

Workshop 1

ANDRES CAMILO RAMOS ROJAS – 20242020005

Project:

Outlook (message app with a close environment)

Objectives:

- The user will registrant his self with an identificatory
- The user will see all the other users that exist in the system
- The user will be able to send messages to other users

Requirements:

- Functional requirements:
 - User register
 - User login
 - User identification
 - User communication
 - User update (was added with human error in mind)
 - User elimination (was added for users that leave the system)
 - User logout
 - Message management
 - Archive management
- Non - functional requirements:
 - Multiple users simultaneously support
 - Outlook-like interface
 - Standard security parameters to password defining
 - Unique identifiers (usernames)

User stories:

Title: user register	Priority: high	Estimate:
User story: As a user , I want to register my self, so that I can send messages to other users.		
Acceptance criteria: Given a user When he register her self Then he will be able to sing in		

Title: user access	Priority: High	Estimate
User story:		

As a user, I want to be able to sing in, so that I will have acces to the system.
Acceptance criteria: Given 10 users When they acces to the system Then will be able to send a message

Title: user identification	Priority: high	Estimate:
User story: As a user, I want have an identifier , so that my familiars, acquaintances and another users recognize me on the system.		
Acceptance criteria: Given hundreds of users When they belong to the system Then they will be able yo recognizes each other easily		

Title: user communication	Priority: high	Estimate;
User story: As a user, I want to send messages to the other users when I need to, so that I can communicate with other people.		
Acceptance criteria: Given hundreds users When they send a message to other user Then the second user will have the message in her messages		

Title: user update	Priority: high	Estimate;
User story: As a user, I want to update my profile, so that If I make a mistake when registering I can correct it.		
Acceptance criteria: Given a user When he update his information Then he will se the changes		

(If a user want to change his password for security reasons, also if there was an error in the register)

Title: user disable	Priority: high	Estimate;
User story: As an administrator, I want to disable profiles, so that I can manage how many users the system have		
Acceptance criteria:		

Title: user log out	Priority: High	Estimate
User story:		

As a user, I want to be able to log out, so that I will disconnect from the system
Acceptance criteria: Given 10 users When they log out Then they will be able to access again

(If there is a user without use the administrator will be able to eliminate this user and it wont be able to login and receive messages)

Title: message query	Priority: high	Estimate:
User story: As a user, I want to see the messages the people send me, so that I can stay in contact with them.		
Acceptance criteria: Given hundred users When the user receive a message Then the user will be able to see al the messages he received		

Title: chats differentiation	Priority: high	Estimate:
User story: As a user, I want a menu of important chats , so that I can I can differentiate between important chats and those that are not		
Acceptance criteria: Given 10 chats When the user mark a chat as important Then he will be able to see faster the more important messages		

Title: archive management	Priority: low	Estimate
User story: As a user, I want to be able to attach archives, so that I can send photos or reports to other people.		
Acceptance criteria: Given a chat When the user attach an archive Then the other user will see and be able to open the archive		

Title:	Priority:	Estimate
User story: As a user, I want to change the color of the interface , so that if I'm used to dark interfaces		
Acceptance criteria: Given [how things begin] When [action taken]		

Then [outcome of taking action]

Mockups:

<https://www.figma.com/design/MdKn3vvMIK9hrLO47QKTft/Untitled?node-id=2-127&t=jVvi57LOTBsAzuqb-1>

CRC cards:

Class: user	
Responsibility: <ul style="list-style-type: none"> ○ Register ○ Sing in ○ Log out ○ Update ○ elimination ○ Send messages ○ Read messages 	Collaborator: <ul style="list-style-type: none"> ○ Message ○ Chat

Class: Message	
Responsibility: <ul style="list-style-type: none"> ○ Create ○ Consult ○ update 	Collaborator: <ul style="list-style-type: none"> ○ User ○ Chat

Class: Chat	
Responsibility: <ul style="list-style-type: none"> ○ Create ○ Consult ○ Clean chat 	Collaborator: <ul style="list-style-type: none"> ○ User ○ Chat

All the crc card have changed to be equal to the classes diagram, the responsibilities as the methods of each class