IoT Application Requirements

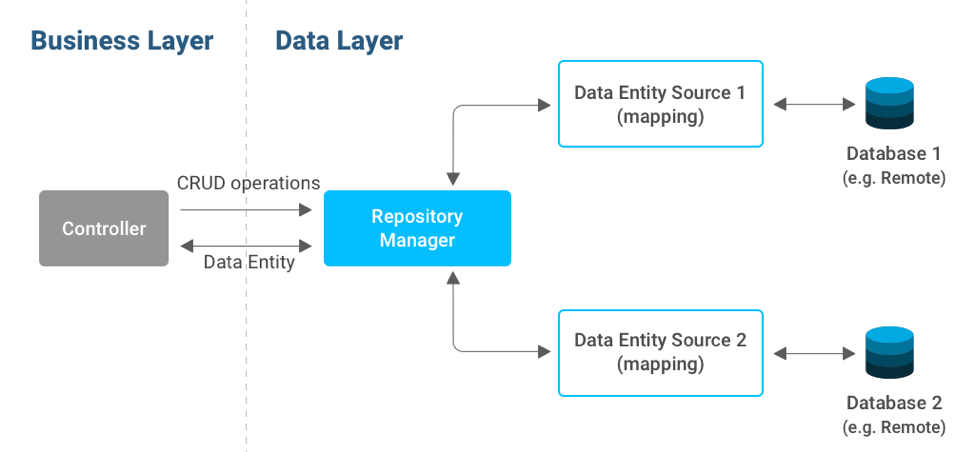
In this section we will describe the requirements of IoT application arising due to the application.

Security Privacy, trust, confidentiality and integrity are considered important security principles for IoT due to the large number of devices to the internet [24]. Since IoT application uses data in various forms, speed and from a variety of sources, it is important it incorporates trust mechanisms that enforce privacy and confidentiality. In addition, IoT application must integrate mechanisms to check for the integrity of data to avoid the erroneous operation of IoT applications. 3.2. Adaptability IoT systems will consist of several nodes, which will be resource constrained mobile and wirelessly connected to the Internet. Due to factors such as poor connectivity and power shortage, nodes can be connected and disconnected from the system arbitrarily. Furthermore, the state, location and computing speed of these nodes can change dynamically. All these factors can make IoT systems to be extremely dynamic. In a physical environment that is highly dynamic, IoT application needs to be self-adaptive to manage the communication between the nodes and the services using them. [3] presents that IoT applications need to designed and developed in a way that it can efficiently and effectively react in a timely manner to the continuously changing context in accordance with, for instance, business policies or performance objectives that are defined by humans. IoT applications should be self-optimizing, self-protecting, and self- configuring, resilient and energy-efficient. 3.3. Intelligence Intelligent things and system of systems are the building blocks of Internet of things [26]. IoT applications will power IoT enabling technologies in transforming everyday objects into smart objects that can understand and obtain intelligence by making or enabling context related decisions, resulting in the execution of tasks independently without human intervention. Achieving this requires IoT application to be designed and developed with intelligent decision-making techniques such as context-aware computing service, predictive analytics, complex event processing and behavioural analytics. 3.4. Real time A number of IoT domains requires the timely delivery of data and services. For instance, consider IoT in scenarios such as telemedicine, patient care and vehicle-to-vehicle communications where a delay in seconds can have dangerous consequences Environments where operations are time-critical will require IoT applications that provide on-time delivery of data and services. 3.5. Regulation Compliant IoT applications may collect sensitive personal information about people daily activities such as detailed household energy usage profile and travel history. Many people consider this information as confidential. When such information is exposed to the Internet, there is a possibility of privacy leakage and this could affect the privacy of the individual. In order not to violate the privacy of people, IoT applications must be compliant with the privacy requirements established by law such as EU data protection rules [27], otherwise, they could be prohibited.

(3) (PDF) Developing IoT applications: Challenges and Frameworks. Available from: https://www.researchgate.net/publication/320571136\_Developing\_IoT\_applications\_Challenges\_and\_Frameworks [accessed Jul 11 2020].

The mobile application is built using android platform because Android Market is open, whereas, the Apple App store is gated. The low level diagram of the android software platform is shown below. The whole system contains two parts: a mobile client app on Android OS platform and a remote data server. Wireless Network is the only connection between client and server. After submitting a request from a client, the server will receive and respond this request and return the processing results to client.





The main focus of the application layer is the graphical user interface and the operation logic, In this layer high level APIs can be accessed directly. This is platform-specific programming interface that supports Java when developing application using Android SDK.

Communication between the Mobile and Web Server

The relationship between the mobile device and the Web server is referred to as the client-server model. The client is a mobile device or a computer that initiates contact with the server to use a resource. The server is a server system that selectively shares its resources. Communication between the client and the server occurs when the client sends a request and the server returns a response. To communicate, the computers must have a common language so that both the client and the server know what to expect. Figure 9 shows the client-server architecture. The client is an Android mobile device, and the server side consists of a combination of PHP script and MySQL database. The PHP script acts as the connection protocol. Communication occurs when the user clicks an upload button to synchronize the health data. To connect to the PHP script, we use the HTTP protocol from the Android system and the JSON format for data synchronization of the name-value pairs.

Synchronization of Health Data

This feature is intended to synchronize the tables in the database between the Android-based and the Web-based health monitoring systems. To achieve a cloud computing service in the proposed system, synchronization of health data is implemented using HTTP POST and JSON. The algorithms for the uploading and downloading of the health data are shown in Figure 10. Figure 10. (a) Uploading process from the mobile device to the Web server, which involves the creation of a database table and insertion of the health data into the Web server database; (b) Downloading process from the Web server to the client side where the clients retrieve the health data from the Web server database.

On the client side, which is the Android mobile device application, when the user clicks the upload button, the Android application will call the HttpPost() method defined in HttpClient class, which Sensors 2013, 13 16462 results in running a PHP script in the Web server. The uploading process from the mobile device to the Web server database is shown in Figure 10a. In this scenario, the date, time, HR, and location are synchronized into the Web server database. First, the exact date and time are inserted in the database with the table name Alldates, followed by a new table HealthStatus\_date based on the newly created date. Then, all the health data, HR, and location are uploaded and stored in the SQL database. At the same time, the Android application calls the jsonInsert() method to build up an array of name/value pairs for the date, table name, date and time in a timestamp format, HR, and location. Then, the PHP script (jsonInsert.php) runs a database query to store the JSON data into the Web server database. The description of the JSON function and the PHP pair is shown in Table 2. Table 2. Description of the JSON function and PHP script pair. JSON Function Description jsonInsert() This JSON function and PHP script pair is used to upload the date, table name, date and time in timestamp format, HR, and location into the Web server database. PHP Script jsonInsert.php To download health data from the Web server database to the browser, the HTML program calls a PHP script in the Web server where the PHP script retrieves and obtains the health data from the database. The downloading process from the Web server database to the mobile device is shown in Figure 10b. In this process, the PHP script connects to the MySQL database and searches for the table Alldates. Then, the number of rows in the table Alldates is counted by getting the lastrow() of the table. Next, the PHP script retrieves the latest modified health data from the table HealthStatus\_date and presents them on the operational Web page. In addition, the retrieved HR is plotted accordingly in a graph using JavaScript.