
1. Introduction

1.1. Introduction

Imagine a world without a social media application, well, the universe will not collapse without the social media and people will not die if it has vanished, But the advantage of the social media that would really destroy huge businesses and disassemble the enormous networking connections among people.

However, "social media" are defined as any Web sites, portals or applications that allow individuals to post and share content publicly, which allow other individuals to view, respond and share this content further. These media include digital-based applications such as public blogging and micro-blogging; music, image, audio and video sharing; review and opinion sharing; gaming and entertainment; location-based, event-based and occupation-based networks; information and news aggregation; presentation sharing; advocacy and fundraising; creation of virtual worlds; and all-purpose wide-appeal networks geared toward general audiences.[1]

Moreover, before 1999 when the social media applications were not well recognized by the majority of people it would not really matter the idea of having a real connection via the internet even theoretical because of the social media complexity which people cannot absorb, and also because of the explosion of mobile phone then that made people think that they are in fantasy world for the reason that they can talk to the someone with a wireless mobile device, But now after the penetration of the social media applications in our lives, we actually cannot give it away under any circumstances; that's why it is a must to develop social media applications that supply their demands and strengthen their networking bonds by allowing them to do mostly everything they want with a single click.

For context, as of October 2017, total worldwide population is 7.6 billion and The internet has 3.5 billion users[2], and that emphasizes the importance of the social media applications in our live, as a matter of fact not all of them or even the half of them are individual end users accounts, they can be brands retails and small businesses marketing accounts, There are 3.03 billion active social media users 91% of retail brands use 2 or more social media channels, and 81% of all small and medium businesses use some kind of social platform [2].

Internet users have an average of 7.6 social media accounts Social media users grew by 121 million between Q2 2017 and Q3 2017.

1.2. Problem Definition

1.2.1. Introduction

Basically people by nature are seeking for the better life standards, they want easier life with more options while feeling free to choose any of them.

However, a lot of social media applications are developed, and by many I mean almost thousands of them, but in this application we would be developing a better way for communication that makes it easy for the users to use the application in various situations, we are here supplying the demands of users more than the available applications to rise their ability to communicate and to express themselves under most of circumstances even if the user is driving and cannot use the keypad or key touch to communicate this application offers the recording facility, in this application we offer the users friends' organization into circles to guarantee the user's privacy and to have them freely expressing without any fear, also it includes the parental monitor for the children to calm the parents concerns' when having their kids using the application, they will guarantee the safe environment for them.

1.2.2. Limited Expressivity

if the user is driving then he will be able to focus on writing the post and look for the letters in the key touch while he got to be driving at the same time he would wait until he stops by some place and finish driving then create his own post or talk to his friend or else he would have a terrible accident I'm most cases which will sacrifice his own life and might sacrifice other additional people because of his confusion between posting and driving, even if no accidents will happen he might be accused by traffic violation.

Moreover, if the user wants to express by some kind of way that shows a lot of details in the voice recording while the writing and the emojis are not helping him, also if the user is bored of writing or having some sort of illness that does not allow him to write properly.

This problem also arises when you are in an emergency situation where there are not enough patience to write and you are in hurry to inform what you intend to for the specified people, so in the normal case you will probably must write the post because the recording capability is not available there or record a voice note to one of his friends privately in messages - where this is available in some applications - and that one writes his post, and this is obviously will not be convenient for the nowadays users

Basically, the feeling that the user gets when he has limited expressively option will gradually decrease his satisfaction degree of the application being used, and the options limiting might get him bored of using this application and consider it an unreliable one that is not helpful in various situations.

1.2.3. Parental Monitor Missing

The emotional, social and physical development of young children has a direct effect on their overall development and on the adult they will become. That is why understanding the need to invest in very young children is so important, so as to maximize their future well-being, that's why keeping an eye on children is a must by their parents.

However, if there is no parental monitor for the child, the child can easily corrupted either by the outrageous website, or the violence in the surrounding environment, or even the inappropriate age contents.

The previous inappropriate contexts mentioned above shall be obscured from the children to ensure healthy raising for them without having mental illnesses or issues, and as long as the child is raised by his parents then the most suitable people to control what is allowed and what is forbidden for him is their parents or his patrons.

Furthermore, allowing the patrons to even suggest a specific contents or topics for their child to see or to interact with would be the most beneficial options for them and then you are really rendering them the easiest way to raise their child properly without having any fear of the corruption that would reach their child without any suspiciousness over and out.

1.2.4. User Privacy

The main purpose of having social media account is to easily expressing yourself while protecting your own privacy, imagine that you want to create a post but you like to show it for specific group of people and the application that you are using does not offer this option, well certainly you will not get comfortable and you will call off posting.

Actually, this is not the goal of the social media; it should encourage you express yourself without any fear of showing it to people other than the desired ones.

There got to be an organization for the users contacts' to allow them to specify specific group of contacts that can show what they have posted, if this kind of customization does not exist then the user will not be fully satisfied with the application usage, and he will not be freely expressing himself, while this is the most fatal issues in the social media application that must be modified and maintained to meet the users' privacy requirements.

1.2.5. Unseen talents

A Talent is the skill that someone naturally has to do something that is hard. It is an ability that someone is born with. Talented people as rule have many talents, for music, dancing, acting, sports, or other skills, but often only in single direction or genre, unlike genius.

Most of people have talent that they good at, but what is the use without glorifying it to whom it may concern like public figures who can help to expose this obscured talent to the whole world !

Well, most of popular social media applications didn't give this prospective the required concern although it is really important to encourage creativity that will finally benefit the society.

The real problem arises when the talents are ignored because it can't be glorified, the effect of ignoring talents are: the responsibility to others has been failed to accomplish because the talent is used mainly to acknowledge and benefit the surrounding people, also the responsibility to the ones self for the reason that he/she is not exposing his/her gifts to the world, and finally it is waste of life to not use or ignore the talent of ones self.

1.3. Proposed system

1.3.1. Overview

With the exposed problems mentioned above, we propose a social media application that would satisfy and supply the demands of users.

The application will expand the limited options of posting for the user by not only allowing for the text, links, picture, albums and videos but also the voice notes posts.

The user will be able to record a voice note and publish it as a post in case of emergency situation or driving or for any other reason to let him express with various ways.

The application will allow the user to organize his/her account as desired into circles and for each circle it has a privacy set - called privacy circle set - that always can be modified by the authenticated user and also there will be privacy set for each user - called privacy user set - as it is considered as user exceptions .For each user or circle the authenticated user can choose among three privacy options which are: valid (can display the authenticated user actions), invalid (cannot display the authenticated user actions until a valid option is shown for the same category, for instance if one user is in two circles and the first circle privacy set is invalid and the second circle is valid then the user privacy set is valid) and invalid for all (cannot display user action always).

For the users who are below 12 years old the the candy session (parental monitor) is enabled, where the parents take control of what is being displayed to their children and also they can suggest specific contents or activities for them to join and exclude what they think that it is harmful for their children to guarantee the children wellbeing.

In Knocks every one is allowed to unleash his/her potentials by exposing their talents freely and confidently to whoever they want.

Knocks team has arranged sessions that are called talent session which are directly linking the talents owners and the talents chefs or public figures who are related to this talent to offer then a better opportunity to unleash their talents and give themselves a step forward to catch their dreams and become what they want to be, Also Knocks is allowing the personal contact between the talented one and the talent sponsor in case of acceptance.

1.4. Functional Requirements

Knocks Application will have some privileges such as :

- System Admin Privilege.
- Security Engineer Privilege.
- Account Owner Privilege.
- Data & Analytics scientist Privilege.
-

Knocks Application premises the user to have the privilege to operate functions :

- **With Reference to System Admin**
 - ❖ **Engineering and Provisioning**
 - Engineering of SA-related solutions for various project and operational needs.
 - Install new / rebuild existing servers and configure hardware, peripherals, services, settings, directories, storage, etc. in accordance with standards and project/operational requirements.
 - Develop and maintain installation and configuration procedures.
 - Contribute to and maintain system standards.
 - Research and recommend innovative, and where possible automated approaches for system administration tasks. Identify approaches that leverage our resources and provide economies of scale.
 - ❖ **Operations and Support**
 - Perform daily system monitoring, verifying the integrity and availability of all hardware, server resources, systems and key

processes, reviewing system and application logs, and verifying completion of scheduled jobs such as backups.

- Perform regular security monitoring to identify any possible intrusions.
- Perform daily backup operations, ensuring all required file systems and system data are successfully backed up to the appropriate media, recovery tapes or disks are created, and media is recycled and sent off site as necessary.
- Perform regular file archival and purge as necessary.
- Create, update, and delete user accounts per request.
- Investigate and troubleshoot issues.
- Repair and recover from hardware or software failures.
- Coordinate and communicate with impacted constituencies.

❖ Maintenance

- Apply OS patches and upgrades on a regular basis, and upgrade administrative tools and utilities. Configure / add new services as necessary.
- Maintain operational, configuration, or other procedures.
- Perform periodic performance reporting to support capacity planning.
- Perform ongoing performance tuning, hardware upgrades, and resource optimization as required. Configure CPU, memory, and disk partitions as required.
- Maintain data center environmental and monitoring equipment.
- **With Reference to account owner**
 1. Publish a knock (create post).
 - Voice note
 - Text (plain text)
 - Photo (in any extension)
 - Album
 - Video
 - file (in any extension)
 - Check-in location
 2. Write a comment- when allowed to do so.
 3. Write a reply- when allowed to do so.
 4. Create a circle.
 5. Delete a circle
 6. Update circle name
 7. Update circle thumbnail

-
8. Include a friend to a circle
 9. Include group friends to a circle at a time.
 10. Exclude a friend from a circle
 11. Exclude group friends from a circle at a time.
 12. Regulate a privacy set for users
 13. Regulate a privacy set for circles
 14. Regulate a parent object privacy set for circle.
 15. Download files (with any extension).
 16. Upload profile picture
 17. Upload cover picture
 18. Update profile picture
 19. Update cover picture
 20. Block a friend
 21. Report a friend
 22. Unfriend a friend
 23. Suggest a content (in case of adult user to his kid account).
 24. Update personal information
 25. React to post- when allowed to do so with allowed reactions.
 26. React to comment- when allowed to do so with allowed reactions.
 27. React to reply- when allowed to do so with allowed reactions.
 28. Chat with a friend
 29. Create a group chat
 30. Update a group chat - when allowed to do so.
 31. Delete a group chat- when allowed to do so.
 32. React to messages
 33. Create a group
 34. Add friends to group
 35. Update a group- when allowed to do so.
 36. Delete a group- when allowed to do so.
 37. Post in a group- when allowed to do so.
 38. Comment in a group- when allowed to do so.
 39. Reply in a group- when allowed to do so.
 40. React to an object a group- when allowed to do so.
 41. Change user permission- when allowed to do so.
 42. Change the group privacy set - when allowed to do so.
 43. Ban members of a group- when allowed to do so.
 44. Report a group- when allowed to do so.
 45. Chat in a group

-
- 46. Request for account verification - incase of public figure.
 - 47. View a user profile - when allowed to do so.
 - 48. Explore talents
 - 49. Upload a talent object - when allowed to do so
 - 50. Filter talents
 - 51. Explore hashtags in talents section
 - 52. Rate a talent object
 - 53. View a talent career
 - 54. Rate a talent owner
 - 55. Leave a review on a talent object
 - 56. Leave a review on a talent owner
 - 57. View Analysis for the talent owner himself
 - 58. Monitor for child log.
 - 59. Monitor objects for kid
 - 60. Set admins for public figure
 - 61. Ban for public figure account
 - 62. Set a parent for a child

Any user is privileged to do operations like : login, logout, register, and normal user privilege is allowed to ones for their own accounts.

1.5. Non- Functional Requirements

1.5.1. Product Requirements

- **Efficiency Requirements** : the application should operate it's requested function in a very short amount of time which is less than 2 seconds the response is received by the client.
- **Usability Requirements** : Enduser training shouldn't exceed 4 hours ,a little experience in social media applications and being familiar to IOS, Android, windows or Mac operating system would be enough to easily use the application.
- **Security Requirements** : the application should be requiring a strong password with a unique username, also the password should be automatically stored encrypted in the database, user's information should not be displayed unless he gives the permission.

1.5.2. Organizational Requirements

- **Operational Requirements** : the user's information confidentiality is operated by the security engineer with having some access to the user information, while the data and analytics scientist has access to some user's informations and user's activity log ,to order to avoid roles confusion

-
- **Implementation** : the System is implemented in 2.5 GHz Intel Core i5 processor and memory specification is 4 GB RAM 1600 MHz DDR3, and implementation is tested through white, black and gray testing.

1.5.3. Additional requirements

- **User Interface** : it is a user-friendly interface which can be easily used by the inexperienced user as it has tooltips all over its icons and a manual for usability and also a support center that helps the user to do what he aims to, the design structure of the application is eye relaxing and has its own modern character through the seeding the colors picking choices.
- **Performance** : the application should operate its requested function by the privileged user in a very short amount of time which is less than 2 seconds the response is received by the client.
- **Reliability** : the system can supply the user demands smoothly without abortion or fault occurrence, it can operate the functions under stress of retrieving big amount of data for many users concurrently with a safe data management with efficient and fast response.
- **Supportability** : Knocks application operates on any version of windows, MacOS, IOS, and Android and any other related operating system and Smartphones.
- **Maintainability** : the system can be easily maintained by the readable and detailed documentation and also can be maintained by corrective and perfective maintenance.

1.6. Target environment

1.6.1. Server requirements

-

1.6.2. Client requirements

- Windows, Mac OS, Linux
- IOS, Android
- //Minimum core i5

1.6.3. Programming languages

❖ Client side

- Axios
- MomentJS
- Moment time zone
- Vue-popper
- Vue-HTML
- Croppie
- Google Map API

-
- pNotify
 - jQuery
 - jQuery touch swipe
 - Image compressor
 - XKESHI/ Image Compressor
 - Base64toBlob
 - Message box
 - EnLocal
 - WebPack
- ❖ Mean stack server side
 - Mongo database
 - NodeJS 8.4.0 (google-v8 engine).
 - NPM package manager
 - AngularJS
 - DotJS
 - ExpressJS
 - ❖ Lamp stack
 - Maria Database
 - PHP 7.1.8 zend engine
 - Composer package manager
 - Laravel as an MVC framework
 - Laravel blade template
 - ❖ Server Infrastructure
 - GRUB boot loader
 - Fedora RedHat (RHEL) linux 26 server edition
 - DNF package manager
 - YUM package manager
 - Apache HTTPD
 - Apache SSL
 - Cockpit resource manager
 - SELinux firewall
 - RHEL firewall

1.6.4. Design

- HTML5
- CSS3
- jQuery
- UIKit

-
- Materialize
 - Vuetify
 - VUE
 - Element UI
 - AnimateCSS

1.6.5. Developer tools

- WebPack
- NPM watcher
- Chromium dev tools
- Audits
- Mozilla dev tools
- Sublime
- Atom
- Sequel pro
- Compass
- Xcode
- IcoMoon
- Adobe illustrator
- Adobe photoshop
- Adobe dev tools
- JSON viewer
- JSON formatter
- Google developers API
- Laravel Faker
- Laravel Test tools
- CORSE -corse domain request
- Safari dev tools
- Advanced rest client
- Postman
- Insomnia rest
- Electron
- GitHub
- Slack
- Beaker browser
- Icons8
- MySQL workbench
- Datagrip

2. Background & Survey

2.1. Introduction

Social media applications are tending to satisfy the user, lets him/her feel free to express him/herself with the desired way, and also it is supposed to convey him the fact that the world is just a room so that he can communicate with whoever he wants easily.

However, after the increasing amount of people who are joining the team, well a different categories, genders, orientations, cultures, races, and of course plenty of mindsets which leads us to break the limitations as possible to satisfy his needs and other users needs. After all privacy must be required gain the user trust and to obtain the main purpose of it which is expressivity allowance

Basically, having a social media application that regulates and organizes all the mentioned issues safely would really save the day, so what is needed currently is an application with some desired features such as : privacy protection, safe expressivity, and flexible customization for each user to gain user's satisfaction.

Well, the best way for building a social media application that can satisfy approximately all users is questioning users with different genders, orientations, cultures, races, and mindsets to get the approximate image of what are we heading to build in order to supply their demands.

2.2. System Benefits

Knocks application is meant to build for the purpose of gaining the user satisfaction and supply his demands as possible, well the application features are covering what most of users needs and cannot find in the existing social media application; the features of the applications has been provided according to filtering what really upsets users when using the existing social media application.

First, the primacy is completely protected at it's advanced level and the flexibility of privacy setting is rendered with the smoothest easy way that can be rendered with , so that the user can customize his own desired privacy sets; it is like he can set a privacy set for each friend as an exception so that he can choose between the general privacy set options which are : valid, invalid, and invalid for all and also he can set a privacy setting for the circles that he previously made by choosing a primacy set option for the circle members to be regulated by.

Second, the parental monitoring of the kids who are aged as twelve years old or younger is high defined for the users, so that the parents are having the full access and full control to monitor the activities and action of the kid, this feature is called the Candy session that is enabled when verifying the age of the user that is twelve years or younger and also after the specification of the kid sponsors and they have to be two people regulating the kid account, also this is done by verifying the sponsors of the kid by the kid him/herself, this kind of transparency is really

satisfying the parents or whoever the kid sponsors because then they are safely allowing their kid to socialize and express himself without having the fear to miss what he does, and also the kid sponsors can freely suggest a content that they see that it is useful for their kid to acquaint which means that the kids can indirectly be introduced to subjects that copes with sponsors' vision.

Third, Cyberbullying avoidance by paying attention to kid mentality health with regulating the interaction between him and others kids or with other adult users, well this is done by for example the kid post, comment or reply cannot be reactor to with the angry reaction or any other reactions that may insult him personally or insult his opinion, in Knocks application there is nothing called stupid idea or stupid opinion specially for the kid because the distractive criticism affects the kid feelings and hurts his own self-esteem which is a non-allowed portion of negativity that must not be received by the kid to guarantee the wellness of his mentality health, also this is basically prevents the cyberbullying that frequently most of kids are facing especially in the social media applications by insulting ones opinions publicly, so by this feature and precise filtering with organized functionality the cyberbullying and other related issues can be easily and safely vanished.

Fourth, Unleashing talents by connecting people to each other specifically to those who are interested in having people with given talents, this is actually done by creating the feature called the talents session; the talent session's main job is to connect the talented people with each talent sponsor so that the talents are not veiled anymore, because having a really gifted person with a specific talent is not an easy way to obtain except being born with it, the talent is actually like the plant that should be planted and exposed to the sun in order to grow or else it will be dead, so in Knocks application unleashing people's talents and glorifying them whether they worth it is one of our main concerns that is considered to be included in the application feature structure.

Fifth, the friendly user interface is obtained by the Knocks application at its highest levels, obviously there are lots of existing social media applications that do not have the friendly user interface in their design which arises the problem of having the user unable to use the application as appropriate so he needs trial and errors and hours and hours to actually know how to deal with the application, and it can actually ends up with the user bored of not knowing how to deal with it anymore after the training trials; in Knocks application the user will not mostly face this kind of problems as the the interface is very friendly and the icons are very familiar to most of people, also it has tooltips allover the page where the user navigates in to show what is the object job exactly with simple and human familiar description so that the user will not need hours of training to do what s/he

intended to do , which increases the usability of the application and this leads to more user satisfaction.

2.3. Survey

Having variety of concerns about the social media applications must be considered carefully, so that the questionnaire have to take care about this point by asking for the suggestions and whether they are satisfied with the existing features of the current used social media applications, the survey is considering a wide range of people, some of them are kids and adults, to try to get a big cluster of people to keep the precision of the survey and to be more realistic.

Well, it is obvious that a lot of people with different ages are using the social media applications, so after the survey has been done, it showed that 98% of people are using the social media application and they already have at least one account, and as long as Knocks application is considering the kids and accounting them as a user with a special treatment to keep their wellbeing, we wanted to dive deep in the children issues and try to stop the social media bullying and the survey showed that 54% of people were a victim of the social media bullying while 70% of them were disabled to share their content because of other people, and 70% of those people are saying that the current social media are not safe for children, also their opinion about whether children being bullied affects the children personality negatively, 36% strongly agree, 52% agree, 3% fairly agree, 7% disagree, and no one strongly disagrees; while their opinion about whether having a kid of 12 years old or younger with his parental monitored social network account will ensure the kid wellbeing 20% strongly agree, 47% agree, 23% fairly agree, 9% disagree, and no one disagrees .

For the importance of the social media in people's opinion, the survey shows that 20% of them are strongly agreed, 45% agree, 30% fairly agree, 3% disagree, and no one strongly disagrees; and for a measurement for the level of activity for those people, there was 25% were very active, 56% were active, 16% not active, 1% not active at all; and about the joined social media applications there was 85% using Facebook, 45% using Twitter, 67% using YouTube, 80% using Instagram, 85% using WhatsApp, 47% using Snapchat, and 10% using Kik, and about whether privacy policies are effective in Social Networking sites or not the survey shows that 16% strongly agree, 54% agree, 16% fairly agree, 12% disagree, and no strongly disagree; while whether it will be better to expand the options of interactivity the survey shows that 16% strongly agree, 70% agree, 10% fairly agree, 1% disagree, and no strongly disagree; and about if the user usually able to interact in your social media account the survey shows that 12% strongly agree, 52% agree, 16% fairly agree, 14% disagree, and 1% strongly disagree.

However, because Knocks application is concerned with the comfy of the user it seeks to offer him the opportunity to let him express using most of devices anywhere, so the

question of which device do people use when accessing to social media applications was 21% uses Desktop/PC, 61% uses Laptop, 18% uses Tablet, and 87% uses Smartphones.

As mentioned before the main goal for Knocks application is to ease the process of expressing, so that we were concerned in the survey whether people are satisfied with the current options of expressing, so the result shown in the survey was that

Problem Solutions identified the top cause of a troubled project was poor requirements. By using Questionnaires, or surveys, lead to collect information from many people in relatively short amount of time. This is especially helpful when stakeholders are spread out geographically , or there is dozen to hundreds of respondents whose input will be needed to help establish system requirements. When using questionnaires, the questions should be focused and organized by a feature or project objective . Questionnaires should be not being too long, to ensure that users will complete them. When constructing the questionnaire, general guideline to determine the questions would be to ask is a yes or no with a level of agreement or disagreement or choices to not lose the user's interest and to be able to complete the whole survey.

2.4. System Tools

2.4.1. Axios

Promise based HTTP client for the browser and node.js
Features

- Make XMLHttpRequests from the browser
- Make http requests from node.js
- Supports the Promise API
- Intercept request and response
- Transform request and response data
- Cancel requests
- Automatic transforms for JSON data
- Client side support for protecting against XSRF

2.4.2. MomentJS

A lightweight JavaScript date library for parsing, validating, manipulating, and formatting dates.

2.4.3. Moment time zone

IANA Time Zone Database + Moment.js.

It formats the time, the difference between the time zone of the client and the UTC

2.4.4. Vue-popper

VueJS popover component based on popper.js which is the engine, the library that computes and, optionally, applies the styles to the poppers.

Some of the key points are:

- Position elements keeping them in their original DOM context (doesn't mess with your DOM!);
- Allows to export the computed informations to integrate with React and other view libraries;
- Supports Shadow DOM elements;
- Completely customizable thanks to the modifiers based structure

2.4.5. Vue-HTML

A Vue.js component implementing the HTML editor with the jQuery summernote plugin.

2.4.6. Croppie

VueCroppie is a Vue 2 wrapper for Croppie a beautiful photo cropping tool for Javascript by foliotek.

2.4.7. Google Map API

The Google API Client Library enables you to work with Google APIs such as Google+, Drive, or YouTube on your server.

2.4.8. pNotify

PNotify is a JavaScript notification plugin. PNotify can provide desktop notifications based on the Web Notifications spec. If desktop notifications are not available or not permitted, PNotify will fall back to an in-browser notice.

2.4.9. jQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

2.4.10. jQuery touch swipe

A jQuery plugin to be used on touch devices such as iPad, iPhone, Android etc.

Detects single and multiple finger swipes, pinches and falls back to mouse 'drags' on the desktop.

Time and distance thresholds can be set to distinguish between swipe gesture and slow drag.

Allows exclusion of child elements (interactive elements) as well allowing page scrolling or page zooming depending on configuration.

- Detects swipes in 4 directions, "up", "down", "left" and "right"
- Detects pinches "in" and "out"
- Supports single finger or double finger touch events
- Supports click events both on the touchSwipe object and its child objects
 - Definable threshold / maxTimeThreshold to determine when a gesture is actually a swipe
 - Events triggered for swipe "start", "move", "end" and "cancel"
 - End event can be triggered either on touch release, or as soon as threshold is met
 - Allows swiping and page scrolling
 - Disables user input elements (Button, form, text etc) from triggering swipes

2.4.11. Image compressor

A simple JavaScript image compressor. Uses the Browser's native canvas.toBlob API to do the compression work. General use this to pre-compress a client image file before upload it.

2.4.12. XKESHI/Image Compressor

A simple JavaScript image compressor. Uses the Browser's native canvas.toBlob API to do the compression work. General use this to pre-compress a client image file before upload it.

2.4.13. Base64toBlob

Convert a base64 string to a Blob object.

2.4.14. Message box

An NPM package for defining and getting validation error messages in JavaScript. Meteor apps can pass in `Tracker` to get reactive language selection.

2.4.15. EnLocal

A type of variable that lives only in the scope in which it is declared and then forgotten. A variable declared inside the `setup()` of a program would be local: after the `setup()` finished running, the Galileo would forget that the variable ever existed.

2.4.16. WebPack

webpack is a module bundler. Its main purpose is to bundle JavaScript files for usage in a browser, yet it is also capable of transforming, bundling, or packaging just about any resource or asset.

2.4.17. Mongo database

MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need

2.4.18. NodeJS 8.4.0 (google-v8 engine).

V8 is Google's open source high-performance JavaScript engine, written in C++ and used in Google Chrome, the open source browser from Google, and in Node.js, among others. It implements ECMAScript as specified in ECMA-262, and runs on Windows 7 or later, macOS 10.5+, and Linux systems that use IA-32, ARM, or MIPS processors. V8 can run standalone, or can be embedded into any C++ application.

2.4.19. NPM package manager

npm opens up an entire world of JavaScript talent to you and to your team. It's the world's largest software registry, with approximately 3 billion downloads per week. The registry contains over 600,000 *packages* (building blocks of code). Open-source developers from every continent use npm to share and borrow packages. Packages are structured to enable you to keep track of dependencies and versions.

2.4.20. AngularJS

AngularJS lets you write client-side web applications as if you had a smarter browser. It lets you use good old HTML (or HAML, Jade/Pug and friends!) as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. It automatically synchronizes data from your UI (view) with your JavaScript objects (model) through 2-way data binding. To help you structure your application better and make it easy to test, AngularJS teaches the browser how to do dependency injection and inversion of control.

It also helps with server-side communication, taming async callbacks with promises and deferred objects, and it makes client-side navigation and deep linking with hashbang urls or HTML5 pushState a piece of cake. Best of all? It makes development fun!

2.4.21. DotJS

Created in search of the fastest and concise JavaScript templating function with emphasis on performance under V8 and nodejs. It shows great performance for both nodejs and browsers.

2.4.22. ExpressJS

Fast, un-opinionated, minimalist web framework for node.

2.4.23. Lamp stack

LAMP stack is a popular open source web platform commonly used to run dynamic web sites and servers. It includes Linux, Apache, MySQL, and PHP/Python/Perl and is considered by many the platform of choice for development and deployment of high performance web applications which require a solid and reliable foundation.

LAMP stack includes all the standard features in TurnKey Core, and on top of that:

- SSL support out of the box.
- PHP, Python and Perl support for Apache2 and MySQL.
- PHP development helpers

2.4.24. Maria Database

MariaDB is designed as a drop-in replacement of MySQL(R) with more features, new storage engines, fewer bugs, and better performance.

MariaDB is brought to you by the MariaDB Foundation. Please read the CREDITS file for details about the MariaDB Foundation, and who is developing MariaDB.

MariaDB is developed by many of the original developers of MySQL who now work for the MariaDB Foundation and the MariaDB Corporation, and by many people in the community.

2.4.25. PHP 7.1.8 zend engine

2.4.26. Composer package manager

Composer helps you declare, manage, and install dependencies of PHP projects.

2.4.27. Laravel as an MVC framework

Laravel is a web application framework with expressive, elegant syntax. We believe development must be an enjoyable, creative experience to be truly fulfilling. Laravel attempts to take the pain out of development by easing common tasks used in the majority of web projects, such as:

- Simple, fast routing engine.
- Powerful dependency injection container.
- Multiple back-ends for session and cache storage.
- Database agnostic schema migrations.
- Robust background job processing.
- Real-time event broadcasting.

Laravel is accessible, yet powerful, providing tools needed for large, robust applications. A superb combination of simplicity, elegance, and innovation gives you a complete toolset required to build any application with which you are tasked

2.4.28. Laravel blade template

The standalone version of Laravel's Blade templating engine for use outside of Laravel.

2.4.29. Server Infrastructure

Support documents and files for deploying the Chibitronics Love-to-Code server infrastructure.

2.4.30. GRUB boot loader

GNU GRUB (short for GNU GRand Unified Bootloader) is a boot loader package from the GNU Project. GRUB is the reference implementation of the Free Software Foundation's Multiboot Specification, which provides a user the choice to boot one of multiple operating systems installed on a computer or select a specific kernel configuration available on a particular operating system's partitions.

2.4.31. Fedora RedHat (RHEL) linux 26 server edition

is a Unix-like operating system based on the Linux kernel and GNU programs (a Linux distribution), developed by the community-supported Fedora Project, and sponsored by the Red Hat company. Fedora contains software distributed under various free and open-source licenses and aims to be on the leading edge

of such technologies. Fedora is the upstream source of the commercial Red Hat Enterprise Linux distribution.

2.4.32. DNF package manager

Dandified Yum (DNF) is the next upcoming major version of Yum. It does package management using RPM, libolv and hawkey libraries. For metadata handling and package downloads it utilizes librepo. To process and effectively handle the comps data it uses libcomps.

2.4.33. YUM package manager

Yum is an automatic updater and package installer/remover for rpm systems. It automatically computes dependencies and figures out what things should occur to install packages. It makes it easier to maintain groups of machines without having to manually update each one using rpm. Yum has a plugin interface for adding simple features. Yum can also be used from other python programs via its module interface.

2.4.34. Apache HTTPD

This cookbook provides a complete Debian/Ubuntu style Apache HTTPD configuration. Non-Debian based distributions such as Red Hat/CentOS, ArchLinux and others supported by this cookbook will have a configuration that mimics Debian/Ubuntu style as it is easier to manage with Chef.

2.4.35. Apache SSL

The Apache HTTP Server, colloquially called Apache , is free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

2.4.36. Cockpit resource manager

is a set of training procedures for use in environments where human error can have devastating effects. Used primarily for improving air safety, CRM focuses on interpersonal communication, leadership, and decision making in the cockpit of an airliner.

Crew resource management formally began with a National Transportation Safety Board (NTSB) recommendation made during their investigation of the 1978 United Airlines Flight 173 crash. The issues surrounding that crash included a DC-8 crew running out of fuel over Portland, Oregon while troubleshooting a landing gear problem.

2.4.37. SELinux firewall

SELinux is a great component to have for security. There are a ton of benefits of configuring it for your servers, whether they are publicly facing or not.

Firewalls do a great job on filtering out unwanted or malicious activity, but flaws in your security can still be leveraged by external exploits as well as internal flaws in configuration or buggy software.

2.4.38. RHEL firewall

The real power of the tool is the use of custom commands/scripts you can use to populate the option and finally have a fully working firewall setup for both IPv4 and IPv6.

There a set of predefined rules called default.

On both scripts **host_manager.py** or **iptables_manager.py** you have to specify the **--deploy** argument in order for the configuration to be applied on the machine. There is also an option of generating the actual files so you can use them along with **iptables-restore**, **ip6tables-restore**, **ipset-restore** commands.

2.4.39. HTML5

HTML5 Boilerplate is a professional front-end template for building fast, robust, and adaptable web apps or sites.

This project is the product of years of iterative development and community knowledge. It does not impose a specific development philosophy or framework, so you're free to architect your code in the way that you want.

2.4.40. CSS3

CSS3 is the latest evolution of the Cascading Style Sheets language and aims at extending CSS2.1. It brings a lot of long-awaited novelties, like rounded corners, shadows, gradients, transitions or animations, as well as new layouts like multi-columns, flexible box or grid layouts. Experimental parts are vendor-prefixed and should either be avoided in production environments, or used with extreme caution as both their syntax and semantics can change in the future.

2.4.41. jQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of

browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

2.4.42. UIKit

A lightweight and modular front-end framework for developing fast and powerful web interfaces.

2.4.43. Materialize

A modern responsive front-end framework based on Material Design.

2.4.44. Vuetify

Vuetify is an open source MIT project that has been made possible due to the generous contributions by community backers. If you are interested in supporting this project, please consider becoming a patron.

2.4.45. VUE

Vue.js (commonly referred to as Vue; pronounced /vju:/, like view) is an open-source progressive JavaScript framework for building user interfaces.[4] Integration into projects that use other JavaScript libraries is made easy with Vue because it is designed to be incrementally adoptable. Vue can also function as a web application framework capable of powering advanced single-page applications.

2.4.46. Element UI

Element, a Vue 2.0 based component library for developers, designers and product managers.

2.4.47. AnimateCSS

animate.css is a bunch of cool, fun, and cross-browser animations for you to use in your projects. Great for emphasis, home pages, sliders, and general just-add-water-awesomeness.

2.4.48. NPM watcher

The intention of this module is provide tools that make managing the watching of file & directory trees easier.

2.4.49. Chromium dev tools

The Chrome Developer Tools (DevTools for short), are a set of web authoring and debugging tools built into Google Chrome. The DevTools provide web developers deep access into the internals of the browser and their web application. Use the DevTools to efficiently track down layout issues, set JavaScript breakpoints, and get insights for code optimization.

2.4.50. Audits

Website audit is a full analysis of all the factors that affect website's visibility in search engines. This standard method gives a complete insight into any website, overall traffic and individual pages. Website audit is completed solely for marketing purposes. The goal is to detect weak points in campaigns that affect web performance.

2.4.51. Mozilla dev tools

To Examine, edit, and debug HTML, CSS, and JavaScript on the desktop and on mobile.

2.4.52. Sublime

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

2.4.53. Atom

Atom is a text editor that's modern, approachable, yet hackable to the core a tool you can customize to do anything but also use productively without ever touching a config file.

2.4.54. Sequel pro

Sequel Pro is open source and built by people like you. We'd love your input whether you've found a bug, have a suggestion or want to contribute some code.

2.4.55. Compass

Compass is an open-source CSS Authoring Framework.

2.4.56. Xcode

Xcode is an integrated development environment (IDE) for macOS containing a suite of software development tools developed by Apple for developing software for macOS, iOS, watchOS, and tvOS. First released in 2003, the latest stable release is version 9.2 and is available via the Mac App Store free of charge for macOS High Sierra and macOS Sierra users. Registered developers can download preview releases and prior versions of the suite through the Apple Developer website.

2.4.57. IcoMoon

Pixel Perfect Icon Solutions.

2.4.58. Adobe illustrator

The industry-standard vector graphics app lets you create logos, icons, drawings, typography, and illustrations for print, web, video, and mobile.

2.4.59. Adobe photoshop

Adobe Photoshop is a raster graphics editor developed and published by Adobe Systems for macOS and Windows.

2.4.60. Adobe dev tools

The following developer tools are available for HTL:

- HTML Read Eval Print Loop
- A live execution environment for playing with the language, great for learning.
- AEM Brackets extension
- An easy tool for front-end developers to work on AEM components.
- AEM Eclipse extension
- A tool for back-end developers to create AEM projects and easily edit them.

2.4.61. JSON viewer

The most beautiful and customizable JSON/JSONP highlighter that your eyes have ever seen. It is a Chrome extension for printing JSON and JSONP.

- This extension might crash with other JSON highlighters/formatters, you may need to disable them
- To highlight local files and incognito tabs you have to manually enable these options on the extensions page

-
- Sometimes when the plugin updates chrome leaves the old background process running and revokes some options, like the access to local files. When this happens just recheck the option that everything will work again
 - Works on local files (if you enable this in chrome://extensions)

Features:

- Syntax highlighting
- 27 built-in themes
- Collapsible nodes
- Clickable URLs (optional)
- URL does not matter (the content is analysed to determine if it's a JSON or not)
 - Inspect your json typing "json" in the console
 - Hot word json-viewer into omnibox (type json-viewer + TAB and paste your JSON into omnibox, hit ENTER and it will be highlighted)
 - Toggle button to view the raw/highlighted version
 - Works with numbers bigger than Number.MAX_VALUE
 - Option to show line numbers
 - Option to customize your theme
 - Option to customize the tab size
 - Option to configure a max JSON size to highlight
 - Option to collapse nodes from second level + Button to unfold all collapsed nodes
 - Option to include a header with timestamp + url
 - Option to allow the edition of the loaded JSON
 - Option to sort json by keys
 - Option to disable auto highlight
 - Option for C-style braces and arrays

2.4.62. JSON formatter

Chrome extension for printing JSON and JSONP nicely when you visit it 'directly' in a browser tab.

Features

- JSONP support
- Fast, even on long pages
- Works on any valid JSON page – URL doesn't matter
- Syntax highlighting
- Collapsible trees, with indent guides
- Clickable URLs

-
- Buttons for switching between raw and parsed JSON
 - Parsed JSON is exported as a global variable, `json`, so you can inspect it in the console
- A background worker is used to prevent the UI freezing when processing very long JSON pages.

2.4.63. Google developers API

They have the world mapped. With more than one billion global monthly active users in over 200 countries, their data gives you accurate real-time information for mapping, navigation and places.

2.4.64. Laravel Faker

Faker is a PHP library that generates fake data for you. Whether you need to bootstrap your database, create good-looking XML documents, fill-in your persistence to stress test it, or anonymize data taken from a production service, Faker is for you.

2.4.65. Laravel Test tools

Testing a Laravel application really is an easy task - the "Integrated" package from Jeffrey Way, that later got merged into the core framework is fantastic and helps you with the otherwise cumbersome task of testing and interacting with your application. But still - are *you* actually using tests?

A lot of times people really love the idea of tests, but simply don't get their asses up to start using them in their own projects. That's why I created a chrome extension that hopefully saves you some time when you need to test your app.

2.4.66. CORSE -corse domain request

Cross-Origin Resource Sharing (CORS) is a mechanism that uses additional HTTP headers to let a user agent gain permission to access selected resources from a server on a different origin (domain) than the site currently in use. A user agent makes a cross-origin HTTP request when it requests a resource from a different domain, protocol, or port than the one from which the current document originated.

2.4.67. Safari dev tools

Apple has brought its expertise in macOS and iOS development tools to the web. Safari includes Web Inspector, a powerful tool that makes it easy to modify, debug, and optimize a website for peak performance and compatibility on both platforms. And with Responsive Design Mode, you can even preview

your webpages for various screen sizes, orientations, and resolutions. To access these tools, enable the Develop menu in Safari's Advanced preferences.

2.4.68. Advanced rest client

Cross platform API testing, Use ARC on any major platforms including Windows, MacOS and Linux.

2.4.69. Postman

Developers use Postman to build, modern software for the API-first world.

2.4.70. Insomnia rest

Powerful HTTP tool belt, in one intuitive app, Free and open source on Mac, Windows, and Linux.

2.4.71. Electron

Electron is a framework for creating native applications with web technologies like JavaScript, HTML, and CSS. It takes care of the hard parts so you can focus on the core of your application.

2.4.72. GitHub

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside millions of other developers.

2.4.73. Slack

Slack is a cloud-based set of proprietary team collaboration tools and services, founded by Stewart Butterfield.[1] Slack began as an internal tool used by their company, Tiny Speck, in the development of Glitch, a now defunct online game. The name is an acronym for "Searchable Log of All Conversation and Knowledge".

2.4.74. Beaker browser

Beaker is a peer-to-peer browser with tools to create and host websites. Don't just browse the Web, build it.

2.4.75. Icons8

0 Free Flat Icons, in any format, size and color in 20 seconds.

2.4.76. MySQL workbench

MySQL Workbench is a visual database design tool that integrates SQL development, administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system. It is the successor to DBDesigner 4 from fabFORCE.net, and replaces the previous package of software, MySQL GUI Tools Bundle.

2.4.77. Datagrip

new database IDE that is tailored to suit specific needs of professional SQL developers.

3. System analysis

3.1. Scenario

3.1.1. Publish a Knock

Scenario ID	Knocks-1
Scenario Name	Publish a Knock
Actors	User
Initial Assumption	The user is logged in and wants to create a post and then publish it to be shown for the previously mentioned people
Normal	<p>the user clicks the button publish in the user interface with the given userid as he is logged in already, then the click button emits the click event which will be listened by the record event listener and the media file uploader event listener and if the post have media blobs then the media is uploaded and then the record and media file uploader sends an XMLHttpRequest with the headers that mainly contains the CSRF of the user and the userid to the NodeJS and then NodeJs validates the CSRF header with the PHP side to pass the validation operation to mysql database and if validated successfully then NodeJS inserts the blobs (the contained media) into the MongoDB with the userid then mongoDb responds after insertion with a token that acts as the index of the blob in the database then nodeJs retrieves the token to the vue component, after that the record event listener and the media file uploader event listeners are emitting the media query logged with the data included to the due component knock then the knock prepares and submit the data, and sends the final sub the App API and the knock button sends the XMLHttpRequest to the PHP side and then PHP validates the XMLHttpRequest and data and inserts the data and the user id into Mysql database and then PHP and user are acknowledged with successful message.</p>
What can go wrong	<ol style="list-style-type: none">If the content of the post is empty, so that the media query logged has false as a response from the media file uploader and the recorder to store in mongoDB, and it has no plain text to store in MySQL so that the process terminates with a failure messageIf the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.
Other Activities	The user can proceed all operations allowed while publishing a new post but not allowed to edit the privacy sets that has been regulated previously while the publishing operation is in progress
System on Completion	A success message that the post has been published successfully and safely inserted into the database which can be retrieved securely, or a failure message that the condition of publishing the post has not been satisfied.

3.1.2. Write a comment

Scenario ID	Knocks-2
Scenario Name	Write a comment
Actors	User
Initial Assumption	The user is logged in and wants to write a comment and then publish it to be shown for the previously mentioned people in the privacy set
Normal	<p>the user clicks the button publish in the user interface with the given userid as he is logged in already, then the click button emits the click event which will be listened by the record event listener and the media file uploader event listener and if the post have media blobs then the media is uploaded and then the record and media file uploader sends an XMLHttpRequest with the headers that mainly contains the CSRF of the user and the userid to the NodeJS and then NodeJs validates the CSRF header with the PHP side and if validated successfully then NodeJS inserts the blobs (the contained media) into the MongoDB with the userid then mongoDb responds after insertion with a token that acts as the index of the blob in the database then nodeJs retrieves the token to the vue component, after that the record event listener and the media file uploader event listeners are emitting the media query logged with the data included to the due component knock then the knock prepares and submit the data, and sends the final sub the App API and the knock button sends the XMKHTTPRequest to the PHP side and then PHP validates the XMLHttpRequest and data and inserts the data and the user id and the object parent id into Mysql database and then PHP and user are acknowledged with successful message.</p>
What can go wrong	<ol style="list-style-type: none"> If the content of the comment is empty, so that the media query logged has false as a response from the media file uploader and the recorder to store in mongoDb, and it has no plain text to store in MySql so that the process terminates with a failure message If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.
Other Activities	The user can proceed all operations allowed while writing a new comment but not allowed to edit the privacy sets that has been regulated previously while the publishing operation is in progress
System on Completion	A success message that the comment has been published successfully and safely inserted with the parent object id into the database which can be retrieved securely, or a failure message that the condition of writing the comment has not been satisfied.

3.1.3. Writing a reply

Scenario ID	Knocks-3
Scenario Name	Write a reply
Actors	User
Initial Assumption	The user is logged in and wants to write a comment and then publish it to be shown for the previously mentioned people in the privacy set
Normal	<p>the user clicks the button publish in the user interface with the given userid as he is logged in already, then the click button emits the click event which will be listened by the record event listener and the media file uploader event listener and if the post have media blobs then the media is uploaded and then the record and media file uploader sends an XMLHttpRequest with the headers that mainly contains the CSRF of the user and the userid to the NodeJS and then NodeJs validates the CSRF header with the PHP side and if validated successfully then NodeJS inserts the blobs (the contained media) into the MongoDB with the userid then mongoDb responds after insertion with a token that acts as the index of the blob in the database then nodeJs retrieves the token to the vue component, after that the record event listener and the media file uploader event listeners are emitting the media query logged with the data included to the due component knock then the knock prepares and submit the data, and sends the final sub the App API and the knock button sends the XMKHTTPRequest to the PHP side and then PHP validates the XMLHttpRequest and data and inserts the data and the user id and the object parent id into Mysql database and then PHP and user are acknowledged with successful message.</p>
What can go wrong	<ol style="list-style-type: none"> If the content of the comment is empty, so that the media query logged has false as a response from the media file uploader and the recorder to store in mongoDb, and it has no plain text to store in MySql so that the process terminates with a failure message If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.
Other Activities	The user can proceed all operations allowed while writing a new comment but not allowed to edit the privacy sets that has been regulated previously while the publishing operation is in progress
System on Completion	A success message that the comment has been published successfully and safely inserted with the parent object id into the database which can be retrieved securely, or a failure message that the condition of writing the comment has not been satisfied.

3.1.4. Create a circle

Scenario ID	Knocks-4
Scenario Name	Create a circle
Actors	User
Initial Assumption	The user is logged in and wants to add a new circle to his account
Normal	The user clicks the add new circle button after supplying the circle name and the selected thumbnail for the circle then the circle name and the thumbnail and the user id is passed to the php side , and then the php validates the circle with the circle name and the username from mysql database and if there is no such a circle with the same name for such a user then the circle is inserted in a new record in the circles table in mysql database with the user id and circle name and the thumbnail.
What can go wrong	1. If the circle is already exists with the same name for the logged in user
Other Activities	The user can proceed all operations allowed while adding new circle
System on Completion	A success message that the circle has been added successfully with the default privacy circle set, or a failure message that the circle cannot be added to the specified user

3.1.5. Delete a Circle

Scenario ID	Knocks-5
Scenario Name	Delete a circle
Actors	User
Initial Assumption	The user is logged in and wants to delete a circle to from his account
Normal	The user selects a specific circle or group of circles as desired and then he clicks the delete button so this event is passed to the php side with the circle name and the user id, after that the php side validates the given circle with the circle name and the user id and checks if it already exists in the database bounded to the requested user id or not, and if the validation returns success then the php side deletes the circle form mysql database with the given user id and circle name.
What can go wrong	1. If the circle is already does not exists with the same name for the logged in user in the database
Other Activities	The user can proceed all operations allowed while deleting a circle but adding a new circle with the name name of the desired circle to be deleted cannot be done unless the process of deletion has been successfully done, and also users cannot be added to the circle that is being deleted or even modifying the primacy set of the circle.
System on Completion	A success message that the circle has been deleted successfully, or a failure message that the circle cannot be deleted because the validation failed.

3.1.6. Update a circle name

Scenario ID	Knocks-6
Scenario Name	Update a circle name
Actors	User
Initial Assumption	The user is logged in and wants to update a circle name that he already has
Normal	The user selects a specific circle name as desired and then he clicks the update button so this event is passed to the php side with the circle name and the user id, after that the php side validates the given circle with the circle name and the user id and checks if it already exists in the database bounded to the requested user id or not, and if the validation returns success then the php side updates the circle name from mysql database with the given user id and circle name.
What can go wrong	1. If the circle does not exist with the same name for the logged in user in the database
Other Activities	The user can proceed all operations allowed while updating existing circle but adding a new circle with the name or deleting the circle cannot be done unless the process of updating has been successfully done.
System on Completion	A success message that the circle name has been updated successfully, or a failure message that the circle name cannot be updated because the validation failed.

3.1.7. Update a circle thumbnail

Scenario ID	Knocks-7
Scenario Name	Update a circle thumbnail
Actors	User
Initial Assumption	The user is logged in and wants to update a circle thumbnail that he already has
Normal	The user selects a specific circle thumbnail as desired and then he clicks the update button so this event is passed to the php side with the circle name and the circle thumbnail and the user id, after that the php side validates the given circle with the circle name and the user id and checks if it already exists in the database bounded to the requested user id or not, and if the validation returns success then the php side updates the circle thumbnail form mysql database with the given user id and circle name and circle thumbnail.
What can go wrong	1. If the circle does not exist with the same name for the logged in user in the database
Other Activities	The user can proceed all operations allowed while updating existing circle but adding a new circle with the name or deleting the circle cannot be done unless the process of updating has been successfully done.
System on Completion	A success message that the circle thumbnail has been updated successfully, or a failure message that the circle thumbnail cannot be updated because the validation failed.

3.1.8. Include a friend to a circle

Scenario ID	Knocks-8
Scenario Name	Include a friend to a circle
Actors	User
Initial Assumption	The user is logged in and has existing circle and wants to add a friend to this circle
Normal	<p>Including a friend to a circle is done with two ways, The first one is while accepting the friend request there is add to circle button that the friend who sent a friend request to be added to a circle which the user specifies.</p> <p>The second way is adding already friend user to add him into a circle that the account user owns.</p> <p>In both ways after the user clicks the add to circle button the user id and the friend id with the circle name is passed to the php side to validate the circle name and the user id from the mysql database to check if the account owner already have a circle that has the already passed name or not and also it validates if the friend id is already existing in one of the circles or at least in the all circle that has all the friends of the user and after validation responses with success then the friend is added to the circle with the user id and friend id and the circle name in the database and then it returns a success message that the friend has been successfully added to the specified circle.</p>
What can go wrong	<ol style="list-style-type: none"> 1. If the circle does not exist with the same name for the logged in user in the database 2. If the friend user does not exist in any circle which the account owner has
Other Activities	The user can proceed all operations allowed while adding a friend to circle but modifying the same circle privacy set is not allowed until the transactions on the circle has been completed or even deleting the same circle or updating its name.
System on Completion	A success message that friend has been added to the specified circle successfully, or a failure message that the friend cannot be added to the circle because the validation failed.

3.1.9. Include a group of friends to a circle at a time

Scenario ID	Knocks-9
Scenario Name	include a group of friends to a circle at a time
Actors	User
Initial Assumption	The user is logged in and has existing circle and wants to add a group of friends to this circle
Normal	<p>Including a group of friends to a circle is done with two ways, The first one is while accepting the friend requests there is add to circle button that the friends who sent a friend requests to be added to a circle which the user specifies.</p> <p>The second way is adding already friend users to add them into a circle that the account user owns.</p> <p>In both ways after the user clicks the add to circle button the user id and the friend ids with the circle name is passed to the php side to validate the circle name and the user id from the mysql database to check if the account owner already have a circle that has the already passed name or not and also it validates if the friend id is already existing in one of the circles or at least in the all circle that has all the friends of the user and after validation responses with success then the friends are added to the circle with the user id and friend ids and the circle name in the database and then it returns a success message that the friends have been successfully added to the specified circle.</p>
What can go wrong	<ol style="list-style-type: none"> If the circle does not exist with the same name for the logged in user in the database If the friend users do not exist in any circle which the account owner has
Other Activities	The user can proceed all operations allowed while adding a friend circle but modifying the same circle privacy set is not allowed until the transactions on the circle has been completed or even deleting the same circle or updating its name.
System on Completion	A success message that group of friends has been added to the specified circle successfully, or a failure message that the group of friends cannot be added to the circle because the validation failed.

3.1.10. Exclude a friend from a circle

Scenario ID	Knocks-10
Scenario Name	Exclude a friend from a circle
Actors	User
Initial Assumption	The user is logged in and has existing circle and wants to exclude a friend from this circle
Normal	<p>Excluding a friend from a circle is done by removing already friend user to exclude him from a circle that the account user owns.</p> <p>after the user clicks the remove from circle button the user id and the friend id with the circle name is passed to the php side to validate the circle name and the user id from the mysql database to check if the account owner already have a circle that has the already passed name or not and also it validates if the friend id is already existing in the specified circle and after validation responses with success then the friend is added to the circle with the user id and friend id and the circle name in the database and then it returns a success message that the friend has been successfully excluded from the specified circle.</p>
What can go wrong	<ol style="list-style-type: none"> If the circle does not exist with the same name for the logged in user in the database If the friend user does not exist in the specified circle which the account owner wants to exclude the friend from.
Other Activities	The user can proceed all operations allowed while excluding a friend from a circle, but it is not allowed to update the circle name until the exclusion process has been completed.
System on Completion	A success message that friend has been excluded the specified circle successfully, or a failure message that the friend cannot be excluded from the circle because the validation failed.

3.1.11. Exclude a group of friends to a circle at a time

Scenario ID	Knocks-11
Scenario Name	Exclude a group of friends from a circle
Actors	User
Initial Assumption	The user is logged in and has existing circle and wants to exclude a group of friends from this circle
Normal	<p>Excluding a group of friends from a circle is done by removing already friend users to exclude them from a circle that the account user owns.</p> <p>after the user clicks the remove from circle button the user id and the friends ids with the circle name is passed to the php side to validate the circle name and the user id from the mysql database to check if the account owner already have a circle that has the already passed name or not and also it validates if the friends ids is already existing in the specified circle and after validation responses with success then the group of friends are added to the circle with the user id and friends ids and the circle name in the database and then it returns a success message that the group of friends have been successfully excluded from the specified circle.</p>
What can go wrong	<ol style="list-style-type: none"> If the circle does not exist with the same name for the logged in user in the database If the group of friends do not exist in the specified circle which the account owner wants to exclude the group of friends from.
Other Activities	The user can proceed all operations allowed while excluding a group of friends from a circle, but it is not allowed to update the circle name until the exclusion process has been completed.
System on Completion	A success message that the group of friends have been excluded the specified circle successfully, or a failure message that the group of friends cannot be excluded from the circle because the validation failed.

3.1.12. Regulate a privacy set for users

Scenario ID	Knocks-12
Scenario Name	Regulate a privacy set for users
Actors	User
Initial Assumption	The user is logged in and has friends and wants to regulate a primacy set for a specific user
Normal	<p>The user can regulate a privacy set for other users in two ways, The first way is setting a privacy set for a user after accepting the friend request, and the second way can be done by regulating the user privacy set when he is already a friend to the account owner.</p> <p>Both ways are counting the process that the user will select the user and then he will choose the desired user privacy set whether it is valid, invalid or invalid for all and then clicks on the update button then this event is emitted to the php side with the user id and the friend id and the desired privacy set, the the php slide validates the friend that it checks if the friend already exist in a circle that the account owner has , and if this validation returns success then the php side checks whether the privacy set is already existing in the available privacy sets specified in the database and if this validation also returns success then mysql is ready to update the privacy set fo the friend with the arguments: user id and the friend id and the user privacy set.</p>
What can go wrong	<ol style="list-style-type: none"> 1. If the friend does not exist in any circle which the account owner has 2. If the privacy set given is not in the specified option in the database
Other Activities	The user can proceed all operations allowed while regulating a user privacy set, but he cannot unfriend or block this specified friend
System on Completion	A success message that the friend privacy set has been successfully updated, or a failure message that the specified friend privacy set cannot be updated to the desired privacy set because the validation failed.

3.1.13. Regulate a privacy set for circles

Scenario ID	Knocks-13
Scenario Name	Regulate a privacy set for circles
Actors	User
Initial Assumption	The user is logged in and has friends and wants to regulate a primacy set for a specific circle
Normal	The user can regulate a privacy set for other users by selecting a specific circle and select the privacy circle set for it whether it is valid, invalid or invalid for all and then clicks on the update button then this event is emitted to the php side with the user id and the circle name and the desired privacy set, the the php slide validates the circle name that it checks if the specified circle name is in the circle list of the account owner or not, and if this validation returns success then the php side checks whether the privacy set is already existing in the available privacy sets specified in the database and if this validation also returns success then mysql is ready to update the privacy set fo the circle with the arguments: user id and the circle id and the circle privacy set.
What can go wrong	<ol style="list-style-type: none"> 1. If the circle name does not exist in the circle list which the account owner has 2. If the privacy set given is not in the specified option in the database
Other Activities	The user can proceed all operations allowed while regulating a circle privacy set, but he cannot delete the same specified name or exclude and include friends from this circle or change the circle name until the updating process is completed.
System on Completion	A success message that the circle privacy set has been successfully updated, or a failure message that the specified circle privacy set cannot be updated to the desired privacy set because the validation failed.

3.1.14. Download files

Scenario ID	Knocks-14
Scenario Name	Download files
Actors	User
Initial Assumption	The user is logged in and he wants to download specified file
Normal	<p>The user can download a specified file by selecting the desired file and press the file icon or press the button download to start downloading, well it starts with the user click then a get request with the object id and the user id with the CSRF generated is sent to the nodeJS side to check the availability of the object to the user with the user id and the object id with the php side that authorizes the availability of the user id ad the object Id in mysql and after sending the authorization to the php side and acknowledgement to the nodeJS side then nodeJS will check the headers with the given CSRF the it fitches for the ole with the object id of the file then when rendering the file the nodeJS side it sets the file initiates like the encoding process and the extension setting then it can be downloaded to the user.</p>
What can go wrong	<ol style="list-style-type: none">1. The file to be uploaded can't be available to the user according to the object privacy set2. The file is not available at the database anymore
Other Activities	The user can proceed all operations will downloading file.
System on Completion	A success message that the file is downloading and it stars downloading to the user device, or a failure message that the specified file cannot be downloaded because the validation failed.

3.1.15. Upload a profile picture

Scenario ID	Knocks-15
Scenario Name	Upload a profile picture
Actors	User
Initial Assumption	The user is logged in and he wants to upload his profile picture
Normal	<p>The user will be uploading his profile picture that he had his profile picture empty before it, so he would be going to the side that is for displaying his own profile picture and clicks the button of uploading his profile picture in the dropdown list, then the user will choose from his device pictures and then he would be able to crop it and process on this picture then internal the after clicking the button upload the button upload will be triggering the click event with the user id and the profile picture chosen by the user.</p> <p>After that, the media file uploader will be uploading the profile picture and then it sends XMLHttpRequest with the CSRF header and the user id to nodeJS side which will validate the header with the php side then nodeJS side will be inserting the profile picture as a blob into MongoDB and then nodeJs will be delivering the token generated from mongoDb to the vue component and then the cue component will be sending acknowledgement to the media file uploader then the media file uploader will be emitting the media query logged to the app API.</p> <p>Finally, the app API will be sending an acknowledgement to the due component which will trigger the process of preparing data to be submitted with the given token by the due component and also the vue component will be responsible for the final submit sent for the app API and after that due component sends an acknowledgement to the click event listener so that the media file uploader after listening on this click event listener it will send XMLHttpRequest with the token and user id to the php side which will validate XMLHttpRequest with the header then it inserts the profile picture with token and the user id into mysql database.</p>
What can go wrong	<ol style="list-style-type: none"> If the content of the profile picture is empty, so that the media query logged has false as a response from the media file uploader to store in mongoDB, so that the process terminates with a failure message If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.
Other Activities	The user can proceed all operations allowed while uploading a profile picture but not allowed to edit the privacy sets that has been regulated previously while the uploading operation is in progress
System on Completion	A success message that the profile picture has been uploaded successfully and safely inserted into the database, or a failure message that the condition of uploading a profile picture has not been satisfied.

3.1.16. Upload a cover picture

Scenario ID	Knocks-16
Scenario Name	Upload a cover picture
Actors	User
Initial Assumption	The user is logged in and he wants to upload his cover picture
Normal	<p>The user will be uploading his cover picture that he had his cover picture empty before it, so he would be going to the side that is for displaying his own cover picture and clicks the button of uploading his cover picture in the dropdown list, then the user will choose from his device pictures and then he would be able to crop it and process on this picture then internal the after clicking the button upload the button upload will be triggering the click event with the user id and the cover picture chosen by the user.</p> <p>After that, the media file uploader will be uploading the cover picture and then it sends XMLHttpRequest with the CSRF header and the user id to nodeJS side which will validate the header with the php side then nodeJS side will be inserting the cover picture as a blob into MongoDB and then nodeJs will be delivering the token generated from mongoDb to the vue component and then the cue component will be sending acknowledgement to the media file uploader then the media file uploader will be emitting the media query logged to the app API.</p> <p>Finally, the app API will be sending an acknowledgement to the due component which will trigger the process of preparing data to be submitted with the given token by the due component and also the vue component will be responsible for the final submit sent for the app API and after that due component sends an acknowledgement to the click event listener so that the media file uploader after listening on this click event listener it will send XMLHttpRequest with the token and user id to the php side which will validate XMLHttpRequest with the header then it inserts the cover picture with token and the user id into mysql database.</p>
What can go wrong	<ol style="list-style-type: none"> If the content of the cover picture is empty, so that the media query logged has false as a response from the media file uploader to store in mongoDB so that the process terminates with a failure message If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.
Other Activities	The user can proceed all operations allowed while uploading a cover picture but not allowed to edit the privacy sets that has been regulated previously while the uploading operation is in progress
System on Completion	A success message that the cover picture has been uploaded successfully and safely inserted into the database, or a failure message that the condition of uploading a cover picture has not been satisfied.

3.1.17. Update a profile picture

Scenario ID	Knocks-17
Scenario Name	Update a profile picture
Actors	User
Initial Assumption	The user is logged in and he wants to update his profile picture
Normal	<p>The user will be uploading his profile picture that he had his profile picture set before it, so he would be going to the side that is for displaying his own profile picture and clicks the button of updating his profile picture in the dropdown list, then the user will choose from his device pictures and then he would be able to crop it and process on this picture then internal the after clicking the button upload the button update will be triggering the click event with the user id and the profile picture chosen by the user.</p> <p>After that, the media file uploader will be updating the profile picture and then it sends XMLHttpRequest with the CSRF header and the user id to nodeJS side which will validate the header with the php side then nodeJS side will be inserting the profile picture as a blob into MongoDB and then nodeJs will be delivering the token generated from mongoDb to the vue component and then the cue component will be sending acknowledgement to the media file uploader then the media file uploader will be emitting the media query logged to the app API.</p> <p>Finally, the app API will be sending an acknowledgement to the due component which will trigger the process of preparing data to be submitted with the given token by the due component and also the vue component will be responsible for the final submit sent for the app API and after that due component sends an acknowledgement to the click event listener so that the media file uploader after listening on this click event listener it will send XMLHttpRequest with the token and user id to the php side which will validate XMLHttpRequest with the header then it updates the profile picture with token and the user id into mysql database.</p>
What can go wrong	<ol style="list-style-type: none"> If the content of the profile picture is empty, so that the media query logged has false as a response from the media file uploader to store in mongoDB, so that the process terminates with a failure message If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.
Other Activities	The user can proceed all operations allowed while updating a profile picture but not allowed to edit the privacy sets that has been regulated previously while the updating operation is in progress
System on Completion	A success message that the profile picture has been updated successfully and safely updated into the database, or a failure message that the condition of updating a profile picture has not been satisfied.

3.1.18. Update a cover picture

Scenario ID	Knocks-18
Scenario Name	Update a cover picture
Actors	User
Initial Assumption	The user is logged in and he wants to update his cover picture
Normal	<p>The user will be uploading his cover picture that he had his cover picture set before it, so he would be going to the side that is for displaying his own cover picture and clicks the button of updating his cover picture in the dropdown list, then the user will choose from his device pictures and then he would be able to crop it and process on this picture then internal the after clicking the button upload the button update will be triggering the click event with the user id and the cover picture chosen by the user.</p> <p>After that, the media file uploader will be updating the cover picture and then it sends XMLHttpRequest with the CSRF header and the user id to nodeJS side which will validate the header with the php side then nodeJS side will be inserting the cover picture as a blob into MongoDB and then nodeJs will be delivering the token generated from mongoDb to the vue component and then the cue component will be sending acknowledgement to the media file uploader then the media file uploader will be emitting the media query logged to the app API.</p> <p>Finally, the app API will be sending an acknowledgement to the vue component which will trigger the process of preparing data to be submitted with the given token by the due component and also the vue component will be responsible for the final submit sent for the app API and after that due component sends an acknowledgement to the click event listener so that the media file uploader after listening on this click event listener it will send XMLHttpRequest with the token and user id to the php side which will validate XMLHttpRequest with the header then it updates the cover picture with token and the user id into mysql database.</p>
What can go wrong	<ol style="list-style-type: none"> If the content of the cover picture is empty, so that the media query logged has false as a response from the media file uploader to store in mongoDB, so that the process terminates with a failure message If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.
Other Activities	The user can proceed all operations allowed while updating a cover picture but not allowed to edit the privacy sets that has been regulated previously while the updating operation is in progress
System on Completion	A success message that the cover picture has been updated successfully and safely updated into the database, or a failure message that the condition of updating a cover picture has not been satisfied.

3.1.19. Unfriend a friend

Scenario ID	Knocks-19
Scenario Name	Unfriend a friend
Actors	User
Initial Assumption	The user is logged in and he wants to unfriend one of his friends
Normal	The user will be selecting a specific friend and then he would press the button unfriend to remove him from his friends list so that php side will be validating the friend if he is already in the account user friends by fetching in the circles which the account user have in mysql database and if the validation returns success then the php side will be updating the privacy setoff this user with the given account owner id and the other user id then it should be removing this user from all circles that the account user have.
What can go wrong	1. If the account owner does not have the user id in any of the circles.
Other Activities	The user can proceed all operations allowed while unfriending a friend , except for updating the privacy set fo the specific user or excluding members from the circles where the user exist in
System on Completion	A success message that the specific friend has been unfriended successfully and safely removed from the account user circles in the database, or a failure message that the condition of unfriending that friend has not been satisfied.

3.1.20. Report a user

Scenario ID	Knocks-20
Scenario Name	Report a user
Actors	User
Initial Assumption	The user is logged in and he wants to report a user
Normal	The user will be selecting a specific friend and then he would press the button report to report this specific user with a reason that can be given by the user optionally.
What can go wrong	1. If the account owner has reported this user before
Other Activities	The user can proceed all operations allowed while reporting a friend but he cannot logout until the reporting process completes
System on Completion	A success message that the specific user has been reported successfully, or a failure message that the condition of reporting the user has not been satisfied.

3.1.21. Blocking a friend

Scenario ID	Knocks-21
Scenario Name	Blocking a friend
Actors	User
Initial Assumption	The user is logged in and he wants to block a friend
Normal	The user will be selecting a specific friend and then he would press the button block to report this specific user, so that the event of blocking is triggered by the user with the given account owner id and the user id then php side would validate the user id by checking in mysql database whether this user exists in any of the circles which the account user owns and then the privacy user of this user will be updated in the database with the invalid for all privacy set with the given account owner id and the user id and finally the blocking list of the account owner will be updated in the database with the new user id.
What can go wrong	1. If the user does not exist in any circle which the account owner have.
Other Activities	The user can proceed all operations allowed while blocking a user except for updating the privacy set for that specific user or remove friends from the circles where the user reside.
System on Completion	A success message that the specific user has been blocked successfully, or a failure message that the condition of blocking the user has not been satisfied.

3.1.22. Update personal information

Scenario ID	Knocks-22
Scenario Name	Update personal information
Actors	User
Initial Assumption	The user is logged in and he wants to update personal information
Normal	The user will be selecting a specific field in his own profile and after supplying the field with the correct information he would press the button update with his own id and the personal information will be used by the php side to update this failed in mysql database.
What can go wrong	1. If the user submitted the fields with empty submission
Other Activities	The user can proceed all operations allowed while updating personal information but cannot update the privacy set of his own personal information until the updating process completes
System on Completion	A success message that the desired personal information is updated successfully, or a failure message that the condition of updating the user personal information has not been satisfied.

3.1.23. React to post

Scenario ID	Knocks-23
Scenario Name	React to a post
Actors	User
Initial Assumption	The user is logged in and he wants to react to a specific post
Normal	While the user browses the timeline he would stop at a post and would like to hit a reaction, so that after the reaction have been pressed then the php side will be taken the arguments reaction name and the reactor id and the post id to validate the the post owner age so that if he is a kid then some reactions cannot be reacted to the user like the angry reaction, then after reaction validation returns success php will insert the reaction into mysql database with the reactor id and the reaction with the post id.
What can go wrong	1. If the reaction receiver user is not validated so that he cannot receive the chosen reaction type
Other Activities	The user can proceed all operations allowed while reacting to post but the post cannot be updated or deleted until the reacting process is completed.
System on Completion	A success message that the desired personal information is updated successfully, or a failure message that the condition of updating the user personal information has not been satisfied.

3.1.24. React to comment

Scenario ID	Knocks-24
Scenario Name	React to a Comment
Actors	User
Initial Assumption	The user is logged in and he wants to react to a specific comment
Normal	While the user browses the timeline he would stop at a comment and would like to hit a reaction, so that after the reaction have been pressed then the php side will be taken the arguments reaction name and the reactor id and the comment id to validate the the comment owner age so that if he is a kid then some reactions cannot be reacted to the user like the angry reaction, then after reaction validation returns success php will insert the reaction into mysql database with the reactor id and the reaction with the comment id.
What can go wrong	1. If the reaction receiver user is not validated so that he cannot receive the chosen reaction type
Other Activities	The user can proceed all operations allowed while reacting to comment but the comment cannot be updated or deleted until the reacting process is completed.
System on Completion	A success message that the desired reaction is added to the specified post successfully, or a failure message that the condition of reacting to the post has not been satisfied.

3.1.25. React to reply

Scenario ID	Knocks-25
Scenario Name	React to a reply
Actors	User
Initial Assumption	The user is logged in and he wants to react to a specific reply
Normal	While the user browses the timeline he would stop at a reply and would like to hit a reaction, so that after the reaction have been pressed then the php side will be taken the arguments reaction name and the reactor id and the reply id to validate the the reply owner age so that if he is a kid then some reactions cannot be reacted to the user like the angry reaction, then after reaction validation returns success php will insert the reaction into mysql database with the reactor id and the reaction with the reply id.
What can go wrong	1. If the reaction receiver user is not validated so that he cannot receive the chosen reaction type
Other Activities	The user can proceed all operations allowed while reacting to reply but the reply cannot be updated or deleted until the reacting process is completed.
System on Completion	A success message that the desired reaction is added to the specified reply successfully, or a failure message that the condition of reacting to the reply has not been satisfied.

3.1.26. Chat with a friend

Scenario ID	Knocks-26
Scenario Name	Chat with a friend
Actors	User
Initial Assumption	The user is logged in and wants to chat with a friend
Normal	<p>the user enters the message part of himself and his friends and it writes his own message then clicks the button send in the user interface with the given sender id and receiver id as they both are logged in already and the message, then the click button emits the click event which will be listened by the record event listener and the media file uploader event listener and if the message has media blobs then the media is uploaded and then the record and media file uploader sends an XMLHttpRequest with the headers that mainly contains the CSRF of the user and the userid to the NodeJS and then NodeJs validates the CSRF header with the PHP side and if validated successfully then NodeJS inserts the blobs (the contained media) into the MongoDB with the userid then mongoDb responds after insertion with a token that acts as the index of the blob in the database then nodeJs retrieves the token to the vue component, after that the record event listener and the media file uploader event listeners are emitting the media query logged with the data included to the due component knock then the knock prepares and submit the data, and sends the final sub the App API and the knock button sends the XMKHTTPRequest to the PHP side and then PHP validates the XMLHttpRequest and data and inserts the data and the sender id and receiver id into Mysql database in the envelopes table and then PHP and sender and receiver are acknowledged with successful message.</p>
What can go wrong	<ol style="list-style-type: none"> If the content of the message is empty, so that the media query logged has false as a response from the media file uploader and the recorder to store in mongoDB, and it has no plain text to store in MySql so that the process terminates with a failure message If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side. If the receiver is already blocking the sender user so that the message will not be sent to the receiver successfully.
Other Activities	The user can proceed all operations allowed while chatting but not allowed to delete the message the is still sending until the sending operation finishes
System on Completion	A success message that the message has been sent successfully and safely inserted into the database which can be retrieved securely, or a failure message that the condition of sending the message has not been satisfied.

3.1.27.Create a group chat

Scenario ID	Knocks-27
Scenario Name	Create a group chat
Actors	User
Initial Assumption	The user is logged in and wants to create a group chat
Normal	<p>The user will be selecting a group of friends and then he will press the button "create a group chat" through the interface provided, then the click button emits the click event with the users ids which will be listened by the record event listener and the media file uploader event listener and if the message has media blobs then the media is uploaded and then the record and media file uploader sends an XMLHttpRequest with the headers that mainly contains the CSRF of the user and the userid to the NodeJS and then NodeJs validates the CSRF header with the PHP side and if validated successfully then NodeJS inserts the blobs (the contained media) into the MongoDB with the userid then mongoDb responds after insertion with a token that acts as the index of the blob in the database then nodeJs retrieves the token to the vue component, after that the record event listener and the media file uploader event listeners are emitting the media query logged with the data included to the due component knock then the knock prepares and submit the data, and sends the final sub the App API and the knock button sends the XMKHTTPRequest to the PHP side and then PHP validates the XMLHttpRequest and data and inserts the data and the users ids into Mysql database in the envelopes table and then PHP and sender and receiver are acknowledged with successful message.</p>
What can go wrong	<ol style="list-style-type: none"> If the content of the message is empty, so that the media query logged has false as a response from the media file uploader and the recorder to store in mongoDB, and it has no plain text to store in MySql so that the process terminates with a failure message If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side. If the receiver user is already blocking the sender user so that the message cannot be sent to the receiver successfully.
Other Activities	The user can proceed all operations allowed while chatting but not allowed to delete the message the is still sending until the sending operation finishes
System on Completion	A success message that the group chat message has been sent successfully and safely inserted into the database which can be retrieved securely, or a failure message sending that the condition of group chat message has not been satisfied.

3.1.28.Update a group chat

Scenario ID	Knocks-28
Scenario Name	Update a group chat
Actors	User
Initial Assumption	The user is logged in and wants to update his own group chat by removing some users from it
Normal	The user will enter the group chat and he would exclude the group chat members that he selects and then he will press the button update that will emit the event to the php side with the user id and the other group chat members ids and then the php side will validate the users and check if they are already in a group chat with the account owner or not , then after validation the php side will update the members in the group.
What can go wrong	<ol style="list-style-type: none">1. If the specified users to be removed are not already in any group chat with the account owner2. If the account owner has no privilege to remove or add users to the group chat
Other Activities	The user can proceed all operations allowed while updating a group chat but he cannot add or remove users from the group chat unless the updating process is done.
System on Completion	A success message that the group chat updating has been done successfully and safely updated in the database, or a failure message that the condition of group chat updating has not been satisfied.

3.1.29.Delete a group chat

Scenario ID	Knocks-29
Scenario Name	Delete a group chat
Actors	User
Initial Assumption	The user is logged in and wants to delete his own group chat.
Normal	The user will enter the group chat and he would press the button delete that will emit the event to the php side with the user id and the other group chat members ids and then the php side will validate the users and check if they are already in a group chat with the account owner or not , then after validation the php side will delete all the envelopes by those users and removing all the users from the group chat.
What can go wrong	<ol style="list-style-type: none">1. If the specified users to be removed are not already in any group chat with the account owner2. If the account owner has no privilege to remove or add users to the group chat
Other Activities	The user can proceed all operations allowed while deleteng a group chat but he cannot add or remove users from the group chat unless the updating process is done.
System on Completion	A success message that the group chat deletion has been done successfully and safely deleted from the database, or a failure message that the condition of group chat deletion has not been satisfied.

3.1.30.React to messages

Scenario ID	Knocks-30
Scenario Name	React to message
Actors	User
Initial Assumption	The user is logged in and he wants to react to a message that he received
Normal	<p>The user will enter the messages section between him and his friend and he would press on the reactions to view the reactions and then he will choose one by pressing on it so that this click emits an event to the php side with the object id which the message selected and the user id who reacts and the reaction type then the php side will validate the user age by fetching for the owner of the object id to see if he is allowed to receive the reaction or not.</p> <p>After the validation then php side will insert the reaction in mysql database with the reaction type and the reactor user id.</p>
What can go wrong	<ol style="list-style-type: none">1. If the reaction type cannot be proceeded to be received by the receiver user.2. If the account owner has no privilege to remove or add users to the group chat
Other Activities	The user can proceed all operations allowed while
System on Completion	A success message that the message has been reacted to successfully and the second party can display this reaction, or a failure message that the condition of reacting to a message has not been satisfied.

3.1.31.Create a group

Scenario ID	Knocks-31
Scenario Name	Create a group
Actors	User
Initial Assumption	The user is logged in and he wants to react to a create a group
Normal	The user will be clicking the button add a new group and then he will provide the group name , the group picture, the group thumbnail, the group category, the group preset and the user id who creates this group and then after he presses this button he will be emitting the event of adding a group from the php side with the given group name, thumbnail, picture, category and preset , and then after the insertion process is completed then the group will have the new one member who already created this group with the position admin and it is inserted in the sql database in the group members table.
What can go wrong	<ol style="list-style-type: none">1. If the user have submitted the group failed with empty submission of one of the required fields
Other Activities	The user can proceed all operations allowed while creating a new group but he cannot add member of the currently creating group unless the creation process already finishes.
System on Completion	A success message that the group has been created successfully, or a failure message that the condition of group creation has not been satisfied.

3.1.32.Add friends to a group

Scenario ID	Knocks-32
Scenario Name	Add friends to a group
Actors	User
Initial Assumption	The user is logged in and he wants to add friends to the group that he is member of.
Normal	The user will be pressing the button add members when he is in the group that he is a member of and then he will selecting the user that he wants to add in the group with the user id and the group id and the position that will be labeled with this added user, and then the php side will be checking the user privilege to check if he can add members to the group or not with the group id and the user id , and then after the checking process returns success the the request will be sent to the user who will be added to the group and then after that user accepts the request then php side will be able to dd this user in the group members with the group is and the user id and the position of this user.
What can go wrong	<ol style="list-style-type: none">1. If the group is already deleted by other party so that no members can be added to that group2. If the user is not having the privilege to add member to that group according to his position in the group.
Other Activities	The user can proceed all operations allowed while adding friends to the group except for deleting the group itself or updating its members, or changing the users privileges.
System on Completion	A success message that the member has been added to the group successfully and he can display what the group posts or any other activities done in the group, or a failure message that the condition of adding a member to the group has not been satisfied.

3.1.33.Update a group

Scenario ID	Knocks-33
Scenario Name	Update a group
Actors	User
Initial Assumption	The user is logged in and he to update a group that he is a member of.
Normal	The user will be selecting the one group that he wants to update and then after changing the name or the thumbnail or the preset or the picture and then he will press the button update which will emit an event to the php side with the group id and the user id and the name for example to updated and then the php side will be checking the user privilege with the user id and the group id and the position and then after the confirmation the php side will be updating the group name with the group id and the group name.
What can go wrong	<ol style="list-style-type: none">1. If the group is already deleted by other party so that no updating can be processed on the group2. If the user is not having the privilege to update the group according to his position in the group.
Other Activities	The user can proceed all operations allowed while updating group except for deleting the group itself or changing the users privileges.
System on Completion	A success message that the group has been updated successfully and it can be now showed with the updating features, or a failure message that the condition of updating the group has not been satisfied.

3.1.34.Delete a group

Scenario ID	Knocks-34
Scenario Name	Delete a group
Actors	User
Initial Assumption	The user is logged in and he wants to delete a group that he is a member of while he is privileged to do so.
Normal	The user will be selecting the group that he wants to delete and then he will press the button delete which will emit an event to the php side with the group id and the user id and then the php side will check the user with the user id and the group id and after the validation in mysql database the php side will again check for the user privilege with the user id and the group id and the position and then after the validation return success then the php will delete the group with the group id from mysql database.
What can go wrong	1. If the user is not having the privilege to delete the group according to his position in the group.
Other Activities	The user can proceed all operations allowed while updating group except for changing the users privileges.
System on Completion	A success message that the group has been deleted successfully, or a failure message that the condition of deleting the group has not been satisfied.

3.1.35.Post in a group

Scenario ID	Knocks-35
Scenario Name	Post in a group
Actors	User
Initial Assumption	The user is logged in and he to post in group the he is a member of.
Normal	<p>The user will be entering the group that he is already a member of it and then he will post what he wants then he will hit the button publish, and with the user id and the group id and the post the click button emits the click event which will be listened by the record event listener and the media file uploader event listener and if the post have media blobs then the media is uploaded and then the record and media file uploader sends an XMLHttpRequest with the headers that mainly contains the CSRF of the user and the userid to the NodeJS and then NodeJS validates the CSRF header with the PHP side and if validated successfully then NodeJS inserts the blobs (the contained media) into the MongoDB with the userid then mongoDb responds after insertion with a token that acts as the index of the blob in the database then nodeJs retrieves the token to the vue component, after that the record event listener and the media file uploader event listeners are emitting the media query logged with the data included to the due component knock then the knock prepares and submit the data, and sends the final sub the App API and the knock button sends the XMKHTTPRequest to the PHP side and then PHP validates the XMLHttpRequest and data and inserts the data bounded to the specified user and specified group into Mysql database and then PHP and user are acknowledged with successful message.</p>
What can go wrong	<ol style="list-style-type: none"> If the content of the post is empty, so that the media query logged has false as a response from the media file uploader and the recorder to store in mongoDB, and it has no plain text to store in MySql so that the process terminates with a failure message If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side. If the user is not a member of the group where he wants to post
Other Activities	The user can proceed all operations allowed while publishing a knock in group except for editing the members positions or deleting the group itself until the posting process finishes.
System on Completion	A success message that the post is published in the group successfully, or a failure message that the condition of publishing a knock in the group has not been satisfied.

3.1.36.Comment in a group

Scenario ID	Knocks-36
Scenario Name	Comment in a group
Actors	User
Initial Assumption	The user is logged in and he to comment in group the he is a member of.
Normal	<p>The user will be entering the group that he is already a member of it and then he will comment on a specific post what he wants then he will hit the button publish, and with the user id and the group id and the comment the click button emits the click event which will be listened by the record event listener and the media file uploader event listener and if the comment have media blobs then the media is uploaded and then the record and media file uploader sends an XMLHttpRequest with the headers that mainly contains the CSRF of the user and the userid to the NodeJS and then NodeJs validates the CSRF header with the PHP side and if validated successfully then NodeJS inserts the blobs (the contained media) into the MongoDB with the user id then mongoDb responds after insertion with a token that acts as the index of the blob in the database then nodeJs retrieves the token to the vue component, after that the record event listener and the media file uploader event listeners are emitting the media query logged with the data included to the due component knock then the knock prepares and submit the data, and sends the final sub the App API and the knock button sends the XMKHTTPRequest to the PHP side and then PHP validates the XMLHttpRequest and data and inserts the data bounded to the specified user and specified group with the parent object id of the comment into Mysql database and then PHP and user are acknowledged with successful message.</p>
What can go wrong	<ol style="list-style-type: none"> If the content of the comment is empty, so that the media query logged has false as a response from the media file uploader and the recorder to store in mongoDB, and it has no plain text to store in MySql so that the process terminates with a failure message If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side. If the user is not a member of the group where he wants to comment
Other Activities	The user can proceed all operations allowed while publishing a comment in group except for editing the members positions or deleting the group itself until the commenting process finishes.
System on Completion	A success message that the comment is published in the group successfully, or a failure message that the condition of publishing a comment in the group has not been satisfied.

3.1.37.Reply in a group

Scenario ID	Knocks-37
Scenario Name	Reply in a group
Actors	User
Initial Assumption	The user is logged in and he to reply in group the he is a member of.
Normal	<p>The user will be entering the group that he is already a member of it and then he will reply on a specific post what he wants then he will hit the button publish, and with the user id and the group id and the reply the click button emits the click event which will be listened by the record event listener and the media file uploader event listener and if the reply have media blobs then the media is uploaded and then the record and media file uploader sends an XMLHttpRequest with the headers that mainly contains the CSRF of the user and the userid to the NodeJS and then NodeJs validates the CSRF header with the PHP side and if validated successfully then NodeJS inserts the blobs (the contained media) into the MongoDB with the user id then mongoDb responds after insertion with a token that acts as the index of the blob in the database then nodeJs retrieves the token to the vue component, after that the record event listener and the media file uploader event listeners are emitting the media query logged with the data included to the due component knock then the knock prepares and submit the data, and sends the final sub the App API and the knock button sends the XMKHTTPRequest to the PHP side and then PHP validates the XMLHttpRequest and data and inserts the data bounded to the specified user and specified group with the parent object id of the reply into Mysql database and then PHP and user are acknowledged with successful message.</p>
What can go wrong	<ol style="list-style-type: none"> If the content of the comment is empty, so that the media query logged has false as a response from the media file uploader and the recorder to store in mongoDB, and it has no plain text to store in MySql so that the process terminates with a failure message If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side. If the user is not a member of the group where he wants to reply
Other Activities	The user can proceed all operations allowed while publishing a replying in group except for editing the members positions or deleting the group itself until the replying process finishes.
System on Completion	A success message that the reply is published in the group successfully, or a failure message that the condition of publishing a reply in the group has not been satisfied.

3.1.38.React to an object in a group.

Scenario ID	Knocks-38
Scenario Name	React to an object in a group
Actors	User
Initial Assumption	The user is logged in and he wants to react to an object included in the group which he is a member of.
Normal	<p>The user will be navigating in the group that he joined before and then he will react to an object in this group, this object can be post or comment or reply.</p> <p>The user will choose the desired reaction type to react on this object and then he will press on it, so that an event is emitted by this press with the user id and the group id and the object id and the reaction type to the php side then the php side would check the object with the given object id and the group id to make sure that this object has been published in this specific group.</p> <p>Then the php side would check for the user privilege in this group with the user id and the group id to make sure that this user who reacts to the object has the privilege to react to an object in the group.</p> <p>Then the php side will check the reaction type itself with the object id and reaction type to assure that this chosen reaction can be reacted to this object according to its user id and checking his age.</p> <p>Then the php side will insert the reaction with the user id who reacts and the object id and the group id and the reaction type, and also if this object is a comment or a reply that it has a parent id then the parent id is also considered in the insertion process.</p>
What can go wrong	<ol style="list-style-type: none"> 1. If the selected object is not in the group anymore or it has been deleted while this process is in progress. 2. If the user privilege does not contain the reaction privilege on an object in the group. 3. If the reaction type is not compatible with the age of the objects owner. 4. If the group has been banning the member who reacts at the moment 5. If the group has been deleted. 6. If the member has been blocked from the group
Other Activities	The user can proceed all operations allowed while reacting to an object in a group except for deletion of the object itself or process on this object, or also changing the group member privileges until the process of reacting finishes, or also the group deletion, the member banning, or blocking the user who reacts.
System on Completion	A success message that the object reaction in a group is inserted successfully, or a failure message that the condition of reacting to an object in a group has not been satisfied.

3.1.39.Change a group privacy preset

Scenario ID	Knocks-38
Scenario Name	Change a group privacy preset
Actors	User
Initial Assumption	The user is logged in and he wants to change the privacy preset of the group that he is a member of.
Normal	<p>The user will be selecting the desired page where he wants to update the privacy preset, so that he will press the button update privacy preset and then he will choose the demanded privacy preset then confirms his choice.</p> <p>This will emit an event to the php side with the user id who wants to update and the new desired privacy preset and the group id which will make the php side to check the user with the group id and the user id to check if this user will be able to update the preset with his current privilege or not and also check if he is still a member of this group or not.</p> <p>Then the php will insert the new privacy preset with the given group id and the new privacy preset.</p>
What can go wrong	<ol style="list-style-type: none"> 1. If the user privilege does not contain the preset changing privilege in the group. 2. If the group has been banning the member who reacts at the moment 3. If the group has been deleted. 4. If the member has been blocked from the group
Other Activities	The user can proceed all operations allowed while changing the privacy preset of a group except for changing the group member privileges until the process of updating finishes, or also the group deletion, the member banning, or blocking the user who updates.
System on Completion	A success message that the privacy preset updating process in a group has been updated successfully, or a failure message that the condition of changing the privacy preset in a group has not been satisfied.

3.1.40.Report a group

Scenario ID	Knocks-40
Scenario Name	Report a group
Actors	User
Initial Assumption	The user is logged in and he wants to report a group.
Normal	The user will be selecting a specific group and then he would press the button report with a reason that can be given by the user optionally.
What can go wrong	1. If the user has reported this group before
Other Activities	The user can proceed all operations allowed while reporting a group but he cannot logout until the reporting process completes
System on Completion	A success message that the specific group has been reported successfully, or a failure message that the condition of reporting the group has not been satisfied.

3.1.41. View user information

Scenario ID	Knocks-41
Scenario Name	View user information
Actors	User
Initial Assumption	The user is logged in and he wants to view other user profile details.
Normal	<p>The user will be selecting a specific user in order to view his information and then he would press on his name, which will emit an event to the php side with the user1 id and the user2 id, and then the php side will validate the user1 privilege towards the user2 in the user privacy set and then in the circle privacy set if the validation in the user privacy set gives no result.</p> <p>Then after validation the user will be able to view the other user's allowed information.</p>
What can go wrong	<ol style="list-style-type: none">1. If the user has reported this requester user before.2. If the user has blocked the requester.3. If the privilege of the user1 doesn't satisfy the information viewing for user2.
Other Activities	The user can proceed all operations allowed while viewing other user information except for the other user to change his own privacy set towards the requester user or block him until the process finishes.
System on Completion	A success message that the requester user have viewed the other user's allowed information successfully, or a failure message that the condition of viewing other user's information has not been satisfied.

3.1.42.Explore Talents

Scenario ID	Knocks-42
Scenario Name	Explore Talents
Actors	User
Initial Assumption	The user is logged in and wants to explore the talents available
Normal	<p>The user will be entering the talents section from his own account by clicking the explore talent button.</p> <p>The clicking event will be triggered with the user id given to the php side to retrieve all existing talents after ordering it ascending according to the date of publishing from Mysql database .</p>
What can go wrong	1. If the user is not logged in then he won't be able to explore talents exist
Other Activities	The user can proceed all operations allowed while exploring talents but he cannot log out until the talents has been retrieved or else the retrieval process will fail.
System on Completion	A success message that talents are retrieved successfully and so it can be displayed for the requesting user, or a failure message that the conditions of retrieving existing talents cannot be satisfied to accomplish this function.

3.1.43.Filter talents

Scenario ID	Knocks-43
Scenario Name	Filter talents
Actors	User
Initial Assumption	The user is logged in and wants to filter talents
Normal	<p>The user will be entering the talents section from his own account by clicking the filter talents button.</p> <p>The click will trigger an event to pass the user id and the talent name or the category or by a search key to the php side and then the php side will be displaying the the specified group of talents to the requesting user</p>
What can go wrong	<ol style="list-style-type: none">1. If the user is not logged in then he won't be able to filter talents2. If the user have entered an invalid search key for filtering
Other Activities	The user can proceed all operations allowed while filtering talents but he will not be able to log out until the specified filtered talents are retrieved to the user or else the filtering process will be called off
System on Completion	A success message that the talents are filtered and displayed according to the search key successfully, or a failure message that the conditions of filtering talents cannot be satisfied to accomplish this function.

3.1.44.Upload talent object

Scenario ID	Knocks-44
Scenario Name	Upload talent object
Actors	User
Initial Assumption	The user is logged in and wants to upload his own talent object.
Normal	<p>The user will be entering the talents section from his own account by clicking the add talent button.</p> <p>Then the given userid and the talent object and the talent category and the talent specificityon if any as he is logged in already, then the click button emits the click event which will be listened by the record event listener and the media file uploader event listener and if the post have media blobs then the media is uploaded and then the record and media file uploader sends an XMLHttpRequest with the headers that mainly contains the CSRF of the user and the userid to the NodeJS and then NodeJsvalidates the CSRF header with the PHP side to pass the validation operation to mysql database and if validated successfully then NodeJS inserts the blobs (the contained media) into the MongoDB with the userid then mongoDb responds after insertion with a token that acts as the index of the blob in the database then nodeJs retrieves the token to the vue component, after that the record event listener and the media file uploader event listeners are emitting the media query logged with the data included to the due component knock then the knock prepares and submit the data, and sends the final sub the App API and the knock button sends the XMLHttpRequest to the PHP side and then PHP validates the XMLHttpRequest and data that contains the talent object and the talent category and any specification for this talent if needed and inserts the data and the user id into Mysql database and then PHP and user are acknowledged with successful message.</p>
What can go wrong	<ol style="list-style-type: none"> If the user is not logged in then he won't be able toads a talent object. If the content of the talent is empty, so that the media query logged has false as a response from the media file uploader and the recorder to store in mongoDB, and it has no talent category or talent specification to store in MySQL so that the process terminates with a failure message If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.
Other Activities	The user can proceed all operations allowed while uploading a talent object but he cannot log out until the talent object is uploaded or else the uploading process will fail.
System on Completion	A success message that talent has been uploaded successfully and so it can be displayed for all users, or a failure message that the conditions of uploading a new talent cannot be satisfied to accomplish this function.

3.1.45.Explore talents hashtags

Scenario ID	Knocks-45
Scenario Name	Explore talents hashtags
Actors	User
Initial Assumption	The user is logged in and wants to explore the talents hashtags available
Normal	<p>The user will be entering the talents section from his own account by clicking the explore talents hashtags button.</p> <p>The clicking event will be triggered with the user id given to the php side to retrieve all existing talents after ordering it ascending according to the most trending hashtags from Mysql database .</p>
What can go wrong	<ol style="list-style-type: none">1. If the user is not logged in then he won't be able to explore talents hashtags exist
Other Activities	The user can proceed all operations allowed while exploring talents hashtags but he cannot log out until the talents hashtags has been retrieved or else the retrieval process will fail.
System on Completion	A success message that talents hashtags are retrieved successfully and so it can be displayed for the requesting user, or a failure message that the conditions of retrieving existing talents hashtags cannot be satisfied to accomplish this function.

3.1.46.Rate talent object

Scenario ID	Knocks-46
Scenario Name	Rate talent object
Actors	User
Initial Assumption	The user is logged in and wants to rate talent object.
Normal	<p>The user will be selecting the specific talent object and then he will be clicking the button “rate this talent”.</p> <p>This button clicking will be triggering the event of passing the user id who rates the talent object and the talent id that will be rated and the rate value to the php side.</p> <p>The php side will be checking the user privilege and validate if he can rate this talent object or not by the given user id and the talent id.</p> <p>After validation returns success the rate will be inserted with the given user id and the talent id and the rate value.</p>
What can go wrong	<ol style="list-style-type: none"> 1. If the user is not logged in then he won't be able to explore talents hashtags exist 2. If the user has no privilege to rate this specific talent object 3. If the rate is empty, and has no value 4. The talent owner deleting his talent object while the other user is rating.
Other Activities	The user can proceed all operations allowed while rating a talent object but he cannot enter an empty value for the rate.
System on Completion	A success message that talent rate has been inserted successfully and so it can be displayed later, or a failure message that the conditions of rating existing talents hashtags cannot be satisfied to accomplish this function.

3.1.47. View talent career

Scenario ID	Knocks-47
Scenario Name	Filter talents
Actors	User
Initial Assumption	The user is logged in and wants to view talents
Normal	<p>The user will be entering the talents section from his own account by clicking the talents button and then he will provide the talent owner and the talent category then clicks confirm .</p> <p>The click will trigger an event to pass the user id and the talent category to the php side and then the php side will be displaying the the specified group of talents to the requesting user</p>
What can go wrong	<ol style="list-style-type: none">1. If the user is not logged in then he won't be able to filter talents2. If the user have entered an invalid talent owner name or talent category name for filtering
Other Activities	The user can proceed all operations allowed while filtering talents but he will not be able to log out until the specified filtered talents are retrieved to the user or else the filtering process will be called off
System on Completion	A success message that the talents are viewed and displayed according to the talent owner and talent category successfully, or a failure message that the conditions of viewing talents career cannot be satisfied to accomplish this function.

3.1.48.Rate talent owner

Scenario ID	Knocks-48
Scenario Name	Rate talent owner
Actors	User
Initial Assumption	The user is logged in and wants to rate talent owner.
Normal	<p>The user will be selecting the specific talent object and then he will be clicking the button “rate this talent owner”. This button clicking will be triggering the event of passing the user id who rates the talent owner and the talent id that will be rated and the rate value to the php side. The php side will be checking the user privilege and validate if he can rate this talent user to see if he is a kid or an adult to choose the right rate for him or not by the given user id and the talent id. After validation returns success the rate will be inserted with the given user id and the talent owner id and the rate value.</p>
What can go wrong	<ol style="list-style-type: none"> 1. If the user is not logged in then he won't be able to rate this talent owner. 2. If the user has no privilege to rate this specific talent owner. 3. If the rate is empty, and has no value.
Other Activities	The user can proceed all operations allowed while rating a talent owner but he cannot enter an empty value for the rate.
System on Completion	A success message that talent owner rate has been inserted successfully and so it can be displayed later, or a failure message that the conditions of rating existing talent owner cannot be satisfied to accomplish this function.

3.1.49. Write review for a talent object

Scenario ID	Knocks-49
Scenario Name	Write a review for a talent object
Actors	User
Initial Assumption	The user is logged in and wants to write a review for a talent object
Normal	<p>The user will be selecting the specific talent object and then he will be writing his own review for this talent object and then he will be clicking the button "submit review".</p> <p>This button clicking will be triggering the event of passing the user id who write the review and the review content and the talent id to the php side.</p> <p>The php side will be checking the user privilege to check if he can write a review for this talent or not with the given talent id and the user id.</p> <p>After the validation returns success the php side will be inserting the review with the given user id and the review content and the talent id.</p>
What can go wrong	<ol style="list-style-type: none">1. If the user is not logged in then he won't be able to rate this talent object.2. If the user has no privilege to insert his review.3. If the review is empty.
Other Activities	The user can proceed all operations allowed while writing a review for a talent object but he cannot log out until the review insertion is finished or else the review insertion operation will be cancelled.
System on Completion	A success message that talent object review has been inserted successfully and so it can be displayed later for the talent owner, or a failure message that the conditions of writing a review for the talent object cannot be satisfied to accomplish this function.

3.1.50. Write review for a talent owner

Scenario ID	Knocks-50
Scenario Name	Write a review for a talent owner.
Actors	User
Initial Assumption	The user is logged in and wants to write a review for a talent owner.
Normal	<p>The user will be selecting the specific talent owner and then he will be writing his own review for this talent owner and then he will be clicking the button "submit review".</p> <p>This button clicking will be triggering the event of passing the user id who write the review and the review content and the talent owner id to the php side.</p> <p>The php side will be checking the user privilege to check if he can write a review for this talent or not with the given talent owner id and the user id.</p> <p>After the validation returns success the php side will be inserting the review with the given user id and the review content and the talent owner id.</p>
What can go wrong	<ol style="list-style-type: none">1. If the user is not logged in then he won't be able to rate this talent owner.2. If the user has no privilege to insert his review.3. If the review is empty.
Other Activities	The user can proceed all operations allowed while writing a review for a talent owner but he cannot log out until the review insertion is finished or else the review insertion operation will be cancelled.
System on Completion	A success message that talent owner review has been inserted successfully and so it can be displayed later for the talent owner, or a failure message that the conditions of writing a review for the talent owner cannot be satisfied to accomplish this function.

3.1.51.Analysis for the talent owner

Scenario ID	Knocks-51
Scenario Name	Analysis for the talent owner.
Actors	User
Initial Assumption	The user is logged in and wants to view the analysis for his talents.
Normal	<p>The user will be entering his talents section and then clicks “analysis” button. The button being clicked will be triggering the event that will pass the user id to the php side.</p> <p>The php side will be validating the user privilege to check if he has the permission to analyze the user talents and the people rating of them.</p> <p>After the validation returns success the analysis will be retrieved according to the user id</p>
What can go wrong	<ol style="list-style-type: none"> 1. If the user is not logged in then he won’t be able to view his talent analysis. 2. If the user has no privilege to view his analysis . 3. If the user ha not been publishing a talent yet or have deleted all of them.
Other Activities	The user can proceed all operations allowed while viewing talents analysis displaying but he cannot add a new talent or do any action related to the his talents section until the analysis is accomplished.
System on Completion	A success message that the analysis is retrieved successfully and so it can be displayed for the talent owner, or a failure message that the conditions of retrieving the analysis cannot be satisfied to accomplish this function.

3.1.52.Monitor the Child Log

Scenario ID	Knocks-52
Scenario Name	Monitor the Child Log
Actors	User
Initial Assumption	The user is logged in and he is already a parent for a child account and he wants to monitor the log of his child.
Normal	<p>The user will request to monitor the log of his child by entering his kids section and then he will click the button monitor log for a specific child.</p> <p>The button clicking will trigger an event for the php side with the parent id and the child id and then php side will validate the parent will mysql database.</p> <p>After validation the php side will get the child log with the given parent id and the child id, and then the log is retrieved to the parent and displayed as he requested.</p>
What can go wrong	<ol style="list-style-type: none">1. If the user is not logged in then he won't be able to monitor the child log.2. If the user has no privilege to monitor the child log .
Other Activities	The user can proceed all operations allowed while monitoring the child log except for logging out until the monitoring process completes.
System on Completion	A success message that the child log has been retrieved and displayed successfully, or a failure message that the conditions of retrieving the child log cannot be satisfied to accomplish this function.

3.1.53. Set a parent for a Child

Scenario ID	Knocks-53
Scenario Name	Set a Parent for a child
Actors	User
Initial Assumption	The user is logged in and he wants to add a user to be recognized as a parent with the privilege of monitoring his log.
Normal	<p>The user will be entering his profile and then at the family section he will be adding his parent as a parent by selecting his account name and then clicks add. Then the php side will be validating the number of parents that child has and if it is less than two then this parent can be added with the child id, after validation the php side will request the parent user to accept as added to the child account as his parent with the parent id and the child id.</p> <p>After confirmation is created from the parent the child will be able to add the parent with the child id and the parent id.</p>
What can go wrong	<ol style="list-style-type: none"> If the user is not logged in then he won't be able to add the parent. If the parent has not confirmed the request from the child. If the number of parents for the child is more than one so that the child cannot have more than two parents to monitor his log.
Other Activities	The user can proceed all operations allowed while adding a user as a parent but he cannot delete a parent user or update a parent user or log out until the process of adding a parent is completed.
System on Completion	A success message that the parent has been added to the child account successfully, or a failure message that the conditions of adding a parent for a child account cannot be satisfied to accomplish this function.

3.1.54.Ban for a public figure

Scenario ID	Knocks-54
Scenario Name	Ban for a public figure
Actors	User
Initial Assumption	The user is logged in and he is a public figure also he wants to ban a user
Normal	The user will select a specific person and then he will click the button ban. This button clicking will trigger an event for the php side to ban this user with the user id and the public figure id to be deleted as a follower for that public figure and inserted as banned user for this public figure.
What can go wrong	1. If the user is not logged in then he won't be able to ban a user.
Other Activities	The user can proceed all operations allowed while banning a user but he cannot log out until the process of banning is completed
System on Completion	A success message that the user banning successfully, or a failure message that the conditions of user banning cannot be satisfied to accomplish this function.

3.1.55. Set an admin for a public figure

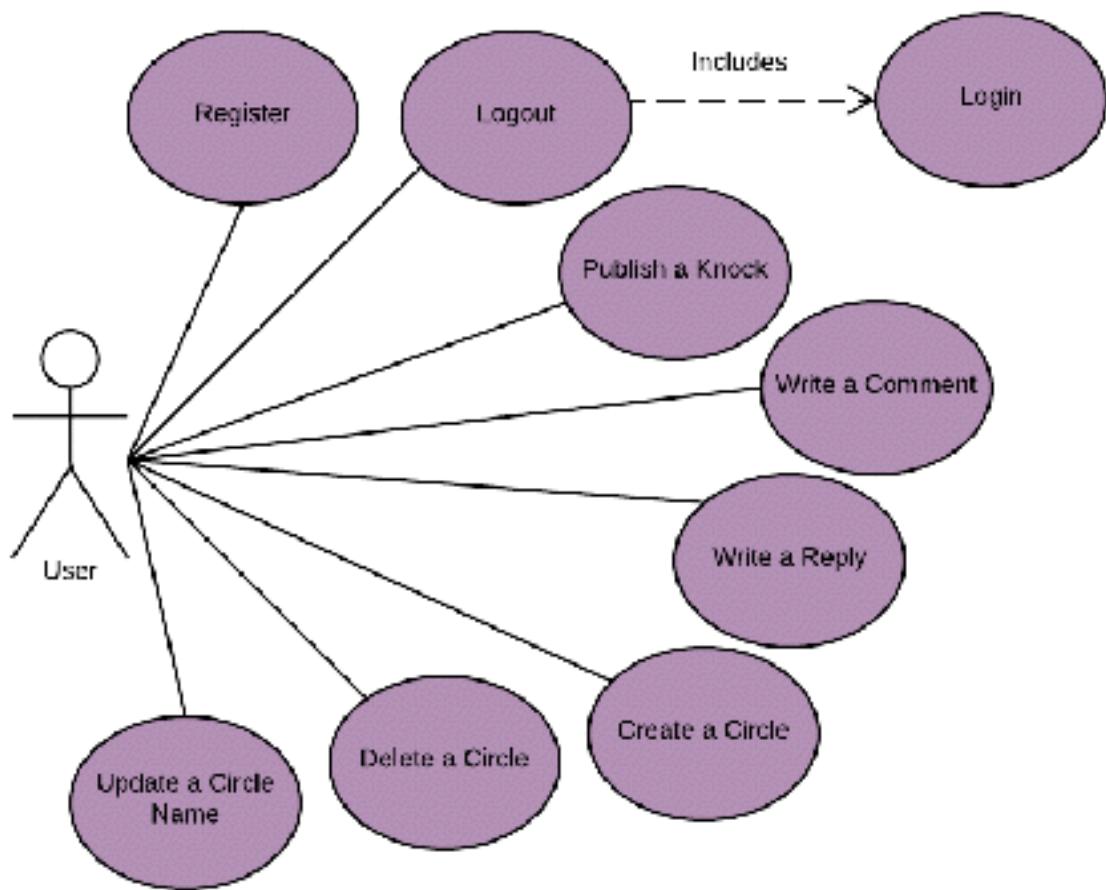
Scenario ID	Knocks-55
Scenario Name	Set an Admin for a Public Figure
Actors	User
Initial Assumption	The user is logged in and he is a public figure also he wants to add an admin for his account.
Normal	<p>The user will select specific user and then clicks the button add as an admin, this click will trigger an event for the php side with the user id and the public figure id.</p> <p>The php side will request the user to accept being an admin for the specific public figure, after the user confirms the request, the php side will add an admin with the public figure id and the user id.</p>
What can go wrong	<ol style="list-style-type: none">1. If the user is not logged in then he won't be set as an admin.2. If the user did not confirm the request of being an admin
Other Activities	The user can proceed all operations allowed while setting an admin for his public figure account but he cannot ban this user.
System on Completion	A success message that the setting an admin for a public figure process had been done successfully, or a failure message that the conditions of admin setting cannot be satisfied to accomplish this function.

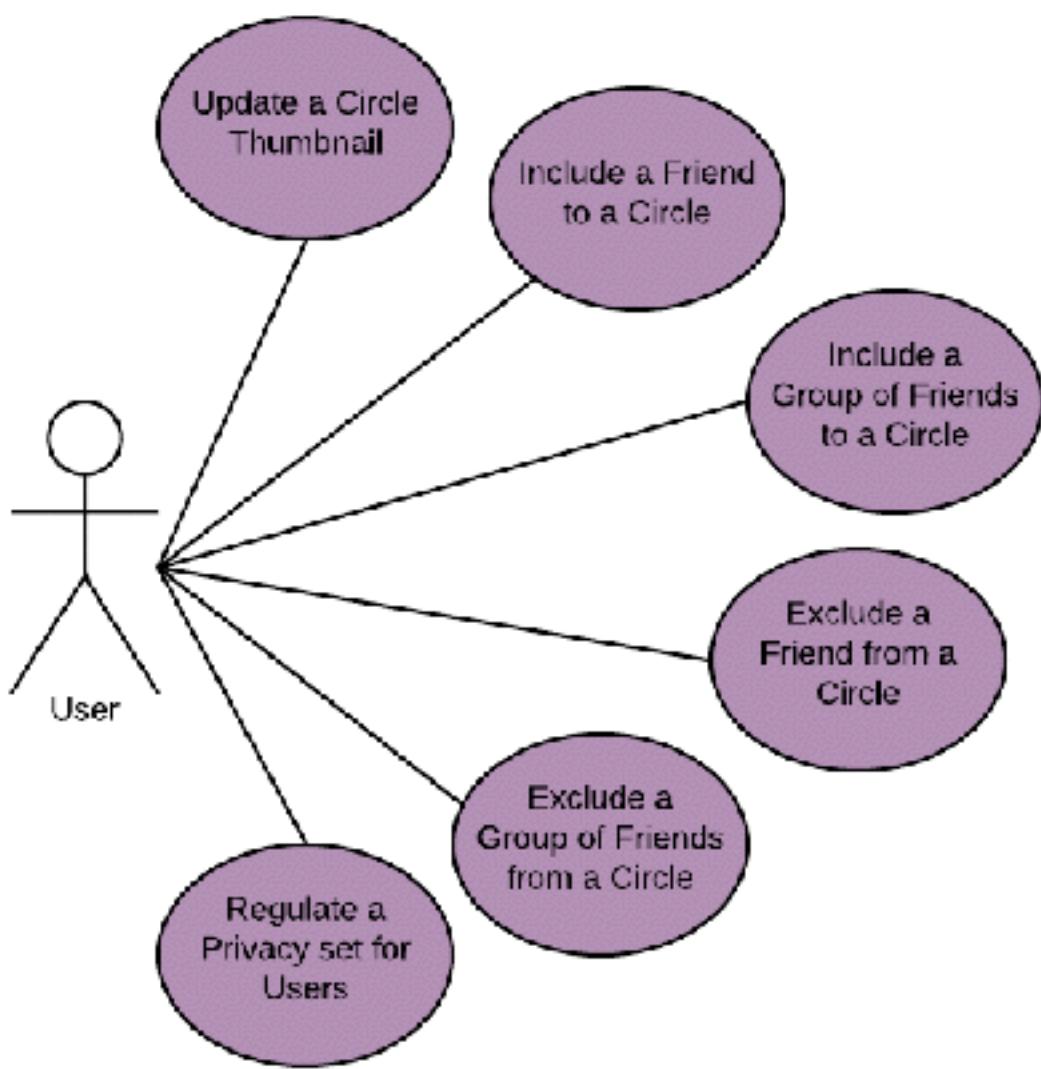
3.1.56.Monitor an Object for a Kid

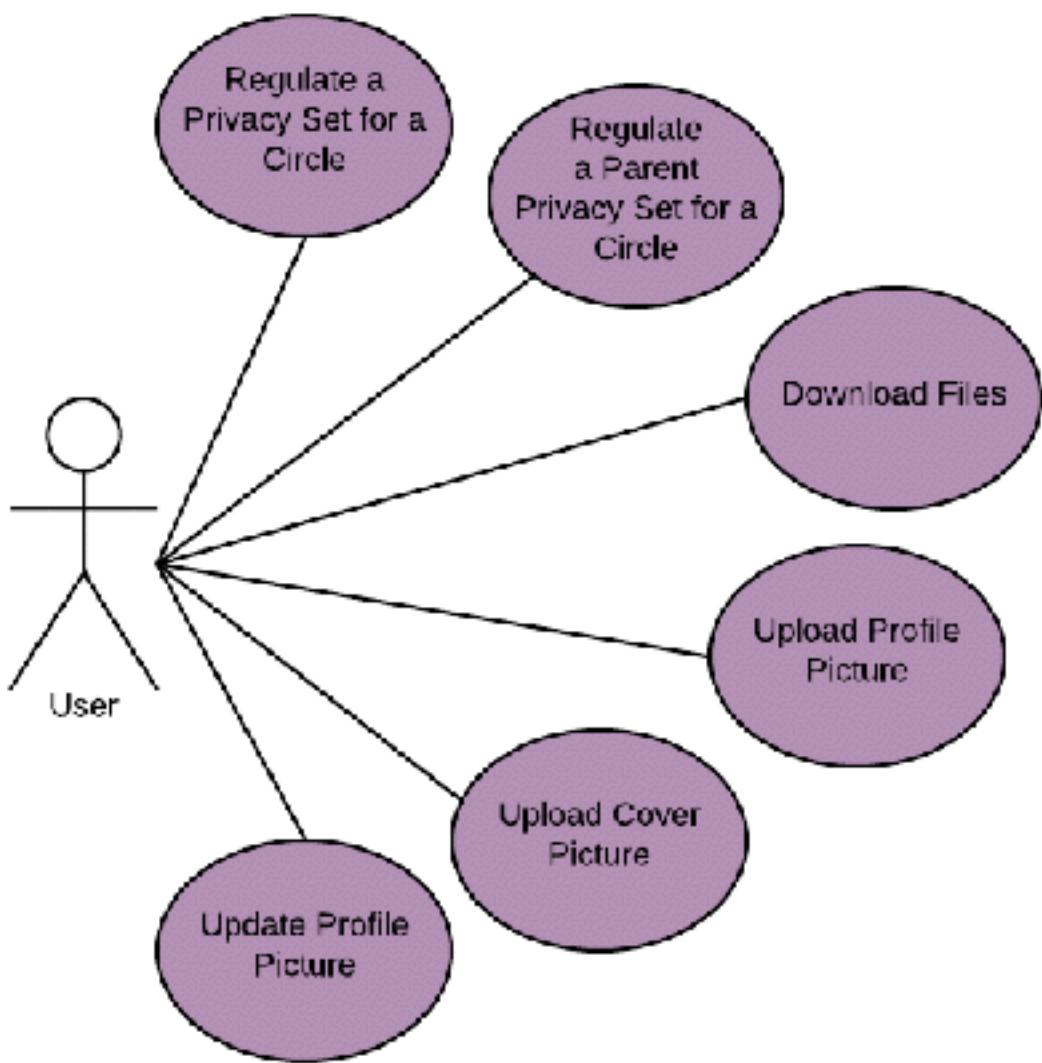
Scenario ID	Knocks-56
Scenario Name	Monitor an Object for a Kid
Actors	User
Initial Assumption	The user is logged in and he is a child and there is a content that he will receive from another user
Normal	The kid will be receiving an object with the object id and the kid id and the object creator id, ad the system will automatically monitor this object without being displayed to the kid with the php side validating the object with the object id and the kid id and after validation arriving from mysql, the php side will insert the object to be belonged to the kid with the object id and the object creator id and kid id, and then it displays the object for him
What can go wrong	1. If the user is not logged in then he won't be able monitor an object.
Other Activities	The user can proceed all operations allowed while monitoring the object.
System on Completion	A success message that the object monitoring had been done successfully, or a failure message that the conditions of object monitoring cannot be satisfied to accomplish this function.

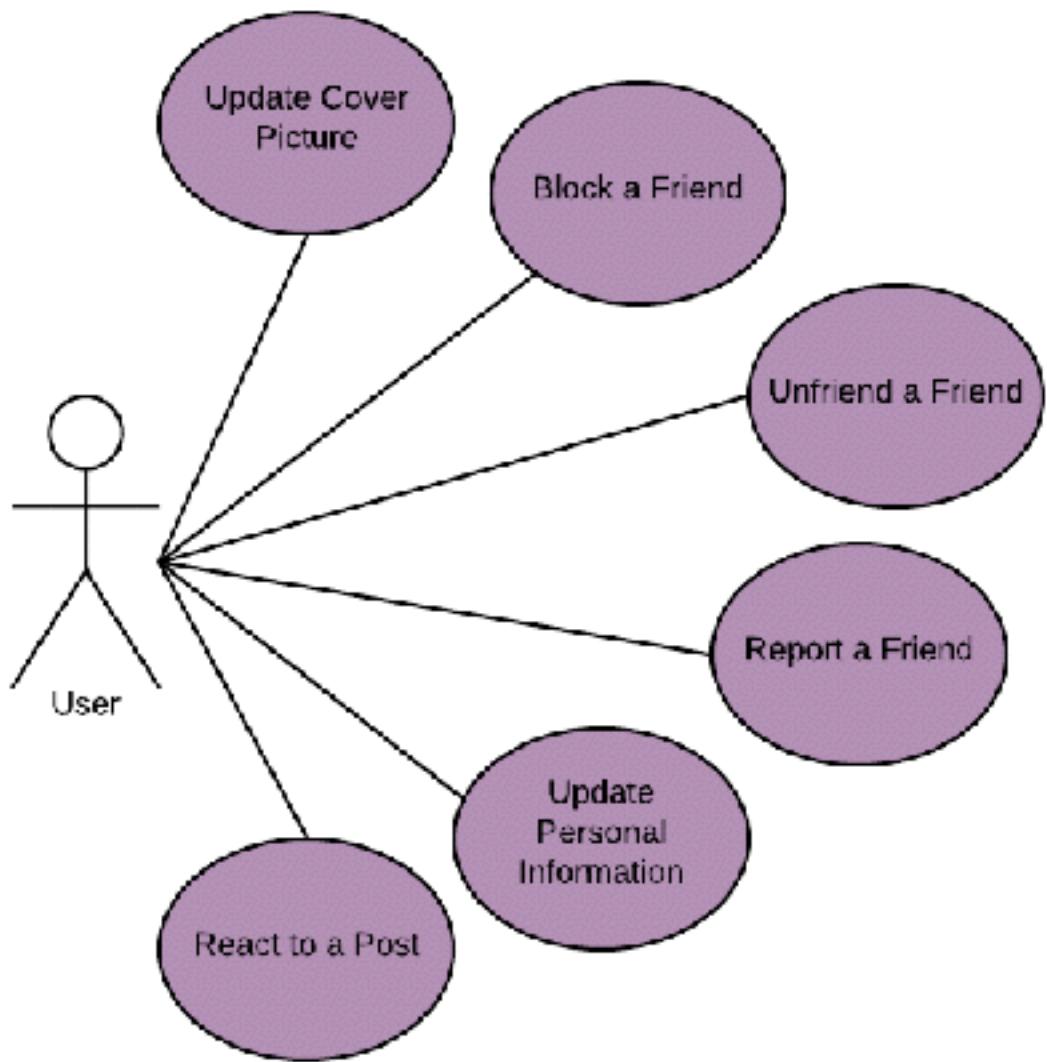
3.2. Use case Model

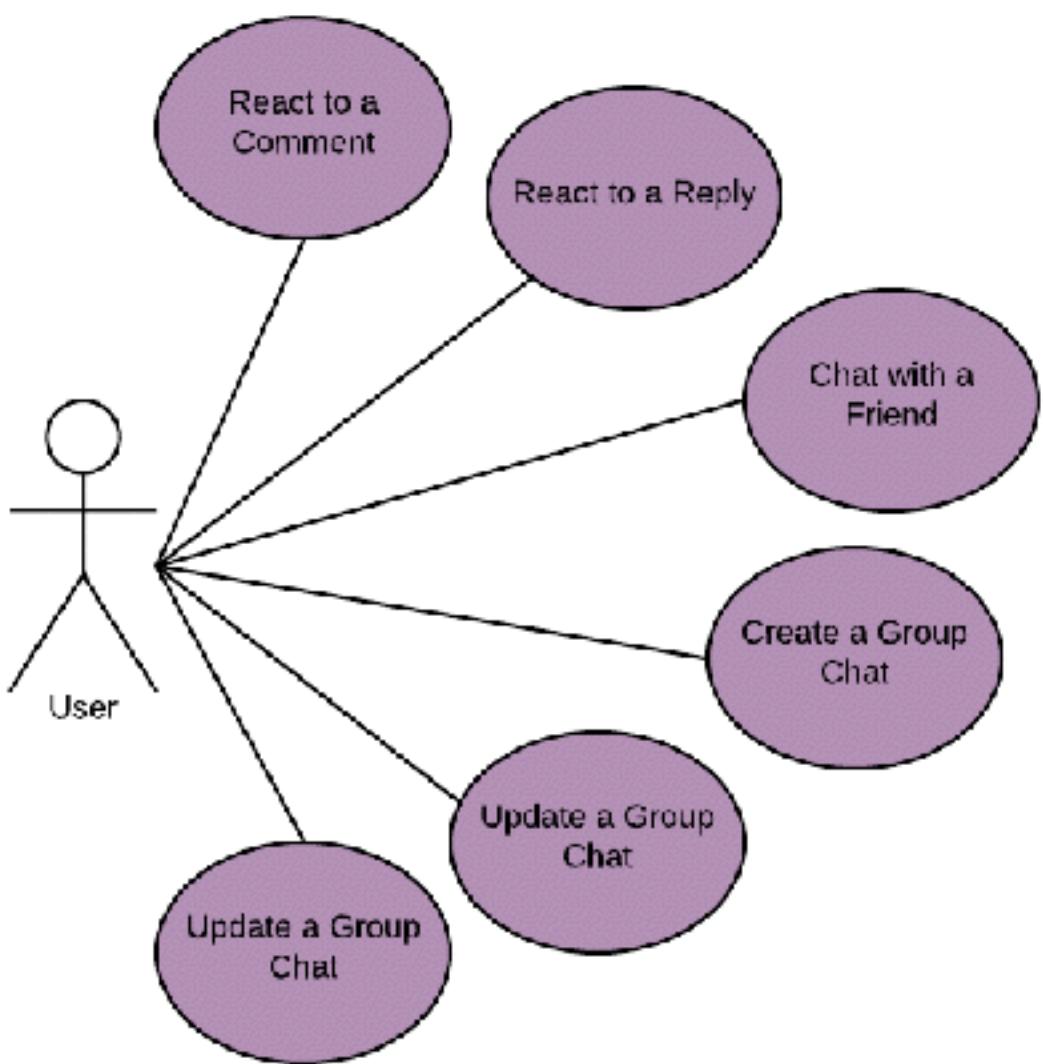
3.2.1. User

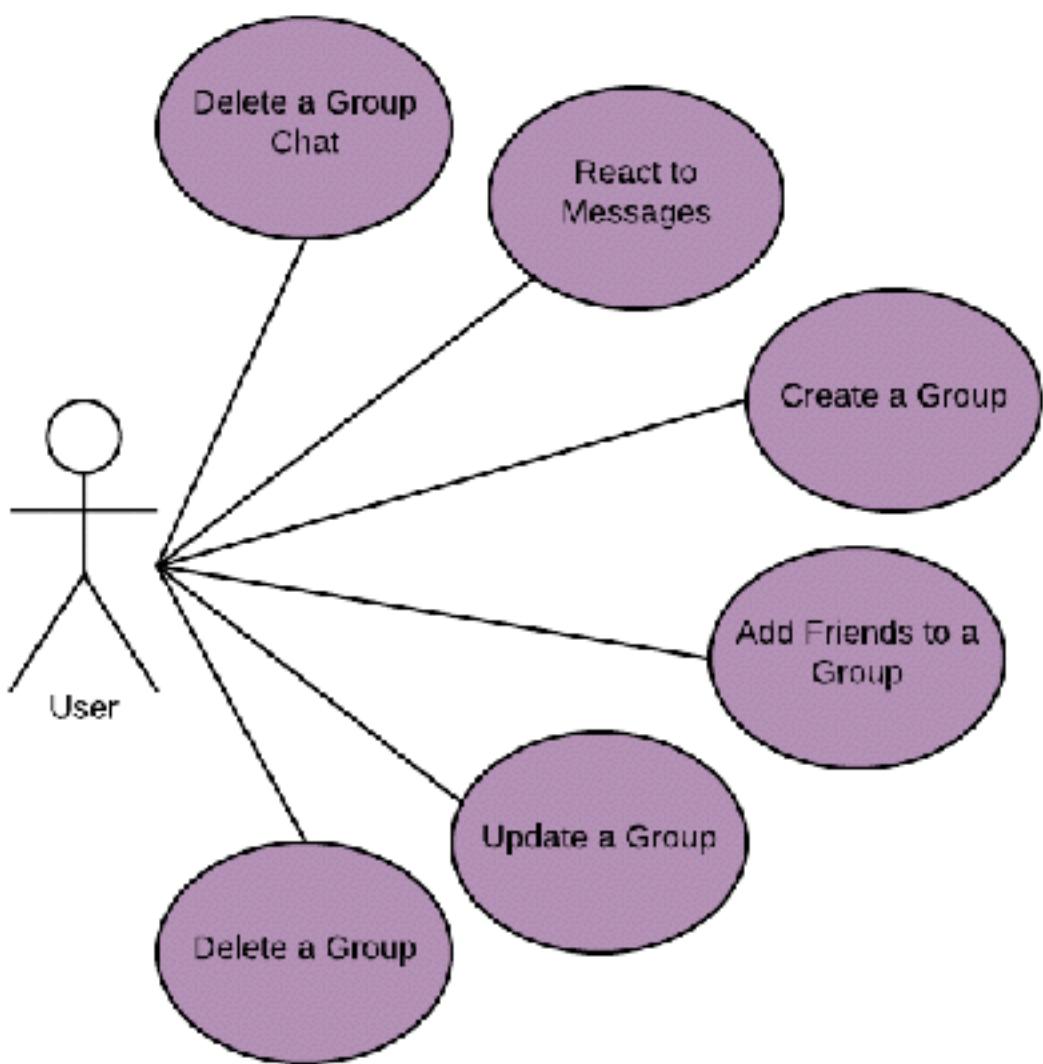


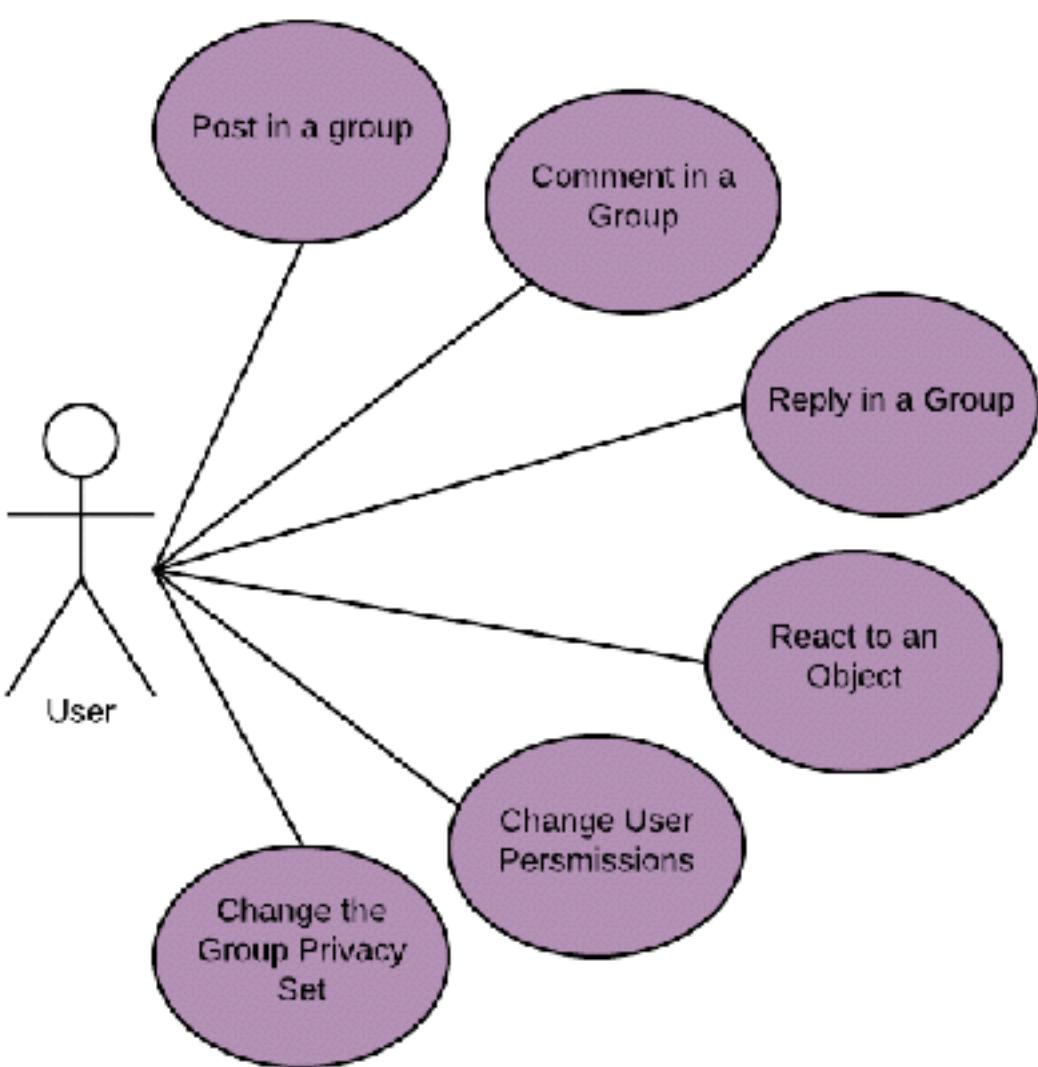


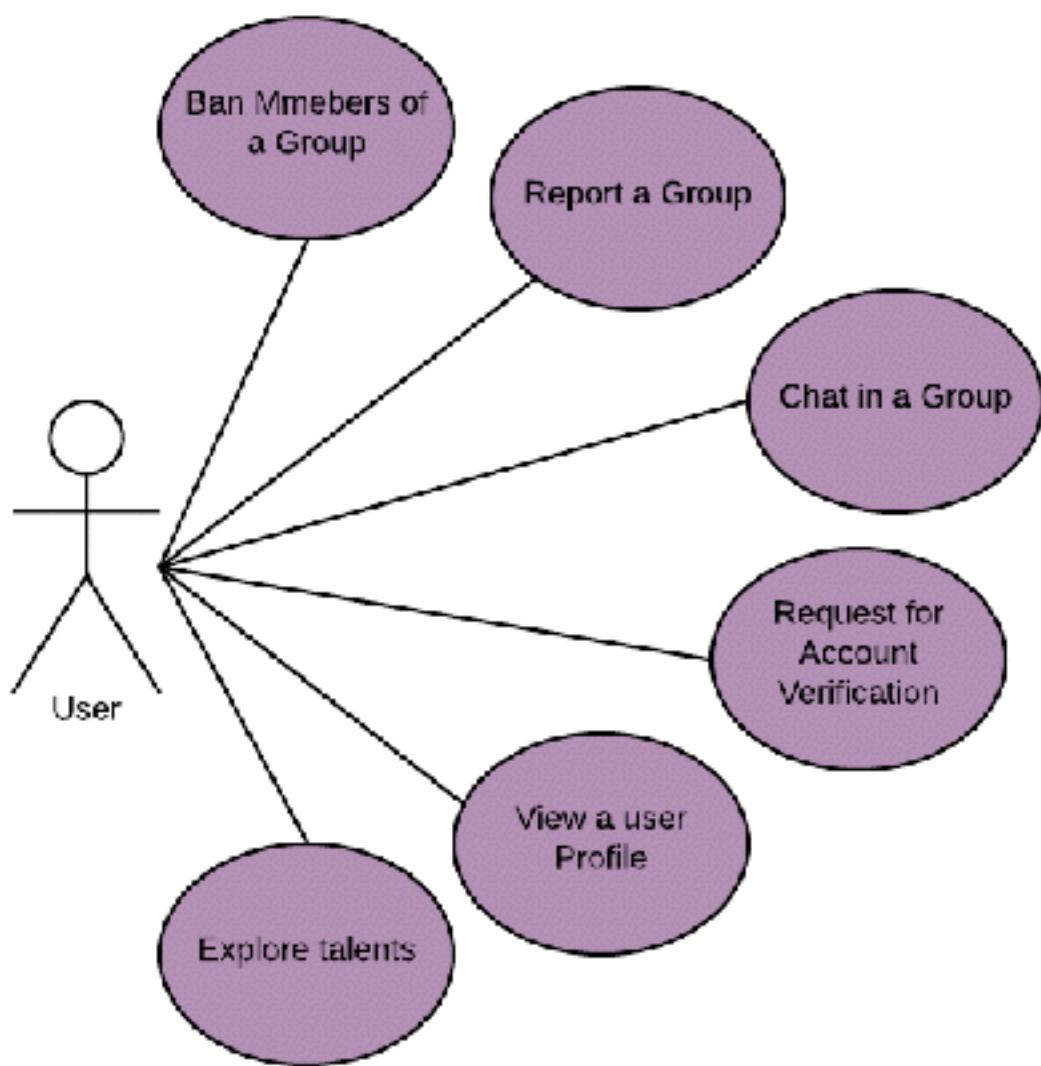


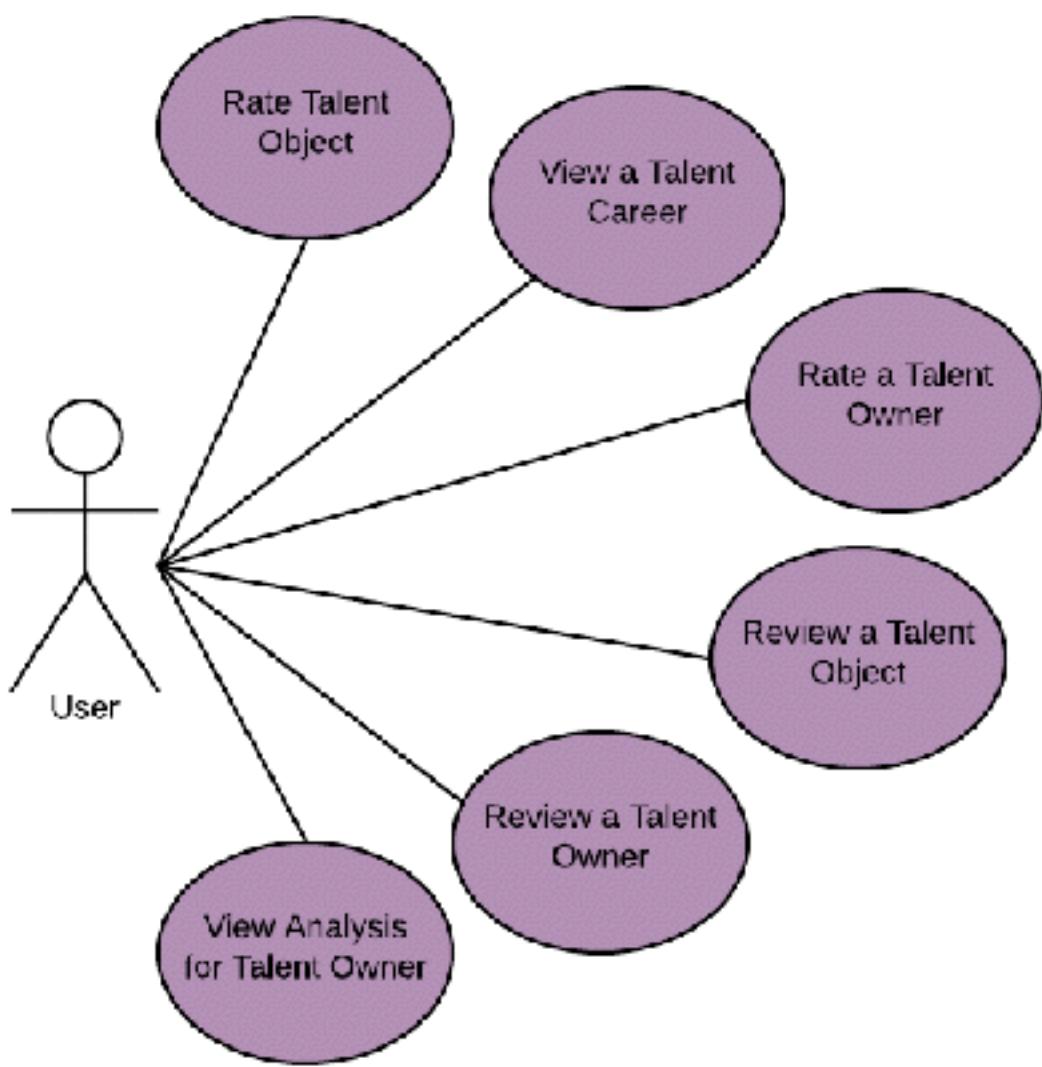


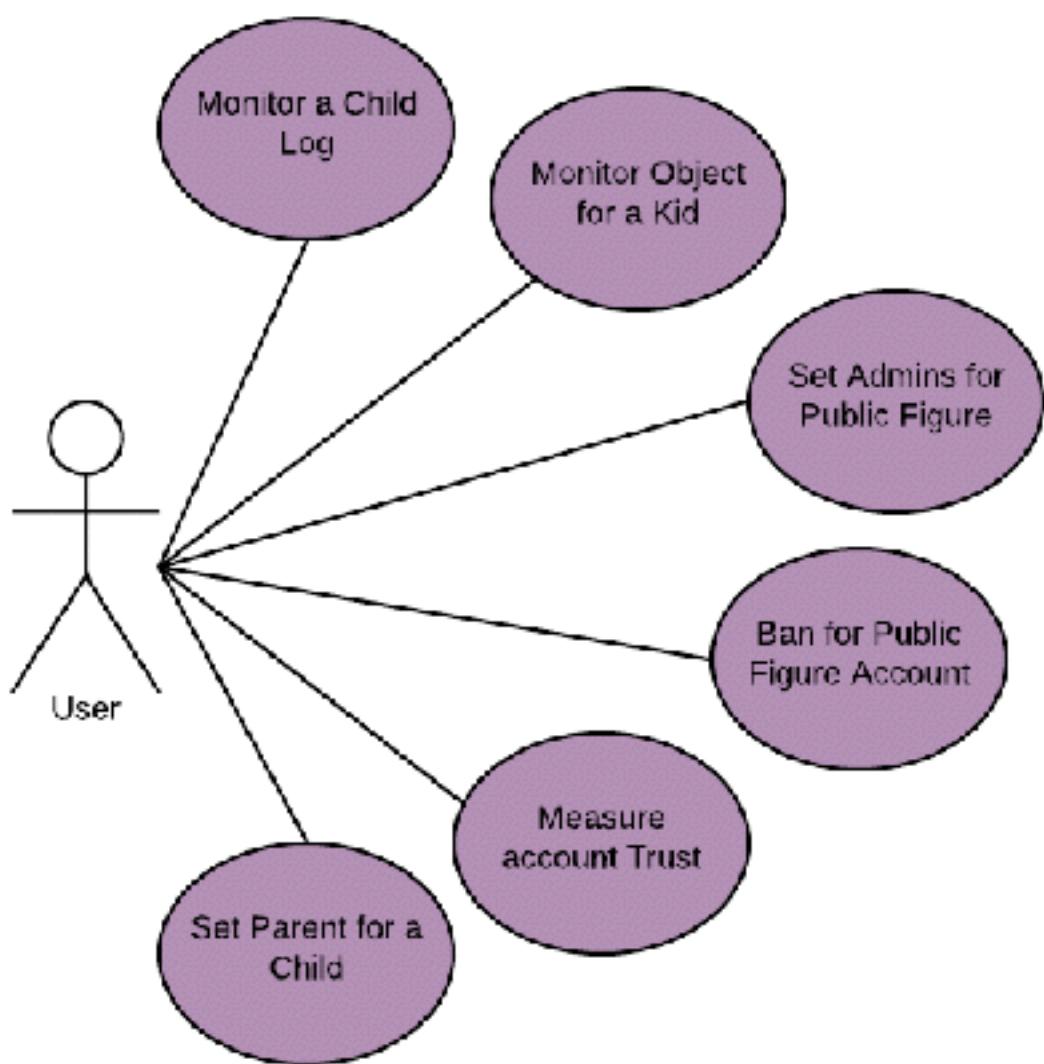












3.3. Use case Specification Table

3.3.1. Publish a knock —

Use case name	Publish a Knock	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will fill in the post area with text, upload a picture, album, video, record, check in, or multiple options among them.2. The user will then click the button "Publish".	<ol style="list-style-type: none">3. If the user has left the post empty, the system will display an error message "please fill in the post area", or else the system will publish the post and display success message " ".

3.3.2. Write a Comment

Use case name	Write a Comment	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• There should be a post to comment for.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">1. The user will fill in the comment area with text, upload a picture, album, video, record, or multiple options among them.2. The user will then click the button "".	<ol style="list-style-type: none">3. If the user has left the comment empty, the system will display an error message "please fill in the comment area", or else the system will publish the comment and display success message " ".

3.3.3. Write a Reply

Use case name	Write a Reply	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in• There should be a comment to reply for	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will fill in the reply area with text, upload a picture, album, video, record, or multiple options among them.2. The user will then click the button "".	<ol style="list-style-type: none">3. If the user has left the reply empty, the system will display an error message "please fill in the reply area", or else the system will publish the reply and display success message "".

3.3.4. Add a New Circle

Use case name	Add a New Circle	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The user should not duplicate circle names. 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> The user will click on circle icon, then clicking on "adding new circle" button. The user will provide the circle name, thumbnail, and the participant friends there. The user will click on "add circle" button. 	<ol style="list-style-type: none"> If the user has left the circle name empty, or he did not choose the thumbnail for the circle, the system will display an error message "please fill in the required fields" or if the circle name is duplicated from existing circle that this user has, the system then will display an error message "this circle name is already existing as a name of a circle you have"/"choose another circle name", else the system will add the new circle and display success message "a circle has been added successfully"/"one circle added" to the user. The user then can optionally select specified friends to add them to his new added circle and the system will display message "a friend has been added successfully"/"a member is added" or a failure message "unable to add this member" / "already existing member".

3.3.5. Delete a Circle

Use case name	Add a New Circle	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The user should have the specified circle available to be deleted.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will select a circle and then in circle settings he will click the button "delete".	<ol style="list-style-type: none">4. If the user has not selected a circle, the system will display a failure message "please choose a circle to delete"/"no circle chosen yet" or else the system will delete all the circle members first and then he will delete the circle itself and will display a success message "circle is deleted"5. The user can later add a new circle with the same name of the deleted circle

3.3.6. Update a Circle Name

Use case name	Update a Circle Name	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The user should have the specified circle available to be Updated.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will select a circle and then in circle settings he can click the button “update” and then he will edit to the new name of the circle.2. The user will click the “update” button.3. The system will update the circle name and will display a success message “circle name is updated”.4. If the user has not selected a circle, the system will display a failure message “please choose a circle to Update” / “no circle chosen yet” or if the user did not fill the new name of the circle so that the system will display a failure message “please fill in the required fields”, or else the system will update the circle name and will display a success message “circle name is updated”.5. The user can later add members to the updated circle or delete this circle or update it again.	

3.3.7. Update a Circle Thumbnail

Use case name	Update a Circle Thmbnail	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The user should have the specified circle available to be Updated.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will select a circle and then in circle settings he can click the button "update" and then he will edit to the new thumbnail of the circle.	<ol style="list-style-type: none">4. If the user has not selected a circle, the system will display a failure message "please choose a circle to Update"/"no circle chosen yet" or if the user did not select the new Thumbnail of the circle so that the system will display a failure message "please fill in the required fields", or else the system will update the circle thumbnail and will display a success message "circle thumbnail is updated".5. The user can later add members to the updated circle or delete this circle or update it again.

3.3.8. Include a Friend to a Circle

Use case name	Include a Friend to a Circle	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The user should have the specified circle available to add friends to it. The account owner should have that friend added to one of his circles, or maybe the "All" circle prior to including him to a circle. 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> 1. The user will select a circle and then in circle settings he can click the button "Add Friends" and then he will select the desired friend, the clicks "add to circle" button. 	<ol style="list-style-type: none"> 4. If the user has not selected a friend, the system will display a failure message "please choose a friend to add"/"no friends chosen yet", or else the system will display success message "a Friend is added successfully"/" one member added to the circle". 5. The user can later add more members or exclude members from the circle.

3.3.9. Include a Group of Friends to a Circle

Use case name	Include a Group of Friends to a Circle	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The user should have the specified circle available to add friends to it. The account owner should have these friends added to one of his circles, or maybe the "All" circle prior to including them to a circle. 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> 1. The user will select a circle and then in circle settings he can click the button "Add Friends" and then he will select the desired friends, then clicks "add to circle" button. 	<ol style="list-style-type: none"> 4. If the user has not selected friends or at least one friend, the system will display a failure message "please choose friends to add"/"no friends chosen yet", or else the system will display success message "Friends are added successfully" / "members are added to the circle". 5. The user can later add more members or exclude members from the circle.

3.3.10.Exclude a Friend from a Circle

Use case name	Exclude a Friend from a Circle	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The user should have the specified circle available to add friends to it. The account owner should have that friend added to the selected circle. 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> 1. The user will select a circle and then in circle settings he can click the button “Exclude Friends” and then he will select the desired friend, the clicks “Exclude” button. 	<ol style="list-style-type: none"> 4. If the user has not selected a friend, the system will display a failure message “please choose a friend to add”/“no friends chosen yet”, or else the system will display success message “a Friend is excluded successfully”/“ one member excluded from the circle”. 5. The user can later add more members or exclude members from the circle.

3.3.11.Exclude a Group of Friends from a Circle

Use case name	Exclude a Group Friends from a Circle					
Initiating Actor	User					
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The user should have the specified circle available to add friends to it.• The account owner should have that friend added to the selected circle.					
Flow of Events	<table><thead><tr><th>User Steps</th><th>System Steps</th></tr></thead><tbody><tr><td><ol style="list-style-type: none">1. The user will select a circle and then in circle settings he can click the button “Exclude Friends” and then he will select the desired friend, the clicks “Exclude” button.4. If the user has not selected friends, the system will display a failure message “please choose friends to exclude”/“no friends chosen yet”, or else the system will display success message “Friends are excluded successfully”/“members are excluded from the circle”.5. The user can later add more members or exclude members from the circle.</td><td></td></tr></tbody></table>		User Steps	System Steps	<ol style="list-style-type: none">1. The user will select a circle and then in circle settings he can click the button “Exclude Friends” and then he will select the desired friend, the clicks “Exclude” button.4. If the user has not selected friends, the system will display a failure message “please choose friends to exclude”/“no friends chosen yet”, or else the system will display success message “Friends are excluded successfully”/“members are excluded from the circle”.5. The user can later add more members or exclude members from the circle.	
User Steps	System Steps					
<ol style="list-style-type: none">1. The user will select a circle and then in circle settings he can click the button “Exclude Friends” and then he will select the desired friend, the clicks “Exclude” button.4. If the user has not selected friends, the system will display a failure message “please choose friends to exclude”/“no friends chosen yet”, or else the system will display success message “Friends are excluded successfully”/“members are excluded from the circle”.5. The user can later add more members or exclude members from the circle.						

3.3.12. Regulate a User Privacy set

Use case name	Regulate a User Privacy Set	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The account owner should have that friend added to the any circle of his/her. 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> 1. The user will select the user and then in the user settings he will click "Change Privacy" button and he will choose one of the available privacy set from the dropdown list "Valid","Invalid",or "Invalid For All"and then he will click "Confirm" button. 	<ol style="list-style-type: none"> 4. If the user has not selected a privacy set and left this field empty, the system must display failure message "Please Choose a Privacy Option"/"No privacy option is selected yet", or else the system will display a success message "The Privacy set for this user is updated". 5. The user can later update this privacy set for this user again.

3.3.13. Regulate a Circle Privacy set

Use case name	Regulate a User Privacy Set	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The account owner should have that circle owned by him/her. 	
Flow of Events	User Steps	System Steps
	<p>1. The user will select the circle and then in the circle settings he will click "Change Privacy" button and he will choose one of the available privacy set from the dropdown list "Valid", "Invalid", or "Invalid For All" and then he will click "Confirm" button.</p>	<p>4. If the user has not selected a privacy set and left this field empty, the system must display failure message "Please Choose a Privacy Option"/"No privacy option is selected yet", or else the system will display a success message "The Privacy set for this circle is updated".</p> <p>5. The user can later update this privacy set for this circle again.</p>

3.3.14.Download a File

Use case name	Download a File	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The File has to be Available at the site.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will click "Download" button below the desired File.	<ol style="list-style-type: none">4. If the user has not the privacy set that gives him the privilege to download this file the system will display a failure message "This File is not Available" / "Unable to Download", or else the system will display a success message "downloading" / "the File is downloading".

3.3.15.Upload a Profile Picture

Use case name	Upload a Profile Picture	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">The site is available and the user must log in.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will be in his own profile page and he will click on the profile picture area and he will click on "Edit Profile Picture" then the user will choose the new profile picture from his own device, then he will be able to crop or rotate the chosen picture then he will click "ok" button.4. If the user has not chosen a profile picture, the system will display a failure message "Please choose a Profile Picture"/"No Profile Picture is Chosen yet", or else the system will display a success message "The Profile Picture has been uploaded Successfully".	

3.3.16.Upload a Cover Picture

Use case name	Upload a Cover Picture	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">The site is available and the user must log in.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will be in his own profile page and he will click on the cover picture area and he will click on "Edit Cover Picture" then the user will choose the new cover picture from his own device, then he will be able to crop or rotate the chosen picture then he will click "ok" button.4. If the user has not chosen a cover picture, the system will display a failure message "Please choose a Cover Picture"/"No Cover Picture is Chosen yet", or else the system will display a success message "The Cover Picture has been Uploaded Successfully".	

3.3.17.Update a Profile Picture

Use case name	Update a Profile Picture	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">The site is available and the user must log in.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will be in his own profile page and he will click on the profile picture area and he will click on "Edit Profile Picture" then the user will choose the new profile picture from his own device, then he will be able to crop or rotate the chosen picture then he will click "ok" button.4. If the user has not chosen a profile picture, the system will display a failure message "Please choose a Profile Picture"/"No Profile Picture is Chosen yet", or else the system will display a success message "The Profile Picture has been Updated Successfully".	

3.3.18.Update a Cover Picture

Use case name	Update a Cover Picture	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">The site is available and the user must log in.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will be in his own profile page and he will click on the cover picture area and he will click on "Edit Cover Picture" then the user will choose the new cover picture from his own device, then he will be able to crop or rotate the chosen picture then he will click "ok" button.4. If the user has not chosen a cover picture, the system will display a failure message "Please choose a Cover Picture"/"No Cover Picture is Chosen yet", or else the system will display a success message "The Cover Picture has been Updated Successfully".	

3.3.19.Unfriend a Friend

Use case name	Unfriend a Friend	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The friend has to be joined in one of the account owner circles.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will enter the friend's profile and then he will click "Unfriend" button.	<ol style="list-style-type: none">4. The system will display a success message "The friend has been unfriended successfully".5. The user can later send a friend request to that friend.

3.3.20. Block a Friend

Use case name	Block a Friend	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The friend has to be joined in one of the account owner circles.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will enter the friend's profile and then he will click "Block" button.	<ol style="list-style-type: none">4. The system will display a success message "The friend has been Blocked successfully".5. The user can later unblock this friend.

3.3.21.Update Personal Information

Use case name	Update Personal Information	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• There must be a previously initialized personal information.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will enter his own profile and select the desired section of personal information, then he will click "Edit" button.	<ol style="list-style-type: none">4. If the user did not fill the new updated information, the system will display a failure message "No Information to be Updated" / "Please fill in the Required fields", or else the system will display a success message "Personal Information has been updated".5. The user can later update personal information again.

3.3.22.React to a Post

Use case name	React to a Post	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The post privacy set should allow the user to react to that post.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will be clicking on the emojis icons in the post to choose the desired emoji.	<ol style="list-style-type: none">4. The system should indicate success by placing the chosen emoji below the post.5. The user can later delete his emoji or update the chosen emoji.

3.3.23.React to a Comment

Use case name	React to a Comment	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The post privacy set should allow the user to react to that comment.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">1. The user will be clicking on the emojis icons in the comment to choose the desired emoji.	<ol style="list-style-type: none">4. The system should indicate success by placing the chosen emoji below the comment.5. The user can later delete his emoji or update the chosen emoji.

3.3.24.React to a Reply

Use case name	React to a Reply	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The post privacy set should allow the user to react to that reply.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">1. The user will be clicking on the emojis icons in the reply to choose the desired emoji.	<ol style="list-style-type: none">4. The system should indicate success by placing the chosen emoji below the reply.5. The user can later delete his emoji or update the chosen emoji.

3.3.25. Report a User

Use case name	Report a User	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The account owner should not have reported this user before.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will be clicking on "Report" button on the other user's profile and then he can optionally provide the reason of reporting, and then the user clicks "Confirm Report" button.	<ol style="list-style-type: none">4. The system should display a success message "User has been Reported"/"Report Confirmed".5. The user can later remove the report for that user.

3.3.26.Chat with a Friend

Use case name	Chat with a friend					
Initiating Actor	User					
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The receiver user should not be blocking the sender user. The user should be one of the chat starter friends. 					
Flow of Events	<table border="1"> <thead> <tr> <th>User Steps</th><th>System Steps</th></tr> </thead> <tbody> <tr> <td> <ol style="list-style-type: none"> The user will enter the chat section and he will click on the previous chat between him/her and the desired user or he can initiate the chat by searching for the friend and then he can start the chat by clicking on "Chat" button. </td><td> <ol style="list-style-type: none"> If the user has provided a non friend to search for in order to initiate the chat, the system will display a null result or it can display "Not a friend of yours" or if the receiver friend has blocked the sender, then the system will display a failure message "You cannot send a message to this User anymore" / "Unable to Chat", or else the system will send the message to the receiver successfully and it can be displayed for the other user too . The user can later delete the chat or initiate another chat. </td></tr> </tbody> </table>		User Steps	System Steps	<ol style="list-style-type: none"> The user will enter the chat section and he will click on the previous chat between him/her and the desired user or he can initiate the chat by searching for the friend and then he can start the chat by clicking on "Chat" button. 	<ol style="list-style-type: none"> If the user has provided a non friend to search for in order to initiate the chat, the system will display a null result or it can display "Not a friend of yours" or if the receiver friend has blocked the sender, then the system will display a failure message "You cannot send a message to this User anymore" / "Unable to Chat", or else the system will send the message to the receiver successfully and it can be displayed for the other user too . The user can later delete the chat or initiate another chat.
User Steps	System Steps					
<ol style="list-style-type: none"> The user will enter the chat section and he will click on the previous chat between him/her and the desired user or he can initiate the chat by searching for the friend and then he can start the chat by clicking on "Chat" button. 	<ol style="list-style-type: none"> If the user has provided a non friend to search for in order to initiate the chat, the system will display a null result or it can display "Not a friend of yours" or if the receiver friend has blocked the sender, then the system will display a failure message "You cannot send a message to this User anymore" / "Unable to Chat", or else the system will send the message to the receiver successfully and it can be displayed for the other user too . The user can later delete the chat or initiate another chat. 					

3.3.27.Create a Group Chat

Use case name	Create a Group Chat	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The receiver users in the group chat should not be blocking the sender user.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">1. The user will enter the group chat if the group chat has been initiated previously, or else the user can initiate the group chat by clicking "Chat" button and then the user will select the specified user to be included.2. If the user has been supplying the friends selection with a null group that there is no friend has been selected, the system will display a failure message "No Friends Selected"/"Please Select Friends", or else the system will display success message "The group chat has been successfully created".	

3.3.28.Update a Group Chat

Use case name	Update a Group Chat	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The user have to be a member of the group chat.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will enter the group chat that has been previously created and then he will click "Edit" button then he will exclude the selected members from the group chat then clicks on "Confirm" button.2. If the user has not removed friends from the group chat, or updated the group chat, the system will display failure message "No Update Detected"/"Please Update the Group chat", or else the system will display a success message "The Group Chat has been updated Successfully" / " Updated Successfully".3. The user can later update the group chat or he can delete it if needed.	

3.3.29.Delete a Group Chat

Use case name	Delete a Group Chat	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The user have to be the group chat initiator. The group chat has to be available to be deleted. 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> 1. The user will enter the group chat that has been previously created and then he will click “Delete Group Chat” button. 2. then a confirmation message will be displayed for the user, then the user will click “Confirm” button. 	<ol style="list-style-type: none"> 3. If the user has not confirmed the Deletion Process the system will call off the deletion operation. 4. If the user confirmed the deletion operation, the system will display a success message “The Group Chat has been deleted Successfully”/“Group Chat id Deleted”. 5. The user can later add a new group chat with the specified friends.

3.3.30.React to a Message

Use case name	React to a Message	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The user have to be one of the group chat members. The group chat message has to be available. 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> 1. The user will enter the group chat and will choose an emoji for the selected emoji either "Like", "Angry", "Finger", "Dislike", "Poker", "Love" icon. 	<ol style="list-style-type: none"> 2. If the emoji selected is not compatible with the object creator, the system will display a failure message "Reaction cannot be confirmed"/"Forbidden Emoji is selected", or else the system will display the selected emoji on the object selected. 3. If the user has chosen the emoji, the system will display the selected emoji on the message selected. 4. The user can later update the emoji with another one or just delete his reaction.

3.3.31.Create a Group

Use case name	Create a Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> The user will be in his profile page and then click "Create a Group" button. Then the user will supply the group information to initiate the group such as "Name", "Category", "Thumbnail", "Preset", "Picture", "Group Preset". After that the user can optionally choose the members of his group to be added by selecting the friends after searching on them. Then the user will click "Confirm" button. 	<ol style="list-style-type: none"> If the user has not chosen friends or he searched for non friends users, the system will display a failure message "No Friends Selected"/"Please Select Friends", or else the system will display a success message "The Group has been Created"/"A Group is Created". The user can later update the group initials, Remove members, or delete this group.

3.3.32.Add Friends to a Group

Use case name	Add Friends to a Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The user must have the privilege to add members to the group. 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> The user will enter the Group that has been created previously and then he will click "Add Members" button. The user will search for his friends to select the friends to be added. Then he will click "Add" button. 	<ol style="list-style-type: none"> If the user has not chosen friends or he searched for non friends users, the system will display a failure message "No Friends Selected"/"Please Select Friends", or else the system will display a success message "Members have been added"/"Members are added successfully". The user can later update the group initials, Remove members, or delete this group.

3.3.33.Update Group

Use case name	Update Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The user must have the privilege to Update the group. 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> The user will enter the Group that has been created previously and then he will click "Update" button. The user will click on the fields to be updated then he will supply the new information. Then he will click "Update" button. 	<ol style="list-style-type: none"> If the user has not updated the selected fields or supplied empty input, the system will display a failure message "No Update Detected" / "Please Update the Group chat", or else the system will display a success message "Group is updated Successfully", "Group is Updated now". The user can later update the group again initials, Remove members, or delete this group.

3.3.34.Delete Group

Use case name	Delete a Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The user must have the privilege to Delete the group.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">1. The user will enter the Group that has been created previously and then he will click "Delete Group" button.2. Then a confirmation message will be displayed for the user, then the user will click "Confirm" button.	<ol style="list-style-type: none">3. If the user has not confirmed the Deletion Process the system will call off the deletion operation.4. If the user confirmed the deletion operation, the system will display a success message "The Group has been deleted Successfully" / "Group is Deleted".5. The user can later add a new group with the specified friends.

3.3.35.Post in a Group

Use case name	Post in a Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The user have to be a member of the group.• The user must have the privilege to post in the group.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will enter the Group that has been created previously then he will fill in the post area with text, picture, album, video, record, or checkin, then he will click “Publish” button.2. If the user has supplied an empty post, the system will display a failure message “Please fill the Post area” / “Empty Post is provided”, or else the system will display a success message “The Post is published Successfully”.3. The user can later delete the post or edit it.	

3.3.36.Comment in a Group

Use case name	Comment in a Group					
Initiating Actor	User					
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The user have to be a member of the group.• The user must have the privilege to comment in the group.					
Flow of Events	<table><thead><tr><th>User Steps</th><th>System Steps</th></tr></thead><tbody><tr><td><ol style="list-style-type: none">1. The user will enter the Group that has been created previously then for specific post he will fill in the comment area with text, picture, album, video, record, or checkin, then he will click "Comment" button.2. If the user has supplied an empty comment, the system will display a failure message "Please fill the Comment area"/"Empty Comment is provided", or else the system will display a success message "The Comment is published Successfully".3. The user can later delete the comment or edit it.</td><td></td></tr></tbody></table>		User Steps	System Steps	<ol style="list-style-type: none">1. The user will enter the Group that has been created previously then for specific post he will fill in the comment area with text, picture, album, video, record, or checkin, then he will click "Comment" button.2. If the user has supplied an empty comment, the system will display a failure message "Please fill the Comment area"/"Empty Comment is provided", or else the system will display a success message "The Comment is published Successfully".3. The user can later delete the comment or edit it.	
User Steps	System Steps					
<ol style="list-style-type: none">1. The user will enter the Group that has been created previously then for specific post he will fill in the comment area with text, picture, album, video, record, or checkin, then he will click "Comment" button.2. If the user has supplied an empty comment, the system will display a failure message "Please fill the Comment area"/"Empty Comment is provided", or else the system will display a success message "The Comment is published Successfully".3. The user can later delete the comment or edit it.						

3.3.37.Reply in a Group

Use case name	Reply in a Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The user have to be a member of the group.• The user must have the privilege to reply in the group.	
	User Steps	System Steps
Flow of Events		<ol style="list-style-type: none">1. The user will enter the Group that has been created previously then for specific post he will fill in the reply area with text, picture, album, video, record, or checkin, then he will click "Reply" button.2. If the user has supplied an empty reply, the system will display a failure message "Please fill the Reply area"/"Empty Reply is provided", or else the system will display a success message "The Reply is published Successfully".3. The user can later delete the reply or edit it.

3.3.38.React to an Object in a Group

Use case name	React to a Message	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The user have to be one of the group members. The user must have the privilege to reply in the group. 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> The user will enter the group then for a specific object he will choose an emoji for the selected emoji either "Like", "Angry", "Finger", "Dislike", "Poker", "Love" icon. 	<ol style="list-style-type: none"> If the emoji selected is not compatible with the object creator, the system will display a failure message "Reaction cannot be confirmed"/"Forbidden Emoji is selected", or else the system will display the selected emoji on the object selected. If the user has chosen the emoji, the system will display the selected emoji on the object selected. The user can later update the emoji with another one or just delete his reaction.

3.3.39.Update a Privacy Preset in a Group

Use case name	Update Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The user must have the privilege to Update the group privacy preset. 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> The user will enter the Group that has been created previously and then he will click "Update" button. The user will click on the privacy preset field to be updated then he will choose the new privacy preset. Then he will click "Update" button. 	<ol style="list-style-type: none"> If the user has not updated the privacy preset or supplied empty input, the system will display a failure message "No Update Detected"/"Please Update the Privacy Preset", or else the system will display a success message "Group Privacy preset is updated Successfully","Group Privacy Preset is Updated now". The user can later update the group again initials, Remove members, or delete this group.

3.3.40. Report a Group

Use case name	Report a User	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The account owner should not have reported this group before.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will be clicking on "Report" button on the group page and then he can optionally provide the reason of reporting, and then the user clicks "Confirm Report" button.	<ol style="list-style-type: none">4. The system should display a success message "Page has been Reported"/"Report Confirmed".5. The user can later remove the report for that page.

3.3.41. View User's Information

Use case name	View User's Information	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">The site is available and the user must log in.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">The user will be clicking on the other user name.	<ol style="list-style-type: none">If the requester does not have the privilege to view the other user's personal information, the system will display a failure message "Not allowed user information to be viewed"/"Cannot Show this User's Personal Information", or else the system will display the allowed user's personal information.

3.3.42.Explore Talents

Use case name	Explore Talents	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">The site is available and the user must log in.	
	User Steps	System Steps
Flow of Events		<ol style="list-style-type: none">The user will be clicking "Explore Talents" button.If there are no talents to show then the system will simply a failure message "No Talents Available"/"Cannot Load Talents", or else the system will display all the available talents.The user can later search and rate or write reviews for those talents.

3.3.43.Filter Talents

Use case name	Filter Talents	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">1. The user will be entering the talents section then he will provide his search key for filtering in the search bar and then he will click “Filter Talents” button.2. If there are no talents to show for the given search key then the system will simply a failure message “No Talents Available”/“Cannot Load Talents”, or else the system will display all the available filtered talents.3. The user can later rate or write reviews for those talents.	

3.3.44.Upload Talent Object

Use case name	Upload Talent Object	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">1. The user will be entering the talents section then he will click "Upload a Talent" button.2. The user will select a media such as: photo, album, video, record, or maybe a plain text, and then he will provide the talent name, category, and thumbnail.3. The user will click "Upload" button.	<ol style="list-style-type: none">4. If the user did not provide the required fields for uploading the talent such as: name, category, talent object, then the system will display a failure message "Please Fill in the Required Fields" / "Cannot Upload Talent,Please check all Fields", or else the system will display a success message "The Talent is U p l o a d e d " / " U p l o a d e d Successfully", and the talent will be added to the talents with the same category.

3.3.45.Explore Talents Hashtags

Use case name	Explore Talents	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">The site is available and the user must log in.	
	User Steps	System Steps
Flow of Events		<ol style="list-style-type: none">1. The user will be clicking "Explore Talents Hashtags" button.2. If there are no talents hashtags to show then the system will simply a failure message "No Talents hashtags Available"/"Cannot Load Talents", or else the system will display all the available talents hashtags.3. The user can later search and rate or write reviews for those talents.

3.3.46.Rate Talent Object

Use case name	Rate Talent Object	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The Talent Object must be available and the one who rates has the privilege to do so.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">1. The user will be selecting a specific talent and he will click "Rate" button then he will provide his rate and then he will click "Confirm" button.2. If the user did not supply the rate fields then the system will display a failure message "Please enter your Rate"/"No Rate is Provided", or else the system will display a success message "The rate has been added","Rate has been Confirmed Successfully".3. The user can later update his rate for the specific talent.	

3.3.47. View Talents Career

Use case name	View Talents Career	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">The site is available and the user must log in.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">The user will be entering the talents section then he will provide the talent owner and the talent category for filtering in the search bar and then he will click "View" button.If there are no talents to show for the given talent owner and talent category then the system will simply a failure message "No Talents Available"/"Cannot Load Talents", or else the system will display all the available talents.The user can later rate or write reviews for those talents.	

3.3.48.Rate Talent owner

Use case name	Rate Talent Owner	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The one who rates has the privilege to do so. 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> The user will be selecting a specific talent owner and he will click "Rate" button then he will provide his rate and then he will click "Confirm" button. 	<ol style="list-style-type: none"> If the user did not supply the rate fields then the system will display a failure message "Please enter your Rate"/"No Rate is Provided", or else the system will display a success message "The rate has been added"/"Rate has been Confirmed Successfully". The user can later update his rate for the specific talent owner.

3.3.49. Write a Review for a Talent Object

Use case name	Write a Review for a Talent Object					
Initiating Actor	User					
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The Talent Object must be available and the one who rates has the privilege to do so.					
Flow of Events	<table><thead><tr><th>User Steps</th><th>System Steps</th></tr></thead><tbody><tr><td><ol style="list-style-type: none">1. The user will be selecting a specific talent and he will click "Review" button then he will provide his review and then he will click "Confirm" button.2. If the user did not supply the review then the system will display a failure message "Please enter your Review" / "No Review is Provided", or else the system will display a success message "The Review has been added", "Review has been Confirmed Successfully".3. The user can later update his review for the specific talent object.</td><td></td></tr></tbody></table>		User Steps	System Steps	<ol style="list-style-type: none">1. The user will be selecting a specific talent and he will click "Review" button then he will provide his review and then he will click "Confirm" button.2. If the user did not supply the review then the system will display a failure message "Please enter your Review" / "No Review is Provided", or else the system will display a success message "The Review has been added", "Review has been Confirmed Successfully".3. The user can later update his review for the specific talent object.	
User Steps	System Steps					
<ol style="list-style-type: none">1. The user will be selecting a specific talent and he will click "Review" button then he will provide his review and then he will click "Confirm" button.2. If the user did not supply the review then the system will display a failure message "Please enter your Review" / "No Review is Provided", or else the system will display a success message "The Review has been added", "Review has been Confirmed Successfully".3. The user can later update his review for the specific talent object.						

3.3.50. Write a Review for a Talent Owner

Use case name	Write a Review for a Talent Owner	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The one who rates must have the privilege to do so.	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none">1. The user will be selecting a specific talent owner and he will click "Review" button then he will provide his review and then he will click "Confirm" button.2. If the user did not supply the review then the system will display a failure message "Please enter your Review"/"No Review is Provided", or else the system will display a success message "The Review has been added"/"Review has been Confirmed Successfully".3. The user can later update his review for the specific talent owner.	

3.3.51.Analysis for the Talent Owner

Use case name	Analysis for the Talent Owner	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">1. The user will be entering his own talent page and then he will click "Analyze" button.	<ol style="list-style-type: none">2. If the user has no talents published yet, then the system will display a failure message "No Talents given for Analysis"/"Cannot go through Analysis", or else the system will display the analysis as a success response.

3.3.52.Monitor Child Log

Use case name	Analysis for the Talent Owner	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> The site is available and the user must log in. The user who is monitoring has to be a verified parent for that kid 	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> The parent user will be entering his kids section and with selecting a specific kid if he has more than one kid, he will click "Monitor Log" button. If the user is no longer a parent for the kid, The system will display a failure message "You Cannot Monitor this Kid Log"/"Sorry, you do not have the Privilege To Monitor this User", or else the system will display a success action that the Log of the kid is successfully displayed for that parent. 	

3.3.53. Set a Parent for a Child

Use case name	Set a Parent for a Child	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">The site is available and the user must log in.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">The kid will be entering the parental monitor section and he will click "Add a Parent" and then he will select a user and clicks "Confirm" button.If the number of parents will exceed two after this request for that specified kid, the system will display a failure message "You cannot Add more Parents"/"Sorry, It Exceeds the Maximum number which is two", or else the system will display a success message "This User is Successfully added as a Parent" / "Added as a Parent".If the parent that is receiving the request did not confirm the request, the system will display a failure message to the kid "Sorry, This User did not Confirm the Request" / "Unable to add Parent", or else the system will display a success message "This User is Successfully added as a Parent" / "Added as a Parent".Later the parent can monitor the log of this kid and the kid may update parents account or deleted them.	

3.3.54.Ban for a Public Figure

Use case name	Ban for a Public Figure	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">• The site is available and the user must log in.• The user that is being banned must be still following this public figure account.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">1. The Public figure or one of his admins can select a specific user and clicks "Ban User" button.	<ol style="list-style-type: none">2. If the user being banned is no longer following the public figure account, the system will display a failure message "Sorry, cannot Ban this User"/"This User is not a Follower for this Account", or else the system will display a success message "The User is Banned"/"Banned Successfully".3. Later the public figure or one of his admins may remove the ban.

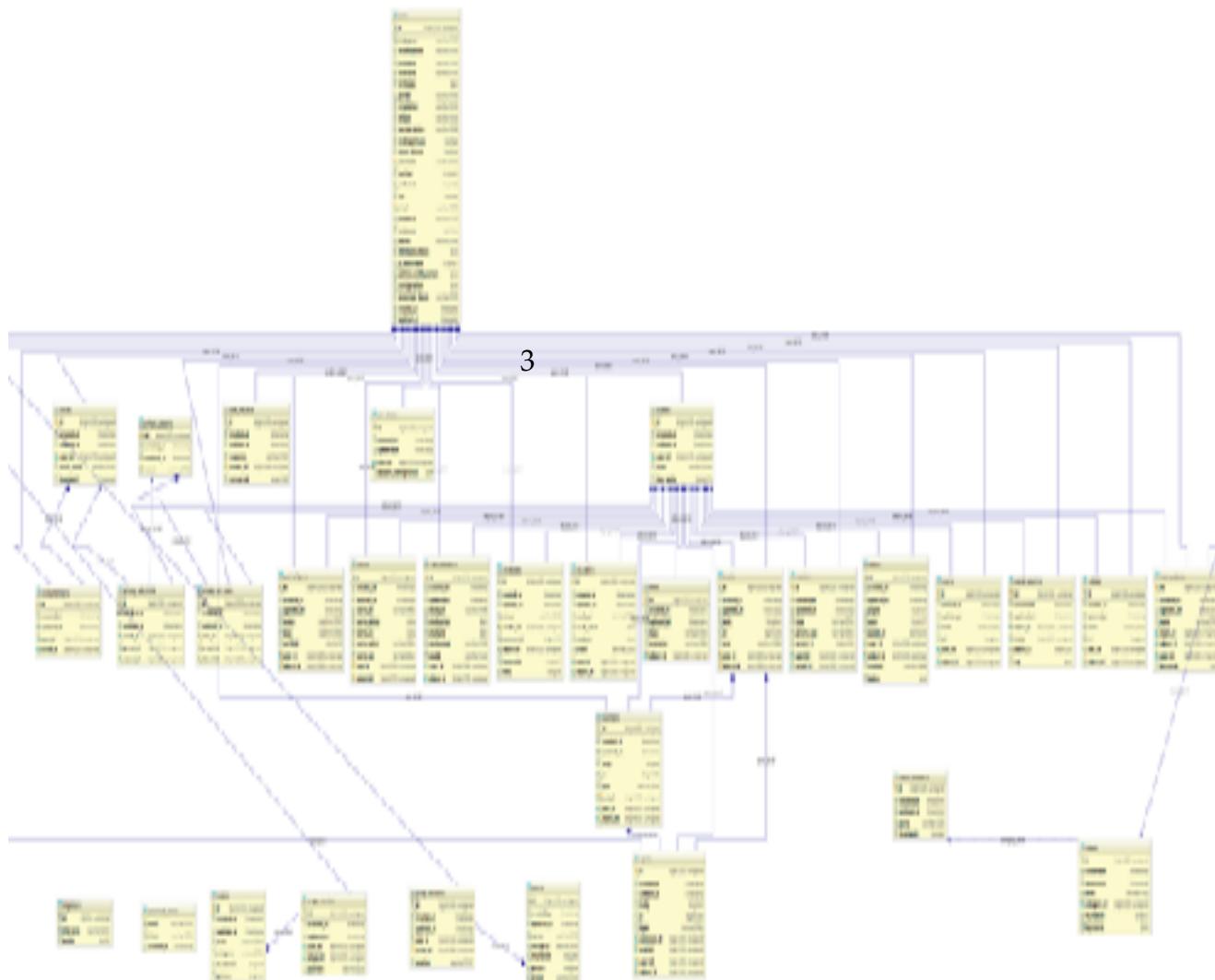
3.3.55. Set an Admin for a Public Figure

Use case name	Set an Admin for a Public Figure	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">The site is available and the user must log in.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">The Public figure will be entering his profile page and clicks "Add Admin" button and then he will be selecting user or many users and then he clicks "Confirm" button.If the user that is requested to be an admin did not confirm the request, the system will display failure message "Sorry, Cannot add this User as an Admin"/"The User did not Confirm the Request", or else the system will display a success message "An Admin has been Added"/"Added as an Admin Successfully".Later the public figure can remove this admin or can add more admins.	

3.3.56.Monitor an Object for a Kid

Use case name	Monitor an Object for a Kid	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none">The site is available and the user must log in.	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none">The kid will be receiving an object but before it displays it to the kid, the system will automatically monitor the object for a kid as a part from publishing an object or reacting.	<ol style="list-style-type: none">If the object is not appropriate for the child, the system will display a failure message to the object creator "Cannot Publish"/"Unable to proceed because of the kid inappropriate content", or else the system will display a success message "Published successfully".

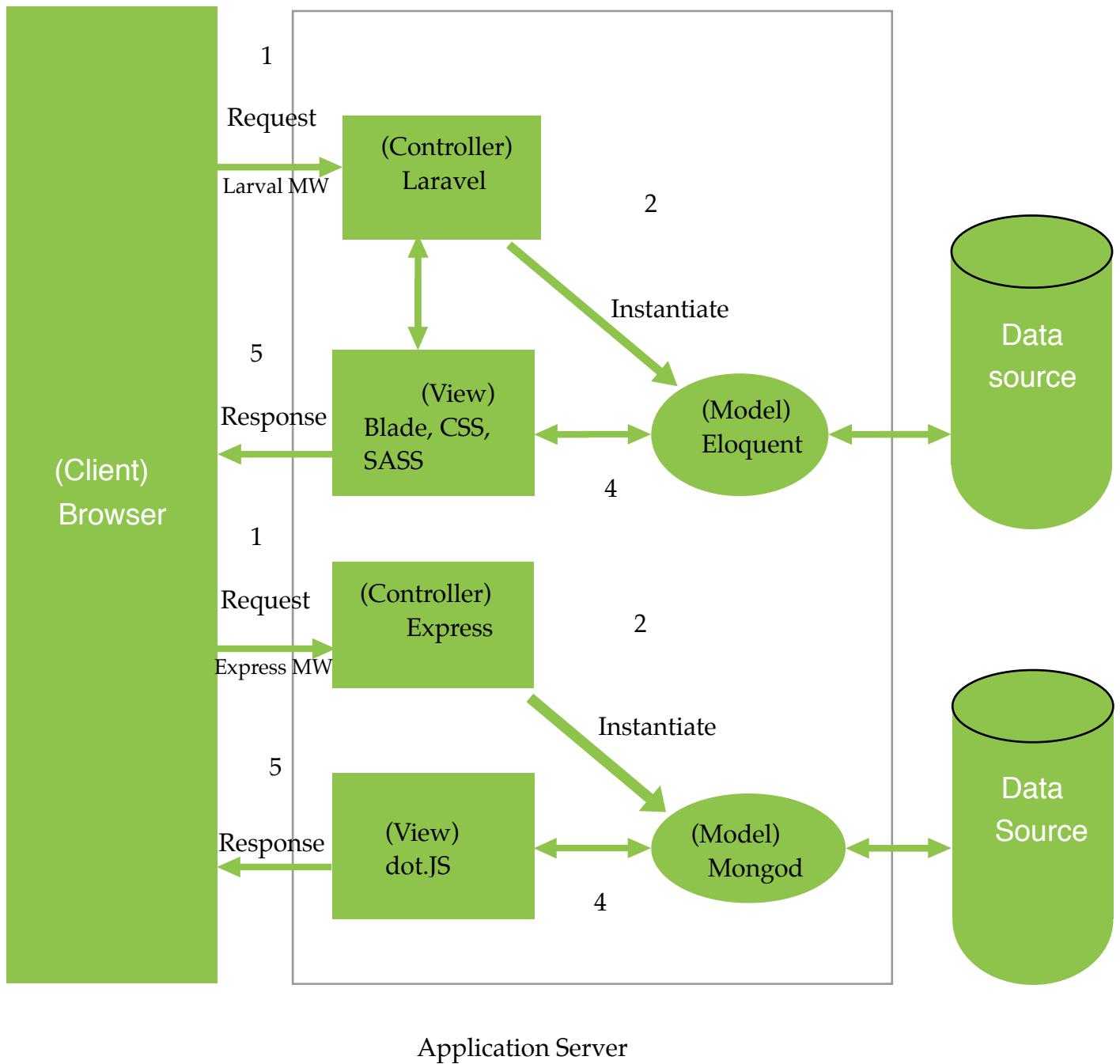
3.4. ERD



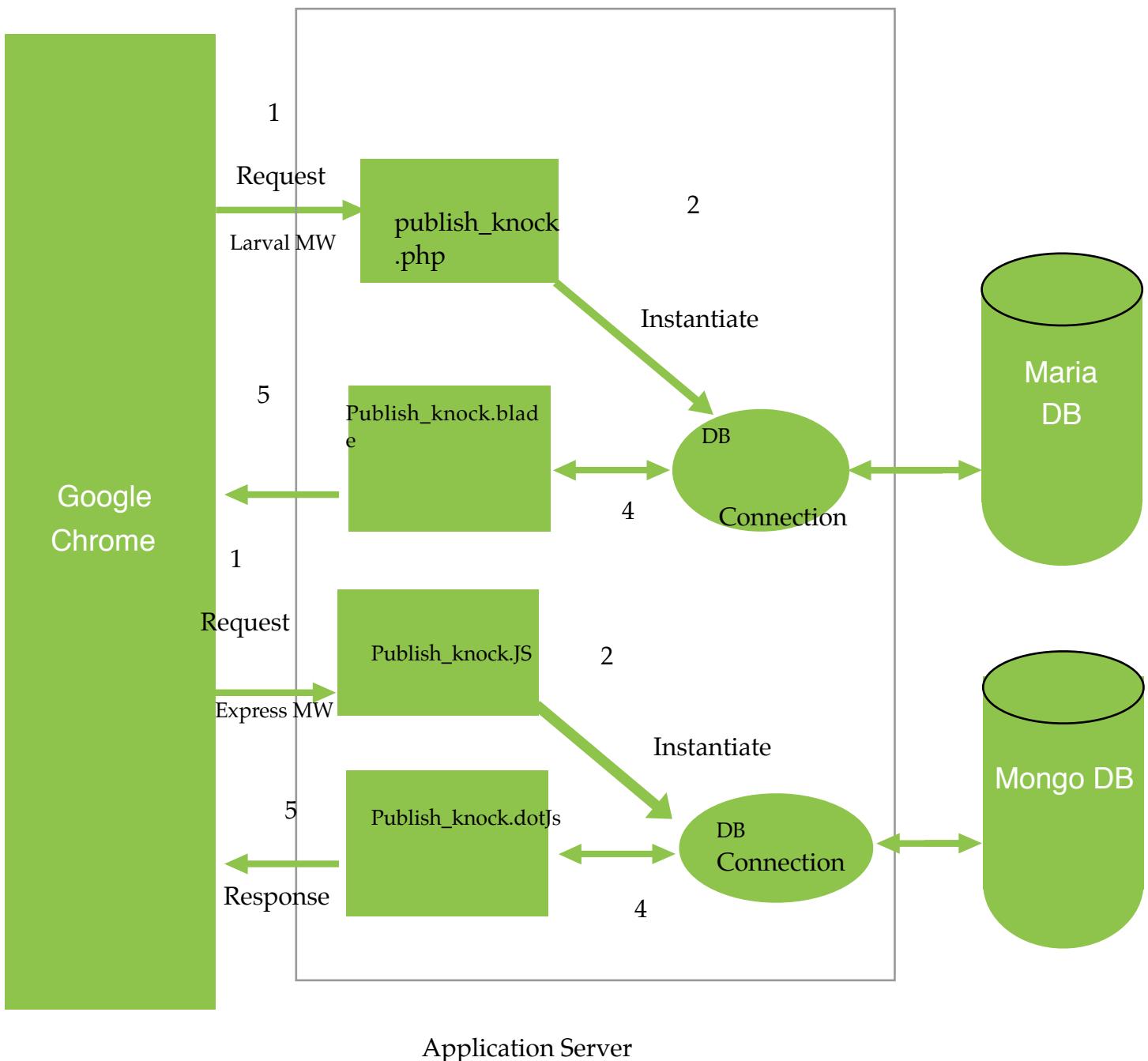
4. System Design

4.1. System Architecture

4.1.1. Application Architecture



4.1.2. Application Architecture (Example)

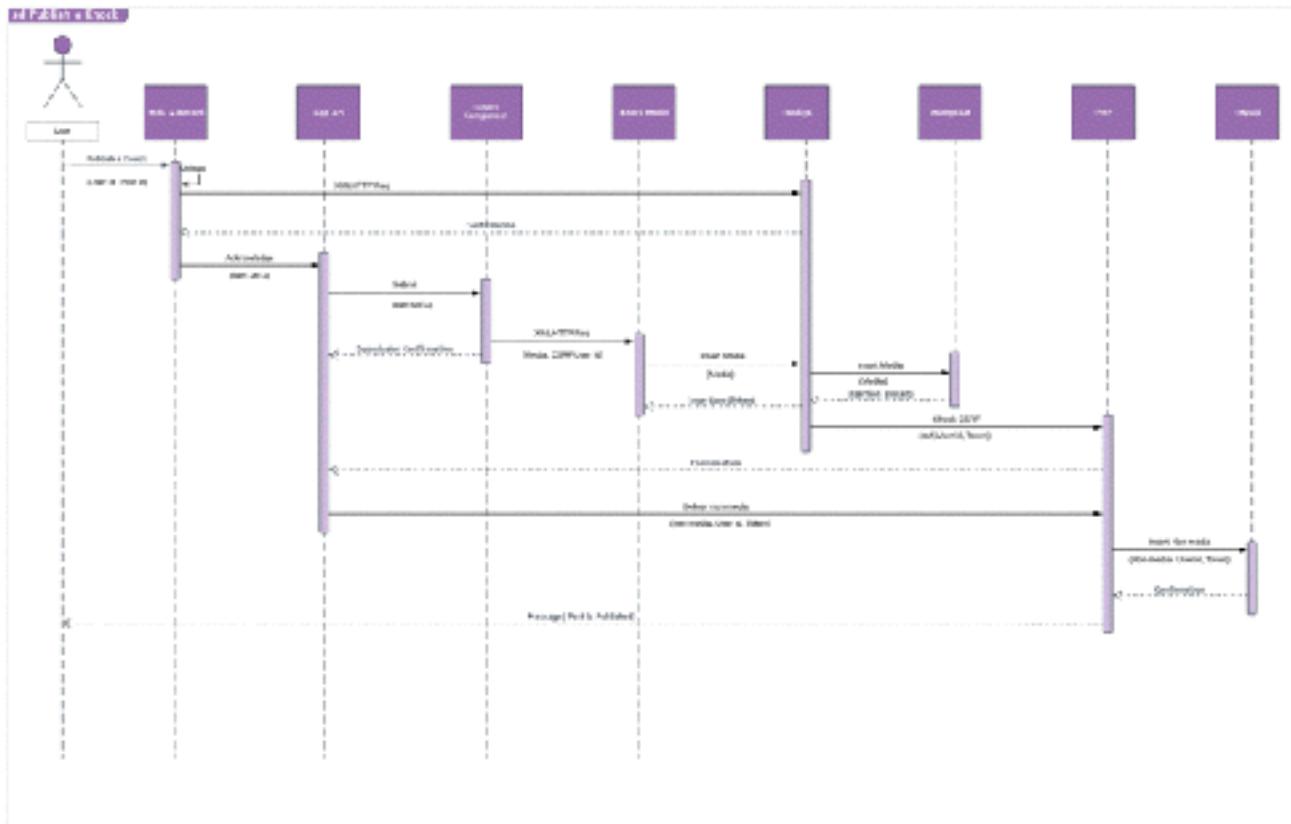


4.2. Class Diagram

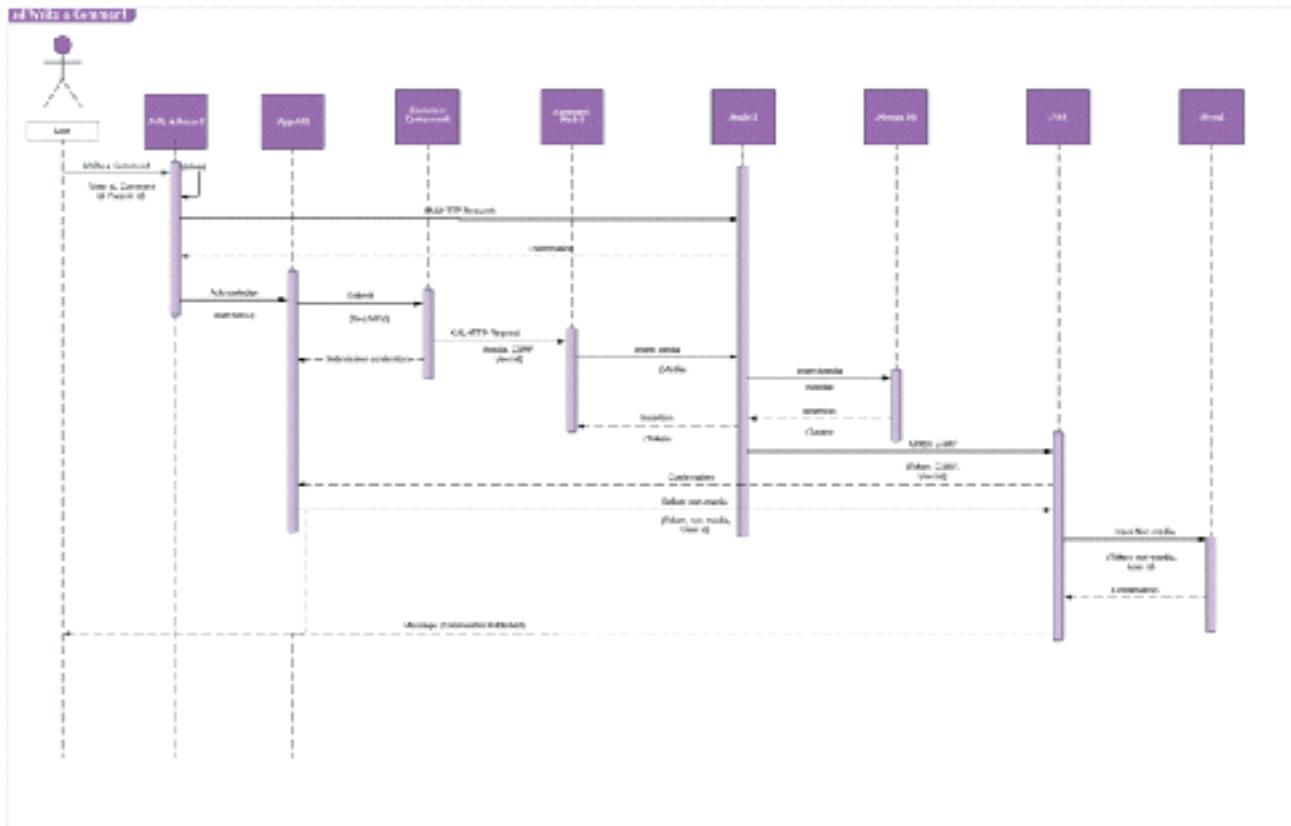
4.3. Physical Schema

4.4. Sequence Diagram

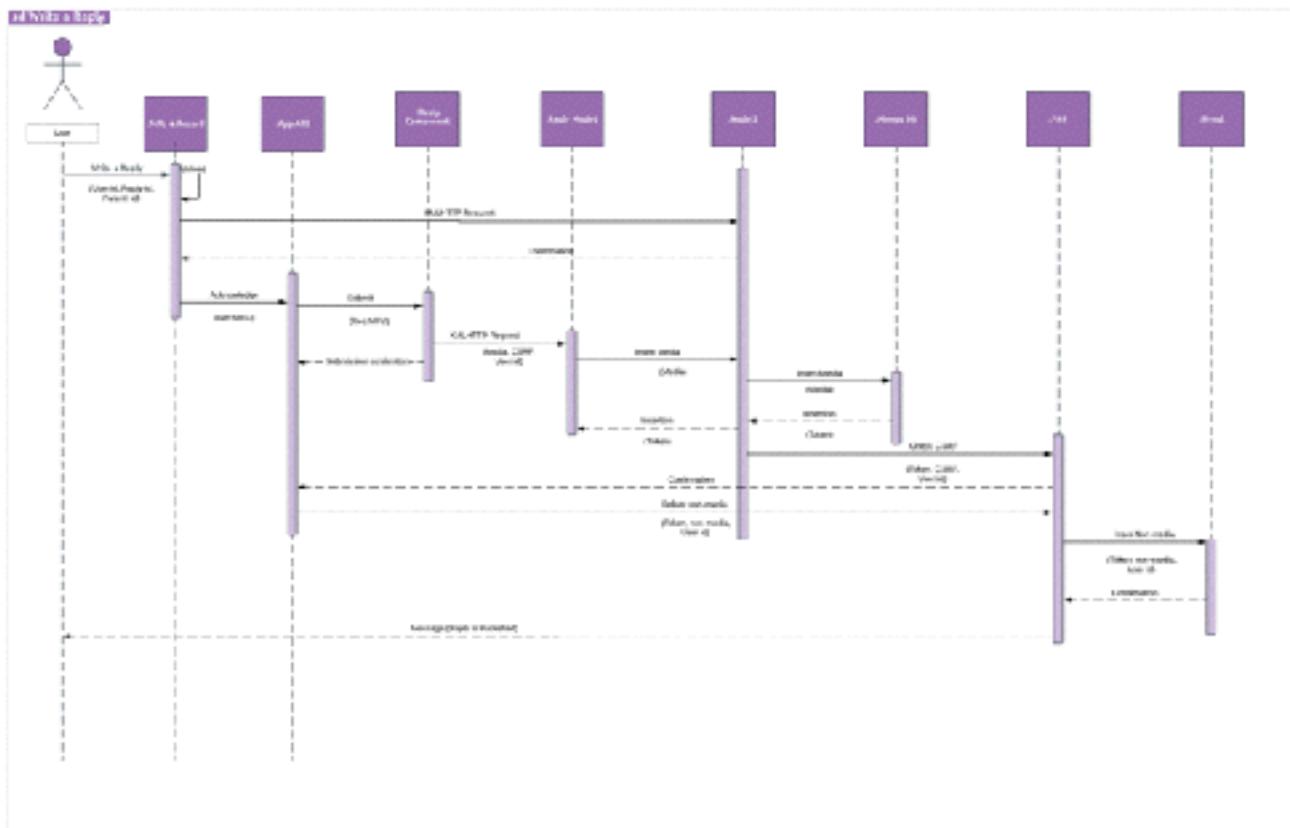
- Publish a knock (create post).



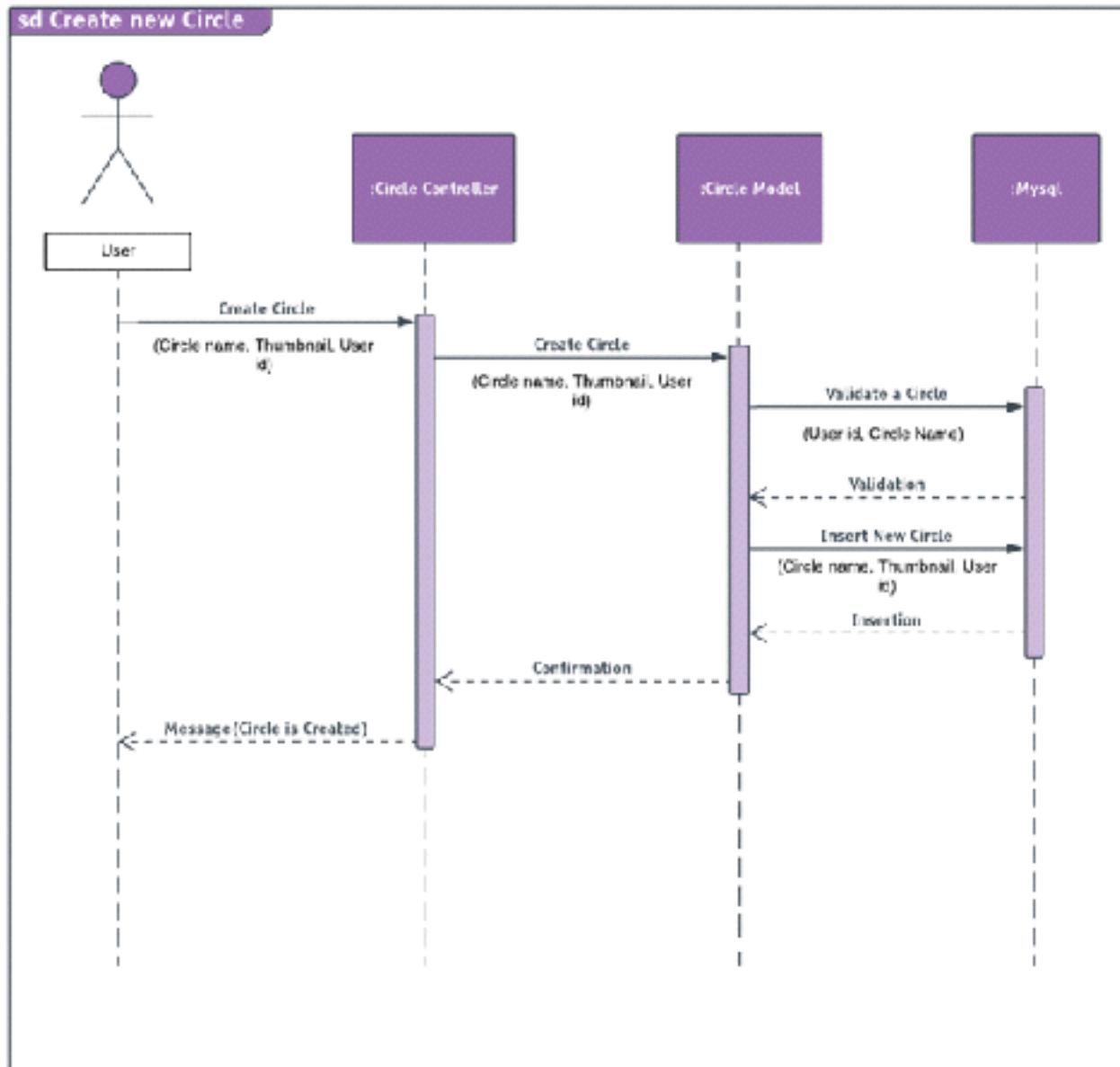
- Write a comment- when allowed to do so.



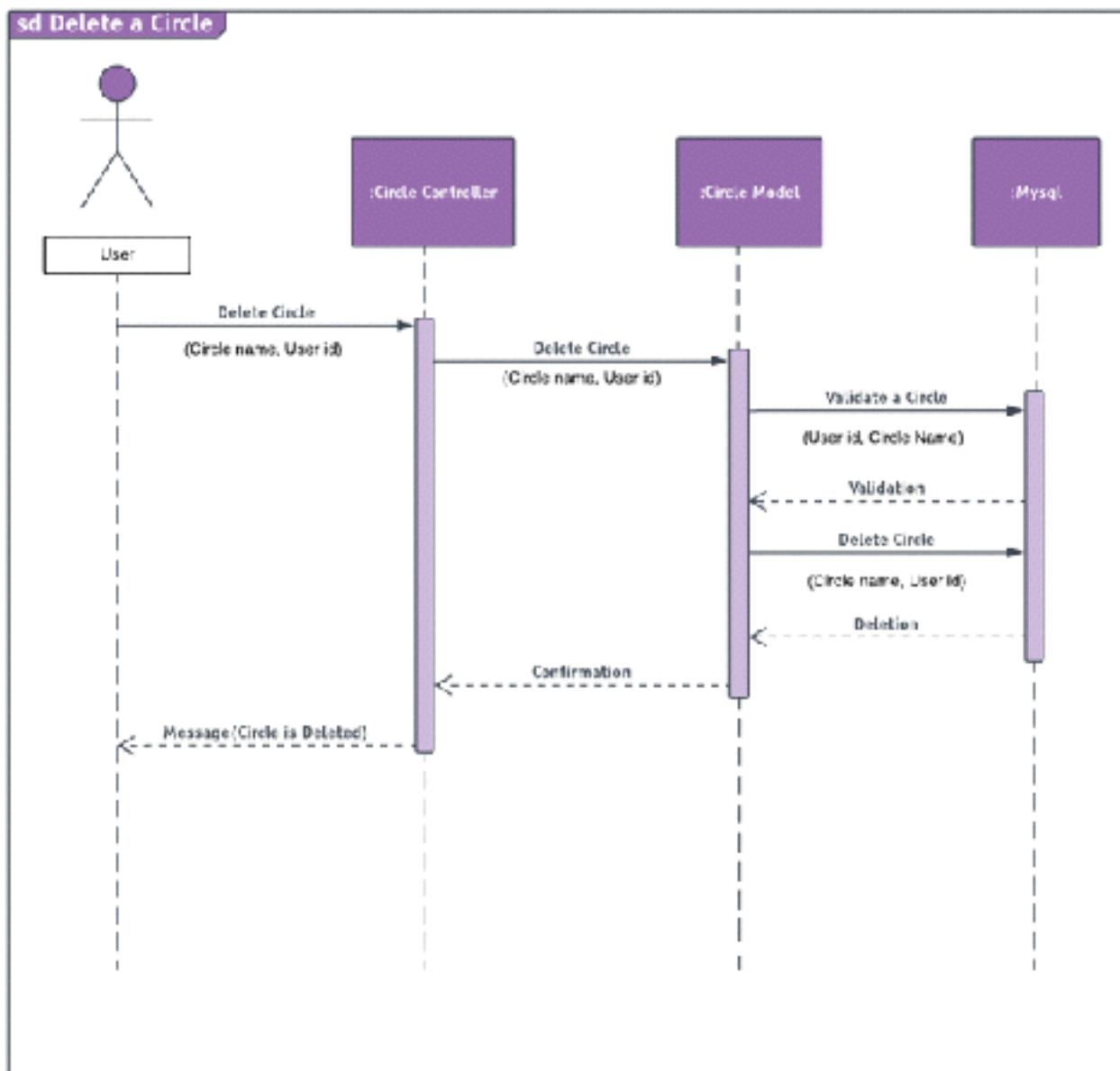
- Write a reply- when allowed to do so.



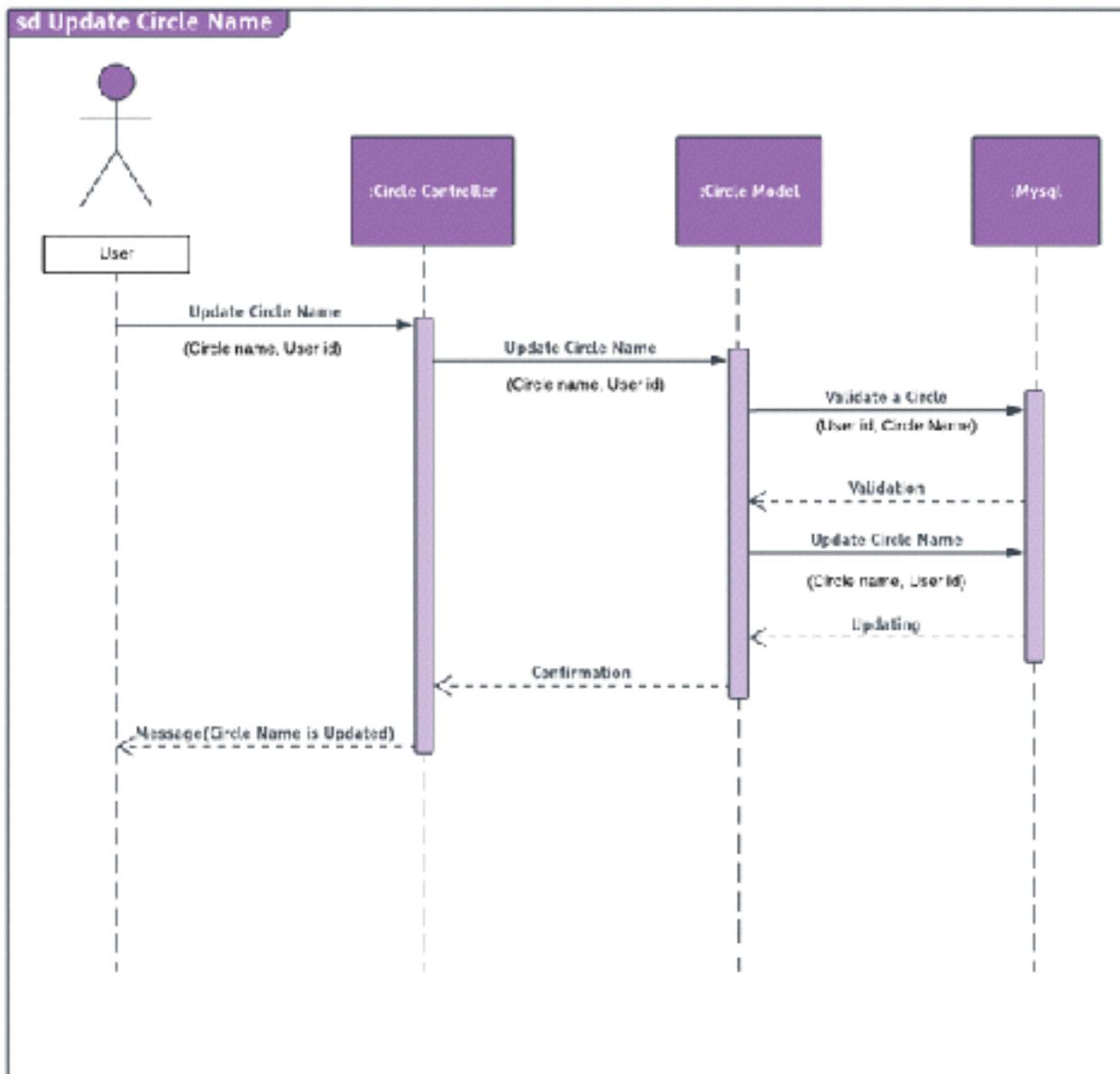
- Create a circle.



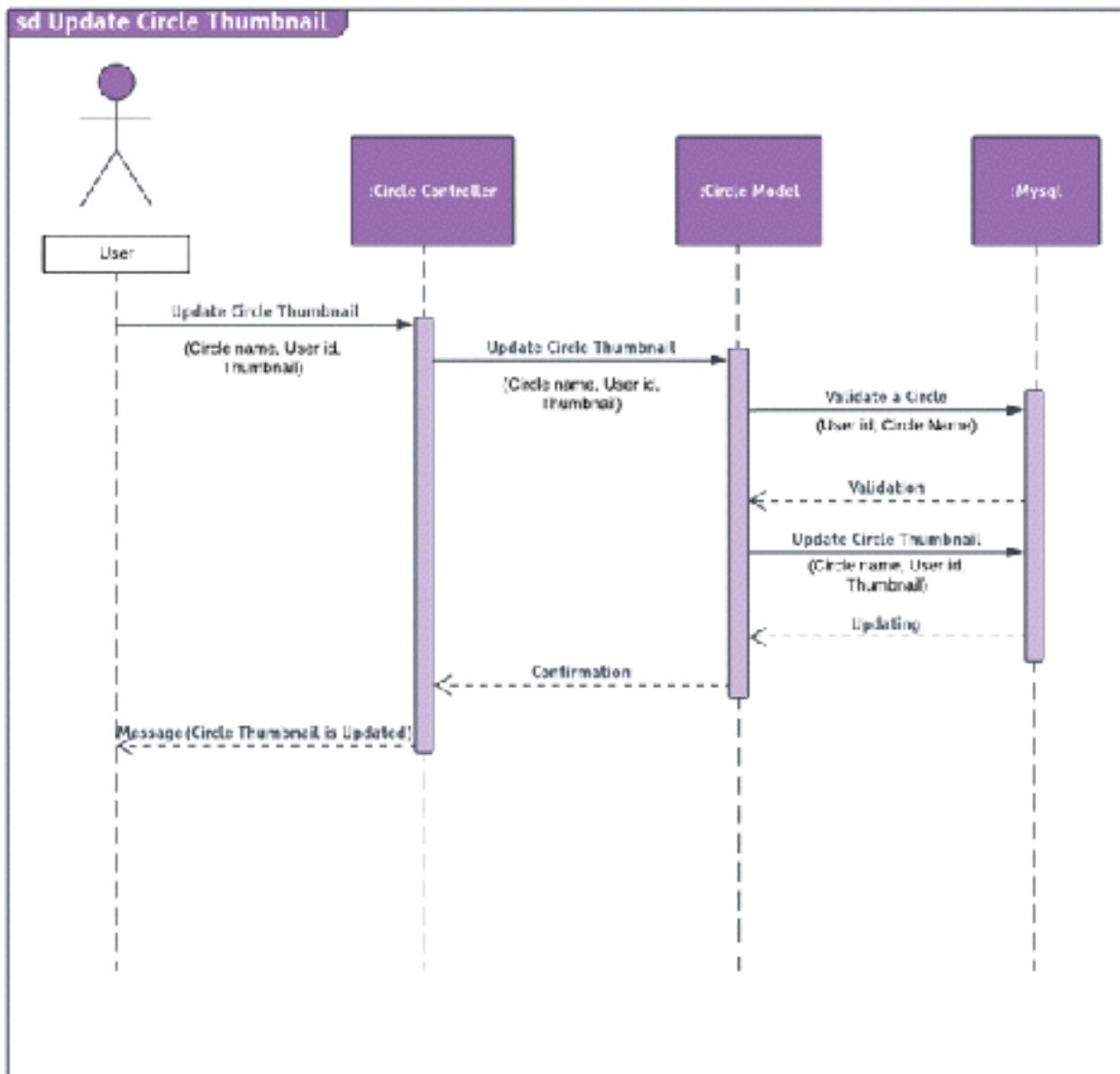
- Delete a circle



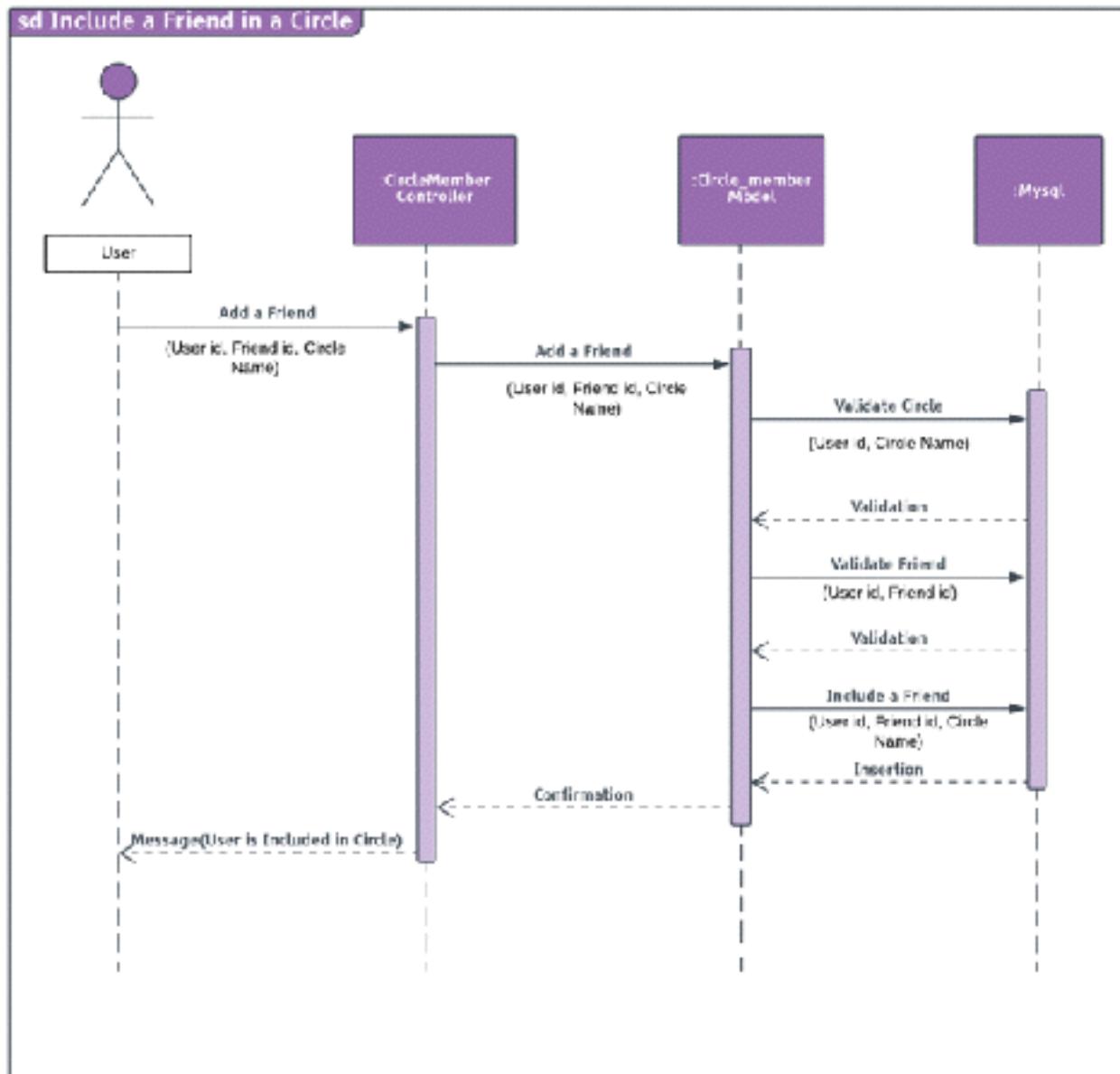
- Update circle name



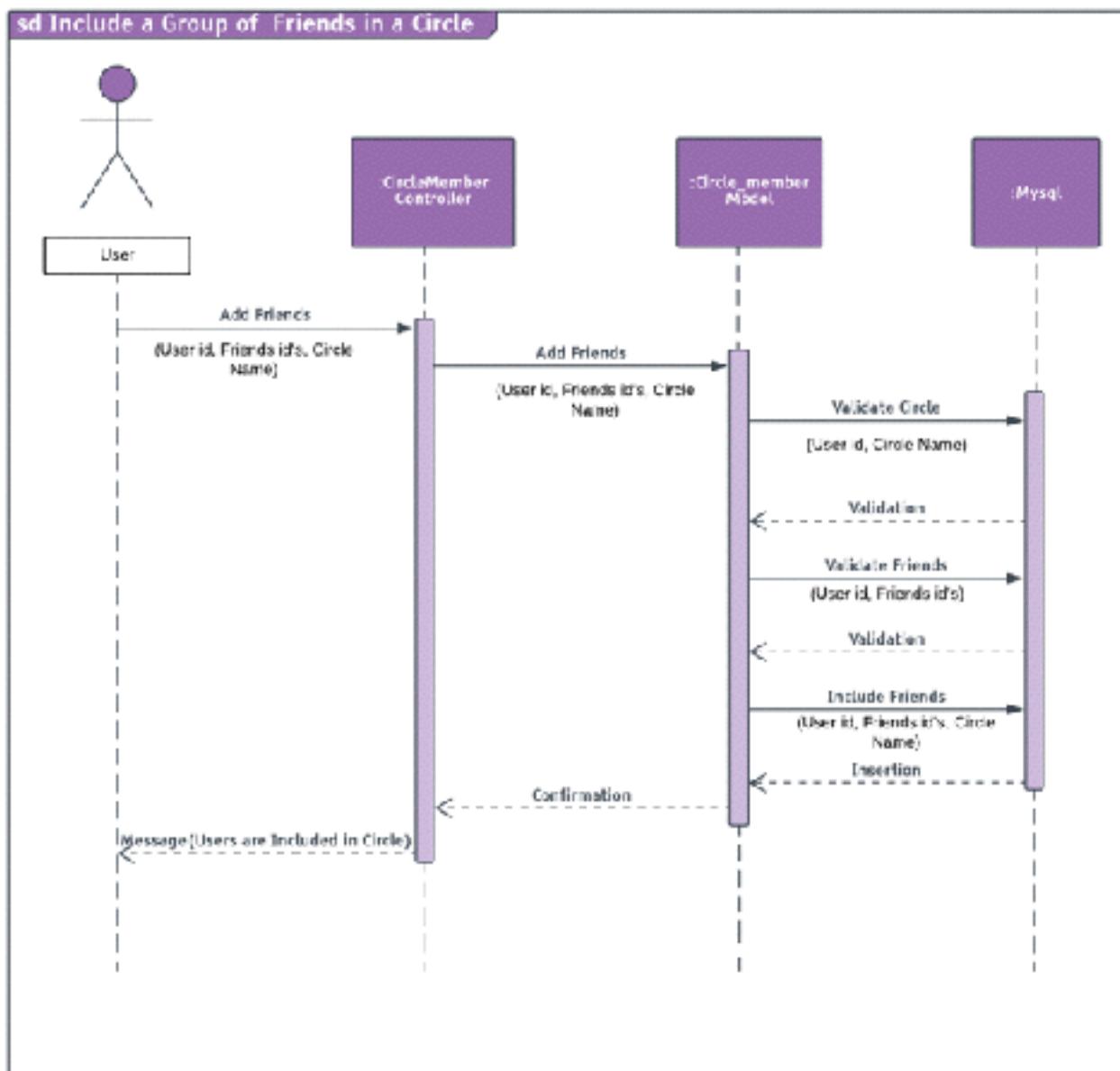
- Update circle thumbnail



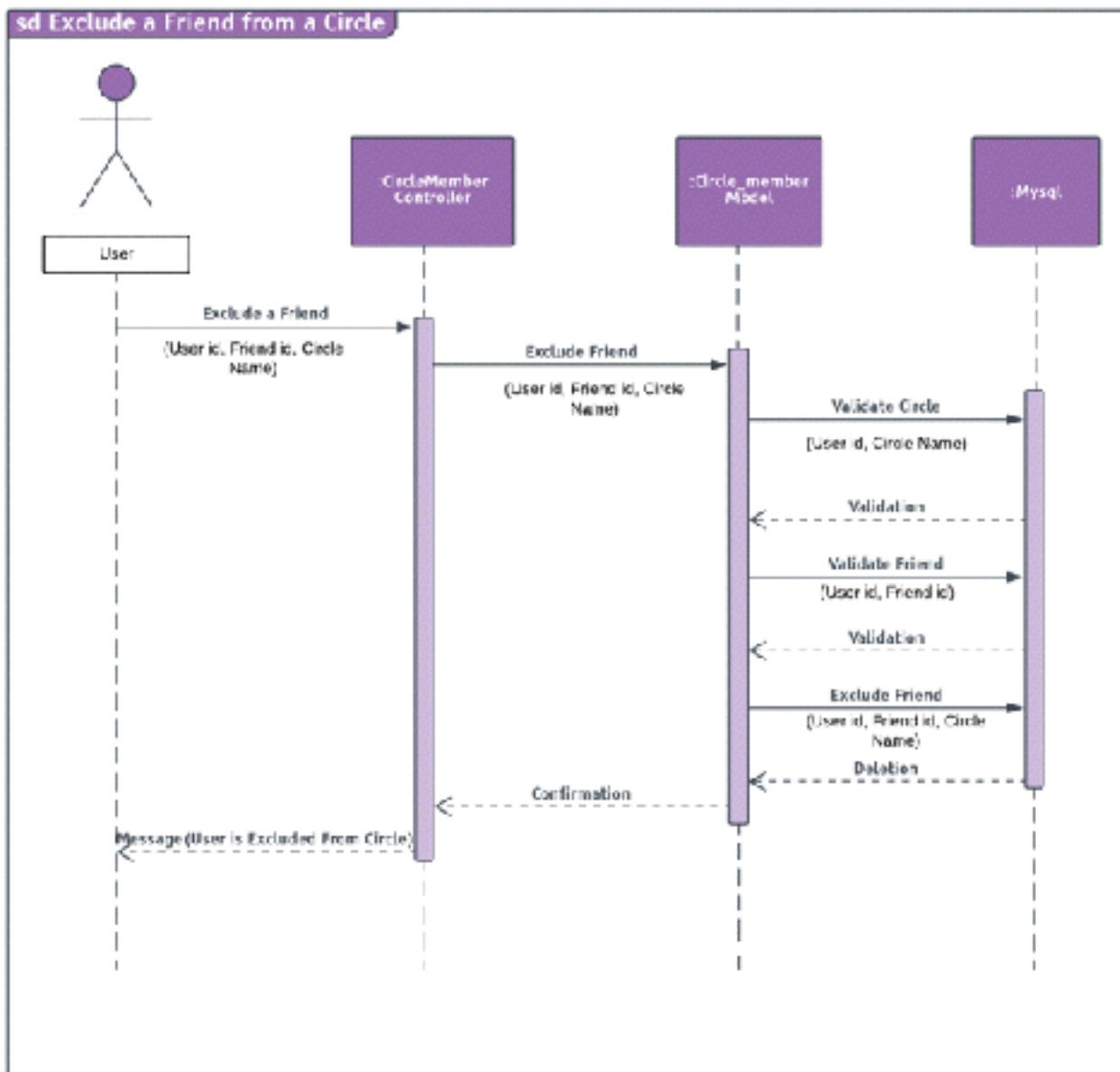
- Include a friend to a circle



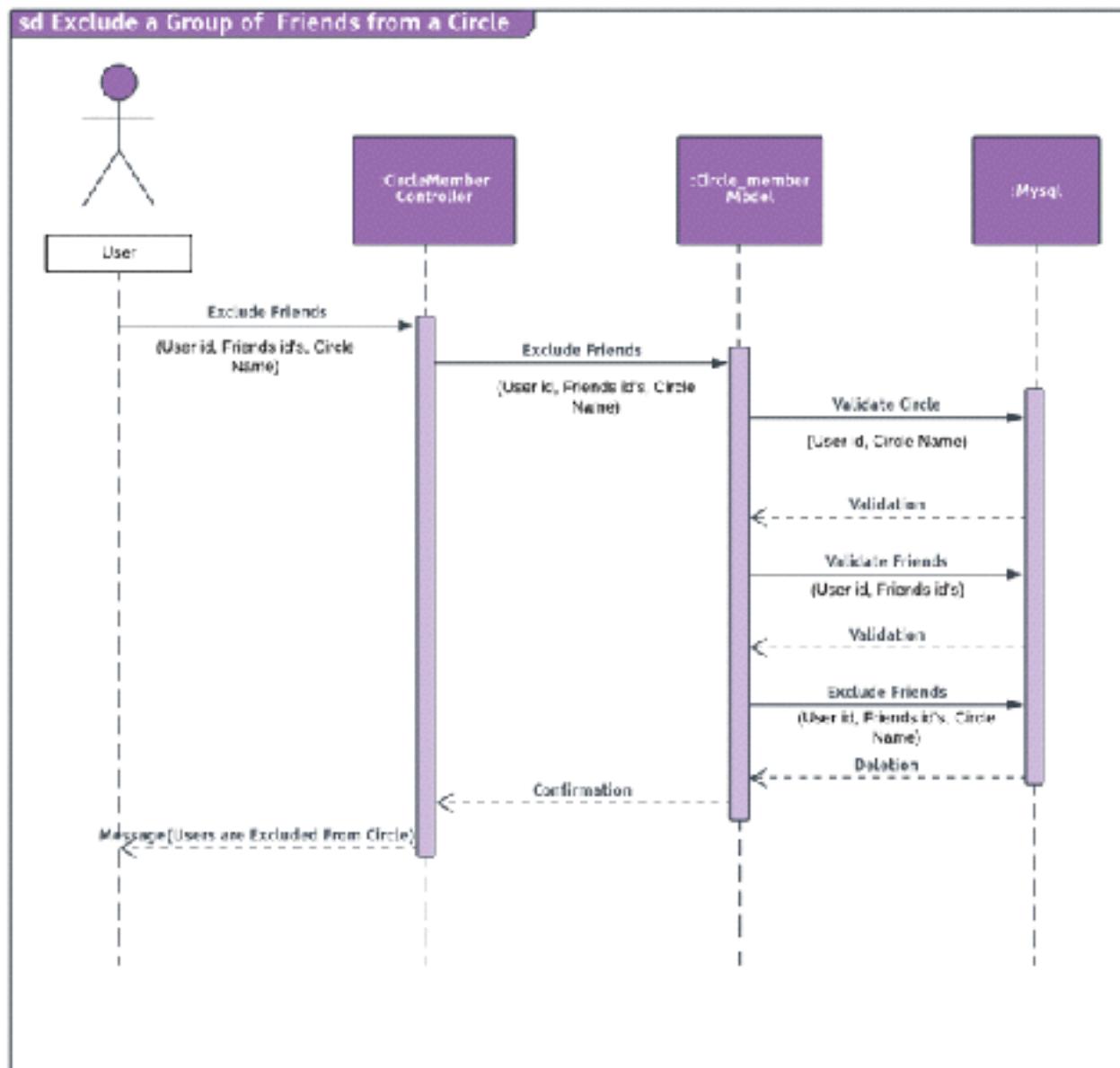
- Include group friends to a circle at a time.



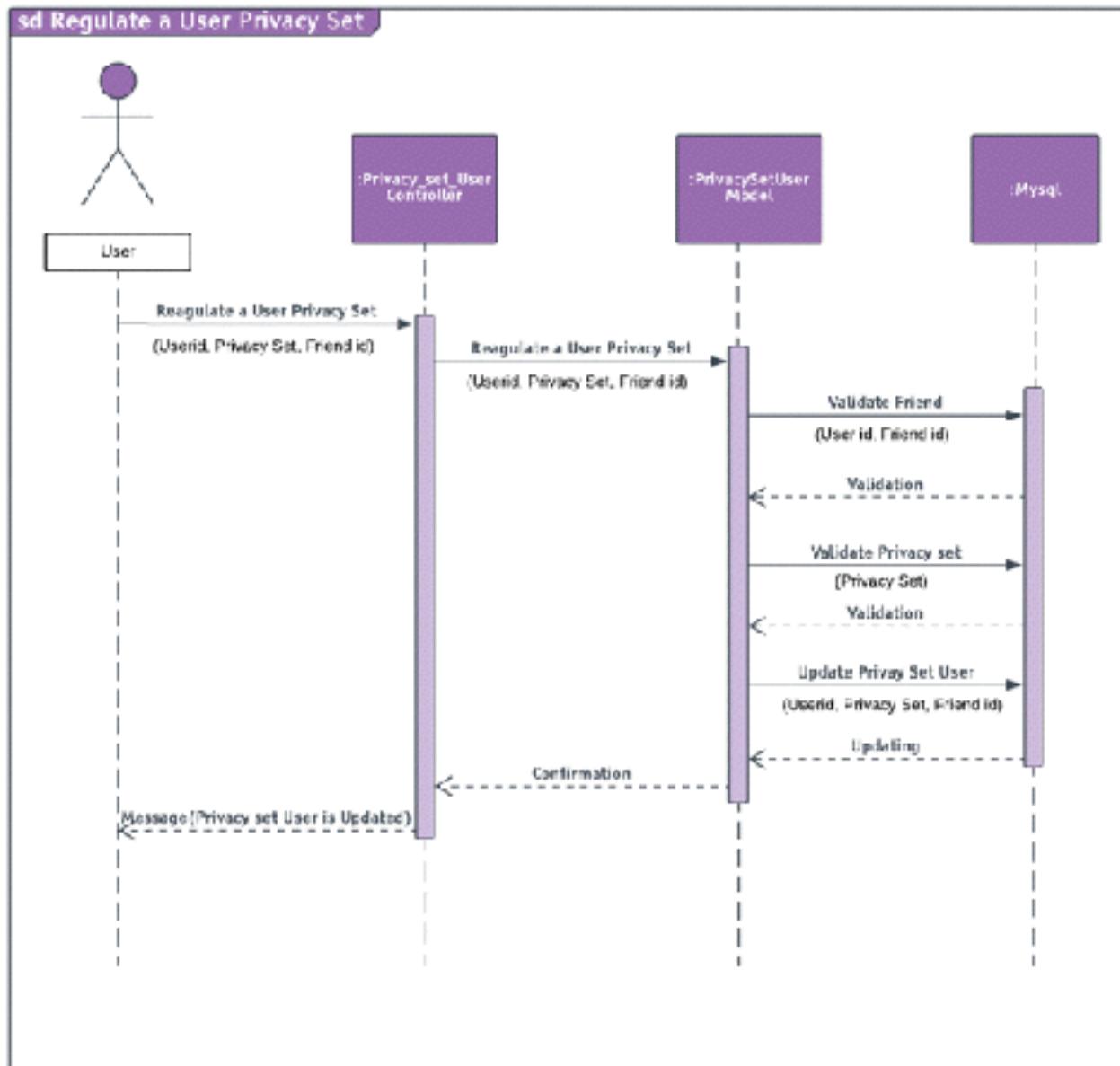
- Exclude a friend from a circle



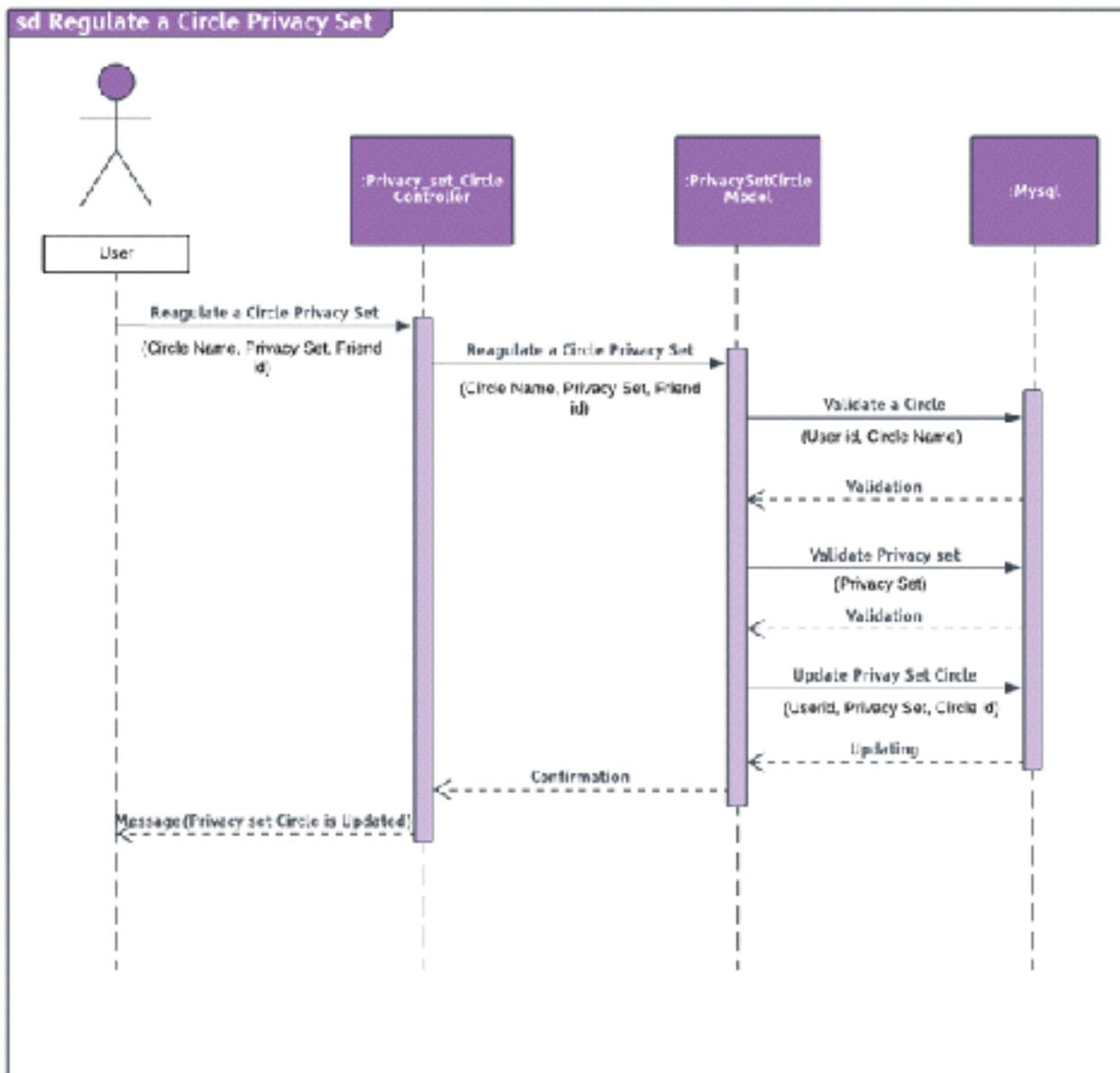
- Exclude group friends from a circle at a time.



- Regulate a privacy set for users

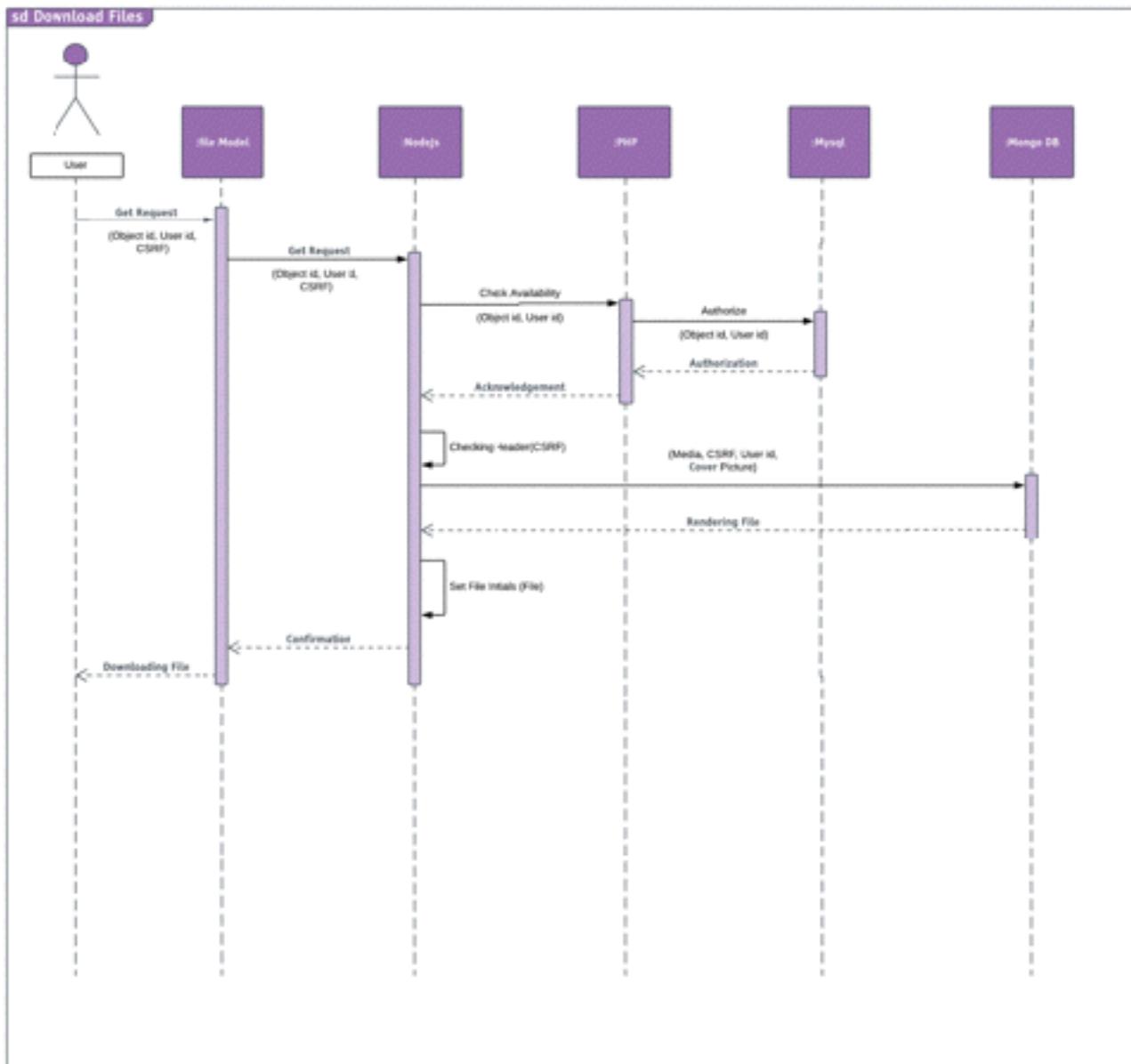


- Regulate a privacy set for circles

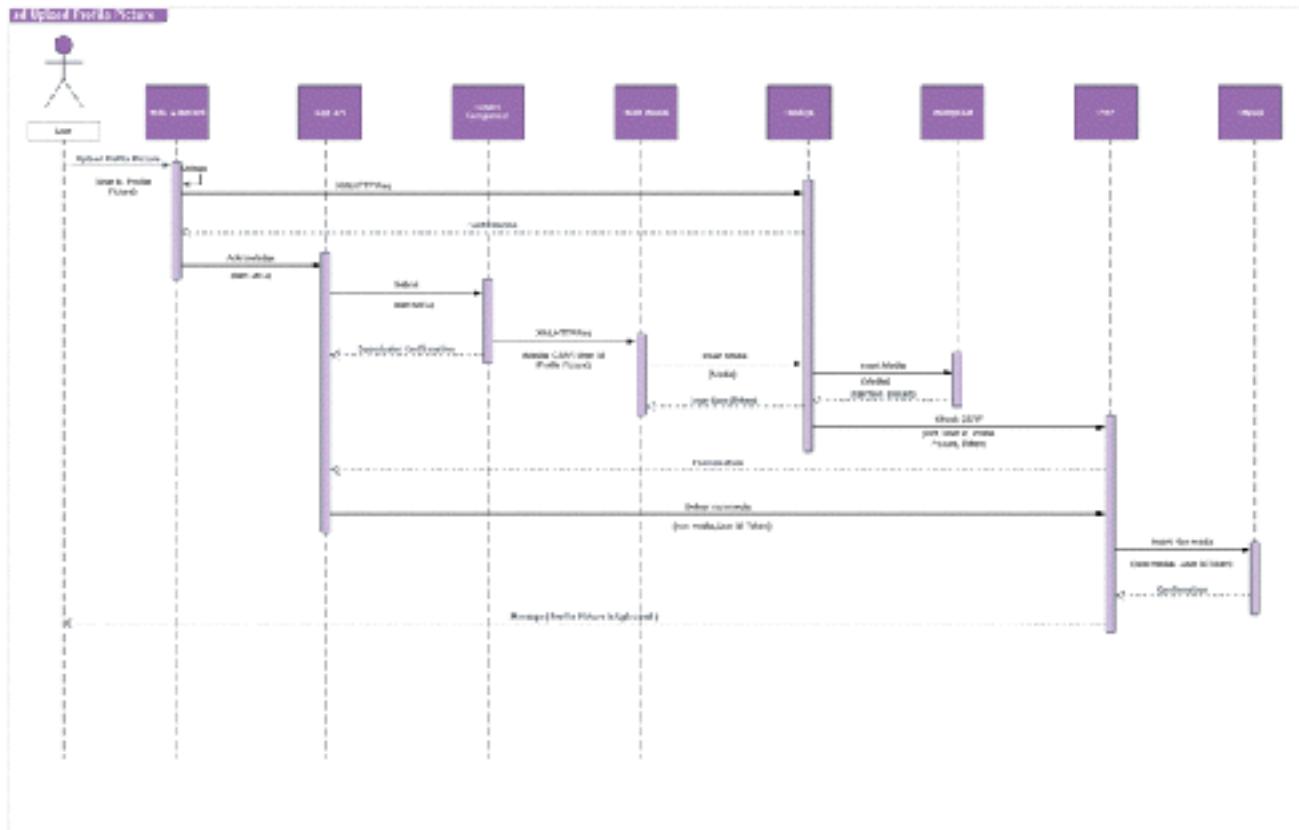


-
- Regulate a parent object privacy set for circle.

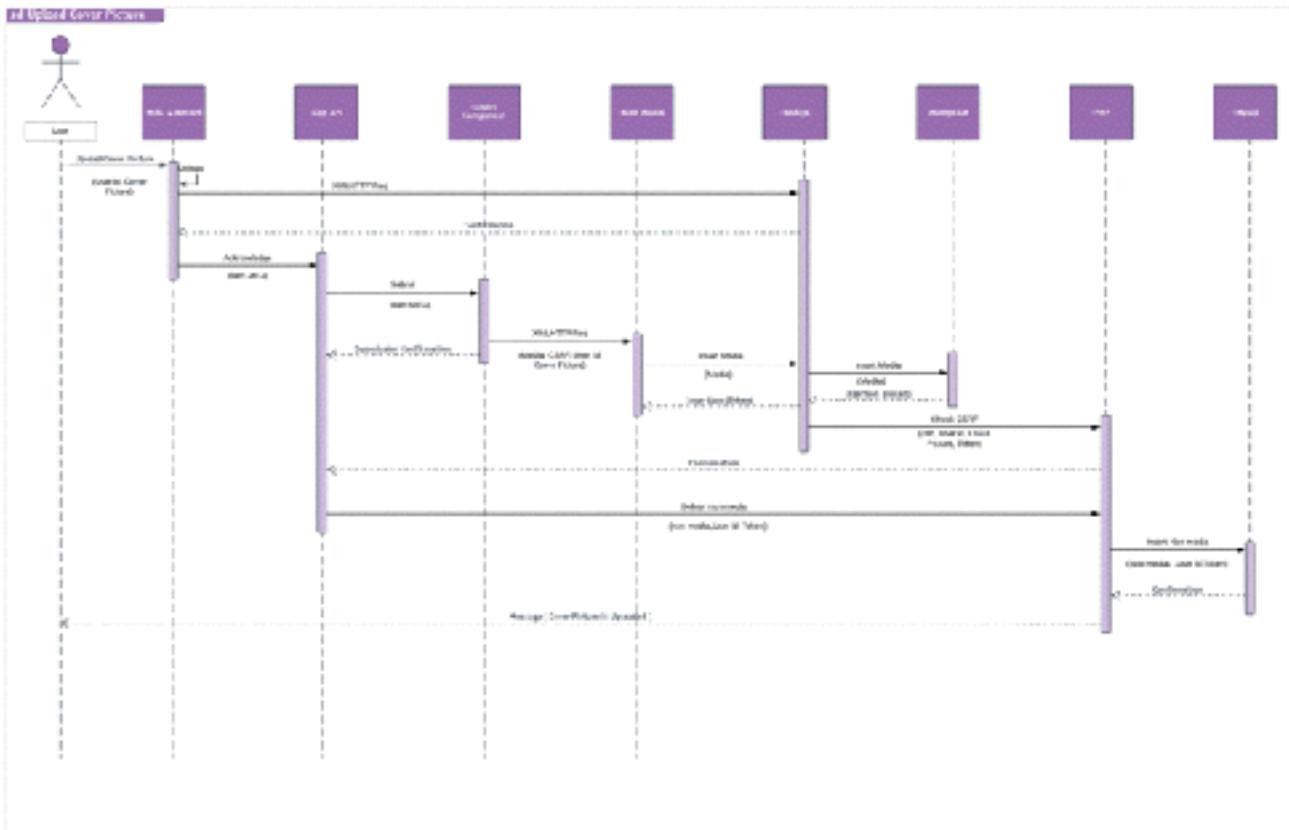
- Download files (with any extension).



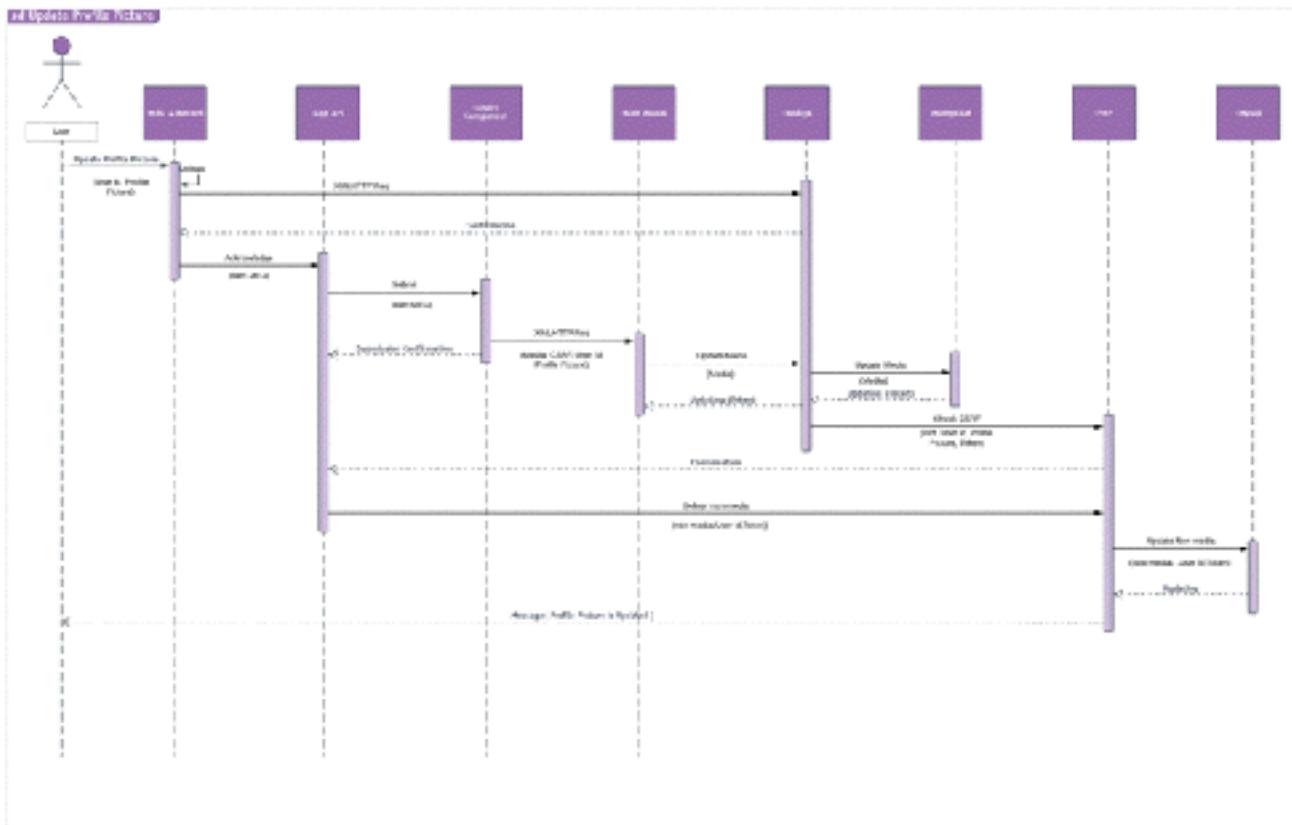
- Upload profile picture



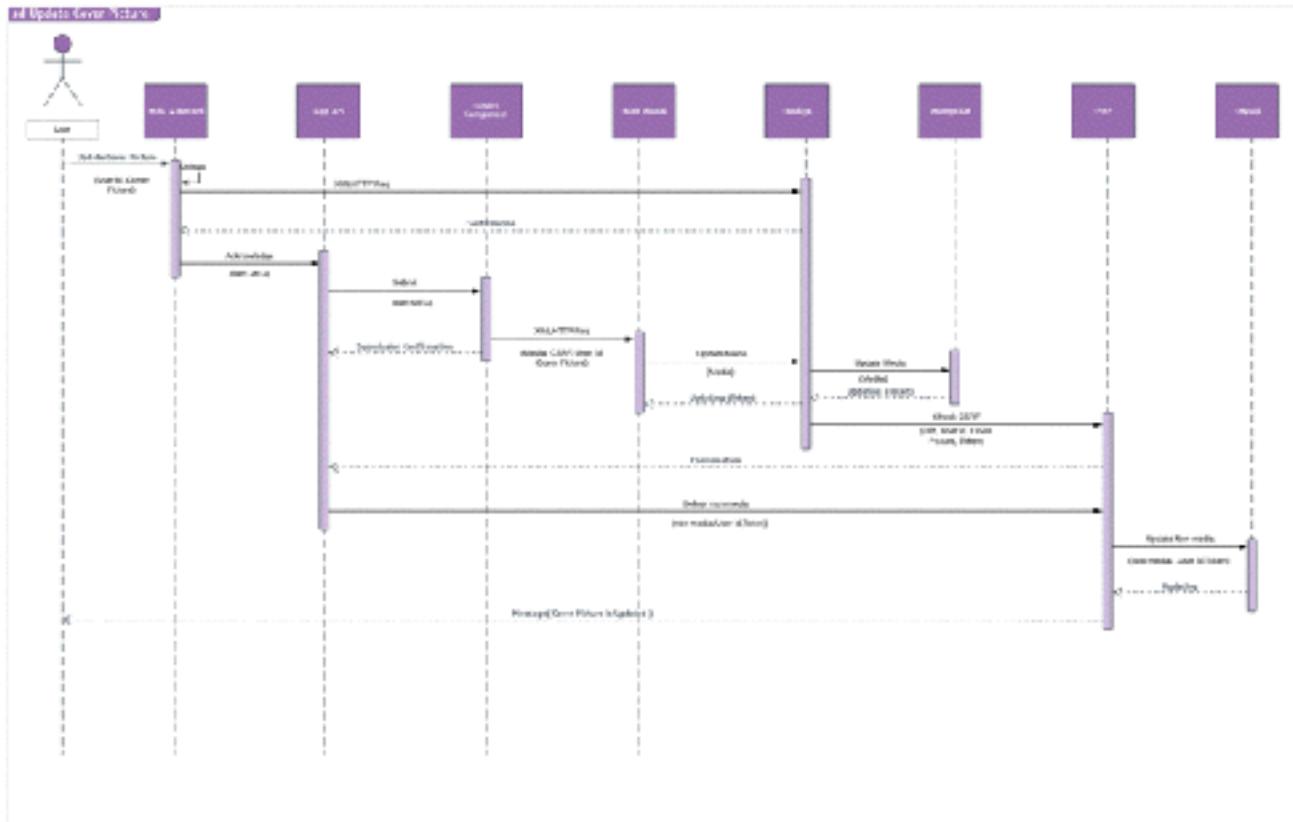
- Upload cover picture



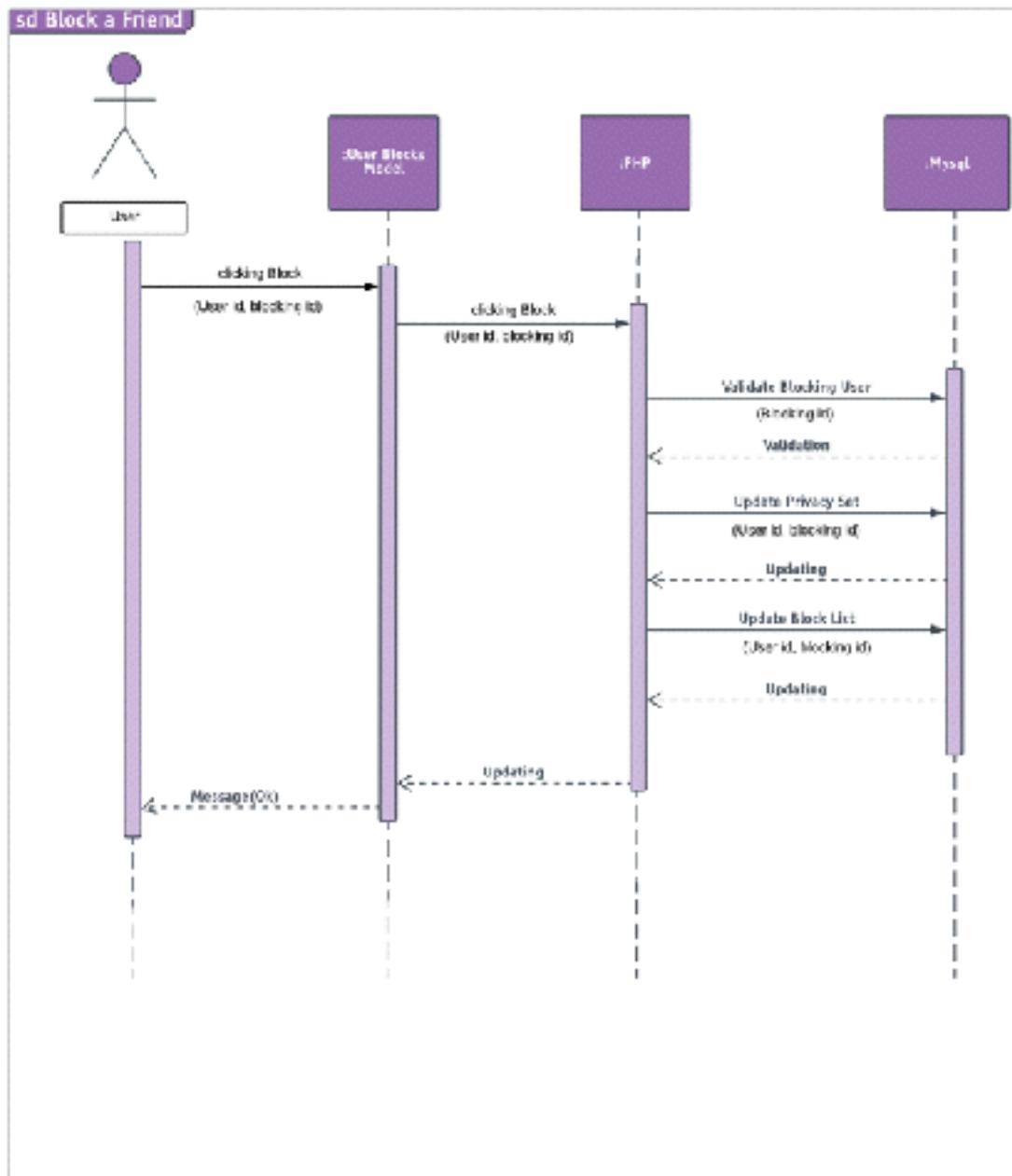
- Update profile picture



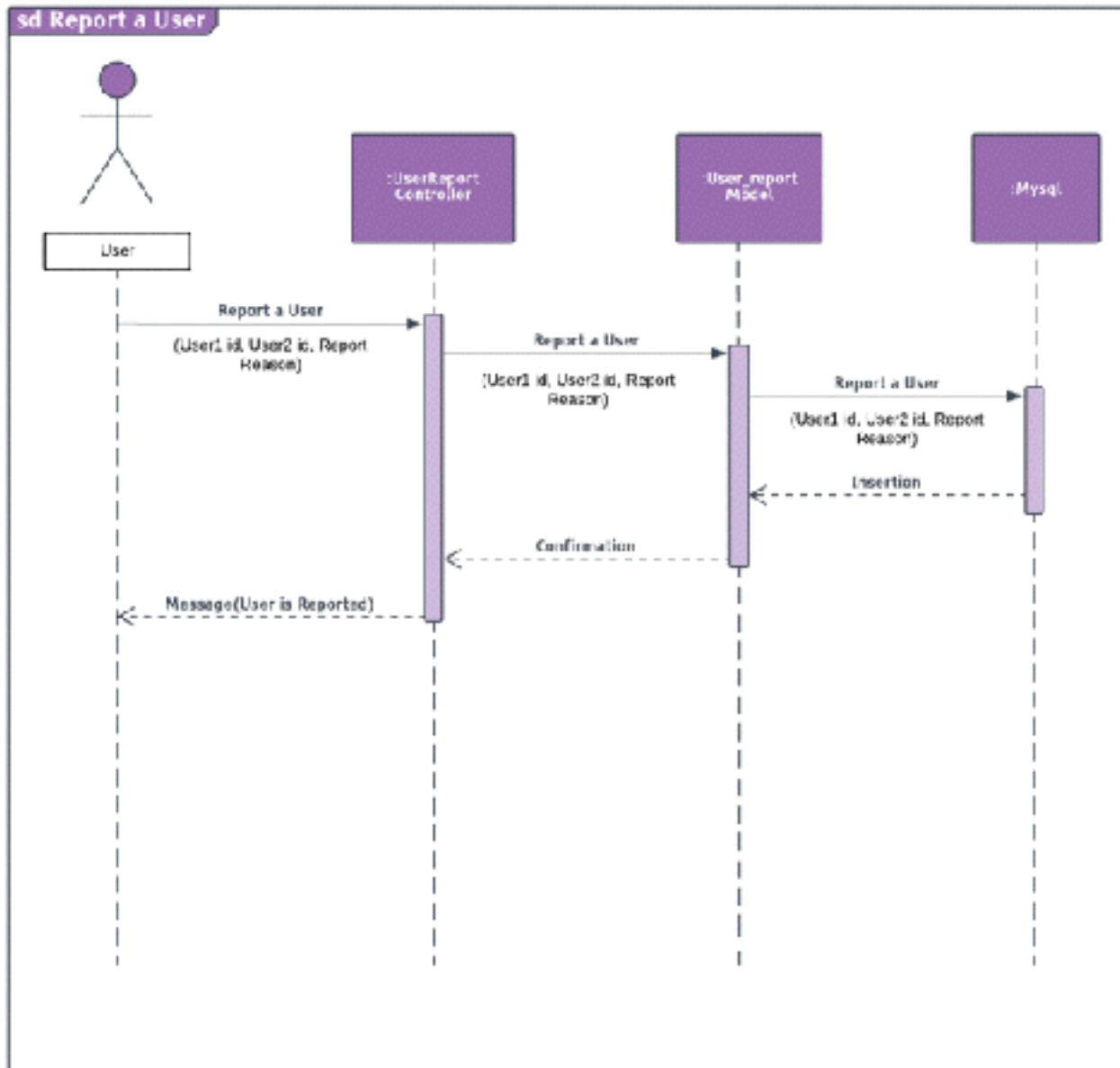
- Update cover picture



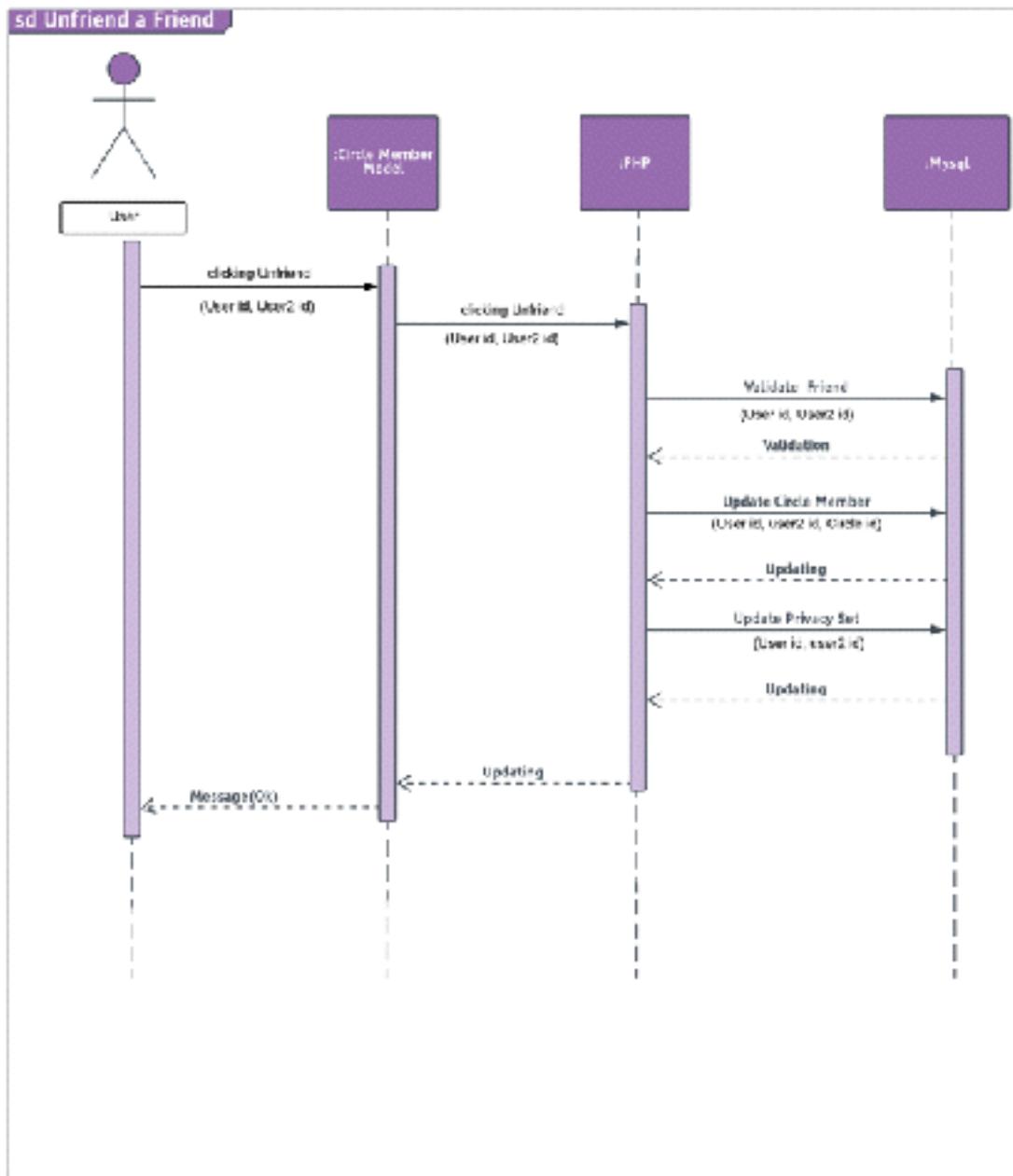
- Block a friend



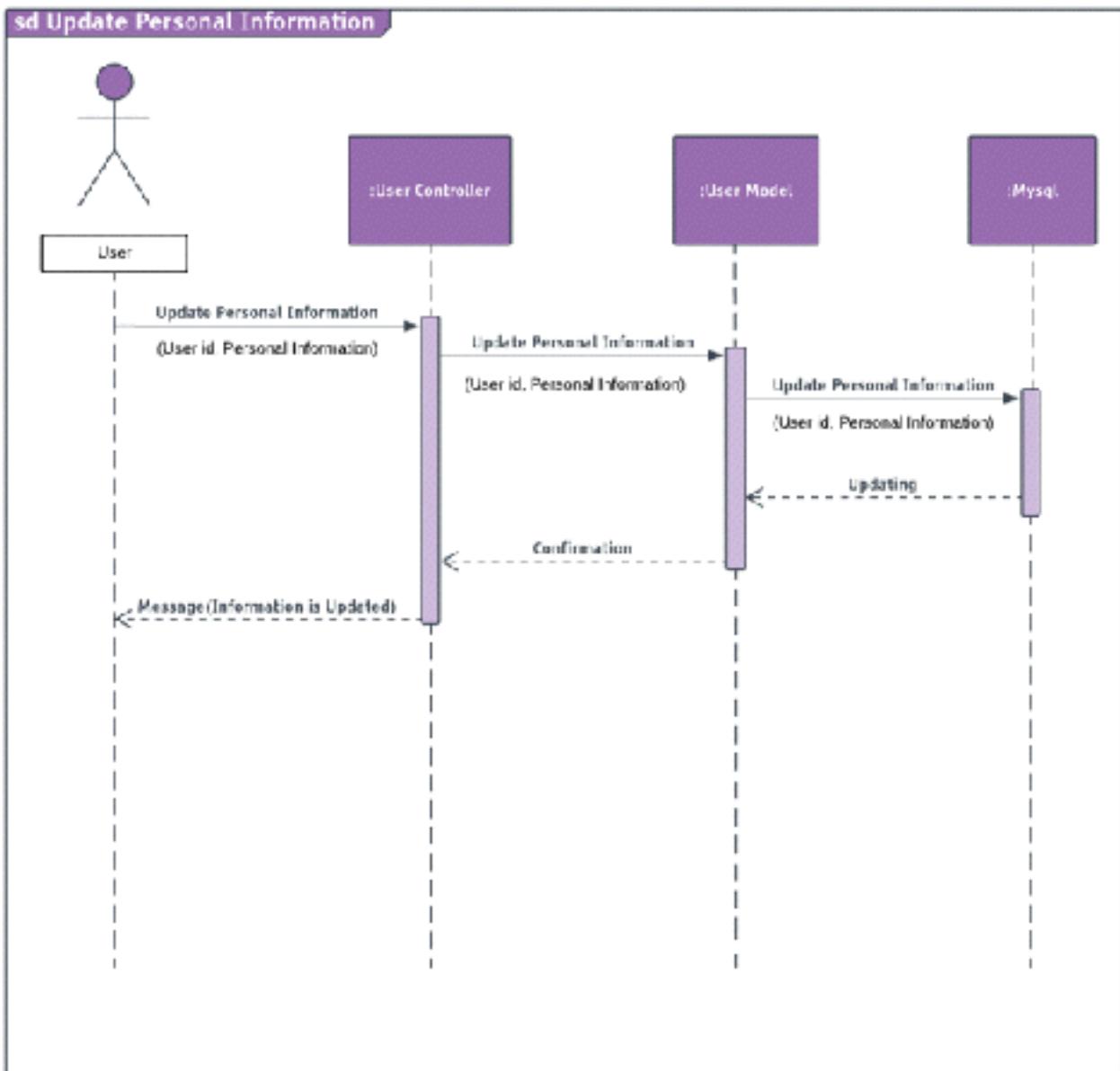
- Report a User



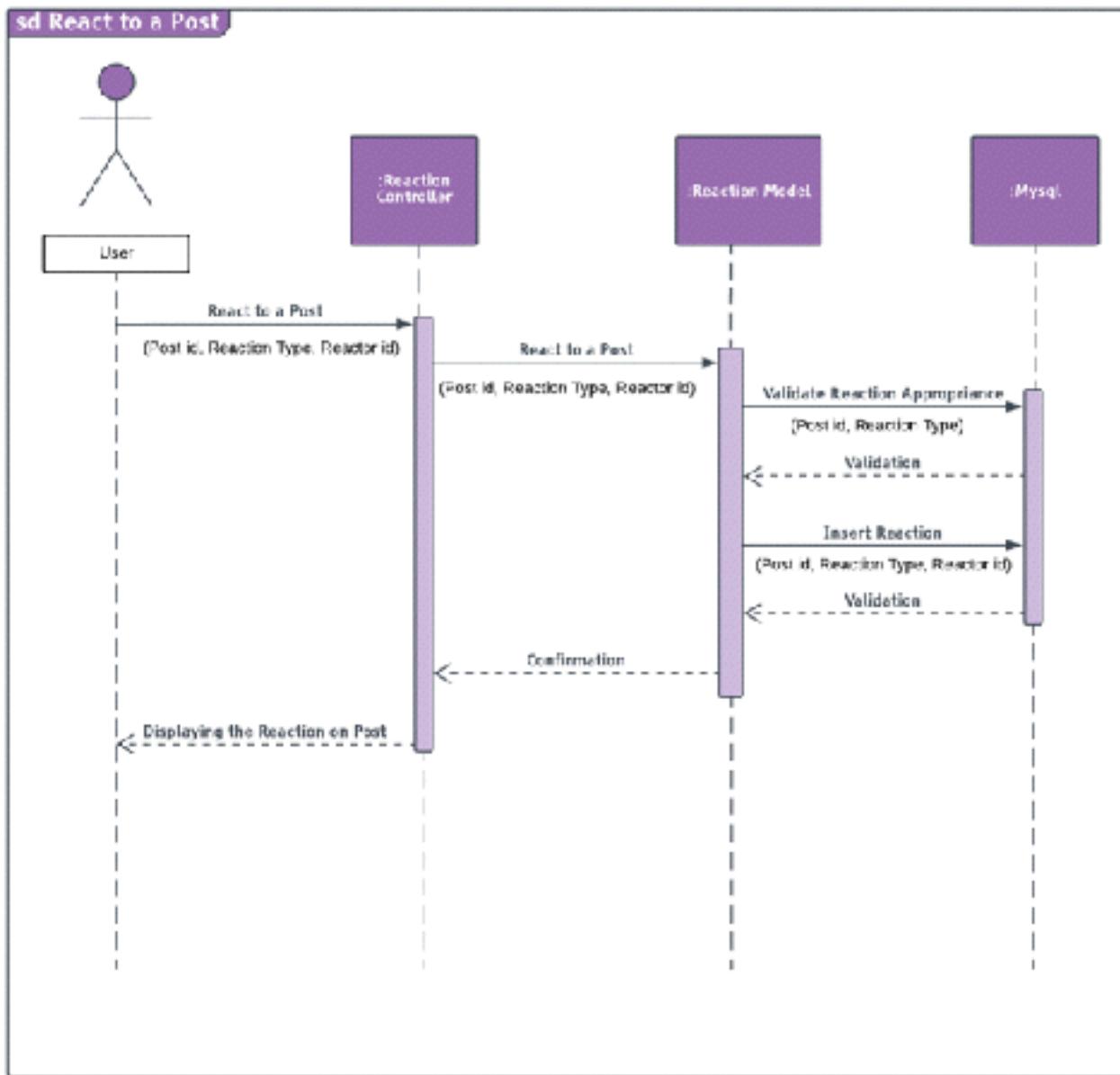
- Unfriend a friend



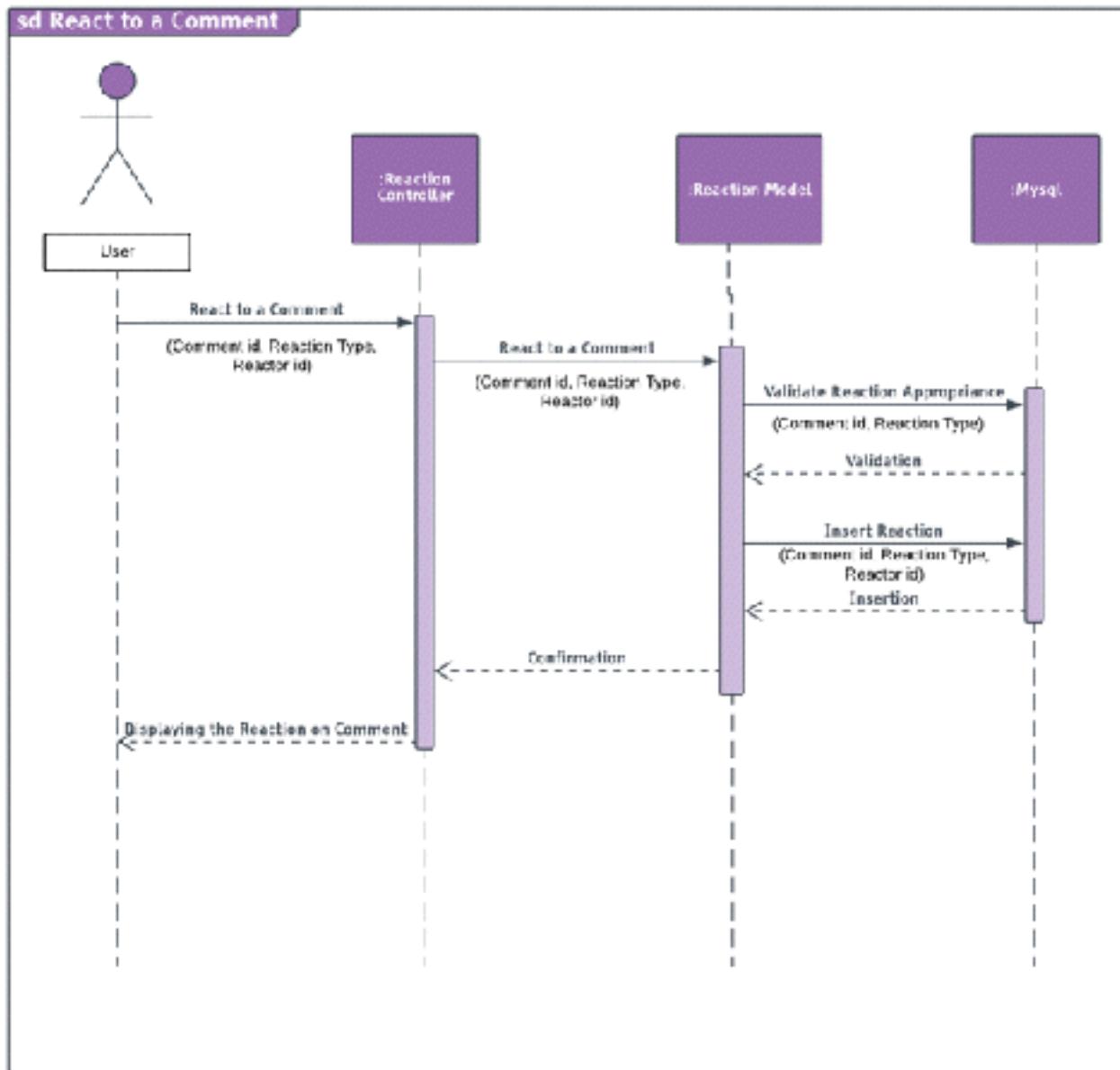
- Update personal information



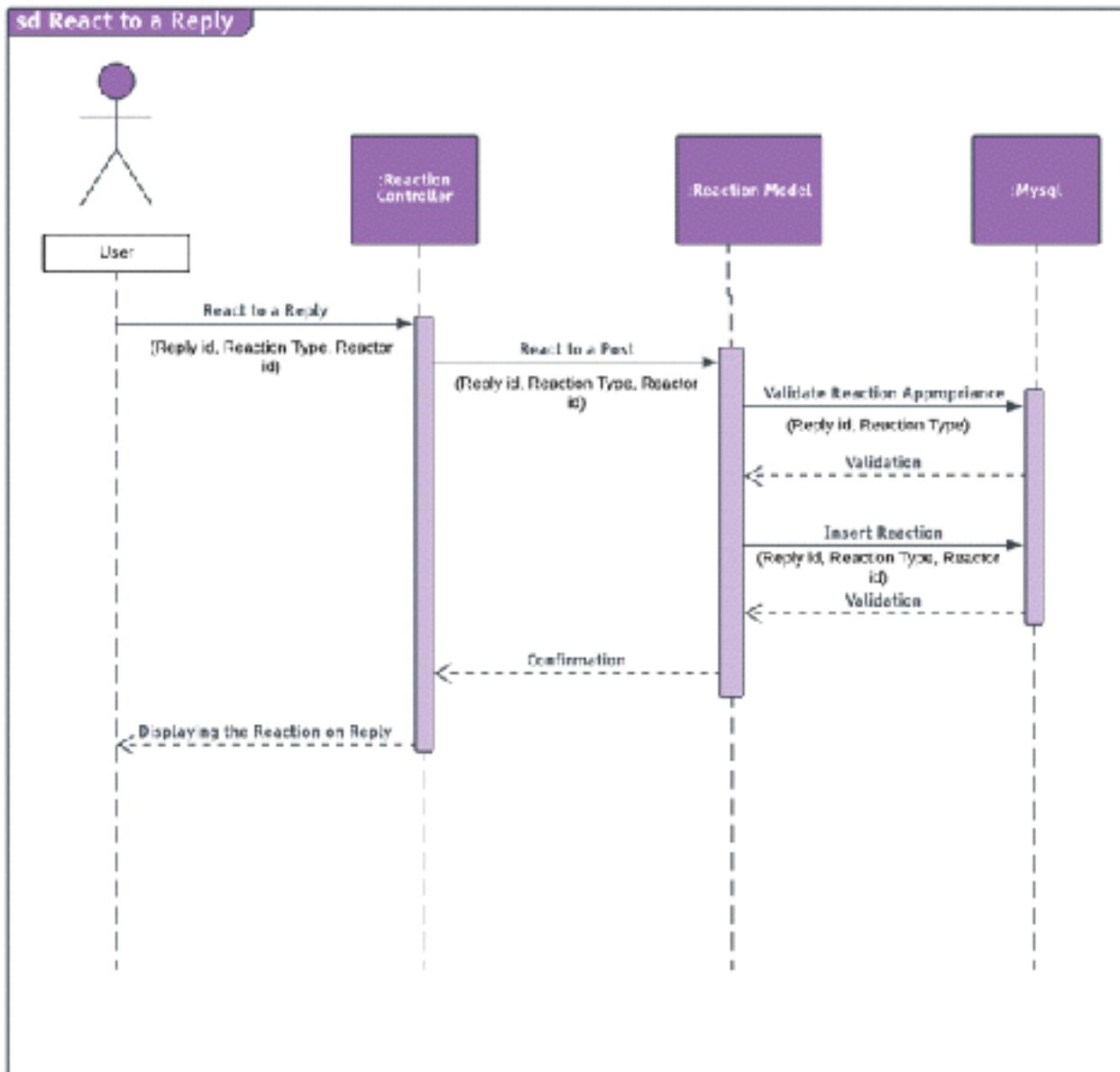
- React to post- when allowed to do so with allowed reactions.



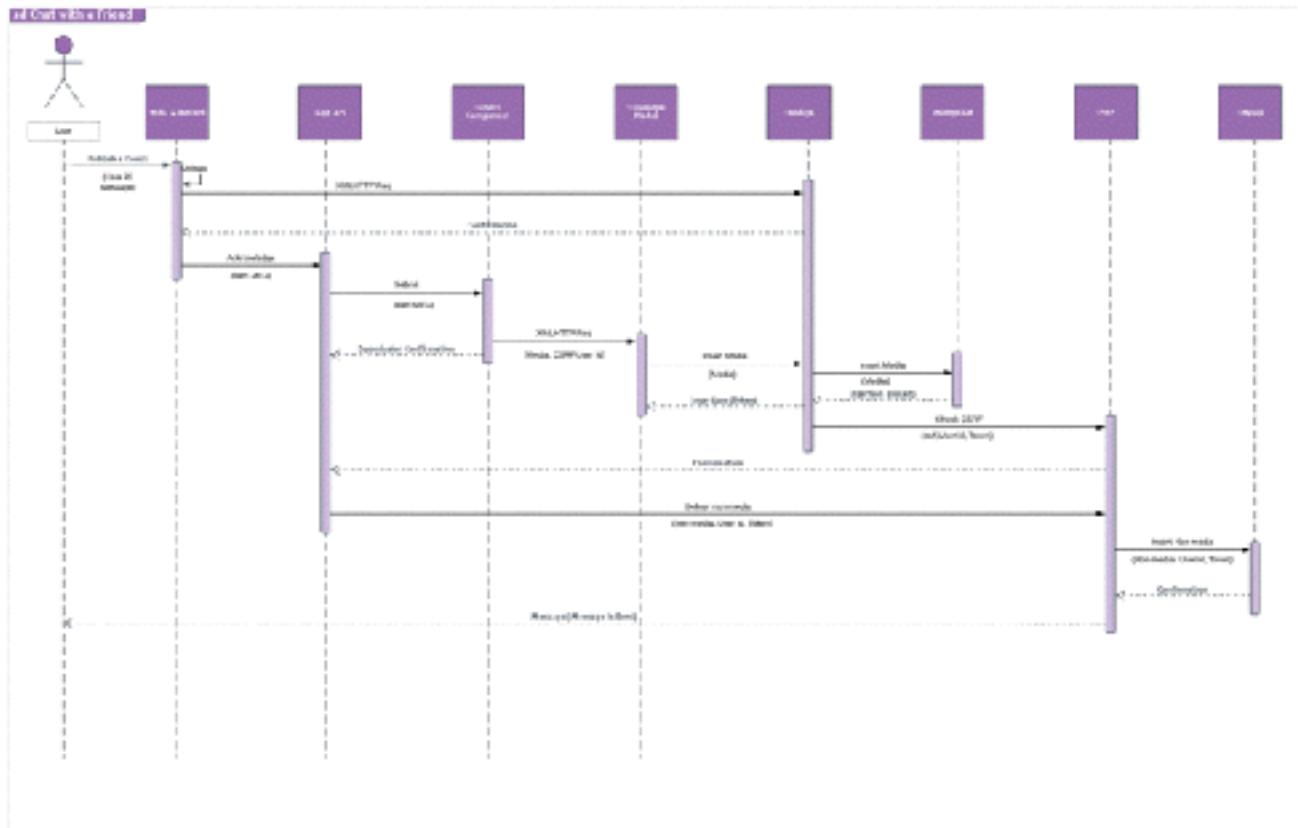
- React to comment- when allowed to do so with allowed reactions.



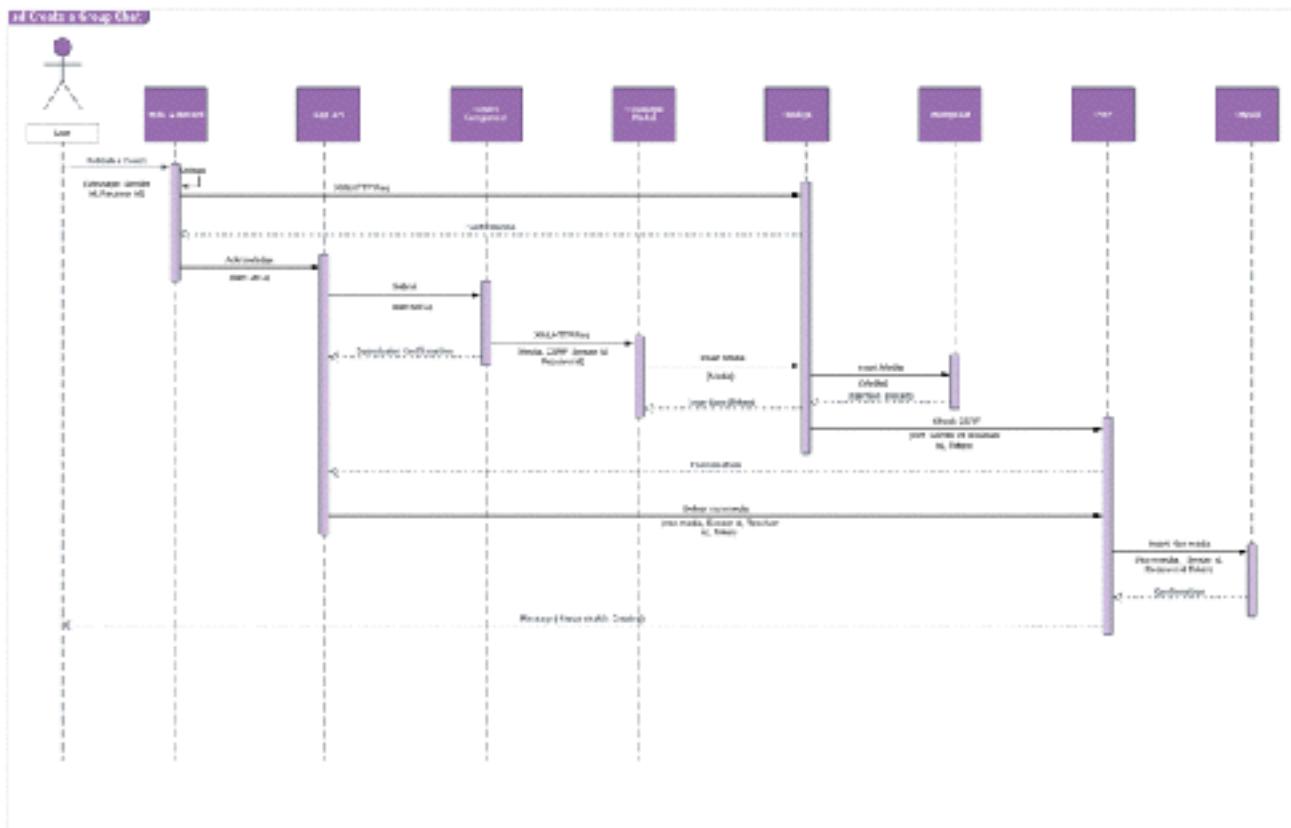
- React to reply- when allowed to do so with allowed reactions.



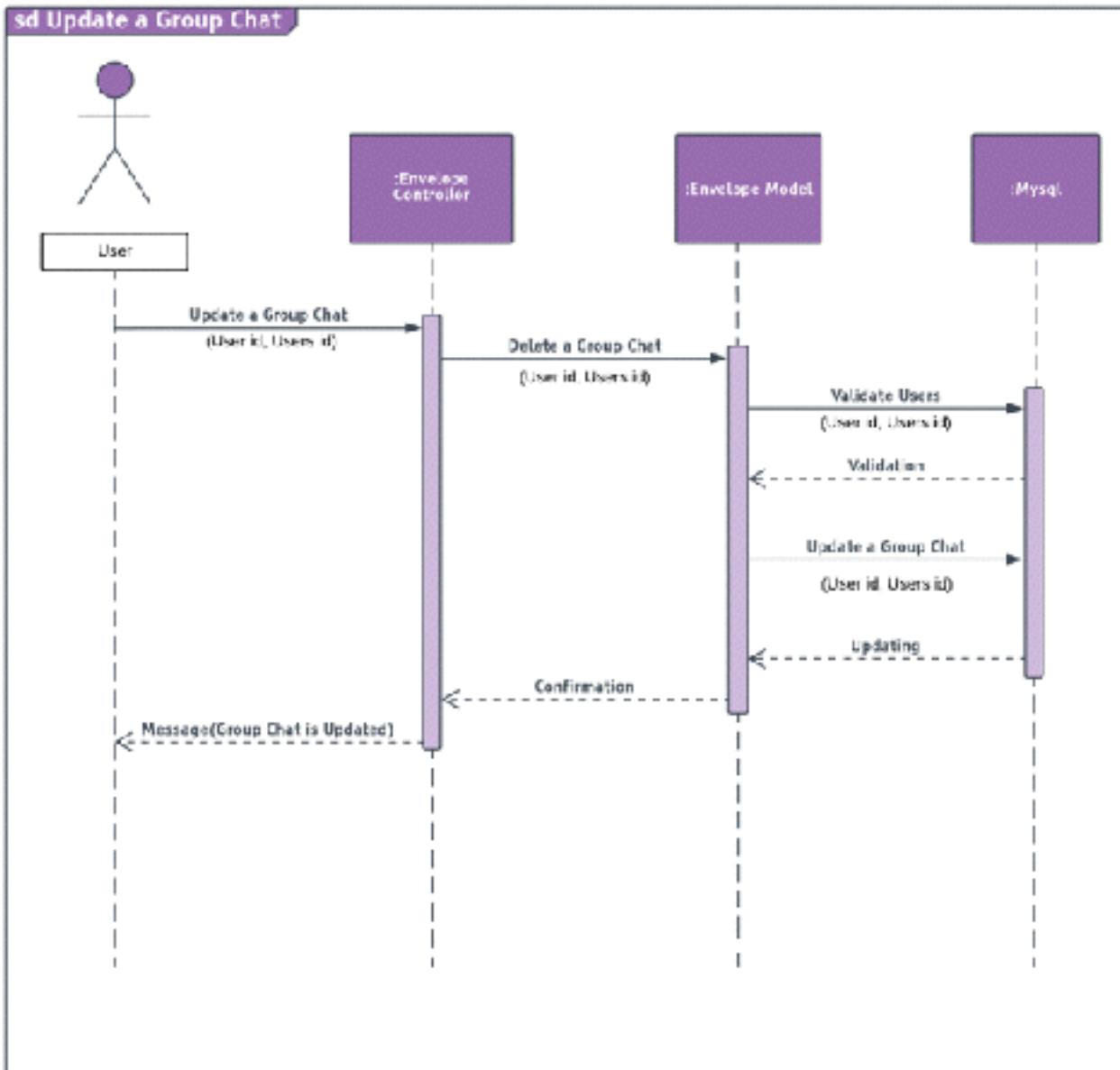
- Chat with a friend



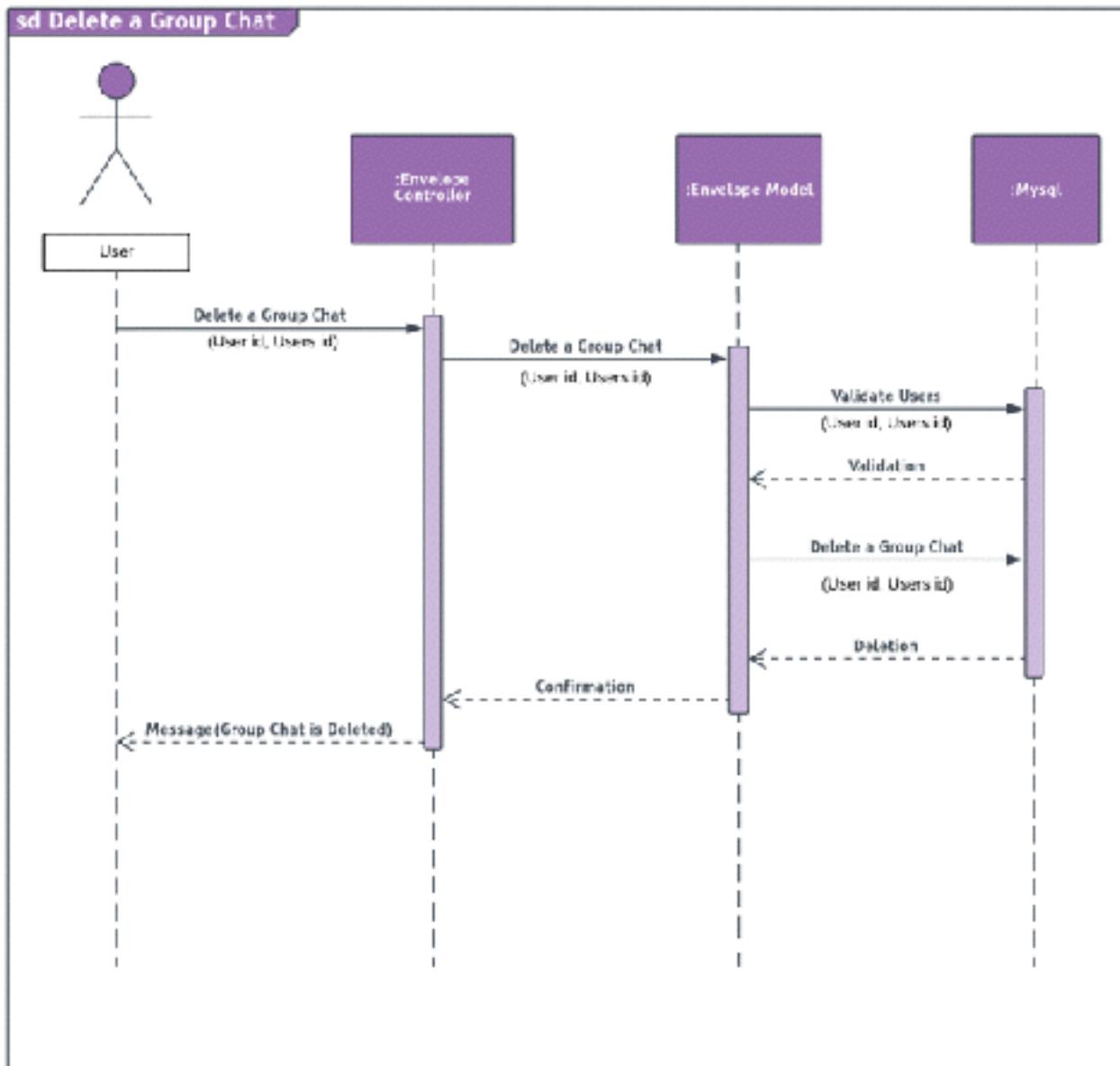
- Create a group chat



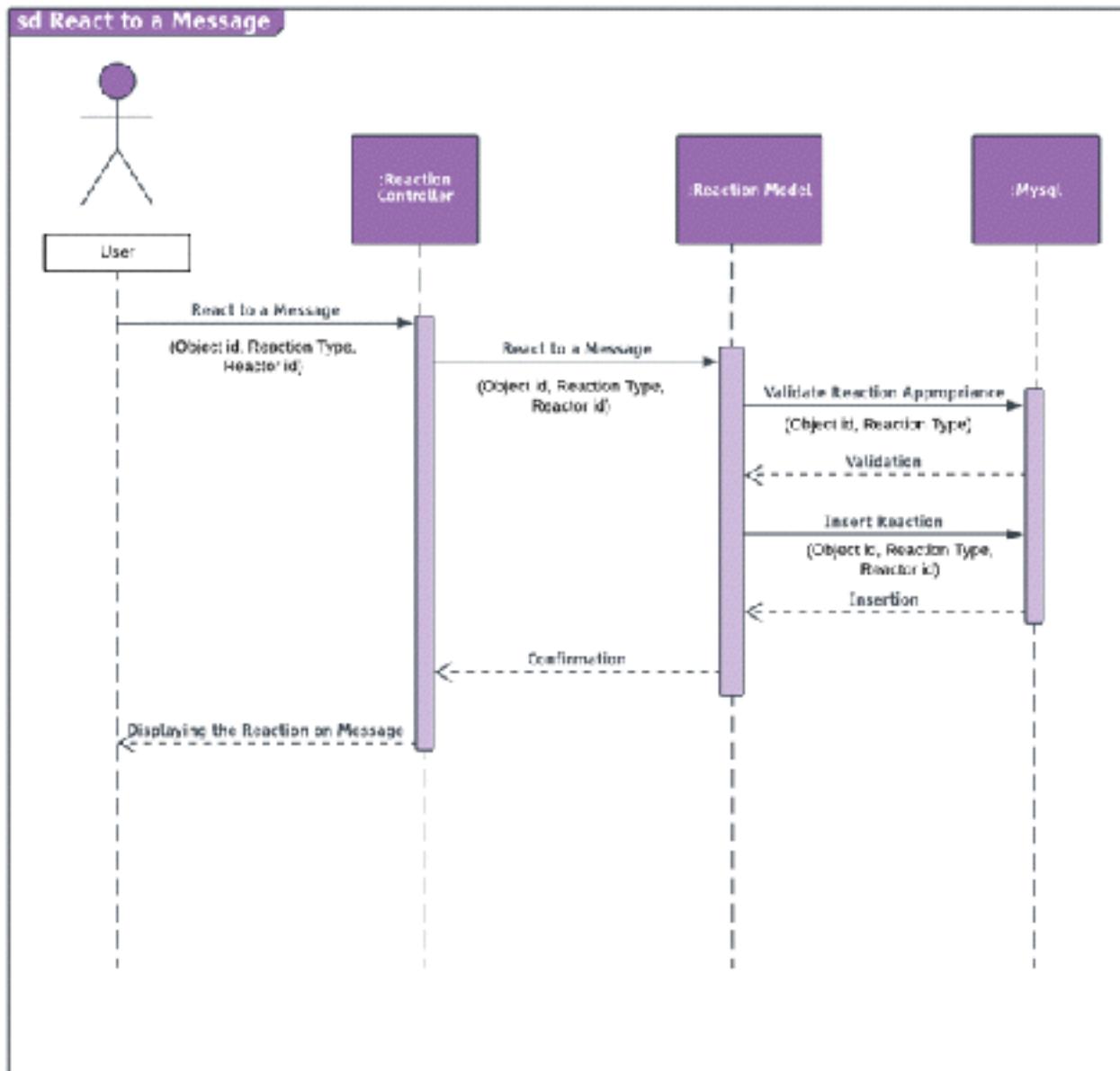
- Update a group chat - when allowed to do so.



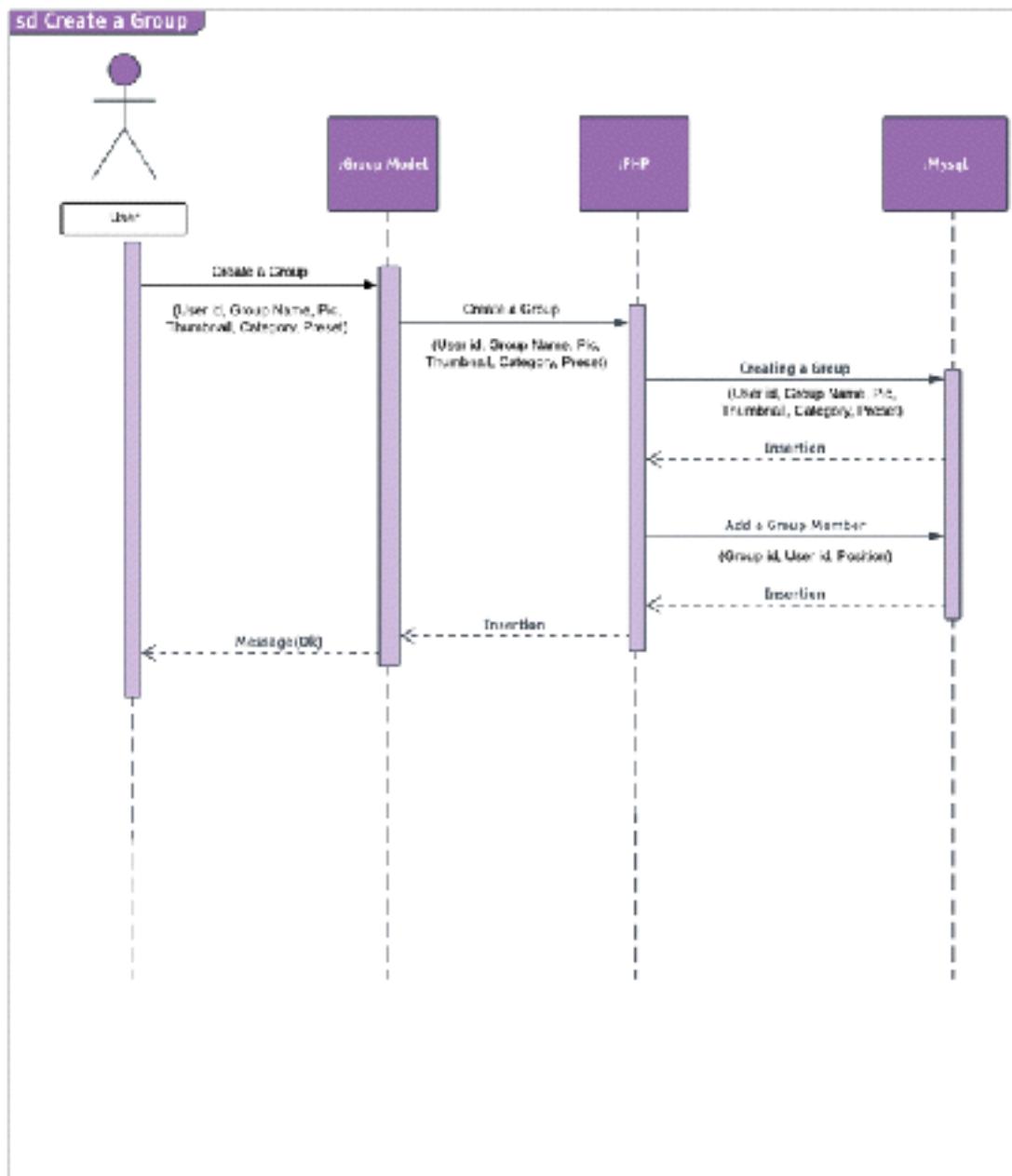
- Delete a group chat- when allowed to do so.



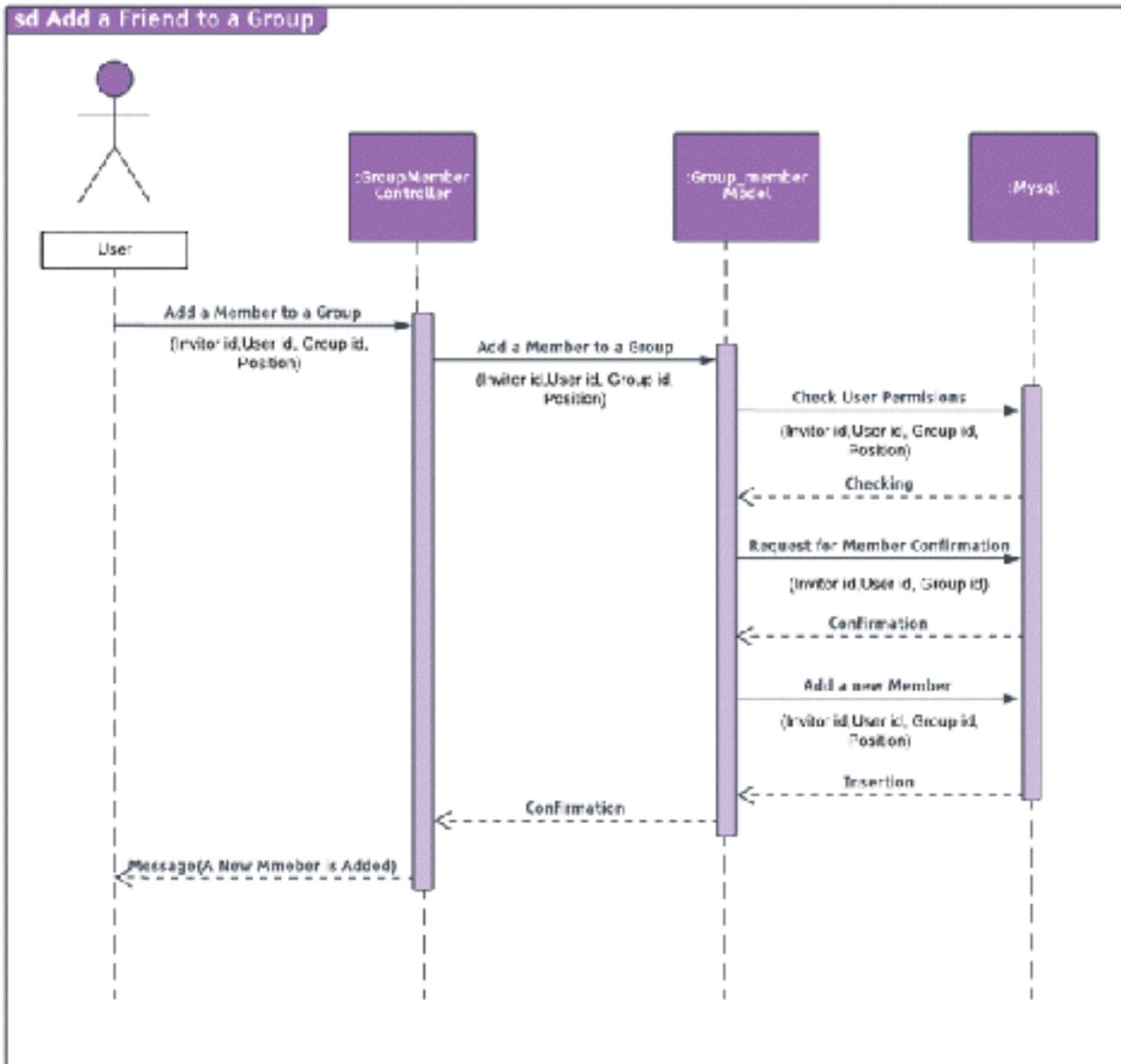
- React to messages



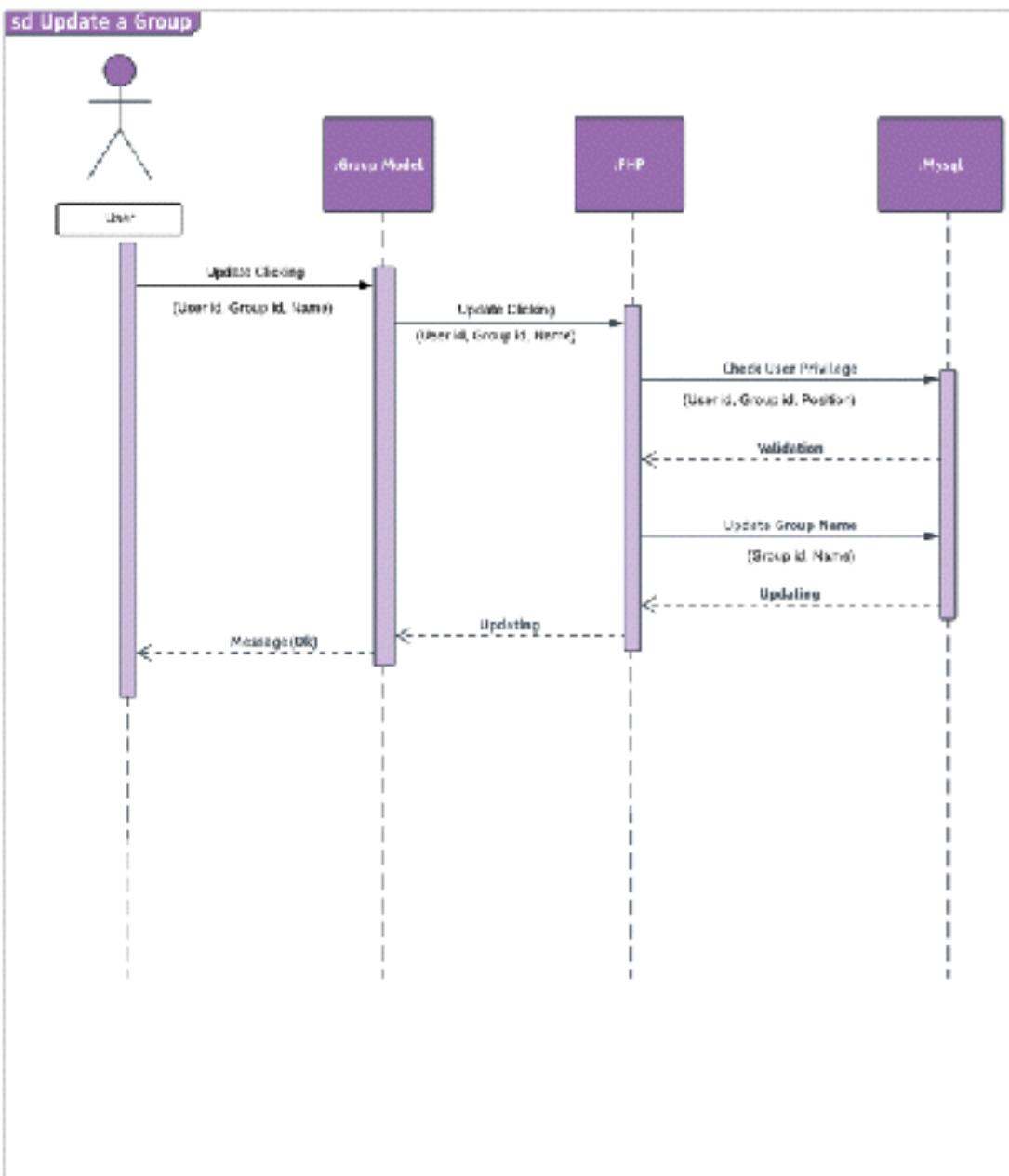
- Create a group



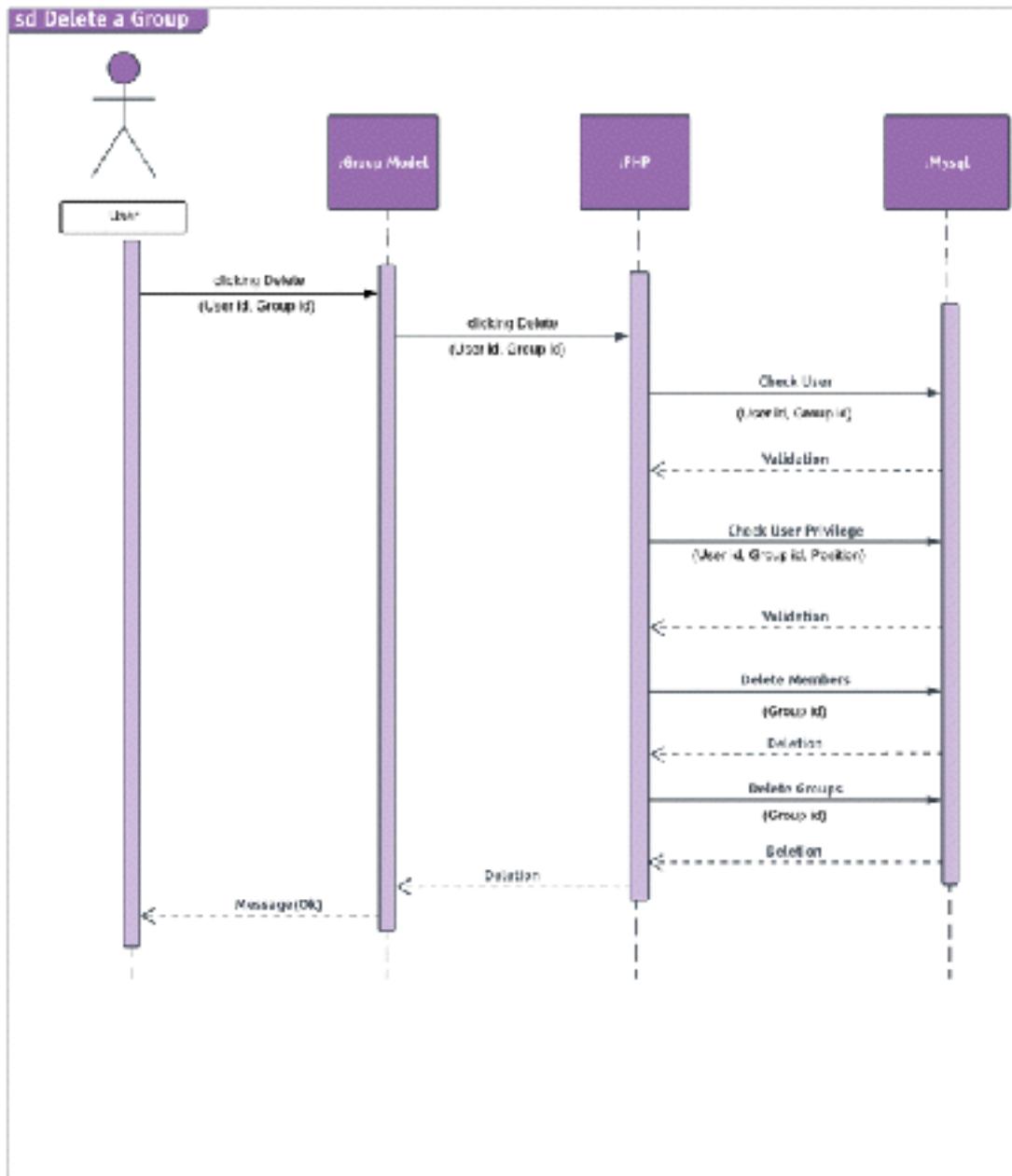
- Add friends to group



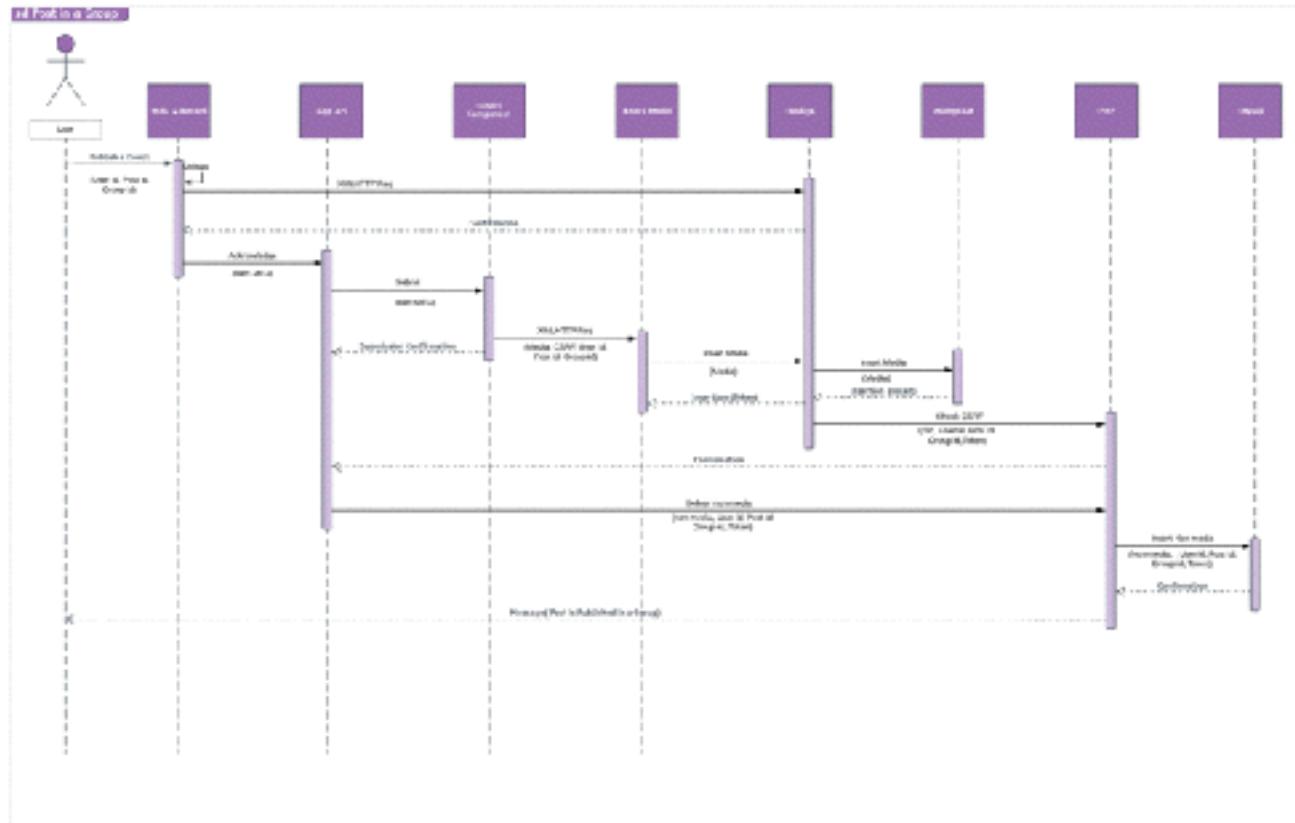
- Update a group- when allowed to do so.



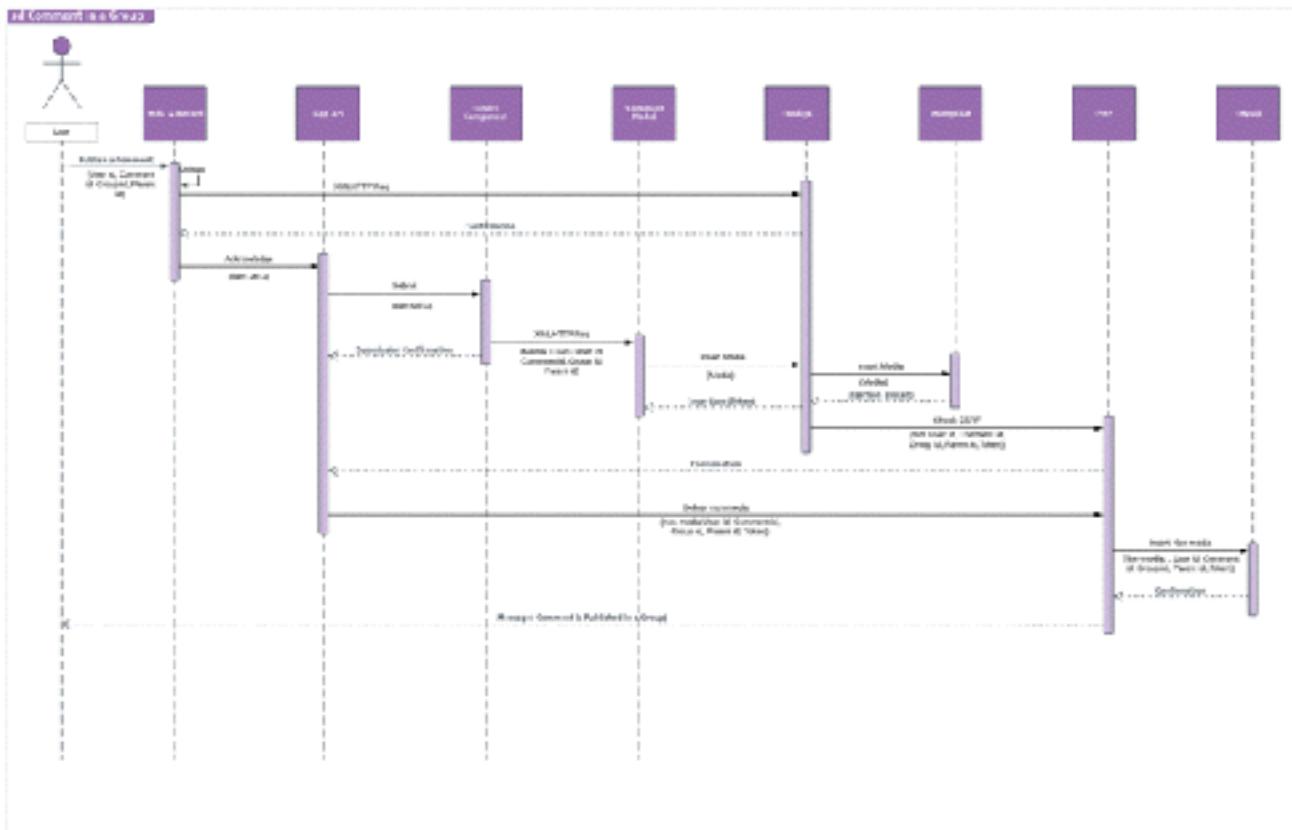
- Delete a group- when allowed to do so.



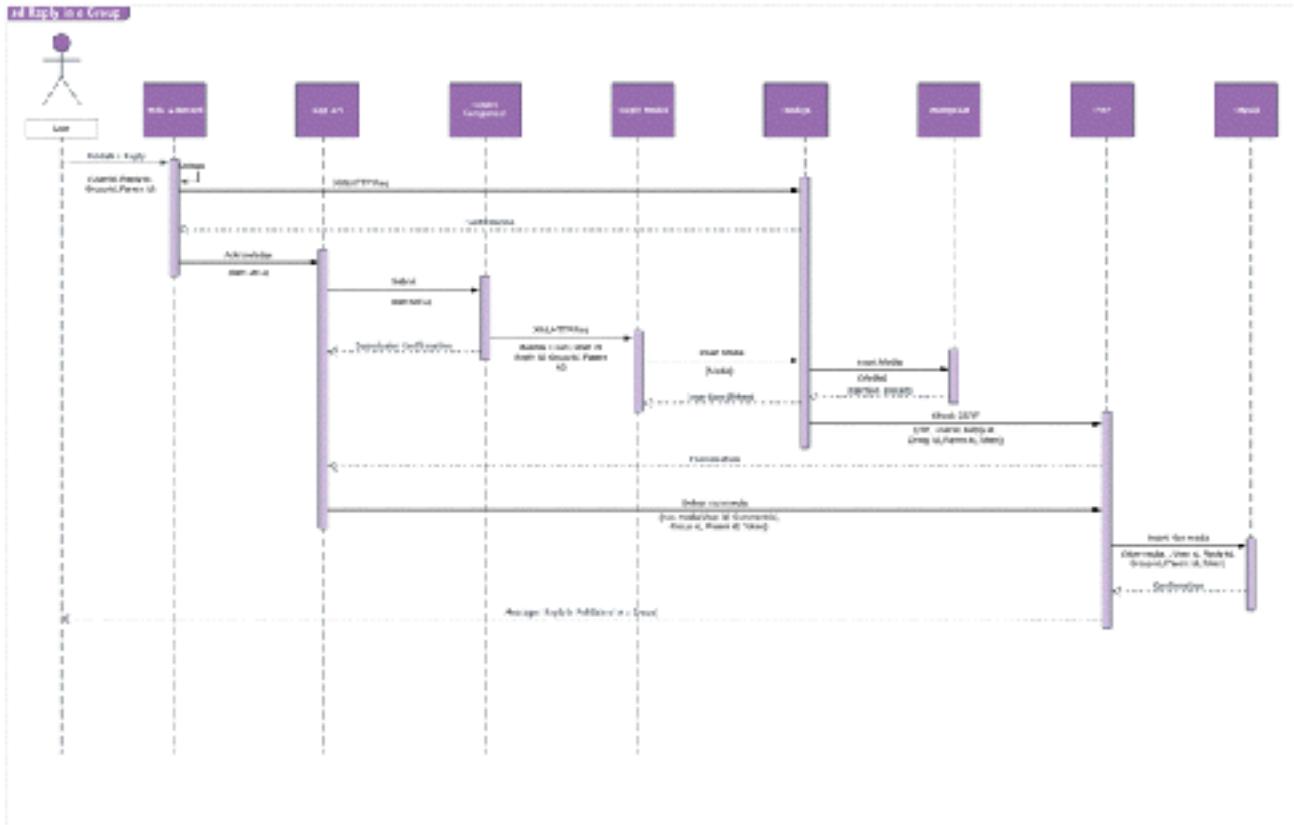
- Post in a group- when allowed to do so.



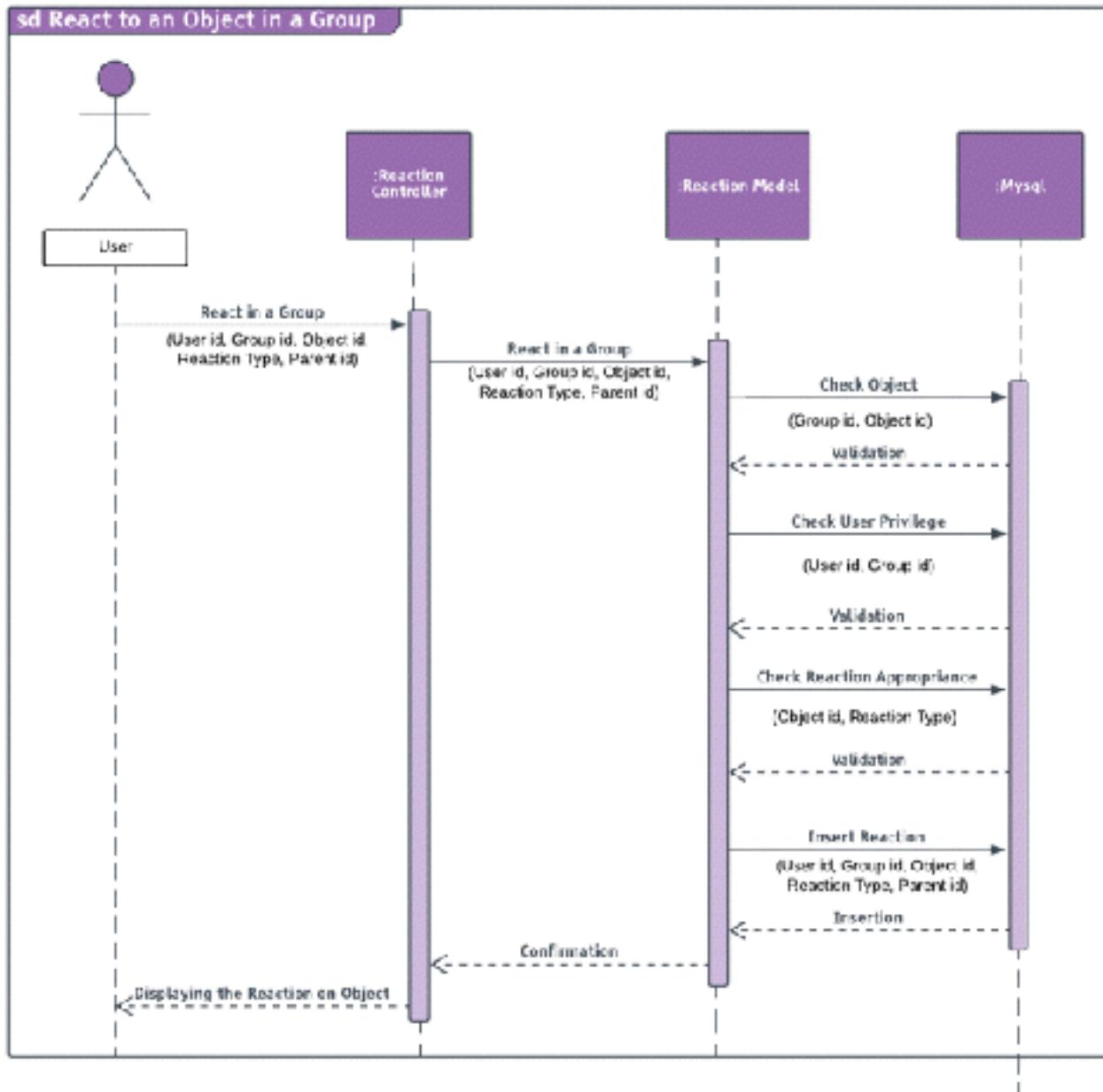
- Comment in a group- when allowed to do so.



- Reply in a group- when allowed to do so.

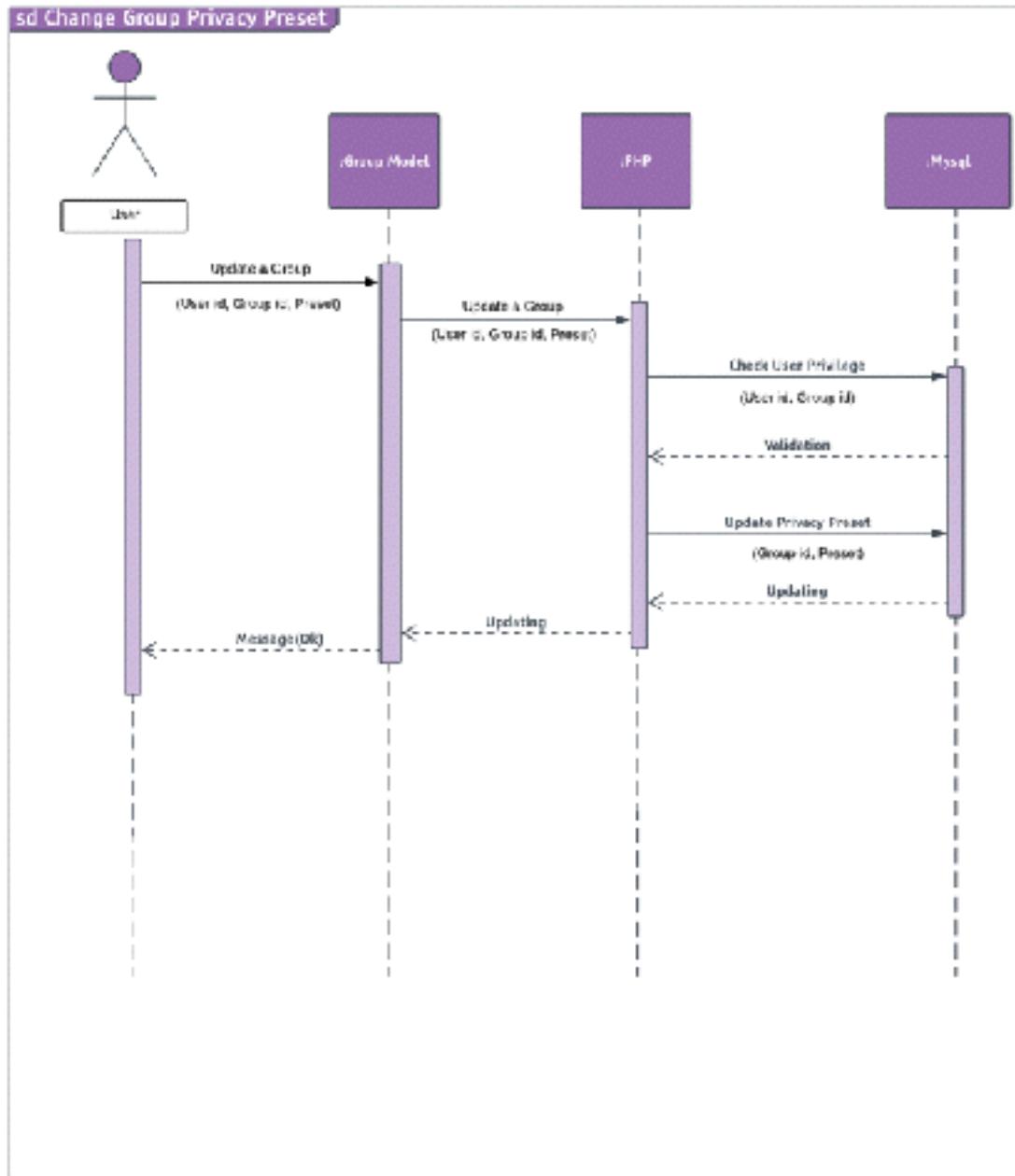


- React to an object a group- when allowed to do so.

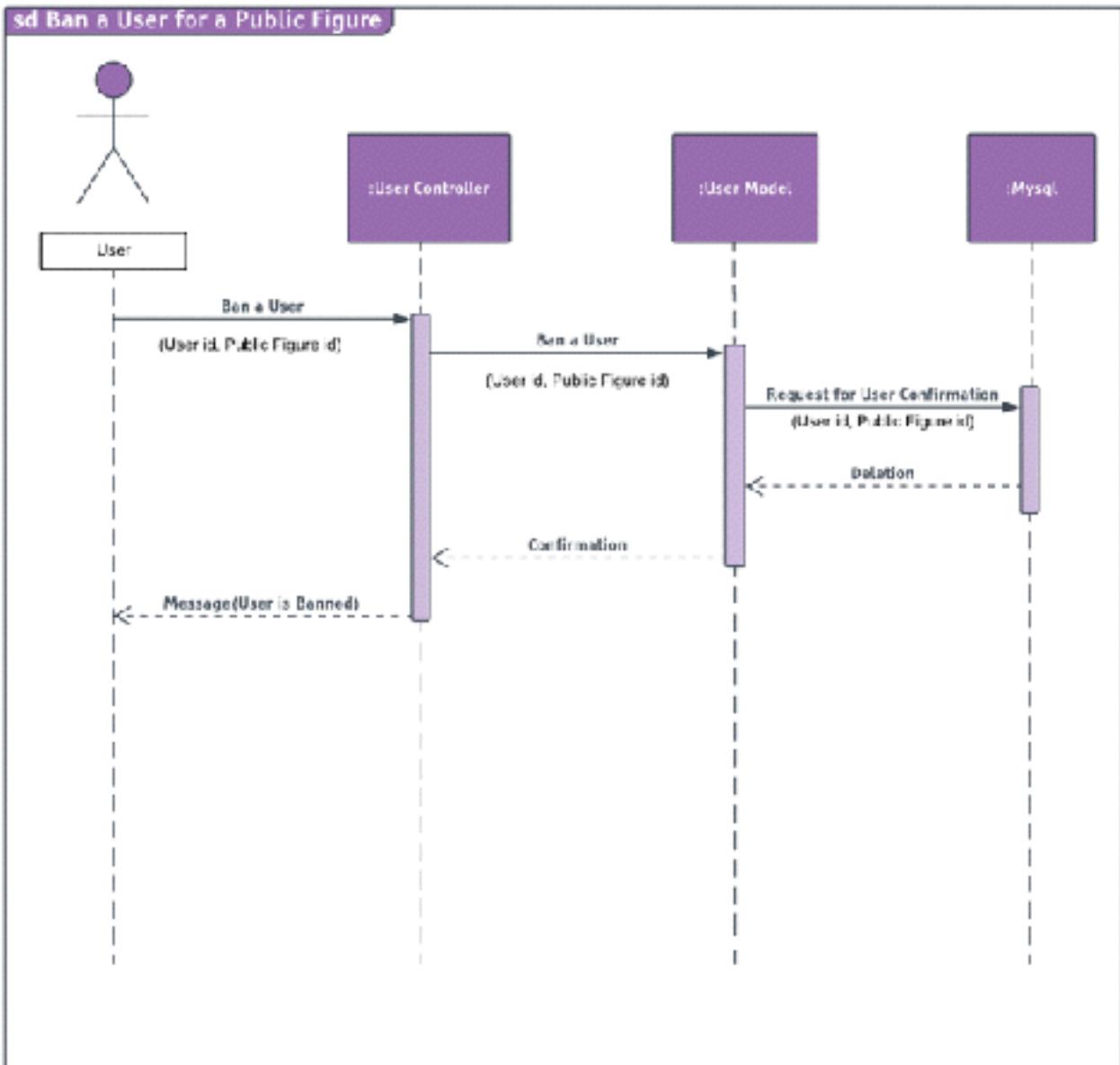


-
- Change user permission- when allowed to do so.

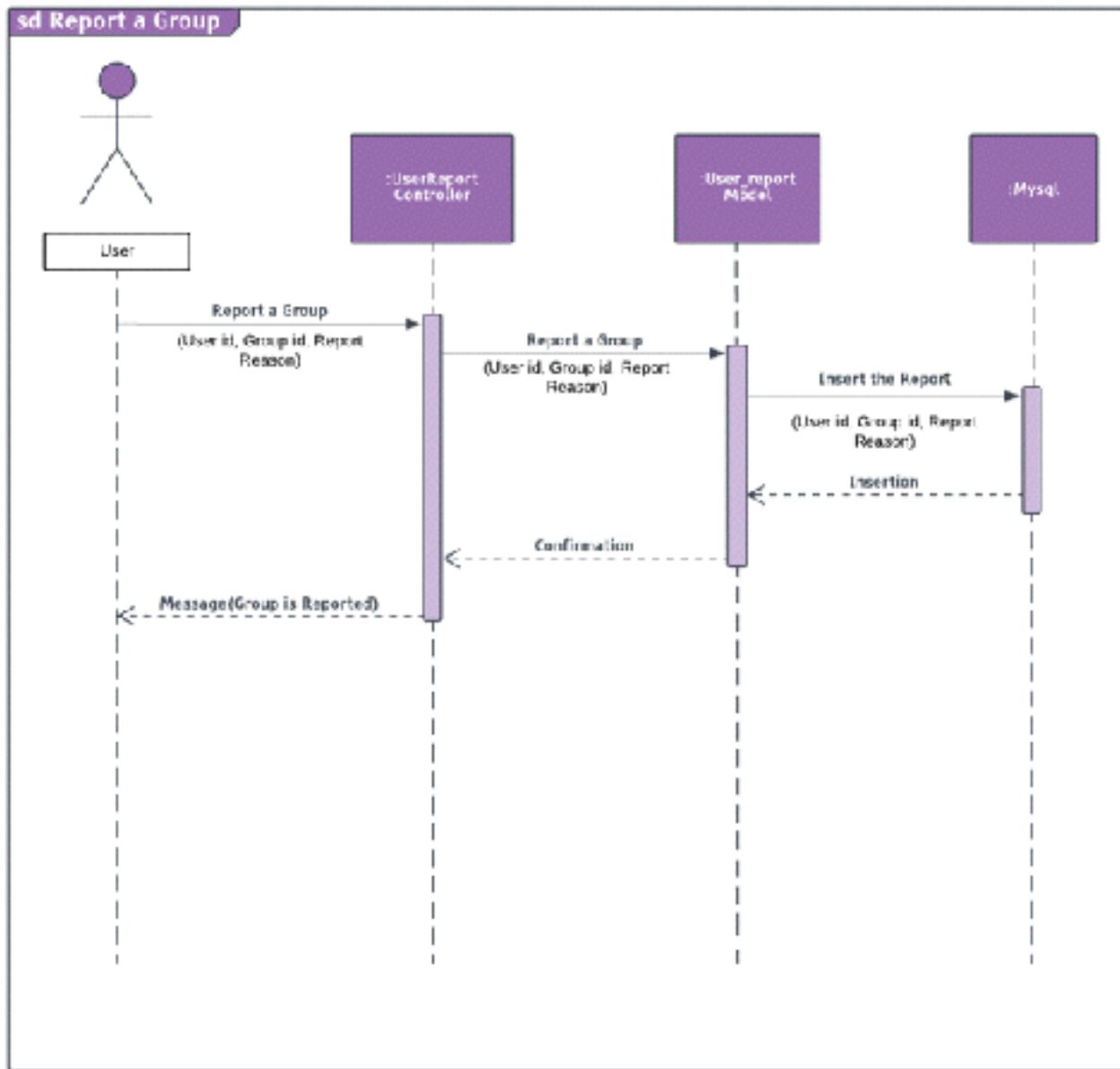
- Change the group privacy preset - when allowed to do so.



- Ban members of a group- when allowed to do so.



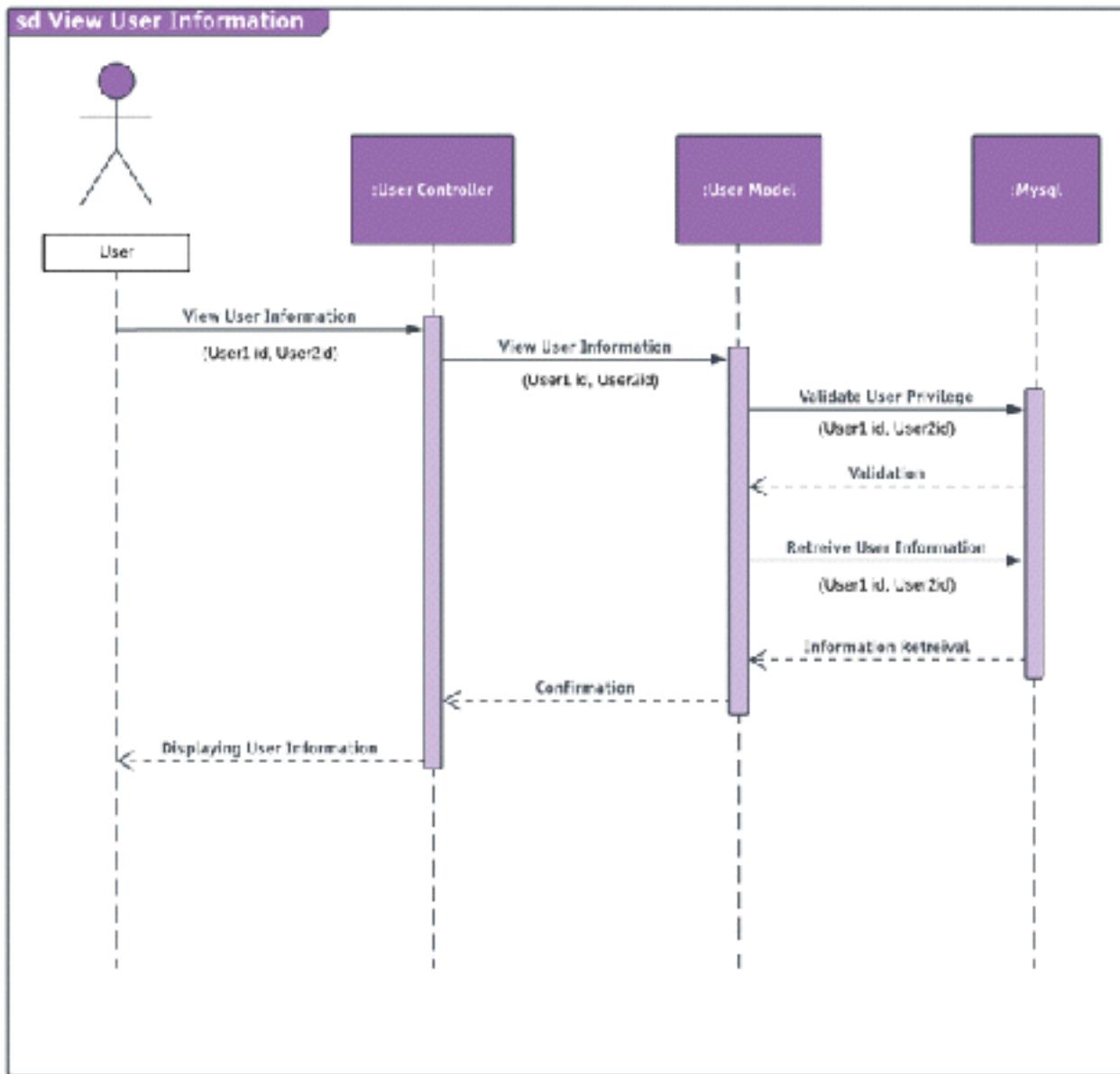
- Report a group- when allowed to do so.



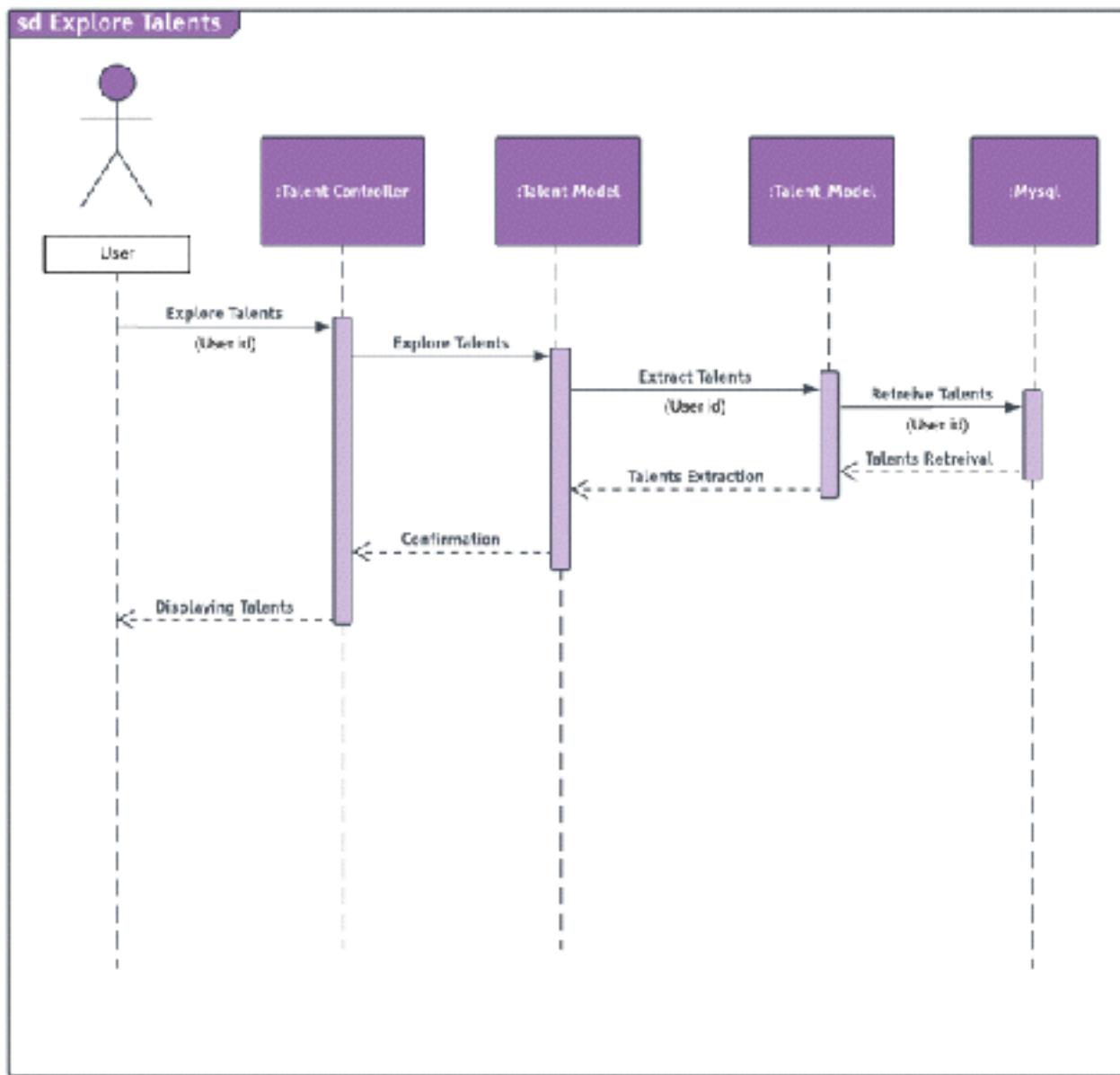
-
- Chat in a group

-
- Request for account verification - incase of public figure. //

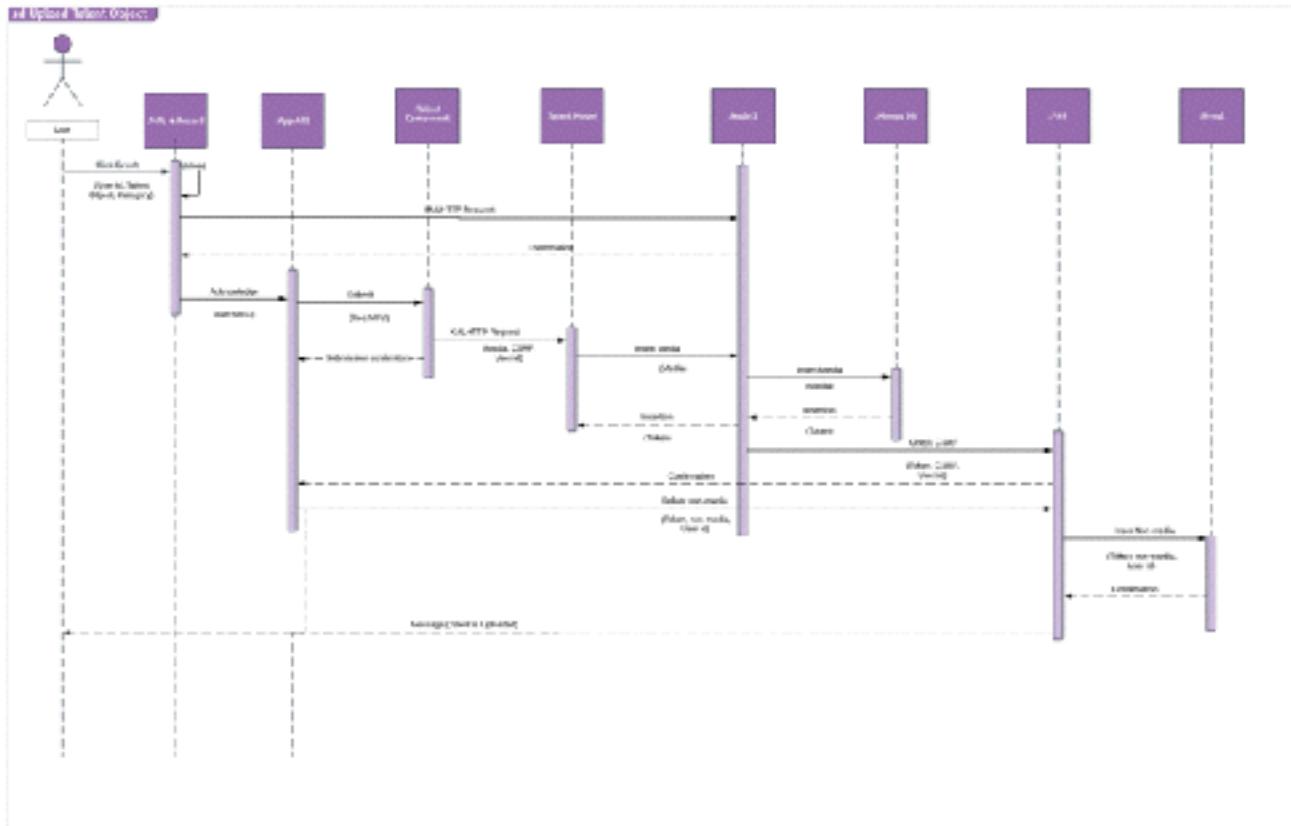
- View a user profile - when allowed to do so.



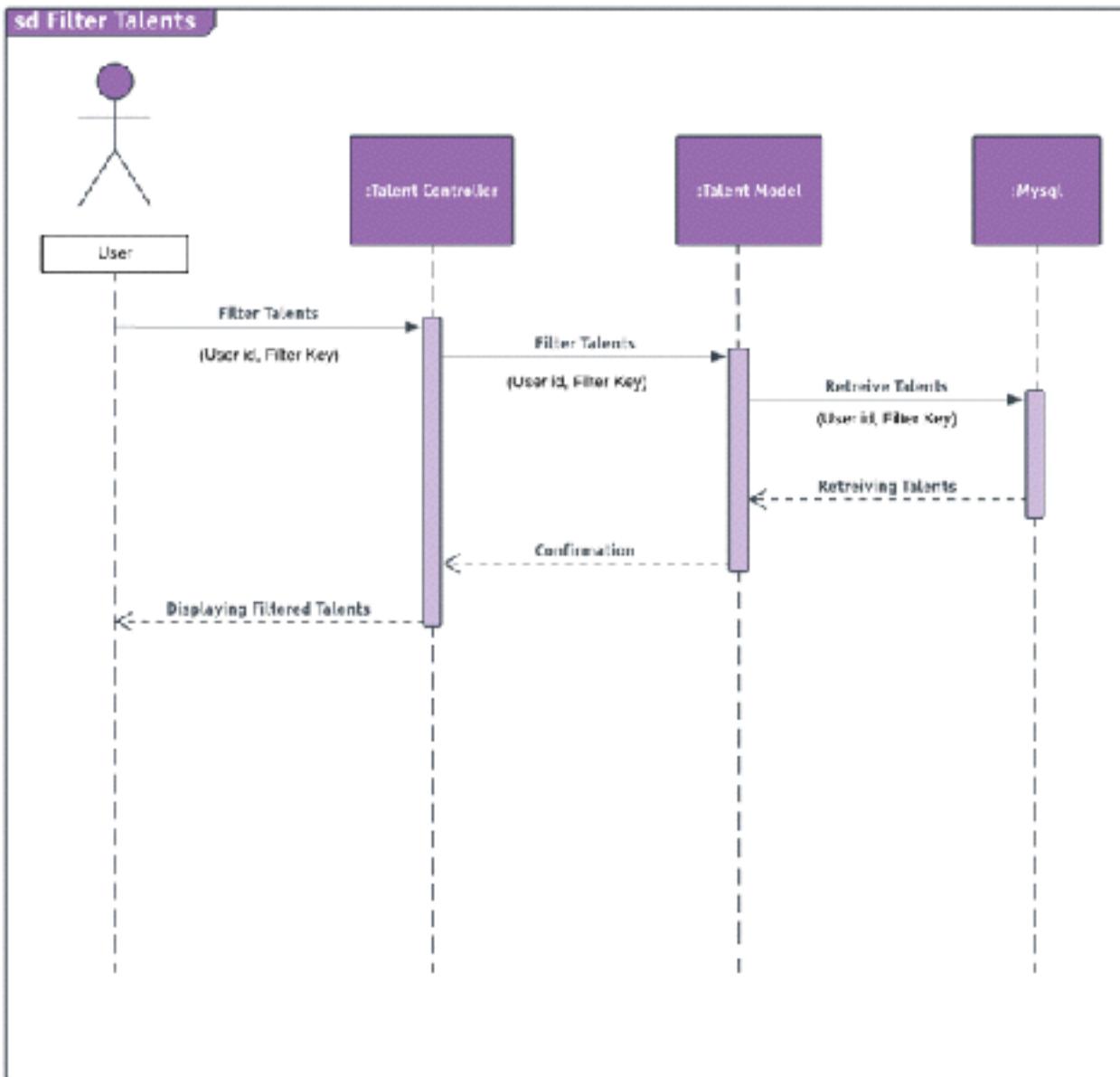
- Explore talents



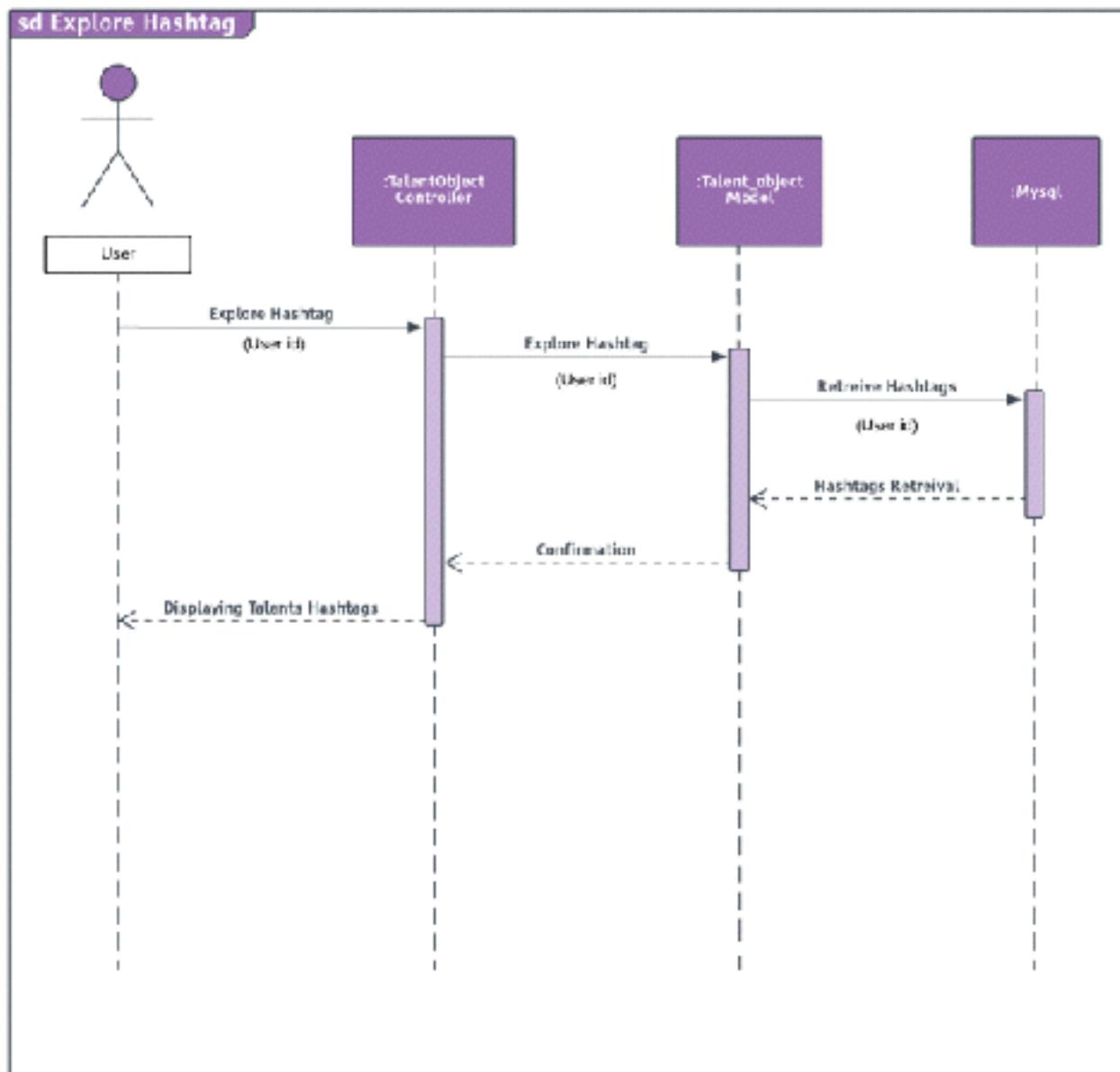
- Upload a talent object - when allowed to do so



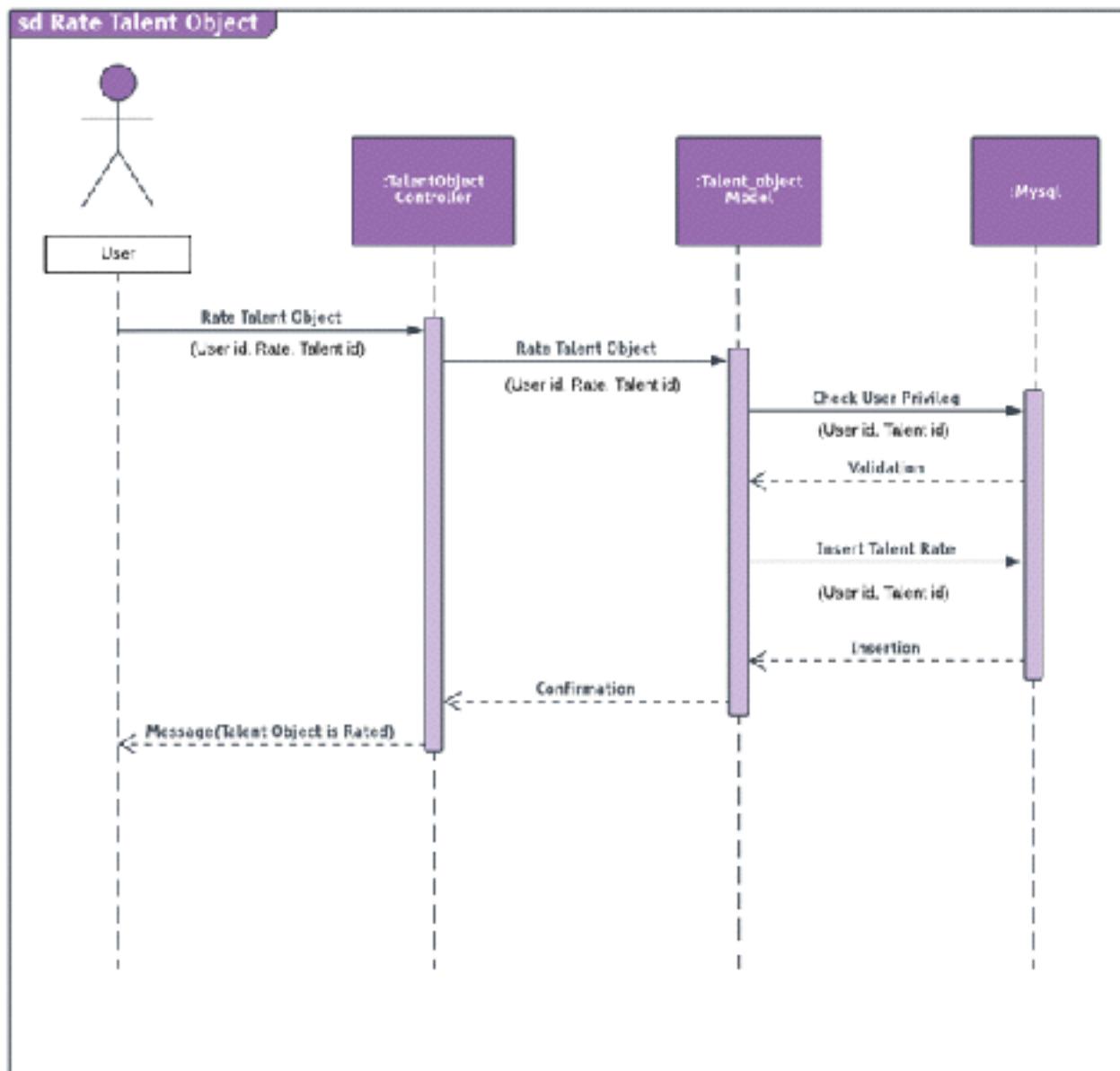
- Filter talents



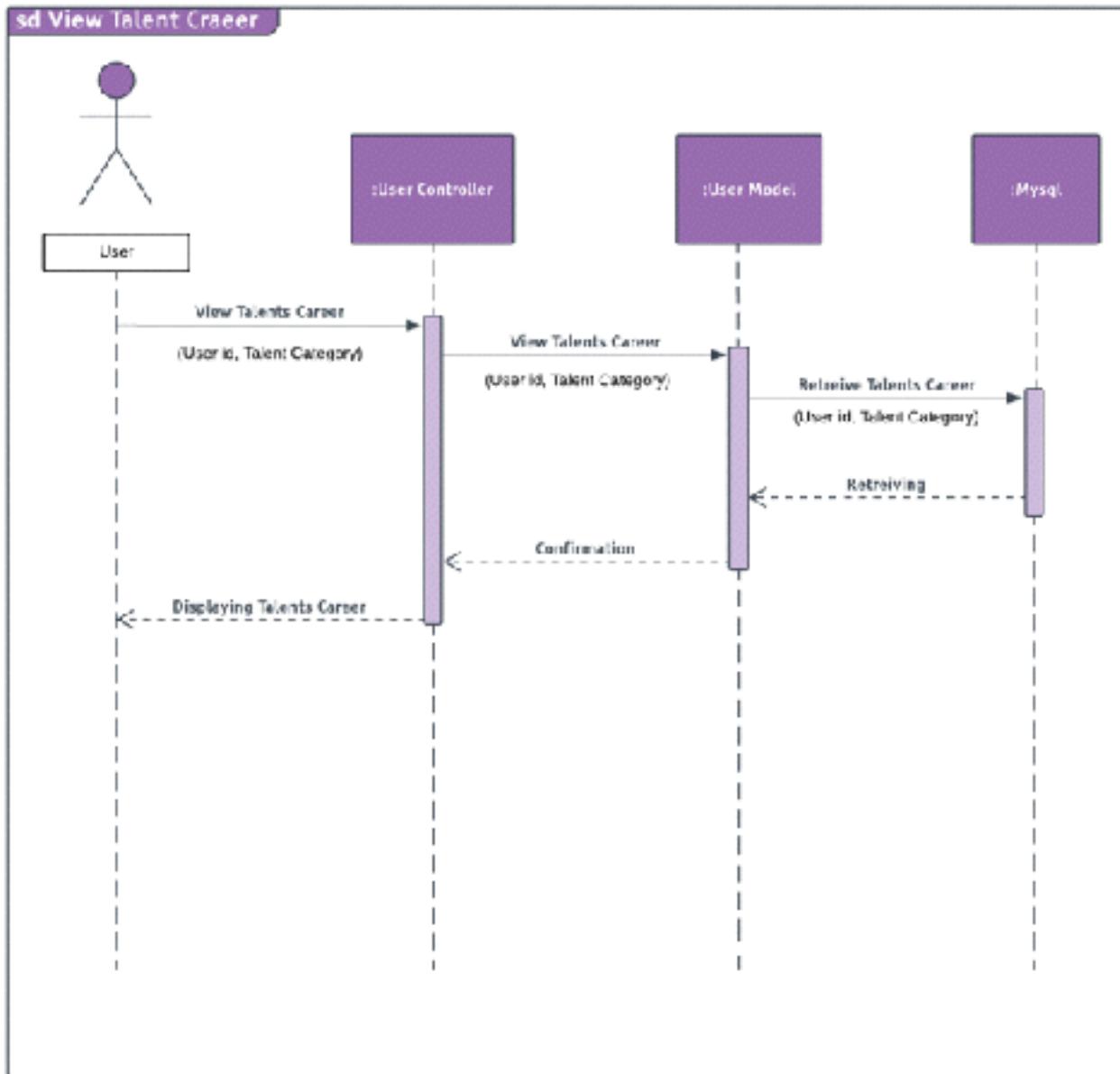
- Explore hashtags in talents section



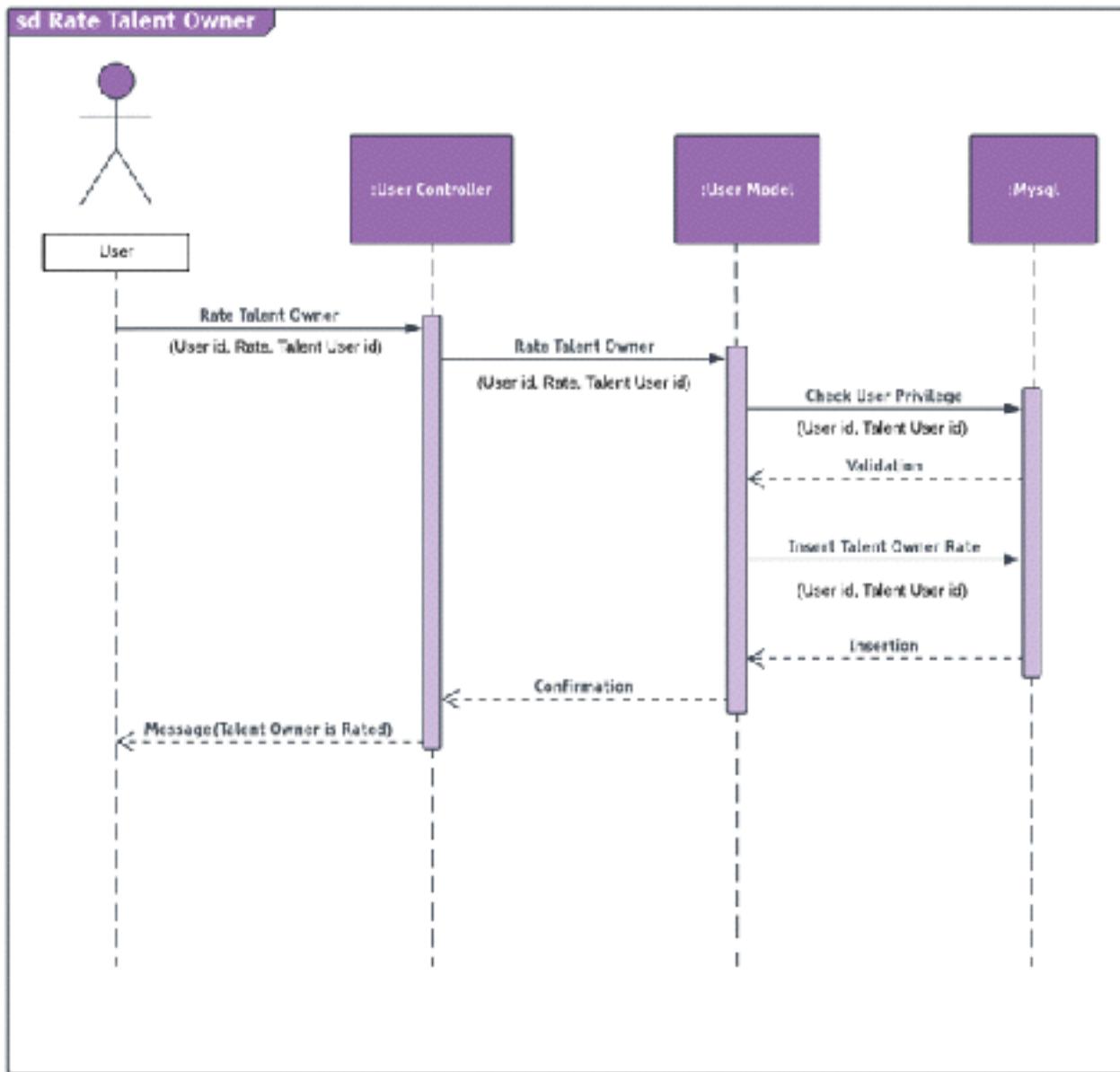
- Rate a talent object



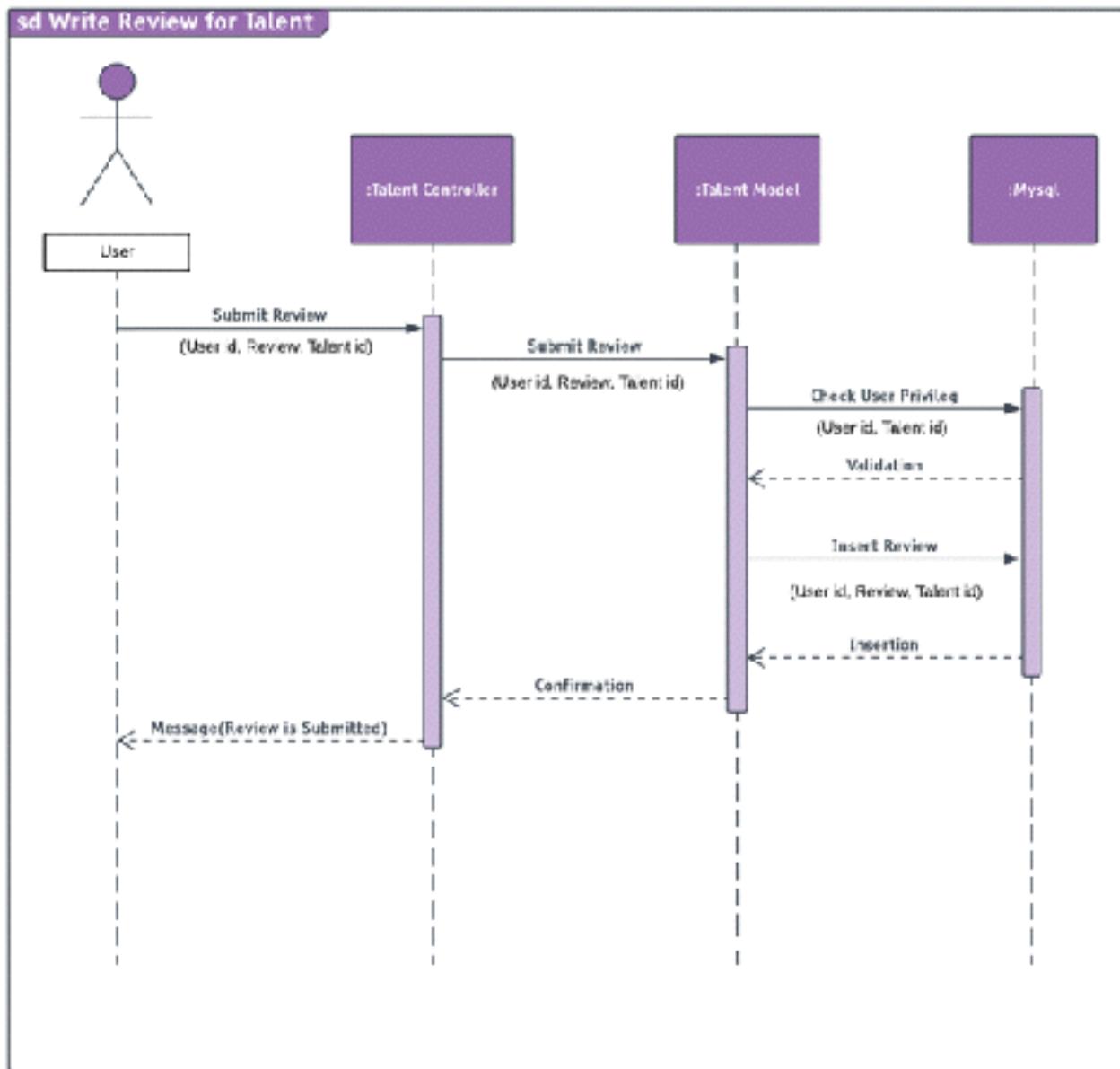
- View a talent career



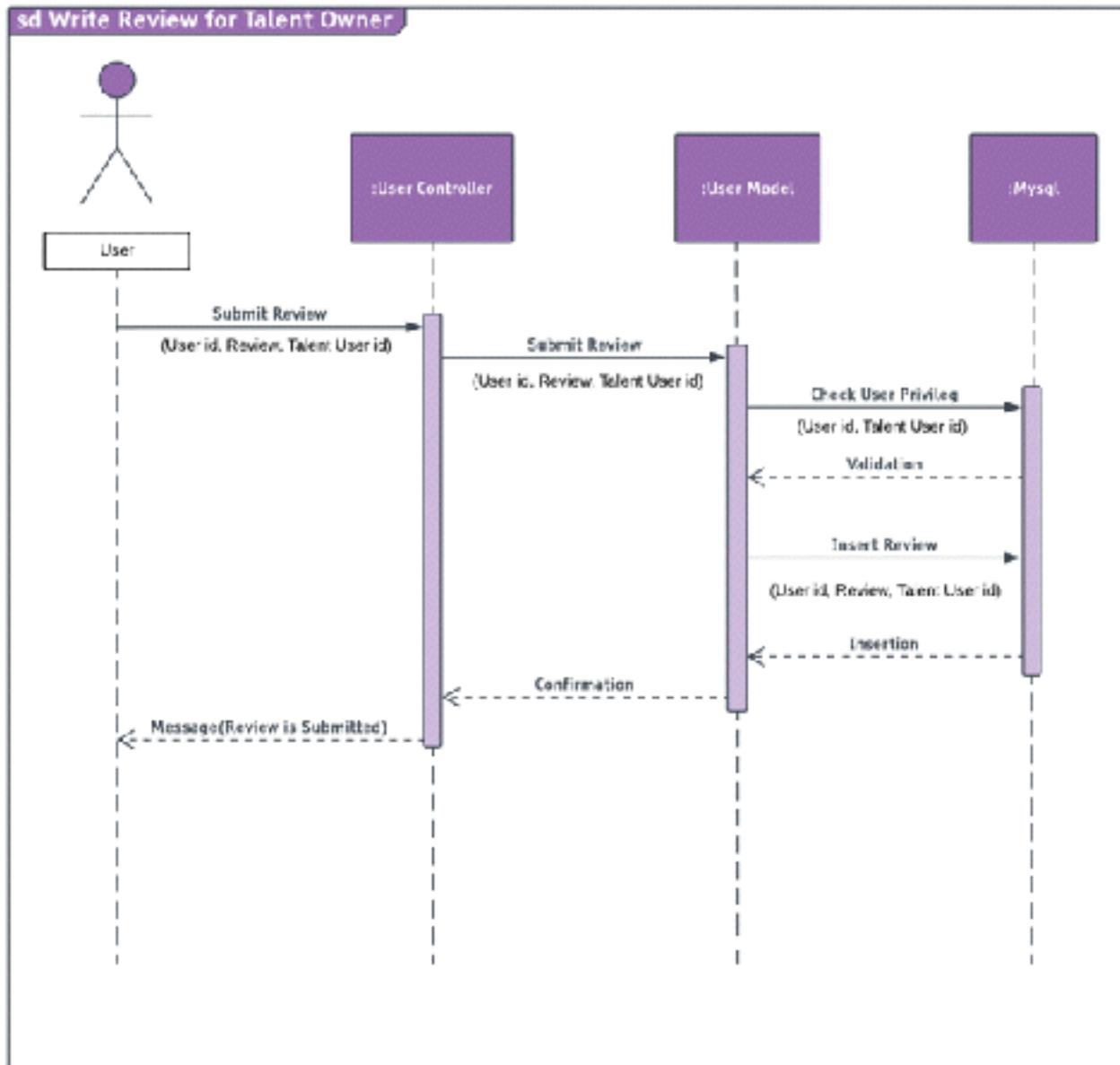
- Rate a talent owner



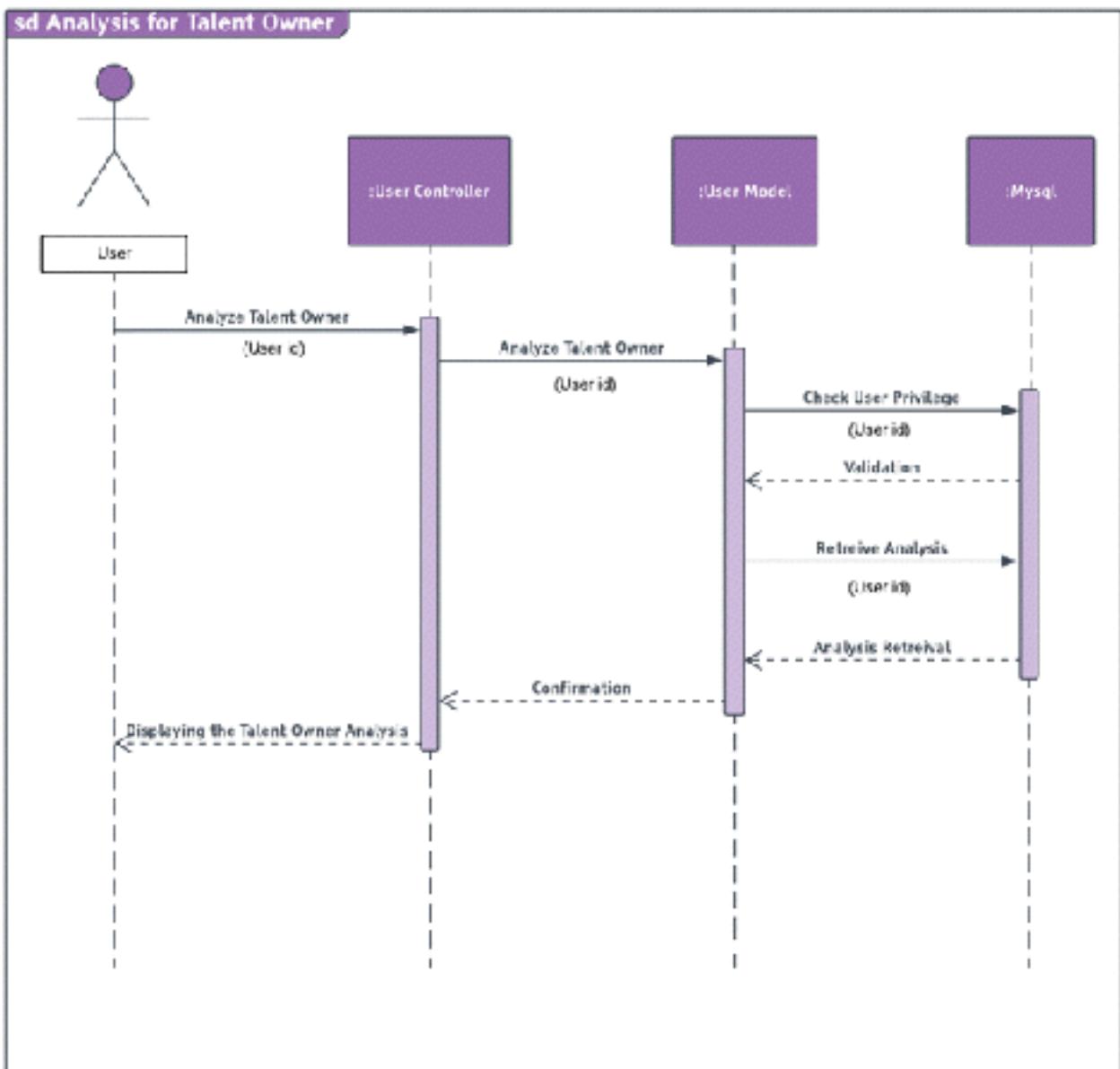
- Leave a review on a talent object



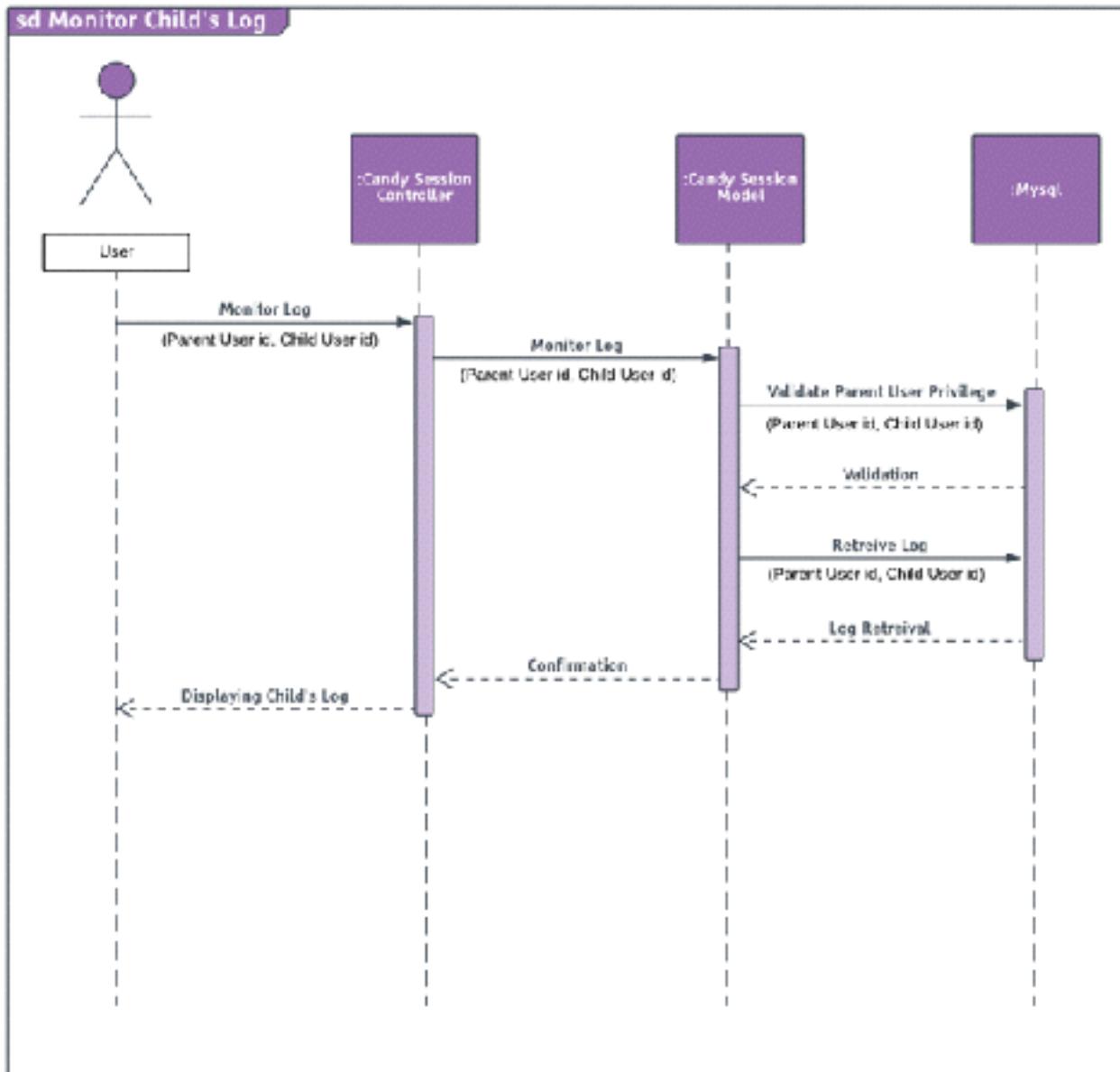
- Leave a review on a talent owner



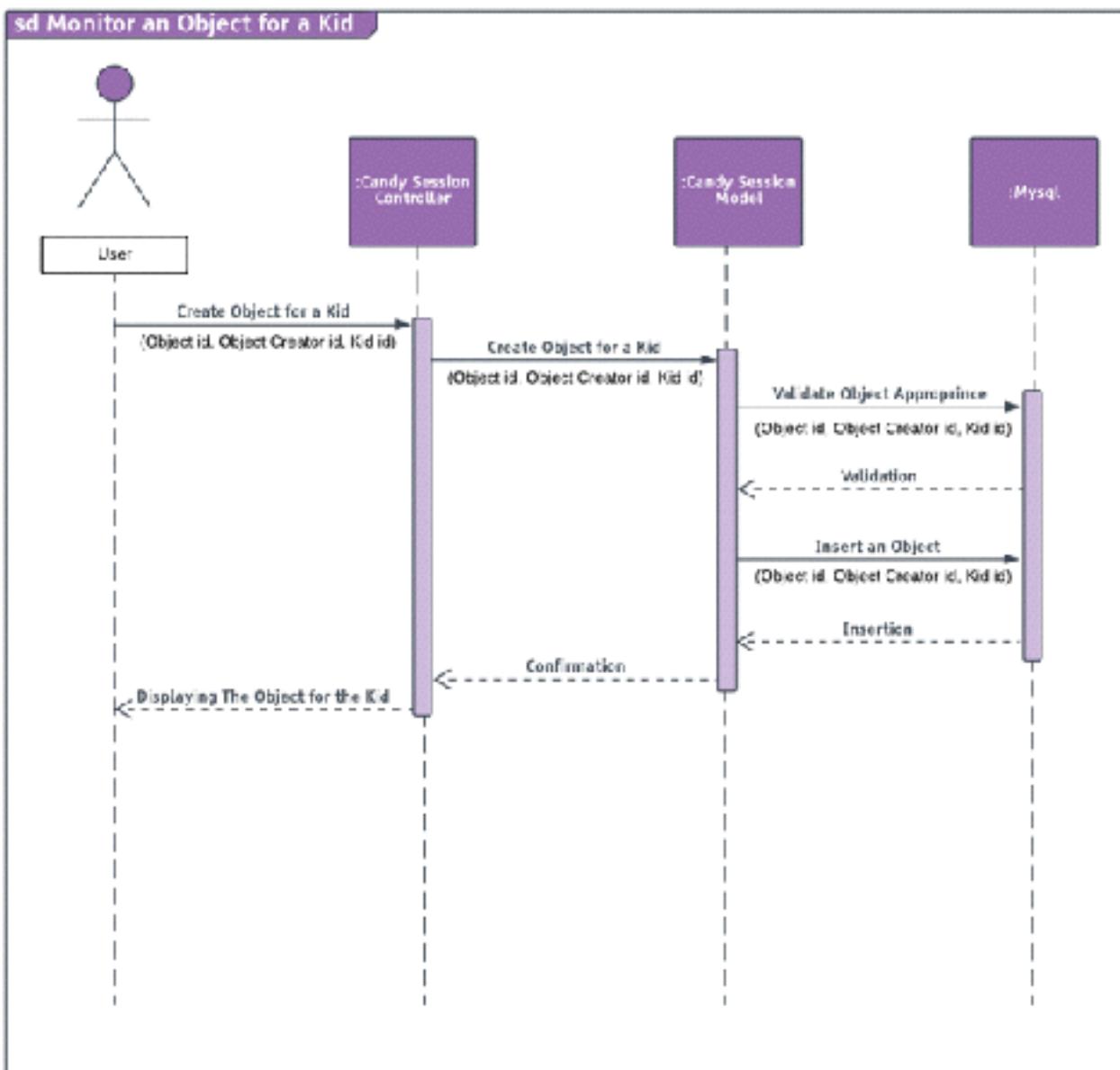
- View Analysis for the talent owner himself



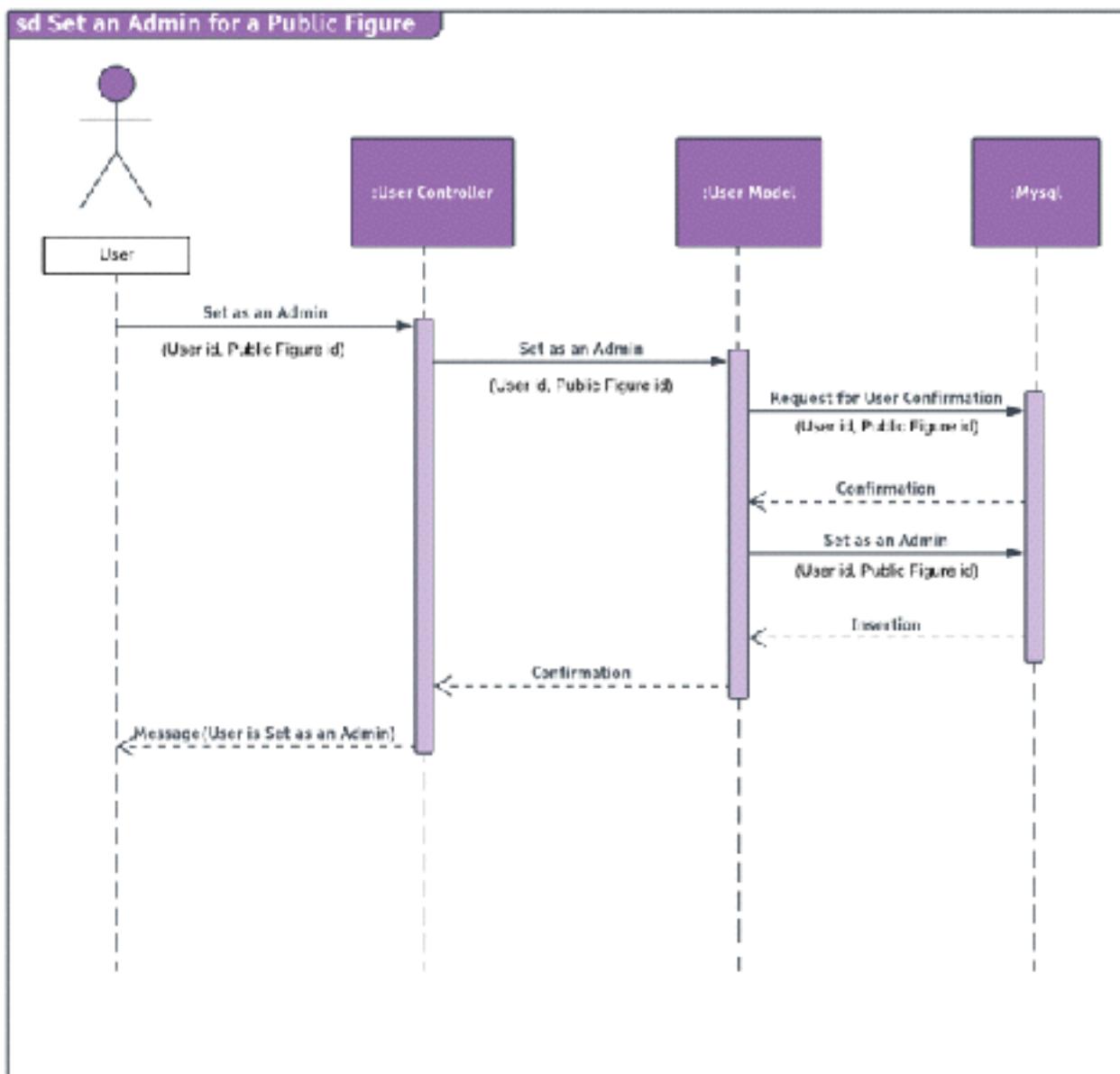
- Monitor for child log.



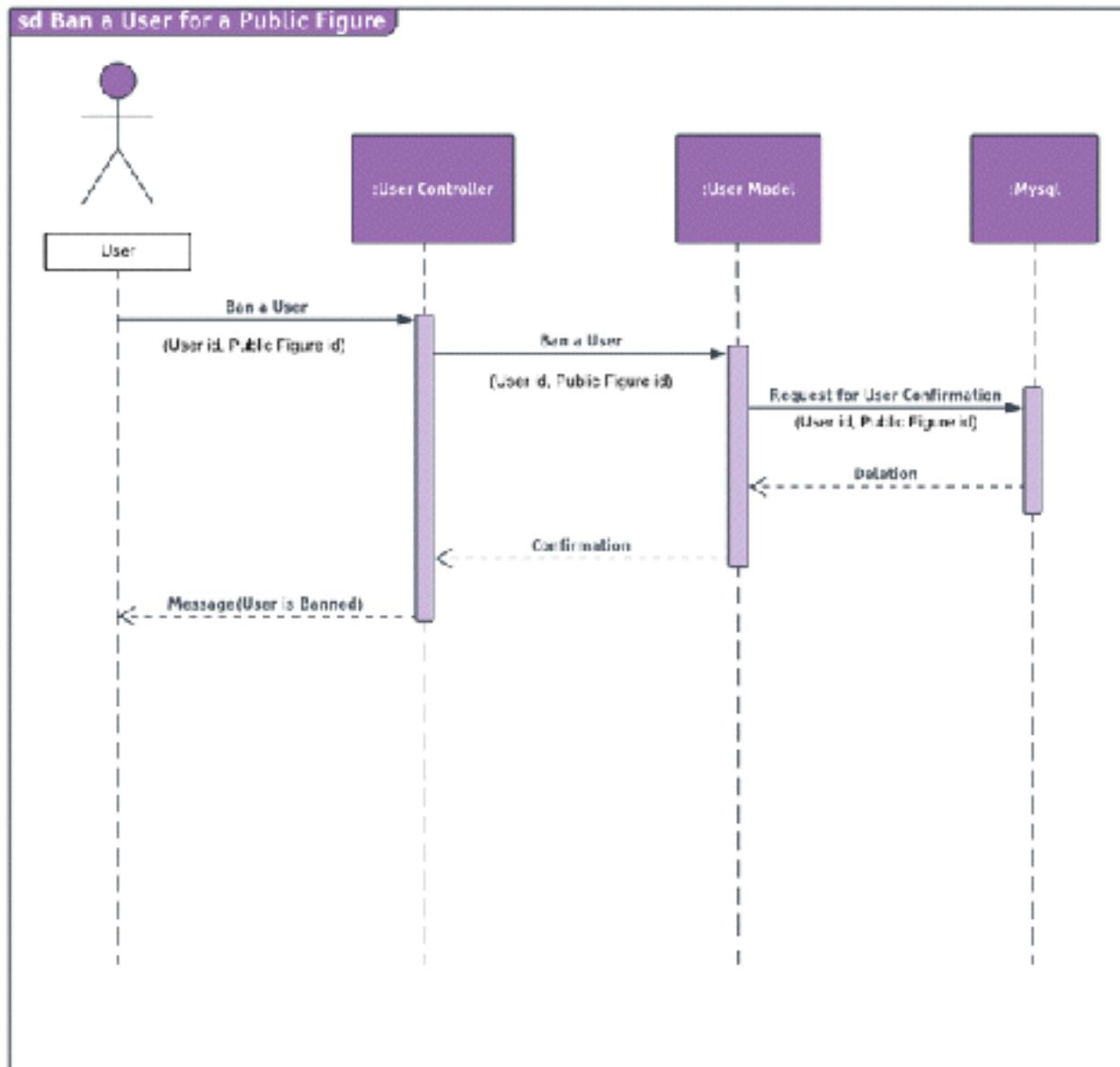
- Monitor objects for kid



- Set admins for public figure



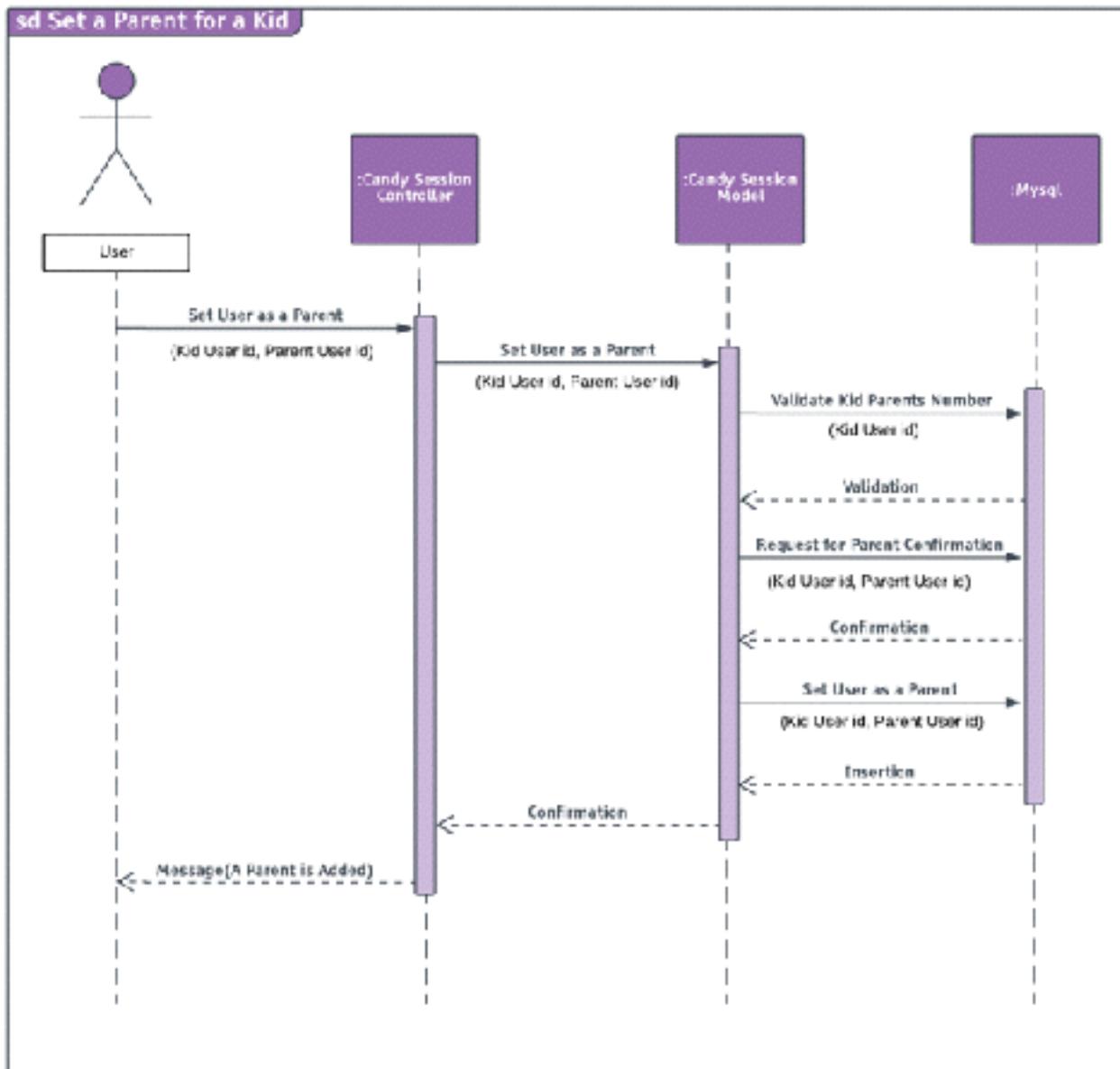
- Ban for public figure account



-
- measuring account trust //

-
- Recognize the user (Random Time camera). / /

- Set a parent for a child



5. Implementation

5.1.Controllers

5.1.1.Controllers List

AnswerController

BallonController

BlobController

CandySessionController

CareerController

CircleController

CircleMemberController

CommentController

Controller

CountryController

DevController

EducationController

EnvelopeController

FavUserController

GroupController

GroupMemberController

HashtagsController

HighEducationController

HobbyController

HomeController

IgnoreObjectController

JamCircleController

KnockController

LanguageController

MessageBusController

ObjController

PrivacyPresetController
PrivacySetCircleController
PrivacySetUserController
PublicFigureController
QuestionController
ReactionController
ReplyController
SavedObjectController
SavedPresetsController
SportController
StageController
StageMemberController
StaticMessageController
TalentCategoryController
TalentController
TalentObjectController
UserAddressController
UserBlocksController
UserCheckinsController
UserController
UserHashtagsController
UserKeywordsController
UserLogController
UserReportsController
UserRequestController

5.1.2.Controllers Specifications

Controller Name	Modifier Type	Method and Description
AnswerController	String	<ul style="list-style-type: none"> • create(Request \$req) • This method takes a req object as a parameter and it works as to create an instance of the answer
	Array	<ul style="list-style-type: none"> • userAnswers(Request \$req) • This method takes a req object as a parameter and it works as to get the answers of a specific user
	Array	<ul style="list-style-type: none"> • patch(Request \$req) • This method takes a req object as a parameter and it works as to get all the users answers for the analysis
BallonController	Object	<ul style="list-style-type: none"> • getUserNotification(Request \$request) • This method takes a req object as a parameter and it works as to get the notifications for a specific user
	Object	<ul style="list-style-type: none"> • getAllUserNotification(Request \$request) • This method takes a req object as a parameter and it works as to get all the notifications for a specific user
	Object	<ul style="list-style-type: none"> • getAllUserNotificationFr(Request \$request) • This method takes a req object as a parameter and it works as to get all the notifications of friend request for a specific user
	Object	<ul style="list-style-type: none"> • setToPoped(Request \$request) • This method takes a req object as a parameter and it works as to set the notification as popped before, so that it wont be notifying the user again.

Controller Name	Modifier Type	Method and Description
BlobController	String	<ul style="list-style-type: none"> • setToseen(Request \$request) • This method takes a req object as a parameter and it works as to set the notification as seen before, so that it won't be notifying the user again.
	Object	<ul style="list-style-type: none"> • createRecord(Request \$request) • This method takes a req object as a parameter and it works as to create an instance of a record
	Object	<ul style="list-style-type: none"> • retrieveRecord(Request \$request , \$id) • This method takes a req object and id as a parameter and it works as to retrieve a specific record
	Double	<ul style="list-style-type: none"> • retrieveRecordMeta(Request \$request) • This method takes a req object as a parameter and it works as to retrieve record meta data
	Array	<ul style="list-style-type: none"> • retrieveFileMeta(Request \$request) • This method takes a req object as a parameter and it works as to retrieve file meta data
	Integer	<ul style="list-style-type: none"> • uploadImage(Request \$request) • This method takes a req object as a parameter and it works as to upload an image
	Integer	<ul style="list-style-type: none"> • uploadFile(Request \$request) • This method takes a req object as a parameter and it works as to upload a file
	String	<ul style="list-style-type: none"> • uploadAvatar(Request \$request) • This method takes a req object as a parameter and it works as to upload an avatar

Controller Name	Modifier Type	Method and Description
	String	<ul style="list-style-type: none"> • uploadGroupPicture(Request \$request) • This method takes a req object as a parameter and it works as to upload group picture
	String	<ul style="list-style-type: none"> • uploadCover(Request \$request) • This method takes a req object as a parameter and it works as to upload cover picture
	Object	<ul style="list-style-type: none"> • retrieveImage(Request \$request , \$id) • This method takes a req object and an id as a parameter and it works as to retrieve a specific image
	Object	<ul style="list-style-type: none"> • retrieveFile(Request \$request , \$id) • This method takes a req object and an id as a parameter and it works as to retrieve a specific file
	Object	<ul style="list-style-type: none"> • retrieveAvatar(Request \$request , \$id) • This method takes a req object and an id as a parameter and it works as to retrieve a specific avatar
	Object	<ul style="list-style-type: none"> • retrieveGroupPicture(Request \$request , \$id) • This method takes a req object and an id as a parameter and it works as to retrieve a specific group picture
	Object	<ul style="list-style-type: none"> • retrieveGroupCompressed(Request \$request , \$id) • This method takes a req object and an id as a parameter and it works as to retrieve a specific group picture but compressed

Controller Name	Modifier Type	Method and Description
CandySessionController	Object	<ul style="list-style-type: none"> • retrieveAvatarCompressed(Request \$request , \$id) • This method takes a req object and an id as a parameter and it works as to retrieve a specific avatar but compressed
	Object	<ul style="list-style-type: none"> • retrieveCover(Request \$request , \$id) • This method takes a req object and an id as a parameter and it works as to retrieve a specific cover
	Object	<ul style="list-style-type: none"> • retrieveCoverCompressed(Request \$request , \$id) • This method takes a req object and an id as a parameter and it works as to retrieve a specific cover but compressed
	String	<ul style="list-style-type: none"> • quote(Request \$request) • This method takes a req object as a parameter and it works as to insert the picture quote
	String	<ul style="list-style-type: none"> • imageStates(Request \$req) • This method takes a req object as a parameter and it works as to get the initials o the image
	String	<ul style="list-style-type: none"> • imageComments(Request \$req) • This method takes a req object as a parameter and it works as to get the comments of the image
CareerController	Integer	<ul style="list-style-type: none"> • createCareer(Request \$request) • This method takes a req object as a parameter and it works as to create an instance of a career
	Array	<ul style="list-style-type: none"> • retrieveCareer(Request \$request) • This method takes a req object and id as a parameter and it works as to retrieve a specific career

Controller Name	Modifier Type	Method and Description
CircleController	String	<ul style="list-style-type: none"> • updateCareer(Request \$request) • This method takes a req object as a parameter and it works as to update career
	String	<ul style="list-style-type: none"> • deleteCareer(Request \$request) • This method takes a req object as a parameter and it works as to delete career
	String	<ul style="list-style-type: none"> • create (Request \$request) • This method takes a req object as a parameter and it works as to create new circle
	Object	<ul style="list-style-type: none"> • view (Request \$request) • This method takes a req object as a parameter and it works as to view a circle
	Object	<ul style="list-style-type: none"> • retrieve(Request \$request) • This method takes a req object as a parameter and it works as to retrieve circles
	Object	<ul style="list-style-type: none"> • search(Request \$request) • This method takes a req object as a parameter and it works as to search for circles
	Object	<ul style="list-style-type: none"> • check(Request \$request) • This method takes a req object as a parameter and it works as to check if a specified circle exists or not
CircleMemberController	String	<ul style="list-style-type: none"> • deleteCircle(Request \$request) • This method takes a req object as a parameter and it works as to delete a specific circle.
	String	<ul style="list-style-type: none"> • acceptGroup(Request \$request) • This method takes a req object as a parameter and it works as to accept group of members in a circle.

Controller Name	Modifier Type	Method and Description
CircleController	Object	<ul style="list-style-type: none"> groupPushMembers(Request \$request) This method takes a req object as a parameter and it works as to get circle members
	String	<ul style="list-style-type: none"> unpairFriends(Request \$request) This method takes a req object as a parameter and it works as to delete a specific friend from all circles
	String	<ul style="list-style-type: none"> addMember(Request \$req) This method takes a req object as a parameter and it works as to add circle members
	String	<ul style="list-style-type: none"> removeMember(Request \$req) This method takes a req object as a parameter and it works as to remove friend from a circle
CommentController	String	<ul style="list-style-type: none"> create(Request \$request) This method takes a req object as a parameter and it works as to create comment instance
	String	<ul style="list-style-type: none"> retrieve(Request \$request) This method takes a req object as a parameter and it works as to retrieve comments
	Array	<ul style="list-style-type: none"> getComments(Request \$request) This method takes a req object as a parameter and it works as to get all comments
	Array	<ul style="list-style-type: none"> getReplies(Request \$request) This method takes a req object as a parameter and it works as to get comment replies
CountryController		

Controller Name	Modifier Type	Method and Description
DevController	String	<ul style="list-style-type: none"> • resetKnocks(Request \$request) • This method takes a req object as a parameter and it works as to Truncate all the Knocks, which means deleting its childs too, this will truncate all of Ballons, Reactions, Blobs, Comments and Knocks tables.
	String	<ul style="list-style-type: none"> • resetUsers(Request \$request) • This method takes a req object as a parameter and it works as to Truncate all the Users, which means deleting its childs too, this will truncate all of Ballons, Reactions, Blobs, Comments, Knocks, Circles, Circle Members, Privacy Circle Sets, Privacy User Sets, Objects and Users tables.
	String	<ul style="list-style-type: none"> • reinstall(Request \$request) • This method takes a req object as a parameter and it works as to If you have a fresh install or migration for the Database, so this will rebound the initial data for the whole App , You will need to reinsert the Static Messages manually from the database but still you cant if you don't have this kind of data
	String	<ul style="list-style-type: none"> • removeAllFriends(Request \$request) • This method takes a req object as a parameter and it works as to remove all friends from a specific circle
	String	<ul style="list-style-type: none"> • createRandomKnocks(Request \$request) • This method takes a req object as a parameter and it works as to create random knocks for testing purposes.

Controller Name	Modifier Type	Method and Description
EducationController	String	<ul style="list-style-type: none"> • <code>createRandomSocial(Request \$request)</code> • This method takes a req object as a parameter and it works as to create random knocks for testing purposes.
	String	<ul style="list-style-type: none"> • <code>createRandomReactions(Request \$request)</code> • This method takes a req object as a parameter and it works as to create random reactions for testing purposes.
	String	<ul style="list-style-type: none"> • <code>createRandomGroups(Request \$request)</code> • This method takes a req object as a parameter and it works as to create random groups for testing purposes.
	String	<ul style="list-style-type: none"> • <code>createRandomCircles(Request \$request)</code> • This method takes a req object as a parameter and it works as to create random circles for testing purposes.
	Array	<ul style="list-style-type: none"> • <code>watchMembershipPublications()</code> • This method takes a req object as a parameter and it works as to check and remove duplications for testing purposes.
	String	<ul style="list-style-type: none"> • <code>addRandomEntry(Request \$request)</code> • This method takes a req object as a parameter and it works as to add random entries for personal information for testing purposes.
	Integer	<ul style="list-style-type: none"> • <code>createEducation(Request \$request)</code> • This method takes a req object as a parameter and it works as to create an education instance

Controller Name	Modifier Type	Method and Description
	Array	<ul style="list-style-type: none"> • <code>retrieveEducation(Request \$request)</code> • This method takes a req object as a parameter and it works as to retrieve education objects
	String	<ul style="list-style-type: none"> • <code>updateEducation(Request \$request)</code> • This method takes a req object as a parameter and it works as to update education objects
	String	<ul style="list-style-type: none"> • <code>deleteEducation(Request \$request)</code> • This method takes a req object as a parameter and it works as to delete education objects
EnvelopeController		
FavUserController		
GroupController	String	<ul style="list-style-type: none"> • <code>createGroup(Request \$request)</code> • This method takes a req object as a parameter and it works as to create a group
	String	<ul style="list-style-type: none"> • <code>joinPublicGroup(Request \$request)</code> • This method takes a req object as a parameter and it works as to enable to join a public group
	String	<ul style="list-style-type: none"> • <code>joinClosedGroup(Request \$request)</code> • This method takes a req object as a parameter and it works as to enable to join a closed group
	String	<ul style="list-style-type: none"> • <code>addMemberPublicGroup(Request \$request)</code> • This method takes a req object as a parameter and it works as to process the request of adding a member to a public group

Controller Name	Modifier Type	Method and Description
	Array	<ul style="list-style-type: none"> • <code>getGroups(Request \$request)</code> • This method takes a req object as a parameter and it works as to get all the groups
	String	<ul style="list-style-type: none"> • <code>getGroupName(Request \$request)</code> • This method takes a req object as a parameter and it works as to get all the group name
	Integer	<ul style="list-style-type: none"> • <code>routeToGroup(Request \$request)</code> • This method takes a req object as a parameter and it works as to route for a specific group
	Object	<ul style="list-style-type: none"> • <code>routeToGroupPictures(Request \$request)</code> • This method takes a req object as a parameter and it works as to route for a group pictures
	Object	<ul style="list-style-type: none"> • <code>routeToGroupFiles(Request \$request)</code> • This method takes a req object as a parameter and it works as to route for a group files
	Object	<ul style="list-style-type: none"> • <code>routeToGroupVoices(Request \$request)</code> • This method takes a req object as a parameter and it works as to route for a group voices
	Object	<ul style="list-style-type: none"> • <code>routeToGroupVideos(Request \$request)</code> • This method takes a req object as a parameter and it works as to route for a group videos
	Object	<ul style="list-style-type: none"> • <code>routeToGroupSettings(Request \$request)</code> • This method takes a req object as a parameter and it works as to route for a group settings

Controller Name	Modifier Type	Method and Description
	Integer	<ul style="list-style-type: none"> • retrieveGroupKnocks(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the group knocks
	Integer	<ul style="list-style-type: none"> • retrieveOlderGroupKnocks(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the old group knocks
		retrieveNewerGroupKnocks(Request \$request)
	Object	<ul style="list-style-type: none"> • retrieveGroupForJoin(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the groups in order to join it.
	Object	<ul style="list-style-type: none"> • getPictures(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the group's pictures
	Object	<ul style="list-style-type: none"> • getFiles(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the group's files
	Object	<ul style="list-style-type: none"> • getVoices(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the group's records
	Object	<ul style="list-style-type: none"> • getVideos(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the group's videos
	String	<ul style="list-style-type: none"> • updateGroupInfo(Request \$request) • This method takes a req object as a parameter and it works as to update the information for a specific group

Controller Name	Modifier Type	Method and Description
GroupMemberController	String	<ul style="list-style-type: none"> updateGroupPrivacy(Request \$request) This method takes a req object as a parameter and it works as to update the privacy for a specific group
	String	<ul style="list-style-type: none"> deleteGroup(Request \$request) This method takes a req object as a parameter and it works as to delete a specific group
	Object	<ul style="list-style-type: none"> getGroupMembers(Request \$request) This method takes a req object as a parameter and it works as to get the members of a specific group
		checkUserInGroup(Request \$request)
	String	<ul style="list-style-type: none"> checkOwner(Request \$request) This method takes a req object as a parameter and it works as to check if a specific user is the owner of the group or not
	String	<ul style="list-style-type: none"> checkAdmin(Request \$request) This method takes a req object as a parameter and it works as to check if a specific user is the admin of the group or not
	String	<ul style="list-style-type: none"> removeMember(Request \$request) This method takes a req object as a parameter and it works as to remove a member from the group
	Object	<ul style="list-style-type: none"> getMembersPosition(Request \$request) This method takes a req object as a parameter and it works as to get the position of the members in the group

Controller Name	Modifier Type	Method and Description
HashtagsController	String	<ul style="list-style-type: none"> • setMembersToAdmin(Request \$request) • This method takes a req object as a parameter and it works as to set a member as an admin
	String	<ul style="list-style-type: none"> • setAdminToMember(Request \$request) • This method takes a req object as a parameter and it works as to set admin to a normal group member
	String	<ul style="list-style-type: none"> • setToOwner(Request \$request) • This method takes a req object as a parameter and it works as to set specific user as an owner
	Object	<ul style="list-style-type: none"> • lazy(Request \$req) • This method takes a req object as a parameter and it works as to retrieve all the hashtags that matches the given hashtag
	Array	<ul style="list-style-type: none"> • getRecentHashtags(Request \$request) • This method takes a req object as a parameter and it works as to retrieve all the recent hashtags.
	Object	<ul style="list-style-type: none"> • findHashTag(Request \$req, \$hashtag) • This method takes a req object and the hashtag as a parameter and it works as to retrieve specific hashtag
	Object	<ul style="list-style-type: none"> • retrieveTrendKnocks(Request \$request) • This method takes a req object as a parameter and it works as to retrieve trend knocks
	Object	<ul style="list-style-type: none"> • retrieveOlderTrendKnocks(Request \$request) • This method takes a req object as a parameter and it works as to retrieve older trend knocks

Controller Name	Modifier Type	Method and Description
HighEducationController	Object	<ul style="list-style-type: none"> • retrieveNewerTrendKnocks(Request \$request) • This method takes a req object as a parameter and it works as to retrieve Newer trend knocks
	Integer	<ul style="list-style-type: none"> • createHighEducation(Request \$request) • This method takes a req object as a parameter and it works as to create a high education instance
	Array	<ul style="list-style-type: none"> • retrieveHighEducation(Request \$request) • This method takes a req object as a parameter and it works as to retrieve high education objects
	String	<ul style="list-style-type: none"> • updateHighEducation(Request \$request) • This method takes a req object as a parameter and it works as to update high education objects
HobbyController	String	<ul style="list-style-type: none"> • deleteHighEducation(Request \$request) • This method takes a req object as a parameter and it works as to delete high education objects
	Integer	<ul style="list-style-type: none"> • createHobby(Request \$request) • This method takes a req object as a parameter and it works as to create a hobby instance
	Array	<ul style="list-style-type: none"> • retrieveHobby(Request \$request) • This method takes a req object as a parameter and it works as to retrieve hobby objects
	String	<ul style="list-style-type: none"> • updateHobby(Request \$request) • This method takes a req object as a parameter and it works as to update hobby objects
	String	<ul style="list-style-type: none"> • deleteHobby(Request \$request) • This method takes a req object as a parameter and it works as to delete hobby objects

Controller Name	Modifier Type	Method and Description
		hobbies(Request \$req)
HomeController	View	<ul style="list-style-type: none"> • index() • This method returns the home view
IgnoreObjectController		
JamCircleController		
KnockController	String	<ul style="list-style-type: none"> • create(Request \$request) • This method takes a req object as a parameter and it works as to create a new knock
	String	<ul style="list-style-type: none"> • delete(Request \$request) • This method takes a req object as a parameter and it works as to delete a knock
	Object	<ul style="list-style-type: none"> • retrieve(Request \$request) • This method takes a req object as a parameter and it works as to receive the knocks
	Object	<ul style="list-style-type: none"> • retrieveOlder(Request \$request) • This method takes a req object as a parameter and it works as to receive the older knocks
	Array	<ul style="list-style-type: none"> • getComments(Request \$request) • This method takes a req object as a parameter and it works as to get comments of a knock
	String	<ul style="list-style-type: none"> • tickSeen(Request \$request) • This method takes a req object as a parameter and it works as to indicate that the knock is seen
	View	<ul style="list-style-type: none"> • viewKnock(Request \$request , \$knock) • This method takes a req object as a parameter and it works as to view the knocks

Controller Name	Modifier Type	Method and Description
KnockController	Array	<ul style="list-style-type: none"> • knockMasterData(Request \$request) • This method is taking the eq object as a parameter to work as to retrieve the initial knock data
	View	<ul style="list-style-type: none"> • viewReply(Request \$request, \$reply) • This method takes a req object and reply as a parameter and it works as to view reply of the knock
	View	<ul style="list-style-type: none"> • viewComment(Request \$request, \$comment) • This method takes a req object and comment as a parameter and it works as to view comment of the knock
	View	<ul style="list-style-type: none"> • viewKnockWithComment(Request \$request, \$knock , \$comment) • This method takes a req object and knock and comment as a parameter and it works as to view knock and the comment of the knock
LanguageController	String	<ul style="list-style-type: none"> • create(Request \$request) • This method takes a req object as a parameter and it works as to create a new language instance
	Object	<ul style="list-style-type: none"> • collect(Request \$request) • This method takes a req object as a parameter and it works as to collect and retrieve all the languages.
MessageBusController		
ObjController	String	<ul style="list-style-type: none"> • hide(Request \$request) • This method takes a req object as a parameter and it works as to hide a specific object
PrivacyPresetController		
PrivacySetCircleController		

Controller Name	Modifier Type	Method and Description
PrivacySetUserController		
PublicFigureController		
QuestionController		
ReactionController	String	<ul style="list-style-type: none"> insert_reaction (Request \$request) This method takes a req object as a parameter and it works as to insert new reaction for specific object.
	String	<ul style="list-style-type: none"> checkinit_reaction(Request \$request) This method takes a req object as a parameter and it works as to check the initial reaction for the specific object
	String	<ul style="list-style-type: none"> delete_reaction (Request \$request) This method takes a req object as a parameter and it works as to delete reaction for an object
	Object	<ul style="list-style-type: none"> getstats_reaction (Request \$request) This method takes a req object as a parameter and it works as to get the count of each reaction for an object
ReplyController	String	<ul style="list-style-type: none"> create(Request \$request) This method takes a req object as a parameter and it works as to create replies instance
	String	<ul style="list-style-type: none"> retrieve(Request \$request) This method takes a req object as a parameter and it works as to retrieve replies
	Array	<ul style="list-style-type: none"> getReplies(Request \$request) This method takes a req object as a parameter and it works as to get replies
SavedObjectController		

Controller Name	Modifier Type	Method and Description
SavedPresetsController	Object	<ul style="list-style-type: none"> • check(Request \$req) • This method takes a req object as a parameter and it works as to check if a specific preset is existing for users or not
	String	<ul style="list-style-type: none"> • save(Request \$req) • This method takes a req object as to save the preset
	Array	<ul style="list-style-type: none"> • Delete(Request \$req) • This method takes a req object as a parameter and it works as to delete preset
	String	<ul style="list-style-type: none"> • setAsDefault(Request \$req) • This method takes a req object as a parameter and it works as to set a specific preset as the default for a user
	Object	<ul style="list-style-type: none"> • get(Request \$req) • This method takes a req object as a parameter and it works as to gets the saved preset for a user
SportController	Integer	<ul style="list-style-type: none"> • createSport(Request \$request) • This method takes a req object as a parameter and it works as to create an sport instance
	Array	<ul style="list-style-type: none"> • retrieveSport(Request \$request) • This method takes a req object as a parameter and it works as to retrieve sport objects
	String	<ul style="list-style-type: none"> • updateSport(Request \$request) • This method takes a req object as a parameter and it works as to update sport objects
	String	<ul style="list-style-type: none"> • deleteSport(Request \$request) • This method takes a req object as a parameter and it works as to delete sport objects

Controller Name	Modifier Type	Method and Description
	Array	<ul style="list-style-type: none"> • sportsSearch(Request \$request) • This method takes a req object as a parameter and it works as to search for sport
	Array	<ul style="list-style-type: none"> • sports(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the sports
StageController		
StageMemberController		
StaticMessageController	String	<ul style="list-style-type: none"> • create(Request \$request) • This method takes a req object as a parameter and it works as to create a new static message
	String	<ul style="list-style-type: none"> • translate(Request \$request) • This method takes a req object as a parameter and it works as to translate the static message
	String	<ul style="list-style-type: none"> • idOf(Request \$request) • This method takes a req object as a parameter and it works as to get the id of a message
	String	<ul style="list-style-type: none"> • idOfGlob(Request \$request) • This method takes a req object as a parameter and it works as to get the id of a message after searching globally in all languages
	String	<ul style="list-style-type: none"> • getTranslation(Request \$request) • This method takes a req object as a parameter and it works as to get translation
	String	<ul style="list-style-type: none"> • getTranslationByWord(Request \$request) • This method takes a req object as a parameter and it works as to get the translation for each word

Controller Name	Modifier Type	Method and Description
	Array	<ul style="list-style-type: none"> • addNewWord(Request \$request) • This method takes a req object as a parameter and it works as to add new word
StaticMessageController	Object	<ul style="list-style-type: none"> • collect(Request \$request) • This method takes a req object as a parameter and it works as to collect and retrieve all the static message
	Array	<ul style="list-style-type: none"> • boundTranslation(Request \$request) • This method takes a req object as a parameter and it works as to translate the messages
	String	<ul style="list-style-type: none"> • forceTranslation(Request \$request) • This method takes a req object as a parameter and it works as to force translate the messages and override the existing translation if exists
TalentCategoryController		
TalentController		
TalentObjectController		
UserAddressController	Array	<ul style="list-style-type: none"> • getStates(Request \$request) • This method takes a req object as a parameter and it works as to get the states in the address
	Array	<ul style="list-style-type: none"> • getRegions(Request \$request) • This method takes a req object as a parameter and it works as to get the regions in the address
	String	<ul style="list-style-type: none"> • create(Request \$request) • This method takes a req object as a parameter and it works as to create the address for the user

Controller Name	Modifier Type	Method and Description
	String	<ul style="list-style-type: none"> • deleteAddresses(Request \$request) • This method takes a req object as a parameter and it works as to delete the address for the user
UserBlocksController	String	<ul style="list-style-type: none"> • blockUser(Request \$request) • This method takes a req object as a parameter and it works as to block specific user
	String	<ul style="list-style-type: none"> • unblockUser(Request \$request) • This method takes a req object as a parameter and it works as to unblock specific user
	Object	<ul style="list-style-type: none"> • isBlocked(Request \$request) • This method takes a req object as a parameter and it works as to check if specific user is blocked or not
	Object	<ul style="list-style-type: none"> • retrieveBlockedUser(Request \$request) • This method takes a req object as a parameter and it works as to retrieve all the blocked users
UserCheckinsController		
UserController	String	<ul style="list-style-type: none"> • activeRequests() • This method works as to get all the active requests
	String	<ul style="list-style-type: none"> • userlogin(Request \$request) • This method takes a req object as a parameter and it works as to let the user log in
	Object	<ul style="list-style-type: none"> • goHome(Request \$request) • This method takes a req object as a parameter and it works as to get the view of home for the user
	Object	<ul style="list-style-type: none"> • offerVerify(Request \$request) • This method takes a req object as a parameter and it works as to offer the user verification

Controller Name	Modifier Type	Method and Description
	Object	<ul style="list-style-type: none"> offerVerifyExpired(Request \$request) This method takes a req object as a parameter and it works as to check if the verification is expired or not
	String	<ul style="list-style-type: none"> requestVerify(Request \$request) This method takes a req object as a parameter and it works as to request for verification
	Object	<ul style="list-style-type: none"> attemptVerify(Request \$request, \$token) This method takes a req object and token as a parameter and it works as to attempt for verification
	Object	<ul style="list-style-type: none"> attemptUnblock(Request \$request, \$user, \$token) This method takes a req object and token and user as a parameter and it works as to attempt for unblock
	Object	<ul style="list-style-type: none"> attemptUnblockTempPassword(Request \$request, \$user, \$token) This method takes a req object and token and user as a parameter and it works as to attempt for unblock for temporary password
	Object	<ul style="list-style-type: none"> lost() This method works as to route the user to the guest page
	String	<ul style="list-style-type: none"> authUsersLanguage() This method works as to authenticate the user language
	Object	<ul style="list-style-type: none"> retrievePeopleKnocks(Request \$request) This method takes a req object as a parameter and it works as to get the people's knocks

Controller Name	Modifier Type	Method and Description
	Integer	<ul style="list-style-type: none"> • retrieveOlderPeopleKnocks(Request \$request) • This method takes a req object as a parameter and it works as to get the people's older knocks
	Integer	<ul style="list-style-type: none"> • retrieveNewerPeopleKnocks(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the newer knocks
	Object	<ul style="list-style-type: none"> • retrieveUserKnocks(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the user knocks
	Object	<ul style="list-style-type: none"> • retrieveOlderUserKnocks(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the older user knocks
	Object	<ul style="list-style-type: none"> • getUserAllCircles(Request \$request) • This method takes a req object as a parameter and it works as to get all circles that the user has
	Object	<ul style="list-style-type: none"> • retrieveNewerUserKnocks(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the newer user knocks
		getDefaultValue(Request \$request)
	Object	<ul style="list-style-type: none"> • retrieveUserGroups(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the user groups
	String	<ul style="list-style-type: none"> • check(Request \$request) • This method takes a req object as a parameter and it works as to check the user name existence

Controller Name	Modifier Type	Method and Description
UserController	String	<ul style="list-style-type: none"> • mailCheck(Request \$request) • This method takes a req object as a parameter and it works as to check his mail while registration
	String	<ul style="list-style-type: none"> • register(Request \$request) • This method takes a req object as a parameter and it works as to register
		getSuggestions(Request \$request)
	Object	<ul style="list-style-type: none"> • initChat(Request \$request) • This method takes a req object as a parameter and it works as to initialize chat for the user
	Object	<ul style="list-style-type: none"> • getInfo(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the user information
	Object	<ul style="list-style-type: none"> • getInfoLazy(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the initial user information
	Object	<ul style="list-style-type: none"> • getUserCircles(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the user circles
	Object	<ul style="list-style-type: none"> • retrieveContact(Request \$request) • This method takes a req object as a parameter and it works as to retrieve the contacts of the user
	String	<ul style="list-style-type: none"> • updateProfileIndex(Request \$request) • This method takes a req object as a parameter and it works as to update the profile picture of the user

Controller Name	Modifier Type	Method and Description
	Array	<ul style="list-style-type: none"> • <code>searchForFriends(Request \$request)</code> • This method takes a req object as a parameter and it works as to search for the friends.
		<code>searchForUsersByNames(Request \$request)</code>
		<code>globalUserSearch(Request \$request)</code>
	Array	<ul style="list-style-type: none"> • <code>mainSearch(Request \$request)</code> • This method takes a req object as a parameter and it works as to search for the knocks, comment, reply and groups.
	Object	<ul style="list-style-type: none"> • <code>friendsToChat(Request \$request)</code> • This method takes a req object as a parameter and it works as to get the friends for the chats
	View	<ul style="list-style-type: none"> • <code>routeToProfile(Request \$request , \$user)</code> • This method takes a req object and user as a parameter and it works as to route for the group page
		<code>routeToProfileById(Request \$request, \$user)</code>
	String	<ul style="list-style-type: none"> • <code>updateUserfirstName(Request \$request)</code> • This method takes a req object as a parameter and it works as to update the user's first name
	String	<ul style="list-style-type: none"> • <code>updateUsermiddleName(Request \$request)</code> • This method takes a req object as a parameter and it works as to update the user's middle name

Controller Name	Modifier Type	Method and Description
	String	<ul style="list-style-type: none"> • updateUserlastName(Request \$request) • This method takes a req object as a parameter and it works as to update the user's last name
	String	<ul style="list-style-type: none"> • updateUsernickName(Request \$request) • This method takes a req object as a parameter and it works as to update the user's nickname
	String	<ul style="list-style-type: none"> • updateUserbirthdate(Request \$request) • This method takes a req object as a parameter and it works as to update the user's birthdate
	String	<ul style="list-style-type: none"> • updateUserorientation(Request \$request) • This method takes a req object as a parameter and it works as to update the user's orientation
	String	<ul style="list-style-type: none"> • updateUserreligion(Request \$request) • This method takes a req object as a parameter and it works as to update the user's religion
	String	<ul style="list-style-type: none"> • updateUsermaritalstatus(Request \$request) • This method takes a req object as a parameter and it works as to update the user's marital status
	String	<ul style="list-style-type: none"> • updateUserServiceBio(Request \$request) • This method takes a req object as a parameter and it works as to update the user's bio
	String	<ul style="list-style-type: none"> • updateUserphone(Request \$request) • This method takes a req object as a parameter and it works as to update the user's phone

Controller Name	Modifier Type	Method and Description
	String	<ul style="list-style-type: none"> • updateUsergender(Request \$request) • This method takes a req object as a parameter and it works as to update the user's gender
	String	<ul style="list-style-type: none"> • deleteUsermiddleName(Request \$request) • This method takes a req object as a parameter and it works as to delete the user's middle name
	String	<ul style="list-style-type: none"> • deleteUsernickName(Request \$request) • This method takes a req object as a parameter and it works as to delete the user's nick name
	String	<ul style="list-style-type: none"> • deleteUserorientation(Request \$request) • This method takes a req object as a parameter and it works as to delete the user's orientation
	String	<ul style="list-style-type: none"> • deleteUserreligion(Request \$request) • This method takes a req object as a parameter and it works as to delete the user's religion
	String	<ul style="list-style-type: none"> • deleteUsermaritalstatus(Request \$request) • This method takes a req object as a parameter and it works as to delete the user's marital status
	String	<ul style="list-style-type: none"> • deleteUserbio(Request \$request) • This method takes a req object as a parameter and it works as to delete the user's bio
	String	<ul style="list-style-type: none"> • deleteUserphone(Request \$request) • This method takes a req object as a parameter and it works as to delete the user's phone
		hasAddresses(Request \$request)

Controller Name	Modifier Type	Method and Description
	String	<ul style="list-style-type: none"> • <code>updateSettings(Request \$request)</code> • This method takes a req object as a parameter and it works as to update the user configurations
	String	<ul style="list-style-type: none"> • <code>updateName(Request \$req)</code> • This method takes a req object as a parameter and it works as to update the username
	String	<ul style="list-style-type: none"> • <code>updatePassword(Request \$req)</code> • This method takes a req object as a parameter and it works as to update the user password
	String	<ul style="list-style-type: none"> • <code>updateDisplayName(Request \$req)</code> • This method takes a req object as a parameter and it works as to update the user display name
	String	<ul style="list-style-type: none"> • <code>updatePrivacy(Request \$req)</code> • This method takes a req object as a parameter and it works as to update the user privacy
	String	<ul style="list-style-type: none"> • <code>updateAttr(Request \$req)</code> • This method takes a req object as a parameter and it works as to update the user attributes
	String	<ul style="list-style-type: none"> • <code>getDeviceInfo(Request \$request)</code> • This method takes a req object as a parameter and it works as to get the user devices information where the user logs in
	String	<ul style="list-style-type: none"> • <code>getUserDevices(Request \$request)</code> • This method takes a req object as a parameter and it works as to get the user devices where the user logs in

Controller Name	Modifier Type	Method and Description
	String	<ul style="list-style-type: none"> • guiedBlockedAccount(Request \$request) • This method takes a req object as a parameter and it works as to get the user blocked accounts
	String	<ul style="list-style-type: none"> • forgotMyPasswordAsk(Request \$request) • This method takes a req object as a parameter and it works as to set the user to be forgotten his password
UserHashtagsController		
UserKeywordsController		
UserLogController		
UserReportsController		
UserRequestController	String	<ul style="list-style-type: none"> • sendGroup(Request \$request) • This method takes a req object as a parameter and it works as to send request for a group
	String	<ul style="list-style-type: none"> • sendOne(Request \$request) • This method takes a req object as a parameter and it works as to send a request for a user
	String	<ul style="list-style-type: none"> • cancelOne(Request \$request) • This method takes a req object as a parameter and it works as to cancel the request
	String	<ul style="list-style-type: none"> • ignoreOne(Request \$request) • This method takes a req object and user as a parameter and it works as to ignore the request and make its notification hidden
	String	<ul style="list-style-type: none"> • accept(Request \$request) • This method takes a req object and user as a parameter and it works as to accept the request

Controller Name	Modifier Type	Method and Description
	Object	<ul style="list-style-type: none"> • <code>getGroupWaitResponse(Request \$request)</code> • This method takes a req object as a parameter and it works as to get the requests that are sent for groups and these requests are still waiting for a response
	String	<ul style="list-style-type: none"> • <code>sendGroupRequest(Request \$request)</code> • This method takes a req object as a parameter and it works as to initialize a request for a group
	Object	<ul style="list-style-type: none"> • <code>getGroupResponse(Request \$request)</code> • This method takes a req object as a parameter and it works as to get the group requests response
	Void	<ul style="list-style-type: none"> • <code>declineRequestForGroup(Request \$request)</code> • This method takes a req object as a parameter and it works as to decline the group requests
	Void	<ul style="list-style-type: none"> • <code>declineRequestForGroup(Request \$request)</code> • This method takes a req object as a parameter and it works as to decline the group requests
	Boolean	<ul style="list-style-type: none"> • <code>checkGroupResponse(Request \$request)</code> • This method takes a req object as a parameter and it works as to get the group requests response

5.2.Models

5.2.1.Models List

Answer
Assistant
Ballon
Blob
Candy_session
Career
Circle
Circle_member
Comment
Country
Education
Envelope
FavUser
Group
Group_member
hashtags
High_education
Hobby
Ignore_object
Jam_circle
Knock
Language
MessageBus
Obj
Privacy_preset
Privacy_set_circle

Privacy_set_user
Public_figure
Question
Reaction
Reply
saved_object
Saved_presets
Sport
Stage_member
Stage
Static_message
Talent_category
Talent_object
Talent
user_blocks
User_checkins
User_hashtags
User_keywords
User_log
user_reports
User_request
User
UserAddress

5.2.2.Models specification

Model Name	Modifier Type	Method and Description
Answer		
Assistant		
Ballon	<p>Void</p> <ul style="list-style-type: none"> • initialize(\$object) • This method takes an object as a parameter and it works as to initialize a new balloon and insert it in the database. <p>Object</p> <ul style="list-style-type: none"> • index() • This method takes an object as a parameter and it works as to get the index of the balloon <p>Void</p> <ul style="list-style-type: none"> • friendRequestBalloon(\$sender , \$reciever , \$request) • This method takes sender and receiver variables and request object as a parameter and it works as to create a friend request balloon. <p>Void</p> <ul style="list-style-type: none"> • friendRequestAccepted(\$sender , \$reciever) • This method takes sender and receiver variables as a parameter and it works as to create a balloon to indicate that the friend request is accepted. <p>Void</p> <ul style="list-style-type: none"> • userComment(\$sender, \$reciever, \$knock, \$comment, \$parent, \$object_type) • This method takes sender and receiver , knock, comment, parent and object type variables as a parameter and it works as to create a balloon of the user comment 	

Model Name	Modifier Type	Method and Description
	Void	<ul style="list-style-type: none"> • userReply(\$sender, \$reciever, \$knock, \$comment, \$reply, \$parent) • This method takes sender and receiver , knock, reply and parent variables as a parameter and it works as to create a balloon of the user reply
	Void	<ul style="list-style-type: none"> • userReaction(\$sender, \$reciever, \$reaction, \$object, \$object_type, \$child, \$parent) • This method takes sender and receiver , reaction, object, object type, child and parent variables as a parameter and it works as to create a balloon of the user reactions
	Void	<ul style="list-style-type: none"> • sendNotification(\$balloon_type, \$not ,\$object , \$reference_url, \$is_replyable , \$sender_name, \$has_picture , \$image_url , \$content) • This method takes balloon type, notification, object -to be notifying-, reference url-for routing the user for the notification location in case of clicking on it- and boolean variable of whether this notification is reliable or not, and the sender name, and if the notification includes a picture or not, and if there exist a picture then the image url and the notification content variables as a parameter and it works as to create a balloon to indicate that the friend request is accepted.
Blob	Boolean	<ul style="list-style-type: none"> • imageBlob(\$object) • This method takes object named object as a parameter and it works as to insert new image blob.

Model Name	Modifier Type	Method and Description
File	Boolean	<ul style="list-style-type: none"> • fileBlob(\$object) • This method takes object named object as a parameter and it works as to insert new file blob.
	Boolean	<ul style="list-style-type: none"> • avatarBlob(\$object) • This method takes object named object as a parameter and it works as to insert new avatar blob.
	Boolean	<ul style="list-style-type: none"> • coverBlob(\$object) • This method takes object named object as a parameter and it works as to insert new cover blob.
	Integer	<ul style="list-style-type: none"> • recordBlob(\$object) • This method takes object named object as a parameter and it works as to insert new record blob.
	Void	<ul style="list-style-type: none"> • assignParent(\$object) • This method takes object named object as a parameter and it works as to assign a parent for a blob.
	Object	<ul style="list-style-type: none"> • index() • This method works as to retrieve the index of a blob
	String	<ul style="list-style-type: none"> • mongoToken() • This method works as to retrieve the token from mongo database of a blob
	String	<ul style="list-style-type: none"> • blobQuote() • This method works as to retrieve the quote of a blob
	Double	<ul style="list-style-type: none"> • blobDuration() • This method works as to retrieve the duration of a blob
	String	<ul style="list-style-type: none"> • vnBlob() • This method works as to enable the format of the file

Model Name	Modifier Type	Method and Description
Candy_session	String	<ul style="list-style-type: none"> • retrieveImgBlob() • This method works as to retrieve the image blob.
	String	<ul style="list-style-type: none"> • retrieveImgCompressed() • This method works as to retrieve the image blob but compressed.
Career	Void	<ul style="list-style-type: none"> • initialize(\$works_at, \$works_since,\$works_to, \$works_what,\$works_as) • This method takes works as, works since, works to, works what and works as, as a parameter and it works as to initialize career instance.
Circle_member	Void	<ul style="list-style-type: none"> • initialize(\$user , \$circle) • This method takes user and circle objects as a parameter and it works as to initialize circle member instance.
Circle	Boolean	<ul style="list-style-type: none"> • isMember (\$user) • This method takes user as a parameter and it works as to check if a specific user is a member in a circle or not.
	Void	<ul style="list-style-type: none"> • initialize(\$name , \$thumbnail) • This method takes name of the circle and the thumbnail as a parameter and it works as to initialize a new circle.
	Object	<ul style="list-style-type: none"> • retrieveCircle() • This method works as to retrieve a specific circle.
Comment	Object	<ul style="list-style-type: none"> • commentReplies() • This method works as to retrieve a specific comment replies.
	Void	<ul style="list-style-type: none"> • addFollower(\$user) • This method works as to add a follower for a comment

Model Name	Modifier Type	Method and Description
Country	Object	<ul style="list-style-type: none"> knockIndex() This method works as to get the knock index
	Void	<ul style="list-style-type: none"> initialize(\$object) This method takes comment object named object as a parameter and it works as to initialize a new comment.
Education	Void	<ul style="list-style-type: none"> initialize(\$study_at , \$study_what, \$study_since , \$study_to , \$grade) This method takes study_at , study_what, study_since , study_to ,and grade as a parameter and it works as to initialize education instance.
Envelope		
fav_user		
Group_member	Void	<ul style="list-style-type: none"> initialize(\$user_id , \$group_id, \$position) This method takes user_id ,group_id, and position as a parameter and it works as to initialize group member instance.
	Void	<ul style="list-style-type: none"> isAdmin() This method works as to check if a specific user is an admin or not
Group	Object	<ul style="list-style-type: none"> memberPosition() This method works as to retrieve the user position
	Integer	<ul style="list-style-type: none"> groupRequests() This method works as to retrieve the group requests count

Model Name	Modifier Type	Method and Description
Group	Void	<ul style="list-style-type: none"> • initialize(\$name , \$category, \$thumbnail,\$preset) • This method takes \$name ,category, thumbnail, and preset as a parameter and it works as to initialize group instance.
	Object	<ul style="list-style-type: none"> • index() • This method works as to retrieve the index of a group
	Void	<ul style="list-style-type: none"> • increaseMembers() • This method works as to update the group members and increase it by one.
	Boolean	<ul style="list-style-type: none"> • checkUser() • This method works as to check if a user is a member of a group or not
	Array	<ul style="list-style-type: none"> • photos() • This method works as to get the group photos
	Array	<ul style="list-style-type: none"> • voices() • This method works as to get the group voice records
	Array	<ul style="list-style-type: none"> • files() • This method works as to get the group files
	Array	<ul style="list-style-type: none"> • videos() • This method works as to get the group videos
	Void	<ul style="list-style-type: none"> • decreaseMembers() • This method works as to update the group members and decrease it by one.
	Array	<ul style="list-style-type: none"> • getGroupKnocks(\$limits) • This method takes limits object as a parameter and it works as to retrieve the knocks of the group in order.

Model Name	Modifier Type	Method and Description
knock	Array	<ul style="list-style-type: none"> • <code>getGroupKnocksRegular()</code> • This method works as to retrieve the knocks of the group.
	Array	<ul style="list-style-type: none"> • <code>getGroupKnocksRegularMin(\$min)</code> • This method takes min object as a parameter and it works as to retrieve the group knocks with the given minimum limit.
	Array	<ul style="list-style-type: none"> • <code>getGroupKnocksRegularMax(\$max)</code> • This method takes max object as a parameter and it works as to retrieve the group knocks with the given maximum limit.
hashtags	Void	<ul style="list-style-type: none"> • <code>createOrUpdate(\$q)</code> • This method works as to create a hashtag if it is the first hashtag or to update its count if it already exists
	Array	<ul style="list-style-type: none"> • <code>retrieveOlderKnocks(\$min)</code> • This method takes min object as a parameter and it works as to retrieve the older knocks
	Array	<ul style="list-style-type: none"> • <code>retrieveNewerKnocks(\$max)</code> • This method takes max object as a parameter and it works as to retrieve the newer knocks
	Array	<ul style="list-style-type: none"> • <code>retrieveKnocks()</code> • This method takes min object as a parameter and it works as to retrieve the knocks
High_education	Void	<ul style="list-style-type: none"> • <code>initialize(\$study_at, \$study_what, \$study_since, \$study_to, \$grade)</code> • This method takes study_at, study_what, study_since, study_to, and grade as a parameter and it works as to initialize high education instance.

Model Name	Modifier Type	Method and Description
Hobby	Void	<ul style="list-style-type: none"> • initialize(\$name) • This method takes name of the hobby as a parameter and it works as to initialize hobby instance.
ignore_object		
Jam_circle		
Knock	Void	<ul style="list-style-type: none"> • addFollower(\$user) • This method works as to add a follow for a specific knock
	Boolean	<ul style="list-style-type: none"> • canEdit(\$user) • This method works as to check if a user can edit a knock or not
	Void	<ul style="list-style-type: none"> • deleteKnock() • This method works as to delete a knock
	Object	<ul style="list-style-type: none"> • isIgnored(\$user) • This method works as to check if a knock is being notified with or not
	String	<ul style="list-style-type: none"> • view(\$requester) • This method works as to view the knocks for the permitted users
	Void	<ul style="list-style-type: none"> • initialize(\$object) • This method takes knock object as a parameter and it works as to initialize knock instance.
	Object	<ul style="list-style-type: none"> • knockIndex() • This method works as to retrieve the index of a knock
	Boolean	<ul style="list-style-type: none"> • typing(\$user) • This method takes user object as a parameter and it works as to create in the knock index that there is a comment is being typed.

Model Name	Modifier Type	Method and Description
Knock	Array	<ul style="list-style-type: none"> • runningTypings(\$user) • This method takes user object as a parameter and it works as to indicate the user who is typing for this knock.
	Boolean	<ul style="list-style-type: none"> • watchSeen(\$user) • This method takes user object as a parameter and it works as to monitor whether this knock is being seen or not.
	Boolean	<ul style="list-style-type: none"> • hasSeen(\$user) • This method takes user object as a parameter and it works as to indicate whether this knock is being seen or not.
Language	Void	<ul style="list-style-type: none"> • initialize(\$object) • This method takes language object as a parameter and it works as to initialize language instance.
MessageBus	Void	<ul style="list-style-type: none"> • defineDefault(\$englishString) • This method takes English String as a parameter and it works as to set this English string as the default language.
	Void	<ul style="list-style-type: none"> • addTranslation(\$string , \$language) • This method takes the String and the language as a parameter and it works as to add this new translation.
obj	Object	<ul style="list-style-type: none"> • getMyChild() • This method works as to get the object child
	Object	<ul style="list-style-type: none"> • addFollower(\$user) • This method works as to add a follower to the object with specific user
	Object	<ul style="list-style-type: none"> • knockIndex() • This method works as to get the knock index

Model Name	Modifier Type	Method and Description
		index() updatePublicPresetNum(\$num)
	Void	<ul style="list-style-type: none"> initialize(\$type) This method takes the object type as a parameter and it works as to initialize a new object.
	Boolean	<ul style="list-style-type: none"> isAvailable (\$requestMaker) This method takes the requestMaker object as a parameter and it works as to check for the object availability if it is public or private or customized.
Privacy_preset		
Privacy_set_circle		
Privacy_set_user		
Public_figure	Void	<ul style="list-style-type: none"> initialize(\$name,\$type,\$verified) This method takes the name and type of the public figure and if this public figure is verified or not as a parameter and it works as to initialize public figure instance.
Question		
Reaction	Void	<ul style="list-style-type: none"> init(\$user, \$object, \$reaction, \$type) This method takes reaction object and reaction type and user as a parameter and it works as to initialize new reaction instance.
	Void	<ul style="list-style-type: none"> initialize(\$object) This method takes reaction object as a parameter and it works as to initialize new reaction instance.

Model Name	Modifier Type	Method and Description
	String	<ul style="list-style-type: none"> • initializeReaction(\$parent_id, \$type) • This method takes parent object id and the reaction type as a parameter and it works as to initialize reaction instance for a specific object.
Reply	Object	<ul style="list-style-type: none"> • replyReplies() • This method works as to get the replies for a specific reply.
	Void	<ul style="list-style-type: none"> • initialize(\$object) • This method takes reply object as a parameter and it works as to initialize new reply.
saved_object		
Saved_presets		
Sport	Void	<ul style="list-style-type: none"> • initialize(\$name) • This method takes sport name as a parameter and it works as to initialize new sport.
Stage_member		
Stage		
Static_message	Void	<ul style="list-style-type: none"> • initialize(\$object) • This method takes static message object as a parameter and it works as to initialize new static message.
	Void	<ul style="list-style-type: none"> • translate(\$object) • This method takes static message object as a parameter and it works as to translate it.
Talent_category		
Talent_object		
Talent		

Model Name	Modifier Type	Method and Description
user_blocks	Void	<ul style="list-style-type: none"> • initialize(\$blocked_user_id) • This method takes the blocked user id and initialize the block for him
	Object	<ul style="list-style-type: none"> • index() • This method returns the Index of the block
User_checkins		
User_hashtags		
User_keywords		
User_log	Void	<ul style="list-style-type: none"> • autoLog(\$url , \$ip , \$method) • This method takes the url and the ip of the user and the method as a parameter and it works as to log in. specified for him or anonymously,
	Void	<ul style="list-style-type: none"> • addUserLog(\$user , \$url , \$ip) • This method takes user id and the url and the ip of the user as a parameter and it works as to log in.
	Void	<ul style="list-style-type: none"> • addAnonymousLog(\$url , \$ip) • This method takes the url and the ip of the user as a parameter and it works as to log in as a guest without the user id.
user_reports		
User_request	Void	<ul style="list-style-type: none"> • initialize(\$sender_id , \$receiver_id) • This method takes the receiver id and the sender id for this request as a parameter and it works as to initialize new friend request.
	Void	<ul style="list-style-type: none"> • initializeForGroups(\$sender_id, \$receiver_id) • This method takes the receiver id and the sender id for this request as a parameter and it works as to initialize new group request.

Model Name	Modifier Type	Method and Description
User	Boolean	<ul style="list-style-type: none"> • <code>isMemberIn(\$circleToCheck, \$userToCheck)</code> • This method takes the circle object and the user object as a parameter and it works as to check if the given user is a member of the given circle or not.
	Array	<ul style="list-style-type: none"> • <code>entryStatus()</code> • This method works as to get the entry for each user
	Array	<ul style="list-style-type: none"> • <code>getAddresses(\$requester)</code> • This method works as to get the address for each user
	Array	<ul style="list-style-type: none"> • <code>getSuggestions()</code> • This method works as to get the suggestion for each user
	Array	<ul style="list-style-type: none"> • <code>getSuggestionsAvoid(\$arr)</code> • This method works as to get the suggestion that are avoided for each user
	Object	<ul style="list-style-type: none"> • <code>isSuggestableInList(\$arr, \$user)</code> • This method works as to get the suggestions list
	Object	<ul style="list-style-type: none"> • <code>isSuggestable(\$user)</code> • This method works as to check if the user is suggestible or not
	Object	<ul style="list-style-type: none"> • <code>passFriends(\$friend, \$prev)</code> • This method works as to check if the user is suggestible or not
	Object	<ul style="list-style-type: none"> • <code>fullName()</code> • This method works as to get the full name of the user
	Array	<ul style="list-style-type: none"> • <code>friendsByWeight()</code> • This method works as to set weight for friends

Model Name	Modifier Type	Method and Description
User	Object	<ul style="list-style-type: none"> • <code>isBlockedBy(\$user)</code> • This method works as to check if the user is being blocked by other user
	Object	<ul style="list-style-type: none"> • <code>isBlocking(\$user)</code> • This method works as to check if the user is blocking other users or not
	Object	<ul style="list-style-type: none"> • <code>hasNoBlocks(\$user)</code> • This method works as to check if the user has no blocks
	Object	<ul style="list-style-type: none"> • <code>friends()</code> • This method works as to get the user friends
	Object	<ul style="list-style-type: none"> • <code>userAddresses()</code> • This method works as to get the user address
	Object	<ul style="list-style-type: none"> • <code>knocks()</code> • This method works as to get the user knocks
	Void	<ul style="list-style-type: none"> • <code>updateFriendshipWeight(\$fr, \$weight)</code> • This method works as to update the friendship weight
	Boolean	<ul style="list-style-type: none"> • <code>updateLastSeen()</code> • This method works as to update the last seen for a user.
	Boolean	<ul style="list-style-type: none"> • <code>turnOffChat()</code> • This method works as to turn off the chat for the user and indicate that this user is offline.
	Object	<ul style="list-style-type: none"> • <code>cog()</code> • This method works as to retrieve the user configurations.
	Void	<ul style="list-style-type: none"> • <code>initialaize(\$object)</code> • This method takes the user object as a parameter and it works as to initialize a new user.
	Void	<code>initForTesting(\$object)</code>

Model Name	Modifier Type	Method and Description
	Void	<ul style="list-style-type: none"> • updateToken(\$token) • This method takes the token as a parameter and it works as to update it for the user tokens.
	Object	<ul style="list-style-type: none"> • blobObject() • This method works as to retrieve the user profile picture blob.
	Object	<ul style="list-style-type: none"> • defaultPreset() • This method works as to get the default preset for a user
	Boolean	<ul style="list-style-type: none"> • setDefaultPreset(\$preset) • This method works as to set the default preset for a user
	Array	<ul style="list-style-type: none"> • retrieveForUserLazy(\$requester) • This method takes the request object as a parameter and it works as to retrieve the initial low load user information.
	Object	<ul style="list-style-type: none"> • retrieveForUser(\$requester) • This method takes the request object as a parameter and it works as to retrieve all the user information.
	Object	<ul style="list-style-type: none"> • hasSentRequest(\$target) • This method takes the target user id as a parameter and it works as to retrieve the requests that the user sent.
	Object	<ul style="list-style-type: none"> • hasFriendRequest(\$from) • This method takes the request sender user id as a parameter and it works as to retrieve the friends requests from the given user.
	Object	<ul style="list-style-type: none"> • hasFriendRequestObject(\$from) • This method takes the request sender user id as a parameter and it works as to retrieve the latest request from the given user.

Model Name	Modifier Type	Method and Description
	Object	<ul style="list-style-type: none"> • defaultPresets() • This method works as to retrieve the user default privacy set.
	Boolean	<ul style="list-style-type: none"> • isFriend(\$user) • This method takes the user object as a parameter and it works as to check if the given user is a friend with the account owner or not by checking the main cycle of the account owner.
	Object	<ul style="list-style-type: none"> • mainCircle() • This method works as to retrieve the main circle of the account owner.
	Object	<ul style="list-style-type: none"> • getCircleId(\$name) • This method takes the circle name as a parameter and it works as to get the circle id of the given circle name.
	Boolean	<ul style="list-style-type: none"> • hasCircle(\$circle) • This method takes the circle object as a parameter and it works as to check if this user has circles or not.
	Boolean	<ul style="list-style-type: none"> • hasCircleById(\$circle) • This method takes the circle object as a parameter and it works as to check if this user has circles with the given circle id or not.
	Boolean	<ul style="list-style-type: none"> • hasReceivedRequest(\$id) • This method takes the request id as a parameter and it works as to check if this user has received request or not.
	Boolean	<ul style="list-style-type: none"> • hasCareer(\$works_at, \$works_since) • This method takes works at and works since as a parameter and it works as to check if this user has career or not.

Model Name	Modifier Type	Method and Description
	Boolean	<ul style="list-style-type: none"> • has_hobby(\$name, \$url) • This method takes hobby name and hobby url as a parameter and it works as to check if this user has hobby or not.
	Boolean	<ul style="list-style-type: none"> • has_public_figure(\$name, \$url) • This method takes public figure name and hobby url as a parameter and it works as to check if this user is public figure or not.
	Boolean	<ul style="list-style-type: none"> • hasEducation(\$uniname, \$study_since) • This method takes university name and the study since as a parameter and it works as to check if this user has education feature added by him in his account or not.
	Boolean	<ul style="list-style-type: none"> • hasHighEducation(\$uniname, \$study_since) • This method takes university name and the study since as a parameter and it works as to check if this user has high education feature added by him in his account or not.
	Boolean	<ul style="list-style-type: none"> • hasSport(\$sportname, \$url) • This method takes sport name and sport url as a parameter and it works as to check if this user has sport or not.
	Object	<ul style="list-style-type: none"> • userLanguage() • This method works as to retrieve the user languages.
	Object	<ul style="list-style-type: none"> • userLanguageFont() • This method works as to retrieve the user font family.
	Object	<ul style="list-style-type: none"> • userLanguageAlignment() • This method works as to retrieve the user language alignment.

Model Name	Modifier Type	Method and Description
User	String	<ul style="list-style-type: none"> • profilePicture() • This method works as to retrieve the user profile picture .
	Object	<ul style="list-style-type: none"> • profilePictures() • This method works as to retrieve the user profile pictures .
	Object	<ul style="list-style-type: none"> • coverPictures() • This method works as to retrieve the user cover pictures .
	Object	<ul style="list-style-type: none"> • profilePictureBlob() • This method works as to retrieve the user profile picture blob .
	Array	<ul style="list-style-type: none"> • profilePictureBlobObject() • This method works as to retrieve the user profile picture blob as object .
	Array	<ul style="list-style-type: none"> • coverPictureBlob() • This method works as to retrieve the user cover picture blob.
	String	<ul style="list-style-type: none"> • nextProfilePicture() • This method works as to retrieve the user's next profile picture .
	String	<ul style="list-style-type: none"> • currentProfilePicture() • This method works as to retrieve the user current profile picture .
	Object	<ul style="list-style-type: none"> • publicPreset() • This method works as to retrieve the user public preset .
	Array	<ul style="list-style-type: none"> • getPeopleKnocks(\$limits) • This method takes limits object as a parameter and it works as to retrieve the people knocks.
	Array	<ul style="list-style-type: none"> • getPeopleKnocksRegular() • This method works as to retrieve the knocks.

Model Name	Modifier Type	Method and Description
	Array	<ul style="list-style-type: none"> • <code>getPeopleKnocksRegularMin(\$min)</code> • This method takes min object as a parameter and it works as to retrieve the knocks with the given minimum limit.
	Array	<ul style="list-style-type: none"> • <code>getPeopleKnocksRegularMax(\$max)</code> • This method takes max object as a parameter and it works as to retrieve the knocks with the given maximum limit.
	Array	<ul style="list-style-type: none"> • <code>getUserKnocks(\$limits)</code> • This method takes limits object as a parameter and it works as to retrieve the user knocks.
	Integer	<ul style="list-style-type: none"> • <code>age()</code> • This method works as to retrieve the user's age .
	Integer	<ul style="list-style-type: none"> • <code>birthYear()</code> • This method works as to retrieve the user's birth year .
	Boolean	<ul style="list-style-type: none"> • <code>isKid()</code> • This method works as to check if the user is a kid or not
	Array	<ul style="list-style-type: none"> • <code>getUserKnocksRegular()</code> • This method works as to retrieve the user knocks.
	Array	<ul style="list-style-type: none"> • <code>getUserKnocksRegularMin(\$min)</code> • This method takes min object as a parameter and it works as to retrieve the user knocks with the given minimum limit.
	Array	<ul style="list-style-type: none"> • <code>getUserKnocksRegularMax(\$max)</code> • This method takes max object as a parameter and it works as to retrieve the user knocks with the given maximum limit.

Model Name	Modifier Type	Method and Description
	Object	<ul style="list-style-type: none"> • soundsLike(\$q) • This method takes q object as a parameter and it works as to search for the user with the given parameter if it has one of his information -naming- matched with q parameter.
	Object	<ul style="list-style-type: none"> • soundsLikeID(\$q) • This method takes q object as a parameter and it works as to search for the user with the given parameter if it has one of his information matched with q parameter.
	Object	<ul style="list-style-type: none"> • FriendsSoundsLikeID(\$q) • This method takes q object as a parameter and it works as to search for the user finds with the given parameter if it has one of his information matched with q parameter.
	Object	<ul style="list-style-type: none"> • friendsObjects() • This method works as to retrieve the user friends from the main circle.
	Object	<ul style="list-style-type: none"> • friendsToChat() • This method works as to retrieve the user friends and their last seen and their status.
	Boolean	<ul style="list-style-type: none"> • isMatched(\$q) • This method takes q object as a parameter and it works as to get the similarity between users accounts.
	String	<ul style="list-style-type: none"> • generateRandomString(\$length) • This method takes length object as a parameter and it works as to generate random strings

Model Name	Modifier Type	Method and Description
User	Void	<ul style="list-style-type: none"> pairAsFriend(\$friend) This method takes friend object as a parameter and it works as to pair both the friend and the account owner as friends with the same circle.
	Object	<ul style="list-style-type: none"> devices() This method works as to get the user devices
	Array	<ul style="list-style-type: none"> deviceInfo(\$device) This method works as to get the user devices information
	Void	<ul style="list-style-type: none"> createCirclesMembership(\$circles) This method takes circle object as a parameter and it works for the user to be a circle member.
	Object	<ul style="list-style-type: none"> isBlockedAccount() This method works as to check if the account is blocked
	Void	<ul style="list-style-type: none"> init(\$object) This method works as to initialize user address
UserAddress		

5.3. Specific Algorithms Used

5.3.1. Searching Algorithm

5.3.1.1. Algorithm Flowchart



5.3.1.2.Algorithm Description

First of all the user would enter the search key which he wants to search with, and then an associative array containing all the knocks, comments, reply, groups and users is created to be filtered through the algorithm.

The system would measure the language matching, so that if the language of the search key does match the language in the given associative array then the second step is to measure the block similarity which is done by the "Like" operator in the database and either the block similarity is existing or not the system proceeds to the third and the final step which is the similarity percentage , and if the similarity percentage is 50% or more then this object is accepted and pushed in the result array, along with the condition being false or true in the similarity percentage in connection with the false case of the language matching the system will proceed to the next step.

The next step would be filtering the users by creating an array for the users and then The system would measure the language matching, so that if the language of the search key does match the language in the given users array then the second step is to measure the block similarity which is done by the "Like" operator in the database and either the block similarity is existing or not the system proceeds to the third and the final step which is the sounds like similarity, and if the sounds like similarity is true then this object is accepted and pushed in the result array, and then the system finally prints the result array which contains the matched objects, or else the system will eventually return false, and the system terminates.

5.4.Vue Components

5.4.1.Vue Component List

example

knocksinput

knockselinput

knocksfileupload

knockscoveruploader

knocksdatepicker

knocksbutton

static_message

knocksloader

knocksloaderbar

knocksloaderprogress

selector

knocksnotification

knockssuser

knocksrecorder

knockssplayer

knocksmimg

knocksmimgframeless

knocksviewcircle

knocksreactor

knock

knocksmultipleuploader

knockssimageeditor

knockscirclechip

knocksmultipleswitch

knocksprivacysetter

knockspopover
knocksimageviewer
knocksaddcircle
knockscircleseditor
knocksknock
knockscomment
knockschildreply
knocksreply
knocksreactionstats
knockstaps
imagequote
knocksfileviewer
knocksballon
knocksuseractions
knocksquickaddcircle
knockselbutton
knocksknockinjector
knocksusercareers
knocksusereducation
knocksuserhigheducation
knocksuserhobby
knocksusersport
knocksuserabout
knocksretriver
knocksvoicerecognition
knockscroppie
knocksuseraboutedit
knocksusersportedit

knocksuserhobbyedit
knocksusereducationedit
knocksusercareeredit
knocksuseraboutdelete
knocksgroupcreation
knocksgroupslist
knockspagesearch
knocksgroupmembers
knocksimg
knockschattingzone
knocksconversation
knocksmessagesender
knocksgroupshortcut
knocksgroupjoining
knocksgroupmemberdelete
knocksgrouppictures
knocksgroupfiles
knocksgroupvoices
knocksrightbar
knocksprivacyadjustments
knocksgroupsettings
knocksuserinfo
knocksgroupmemberposition
knockshashhtag
knocksusercircles
knocksquickcircleadder
knockshashtagchip
knocksimagestates

knocksuserinfo.delete
knocksphoto.comments
knocksuser.general.info
knocksdepimg.viewer
knockscircle.members
knockscollapse
knocksmedia.object
knocksshow.keys
knocksrecord.mpeg
knocksvoice.commands.vue
knocksqick.address.vue
knocksaddress.viewer.vue
knocksel.datepicker.vue
knockselect.vue
knocksusers.suggestions.vue
knocksusers.settings.vue
knockswatchmy.window.vue
knocksblock.user.vue
knocksblock.userlist.vue
knocksverify.user.vue
knockstips.vue
knocksshow.device.vue
knocksdate.viewer.vue
knocksedit.circlename.vue
knockswelcome.slider.vue
knocksforgot.password.vue
knockssitemap.vue
knocksprivacy.tips.vue

knocksmeganumber.vue

knockstrendslist.vue

knocksgenderviewer.vue

5.4.2.Vue Component specification

1. knockselinput.vue

- Brief Description
- Props
- API Events
- Used Components

2. knocksfleinput.vue

- Brief Description

This component is used for uploading the profile picture for the user, Also it makes use of knockscroppie component with specific aspect ratio

- Props
- API Events
- Used Components

3. knockscoveruploader.vue

- Brief Description

This component is used for uploading the cover picture for the user, Also it makes use of knockscroppie component with specific aspect ratio

- Props
- API Events
- Used Components

4. knockscdatepicker.vue

- Brief Description

This component is used for picking a date from a calendar, it takes as input the scope, maximum date, minimum date, is required and the CSS customizations, it can also add and subtract dates for determining the minimum and the maximum values, it also can finally return the database format.

- Props
- API Events

- Used Components

5. knocksbutton.vue

- Brief Description

This component is used for viewing the button in the application interface and it can be considered as the basic form of the buttons being used

- Props

it takes icon, alignment direction to be a language flexible component, placeholder, disable placeholder, submit flag(in case of remotely submission or not), timeout (the maximum time for XHR), submit at (Submission URL), success at, success message, error at (array of objects which has the target of objects and the response), reset on success (reset the components on the same scope), submit on (array of scopes for scope variations), validation error, connection error, validate, materialize feedback, disabled, computed response (for example : if there is no error message so it returns success), precondition (additional validation)

- API Events

For events API global: knocks input status (the button reacts on it if it finds an error), knocks final submit (validates the error stack), knocks presubmit (executes the validation so it does the remote submission when the next action takes place), knocks button remote click (takes place when needing to enable component behind scenes), knocks submit past (emitted when the validations are true), knocks put reset (resets the component)

For events API local: knocks button clicked (does not have any payloads unless the button is clicked), knocks stack failed, knocks submit accepted (have payloads - which is an object which submits data and response), knocks submit rejected (when the XHR gets an error that we are targeting), knocks submit error (when XHR has an internal error and its payload is the error).

- Used Components

6. static_message.vue

- Brief Description

This component is used for the messages that are located in the database to be displayed with the user's language with specific alignment and font family

- Props

it takes as property : message id (and it translates this message from the database with its id), message (and it translates this message from the database with its name), align (to have specific alignment), replaceable : to enable it just set some target in your static messages, then enable the replacement option in your component and provide it with target and body.

EG: The next text is stored in the database as 'replace ** By Foo', in my case I need to give this text to involve it between the string, so I'll enable the replaceable option and setup my component like the following sample.

```
< static_message  
msg = "replace ** by Foo"  
replaceable  
:replacements = "[ { target : '**' , body : 'Bar' } ]" >  
< /static_message>
```

The Result : replace Bar by Foo

You can also set many replacements as much as you want, as you define your replacements as an array you can define any number of replacements you need, the next is the same like this but having 2 replacements.

The original message in the database will be 'replace ** by @@'

```
< static_message  
msg = "replace ** by @@"  
replaceable  
:replacements = "[ { target : '**' , body : 'Bar' } , { target : ' {@@' , body :  
'Foo'} ]" >  
< /static_message>
```

The Result : replace Bar by Foo

The input language (checks if the message is with its body and has no replace meant and the language of it is the same as the message language so no translation needed), and some CSS customization.

- API Events
- Used Components

7. **loader.vue**

- Brief Description
This component is for show loading
- Props
props used are the bg and progs and both have CSS styling classes
- API Events
- Used Components

8. **knocksloaderbar.vue**

- Brief Description
This component is for show loading
- Props
The props used are the bg and progs and the progs here is numeric and both have CSS styling classes
- API Events
- Used Components

9. **knocksprogressbarindecor.vue**

- Brief Description
This component is for show loading
- Props
The props used are the bg and progs and the progs here is numeric and can be displayed as animation and both have CSS styling classes
- API Events
- Used Components

10. **selector.vue**

- Brief Description

- Props
- API Events
- Used Components

11. knocksusershortcut.vue

- Brief Description

This component is for showing a quick presentation for the user information
- Props

The props : user (user id), lazy user (flag activates the knocks retriever component for respond), show image (boolean to hide or show the image), show accept shortcut, hide username, hide name, hide text info, hide popover, hide image, as chip (displays the profile picture in tiny way and the name to be fit for small areas), as result (displays the user and his name and the user actions and does not show the pop over to not confuse the user), as report (displays the user details of the user in a list such as first name, last name, religion), as call back (displays the name and the image and for the click event it calls back a function instead of showing the user profile), as URL and as name (displays only the text content and this fits the text content), as label (displays the profile picture in tiny way and the name to be fit for small areas), as card (has the profile picture and initialized with the name and the common friends and can be toggled to show more details and list of all common people and has reactors such as friend request), as smaller card (has the profile picture and shows the cover photo and initialized with the name and the common friends and can be toggled to show more details and list of all common people and has reactors such as friend request, and it has a fixed size), no rebound, extended, extras.
- API Events

The event API : knocks content changed (refresh the component), knocks user reload (forces the retriever to rebound the component), knocks user key update (update some keys in user object, payloads : user, patch -patch has key and value).

For example

```
App.$emit('knocksUserKeyUpdate', {user:1 ,
patch :[ { key : gender , value : 'male'} ] })
```

The users are retrieved remotely, so if I retrieve the user for the first time to stores them for next times.

- Used Components

This component uses previously made components : knocks retriever, knocksuserbar, elPopover, knockspopover, knocksimage, knocksueractions, knockscollapse, knocksaddressviewer, knocksshowkeys, staticmessage, and eltooltip.

12. knocksrecorder.vue

- Brief Description

This component is for recording user knocks and encode and compress and upload the user records and react with other components such as knock, knocksplayer and knock reply.

- Props

The props : upload data (object of XHR form), uploading finish, CSS classes, timer right, scope, hide player, player show timer, player show options, player show runtime, player show volume, upload in scope, loop recording, volatile (used in recognition and for memory management generally), record limit.

- API Events

Event API : recorder finished (emitted on the user release recorder button), knocks submit (react with the submission to execute the uploading), blob encoded (once the blob is encoded), record started, record stopped, record uploaded, knocks switch recording kit (for interest), record reset (for recognition), push to knock, knocks media query logged (in case if the recorder was not user or null recorder the recorder will have payload with its scope and null token and query token and has recored is set to false, for example : App.\$emit('knocksMediaQueryLogged' , { scope : this.scope , token : null , query : 'record' , hasRecord : false });

in other case if the user has a record and the event has the scope, token of the record in mongo database and the query type set to record and has record is set to true, for example :App.\$emit('knocksMediaQueryLogged' , { scope : vm.scope , token : record.token , query : 'record' , hasRecord : true });).

Window events : recognition events (has own property :if the browser supports the voice recognition or not, sound start : once

the user starts speaking, sound end, result : when recognition is detected, on error : when error is detected like no speech or network failure), recorder events (on data available : once the recorder has enough chunks, on stop : once the user finishes the recording, on load need : when all chunks are ready), DOM events (mouse over : show interest and react with knock and knocks reply components, mouse up and mouse down : toggling recording status.

- Used Components

It makes use of elpopover, static message, els elect, elocation, transition, elbutton, knockspopover, knocks button, knocks player.

13. knocksplayer.vue

- Brief Description

This component is used for running audio sources and validate its type according to the browser and plays base 64 blob and gives the option to the developer to convert the URL then run it or run it only and acknowledge the user if he has compatibility problems or loading for audio source.

- Props

The props : source, duration, live, gid, specifications, fill from, meta ,load on mount, show options, runtime, hide sniper, show volume, live meta, full back loading, not web, CSS classes.

- API Events

Event API global: knocksGlobalPlayerRequested (has the gid, and when the gid does not match pauses so it toggles between players), running record (if a player is running and the user is recording so it pauses).

Event API global : liveMetaReady (acknowledge the component that live meta lie data ready in case the user runs live source), live loading done (when the audio source is finished).

DOM events :mouse over (shows interest), ended (resets the player after it finishes), loaded meta data (gets the runtime), loaded date (acknowledge both of users and components to resume the player), at waiting (indicates that there are new

chunks to be loaded) , range changed (updates the component timer and updates the requested chunks if there any).

- Used Components

It uses components : elpopover, elbutton, elslider.

14. knocksreactor.vue

- Brief Description
- Props
- API Events
- Used Components

15. knock.vue

- Brief Description
- Props
- API Events
- Used Components

16. knocksmultipleuploader.vue

- Brief Description

This component is used for taking the multiple file as input and maps them according to size, type and name and validates the type and size and the duplication and upload them remotely.

- Props

gid, images_url, regular_files_url, scope, maximum_size, and css customization.

- API Events

knocks_submit (takes scope and determines if this component submits or not), knocks_multiple_uploader_reset (gets the scope and determines if it resets itself or not and it resets as it empties the files and gets back to the normal state), knocks_multiple_uploader_trigger (switches the dragging mode on and off), knocks_multiple_uploader_dragging (activates the dragging mode for user actions), knocks_media_query_logged (it has scope, token, query, has files, has images as payloads and it considers the multiple uploader state if it has files and pictures or not).

Local events : media_query_counter (counts files gets uploaded)

- Used Components

It uses the previously made components : knocksimageditor, knocks popover.

17. knocksimageditor.vue

- Brief Description

This component is used for editing images and performs the cropping.

- Props

gid, crops

- API Events

- Used Components

18. knockscirclechip.vue

- Brief Description

This component is used for represents the circle name and icon and also retrieving a model that retrieves the circle index.

- Props

circle, circle_url, circle_background_color, only_icon, popover, no_rebound, as_list, toggled, show_key.

- API Events

Global events API : knockscirclechip (payloads : circle and if it matched with the circle id it bents the circle data), knockscirclekeyupdate (it does the same function of knockscirclekeyupdate), knockscircleaddmember (payloads : circle, member and updates the circle model to add a member programmatically), knockscircleremovemember (the same as add member but it removes).

- Used Components

19. knocksmultipleswitch.vue

- Brief Description

This component is used for giving options for the user to choose between the privacy sets for users and circles as dropdown menu

and toggle button and it can be showing the options as icons or static messages.

- Props
 - options, static messages, send code, receive code, start up value.
- API Events
 - kncoks_switch_change (payloads : focus, value and it updates the value).
- Used Components

20. knockspopover.vue

- Brief Description
 - This component is used for adding details as a tooltip
- Slots
 - container (what is shown in pop over), content (what is to be shown)

21. knocksknock.vue HESHAM

- Brief Description
- Props
- API Events
- Used Components

22. knocksccomment.vue ABASIRY

- Brief Description
- Props
- API Events
- Used Components

23. knockscchildreply.vue

- Brief Description
 - This component is used for giving banning some inappropriate actions from the child to see it.
- Props

Props : knock, current user, parent comments, parent object is, gid, knock type, as short cut, show reply on mount.

- API Events
knocks_show_interest (activates the interest mode).
- Used Components

24. knocksreply.vue

- Brief Description
- Props
- API Events
- Used Components

25. knocksreactionstats.vue

- Brief Description
- Props
- API Events
- Used Components

26. knockstaps.vue

- Brief Description
This component is used for representing options as radio button and check boxes and it can be filled remotely and can translate with the static messages
- Props
options, multiple, define first, define with index, scope, is required, radio unset, unlabeled, untooltiped, CSS customizations.
- API Events
knocks_submit (explained before) kncks_changed_taps_value (payload : scope and value and it changes its value).
- Used Components
None

27. knocksfileviewer.vue

- Brief Description

This component is used for representing the file with the name and icon according to it's url

- Props
file, CSS customization
- API Events
None
- Used Components
None

28. knocksballon.vue

- Brief Description
This component is used for representing the balloons and notifications and makes them as seen in the database based on the user interaction.
- Props
gid, constraints, keepShowing, mute, hideReplies, showBrowserNotifications, extended, index time.
- API Events
knocks_stop_timer, knocks_return_timer (for starting and stopping the timer), knocks_ballon_global_seen .
- Used Components
This makes use of previously made components : knocksretreiver, knocks user, static messages, knocksuseractions, knocks collapse, knocks showkeys, knocks knock, knockscomment, knocks child reply, knocks image viewer.

29. Knocksimageviewer.vue

- Brief Description
- Props
- API Events
- Used Components

30. knocksuseractions.vue

- Brief Description
This component is for responding for a friend request and accept a friend request
- Props
user, start as, show accept shortcut, extended, add remove only, extras
- API Events
knocks global circle list (assigns the main circle object to be seen to the old components), knocks user reset content (reset contents for the user actions).
- Used Components
None

31. knockssqladdcircle.vue

- Brief Description
- Props
- API Events
- Used Components

32. knocksselbutton.vue

- Brief Description
- Props
- API Events
- Used Components

33. knocksknockinjector.vue

- Brief Description
- Props
- API Events
- Used Components

34. knockssqlcareers.vue

- Brief Description
This component is for viewing the user's career and it has the fields, works at, works as, works since, works to, and it also has the previously made date picker component.for the works since and

works to cannot be before his birthdate and also the works to cannot be before his works since date

- Props
- API Events
- Used Components
 - using the previously made component knocks selector.

35. knocksusereducation.vue

- Brief Description

This component is for viewing the user's education and it has the fields, study at, study what, grade, study since, study to, and it also has the previously made date picker component. for the study since and study to cannot be before his birthdate and also the study to cannot be before his study since date
- Props
- API Events
- Used Components
 - using the previously made component knocks selector

36. knocksuserhigheducation.vue

- Brief Description

This component is for viewing the user's education and it has the fields, study at, study what, grade, study since, study to, and it also has the previously made date picker component. for the study since and study to cannot be before his birthdate and also the study to cannot be before his study since date
- Props
- API Events
- Used Components
 - using the previously made component knocks selector

37. knocksuserhobby.vue

- Brief Description
This component is for viewing the user's hobby
- Props
- API Events
- Used Components
using the previously made component knocks selector

38. knocksusersport.vue

- Brief Description
This component is for viewing the user's sport
- Props
- API Events
- Used Components
using the previously made component knocks selector

39. knocksuserabout.vue

- Brief Description
Knocks user about is a component that show the user information such as : General Info, Career, Education, High Education, Hobby and Sports by using other components (knocksuser, knocksusercareer, knocksusercareer, knocksusereducation, knocksuserhigheducation, knocksuserhobby and knocksusersport) We are showing the data in tabs and in each tab has it's own Add new, edit and delete tool except General, only the owner who can edit or add or delete his own Information other users can only see the information without doing any action on it.
- Props
- API Events
- Used Components

40. knocksretriver.vue

- Brief Description

Knocks retriever is a component that performs an XHR requests using ‘axios’ package, The component is designed to make performing XHR requests easier and richer, the developer can take the benefit of many events without writing a painful code, it only requires a url then the component can take the rest to provide the developer by all the events he may need in details.

- Props
- API Events
- Used Components

41. knocksvoicerecognition.vue

- Brief Description
- Props
- API Events
- Used Components

42. knockscroppie.vue

- Brief Description

This component is used for cropping n image as it takes the aspect ration to use for cropping, and also takes the area where this picture is uploaded, and the picture information, also it internally takes the error messages and call back events and the scope, the package crappie is used to assist for this component service.

- Props
- API Events
- Used Components

43. knocksuseraboutedit.vue

- Brief Description

Knocks user about edit is a component that select the demand index and start do editing on it

- Props
- API Events
- Used Components

44. knocksusersportedit.vue

- Brief Description

Knocks user about edit is a component that select the demand index and start do editing on it .
- Props
- API Events
- Used Components

45. knocksuserhobbyedit.vue

- Brief Description

Knocks user about edit is a component that select the demand index and start do editing on it .
- Props
- API Events
- Used Components

46. knocksusereducationedit.vue

- Brief Description

Knocks user education edit is a component that select the demand index and start do editing on it .
- Props
- API Events

- Used Components

47. knocksusercareeredit.vue

- Brief Description

Knocks user career edit is a component that select the demand index and start do editing on it .

- Props
- API Events
- Used Components

48. knocksuseraboutdelete.vue

- Brief Description

Knocks user about delete is a component see the user data if it was career, education, high education, hobby or sport and select the index that will be deleted .

- Props
- API Events
- Used Components

49. knocksgroupcreation.vue

- Brief Description

This component is for group creation and it has fields group name, category and it has user searching to add them to that group, and it has the privacy setting for this group like public, closed or secret, and it has the option all so that this group can add all the users

- Props

show-toggler it's type is a boolean, so if the prop is true then it uses the button to display the component, and if it is false it responds to an event called knocksgroupcreationToggle and it takes the boolean payloads .

- API Events
- Used Components

50. knocksgroupslist.vue

- Brief Description

This component is for listing the user groups that he is joined in, and its also for searching the groups with the search key to filter them
- Props

it responds for events called knockspushNewGroup and it's payload is group id.
- API Events
- Used Components

it uses the previously made components knocksgroupshortcut and knocksshowkeys

51. knockspagesearch.vue

- Brief Description

This component is for searching in all, people, knocks, groups sections with the search key, and by default it searches and views the search result in the all section and in case there was a desired separation you can look in the other sections, in all section it views the results in people, knocks, groups sections respectively
- Props
- API Events
- Used Components

it has a previously made components knocksshortcut, knocksshowkeys, knocksshowpeople, knocksshowknock, knocksshowgroups.

52. knocksgroupmembers.vue

- Brief Description

This component is for viewing the group's group info, group members, adding members, group photos, group files, group voices and group videos.

- Props

groupObject which takes the group object to get the group name, group category, group preset and group date creation.

- API Events

- Used Components

it uses the previously made components knockssusershortcut, knocksgroupmemberdelete, knocksrotriver, knocksfleviewer, knockssusershortcut, knockspayer, knocksimg and knocksgroupjoining .

53. knocksimg.vue

- Brief Description
- Props
- API Events
- Used Components

54. knockschattingzone.vue

- Brief Description
- Props
- API Events
- Used Components

55. knockscconversation.vue

- Brief Description
- Props
- API Events
- Used Components

56. knocksmessagesender.vue

- Brief Description
- Props
- API Events
- Used Components

57. knocksgroupshortcut.vue

- Brief Description

This component is for viewing the group in various patterns, it can be showed as chip, dialog, result, URL and label

- Props

group_id (type : number), as_chip (boolean), as_dialog (boolean), as_result (boolean), as_url (boolean), as_label (boolean).

- API Events

- Used Components

it uses the previously made component knockstriver(to retrieve the group info or group attributes), knocksimg (to show group image) and knocksgroupjoining (to show the button of showing the group or leaving the group or get into the group).

58. knocksgroupjoining.vue

- Brief Description

This component is actually a button and is for including the user in many patterns according to his privileges and to the group privacy as a result or adding member mode.

- Props

group_id (type : number), as_result (boolean), add_member_mode (boolean), user_id (number), as_owner(boolean).

- API Events

- Used Components

It also uses the previously made component knockstriver (to check the group request and sends a request to join the group), and who clicking that button it sends a message that you asked to join the group, and you can open that group if you are a member of it, also it can be for joining the group if it is a public group.

59. knocksgroupmemberdelete.vue

- Brief Description

This component which is a button is for removing the members of the group (kick) from it by the admin and also the user can leave himself (leave) .

- Props

Props : group_id (number), member_delete (number), gid (number), position (string), authposition (string).

- API Events

knocksContentChange (it emits an event if one member is deleted to refresh the member list)

- Used Components

It also uses the previously made component knocksretriver (for retrieving the group members and exclude the group member from the database too and if that group has one member and leaving action takes place then the group is deleted), and knocksgroupmember position.

60. knocksgrouppictures.vue

- Brief Description

This component is for viewing the group pictures.

- Props

Props : group_id (number).

- API Events

- Used Components

It also uses the previously made component knocksimg (to show the group picture)

61. knocksgroupfiles.vue

- Brief Description

This component is for viewing the group files.

- Props

Props : group_id (number).

- API Events

- Used Components

It also uses the previously made component knocksfileviewer (to show the group files).

62. knocksgroupvoices.vue

- Brief Description

This component is for viewing the group voices.

- Props

Props : group_id (number).

- API Events

- Used Components

It also uses the previously made component knocksplayer(to show the group voices).

63. knocksrightbar.vue

- Brief Description

- Props

- API Events

- Used Components

64. knocksprivacyadjustments.vue

- Brief Description

- Props

- API Events

- Used Components

65. knocksgroupsettings.vue

- Brief Description

This component which is tabs and is for editing the group name , group category or the group picture in the group information, and it also has the group privacy for editing the group privacy in three options secret, public or closed, and it also the manage member position to mange the positions of owner and admin, also it receives the group requests and it can has two responses: accept or decline.

- Props
Props : group_object (which contains all the target groups attributes)
- API Events
- Used Components
It also uses the previously made component knockstriver (retrieves the group members and the group requests), knockscropie (for changing the group picture), knocksgroupjoining (in the partition of group requests to let the admin or owner to accept or decline the request), knockssusershortcut (to list the group members as chips), and knocksmemberposition (to set the group member positions by owner).

66. knocksuserinfo.vue

- Brief Description
This component is used for viewing profile settings for the user, for the fields where the user didn't add when registering like biography ,phone, religion, marital status, orientation , so he can add it and edit and delete it later, and for other registration mandatory fields he can also add it and edit it, also for the display name of the user he can show his name for other people the way he likes, and it has its own restrictions, and some fields can be chosen from choices or can be written freely by the user to be added by him as an option.
- Props
- API Events
- Used Components
date picker component

67. knocksgroupmemberposition.vue

- Brief Description
This component is for checking the user position so the privileges can be tailored for him, also in case of the owner intends to leave

the group that he owns then this component displays a dialog for him to ask him to transfer his ownership to another user which he desires.

- Props
 - user_id (number), group_id (number), dialog (boolean).
- API Events
- Used Components
 - It also uses the previously made component knockstriver (to retrieve the group member positions)

68. knockshash.tag.vue

- Brief Description
- Props
- API Events
- Used Components

69. knockscircles.user.vue

- Brief Description
- Props
- API Events
- Used Components

70. knockscircle.adder.quick.vue

- Brief Description
- Props
- API Events
- Used Components

71. knockshash.tagchip.vue

- Brief Description
- Props
- API Events
- Used Components

72. knockscircle.states.image.vue

- Brief Description

- Props
- API Events
- Used Components

73. knocksuserinfodelete.vue

- Brief Description

This component is used for deleting profile settings for the user, for the fields where the user didn't add when registering like biography ,phone, religion, marital status, orientation , so he can delete them if there were not null by clicking the see more button.

- Props
- API Events
- Used Components

74. knocksphotocomments.vue

- Brief Description
- Props
- API Events
- Used Components

75. knocksusergenralinfo.vue

- Brief Description

This component is used for viewing the profile settings for the user, for the fields where the user didn't add when registering like biography ,phone, religion, marital status, orientation , so he can view them if there were not null.

- Props
- API Events
- Used Components

76. knocksdepimgviewer.vue

- Brief Description
- Props

- API Events
- Used Components

77. knockscirclemembers.vue

- Brief Description
- Props
- API Events
- Used Components

78. knockscollapse.vue

- Brief Description
- Props
- API Events
- Used Components

79. knocksmediaobject.vue

- Brief Description
- Props
- API Events
- Used Components

80. knocksshowkeys.vue

- Brief Description
- Props
- API Events
- Used Components

81. knocksrecordermpeg.vue

- Brief Description
- Props
- API Events
- Used Components

82. knocksvoicecommands.vue

- Brief Description
- Props
- API Events
- Used Components

83. knocksquickaddress.vue

- Brief Description
- Props
- API Events
- Used Components

84. knocksaddressviewer.vue

- Brief Description
- Props
- API Events
- Used Components

85. knockseldatepicker.vue

- Brief Description
- Props
- API Events
- Used Components

86. knockselect.vue

- Brief Description
- Props
- API Events
- Used Components

87. knocksusersuggestions.vue

- Brief Description
- Props
- API Events
- Used Components

88. knocksusersettings.vue

- Brief Description
- Props
- API Events
- Used Components

89. knockswatchmywindow.vue

- Brief Description
- Props

- API Events
- Used Components

90. knocksblockuser.vue

- Brief Description
- Props
- API Events
- Used Components

91. knocksblockuserlist.vue

- Brief Description
- Props
- API Events
- Used Components

92. knocksverifyuser.vue

- Brief Description
- Props
- API Events
- Used Components

93. knockstips.vue

- Brief Description
- Props
- API Events
- Used Components

94. knocksshowdevice.vue

- Brief Description
- Props
- API Events
- Used Components

95. knocksdateviewer.vue

- Brief Description
- Props
- API Events
- Used Components

96. knockseditcirclename.vue

- Brief Description
- Props
- API Events
- Used Components

97. knockswelcomeslider.vue

- Brief Description
- Props
- API Events
- Used Components

98. knocksforgotmypassword.vue

- Brief Description
- Props
- API Events
- Used Components

99. knockssitemap.vue

- Brief Description
- Props
- API Events
- Used Components

100. knocksprivacytips.vue

- Brief Description
- Props
- API Events
- Used Components

101. knocksmeganumber.vue

- Brief Description
- Props
- API Events
- Used Components

102. knockstrendslist.vue

- Brief Description
- Props

- API Events
- Used Components

103.knocksgenderviewer.vue

- Brief Description
- Props
- API Events
- Used Components

6. Testing

6.1. Testing Strategies

6.1.1. Black-Box Testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It is sometimes referred to as specification-based testing.

6.1.2. White-Box Testing

White-box testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.

6.1.3. Gray-Box Testing

Gray box testing, also called gray box analysis, is a strategy for software debugging in which the tester has limited knowledge of the internal details of the program. A gray box is a device, program or system whose workings are partially understood.

6.1.4. Unit Testing

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

6.1.5. Integration Testing

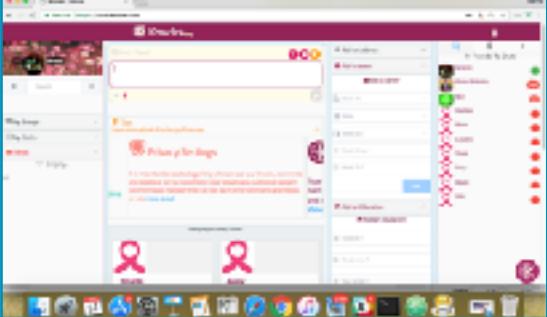
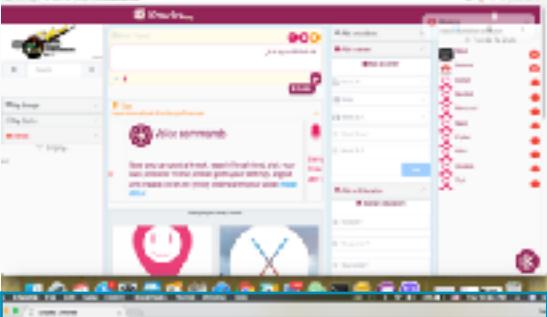
Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

6.1.6. System Testing

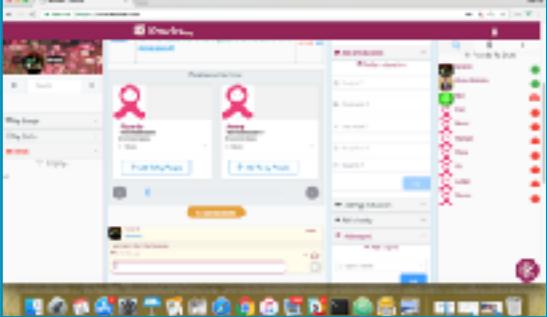
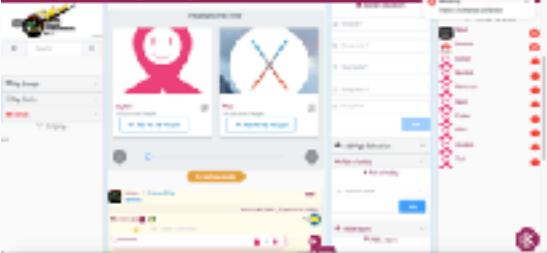
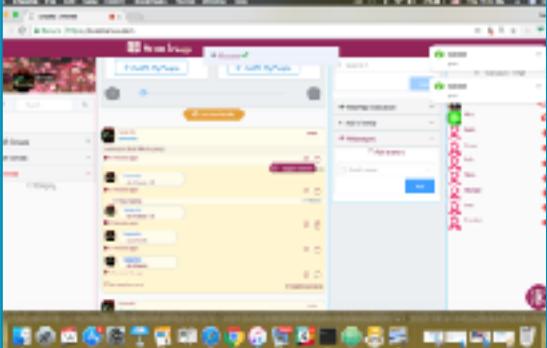
System Testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

6.2. Test Cases

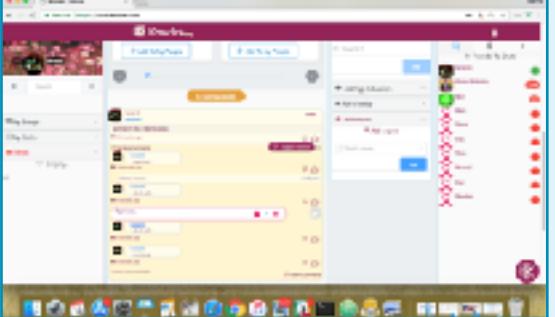
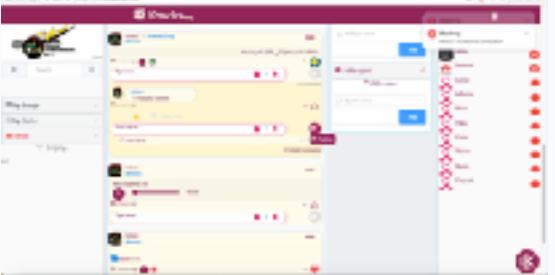
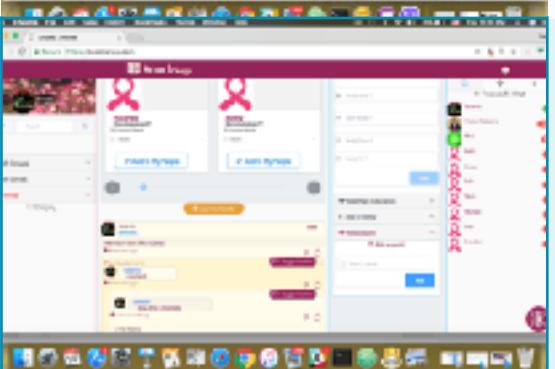
- Publish a Knock

Input	Expected Output	Actual Output	Comment
Knock content : Empty	Disabled Knock Button		Success
Knock content : valid But there is no internet connection.	Disabled Knock Button, "There is no Internet Connection"		Success
Knock content : valid	Enabled Knock Button, "Knock is Published successfully".		Success

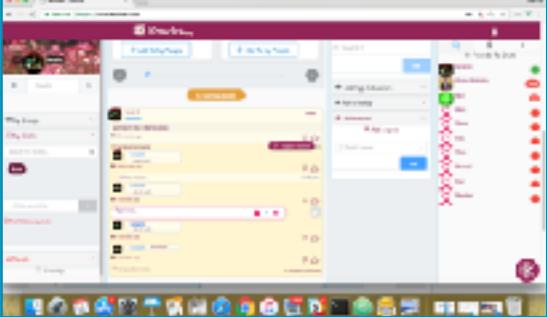
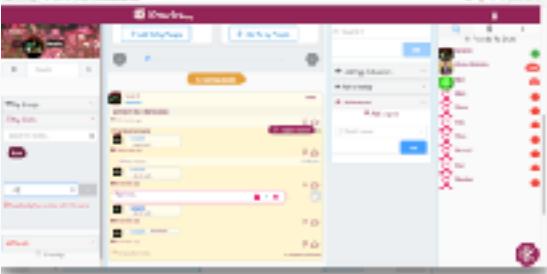
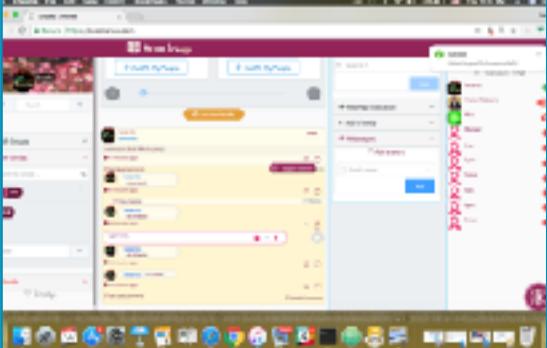
- **Write a comment**

Input	Expected Output	Actual Output	Comment
Comment content : Empty	Disabled "publish comment" Button		Success
Comment content : valid, "There is no Internet connection."			Success
Comment content : valid	Comment is added to the post, "Success".		Success

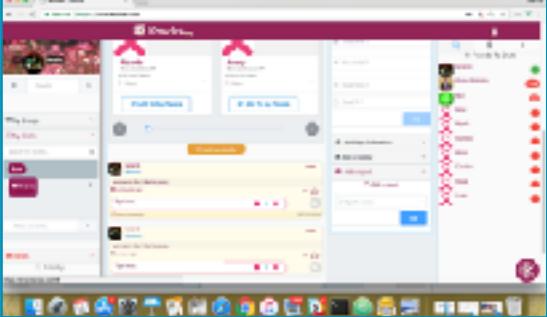
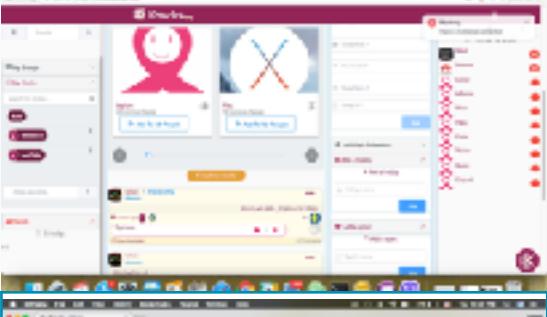
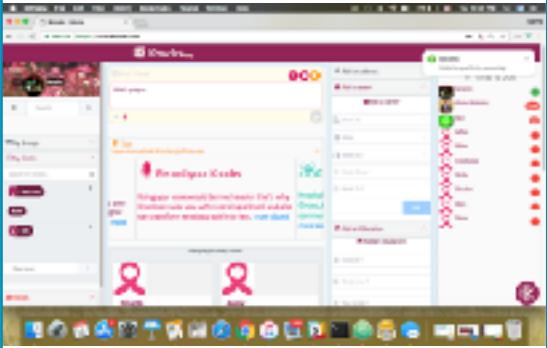
- Write a reply

Input	Expected Output	Actual Output	Comment
Reply content : Empty	Disabled "publish reply" Button		Success
Reply content :valid, but "There is no Internet there is not internet Connection" connection.			Success
Reply content :valid	Reply is added to the comment, "Success".		Success

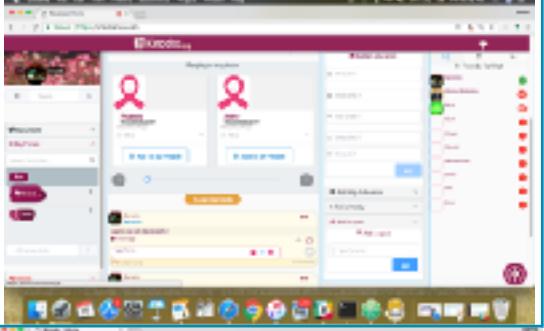
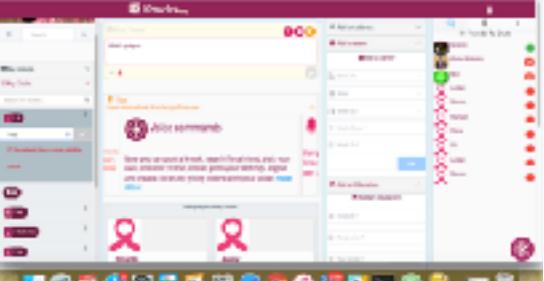
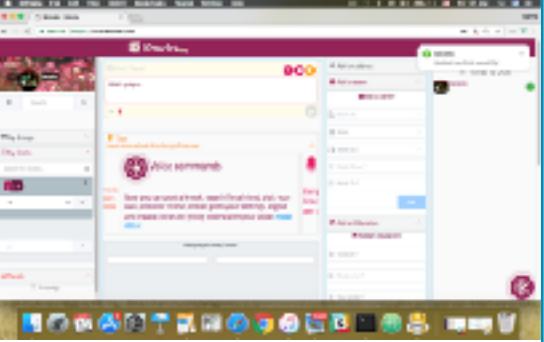
- **Create a circle.**

Input	Expected Output	Actual Output	Comment
Circle Name : Empty	Disabled adding circle button, "This field is required"		Success
Circle Name :invalid (already existing circle name)	Disabled adding circle button, "You already have a circle with this name"		Success
Circle Name :valid (A new circle name)	"Added to your circles successfully"		Success

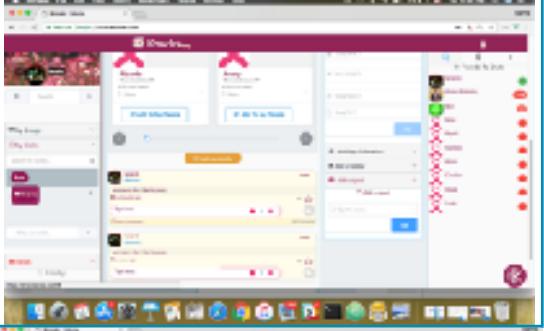
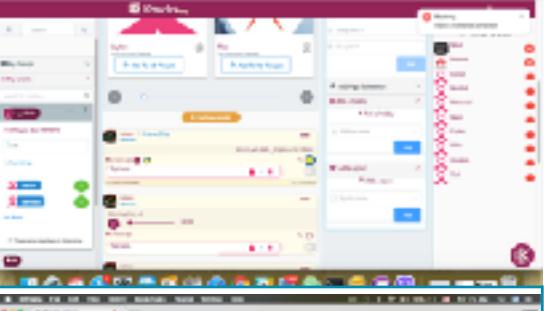
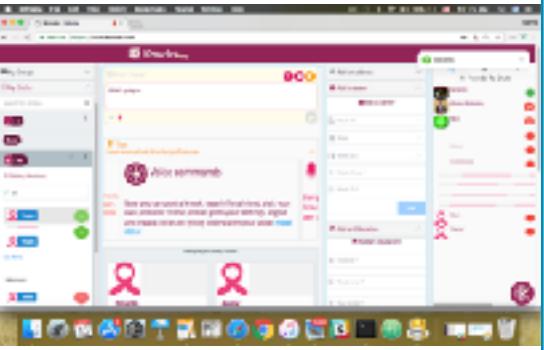
- **Delete a circle**

Input	Expected Output	Actual Output	Comment
Selected Circle : invalid (All circle is selected)	Disabled deleting circle button		Success
Selected Circle : valid (a "There is no Internet connection" created before except for all circle, but there is no internet connection)			Success
Selected Circle : a circle "Selected Circle has been created successfully"			Success

- **Update circle name**

Input	Expected Output	Actual Output	Comment
Selected Circle : invalid (All circle is selected)	Disabled editing circle name button		Success
Selected Circle : invalid (An already existing circle name)	Disabled editing circle name button, "You already have a circle with this name"		Success
Selected Circle : valid (A circle that has been successfully created before except for all circle)	"Updated your circle successfully"		Success

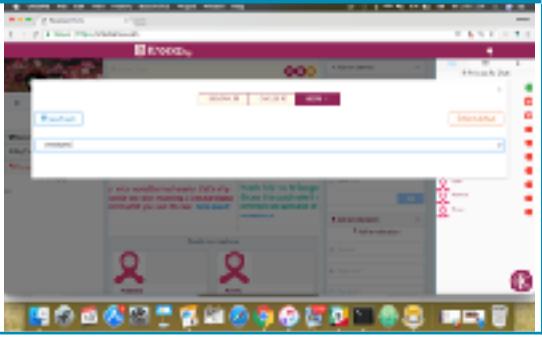
- **Include a friend to a circle**

Input	Expected Output	Actual Output	Comment
Selected Circle : invalid (All circle is selected) Selected User :invalid/ Empty	Disabled editing members to circle		Success
Selected Circle : valid (A circle that has been created before except for all circle, but there is no internet connection) Selected User : valid	"There is no Internet Connection"		Success
Selected Circle : valid (A "success" circle that has been created before except for all circle) Selected User : valid	"success"		Success

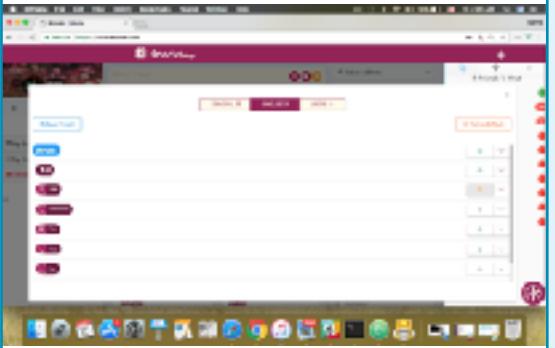
- **Exclude a friend from a circle**

Input	Expected Output	Actual Output	Comment
Selected Circle : invalid (All circle is selected) Selected User :invalid/ Empty	Disabled editing members to circle		Success
Selected Circle : valid (A "There is no Internet connection" message is displayed) Selected User : valid	"There is no Internet connection" message is displayed		Success
Selected Circle : valid (A "Success" message is displayed) Selected User : valid	"Success" message is displayed		Success

- Regulate a privacy set for users

Input	Expected Output	Actual Output	Comment
Selected User : invalid (no selection/blocked user) Privacy switch : Valid	No action is triggered / "Please select a user"		Success
Selected User :valid Privacy switch : Valid but there is no internet connection	"There is no Internet Connection"		Success
Selected User :valid Privacy switch : Valid	Changed privacy set for the selected user with the selected privacy set		Success

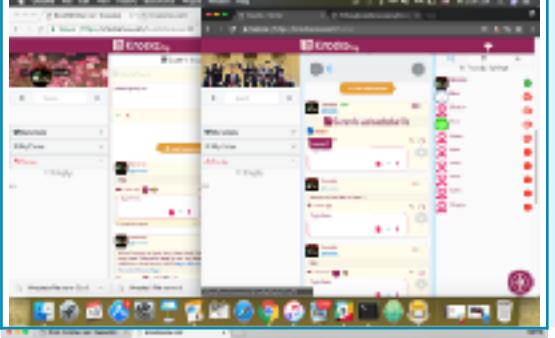
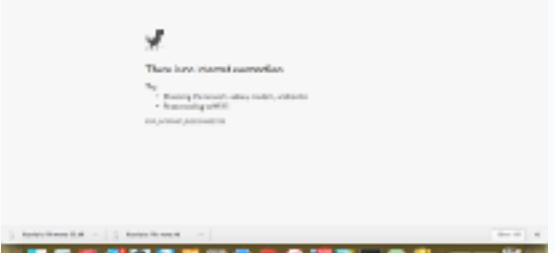
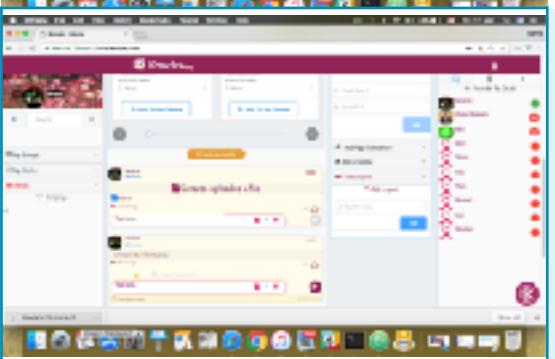
- Regulate a privacy set for circles

Input	Expected Output	Actual Output	Comment
Selected Circle : invalid (no selection/deleted circle) Privacy switch : Valid	No action is triggered / "Please select a circle"		Success
Selected Circle :valid Privacy switch : Valid but there is no internet connection	"There is no Internet Connection"		Success
Selected Circle :valid Privacy switch : Valid	Changed privacy set for the selected circle with the selected privacy set		Success

- Regulate a parent object privacy set for circle. //

Not yet a ready function

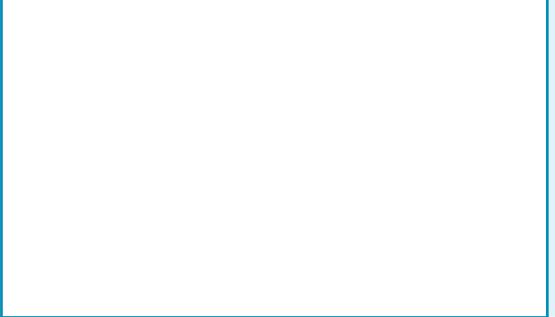
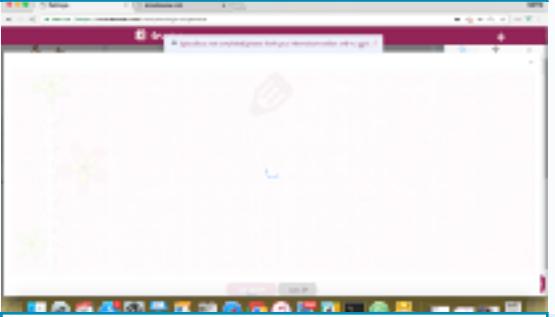
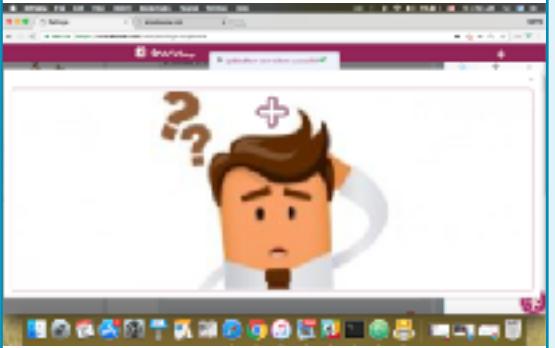
- Download files (with any extension).

Input	Expected Output	Actual Output	Comment
Selected File : invalid (a “Invalid” deleted file)			Success
Selected file : valid But there is no internet connection	“No Internet Connection”		Success
Selected File :valid	The file is being downloaded and saved at the user’s computer		Success

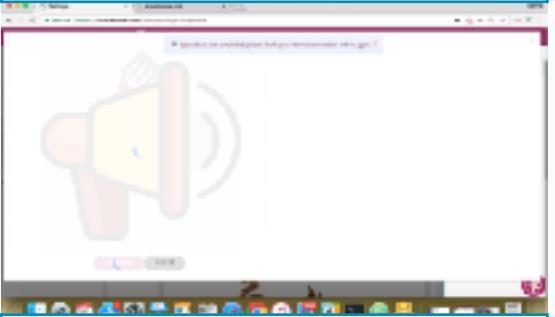
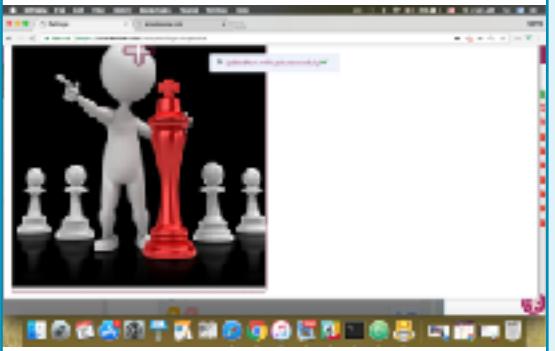
- **Upload profile picture**

Input	Expected Output	Actual Output	Comment
Selected file : invalid	"Cannot use the selected object as a profile picture"		Success
Selected file : valid But there is no internet connection	"Upload was not completed, Please check your internet connection and try again."		Success
Selected file :valid	"updated your profile picture successfully"		Success

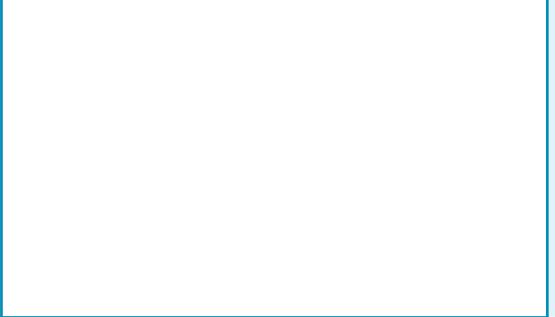
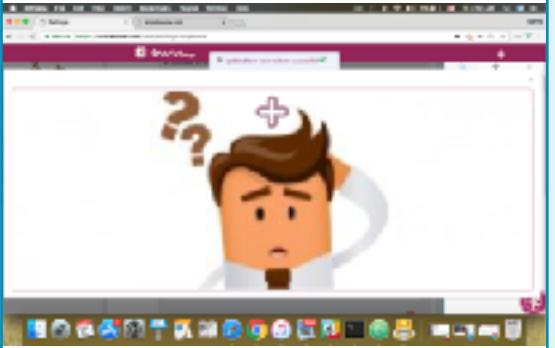
- **Upload cover picture**

Input	Expected Output	Actual Output	Comment
Selected file : invalid	“Cannot use the selected object as a cover picture”		Success
Selected file : valid But there is no internet connection	“Upload was not completed, Please check your internet connection and try again.”		Success
Selected file :valid	“updated your cover picture successfully”		Success

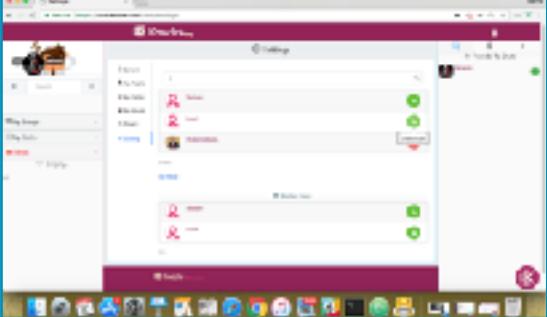
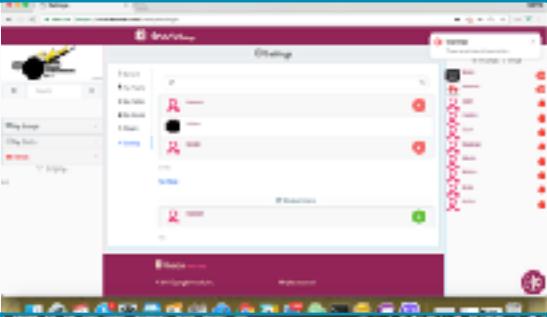
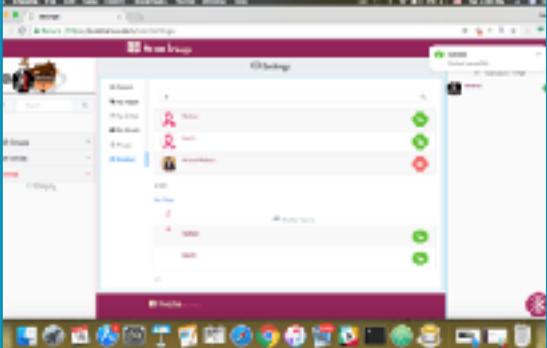
- **Update profile picture**

Input	Expected Output	Actual Output	Comment
Selected file : invalid	"Cannot use the selected object as a profile picture"		Success
Selected file : valid But there is no internet connection	"Upload was not completed, Please check your internet connection and try again."		Success
Selected file :valid	"updated your profile picture successfully"		Success

- **Update cover picture**

Input	Expected Output	Actual Output	Comment
Selected file : invalid	"Cannot use the selected object as a cover picture"		Success
Selected file : valid But there is no internet connection	"Upload was not completed, Please check your internet connection and try again."		Success
Selected file :valid	"updated your cover picture successfully"		Success

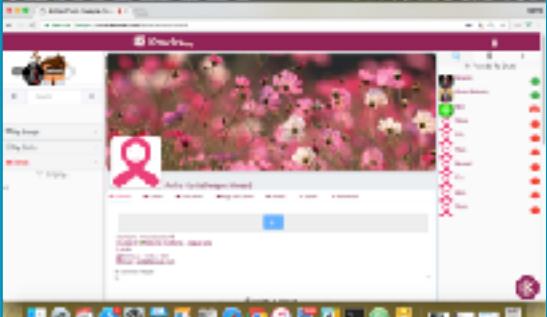
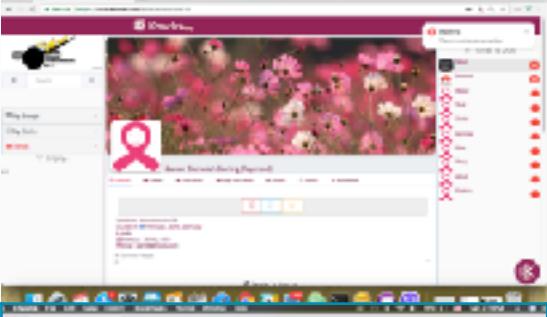
- **Block a friend**

Input	Expected Output	Actual Output	Comment
Selected User : invalid (already blocked users)	"You can only unblock this user"		Success
Selected User : valid But there is no internet connection	"Please check your internet connection"		Success
Selected User :valid	" Blocked successfully"		Success

- Report a friend

Not yet a ready function

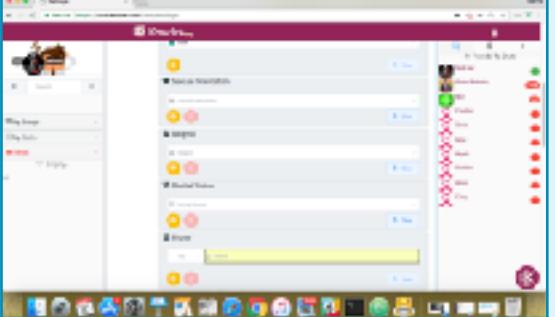
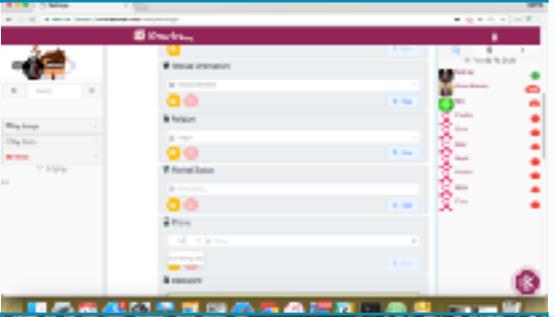
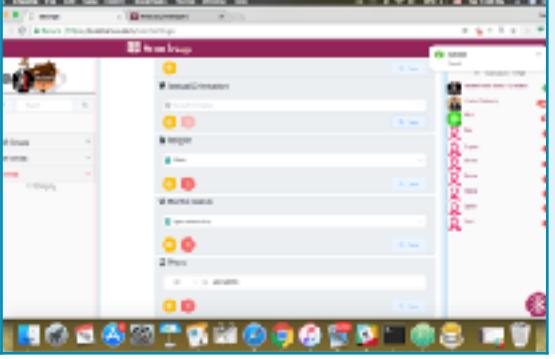
- **Unfriend a friend**

Input	Expected Output	Actual Output	Comment
Selected User : invalid user	"This User is already not a friend of you", only "Add a friend" button is shown		Success
Selected User : valid connection	"There is no Internet Connection"		Success
Selected User :valid	"This user is unfriended successfully"		Success

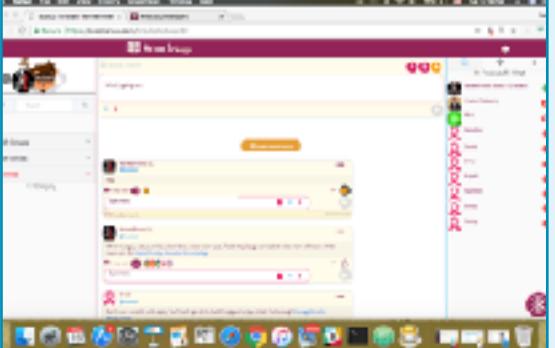
- Suggest a content (in case of adult user to his kid account).

Not yet a ready function

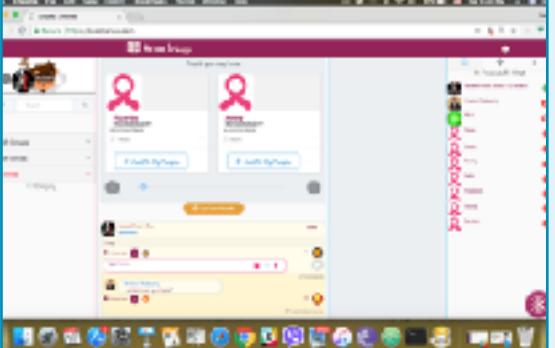
- Update personal information (Eg :Phone number)

Input	Expected Output	Actual Output	Comment
Key : Empty Phone : valid	Disabled updating button		Success
Key : Invalid Phone : valid	"Not matching data"		Success
Key : Valid Phone : Valid	"Saved"		Success

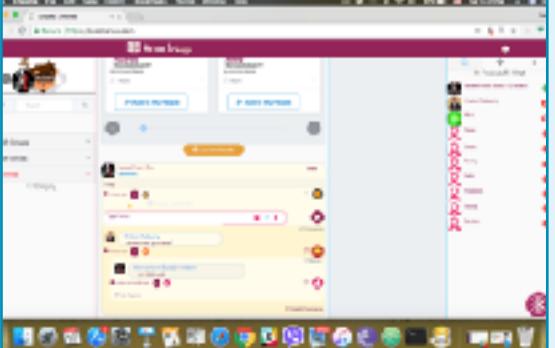
- **React to post**

Input	Expected Output	Actual Output	Comment
Selected Post : invalid Reaction Type : valid	"cannot react to post, it is not available anymore "		Success
Selected Post : valid Reaction Type : valid But there is no internet connection	"There is no Internet Connection"		Success
Selected Post : invalid Reaction Type : valid	Displays your reaction across the post		Success

- **React to comment**

Input	Expected Output	Actual Output	Comment
Selected Comment : invalid Reaction Type : valid	: "cannot react to comment, it is not available anymore "		Success
Selected Comment : valid Reaction Type : valid But there is no internet connection	"There is no Internet Connection"		Success
Selected Comment : valid Reaction Type : valid	"Your reaction is added to this comment"		Success

- **React to reply**

Input	Expected Output	Actual Output	Comment
Selected Reply : invalid Reaction Type : valid	"cannot react to reply, it is not available anymore"		Success
Selected Reply : valid Reaction Type : valid But there is no internet connection	"There is no Internet Connection"		Success
Selected Reply : valid Reaction Type : valid	"Your reaction is added to this reply"		Success

- Chat with a friend

Not yet a ready function

- Create a group chat

Not yet a ready function

- Update a group chat - when allowed to do so.

Not yet a ready function

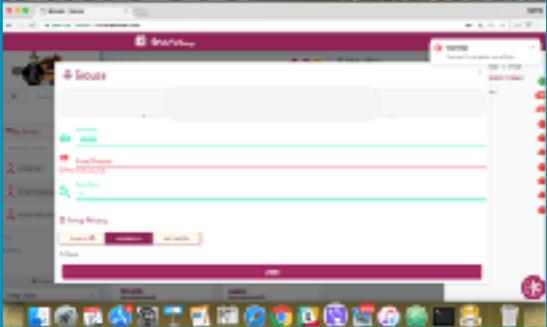
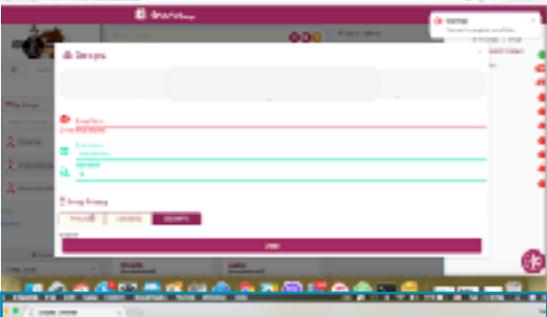
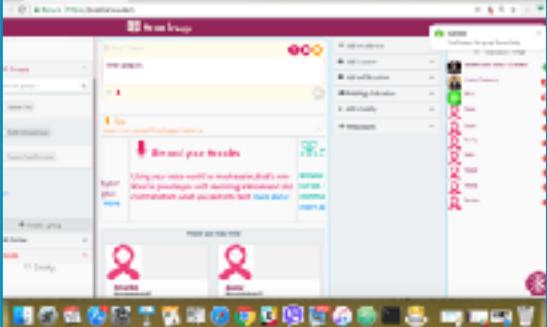
- Delete a group chat- when allowed to do so.

Not yet a ready function

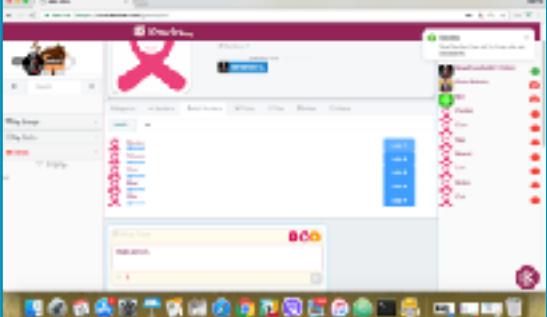
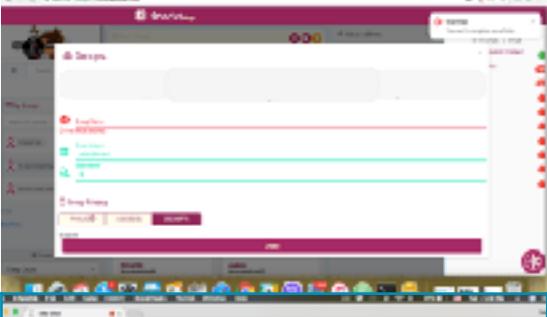
- React to messages

Not yet a ready function

- **Create a group**

Input	Expected Output	Actual Output	Comment
Group Name : valid Group Category : empty Search Name : valid Group Privacy : valid	"You need to complete some fields"		Success
Group Name : empty Group Category : valid Search Name : valid Group Privacy : valid	"You need to complete some fields"		Success
Group Name : valid Group Category : valid Search Name : valid Group Privacy : valid	"You created the group successfully"		Success

- **Add friends to group**

Input	Expected Output	Actual Output	Comment
User Position : Invalid (not a group owner) Group Name : Valid Selected Member : Valid	Not displaying "Invite Members" button		Success
User Position : Valid Group Name : Valid Selected Member : Valid, But there is no internet connection	"There is no Internet Connection"		Success
User Position : Valid Group Name : Valid Selected Member : Valid	"New member has add to group 'group name' successfully"		Success

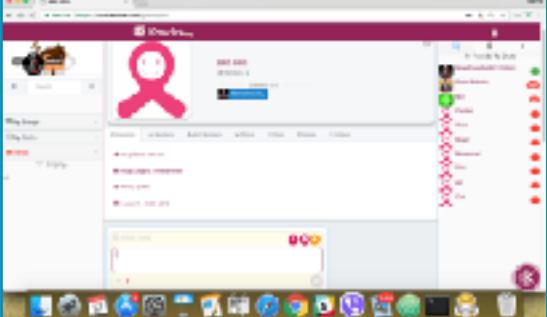
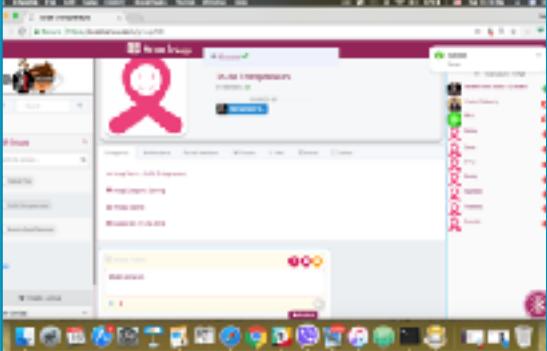
- **Update a group**

Input	Expected Output	Actual Output	Comment
Group Name : valid Group Category : empty User Position : Invalid	Not displaying group information updating section		Success
Group Name : valid Group Category : empty User Position : Valid			Success
Group Name : valid Group Category : valid User Position : valid	"You updated the group successfully"		Success

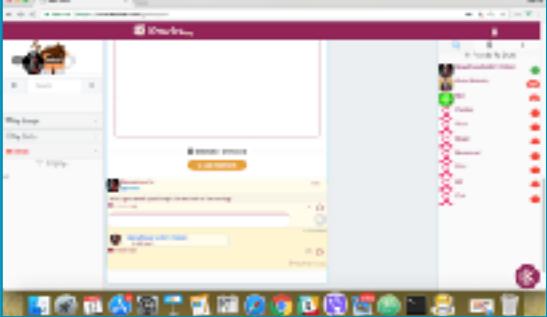
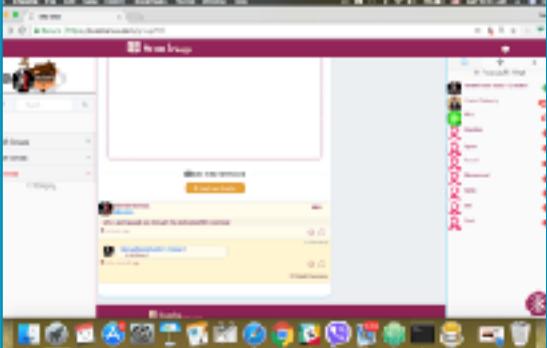
- Delete a group

Not yet a ready function

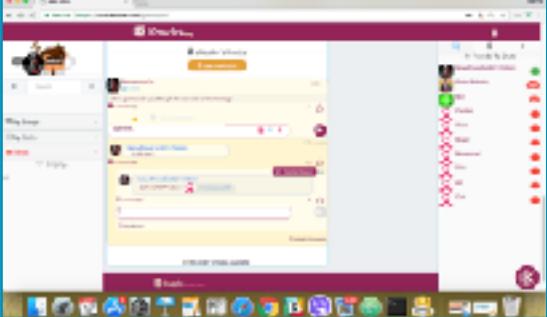
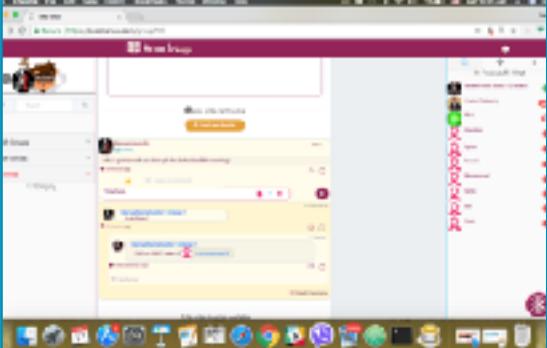
- **Post in a group**

Input	Expected Output	Actual Output	Comment
Selected Group : Valid Knock Content : Empty User Position : Valid	Disabled “Publish” Button		Success
Selected Group : Invalid (The group is already deleted or the user has been kicked out from the group) Knock content : Valid User Position : Valid			Success
Selected Group : Valid Knock content : Valid User Position : valid	Enabled Publish Button, “Success”.		Success

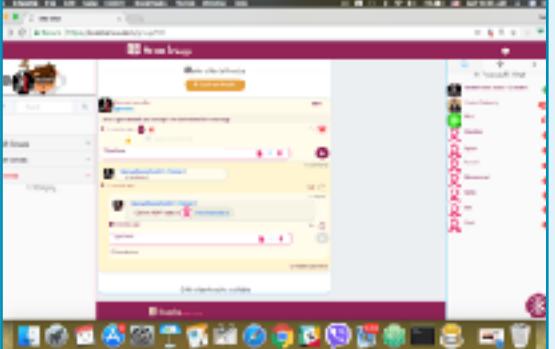
- **Comment in a group**

Input	Expected Output	Actual Output	Comment
Selected Group : Valid Comment Content : Empty User Position : Valid	Disabled “Publish” Button		Success
Selected Group : Invalid (The group is already deleted or the user has been kicked out from the group) Comment content : Valid User Position : Valid			Success
Selected Group : Valid Comment content : Valid User Position : valid	Enabled Publish Button, “Success”.		Success

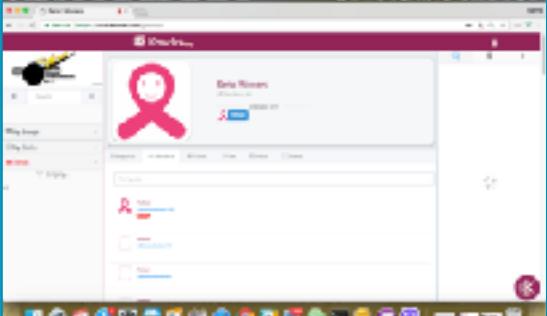
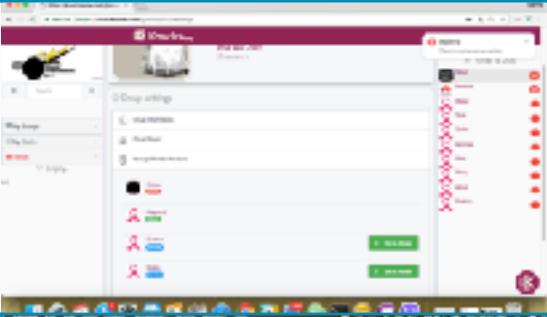
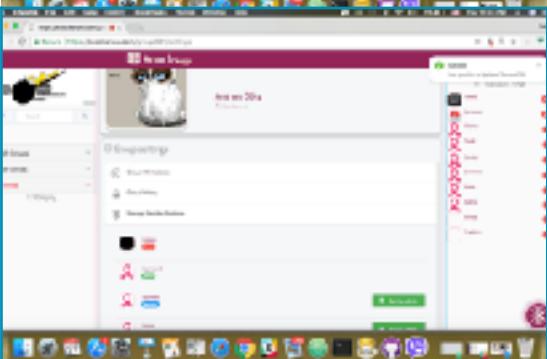
- **Reply in a group**

Input	Expected Output	Actual Output	Comment
Selected Group : Valid Reply Content : Empty User Position : Valid	Disabled “Publish” Button		Success
Selected Group : Invalid (The group is already deleted or the user has been kicked out from the group) Reply content : Valid User Position : Valid			Success
Selected Group : Valid Reply content : Valid User Position : valid	Enabled Publish Button, “Success”.		Success

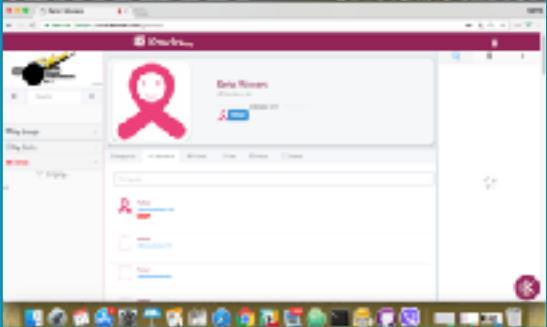
- React to an object a group (Eg : Post in a group)

Input	Expected Output	Actual Output	Comment
Selected Group : Valid User Position : Valid Selected Post : Invalid Reaction Type : Valid	"cannot react to post, it is not available anymore "		Success
Selected Group : Valid User Position : Valid Selected Post : Invalid Reaction Type : Valid But there is no internet connection	"There is no Internet Connection"		Success
Selected Group : Valid User Position : Valid Selected Post : Valid Reaction Type : Valid	Displays your reaction across the post in the group		Success

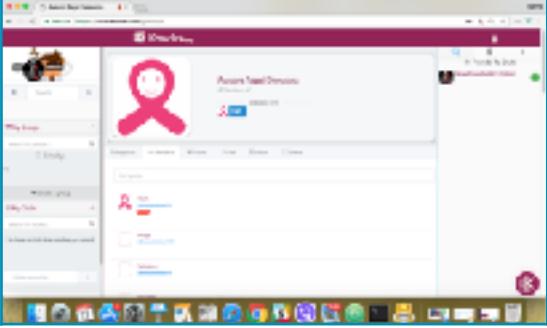
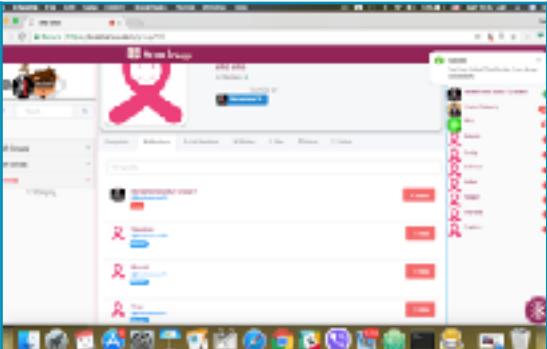
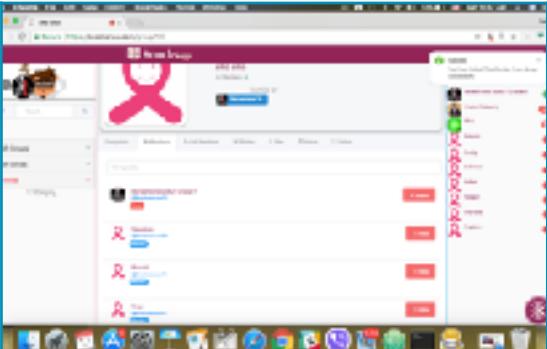
- Change user permission

Input	Expected Output	Actual Output	Comment
Selected Group : Valid User Position : Invalid New Position : Valid	Not displaying the "Manage positions" Section		Success
Group Name : valid Group Category : empty User Position : Valid But there is no internet connection	"There is no Internet Connection"		Success
Selected Group : Valid User Position : Valid New Position : Valid	"You changed this user's position successfully"		Success

- Change the group privacy set

Input	Expected Output	Actual Output	Comment
Privacy Set : Valid User Position : Invalid	Not displaying group privacy set updating section		Success
Privacy Set : Valid User Position : Valid But there is no internet connection	"There is no Internet Connection"		Success
Privacy Set : Valid User Position : Valid	"Group privacy set is updated successfully"		Success

- Ban members of a group- when allowed to do so.

Input	Expected Output	Actual Output	Comment
User Position : Invalid (not a group owner) Group Name : Valid Selected Member : Valid	Not displaying "Kick" button		Success
User Position : Valid Group Name : Valid Selected Member : Valid, But there is no internet connection	"There is no Internet Connection"		Success
User Position : Valid Group Name : Valid Selected Member : Valid	"You have kicked this member from group successfully"		Success

- Report a group- when allowed to do so.

Not yet a ready function

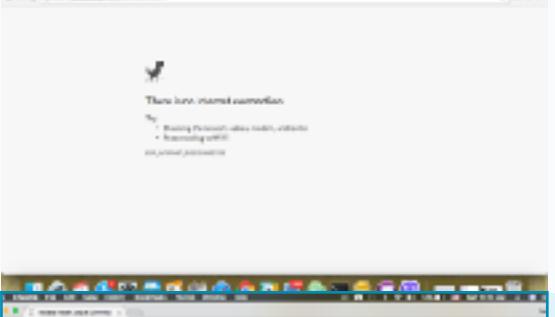
- Chat in a group

Not yet a ready function

- Request for account verification - incase of public figure.

Not yet a ready function

- **View a user profile**

Input	Expected Output	Actual Output	Comment
Selected User : invalid (blocking friend)	"You cannot view this user profile"		Success
Selected User : valid But there is no internet connection	"There is no Internet Connection"		Success
Selected User : valid	Viewing the user's profile		Success

- Explore talents

Not yet a ready function

- Upload a talent object - when allowed to do so

Not yet a ready function

- Filter talents

Not yet a ready function

- Explore hashtags in talents section

Not yet a ready function

- Rate a talent object

Not yet a ready function

- View a talent career

Not yet a ready function

- Rate a talent owner

Not yet a ready function

- Leave a review on a talent object

Not yet a ready function

- Leave a review on a talent owner

Not yet a ready function

- View Analysis for the talent owner himself

Not yet a ready function

- Monitor for child log.

Not yet a ready function

- Monitor objects for kid

Not yet a ready function

.

- Set admins for public figure

Not yet a ready function

- Ban for public figure account

Not yet a ready function

- Set a parent for a child

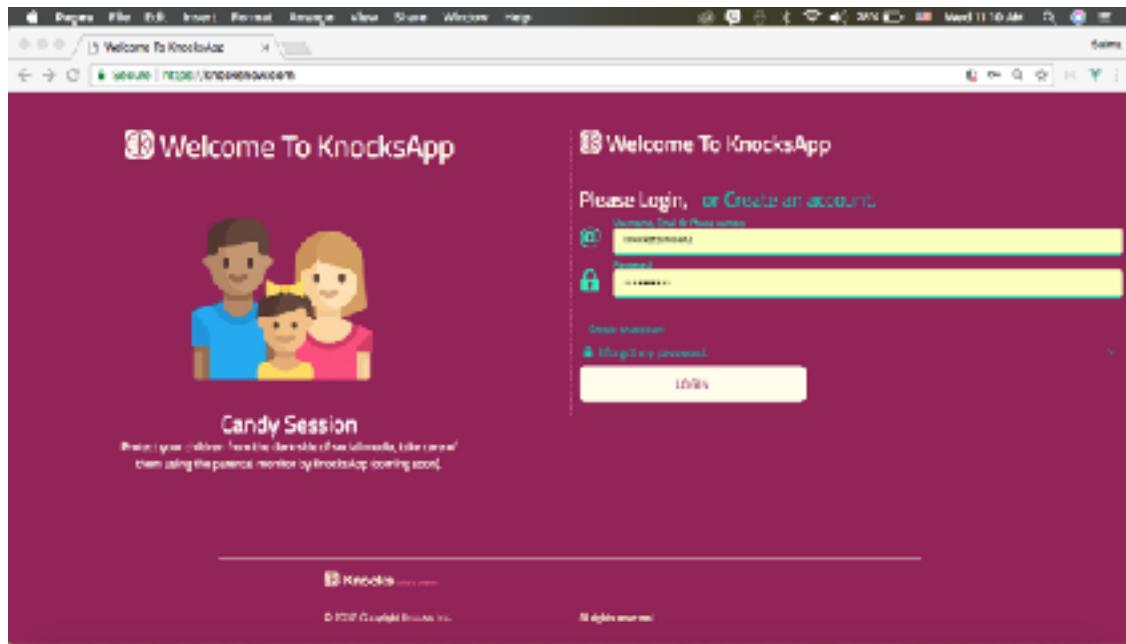
Not yet a ready function

7. Deployment

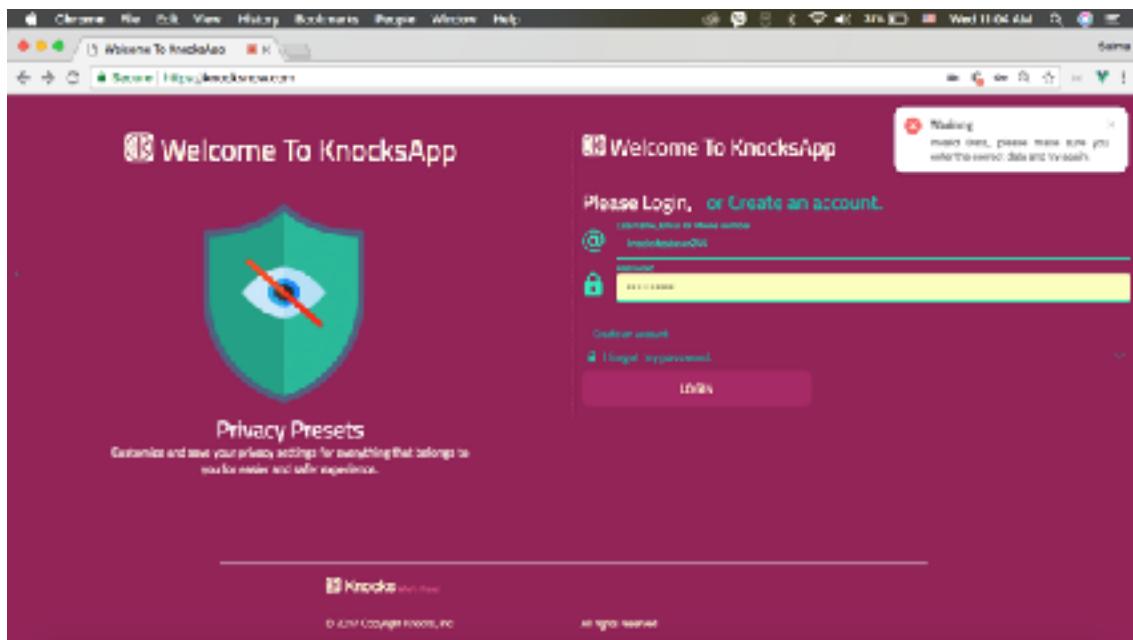
8. User Manual

- **Log in**

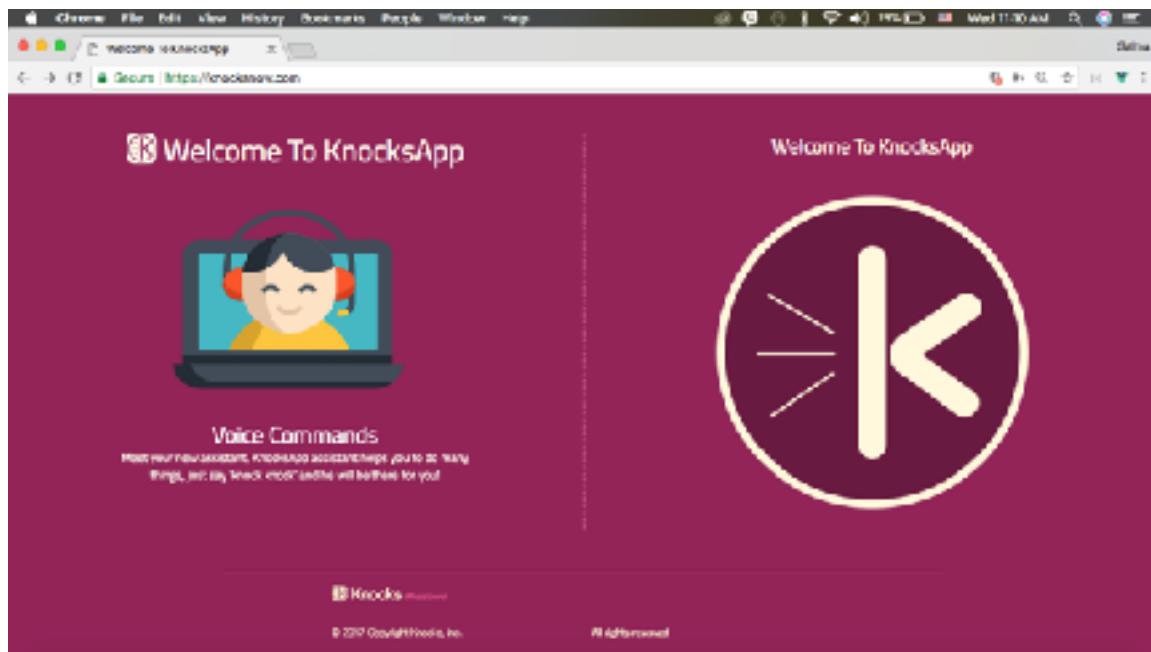
The user can log in using the phone number, e-mail, or username with the password which he/she registered before, then clicks on “LOGIN”



If the date are incorrect, the system will display a message “Invalid Data, Please make sure you enter the correct data and try again”

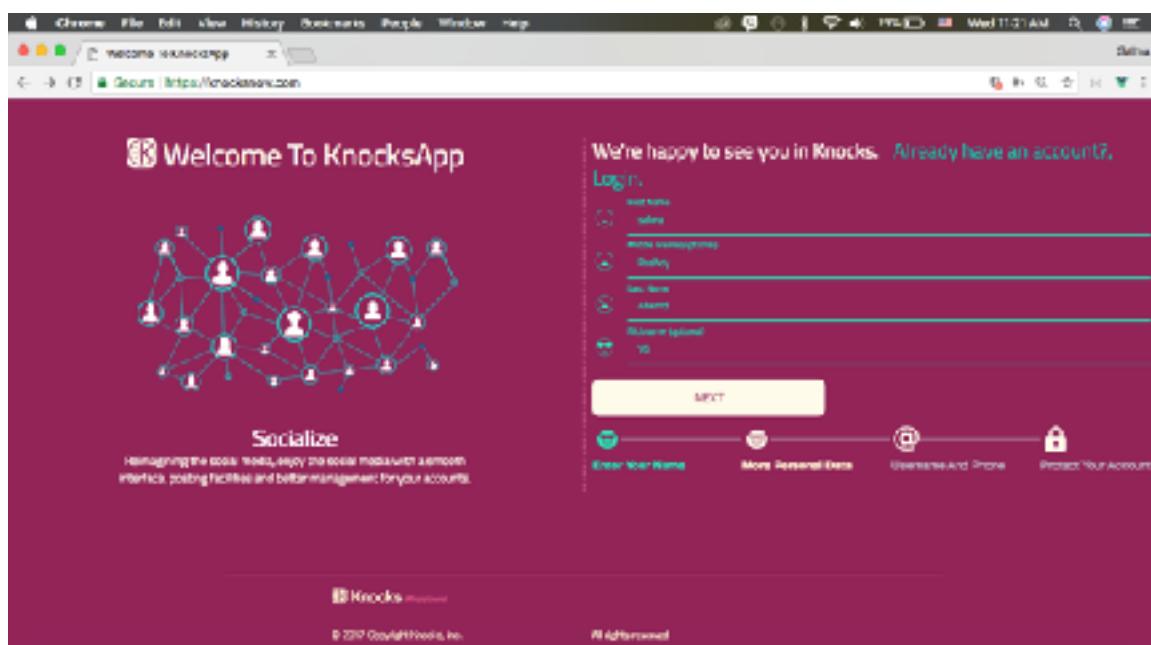


Else the system will display message “Welcome to KnocksApp” and system direct the user to his home page.

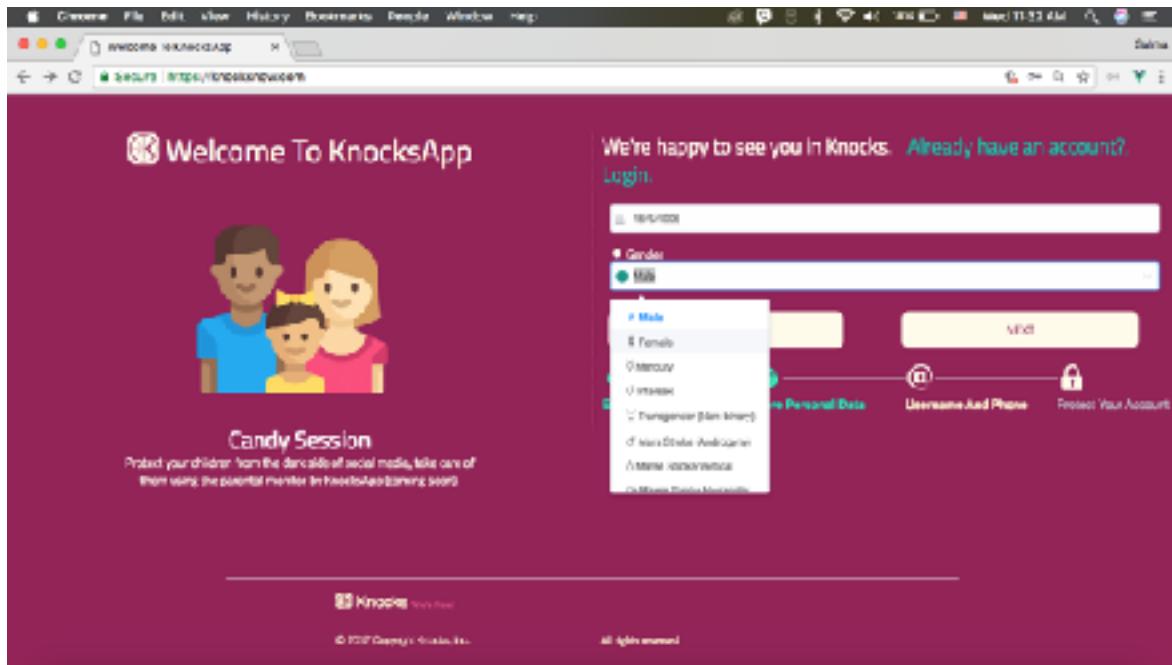


- **Sign up**

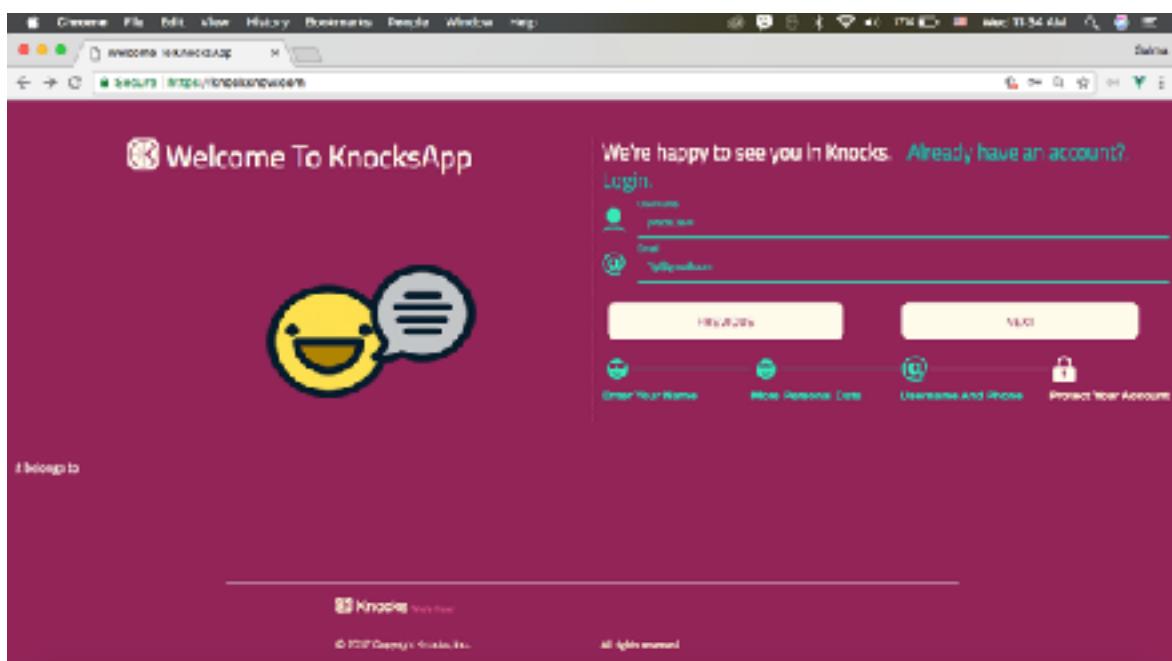
The user can sign up by providing First name (length should not exceed 15 characters) , Middle name (Optional and length should not exceed 15 characters), Last name (length should not exceed 15 characters), Nickname (Optional and length should not exceed 15 characters), then clicks on “NEXT” to finish first stage of sign up



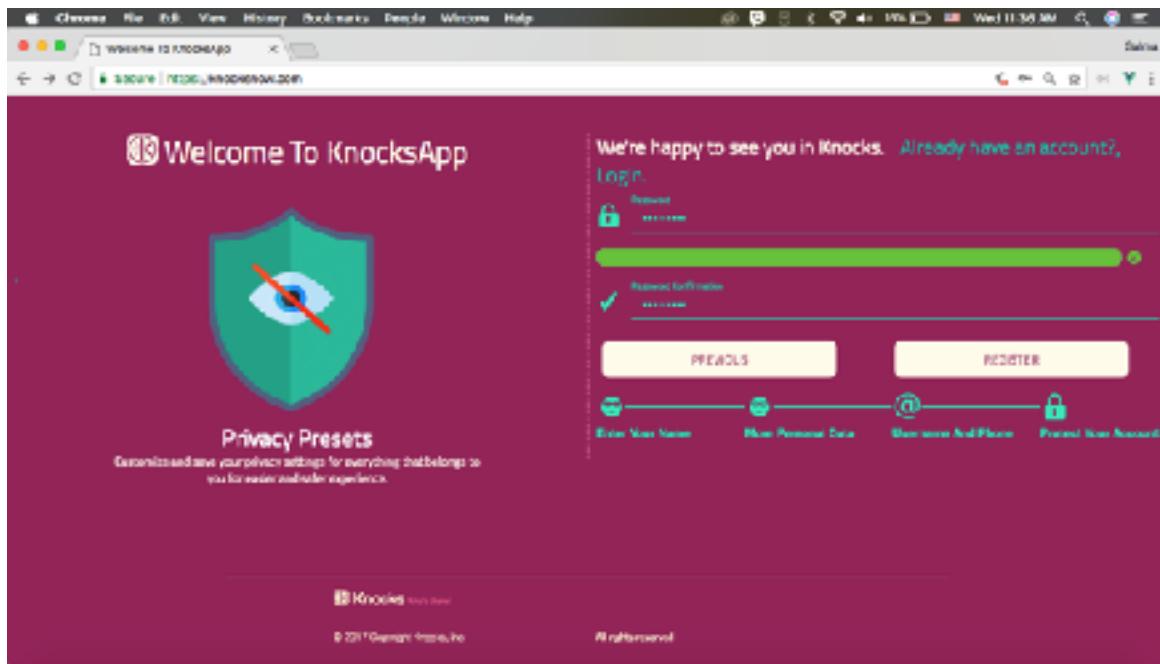
Then the user provides birthdate from a calendar and the gender from dropdown list (Female, Male, Genderless, Mars stroke, Mercury, intersex, Transgender, Mars stroke vertical, Mars stroke Horizontal), then clicks on “NEXT” to finish second stage of sign up or “PREVIOUS” to get back to previous stage for editing.



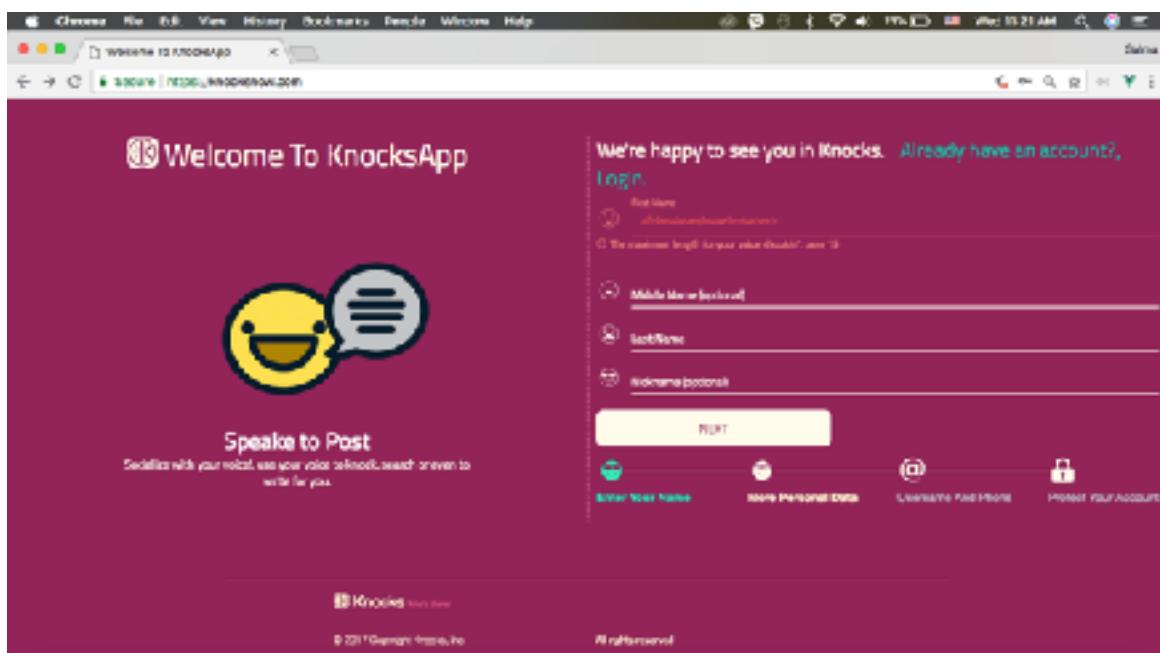
Then the user provides the user name (which must be non used by other users) and the e-mail which must be valid one then clicks on “NEXT” to finish third stage of sign up or “PREVIOUS” to get back to previous stage for editing.



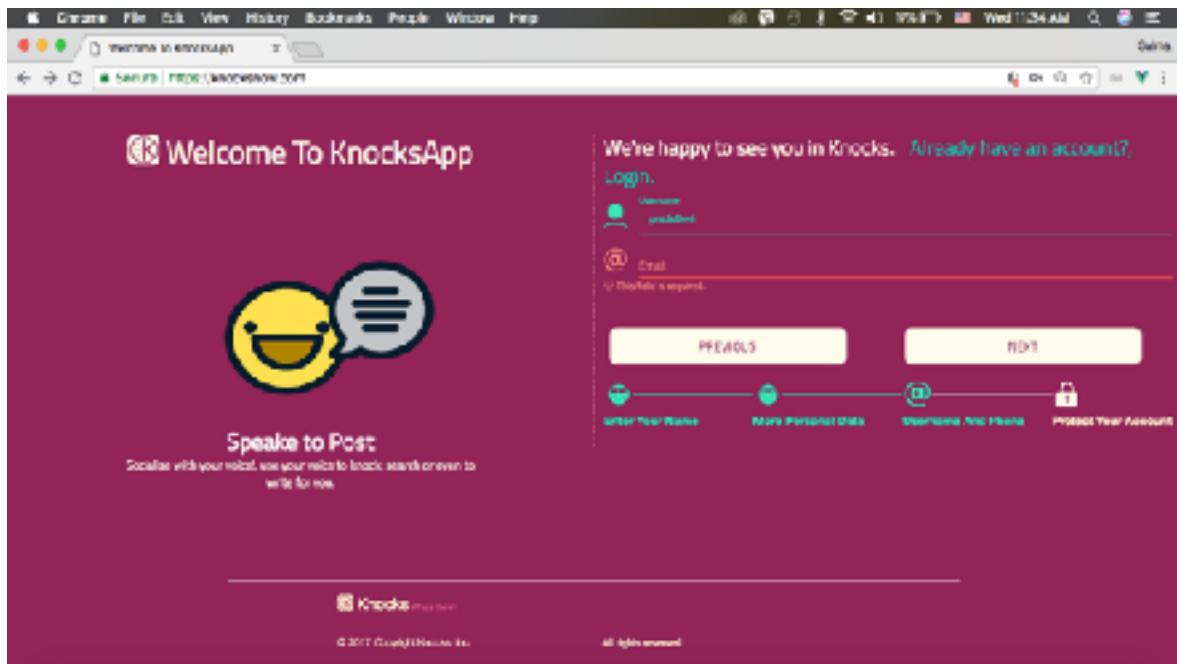
Then the user provides password (which must contain characters and numbers at least and not too close to the provided personal information), and then confirms the password (which must match the password), then clicks on “Register” to finish fourth and last stage of sign up or “PREVIOUS” to get back to previous stage for editing.



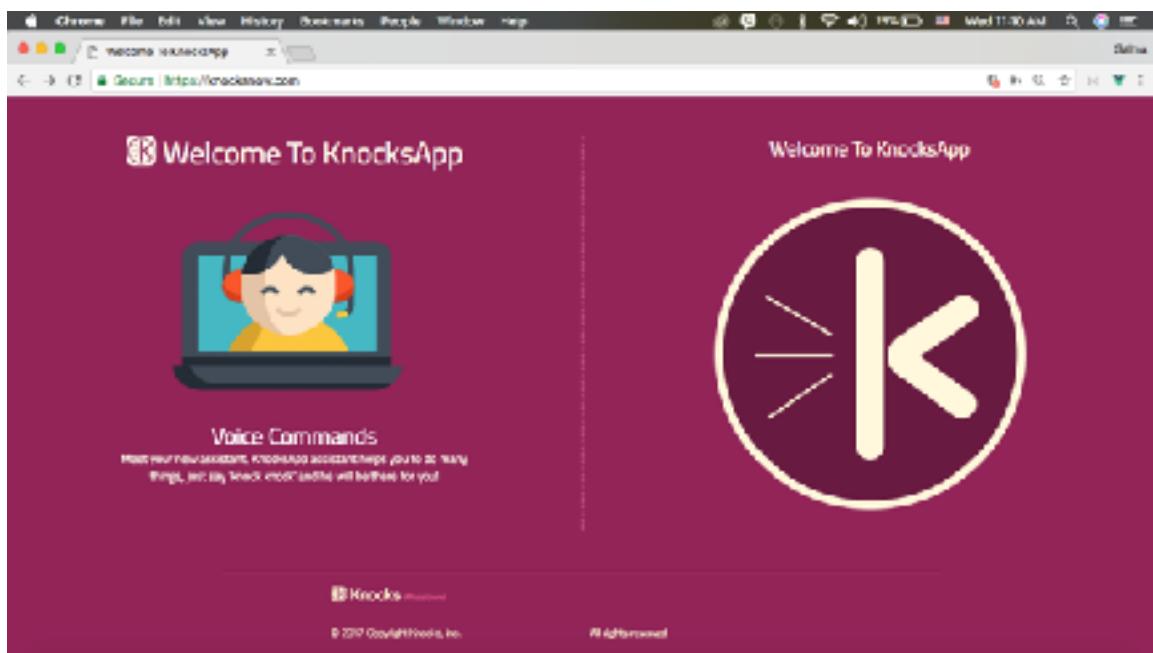
If the date are incorrect, for example providing incorrect first name, the system will display a message “The maximum length of the first name should not pass 15 characters”



If the date are empty, for example providing empty e-mail, the system will display a message “This field is required”

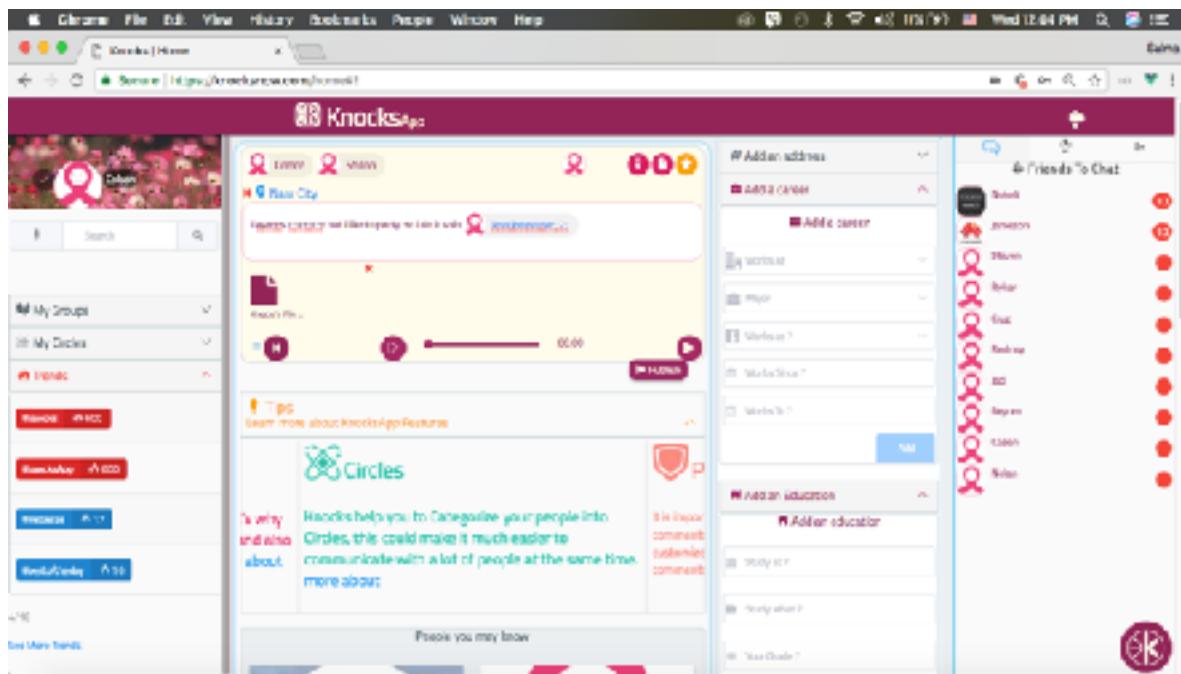


Else the system will display message “Welcome to KnockssApp” and system direct the user to his home page.

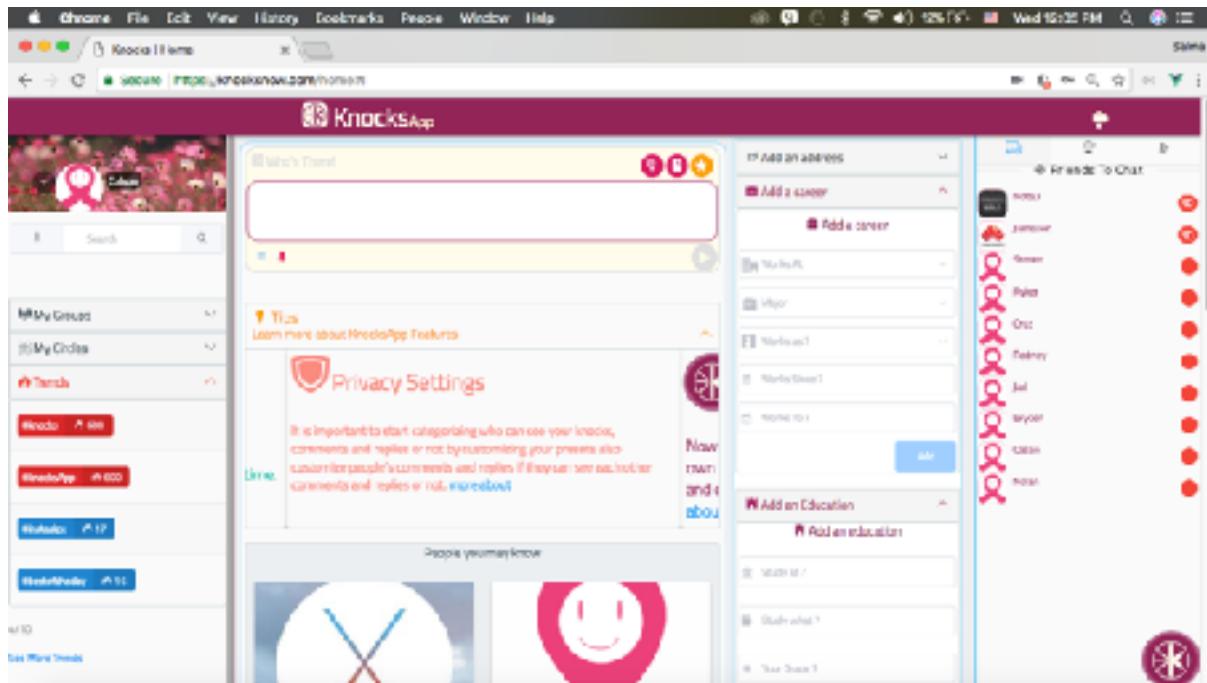


- **Publish a knock (create post).**

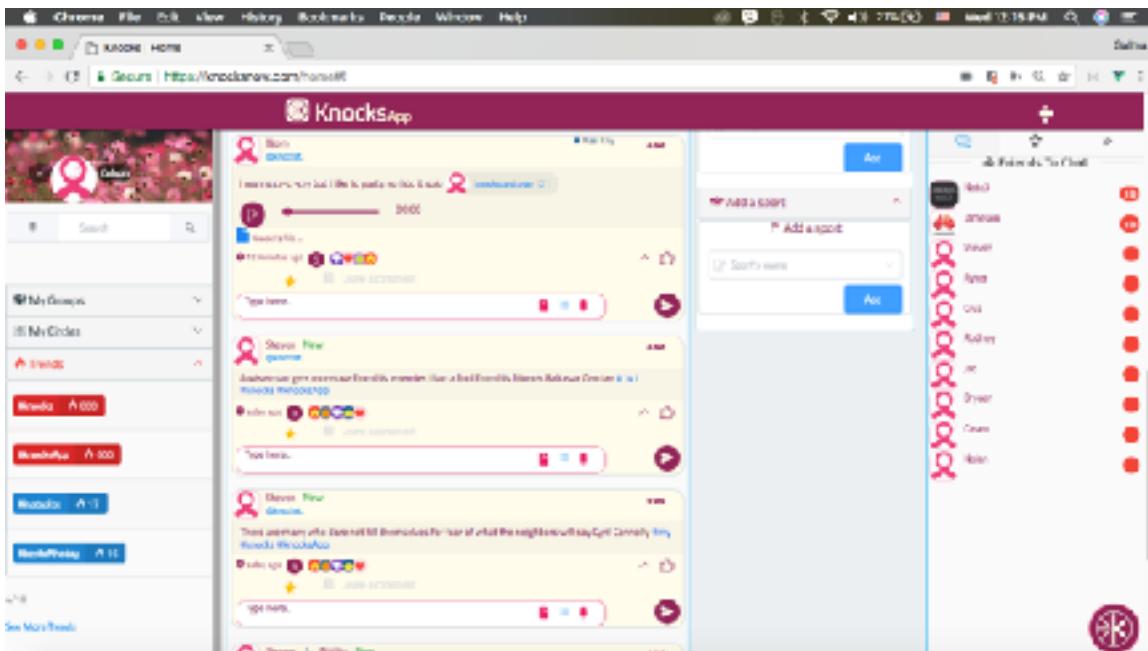
The user should provide text, check in, voice note, tagging a friend, uploading a file, voice to be converted into text or combination between then, then clicks on “Publish”



If the date are incorrect (Empty Knock), the system will not enable the publish button

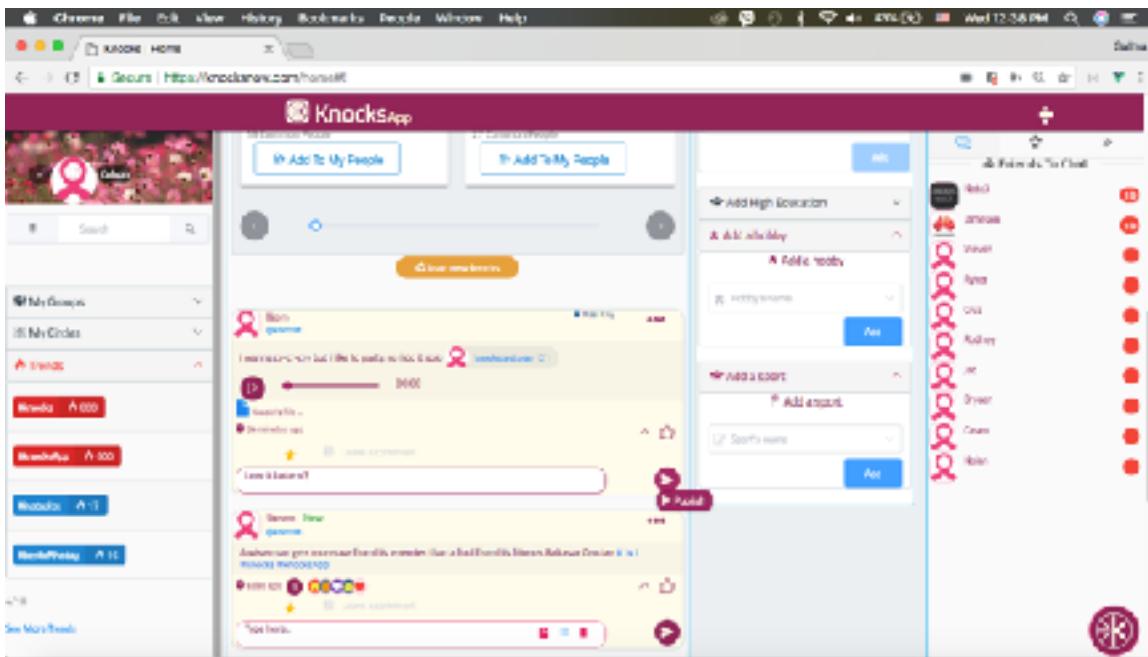


Else the system will display the knock at the user's time line



- **Write a comment**

The user should provide text, voice note, tagging a friend, uploading a file, voice to be converted into text or combination between them, then clicks on "Publish"



If the date are incorrect (Empty Comment), the system will display a message “ There is no content in your comment”

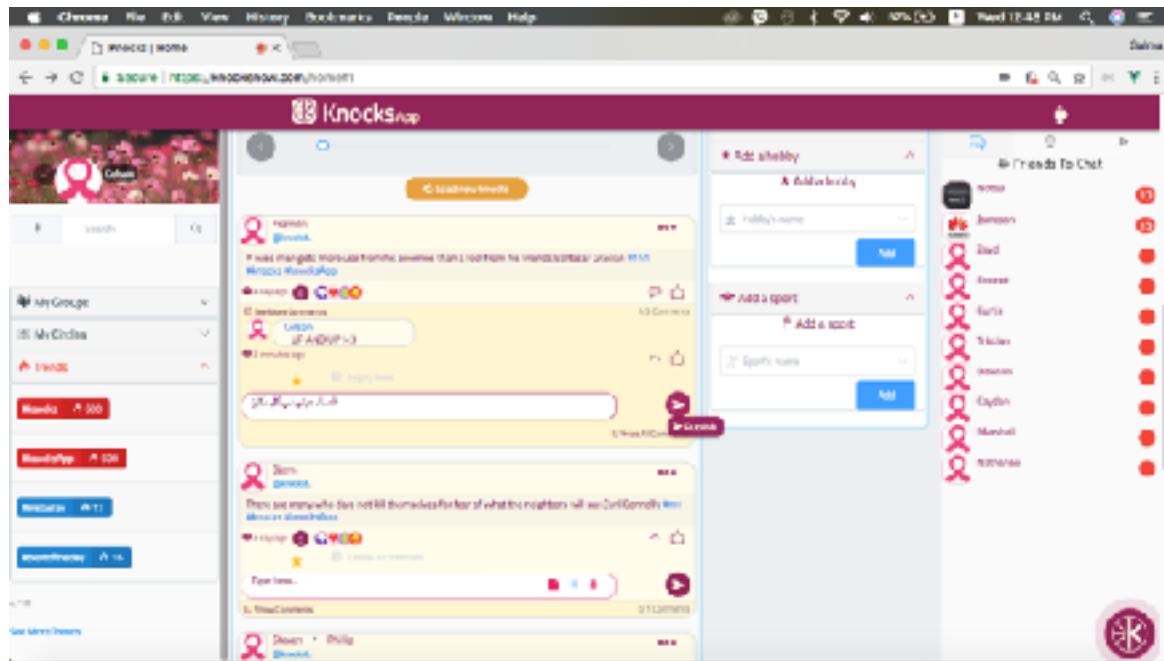
The screenshot shows the KnockStops application interface. In the center, there is a message box with the text "There is no content in your comment". Below this, there is a timeline feed with several posts from different users. One post by "Spicy_gamer" says: "I am going to my birthday party, so I do it like this". Another post by "Spicy_gamer" says: "A wise man gets more joy from his enemies than from his friends. Goliath Doctor". On the right side, there is a sidebar titled "Friends To Chat" with a list of friends: Jennifer, Zayil, Peacock, Gami, Trishler, Devonee, Dylar, Marissa, and Asherad.

Else the system will display the Comment towards the knock at the user's time line

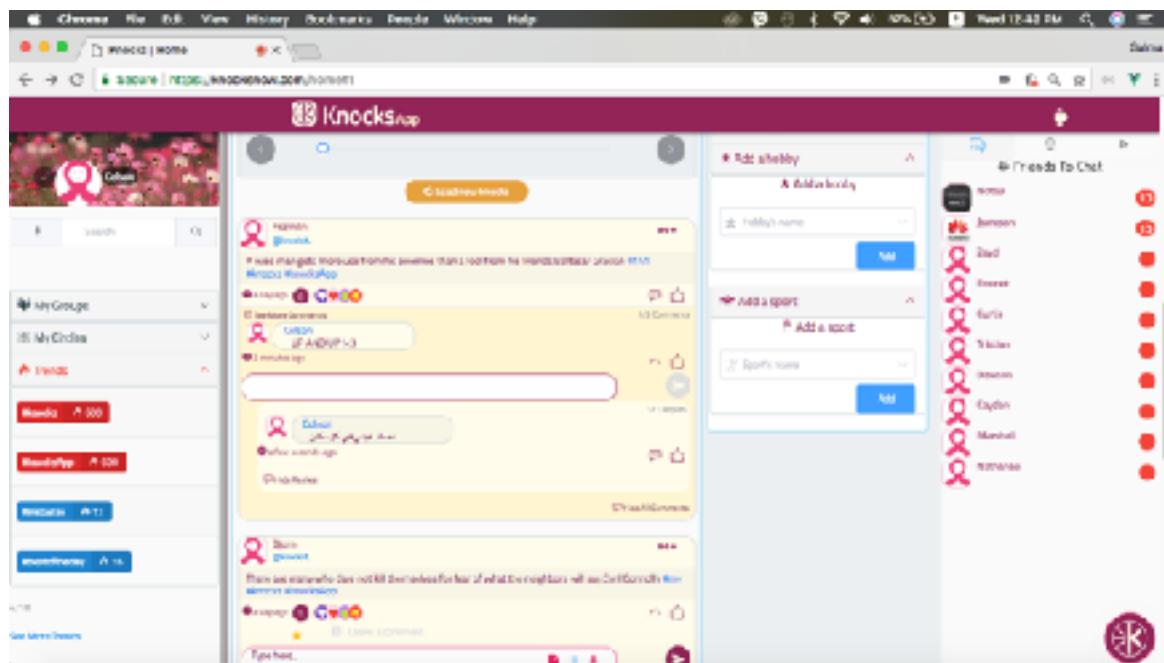
The screenshot shows the KnockStops application interface. The timeline feed now displays comments. A comment by "Spicy_gamer" says: "A wise man gets more joy from his enemies than from his friends. Goliath Doctor". Another comment by "Spicy_gamer" says: "There are many ways how we will measure the success of our life. The neighborhood survey is currently being conducted". On the right side, there is a sidebar titled "Friends To Chat" with a list of friends: Jennifer, Zayil, Peacock, Gami, Trishler, Devonee, Dylar, Marissa, and Asherad.

- **Write a reply.**

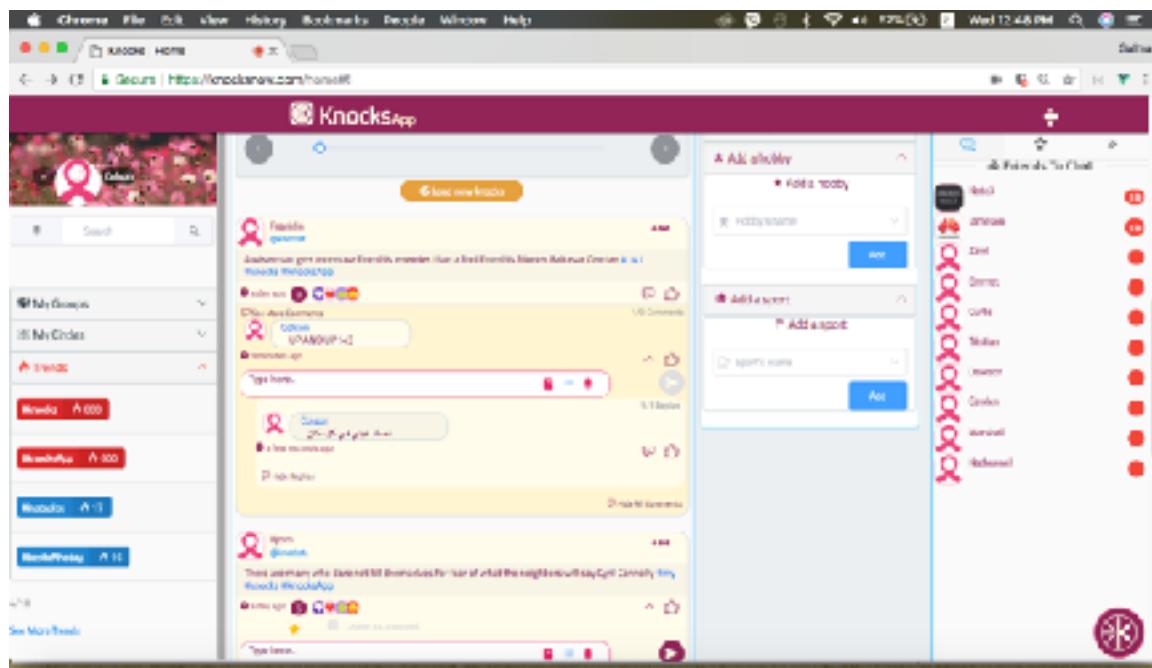
The user should provide text, voice note, tagging a friend, uploading a file, voice to be converted into text or combination between then, then clicks on “Publish”



If the date are incorrect (Empty Reply), the system will not enable the publish button

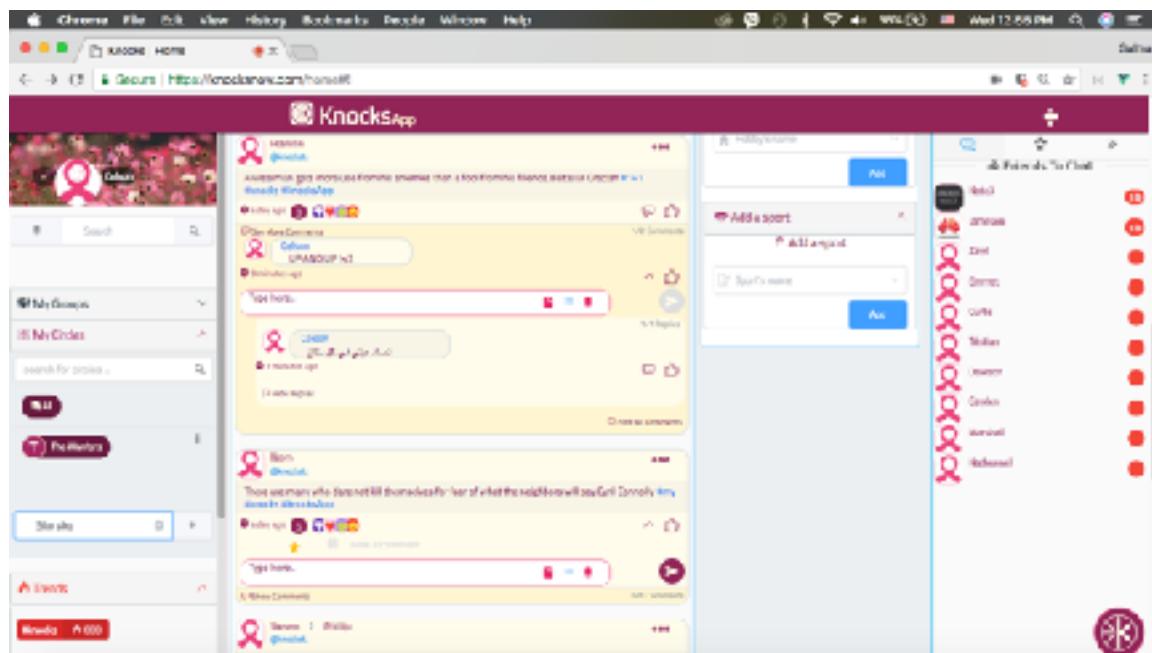


Else the system will display the reply towards the Comment at the user's time line

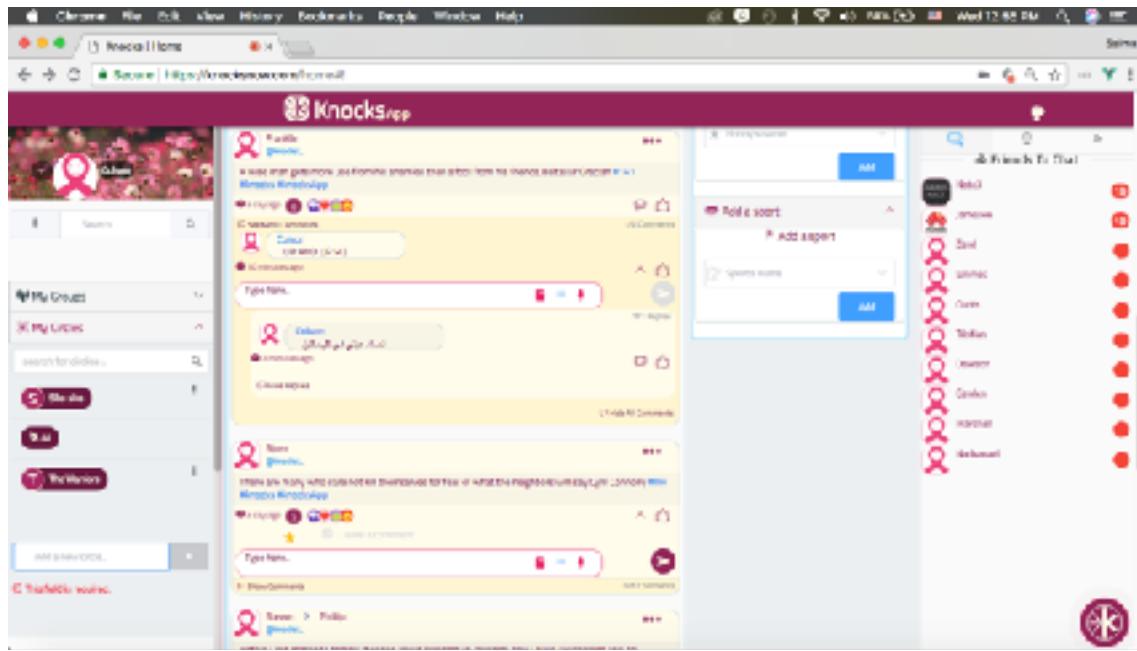


- **Create a circle.**

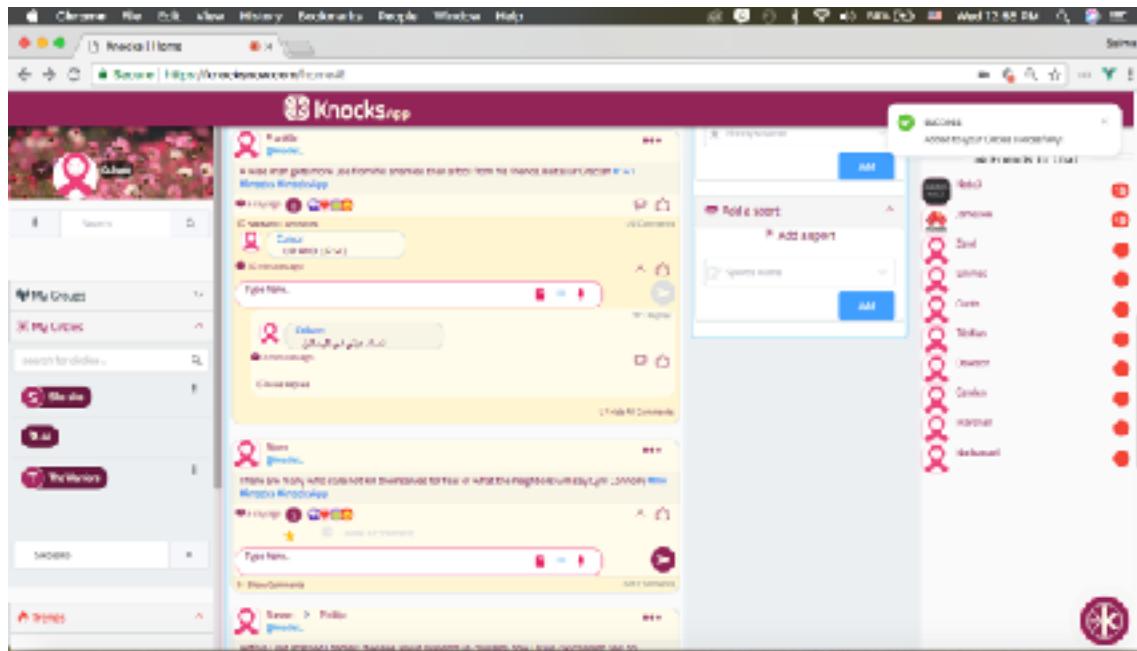
In the circles section on the middle left corner of the timeline the user should provide name of a circle that must be unique for this user's circles, then clicks on Add Circle icon



If the date are incorrect (Repeated circle name), the system will display a message “You already have a circle with this name”

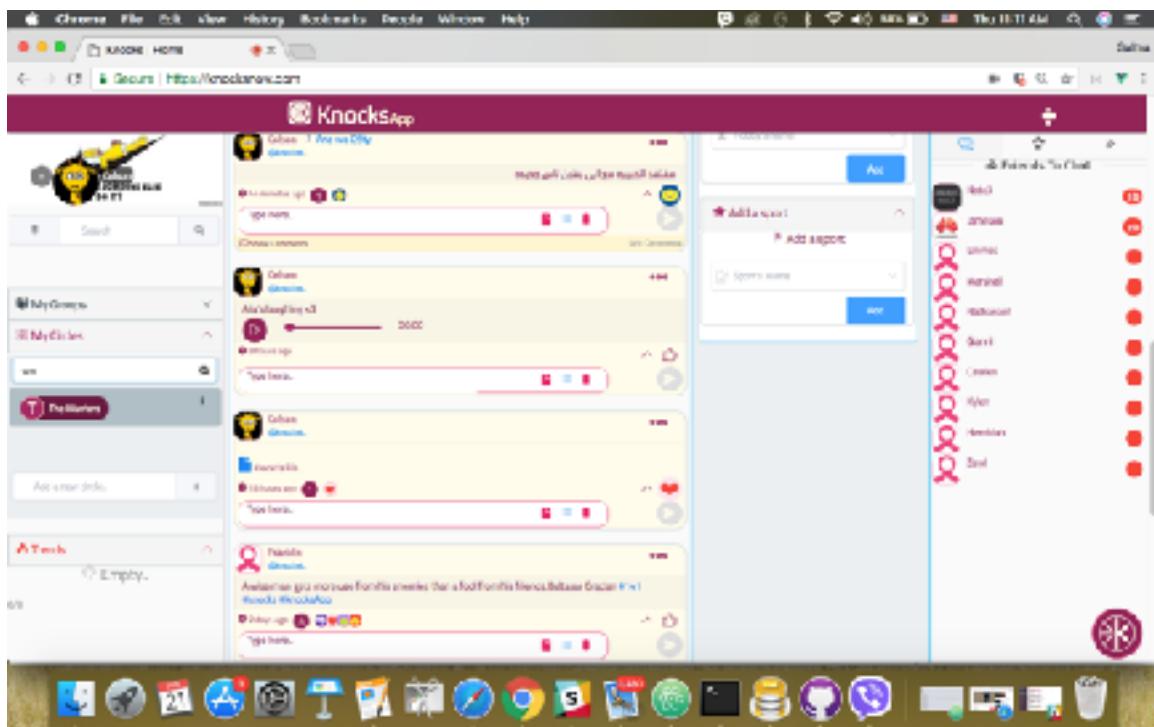


Else the system will display message “Added to your circles successfully” and displays the newly created user circle in the circles section

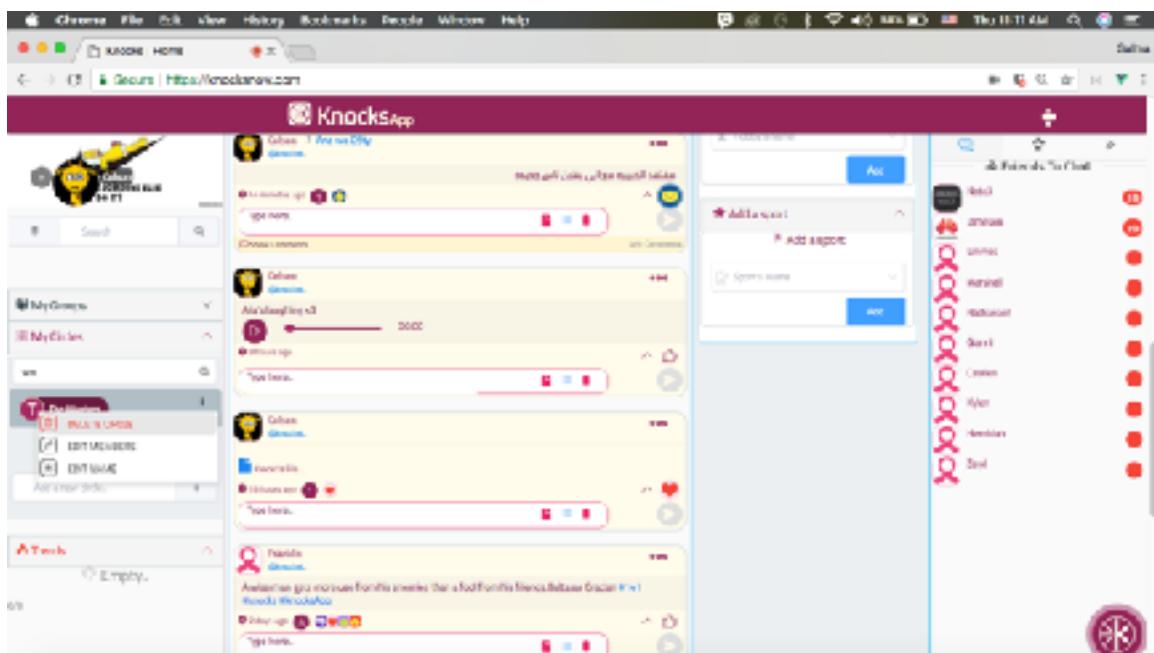


- **Delete a circle**

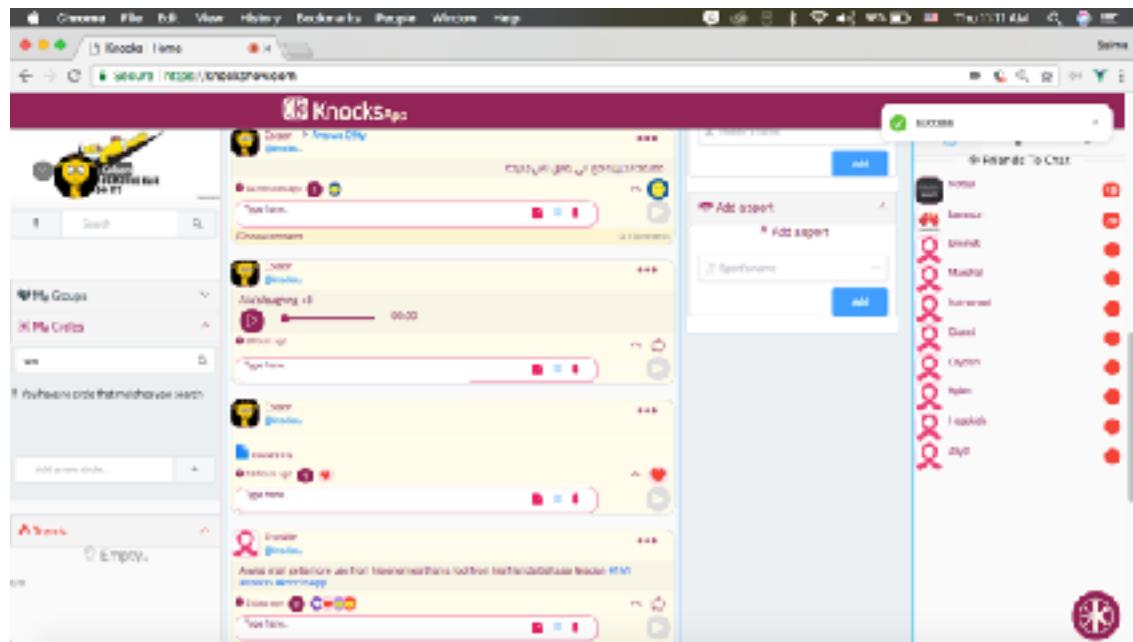
In the circles section on the middle left corner of the timeline the user should provide selection for the previously created circle by searching for it.



And then chooses from the dropdown list (Delete Circle, Edit Member, Edit Name), then he chooses “Delete Circle” option.

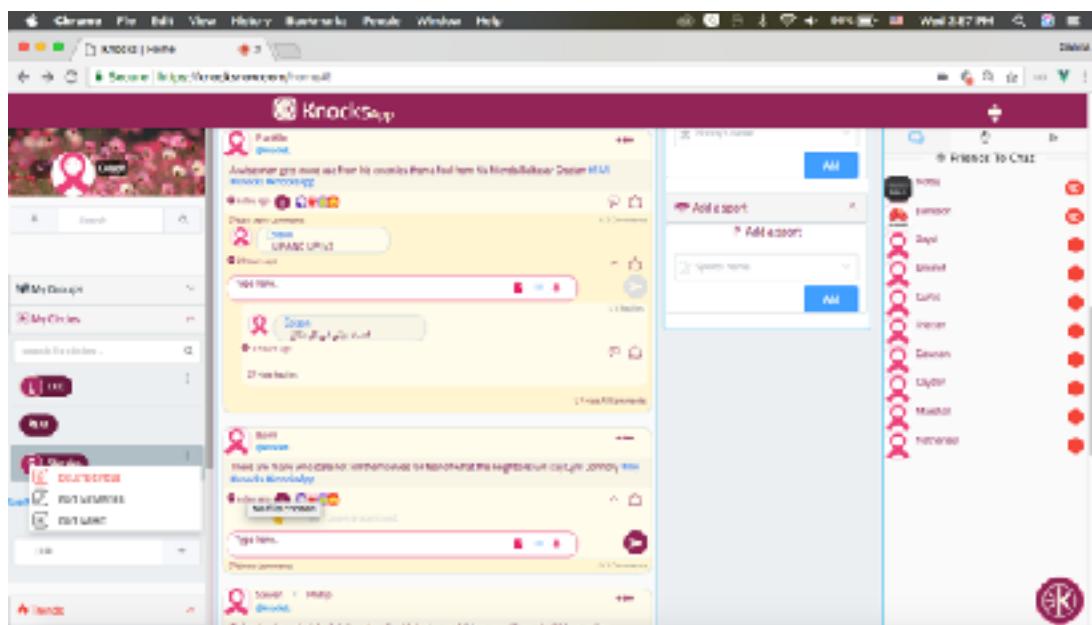


Then the system will display a message “Success”, and the deleted circle will be hidden from the circles section and never be shown again.

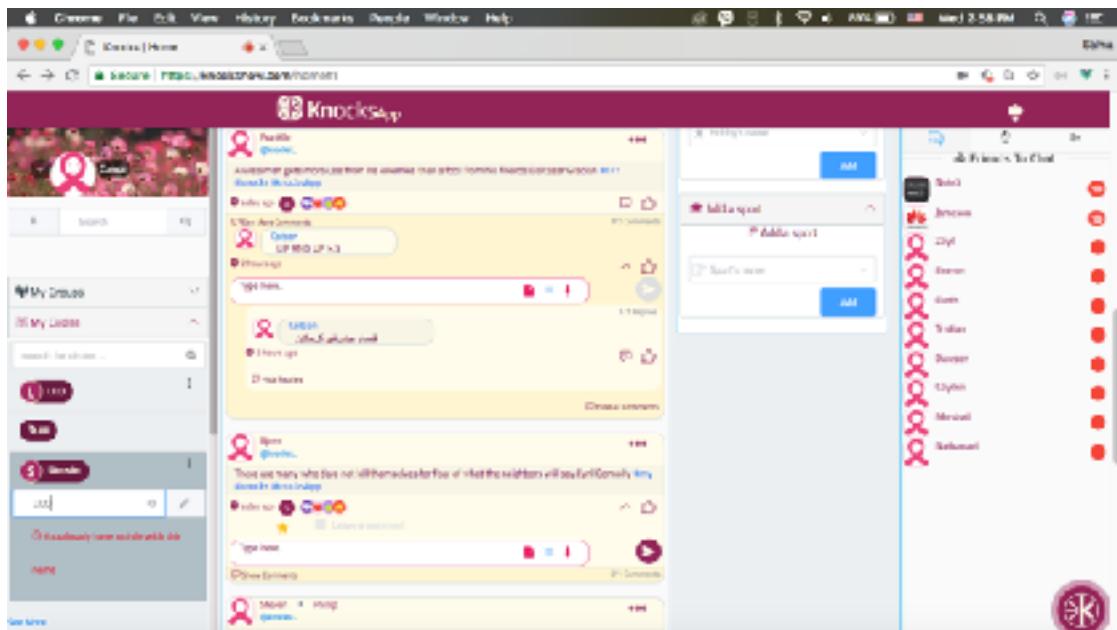


- **Update circle name**

In the circles section on the middle left corner of the timeline the user should provide select the previously created circle and then chooses from the dropdown list (Delete Circle, Edit Member, Edit Name), then he chooses “Edit Name” option then he should provide the new circle name that must be unique for this user’s circles, then clicks on Add Circle icon

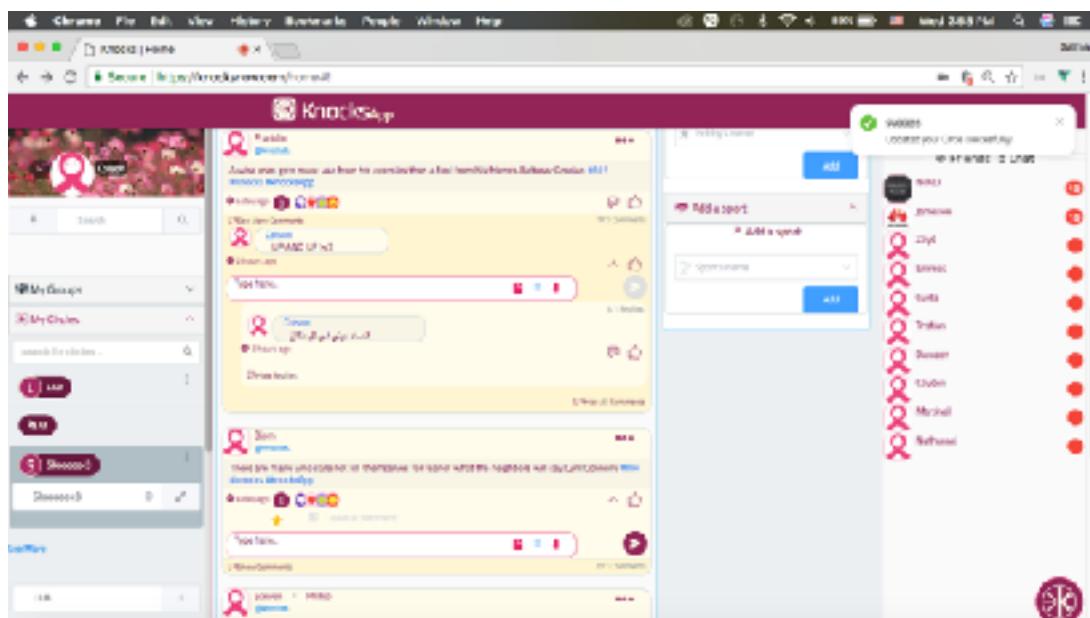


If the date are incorrect (Repeated circle name), the system will display a message “You already have a circle with this name”



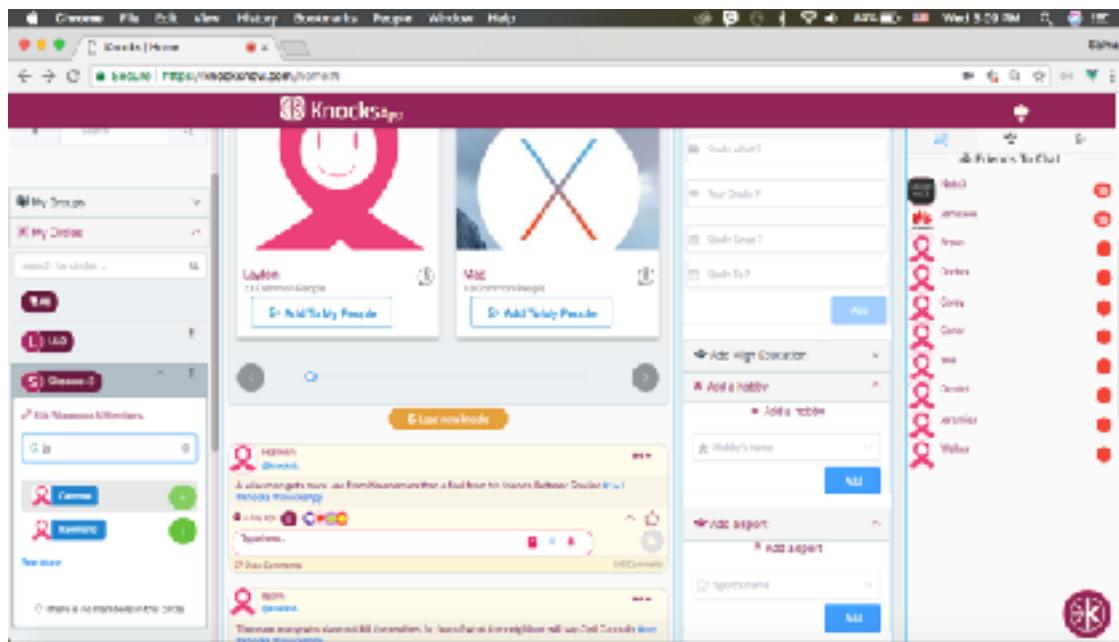
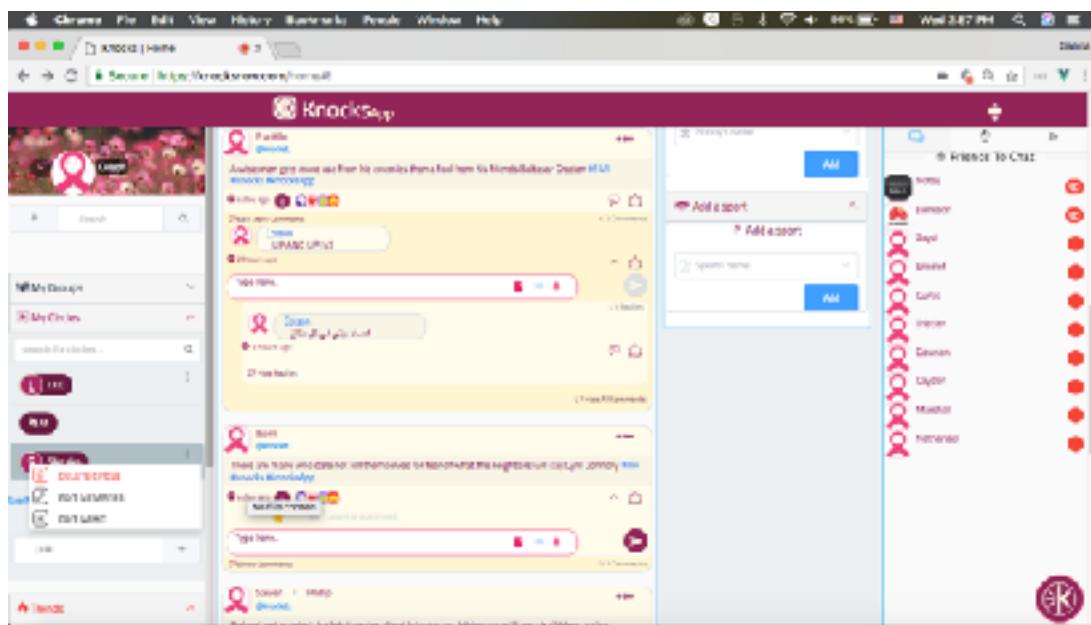
Else
the
system

will display message “Updated your circle successfully” and displays the newly updated user circle in the circles section

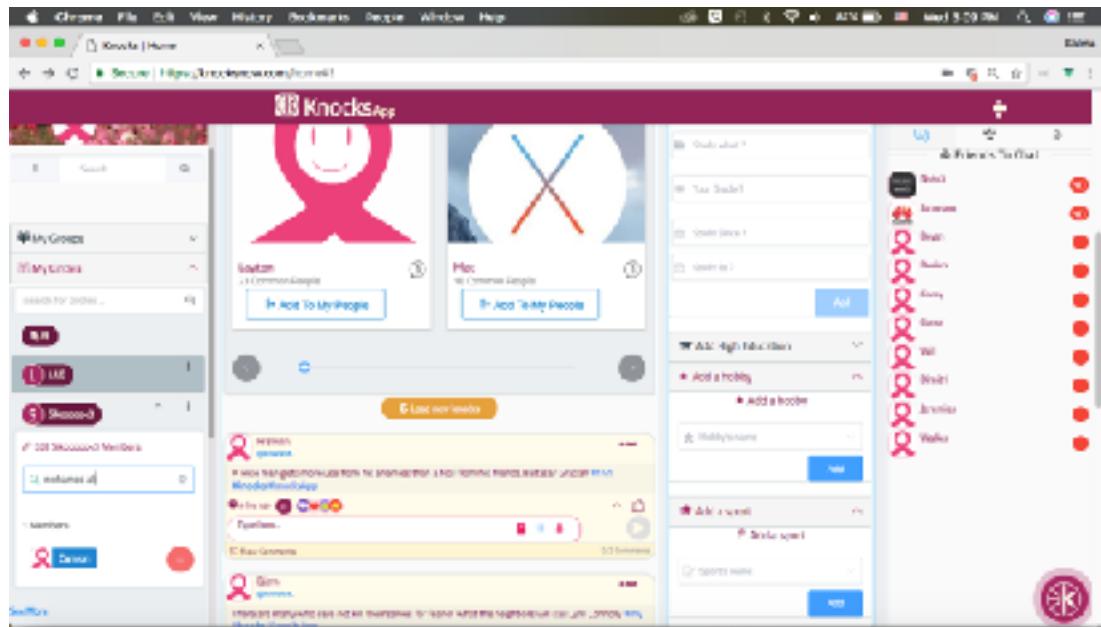


- **Include a friend to a circle**

In the circles section on the middle left corner of the timeline the user should provide select the previously created circle and then chooses from the dropdown list (Delete Circle, Edit Member, Edit Name), then he chooses “Edit Members” option then he should search for the user that he wants to add then after this member appears he should clicks on red icon to add member icon

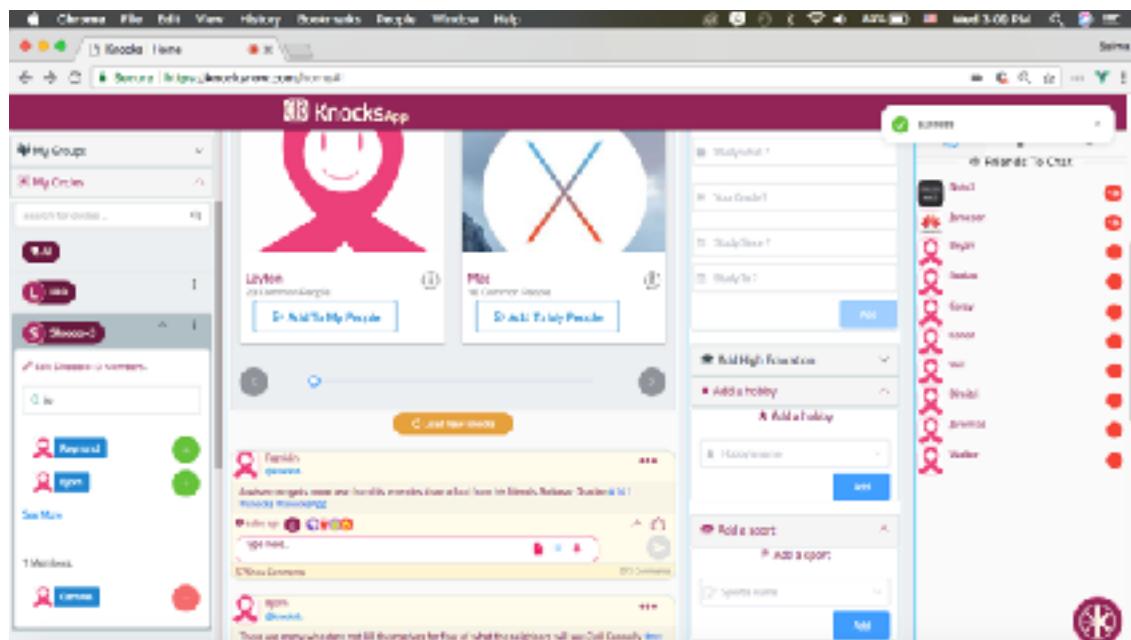


If the date are incorrect (searching for a member that is not a friend to add in the circle), the system will not display any member to be added



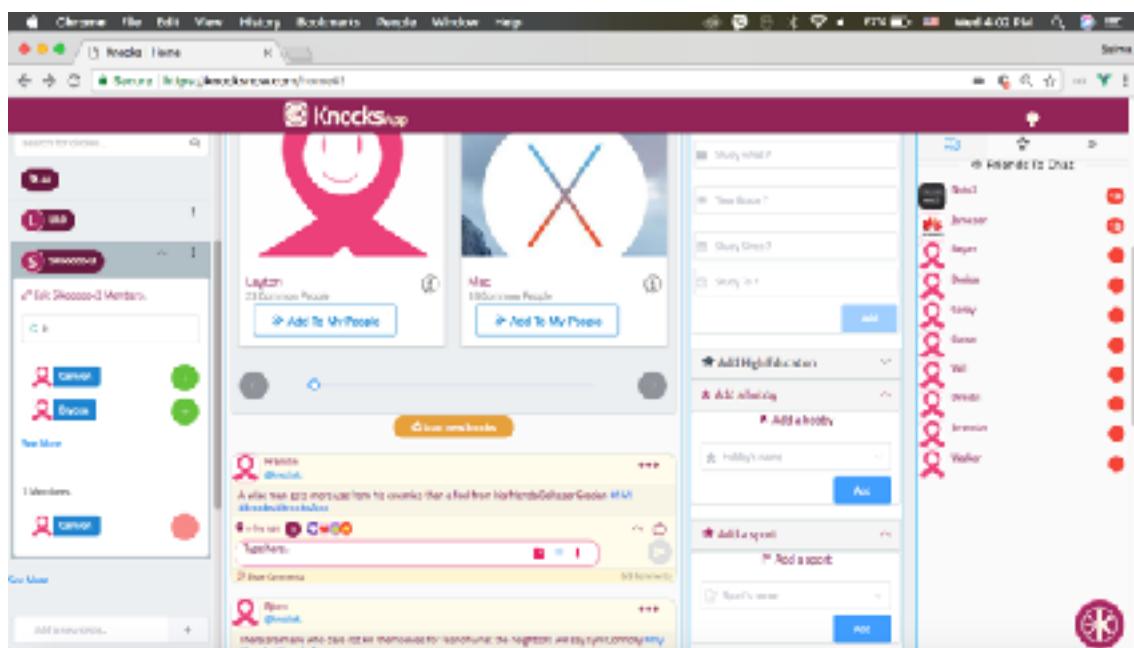
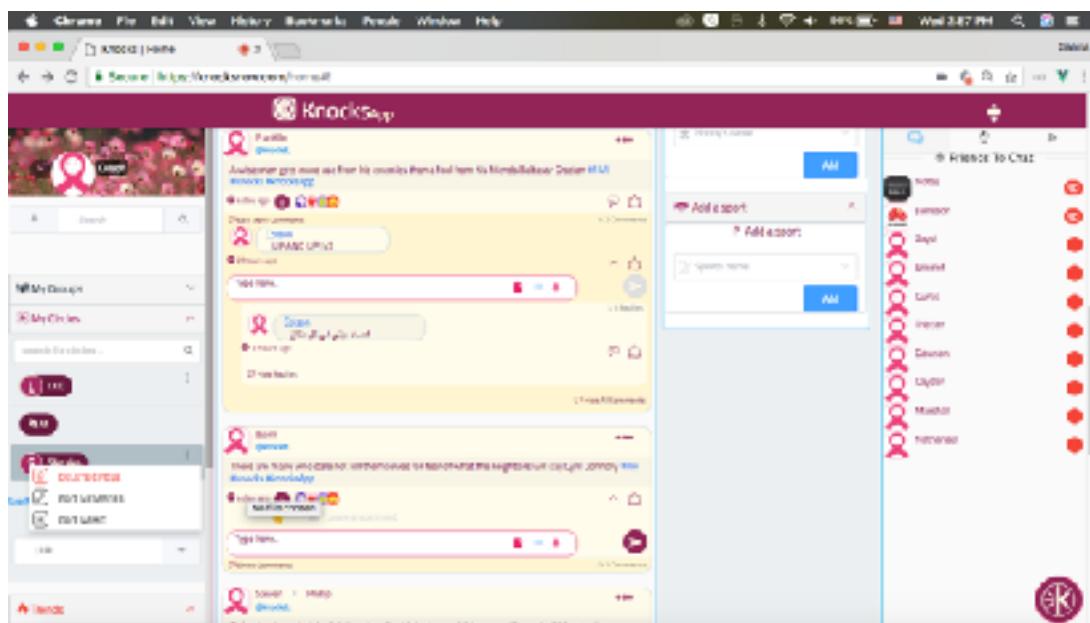
Else
the

system will display message “success” and displays the newly added member below the circle with red icon to indicate that this user is a member of the circle in the circles section

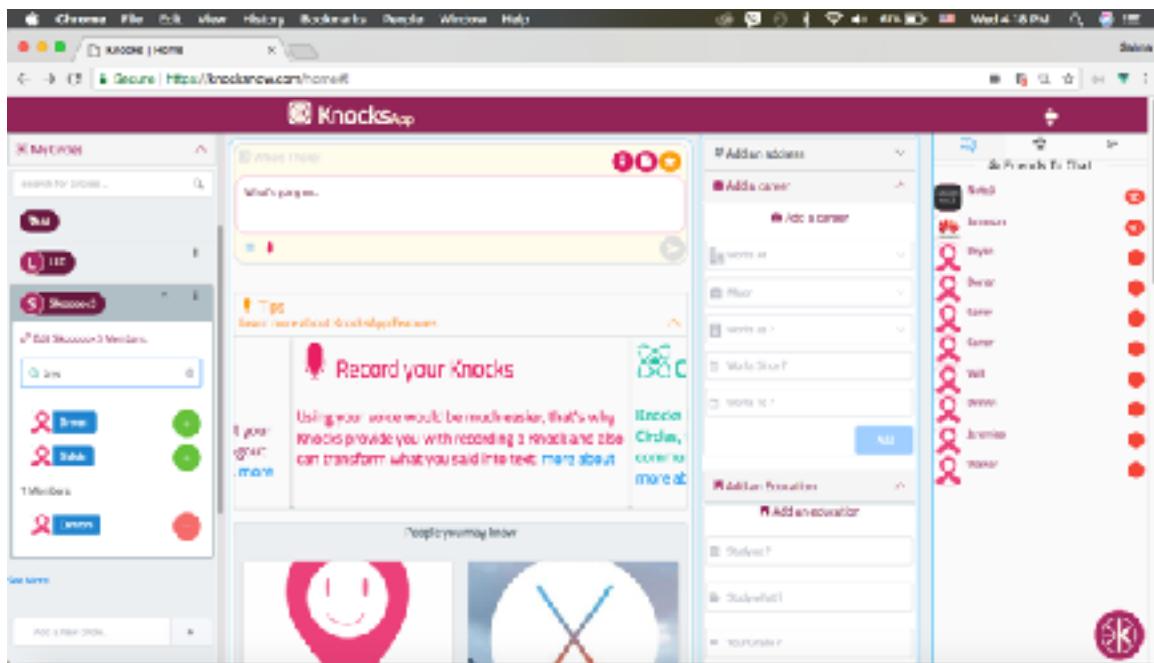


- Exclude a friend from a circle**

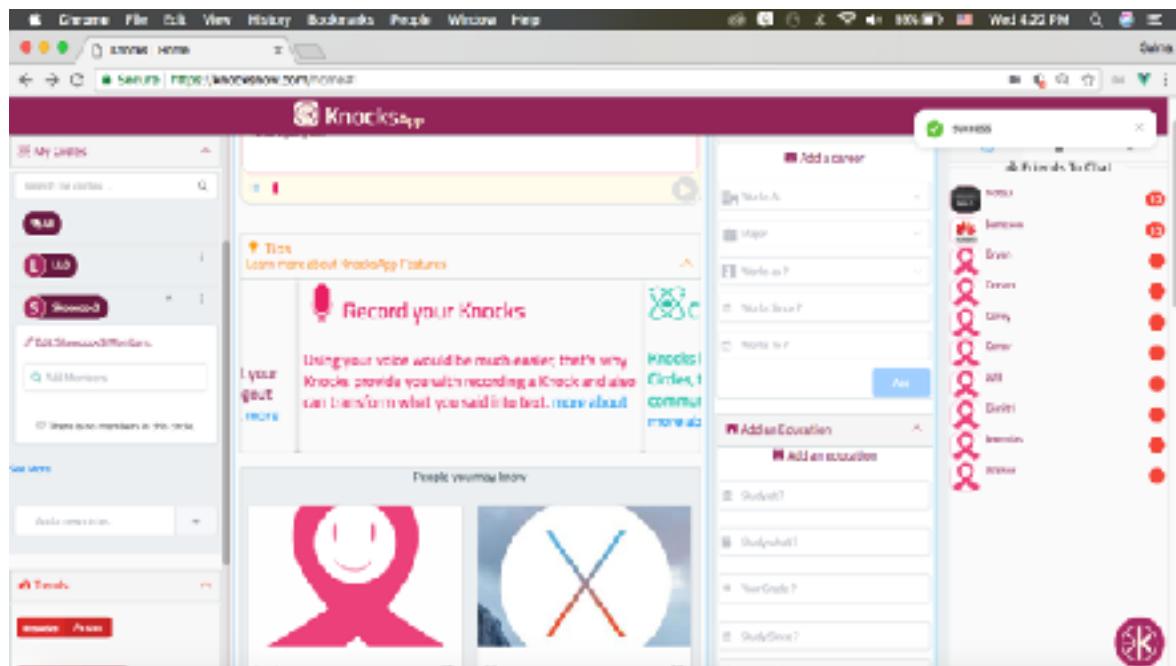
In the circles section on the middle left corner of the timeline the user should provide select the previously created circle and then chooses from the dropdown list (Delete Circle, Edit Member, Edit Name), then he chooses “Edit Members” option then he should search for the user that he wants to exclude then after this member appears he should clicks on red icon to remove member icon



If the date are incorrect (searching for a member that is not in a circle to remove), the system will not display any member to be removed

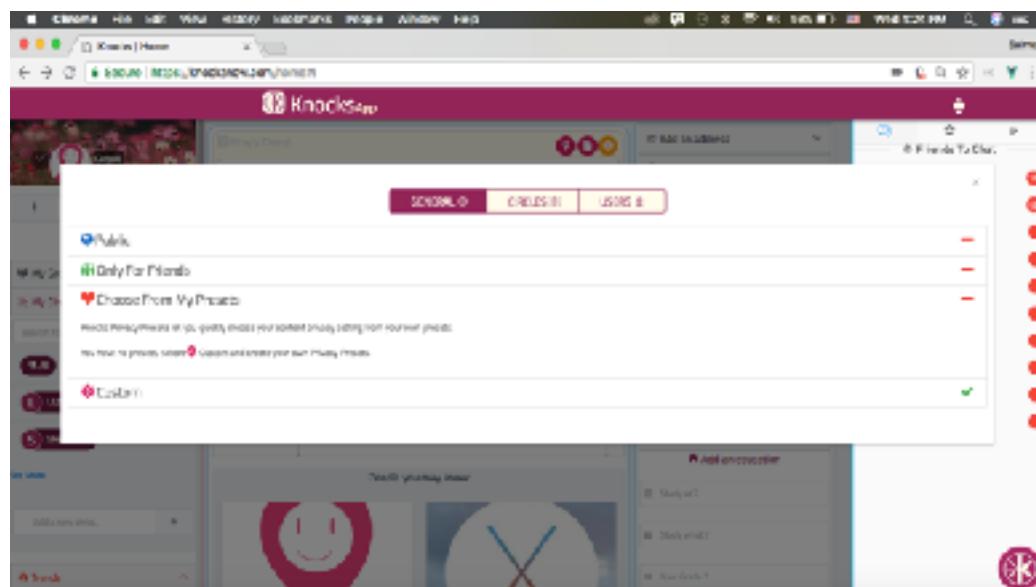
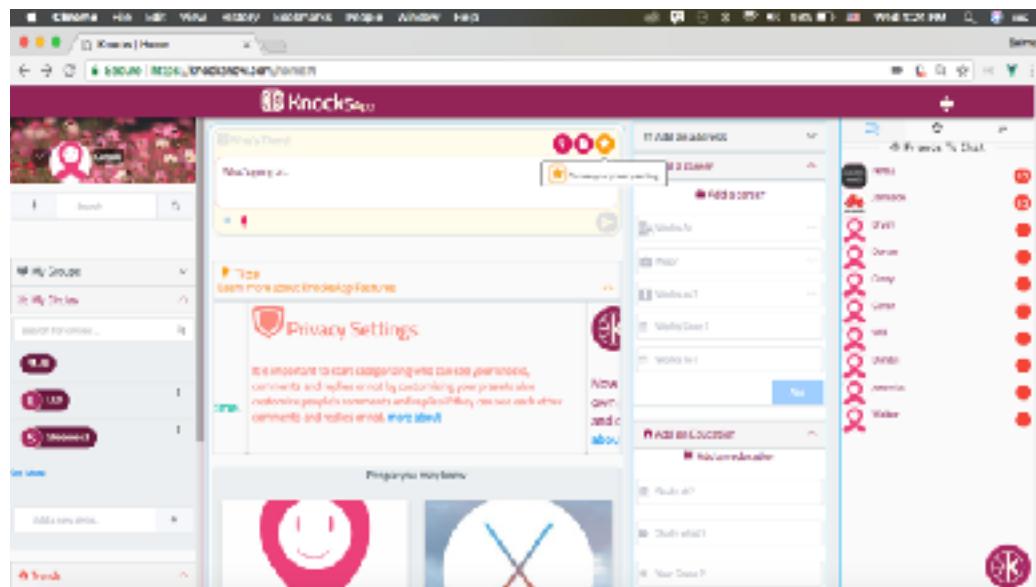


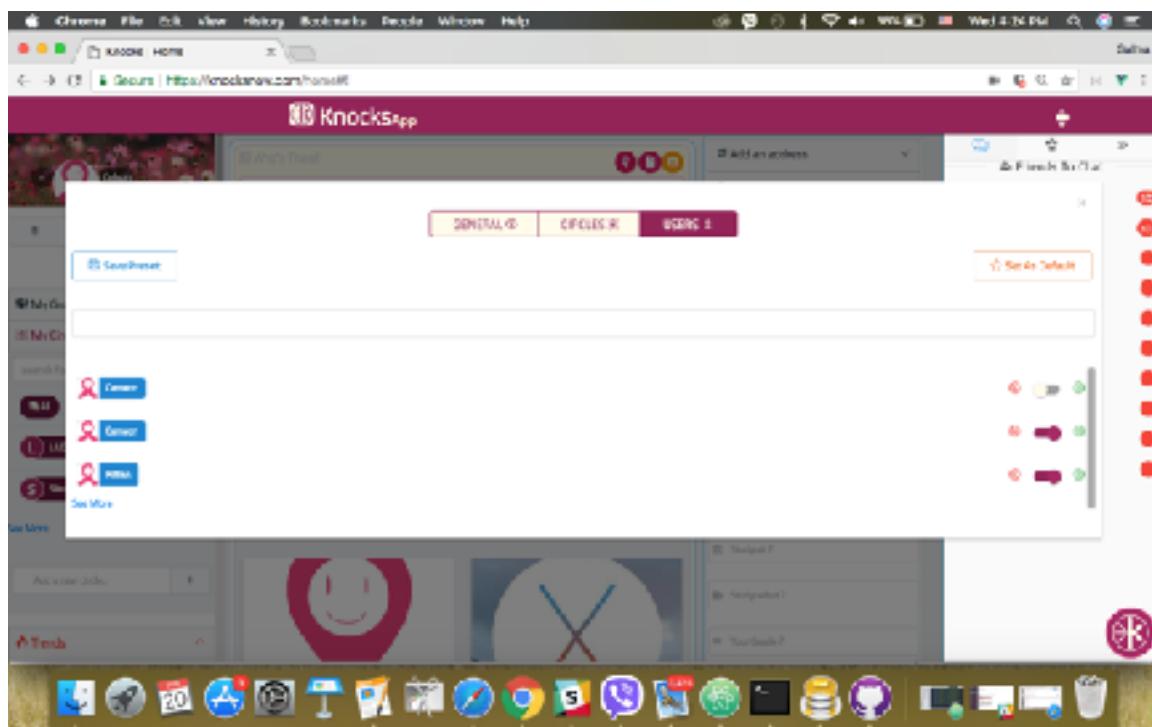
Else the system will display message “success” and displays the newly added member below the circle with red icon to indicate that this user is a member of the circle in the circles section



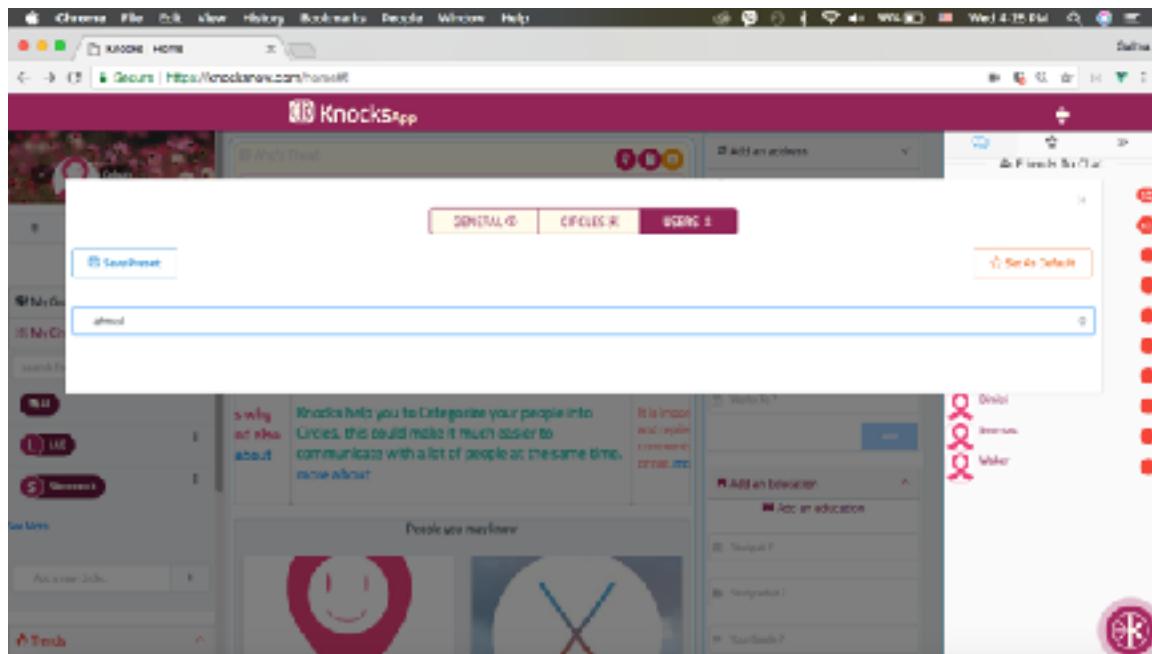
- **Regulate a privacy set for users**

At the knock section on the top of the page, at the General tab the user should click at the star icon that displays “Choose your privacy settings”, then from the options (which are : Public, Only for friends, Choose my Preset, Custom) clicks on Custom and clicks on “Users” tab, he can choose by the switch or the icons if the selected user can see his activities or not, and if the desired user is not shown he can search for him.

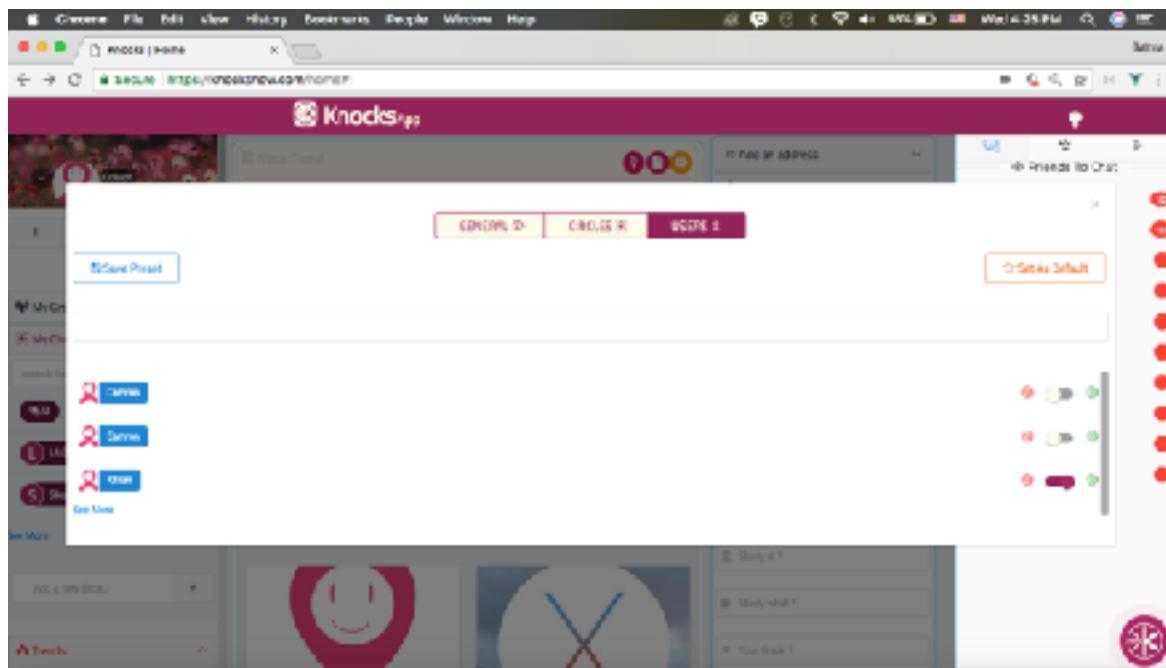




If the date are incorrect (searching for a user who is not a friend), the system will not display any user to choose privacy set for him.

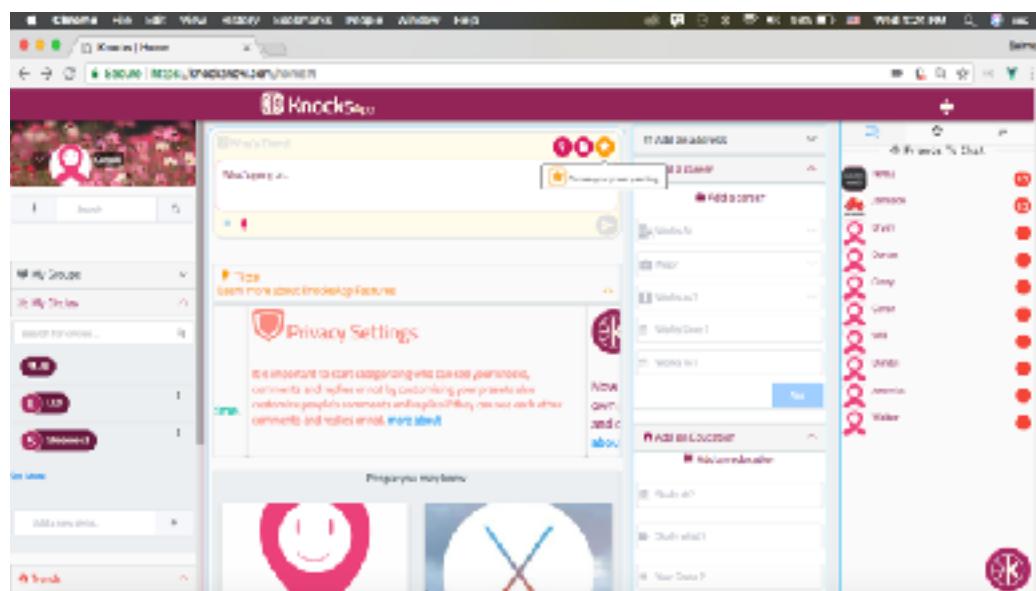


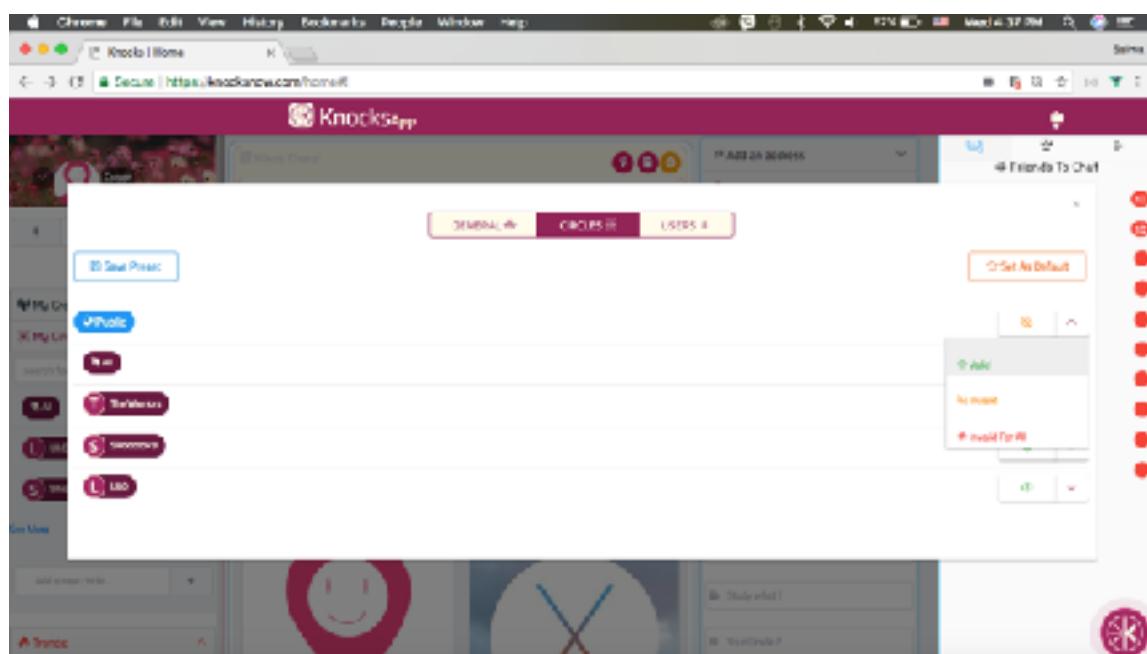
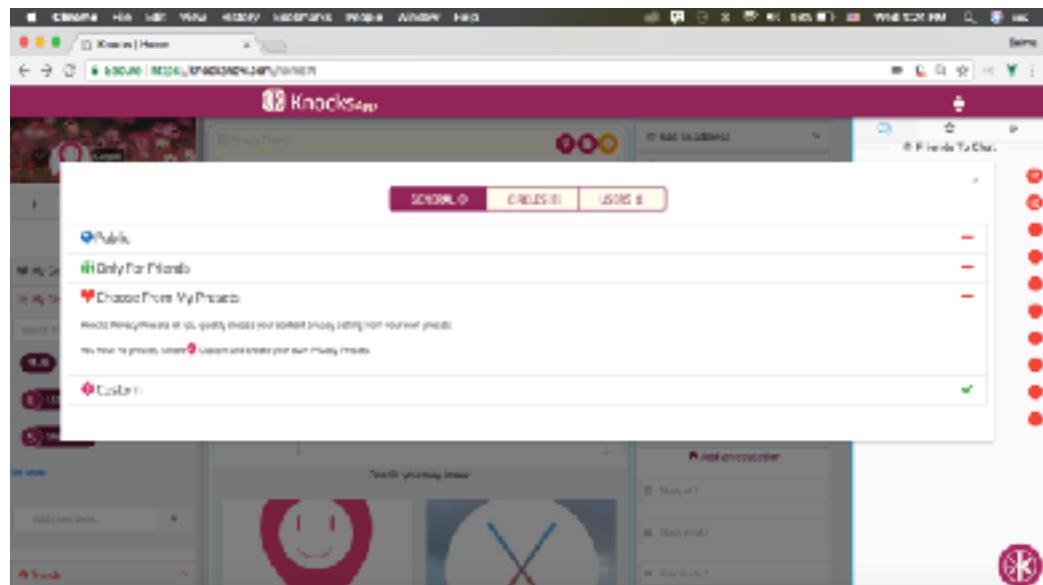
Else the system will display the new privacy set for the selected user.



- **Regulate a privacy set for circles**

At the knock section on the top of the page, at the General tab the user should click at the star icon that displays “Choose your privacy settings”, then from the options (which are : Public, Only for friends, Choose my Preset, Custom) clicks on Custom and clicks on “Circles” tab, he can choose by the dropdown list (Valid, Invalid, Invalid for all) or the icons if the selected Circles can see his activities or not, and if the desired Circle is not shown he can search for it.



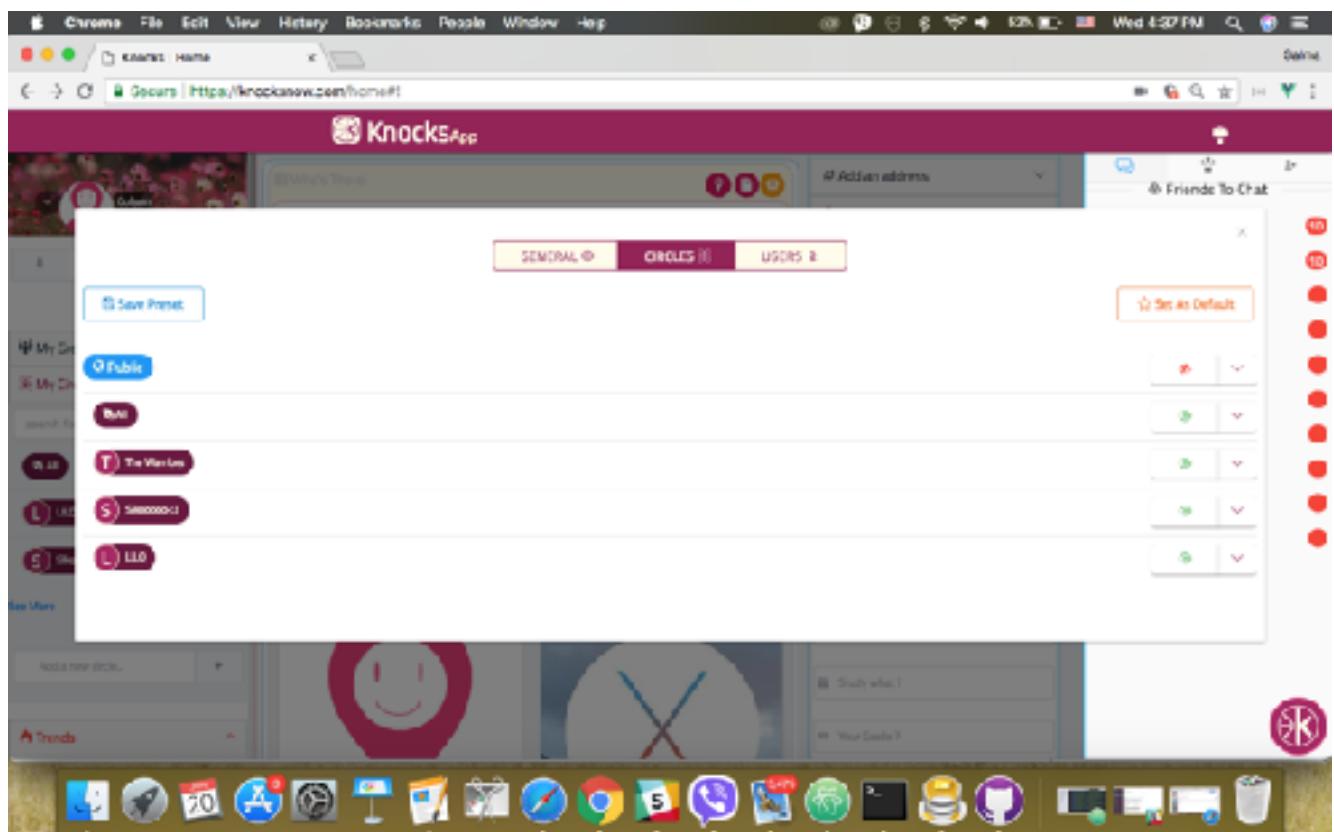


If

the date are incorrect (searching for a circle that the user has not created), the system will not display any circle to choose privacy set for him.

Still not fixed

Else the system will display the new privacy set for the selected user.

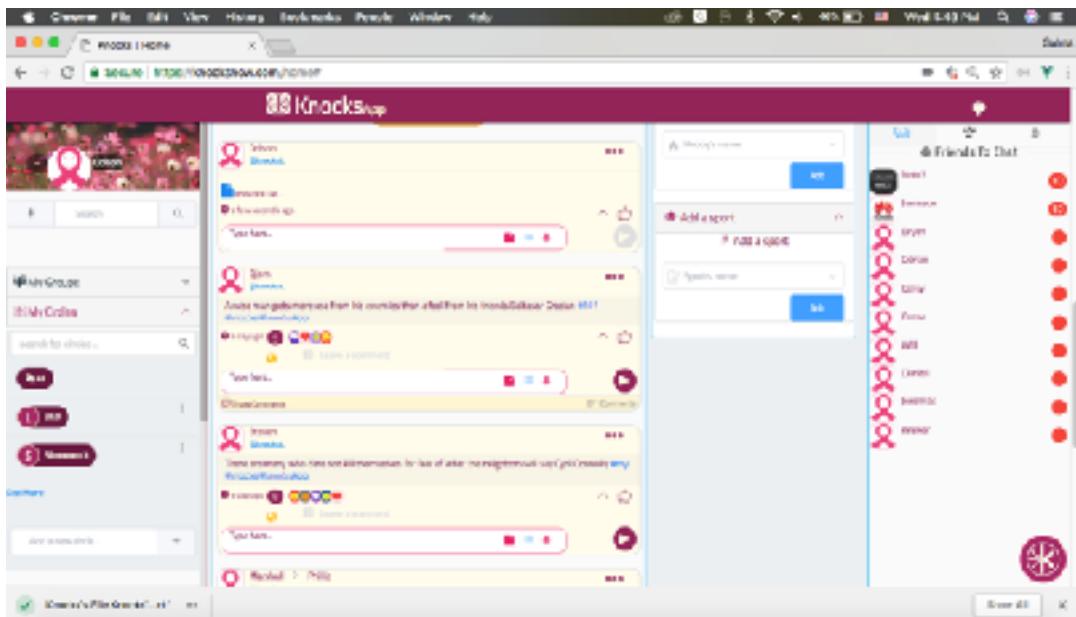


- Regulate a parent object privacy set for circle.

Not yet a ready function

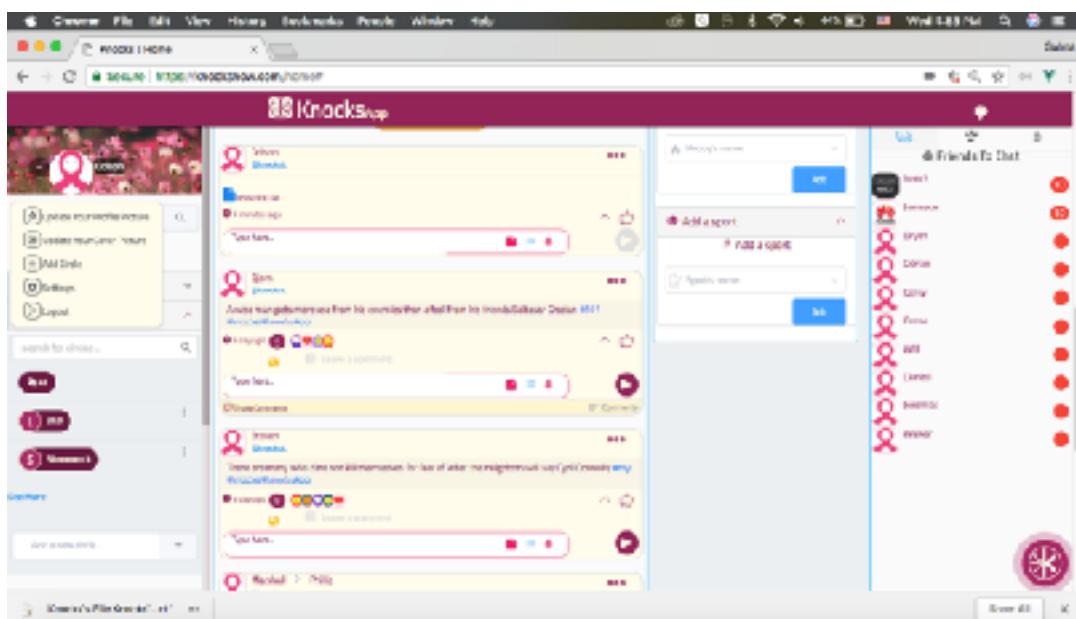
- **Download files (with any extension).**

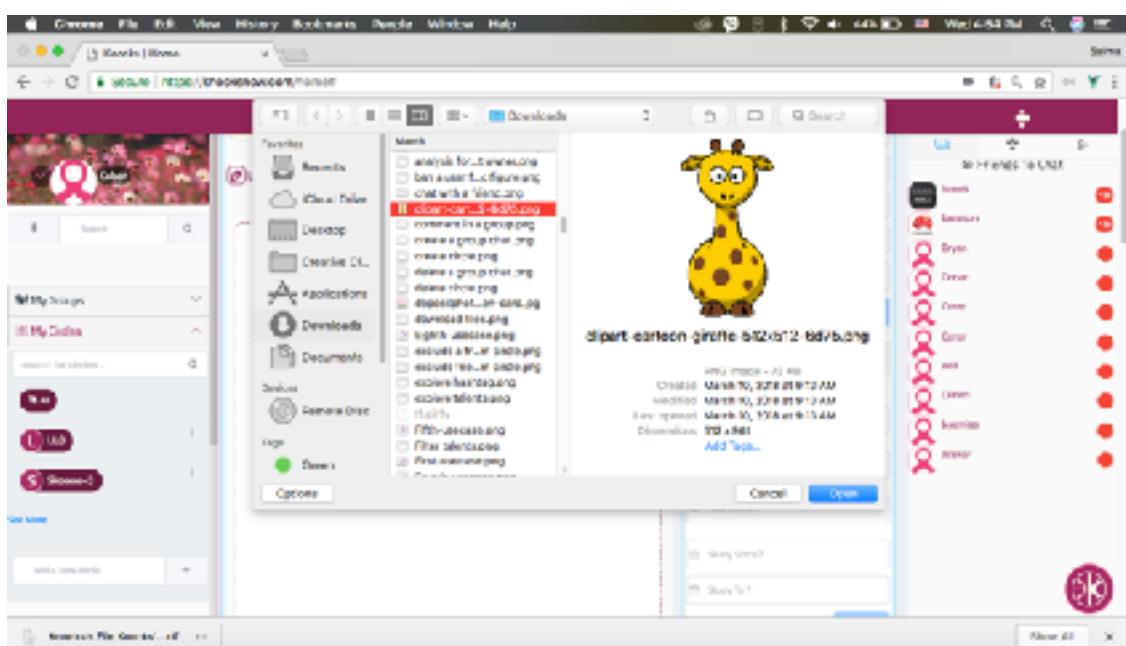
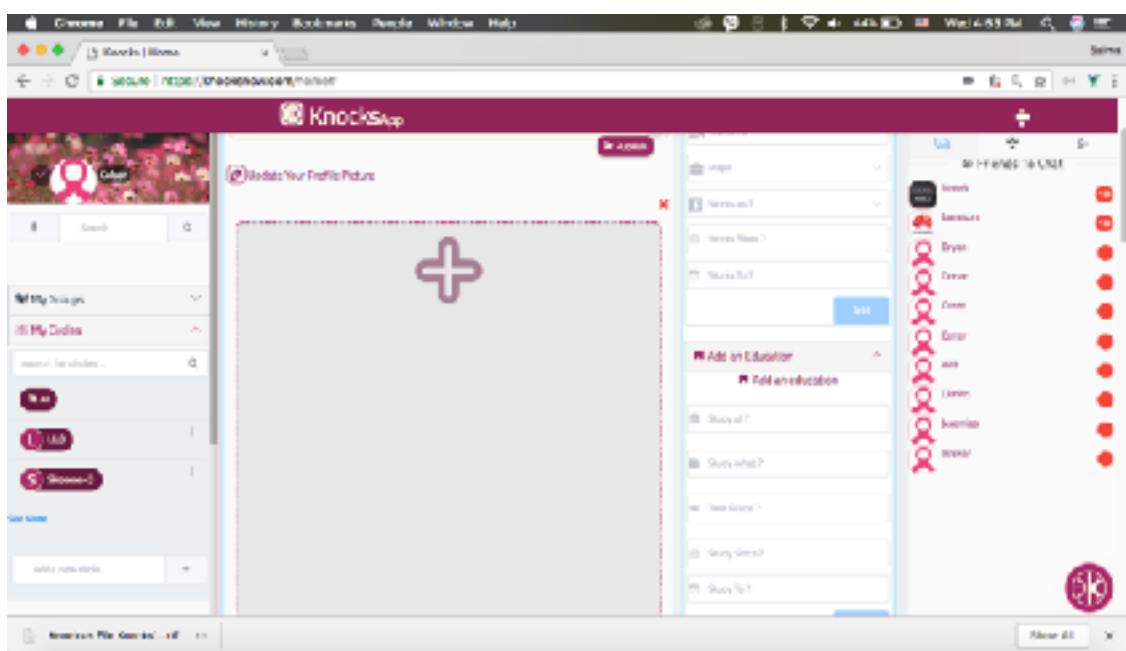
In any Knock or comment or reply where the user desires to download an uploaded file, he should click on the file to download it to his computer.

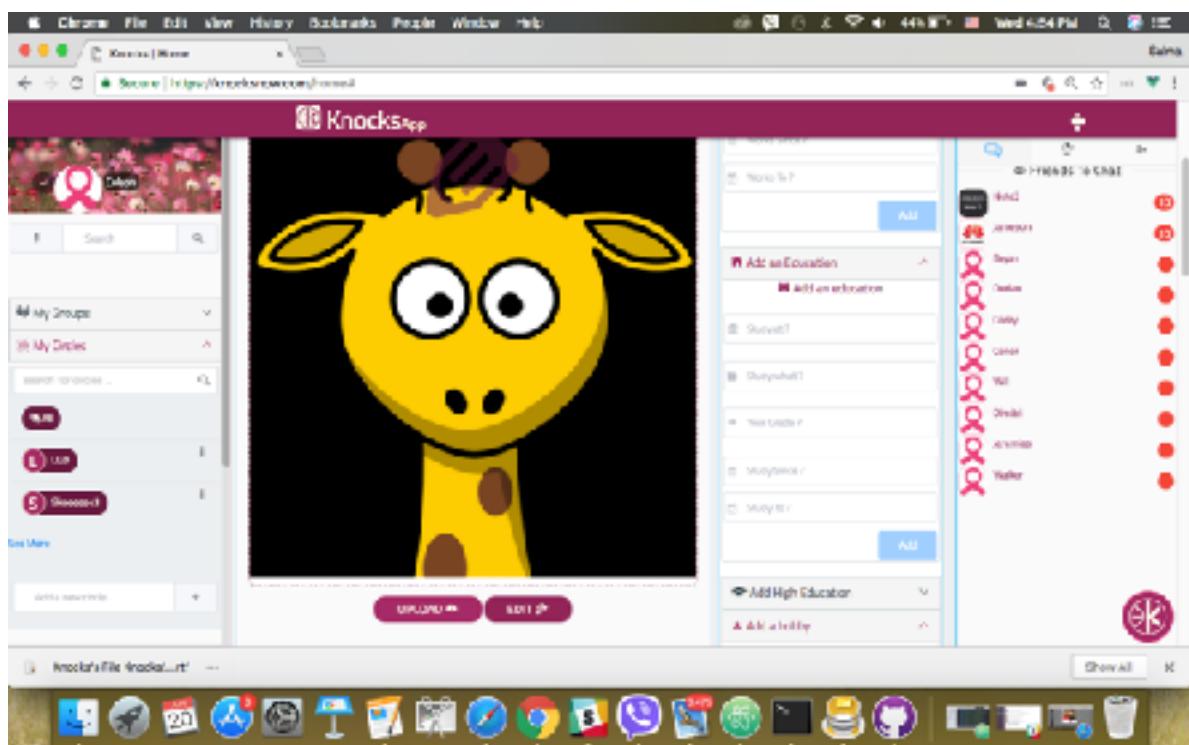
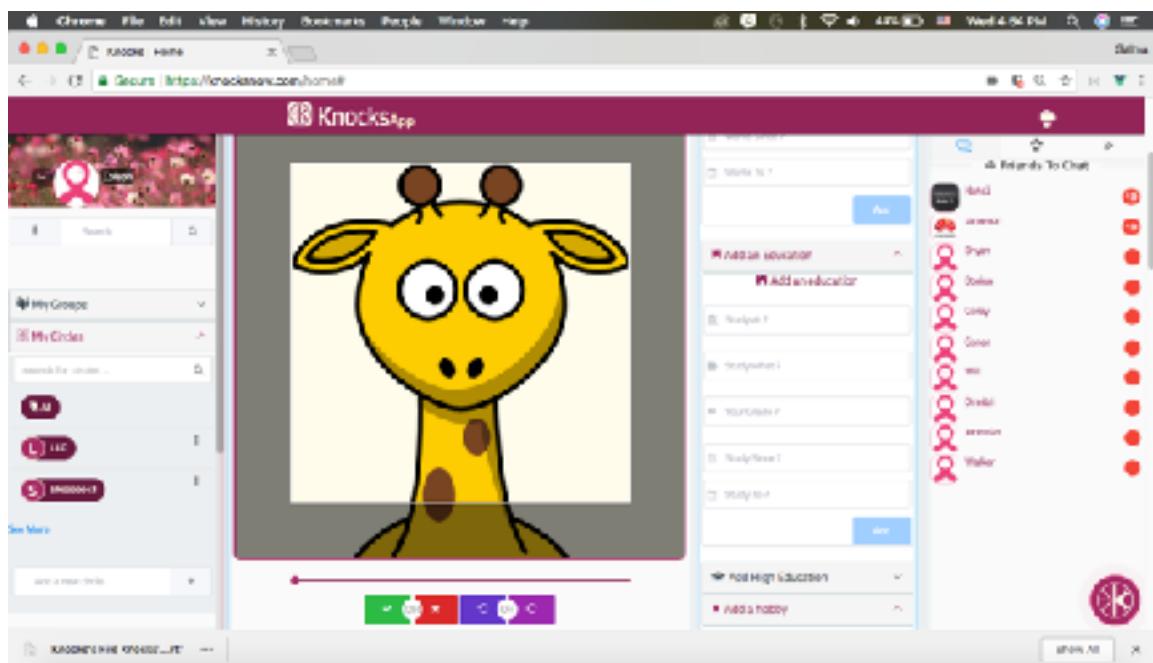


- **Upload or Update profile picture**

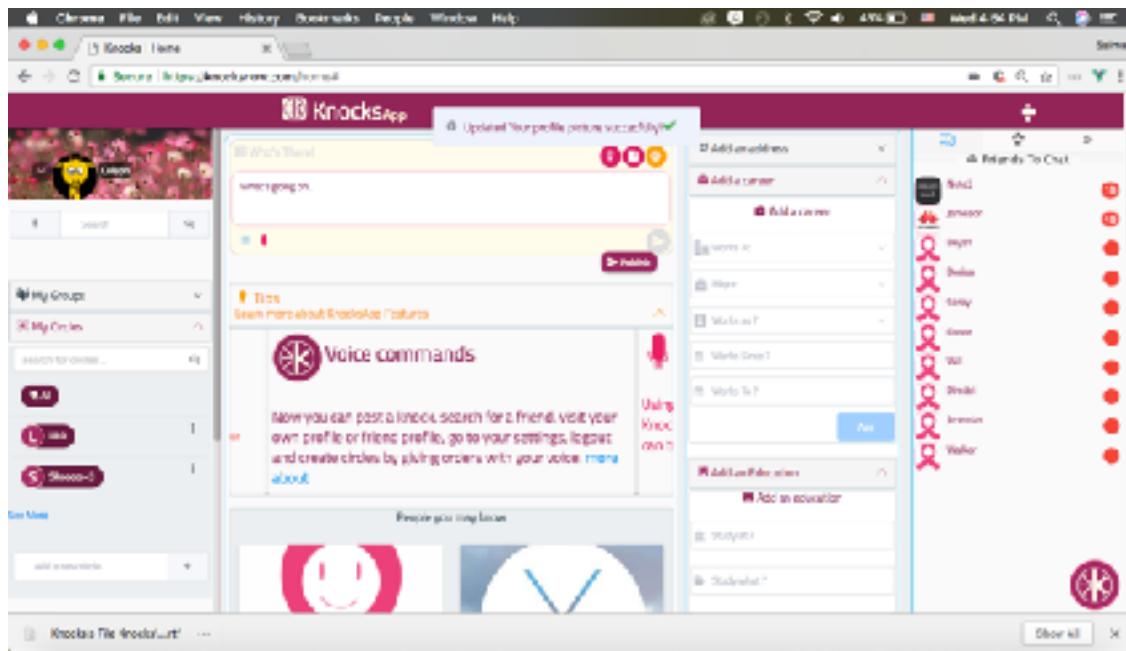
At the Profile picture section on the top left of the page, click on the arrow icon to display a dropdown list (Update your profile picture, Update your cover picture, Add circle, Settings, Logout) to choose “Update your profile picture”, then he clicks on the add icon on the middle and chooses a picture from the computer and then clicks on the green tick icon if he does not need to crop or edit the image, then clicks on “Upload” button.





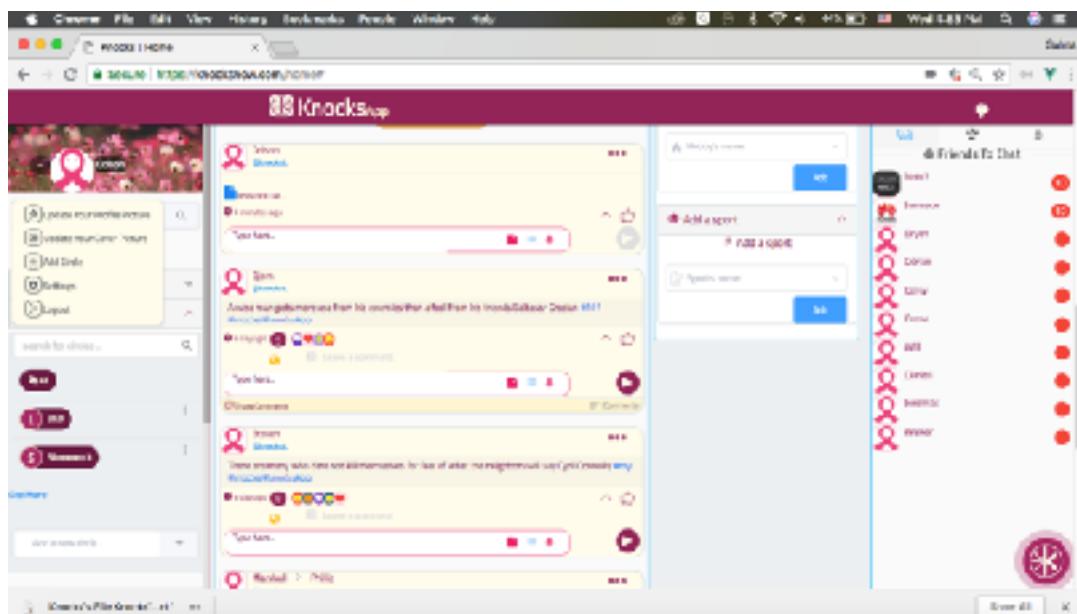


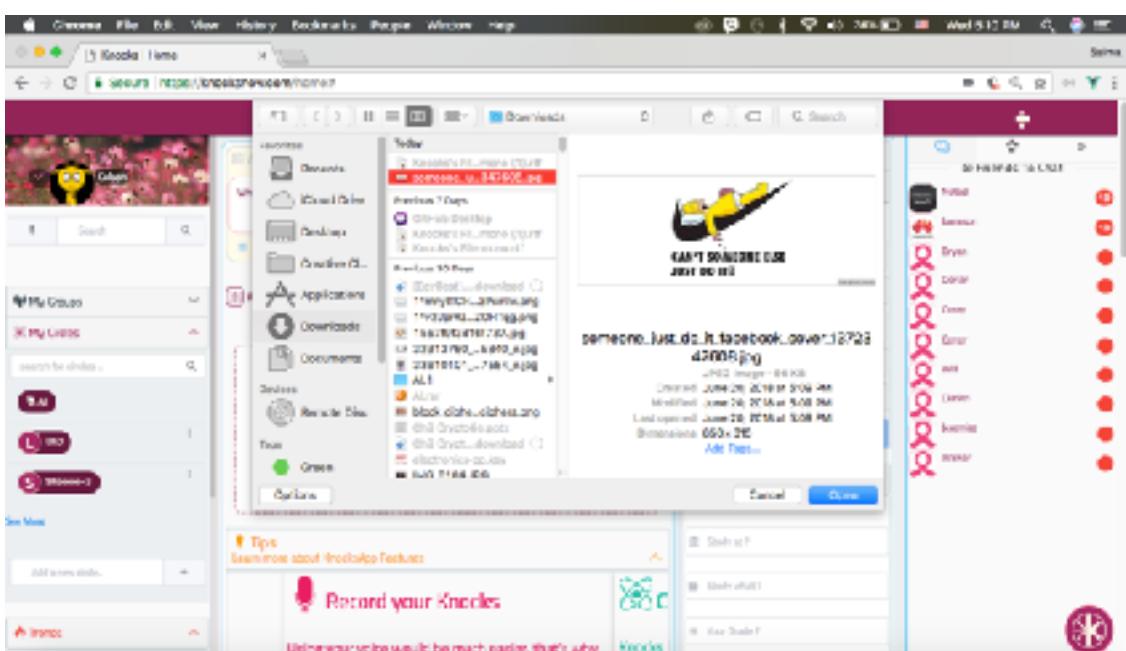
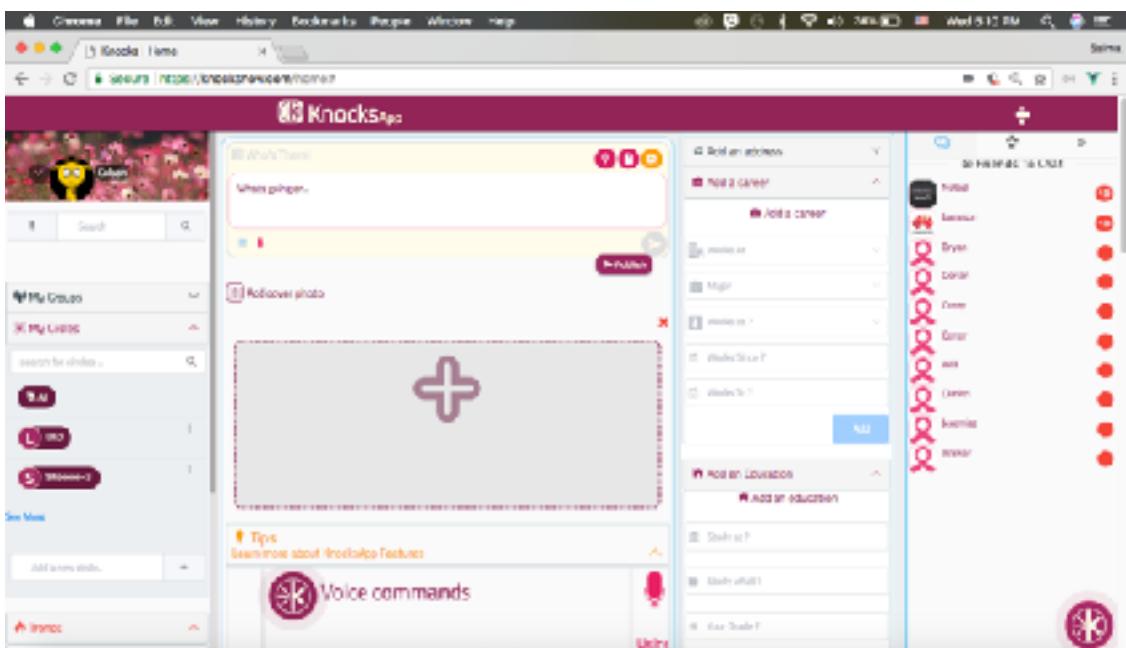
Then the system will display a message “Updated your profile picture successfully” and it will display the profile picture of the user instead of the old one.

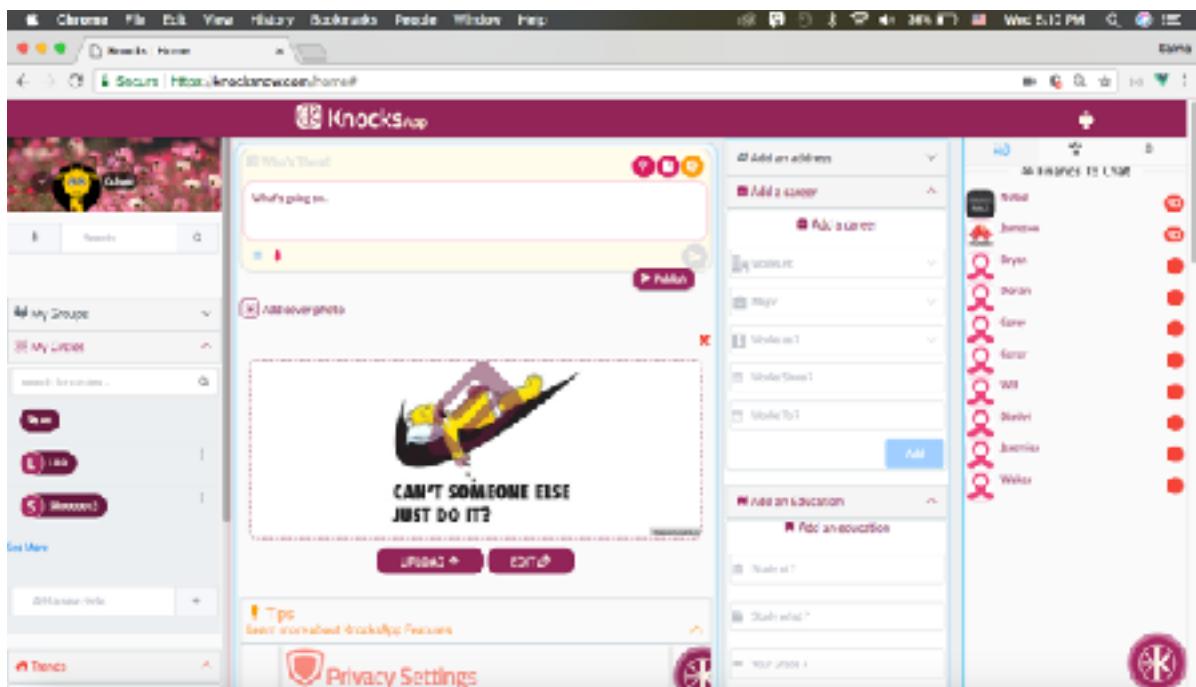
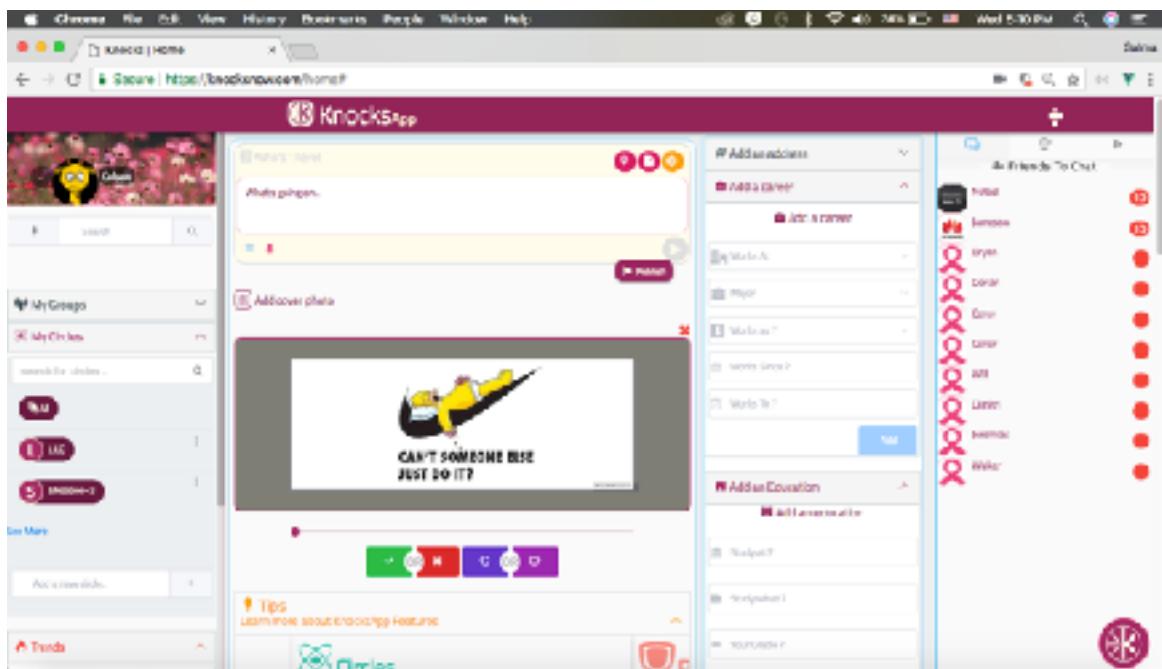


- **Upload or Update cover picture**

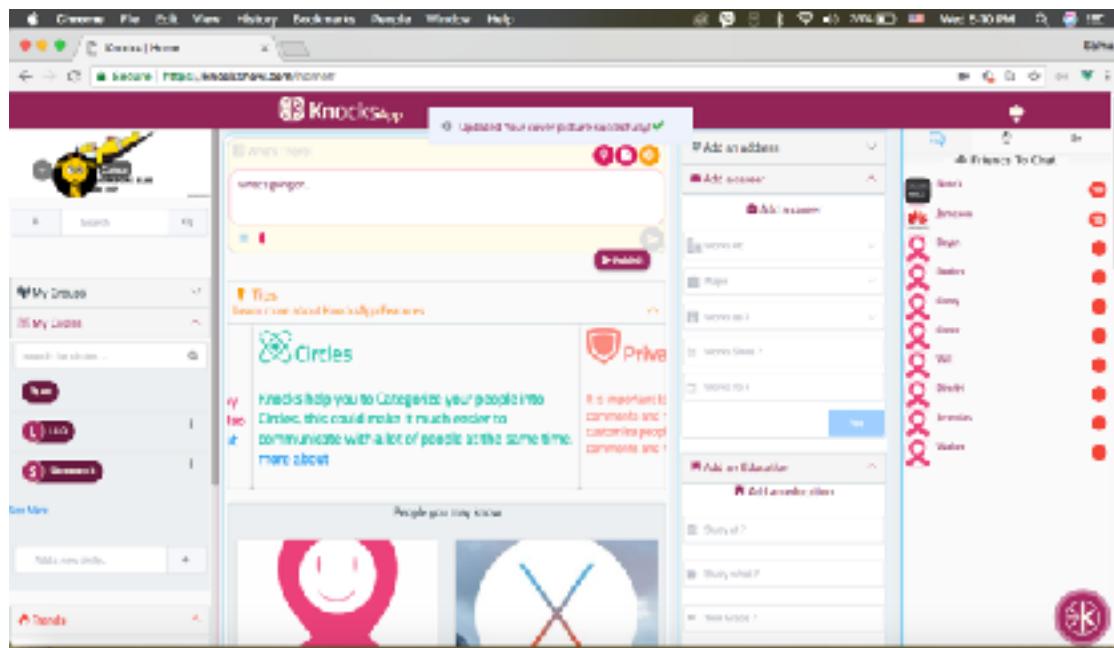
At the cover picture section on the top left of the page, click on the arrow icon to display a dropdown list (Update your profile picture, Update your cover picture, Add circle, Settings, Logout) to choose “Update your cover picture”, then he clicks on the add icon on the middle and chooses a picture from the computer and then clicks on the green tick icon if he does not need to crop or edit the image, then clicks on “Upload ” button.





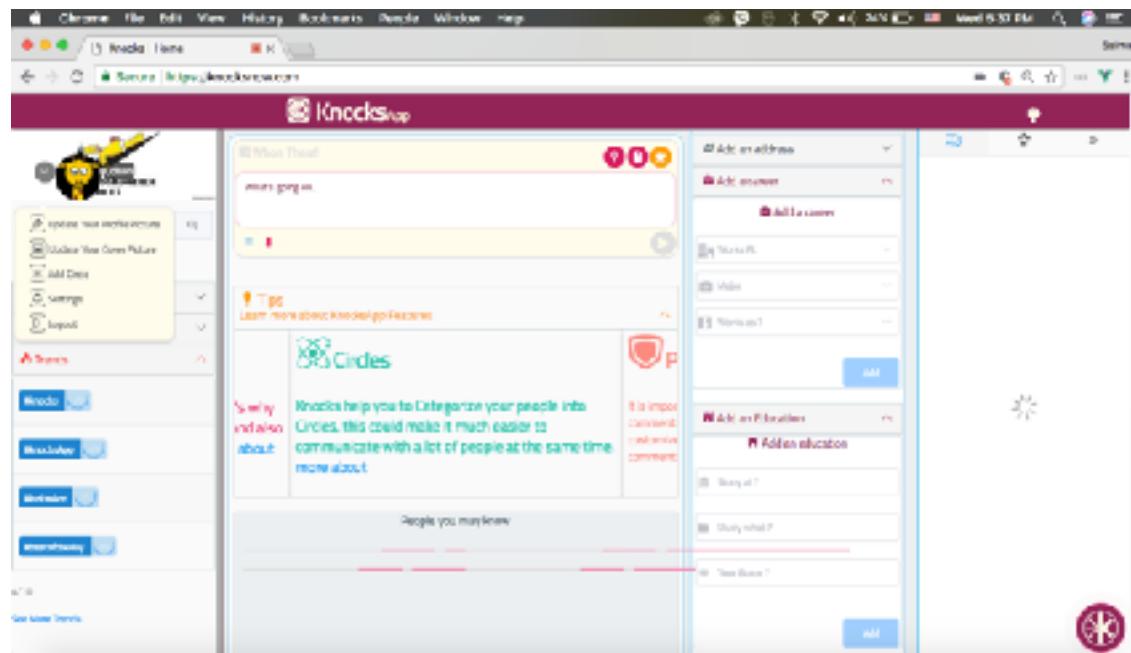


Then the system will display a message “Updated your cover picture successfully” and it will display the cover picture of the user instead of the old one.



• Block a friend

At the cover picture section on the top left of the page, click on the arrow icon to display a dropdown list (Update your profile picture, Update your cover picture, Add circle, Settings, Logout) to choose “Settings”, then he clicks on the left side bar at “Blocking” tab, and search for friends then clicks the red icon of lock.



The screenshot shows the KnockApp user settings interface. On the left, there's a sidebar with navigation links: My Groups, My Orders, Friends, Knocks (selected), KnockBooks (disabled), KnockBooks Pro (disabled), and See More Tools. The main area has a search bar and a list titled 'People' with the following items:

- Mr. People
- Mr. Order
- Mr. Group
- Elmway
- Planning**

A message at the bottom says "empty." and "Add another item". To the right, there's a sidebar titled "All Friends To Chat" with a list of friends: Paul, Connor, Jim, Brian, Sean, Peter, Sam, Gamma, Guest, and Jeffcoast. At the bottom right is a circular logo with a stylized 'K'.

This screenshot is identical to the one above, except the search bar now contains the text "pl". The list of people includes "Planning" along with the others. The rest of the interface and sidebar are the same.

If the date are incorrect (searching for a user who is not a friend), the system will not display any user to be blocked.

The screenshot shows the KnockApp interface. On the left, there's a sidebar with options like 'My Groups', 'My Chats', 'Trends', and 'KnockApp News'. The main area is titled 'Settings' and has a search bar with 'daniel' typed in. Below it, there are two sections: 'Blocked' (empty) and 'Blocked Since' (empty). On the right, there's a list of friends with red status dots, and a 'Blocked' section with a green lock icon next to 'daniel'.

Else the system will block the selected user and display a message “Blocked successfully” and displays the green lock beside the blocked username for unblocking.

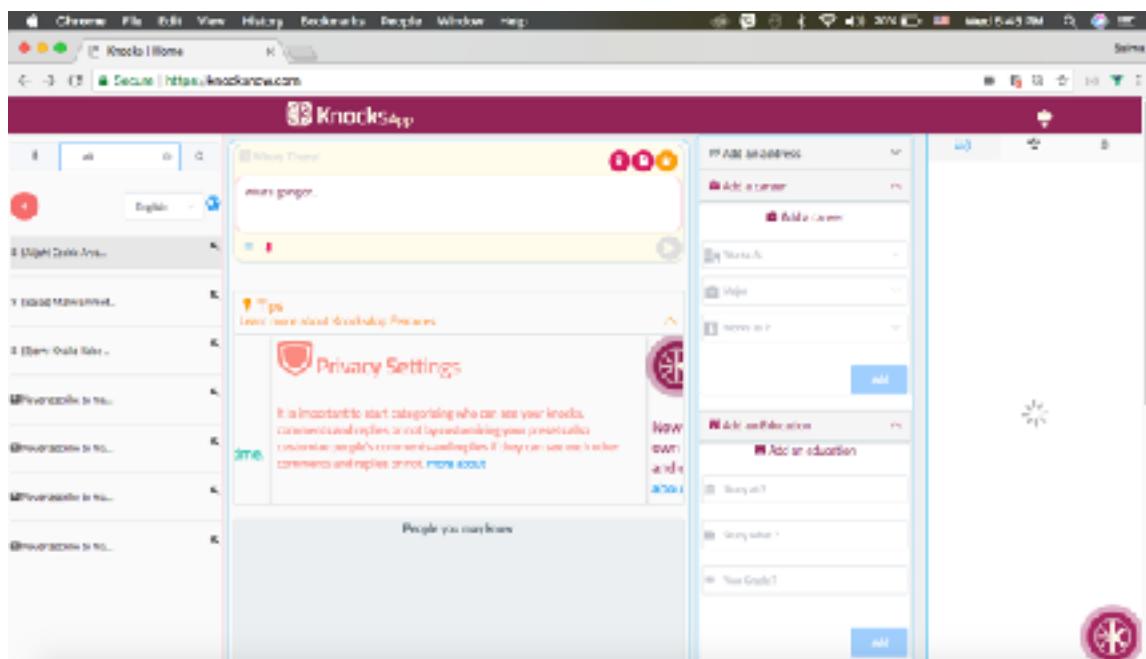
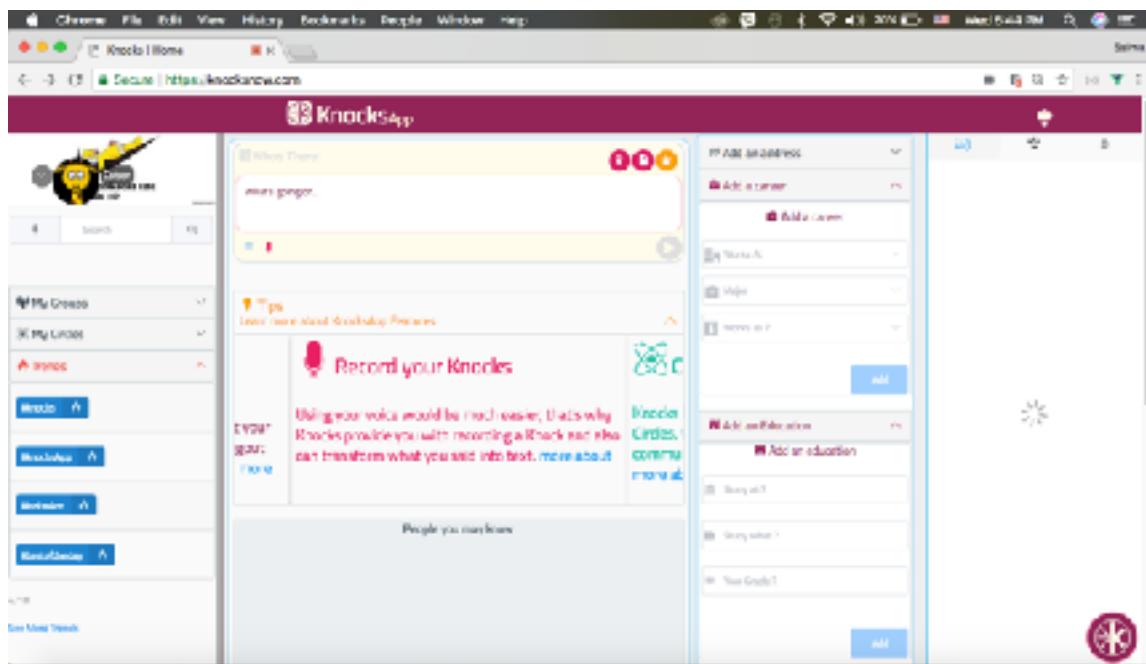
This screenshot is similar to the previous one, but it shows a success message: 'Blocked successfully' with a checkmark icon. The 'Blocked' section now contains 'daniel' with a green lock icon. The 'Blocked Since' section also shows 'daniel' with a green lock icon.

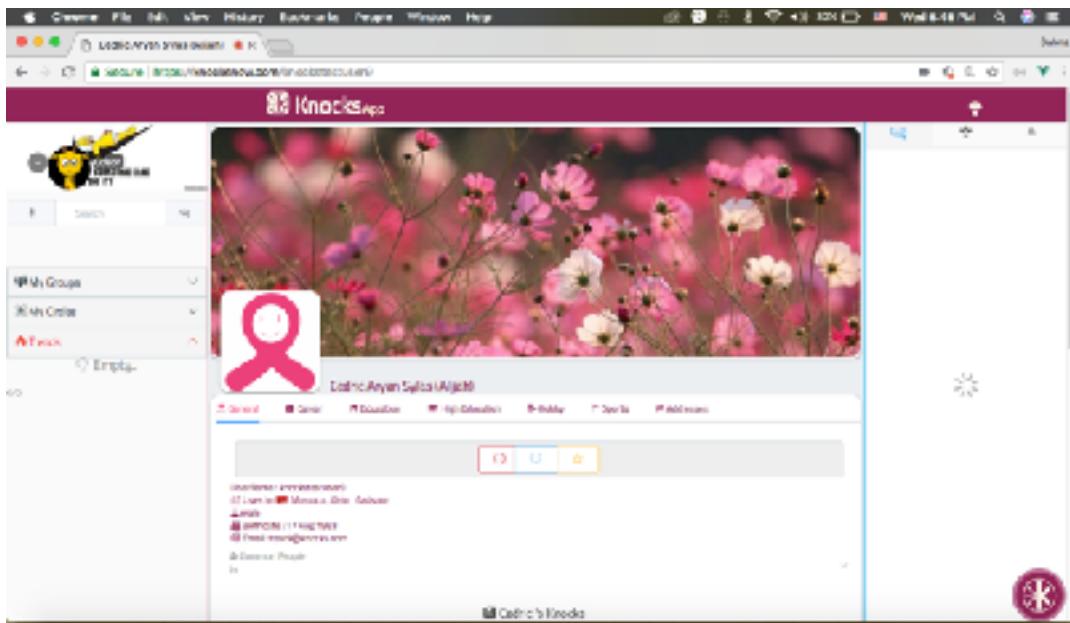
- Report a friend

Not yet a ready function

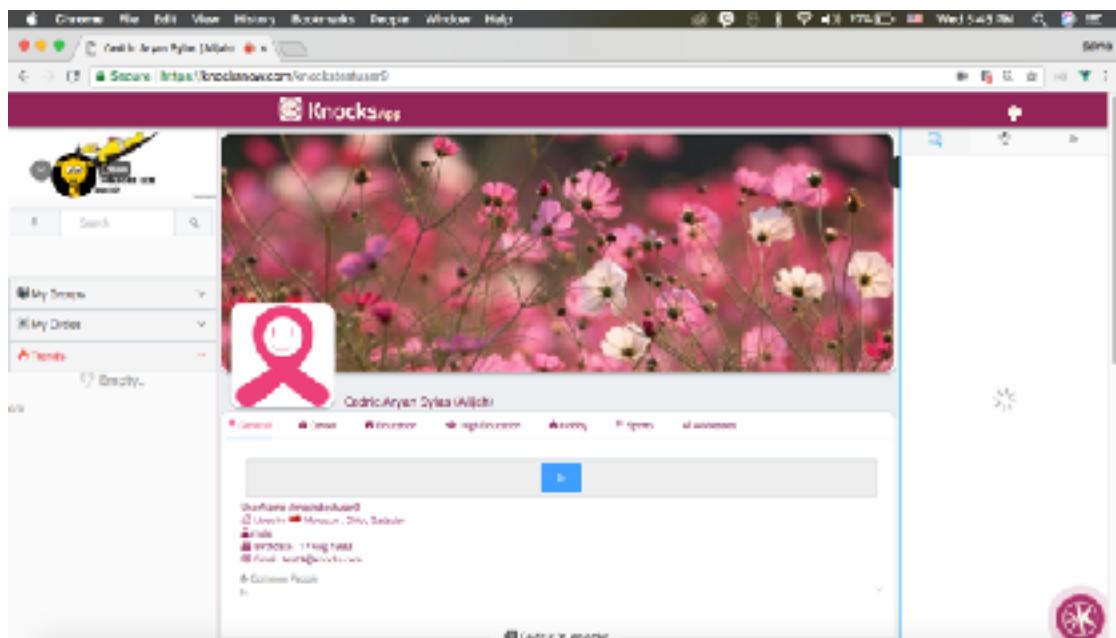
- **Unfriend a friend**

The user can search for friends in the left side bar and clicks on the user chip to be directed to his profile or just click on his username in a knock, comment, reply or any activity where his username appears, then on the user profile at the middle of it, the user clicks on red left icon which has banning logo.





Then the system will display the blue icon of adding friends which indicated that this user is no longer a friend and according to the other user's privacy settings his activities will be displayed.

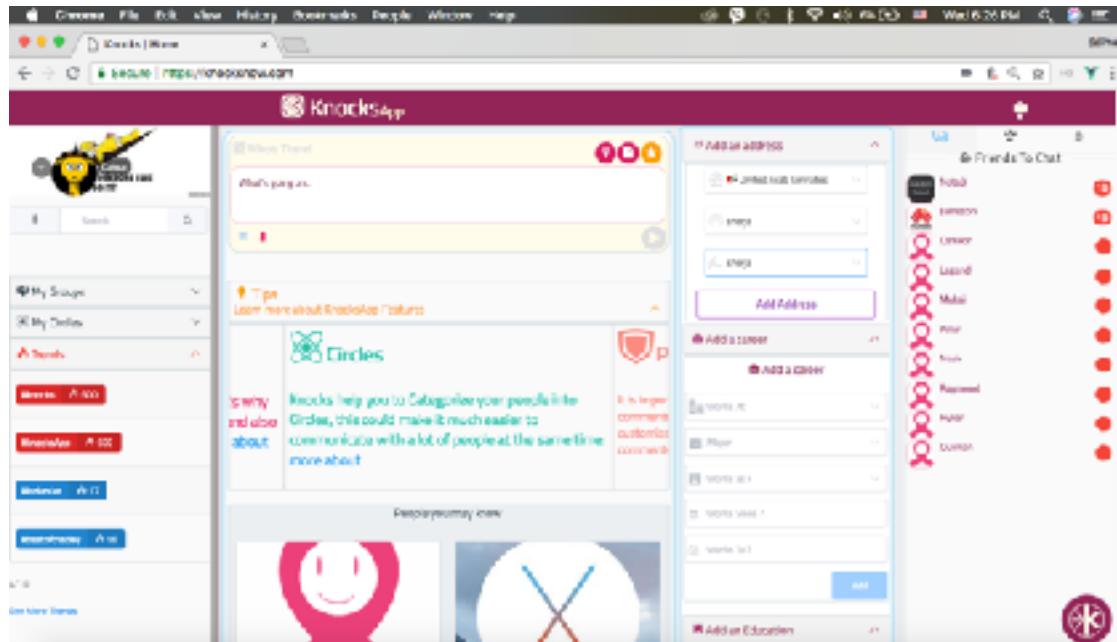


- Suggest a content (in case of adult user to his kid account).

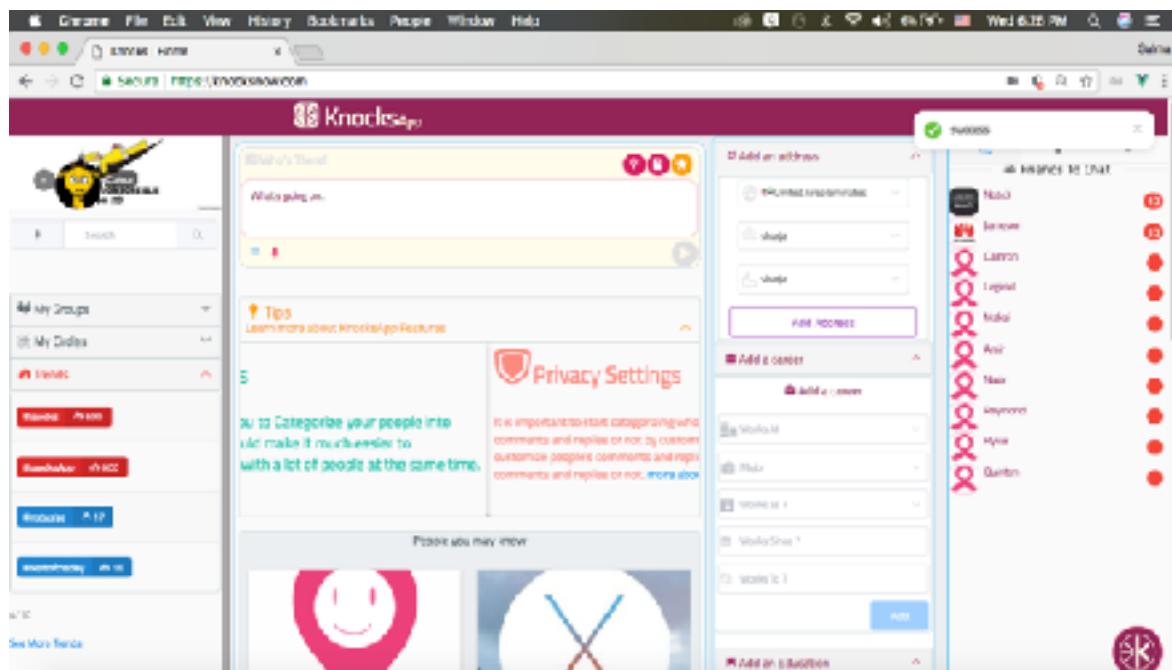
Not yet a ready function

- **Add or update personal information**

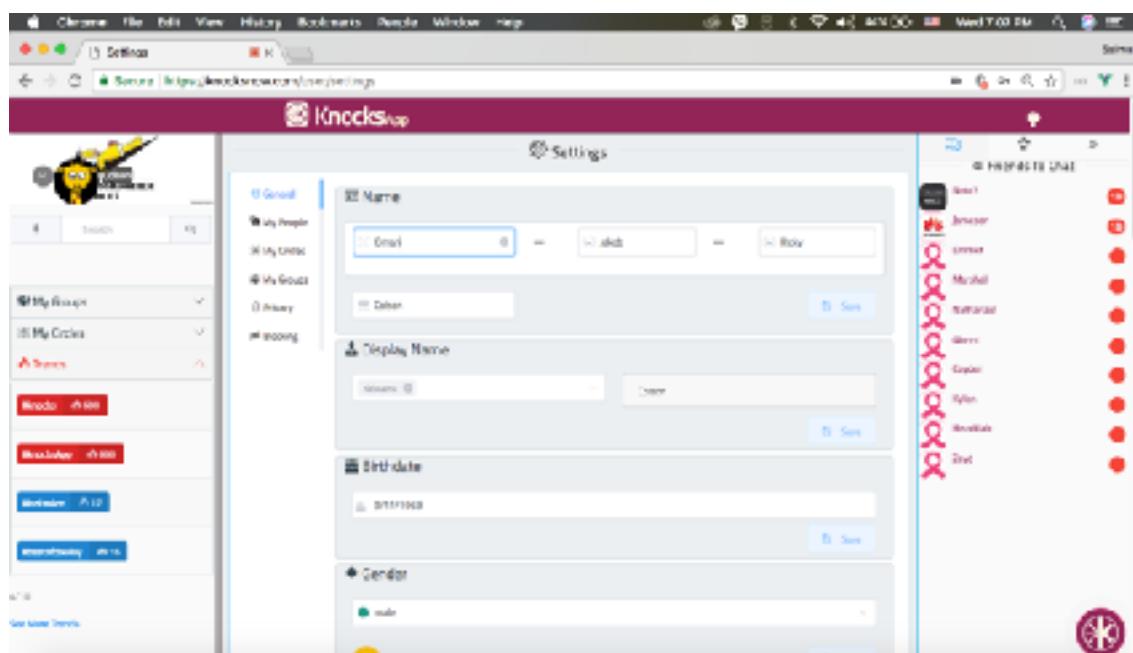
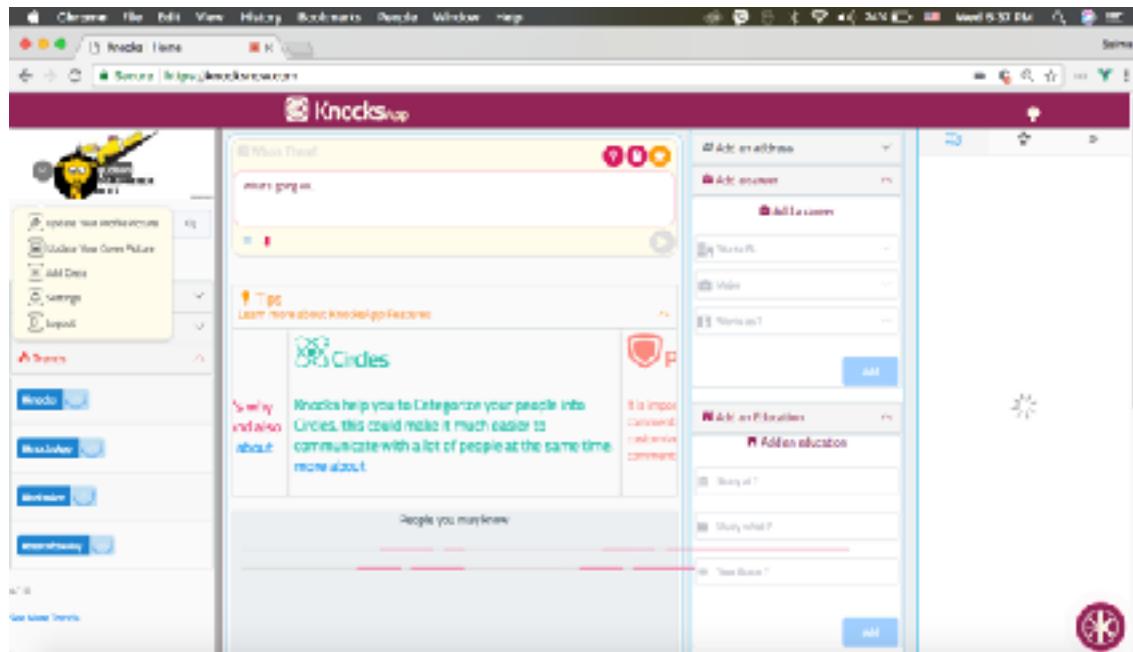
At the right corner of the home page, the user can update the personal information like the address, career, education, high education, sport and hobby, for example in adding address the user should provide the country, state and region then clicks on “Add” button.



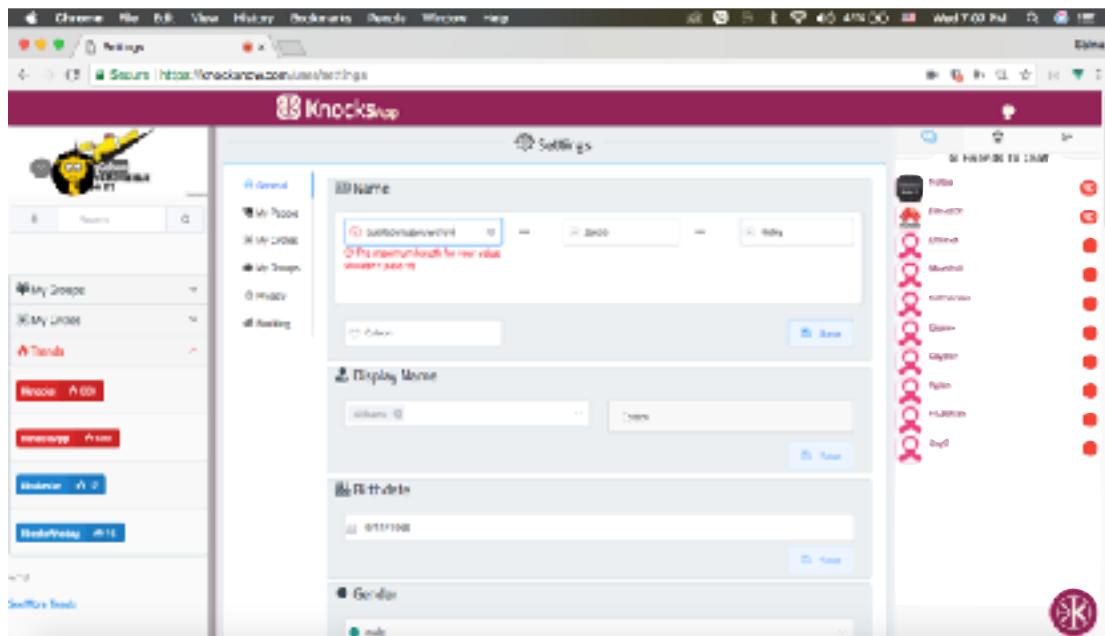
Then the system will display a message “Success” and displays the new address at the user’s profile.



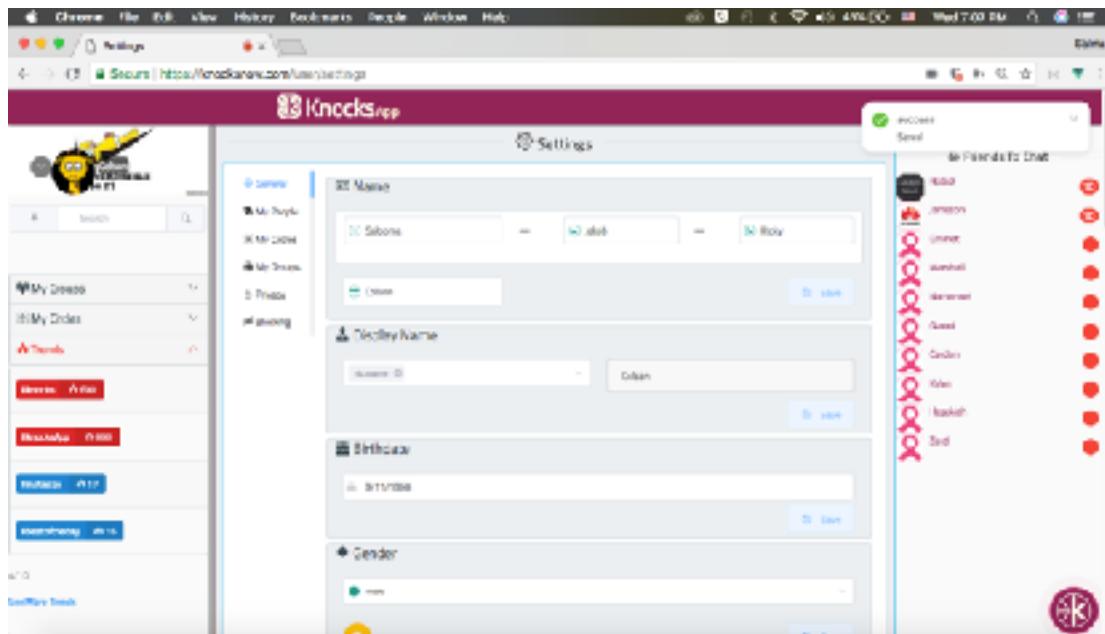
Or at the cover picture section on the top left of the page, click on the arrow icon to display a dropdown list (Update your profile picture, Update your cover picture, Add circle, Settings, Logout) to choose “Settings”, then he clicks on the left side bar at “General” tab, and chooses for example the first name to edit, then clicks “Save” button .



If the date are incorrect (providing a first name that exceeds 15 characters), the system will display message “The maximum length for your value shouldn't pass 15”.

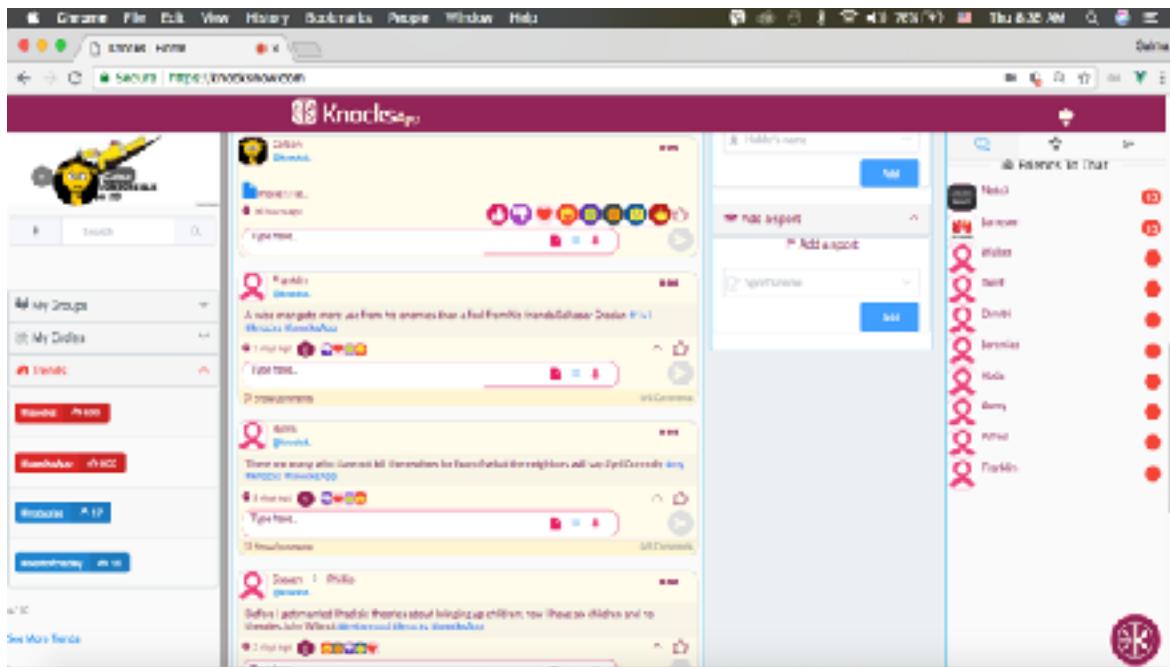


Else the system will block the selected user and display a message “Saved” and displays the the new first name for the user.

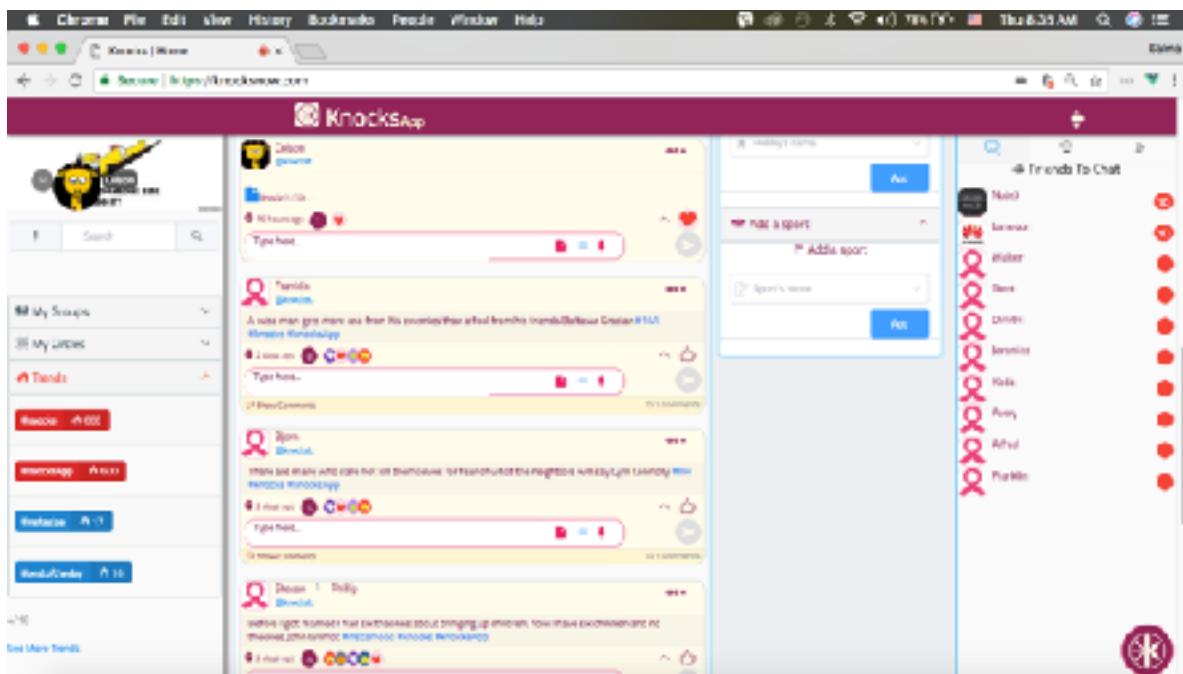


- React to post- when allowed to do so with allowed reactions.

In any Knock where the user desires to react, at right of the the knock section he should hover or click on the reaction icon, then he clicks on the reaction

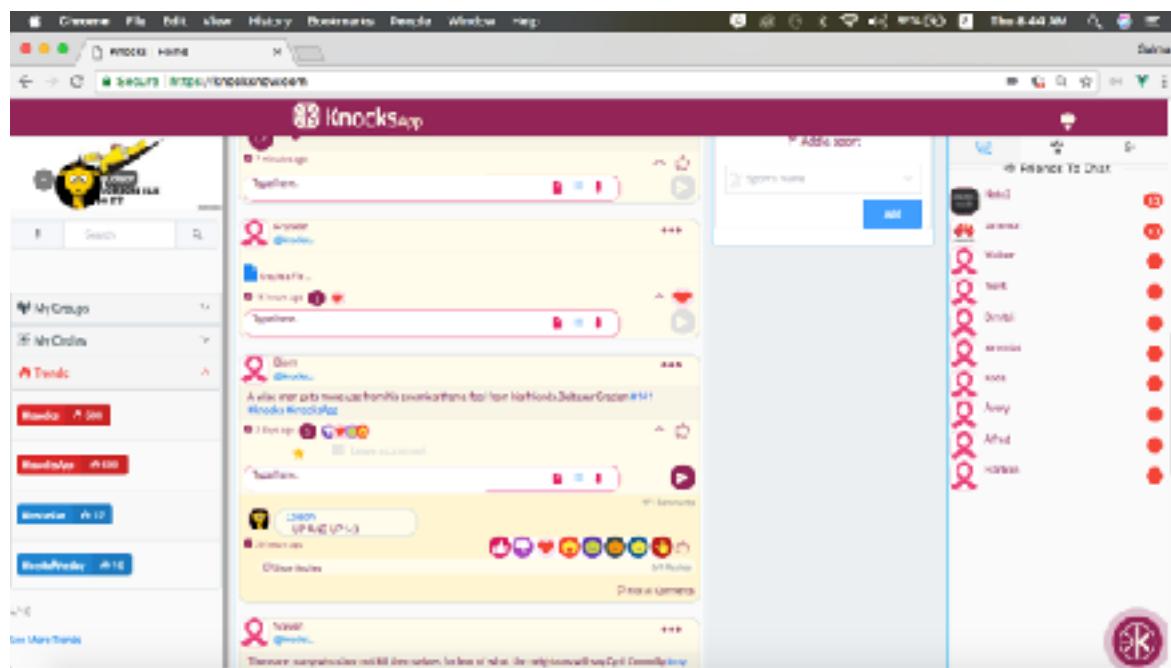


Then the system will display the new reaction towards the post and updates the reactions counter of the knock.

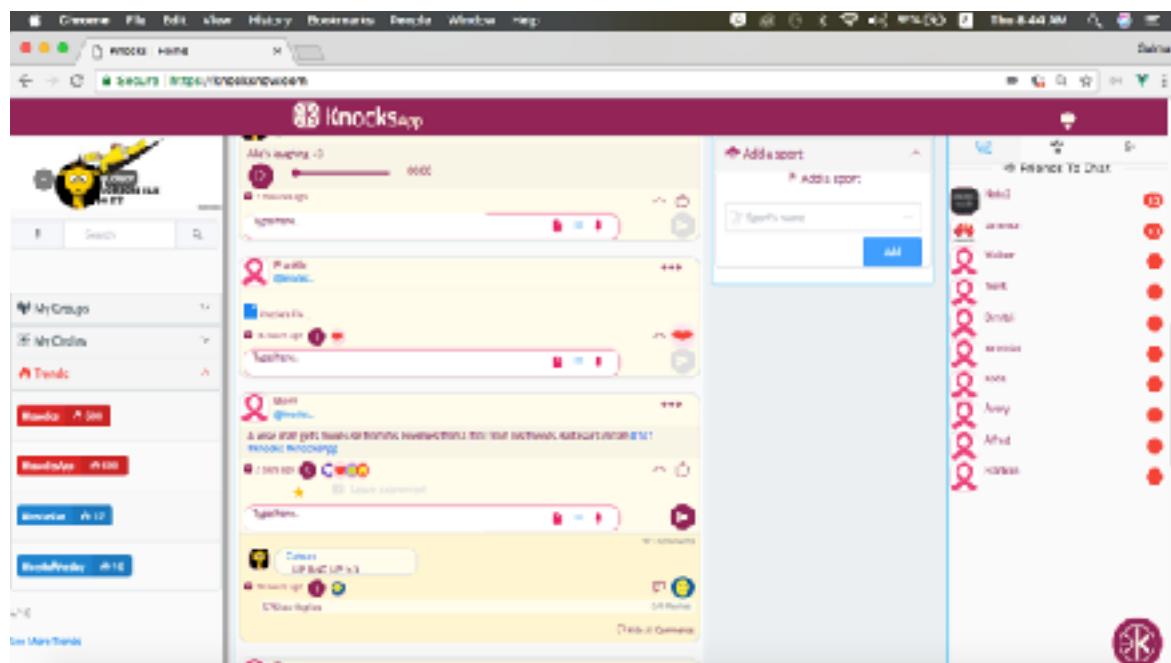


- **React to comment- when allowed to do so with allowed reactions.**

In any comment where the user desires to react, at right of the the comment section in the knock he should hover or click on the reaction icon, then he clicks on the reaction

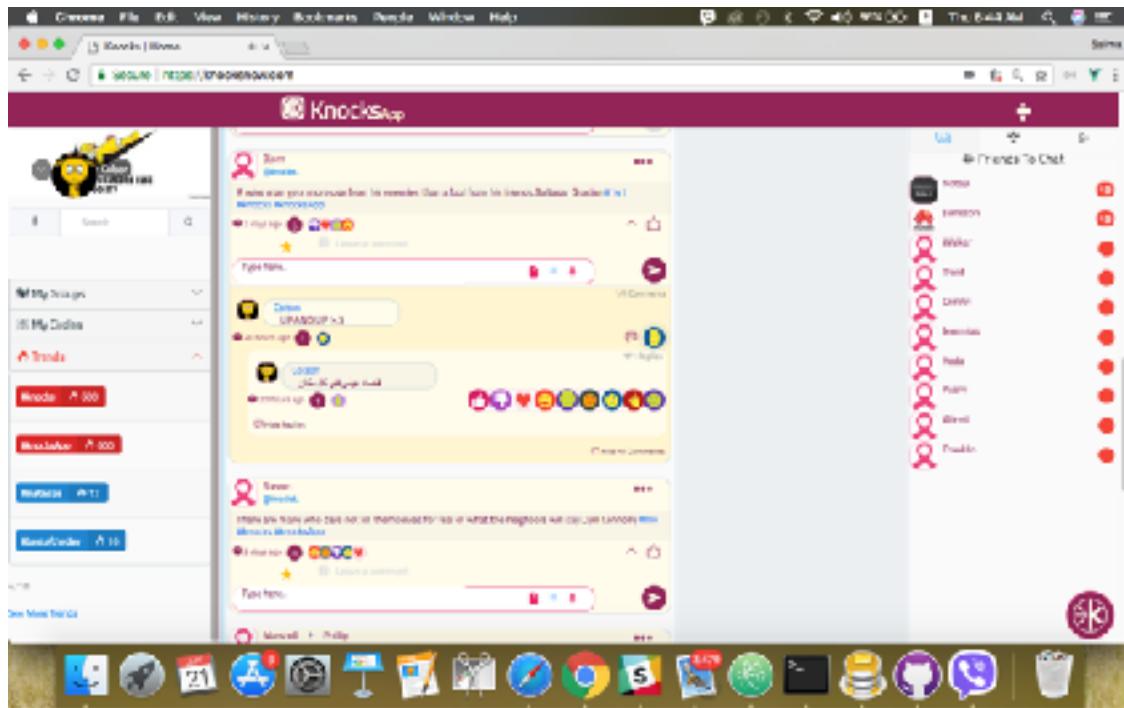


Then the system will display the new reaction towards the comment and updates the reactions counter of the comment.

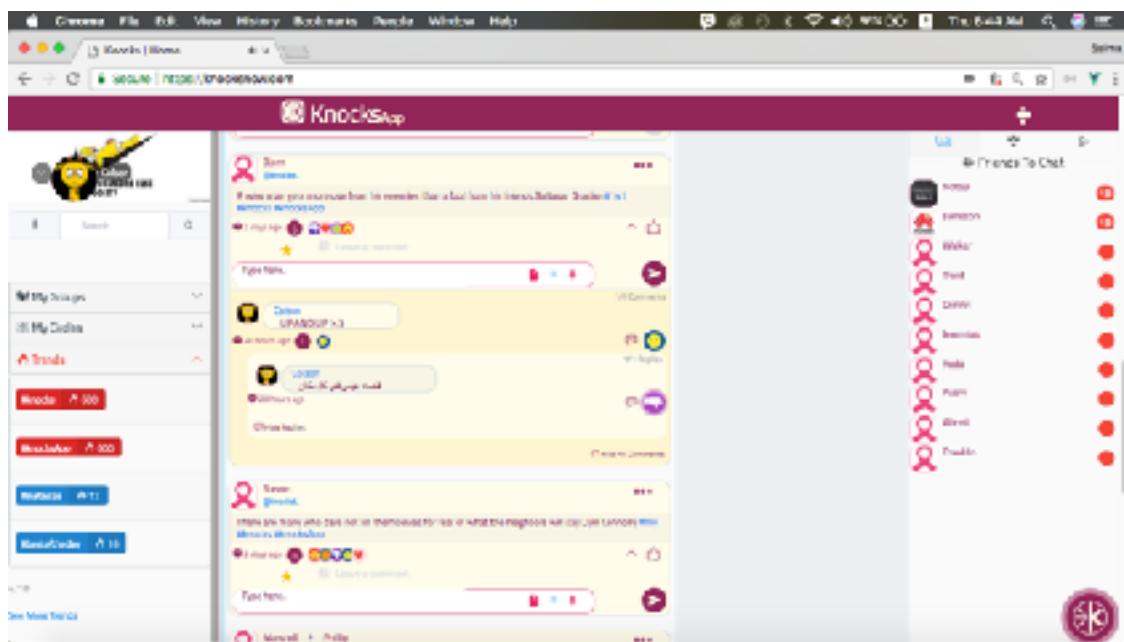


- React to reply- when allowed to do so with allowed reactions.

In any reply where the user desires to react, at right of the the reply section in the knock he should hover or click on the reaction icon, then he clicks on the reaction



Then the system will display the new reaction towards the reply and updates the reactions counter of the reply.



- Chat with a friend

Not yet a ready function

- Create a group chat

Not yet a ready function

- Update a group chat - when allowed to do so.

Not yet a ready function

- Delete a group chat- when allowed to do so.

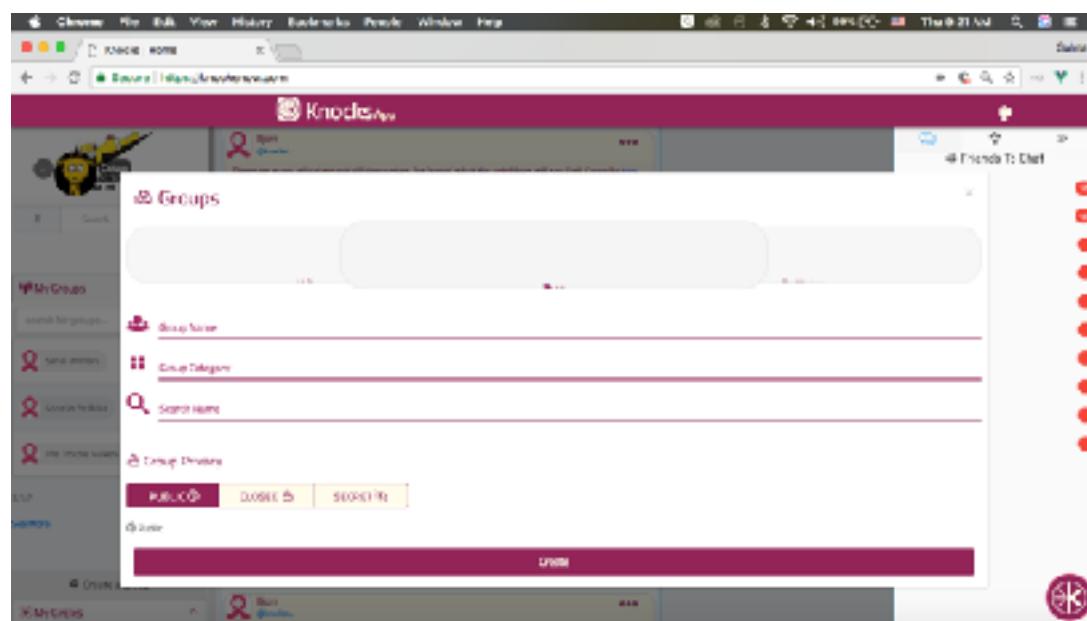
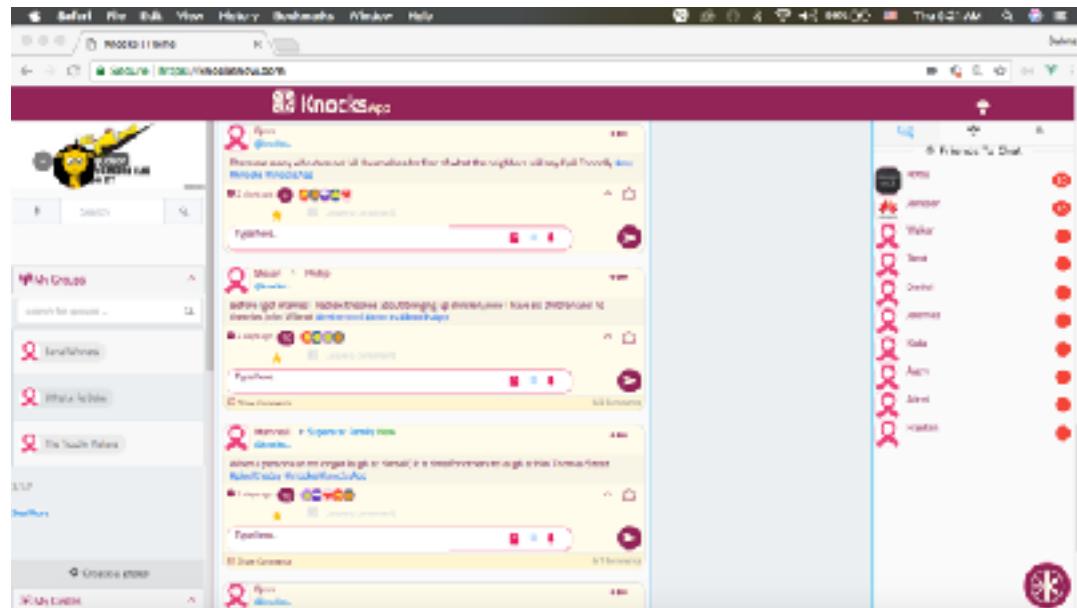
Not yet a ready function

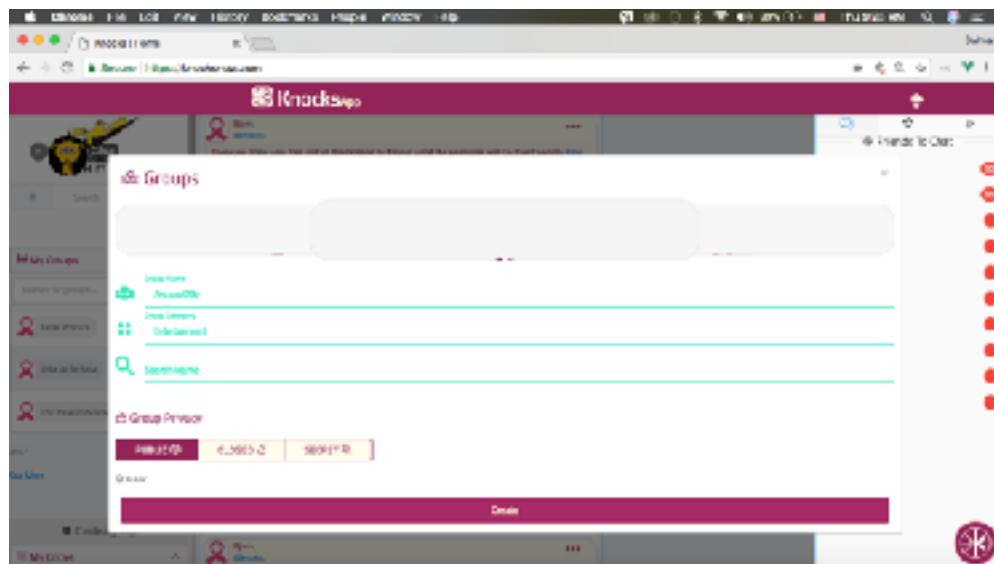
- React to messages

Not yet a ready function

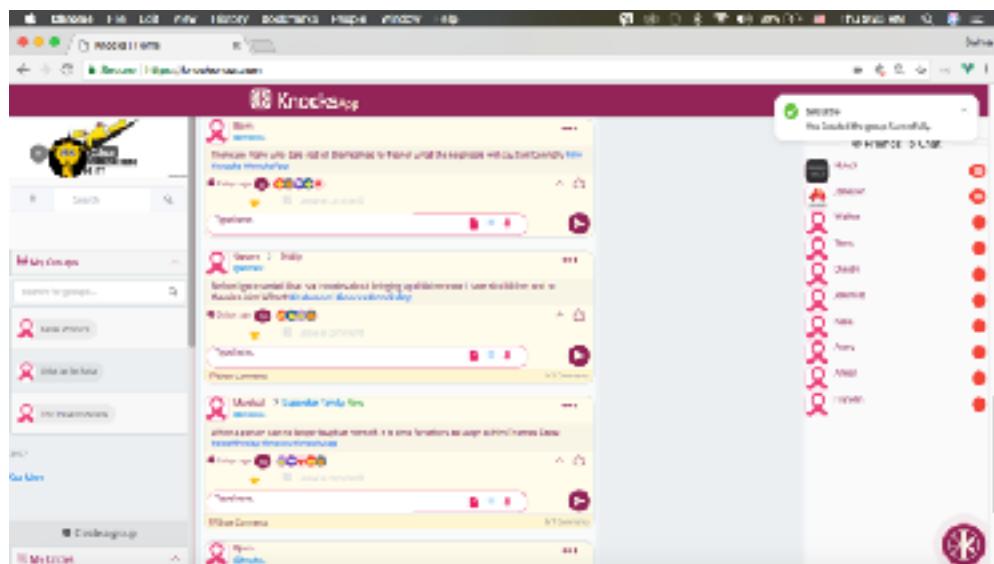
- **Create a group**

At the left side of the home page in groups section, click on “My Groups” tab and click on “Create a Group”, then provide the group name, group category, search name (Optional), and choose the privacy setting of the group (Public, Closed, Secret), and then click on “Create”.



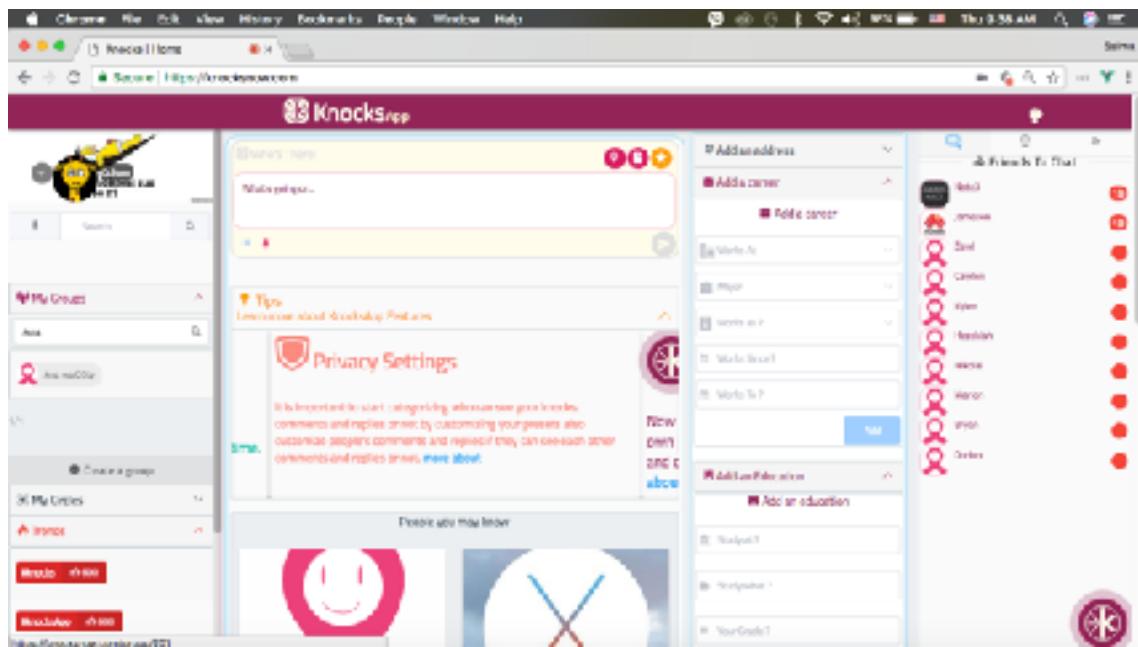


Then the system will display a message "You created the group successfully" and displays the newly created group in the groups section

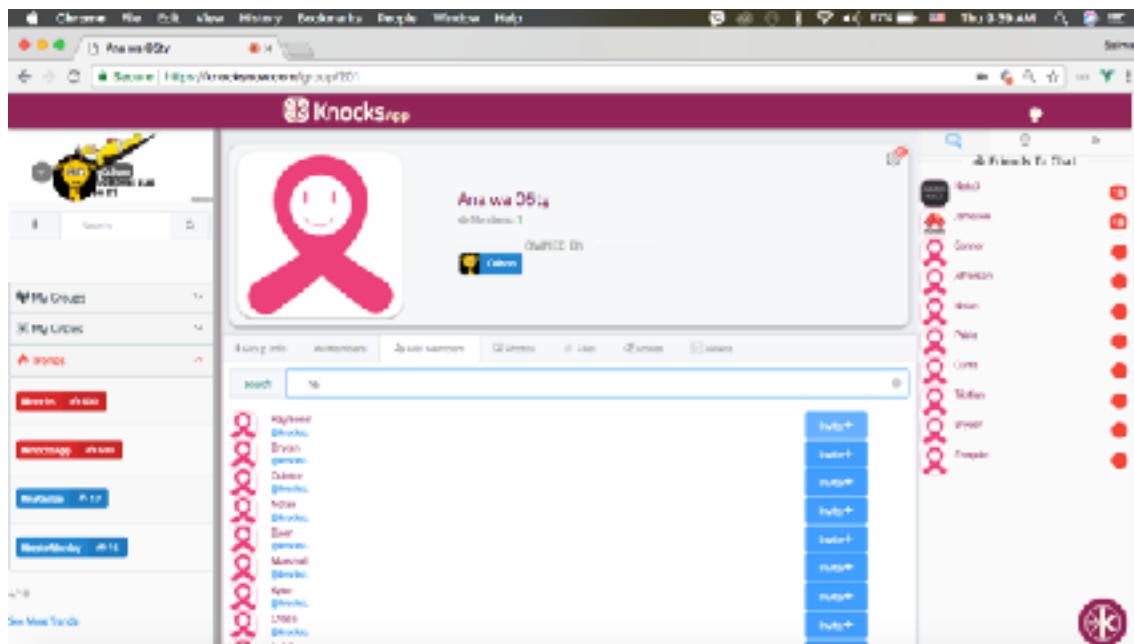


- **Add friends to group**

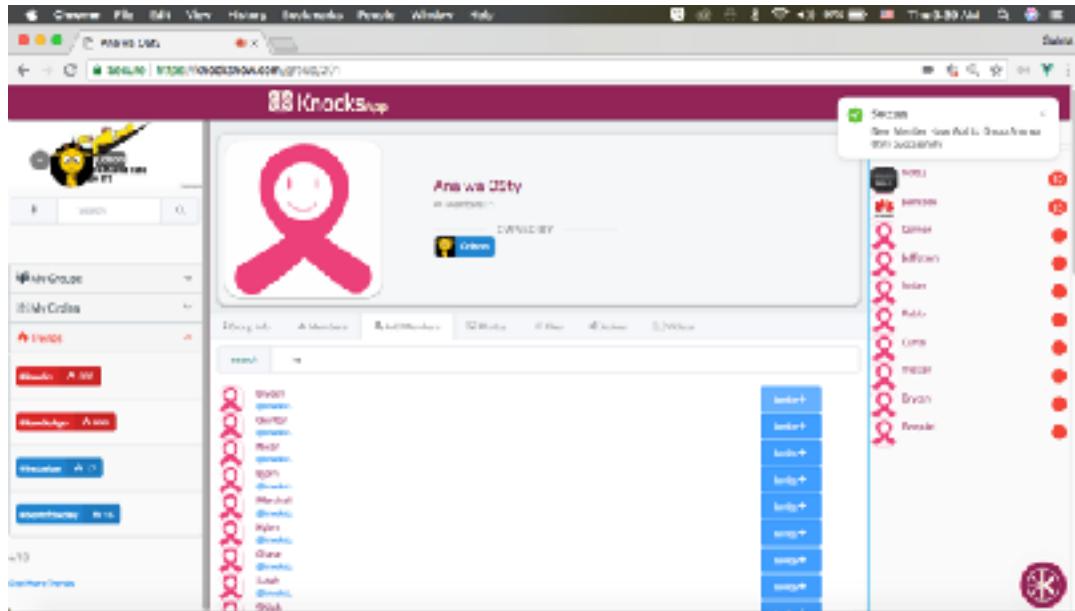
At the left side of the home page in groups section, click on “My Groups” tab and search for a group that he owns or admins.



Then click on the group name chip, it will direct the user to the group page.

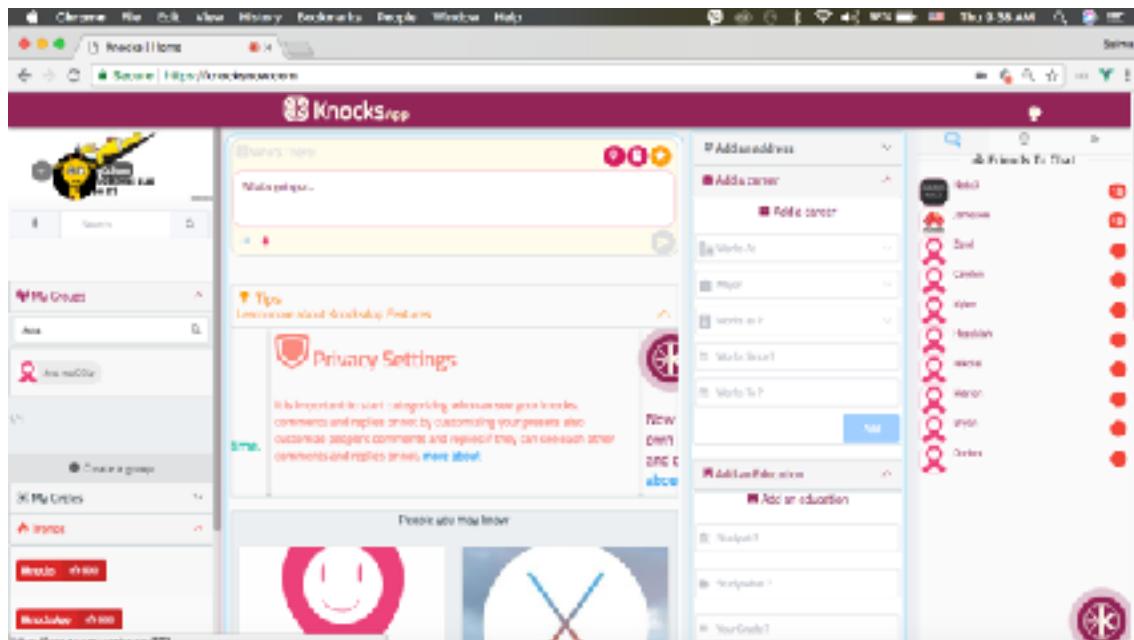


Then he should click on the “Add members” tab an search for users to add them, and clicks “Invite”, then the system will display message “New member has add to the group successfully”.

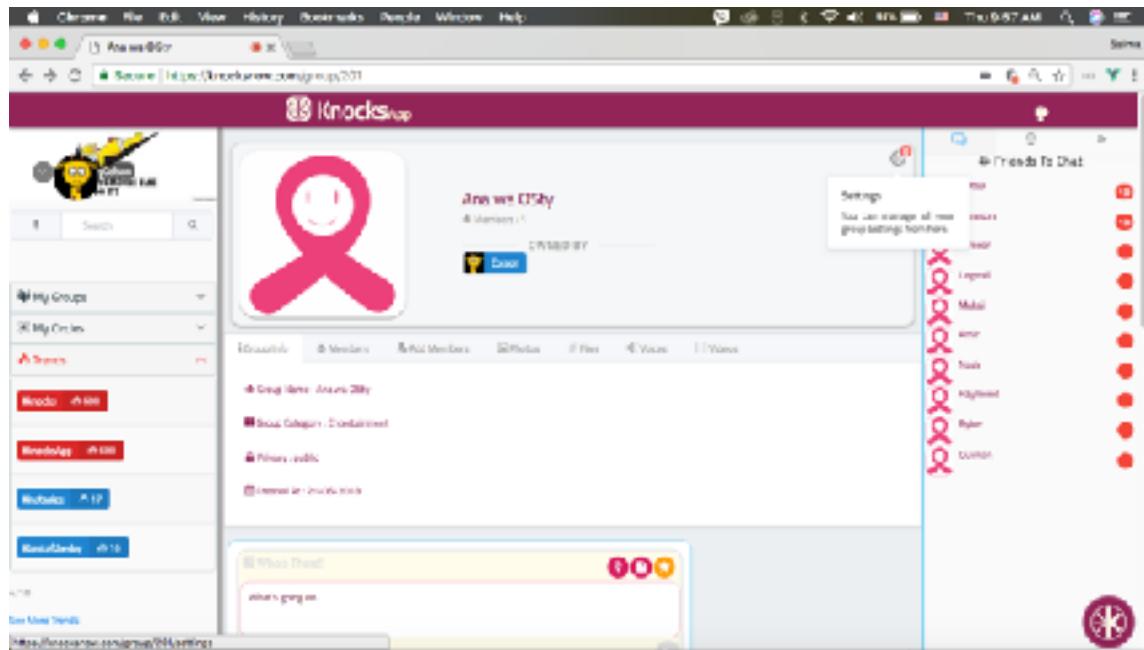


- **Update a group- when allowed to do so.**

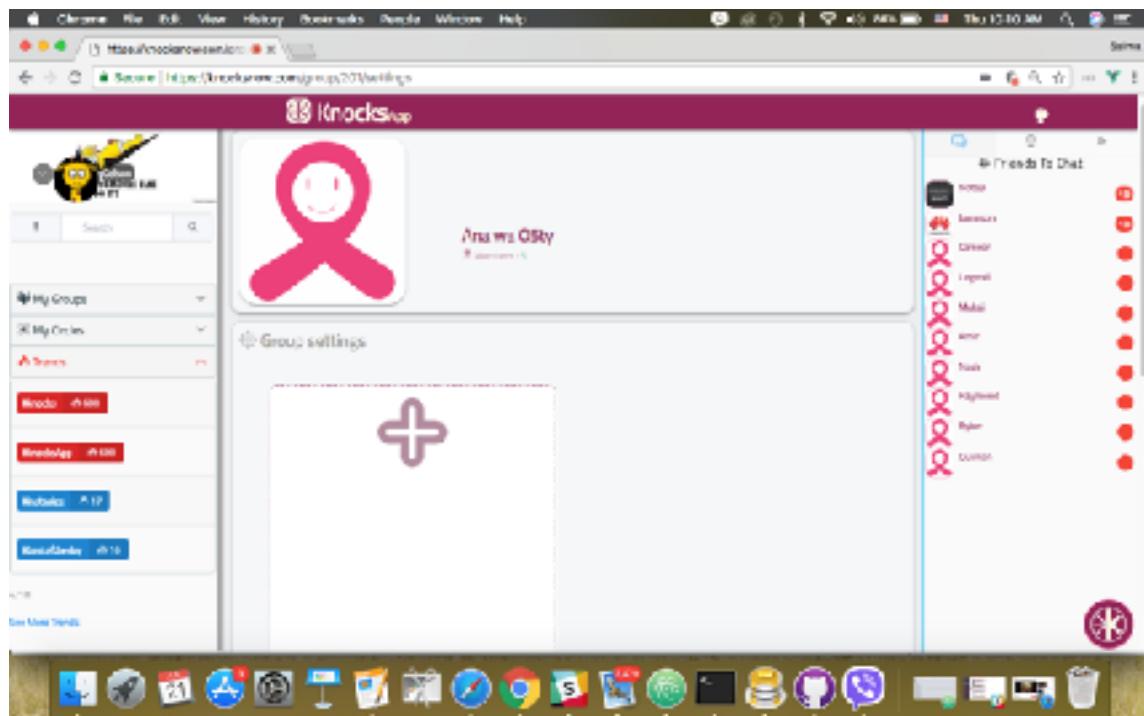
For example updating the group picture, at the left side of the home page in groups section, click on “My Groups” tab and search for a group that he owns or admins.

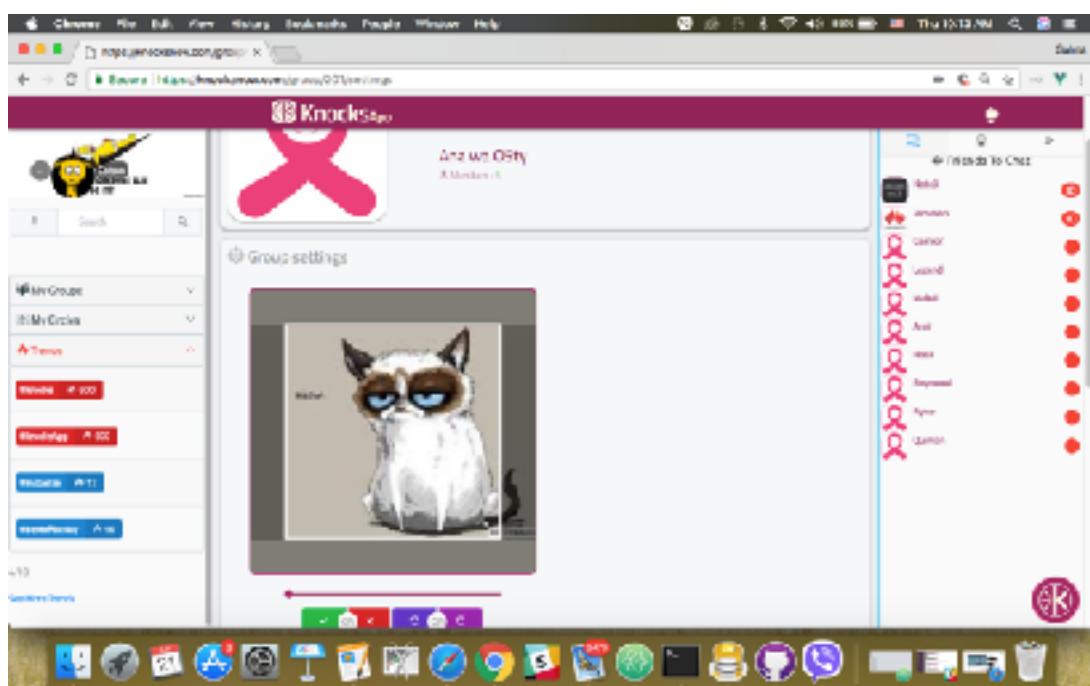
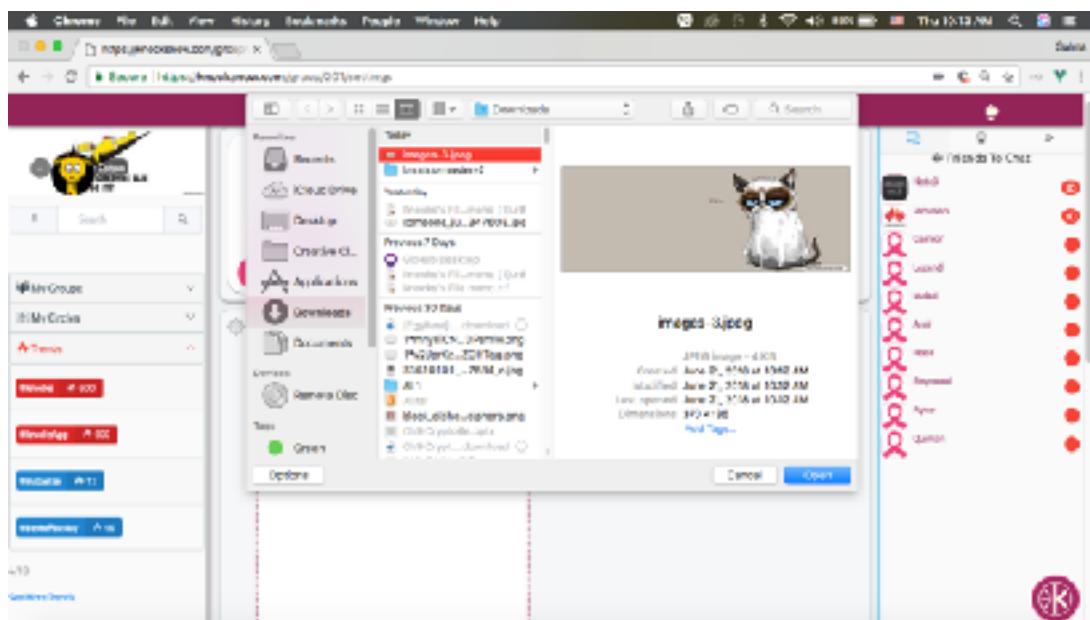


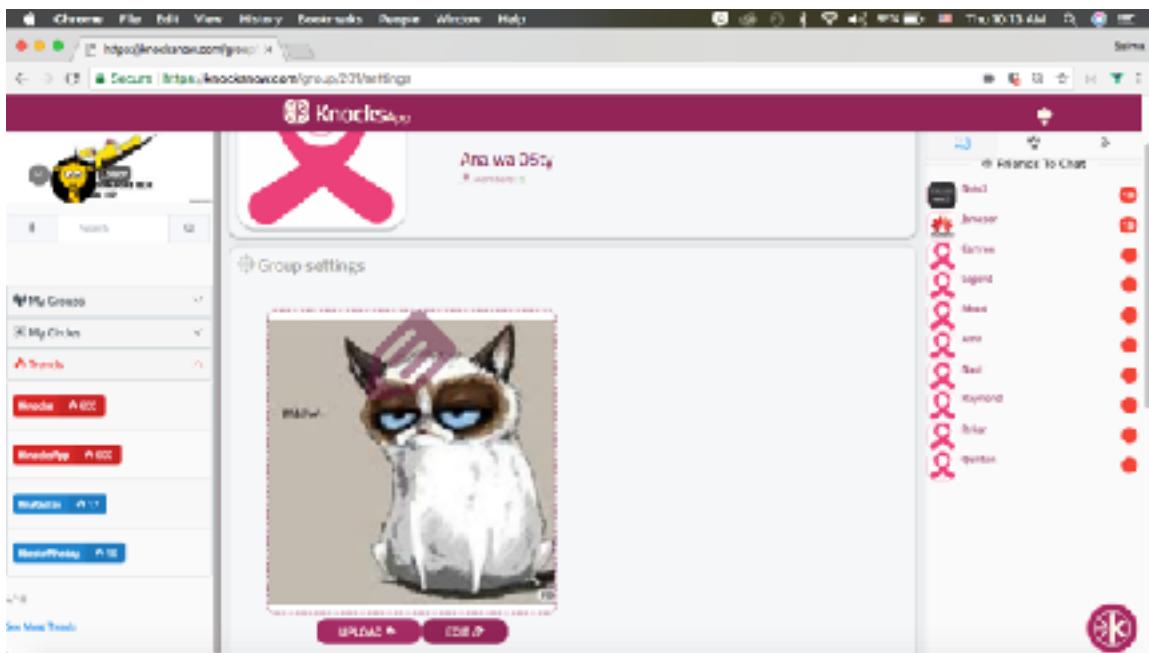
Then click on the group name chip, it will direct the user to the group page, then click on the settings icon.



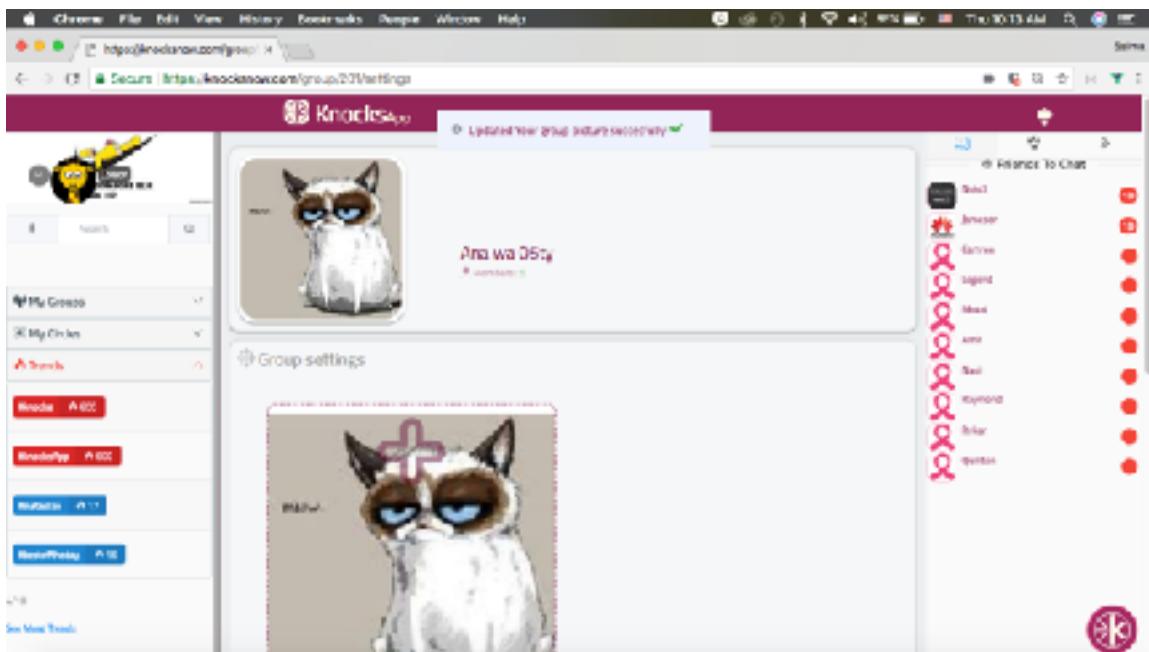
Then click on the change group picture tab and choose the picture to be updated with clicking on the add icon.







Then the System will display message “Updated your group picture successfully”, and changes the group picture into the new one.

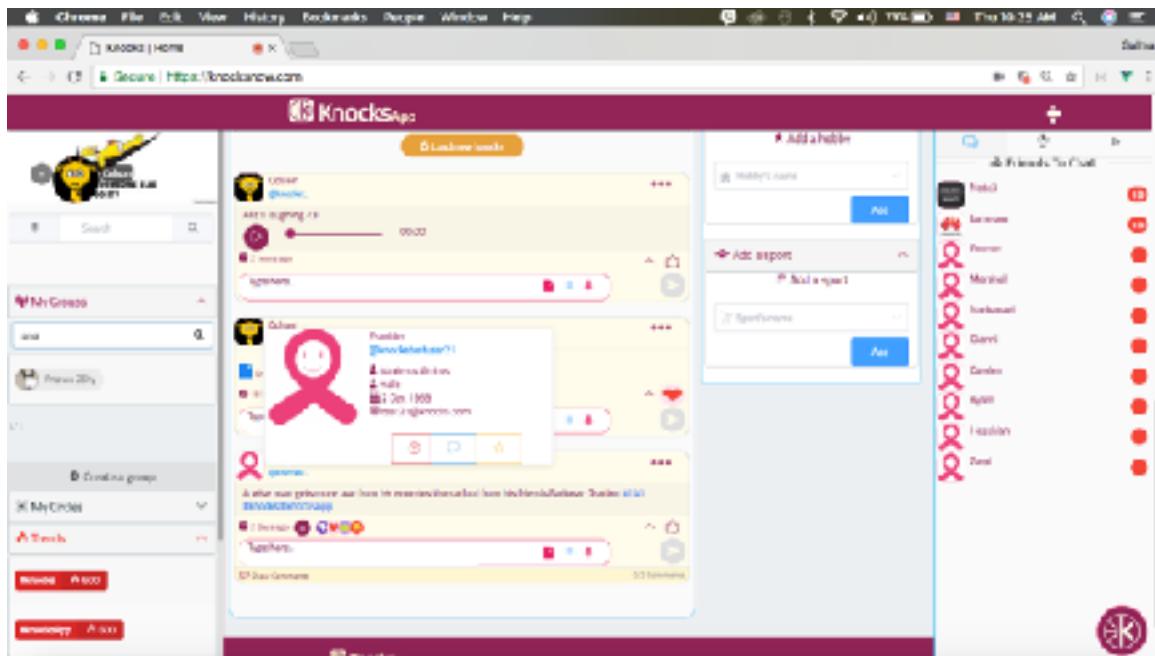


- Delete a group- when allowed to do so.

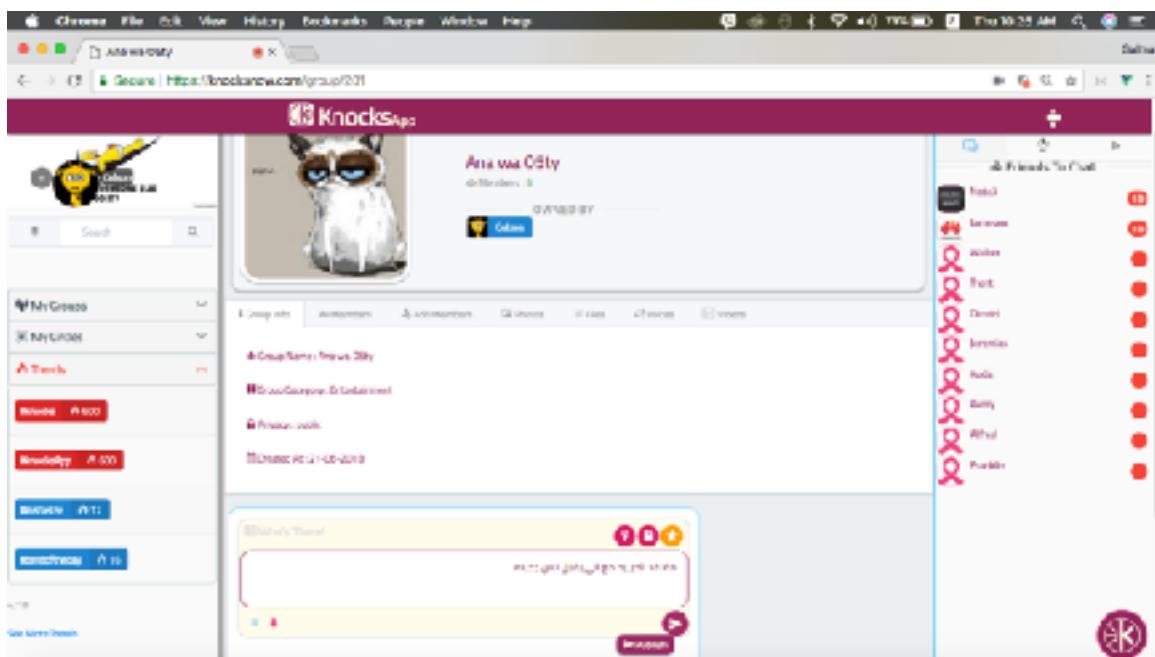
Not yet a ready function

- Post in a group- when allowed to do so.

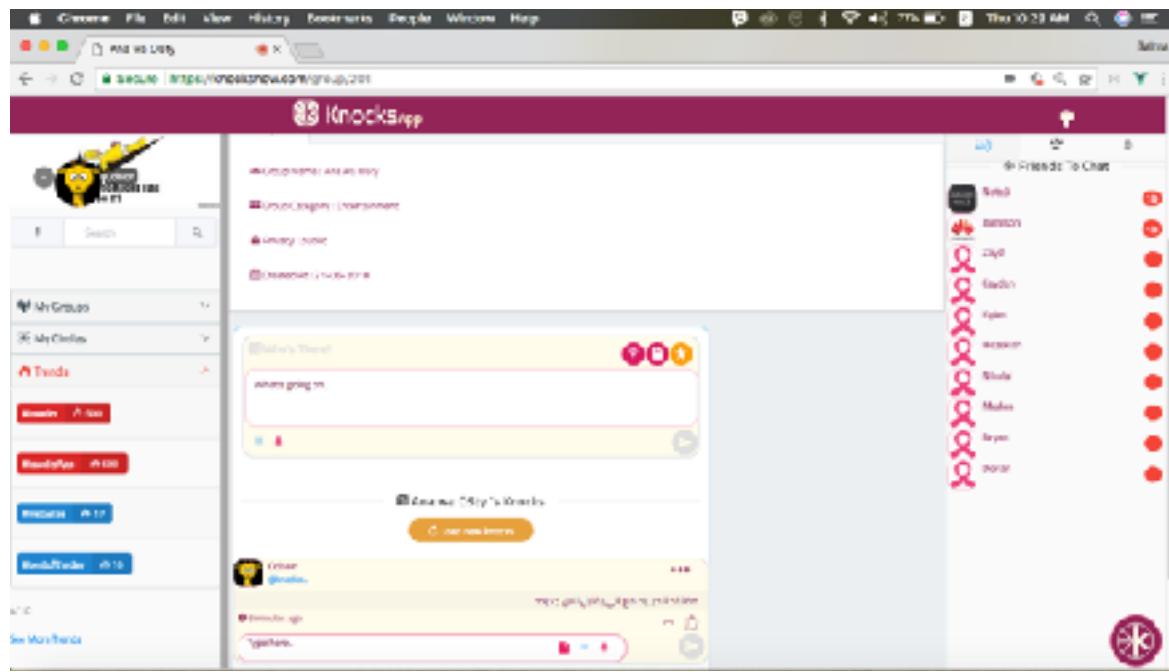
At the left side of the home page in groups section, click on “My Groups” tab and search for a group that he can post at.



Then click on the group name chip, it will direct the user to the group page, then create a post in the group and click “Publish”.

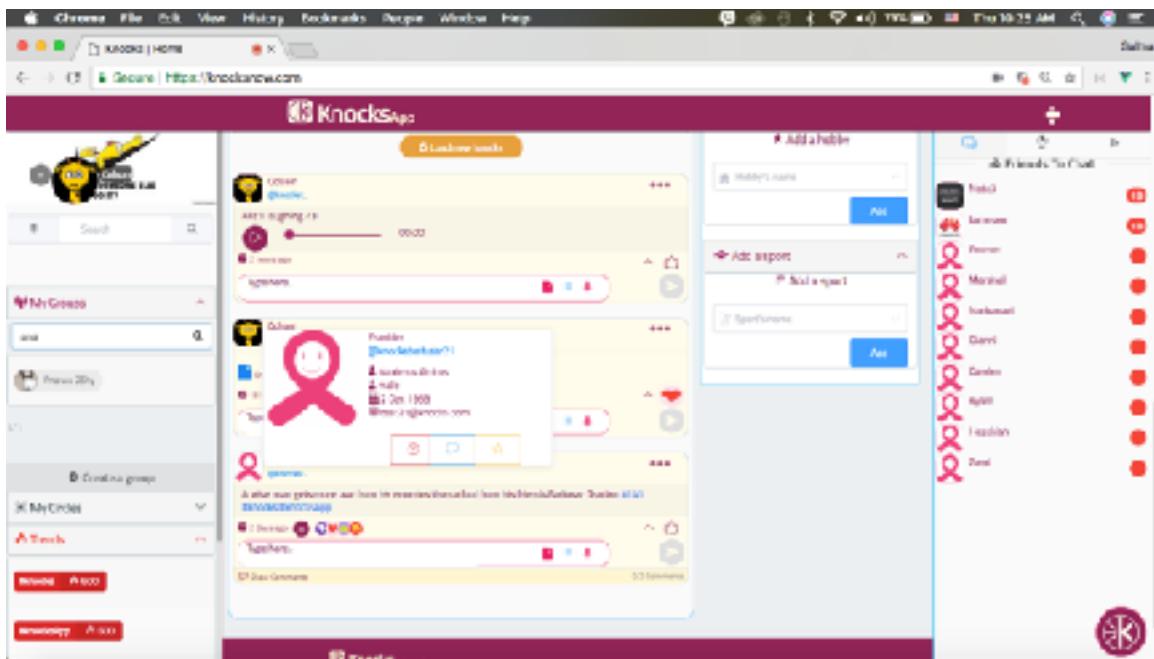


Then the System will display message “success”, and adds your post to the group knocks.

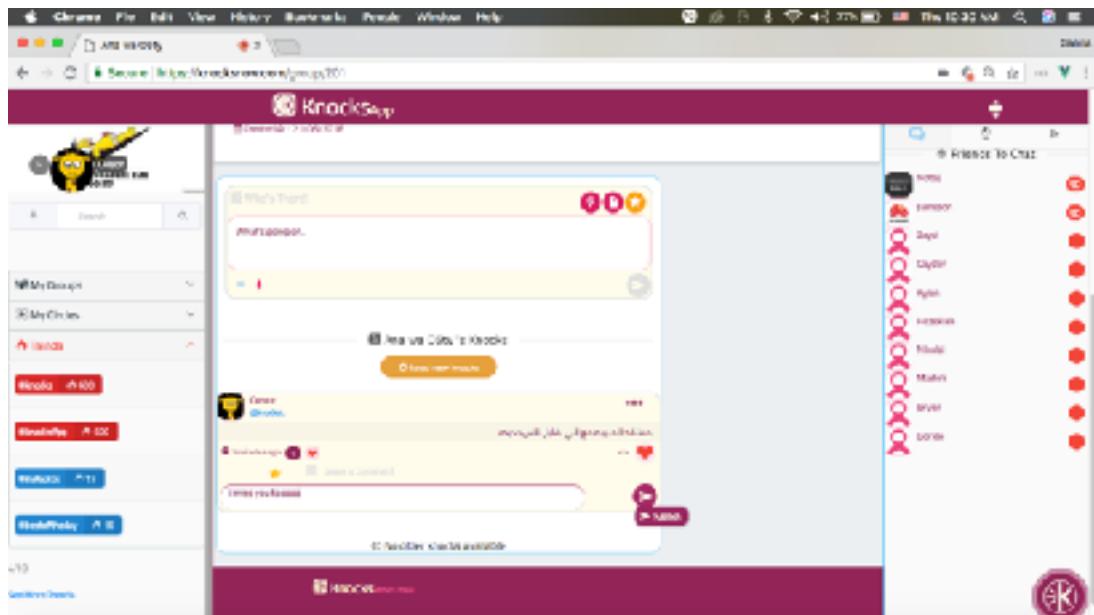


- Comment in a group- when allowed to do so.

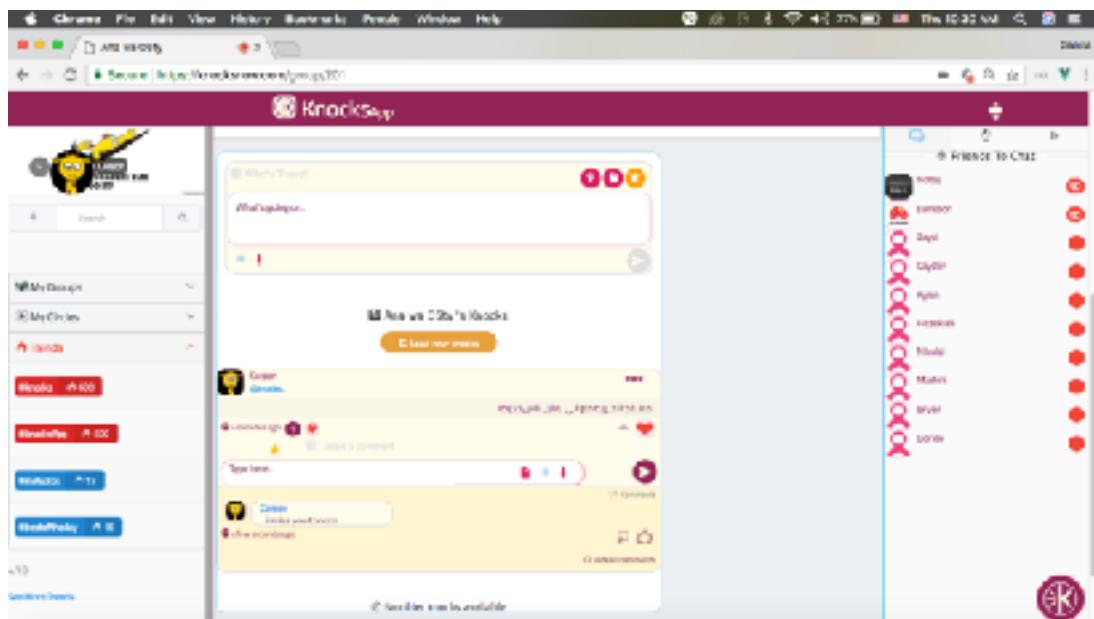
At the left side of the home page in groups section, click on “My Groups” tab and search for a group that he can comment at.



Then click on the group name chip, it will direct the user to the group page, then create a comment for a post in the group and click “Publish”.

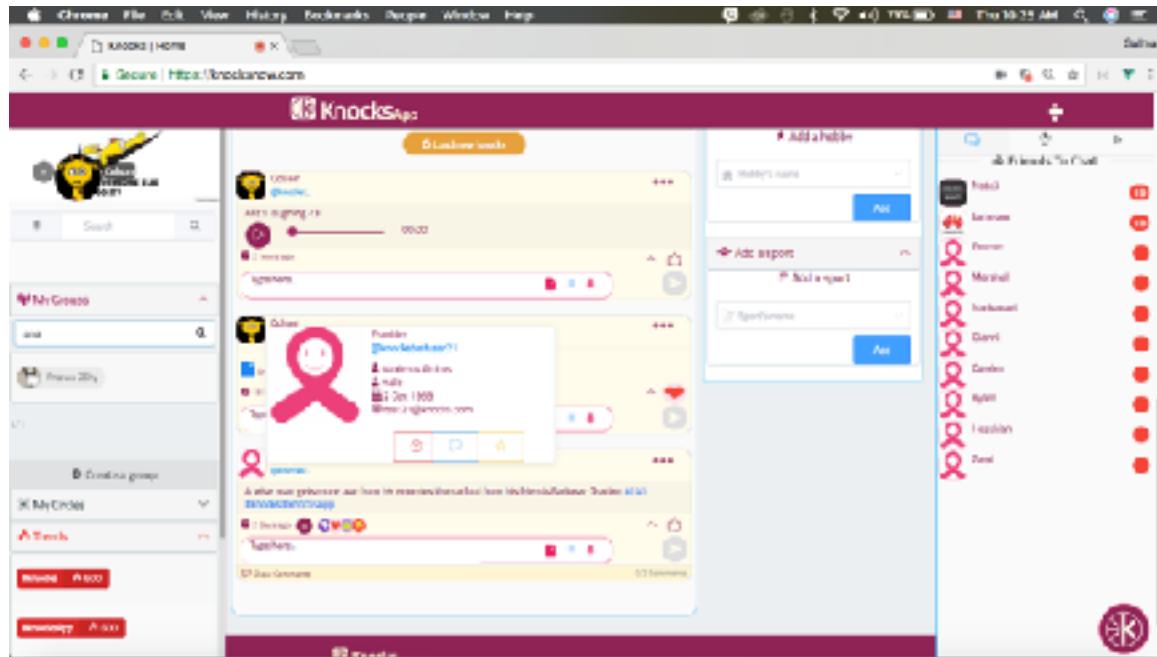


Then the System will display message “success”, and adds your comment to the post comments in the group.

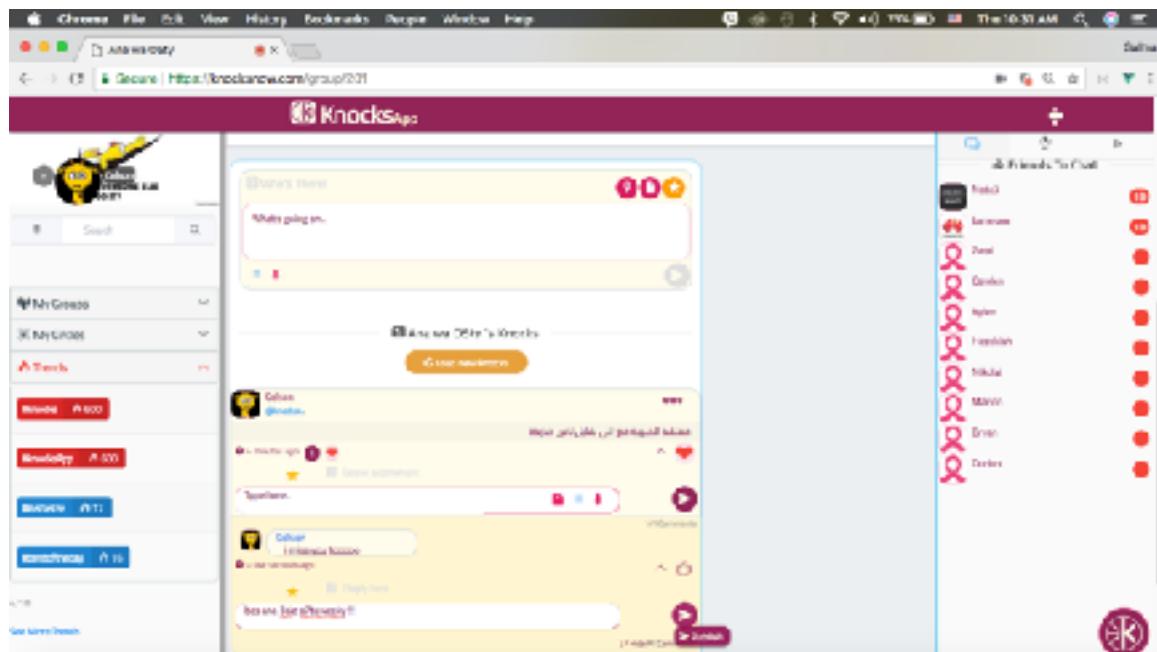


- **Reply in a group- when allowed to do so.**

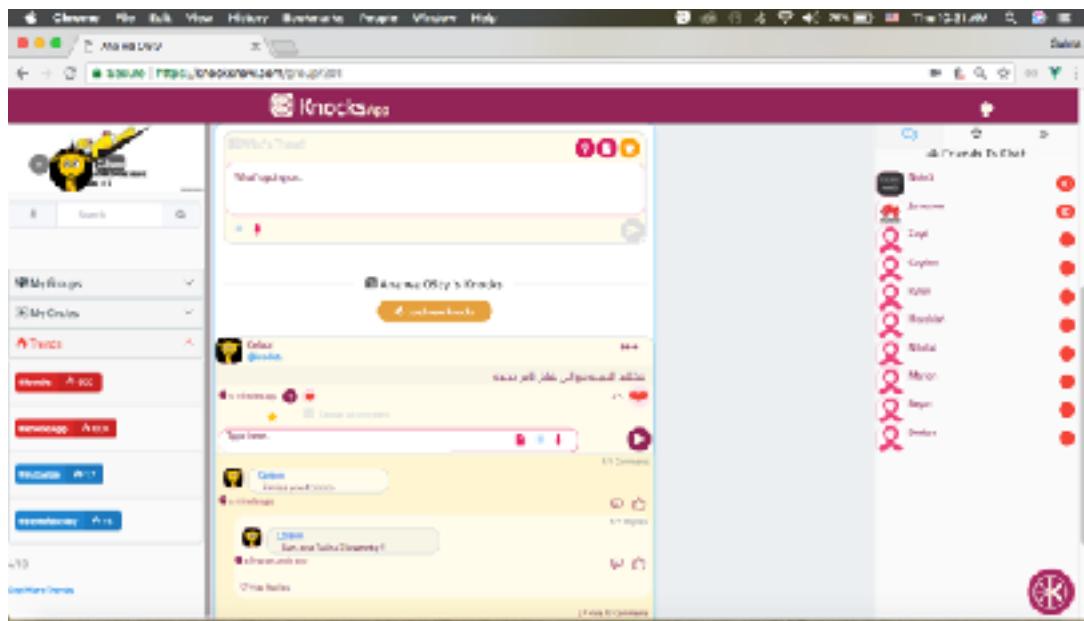
At the left side of the home page in groups section, click on “My Groups” tab and search for a group that he can reply at.



Then click on the group name chip, it will direct the user to the group page, then create a reply for a comment in the group and click “Publish”.

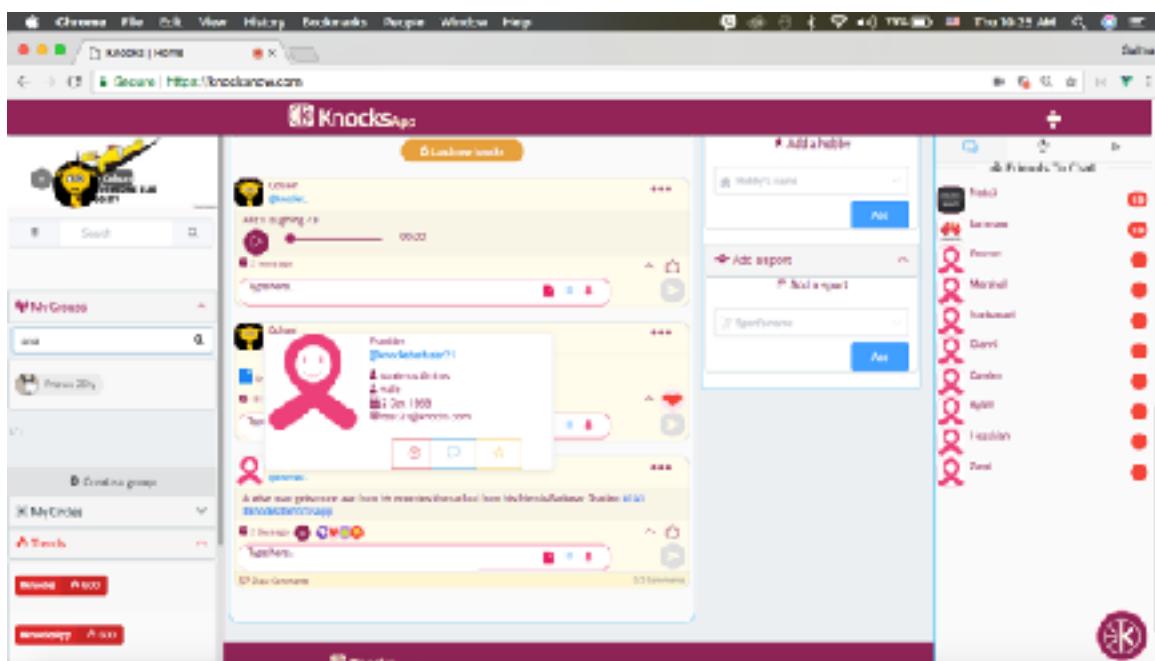


Then the System will display message “success”, and adds your reply to the comment of the post in the group.

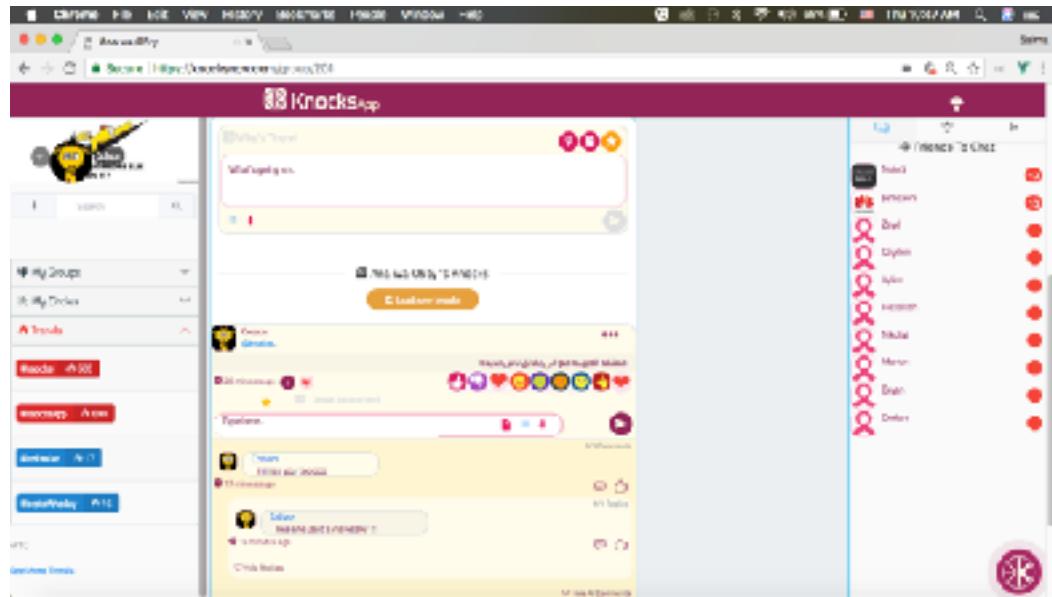


- React to an object a group- when allowed to do so (Eg : Post Reaction).

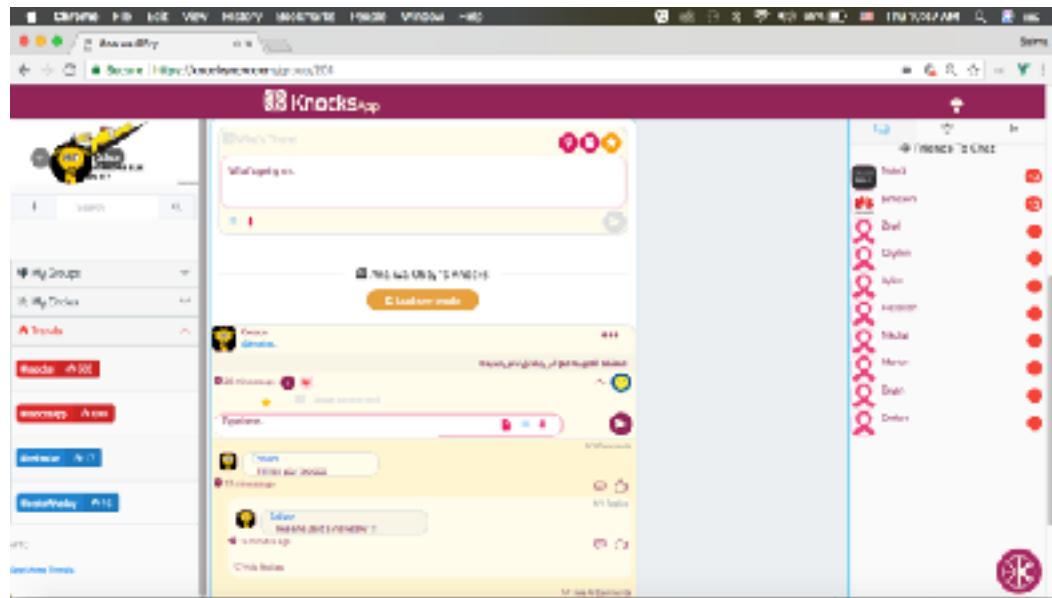
At the left side of the home page in groups section, click on “My Groups” tab and search for a group that he can react.



In any Knock of the group knocks where the user desires to react, at right of the the knock section he should hover or click on the reaction icon, then he clicks on the reaction

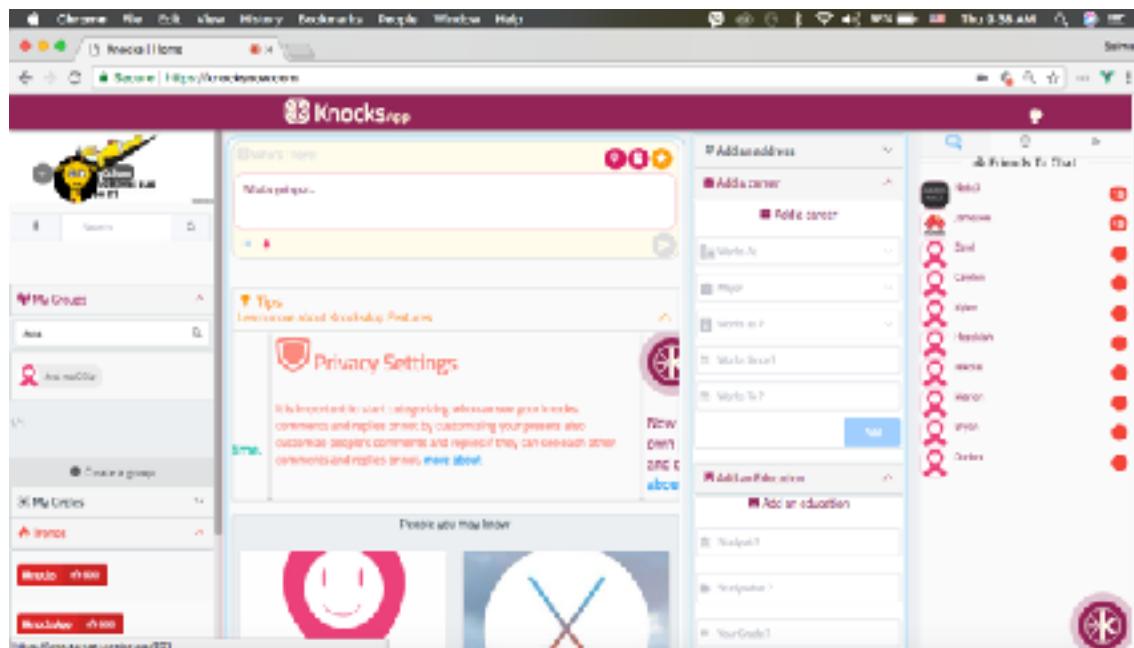


Then the system will display the new reaction towards the post in the group and updates the reactions counter of the post.

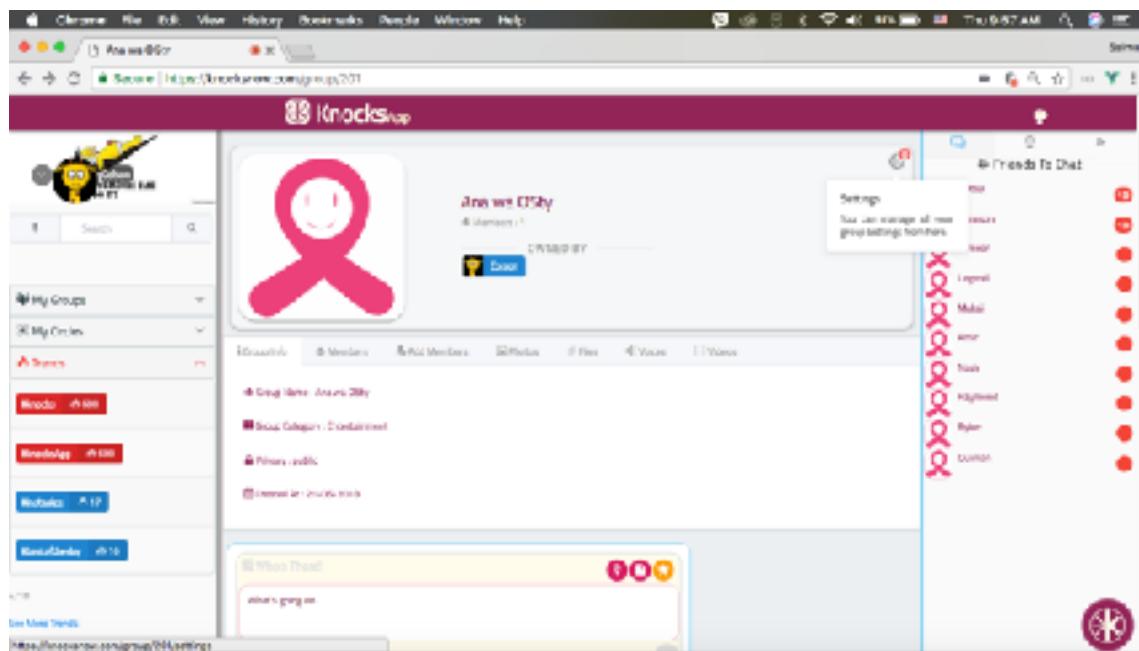


- **Change user permission- when allowed to do so.**

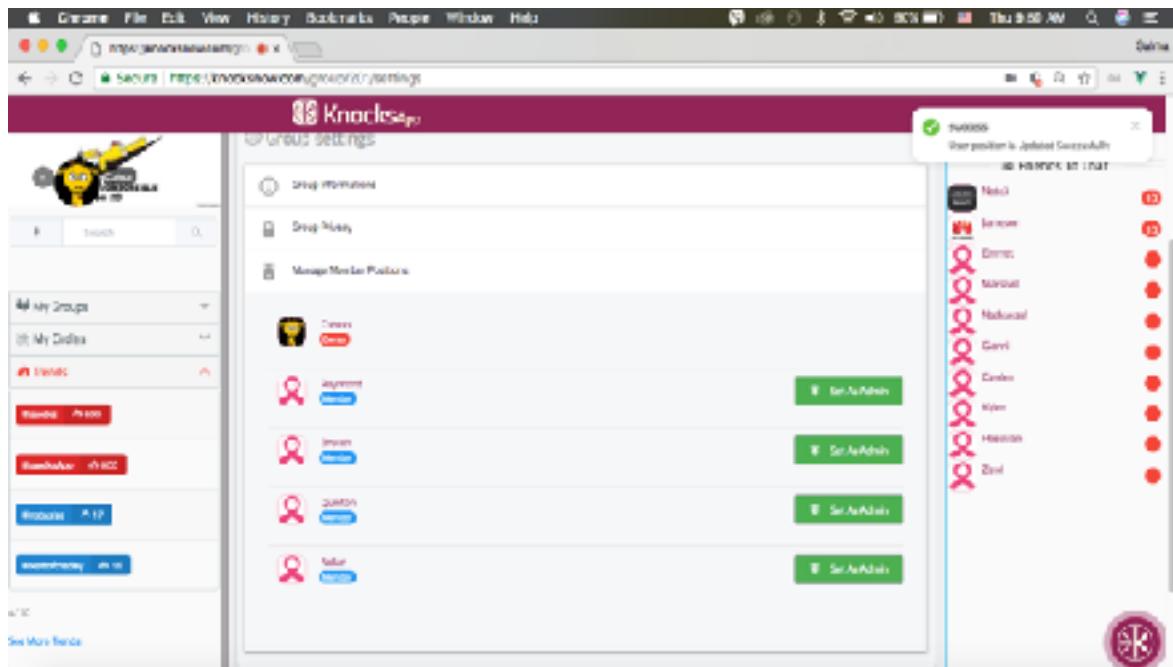
At the left side of the home page in groups section, click on “My Groups” tab and search for a group that he owns or admins.



Then click on the group name chip, it will direct the user to the group page, then click on the settings icon.



Then click on “Manage Member Positions” and click on “Set as Admin” for specific user, then the system will display a message “User Position is updated successfully”.



- Change the group privacy set - when allowed to do so.

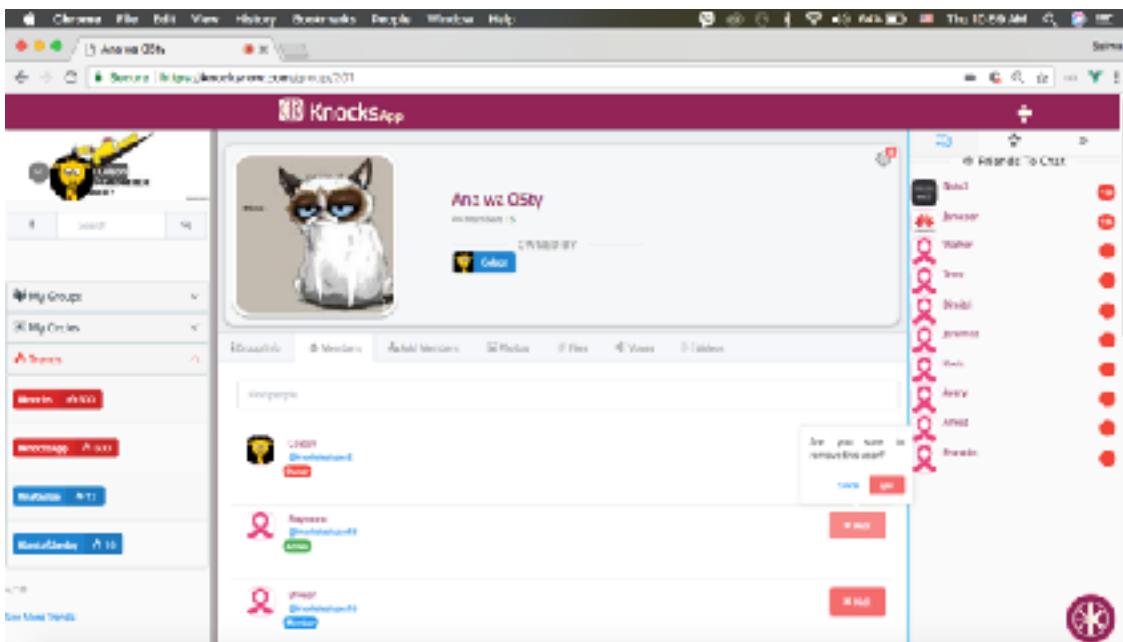
Not yet a ready function

- Ban members of a group- when allowed to do so.**

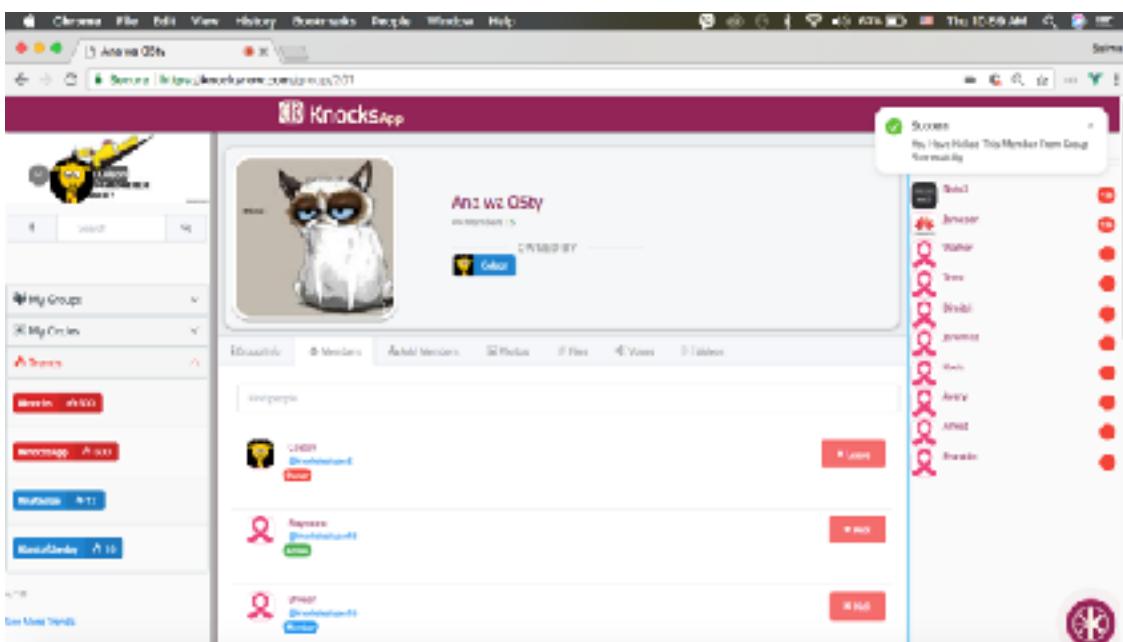
At the left side of the home page in groups section, click on “My Groups” tab and search for a group that he owns or admins.

Then he should click on the “Members” tab, then choose the user to be kicked out of the group (if you are leaving the group that you own, you should transfer the group ownership yo one of the members first), then click “Kick”

A pop over will appear saying “Are you sure you want to remove this user?”, click “Yes”



The system will display a message “You have kicked this member from group successfully”



- Report a group- when allowed to do so.

Not yet a ready function

- Chat in a group

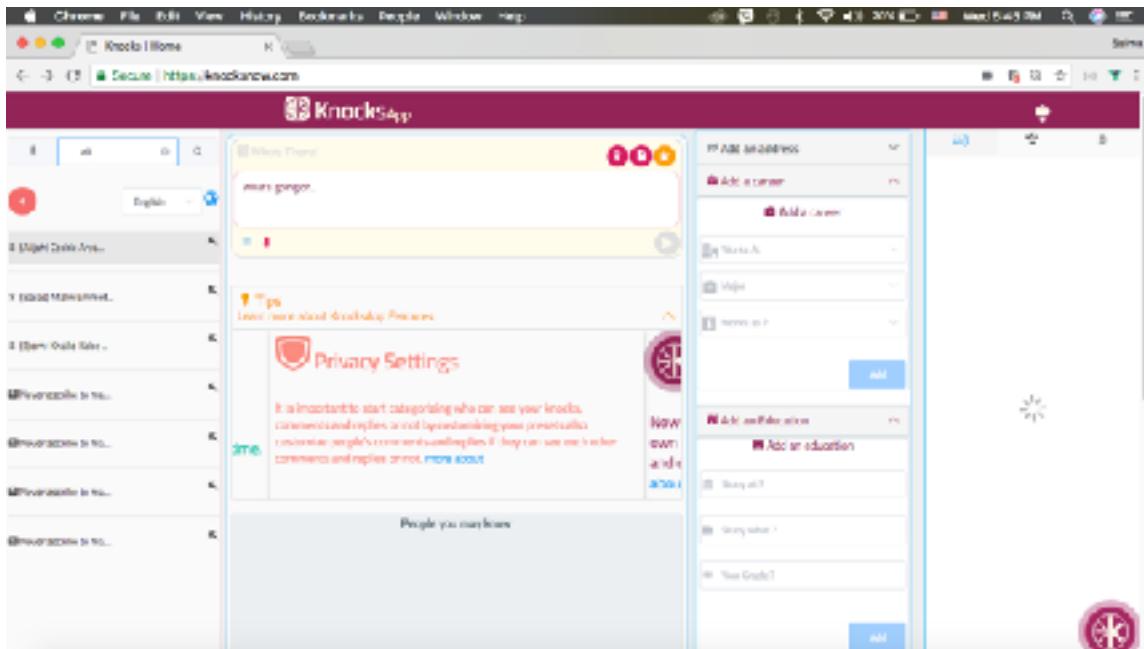
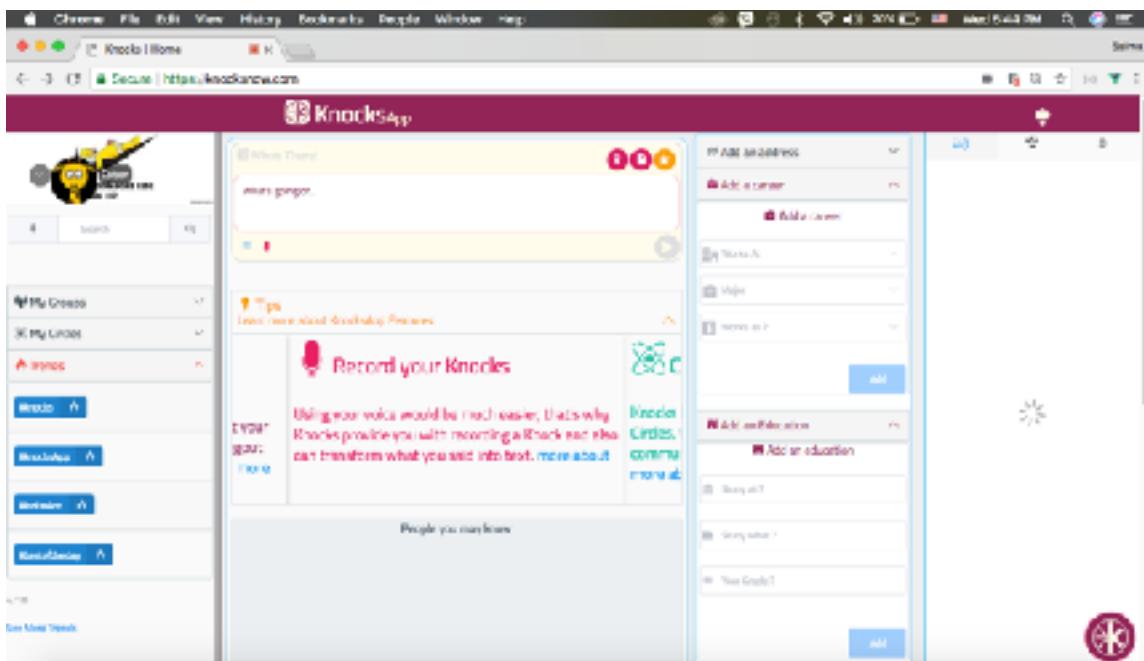
Not yet a ready function

- Request for account verification - incase of public figure.

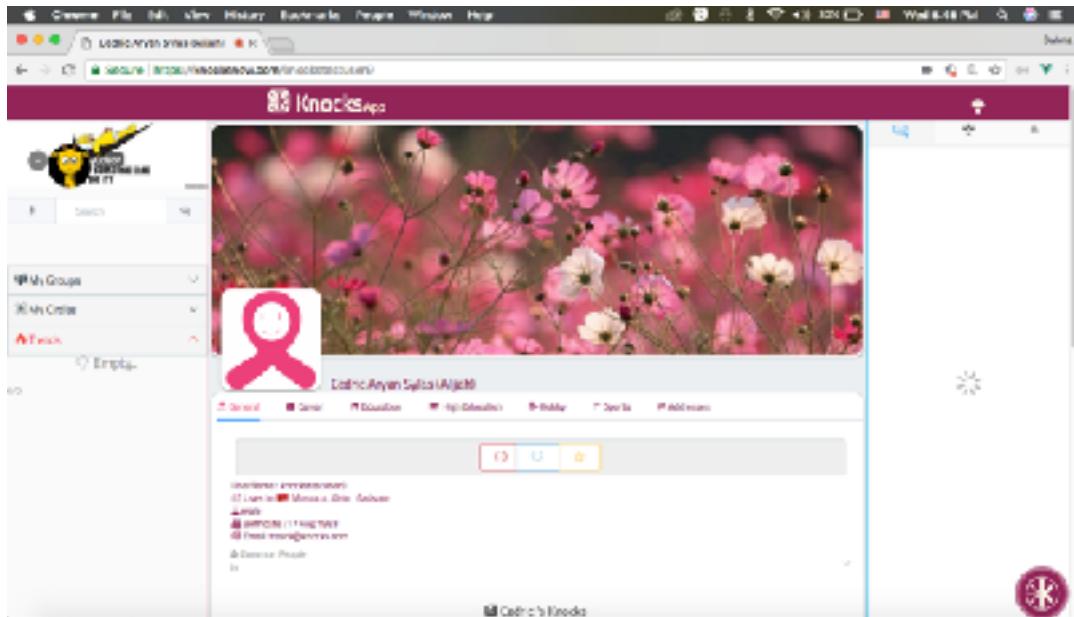
Not yet a ready function

- View a user profile - when allowed to do so.

The user can search for friends in the left side bar and clicks on the user chip to be directed to his profile or just click on his username in a knock, comment, reply or any activity where his username appears.



Then the system will display the user's profile according to the other user's privacy settings.



- Explore talents

Not yet a ready function

- Upload a talent object - when allowed to do so

Not yet a ready function

- Filter talents

Not yet a ready function

- Explore hashtags in talents section

Not yet a ready function

- Rate a talent object

Not yet a ready function

- View a talent career

Not yet a ready function

- Rate a talent owner

Not yet a ready function

- Leave a review on a talent object

Not yet a ready function

- Leave a review on a talent owner

Not yet a ready function

- View Analysis for the talent owner himself

Not yet a ready function

- Monitor for child log.

Not yet a ready function

- Monitor objects for kid

Not yet a ready function

- Set admins for public figure

Not yet a ready function

- Ban for public figure account

Not yet a ready function

- Set a parent for a child

Not yet a ready function