Sri Lanka Institute Of Information Technology

Comprehensive Design/Analysis Projects

Project Proposal

SMS Based Application Management Platform

Group no:

Group Members:

R.M.C.U. Senevirathne	IT10031308
L. Shashika Amali	IT10029114
H.H. Rajamanthrie	IT10029664
P.T.K Perera	IT10025604

Supervisor name: Ms.Aruni Niroshika

Date of submission: 22/02/2013

Abstract

The proposed system SMS Based Application Management Platform, allows service providers to define and build applications converging entertainment, information & communication, in a flexible, easy-to-use manner with virtually no complexities of linking different applications and provide solutions to their subscribers. The framework is going to be built for service providers to create their own pool of easy to use Telco applications for their various needs in a customized manner. According to the requirements gathered, four main SMS based applications have been identified and the proposed system allows the creation of, alert applications, voting applications, service applications and contact applications to introduce an immense number of applications to the subscribers without investing any high operational and capital expenditure. The proposed system is enriched with many features so that the client will be served with the maximum satisfaction. Some of the main functionalities other than the implementation of the four main SMS based application types are, the real time report generating, provide the clients to log on to their apps with SSO, subscriptions & un-subscription handling, filtering data which transferred from apps to its' subscribers checking for informal or any abusive contents. Outcome of this project is simplified user interfaces for creating customized SMS based applications and it will be an open platform that allows the creation of SMS applications, with a wide variety of reports (real time reporting) with a user friendly dashboard.

Contents

1. Introduction

1.1 Background

Mobile application management has been driven by the widespread adoption and use of mobile devices in business settings. There have been a very few developments in SMS based application management platforms in the present day which allow the users to browse the mobile applications and also gives the opportunity for mobile application developers to create, test and sell their own unique applications. The major drawback of the current solutions for SMS based application management platforms is application developers have to implement the coding for the application they are creating and they need to be very familiar with the technologies to be used in the development of such applications. This requires a prior knowledge about the technologies and languages used to implement an application of such kind.

The proposed SMS based application management platform allows it's user to create SMS based applications without having a prior knowledge and without knowing the underlying technologies of the application. This facility may address a wide range of customers who wish to create SMS based applications for their businesses, organizations etc. through web interfaces.

The proposed system provides an extensive range of voting, alerting, subscribing, competitions, viewing request messages, and responds on requests applications. Normal scenario for a voting application or some other SMS based application in existence is that when a client (mostly media partners who hold reality programs or other needs) contacts a service provider requesting a voting application; the service provider has to make a whole new application to each individual client separately.

With the proposed system what will be done is to build an application management platform which allows multiple Telco partners to customize the same application as per the requests of multiple clients without wasting resources or time. Each client will get a unique identification for all of their applications and a login per each application they create. CAS is used in order to facilitate clients to register and log on to the applications using open ID integration.

1.2 Research Problem

The main research problem identified is clustering and concurrent high load data handling and mongoDB is to be used to implement the solution for the mentioned problem. mongoDB is a scalable, high-performance, open source NoSQL database. Instead of storing data in tables as is done in a "classical" relational database, MongoDB stores structured data, making the integration of data in certain types of applications easier and faster. MongoDB is the most popular NoSQL database management system. The database is used by MTV Networks, Craigslist, Foursquare, UIDAI Aadhaar and many other large organizations for the purpose of clustering and concurrent high load data handling. mongoDB consist with important features like Ad hoc queries, Indexing, Replication, load balancing, file storage etc.

Integration of the **central authentication service** (CAS) with open ID integration for user interfaces also has been identified as other research problems.

Furthermore several main research questions have been acknowledged in order to produce the proposed project outcomes as per the level expected by the client & to serve the major research problem. How to handle large amount of data without affecting system performance is a major concern in the development & the suitable technologies, database systems (mongoDB) are under consideration of the research group and study of the necessary technologies has been taken place in the initial parts of the development. Also the benchmark results for other databases comparing with mongoDB are thoroughly conducted in the early study of the project. How to handle server event occurring schedules, how to simulate large number of sample data without taking longer time and how to tune database performing with large number of data are some other major concerns. Mechanism to do the data filtering is planned to be done by developing an algorithm. How to generate real time reporting for voting applications, how to authorize & authenticate multi module access and how to handle un-subscriptions for a particular application are the additional questions to be answered when the completion of the project is done.

2. Objectives of the Project

Both specific and general objects have been identified for the proposed system; SMS based application management platform and the main objectives are listed below.

2.1 Specific Objectives

Specific objects of the proposed system; SMS based application management platform may fall in to two main areas. Those are implementation of the framework for service providers to create their own pool Telco applications, clustering and concurrent high load data handling.

- Implementation of the framework for service providers to create their own pool of easy to use Telco applications
    - Implementation of alert, voting, contact & service applications.
    - Real time reporting, common registration, subscriber handling and data filtering.
    - Provide third party developers to create applications without knowing the underlying technologies.

- Clustering and concurrent high load data handling
    - Replicate data to more than two servers.
    - mongoDB shading and find indexes (Support clustering).
    - SNMP trap implementation.
    - Can analyze large amount of data. This framework can be used by any Telco company with smaller configurations.

2.2 General Objectives

Other than the specific objectives listed above there are some important general objectives the proposed system may serve after the implementation. One main general objective identified is central authentication Service with Single Sign On. It permits a user to access multiple applications while providing their credentials (such as user id and password) only once. It also allows web applications to authenticate users without gaining access to a user's security credentials, such as a password. Open ID Integration is one other general objective which allows users to be authenticated by certain co-operating sites (known as Relying Parties or RP) using a third party service, eliminating the need for webmasters to provide their own ad hoc systems and allowing users to consolidate their digital identities. OpenID authentication is now used and provided by several large websites. Providers include Google, Yahoo!, PayPal, BBC, AOL etc.

Other main identified general objectives are listed below.

- High Availability of data
- Security
- Concurrent access
- Generate Reports

3. Methodology

3.1 Project activities and work plan

SMS Based Application Management Platform is based on implementing the framework to create Telco applications, clustering data and concurrent high load data handling. To achieve the objectives of it is needed to focus on research and development with initial requirements.

The proposed system is divided in to four major modules according to the requirements gathered,

- **Voting application implementation & real time reporting**
- **Service application implementation, common registration & login with SSO**
- **Alert application implementation, admin & subscriber handling**
- **Contact application implementation & data filtering**

Voting application implementation & real time reporting module is responsible for implementing front end UIs and backend UIs of the application. Voting application shall allow service provider to create an application where subscribers can vote on the options provided by the service provider. Real time reports are generated in the system to obtain the results of the voting applications in a wide variety such as graphs, charts, data reports etc. Additionally reporting module will be used as an independent module by other main modules to generate reports relevant to those particular applications when necessary.

Service application implementation, common registration & login with SSO modules are responsible for implementing front end UIs and backend schedulers of the service application. Service/Channel application would allow service provider to create an application which would allow scheduled message sending to the subscribed subscriber base. Hourly, Daily, Weekly and Monthly message scheduling shall be possible. Registration of the clients who need to create applications using the application management platform is handled by the system; this includes login handling and user account handling. In addition Central authentication service (CAS) with open ID integration for user interfaces is integrated in this module.

Contact application implementation & data filtering module is responsible implementing front end UIs, Handle response messages of the alert application. Contact application shall allow subscribers to send messages to the application created by the service provider and shall display all the messages in a web UI allowing service provider to reply for each and every message or to send a common response message to all the messages. The content of the

SMS sent from the service provider to the subscribed subscriber is checked for informal or abusive content in the data filtering module and those are informal or abusive is filtered to be analyzed by the admin.

Alert application implementation, admin & subscriber handling modules are responsible for implementing front end UIs and handling subscription and un-subscription of the alert application, implementing admin module and handle subscriber details and operations. Data filtered in the filtering module is sent to the admin UIs to manually approve the content and send them to the application or reject the content alerting the application owner that the content contains informal or abusive content.

Below is the system high level architecture diagram of the proposed system.

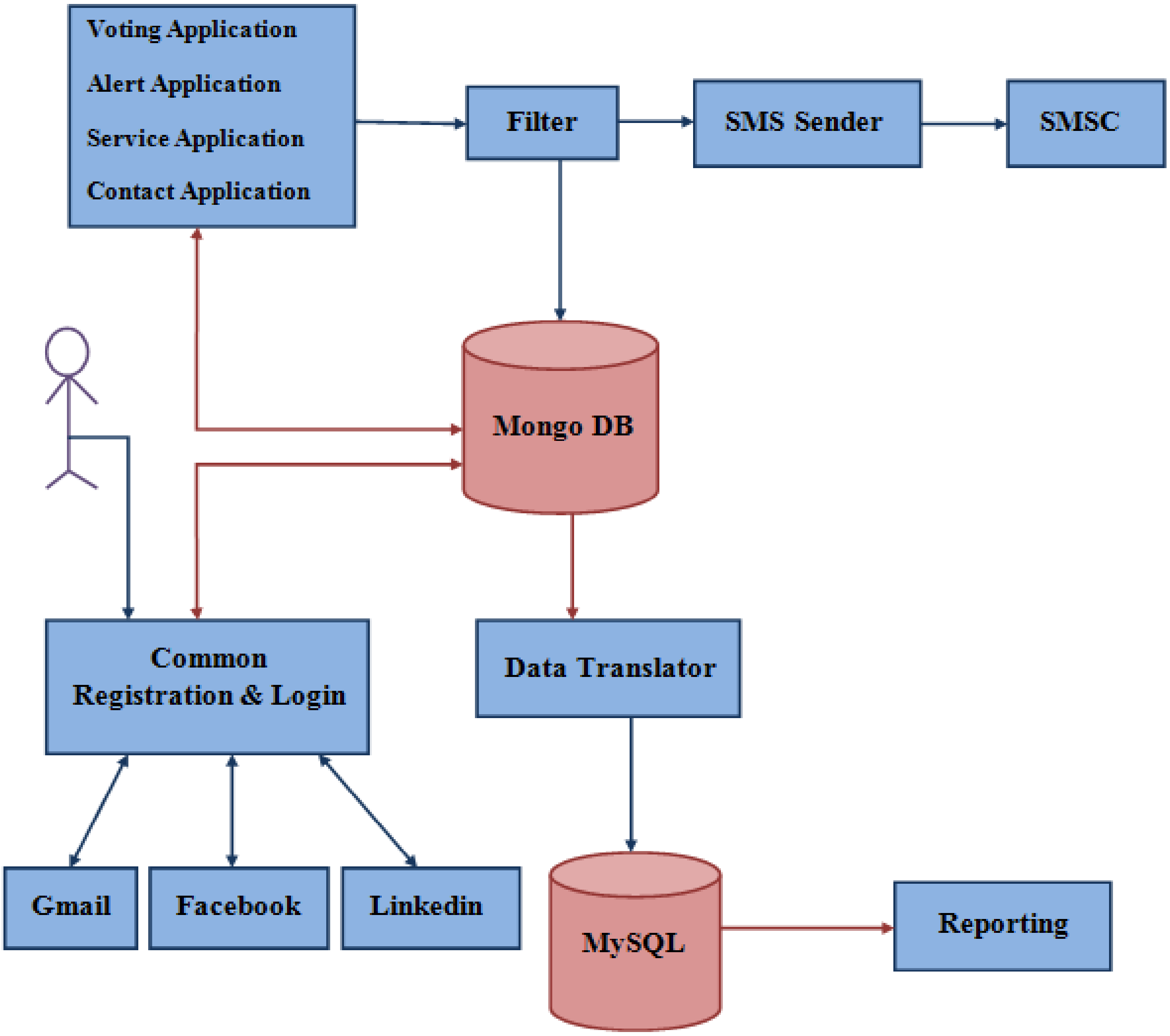


Figure 1

With the time of the development it will be a necessity to add more requirements and more developments to be done to the project. To fulfill above features **Agile Methodology** and **Scrum** sub methodology is used out of the many methodologies available today.

Scrum Methodology

Of all the agile methodologies scrum is unique because it introduced the idea of 'empirical process control'. Scrum use real world progress of a project to plan and schedule releases. Scrum is based on sprint, which is can be two weeks to give working system to customer. Four weeks for each sprint and two sprints for a release are used in this system.

Scrum Development Cycle

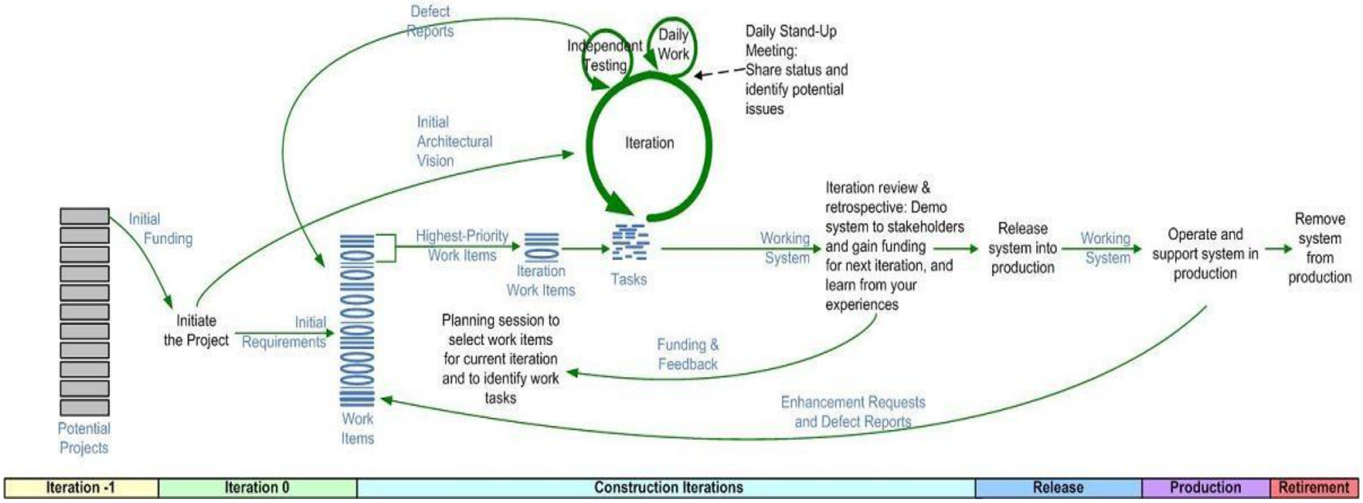

Defect Reports
Independent Testing
Daily Work
Daily Stand-Up Meeting: Share status and identify potential issues
Initial Architectural Vision
Iteration
Iteration review & retrospective: Demo system to stakeholders and gain funding for next iteration, and learn from your experiences
Initial Funding
Highest-Priority Work Items
Iteration Work Items
Tasks
Working System
Release system into production
Working System
Operate and support system in production
Remove system from production
Initiate the Project
Initial Requirements
Planning session to select work items for current iteration and to identify work tasks
Funding & Feedback
Enhancement Requests and Defect Reports
Potential Projects
Work Items
Iteration -1
Iteration 0
Construction Iterations
Release
Production
Retirement

Figure 2

Scrum Roles

- **Product Owner**
    - Responsible for communicating the vision of the product to the development team.
    - Requirement prioritizing.
    - Decide when product is ready to ship.

- **Scrum Master**
  - Give hardware and software requirements.
  - Protect the team from disruption.
  - Teach scrum to team.

- **Team**
  - Self-Organizing.
  - Cross functional.

3.2 Initial User Stories

In scrum entire system need to describe with user stories. Following are the user stories what we identified.

1. Spring Integration
2. Initial Services
3. CAS integration with Single Sign On
4. Open ID Integration
5. Configure properties
6. Initial Reports
7. Identify Drill down reports and charts
8. Identify necessary indexes
9. Spring Birt integration
10. Test cases
11. Concurrent access
12. Security
13. SSL for User interfaces
14. Schedule services
15. mongoDB sharding and clustering
16. Identify shard keys for necessary databases and collections
17. Implement filtering mechanisms
18. Spring MVC integration
19. Spring mongo integration
20. Spring quartz scheduler integration

Initial user stories are listed below. After identifying the user stories, estimation of those above stories are to be done.

User Stories																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Chamika	3	3	13	8	3	8	5	3	3	3	13	5	8	5	13	3	5	0	0	0
Shashika	3	3	13	8	3	5	8	5	3	3	13	5	5	5	13	3	5	0	0	0
Hasini	3	3	13	13	3	8	8	8	3	3	13	5	8	5	13	3	5	0	0	0
Tharindra	3	3	13	8	3	8	8	5	3	3	13	5	8	5	13	3	5	0	0	0
Final Estimation	3	3	13	8	3	8	8	5	3	3	13	5	8	5	13	3	5	0	0	0

3.3 Contribution

Voting application implementation & real time reporting - R.M.C.U. Senevirathna

This module is responsible for implementing front end UIs and backend UIs of the application. Voting application shall allow service provider to create an application where subscribers can vote on the options provided by the service provider. Multiple Telco partners can customize the same voting application as per the requests of multiple clients without wasting resources or time. By using this method, service provider do not want to make a whole new voting application to each individual client separately.
When a request is been made for a voting application, each client will get a unique short code for all the applications and a unique keyword for the voting application. Short code and the key word together is compositely unique combination. Depending on the requirements clients can customize the voting application and subscribers can vote on the options provided by the service provider. To obtain and analyze the results of a particular voting application, real time reports are generated from the system in a wide variety such as graphs, charts, data reports etc.

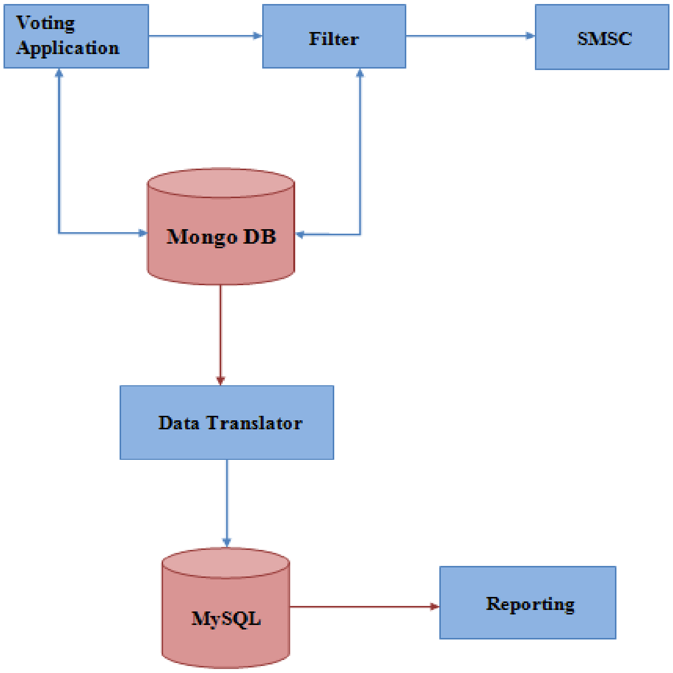


Figure 3

Service application implementation, common registration & login with SSO - L. Shashika Amali

Service application implementation, common registration & login with SSO modules are responsible for implementing front end UIs and backend schedulers of the service application. Service/Channel application would allow service provider to create an application which would allow scheduled message sending to the subscribed subscriber base. Hourly, Daily, Weekly and Monthly message scheduling shall be possible. Registration of the clients who need to create applications using the application management platform is handled by the system; this includes login handling and user account handling. In addition Central authentication service (CAS) with open ID integration for user interfaces is integrated in this module.

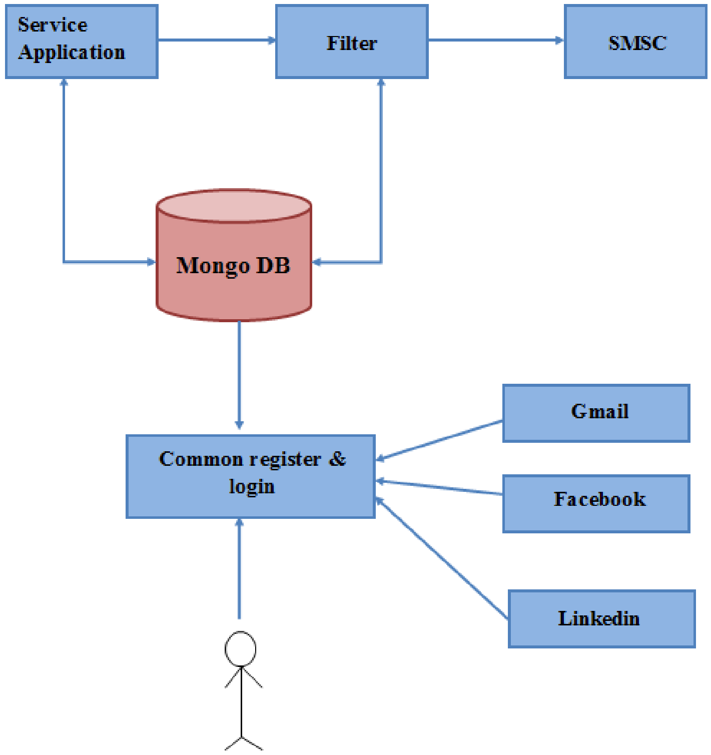


Figure 4

Alert application implementation, admin & subscriber handling - H. H. Rajamanthrie

Alert application implementation, admin & subscriber handling modules are responsible for implementing front end UIs and handling subscription and un-subscription of the alert application, implementing admin module and handle subscriber details and operations.

When a request is been made for an alert application by any client, they get to customize and generate the application through the front end UI and afterwards they can log in to this alert application. Then the owner of the application can send alerts to the subscribers through the service provider. Customers who wish to use this service can subscribe to alerts from this client as per their preference. Also the facility to unsubscribe whenever needed is possible in the proposed system.

These receiving messages will be processed according to the client and the type of alerts subscribed, by this system so the subscriptions and un-subscriptions details will be handled properly.

Data filtered in the filtering module is sent to the admin UIs from the service provider's side to manually approve the content and send them to the application or reject the content alerting the application owner that the content contains informal or abusive content. This feature is used to assure that the alerts or any sort of message transmitted through the service provider is clean as it becomes their responsibility to ensure that in the end.

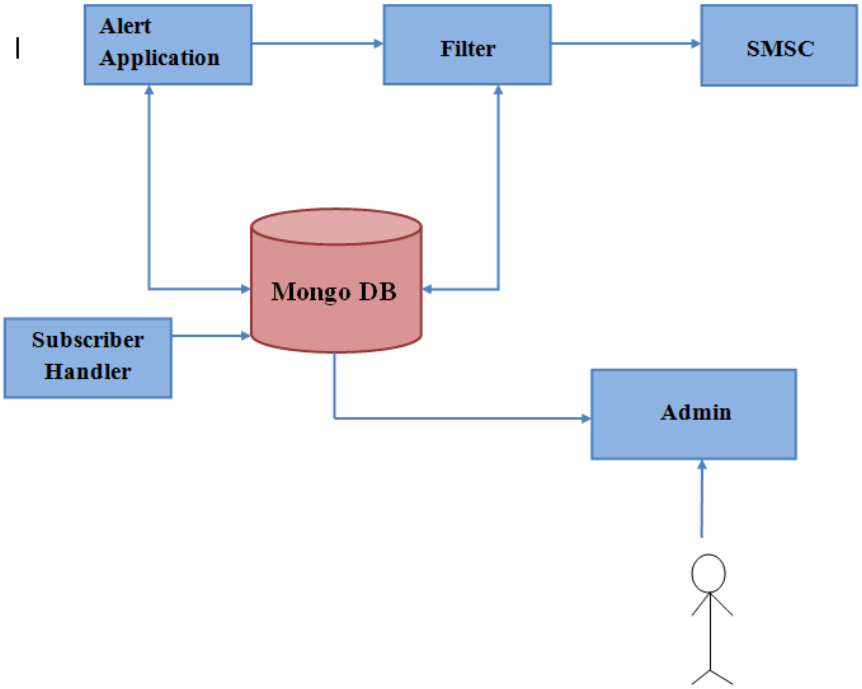


Figure 5

Contact application implementation & data filtering - P.T.K Perera

Under Contact application implementation, it supposed to implement front end UIs, which handle response messages of the alert application. This shall allow subscribers to send messages to the application created by the service provider and display all the messages in a web UI allowing service provider to reply for each and every message or to send a default/common response message to all the messages. As an example, assume that TV Derana wants to get a feedback from their news alert subscribers and their question is; Question: “Please send a feedback about Derana News Alert service” They simply send the above message without any selectable answers such as “Good/Bad”, “100%, 0%” rating and wait for subscriber’s manually typed feedback message. Such as; Answer: “News alert service is really worth. Thank You Derana” The Contact application implementation is a place where subscribers can reply back to their media partners. Basically under data filtering, all the informal and formal messages will be checked automatically. Client (media partners) will use an application which was already built in –Name- , such as Service Providing Apps and send to the subscribers who subscribed into those Apps, in order to serve them with information. Data filtering process will use an automatic mechanism to filter all application details which are about to send to the subscribers and if the process found an informal message it will send to the Admin to verify and analyze and other remaining formal messages will be sent to the subscribers.

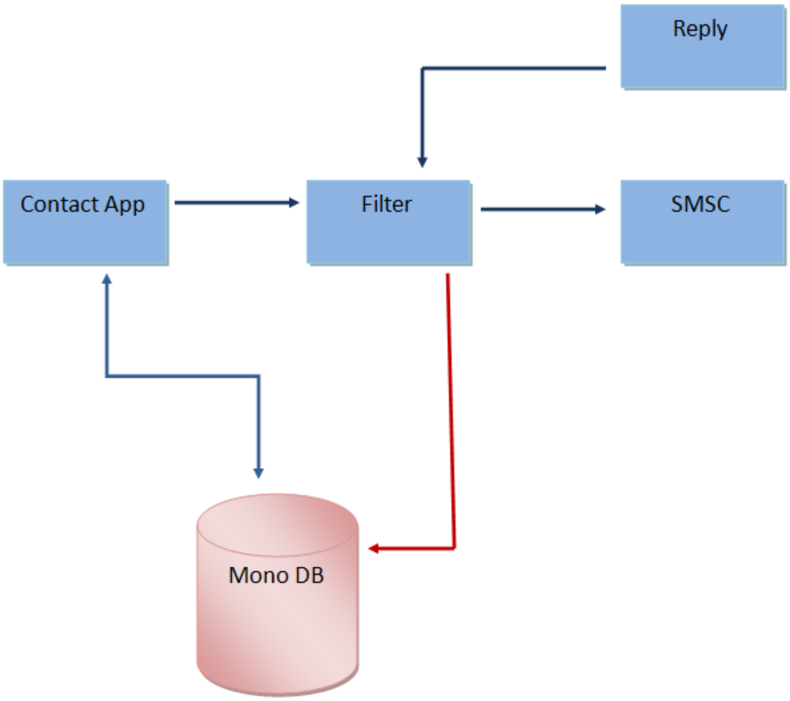
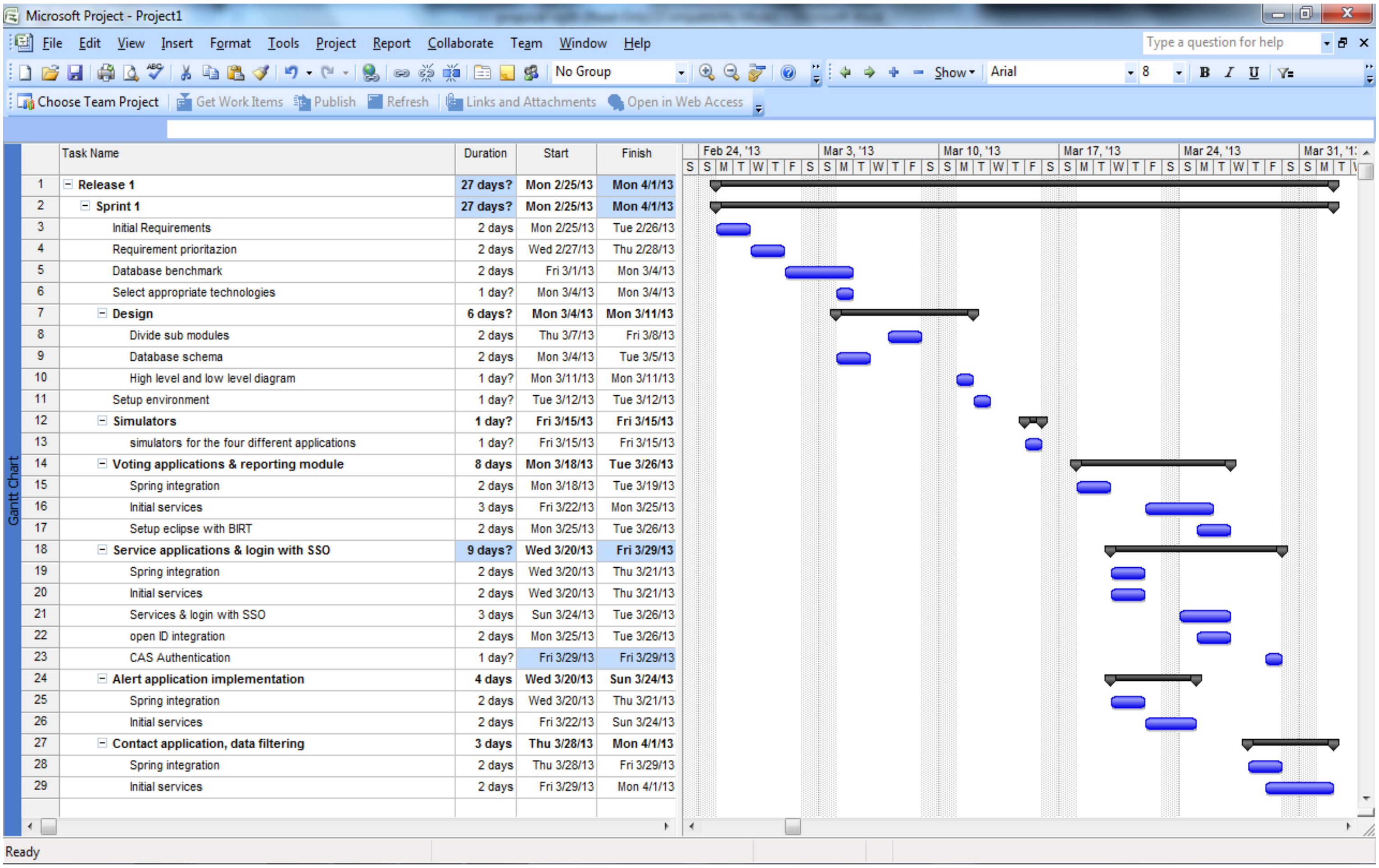


Figure 6

3.4 Project Plan



	Task Name	Duration	Start	Finish
1	**Release 1**	**27 days?**	**Mon 2/25/13**	**Mon 4/1/13**
2	**Sprint 1**	**27 days?**	**Mon 2/25/13**	**Mon 4/1/13**
3	Initial Requirements	2 days	Mon 2/25/13	Tue 2/26/13
4	Requirement prioritazion	2 days	Wed 2/27/13	Thu 2/28/13
5	Database benchmark	2 days	Fri 3/1/13	Mon 3/4/13
6	Select appropriate technologies	1 day?	Mon 3/4/13	Mon 3/4/13
7	**Design**	**6 days?**	**Mon 3/4/13**	**Mon 3/11/13**
8	Divide sub modules	2 days	Thu 3/7/13	Fri 3/8/13
9	Database schema	2 days	Mon 3/4/13	Tue 3/5/13
10	High level and low level diagram	1 day?	Mon 3/11/13	Mon 3/11/13
11	Setup environment	1 day?	Tue 3/12/13	Tue 3/12/13
12	**Simulators**	**1 day?**	**Fri 3/15/13**	**Fri 3/15/13**
13	simulators for the four different applications	1 day?	Fri 3/15/13	Fri 3/15/13
14	**Voting applications & reporting module**	**8 days**	**Mon 3/18/13**	**Tue 3/26/13**
15	Spring integration	2 days	Mon 3/18/13	Tue 3/19/13
16	Initial services	3 days	Fri 3/22/13	Mon 3/25/13
17	Setup eclipse with BIRT	2 days	Mon 3/25/13	Tue 3/26/13
18	**Service applications & login with SSO**	**9 days?**	**Wed 3/20/13**	**Fri 3/29/13**
19	Spring integration	2 days	Wed 3/20/13	Thu 3/21/13
20	Initial services	2 days	Wed 3/20/13	Thu 3/21/13
21	Services & login with SSO	3 days	Sun 3/24/13	Tue 3/26/13
22	open ID integration	2 days	Mon 3/25/13	Tue 3/26/13
23	CAS Authentication	1 day?	Fri 3/29/13	Fri 3/29/13
24	**Alert application implementation**	**4 days**	**Wed 3/20/13**	**Sun 3/24/13**
25	Spring integration	2 days	Wed 3/20/13	Thu 3/21/13
26	Initial services	2 days	Fri 3/22/13	Sun 3/24/13
27	**Contact application, data filtering**	**3 days**	**Thu 3/28/13**	**Mon 4/1/13**
28	Spring integration	2 days	Thu 3/28/13	Fri 3/29/13
29	Initial services	2 days	Fri 3/29/13	Mon 4/1/13

Figure 7

3.5 Work Break down Structure

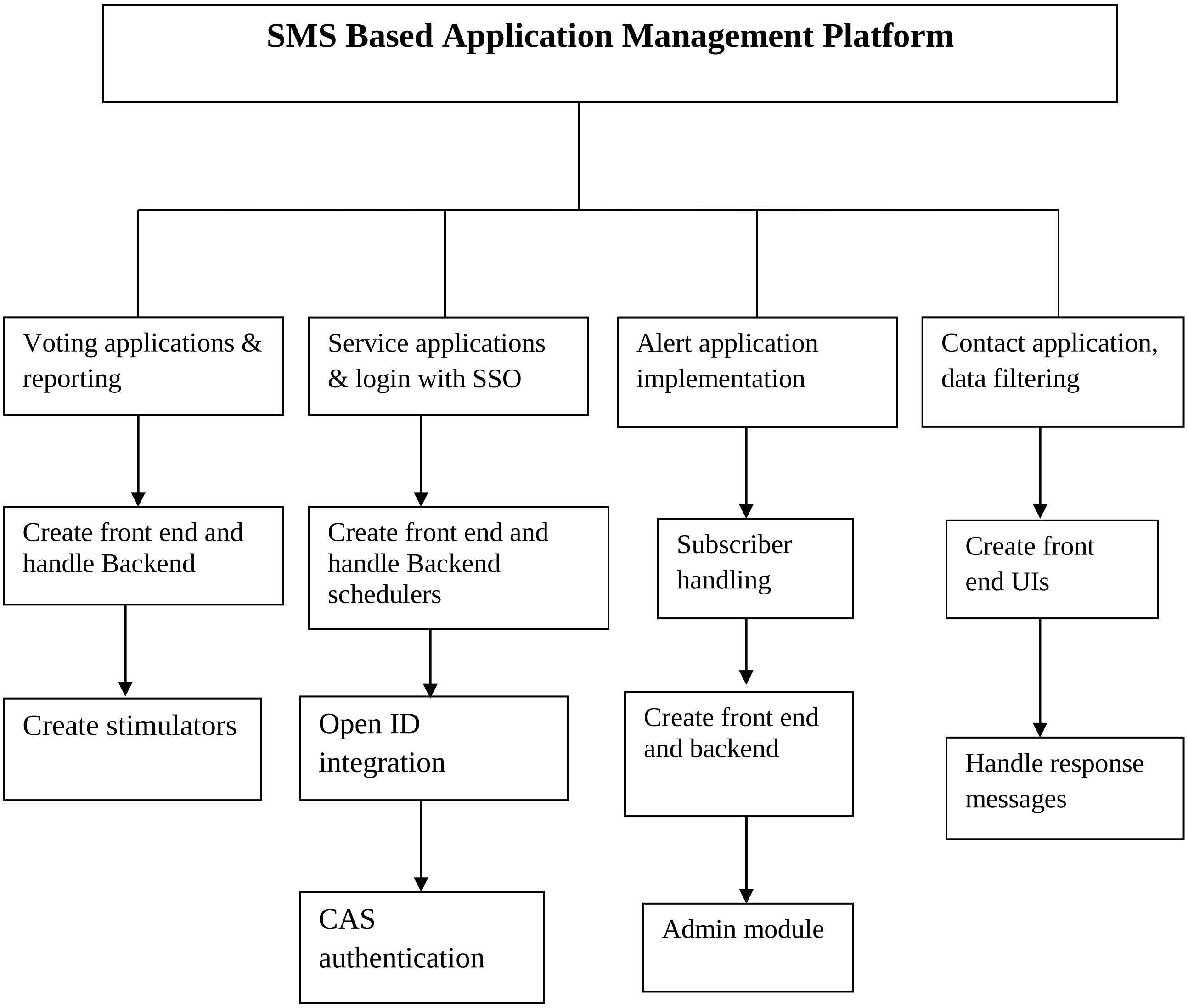

SMS Based Application Management Platform
Voting applications & reporting
Service applications & login with SSO
Alert application implementation
Contact application, data filtering
Create front end and handle Backend
Create front end and handle Backend schedulers
Subscriber handling
Create front end UIs
Create stimulators
Open ID integration
Create front end and backend
Handle response messages
CAS authentication
Admin module

Figure 8

Software Requirements for development

- Intellij Idea 11.0 version
- DB Visualizer
- Eclipse – Use for report design
- MongoVUE
- mongo

Graphics

- Microsoft project2007 - Use to develop Gant chart.
- Corel Draw 12 – Use to draw diagrams.
- Photoshop - To design the interfaces of the system
- Adobe flash – To use in the power point presentations

Technologies

- Apache maven
- Apache tomcat
- Spring framework
- Quartz Schduler
- BIRT report

4. Description of Personal and Facilities

Name	DIT No.	Assigned work
R.M.C.U. Senevirathna	IT10031308	Voting application implementation & real time reporting ○ Implement Voting application - Create front end and handle Backend ○ Real time reporting
L. Shashika Amali	IT10029114	Service application implementation, common registration & login with SSO ○ Implement Service application - Create front end UIs - Handle Back end schedulers ○ CAS with SSO ○ Open ID integration
H. H. Rajamanthrie	IT10029664	Alert application implementation, admin & subscriber handling ○ Implement Alert application - Create front end UIs - Handle Subscription and un-subscription ○ Implement admin module ○ Subscriber detail handling
P.T.K Perera	IT10025604	Contact application implementation & data filtering ○ Implement Contacts application - Create front end UIs - Handle response messages ○ Filtering Implementation

5. Conclusion

The project will be most useful for Telco companies and external companies willing to advertise through use of messages. Once the proposed project is successfully implemented it will provide an extensive range of voting, alerting, subscribing, competitions, viewing request messages, and respond on requests applications, provide an open platform that allows the creation of applications, provide a wide variety of reports- real time reporting with a User friendly dashboard. Response options based on the user requirements will be handled by the system. The proposed system is expected to be done with no problems and be performed its' tasks without exception.

6. References

[1] Madhusudhan Konda, “what’s new in Java 7?”, 2011, pp. 20-121

[2] Craig walls, “Spring in Action”, 2012, pp. 20-121

[3] Christian Bauer, “Java Persistence with Hibernate”, 2010, pp. 54-79.

[4]Marten Deinum , ”MongoDb in Action”, 2011, pp.35-155

[5] Kristina Chodorow , “Scaling MongoDB”, 2012,pp. 25-98

[6] Sonat Command, “Maven: The De_nitive Guide”, 2011, pp. 120-156.

[7] “Scrum Methodology”, 10-10-2011, Available: http://scrummethodology.com/scrum-sprint/. [Accessed: 21-02-2013].

[8] “Introduction to Scrum – An Agile process”, 13-12-2011, Available: http://www.mountaingoatsoftware.com/topics/scrum. [Accessed: 21-02-2013].

[9]"Sharding-mongoDB Manual", 2011, Available: http://docs.mongodb.org/manual/sharding/. [Accessed: 21-02-2013].

7. Appendices

Appendix A: List of Acronyms and Abbreviations

SMS - Short Message Service

WBS – Work Break Structure

SSO – Single Sign On

CAS – Central Authentication Service

SSL – Secure Socket Layer

SMSC – Short Messaging Service Centre

Appendix B: Diagrams and Figures

Figure 1: The High Level Diagram for the entire system. It shows how each component is connected to each other.

Figure 2: Scrum project management method. Shows how the scrum process works.

Figure 3: The High Level Diagram for the voting application implementation & real time reporting module

Figure 4: The High Level Diagram for the service application implementation, common registration & login with SSO module

Figure 5: The High Level Diagram for the alert application implementation, admin & subscriber handling module

Figure 6: The High Level Diagram for the contact application implementation & data filtering module

Figure 7: Project plan

Figure 8: Work break structure shows the main components and the functions that fall under it.