



Faculty of Engineering and Technology
Master of Software Engineering (SWEN)

SWEN6307: SERVICE-ORIENTED SOFTWARE ENGINEERING
Second Semester 2016/2017

SOA Project Document for Team3 (Paltel telecommunications system)

Instructor: Dr. Nariman Ammar

Date: 28 May 2017

Team Members

Student ID	Student Name
1155124	Ibrahim Assi
1155073	Nadeem Al Kilani
1155072	Tarik Hannoun

TABLE OF CONTENTS:

Team Members.....	1
Table of contents:	2
1. Introduction	3
2. Our Project	4
3. Keywords:	4
4. Scope	4
5. Description (Paltel Enviroment):	5
6. Implementation:	8
6.1. Milestone1 (Implementation Issues):	8
6.2. Mllestone2 (Interface Definition + Api Documentation):	11
6.2.1. Web Servers (Middleware):	11
6.2.2. Web Services:	11
6.3. Mllestone3 (Actual Service Code Implementation):	13
6.3.1. Code	13
6.3.2. Implementation Process:	13
6.3.3. Test Process:	15
6.4. Mllestone4 (Client Applicaions):	16
6.4.1. Web Site	16
6.4.2. Mobile APP	18
7. Future Enhancement	19
8. Risks	20
9. Conclusion	20
10. Appendix B (GetHub Paths)	21
11. Appendix A of Web services used	21

1. INTRODUCTION

Service Oriented Architecture (SOA) is a distributed objects style of software design & Architecture where services as an objects of specific functionality are provided by application to other application via a procedure call over network using communication protocols such as SOAP or REST.

SOA includes objects applied in the logic of object-oriented style, these objects can be a data structure or legacy systems, these objects as components can be hosted on the same computer or on multiple platforms implemented in different programming languages, and communicate to each other's over a network as connectors or multiple networks, they shared data and call methods to retrieve stored data, all to achieve business object or goal. Also a client-server architecture as a base for n-tier architectures included in SOA to achieve the component distribution over several independent computers (for cross-machine and cross-language communication)

SOA implemented by a method of web server standards, in which Web services are independent of applied technology, achieved by using XML for communication as a request/response over HTTP. The request process done via XML form on client side (include details of the code that needs to be invoked), server that include services listens and accept XML request via port (open mostly on all networks) and then send results back to the client in standard XML, the request/response can implement in different programming languages.

web services have multiple communication protocols such as SOAP that help accomplish invocation over HTTP, SOAP uses its own XML-RPC request/response over HTTP (over TCP/IP) and can transport over other protocols such as SMTP

In this project as a goal, we going to understand the implementation of SOA in a real environment "Paltel Telecommunications environment", and then for best practice we going to simulate the environment by construct the required network and web servers (middleware) with 2 different technologies over public internet, then to develop and implement some web services (composite and atomic) in our own simple clients (web page and mobile app), also to used web services from other teams in the course.

In this documentation we will mention the 4 milestones done in our work including the initial architectural design we have proposed, then we will include the implementation process and the problems we faced during it, then we will show the clients that we developed and how they work, then we will talk about future enhancement and the conclusion.

2. OUR PROJECT

With the rapid improvement of the technologies, PALTEL find that it's become necessary to stratify the customer's needs, and tailor system to their customers in order to allow them to interest with many services and campaigns, for this reason Paltel decide to make a serious change on the integration environment between the legacy systems and the new required systems for their improvement plan.

What we try to do on this project is to simulate how that Paltel as a telecommunication company decided few years ago to utilize the existing technologies to build their architecture design by take a decision to build a service oriented architecture after they were using not effective technologies for years such as a Database links, which prevented them from improving their business needs and Keep pace with development.

So we introduce some of the enterprise systems like CRM and billing systems, and showed how service oriented technology (web servers) used as an integration point between Paltel systems to interact effectively and used as a secure layer to expose Paltel systems to public internet and how they integrate with BANK system.

3. KEYWORDS:

CRM System, Billing System, Legacy Systems, LAN network, DMZ network, SOAP, Web server, Portal, Mobile app, Cloud server

4. SCOPE

In this project we will try to show SOA-based solution as a simulation environment on Paltel problem domain, including hosting organizations and client applications, this done through implement a few services (composite services involved atomic services) such as get customer's information, get customer's bills, get customer's services and Add Payment, and use other team's services, and finally develop 2 clients on 2 different technology (web site and mobile app) to apply the mentioned web services.

5. DESCRIPTION (PALTEL ENVIROMENT):

PALTEL company is a big telecommunication company deals with Hundreds of thousands customers and operates millions of transactions, so that it has enterprise systems to deals with customer's bills, payments, installations, etc. They have CRM and billing system to manage the customer's fixed lines transactions, in addition to many types of "Value Added Services" and other new services, also they used to manage complex campaigns to encourage their customers to subscribe with these new services.

Of course all of these things operates by an enterprise system that all integrates with each other In order to facilitate the customer to deal with these services in terms of purchase, activate, deactivate services, installment payments, apply faults, complaints with one click either via mobile app or portal we site.

Since PALTEL allows the external customers to access their applications, they divided their networks to two networks:

1. LAN network (Local secured network), which contains:
 - a. Legacy systems (CRM and billing) with their local Databases.
 - b. Local web server
 - c. Portal and Mobile app Administration
2. DMZ network (Public network, exposed to public internet), which contains:
 - a. Paltel web site (portal)
 - b. Public web server

Public network DMZ opened for public users with least privilege that is enabled them to brows the portal, and act as a mediation secure layer include (Public web server) to allow mobile application to access local network securely. Paltel has 3-tier architecture style (backend – middleware – client), **Figure1**,

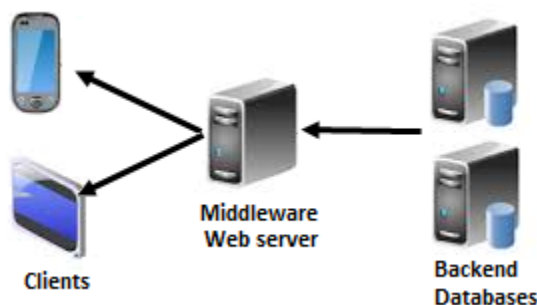


Figure1

But when deciding the network to LAN and DMZ and including the Internet as tier for mobile apps and integrated system such as the bank, we can say that we have N-tier architecture style,

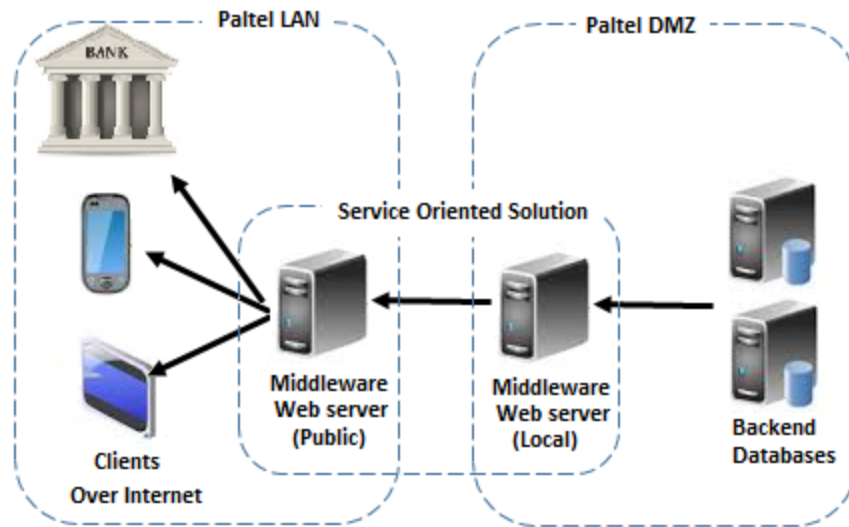


Figure2

All in order to provide the customer with an easy and effective mechanism to manage his accounts and interesting with PALTEL service in a secure way without having to visit the showroom. of course these DMZ applications integrates with the CRM and Billing system through web services.

By adopting Service Oriented Architecture, and to achieve this architecture, PALTEL implemented 2 web servers, as mentioned previously, one on LAN and the other on DMZ for security issues and used a SOAP technology as they have many legacy systems integrated with CRM and billing such as financial system, pointing, etc., so it makes deal with SOAP technology more flexible and easy.

Also by using SAOP technology PLATEL increase their benefits form the existing legacy system that were built on different technologies and increased the security level, so each team is responsible only to his job with no need to know more about the needed integration system, or to know how the other system built , what is only to get an API , also as it known by using web services we support the LESKOV principle which is apply open –close design , so each system doesn't affect with any change happened to the other system , also we can rebuilt the other system and changed the technology with any affect to the other systems.

The below figure (**Figure3**) shows PALTEL SOA architecture that describe how local network zone, consists of CRM and billing systems, interconnect with DMZ zone via web services implemented on web servers,

The customer will connect with Paltel system through the Portal web application in Paltel DMZ network or mobile application on public internet. The web server in DMZ zone contains all services related to customer requests. Customer can perform multi services such as view his information, view his bills and pay unpaid bills using VISA, these operations are complex and done on different level of integration between PALTEL organization itself (LAN, DMZ), the BANK organization and Public Internet.

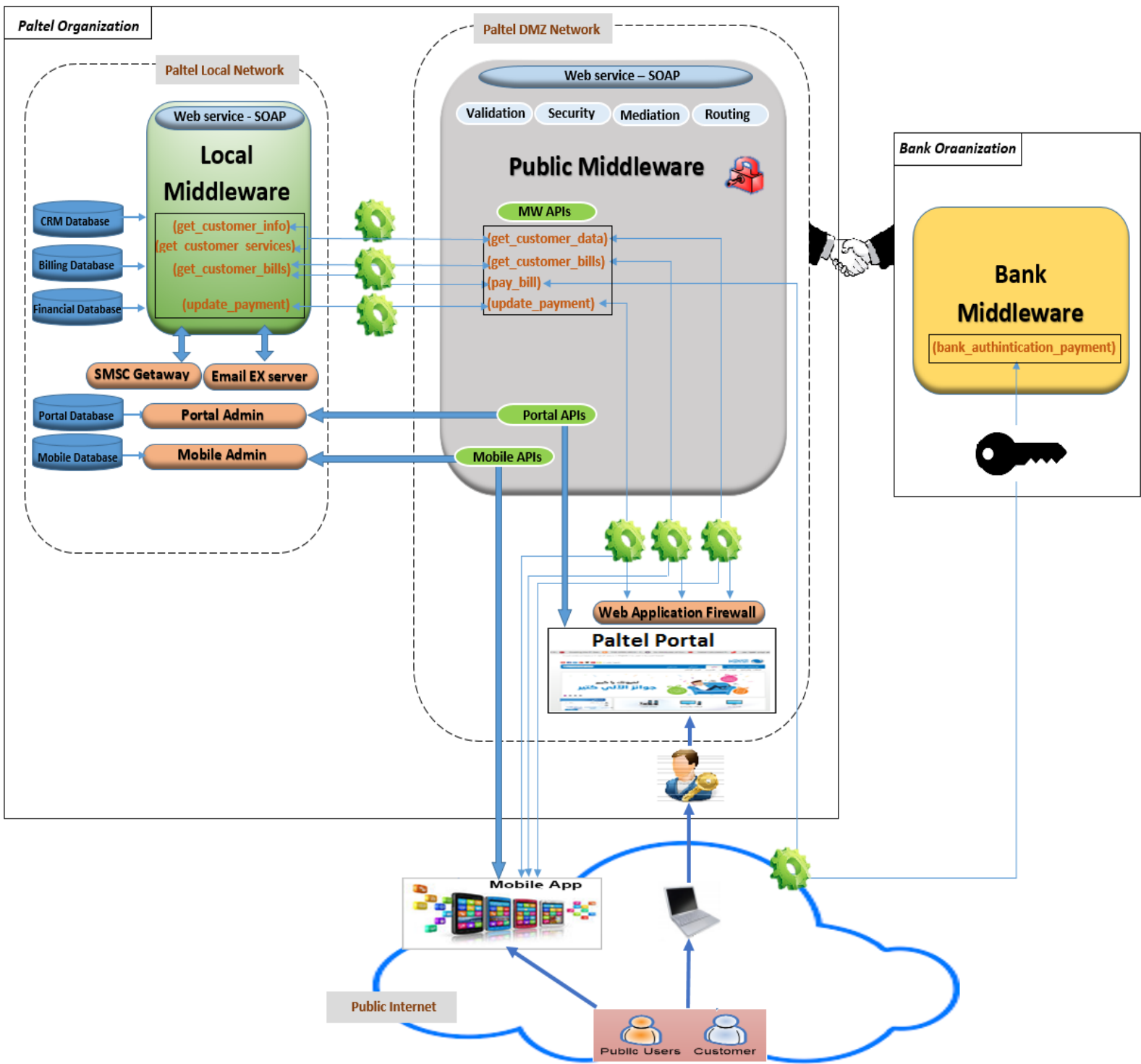


Figure 3: Simulation for Paltel Service Oriented Architecture – middleware (Web Services Architecture), includes networks LAN & DMZ

6. IMPLEMENTATION:

6.1. MILESTONE1 (IMPLEMENTATION ISSUES):

1. Properties Client applications (portal and mobile systems):

1. Portal application will be hosted under Apache Application server, using php programming language and SAOP web services.
2. Mobile application will be hosted under JBoss Application server, using Android framework, and SAOP web services.

2. Basic Operations:

1. Customer type his Tel no and password
2. authenticate the customer.
3. Client App shows the customer information like his name, address, email, etc., and his services
4. Client App shows the active services of the customers and the subscription date of each service
5. Customer request to see his bills.
6. Client App shows unpaid bills.
7. Once the customer request to pay his bill, the system will make authentication on the bank gateway and call the web service to perform transaction payment.
8. Client App check the returned value from the Bank web services in order to know the status of the payment.
9. Either of the payment was success or fail, the system will send email and SMS to the customer to inform him with transaction status
10. If the payment was success the additional step will be performed which is to register the successful payment to the customer through call web service to transfer this payment

3. Basic Operations (detailed description of the architecture operations):

Note: duplicated web services on both web servers (LAN & DMZ), because of security issues.

1. Login to the portal (Tel no and password)
2. Retrieve Customer Information automatically:
 - a. Portal main page retrieve automatically customer information via composite service “get_customer_data” on the DMS web server.

- b. composite service “get_customer_data” on the DMS web server, call 2 atomic web services from LAN web server:
 - i. 1st atomic service “get_customer_info”, retrieve basic customer information (customer name, address, mobile, email, telephone number, area code, fax number, customer classification)
 - ii. 2nd atomic service “get_customer_services”, retrieve customer active services (telephone number, service name)

3. Retrieve Billing information:

Customer request to view his bills by click a button, then portal call web service “get_customer_bills” from DMZ web server, which call a copy of the same web service on the LAN web server, this service retrieve (bill number, telephone number, bill period, bill amount, currency, due date, service name, amount).

4. Bill Payment:

- a. Customer request to pay for a specific bill via choose a bank payment
- b. Portal request to pay specific bill through web service “pay_bill” on DMZ web server, this is a composite web service that call 2 other web services:
 - i. 1st web services from LAN web server that retrieve customer bill information to send it to bank web service “get_customer_bills”
 - ii. 2nd web service, is the bank web service “bank_authentication_payment”, ‘pay_bill’ service call this service to process the payment on the bank, portal call this web service to authenticate customer bank account and send customer bill info.
- c. Bank web service “bank_authentication_payment”, return payment status.
- d. Portal update payment status as paid via web service “update_payment”, from DMZ web server, which call a copy of the same web service on the LAN web server, the LAN web service update the payment with the payment type and bank transaction number.

4. Data models:

1. CRM data model:

- a. customer information: (account number, customer name, address, mobile, email, telephone number, area code, fax number, customer classification, customer type “residential, corporate”, subscription date, line status)
- b. Payments: (bill number “unique”, account number, telephone number, payment due date, bill amount, bill currency, payment amount, payment currency, currency rate, payment type “cash, bank”, payment location name, bank trans number)
- c. Services: (account number, telephone number, service name, service status, subscription date)

2. Billing data model:

- a. Bill header: (bill number, account number, telephone number, bill period, bill amount, currency, due date)
- b. Bill summary: (bill number, account number, telephone number, service name, amount)

3. Bank data model: Bank trans: (bank name, bank trans number, amount, currency)

5. Technologies:

1. Three backend systems as a database schemes on "MySQL" to provide data:

- a. CRM (2tables customer info, services and payment)
- b. Billing (bill information)
- c. Bank (transaction)

2. Two middleware servers:

- a. First middleware exists on LAN environment, built on Apache/php technology and has web services based on SOAP technology and provide WSDL to help client to connect its services.
- b. Second middleware exists on DMZ (public to Internet) environment, built on JBoss technology and has web services based on SOAP technology, provide WSDL to help client to connect its services.

3. Two Clients, each will use web services from the second middleware DMZ (JBoss):

- a. First client is a web application built on php technology
- b. Second client is a mobile application built on android technology

6. Paradigm to be used :

1. SOAP: will be implemented on the LAN web servers.
2. SOAP: will be implemented on the DMZ web servers.

7. Data representation:

1. LAN MW services return in JSON
2. DMZ MW services return in XML

6.2. MILESTONE2 (INTERFACE DEFINITION + API DOCUMENTATION):

6.2.1. WEB SERVERS (MIDDLEWARE):

1. Web server1 (simulate Public DMZ) (Middleware1): built on JBoss, use SOAP technology and return XML format
2. Web server2 (simulate Local LAN) (Middleware2): built on Apache, use SOAP technology and return JSON format
3. Web server3 (Bank Network): as it's hard to use such real web server, we will simulate its web service on the Apache web server (middleware2)

Note: We deployed our web servers (Apache, jboss) on cloud servers (<http://paas.eapps.com/>)

6.2.2. WEB SERVICES:

1. WSDLs



Apache Server - LAN Middleware.rar



JBoss Server - DMZ Middleware.rar

2. Services Paths:

<http://yousefazem.com/SWEN6307/getCustomerInfo.php?wsdl>
<http://yousefazem.com/SWEN6307/getCustomerBill.php?wsdl>
<http://yousefazem.com/SWEN6307/getCustomerServices.php?wsdl>
<http://yousefazem.com/SWEN6307/login.php?wsdl>
<http://yousefazem.com/SWEN6307/updatePayment.php?wsdl>
<http://yousefazem.com/SWEN6307/bankAuthntectionPayment.php?wsdl>

3. Detailed description and Invocation scenarios of the existing web Services:

- a. Login: Portal (Php) or Mobile (Andriod), use web service "Login" built on Middleware1, this service call web service "Login" built on Middleware2 which return authentication as a value of "true", "false"
- b. Retrieve Customer Information:
Portal (Php) or Mobile (Android), use composite web service "GetCustomerData" built on Middleware1, this composite web service calls 2 atomic web services on Middleware2:
 - i. 1st atomic service "GetCustomerInfo", return customer information (name, address, email, etc.)
 - ii. 2nd atomic service "GetCustomerServices", return customer services such as (subscription fees, caller_id, international calls, etc.)
- c. Retrieve Bills:
Portal (Php) or Mobile (Andriod), use web service "GetCustomerBills" built on Middleware1. This web service calls atomic web service "GetCustomerBills" on Middleware2 to retrieve all customer unpaid bills information.
- d. Pay for bill:
Portal (Php) or Mobile (Android), use composite web service "PayBill" built on Middleware1 to pay for specific bill, this web service calls 3 atomic web services on Middleware 2:
 - i. 1st atomic web service "GetCustomerBills" retrieve customer bill amount.

- ii. 2nd atomic web service “BankAuthinticationPayment”, this service to pay the bill amount from “GetCustomerBills” on the bank and return payment status.
- iii. 3rd atomic web service “UpdatePayment”, to update bill status with the payment status from “BankAuthinticationPayment”.

4. Table1 shows the services and the relations between them:

<i>Is compsite</i>	<i>Web server1 (DMZ) (Middleware1) JBoss server</i>	<i>Web server2 (LAN) (Middleware2) PHP Apache server</i>
False	Login	Login
True	getCustomerData	getCustomerInfo
		getCustomerServices
False	getCustomerBills	getCustomerBills
True	payBill	getCustomerBills
		bankAuthntectionPayment
		updatePayment

Table1

5. Table2, Table3 shows Web services (Input/Output)

<i>Web server1 (DMZ) (Middleware1) JBoss server</i>		
Service	Input	Output
Login	areaCode, telNo, password	True or false
getCustomerData	areaCode,telNo	accountNumber, customerName, address, mobile, email, telephoneNumber, areaCode, faxNumber, customerClassification, customerType, subscriptionDate, lineStatus and list of services (serviceName, serviceStatus, subscriptionDate)
getCustomerBills	areaCode,telNo	accountNumber, billNumber, telephoneNumber, billPeriod, billAmount, currency, dueDate
payBill	areaCode,telNo,billNumber	Message

Table2: (Input/Output) for web service on DMZ MW

<i>Web server2 (LAN) (Middleware2) PHP Apache server</i>		
Service	Input	Output
Login	areaCode, telNo, password	True or false
getCustomerInfo	areaCode, telNo	accountNumber, customerName, address, mobile, email, telephoneNumber, areaCode, faxNumber, customerClassification, customerType, subscriptionDate, lineStatus
getCustomerServices	areaCode, telNo	serviceName, serviceStatus, subscriptionDate
getCustomerBills	areaCode, telNo, billNumber	accountNumber, billNumber, telephoneNumber, billPeriod, billAmount, currency, dueDate
bankAuthntectionPayment	billNumber,billAmount,curr ency,visaNumber	True or false
updatePayment	areaCode,telNo,billNumber	True or false

Table3: (Input/Output) for web service on LAN MW

6.3. MILESTONE3 (ACTUAL SERVICE CODE IMPLEMENTATION):

6.3.1. CODE

Code link on GitHub for LAN (Apache php) services and DMZ (jboss java) services:

https://github.com/IbrahimAssi/SWEN6307_SERVICE-ORIENTED_nadeem-tarik-ibrahim/tree/Milestone3

6.3.2. IMPLEMENTATION PROCESS:

As we mentioned on the previous milestone we have two MW one local network another on DMZ , we go on this approach for the security reason , also we applied composite practice on many layers .

Local Middleware services:

- 6 services developed using on PHP, and deployed on Apache on local MW, we succeed to deployed it live, so its accessible from anywhere.
- We used Apache with My SQL DB, so the application is fully implemented **with no static data**, all transactions read/write on MySql DB, see the published services on the URLs below.
- We already test all of the using SOAP-UI and wsdler plugin tool on the chrome, we recommend anyone to use it since it very easy for the testing.

<https://chrome.google.com/webstore/detail/wizardler/oebpmncolmhiapingjaagmapififiakb>

Service name with URL	Functions
http://yousefazem.com/SWEN6307/getCustomerInfo.php?wsdl	Returned customer information based on Tel number
http://yousefazem.com/SWEN6307/getCustomerBill.php?wsdl	Returned all customer's bills , also it designed to return the information of one bill if the bill no is passed , we implemented this scenario for the payment in order to prevent the hacker to change the amount of the bill No
http://yousefazem.com/SWEN6307/getCustomerServices.php?wsdl	Returned customer's services
http://yousefazem.com/SWEN6307/login.php?wsdl	Check authentication if the logged in user
http://yousefazem.com/SWEN6307/updatePayment.php?wsdl	Update the status of the payment once the user pay his bill
http://yousefazem.com/SWEN6307/bankAuthntectionPayment.php?wsdl	Check if the customer's authenticated to pay his bill , since not all customer allow to pay since one of them may could misuse this function

Table 4: Description of web service on LAN MW

2. DMZ Middleware services:

- 4 services developed using on java, and deployed on jboss on DMZ MW, we succeed to deployed it live, so its accessible from anywhere.
- Services is fully implemented with no static data, all transactions read/write via the local MW web services as clarified in previous milestone and in the below table.

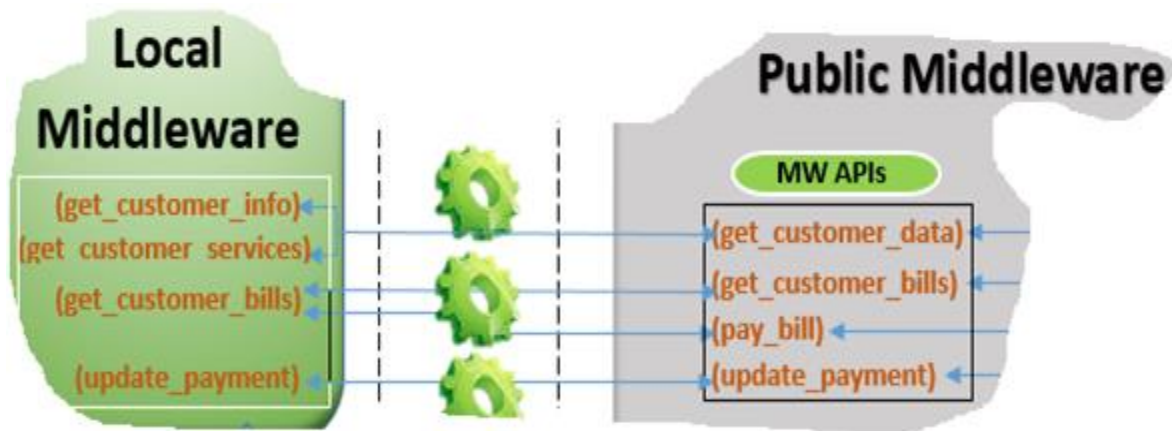


Figure4 : snapshot of Paltel architecture shows which services on Local MW called by DMZ services.

Services on DMZ MW (jboss - java)	Services on Local MW (Apache – php)
http://team3jboss.njs.jelastic.vps-host.net/team3/getCustomerData?wsdl	http://yousefazem.com/SWEN6307/getCustomerInfo.php?wsdl
	http://yousefazem.com/SWEN6307/getCustomerServices.php?wsdl
http://team3jboss.njs.jelastic.vps-host.net/team3/getCustomerBills?wsdl	http://yousefazem.com/SWEN6307/getCustomerBill.php?wsdl
http://team3jboss.njs.jelastic.vps-host.net/team3/login?wsdl	http://yousefazem.com/SWEN6307/login.php?wsdl
http://team3jboss.njs.jelastic.vps-host.net/team3/payBill?wsdl	http://yousefazem.com/SWEN6307/updatePayment.php?wsdl
	http://yousefazem.com/SWEN6307/bankAuthntectionPayment.php?wsdl

Table5: shows which services on Local MW called by DMZ services

- We faced some obstacles at the beginning regarding the hosting server, as the jboss server exists in Birzeit university (Its Run pending on localhost, it should be run with

pending server IP) because the Apache server hosted on public internet and this required access for both servers to communicate.

- d. Later on we found a cloud server solution to host our DMZ middleware server, and the work published successfully.

6.3.3. TEST PROCESS:

The test will be on the DMZ web services as they are the ones that will be used by the client applications (web site and mobile app), please see the following steps for testing:

To test the web service go to this Test Page (<http://team3jboss.njs.jelastic.vps-host.net/team3/>)

Or see the following:

- Web Service List:

1. <http://team3jboss.njs.jelastic.vps-host.net/team3/login?wsdl>
2. <http://team3jboss.njs.jelastic.vps-host.net/team3/getCustomerData?wsdl>
3. <http://team3jboss.njs.jelastic.vps-host.net/team3/getCustomerBills?wsdl>
4. <http://team3jboss.njs.jelastic.vps-host.net/team3/payBill?wsdl>

- Values for Test: (3 Tel Numbers)

Area code	Telephone Number	Password
09	2345788	ibrahim123
02	2810202	Tarik123
09	2345767	nadeem123

- Web Site Link to Test web service:
<http://wsdlbrowser.com/soapclient>

6.4. MILESTONE4 (CLIENT APPLICATIONS):

We have developed 2 clients (web site and Mobile app)

1. Web site (<http://env-3937847.njs.jelastic.vps-host.net/portal/public/index.php/auth/login>)
2. Mobile app, you can download it from GitHub (https://github.com/IbrahimAssi/SWEN6307_SERVICE-ORIENTED_nadeem-tarik-ibrahim/tree/Milestone4)
3. We used 3 web services from 3 different teams (Team4, Team5, Team7), in addition to our services (6 services)
4. We deployed our web servers (Apache, jboss) on cloud servers (<http://paas.eapps.com/>)
5. Login credentials for both clients:

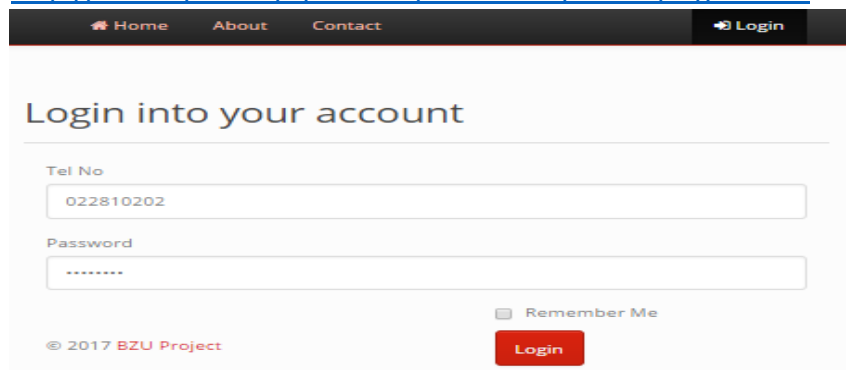
Tel No.	Password
092345788	ibrahim123
022810202	Tarik123
092345767	nadeem123

6.4.1. WEB SITE

(<http://env-3937847.njs.jelastic.vps-host.net/portal/public/index.php/auth/login>)

1. Used web services:
 - a. from **Team4** (university web service)
 - b. from Team7 (county information weather, currency, etc.)
 - c. Our web services (customer info, customer services, customer bills, pay bill, bank authentication)
2. Steps to use the site:
 - a. Login using the provided credentials: use our login web service

"<http://team3jboss.njs.jelastic.vps-host.net/team3/login?wsdl>"



Home About Contact Login

Login into your account

Tel No
022810202

Password
.....

☐ Remember Me

© 2017 BZU Project Login

b. The below is site main page after login, it shows the use of all other web services

Go to frontend

Logout

Service Team 7

31.8354533

35.6674354

Get

Country: Jordan
ExchangeRate: 1
Description: clear sky
Humidity: 100
Temp: 285.514
Temp min: 285.514
Temp max: 285.514

Service from Team7:
get country info,
weather status,
Exchange rate,,,,etc

Customer Info

10 Show elements

Search:

Account Number	Customer Name	Customer Classification	Customer Type	Address	Balance
176677	Tarik Hannoun	Golden	Residential	Ramallah	0

Showing 1 do 1 of 1 elements

Start Prev 1 Next Last

Customer Services

10 Show elements

Search:

Service Name	Service Status	Subscription Date
Follow Me	Active	2016-12-01
Internation 50 minutes Free Calls	Active	2016-11-01
Mobile 100 minutes Free calls	Active	2017-02-01

Showing 1 do 3 of 3 elements

Start Prev 1 Next Last

Customer Bills

10 Show elements

Search:

Bill Number	Bill Amount	Bill Period	Currency	Due Date	Payment Status
2211	118	022017	NIS	2017-05-10	Paid
2222	78	012017	NIS	2017-02-10	Paid
2233	56	122016	NIS	2017-01-10	Pay

Showing 1 do 5 of 5 elements

Start Prev 1 Next Last

Service Team 4

10 Show elements

Search:

ID	Title	Description	Branch	Mingrade	Program Type	University
PPU2	Computer Engennering		scientific	80	master	PPU
PPU3	Biomedical Engineering		scientific	77	bachelor	PPU
PPU4	Information System		literature	70	bachelor	PPU

Showing 1 do 6 of 6 elements

Start Prev 1 Next Last

Our web service: get customer info

Our web service: get customer telephone services

Our web service: get customer bills

Our web service: pay for bill

6.4.2. MOBILE APP

1. Used web services:

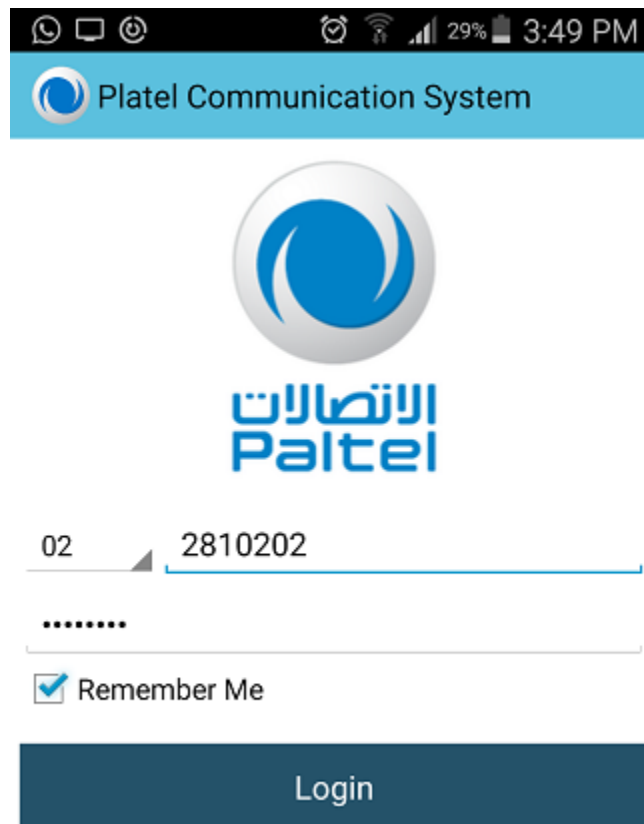
- a. from **Team5** (add payment) payment information.
- b. from **Team7** (county information weather, currency, etc)
- c. Our web services (customer info, customer services, customer bills, pay bill, bank authentication)

2. Steps to use the site

- a. Login using one of the provided credentials

Tel No.	Password
092345788	ibrahim123
022810202	Tarik123
092345767	nadeem123

- b. we use our login web service "<http://team3iboss.njs.jelastic.vps-host.net/team3/login?wsdl>"



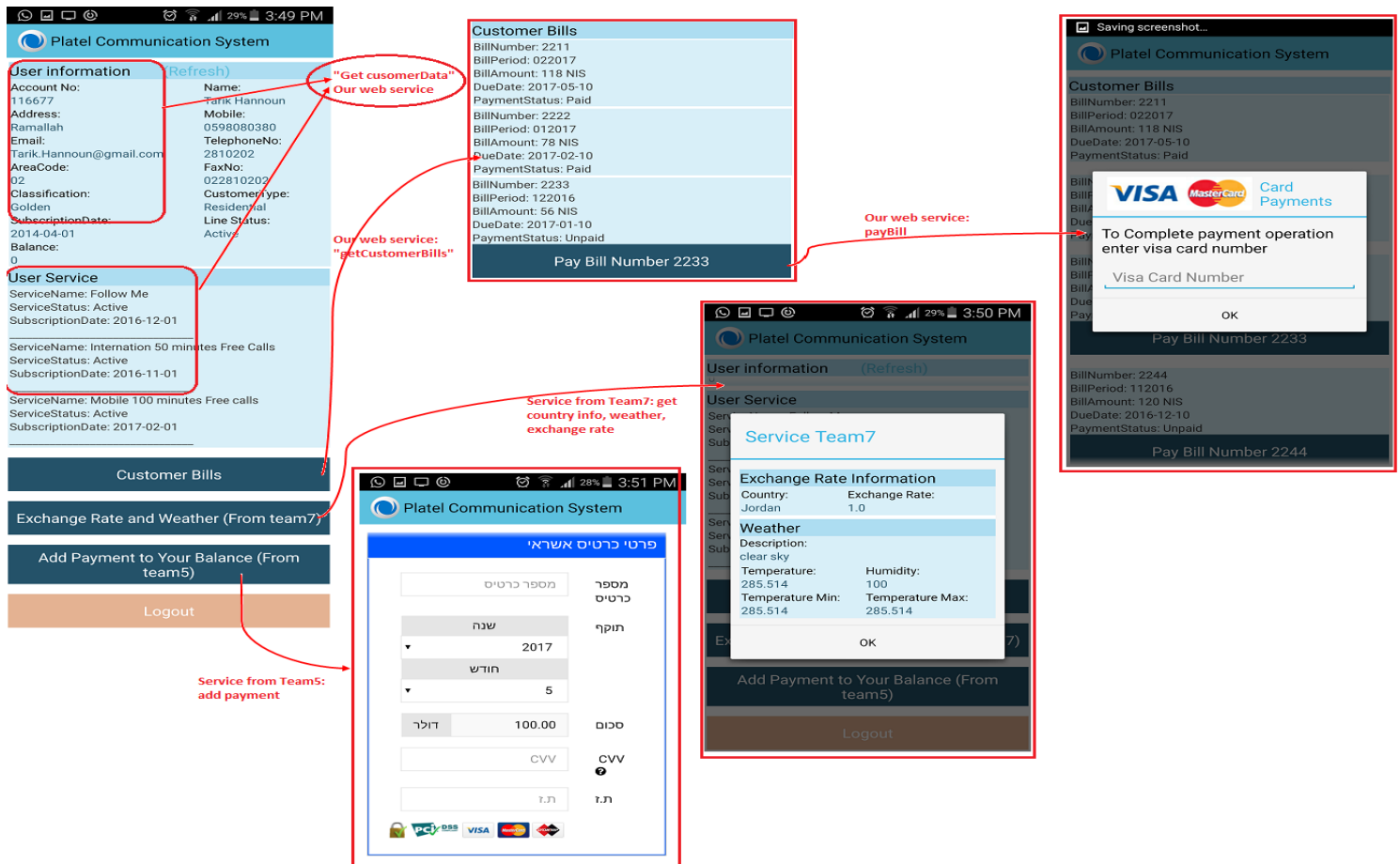
02 2810202

.....

☒ Remember Me

Login

c. Below screen shots, shows the process through mobile pages and the used web services



d. Use the following credit card info after click the button an “add payment to your balance”, for the web service from Team5

Credit card number referred to as מספר כרטיס	4580000000000000
Expiration date referred to as - תוקף	5-2017
CVV:	123
Id number referred to as - ת.ז	00000

e. After adding successful payment through Team5 web service, you will notice that the customer balance in “Customer Info” will change

f. When click pay bill, the Visa box will appear, then put any credit number will work and give status “Paid”

7. FUTURE ENHANCEMENT

Our suggestions for future enhancement:

1. As a creative thinking, we may use a Blackboard Architecture style to make a pool of web services, where the legacy systems and external vendors may provide their web services and added them dynamically to the blackboard pool as a repository for web services, then any component or client need specific web service may be looking for it on this blackboard via access request controlled by a controller.
2. Add a security layer between the clients and the DMZ middleware, this may act as a service buss or Web Application firewall
3. Add extra web services to deal with email exchange server
4. Use specific web service through specific tool to deal with Mobile telecommunication companies to send SMS and MMS messages.
5. Include push notification service from Google Android, Apple IOS and other OSs

8. RISKS

In our work we faced some risks and problems:

1. It was not easy to find 2 servers for the 2 middlewares, as this was our main idea to have 2 middlewares for security purposes, to prevent devices on the internet from direct access to Paltel LAN.
we solved the issue by working with a cloud servers provider and implement the jboss server on it.
2. Not to simulate the exact Paltel environment as it's a confidential environment and cannot be displayed to public, we solved this issue by suggested a new architecture that solve the same problem that telecommunication companies seek to solve.

9. CONCLUSION

We as a team of this project, on one hand, we already have a good experience in applying distributed object architectures through SOA, using different protocols and programming. On the other hand, from the previous work and through our completion of all the project milestones, we can say the truth that we really benefited from the scientific and thoughtful way of applying SOA environment, which is a value added, refines and correctness to our practical experience.

10. APPENDIX B (GETHUB PATHS)

Repository (SWEN6307_SERVICE-ORINTED_nadeem-tarik-ibrahim)

Paths:

1- Milestone1 (architecture (assignment 2)):

link: [https://github.com/IbrahimAssi/SWEN6307_SERVICE-ORIENTED_nadeem-tarik-ibrahim/tree/Milestone1\(assignment-2\)](https://github.com/IbrahimAssi/SWEN6307_SERVICE-ORIENTED_nadeem-tarik-ibrahim/tree/Milestone1(assignment-2))

2- Milestone2 (interface definition+api documentation) :

link: https://github.com/IbrahimAssi/SWEN6307_SERVICE-ORIENTED_nadeem-tarik-ibrahim/tree/Milestone2

3- Milestone3 (actual service code implementation) :

link: https://github.com/IbrahimAssi/SWEN6307_SERVICE-ORIENTED_nadeem-tarik-ibrahim/tree/Milestone3

4- Milestone4 (Clients) link: (https://github.com/IbrahimAssi/SWEN6307_SERVICE-ORIENTED_nadeem-tarik-ibrahim/tree/Milestone4)

11. APPENDIX A OF WEB SERVICES USED

WS-01 Get Customer Information :

This process returns all information related to the customer

WSDL URL: <http://...../.... /... /getCustomerinfo?wsdl>

Inputs

Name	Data Type	Comment
areaCode	String	Like 09 , 02
TelNo	String	Like 2810202 (without area code)

Outputs:

The output will be Text XML object

Name	Data Type	Comment
Account_number	String	This account used as umbrella to hold many tel number related to the same customer
Customer_name	String	

Address	String	
Mobile	String	
Email	String	
Telephone_number	String	
Area_code	String	
Fax_number	String	
Customer_classification	String	Platinum ,Golden ,Sliver,...ect.
Customer_type	String	Residential, Corporate
Subscription_date	String	
Line_status	String	Active or inactive

WS-02 Get Customer services :

This process returns all active services that is installed on customer's Tel number

WSDL URL: <http://...../.../.../getCustomerServices?wsdl>

Inputs

Name	Data Type	Comment
areaCode	String	Like 09 , 02
TelNo	String	Like 2810202 (without area code)

Outputs:

The output will be Text XML object

Name	Data Type	Comment
Account_number	String	This account used as umbrella to hold many tel number related to the same customer
Telephone_number		
Service_name		Like caller ID ,caller ID+ , Follow me ,...etc
Service_status		Active .inactive
Subscription_date		

WS-03 Get Bill Header :

This process returns the last 6 bills of the customers with a brief information as mentioned on the table below

WSDL URL: <http://...../.../.../getBillHeader?wsdl>

Inputs

Name	Data Type	Comment
areaCode	String	Like 09 , 02
TelNo	String	Like 2810202 (without area code)

Outputs:

The output will be Text XML object

Name	Data Type	Comment
------	-----------	---------

Account_number	String	This account used as umbrella to hold many tel number related to the same customer
Bill_number	String	
Telephone_number	String	
Bill_period	String	
Bill_amount	String	
Currency	String	
Due_date	String	

WS-04 Get Bill Summary :

This process returns summary of the bills based on each bill period

WSDL URL: <http://...../.../.../getBillSummary?wsdl>

Inputs

Name	Data Type	Comment
areaCode	String	Like 09 , 02
TelNo	String	Like 2810202 (without area code)
Bill_period	String	Like 072017

Outputs:

The output will be Text XML object

Name	Data Type	Comment
Account_number	String	This account used as umbrella to hold many tel number related to the same customer
Bill_number	String	
Telephone_number	String	
Service_name	String	
Amount	String	

WS-05 Pay Bill :

This process is responsible to add payment to the bank

WSDL URL: <http://...../.../.../payBill?wsdl>

Inputs

Name	Data Type	Comment
areaCode	String	Like 09 , 02
TelNo	String	Like 2810202 (without area code)

Outputs:

The output will be Text XML object

Name	Data Type	Comment
------	-----------	---------

Account_number	String	This account used as umbrella to hold many tel number related to the same customer
Bill_number	String	
Telephone_number	String	
Amount	String	
Currency		