Exercise 3: Small IoT Implementation

Sofeem Nasim, Aditya Mehta, Xinyi Yang
sofeem.nasim@oulu.fi, amehta20@student.oulu.fi, xiyang18@student.oulu.fi

10 May 2020

1 Brief introduction and demo implementation

The project, named as WhereRU, is a smartphone application (current implementation for Android Operating System platform) which provides the functionality of one-click location sharing to the users with a simple user interface.

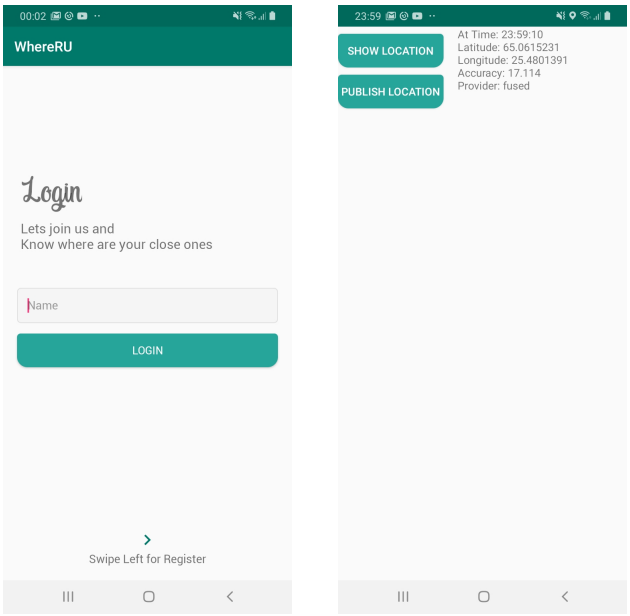
The demonstration video can be accessed at https://youtu.be/1NOJ-s9mopQ

2 Description of the design

2.1 Design description

This application lets the user share their location easily with their loved ones. The location sharing could be unidirectional as well as bidirectional. Although existing applications such as WhatsApp or Facebook may provide location sharing capabilities, however they are not focused on location sharing. Our proposed application (WhereRU) enables the user for easy one-click location sharing.

As depicted in application screenshot in figure 1a, user can register by providing basic information such as user's name and names of people, with whom user wants to share the location. Figure 1b illustrates that user needs to provide their unique username to login to the application. The application is designed so as to provide easy login. No need to remember complicated passwords. On the home page, the user gets the option to update their location. The other users of application such as friends and family members of this user can subscribe to the update of this user. As soon as the user updates the location in the application, notifications are shared with other users who are subscribed to this user's location updates.



(a) (b)

Figure 1: Register and Login screen in application

2.2 Need of this design

The students as well as working professionals have to travel frequently to places, such as university, company office, supermarket among others. The family members and friends are often concerned about user's whereabouts. The user often needs to make a call to their family to share their location. The applications such as WhatsApp or Facebook are not focused on location sharing, also it could be difficult to use them for location sharing, specially for elder people. This application is lightweight, and the user can share their location in just one click. The objective of WhereRU is: "Your location message to friends and family is just one click away."

2.3 Target users who will be benefited from the solution

WhereRU is best suited for users living away from their family with everyday frequent travel needs. For instance, an international student at the University or their family members and friends. Often their family members are worried for their whereabouts and if they are doing well. Also, existing applications with many features can be confounding to elder age users. The application is designed in such a way that users of all age groups can use it easily and be benefited by its functionality. This application is lightweight, and its sole functionality is one-click location sharing with a simple user interface.

For instance, three customer journeys are explained in the following sections.

2.3.1 Portfolio 1

Michael is an international student at the university. When he goes to the university, he can easily share his location with his parents living abroad, so they are not concerned for his whereabouts.

Figure 2: Customer journey for Michael

2.3.2 Portfolio 2

Laura is a working professional in city of Oulu. On weekdays, she travels to her office in city center. On Friday evening, when she gets an important thing to finish before leaving for meeting with her friends, she can easily share her location with her friends so that they do not keep waiting for her.

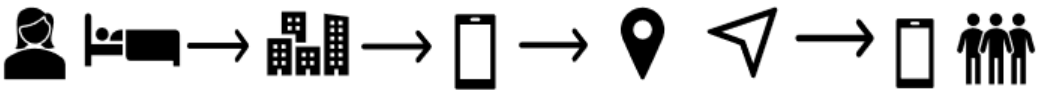


Figure 3: Customer journey for Laura

2.3.3 Portfolio 3

On the other hand, when Laura's parents plan to go to supermarket while she is in her office. They can easily share their location in the app, so that she is relaxed and not constantly worrying about them.

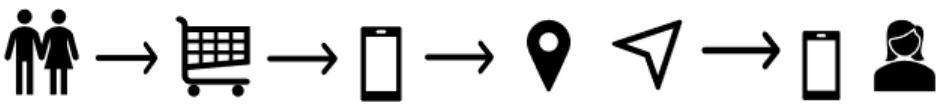


Figure 4: Customer journey for Laura's parents

2.4 Value proposition

A value proposition is a promise of value to be delivered, communicated, and acknowledged to the users of the product (WhereRU). It is a belief from the customer perspective about how value will be given, experienced, and acquired by the users.

Need, approach, benefit, and competition (NABC) is a useful tool to illustrate value proposition as it highlights the value for the end-users. This section follows the NABC model to describe the value proposition.

- Need means what the customer and market needs are.
- Approach means what your unique approach for addressing this need is.
- Benefit means what the specific benefits for the stakeholders are.
- Competition means how the benefits are superior to the competition and the alternatives.

The NABC model for WhereRU has been depicted in table 1.

Table 1: Need, Approach, Benefit, Competition for WhereRU

Category	**Description**
Need	Helping users to track their kids Helping users to track and meet up with friends Concise interface
Approach	WhereRU smartphone application Message Queue Telemetry Transport protocol Android Studio (Paho MQTT Client with Java) CloudMQTT
Benefit	Users able to track their friends easily
Competition	WhatsApp real-time location Facebook location Owntracks sharing

3 Technical implementation

WhereRU application implementation consists of these main components.

1. Smartphone devices (location sensors)
2. User interface for mobile application
3. Communication protocol (MQTT)

3.1 Smartphone devices

For this application, we needed at least two smartphones devices, so that location of one device can be shared with other device. Android-powered smartphones have built-in sensors that can be used to track location of the user carrying those devices. We used the location sensor data, such as latitude and longitude of the device, which can be transmitted to other user through communication protocol MQTT, which is discussed in following section.

3.2 User Interface

We used Android Studio to build user interface for smartphone application. Once app is installed, the user needs to provide permission for reading few things such as fine and coarse location, internet, among others. Android Studio offers features such as layout for the interface, with full control on behavior of buttons, texts, images, positions, functionality, as well as testing of the application. There are three main screens in WhereRU, for each of registration, login, and location activities. The Paho Android Service is an MQTT client library written in Java for developing applications on Android, which contains a programming interface enabling applications to communicate using MQTT. The reason for choosing this library is that our application design

requirements are for Android platform, hence choosing Paho library. The source code for the application can be accessed at public repository https://github.com/Sofeem/WhereRUIot. Also, we decided to choose MQTT for our application communication needs which is discussed in detail in next section.

3.3 Communication Protocol (MQTT)

This application is based on enabling communication between mobile devices for location sharing needs. MQTT (Message Queue Telemetry Transport) protocol is being used, which is a machine-to-machine (M2M) connectivity protocol without human interference [1]. This protocol is specifically suited for Internet of Things applications as it is designed for easy communication and resource sharing between things.

MQTT is an open-sourced messaging transport protocol, which is lightweight, efficiently minimizes bandwidth usage. This is based on publisher-subscriber model, which is used to send and receive messages respectively. The protocol runs over TCP/IP, or over other network protocols that provide ordered, lossless, bi-directional connections [2]. In this model, every sensor can act as a client (which can have role of publisher or subscriber) and connects to a server, also known as a broker, over TCP. Every message is published to an address, known as a topic, which can be configured in-place already and can be used directly [3]. Clients may subscribe to multiple topics. Every client subscribed to a topic receives every message published to the topic [4].

3.3.1 Connection Methods

This model allows clients to communicate in following available connection approaches:

1. One-to-One
2. One-to-Many
3. Many-to-One

3.3.2 Choice of MQTT for WhereRU

For WhereRU design needs, we find MQTT to be the most suited because of specifically these reasons:

- MQTT ensures delivery of messages in a lightweight manner.
- One objective of MQTT design is to minimize the resource usage.
- It can support real-time monitoring by continuous data delivery.
- It is ideal for mobile applications because of its small size, low power usage, minimised data packets, and efficient distribution of information to one or many receivers. [1]
- This is open sourced so it can be deployed easily and free of cost.

3.3.3 Quality of Service

The Quality of Service (QoS) defines how hard the broker will try to ensure that a message is received.
There can be three qualities of service (QoS) for message delivery. [2]:

1. "At most once", also known as "Fire and forget"
    - Messages are delivered with possibilities of few messages loss, which is not highly critical as next message is published soon. No confirmation.
2. "At least once"
    - Messages are assured to arrive but with possibility of duplicates, delivery confirmation is required.
3. "Exactly once"
    - Messages are assured to arrive exactly once, with four step handshake.

3.3.4 Architecture

The major components of architecture can be:

- **Publisher**
  The device, which is sending data on the channel.
- **Subscriber**
  The device, which is receiving data on the channel.
- **Message Queue**
  - Message queues provide an asynchronous communications protocol.
  - The sender and receiver of the message do not need to interact with the message queue at the same time.
  - Messages placed onto the queue are stored until the recipient retrieves them or until the messages times out. [5]
- **Broker**
  - Broker is used to distribute messages to interested clients based on the topic of a message.
  - The broker server task can be to
    * receive all messages
    * filter and rectify the interesting subscriber
    * send/publish a message to all subscriber
- **Database**
  Database is used to store the details of users' names and the topics to be subscribed. SQL Lite database server is used for WhereRU application implementation.

3.3.5 Cloud MQTT (public broker used for WhereRU

Although it is possible to design the broker from scratch to fulfill the needs, however for this application design, we decided to choose the one public broker (CloudMQTT of Mosquitto type implementation), which is suited for IoT applications with messaging needs between low power sensors or mobile devices such as phones. CloudMQTT are managed Mosquitto servers in the cloud.

For CloudMQTT, we created an account with the service. The service let us choose an instance, associated with a specific data center in the cloud in nearby region. After providing all details on the portal, instance is provisioned in few minutes, which shows the default connection information as well.

The technical specifications for CloudMQTT has been described briefly in table 2 and 3.

Table 2: CloudMQTT Broker Details

Field	**Value**
address	m12.cloudmqtt.com
port	Default: 18443, 28443 (SSL) Application: 11111
type	mosquitto

Table 3: CloudMQTTdetails

Instance type	**Plain MQTT**	**MQTT over TLS**	**MQTT over Websockets**
Shared	1XXXX	2XXXX	3XXXX
Dedicated	1883	8883	443

3.3.6 Use Case Architecture

For example, imagine a simple network with three clients and a central broker. All three clients open TCP connections with the broker. Clients B and C subscribe to the topic location.

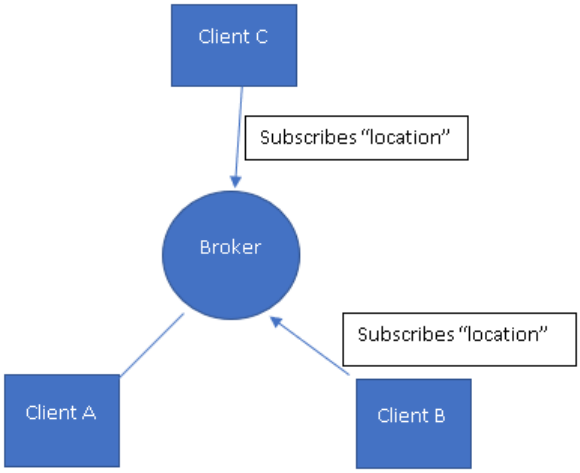


Figure 5: TCP connection with broker and clients B, C subscribed

At a later time, Client A publishes a value of "Uni Oulu" for topic location . The broker forwards the message to all subscribed clients.

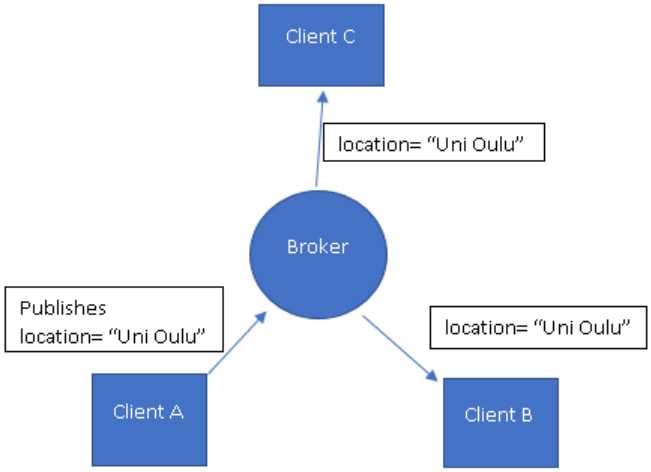


Figure 6: TCP connection with client A publishing a message

References

[1] MQTT.org. Mqtt (machine-to-machine (m2m)/"internet of things" connectivity protocol). http://mqtt.org/, 2020.

[2] A Banks, E Briggs, K Borgendale, and R Gupta. Mqtt version 5.0. *OASIS Standard*, 2019.

[3] R Light. Mqtt man page—eclipse mosquitto. *Disponıvel em: http://mosquitto. org/man/mqtt-7. html. Acesso em*, 9, 2016.

[4] Toby Jaffey. Mqtt and coap, iot protocols. https://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php, 2020.

[5] CloudMQTT. Cloudmqtt faq. https://www.cloudmqtt.com/docs/faq.html, 2020.