

**Name:** Shirley Montero Quesada

**Project:** OTR-Messenger

**Requirements:** Users could be computer illiterate as well as a little savy. The requirements were written to complete the phrase: "As a <user> I need to <task> to <accomplish my goal>".

Table I1. User requirements selected for this report

Requirement	<user>	<task>	<accomplish my goal>	priority
UR-001	client	sign up	to create an account and access chat service	High
UR-002	client	log in	to access the chat service	High
UR-003	client	send messages	to communicate with others	Critical
UR-004	client	receive messages	to communicate with others	Critical
UR-005	client	view friends list	to see who is online	High

### Use Case Document(s):

Table I2. Use Case UR-001 (sign up) documentation.

Use Case ID:	UR-001		
Use Case Name:	sign up		
Description:	client can create a new account by entering an username and a password on the signup window		
Actors:	client		
Pre-Conditions:	There is an user interface where the user can signup		
Post-Conditions:	Client has an account		
Frequency of Use:	Daily		
Flow of Events:		Actor Action	System Response
	1	Activates username field by typing an username on it	
	3	Activates password field by typing a password on it	
	4	Enters password	Verifies username does not exist, password is compliant, Generates master seed
Variations:	None		
Notes and Issues:	Password should comply with BR-001		
Developer Notes:			

Table 13. Use Case UR-002 (log in) documentation.

Use Case ID:	UR-002		
Use Case Name:	log in		
Description:	Client can access the service by logging in		
Actors:	client		
Pre-Conditions:	User has signed up previously and has username/password. There is an interface for client log in.		
Post-Conditions:	client has access to service		
Frequency of Use:	Daily		
Flow of Events:		Actor Action	System Response
	1	Activates username field by typing username on it	
	2	Activates password field by typing a password on it	
	3	Enters password	Verifies username does not exist, password is compliant, Generates master seed
Variations:			
Notes and Issues:			
Developer Notes:			

Table 14. Use Case UR-003 (send messages) documentation.

Use Case ID:	UR-003		
Use Case Name:	send messages		
Description:	to communicate with others		
Actors:	client		
Pre-Conditions:	there is a friends list and a text editing window, client is logged in to system which generates private and public keys		
Post-Conditions:	message is sent to recipient		
Frequency of Use:			
Flow of Events:		Actor Action	System Response
	1	Click recipient	Open chatwindow
	2	Type message	
	3	Click send	move message to display window, encrypt and transmit (request other user's pub encrypt key, encrypt message under shared key, sign message with user's priv key, transmit Message object (encrypted text, time and tag)), draw check by message when succesful
Variations:	1.If recipient not in contact list, then the user must enter the recipient's username		
Notes and Issues:	The chatwindow has two sections. The top one is the display window and the bottom is an editing window.		
Developer Notes:			

Table 15. Use Case UR-004 (receive messages) documentation.

Use Case ID:	UR-004		
Use Case Name:	receive messages		
Description:	to communicate with others		
Actors:	client		
Pre-Conditions:	client is logged in and connected to server, ready to receive messages. Server has received message and it is ready to transmit		
Post-Conditions:	client has received message		
Frequency of Use:	often - whenever someone sends a message to a user		
Flow of Events:		Actor Action	System Response
	1	other user sends a message (UR-003)	server forwards message to receiving client.
Variations:			
Notes and Issues:	actor receiving message does not take any direct actions. The client is listening for messages over an open SSLSocket. When transmitted, the message is then displayed to user in chat window		
Developer Notes:			

Table 16. Use Case UR-005 (view friends list) documentation.

Use Case ID:	UR-005		
Use Case Name:	view friends list		
Description:	allow client to see friends that he has added to his contact list, and which are online		
Actors:	client		
Pre-Conditions:	User has signed up previously and has username/password. There is an interface for client log in.		
Post-Conditions:	client has a viewable window that shows other users that the user has added to their contact list and shows which are online		
Frequency of Use:	every log in, often		
Flow of Events:		Actor Action	System Response
	1	Log in via interface	establishes the connection with the server and retrieves the information to populate the friends list on GUI
Variations:	1. in case the client has no contacts yet, the same GUI allows them to add them (see UR-006)		
Notes and Issues:			
Developer Notes:			

### Activity Diagram:

The following diagram is for use case UR-003 (send messages). Although the user probably doesn't know what goes on with the system once it clicks the send button, I chose to specify it a little bit more as help for the sequence diagram used further below and definitely for the class diagram that we designed for our group report.

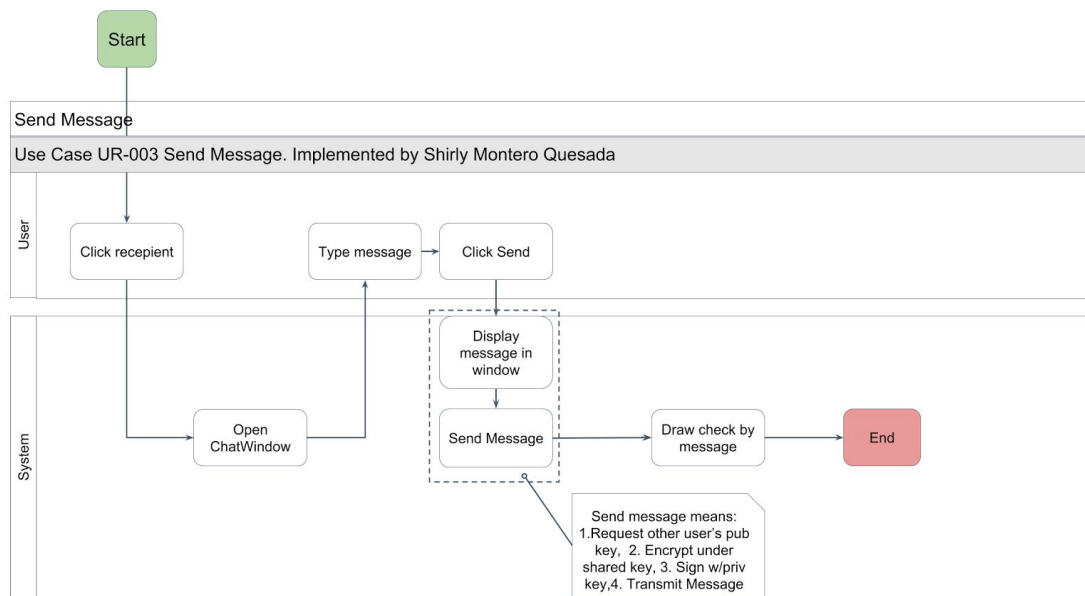


Figure I1. Activity diagram for use case UR-003 Send Message.

## User Interactions

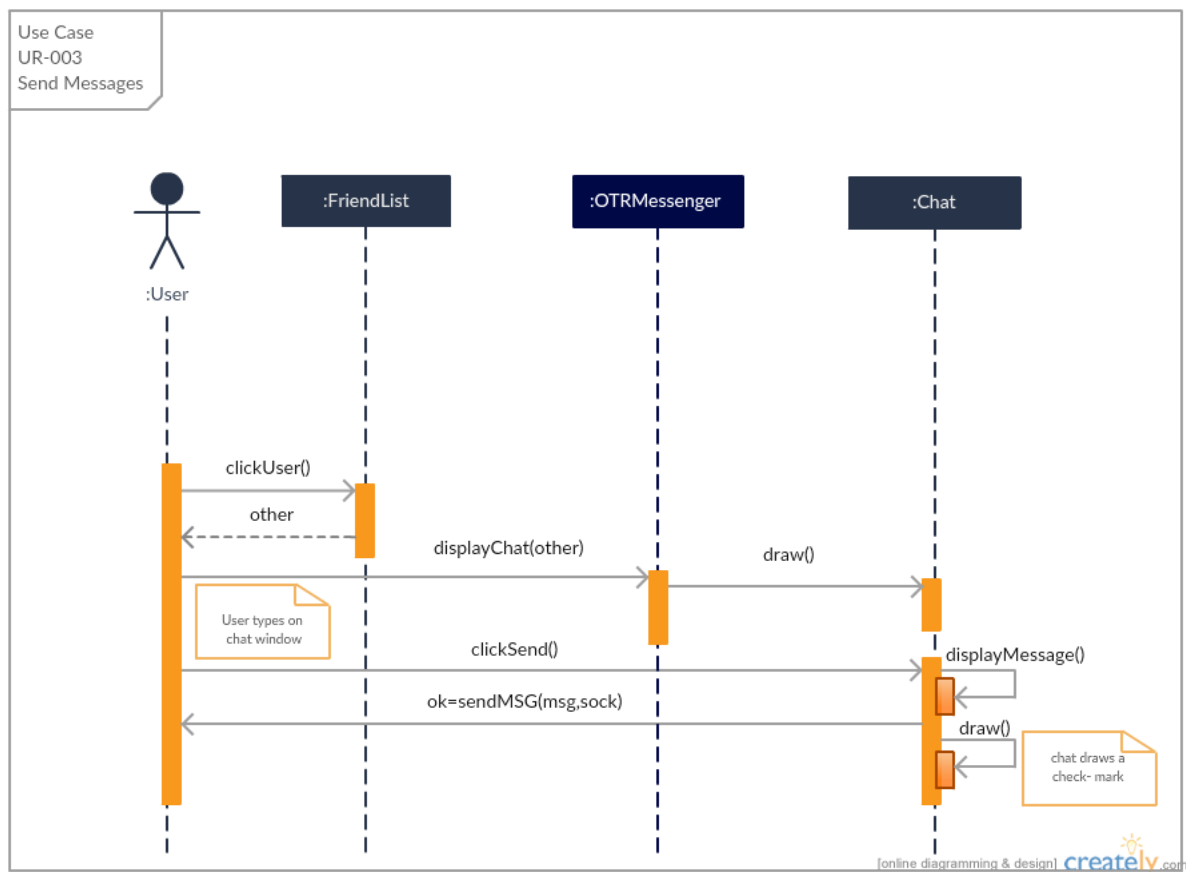


Figure I2. Sequence diagram for use case UR-003 Send Message.