, 2017.

[2] V. Beal, "Firewall Definition," *webopedia*, 2014. [Online]. Available: https://www.webopedia.com/TERM/F/firewall.html. [Accessed: 26-Nov-2017].

[3] T. Branson, "8 Major Advantages of Using MySQL - Datamation," *Datamation - ITBusinessEdge*, 2016. [Online]. Available: http://www.datamation.com/storage/8-major-advantages-of-using-mysql.html. [Accessed: 26-Nov-2017].

[4] B. Mitchell, "Wireless Standards 802.11a, 802.11b/g/n, and 802.11ac," 2017. [Online]. Available: https://www.lifewire.com/wireless-standards-802-11a-802-11b-g-n-and-802-11ac-816553. [Accessed: 03-Dec-2017].

[5] K. T. *et al.*, "TCP vs UDP - Difference and Comparison | Diffen," *Diffen.com*, 2015. [Online]. Available: https://www.diffen.com/difference/TCP_vs_UDP. [Accessed: 03-Dec-2017].

[6] D. Haase, J. Weber, and F. Unger, "AGPS | GPS Data Logger Software - BT747," *bt747.org*. [Online]. Available: https://www.bt747.org/book/agps. [Accessed: 03-Dec-2017].