

Facoltà di Ingegneria - Università di Catania

Smart Intelligent University Communications

Software Engineering Projects

Relatore
Prof. Orazio Tomarchio

Studenti
Russo Leandro
Invincibile Daniele
Didomenico Nicola

Dipartimento di Ingegneria Elettrica, Elettronica e Informatica – DIEEI
Facoltà di Ingegneria Informatica

27 Febbraio 2015



Indice

1. Indice

2. Ideazione - Iterazione 1

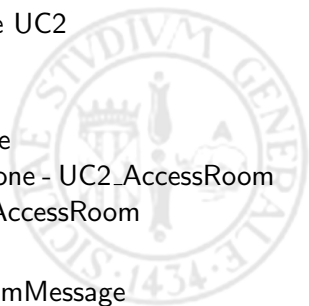
- 2.1. Iterazione 1: Requisiti
- 2.2. Iterazione 1: Analisi - UC1_RequestConnection
- 2.3. Iterazione 1: Analisi - UC2_AccessRoom
- 2.4. Iterazione 1: Progettazione
- 2.5. Iterazione 1: Implementazione - UC1 e UC2
- 2.6. Iterazione 1: Test - UC1 e UC2

3. Ideazione: Refactoring Iterazione 1

- 3.1. Iterazione 1 Refactoring: Progettazione
- 3.2. Iterazione 1 Refactoring: Implementazione - UC2_AccessRoom
- 3.3. Iterazione 1 Refactoring: Test - UC2_AccessRoom

4. Elaborazione - Iterazione 2

- 4.1. Iterazione 2: Requisiti - UC3_SendRoomMessage





Indice

- 4.2. Iterazione 2: Analisi
- 4.3. Iterazione 2: Progettazione
- 4.4. Iterazione 2: Implementazione - UC3_SendRoomMessage
- 4.5. Iterazione 2: Test - UC3_SendRoomMessage

5. Elaborazione - Iterazione 3

- 5.1. Iterazione 3: Requisiti - UC4_SendPrivateMessage
- 5.2. Iterazione 3: Analisi
- 5.3. Iterazione 3: Progettazione
- 5.4. Iterazione 3: Implementazione - UC4_SendPrivateMessage
- 5.5. Iterazione 3: Implementazione - UC4_SendPrivateMessage



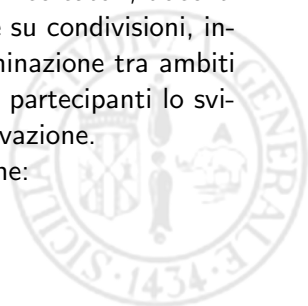
Descrizione realtà d'interesse

Nel Piano Nazionale della Ricerca (PNR) messo a punto da HORIZON ITALIA e MIUR viene bandito un progetto chiamato:

“Smart Intelligent University Communications (SIUC)” destinato a tutti gli atenei italiani.

Tale progetto è caratterizzato dalle seguenti specifiche descritti nel seguito, che consente ad appassionati, studenti, ricercatori, docenti e imprese di scambiarsi messaggi in tempo reale su condivisioni, integrazioni e competenze di idee per una contaminazione tra ambiti disciplinari diversi, realtà diverse e stimolare nei partecipanti lo sviluppo della cultura dell'intraprendere e dell'innovazione.

In tale sistema si richiedono le seguenti specifiche:



Descrizione realtà d'interesse

- Esistono diversi canali o stanze virtuali (ciascuna legata a un corso di laurea per ogni facoltà, includendo sia la triennale e la magistrale di quel determinato corso di laurea) nelle quali un utente può entrare per scambiare messaggi con gli altri utenti presenti nella stessa stanza.
- È possibile inviare due tipi di messaggi:
 - ▶ “pubblici”: dei messaggi inviati da un utente e trasmessi a tutti gli altri partecipanti presenti nella stanza;
 - ▶ “privati”: dei messaggi inviati da un utente e trasmessi ad uno specifico partecipante presente nella stessa stanza, in questo caso il destinatario selezionato sarà l'unico a ricevere il messaggio.
- In ogni istante il sistema prevede la presenza di una tipologia di utente particolare, chiamato amministratore. I compiti più importanti di un amministratore sono i seguenti:



Descrizione realtà d'interesse

- ▶ può creare o eliminare stanze del servizio di messaggistica;
 - ▶ inviare messaggi di avviso ad un utente;
 - ▶ in caso di comportamenti irregolari è possibile espellere un utente dalla stanza e l'utente non può più rientrare fin quando questo non sarà rimosso dalla lista dei partecipanti bannati della stanza presente nel sistema.
- ▶ Il sistema deve mandare dei messaggi di notifica, che permettono di aggiornare l'utente di un cambiamento dello stato del sistema.



Iterazione 1: Requisiti - Documento di Visione

La caratteristica principale del progetto è quella di consentire ad appassionati, studenti, ricercatori, docenti e imprese di scambiarsi messaggi in tempo reale per ogni ateneo. In particolare in questo progetto sono presenti diverse stanze virtuali che rappresentano le facoltà di un ateneo con due tipologie di utilizzatori di sistema, ovvero l'amministratore e gli utenti del servizio di messaggistica. Con le seguenti caratteristiche:

- l'amministratore gestisce le stanze virtuali e supervisiona gli utenti del servizio;

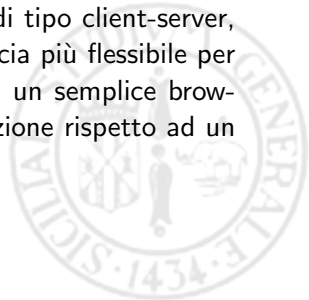




Iterazione 1: Requisiti - Documento di Visione

- gli utenti per usufruire di tale servizio inseriscono un nickname e si collegano al server impostando dei parametri di connessione. Ogni utente può accedere a una stanza e inviare messaggi ad ogni utente presente nella stanza in tempo reale, inoltre ognuno può contattare in maniera privata gli altri partecipanti presenti nella stanza.

L'architettura utilizzata per offrire il servizio è di tipo client-server, questo approccio consente di fornire un'interfaccia più flessibile per l'accesso al servizio di messaggistica anche con un semplice browser, senza modificare pesantemente la progettazione rispetto ad un architettura peer-to-peer.



Iterazione 1: Requisiti - Diagrammi dei casi d'uso

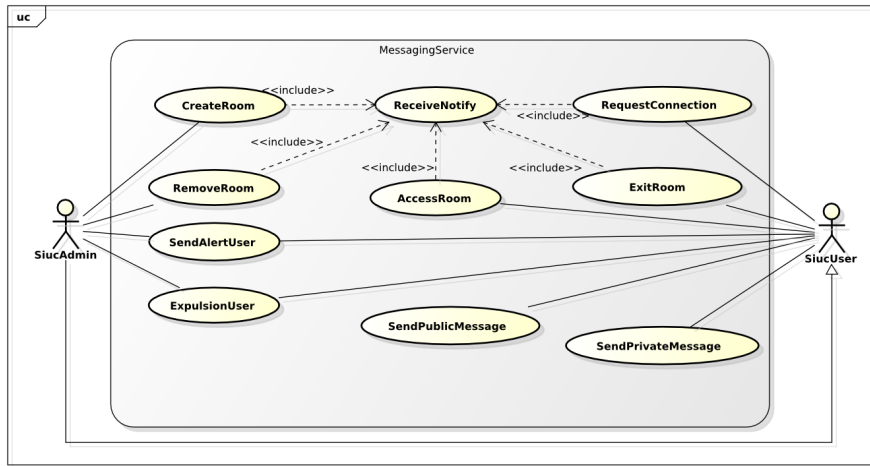


Figura 1 : Diagrammi dei casi uso

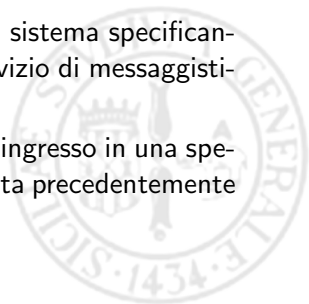
Iterazione 1: Requisiti - Modello dei casi d'uso

Attori Identificati: Amministratore (SiucAdmin) e Utente del Servizio di Messaggistica (SiucUser).

Casi d'uso identificati: RequestConnection, AccessRoom, SendRoomMessage, SendPrivateMessage, ReceiveNotify, ExitRoom, CreateRoom, RemoveRoom, SendAlertUser, ExpulsionUser

Descrizione breve dei casi d'uso identificati:

- 1 RequestConnection. L'utente si connette al sistema specificando il nickname, l'indirizzo e la porta del servizio di messaggistica.
- 2 AccessRoom. L'utente richiede al sistema l'ingresso in una specifica stanza del servizio (che deve essere stata precedentemente creata dall'amministratore del servizio).





Iterazione 1: Requisiti - Modello dei casi d'uso

- 3 SendRoomMessage. Il partecipante al servizio di messaggistica invia un messaggio “pubblico” che viene inviato dal sistema a tutti i partecipanti che si trovano nella stessa stanza di colui che ha inviato il messaggio.
- 4 SendPrivateMessage. Il partecipante al servizio di messaggistica invia un messaggio “privato” ad uno specifico partecipante.
- 5 ReceiveNotify. Il sistema manda dei messaggi di notifica, che permettono di aggiornare l'utente di un cambiamento dello stato del sistema.
- 6 ExitRoom. Un partecipante richiede al sistema l'uscita dal servizio. Questo comporta la sua eliminazione dall'insieme dei partecipanti presenti nella stanza del servizio cui era precedentemente associato l'utente.

Iterazione 1: Requisiti - Modello dei casi d'uso

- 7 CreateRoom. L'amministratore del servizio di messaggistica è responsabile della creazione (preventiva) delle stanze che potranno successivamente essere visitate dai partecipanti.
- 8 RemoveRoom. In ogni momento l'amministratore può eliminare una stanza dal servizio (ad esempio perchè non ci sono partecipanti). Ciò comporta l'invio preventivo di un messaggio a tutti i partecipanti eventualmente presenti nella stanza, i quali potranno in seguito richiedere l'ingresso in una nuova stanza del servizio.
- 9 SendAlertUser. L'amministratore può in ogni momento inviare un messaggio di avviso ad uno specifico partecipante al servizio.
- 10 ExpulsionUser. L'amministratore può in ogni momento espellere da una stanza uno specifico partecipante.

Glossario

- ▶ SiucUser, User, Utente: è l'utente del servizio di messaggistica che usufruisce dei servizi forniti.
- ▶ SiucAdmin, Admin, Amministratore: è il gestore del servizio di messaggistica, che possiede la facoltà di creare e/o eliminare una stanza, di inviare un particolare messaggio di avviso ad un partecipante per un comportamento non corretto ed eventualmente di bannarlo eliminandolo dalla stanza.
- ▶ Messaging Service, Servizio di messaggistica, sistema, chat: sistema che si occupa della gestione.
- ▶ Room, stanza, canale: insieme che si può differenziare dalle altre in base alle tematiche trattate.
- ▶ Messaggio Pubblico: messaggio testuale inviato da un partecipante a tutti gli utenti presenti nella stanza.



Glossario

- ▶ **Messaggio Privato:** messaggio testuale inviato da un partecipante ad un particolare partecipante presente nella stanza.
- ▶ **Notifica:** particolari messaggi di avviso inviati dal server agli utenti del servizio.



Iterazione 1: Requisiti, caso d'uso UC1_RequestConnection

Tabella 1 : Descrizione dettagliata: caso d'uso UC1_RequestConnection

| | |
|--|--|
| Nome caso d'uso | UC1_RequestConnection |
| Portata | Applicazione Smart Intelligent University Communications |
| Livello | Obiettivo Utente |
| Attore primario | SiucUser |
| Parti interessate e interessi | SiucUser: vuole collegarsi al servizio di messaggistica |
| Pre-condizioni | L'utente ha bisogno di una connessione di rete |
| Post-condizioni (garanzia di successo) | L'utente è inserito tra gli utenti online con il nickname confermato dal servizio di messaggistica ottenendo un messaggio di benvenuto |
| Scenario principale di successo | <ol style="list-style-type: none"> ➊ L'utente inserisci un nickname, l'address e la porta del server. ➋ Il sistema esamina il nickname inviato dall'utente e verifica se è presente una omonimia. ➌ Il sistema conferma l'inserimento tra gli utenti online inviando una notifica di benvenuto. |

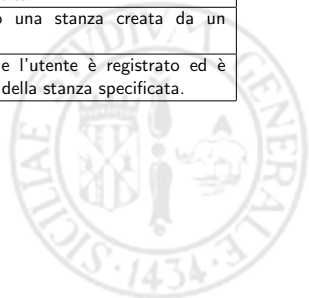
Iterazione 1: Requisiti, caso d'uso UC1_RequestConnection

| | |
|---|---|
| Estensioni (o flussi alternativi) | <p>1A - SiucUser inserisce parametri errati:</p> <ul style="list-style-type: none"> ▶ Viene visualizzato un messaggio di errore e viene richiesto nuovamente l'inserimento di tali parametri. <p>2A - Omonimia del nickname:</p> <ul style="list-style-type: none"> ▶ Il sistema cambia il nickname aggiungendo _ al nickname (es. _nickname). ▶ Il sistema conferma l'inserimento tra gli utenti online inviando una notifica di benvenuto. |
| Requisiti speciali (Requisiti Non Funzionali) | Comunicazione asincrona in cui lo scambio di informazioni avviene in tempo reale, senza sensibili pause tra invio e ricezione del messaggio. |
| Elenco delle varianti tecnologiche | L'applicazione dovrebbe essere flessibile al funzionamento di diversi protocolli di comunicazione (es. TCP, UDP) e con diversi strati middleware (es. Socket, RMI) |
| Frequenza di ripetizione | Potrebbe essere quasi ininterrotta |
| Varie e/o Problemi Aperti | // |

Iterazione 1: Requisiti, caso d'uso UC2_AccessRoom

Tabella 2 : Caso d'uso UC2_AccessRoom

| | |
|--|---|
| Nome caso d'uso | UC2_AccessRoom |
| Portata | Applicazione Smart Intelligent University Communications |
| Livello | Obiettivo Utente |
| Attore primario | SiucUser |
| Parti interessate e interessi | SiucUser: vuole registrarsi e effettuare l'ingresso in una stanza presente nel servizio di messaggistica |
| Pre-condizioni | Nel sistema è presente almeno una stanza creata da un amministratore. |
| Post-condizioni (garanzia di successo) | Nel caso di svolgimento normale l'utente è registrato ed è presente nell'insieme degli utenti della stanza specificata. |



Iterazione 1: Requisiti, caso d'uso UC2_AccessRoom

| | |
|---|--|
| Scenario principale di successo | <ol style="list-style-type: none"> ❶ L'utente richiede una lista di stanze presenti nel sistema di messaggistica. ❷ Il sistema invia la lista delle stanze. ❸ L'utente seleziona la stanza tra quelle presenti in lista. ❹ Il sistema registra l'utente alla stanza. ❺ Il sistema visualizza a video gli utenti presenti nella stanza. ❻ Il sistema visualizza un'area pubblica dove vengono mostrati tutte le conversazioni in corso dal quel momento in poi. |
| Estensioni (o flussi alternativi) | 3A - Il SiucUser inserisce una stanza non in elenco: <ol style="list-style-type: none"> ❶ Il sistema invia un messaggio di errore. |
| Requisiti speciali (Requisiti Non Funzionali) | Comunicazione asincrona in cui lo scambio di informazioni avviene in tempo reale, senza sensibili pause tra invio e ricezione del messaggio. |
| Elenco delle varianti tecnologiche | L'applicazione dovrebbe essere flessibile al funzionamento di diversi protocolli di comunicazione (es. TCP, UDP) e con diversi strati middleware (es. Socket, RMI) |
| Frequenza di ripetizione | Potrebbe essere quasi ininterrotta |
| Varie e/o Problemi Aperti | // |



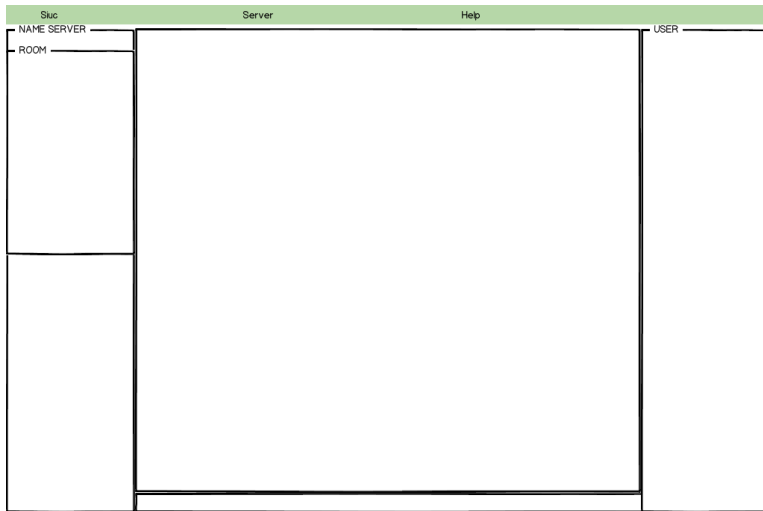
Iterazione 1: Requisiti, descrizione Mockups User Interface

INSERIRE DESCRIZIONE

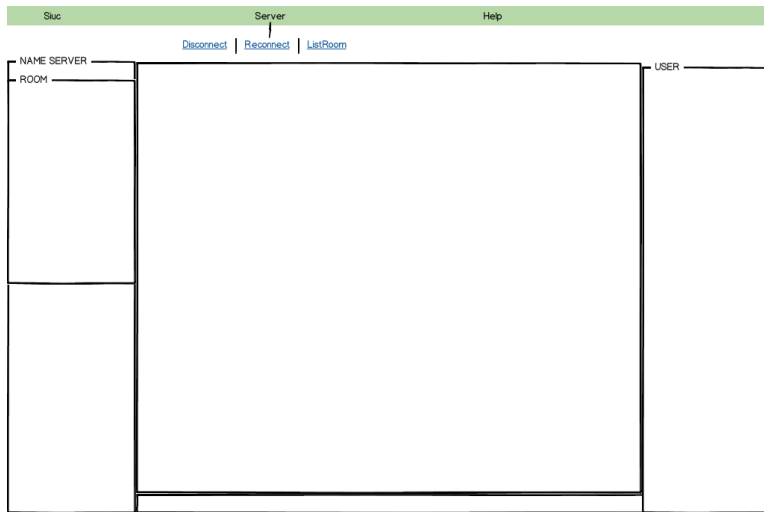




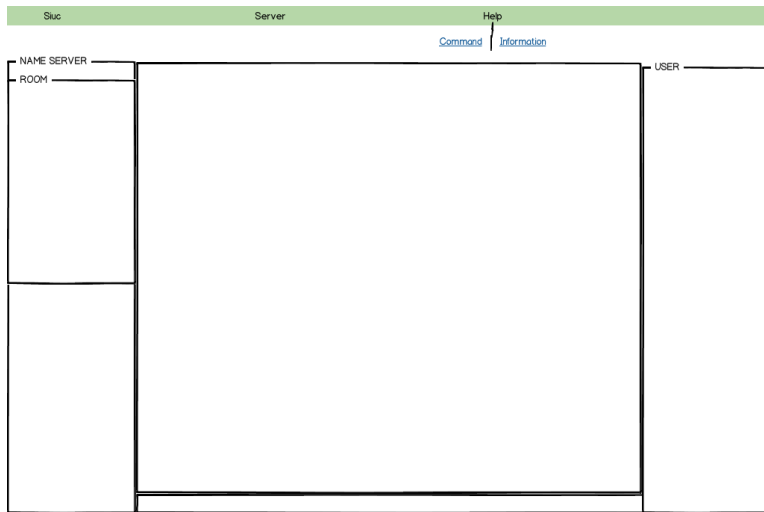
Iterazione 1: Requisiti, Mockups Bozza User Interface



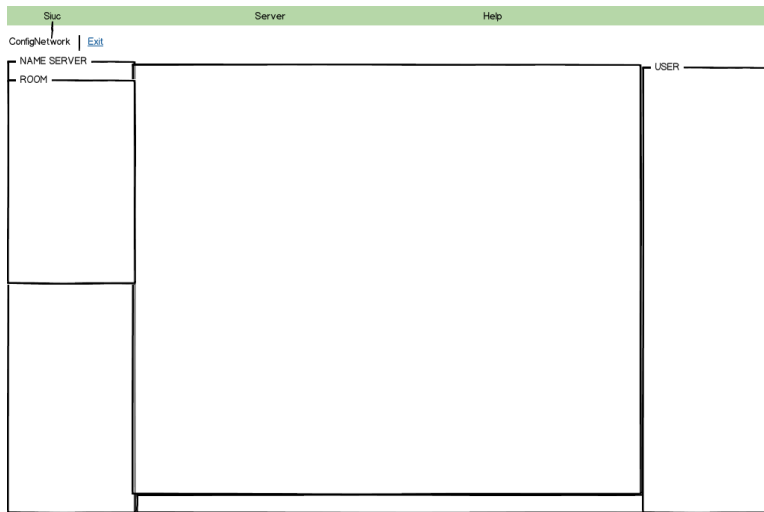
Iterazione 1: Requisiti, Mockups Bozza User Interface



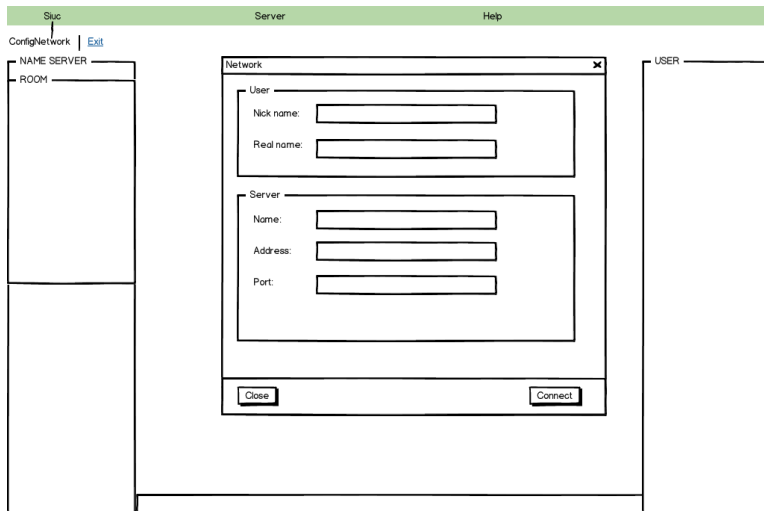
Iterazione 1: Requisiti, Mockups Bozza User Interface



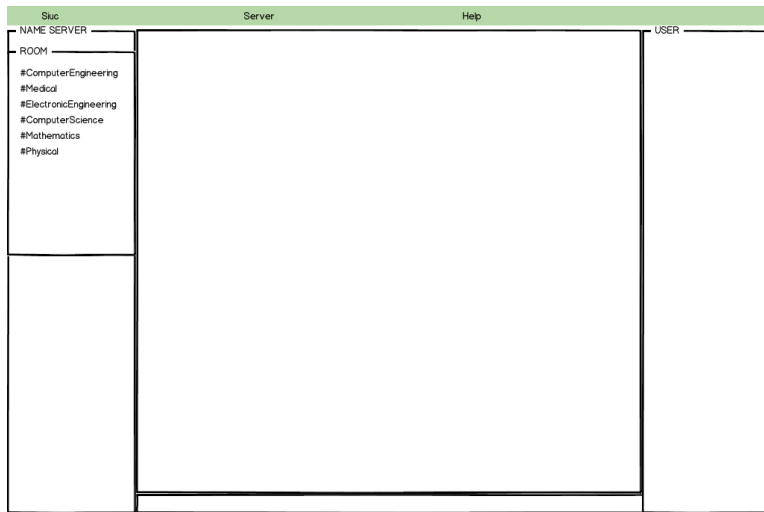
Iterazione 1: Requisiti, Mockups Bozza User Interface



Iterazione 1: Requisiti, Mockups Bozza User Interface

A wireframe mockup of a user interface. At the top is a green header bar with three menu items: "Sluc", "Server", and "Help". Below the header, on the left, is a vertical sidebar with two sections. The top section is labeled "NAME SERVER" and contains a "ROOM" label. The bottom section is empty. To the right of the sidebar is a large central area. In the top-left corner of this area are two links: "ConfigNetwork" and "Exit". In the center is a window titled "Network" with a close button (X) in the top-right corner. The "Network" window has two main sections: "User" and "Server". The "User" section contains two labels, "Nick name:" and "Real name:", each followed by a text input field. The "Server" section contains three labels, "Name:", "Address:", and "Port:", each followed by a text input field. At the bottom of the "Network" window are two buttons: "Close" and "Connect". To the right of the central area is a vertical sidebar with a "USER" label at the top and a large empty space below.

Iterazione 1: Requisiti, Mockups Bozza User Interface

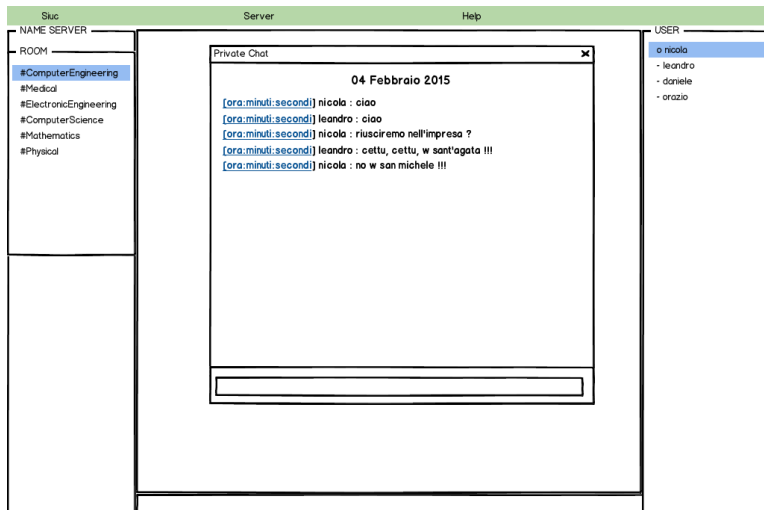


Iterazione 1: Requisiti, Mockups Bozza User Interface

| Sluc | Server | Help | USER |
|------------------------|--|------|------------|
| NAME SERVER | | | |
| ROOM | | | |
| #ComputerEngineering | | | o nicola |
| #Medical | | | - leandro |
| #ElectronicEngineering | | | - danielle |
| #ComputerScience | | | - orazio |
| #Mathematics | | | |
| #Physical | | | |
| | <p>19 Gennaio 2015</p> <p>[ora:minuti:secondi] nicola : ciao</p> <p>[ora:minuti:secondi] leandro : funziona ?</p> <p>[ora:minuti:secondi] danielle : bellissima</p> <p>[ora:minuti:secondi] orazio : è un ottimo modello UP...</p> | | |



Iterazione 1: Requisiti, Mockups Bozza User Interface



Iterazione 1: Requisiti, Mockups Bozza User Interface

| Stuc | Server | Help |
|------------------------|---|------------------|
| NAME SERVER | | USER |
| ROOM | 19 Gennaio 2015 | o user_medical 1 |
| #ComputerEngineering | [ora:minuti:secondi] user_medical 1 : ragazzi questi informatici ne sanno una più del dia | - user_medical 2 |
| #Medical | [ora:minuti:secondi] user_medical 2 : si si... | - user_medical 3 |
| #ElectronicEngineering | [ora:minuti:secondi] user_medical 3 : assurdo | - user_medical 4 |
| #ComputerScience | [ora:minuti:secondi] user_medical 4 : oggi c'è la lezione di anatomia ?? | |
| #Mathematics | | |
| #Physical | | |





Descrizione Iterazione 1: Analisi - UC1_RequestConnection

INSERIRE DESCRIZIONE



Iterazione 1: Analisi - UC1_RequestConnection

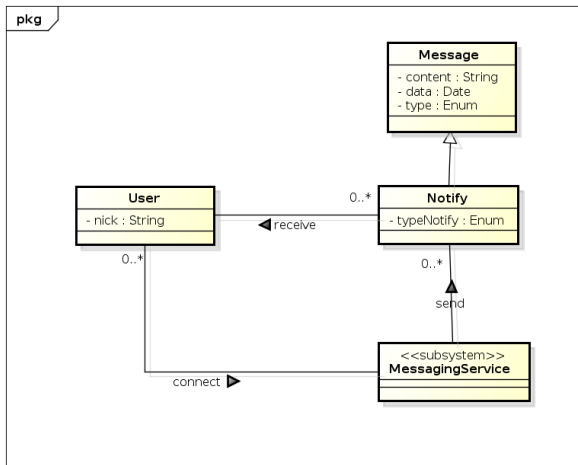


Figura 2 : UC1 - Modello di dominio

Iterazione 1: Analisi - UC1_RequestConnection

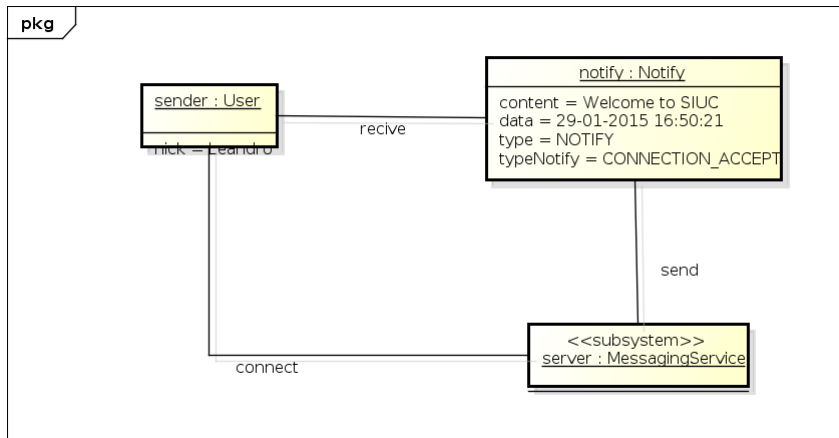


Figura 3 : UC1 - Oggetti di dominio

Iterazione 1: Analisi - UC1_RequestConnection

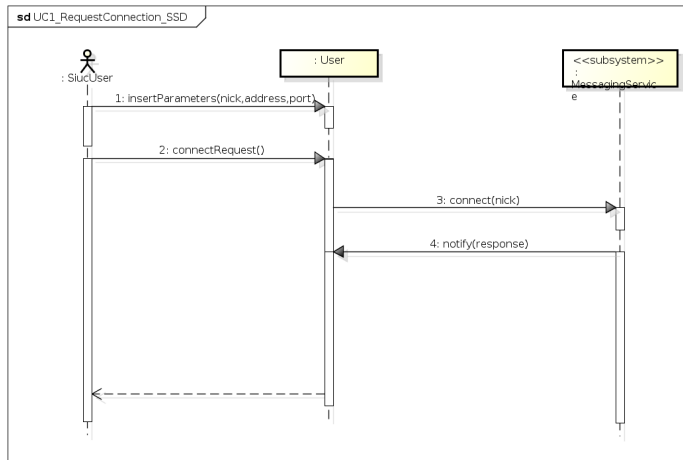


Figura 4 : UC1 - Diagramma di sequenza di sistema



Iterazione 1: Analisi, UC1 contratto CO1 - NameCO1

Tabella 3 : UC1 Contratto CO1 - NameCO1

| | |
|-----------------|--|
| Operazione | |
| Riferimenti | |
| Pre-condizione | |
| Post-condizione | |





Descrizione Iterazione 1: Analisi - UC2_AccessRoom

INSERIRE DESCRIZIONE



Iterazione 1: Analisi - UC2_AccessRoom

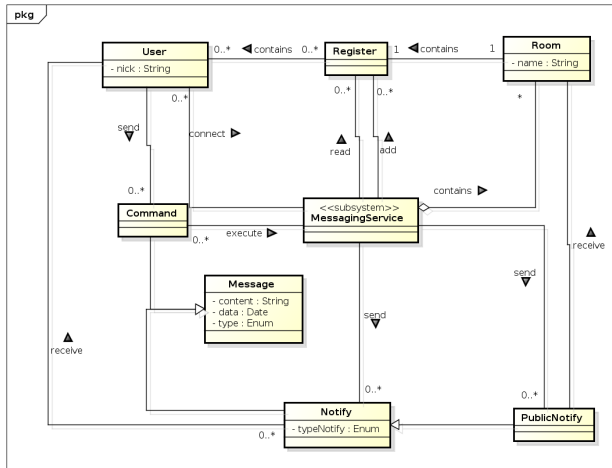


Figura 5 : UC2 - Modello di dominio

Iterazione 1: Analisi - UC2_AccessRoom

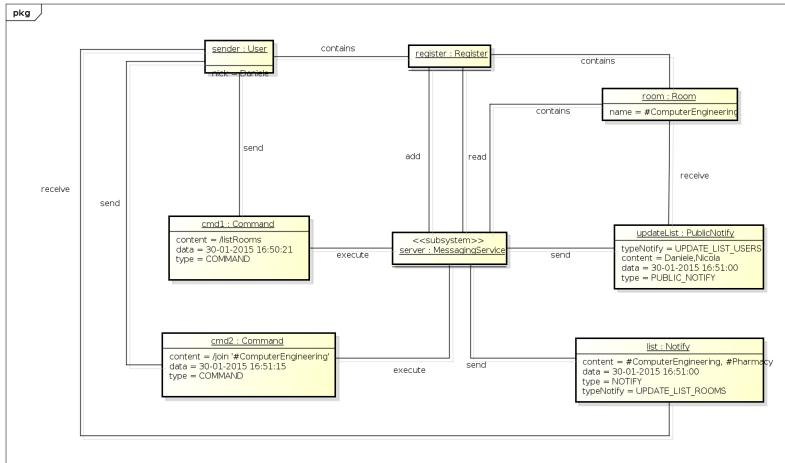


Figura 6 : UC2 - Oggetti di dominio

Iterazione 1: Analisi - UC2_AccessRoom

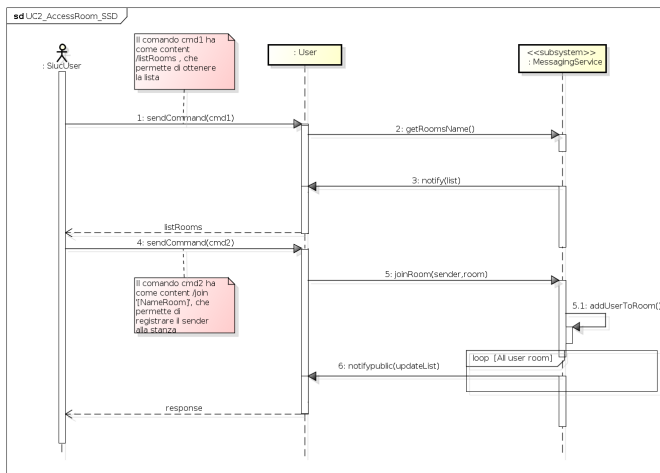


Figura 7 : UC2 - Diagramma di sequenza di sistema



Iterazione 1: Analisi, UC2 contratto CO2 - NameCO2

Tabella 4 : UC2 Contratto CO2 - NameCO2

| | |
|-----------------|--|
| Operazione | |
| Riferimenti | |
| Pre-condizione | |
| Post-condizione | |





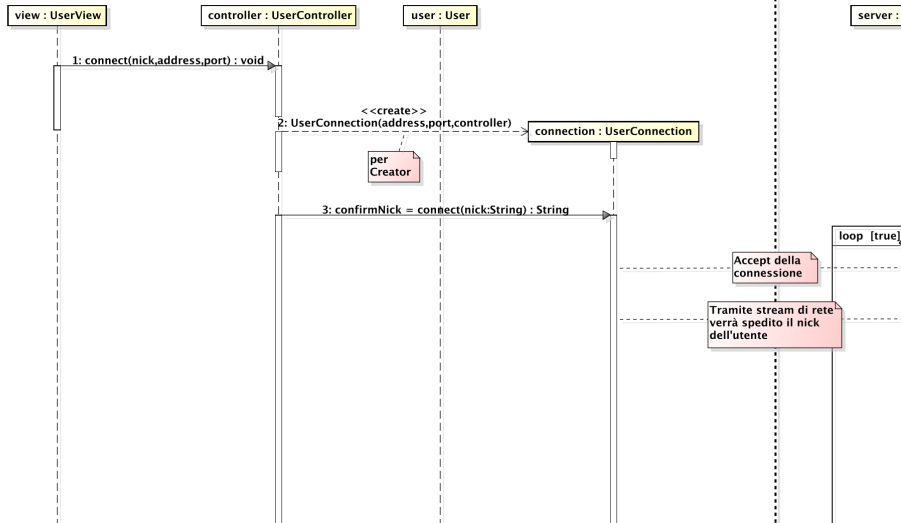
Descrizione Iterazione 1: Progettazione

INSERIRE DESCRIZIONE



Iterazione 1 Progettazione: UC1_RequestConnection Connect

sd UC1_RequestConnection_SSD_3_4: connect





Iterazione 1: Implementazione - UC1 e UC2

INSERIRE DESCRIZIONE





Iterazione 1: Test - UC1 e UC2

INSERIRE DESCRIZIONE



Descrizione Ideazione: Refactoring Iterazione 1

INSERIRE DESCRIZIONE





Iterazione 1 Refactoring: Progettazione

INSERIRE DESCRIZIONE





Iterazione 1 Refactoring: Implementazione - UC2_AccessRoom

INSERIRE DESCRIZIONE





Iterazione 1 Refactoring: Test - UC2_AccessRoom

INSERIRE DESCRIZIONE





Elaborazione: Iterazione 2

INSERIRE DESCRIZIONE



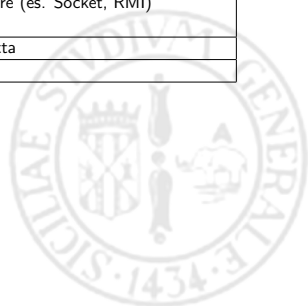
Iterazione 2: Requisiti - UC3_SendRoomMessage

Tabella 5 : Caso d'uso UC3_SendRoomMessage

| | |
|--|---|
| Nome caso d'uso | UC3_SendRoomMessage |
| Portata | Applicazione Smart Intelligent University Communications |
| Livello | Obiettivo Utente |
| Attore primario | SiucUser |
| Parti interessate e interessi | <ul style="list-style-type: none"> ▶ SiucUser, vuole che i messaggi siano inviati a ogni utente della stanza virtuale. ▶ SiucAdmin, è interessato a supervisionare gli utenti del servizio affinché non ci siano abusi. |
| Pre-condizioni | L'utente è registrato nella stanza in cui desidera inviare i messaggi. |
| Post-condizioni (garanzia di successo) | Ogni utente riceve il messaggio inviato. |
| Scenario principale di successo | <ol style="list-style-type: none"> 1 L'utente inserisce in una opportuna area il messaggio da inviare. 2 Il messaggio viene inoltrato agli utenti presenti nella stanza selezionata. |

Iterazione 2: Requisiti - UC3_SendRoomMessage

| | |
|---|--|
| Requisiti speciali (Requisiti Non Funzionali) | Comunicazione asincrona in cui lo scambio di informazioni avviene in tempo reale, senza sensibili pause tra invio e ricezione del messaggio |
| Elenco delle varianti tecnologiche | <ul style="list-style-type: none"> ▶ È possibile inviare messaggi confidenziali, autenticati e integri al server del servizio di messaggistica. ▶ L'applicazione dovrebbe essere flessibile al funzionamento di diversi protocolli di comunicazione (es. TCP, UDP) e con diversi strati middleware (es. Socket, RMI) |
| Frequenza di ripetizione | Potrebbe essere quasi ininterrotta |
| Varie e/o Problemi Aperti | // |





Iterazione 2: Analisi

INSERIRE DESCRIZIONE





Iterazione 2: Progettazione

INSERIRE DESCRIZIONE



Iterazione 2: Implementazione - UC3_SendRoomMessage

INSERIRE DESCRIZIONE



Iterazione 2: Test - UC3_SendRoomMessage

INSERIRE DESCRIZIONE





Descrizione Elaborazione: Iterazione 3

INSERIRE DESCRIZIONE



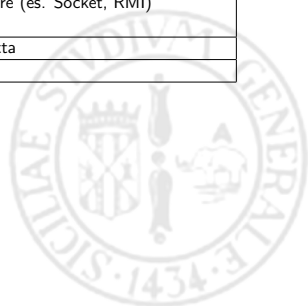
Iterazione 3: Requisiti - UC4_SendPrivateMessage

Tabella 6 : Caso d'uso UC4_SendPrivateMessage

| | |
|--|---|
| Nome caso d'uso | UC3_SendRoomMessage |
| Portata | Applicazione Smart Intelligent University Communications |
| Livello | Obiettivo Utente |
| Attore primario | SiucUser |
| Parti interessate e interessi | SiucUser: vuole che i messaggi siano inviati all'utente selezionato presente nella stanza virtuale |
| Pre-condizioni | L'utente è registrato nella stanza in cui desidera inviare un messaggio ad un altro utente presente |
| Post-condizioni (garanzia di successo) | L'utente selezionato riceve il messaggio inviato |
| Scenario principale di successo | <ol style="list-style-type: none"> ① L'utente seleziona il destinatario del messaggio privato. ② L'utente inserisce da tastiera il messaggio da inviare. ③ Il messaggio viene inoltrato al destinatario selezionato. |

Iterazione 3: Requisiti - UC4_SendPrivateMessage

| | |
|---|--|
| Requisiti speciali (Requisiti Non Funzionali) | Comunicazione asincrona in cui lo scambio di informazioni avviene in tempo reale, senza sensibili pause tra invio e ricezione del messaggio |
| Elenco delle varianti tecnologiche | <ul style="list-style-type: none"> ▶ È possibile inviare messaggi confidenziali, autenticati e integri al server del servizio di messaggistica. ▶ L'applicazione dovrebbe essere flessibile al funzionamento di diversi protocolli di comunicazione (es. TCP, UDP) e con diversi strati middleware (es. Socket, RMI) |
| Frequenza di ripetizione | Potrebbe essere quasi ininterrotta |
| Varie e/o Problemi Aperti | // |





Iterazione 3: Analisi

INSERIRE DESCRIZIONE





Iterazione 3: Progettazione

INSERIRE DESCRIZIONE



Iterazione 3: Implementazione - UC4_SendPrivateMessage

INSERIRE DESCRIZIONE



Iterazione 3: Test - UC4_SendPrivateMessage

INSERIRE DESCRIZIONE



Riferimenti bibliografici



Martin Fowler, L. Baresi, S. Gaburri

UML distilled. Guida rapida al linguaggio di modellazione standard
Pearson Education Italia, 2010



Craig Larman, Luca Cabibbo

Applicare UML e i pattern: analisi e progettazione orientata agli oggetti
Pearson Education Italia, 2005



M.L. Liu

Distributed Computing: Principles and Applications
Addison-Wesley, 2003



Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides

Design patterns - Elementi per il riuso di software ad oggetti
Pearson Education Italia, 1995

