

Process MeNtOR 3.0

Uni-SEP

Iqbal Final SIP

Communication System

Requirements Model

Version:	1.0
Print Date:	
Release Date:	
Release State:	Initial/Core/Final
Approval State:	Draft/Approved
Approved by:	
Prepared by:	
Reviewed by:	
Path Name:	
File Name:	T3-ReqmtsModel.doc

Document No:	
--------------	--

Document Change Control

Version	Date	Authors	Summary of Changes
1.0	Oct. 20 /05		Initial Write

Document Sign-Off

Name (Position)	Signature	Date
		Oct. 20
		Oct. 20
		Oct. 20
		Oct. 20

Contents

1	INTRODUCTION.....	5
1.1	Purpose.....	5
1.2	Overview	5
1.3	References	5
2	BUSINESS SCENARIO MODEL	7
2.1	Actors	7
2.1.1	Overview	7
2.1.2	Actor Diagram.....	7
2.1.3	Actor Definitions.....	8
2.2	Use Case Descriptions.....	10
2.3	Use Case Diagrams	14
3	DOMAIN MODEL.....	15
3.1	Domain Model Class Diagram.....	15
3.2	Domain Model Class Definitions.....	16
4	INTERACTION DIAGRAMS.....	19
4.1	Sequencing Diagrams.....	19
4.2	Collaboration Diagrams	21
5	NON-FUNCTIONAL REQUIREMENTS SPECIFICATION	24
5.1	Overview	24
5.2	Enabling Technologies.....	24
5.2.1	Target Development Environment.....	24
5.2.2	System Interfaces	24
5.3	Capacity Planning	24
5.3.1	Permanent Storage	24
5.4	Network.....	24
5.5	Workstations	24
5.6	Operational Parameters	25
5.6.1	Useability	25
5.6.2	Reliability	25
6	ACTIVITIES PLAN	26
7	DOMAIN DICTIONARY	26

1 Introduction

1.1 Purpose

This document details the requirements of a prototype system that extends the functionality of an existing SIP-Based Communication System to allow for the invocation of Web Services from within a SIP protocol environment. It examines details and scenarios pertaining to the existing system as well as those relevant to the functional extension, and presents models at the business scenario, domain, and interaction levels. It also addresses the non-functional requirements of the system, and the distribution of activities towards system development.

1.2 Overview

The aim of the project is to implement a prototype system which allows the invocation of Web Services from a SIP protocol environment. The JAVA API for Integrated Network Session Instantiation protocol (JAIN-SIP) will be used. JAIN-SIP is the full implementation of the SIP protocol RFC3261 standard by the National Institute of Standards and Technology (NIST).

The SIP Communicator (Client Agent) is a software based IP phone and is built on top of the JAIN-SIP-RI and Java Media Framework API (JMF). It is used to place VoIP phone calls between two PCs or between a PC and an IP phone. The SIP client agent can create a SIP request and send it to the SIP Proxy Server and also generate a response to a SIP request. The SIP Proxy Servers route requests to the user's current locations, authenticate and authorize users for services, implement call routing policies, and provide features to users.

The SIP Registrar Servers allow users to upload their current locations for use by the Proxy Server. To access these servers, the SIP Communicator application is used.

The objectives of the project include: a) to allow the Location Server to contact a Web Service Registry and obtain information on whether the called party has registered any Web Services under his/her account and information on the locations/handlers of these services; b) to allow the Proxy Server to send the above information to the calling party; and c) to allow the caller to invoke one of the Web Services the called party has registered. Under the usual circumstances, if the party has registered any Web Services, then that party would decline calls and prefer those services to be invoked instead.

Non-functional requirements such as security and performance are also considerations within the context of platform deployment.

1.3 References

JAIN-SIP: <https://jain-sip.dev.java.net>
SIP: <https://sip-communicator.dev.java.net/>
Eclipse: <http://www.eclipse.org/downloads/index.php>
WSDP: <http://java.sun.com/webservices/jwsdp/index.jsp>
JAXR: <http://java.sun.com/webservices/jaxr/index.jsp>
JAXRPC: <http://java.sun.com/webservices/jaxrpc/index.jsp>
WSIF: <http://ws.apache.org/wsif/>
ebXML: <http://www.ebxml.org/>
UDDI: <http://www.uddi.org/>
WSDL: <http://www.w3schools.com/wSDL/default.asp>
Java Server: <http://java.sun.com/j2ee/1.4/download.html#sdk>
GANTT: <http://wiki.phprojekt.com/index.php/Gantt-diagram>

2 Business Scenario Model

2.1 Actors

2.1.1 Overview

The actors in our system include end users of software, administrators of the server components, and the web services. End users actors exist in an environment that allows interaction with the SIP Communicator Client Agent software, and their use of the full software functionality is contingent on proper connectivity between the components of the system. Administrator actors have full access (read, write, and execute permissions) on their respective servers, as well as any physical access required for maintenance purposes. Web services are described by .wsdl files linked by the Service Bindings in their respective Web Service Entries in the Web Service Registry.

2.1.2 Actor Diagram

The figure below represents the actors in our system. Based on their interactions with the system, the actors are characterized into three general groups: a) Users, b) Web Service, and c) Administrators. As demonstrated in the diagram, the SIP Communicator Client Agent User is a generalization of both the Calling Party and the Receiving Party, while the Administrator is a generalization of the Registry Server Administrator, the Proxy Server Administrator, and the Location Server Administrator. (Please see next page for diagram).

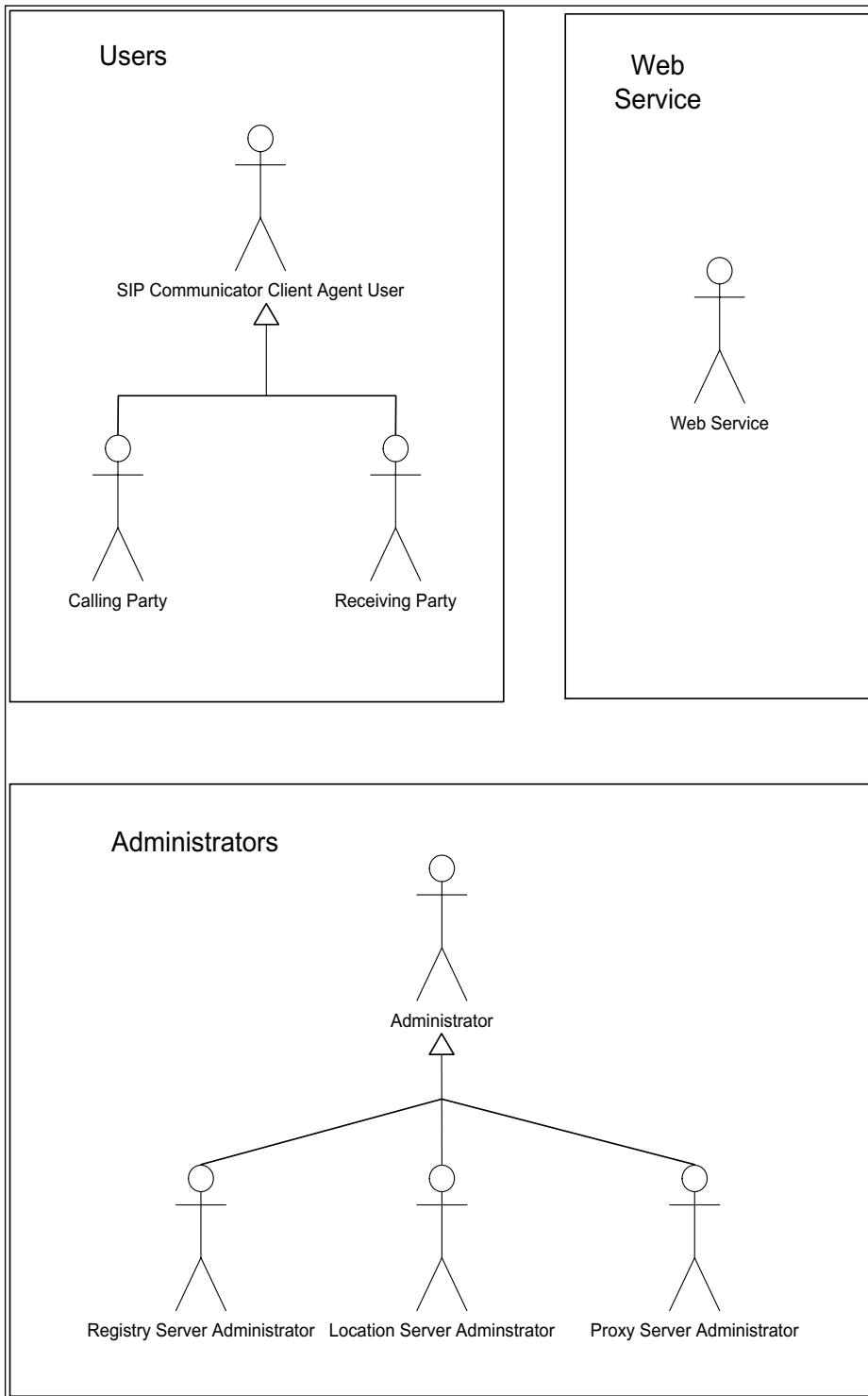


Fig. 2.1 Actor Diagram

2.1.3 Actor Definitions

SIP Communicator Client Agent User

Description	The SIP Communicator Agent User is a human actor that interacts with the system primarily through the “SIP Communicator” software. Any person that uses the “SIP Communicator” (Client Agent) is considered a “SIP Communicator Client Agent User”. The roles of this actor in the context of the system include registering the user’s location with the SIP Registrar servers and using the calling, receiving and Web Service registration/invoke features of the system.
Aliases	SIP Agent User
Inherits	None.
Actor Type	Active Also has passive roles
Contact Person	
Contact Details	

Calling Party

Description	The Calling Party is a human actor that is responsible for initiating a call through the SIP Communicator Client Agent software. Through this call the roles of the Calling Party include the placement of a VOIP phone call to a Receiving Party or the invocation of a Web Service registered by the Receiving Party.
Aliases	Agent A
Inherits	SIP Communicator Agent User
Actor Type	Active Actor
Contact Person	
Contact Details	

Receiving Party

Description	The Receiving Party is a human actor that interacts with the system by accepting or declining a call request in the case that no Web Services have been registered by him. The Receiving Party also has the ability to register Web Services to the Service Repository.
Aliases	Agent B
Inherits	SIP Communicator Agent User
Actor Type	Active Actor Also has passive roles
Contact Person	
Contact Details	

Web Service

Description	The Web Service is an external agent that is invoked through the system. Invoking a Web Service executes the actions defined by that service and returns the invocation result status back to the system.
Aliases	None.
Inherits	None.
Actor Type	Passive Actor
Contact Person	
Contact Details	

Administrator

Description	The administrator is a human actor who is in charge of a certain part of the
--------------------	--

	system (i.e. Registry Server, Proxy Server or Location Server). His roles include the smooth operation, maintenance and updating of the respective servers.
Aliases	None
Inherits	None
Actor Type	Active Actor
Contact Person	
Contact Details	

Registry Server Administrator

Description	The Registry Server administrator is a human actor who is in charge of the Registry Server. His roles include the smooth operation, maintenance and updating of the server which consists of all the registered services.
Aliases	None
Inherits	Administrator
Actor Type	Active
Contact Person	
Contact Details	

Proxy Server Administrator

Description	The Proxy Server administrator is a human actor who is in charge of the Proxy Server. His roles include the smooth operation, maintenance and updating of the server which connects the SIP Communicator Client Agent User to the system.
Aliases	None
Inherits	Administrator
Actor Type	Active
Contact Person	
Contact Details	

Location Server Administrator

Description	The Location Server administrator is a human actor who is in charge of the Location Server. His roles include the smooth operation, maintenance and updating of the server which provides location details, as well as a list of Web Services (if applicable), of the called party to the Proxy Server.
Aliases	None
Inherits	Administrator
Actor Type	Active
Contact Person	
Contact Details	

2.2 Use Case Descriptions

This section documents the complete business scenarios within the scope of this project.

2.2.1 XXXX-0001 Normal Operation Scenario with Web Services (selective email)

In this business scenario, one Agent (Agent A) uses the system to invoke a Web Service registered by another Agent (Agent B). In particular, the Web Service that will be invoked will selectively send an email to Agent B.

Description:

This scenario pertains to the invocation of a Web Service by the Calling Party (Agent A). This process starts off by the Agent A initiating a call to the Receiving Party (Agent B). This scenario requires that Agent B has already registered a Web Service under his/her organization. The system invokes the Web Service based on input from Agent A, which results in a selective email being sent to Agent B and the status result of the invocation being returned to the system.

Actors:

The actors associated with this business scenario are as follows:

1. Calling Party (Agent A)
2. Receiving Party (Agent B)
3. Web Service

The details of the above mentioned actors can be found in Section 2.1.3 (Actor Definitions).

Preconditions:

Before this scenario can be performed:

1. Agent B must have the Selective E-mail Web Service successfully registered with the Service Repository (uses the *Client Registers Web Service Scenario*).
2. The Proxy Server must be on and functioning
3. The Location Server must be on and functioning
4. The Service Repository must be on and functioning

Postconditions:

1. The web service will have performed its intended functionality

Scenario Text:

1. Invite Receiver Party for Call
 - 1.1. Supply Recipient Party Details
 - 1.2. Find User in System
 - 1.3. *Use Graceful Exit* use case to handle errors in contacting servers
2. Check For Web Services

- 2.1. Query Service Repository for Service Handlers
- 2.2. *Use Graceful Exit* use case to handle errors in contacting servers
- 2.3. Send response back to Proxy through Location Server
- 3. Return Web Service Handlers to Calling Party
- 4. Invoke Web Service
 - 4.1. Select Web Service (selective email)
 - 4.2. Call Web Service through Wrapper
 - 4.3. *Use Graceful Exit* use case to handle errors in contacting servers
 - 4.4. Web Service Functionality executed

Alternative Courses:

None.

Extends:

None.

User Interfaces:

None.

Constraints:

None.

Questions:

None.

Notes:

Every effort has been made not to include details that obstruct a high level view of the scenario sequence. Details are provided in the diagrams below for this use case.

No alternative have been listed because of the pre-conditions stating that servers are properly functioning and that the receiver party has web services registered.

Graceful Exit Use Case Note :

The graceful exit use case is behavioral functionality that our system will implement to indicate an error to the user and stop transactions midway with a priority on placing the least amount of interruption and disturbance to the user. This use case is not expanded upon in this version of the SRS to conform with the deliverable requirements. The Graceful Exit use case of course, handles many exit scenarios.

Authors:**Source Documents:**

2.3 Use Case Diagrams

This section presents the business scenarios of the subject area in a graphical form.

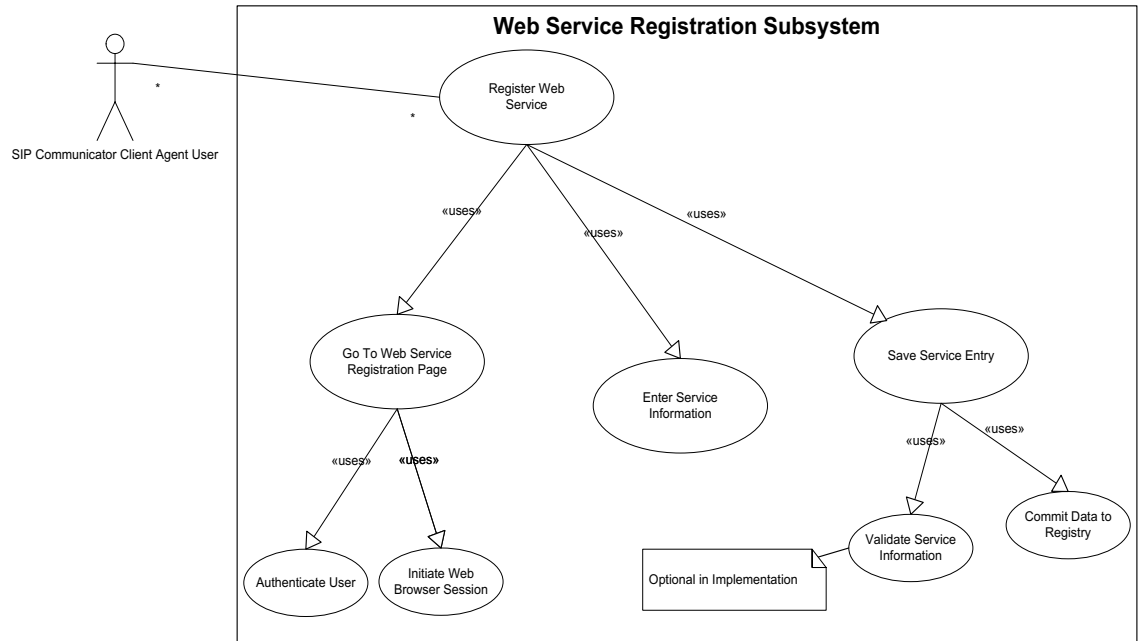


Fig. 2.3.1 Use Case Diagram for Scenario 3.2.1

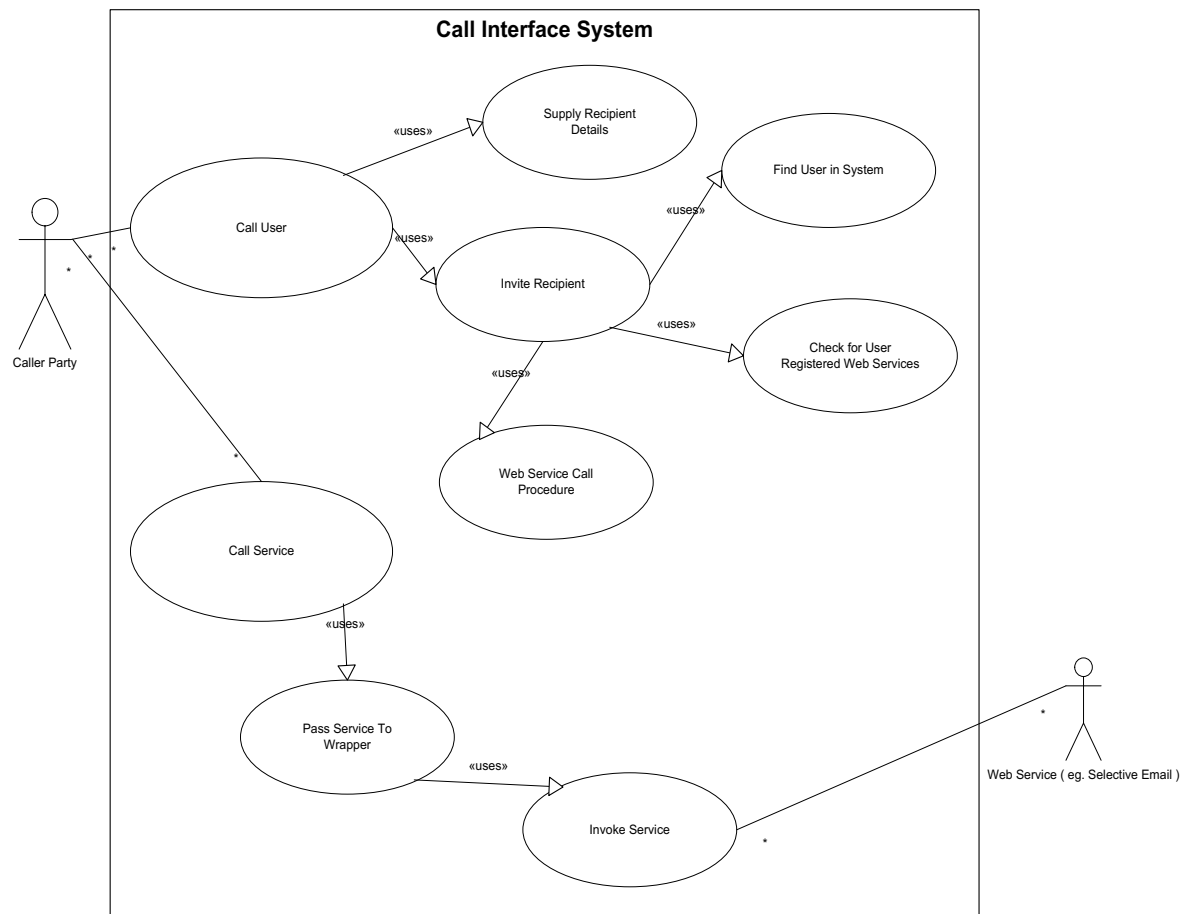


Fig. 2.3.2 Use Case Diagram for Scenario 3.2.2

3 Domain Model

3.1 Domain Model Class Diagram

The domain model class diagram for the SIP-based Communication System appears below:

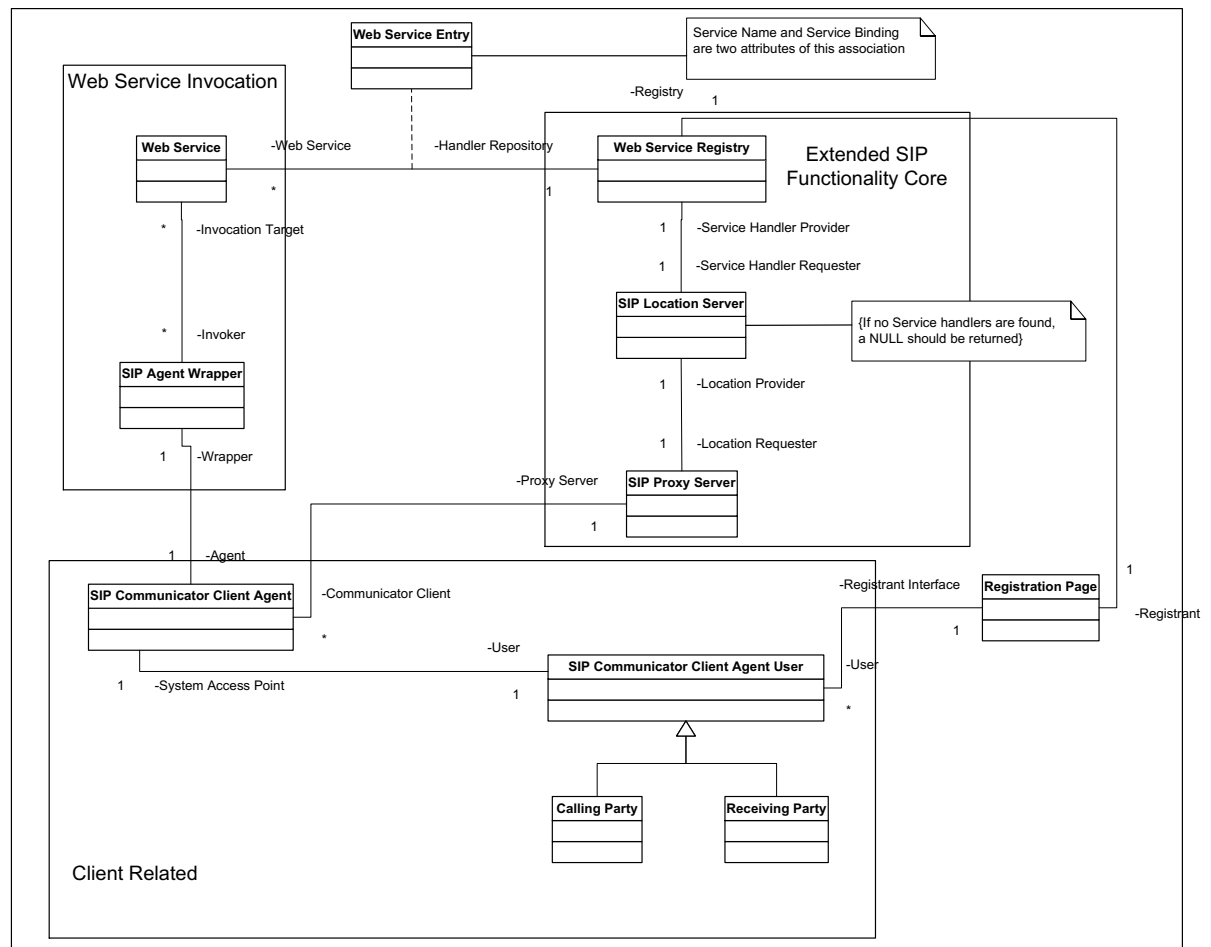


Fig. 3.1 Domain Model Class Diagram

3.2 Domain Model Class Definitions

3.2.1 Web Service

Description	This object represents the actual web service to be invoked during normal operation scenarios that involve the invocation of web services using the SIP-based Communication System. It is invoked by the SIP Agent Wrapper using corresponding description information found in the Web Service Registry for the object. The object is associated with the Web Service Registry via a Web Service Entry object.
Attributes	
Responsibilities	The web service object must be configured to allow invocation of the service by the SIP Agent Wrapper using the information provided in its corresponding Web Service Entry (i.e. as specified in the .wsdl file). The object is also responsible for returning an invocation result to the SIP Agent Wrapper.
Business Rules	

3.2.2 Web Service Entry

Description	This object represents an entry in the Web Service Registry corresponding to a Web Service registered by an Agent. It contains the name of the corresponding Web Service, as well the Service Binding containing an external link to the .wsdl description file for the service. It is an association class, serving as a link between a Web Service object and the Web Service Registry.
Attributes	
Responsibilities	A Web Service Entry object is responsible for providing information regarding the Web Service associated with it (as identified by a Service Name attribute) and a link to the .wsdl file (provided by a Service Binding attribute) that specifies the invocation procedure for the Web Service.
Business Rules	

3.2.3 SIP Agent Wrapper

Description	Each instance of this object is responsible for wrapping exactly one instance of a SIP Communicator Client Agent object. It serves as an interface between the Client Agent and any Web Service that the Agent may choose for invocation.
Attributes	
Responsibilities	This object is responsible for using invocation parameters from the Client Agent (i.e. which service to invoke, the location of the .wsdl file) to actually invoke a selected Web Service. It is also responsible for forwarding the result of this invocation back to the Agent once received.
Business Rules	

3.2.4 SIP Communicator Client Agent

Description	This object represents the SIP Communicator Client Agent software application that allows the SIP Communicator Client Agent User to interface with the rest of the SIP-based Communication System. This object associates with the rest of the system through the Proxy Server, and each instance is wrapped by an instance of the SIP Agent Wrapper.
Attributes	
Responsibilities	This object is responsible for using invocation parameters from the Client Agent (i.e. which service to invoke, the location of the .wsdl file) to actually invoke a selected Web Service. It is also responsible for forwarding the result of this invocation back to the Agent once received.
Business Rules	

3.2.5 SIP Communicator Client Agent User

Description	This object represents the user of SIP Communicator Client Agent software in the context of the domain. This is a human user that can perform general functionality not associated with the user's role as either a Calling Party or Receiving Party.
Attributes	
Responsibilities	This user is responsible for registering any required Web Services by adding them to the Web Service Registry using the Registration Page object as an interface. Specific roles of this user (i.e. Calling Party and Receiving Party) have additional responsibilities.

Business Rules	
-----------------------	--

3.2.6 Calling Party

Description	This object represents the role of the SIP Communicator Client Agent User when it interacts with the system as a party placing a call (i.e. VOIP call or Web Service invocation). It is generalized by the SIP Communicator Client Agent User, and interacts only with the SIP Communicator Client Agent Software in performing its role.
Attributes	
Responsibilities	This object is responsible for instructing the SIP Communicator Client Agent User to initiate a call (i.e. setting call details, dialing). It is also responsible for selecting a Web Service for invocation from the list of returned Web Service Handlers in scenarios that involve the normal invocation of any Web Service.
Business Rules	

3.2.7 Receiving Party

Description	This object represents the role of the SIP Communicator Client Agent User when it interacts with the system as a party receiving a call. It is generalized by the SIP Communicator Client Agent User, and interacts only with the SIP Communicator Client Agent Software in performing its role.
Attributes	
Responsibilities	This object is responsible for indicating to the SIP Communicator Client Agent User whether it will choose to accept or decline an incoming call.
Business Rules	

3.2.8 Web Service Registry

Description	Keeps track of Web Service handlers for Web Services registered by different organization (business entity). The Registration Page contacts the Registry to register or de-register any web services. The Location Server queries it to check whether a called agent has any registered Web Services.
Attributes	
Responsibilities	Stores all information about the services, service handlers and registry that correspond to the registered users
Business Rules	

3.2.9 SIP Location Server

Description	Provides location details of the called party. It is connected to the Proxy Server and the Web Service Registry. Queries about registered services by the Proxy Server are sent through to the Service Registry to return handlers to web services.
Attributes	
Responsibilities	Locates whether the called party has any services registered with the Service Registry. Receives request for Web Service handlers from the Proxy Servers and forwards the handler back to the Proxy Server from the Web Service Registry. If no handler is returned (i.e. the client has no Web Services registered), a NULL is returned.

Business Rules	
-----------------------	--

3.2.10 SIP Proxy Server

Description	Help route requests to the user's current location, authenticate and authorize users for services, implement call-routing policies, and provide features to users.
Attributes	
Responsibilities	Contacts the Location Server to authorize user for services. Locate user and services. Provide call-routing procedures
Business Rules	

3.2.11 Registration Page

Description	A web interface which allows user to register, de-register services. User has direct access to it. It is also connected to the Web Services Registry to retrieve and modify Web Service information.
Attributes	
Responsibilities	Authorizes users to login and look for their registered services. Allows users to register a service through the Web Service Registry. Allows users to de-register a service through the Web Service Registry.
Business Rules	

4 Interaction Diagrams

4.1 Sequencing Diagrams

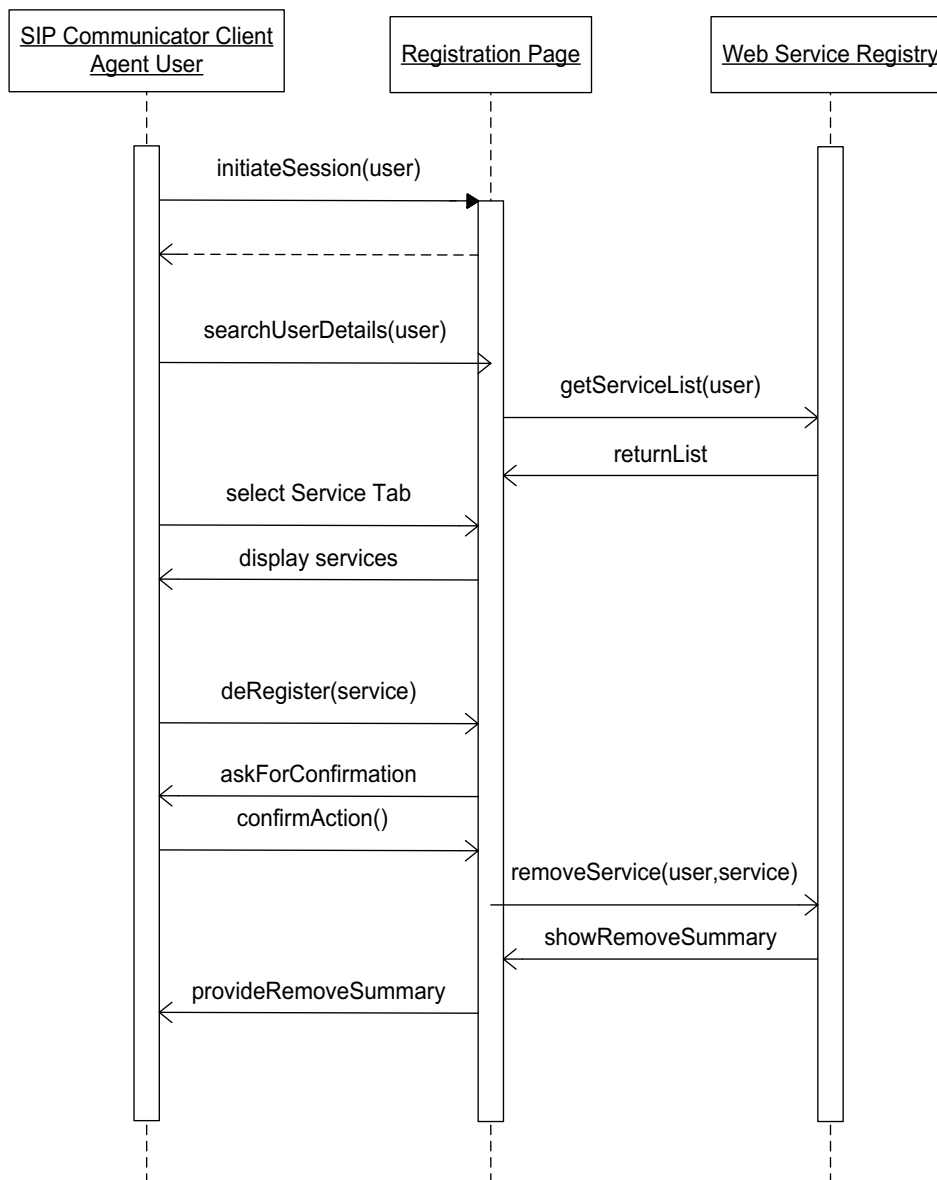


Fig. 4.1.1 Sequencing Diagram for Scenario 3.1.3

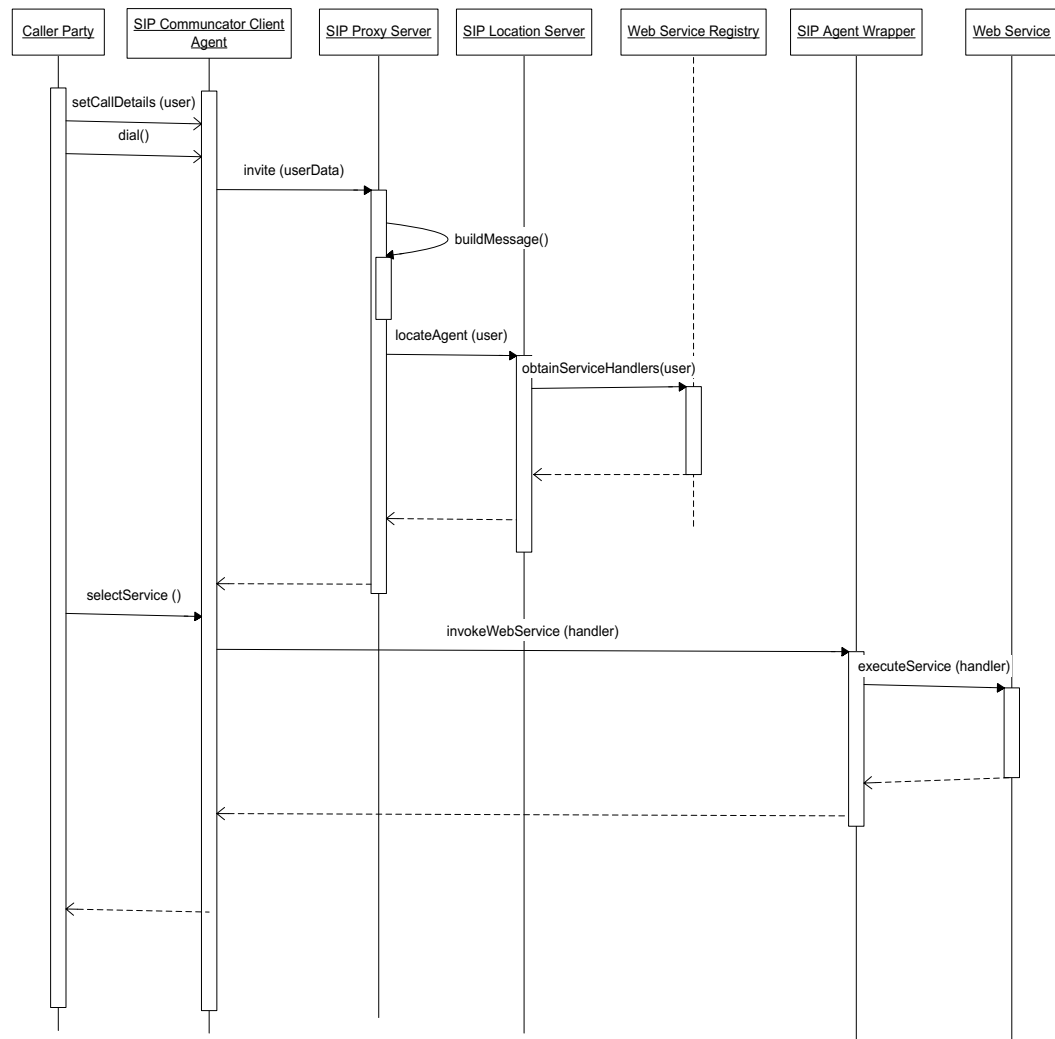


Fig. 4.1.2 Sequencing Diagram for Scenario 3.2.2

4.2 Collaboration Diagrams

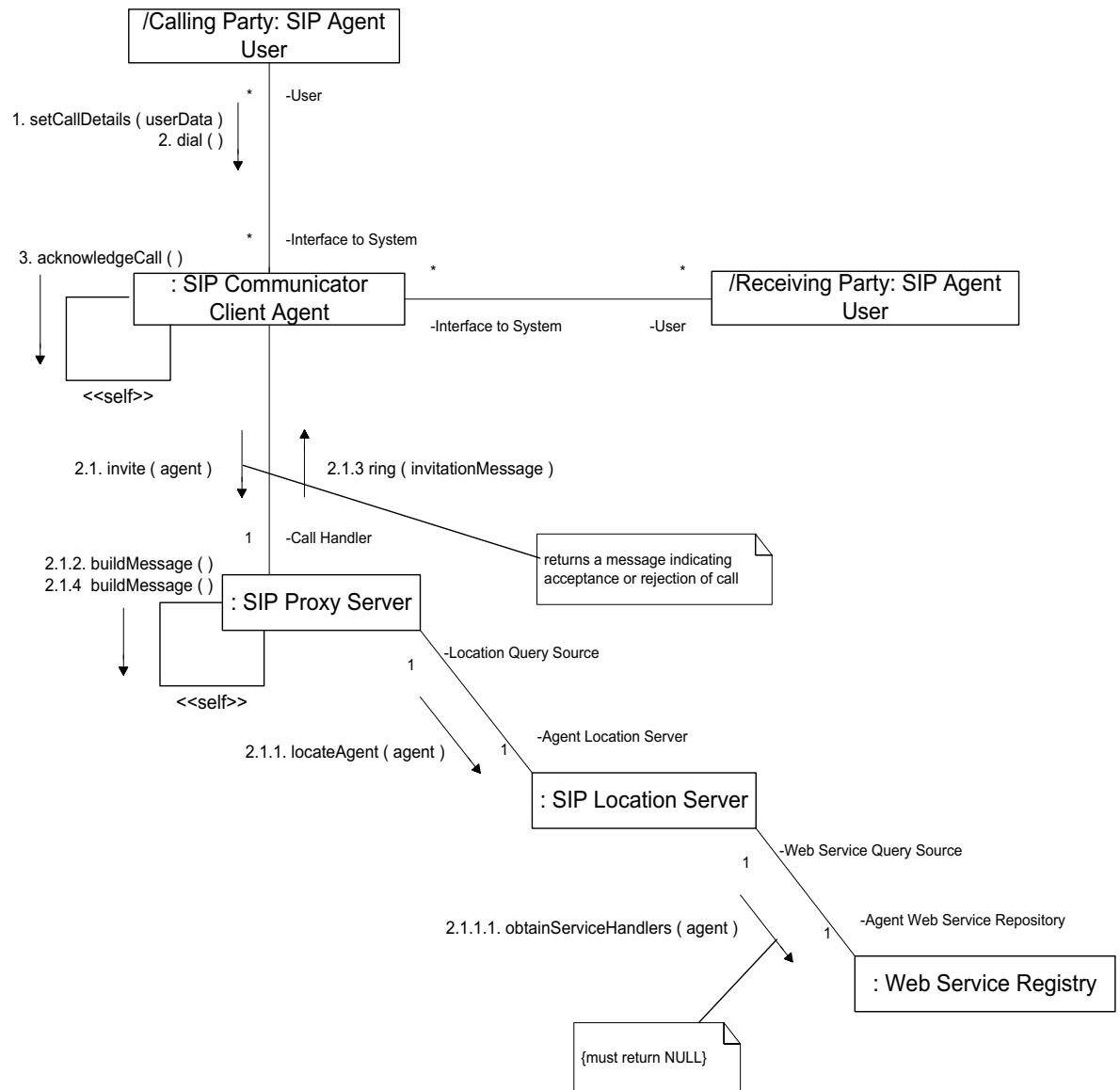


Fig. 4.2.1 Collaboration Diagram for Scenario 3.1.2

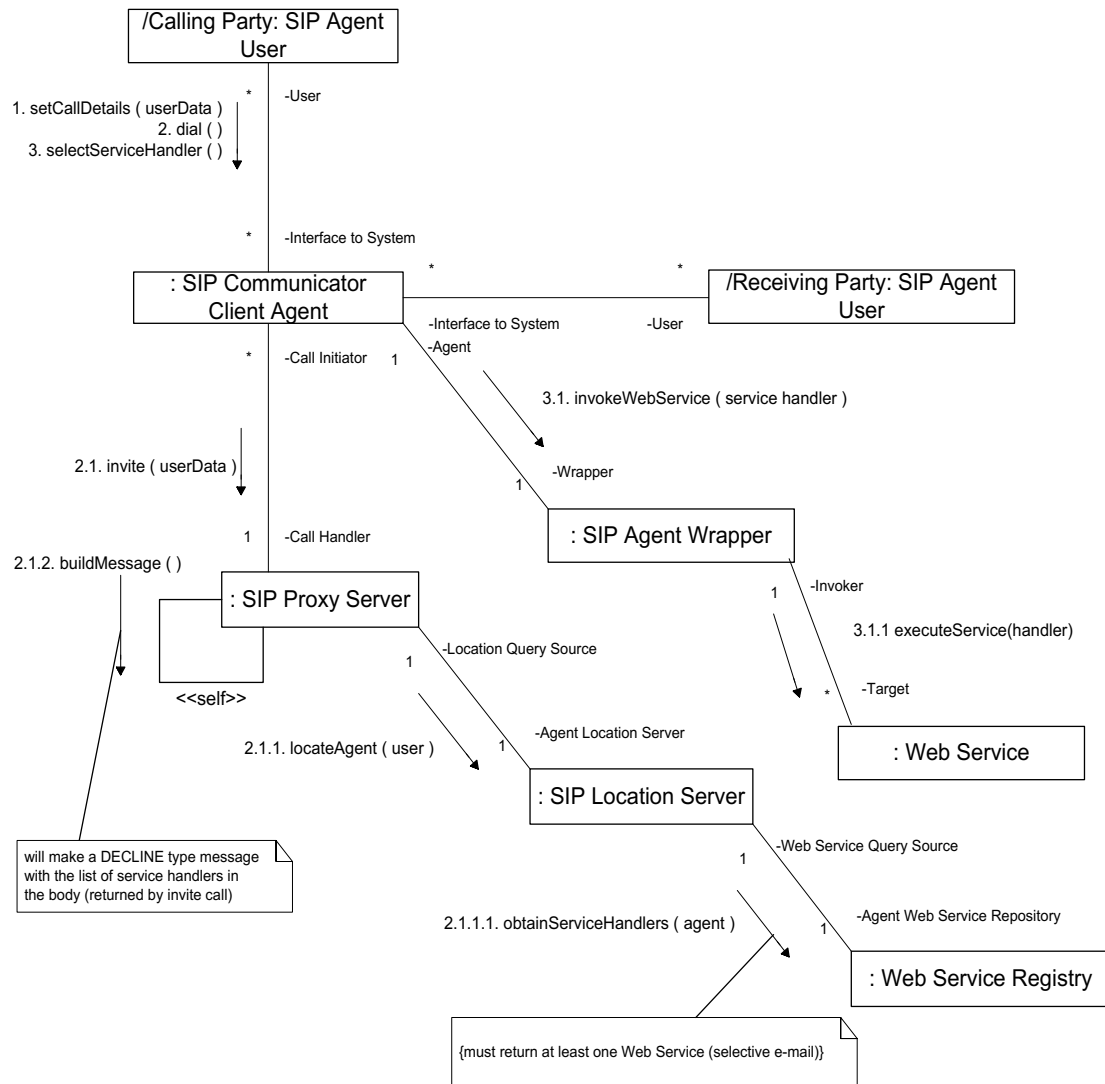


Fig. 4.2.2 Collaboration Diagram for Scenario 3.2.2

5 Non-Functional Requirements Specification

5.1 Overview

The non-functional requirements of the system comprises of utilities, environments and other specifications that are necessary for the smooth operation of the system as a whole. This includes interfaces, development environment, capacity specifications, network and operational parameters.

5.2 Enabling Technologies

5.2.1 Target Development Environment

The system should be developed in a Windows environment and using Java. Eclipse 3.1 will be the Integrated Development Environment (IDE) used for coding purposes.

5.2.2 System Interfaces

Sample code to query the service registry using the JAXR API will be provided. This will be modified to enable the location server to query the service registry. The proxy server will be modified to accept service descriptions from the registry and to compose message body strings to pass on to the client. A wrapper class, which uses WSIF or JAX-RPC API to invoke the web services that has been selected by the client agent UI, would be created. The client agent UI will be provided.

5.3 Capacity Planning

5.3.1 Permanent Storage

In order to be reliable RAID 1 mirroring will be used on the server. This ensures that if one of the disks fails the other disk functions as a single hard drive until the faulty one is replaced. For system installation and activity logging, the disk size used will be 30 GB. Most of that space would be occupied by client information such as number of web services registered, names of the web services and their descriptions, the .wsdl files and user activity logging.

5.4 Network

The system should have connectivity to the Internet with bandwidth sufficient enough to carry audio signals. Furthermore, the network should possess efficient packet switching protocols such as MultiProtocol Layer Switching (MPLS) or Voice Over IP (VoIP) that will transmit the signals without skips or delays.

5.5 Workstations

The minimum system requirements and configurations for the computers used for the development, deployment and execution of the system are:

A hard disk space of 3GB to install Windows, the SIP-Communicator software and a Java Virtual Machine. A processor speed of 300 MHz and memory of 64 MB is sufficient. A display setting of 1024 x 768 resolution and a 16-bit color palette should be used. The workstation should also include a sound card with microphone and speaker jacks and its driver installed. Headphone with an attached microphone is recommended. The computer should have connectivity to a LAN with bandwidth

high enough to carry voice data.

5.6 Operational Parameters

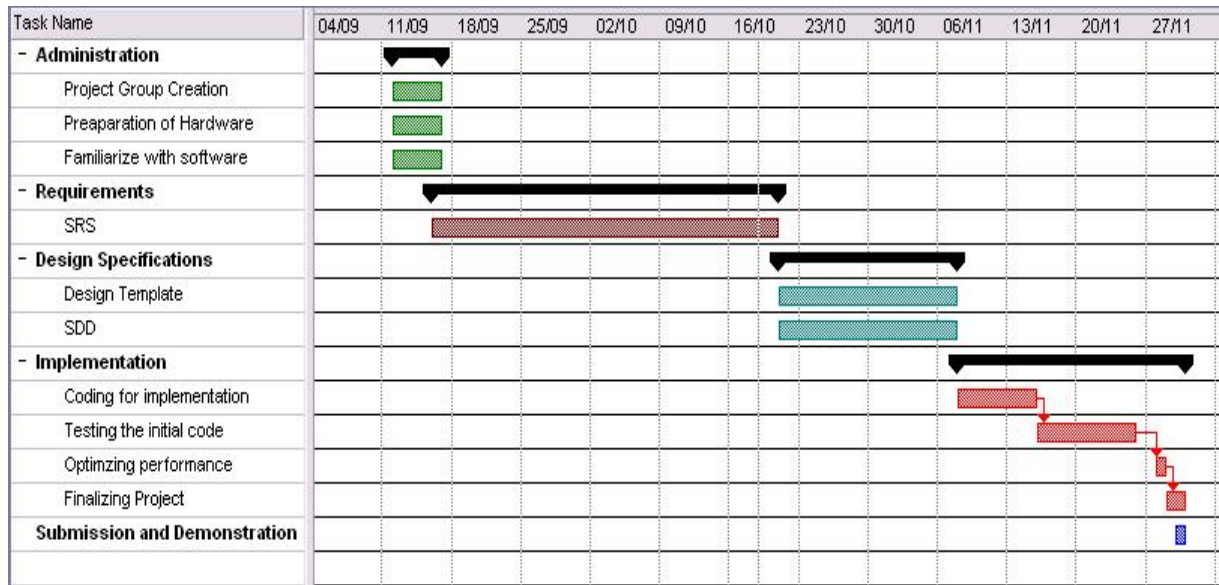
5.6.1 Usability

The system should be learnable and usable by the average computer user who has exposure to VoIP software. The GUI will be designed in such a way that the buttons are conspicuous and meaningful making it easy to access all the functions of the system.

5.6.2 Reliability

The backup and recovery functions of the system will consist of snapshots which will be taken at regular intervals and stored somewhere external to the system. The Mean Time To Repair should be minimized which will in turn minimize Mean Time Between Failures. The system should always be available to the user.

6 Activities Plan



7 Domain Dictionary

All terminology used in this document has been described either in this document or in a referred document. Please consult reference documentation for additional details (see Section 1.3).