

Addressing Review Recommendations

Version 1.0

Table of Contents

1. Introduction	3
1.1 Purpose of this document	3
1.2 Conventions	3
1.3 Document organization	3
1.4 Intended Audience	3
2. Requirements	3
3. Architecture and Design	4
4. Implementation	6
5. Verification and Validation	6
6. Management	7

1. Introduction

1.1 Purpose of this document

This document tries to answer the concerns and doubts expressed by the SCORE evaluators after the first round of selection. Secondly it presents the improvements made in the DPS application after the initial submission and highlights certain areas which otherwise are not visible through rest of the documents.

1.2 Conventions

Texts in **green** are the collection of valuable comments by three evaluators from SCORE after first round of selection.

Texts in **blue** are our responses and clarification to those comments.

1.3 Document organization

The document is organized as follows:

- Section 1, *Introduction*, describes contents and conventions used in this document
- Section 2, *Requirements*
- Section 3, *Architecture and Design*,
- Section 4, *Implementation*,
- Section 5, *Verification and Validation*,
- Section 6, *Management*,

1.4 Intended Audience

Intended Audience of this document is:

- Score Supervisor and Evaluators

2. Requirements

An important asset of the work is that the functionalities have been limited to the point of defining a realistic project considering the capacities and limitations they had. This is very important in real-life projects in order to comply with commitments.

Major: However, they have reduced some functionalities required, like the anonymous vote, with no agreement of the stakeholder (only a misunderstanding is mentioned). This must be done before a product is finally delivered.

We agree, in real life project implementation, mainly in consultancy business, the software is built based on the agreement between the client and the consultancy provider organisation. Even we were supposed to develop the product based upon our agreement of functionalities with the customer.

We carried out this project in a our master's course curriculam (distributed software development) with a deadline of only ten weeks. Whereas according to SCORE suggestion, "*Expectations will be adjusted to the duration. Duration of not less than 5 months and not more than 8 months is suggested*". Even our master's grade in this subject was very much dependent on how we develop (I,e experience flavor on full SDLC starting from requirements gathering till testing and roll-out). Due to bounded time-line of the curriculam, we were eager to produce some live implementation based upon some initial agreed upon requirements. However, the product now offers anonymous voting feature also.

MÄ@dium: It is worth mentioning that the interaction for requirement Elicitation with the stakeholder was limited. Although they mention this as a limitation due to time, it is an important point in real projects and, in particular, in this competition. They adopted an alternative work method defining other "customers" who helped in defining points correctly, but it is the stakeholder acting as a customer who should approve these requirements. If this project was to reach a second phase, it is important to ask for a document with a more complete definition of the functionalities the product will offer.

Yes, We feel we made a mistake in prioritizing among the requirements without involving the customer enough. We can honestly say this is our most important learning from the project. Again we had a burden of bounded timeline of the course curriculum. We would at least like to show the final list of included features. We present the list to our customer on 19-Feb-2008. Please find this list of implemented functionality our product offers in the Requirement Document.

Minor: There should be a better classification between non functional and functional requirements (e.g., Database ones)

This is more of a typo error, as we placed database requirements in functional part. We now have moved them to non functional section.

3. Architecture and Design

Following the line of the requirements, the architecture presented is simple and easy to understand, which is important in projects where future evolution is expected. Besides, the team has presented a previous document with more details with this deliverable, where the decisions taken are explained.

MÄ@dium: An impact of this architecture is that the work is based on the centralization with limited use and development over mobile devices (which was an important part of the challenge)

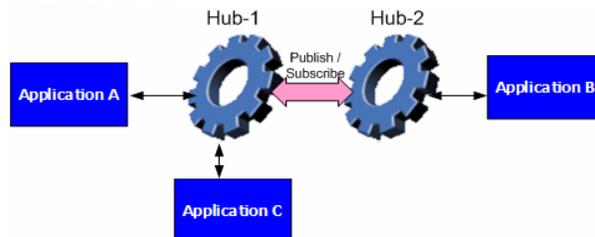
We are sorry, we understood the above remark with two different meaning and we are addressing all as below:

Our interpretation from above remark:

1. Architecture is Centralized as HUB-SPOKE architecture and therefore is of limited use
2. Limited emphasize on mobile communication part and integration with DPS

Our understanding:

1. We agree hub-spoke architecture is centralized in nature and prone to single point of failure. This can cause disintegration of communication among involved applications. We can have two types of resolution for the same.
 - The concept of segregated hub where multiple hubs are connected through publish/subscribe methodology as below. If any hub goes down, the whole system does not shut down.



- Even we can have two different instances of the same hub running in parallel and applications are equally divided among those hubs for communication. Although Fail-over mechanism or so called redundancy is a solution for that.
2. Unfortunately, we could not specify the hurdles we faced for communication with mobile device in detail in the limited 20 pages SCORE report. We had to face a lot of challenges in understanding mobile network and finally successfully integrate with DPS. General architecture for SMS communication in telecom OSS/BSS system is as below:

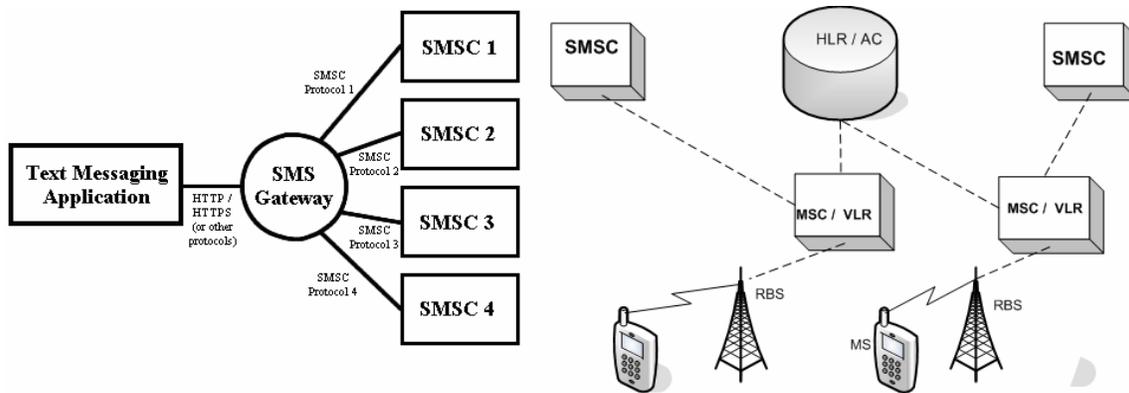


Figure 3-1: A part of Telecom OSS/BSS (operational and business support system) along with sample telecom network

Generally these SMS Gateway and SMSC (SMS Center) are of service providers. So we contacted many local service providers to get access of their SMS Gateway application, so that we can communicate to service provider's SMS Gateway from our Web-DPS application.

At this point of time, we decided that in our architecture, **Text messaging application** (Figure 3-1: A part of Telecom OSS/BSS (operational and business support system) along with sample telecom network) will be replaced by our DPS and we will be able to communicate member through SMS.

However, no service provider was interested to provide access of their internal SMS Gateway application.

Later we found out one more way of prototype development where a mobile phone (handset device) can act as a **SMSC** and a sample java service can act as **SMS Gateway**, which made us enthusiastic on the topic. Finally, approaching the same we have succeeded to communicate and further detailed description will be available at our deliverable '**DPS_Mobile_Communication.doc**'

Minor: It is possible that using SQL writing and reading on an SQL table as means of communication may be inadequate, except in cases of considering high volumes of traffic. It should be pointed out as an adoption only for prototype purpose. Especially, because the design is based on the use of a middleware. Also, as JBOSS is used, why not think of a mechanism of message queues?

Yes. We accept the same.

We have depicted different means of communication approach with web-DPS application in our design document as well as in the SCORE report. Among all the approaches we considered for integration with web-DPS application, includes JMS approach which is much more reliable because of queueing facility. Using JMS approach we can send voting response messages received from target applications (SMS Gateway and Email) to web-DPS application (source) for processing.

However, due to limitation in project timeline due to course curriculum (10 weeks schedule), we could not afford implementation of the queueing approach.

Rather we used MySQL Stored Procedure for prototype purposes and spending affordable time in implementing business logic along with technical communication between applications.

Minor: Choosing a digital signing was a priori known to be ambitious (and which goes against the initial principle of simplicity). This should not have been something that could not be attained but a "nice to have" for future versions and, therefore, the design should have been defined only to be adopted later on.

It's very much true.

Initially at project start up, we were very much optimistic and enthusiastic to implement few security features like password protected PDF generation as email attachment, digital signatures and so on.

However, although we implemented password protected PDF generation and sending as email attachment; we could not afford digital signature implementation due to bounded timeline in our course curriculum of almost ten weeks time line only.

Even we have specified in our SCORE report the perspective and placement of digital signatures that will ensure user's authentication at email server level.

Minor: in this document, the syntax of the messages in order to parse the votes/opinions/messages is not explained.

We are sorry, we could not afford to include these low level details due to number of pages limitation in SCORE report. But definitely we are incorporating *DPS_Message_Routing_Transformation.doc* that will depict the low level parsing of messages, votings and so on.

4. Implementation

The document shows that something has been really implemented and that some problems have arisen. In the presentation some real features will have to be demonstrated since some comments sound like answers from a book, but the distributed experience is still worth mentioning.

The document presents the definitions of the algorithms, in compliance with the requirements in the definition of the project. Here, only descriptions of an extremely high level are included due to the nature of the document. They seem appropriate, but it is important to describe them in detail in the second phase. As perfectly mentioned, we could not yield in details the algorithm for vote calculations. However, definitely we have introduced *DPS_Algorithms.doc* that depicts detailed architecture of capturing votes, analysis and design for algorithms.

Minor: They say they will use MySQL but then they keep mentioning PL/SQL, which is Oracle. Which have they used?

We are sorry, we misused term 'PL/SQL' which is a scripting language used in oracle.

However, we have used MySQL v5.0 for database requirement as it's an open source product.

So, we used MySQL stored procedures as API for communicating synchronously with the respective applications.

Medium: No mention to creating user manuals.

Sorry, we have not specified user manuals in our SCORE report.

But definitely this is part of our deliverable list *DPS_User_Manual.doc* and we will be producing the same.

5. Verification and Validation

Medium: The verification part presented seems limited, and time issues are mentioned. It is a well-known real-life issue and in this sense, should have been foreseen

The only weakness is in the limited verification and validation. Although it is common in real-life projects to cut on VV due to the approaching deadline for delivery, this is a practice to discourage. Therefore, I suggest to improve the work on VV in the final deliverable.

As we depicted the scenario in SCORE report, that our course curriculum of almost ten weeks only, enforced us to limited verification of the developed product.

However, although 15th January, 2009 we submitted our SCORE report but our course curriculum finished on 26th January, 2009 due to christmas vacation. So we managed to perform some more verification on the developed product till 26th January and this has come up with a separate test case document *DPS_Test_Specification.doc*. This was actually mentioned in our test plan document as a deliverable, when it was formulated.

Major: What has been presented sounds a little theoretical and the results are not clear as the vision presented is a "good test cases" one, or one to only prove that it works. No test cases are presented or the number of cases as, for example, on table 7.1. This should be stressed because it is important as a product and in order to understand results.

Although its not an excuse, still we feel, inspite of having test cases scribbled in an excel sheet, we could not specify in the SCORE report due to number of page limitation. Even when we were revising the

SCORE document for second time, we had to cut short the later part of the SCORE report to fill it in 20 pages....:-(. However, definitely we will be specifying the test cases we applied in a separate document *DPS_Test_Specification.doc*

Minor: The scope of the compatibility testing is not clear. For example, it does not reach types of mobile phones, only browsers, which have lower complexity. Does this show that the mobile part has not been very much developed?

Well, it's very true that we have specified only browser part in the compatibility testing. Unfortunately, we have missed all the obstacles we have faced in finding a suitable mobile handset that can interact with the installed driver and can interact using command set from java application.

We borrowed (from friends) and experimented with different models of mobile handsets of different brands (Nokia, Sony Ericsson and Motorola). It also happened that when we connect a mobile device with a laptop, the port assigned (like COM7 and so on) for that needs to be specified in the JAVA service. Each time we use a different handset, the assigned port also changes and we used to execute without verifying that, which used to cause frustration as JAVA code could not work.:-)

6. Management

Another strength of this project: a good team organization. Knowing the constraints, their organization into well defined roles has allowed them to carry out this project.

- The method and process followed is well presented in the paper.

Medium: However, it is doubtful how they implemented the scrum 10-15 minute-stand-up meetings, being geographically distributed. To be validated.

We are students from India, Pakistan and Iran doing our Master's degree in Europe and are participants of the course Distributed Software Development. The course is run jointly between MDH University, Vasteras, Sweden and FER University, Zagreb, Croatia. The setup is like a virtual class room between both universities, and each student can see and hear other student from the other country. Video conferencing using Microsoft Netmeeting is established. Students are provided with number of time slots where they can discuss and communicate with other students located 1500 KM apart from other student of the same project team.

Team pictures and team website. Awareness of cultural and work habit differences of team members. However, on page 3 of the report it states that the students are from India, Iran, and Pakistan, but on page 4 from Sweden and Croatia. Is this a copy and paste error?

We, the students are nationals from India, Iran and Pakistan but studying in Sweden and Croatia.