Smart SMS- SMS Application Management Platform

Software Requirements Specification

Sri Lanka Institute of Information Technology

Special Honors Degree of Bachelor of Science in Information Technology

May 2013

Title of the project: “Smart SMS” SMS Based Application Management Platform

Project ID: 13-009

Team:

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

Author name: L.Shashika Amali

Author’s ID: IT10029114

Signature of the author:

Supervisor name: Ms.Aruni Niroshika

Supervisor signature:

Co-supervisor name: Ms.Umanga Pilapitiya

Co-supervisor signature:

Declaration

I hereby declare that the submitted software requirement specification document for the proposed project work entitled “Smart SMS” SMS Based Application Management Platform submitted to the Sri Lanka Institute of Information Technology is a record of an original work done by me.

L. Shashika Amali ……………………………

Signature

Contents

List of Tables & Figures

1. Introduction

1.1 Purpose

This Document is intended to define the Software Requirements Specification for “Smart SMS” SMS Based Application Management Platform. This platform is developed by 13-009 4th year research group in Sri Lanka Institute Information Technology under the supervision and the guidance of the Ms. Aruni Niroshika. This contains a complete description of the behavior of the platform to be developed. This document is primarily intended for the supervisor, project coordinator and the project developers. All parts are intended primarily for customers of the application, but will also be of interest to software engineers building or maintaining the software. This document contains mainly two sections which are user requirements and system requirements. Both parties that are the developers and the clients can read the document and understand how the system carries on, the processes of each activities and the final outcome of the system. If the client is not satisfied with the content of the document at that moment developers can understand the requirements after further clarification. So the document provides an easy way to confirm user requirements correctly from the beginning.

1.2 Scope

The proposed project “Smart SMS” SMS Based Application Management Platform is intended to implement a platform which allows creation and customizing of SMS based application types with the capability of handling a huge amount of data concurrently. There are four main SMS based applications; alert applications, voting applications, service applications and contact applications. During the completion of the project, all user interfaces necessary to create the four main SMS applications will be designed and developed.

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. 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. The implementing front end UIs and handling subscription and un-subscription of the alert application, implementing admin module and handle subscriber details and operations will be done in this project. 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.

The author of this document will be responsible for Service application implementation and common registration & login with SSO. 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 the system.

The main research problem identified is clustering and concurrent high load data handling and MongoDB will be used to address the issue. 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. Integration of the central authentication service (CAS) with open ID integration for user interfaces, handling large amount of data without affecting system performance are some other major concerns in the developing the system.

After the completion of the project it will facilitate third party developers to create applications without knowing the underlying technologies and the system will be capable to analyze large amount of data. This framework can be used by any Telco company with smaller configurations. Central authentication Service with Single Sign On will permit 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 will allow 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. Final outcome of the system will have a high availability and a high security.

1.3 Definitions, Acronyms, and Abbreviations

SMS	Short Message Service
CAS	Central Authentication Service
SSO	Single Sign-On
SSL	Secure Socket Layer
SMSC	Short Messaging Service Centre

Table 1

1.4 Overview

This document mainly consists of four categories. Those are introduction, user requirements, specific requirements and special details. Since the introduction is completed in above parts of the document as the next step of this document, user requirements, specific requirements and special details are to be discussed further. User requirements are for the user understanding and user perspective. User requirements contain the interfaces of the system, the constraints, diagrams and etc. The specific requirements are used to be understood by the developer. It contains interfaces (standard than interfaces defined under user requirements), classes/objects, non-functional requirements and etc. Later part of this document contains special details and information. It includes all references and appendices. Also this section contains some detail description about diagrams which available at the middle of this document for better understanding. In coming sessions it discuss about the entire system as well as my contribution, which is the Service application implementation, common registration & login with SSO.

2. Overall descriptions

The proposed project is about application for distributed computing. Nowadays one of the most important facts in the world is communication. To communicate currently many methods are used such as mobile phones, internet and etc. There are many features in mobile phones such as SMS, voice calls, LBS, GPRS and USSD. Out of the features mentioned SMS is the most widely used data application in the world and it has been one of the easiest ways to address customers in the emerging economic world. Main focus of the framework to be implanted that is this project is based on SMS based applications creation which allows service providers to define and build applications through a user friendly manner. There are four types of SMS based applications that have been identified, alert applications, contact applications, service applications and voting applications. Normal scenario for a voting application or some other SMS based application in existence is that when a client contacts a service provider requesting a SMS application; the service provider has to make a whole new application to each individual client separately. The proposed system will provide an application management platform that will allow the creation of SMS based applications through web interfaces. The framework also maintains some other options such as CAS in order to facilitate clients to register and log on to the applications using open ID integration. Apart from this the system will facilitate to generate real time reports and use mechanisms to filter data that are passed through the platform. It also capable of subscriber details handling and will be implementing a complete administrator module to monitor and control necessary manual work of the platform. My contribution towards the project will be the Service application implementation, common registration & login with SSO.

This module needs to integrate with spring framework, CAS and open ID integration. Additionally it will be using apache maven, quartz scheduler as main technologies and use many properties to configure the system.

2.1 Product perspective

2.1.1 System interfaces

This kind of similar project has not been done earlier as an industrial project. There is no such system which allows all types of SMS based application creation with web interfaces through the use of a platform. The existing technologies cannot analyze huge amount of data concurrently like the proposed system. But proposed system concerns about clustering and data replication. So it can be concluded that there are no similar projects as proposed project.

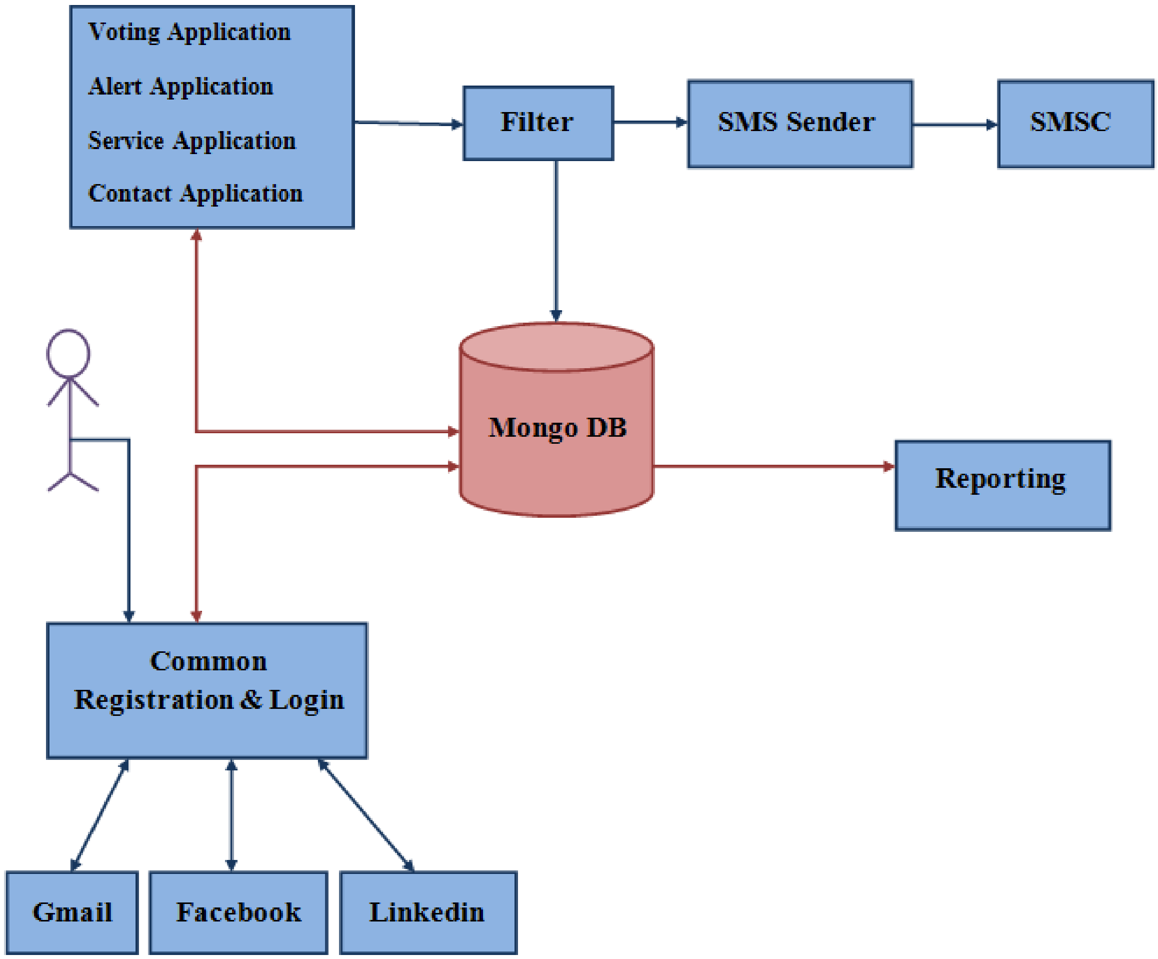


Figure 1: System High Level Architecture Diagram

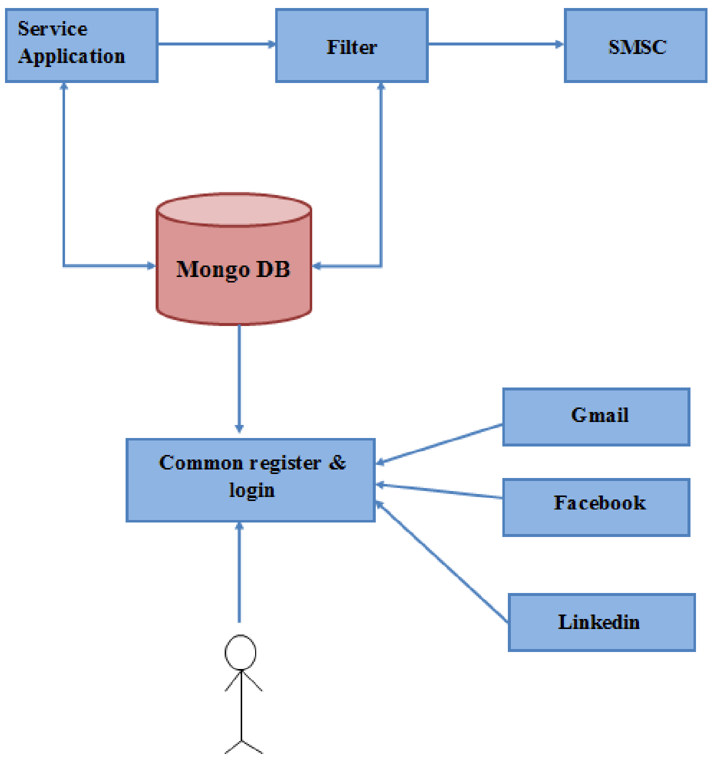


Figure 2: ServiceApp, common registration & SSO module system diagram

2.1.2 User Interfaces

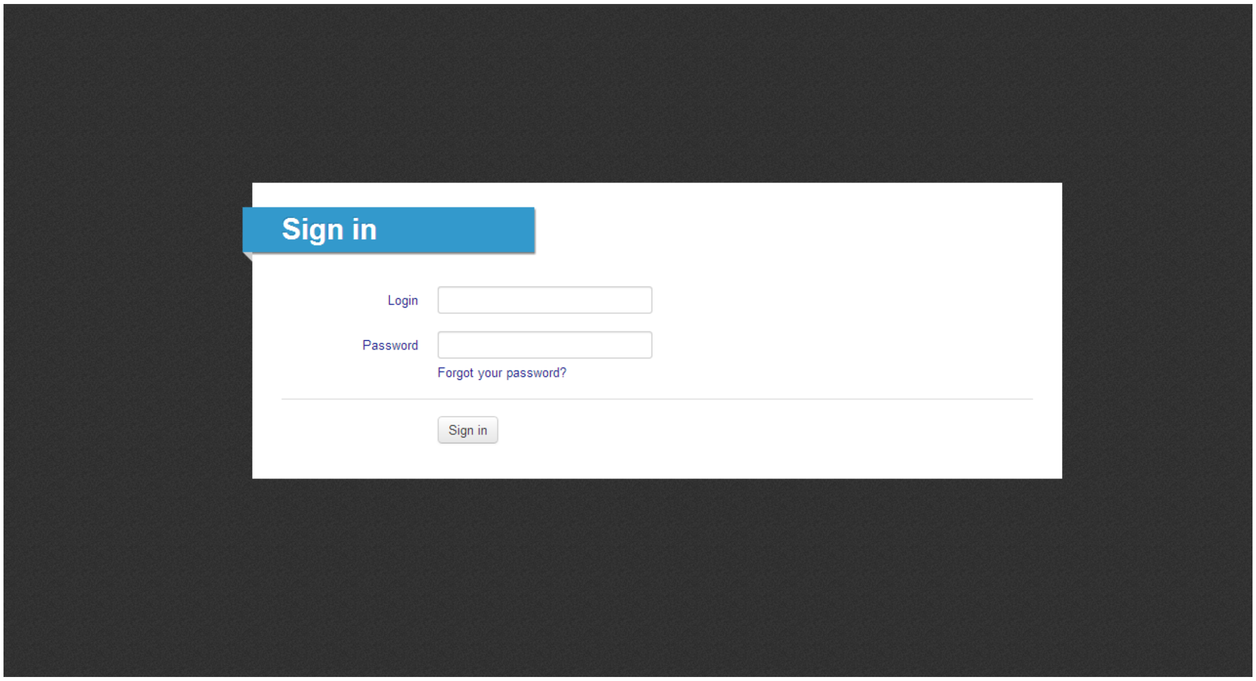


Figure 3: Login page

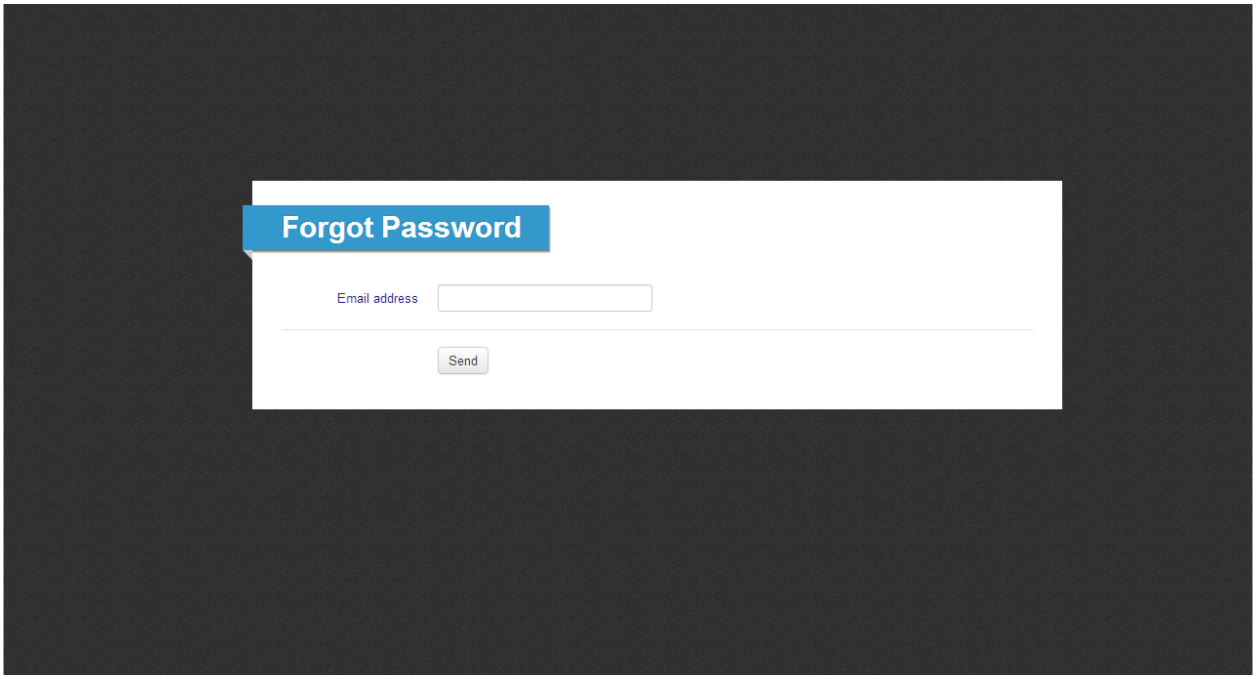
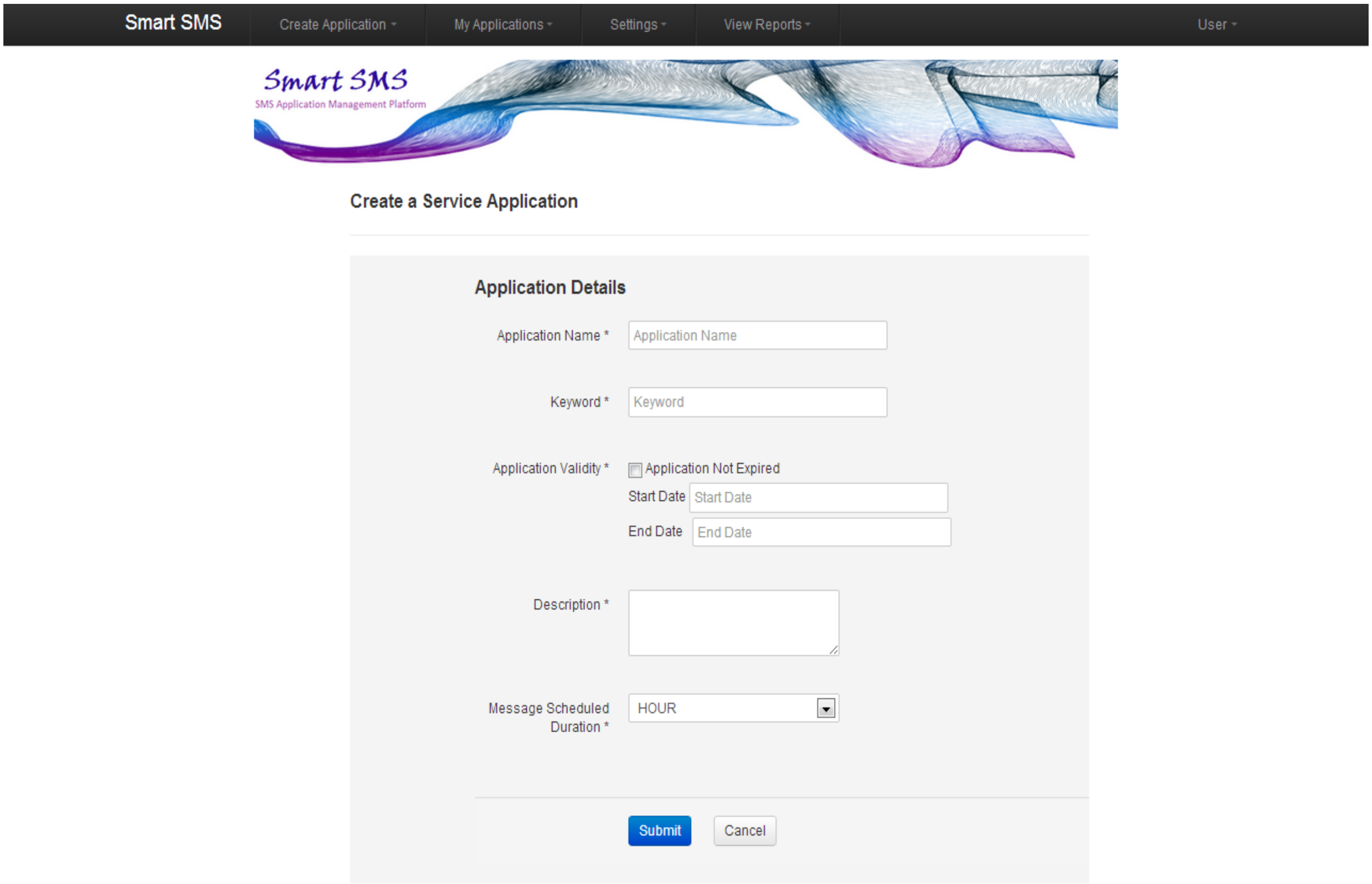


Figure 4: Forgot password page



Create a Service Application

Application Details

Application Name * Application Name

Keyword * Keyword

Application Validity * □ Application Not Expired

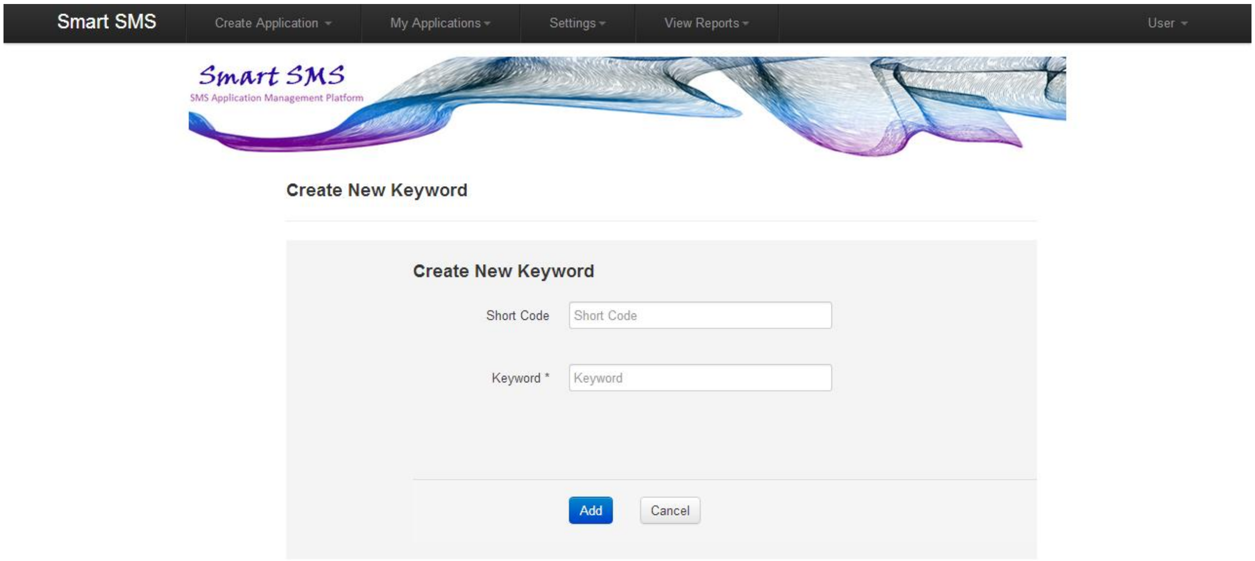
Start Date Start Date

End Date End Date

Description *

Message Scheduled Duration * HOUR

Figure 5: Create Service Application page



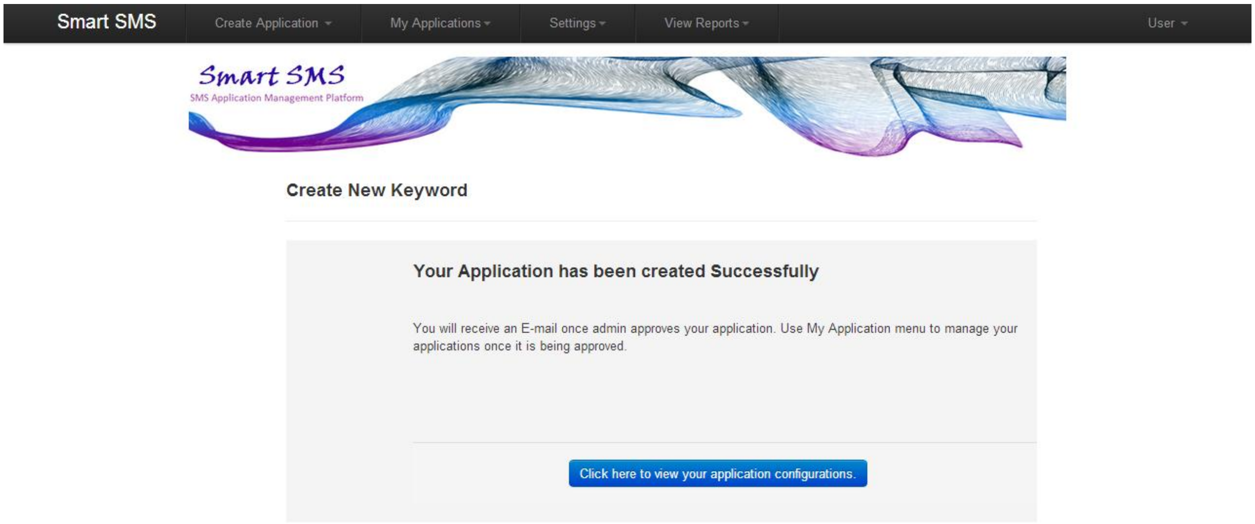
Create New Keyword

Create New Keyword

Short Code: Short Code

Keyword *: Keyword

Figure 6: Create new keyword page


Smart SMS
Create Application
My Applications
Settings
View Reports
User
Smart SMS
SMS Application Management Platform

Create New Keyword

Your Application has been created Successfully

You will receive an E-mail once admin approves your application. Use My Application menu to manage your applications once it is being approved.

Click here to view your application configurations.

Figure 7: Success page

2.1.3 Hardware Interfaces

- High End Server

To run the independent modules of the system; it will require high end server because the modules will be dealing with large amount of data.

- Web server

The system will require a web server to run all the front end modules e.g. Tomcat, Jboss.

- Database server

The system will require Mongodb server to handle a large amount of data.

2.1.4 Software Interfaces

- Any Java Supported IDE (Intellij IDEA, Eclipse, Netbeans)

Since Apache maven as a building tool it will support all the java supported browsers to code the project.

- MongoDB(MongoVUE)

MongoVUE will be used to show the representation of Mongo data.

- Visual paradigm

Visual paradigm will be used to design use cases, activity diagrams, class diagrams and etc.

- Adobe Dreamviewer

Adobe Dreamviewer will be used to design user interfaces.

2.1.5 Communication Interfaces

- LAN Port

LAN Port will connect servers with each other.

2.1.6 Memory Constraints

- Database server need to have more than 4GB ram because it will handle large amount of data.
- High end server need to have more than 2GB ram because the backend modules need to run 24 hours.

2.1.7 Operations

- The system should allow the creation of alert applications, voting applications, service applications and contact applications.
- The system should be able to get database backups
- The system should generate real time reports
- The system should produce central authentication service for users
- The system should filter data which transferred from apps to its’ subscribers checking for informal or any abusive contents
- The system should available at any time
- The system should capable enough to handle large number of data
- The system should respond any crashes and recover data

2.1.8 Site adaptation requirements

- User interfaces support internationalization
- Back-end modules have readme to guide deployment
- Have a user manual to give instruction how to use the system

2.2 Product Functions

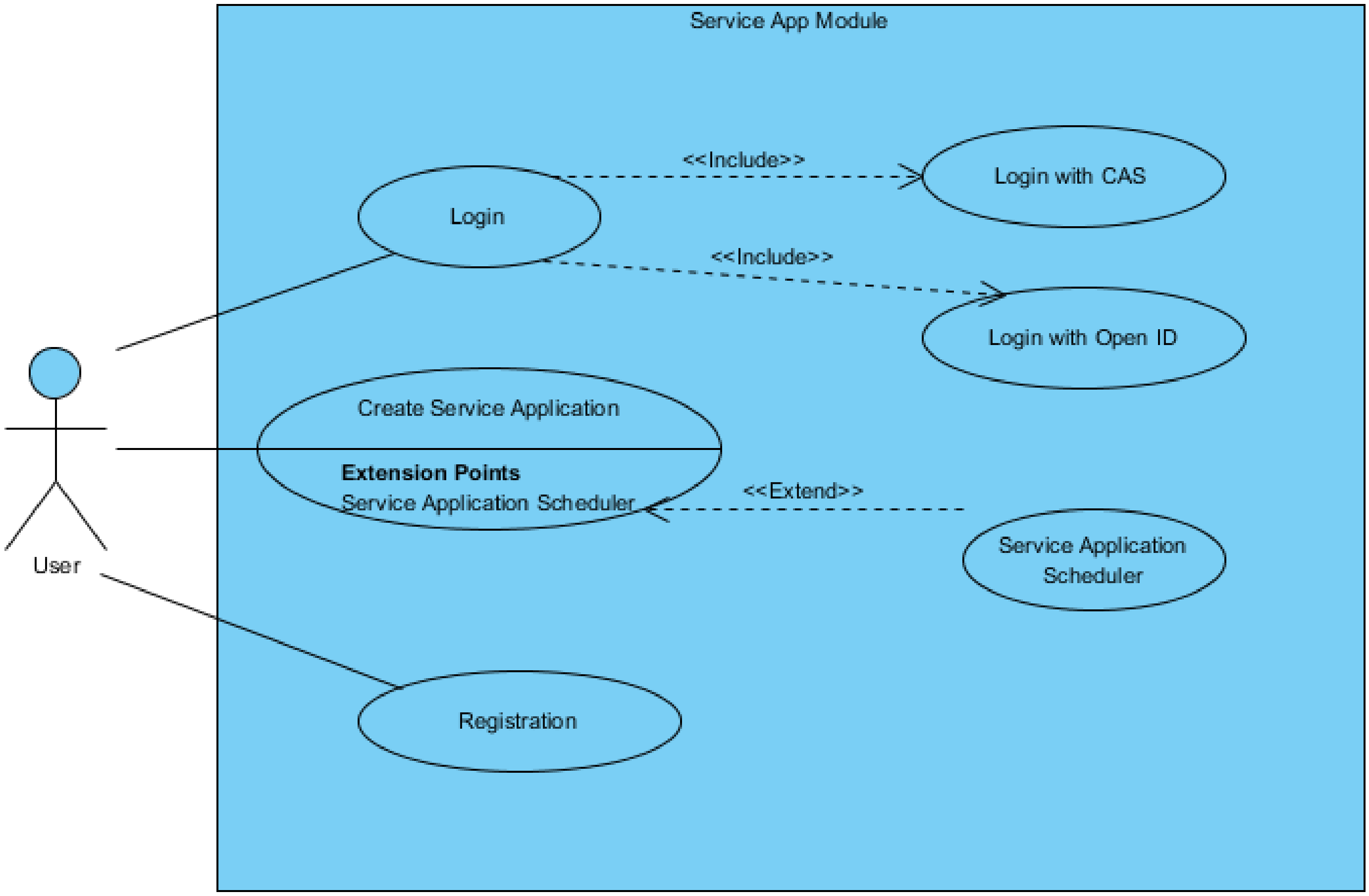


Figure 8: Use case diagram

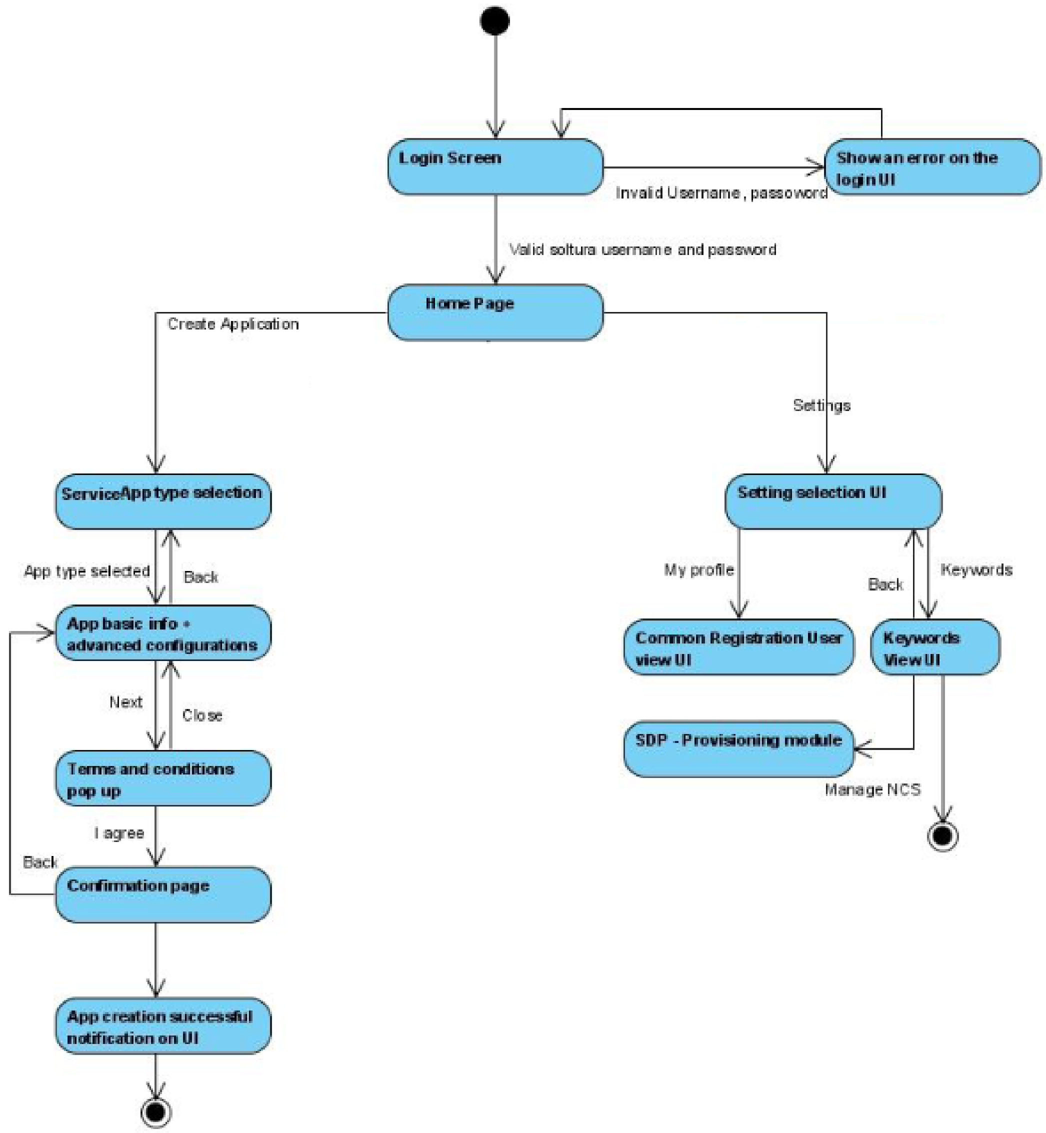

Login Screen
Show an error on the login UI
Invalid Username , passoword
Valid soltura username and password
Home Page
Create Application
Settings
ServiceApp type selection
App type selected
Back
App basic info + advanced configurations
Next
Close
Terms and conditions pop up
I agree
Back
Confirmation page
App creation successful notification on UI
Setting selection UI
My profile
Back
Keywords
Common Registration User view UI
Keywords View UI
SDP - Provisioning module
Manage NCS

Figure 9: Activity diagram

2.3 User characteristics

The system consists of four main modules. Users of the back-end module need to have basic knowledge of implementation. Users of the user interfaces modules can be anyone.

2.4 Constraints

- To the implementation java and MongoDB are required.
- Should use windows XP or above and any UNIX operating system
- Power failures will interrupt the system
- Generate reports from BIRT using MongoDB

2.5 Assumptions and dependencies

Future version of “Smart SMS” will handle other types of sms applications and a larger amount of data concurrently with high performance.

2.6 Apportioning of requirements

The requirements explain above are primary specification. As Agile project management these requirements can be change. Essential requirements will be implementing for this sprint as first release of the system. Desirable requirements are to be implemented in this release if possible.

3. Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

- Login Interface

Use CAS to authenticate users. It protects the system from unauthorized access.

- Create a Service application Interface

User interface designed to create a new service application.

- Create new key word Interface

Interface to allow the users to define a new keyword for the application.

- Application Terms & conditions Interface

- Success Interface

The interface which shall be displayed once the application is being successfully created.

3.1.2 Hardware interfaces

- High End Server

To run the independent modules of the system; it will require high end server because the modules will be dealing with large amount of data.

- Web server

The system will require a web server to run all the front end modules e.g. Tomcat, Jboss.

- Database server

The system will require Mongodb server to handle a large amount of data.

3.1.3 Software interfaces

- Visual paradigm

  To design use cases, activity diagrams, class diagrams and etc.

- Adobe Dreamviewer

Adobe Dreamviewer will be used to design user interfaces.

- Adobe Photoshop

Adobe Dreamviewer will be used to do the graphics editing in the project.

3.1.4 Communication interfaces

- LAN Port

  Network servers with modules

3.2 Classes/ Objects

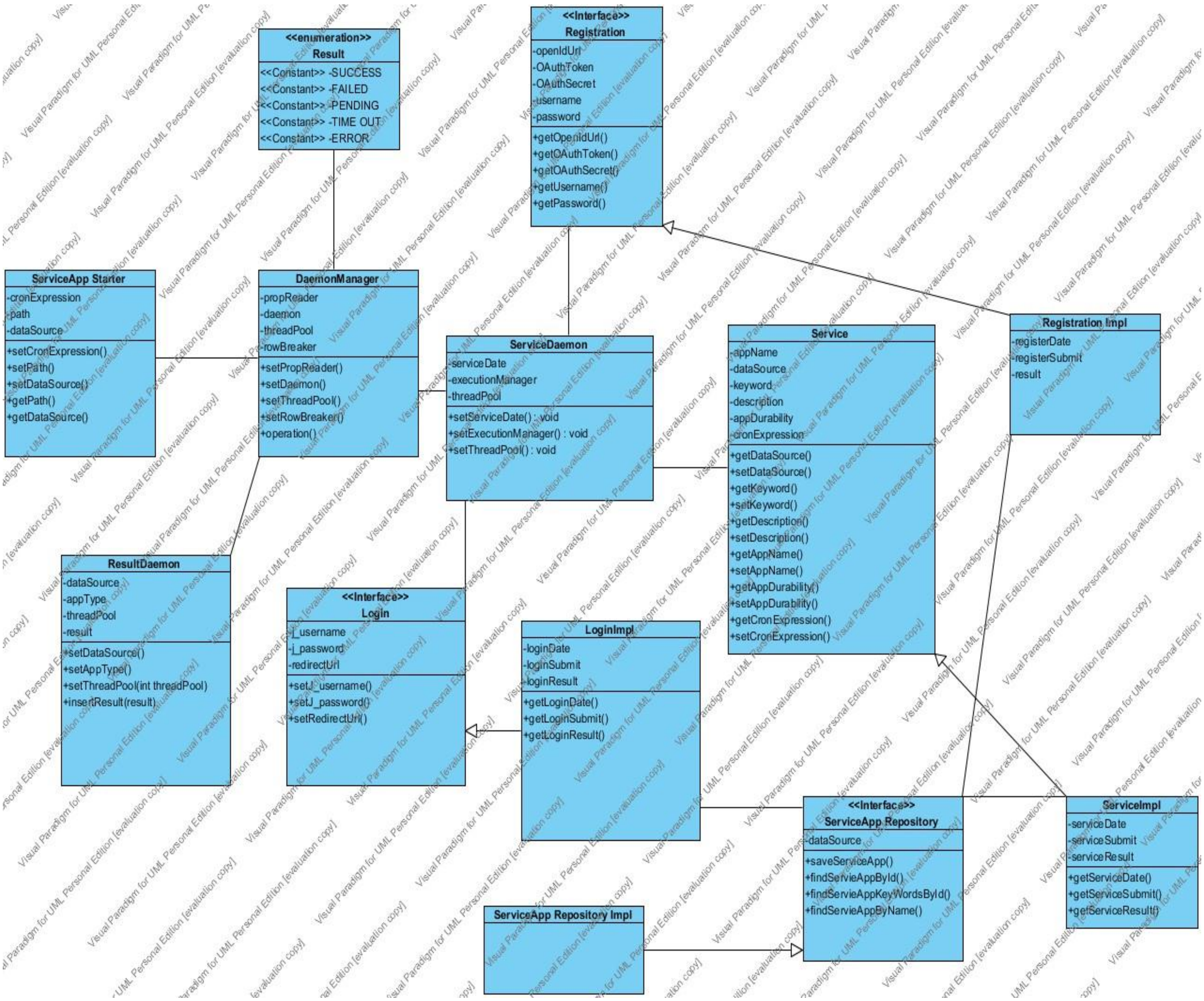


Figure 10: Class diagram

3.3 Performance requirements

The system will require high performance for backend server modules. The key areas needed to be taken into consideration to enhance the performance of the system can be categorized as follows.

- Database handling

All database actions such as, user login authentication, data retrieval and data entering must be done on real time without having the users waiting for a response. So the number of transactions per second to be supported by the web server should be around 100 – 200. Availability of the system should be very high, so as the response time.

- Concurrent user handling

The web application should be capable of handling a large number of users simultaneously, without affecting the performance.

- Memory and disk usage

The system will require multiple servers each with 4 GB of RAM and the servers need to be act as a cluster.

3.4 Design constraints

The system design is use to confirm the validity of the requirements so that it will help to build a system that will provide the functionalities that user expects. Even though the SMS Based application management platform is a server based project, some design constraint are used in it.

- HTML

All the interfaces are built on top HTML. Twitter Bootstrap is used as a CSS 3 framework because it will support all major browsers.

- Main Design Constraint

960px width use for web page because it will fit any screen.

Keep it simple.

Avoid Graphics.

Navigation should be clear and concise.

3.5 Software system attributes

3.5.1 Reliability

Backend modules are highly reliable modules.

3.5.2 Availability

System requires be up and running all the time. If there is a failure it will send SNMP trap to maintain group.

3.5.3 Security

Since system use CAS for authorization and authentication security will be high.

3.5.4 Maintainability

For all the backend modules user guide will be provided describing for how to change configuration, stop threads, how to start the server etc.

3.6 Other requirements

The system must support all web browsers because it will be using the bootstrap framework.

4. Supporting information

4.1 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.

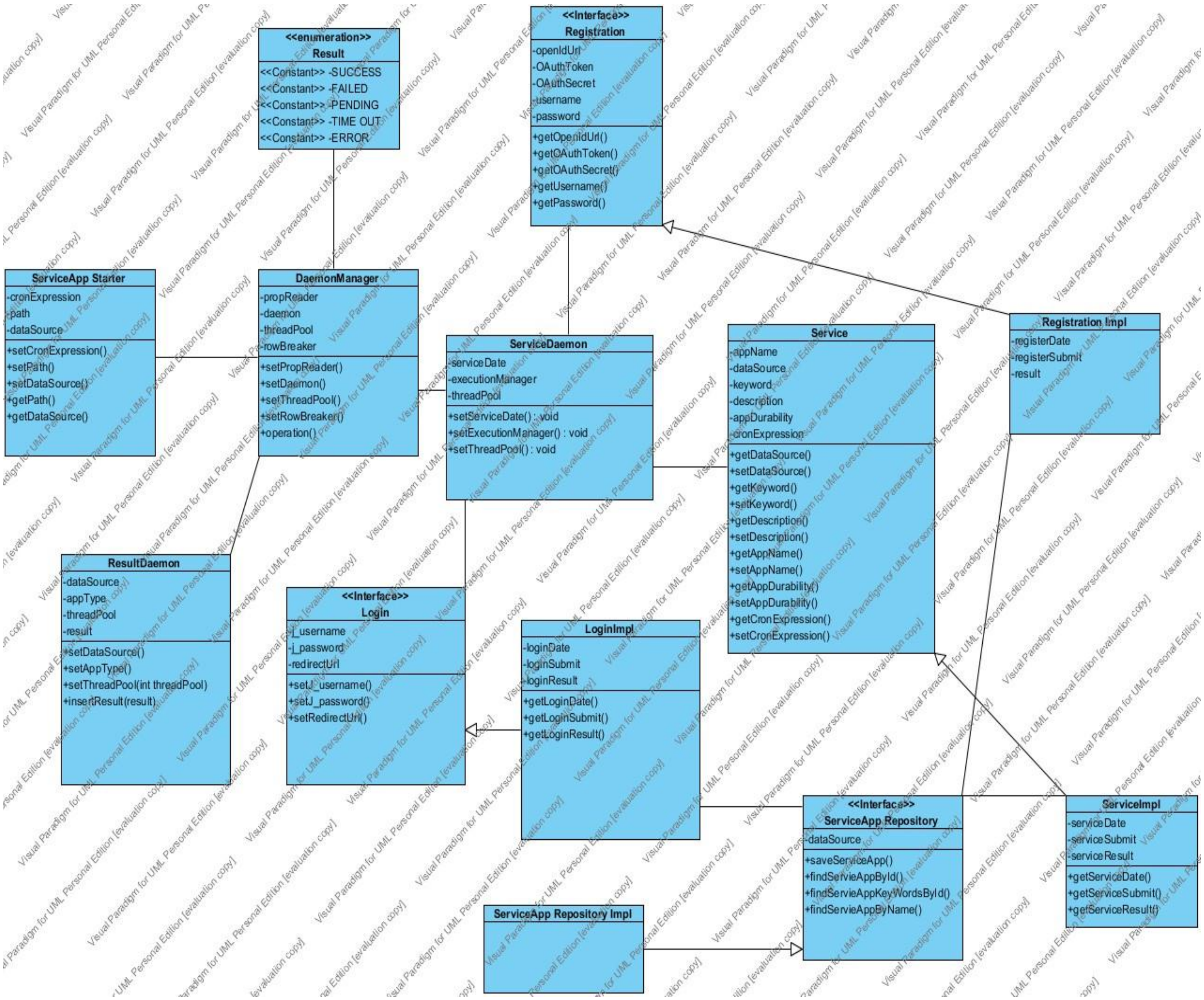
[7] “Scrum Methodology”, 10-10-2011, Available: http://scrummethodology.com/scrum-sprint/. [Accessed: 21-02-2013].

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

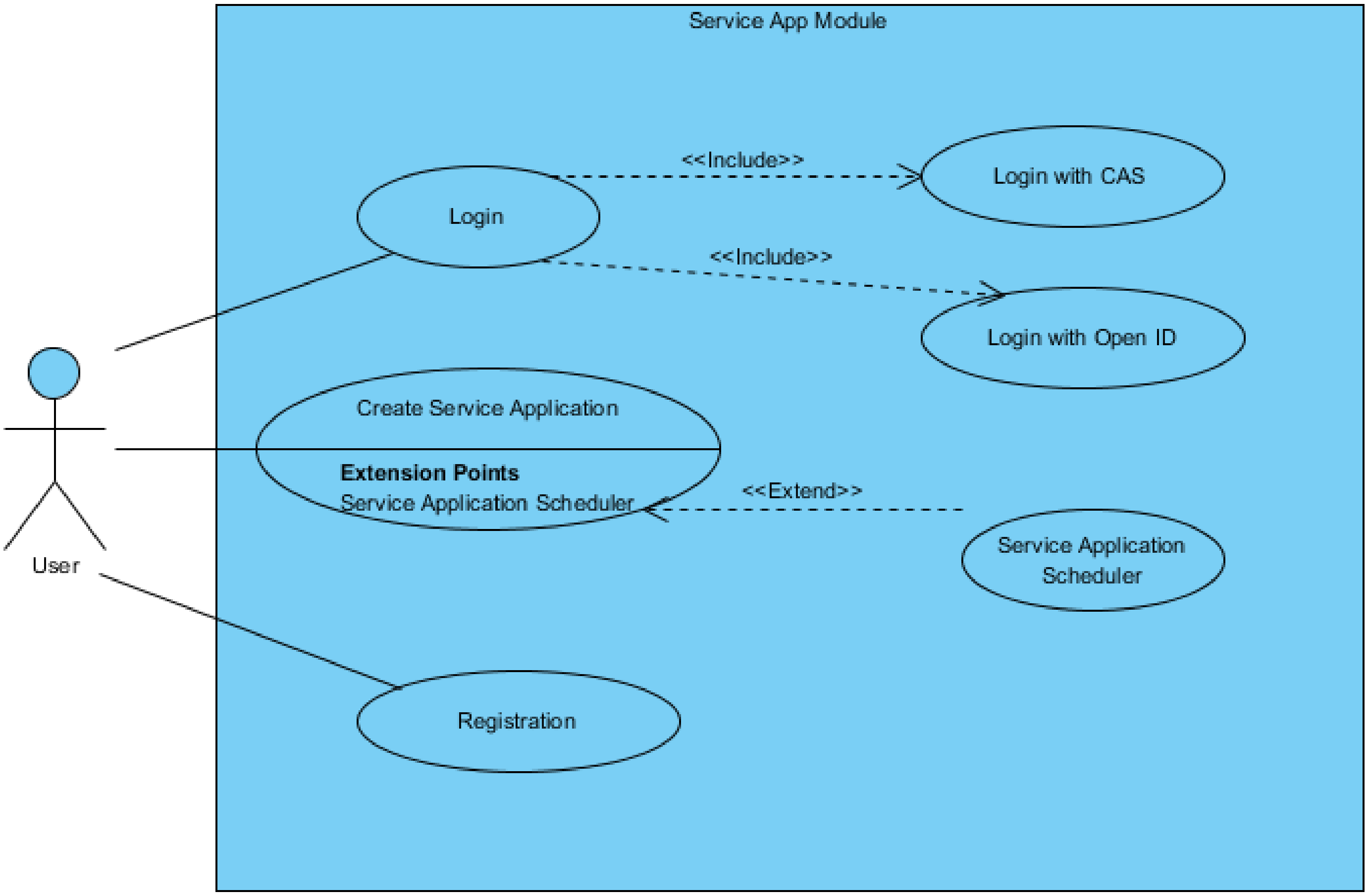
[9]"Sharding-mongoDB Manual", 2011, Available: http://docs.mongodb.org/manual/sharding/. [Accessed: 21-02-2013].

4.2 Appendices

4.2.1 Class diagram



4.2.2 Use case diagram


Service App Module
<<Include>>
Login with CAS
Login
<<Include>>
Login with Open ID
Create Service Application
Extension Points
Service Application Scheduler
<<Extend>>
Service Application Scheduler
User
Registration