iNutriCare: Web Service Developments (REST with JSON)

I-Sheng Chris Wang, The University of Adelaide, a1699186@student.adelaide.edu.au

***Abstract*— The senior citizen determines as a social burden as well as economic liability has now become a common social issue in our society. It has been focusing on our society today and this issue seem to like is inescapable. As a person gets older, they incline to have trouble with speaking, memory loss, hearing, vision problem, losing the muscle and most crucially is having dementia. When the elderly person having dementia, they will forget what they have eaten or even forget when to eat. As time goes further, their body health will get weaker and asthenic due to lacking to good nutrition. Although the elderly person with dementia can be seen as burdens most of the time rather than the blessing, yet sometimes they will need to care for themselves due to the society will forget to look after the older people at home. There is a high demand for the care services for the recent decade as there are an increasing number of the elderly people for those who are living independently. Consequently, with recent high technology development nowadays, it has become capable of exploiting the nutrition system in a mobile application for the elderly people who live independently with dementia. Therefore, in order to assist the elderly with dementia to stay healthy by having a good nutrition and feel more comfortable and unrestrained, we have proposed a mobile application system in our project that can be used by the elderly people, caregivers, and family. In the following sections, I have presented that why most new APIs are built in REST with JSON today.**

***Index Terms*— Caregiver, Internet of things (IoT), JSON, Patient, Representational State Transfer (RESTful) Web services.**

I. Introduction

ONE of the major health problem is dementia and it is the second leading cause of death in Australia [1]. Generally, the symptom of dementia is an intense impact on the lives and caused by the significant death of the brain cells. The disease of the dementia is distinguished by the brain damage function which including losing memory, perspective, the behaviour of personality, planning and fail to control many essential feature of their living style and cognitive skills [2]. It is usually of gradual onset, incapable of being reserved, and progressive even though the level of symptoms and the type of their development pattern are different with the type of dementia [3]. Nonetheless, according to the nutrition monitoring and assistance for the older people's report, it had indicated that whichever the elderly people were having the lower cognitive ability in their daily life, it was correlated to the higher malnutrition. Meaning to say that because of the elderly people having a certain level of dementia, they may not serve themselves a good nutrition for the daily caloric or fluid intake. Consequently, it must have some kind of care systems of care services to assist them in their daily life especially for those older people who live independently with dementia at home despite that they will have a good nutrition for every meal and live for a longer time.

With the increasing age of the population, the primarily affected group will tend to the older people. Meanwhile, based on the prediction of growth of elderly people population in Australia recently, the number of the elderly people who is having dementia is expected to rise to 450,000 in the next five years [2]. Even though the prediction is hard to estimate, yet "*the number of people with dementia is projected to reach around 900,000 by 2050*" dedicated by [4]. Besides that, there are more than 380,000 people in Australia having dementia, whereas nearly 354,000 of the elderly people who defined as more than the age of 65 are living with dementia. In the meantime, nearly half of them are living independently with dementia [2]. It could be said that it can be hazardous for those elderly is living without the care systems or care services. As a result, the number of deaths thus dementia raised 2.5 times from 4975 to 11,120 deaths between 2005 and 2015 [3]. As we can see, one of the main activities that cause death from the elderly people is having malnutrition. It is related to their poor daily food intake and seems to be losing weight.

A study found that in order to prevent the elderly person from becoming depressed or even exploiting the dementia, they need to visit their family or relatives at least three times per week. However, it is hard to maintain this condition about the elderly people that visiting their family or relatives. As time goes, if the condition does not perform well, they will be started to feel lonely and depressed and develop dementia eventually. Undoubtedly, with the recent growth of Internet of things (IoT), everything is now in place for the IoT to work. At the same time, it can be very useful for the ordinary people, especially the elderly with dementia. People can just control almost everything through their smartphone. In general, IoT is a machine-to-machine communication to track machinery operations, provide service alerts and information that widely used in all across the global, especially the manufacturing and energy sectors [6]. Meaning to say that it is not just the people who are connecting or browsing the internet, meanwhile, the objects that surrounding the human beings can be connected to the internet too and we have named it as smart objects.

Specifically, the smart object is always dealt with the Radio Frequency Identification (RFID) technology. It is involving the smart electronic tag of identification's attachment that can be decoded from a certain distance to identify a product and its descriptions as well. In the meantime, those smart objects can collect the information regards the privacy of human's lives as well as the world's safety by controlling the distance poses over

the physical environment [7]. Affirmatively, IoT can lead to change a person behaviour and living style simply because the smartphone can retrieve the background information from a difference context of mutually beneficial databases, moreover it is capable to enlarge the cognitive content and functioning as a search engine for smart objects. Furthermore, we have used the Near Field Communication (NFC) along this project based on the alert message function that proposed by Vidivalianto (One of my group members). The reason that we decided to use NFC is that it is mainly targeted for the mobile devices, so this technology that we have used is compatible with other NFC devices and existing smartcards and readers.

Apart from that, our project's objective is to develop a mobile application for the elderly people of the age over 65 with the stage of mild and moderate dementia that helps them to have a good nutrition and stay healthy. In addition, this mobile application will be designed for two clients in our project which are the caregiver and patient (elderly people). It is still reasonable that the elderly people may have some knowledge of using mobile devices, if not, the caregivers can provide the training for the elderly before started to use the developed system on the mobile application. As a consequence, the user interface of the application that we have developed on the mobile device should be ease to perform and understand easily by the elderly people. There is no reason for us to develop a care application for the elderly people who have the stage of severe or late dementia at this moment because we believe that the majority of people with this stage should be in residential care. Nonetheless, the way the caregiver and the elderly people compromise with the IoT with the mobile application is to alert, notify and provide alarm message if an emergency is developed. On the whole, I have been in charging for the RESTful Web services development along this project. In short, I will demonstrate how the functions in Web service is correlated with the android development based on the input given by caregiver or patient, and at the same time, it will return either JSON array or String output back to the users. In the following sections, I have dedicated why we have been using the JSON over the XML and Simple Object Access Protocol (SOAP) within the RESTful web service development. Meanwhile, also discussed why most new APIs are built in JSON together with REST. On the whole, this paper can be considered as a future work of [26].

II. ANALYSIS BETWEEN API OF JSON, XML, AND SOAP WITH REST

According to [21], JavaScript Object Notation (JSON) is a serialization structured data based on the format of text –based and unaffiliated data of the language that distinct the moveable representation of the structured data by a small set of formatting rules. On the other hand, JSON API is a substance that how those requests to be responded by a server and how the client should modify or fetch that particular resources. It is developed to transmit between clients and servers to minimise both the number of requests and the amount of data without compromising flexibility and comprehensibility [22]. Thus, JSON API is widely using by developers in the recent years. A simple reason that we have chosen JSON API over SOAP and XML in RESTful web service simply because JSON allow us to access in a really logical manner of data collection by human-readable code [19]. It could be said that JSON is a subset of JavaScript. When the new users just starting to get into the web service development, it can easily comprehend and read it because of using very simple grammar and language. Additionally, JSON is easy to add or change new fields. For example, as JSON can provide a simple method for returning serialised formatted responses over HTTP, the developer can transfer the data to mobile clients without the baggage. In short, JSON appeared and began to replace XML as a serialisation technology [18]. It is because the web service is a platform that approachable over standard Internet protocols of programming languages that can engage in functional building-blocks.

When talking about the JSON and XML, undoubtingly JSON is much simpler than XML. Below had shown the structure between XML and JSON about sending the credentials of username and password to certain applications.

XML

```
<credentials>
<username>caregiver001</username>
<password>abcde001</password>
</credentials>
```

JSON

```
{"credentials":[{username ": "caregiver001", "password" :
"abcde001"}]}
```

As we can see, XML contain start and end tags whereas JSON is just directly inserting onto the data structures which has a much smaller grammar and maps. Furthermore, XML will talk more number of characters to show data as compared to JSON [17]. In the meanwhile, when there is required to put in additional data or information within the structure, XML will become heavier due to lots of tags need to be included. On the contrary, JSON does not need the tag within each statement and it is light weighted, consequently, JSON is much faster than XML. With the comparison of the bandwidth of transmitting the message between JSON and XML, they are roughly the same as XML is using less bandwidth and effectively exchangeable. On the other hand, JSON is a newer encoding scheme that can execute the same thing and transmit the same message with XML. In brief, JSON has a lighter format than XML even though XML is more powerful, yet sometimes it is overused for a certain purpose.

Even though using JSON is an order magnitude easier and simple to handle with REST, yet sometimes the developers may using the SOAP with REST. One of the main reasons is that SOAP has a standard way of stipulating the data structure in a maturely formatted document of Web Services Description Language (WSDL), while JSON doesn't have. For instance, when every time setting up the JSON in a particular project, the user may always use the different data structure to accomplish it. Additionally, there is a tool of "try this request" interface that able to work with WSDL to give an enhanced of basic doc-block comment. The last reason is SOAP has infinite utilities and it can work itself without an HTTP transport. Moreover, no security is mandatory in any individual responses and returns whereas JSON doesn't have. This is why SOAP has an industry mature way to develop over the JSON sometimes. However, we

still remained to use the REST with JSON rather than using SOAP because the response of SOAP comprise five times or more junks of information when transferring the data and each request only can use an XML for data transfer [20]. There is a scenario about retrieving user information from a database. If let say using JSON in a RESTful orientation over HTTP to retrieve some information in about 4 lines of code, then a SOAP call over POST HTTP to process the request is comparatively long of over 10 lines of code [18]. Moreover, due to the developer nowadays used to be imported external libraries with their program for the need for third-party's prerequisite, so that we have decided to use the REST with JSON rather than using SOAP-based.

III. RESTful Web service with JSON development

"*Web services are client and server applications that communicate over the World Wide Web's (WWW) HyperText Transfer Protocol (HTTP)*" [13]. For further basic information about the identification of Web Service can refer to Yi Chao's work [14]. The proposed RESTful Web service is those architectural constraints and principles in an application or design whereas the principles for a Web application is completely unsettled between the interaction of the client and server requests [15]. In addition, REST can make accessible with XML and take in by the web page (HTML) without reusing an existing website architecture. RESTful Web service represents an architectural style of networked systems such as Web Expression, Web application within the visual studio, etc. In other words, RESTful Web service is a self-descriptive messages. It can store various format in the content like JSON, XML, HTML, plain text, etc. that resources are decoupled from their statement. In the meanwhile, the statement is used to control the caching, execute authentication, and discover transmission errors.

As we are developing different platforms in mobile devices, so the web service is essential. A simple reason is that it can reuse the existing function over the different network and able to remote by activate using HTTP requests to invoke the HTTP [8]. Meanwhile, web services are application components that can be used by other applications and communicate using open protocols. Full detail of why we have been using RESTful web service rather than other normal web applications along this project can refer to Abdulaziz's work[9].

According to [10], there are three types of components in the web services, which are Simple Object Access Protocol (SOAP), Universal Description, Discovery and Integration (UDDI), and Web Services Description Language (WSDL). In the meantime, the web service works by using open standards of WSDL to describe the availability of service, XML to tag the data, and SOAP to transfer a message that able to communicate with various applications.

The root resource classes are annotated with @Path that requests method designator when developing a REST root resource class [11]. There are four root resource classes that defined as an uniform interface for the operations which are create, update, read, and delete with the annotation of "@PUT", "@POST", "@GET", and "@DELETE" respectively as shown in table 1. Besides these four main annotations, we have been also widely used the annotation of "@Path" and "@Consumers". The description of the annotation as shown below:

@Path – To specify the variables in the Uniform Resource Identifier (URI) to indicate URI's path template. The resource is deployed from the URI base server, the application's context root, and response by the JAX-RS runtime from the URL pattern.

For instance, when the user need to login by using his/her username and password, the @PathParam annotation may be used on the method parameter of a request method such as @Path("*/function's name/variable's input*"), the following code as shown:

```
@Path("/login/{username}/{password}")
public String login(@PathParam("username") String username,
   @PathParam("password") String password) {
   …… ……
```

When we run the server, the proposed website will be shown like this:

URI Path Template	http://localhost:8282/iNutriCareWebServices/ login/{username}/{password}/
URI After Substitution	http://localhost:8282/iNutriCareWebServices/ login/chriswang/78960/

@Consumers – To represent a resource that used to specify the Multipurpose Internet Mail Extensions (MIME) that consume were sent by the client and to customise the requests and responses. The statement that we need to import to the java class is "javax.ws.rs.Consumes;" and the value of the @Consumers is an array of String that allows MIME types, for instance, @Consumers(*MediaType.APPLICATION_FORM_URLENCODED*). The following had shown an example:

```
@POST
@Path("/register")
@Consumes(MediaType.APPLICATION_JSON)
public String register(String jsonInput) {……
```

Full detail about the MIME types can be referred to [12].

By default, @Consumers is implemented at the class level, yet if the user implemented at the function level, it will then overrides the @Consumers annotation. In our project, we have decided to implement at every function level simply because it won't confuse and not easily override by the global annotation that declared outside the function level.

Table 1: Summary of JAX-RS Annotations. Full reference at [11].

Annotation	Description
@PUT	"The Java method annotated with this request method designator will process HTTP PUT requests." The behaviour of a resource is established by the HTTP method to create a new element in the database query.
@POST	"The Java method annotated with this request method designator will process HTTP POST requests." The behaviour of a resource is established by the HTTP method to update an existing element in the database query.
@GET	"The Java method annotated with this request method designator will process HTTP GET requests." The behaviour of a resource is established by the HTTP method to read or retrieve the existing element in the database query.

@DELETE	"The Java method annotated with this request method designator will process HTTP DELETE requests." The behaviour of a resource is established by the HTTP method to delete the existing element in the database query.

However, since we are using the RESTful web service with JSON, we do not require to develop the XML messages or WSDL service. Meaning to say we are not required to implement a client that constructs the request based on the XML interface [10]. All the user interfaces have been developing at the android development.

A. *The difference between "[ ]" and "{ }" in JSON*

Specifically, "{ }" is representing the JSONObject whereas "[ ]" is representing the JSONArray within the JSON structure. All the JSON's nodes will be surrounded by a set of square brackets or curly brackets when accessing the nodes to call appropriate method to access the data [24]. A simple JSON structure as shown as following:

```
{
        "caregiver": [
                {
                "cid" : "caregiver002",
                "FirstName" : "Steve",
                "LastName" : "Job",
                "phoneNo" : {"mobile" : "057473629", "home" :
        "89763456"},
                "gender" : "Male",
                }
        ]
}
```

As a consequence, if the user starts with the "[ ]", the user should use getJSONArray() method and put the JSON node into it. Likewise, if the user starts with the "{ }", the user should use getJSONObject() method and out the JSON node into it. So generally caregiver is a table name of the table structure, whereas the information within the curly brackets is considered as the attributes of the corresponding table. Similarly, there are two JSONObjects within the "phoneNo" array, whereas phoneNo is within the "caregiver" array.

B. *JSON Objects*

An object structure normally surrounded by the names or more than zero values/members that represented within a pair of curly brackets. A single colon comes after each name as a name is considered as a string and it separates a value by a single comma in between. All software implementations obtaining that particular object will go for the mapping of values' name that an object whose name are all unique is able to exchange and make use of information. Thus, all the names within an object should be unique. Otherwise, the result of receiving such a behaviour of an object is unforeseeable.

C. *JSON Arrays*

An array structure normally surrounded by the names or more than zero values/elements that represented within a pair of square brackets. The elements are separated by commas. An example as shown below:

```
{"allergies":[{"aid":"0000","name":"abc0"},{"aid":"0001","na
me":"abc1"},{"aid":"0002","name":"abc2"},{"aid":"0003","na
me":"abc3"}]}
```

From statement of JSON array above, the values within the array will not have the same prerequisites.

IV. IMPLEMENTATION

There is various way to implement the RESTful web services. Along this project, we have implemented the RESTful Web service by using Eclipse Luna IDE for Java EE Developers. RESTful Web service is designed using REST with JSON API and Jersey 2.22.2 API. The framework of Jersey serves as a JAX-RS in Java and provides support for JAX-RS APIs. We have chosen this because it is an open source, and production quality's framework that easy to develop with RESTful Web service in Java. Furthermore, we implement RESTful web service with JSON and deployed them on the Apache Tomcat web server with the version of 7.0.68. Apache Tomcat container runs on ASUS A450LC compatible PC with 2.4GHz Intel® Quad Core processor, 4GB RAM, and operate within Windows 10 Home Single language. For consuming REST Web service in the mobile phone, we have developed Android 4.4 (Kit Kat SDK) version of the client in Eclipse. For the database, we have used MySQL connector Java with the version of 5.1.23. The further detail about how to implement these components (Eclipse, Jersey, Tomcat, and MySQL) can be referred to [16], [23], and [25].

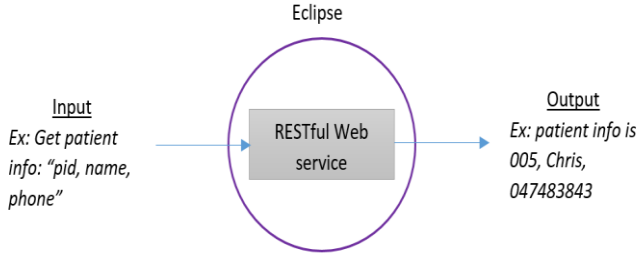


Figure 1: Outline of RESTful Web service

RESTful web service can be involved in the process of deploying, testing, and developing that helped by controlling different tasks and different sequences of steps to achieve the result. There are numbers of steps to execute the Web service on the server and produce the outputs. Firstly, create a new "Dynamic Web Project" in the Eclipse workspace that could host the web service. Secondly, create a new class with "WebService.java" and start to write the functionalities with Java language into it. The full program is attached in the GitHub account. Thirdly, run the project on the server itself and the Eclipse will call the Web service that convert the code and produce a set of web pages based on the components that generated automatically and created as an interface of client. Lastly, try to put some requests on the webpage client and determine the web service's response.

Within the operation of the class, figure 1 had shown the GetPatientInfo whereas it is called and give a string array value "pid,name,phone" and return the corresponding result.

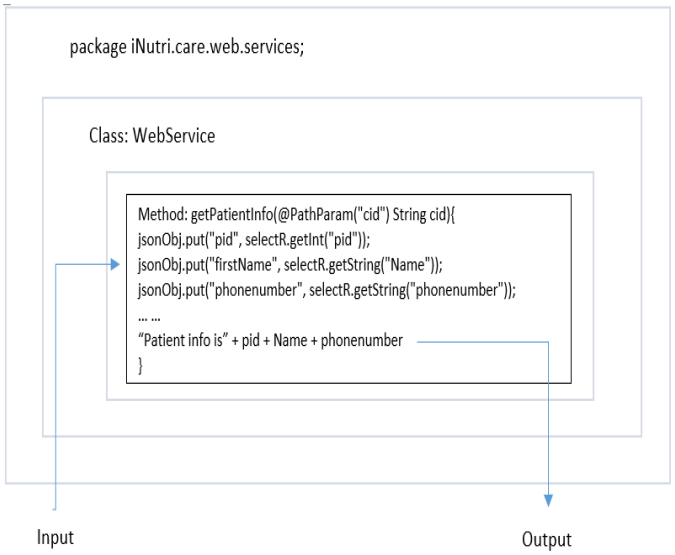


Figure 2: Calling the "GetPatientInfo" method

From the diagrammatic view of figure 2, it had specified that it could be more than a method within a class while it could also more than a class within a package. The method state an operation that can be established by the WebService class and it will take certain input such as caregiver ID (cid) along the parameter. The last statement of the method in figure 2 normally deliver the result within the particular method by returning a string in a Java statement. On the other hand, the user has to be aware of creating a new method's name as the problems will occur when trying to run and test the web service if the first alphabet of a method's name is in capital letter.

Once we have run the project's server, the output of server tab will be showing "Started, synchronized" as shown in figure 3.

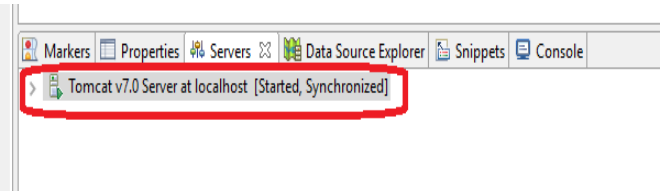


Figure 3: Message prompted after run the tomcat server

V. DEMONSTRATION

I will base my demonstration on how to add a new meal and how to retrieve and view the meal from the database and show the output of the result in mobile phone and data changed in our database. The full web service functions and descriptions will be dedicated in appendix E. For the add meal function in the web service development, certainly that after our server been synchronised, we can test the add meal function with client2.java. Before we test the user input with the add meal function, the attributes of the meal table in the database is to be shown in figure 4. There are only 15 meals in the meal list table.

In the meantime, we have inserted a meal within our add meal function as shown below:

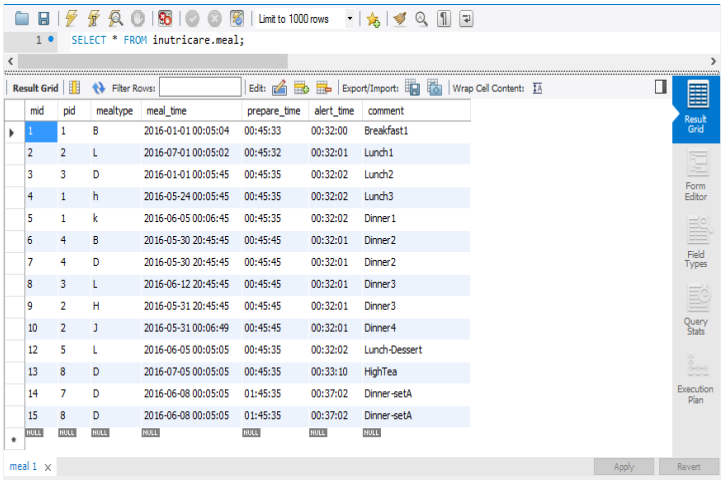


Figure 4: Meal table before insertion

```
JSONObject jsonInput = new JSONObject();

jsonInput.put("pid", 5);
jsonInput.put("mealtype", "T");
jsonInput.put("meal_time", "2016-06-15 17:06:05");
jsonInput.put("prepare_time", "0:30:25");
jsonInput.put("alert_time", "00:05:02");
jsonInput.put("comment", "High-Tea Set G");
```

After we have implemented the add meal function in the client2.java and inserted the user input, we can run the client2.java class as java application. The user input will then store the data in to database as shown in figure 5.

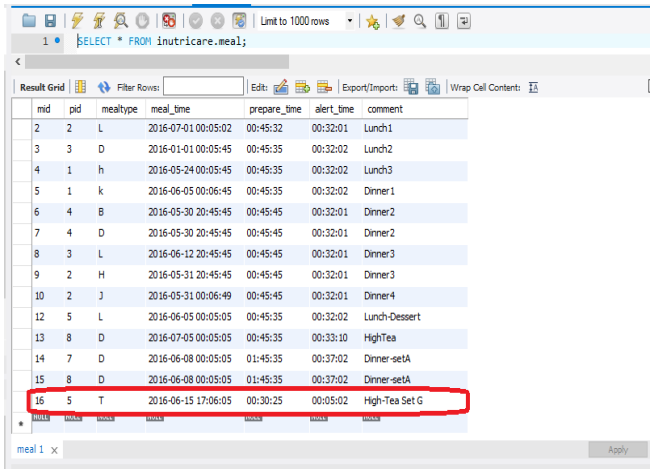


Figure 5: Meal table after insertion

In the android development, the user interface in the mobile device will be shown as JSON array in figure 6.

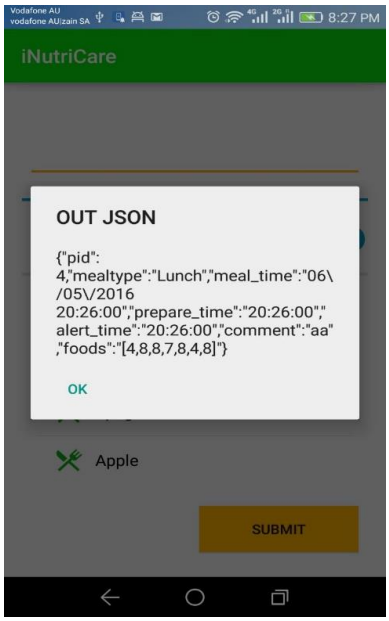


Figure 6: Sample output of JSON in Mobile app

Along the WebService.java class, we have written that every function must have the error message at every level of executing, else the user may not know what is happening next. For instance, within our web service development, we have replaced a digit to represent the error messages instead of putting "System.out.println(e.getMessage());" every time to look after what error had been prompted. When we executing successfully, the result will be shown the user input data, else it will prompt user "-2" for invalid input format, and "-3" for database level error. It is easier to decide which error that user going to fix if executing unsuccessfully.

When the user need retrieves the particular meal, we have specified the URL path with its meal ID in order to get the corresponding meal detail. The input within the method is @Path("/getMeal/{mid}"), so when we run the server, the full URL path should be written like this: "http://localhost:8282/iNutriCareWebServices/rest/getMeal/16 " whereas the number of 16 was we inserted formerly. The result of this output in our testing tomcat server is shown in figure 7.

To show it in the mobile devices, we have to connect to the internet with the same modem or router, otherwise, it will don't work on another platform. The localhost number should be same as the Web service's port number. In addition, in order to show the information in various platform and devices, the firewall of the localhost's PC must be switched off, else it will never be connected and it will prompt the error message to the user. The prefix of the URL is "Your PC's IPv4 address" : "Project's port no."/ "Dynamic Project's name" / rest/ "Web service's function name" / "Particular JSONObject" If your PC's IPv4 address is 192.168.0.10; project's port no is 8282; project name is iNutriCareWebServices; the function name is getMeal; and the input JSONObject is meal ID, so an example as shown below:
http://192.168.0.10:8282/iNutriCareWebServices/rest/getMeal/16

Any devices that connected to this URL will be accessible to the corresponding database's information. Figure 8 had shown the output of the getting particular meal detail in a mobile device.

Figure 8: Specify URL path at mobile application

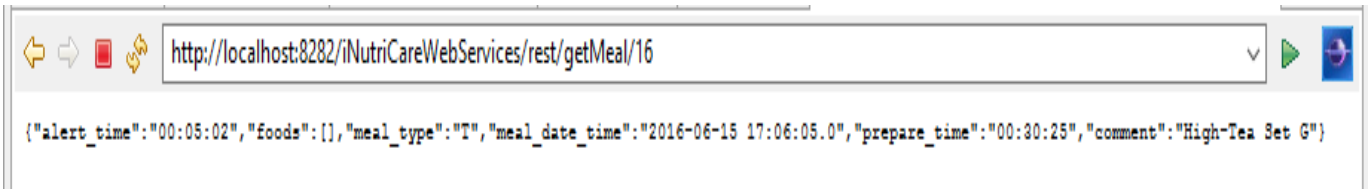


Figure 7: Getting the meal detail with the meal ID of 16

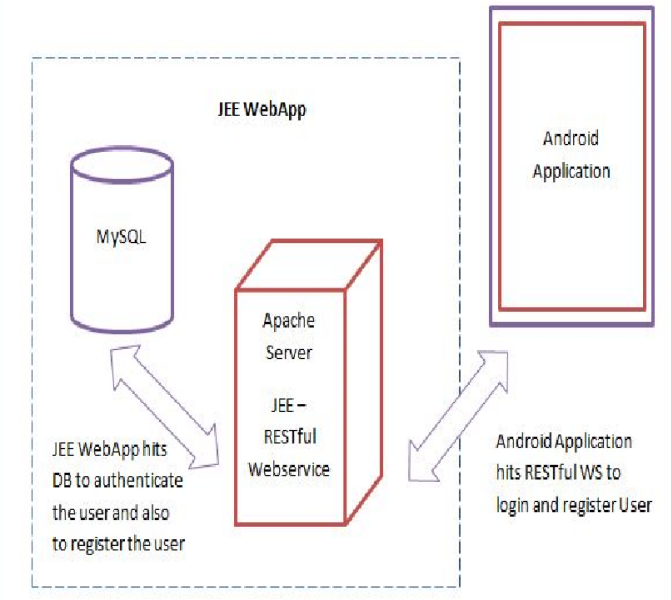
VI. Conclusion

Our studies showed that why most new APIs are built in Rest with JSON are preferable by developers today simply because due to increasing popularity of mobile applications, people are capable to approach to a spacious range of health systems and applications. Because of the demanding of developing the health application, the requirement is prompt to become simple and easier as possible. Nonetheless, that is why we have chosen this RESTful Web service along this project at the first place. It has a less tedious use of bandwidth compared to SOAP and when implementing REST with JSON, it provides improve server loading and responses times due to support from caching. Moreover, the interfaces are much easier to implement and design. Most developers nowadays love to use the third-party system or APIs to import or connect the corresponding components or libraries, so that the simplicity of developing a public APIs is always important at all time. This is the strongest reasons in our project that the major companies like Google and Amazon also moving from SOAP to REST to build their backend services. On the whole, our studies also have shown that with the health application for the elderly people at home, it can reduce the chance that the elderly admittance to the hospital.

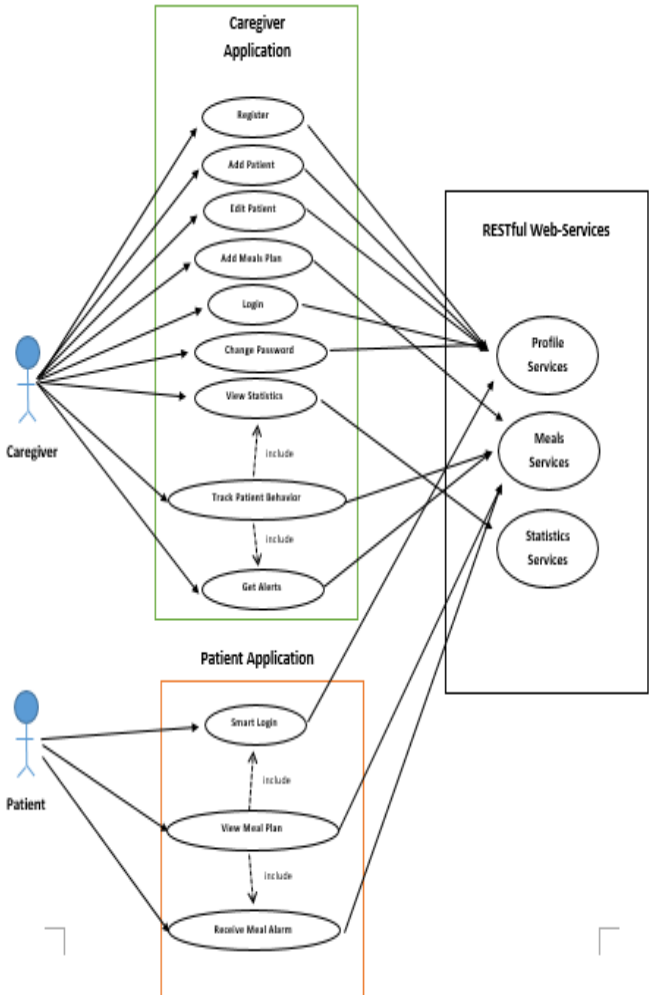
At this stage, we have completed approximately 70% of the proposed functionality for the caregiver and patient application. In the future, we will develop the application for the patient's family, so that it can enhance the mobility and accessibility for the patient to prevent any unforeseen circumstances occurred. We will also like to provide a function of statistics that shows how likely caregivers around the world going use the system, what are the features that need to be added more to the system to persuade more caregivers to use the system.

Appendix

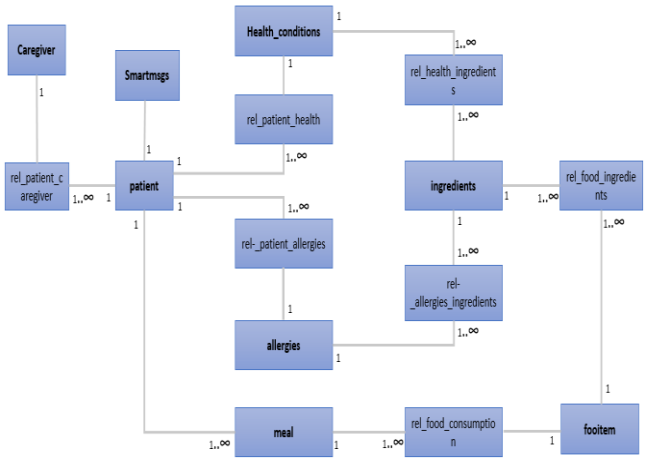
Appendix A: The relationship between MySQL, RESTful web service and Android application



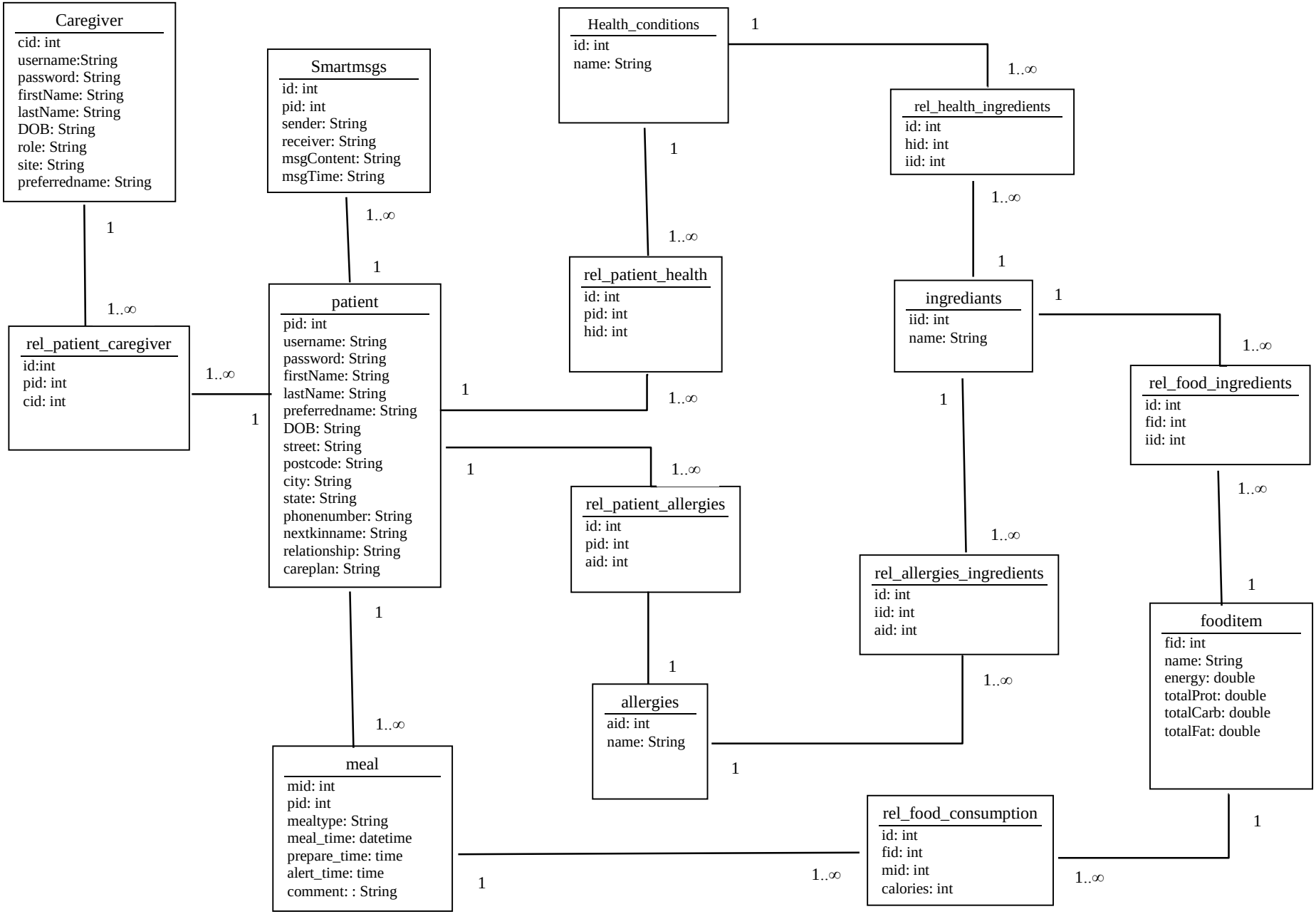
Appendix B: Use-Case Diagram



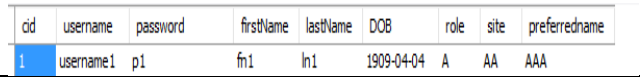
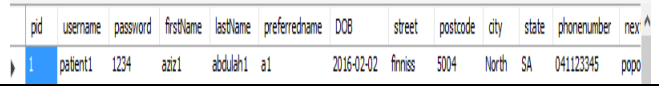
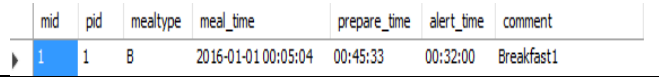
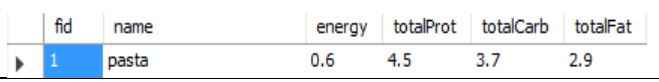
Appendix C: Entity Relationship Diagram



Appendix D: UML Diagram



APPENDIX E: Description of Web Service Functions

Web Service Name	Description	Input	Output
/register	Register a new caregiver.	-	
/login	Login to caregiver application system.	/{username}/{password}	
/addpatient	Add a new patient's detail.	-	
/addmeal	Add a new meal's detail.	-	
/addfood	Add a new food item's detail.	-	
/editpatient	Edit a patient's detail.	-	Before: {"name":"abc0","aid":"0000"},{"name":"abc1","aid":"0001"},{"name":"abc2","aid":"0002"} After: {"name":"Chris","aid":"0000"},{"name":"Jack","aid":"0001"},{"name":"Sam","aid":"0002"}
/changepassword	Change caregiver 's application password.	-	Before: {"password":"p5"} After: {"password":"password123"}
/updateConsumption	Modify patient's food consumption detail.	-	Before: {mid=1,foods:[{fid=1,calories=100},{fid=2,calories=200},{fid=3,calories=300},{fid=4,calories=400}]} After: {mid=1,foods:[{fid=1,calories=3456},{fid=2,calories=278},{fid=3,calories=589},{fid=1,calories=4098}]}
/editmeal	Modify meal's detail.		Before: {mid=2,mealDetail:[{pid=12,mealtype='B', meal_time=2016-01-01 00:05:04, prepare_time=00:45:33, alert_time=00:32:00, comment=Breakfast1}]} After: {mid=2,mealDetail:[{pid=12,mealtype='L', meal_time=2016-06-05 01:04:48, prepare_time=00:45:33, alert_time=00:32:00, comment=Lunch2}]}
/getHealthConditions	Retrieve and view the detail of current health condition.	-	{"HealthConditions":[{"name":"Mild","id":1},{"name":"Intermediate","id":2},{"name":"Bad","id":3},{"name":"Worst","id":4}]}
/getAllergies	Retrieve and view the detail of current allergies' name.	-	{"allergies":[{"name":"sss","aid":1},{"name":"gag","aid":2},{"name":"asdsd","aid":3},{"name":"allergies45","aid":4}]}

/getPatients	Retrieve and view the caregiver's information about how many patients they have.	/{cid}	{"Patients":[{"lastName":"abdulah2","city":"North","kin_phone":"03242341","postcode":"5004","phonenumber":"02342342","kin_state":"SA","pid":2,"preferredname":"a2","nextkinname":"papa","kin_city":"North","firstName":"aziz2","password":"12345","DOB":"2016-03-01","street":"finniss","kin_street":"finniss","state":"SA","kin_postcode":"5004","relationship":"mother","username":"patient2"}]}
/getpatient	Retrieve and view the selected patient's information.	/{pid}	{"livingstatus":"single","allergies":[],"lastName":"abdulah3","city":"North","kin_phone":"03242341","careplan":"up","postcode":"5004","phonenumber":"02342342","kin_state":"SA","pid":3,"preferredname":"a3","nextkinname":"pipi","kin_city":"North","firstName":"aziz3","password":"123444","DOB":"2016-03-04","street":"finniss","MMSE":"sdasdasd","kin_street":"finniss","healthconditions":[],"state":"SA","kin_postcode":"5004","relationship":"mother","username":"patient3"}
/deletepatient	Remove the selected patient's information.	/{pid}	-
/getFoodsList/	Retrieve and view all the food items' detail.	/{pid}	{"foods":[{"fid":"1","name":"pasta"},{"fid":"2","name":"hamburger"},{"fid":"3","name":"spegrahetti"},{"fid":"4","name":"YummyFiendChickenRice"},{"fid":"5","name":"HaiHanWon tun mI"},{"fid":"6","name":"HaiHanWon tun mI"}]}
/deleteMeal	Remove the selected meal's information.	/{mid}	-
/futuremeals	Retrieve and view the future meal. Future meal is indicating the date and the time is greater than current time. Just like "meal_time > now()".	/{pid}	{"meals":[{"date":"2016-07-01 00:05:02.0","mid":"2","type":"L"}]}
/getMeal	Retrieve and view the selected meal's detail.	/{mid}	{"alert_time":"00:32:02","foods":[{"fid":"3","name":"spegrahetti"}],"meal_type":"D","meal_date_time":"2016-01-01 00:05:45.0","prepare_time":"00:45:35","comment":"Lunch2"}
/getMealsTimes/	Retrieve and view all the patient's meal that will be served today. Just like "DATE(meal_time) = CURRENT_DATE()".	/{pid}	{"meals":[{"meal_type":"L","meal_date_time":"2016-06-12 20:45:45.0","mid":"8"}]}
/getMealDetails	Retrieve and view the selected meal's detail in the food item list.	/{mid}	{"foods":[{"fid":"2","name":"hamburger"},{"fid":"4","name":"YummyFiendChickenRice"}],"meal_type":"L","prepare_time":"00:45:32","comment":"Lunch1"}
/getIngredients	Retrieve and view all the current ingredients' detail.	-	{"ingredients":[{"iid":1,"name":"ingredient1"},{"iid":2,"name":"ingredient2"},{"iid":3,"name":"ingredient3"},{"iid":4,"name":"ingredient4"},{"iid":5,"name":"ingredient5"}]}
/getfood	Retrieve and view the selected food's detail	/{fid}	{"fid":2,"totalFat":"7.1","name":"hamburger","ingredients":[],"totalCarb":"3.1","totalProt":"3.6","energy":"1.4"}
/getfoodfrommeals	Retrieve and view the available food item in the meal list.	/{mid}	{"foods":[{"fid":"2","name":"hamburger"},{"fid":"4","name":"YummyFiendChickenRice"}]}

/getConsumption	Retrieve and view the consumption's detail and specify all the date that the meals consumed.	/{year}/{month}/{pid}	{"consumptions":[{"date":"8","consumption":null}]}
/getNewsFeed	Retrieve and view all the latest new feeds. The precedence of the content will be based on the latest date.	-	{"news":[{"receiver":"receiver2","sender":"sender2","fullName":"aziz2 abdulah2","id":"2","message":"Remainder: Eat you lunch"},{"receiver":"receiver34","sender":"sender7","fullName":"aziz4 abdulah4","id":"6","message":"Time to exercise"},{"receiver":"receiver3","sender":"sender4","fullName":"aziz4 abdulah4","id":"5","message":"Watch TV"},{"receiver":"receiver1","sender":"sender3","fullName":"aziz4 abdulah4","id":"3","message":"Remainder: Wash your dishes"},{"receiver":"receiver1","sender":"sender1","fullName":"aziz6 abdulah6","id":"4","message":"Go to jogging"},{"receiver":"receiver1","sender":"sender1","fullName":"aziz1 abdulah1","id":"1","message":"What the fuck you want?"}]}
/getActivity	Retrieve and view the result of how many activity of consumption based on the selected date.	/{year}/{month}/{pid}	{"consumptions":[{"date":"30","consumption":"1"}]}
/getDetailMeal	View the patient's consumption of calories based on selected date and patient's ID. If the meal type is same, sum the calories in one record, else show different record of the consumption.	/{year}/{month}/{position}/{pid}	{"consumptions":[{"mealtype":"B","consumption":"2567"},{"mealtype":"D","consumption":"7896"}]}

APPENDIX F:

The difference between JSON and SOAP API

JSON	SOAP
URL path must be HTTP/HTTPS.	The URL path normally is HTTP/HTTPS, but it can be something else.
Request is transmitted as URI Limit on how long it can be Can use input fields.	Request is transmitted as XML.
It is a style.	It is a standard.
It is powerful when JSON is returned.	It is slow when parsing method differ from browser to browser from JavaScript to XML.
WCF can consume REST.	WCF can consume SOAP
There are third-party add-on's for parsing JSON with C#, so that it may make it easier.	Unable to do so.

The difference between JSON and XML

JSON	XML
No namespaces.	Has namespace.
Objects is used in JavaScript– runtime evaluation of types.	String is used in JavaScript– may require additional parsing.
No validation system.	XML Schema Definition.
Data structure	Data structure

ACKNOWLEDGEMENT

First of all, I would like to thank my course coordinator Dr Damith Ranasinghe of the School of Computer Science at the University of Adelaide. He always points the way out in the fact whenever I needed the assistance and consistently grant this project and paper to be my own work.

Besides my course coordinator, I would like to thank my project supervisors: Dr Cheryl Pope and Ms Asangi Jayatilaka. They always allow me to widen my corresponding research area from many different kinds of perspectives and keep giving me the useful comment and encouragement. I cannot imagine that there is other better project supervisors and mentor like them for my Master study.

I would thank my fellow group mates, Abdulaziz Almoharib, Yi Chao, and Vidivalianto Shaweddy who assisted me to complete this project. Their interesting ideas, enthusiasm and hardworking made the project to become much innovatively and meaningful instead of stimulate the bored. I very appreciate what they have done for us especially when during the discussions and many works that we have been developed in the last four months.

Finally, I must express my gratitude to my parents and for keep providing me the encouragement and financial support along my study as well as accomplishing the project. Without them, the accomplishment of this project will not be done and gained nothing at last. Very appreciate what they have done to me.

REFERENCES

[1] Alzheimer's Australia (Statistics) [Online]. Available: *"https://fightdementia.org.au/about-dementia/statistics"*.

[2] Australian bureau of Statistic, "Australian social trend", abs., Canberra. 4102.0, Dec. 2012. [Online]. Available: *"http://www.abs.gov.au/AUSSTATS/abs@.nsf/Lookup/4102.0Main+Features50Dec+2012"*

[3] Australian Institute of Health and Welfare (2012). Dementia in Australia. Cat. no. AGE 70. Canberra: AIHW. [Online]. Available: *"http://www.aihw.gov.au/WorkArea/DownloadAsset.aspx?id=10737422943"*

[4] Health Direct Australia (2006, Feb.) Dementia statistic. [Online]. Available: *"http://www.healthdirect.gov.au/dementia-statistics"*

[5] Australian Institute of Health and Welfare (2012) Dementia in Australia. [Online]. Available: *"http://www.aihw.gov.au/dementia/"*

[6] Salesforce.com, inc., "The internet of things-building smart machines", Australia Pty Ltd. ABN 91 109 182 989, Sdyney NSW. 2016. [Online]. Available: *"http://www.salesforce.com/au/crm/internet-of-things/?d=7010M0000002IH5A&nc=70130000000f2JI&DCMP=KNC-Google&gclid=CjwKEAjwg6W6BRDn6v__7vzN9QkSJAC9l9C3tkVyhRN5pL71xO_vz8pVJs8THeHxHOyEcYowk75LXRoCMaHw_wcB"*

[7] Kopetz, Hermann. "Internet of things." Real-time systems. Springer US, 2011. 307-323.

[8] Tutorialspoint: Learn web services. "Why web services?". [Online]. Available: *"http://www.tutorialspoint.com/webservices/why_web_services.htm"*

[9] Abdulaziz Almoharib (June, 2016). *Web service and android development*. "Why choose RESTful web service".

[10] Neil Simpkins. (2008). T320 E-business technologies: foundations and practice. "Block 3 Part 1 Activity 5: Implementing a simple web service". [Online]. Available: *"http://www.eclipse.org/webtools/community/education/web/t320/Implementing_a_Simple_Web_Service.pdf"*

[11] Oracle. 2013. The Java EE 6 tutorial. "Creating a RESTful root resource class". P. III, chap 20. [Online]. Available: *"http://docs.oracle.com/javaee/6/tutorial/doc/gilik.html#ginpw"*

[12] IBM, "About MIME types". [Online]. Available: *"https://www.ibm.com/support/knowledgecenter/en/SSNW2F_4.5.1/com.ibm.p8.doc/ce_help/docsandfolders/df_about_mime_types.htm?view=embed"*

[13] Oracle. 2013. The Java EE 6 tutorial. "What are web services". P. III, chap 18. [Online]. Available: *"http://docs.oracle.com/javaee/6/tutorial/doc/giqsx.html#gkcaw"*

[14] Yi Chao (June, 2016). *Development of RESTful Web Services for iNutriCare Android Application*. "Identification of web service".

[15] Bruce Sun. (2009, June). "A multi-tier architecture for building RESTful web services," IBM Corp. [Online]. Available: *"https://www.ibm.com/developerworks/library/wa-aj-multitier/wa-aj-multitier-pdf.pdf"*

[16] S. Tamada, "RESTful Web Services API using Java and MySQL", Egglabs, 9lessons.info, Sep. 16, 2012. [Online]. Available: *"http://www.9lessons.info/2012/09/restful-web-services-api-using-java-and.html"*

[17] A. Gupta, "Difference between JSON and XML", Modern pathshala, Sep. 15, 2015. [Online]. Available: *"http://modernpathshala.com/Article/1016/difference-between-json-and-xml"*

[18] B. Doerrfeld, "Why JSON triumphed over SOAP", API university, Jan. 2, 2015. [Online]. Available: *"http://www.programmableweb.com/news/why-json-triumphed-over-soap/elsewhere-web/2015/01/02"*

[19] Team Stormpath, "The Fundamentals of REST API Design", Jan. 26, 2016. [Online]. Available: *"https://stormpath.com/blog/fundamentals-rest-api-design"*

[20] Shadyyx. (2014, Aug.). Why use XML(SOAP) when JSON so simple and easy to handle?. [Online]. Available:

"http://stackoverflow.com/questions/8323950/why-use-xmlsoap-when-json-so-simple-and-easy-to-handle"

[21] T. Bray. (2014, Mar.). The JavaScript Object Notation (JSON) Data Interchange Format. Google, Inc. [Online]. Available e-mail: tbray@textuality.com

[22] JSON API, 1st ed.. [Online]. Available: *"http://jsonapi.org/format/"*

[23] Y. M. Huang, D. F. Wu, and Q. Guo, "Build a RESTful Web service using Jersey and Apache Tomcat", IBM., Web development, Sep. 24, 2009. [Online]. Available: "*http://www.ibm.com/developerworks/library/wa-aj-tomcat/"*

[24] R. Tamada, "Android JSON Parsing Tutorial", Androidhive, Droid5 Informatics Pvt Ltd., Jan. 21, 2012. [Online]. Available: *"http://www.androidhive.info/2012/01/android-json-parsing-tutorial/"*

[25] L. Vogel, "REST with Java (JAX-RS) using Jersey – Tutorial", Vogella GmbH., Dec. 15, 2015. [Online]. Available: *"http://www.vogella.com/tutorials/REST/article.html"*

[26] A. Jayatilaka, Y. Su and D. C. Ranasinghe, "HoTAAL: home of social things meet ambient assisted living", The Auto-ID Lab, the university of Adelaide, South Australia, 2015.