

---

**Future University in Egypt**  
**Faculty of Computers and Information**  
**Technology Department of Computer Science**



# **Graduation Project**

## **KNOCKS Application**

### **Team Work**

Khaled Ashraf Abasiry

Hesham Ahmed Mohamed

Mohamed Amr Mahdy

Salma Roshdy Ahmed

### **Supervised by**

Dr. Mahmoud Sami Othman

TA. Salma Radwan

### **Academic Year**

2017-2018

---

## Acknowledgement

It has been a great opportunity to gain lots of experience in real time project followed by knowledge of how to design and analysis real project for that we want to thank all people who made it possible for students like us special thanks to the graduation project unit for the efforts they did to provide us with all useful information and making the path clear for the students to implement all the education periods in real time project design and analysis.

We would like to express our special thanks of gratitude to **DR. Mahmoud Sami** encouragement and support during project as well as **TA. Salma Rdawan** who supervised our graduation project, and also helped us to make it the way it is now, which also helped us to earn many skills.

We are really thankful to them.

Secondly, we would also like to thank our university **FUE**. Special thanks to **Prof. Ebada Sarhan**.

## الشکر و التقدیر

لقد كانت فرصة عظيمة لنا ان نكتسب الكثير من الخبرات من خلال تنفيذ مشاريع حقيقية تلاها اكتساب معلومات عن طريقة تصميم مشاريع واقعية وتحليلها و لذا ، فاننا نود ان نشكر كافة الشخصيات التي اتاحت مثل هذه الخبرات

للطلاب و نود ان نتوجه بشكر خاص الي وحدة مشاريع التخرج للجهود التي بذلتها ل توفير كافة المعلومات المفيدة و انارة الدرب امام الطلاب حتى يتمكنوا من تنفيذ مشاريع حقيقة و تحليلها اثناء فترة الدراسة. نود ان نتوجه بشكر خاص **للكتور/ محمود سامي** على ما قدمه لنا من تشجيع و دعم خلال تنفيذ المشروع و كذلك نود

ان نشكر **م/ سلمى رضوان** المشرفة على مشروع التخرج الخاص بنا و علي المساعدات التي قدمتها و كانت السبب وراء وصولنا الي هذا اللحظة و اكتساب الخبرات. و نحن ندين لهم جميعا بالشکر و العرفان. و نود ايضا ان نشكر جامعة المستقبل. و شكر خاص **للكتور / عباده سرحان** رئيس جامعة المستقبل.

---

## TABLE OF CONTENTS

Chapter 1: Introduction	7
1.1. Introduction	8
1.2. Problem Definition	8
1.2.1. Introduction	8
1.2.2. Limited Society service	9
1.2.3. Limited Expressivity	9
1.2.4. Parental Monitor Missing	10
1.2.5. User Privacy	10
1.2.6. Unseen talents	11
1.3. Proposed system	11
1.3.1. Overview	11
1.4. Functional Requirements	12
1.5. Non- Functional Requirements	14
1.5.1. Product Requirements	14
1.5.2. Organizational Requirements	15
1.5.3. Additional requirements	15
1.6. Target environment	16
1.1. Client requirements	16
1.6.2. Server requirements	16
1.6.3. Programming languages	17
1.6.4. Design	18
1.6.5. Developer tools	18
Chapter 2 : Background & Survey	20
2.1. Introduction	21
2.2. System Benefits	21
2.3. Survey	22
2.4. System Tools	24

---

---

## TABLE OF CONTENTS

Chapter 3 : System analysis	37
3.1. Scenario	38
3.2. Use case Model	78
3.3. Use case Specification Tables	86
3.4. ERD	116
4. Chapter 4 : System Design	117
4.1. System Architecture	118
4.2. Class Diagram	120
4.3. Physical Schema	123
4.4. Sequence Diagram	124
Chapter 5 : Implementation	166
5.1. Controllers	167
5.1.1. Controllers List	167
5.1.2. Controllers Specifications	169
5.2. Models	193
5.2.1. Models List	193
5.2.2. Models specification	195
5.3. Specific Algorithms Used	212
5.3.1. Searching Algorithm	212
5.3.1.1. Algorithm Flowchart	212
5.3.1.2. Algorithm Description	213
5.3.2. Voice Commands Algorithm	214
5.3.2.1. Algorithm Flowchart	214
5.3.2.2. Algorithm Description	215
5.3.3. Voice to text conversion Algorithm	215
5.3.3.1. Algorithm Flowchart	215
5.3.3.2. Algorithm Description	216

---

---

## TABLE OF CONTENTS

5.4.Vue Components	216
5.4.1.Vue Component List	217
5.4.2.Vue Component specification	221
Chapter 6 : Testing	237
6.1. Testing Strategies	238
6.2. Test Cases	239
Chapter 7 : System Deployment	291
7.1.Tools Setup	292
7.1.1.Network Authentication	292
7.1.2.DNF Packages Manager	293
7.1.3.Knocks Packages	295
7.1.4.Packages Installation	295
7.1.4.1.HTTPD Apache Server	295
7.1.4.2.Maria DB (MySQL)	296
7.1.4.3.PHP	296
7.1.4.4.Composer	297
7.1.4.5.Laravel	298
7.1.4.6.MongoDB	299
7.1.4.7.NPM	300
7.1.4.8.GNOME	300
7.1.4.9.Mozilla Firefox	301
7.1.4.10.phpMyAdmin	301
7.1.4.11.MongoDB Compass	302
7.1.4.12.Cockpit	302
7.1.5.Knocks Installation Script	303
Chapter 8 : User Manual	304
Chapter 9 : Conclusion and Future Work	381

---

---

## TABLE OF CONTENTS

9.1.Achievements	382
9.2.Future Works	382
Chapter 10 : References	383

---

# Chapter 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 Society service**

Well the existing social media applications are fair enough with the entertainment and connections building means as for Facebook and Twitter and others, but it would be much better to help others even slightly like helping disordered and disabled cluster in the society like those who have hand disorders such as Parkinson's disease, Nerve compression syndromes of the hand, Carpal tunnel syndrome, cubital tunnel syndrome, and radial tunnel syndrome.

However, why not helping those who have weakness of view or some other eyes disorders, well if the user cannot use his fingertips all the time or cannot recognize the small shapes of letters for typing or other related problems, most probably they won't be comfortable to use the application if not stopping using it at all.

so, as a solution for those problems is giving them a tool that is easy to be used by them and does not press on their emotions with their disabilities with appreciating and respecting them, of course it will Wonderfull; that's why developing a voice command tool that provide searching, posting, commenting on posts, assisting with converting the spoken speech into text, visiting other users' profiles and guiding users for tips using it, is a helpful and society serving tool that is needed indeed.

### **1.2.3. 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.

#### **1.2.4. 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.

#### **1.2.5. 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.

### 1.2.6. 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 them 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. Include a friend to a circle
8. Exclude a friend from a circle
9. Regulate a privacy set for users
10. Regulate a privacy set for circles
11. Download files (with any extension).
12. Upload profile picture
13. Upload cover picture
14. Update profile picture
15. Update cover picture
16. Block a friend
17. Unfriend a friend
18. Update personal information
19. React to post- when allowed to do so with allowed reactions.
20. React to comment- when allowed to do so with allowed reactions.
21. React to reply- when allowed to do so with allowed reactions.
22. Create a group
23. Add friends to group
24. Update a group- when allowed to do so.
25. Post in a group- when allowed to do so.

26. Comment in a group- when allowed to do so.
27. Reply in a group- when allowed to do so.
28. React to an object a group- when allowed to do so.
29. Change user permission- when allowed to do so.
30. Change the group privacy set - when allowed to do so.
31. Ban members of a group- when allowed to do so.
32. View a user profile - when allowed to do so.
33. Monitor for child log.
34. Set a parent for a child
35. Search for knock
36. Search for group
37. Search for circle
38. Search by voice for knock
39. Search by voice for group
40. Search by voice for circle
41. Enabling and disabling voice command
42. visiting profiles by voice commands
43. Log out using voice commands
44. visiting authenticated user profiles by voice commands
45. Publishing a knock by voice commands
46. Guiding users by voice commands
47. search by voice commands
48. Assisting the voice commands with text
49. Converting the spoken voice into text
50. Representing the various knock types
51. Comment on photos
52. React on photos
53. Generating report for user's activities
54. Notifying the users
55. Enabling Forgot my password mode
56. Request for e-mail verification
57. Blocking accounts ( for security purposes )
58. Share posts

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 it's 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 it's own modern character through the seeding the colors picking choices.
- **Performance** : the application should operate it's 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

### 1.1. Client requirements

- ❖ Minimum Hardware Specifications
  - 1GB RAM.
  - 1.3 Dual Core Processors.
  - 3.5 Inches display.
  - 450px as a display width.
  - 500MB for GBU.
- ❖ Minimum OS version
  - Mac OS X Tiger 10.4
  - Windows XP Professional NT 5.2
  - Linux based kernel 2.6.14
  - Debian Linux based Etch 4.0
  - Fedora | RH Linux based Bordeaux 2.14
  - Arch Linux kernel 2.6.15
  - iOS 7.1.2
  - Android Ice-cream Sandwich 4.0
- ❖ Minimum Navigator
  - Chrome 4.0
  - Internet Explorer 9.0
  - Mozilla Firefox 3.5
  - Safari 4.0
  - Opera 10.5

### 1.6.2. Server requirements

- ❖ Minimum Hardware Specifications
  - 4GB RAM foreach.
  - 2 vCPUs foreach.
  - 80GB SSD Disk foreach.
  - 4TB transfer bandwidth for network capacity foreach.
- ❖ Minimum OS version
  - RHEL Linux 5
  - Fedora Server Linux 25
  - CentOs linux 7
  - Ubuntu Server 16.04.3

- ❖ Minimum Packages Versions
  - Fedora RedHat Server (26).
  - DNF RedHat Dependency Manager (2.6.3-1).
  - DPKG Debian Dependency Manager (1.18.24).
  - HTTPD Apache HTTP Server (2.4.27).
  - Mariadb MySQL Relational Database (10.2).
  - PHP Server Scripting Compiler Zend-Engine (7.1.8).
  - Composer PHP Dependency Manager (1.5.1).
  - Laravel PHP MVC Framework (5.6)
  - NPM Javascript Node Package Manager (5.3.0).
  - Node.JS Server Scripting Compiler Google V8 Engine (8.4.0).
  - VUE.JS Progressive Javascript MVC Framework (2.4.2).
  - Angular.JS Google Javascript Single Page Framework (1.6.4).
  - MongoDB NoSQL Database (3.4).
  - Cockpit Server Resource Manager (149).
  - Gnome Graphical User Interface (3.24).
  - PhpMyAdmin MySQL Database Graphical Manager (4.7.4).
  - Compass MongoDB Database Graphical Manager (1.8.2).

### 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

---

## CHAPTER 1 : INTRODUCTION

- 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

---

# Chapter 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 it's 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, libsolv 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 are 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:

- HTL 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 -cors 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.

---

# Chapter 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"> <li>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</li> <li>If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.</li> </ol>
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"> <li>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</li> <li>If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.</li> </ol>
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"> <li>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</li> <li>If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.</li> </ol>
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
<b>Scenario Name</b>	Create a circle
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and wants to add a new circle to his account
<b>Normal</b>	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.
<b>What can go wrong</b>	1. If the circle is already exists with the same name for the logged in user
<b>Other Activities</b>	The user can proceed all operations allowed while adding new circle
<b>System on Completion</b>	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
<b>Scenario Name</b>	Delete a circle
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and wants to delete a circle to from his account
<b>Normal</b>	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.
<b>What can go wrong</b>	<ol style="list-style-type: none"> <li>If the circle is already does not exists with the same name for the logged in user in the database</li> </ol>
<b>Other Activities</b>	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.
<b>System on Completion</b>	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
<b>Scenario Name</b>	Updatete a circle name
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and wants to update a circle name that he already has
<b>Normal</b>	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 form mysql database with the given user id and circle name.
<b>What can go wrong</b>	1. If the circle does not exist with the same name for the logged in user in the database
<b>Other Activities</b>	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.
<b>System on Completion</b>	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. Include a friend to a circle

Scenario ID	Knocks-7
<b>Scenario Name</b>	Include a friend to a circle
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and has existing circle and wants to add a friend to this circle
<b>Normal</b>	<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>
<b>What can go wrong</b>	<ol style="list-style-type: none"> <li>1. If the circle does not exist with the same name for the logged in user in the database</li> <li>2. If the friend user does not exist in any circle which the account owner has</li> </ol>
<b>Other Activities</b>	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.
<b>System on Completion</b>	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.

### 3.1.8. Exclude a friend from a circle

Scenario ID	Knocks-8
<b>Scenario Name</b>	Exclude a friend from a circle
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and has existing circle and wants to exclude a friend from this circle
<b>Normal</b>	<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>
<b>What can go wrong</b>	<ol style="list-style-type: none"> <li>1. If the circle does not exist with the same name for the logged in user in the database</li> <li>2. If the friend user does not exist in the specified circle which the account owner wants to exclude the friend from.</li> </ol>
<b>Other Activities</b>	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.
<b>System on Completion</b>	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.

### 3.1.9. Regulate a privacy set for users

Scenario ID	Knocks-9
<b>Scenario Name</b>	Regulate a privacy set for users
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and has friends and wants to regulate a primacy set for a specific user
<b>Normal</b>	<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>
<b>What can go wrong</b>	<ol style="list-style-type: none"> <li>1. If the friend does not exist in any circle which the account owner has</li> <li>2. If the privacy set given is not in the specified option in the database</li> </ol>
<b>Other Activities</b>	The user can proceed all operations allowed while regulating a user privacy set, but he cannot unfriend or block this specified friend
<b>System on Completion</b>	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.

### 3.1.10. Regulate a privacy set for circles

Scenario ID	Knocks-10
<b>Scenario Name</b>	Regulate a privacy set for circles
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and has friends and wants to regulate a primacy set for a specific circle
<b>Normal</b>	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.
<b>What can go wrong</b>	<ol style="list-style-type: none"> <li>1. If the circle name does not exist in the circle list which the account owner has</li> <li>2. If the privacy set given is not in the specified option in the database</li> </ol>
<b>Other Activities</b>	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.
<b>System on Completion</b>	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.

### 3.1.11. Download files

Scenario ID	Knocks-11
<b>Scenario Name</b>	Download files
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and he wants to download specified file
<b>Normal</b>	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.
<b>What can go wrong</b>	<ol style="list-style-type: none"> <li>1. The file to be uploaded can't be available to the user according to the object privacy set</li> <li>2. The file is not available at the database anymore</li> </ol>
<b>Other Activities</b>	The user can proceed all operations will downloading file.
<b>System on Completion</b>	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.

### 3.1.12. Upload a profile picture

Scenario ID	Knocks-12
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"> <li>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</li> <li>If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.</li> </ol>
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.

### 3.1.13. Upload a cover picture

Scenario ID	Knocks-13
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"> <li>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</li> <li>If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.</li> </ol>
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.

### 3.1.14. Update a profile picture

Scenario ID	Knocks-14
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"> <li>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</li> <li>If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.</li> </ol>
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.

### 3.1.15. Update a cover picture

Scenario ID	Knocks-15
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"> <li>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</li> <li>If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.</li> </ol>
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.

### 3.1.16. Unfriend a friend

Scenario ID	Knocks-16
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.

### 3.1.17. Blocking a friend

Scenario ID	Knocks-17
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.

### 3.1.18. Update personal information

Scenario ID	Knocks-18
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.

### 3.1.19. React to post

Scenario ID	Knocks-10
<b>Scenario Name</b>	React to a post
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and he wants to react to a specific post
<b>Normal</b>	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.
<b>What can go wrong</b>	1. If the reaction receiver user is not validated so that he cannot receive the chosen reaction type
<b>Other Activities</b>	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.
<b>System on Completion</b>	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.

### 3.1.20. React to comment

Scenario ID	Knocks-20
<b>Scenario Name</b>	React to a Comment
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and he wants to react to a specific comment
<b>Normal</b>	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.
<b>What can go wrong</b>	1. If the reaction receiver user is not validated so that he cannot receive the chosen reaction type
<b>Other Activities</b>	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.
<b>System on Completion</b>	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.

### 3.1.21. React to reply

Scenario ID	Knocks-21
<b>Scenario Name</b>	React to a reply
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and he wants to react to a specific reply
<b>Normal</b>	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.
<b>What can go wrong</b>	1. If the reaction receiver user is not validated so that he cannot receive the chosen reaction type
<b>Other Activities</b>	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.
<b>System on Completion</b>	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.

### 3.1.22.Create a group

Scenario ID	Knocks-22
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"> <li>If the user have submitted the group failed with empty submission of one of the required fields</li> </ol>
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.

### 3.1.23.Add friends to a group

Scenario ID	Knocks-23
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"> <li>If the group is already deleted by other party so that no members can be added to that group</li> <li>If the user is not having the privilege to add member to that group according to his position in the group.</li> </ol>
Other Activities	The user can proceed all operations allowed while adding friends to the group except for deleting the group itself or updating it is 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.

### 3.1.24.Update a group

Scenario ID	Knocks-24
<b>Scenario Name</b>	Update a group
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and he to update a group that he is a member of.
<b>Normal</b>	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.
<b>What can go wrong</b>	<ol style="list-style-type: none"> <li>1. If the group is already deleted by other party so that no updating can be processed on the group</li> <li>2. If the user is not having the privilege to update the group according to his position in the group.</li> </ol>
<b>Other Activities</b>	The user can proceed all operations allowed while updating group except for deleting the group itself or changing the users privileges.
<b>System on Completion</b>	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.

### 3.1.25.Post in a group

Scenario ID	Knocks-25
Scenario Name	Post in a group
Actors	User
Initial Assumption	The user is logged in and he wants to post in group that 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"> <li>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</li> <li>If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.</li> <li>If the user is not a member of the group where he wants to post</li> </ol>
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.

### 3.1.26.Comment in a group

Scenario ID	Knocks-26
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"> <li>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</li> <li>If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.</li> <li>If the user is not a member of the group where he wants to comment</li> </ol>
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.

### 3.1.27.Reply in a group

Scenario ID	Knocks-27
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"> <li>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</li> <li>If the CSRF of the user that has been generated does not match the one received by the PHP Laravel Side.</li> <li>If the user is not a member of the group where he wants to reply</li> </ol>
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.

### 3.1.28.React to an object in a group.

Scenario ID	Knocks-28
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"> <li>1. If the selected object is not in the group anymore or it has been deleted while this process is in progress.</li> <li>2. If the user privilege does not contain the reaction privilege on an object in the group.</li> <li>3. If the reaction type is not compatible with the age of the objects owner.</li> <li>4. If the group has been banning the member who reacts at the moment</li> <li>5. If the group has been deleted.</li> <li>6. If the member has been blocked from the group</li> </ol>
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.

### 3.1.29.Change a group privacy preset

Scenario ID	Knocks-29
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"> <li>1. If the user privilege does not contain the preset changing privilege in the group.</li> <li>2. If the group has been banning the member who reacts at the moment</li> <li>3. If the group has been deleted.</li> <li>4. If the member has been blocked from the group</li> </ol>
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.

### 3.1.30.View user information

Scenario ID	Knocks-30
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"> <li>1. If the user has reported this requester user before.</li> <li>2. If the user has blocked the requester.</li> <li>3. If the privilege of the user1 doesn't satisfy the information viewing for user2.</li> </ol>
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.

### 3.1.31. Set a parent for a Child

Scenario ID	Knocks-31
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"> <li>1. If the user is not logged in then he won't be able to add the parent.</li> <li>2. If the parent has not confirmed the request from the child.</li> <li>3. 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.</li> </ol>
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.

### 3.1.32.Monitor child log

Scenario ID	Knocks-32
<b>Scenario Name</b>	Monitor an Object for a Kid
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and he is a child and there is a content that he will receive from another user
<b>Normal</b>	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
<b>What can go wrong</b>	1. If the user is not logged in then he won't be able monitor an object.
<b>Other Activities</b>	The user can proceed all operations allowed while monitoring the object.
<b>System on Completion</b>	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.

### 3.1.33.Comment on multimedia object (photo )

Scenario ID	Knocks-33
Scenario Name	Comment on multimedia object ( video or photo )
Actors	User
Initial Assumption	The user is logged in and he is wants to comment on multimedia object ( video or photo )
Normal	<p>The user will be browsing and after selecting the photo or video which he wants to comment on, he would write his comment and publish it.</p> <p>The system will be triggering the event of commenting on multimedia object with the user id, knock type, knock id, and comment content from the controller to the model which will check and validate for the user privilege and the availability of the post, if it returns success then the comment will be inserted into the database.</p>
What can go wrong	<ol style="list-style-type: none"> <li>1. If the user has no privilege to comment on this object</li> <li>2. If the post object is not available anymore</li> </ol>
Other Activities	The user can proceed all operations allowed while commenting on object.
System on Completion	A success message that the comment had been inserted successfully, or a failure message that the conditions of comment cannot be satisfied to accomplish this function.

### 3.1.34.Reply on multimedia object (photo )

Scenario ID	Knocks-34
<b>Scenario Name</b>	Reply on multimedia object ( video or photo )
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and he is wants to reply on multimedia object ( video or photo )
<b>Normal</b>	<p>The user will be browsing and after selecting the photo or video which he wants to reply on, he would write his reply and publish it.</p> <p>The system will be triggering the event of replying on multimedia object with the user id, knock type, comment id, and reply content from the controller to the model which will check and validate for the user privilege and the availability of the comment, if it returns success then the reply will be inserted into the database.</p>
<b>What can go wrong</b>	<ol style="list-style-type: none"> <li>1. If the user has no privilege to comment on this object</li> <li>2. If the comment object is not available anymore</li> </ol>
<b>Other Activities</b>	The user can proceed all operations allowed while replying on object.
<b>System on Completion</b>	A success message that the reply had been inserted successfully, or a failure message that the conditions of reply cannot be satisfied to accomplish this function.

**3.1.35. Generating report for user's activities**

Scenario ID	Knocks-35
<b>Scenario Name</b>	Generating report for user's activities
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in and he is wants check on his logs
<b>Normal</b>	The user will enter his settings and get into his devices section The system will be triggering the event of generating the user logs report and it will retrieve his logs from the database which contains his devices, logins, last log in, first log in and operating system.
<b>What can go wrong</b>	<ol style="list-style-type: none"><li>1. If the user has no privilege to comment on this object</li><li>2. If the comment object is not available anymore</li></ol>
<b>Other Activities</b>	The user can proceed all operations allowed while retrieving his logs.
<b>System on Completion</b>	A success message that the report is generated successfully, or a failure message that the conditions of user logs report cannot be satisfied to accomplish this function.

### 3.1.36.Notifying the user ( Eg : Reaction )

Scenario ID	Knocks-36
<b>Scenario Name</b>	Notifying the user
<b>Actors</b>	User
<b>Initial Assumption</b>	The user is logged in.
<b>Normal</b>	The balloon sender will create the balloon by reaction with the sender id, receiver id, balloon content, popped and seen ( pooped and seen are initialized with false in sending ), so whenever the receiver logs in and be available the receiver will receive the balloon and the popped parameter will be set to true and whenever the user responds to the balloon the seen is set to true.
<b>What can go wrong</b>	<ol style="list-style-type: none"> <li>1. If the sender is blocked by the receiver</li> <li>2. If the reaction is not appropriate to the receiver ( candy session )</li> </ol>
<b>Other Activities</b>	The user can proceed all operations allowed while creating and receiving the knock.
<b>System on Completion</b>	A success message that the balloon is created successfully, or a failure message that the conditions of balloon creation cannot be satisfied to accomplish this function.

### 3.1.37.Enable forgot my password mode

Scenario ID	Knocks-37
Scenario Name	Enable forgot my password mode
Actors	User
Initial Assumption	The user cannot log in and forgot his password.
Normal	The user clicks on “Forgot my password” button and then he will receive e-mail after he provides his mail, and this mail will have a temporary password to allow the user to log in again
What can go wrong	<ol style="list-style-type: none"><li>1. If the user has already blocked account</li><li>2. If the user did not provide correct e-mail</li></ol>
Other Activities	The user cannot log in or do any function unless he logs in after receiving the e-mail which has the temporary password
System on Completion	A success message that the password is reset successfully, or a failure message that the conditions of resetting password cannot be satisfied to accomplish this function.

**3.1.38.Request for e-mail verification**

Scenario ID	Knocks-38
<b>Scenario Name</b>	Request for e-mail verification
<b>Actors</b>	User
<b>Initial Assumption</b>	The site is available and the user has just registered.
<b>Normal</b>	when the user finishes the registration, there will be a dialog asking him to verify his account via e-mail, so the user will click on “Send me an e-mail” button Then the system will generate a code for him and if this code matches the one which the user have, then the system will direct the user to his homepage
<b>What can go wrong</b>	1. If the user did not provide correct e-mail
<b>Other Activities</b>	The user cannot log I or do any function unless he verifies his e-mail
<b>System on Completion</b>	A success message that the e-mail verification successfully, or a failure message that the conditions of e-mail verification cannot be satisfied to accomplish this function.

### 3.1.39.Blocking accounts

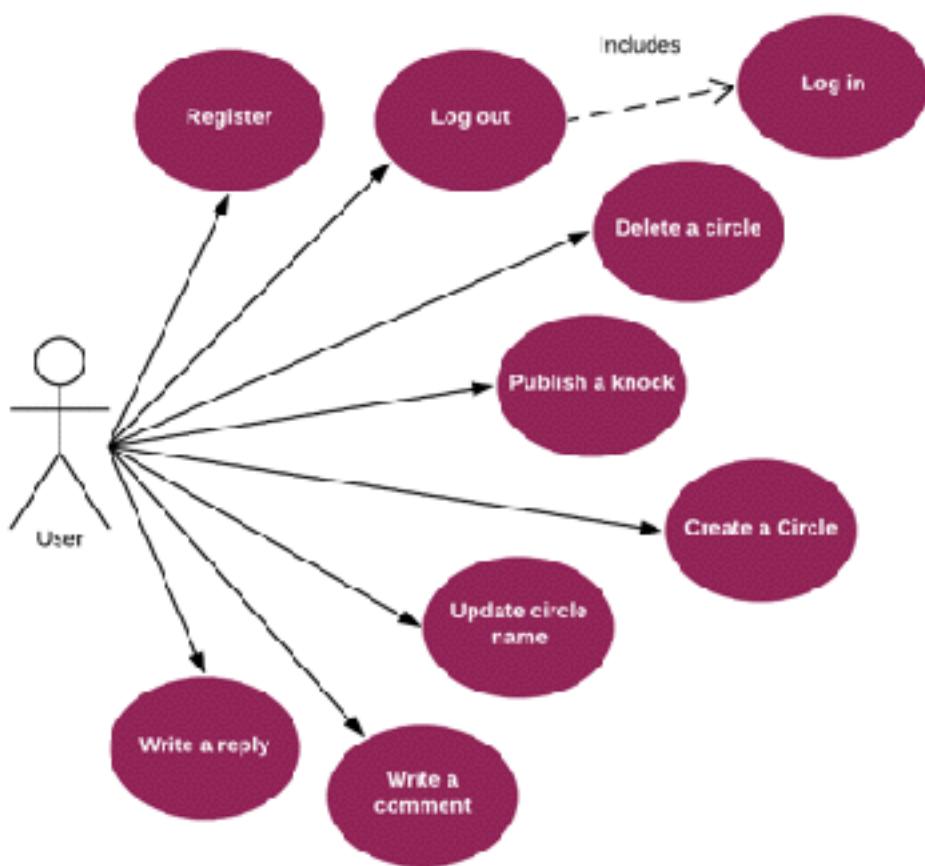
Scenario ID	Knocks-39
<b>Scenario Name</b>	Blocking accounts
<b>Actors</b>	User
<b>Initial Assumption</b>	The site is available and the user already has account but his log in attempts are three or more with invalid data.
<b>Normal</b>	<p>The user will be receiving an e-mail after the system has blocked his system when he attempted three or more log in attempts with invalid mail, phone number, e-mail, or password; so the user will get the e-mail which contains a dialog for account blocking informing, so the user will click “Send me an e-mail” button</p> <p>If the user had provided wrong e-mail and did not verify his e-mail, then the system will keep on blocking his account until the account is verified via e-mail, else the system directs the user to the log in page.</p>
<b>What can go wrong</b>	1. If the user did not provide correct e-mail
<b>Other Activities</b>	The user cannot log I or do any function unless he verifies his e-mail
<b>System on Completion</b>	A success message that the account unblocking has been done successfully, or a failure message that the conditions of account unblocking cannot be satisfied to accomplish this function.

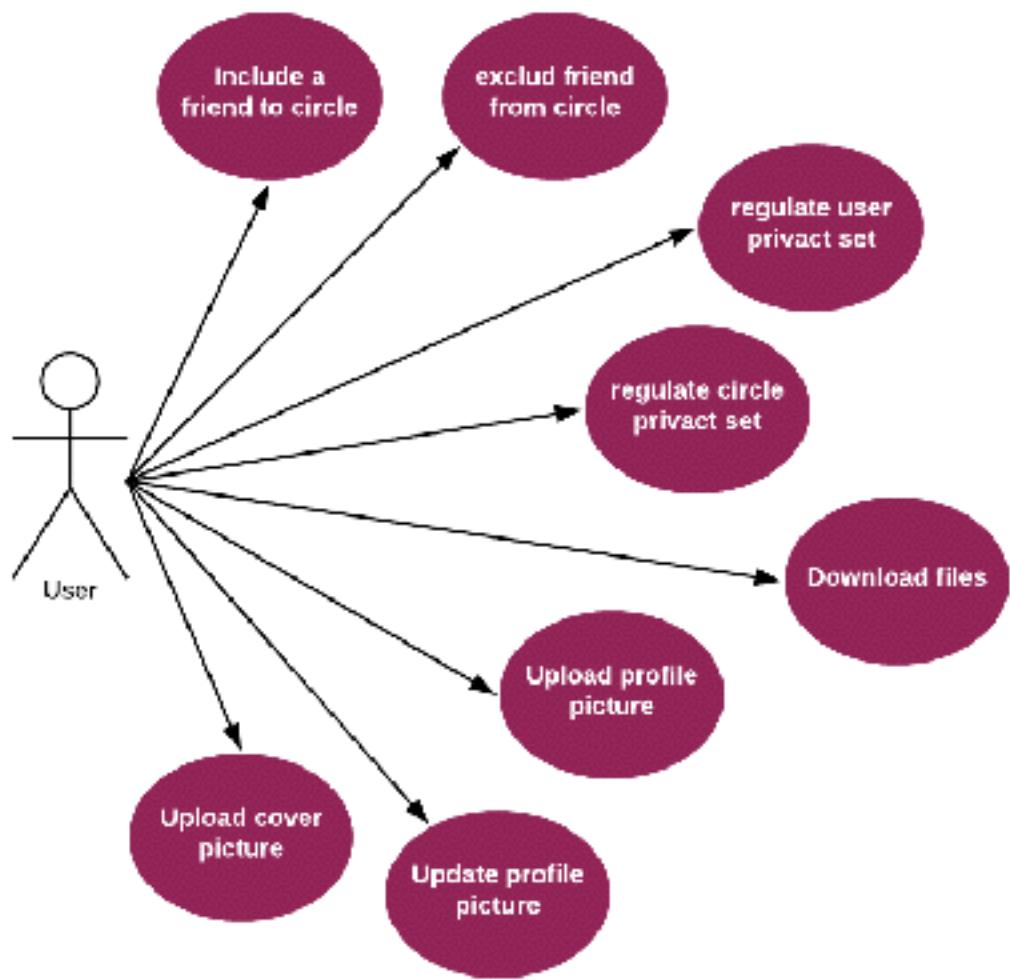
### 3.1.40.Sharing post

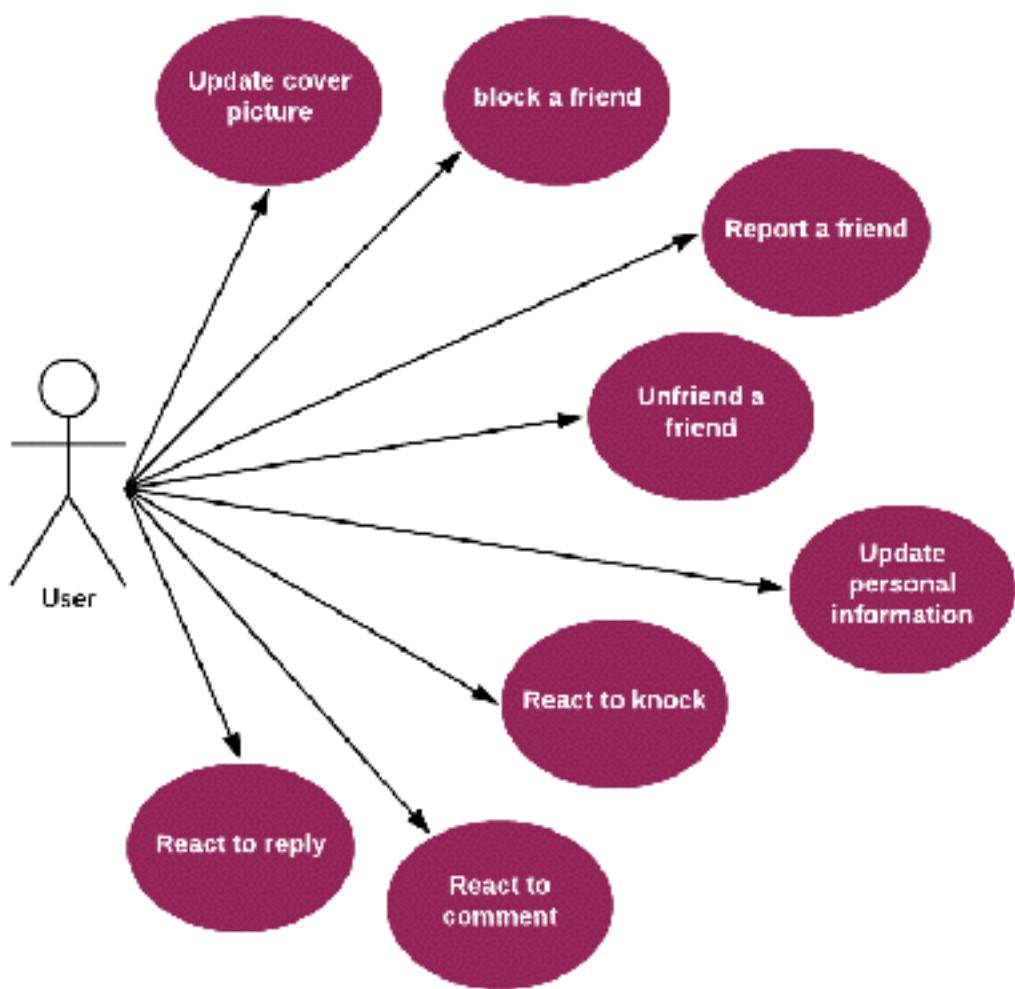
Scenario ID	Knocks-40
<b>Scenario Name</b>	Sharing posts
<b>Actors</b>	User
<b>Initial Assumption</b>	The site is available and the user already has account and he wants to share specific post
<b>Normal</b>	<p>The user will be selecting specific knock and will click on button “Share”, and the user can have the option of sharing the link or sharing the knock.</p> <p>The system will be triggered with the share button clicking and then the model will validate the knock if it is available or not an after the validation, the system displays the shared knock.</p>
<b>What can go wrong</b>	<ol style="list-style-type: none"> <li>1. If the knock is no longer available</li> <li>2. There is not internet connection</li> </ol>
<b>Other Activities</b>	The user can proceed all operations which sharing the post
<b>System on Completion</b>	A success message that the post sharing has been done successfully, or a failure message that the conditions of post sharing cannot be satisfied to accomplish this function.

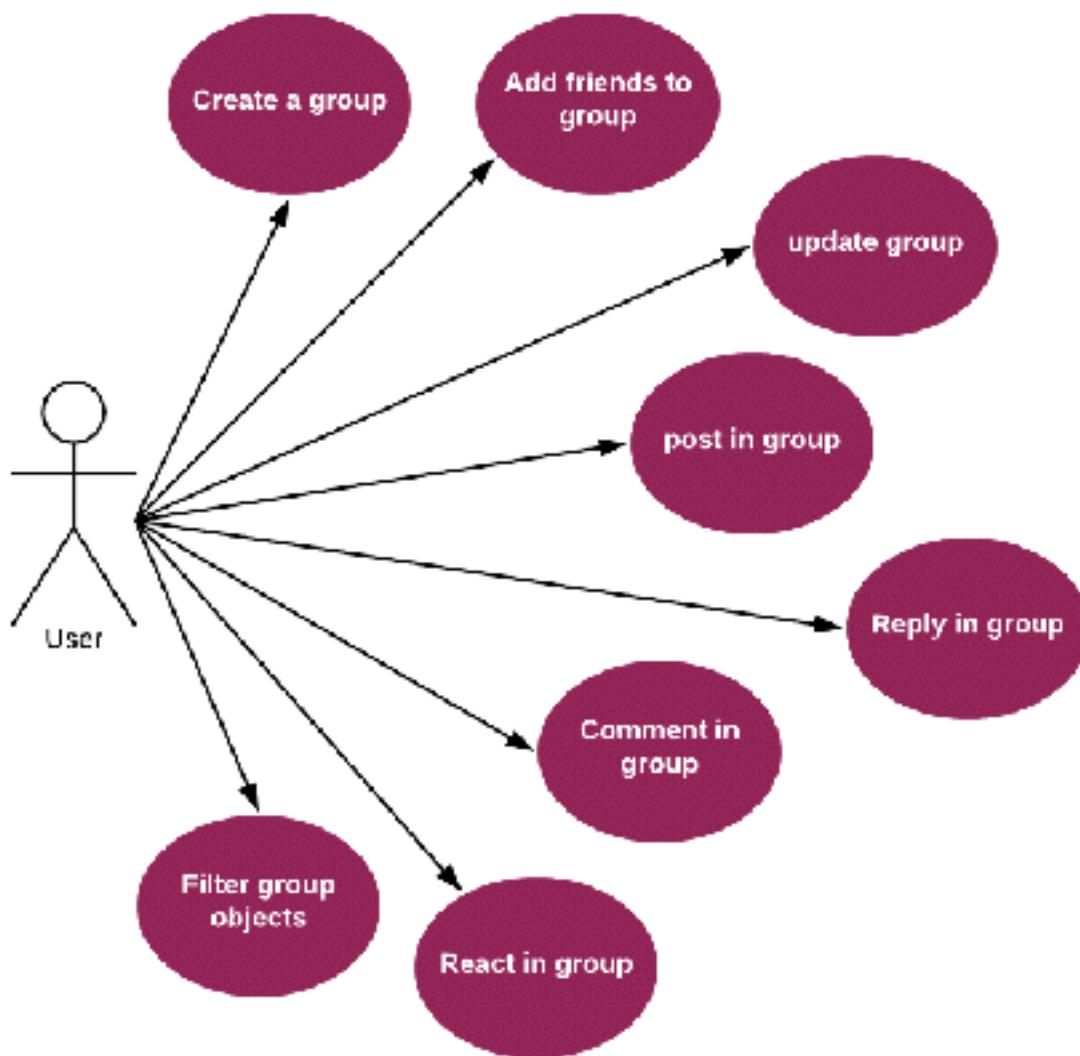
## 3.2. Use case Model

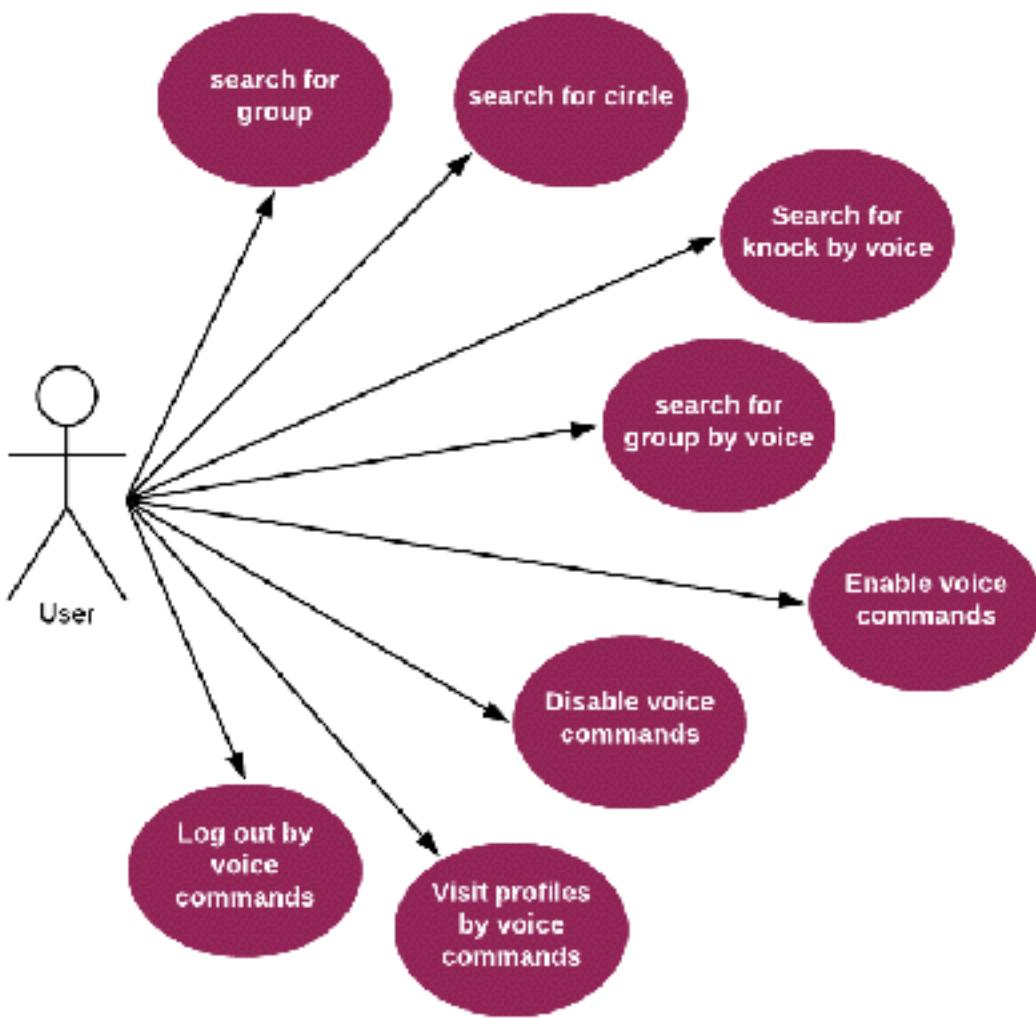
### 3.2.1. User

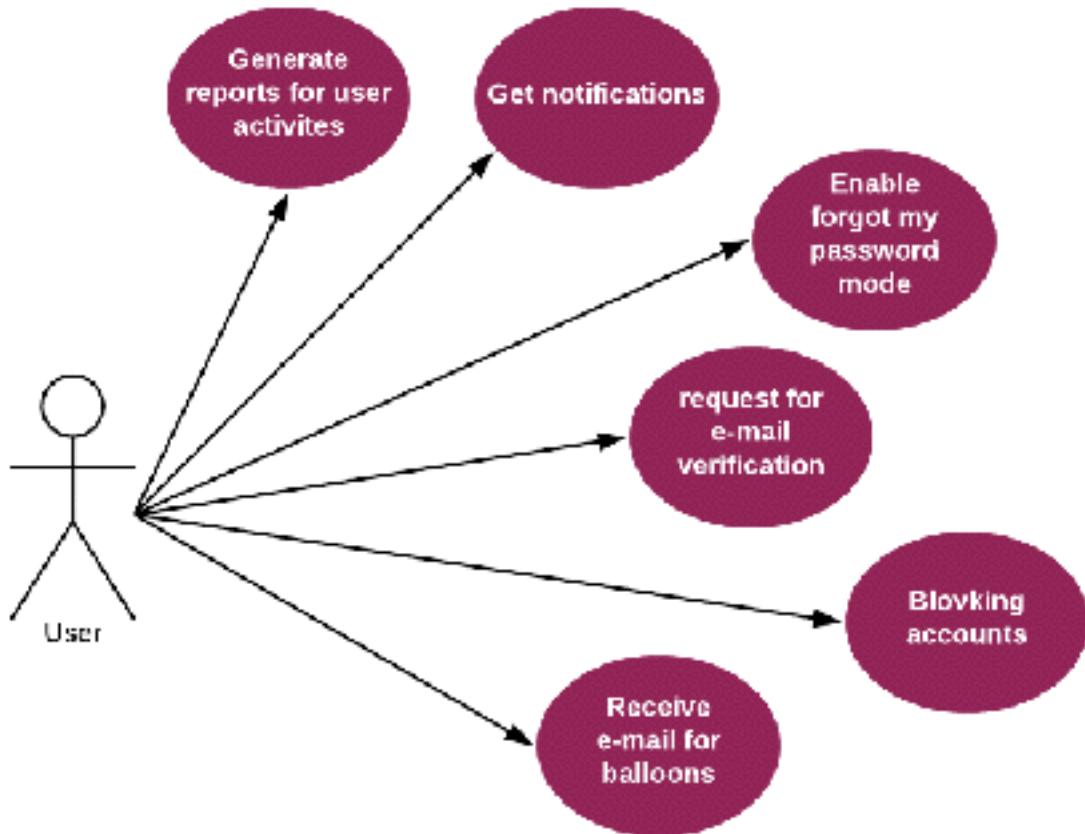


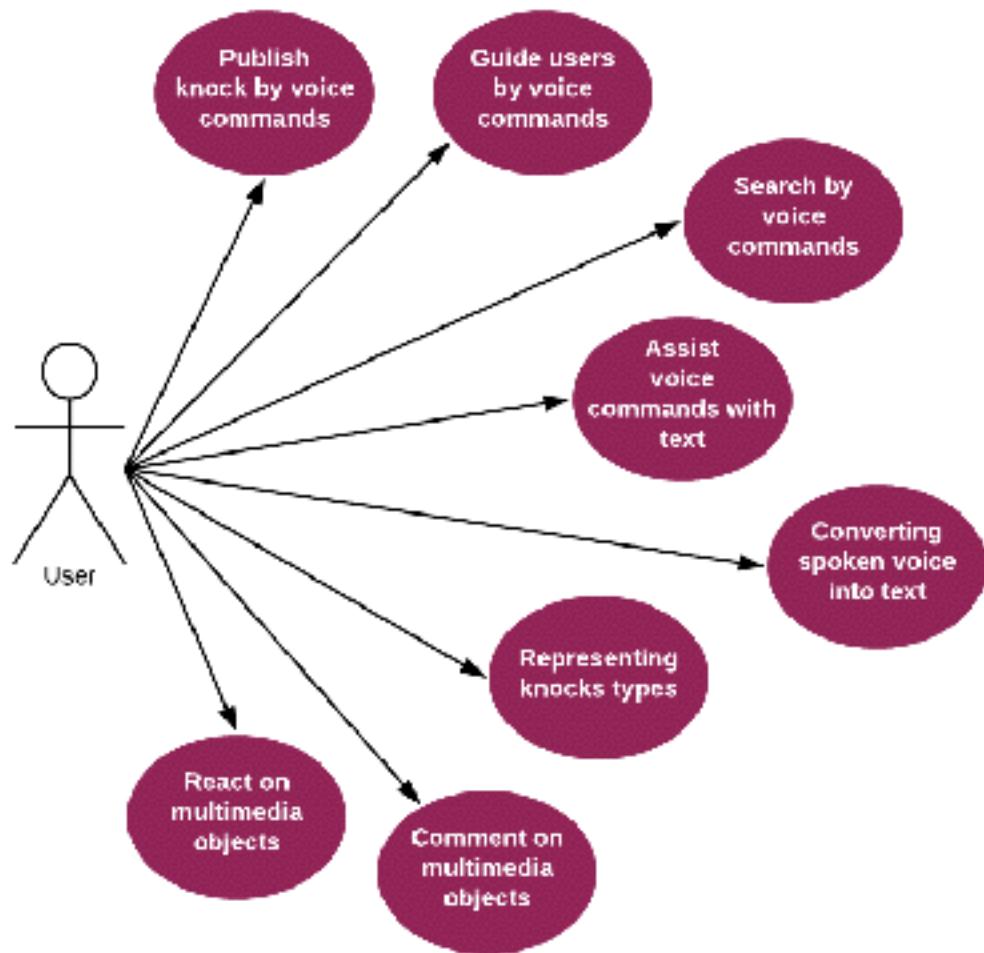


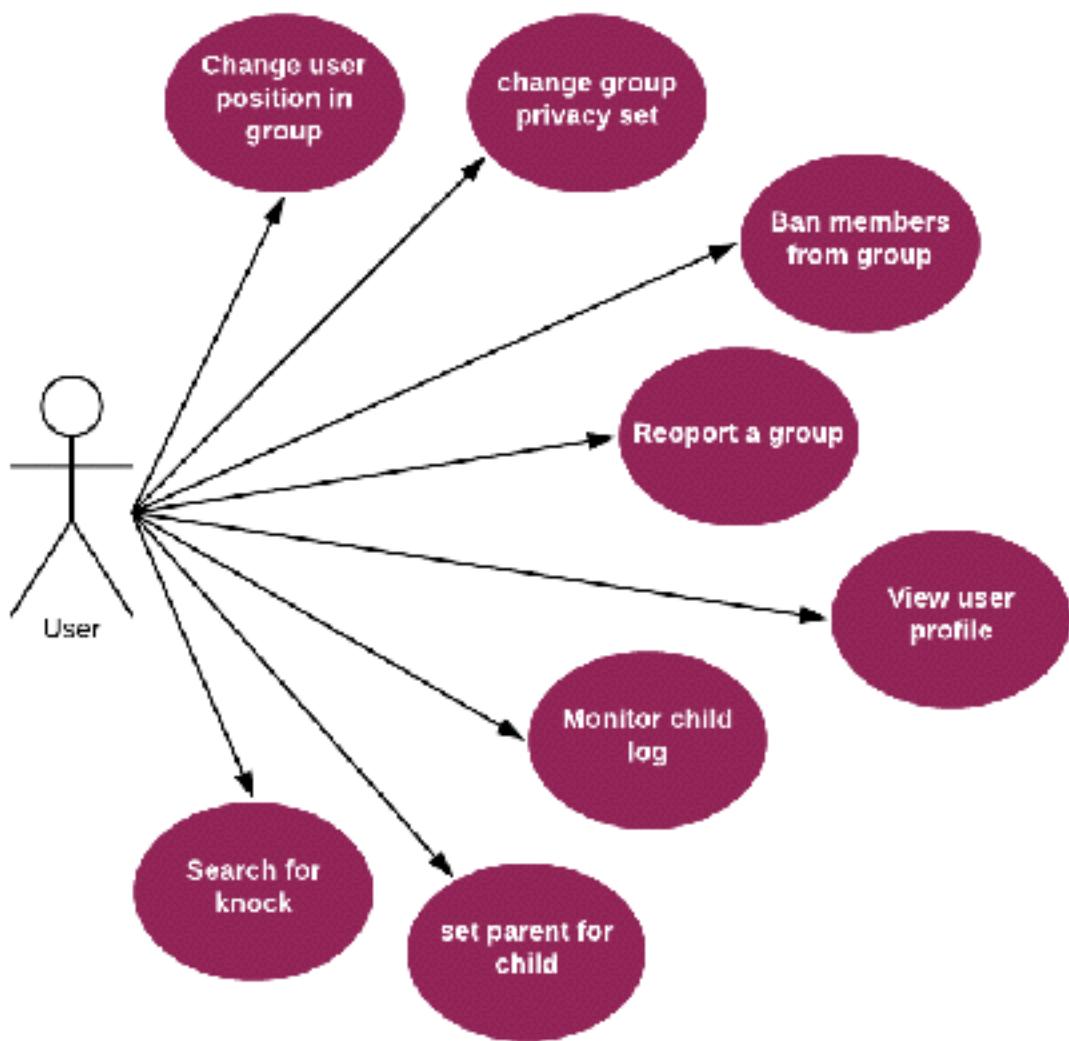












### 3.3. Use case Specification Tables

#### 3.3.1. Publish a knock

Use case name	Publish a Knock	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>The user will fill in the post area with text, upload a picture, album, video, record, check in, or multiple options among them.</li> <li>The user will then click the button “Publish”.</li> </ol>	<ol style="list-style-type: none"> <li>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 “”.</li> </ol>

#### 3.3.2. Write a Comment

Use case name	Write a Comment	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>There should be a post to comment for.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>The user will fill in the comment area with text, upload a picture, album, video, record, or multiple options among them.</li> <li>The user will then click the button “”.</li> </ol>	<ol style="list-style-type: none"> <li>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 “”.</li> </ol>

### 3.3.3. Write a Reply

Use case name	Write a Reply	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in</li> <li>There should be a comment to reply for</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>The user will fill in the reply area with text, upload a picture, album, video, record, or multiple options among them.</li> <li>The user will then click the button “”.</li> </ol>	<ol style="list-style-type: none"> <li>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 ”“.</li> </ol>

### 3.3.4. Add a New Circle

Use case name	Add a New Circle	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The user should not duplicate circle names.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>The user will click on circle icon, then clicking on “adding new circle” button.</li> <li>The user will provide the circle name, thumbnail, and the participant friends there.</li> <li>The user will click on “add circle” button.</li> </ol>	<ol style="list-style-type: none"> <li>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.</li> <li>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”.</li> </ol>

### 3.3.5. Delete a Circle

Use case name	Add a New Circle	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The user should have the specified circle available to be deleted.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>The user will select a circle and then in circle settings he will click the button “delete”.</li> </ol>	<ol style="list-style-type: none"> <li>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”</li> <li>The user can later add a new circle with the same name of the deleted circle</li> </ol>

### 3.3.6. Update a Circle Name

Use case name	Update a Circle Name	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The user should have the specified circle available to be Updated.</li> </ul>	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none"> <li>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.</li> </ol>	<ol style="list-style-type: none"> <li>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”.</li> <li>The user can later add members to the updated circle or delete this circle or update it again.</li> </ol>

### 3.3.7. 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"> <li>The site is available and the user must log in.</li> <li>The user should have the specified circle available to add friends to it.</li> <li>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.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>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.</li> </ol>	<ol style="list-style-type: none"> <li>2. 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”.</li> <li>3. The user can later add more members or exclude members from the circle.</li> </ol>

### 3.3.8. 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"> <li>The site is available and the user must log in.</li> <li>The user should have the specified circle available to add friends to it.</li> <li>The account owner should have that friend added to the selected circle.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>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.</li> </ol>	<ol style="list-style-type: none"> <li>2. 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”.</li> <li>3. The user can later add more members or exclude members from the circle.</li> </ol>

### 3.3.9. Regulate a User Privacy set

Use case name	Regulate a User Privacy Set	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The account owner should have that friend added to the any circle of his/her.</li> </ul>	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none"> <li>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.</li> <li>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”.</li> <li>The user can later update this privacy set for this user again.</li> </ol>	

### 3.3.10. Regulate a Circle Privacy set

Use case name	Regulate a User Privacy Set	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The account owner should have that circle owned by him/her.</li> </ul>	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none"> <li>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.</li> <li>2. 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”.</li> <li>3. The user can later update this privacy set for this circle again.</li> </ol>	

### 3.3.11.Download a File

Use case name	Download a File	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The File has to be Available at the site.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>The user will click “Download” button below the desired File.</li> </ol>	<ol style="list-style-type: none"> <li>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”.</li> </ol>

### 3.3.12.Upload a Profile Picture

Use case name	Upload a Profile Picture	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>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.</li> </ol>	<ol style="list-style-type: none"> <li>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”.</li> </ol>

### 3.3.13.Upload a Cover Picture

Use case name	Upload a Cover Picture	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>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.</li> <li>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”.</li> </ol>	

### 3.3.14.Update a Profile Picture

Use case name	Update a Profile Picture	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>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.</li> <li>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”.</li> </ol>	

### 3.3.15.Update a Cover Picture

Use case name	Update a Cover Picture	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>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.</li> </ol>	<ol style="list-style-type: none"> <li>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”.</li> </ol>

### 3.3.16.Unfriend a Friend

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

### 3.3.17. Block a Friend

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

### 3.3.18. Update Personal Information

Use case name		Update Personal Information			
Initiating Actor	User				
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>There must be a previously initialized personal information.</li> </ul>				
Flow of Events	User Steps	System Steps			
	<ol style="list-style-type: none"> <li>The user will enter his own profile and select the desired section of personal information, then he will click "Edit" button.</li> </ol>	<ol style="list-style-type: none"> <li>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".</li> <li>The user can later update personal information again.</li> </ol>			

### 3.3.19.React to a Post

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

### 3.3.20.React to a Comment

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

### 3.3.21.React to a Reply

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

### 3.3.22.Create a Group

Use case name	Create a Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>The user will be in his profile page and then click “Create a Group” button.</li> <li>Then the user will supply the group information to initiate the group such as “Name”, “Category”, “Thumbnail”, “Preset”, “Picture”, “Group Preset”.</li> <li>After that the user can optionally choose the members of his group to be added by selecting the friends after searching on them.</li> <li>Then the user will click “Confirm” button.</li> </ol>	<ol style="list-style-type: none"> <li>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”.</li> <li>The user can later update the group initials, Remove members, or delete this group.</li> </ol>

### 3.3.23.Add Friends to a Group

Use case name	Add Friends to a Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The user must have the privilege to add members to the group.</li> </ul>	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none"> <li>The user will enter the Group that has been created previously and then he will click “Add Members” button.</li> <li>The user will search for his friends to select the friends to be added.</li> <li>Then he will click “Add” button.</li> </ol>	<ol style="list-style-type: none"> <li>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”.</li> <li>The user can later update the group initials, Remove members, or delete this group.</li> </ol>

### 3.3.24.Update Group

Use case name	Update Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The user must have the privilege to Update the group.</li> </ul>	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none"> <li>The user will enter the Group that has been created previously and then he will click “Update” button.</li> <li>The user will click on the fields to be updated then he will supply the new information.</li> <li>Then he will click “Update” button.</li> </ol>	<ol style="list-style-type: none"> <li>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”.</li> <li>The user can later update the group again initials, Remove members, or delete this group.</li> </ol>

### 3.3.25.Post in a Group

Use case name	Post in a Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The user have to be a member of the group.</li> <li>The user must have the privilege to post in the group.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>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.</li> </ol>	<ol style="list-style-type: none"> <li>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”.</li> <li>The user can later delete the post or edit it.</li> </ol>

### 3.3.26.Comment in a Group

Use case name	Comment in a Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The user have to be a member of the group.</li> <li>The user must have the privilege to comment in the group.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>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.</li> </ol>	<ol style="list-style-type: none"> <li>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”.</li> <li>The user can later delete the comment or edit it.</li> </ol>

### 3.3.27.Reply in a Group

Use case name	Reply in a Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The user have to be a member of the group.</li> <li>The user must have the privilege to reply in the group.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>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.</li> </ol>	<ol style="list-style-type: none"> <li>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”.</li> <li>The user can later delete the reply or edit it.</li> </ol>

### 3.3.28.React to an Object in a Group

Use case name	React to a Message	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The user have to be one of the group members.</li> <li>The user must have the privilege to reply in the group.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>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.</li> </ol>	<ol style="list-style-type: none"> <li>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.</li> <li>If the user has chosen the emoji, the system will display the selected emoji on the object selected.</li> <li>The user can later update the emoji with another one or just delete his reaction.</li> </ol>

### 3.3.29.Update a Privacy Preset in a Group

Use case name	Update Group	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The user must have the privilege to Update the group privacy preset.</li> </ul>	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none"> <li>The user will enter the Group that has been created previously and then he will click “Update” button.</li> <li>The user will click on the privacy preset field to be updated then he will choose the new privacy preset.</li> <li>Then he will click “Update” button.</li> </ol>	<ol style="list-style-type: none"> <li>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”.</li> <li>The user can later update the group again initials, Remove members, or delete this group.</li> </ol>

### 3.3.30. View User's Information

Use case name	View User's Information	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>The user will be clicking on the other user name.</li> </ol>	<ol style="list-style-type: none"> <li>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.</li> </ol>

### 3.3.31. Monitor Child Log

Use case name	Analysis for the Talent Owner	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in.</li> <li>The user who is monitoring has to be a verified parent for that kid</li> </ul>	
Flow of Events	User Steps	System Steps
	<ol style="list-style-type: none"> <li>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.</li> </ol>	<ol style="list-style-type: none"> <li>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.</li> </ol>

### 3.3.32. 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"> <li>The site is available and the user must log in.</li> </ul>	
	User Steps	System Steps
Flow of Events		<ol style="list-style-type: none"> <li>1. 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.</li> <li>2. 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”.</li> <li>3. 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”.</li> <li>4. Later the parent can monitor the log of this kid and the kid may update parents account or deleted them.</li> </ol>

### 3.3.33. Comment on multimedia object

Use case name	Comment on multimedia object	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in, and there is a post object to commented on</li> </ul>	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none"> <li>The user comments ob the multimedia object and clicks the button “Publish”</li> </ol>	<ol style="list-style-type: none"> <li>If the object is not available the system will display a failure message to the comment creator “Cannot Publish”/“Unable to proceed because of the object is not available anymore”, or else the system will publish the comment</li> </ol>

### 3.3.34. Reply on multimedia object

Use case name	Reply on multimedia object	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must log in, and there is a comment object to replied on</li> </ul>	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none"> <li>The user replies ob the multimedia object and clicks the button “Publish”</li> </ol>	<ol style="list-style-type: none"> <li>If the object is not available the system will display a failure message to the reply creator “Cannot Publish”/“Unable to proceed because of the object is not available anymore”, or else the system will publish the reply</li> </ol>

### 3.3.35. Generate report for user's activities

Use case name	Generate reports for user's activities	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must be logged in</li> </ul>	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none"> <li>The user renters settings and then goes into devices section two view his logs</li> </ol>	<ol style="list-style-type: none"> <li>The system will display the user logs with the devices, operating systems, last log in, log in times, and first log in.</li> </ol>

### 3.3.36. Notifying the user ( Eg : Reaction )

Use case name	Notifying the user for reaction balloon	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user must be logged in</li> </ul>	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none"> <li>As a sender balloon, the user can create the balloon reaction by reacting to other user's knock, comment, reply, or any other activity.</li> <li>As a receiver when the user logs in and receives a balloon and he may react with this balloon or just ignore it</li> </ol>	<ol style="list-style-type: none"> <li>The system displays the balloon for the receiver user, but on failure case the system will display a failure message "The balloon is not available anymore"</li> </ol>

### 3.3.37. Enable forgot my password mode

Use case name	Enable forgot my password mode	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user forgets his password and cannot log in</li> </ul>	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none"> <li>The user clicks on “Forgot my password” button and then he will receive e-mail after he provides his mail, and this mail will have a temporary password to allow the user to log in again</li> </ol>	<ol style="list-style-type: none"> <li>If the user did not provide correct e-mail, the system displaying failure message “please provide correct e-mail”, else if the user has a blocked account the system displaying failure message “Your account is already blocked”, else the system will send an e-mail as a success message with the temporary password</li> <li>The user can set it as temporary password or just ignore the e-mail.</li> </ol>

### 3.3.38. Request for e-mail verification

Use case name	Request for e-mail verification	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user has just registered.</li> </ul>	
	User Steps	System Steps
Flow of Events		<ol style="list-style-type: none"> <li>Then the user finishes the registration, there will be a dialog asking him to verify his account via e-mail, so the user will click on “Send me an e-mail” button</li> <li>If the user did not provide correct e-mail, the system displaying failure message “please provide correct e-mail”, else the system will send an e-mail as a success message with the verification link which directs him to his home page</li> </ol>

### 3.3.39. Blocking accounts

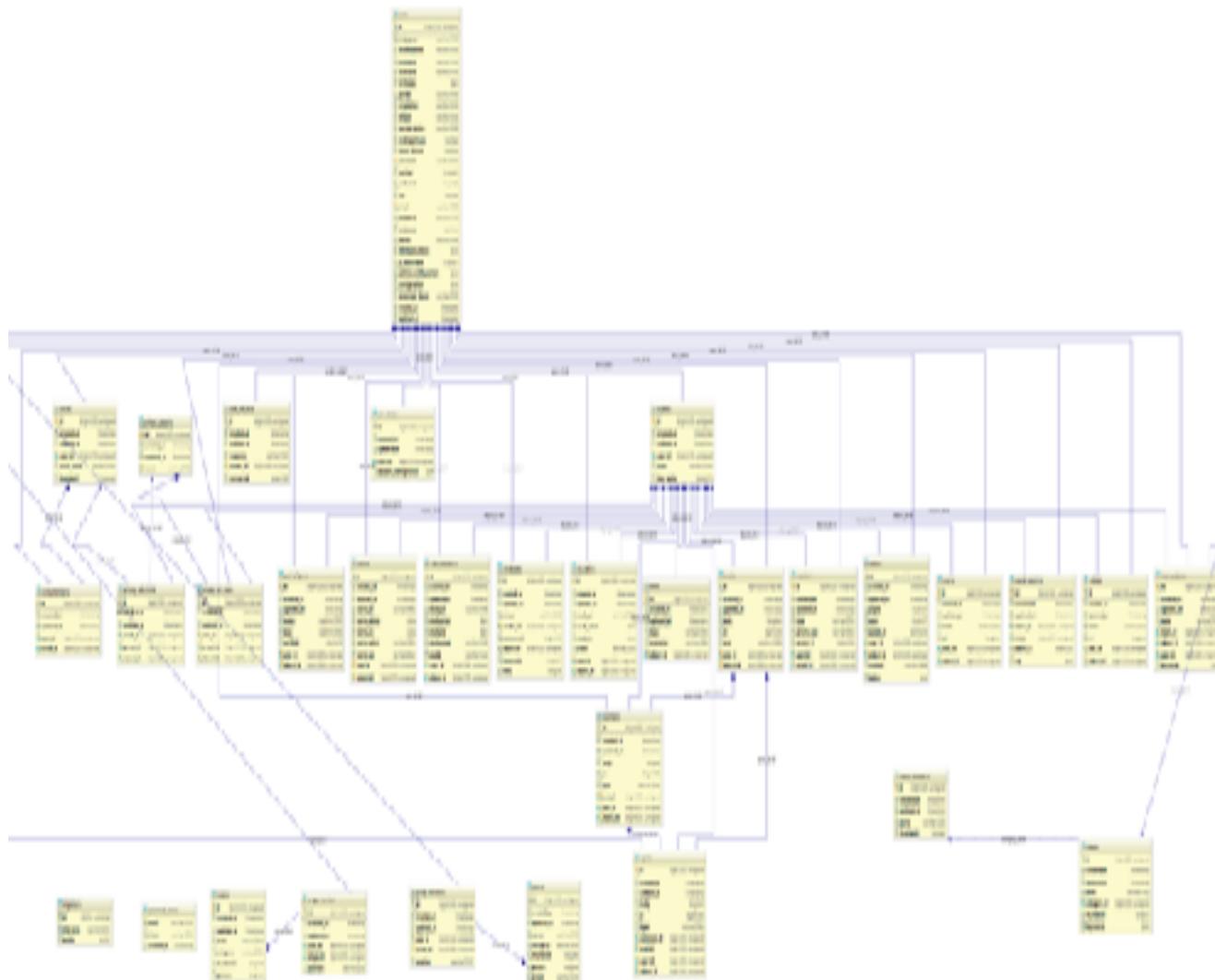
Use case name	Blocking accounts	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user already has account but his log in attempts are three or more with invalid data.</li> </ul>	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none"> <li>The user will be receiving an e-mail after the system has blocked his account when he attempted three or more log in attempts with invalid mail, phone number, e-mail, or password; so the user will get the e-mail which contains a dialog for account blocking informing, so the user will click “Send me an e-mail” button</li> <li>If the user had provided wrong e-mail and did not verify his e-mail, then the system will keep on blocking his account until the account is verified via e-mail, else the system directs the user to the log in page.</li> <li>The user can choose “Forgot my password” in case he forgets his password to avoid account blocking again</li> </ol>	

### 3.3.40. Sharing post

Use case name	Sharing post	
Initiating Actor	User	
Preconditions	<ul style="list-style-type: none"> <li>The site is available and the user already has account and he wants to share specific post</li> </ul>	
	User Steps	System Steps
Flow of Events	<ol style="list-style-type: none"> <li>The user will be selecting specific knock and will click on button “Share”,</li> <li>the user can have the option of sharing the link or sharing the knock.</li> </ol>	<ol style="list-style-type: none"> <li>The system will validate if the post is still available, and if the post is not available it displays error message “This post is not available anymore”/“You cannot share this post”, else the system will display success message “You shared the post successfully”</li> <li>The user can later delete this share</li> </ol>

### 3.4. ERD

---

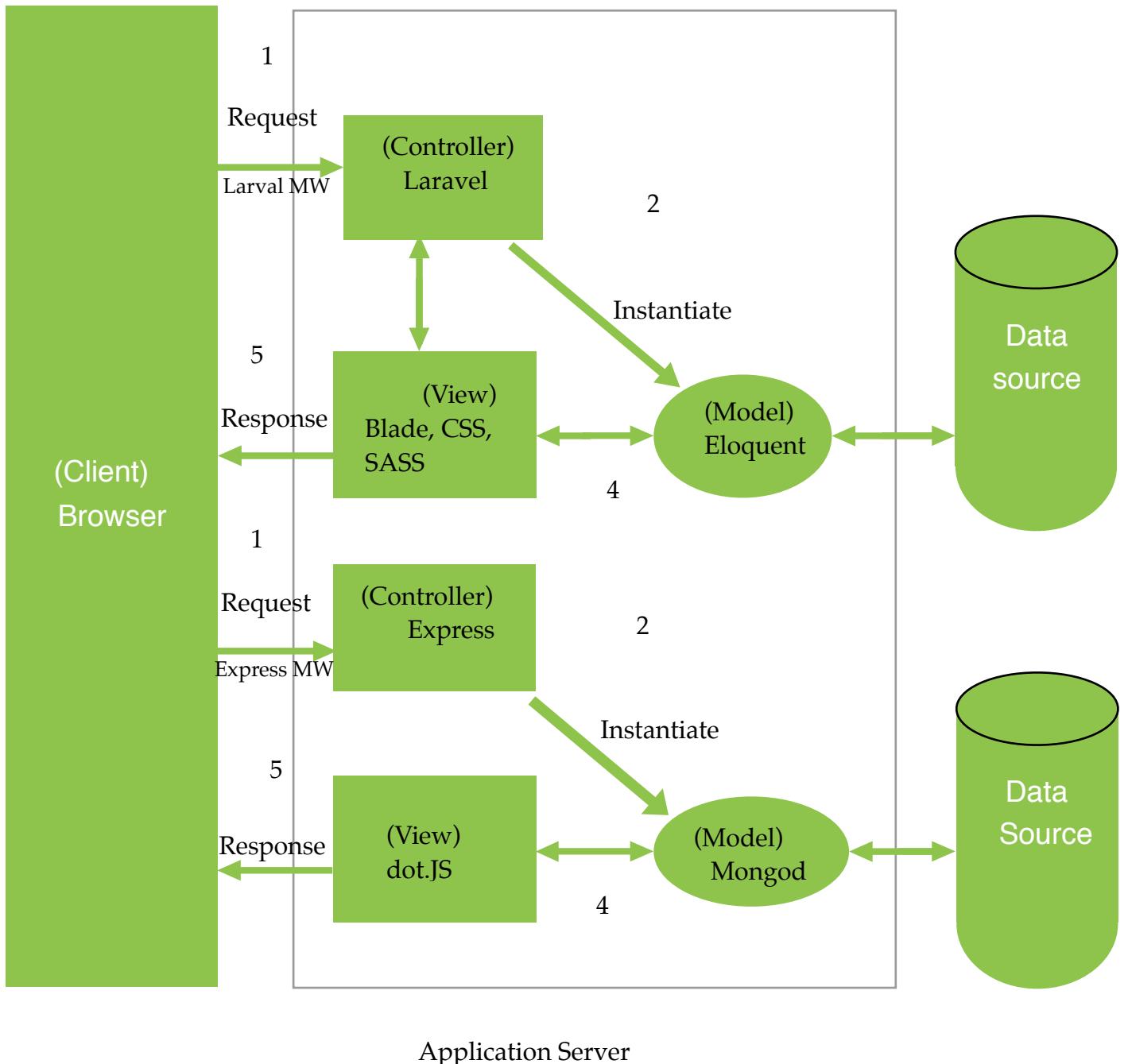


---

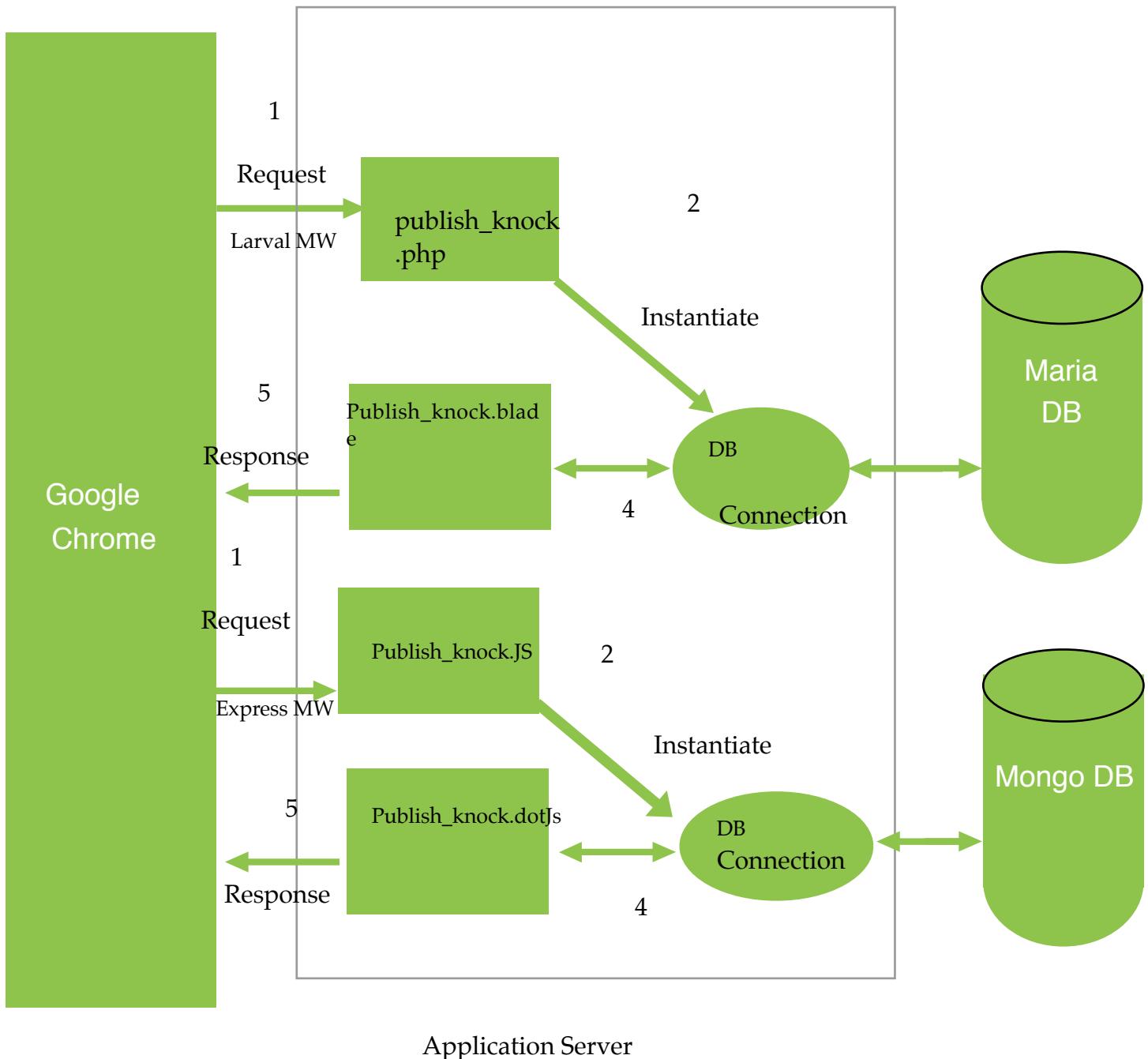
# 4. Chapter 4 : System Design

## 4.1. System Architecture

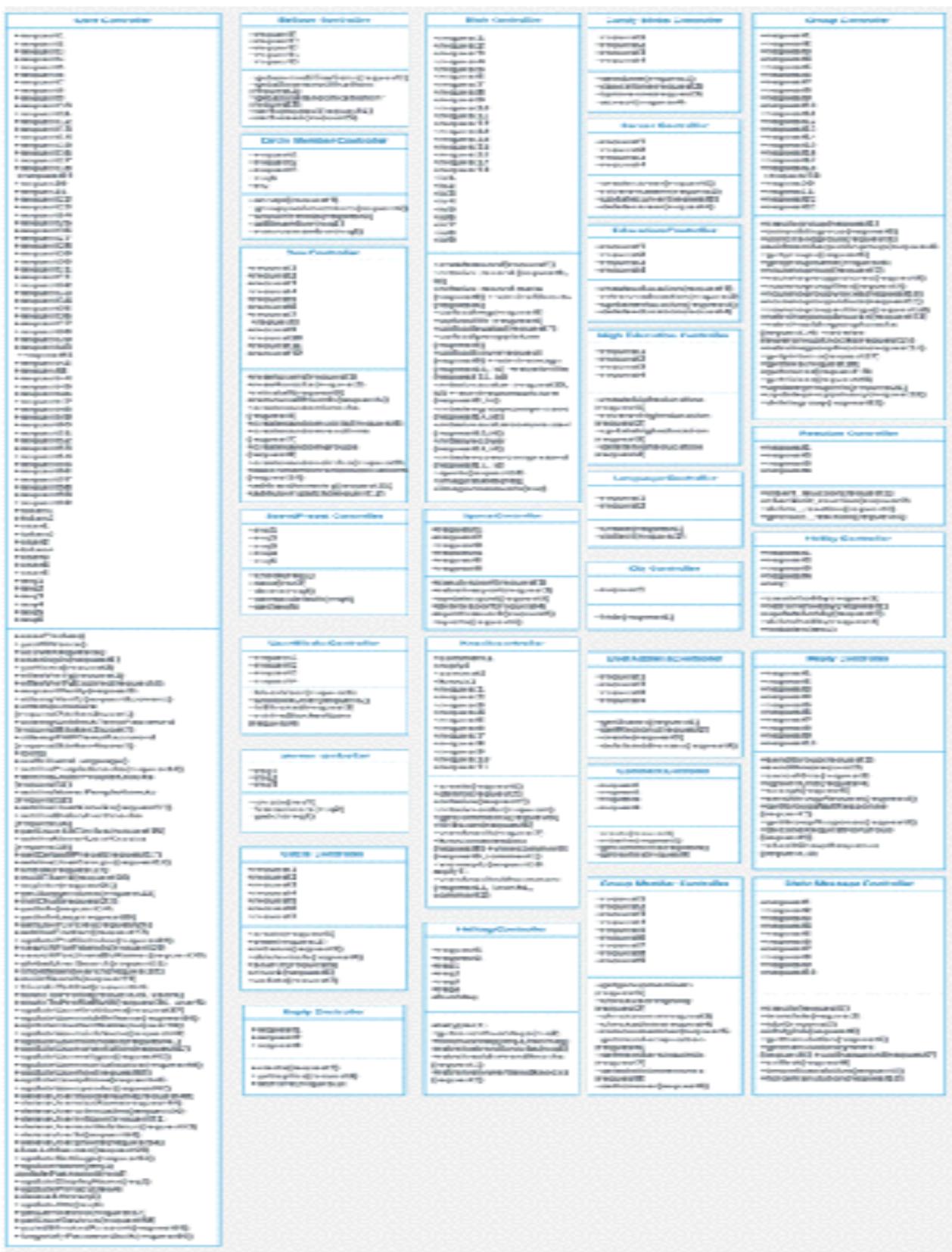
### 4.1.1. Application Architecture



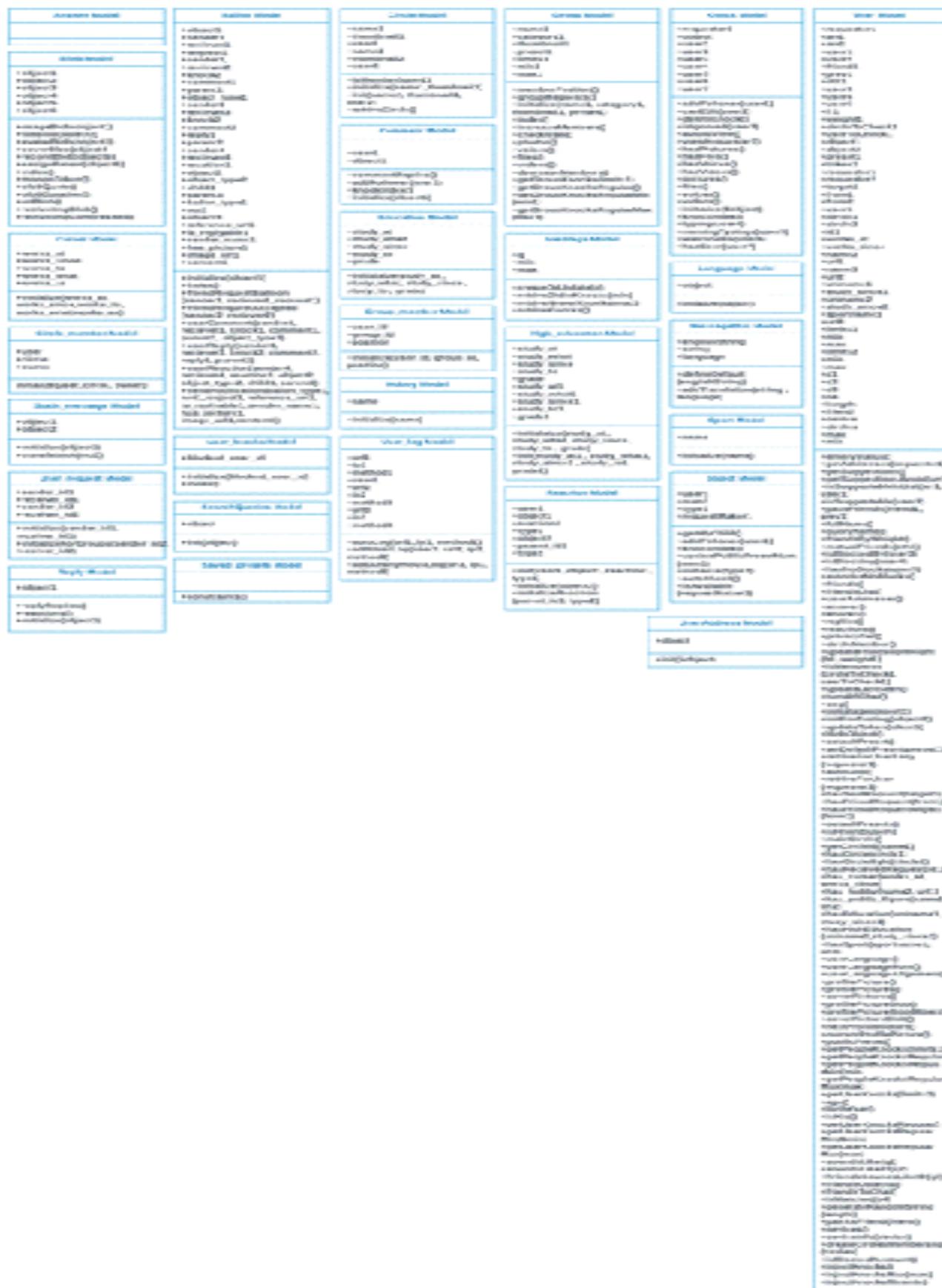
### 4.1.2. Application Architecture (Example)



## 4.2. Class Diagram

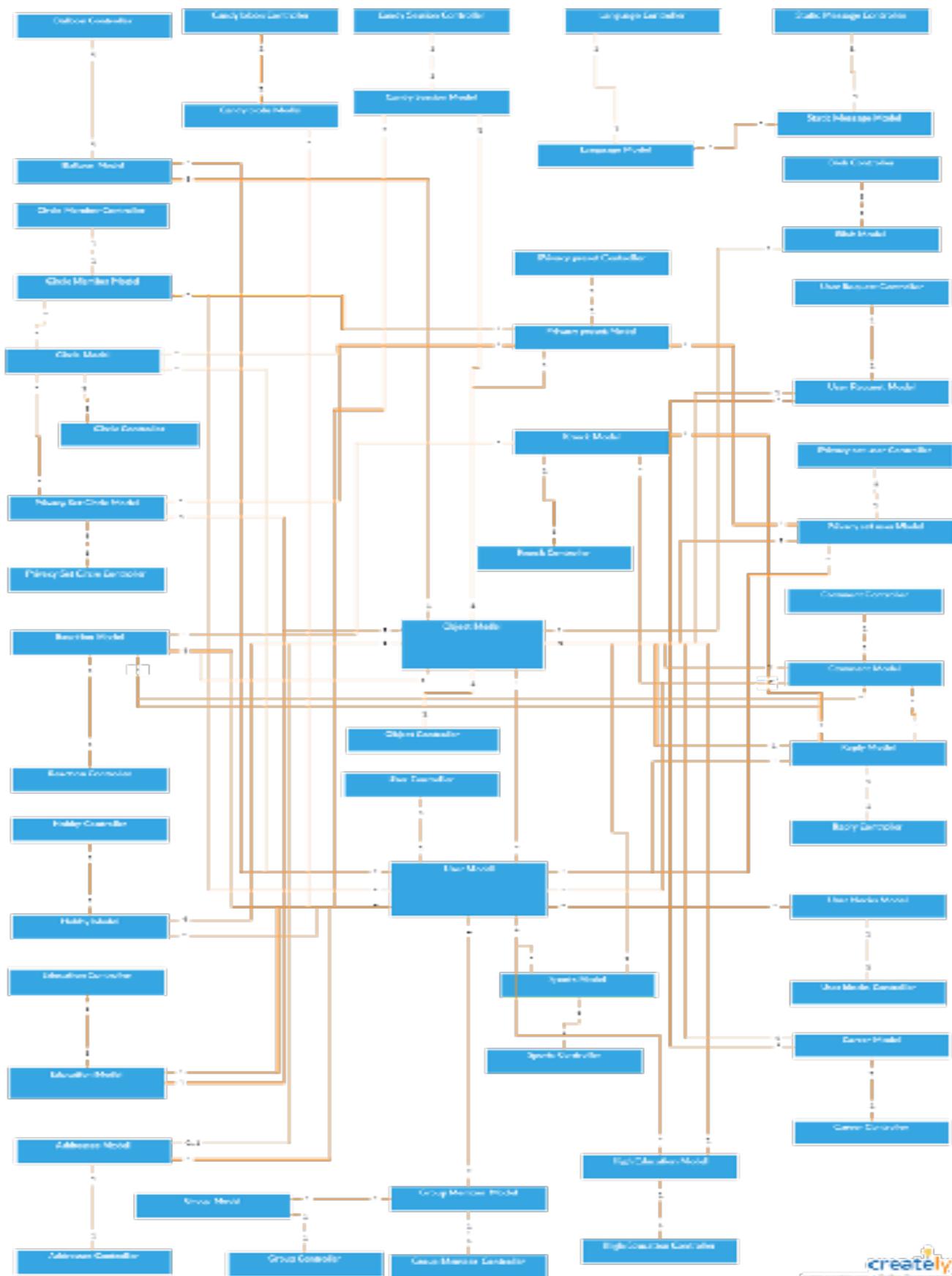


CHAPTER 4 : SYSTEM DESIGN



---

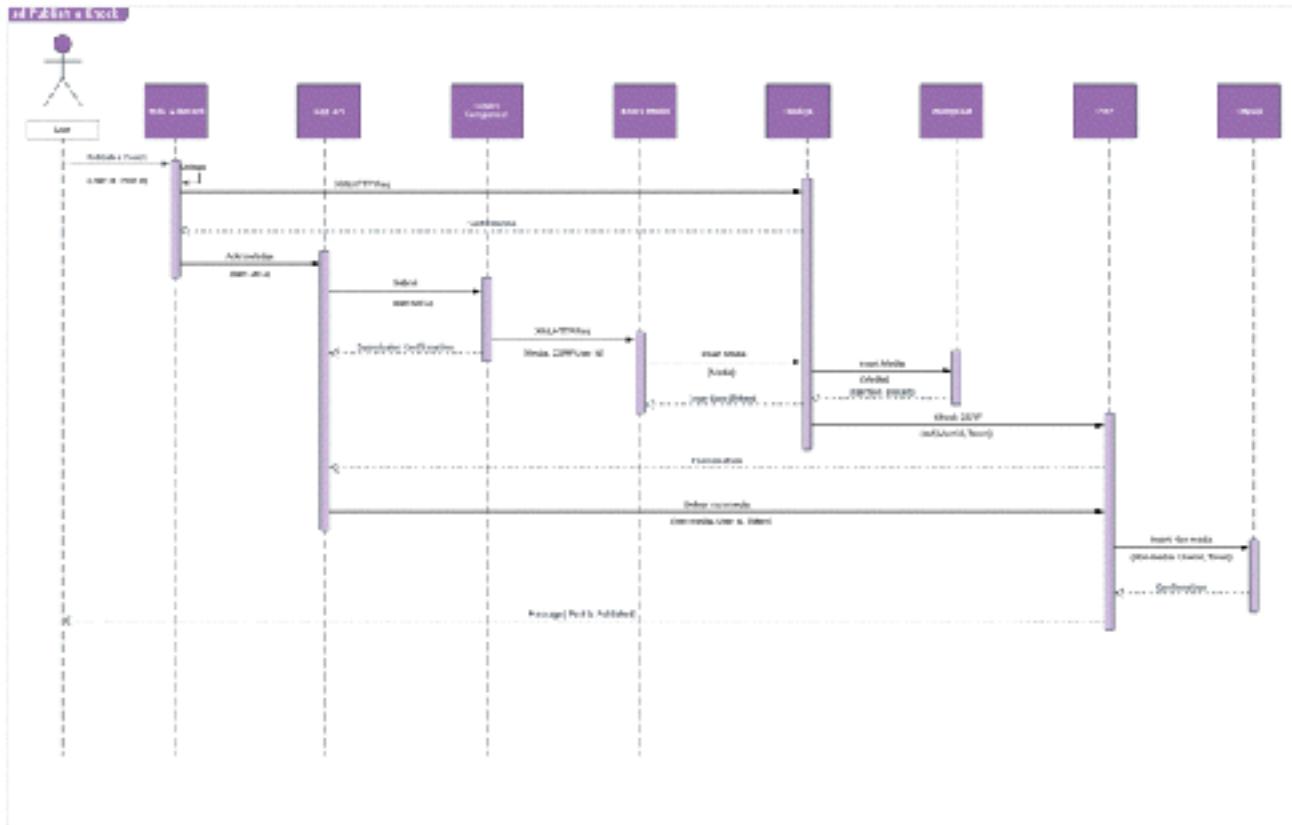
CHAPTER 4 : SYSTEM DESIGN



### 4.3. Physical Schema

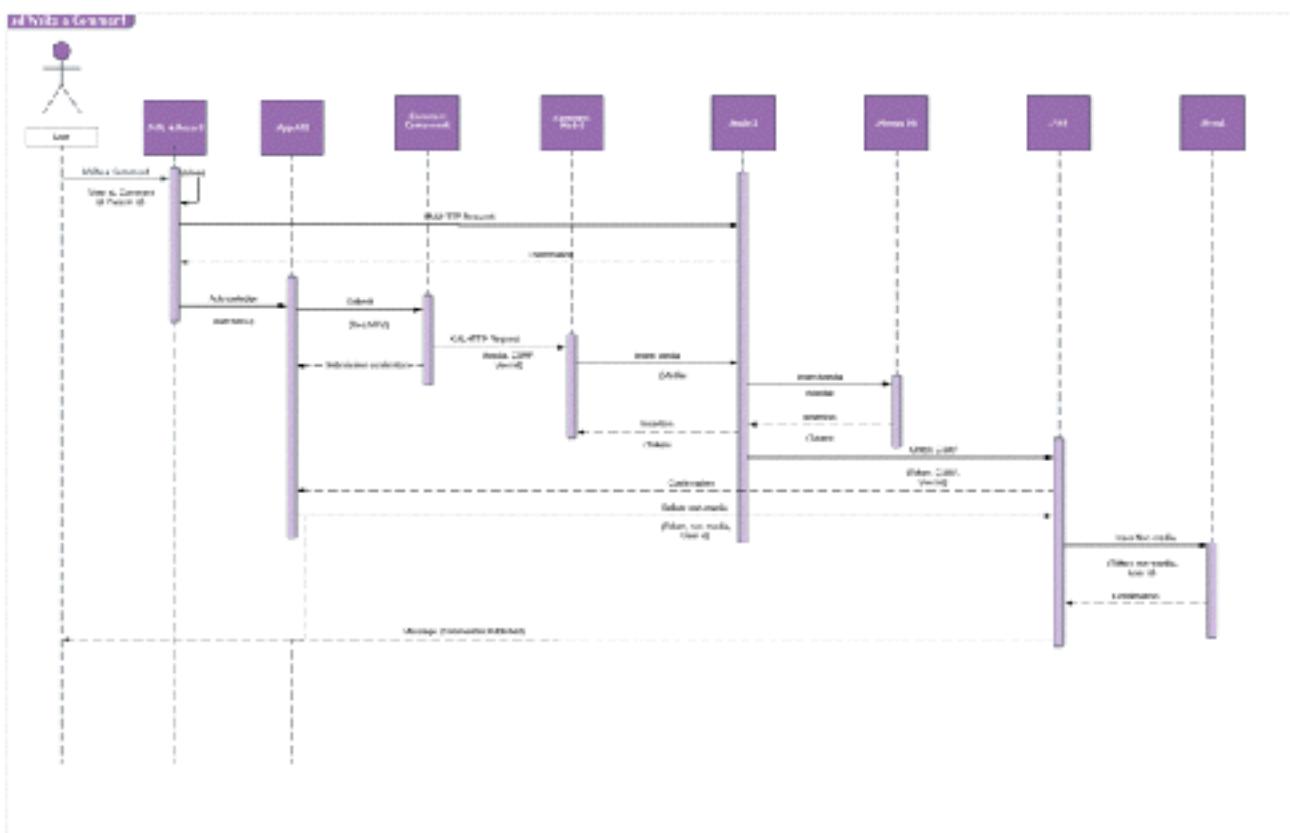
## 4.4. Sequence Diagram

- Publish a knock (create post).



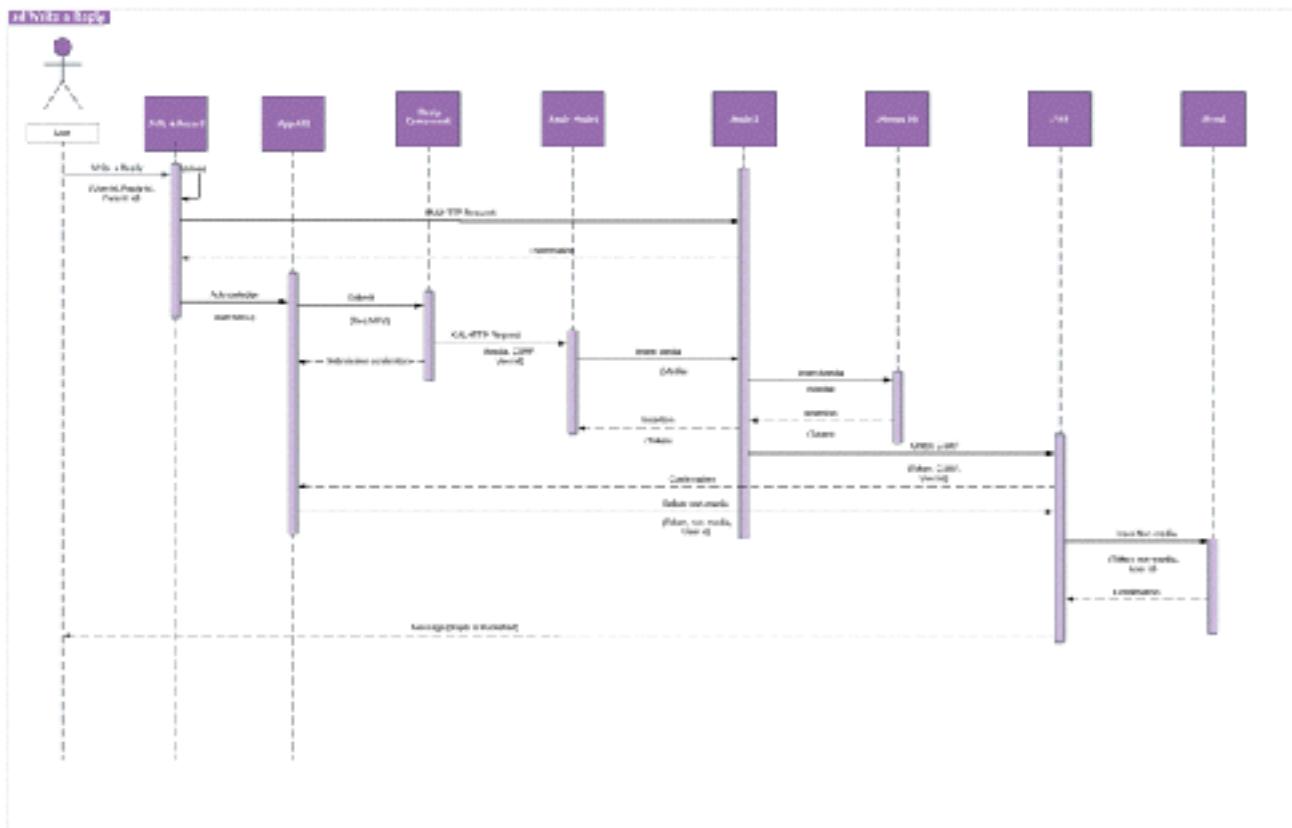
CHAPTER 4 : SYSTEM DESIGN

- Write a comment- when allowed to do so.

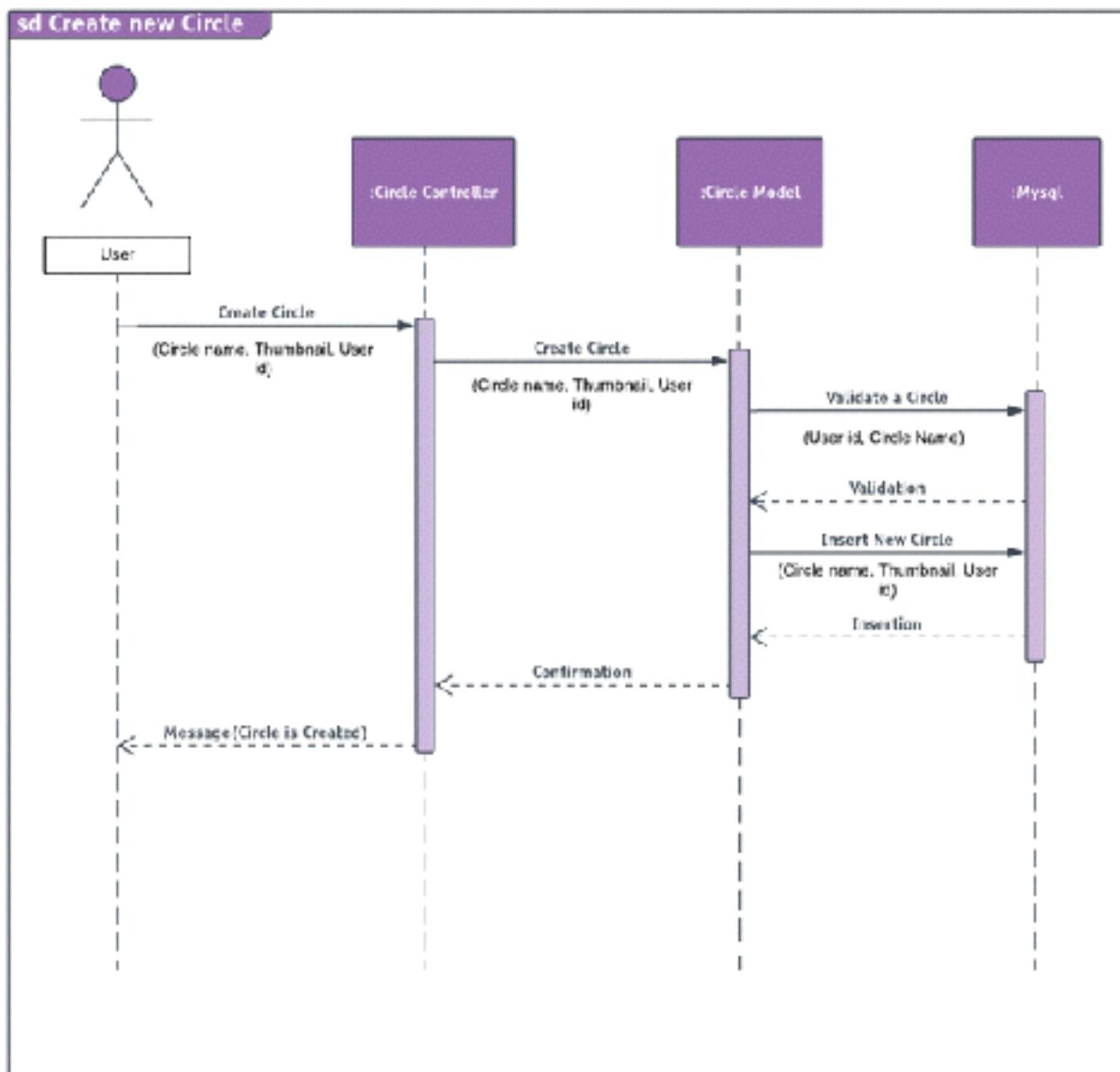


CHAPTER 4 : SYSTEM DESIGN

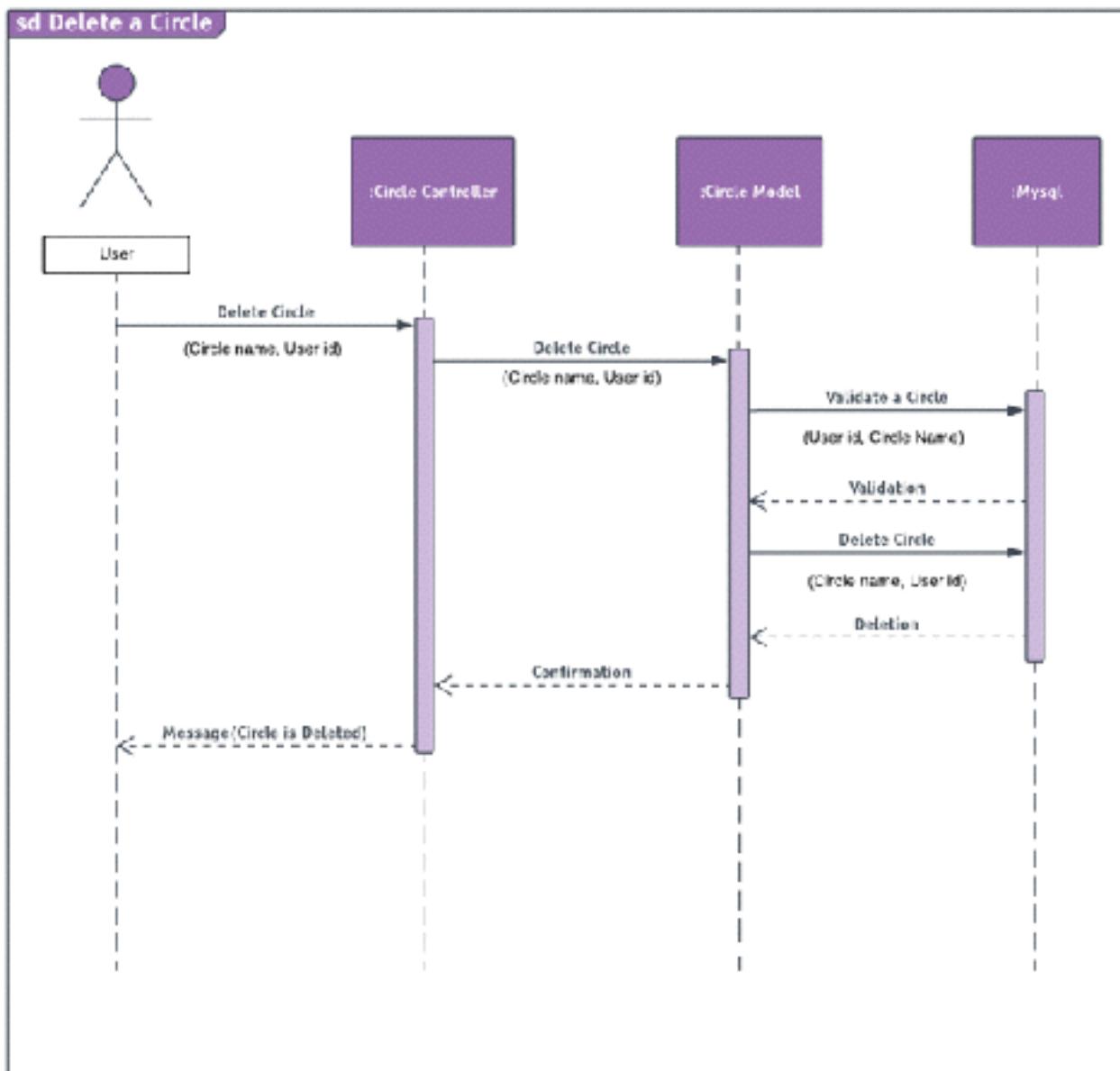
- Write a reply- when allowed to do so.



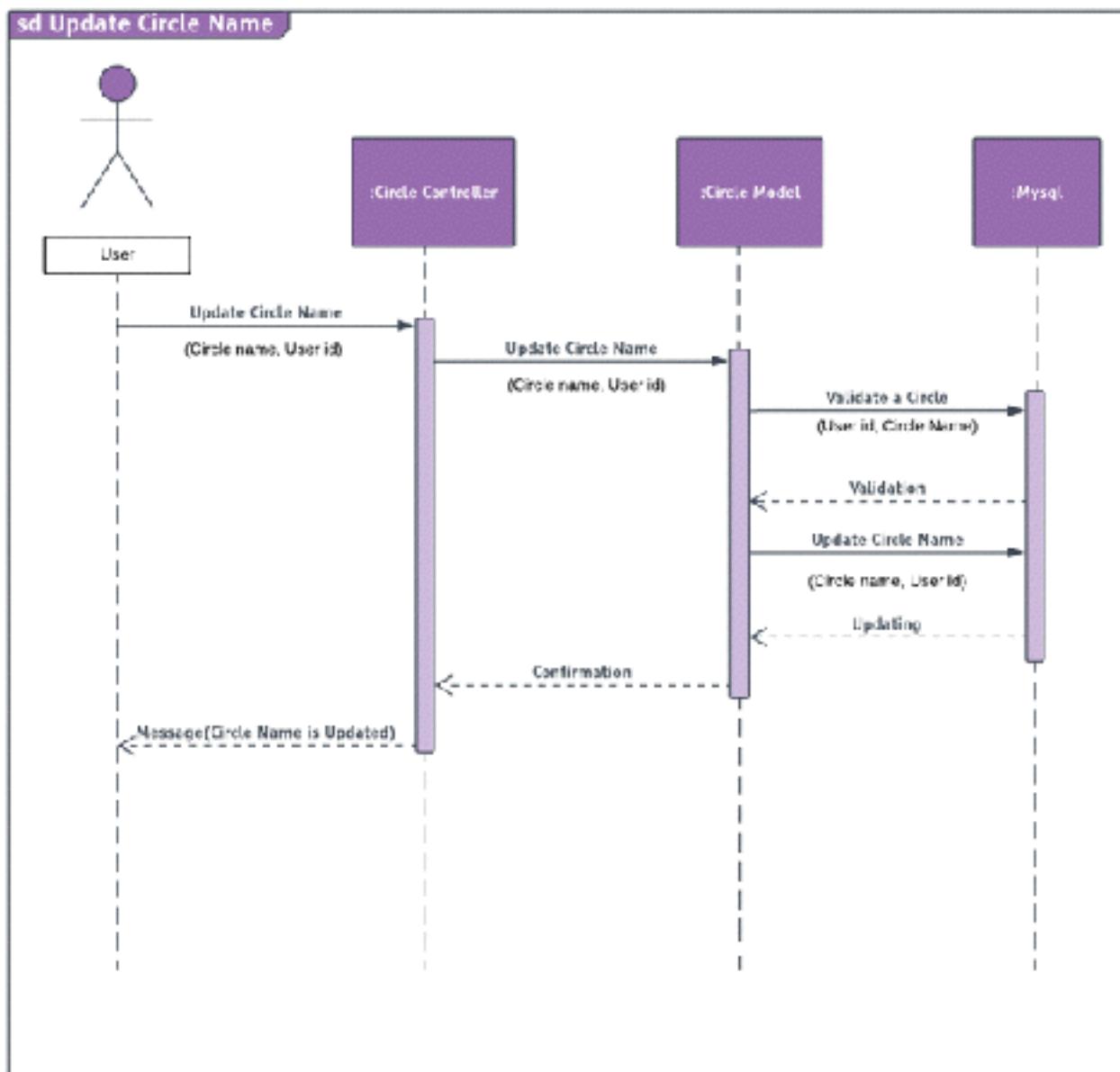
- Create a circle.



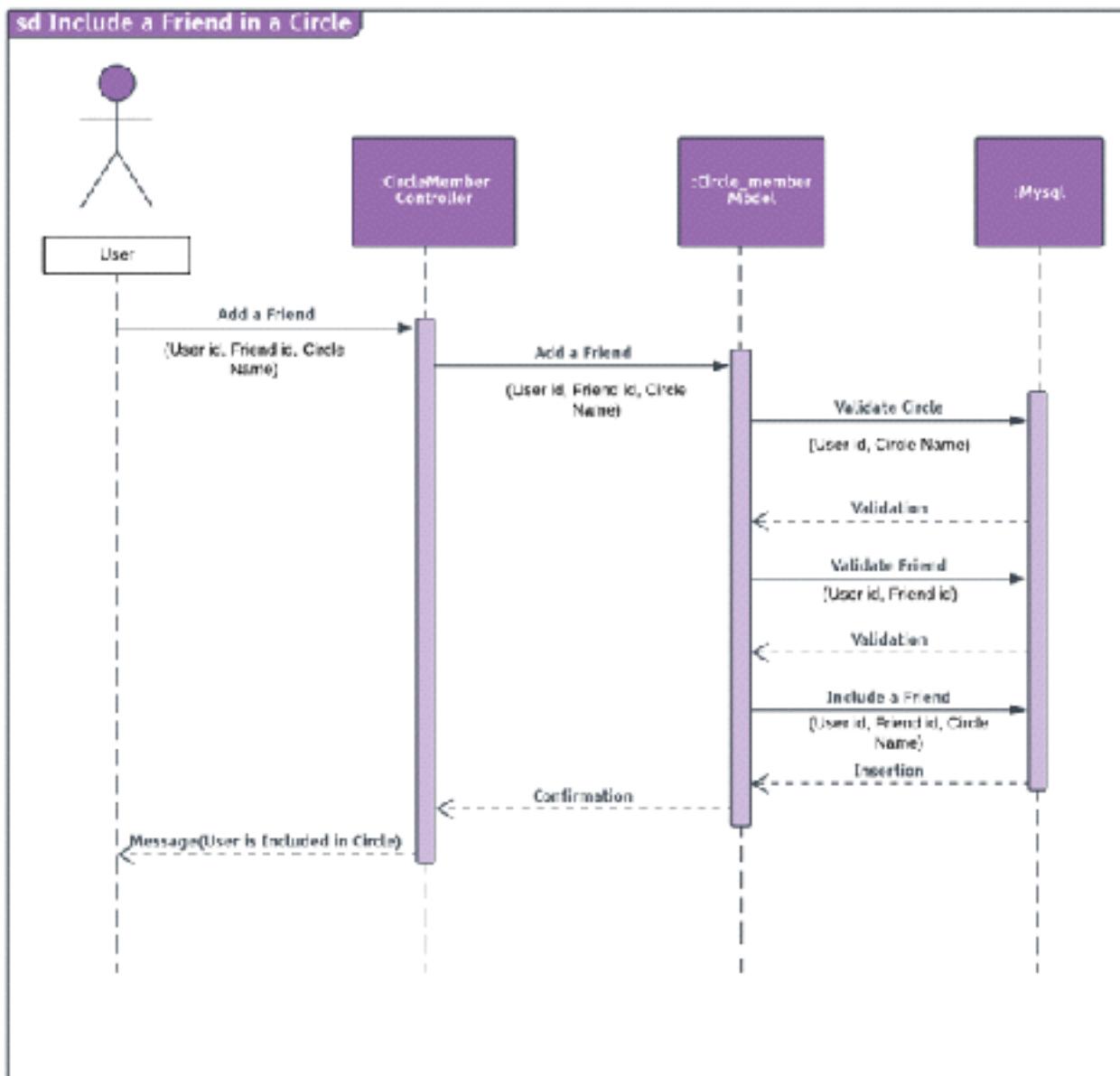
- Delete a circle



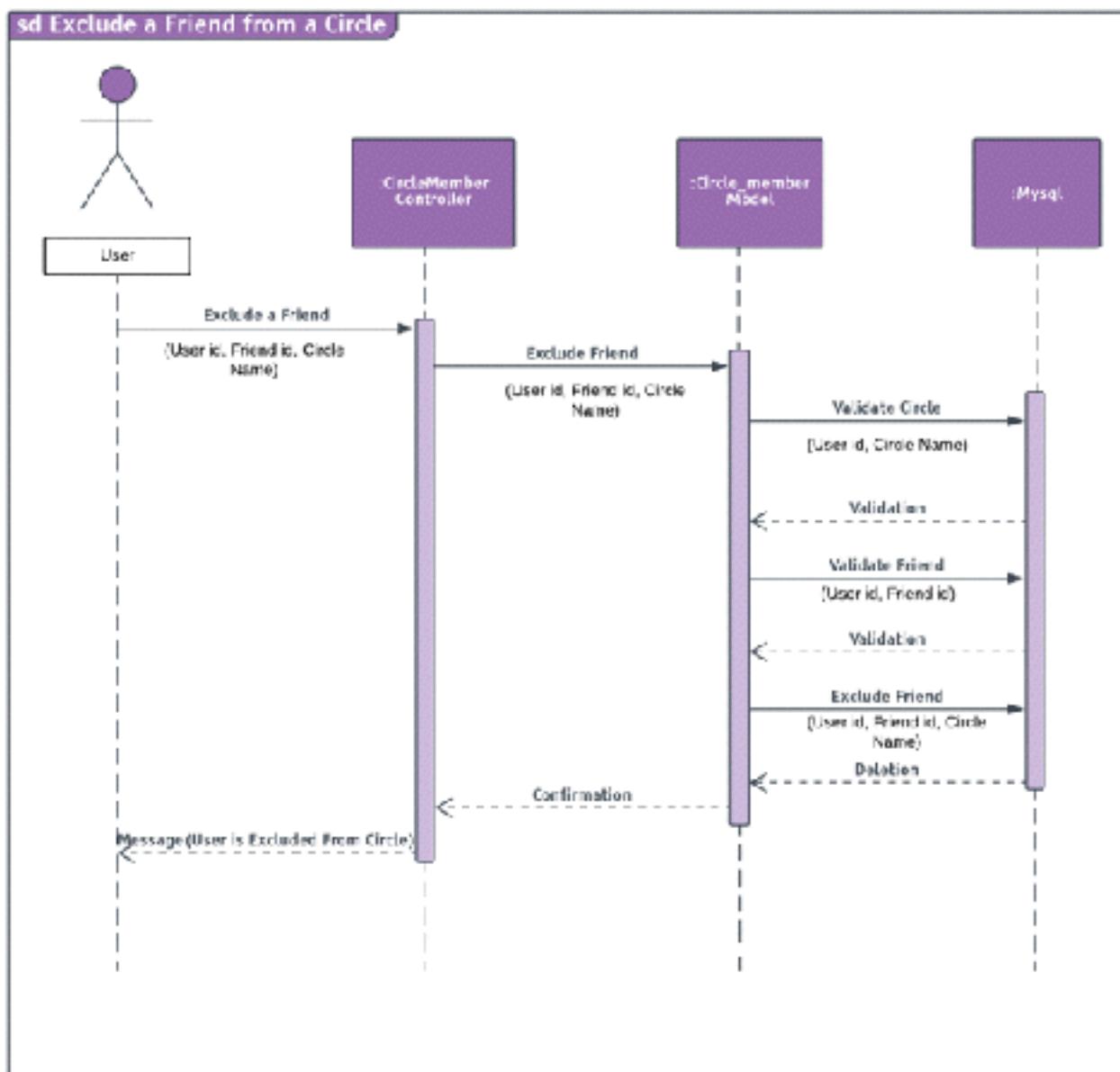
- Update circle name



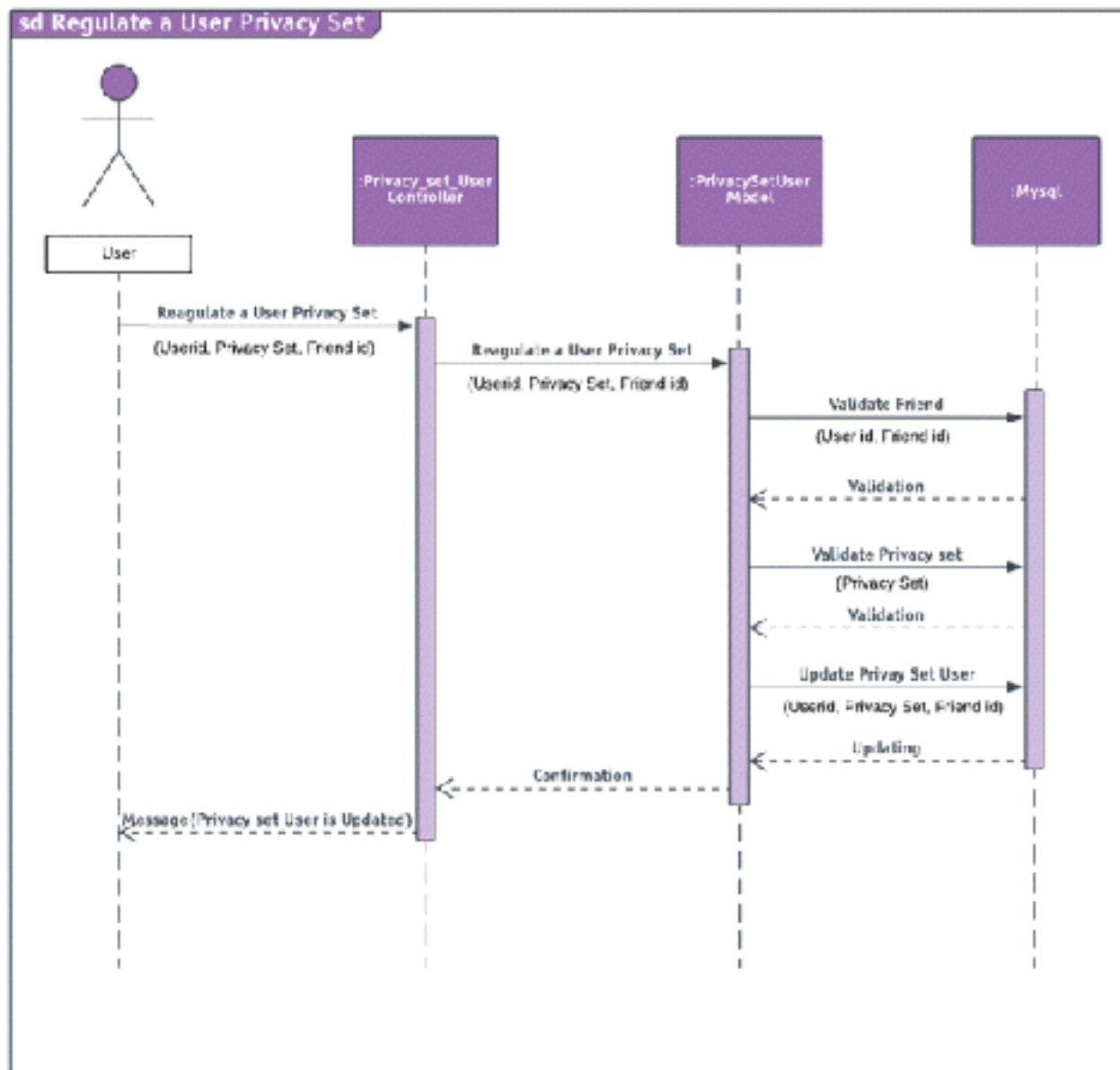
- Include a friend to a circle



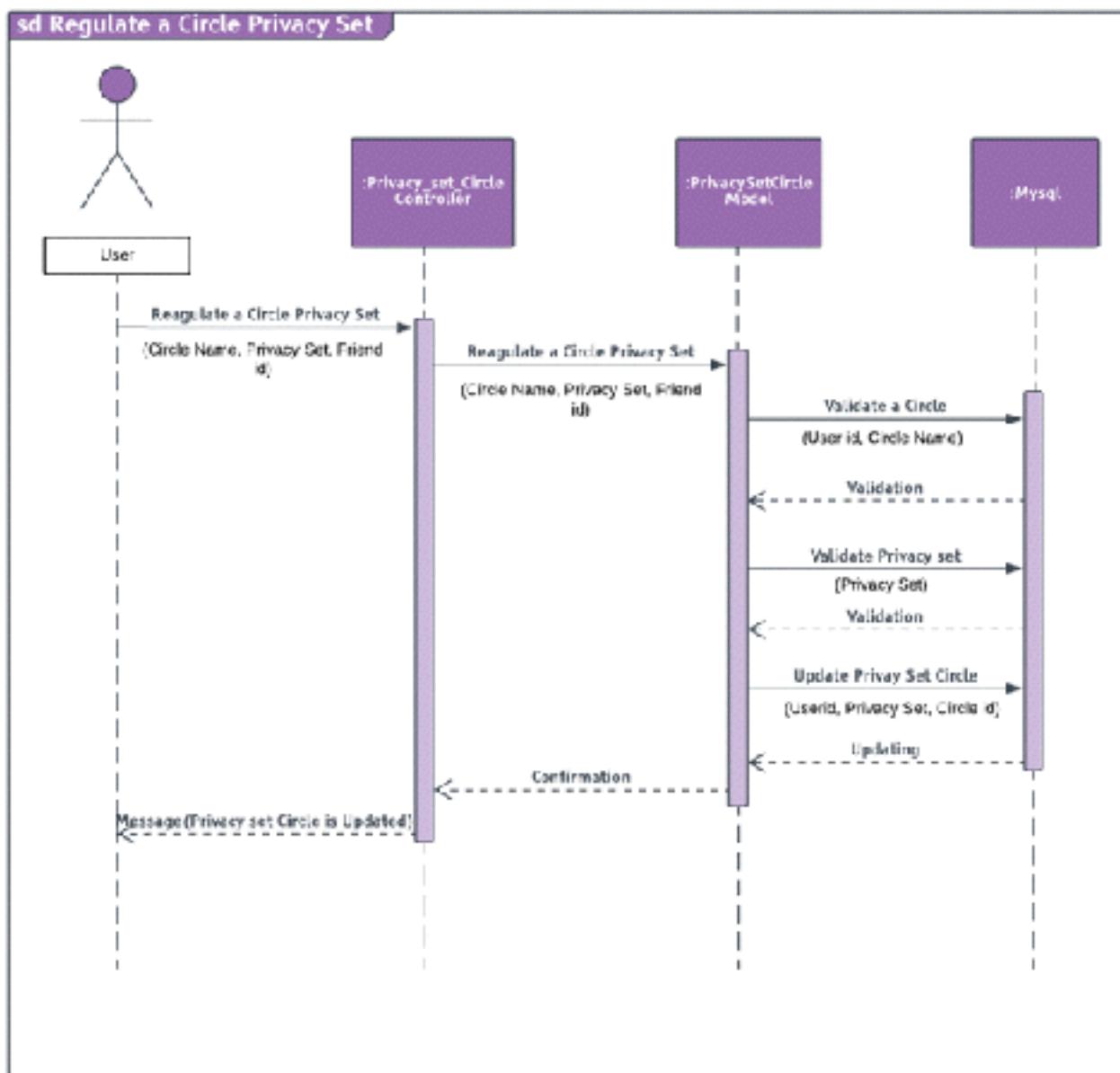
- Exclude a friend from a circle



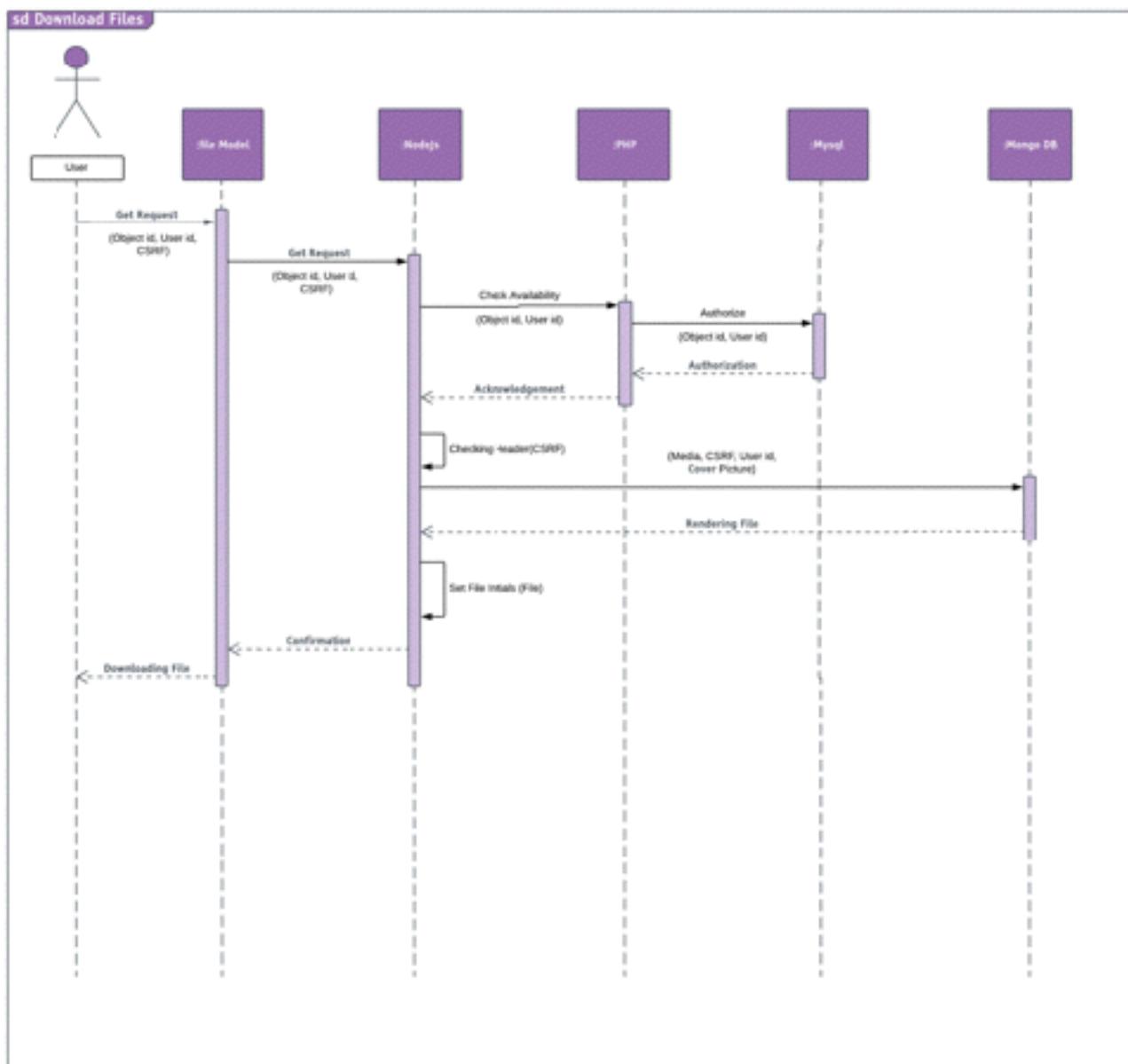
- Regulate a privacy set for users



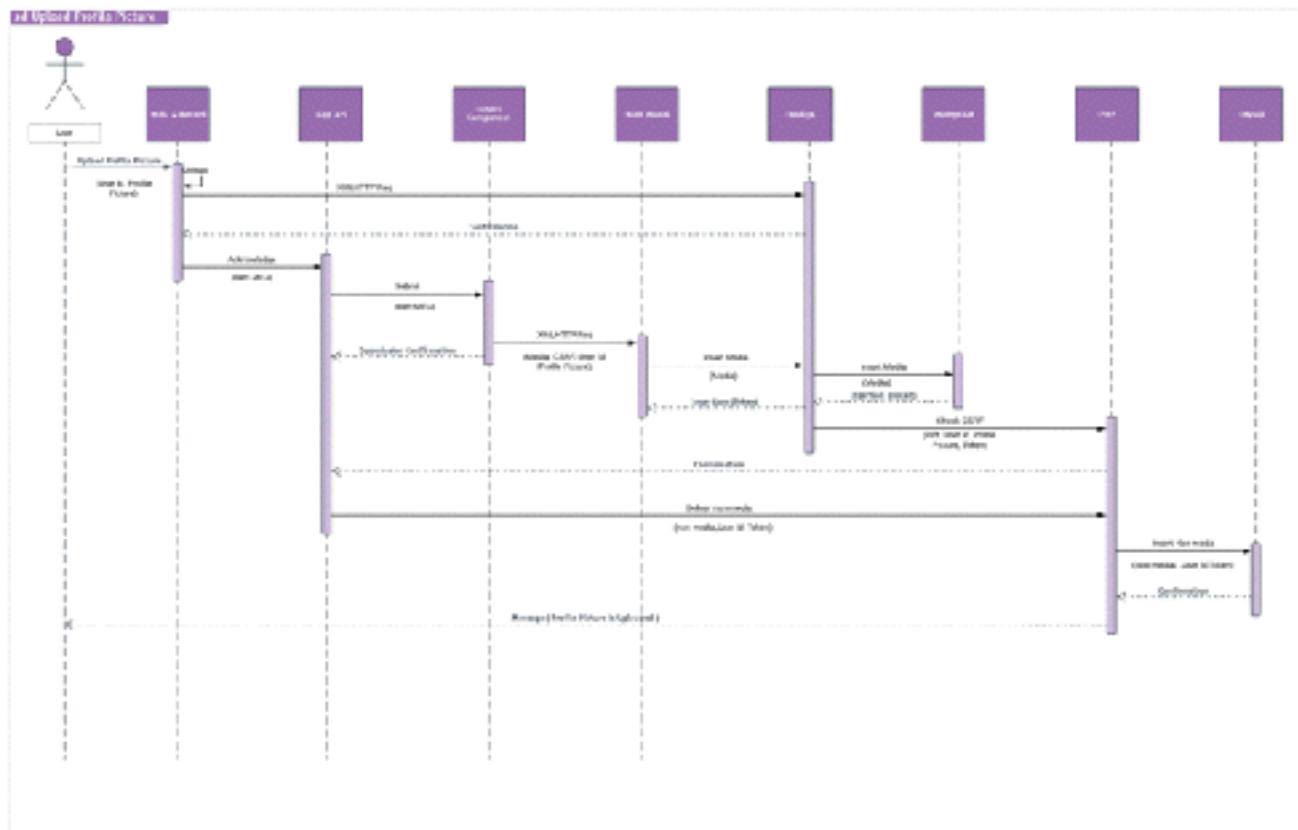
- Regulate a privacy set for circles



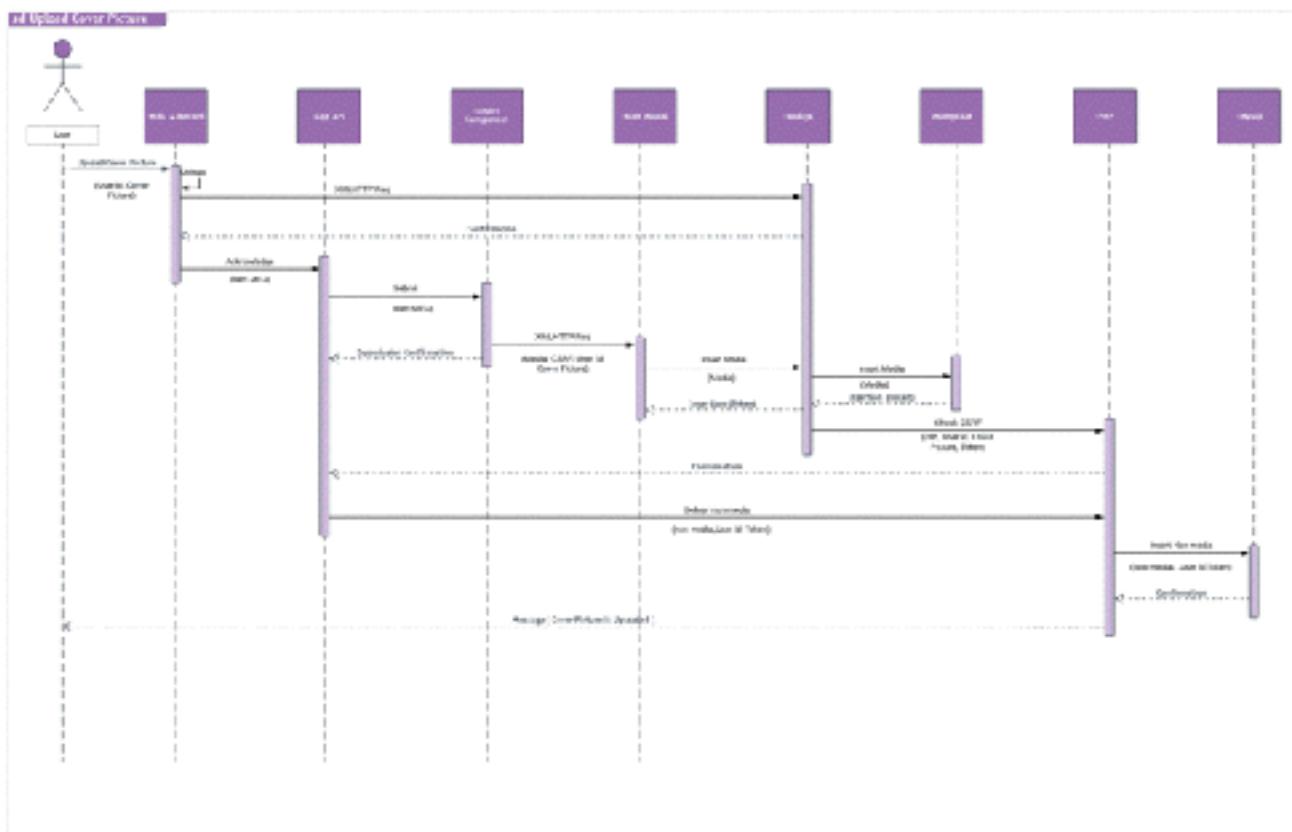
- Download files (with any extension).



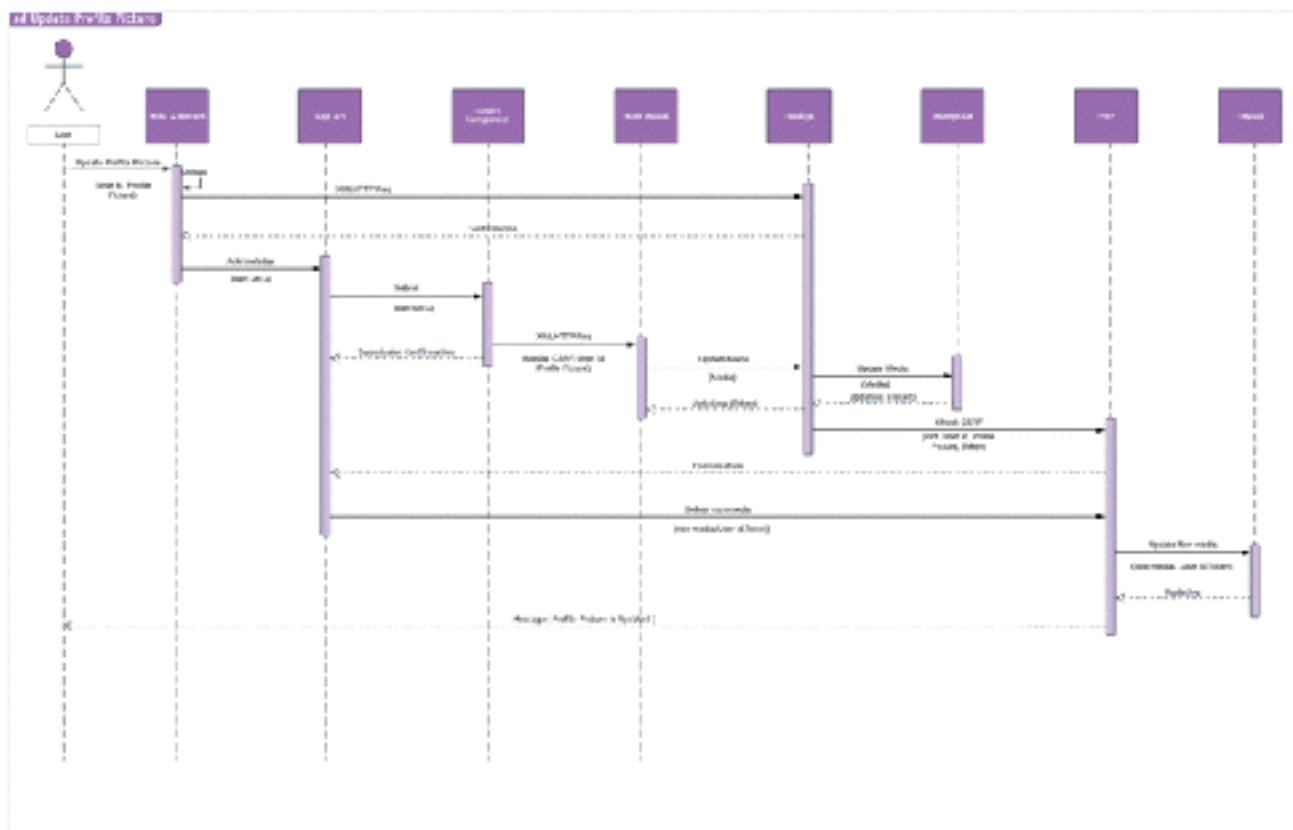
- Upload profile picture



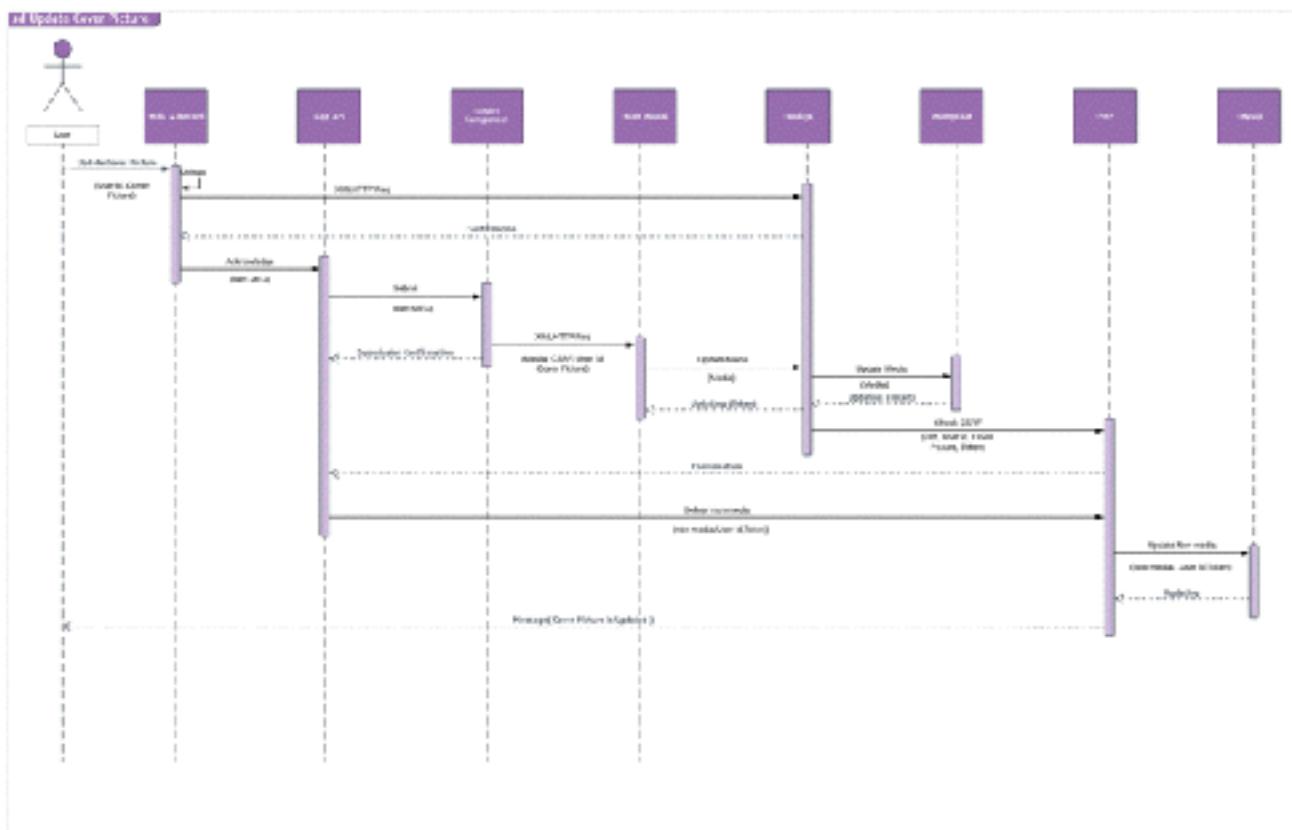
- Upload cover picture



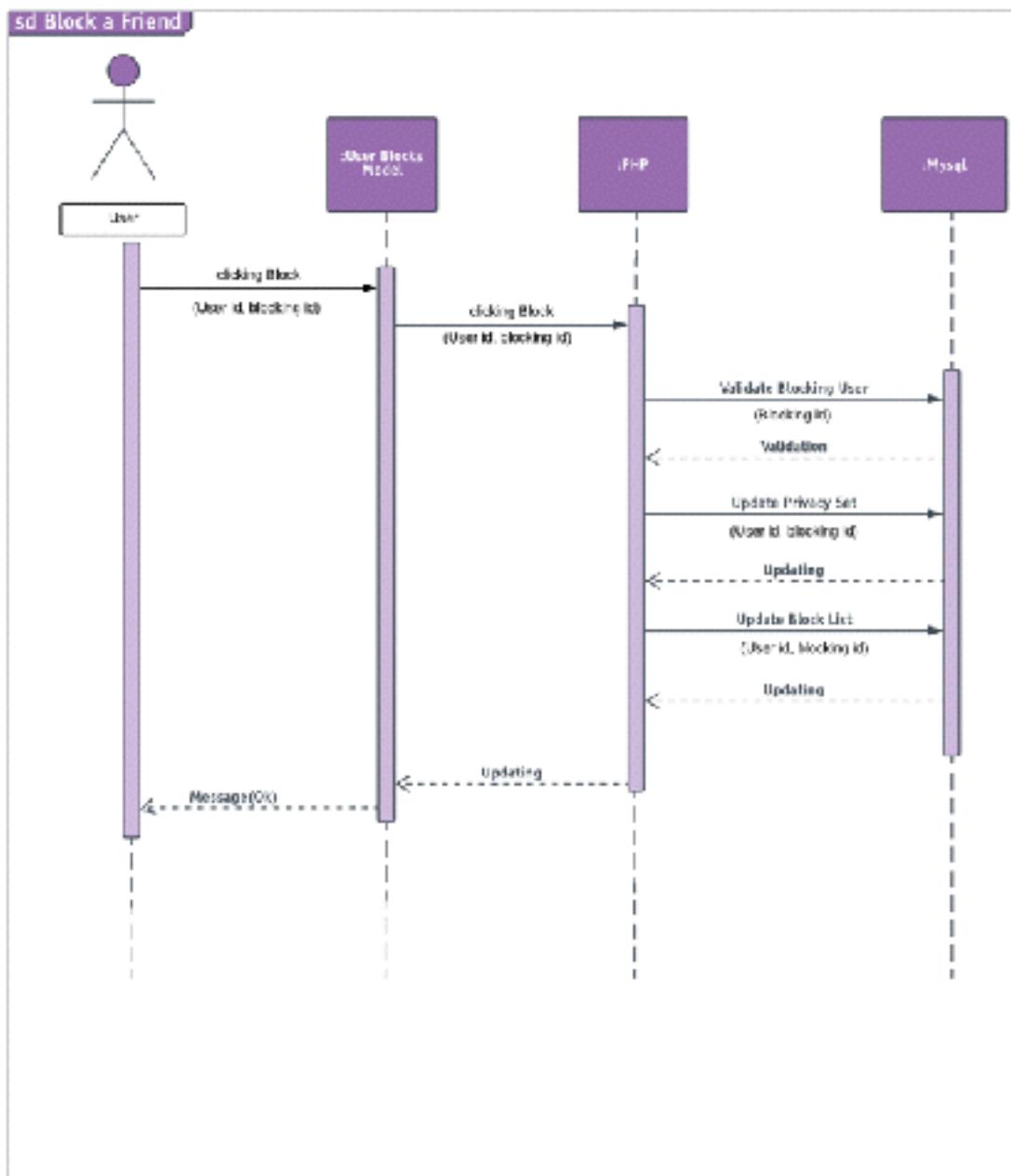
- Update profile picture



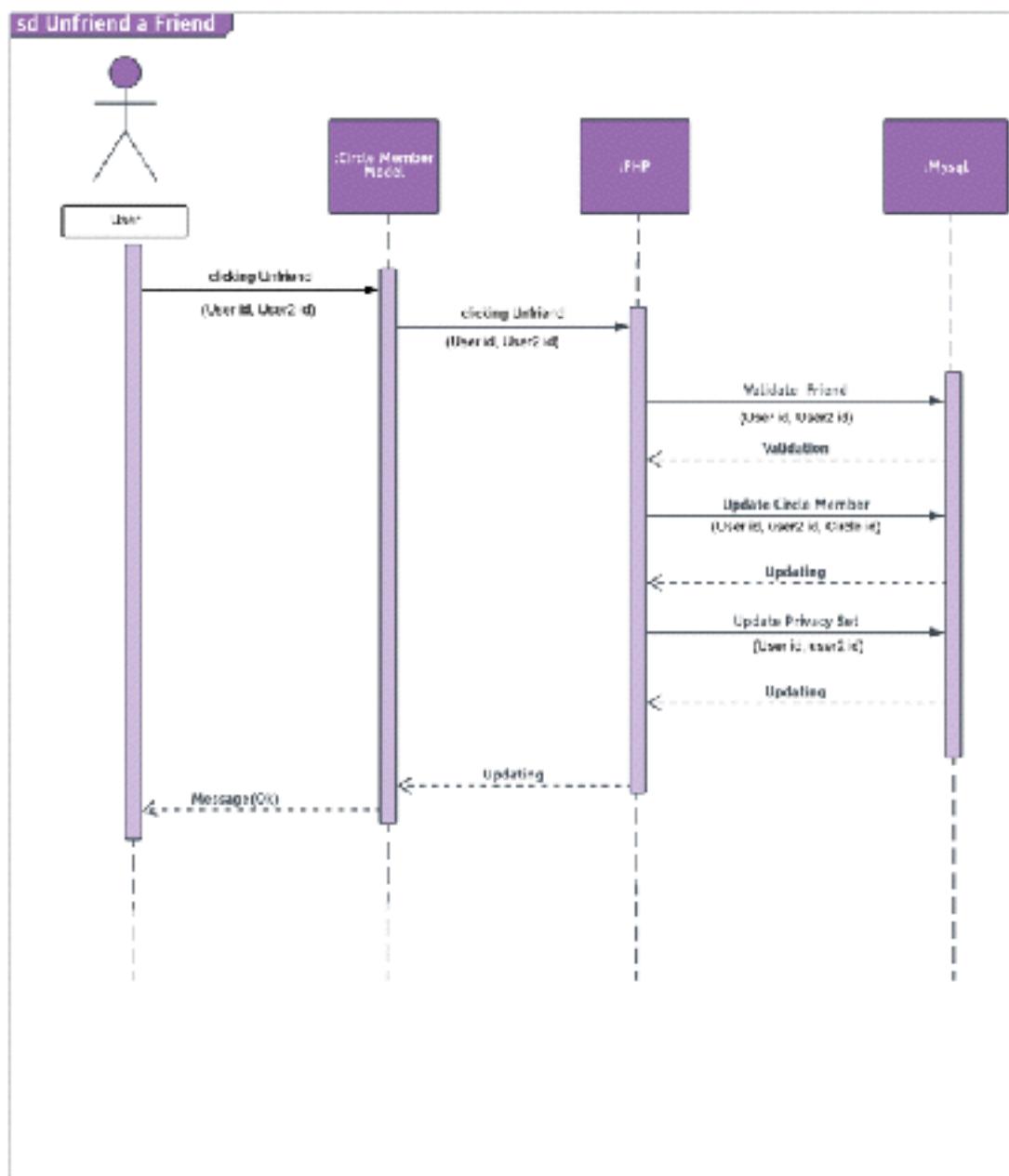
- Update cover picture



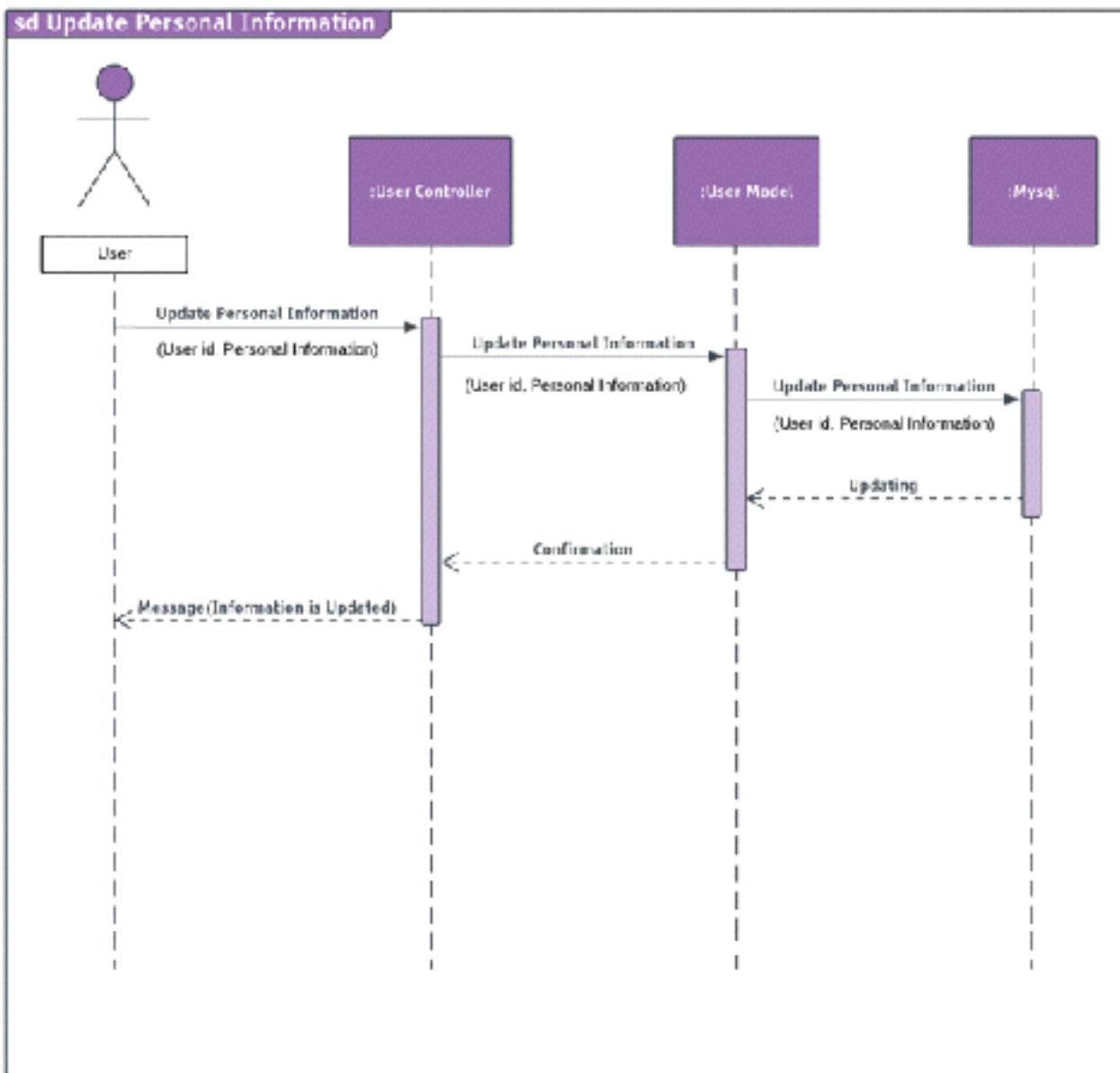
- Block a friend



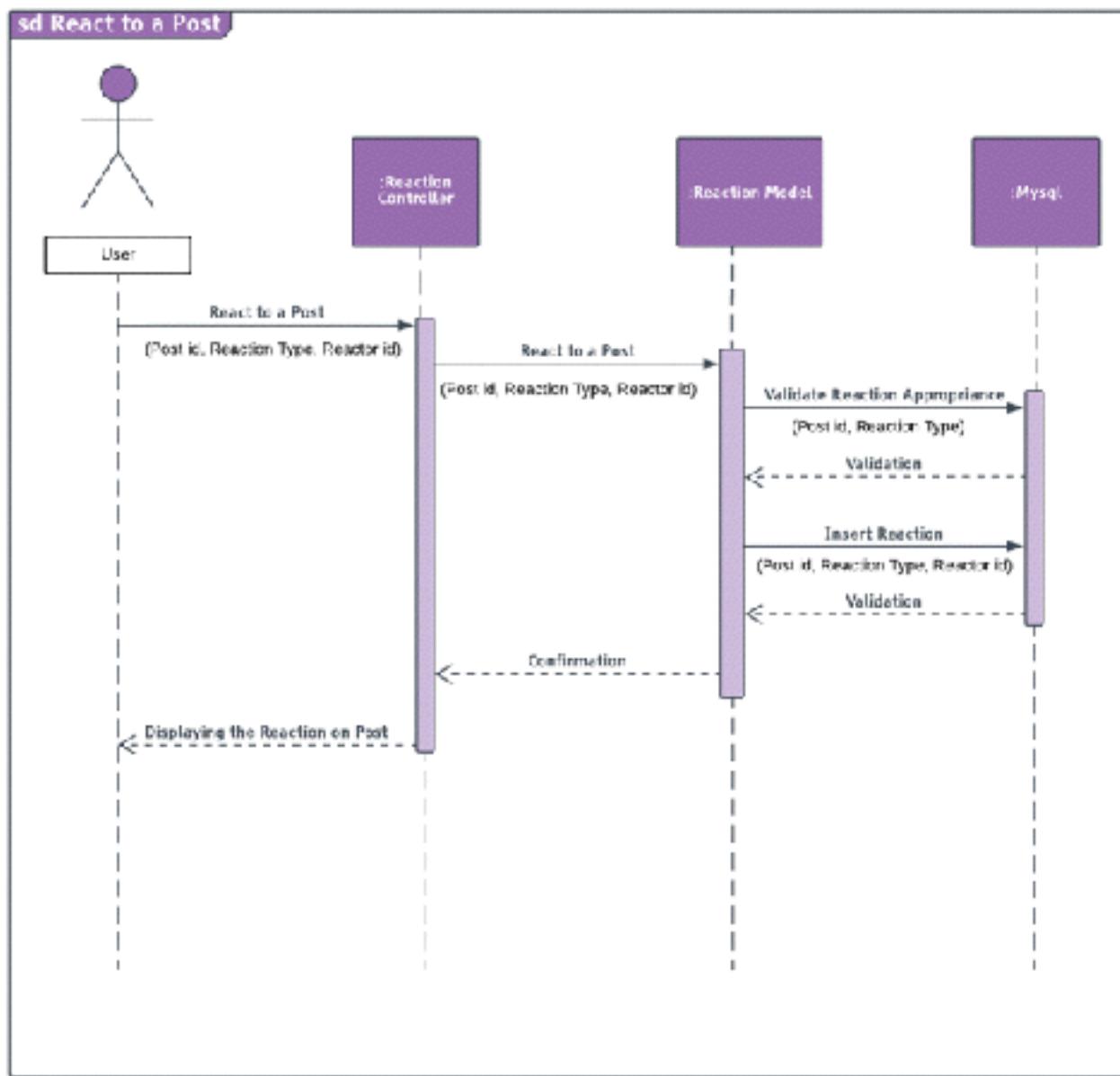
- Unfriend a friend



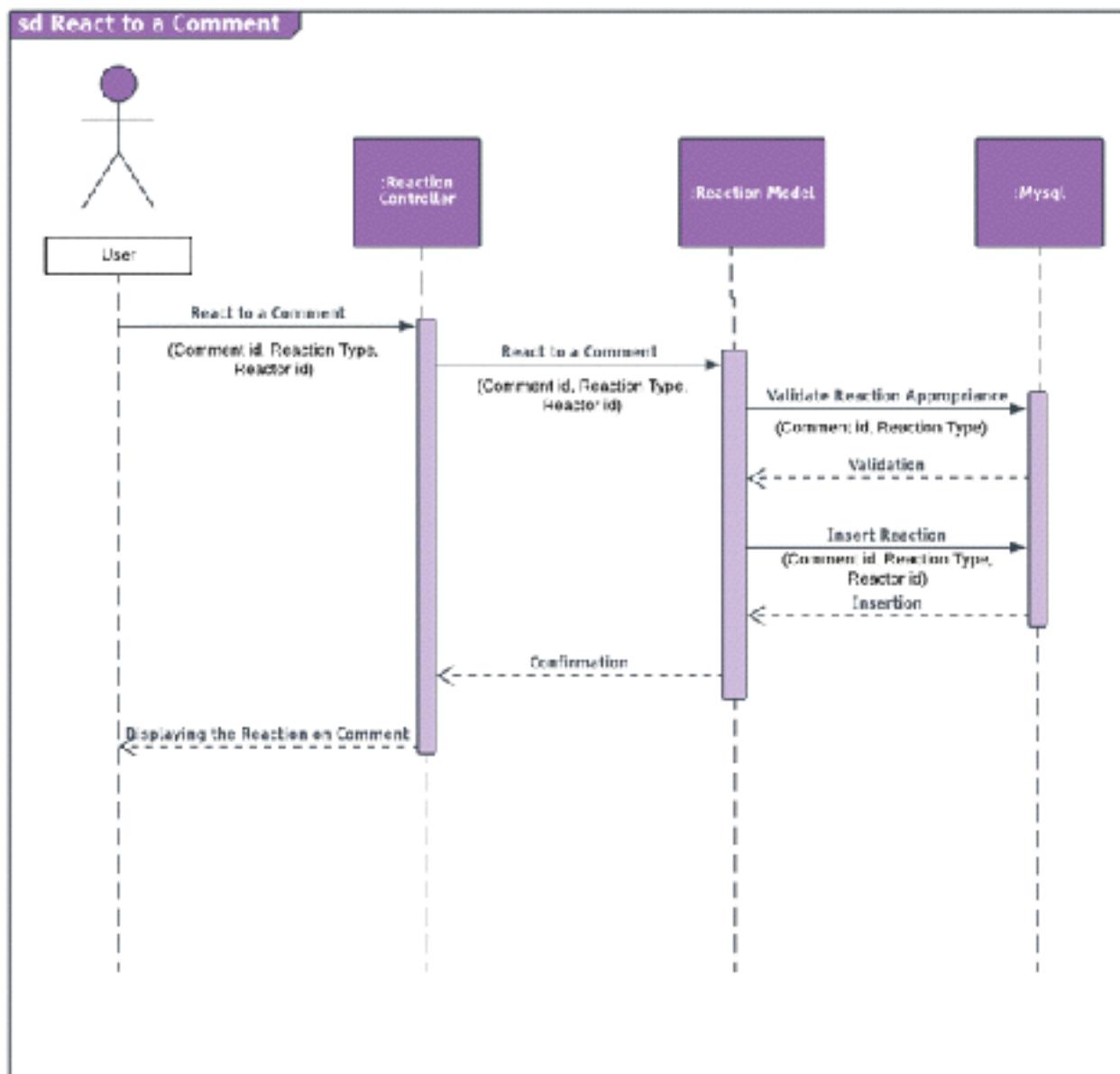
- Update personal information



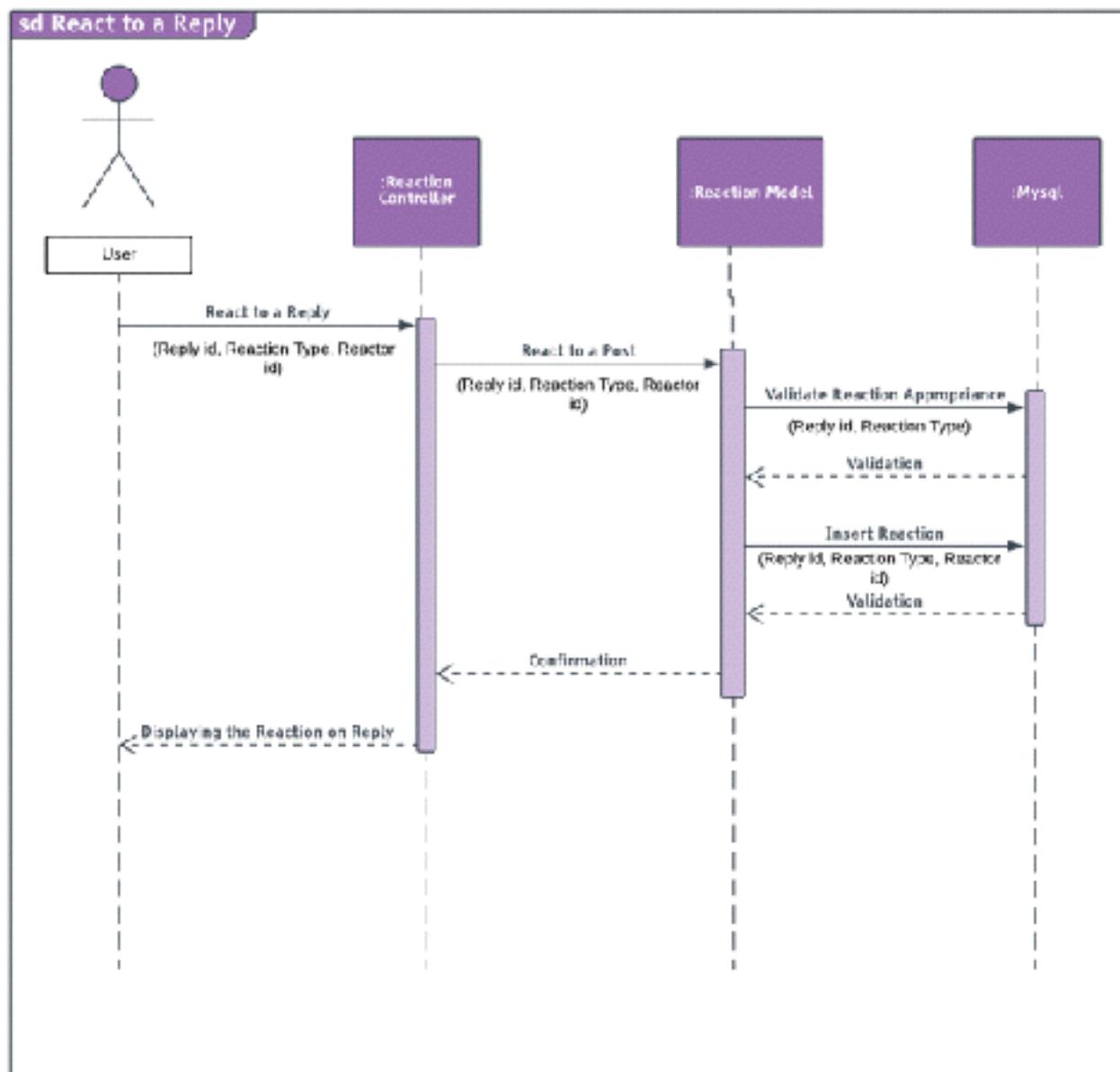
- React to post- when allowed to do so with allowed reactions.



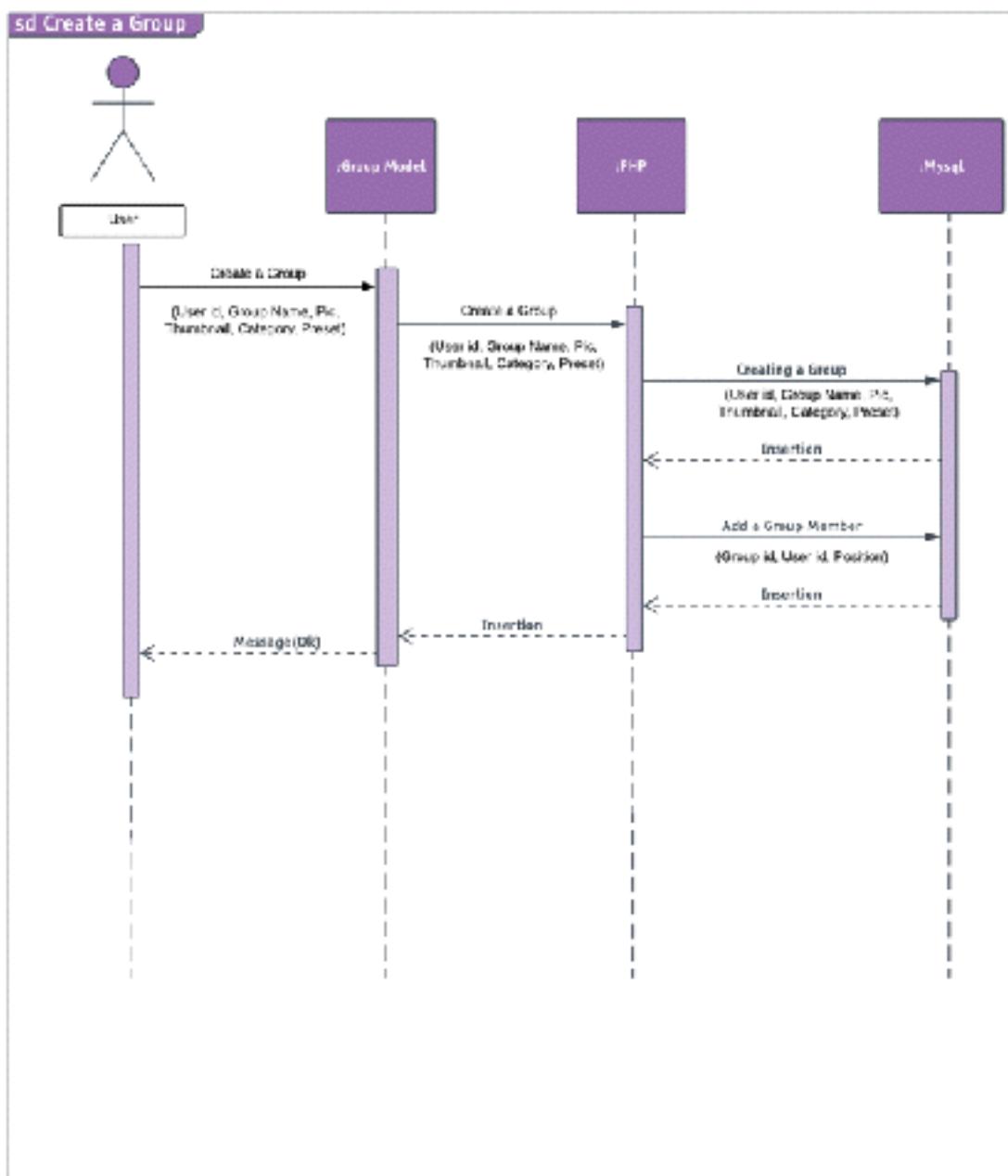
- React to comment- when allowed to do so with allowed reactions.



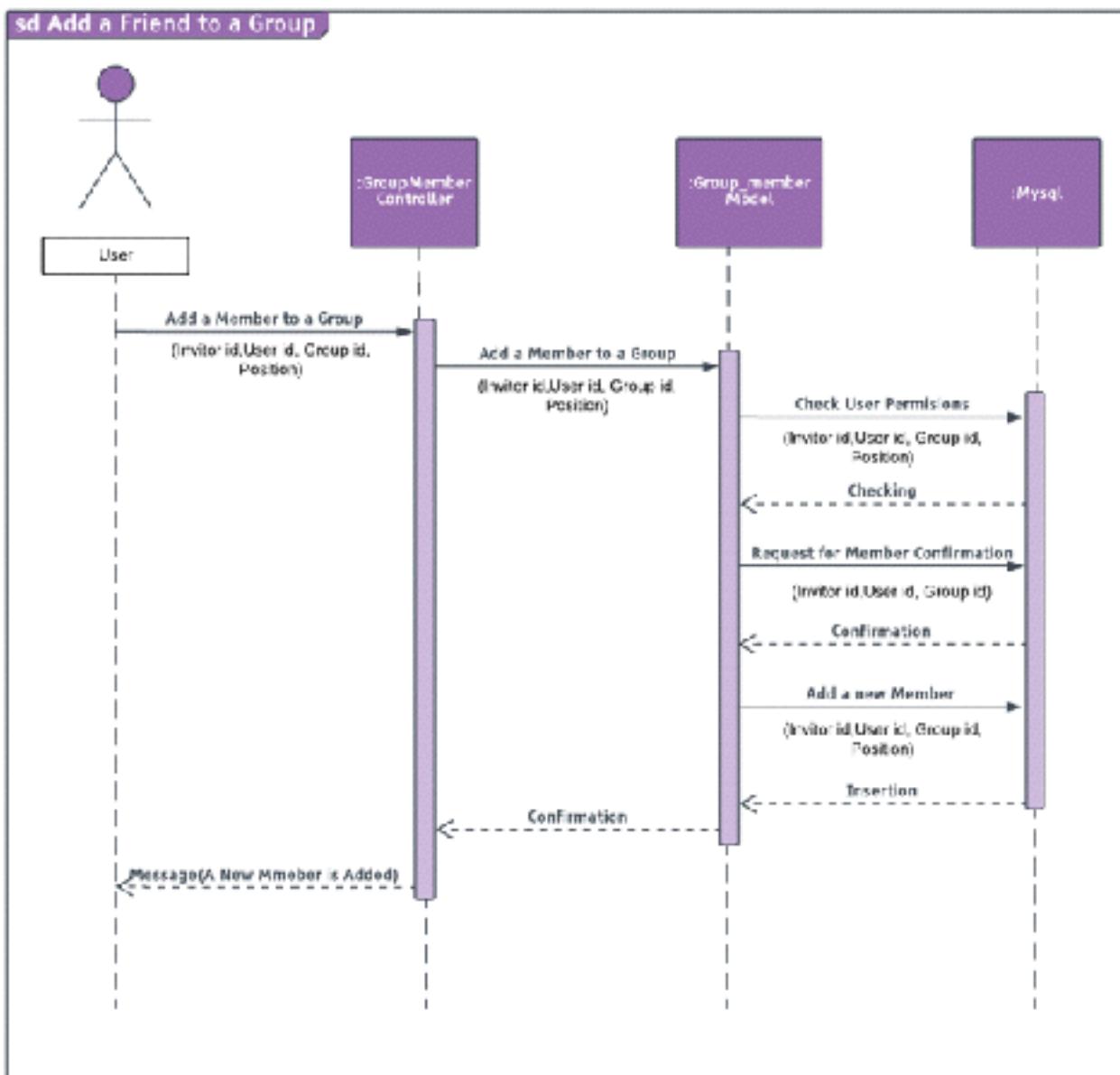
- React to reply- when allowed to do so with allowed reactions.



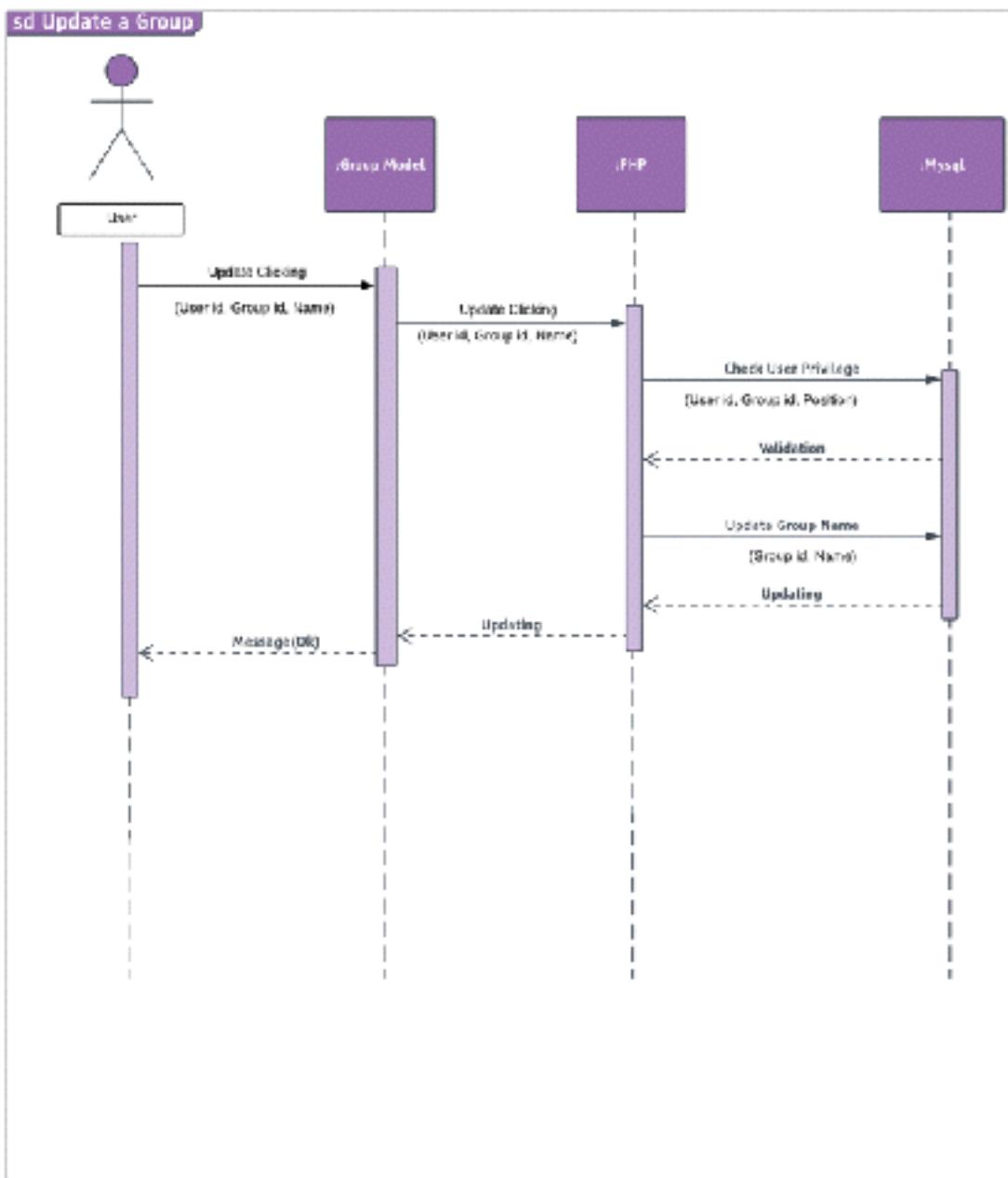
- Create a group



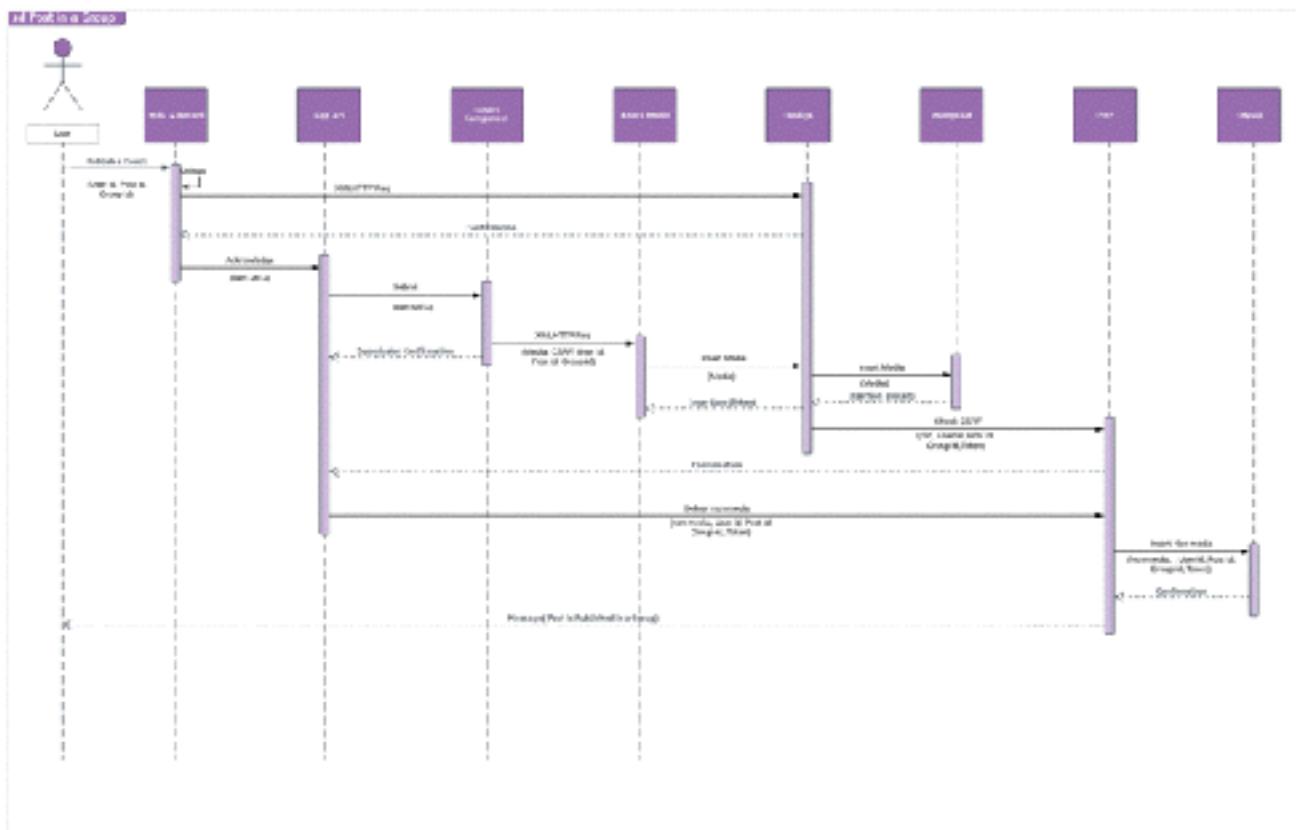
- Add friends to group



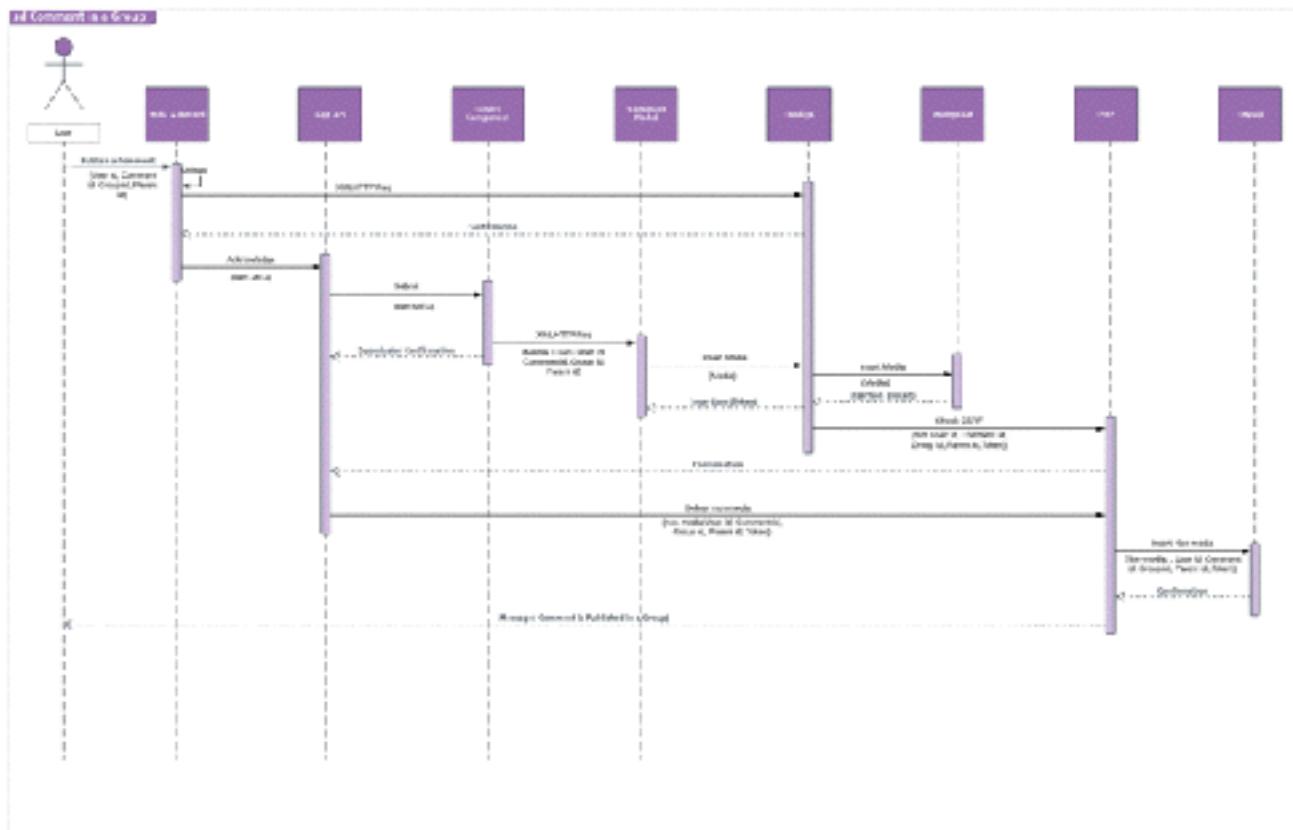
- Update a group- when allowed to do so.



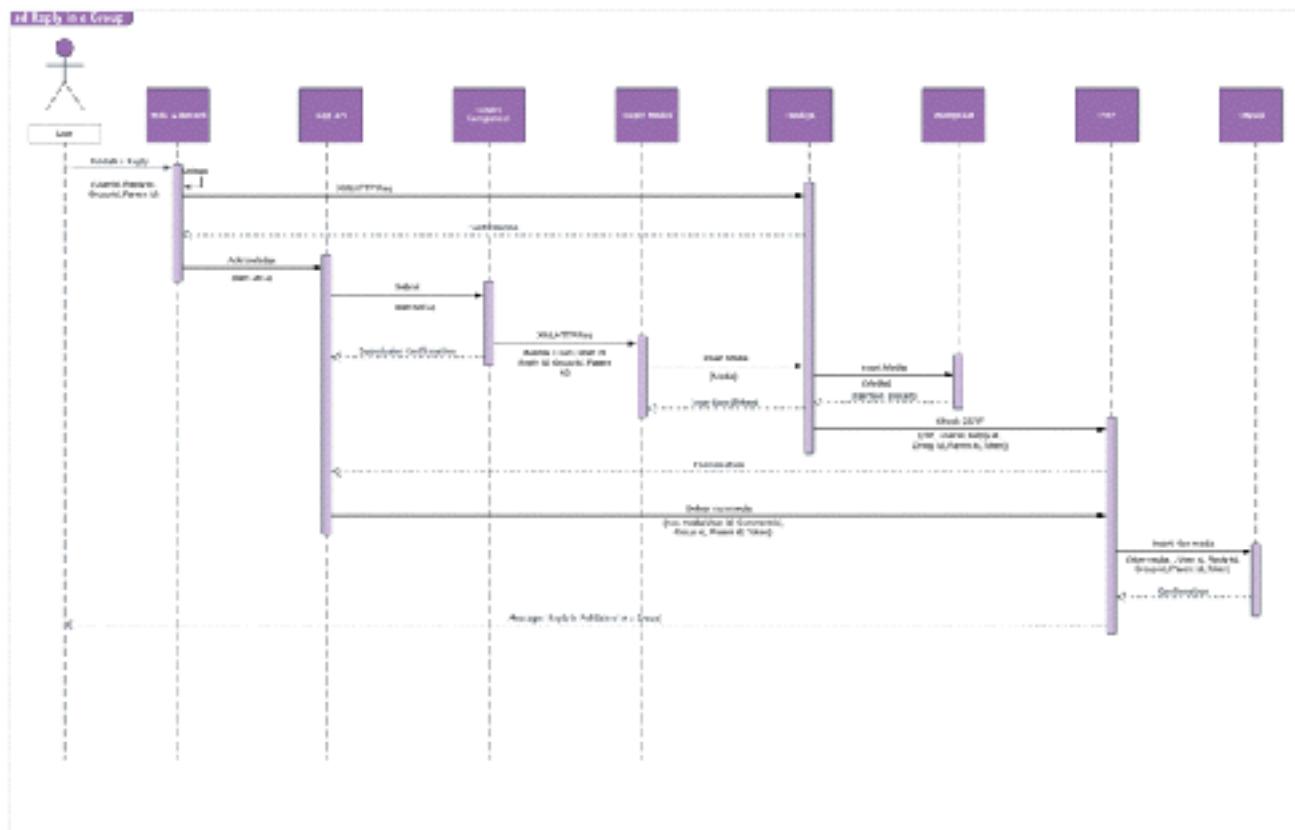
- Post in a group- when allowed to do so.



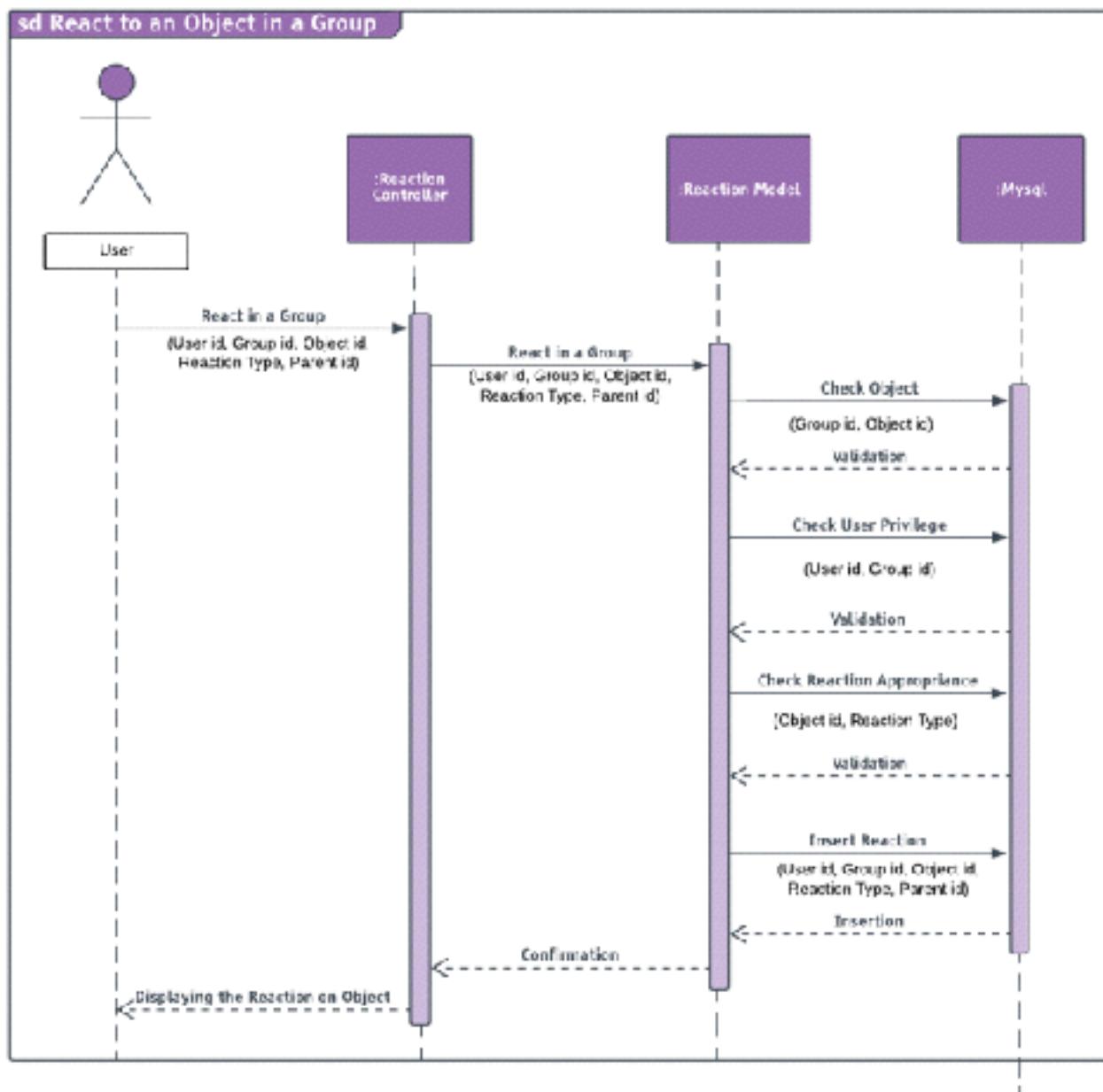
- Comment in a group- when allowed to do so.



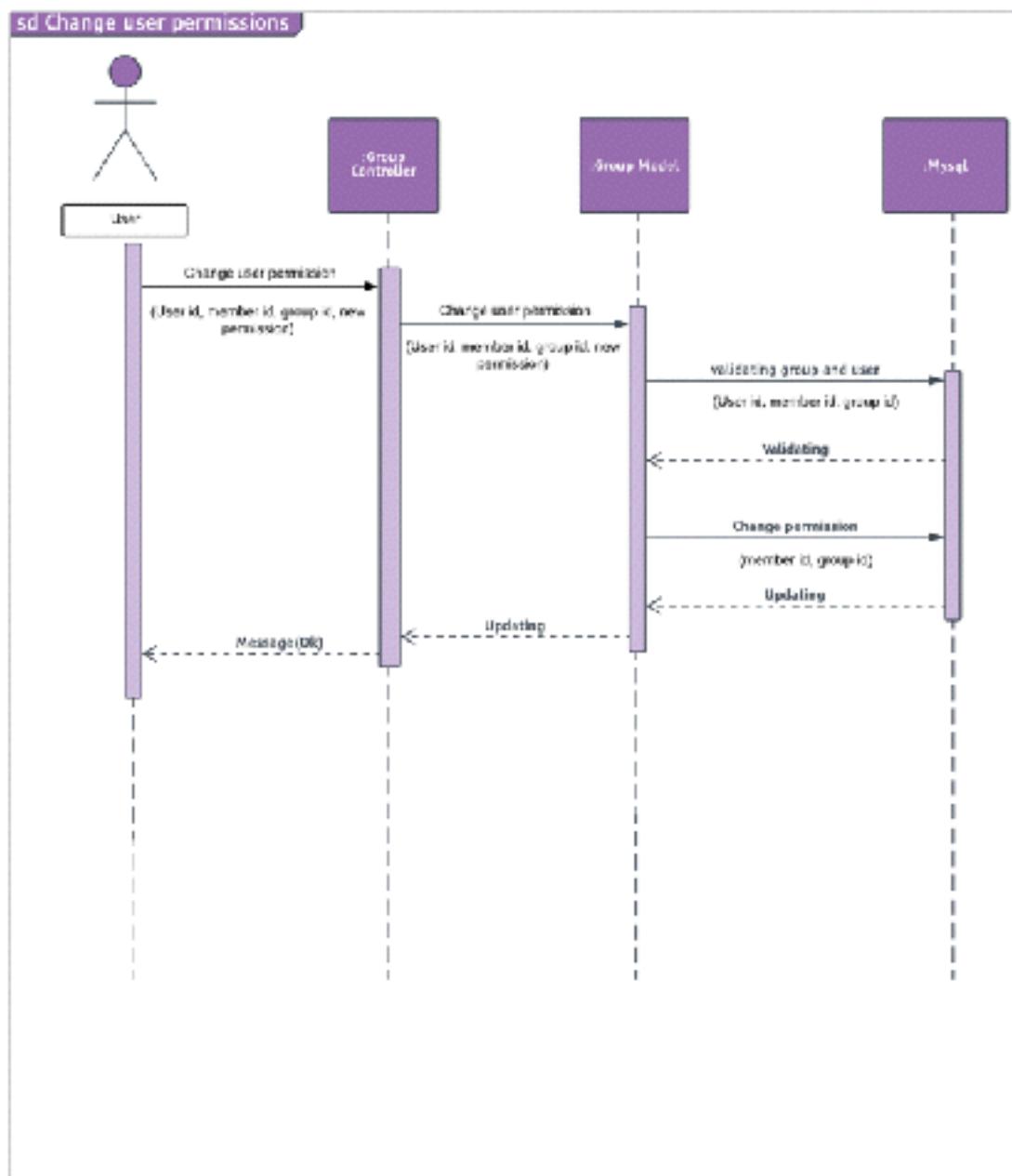
- Reply in a group- when allowed to do so.



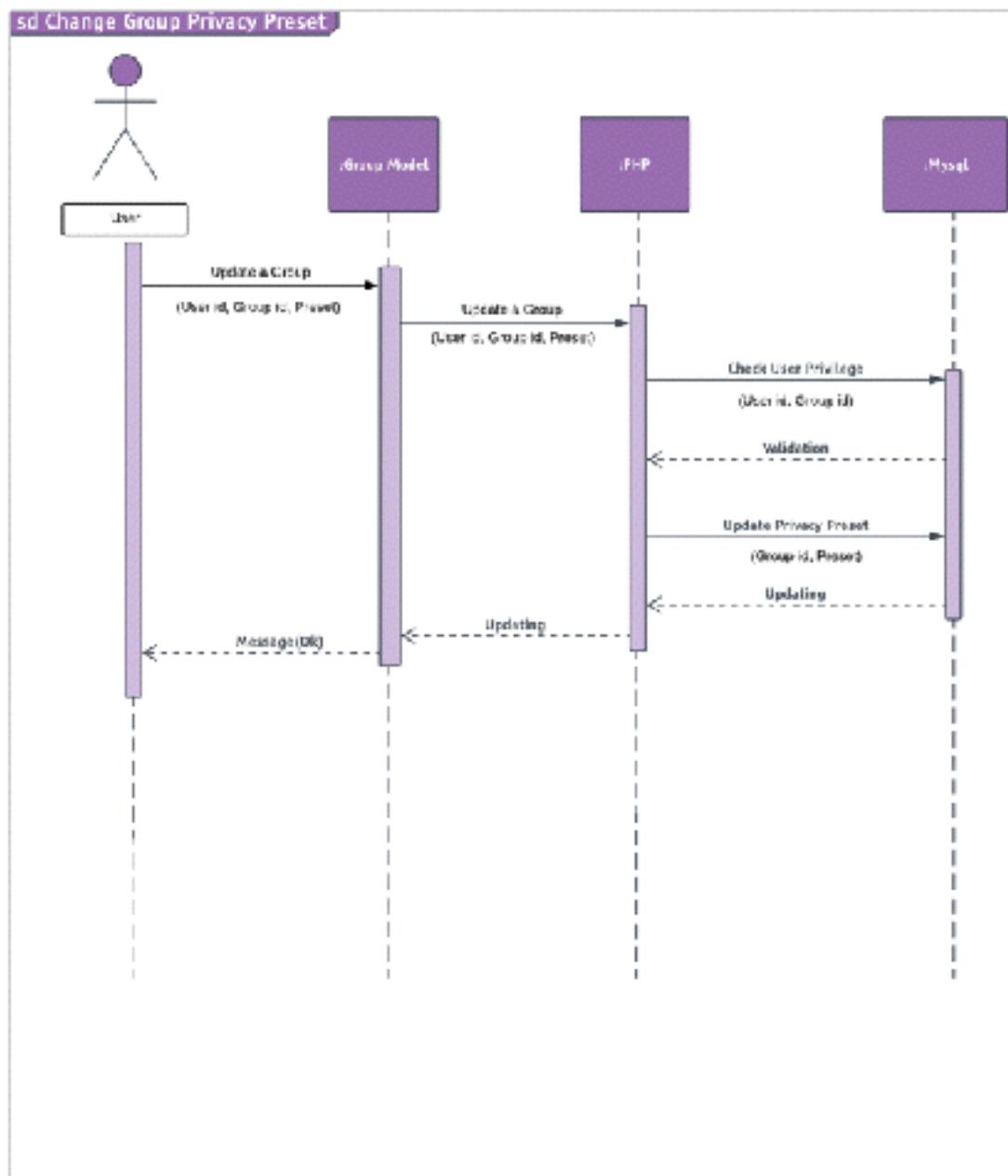
- React to an object a group- when allowed to do so.



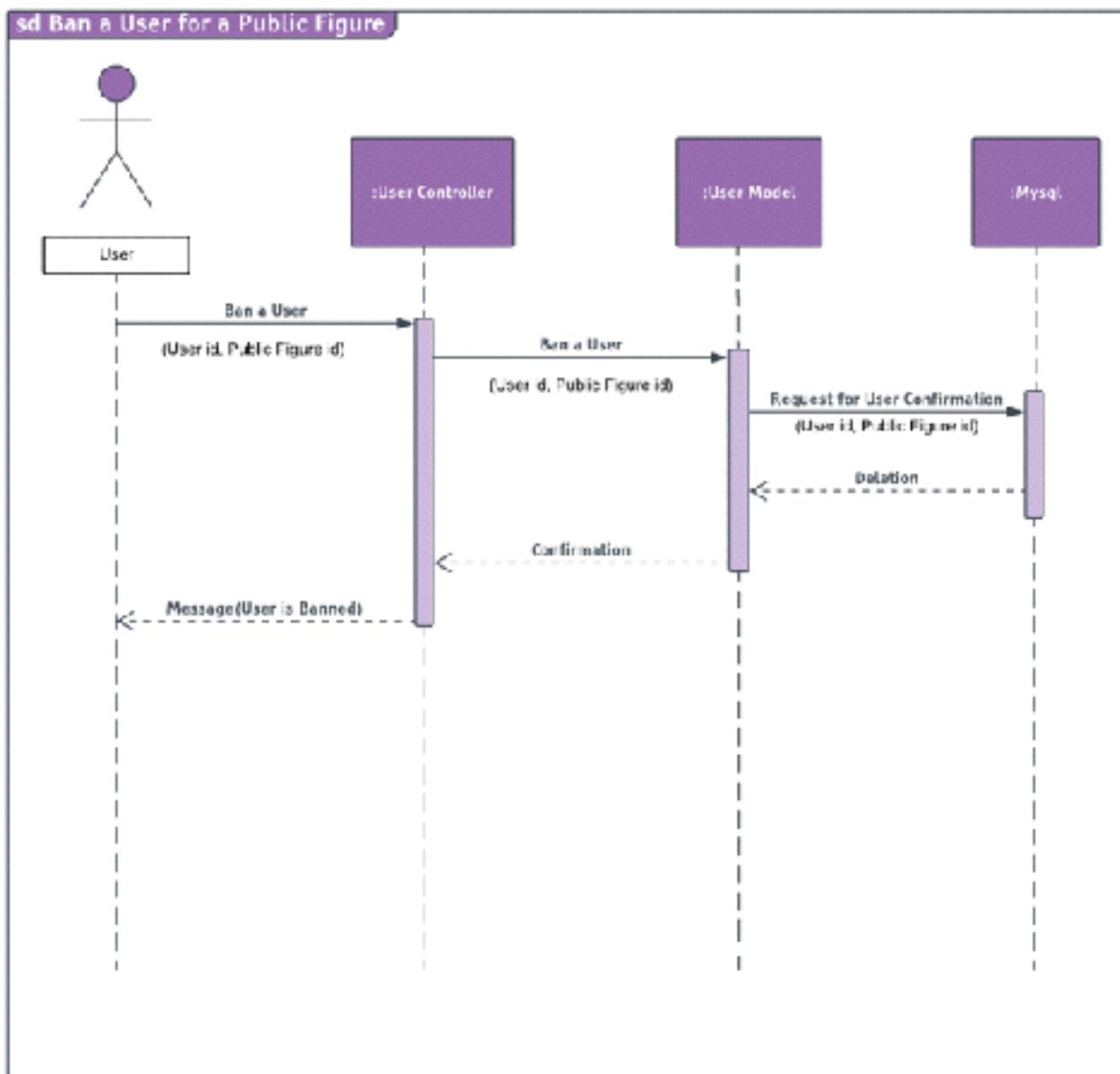
- Change user permission- when allowed to do so.



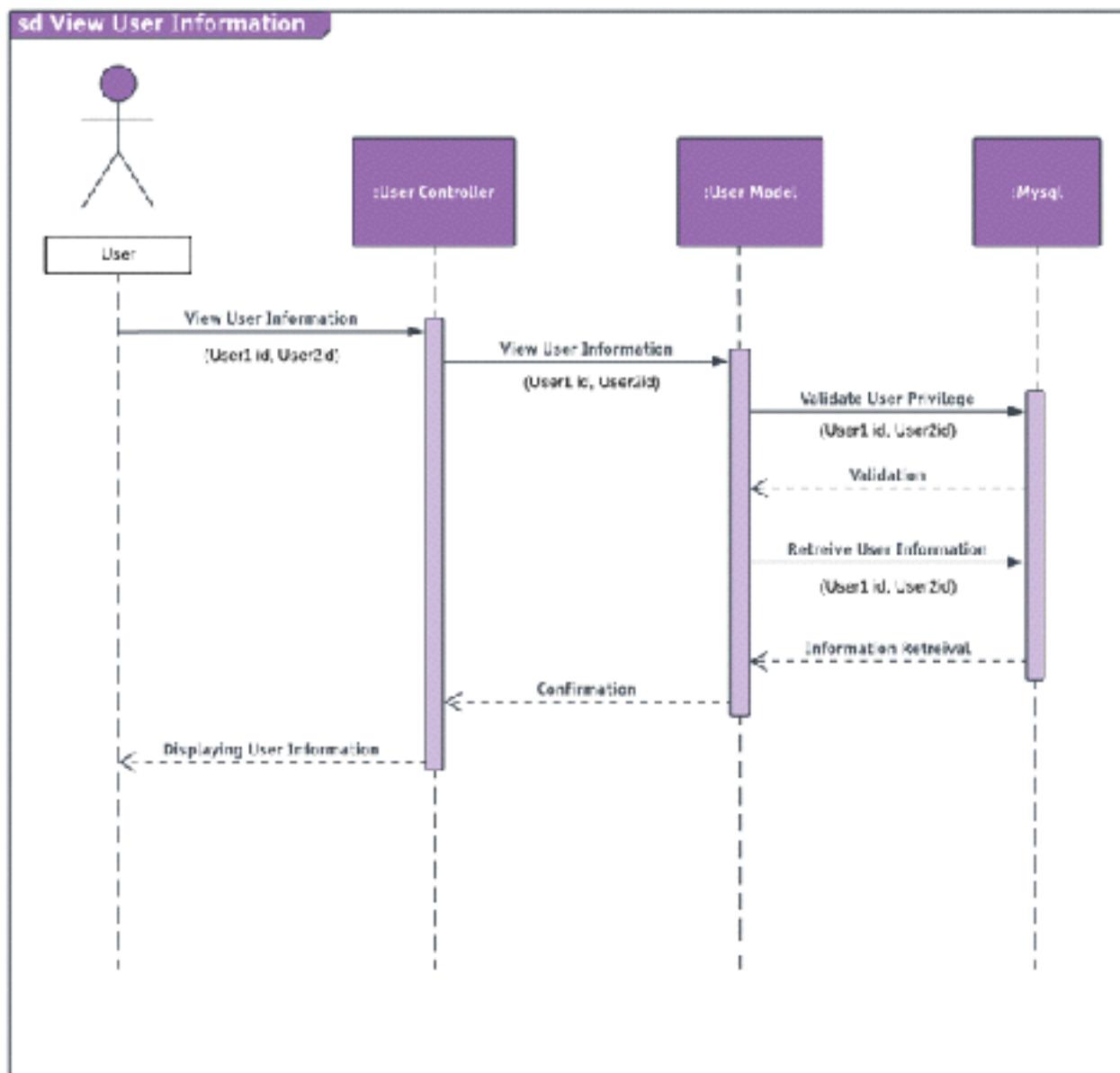
- Change the group privacy preset - when allowed to do so.



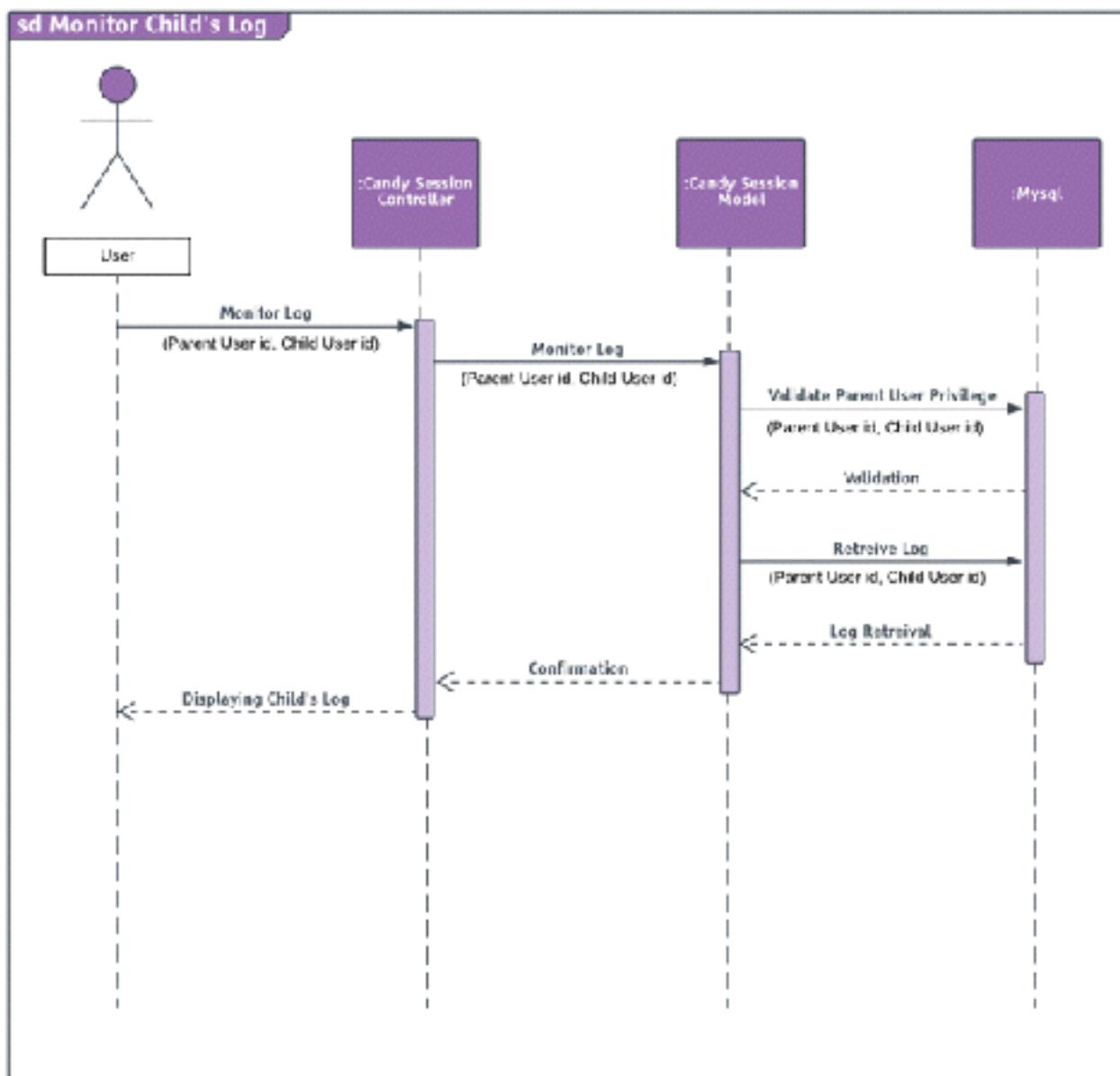
- Ban members of a group- when allowed to do so.



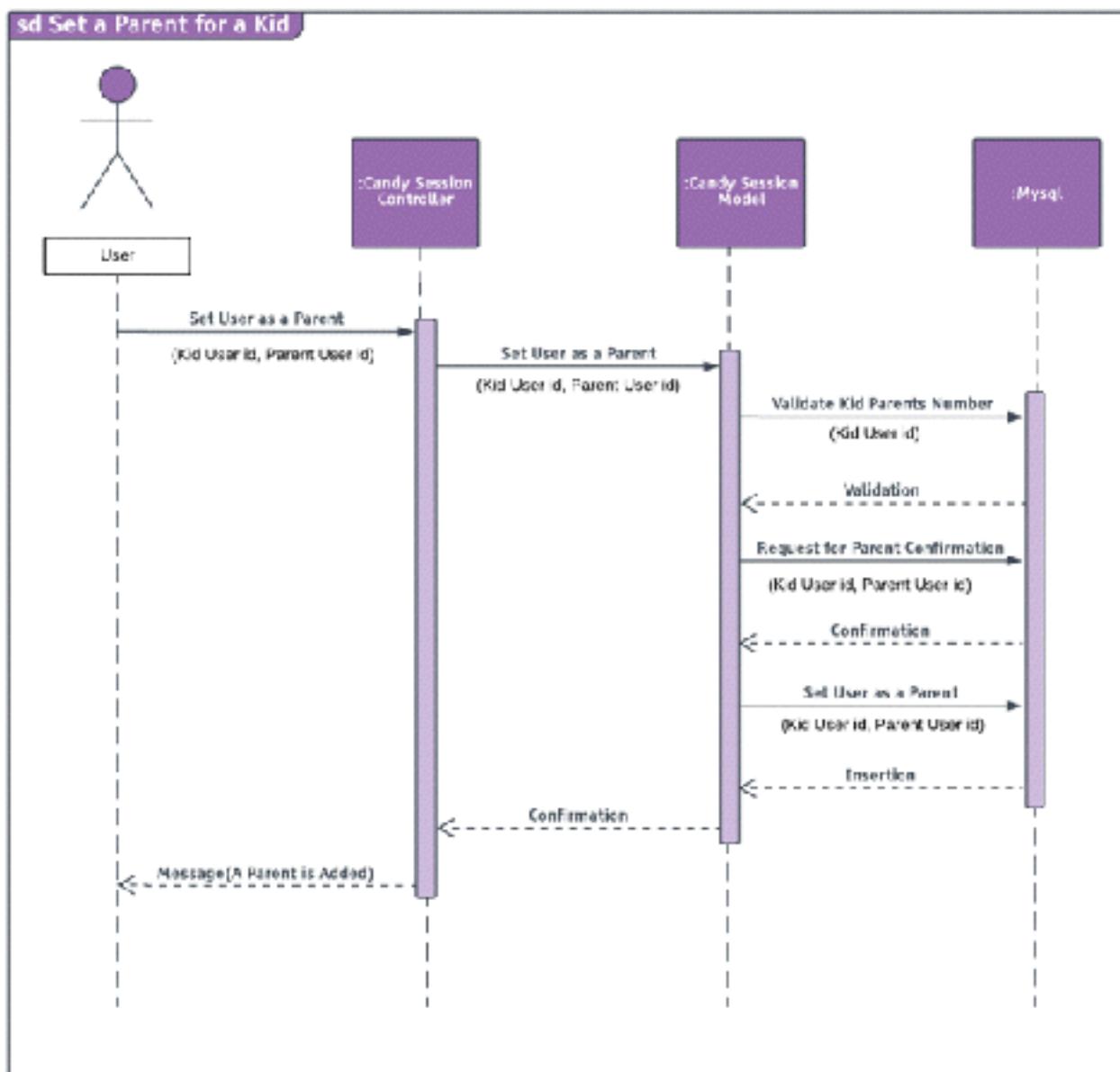
- View a user profile - when allowed to do so.



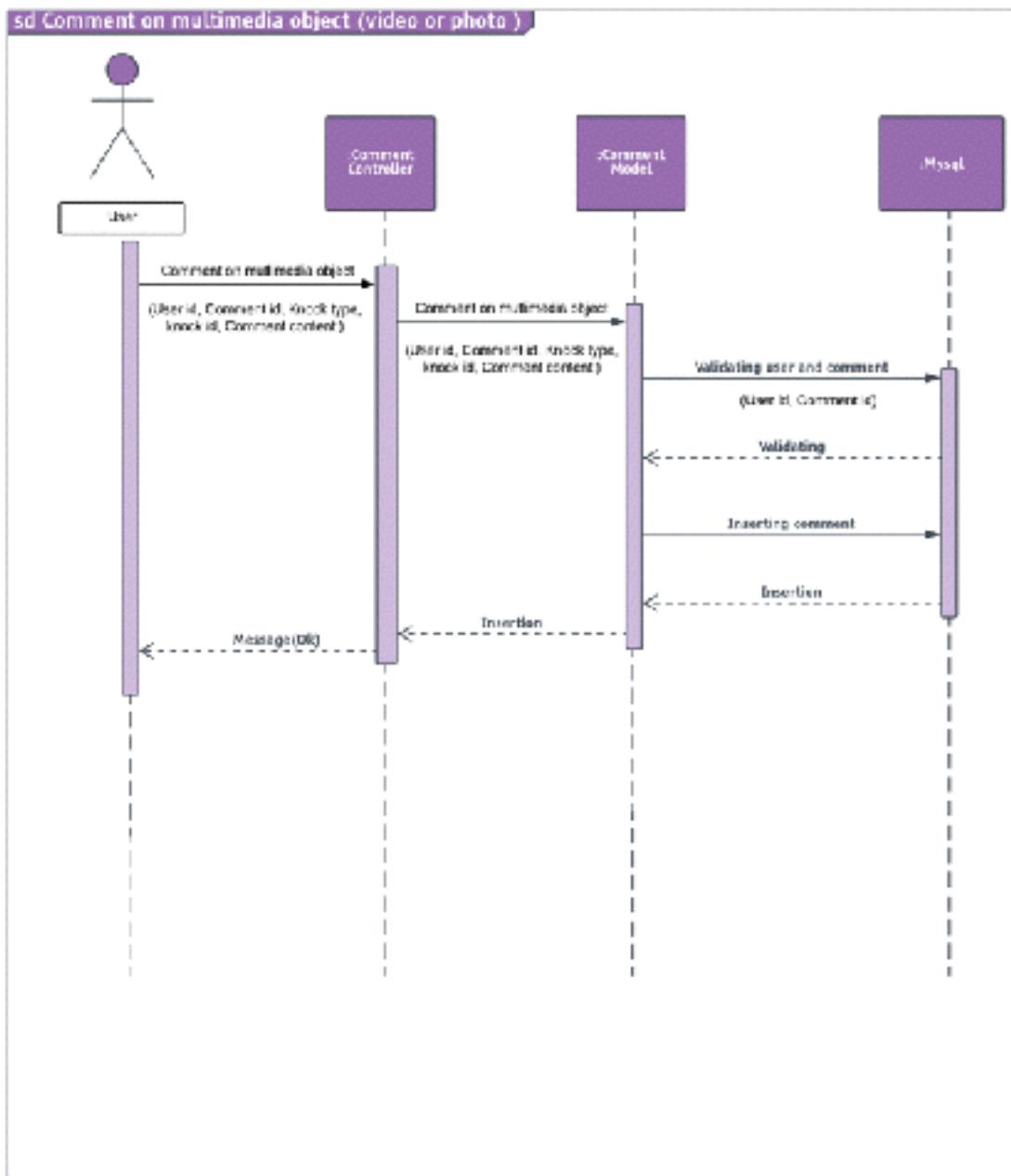
- Monitor for child log.



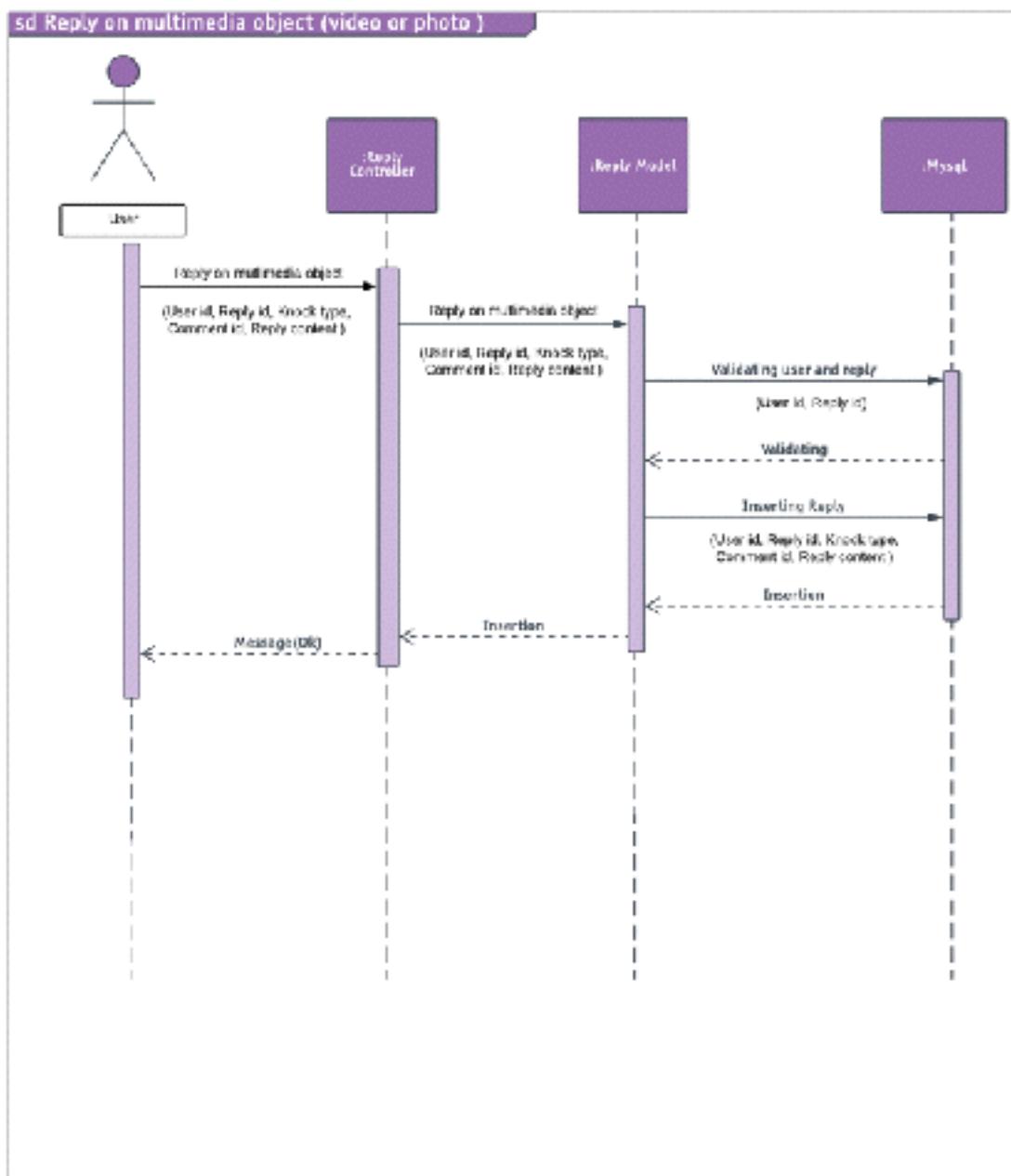
- Set a parent for a child



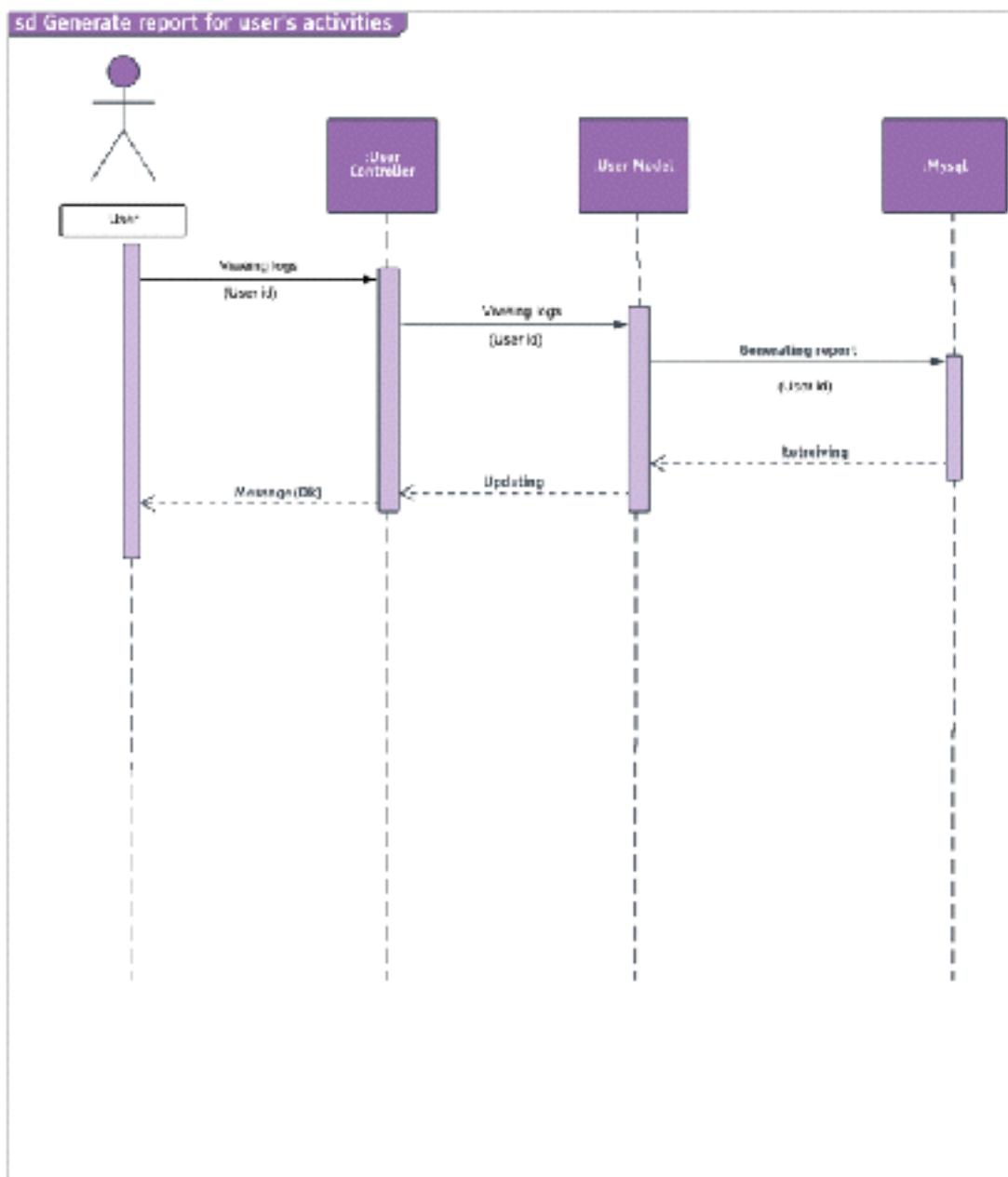
- Comment on multimedia object



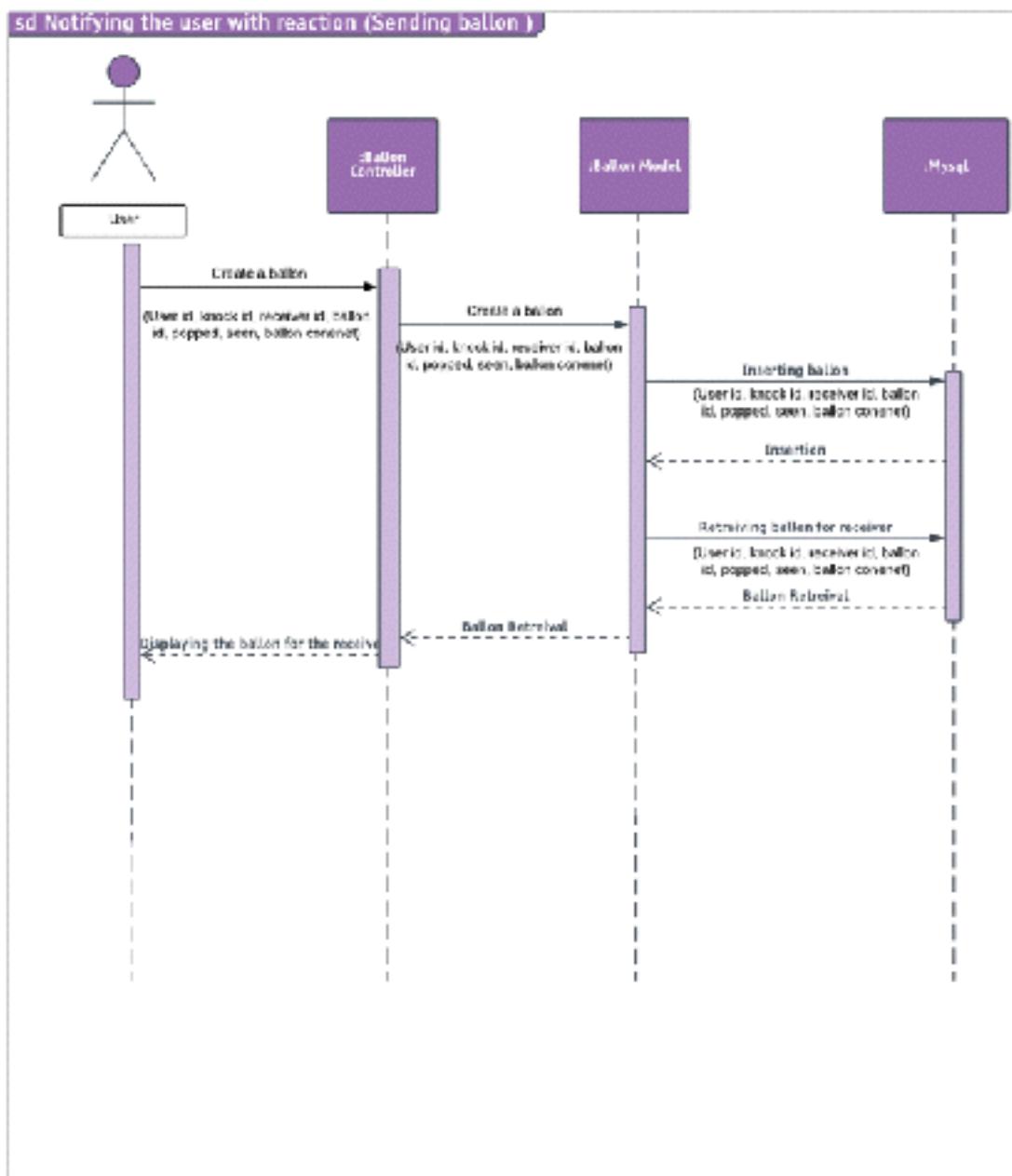
- Reply on multimedia object



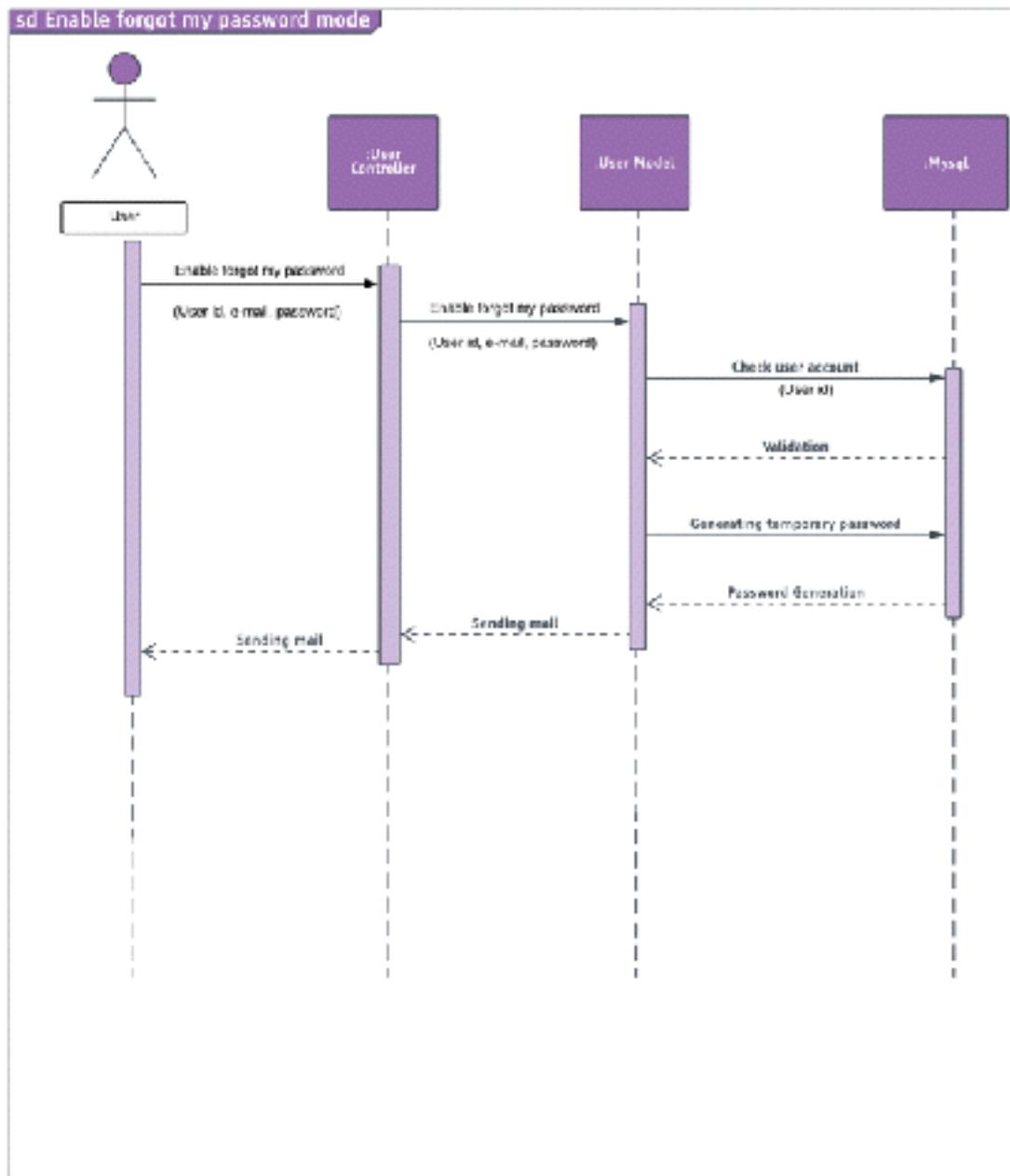
- Generate report for user's activities



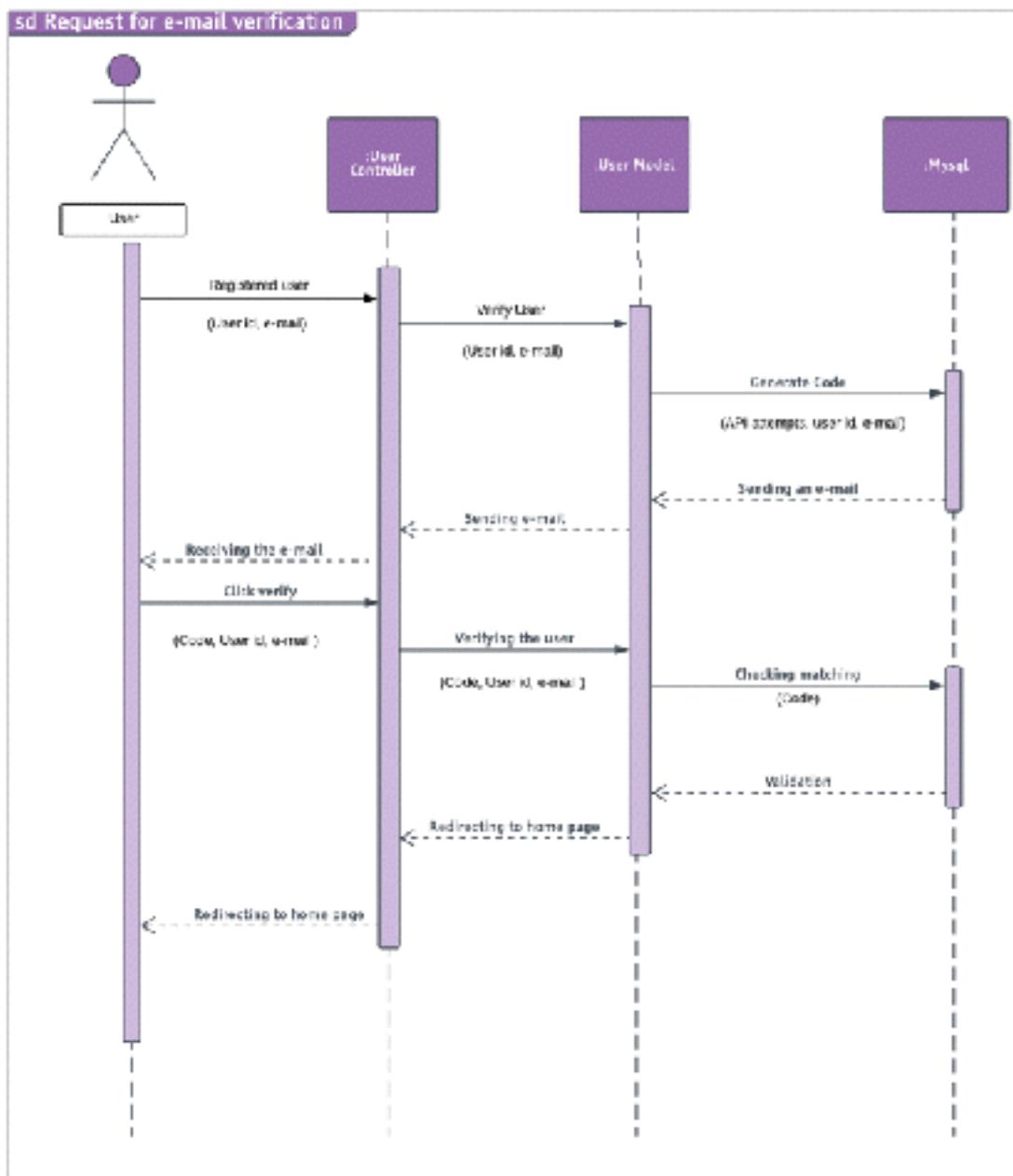
- Notifying the user ( Eg : Reaction )



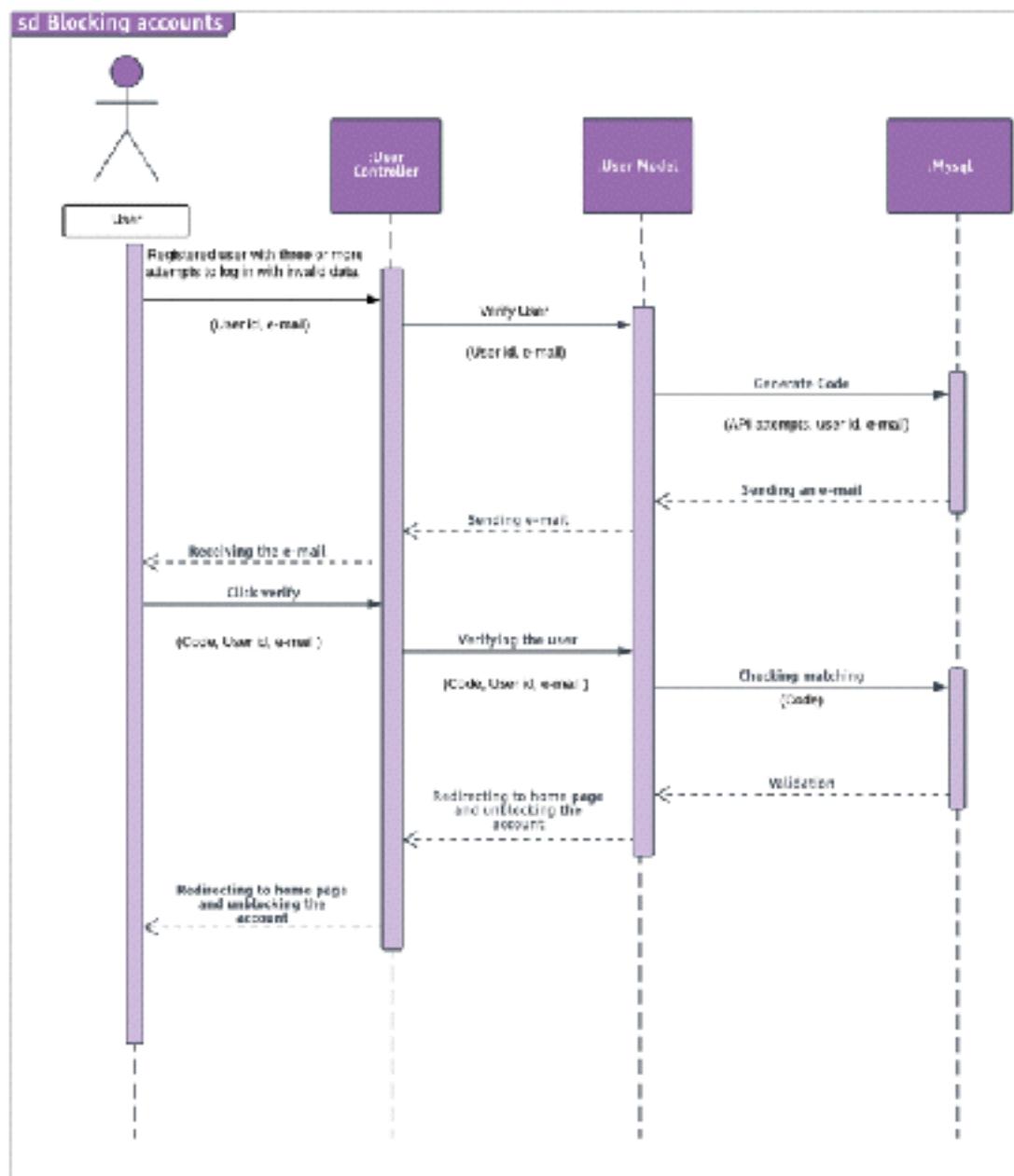
- Enable forgot my password mode



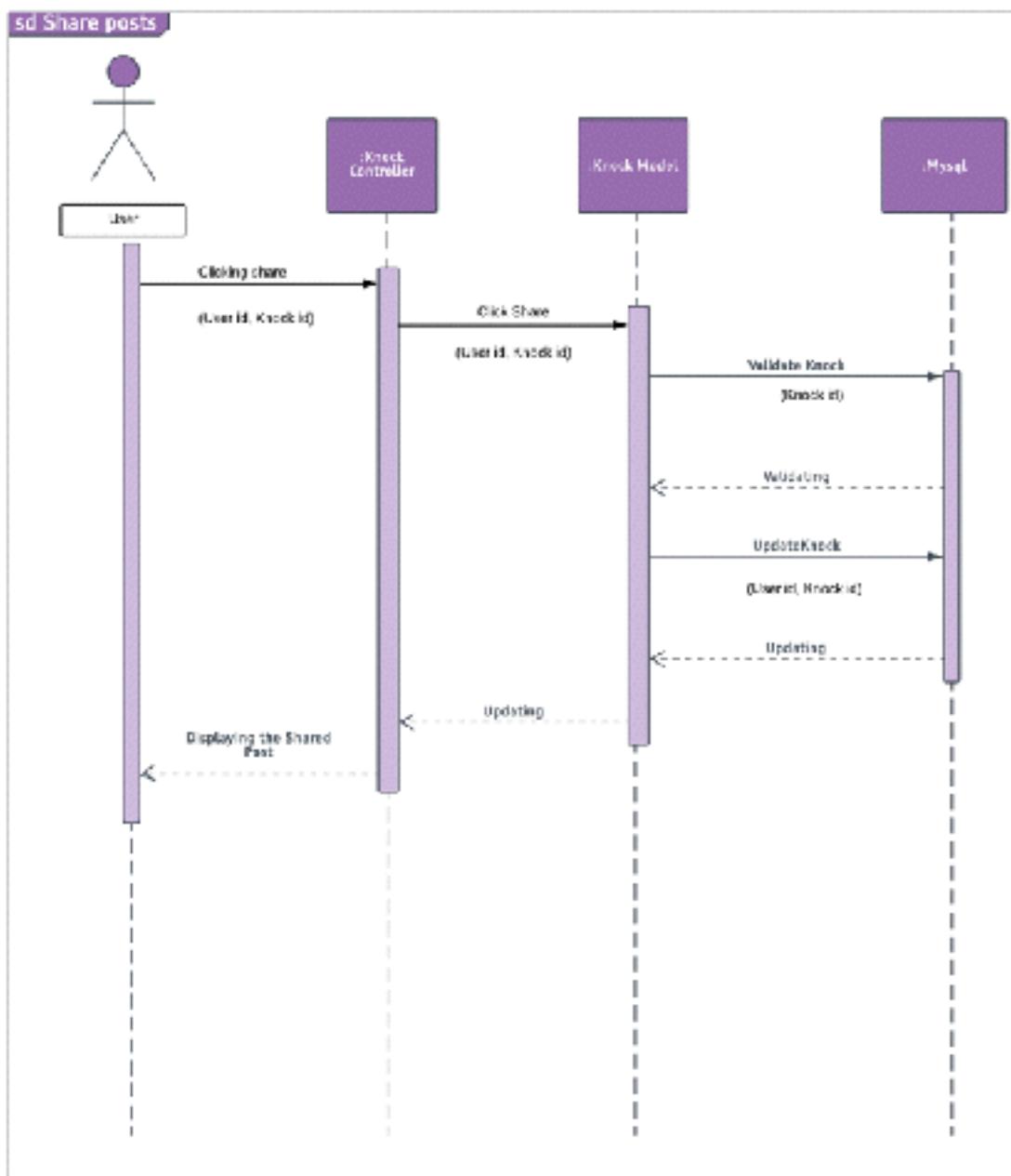
- Request for e-mail verification



- Blocking accounts ( for security purposes )



- Share posts



# Chapter 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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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"> <li>create(Request \$req)</li> <li>This method takes a req object as a parameter and it works as to create an instance of the answer</li> </ul>
	Array	<ul style="list-style-type: none"> <li>userAnswers(Request \$req)</li> <li>This method takes a req object as a parameter and it works as to get the answers of a specific user</li> </ul>
	Array	<ul style="list-style-type: none"> <li>patch(Request \$req)</li> <li>This method takes a req object as a parameter and it works as to get all the users answers for the analysis</li> </ul>
BallonController	Object	<ul style="list-style-type: none"> <li>getUserNotification(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to get the notifications for a specific user</li> </ul>
	Object	<ul style="list-style-type: none"> <li>getAllUserNotification(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to get all the notifications for a specific user</li> </ul>
	Object	<ul style="list-style-type: none"> <li>getAllUserNotificationFr(Request \$request)</li> <li>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</li> </ul>
	Object	<ul style="list-style-type: none"> <li>setToPopped(Request \$request)</li> <li>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.</li> </ul>
	String	<ul style="list-style-type: none"> <li>setToseen(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to set the notification as seen before, so that it wont be notifying the user again.</li> </ul>
BlobController	Object	<ul style="list-style-type: none"> <li>createRecord(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to create an instance of a record</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Controller Name	Modifier Type	Method and Description
FileController	Object	<ul style="list-style-type: none"> <li>• retrieveRecord(Request \$request , \$id)</li> <li>• This method takes a req object and id as a parameter and it works as to retrieve a specific record</li> </ul>
	Double	<ul style="list-style-type: none"> <li>• retrieveRecordMeta(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to retrieve record meta data</li> </ul>
	Array	<ul style="list-style-type: none"> <li>• retrieveFileMeta(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to retrieve file meta data</li> </ul>
	Integer	<ul style="list-style-type: none"> <li>• uploadImage(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to upload an image</li> </ul>
	Integer	<ul style="list-style-type: none"> <li>• uploadFile(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to upload a file</li> </ul>
	String	<ul style="list-style-type: none"> <li>• uploadAvatar(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to upload an avatar</li> </ul>
	String	<ul style="list-style-type: none"> <li>• uploadGroupPicture(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to upload group picture</li> </ul>
	String	<ul style="list-style-type: none"> <li>• uploadCover(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to upload cover picture</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• retrieveImage(Request \$request , \$id)</li> <li>• This method takes a req object and an id as a parameter and it works as to retrieve a specific image</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• retrieveFile(Request \$request , \$id)</li> <li>• This method takes a req object and an id as a parameter and it works as to retrieve a specific file</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• retrieveAvatar(Request \$request , \$id)</li> <li>• This method takes a req object and an id as a parameter and it works as to retrieve a specific avatar</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Controller Name	Modifier Type	Method and Description
	String	<ul style="list-style-type: none"> <li>• reinstall(Request \$request)</li> <li>• 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</li> </ul>
	String	<ul style="list-style-type: none"> <li>• removeAllFriends(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to remove all friends from a specific circle</li> </ul>
	String	<ul style="list-style-type: none"> <li>• createRandomKnocks(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to create random knocks for testing purposes.</li> </ul>
	String	<ul style="list-style-type: none"> <li>• createRandomSocial(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to create random knocks for testing purposes.</li> </ul>
	String	<ul style="list-style-type: none"> <li>• createRandomReactions(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to create random reactions for testing purposes.</li> </ul>
	String	<ul style="list-style-type: none"> <li>• createRandomGroups(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to create random groups for testing purposes.</li> </ul>
	String	<ul style="list-style-type: none"> <li>• createRandomCircles(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to create random circles for testing purposes.</li> </ul>
	Array	<ul style="list-style-type: none"> <li>• watchMembershipPublications()</li> <li>• This method takes a req object as a parameter and it works as to check and remove duplications for testing purposes.</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Controller Name	Modifier Type	Method and Description
EducationController	String	<ul style="list-style-type: none"> <li>• addRandomEntry(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to add random entries for personal information for testing purposes.</li> </ul>
	Integer	<ul style="list-style-type: none"> <li>• createEducation(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to create an education instance</li> </ul>
	Array	<ul style="list-style-type: none"> <li>• retrieveEducation(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to retrieve education objects</li> </ul>
	String	<ul style="list-style-type: none"> <li>• updateEducation(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to update education objects</li> </ul>
	String	<ul style="list-style-type: none"> <li>• deleteEducation(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to delete education objects</li> </ul>
EnvelopeController		
FavUserController		
GroupController	String	<ul style="list-style-type: none"> <li>• createGroup(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to create a group</li> </ul>
	String	<ul style="list-style-type: none"> <li>• joinPublicGroup(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to enable to join a public group</li> </ul>
	String	<ul style="list-style-type: none"> <li>• joinClosedGroup(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to enable to join a closed group</li> </ul>
	String	<ul style="list-style-type: none"> <li>• addMemberPublicGroup(Request \$request)</li> <li>• 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</li> </ul>
	Array	<ul style="list-style-type: none"> <li>• getGroups(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to get all the groups</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Controller Name	Modifier Type	Method and Description
<b>GroupController</b>	Object	<ul style="list-style-type: none"> <li>• retrieveGroupForJoin(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to retrieve the groups in order to join it.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• getPictures(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to retrieve the group's pictures</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• getFiles(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to retrieve the group's files</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• getVoices(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to retrieve the group's records</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• getVideos(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to retrieve the group's videos</li> </ul>
	String	<ul style="list-style-type: none"> <li>• updateGroupInfo(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to update the information for a specific group</li> </ul>
	String	<ul style="list-style-type: none"> <li>• updateGroupPrivacy(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to update the privacy for a specific group</li> </ul>
	String	<ul style="list-style-type: none"> <li>• deleteGroup(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to delete a specific group</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• getGroupMembers(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to get the members of a specific group</li> </ul>
		checkUserInGroup(Request \$request)
<b>GroupMemberController</b>	String	<ul style="list-style-type: none"> <li>• checkOwner(Request \$request)</li> <li>• 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</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Controller Name	Modifier Type	Method and Description
GroupsController	String	<ul style="list-style-type: none"> <li>checkAdmin(Request \$request)</li> <li>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</li> </ul>
	String	<ul style="list-style-type: none"> <li>removeMember(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to remove a member from the group</li> </ul>
	Object	<ul style="list-style-type: none"> <li>getMembersPosition(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to get the position of the members in the group</li> </ul>
	String	<ul style="list-style-type: none"> <li>setMembersToAdmin(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to set a member as an admin</li> </ul>
	String	<ul style="list-style-type: none"> <li>setAdminToMember(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to set admin to a normal group member</li> </ul>
	String	<ul style="list-style-type: none"> <li>setToOwner(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to set specific user as an owner</li> </ul>
	Object	<ul style="list-style-type: none"> <li>lazy(Request \$req)</li> <li>This method takes a req object as a parameter and it works as to retrieve all the hashtags that matches the given hashtag</li> </ul>
	Array	<ul style="list-style-type: none"> <li>getRecentHashtags(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to retrieve all the recent hashtags.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>findHashTag(Request \$req, \$hashtag)</li> <li>This method takes a req object and the hashtag as a parameter and it works as to retrieve specific hashtag</li> </ul>
	Object	<ul style="list-style-type: none"> <li>retrieveTrendKnocks(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to retrieve trend knocks</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Controller Name	Modifier Type	Method and Description
	View	<ul style="list-style-type: none"> <li>viewComment(Request \$request, \$comment )</li> <li>This method takes a req object and comment as a parameter and it works as to view comment of the knock</li> </ul>
	View	<ul style="list-style-type: none"> <li>viewKnockWithComment(Request \$request, \$knock , \$comment )</li> <li>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</li> </ul>
<b>LanguageController</b>	String	<ul style="list-style-type: none"> <li>create(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to create a new language instance</li> </ul>
	Object	<ul style="list-style-type: none"> <li>collect(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to collect and retrieve all the languages.</li> </ul>
<b>MessageBusController</b>		
<b>ObjController</b>	String	<ul style="list-style-type: none"> <li>hide(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to hide a specific object</li> </ul>
<b>PrivacyPresetController</b>		
<b>PrivacySetCircleController</b>		
<b>PrivacySetUserController</b>		
<b>PublicFigureController</b>		
<b>QuestionController</b>		
<b>ReactionController</b>	String	<ul style="list-style-type: none"> <li>insert_reaction (Request \$request)</li> <li>This method takes a req object as a parameter and it works as to insert new reaction for specific object.</li> </ul>
	String	<ul style="list-style-type: none"> <li>checkinit_reaction(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to check the initial reaction for the specific object</li> </ul>
	String	<ul style="list-style-type: none"> <li>delete_reaction (Request \$request)</li> <li>This method takes a req object as a parameter and it works as to delete reaction for an object</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Controller Name	Modifier Type	Method and Description
ReplyController	Object	<ul style="list-style-type: none"> <li>getstats_reaction (Request \$request)</li> <li>This method takes a req object as a parameter and it works as to get the count of each reaction for an object</li> </ul>
	String	<ul style="list-style-type: none"> <li>create(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to create replies instance</li> </ul>
	String	<ul style="list-style-type: none"> <li>retrieve(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to retrieve replies</li> </ul>
	Array	<ul style="list-style-type: none"> <li>getReplies(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to get replies</li> </ul>
SavedObjectController		
SavedPresetsController	Object	<ul style="list-style-type: none"> <li>check(Request \$req)</li> <li>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</li> </ul>
	String	<ul style="list-style-type: none"> <li>save(Request \$req)</li> <li>This method takes a req object as to save the preset</li> </ul>
	Array	<ul style="list-style-type: none"> <li>Delete(Request \$req)</li> <li>This method takes a req object as a parameter and it works as to delete preset</li> </ul>
	String	<ul style="list-style-type: none"> <li>setAsDefault(Request \$req)</li> <li>This method takes a req object as a parameter and it works as to set a specific preset as the default for a user</li> </ul>
	Object	<ul style="list-style-type: none"> <li>get(Request \$req)</li> <li>This method takes a req object as a parameter and it works as to gets the saved preset for a user</li> </ul>
SportController	Integer	<ul style="list-style-type: none"> <li>createSport(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to create an sport instance</li> </ul>
	Array	<ul style="list-style-type: none"> <li>retrieveSport(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to retrieve sport objects</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Controller Name	Modifier Type	Method and Description
<b>UserCheckinsController</b>	Object	<ul style="list-style-type: none"> <li>• isBlocked(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to check if specific user is blocked or not</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• retrieveBlockedUser(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to retrieve all the blocked users</li> </ul>
<b>UserController</b>		
<b>UserController</b>	String	<ul style="list-style-type: none"> <li>• activeRequests()</li> <li>• This method works as to get all the active requests</li> </ul>
	String	<ul style="list-style-type: none"> <li>• userlogin(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to let the user log in</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• goHome(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to get the view of home for the user</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• offerVerify(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to offer the user verification</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• offerVerifyExpired(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to check if the verification is expired or not</li> </ul>
	String	<ul style="list-style-type: none"> <li>• requestVerify(Request \$request)</li> <li>• This method takes a req object as a parameter and it works as to request for verification</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• attemptVerify(Request \$request, \$token)</li> <li>• This method takes a req object and token as a parameter and it works as to attempt for verification</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• attemptUnblock(Request \$request, \$user, \$token)</li> <li>• This method takes a req object and token and user as a parameter and it works as to attempt for unblock</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Controller Name	Modifier Type	Method and Description
	Object	<ul style="list-style-type: none"> <li>attemptUnblockTempPassword(Request \$request, \$user, \$token)</li> <li>This method takes a req object and token and user as a parameter and it works as to attempt for unblock for temporary password</li> </ul>
	Object	<ul style="list-style-type: none"> <li>lost()</li> <li>This method works as to route the user to the guest page</li> </ul>
	String	<ul style="list-style-type: none"> <li>authUsersLanguage()</li> <li>This method works as to authenticate the user language</li> </ul>
	Object	<ul style="list-style-type: none"> <li>retrievePeopleKnocks(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to get the people's knocks</li> </ul>
	Integer	<ul style="list-style-type: none"> <li>retrieveOlderPeopleKnocks(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to get the people's older knocks</li> </ul>
	Integer	<ul style="list-style-type: none"> <li>retrieveNewerPeopleKnocks(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to retrieve the newer knocks</li> </ul>
	Object	<ul style="list-style-type: none"> <li>retrieveUserKnocks(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to retrieve the user knocks</li> </ul>
	Object	<ul style="list-style-type: none"> <li>retrieveOlderUserKnocks(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to retrieve the older user knocks</li> </ul>
	Object	<ul style="list-style-type: none"> <li>getUserAllCircles(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to get all circles that the user has</li> </ul>
	Object	<ul style="list-style-type: none"> <li>retrieveNewerUserKnocks(Request \$request)</li> <li>This method takes a req object as a parameter and it works as to retrieve the newer user knocks</li> </ul>
		getDefaultPreset(Request \$request)

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## 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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Model Name	Modifier Type	Method and Description
Blob	Void	<ul style="list-style-type: none"> <li>• userReaction(\$sender, \$receiver, \$reaction, \$object, \$object_type, \$child, \$parent)</li> <li>• 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</li> </ul>
	Void	<ul style="list-style-type: none"> <li>• sendNotification(\$balloon_type, \$not,\$object , \$reference_url, \$is_replayable , \$sender_name, \$has_picture , \$image_url , \$content)</li> <li>• 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.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• imageBlob(\$object)</li> <li>• This method takes object named object as a parameter and it works as to insert new image blob.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• fileBlob(\$object)</li> <li>• This method takes object named object as a parameter and it works as to insert new file blob.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• avatarBlob(\$object)</li> <li>• This method takes object named object as a parameter and it works as to insert new avatar blob.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• coverBlob(\$object)</li> <li>• This method takes object named object as a parameter and it works as to insert new cover blob.</li> </ul>
	Integer	<ul style="list-style-type: none"> <li>• recordBlob(\$object)</li> <li>• This method takes object named object as a parameter and it works as to insert new record blob.</li> </ul>
	Void	<ul style="list-style-type: none"> <li>• assignParent(\$object)</li> <li>• This method takes object named object as a parameter and it works as to assign a parent for a blob.</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Model Name	Modifier Type	Method and Description
	Object	<ul style="list-style-type: none"> <li>• index()</li> <li>• This method works as to retrieve the index of a blob</li> </ul>
	String	<ul style="list-style-type: none"> <li>• mongoToken()</li> <li>• This method works as to retrieve the token from mongo database of a blob</li> </ul>
	String	<ul style="list-style-type: none"> <li>• blobQuote()</li> <li>• This method works as to retrieve the quote of a blob</li> </ul>
	Double	<ul style="list-style-type: none"> <li>• blobDuration()</li> <li>• This method works as to retrieve the duration of a blob</li> </ul>
	String	<ul style="list-style-type: none"> <li>• vnBlob()</li> <li>• This method works as to enable the format of the file</li> </ul>
	String	<ul style="list-style-type: none"> <li>• retrieveImgBlob()</li> <li>• This method works as to retrieve the image blob.</li> </ul>
	String	<ul style="list-style-type: none"> <li>• retrieveImgCompressed()</li> <li>• This method works as to retrieve the image blob but compressed.</li> </ul>
Candy_session		
Career	Void	<ul style="list-style-type: none"> <li>• initialize(\$works_at,\$works_since, \$works_to,\$works_what,\$works_as)</li> <li>• 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.</li> </ul>
Circle_member	Void	<ul style="list-style-type: none"> <li>• initialize(\$user , \$circle)</li> <li>• This method takes user and circle objects as a parameter and it works as to initialize circle member instance.</li> </ul>
Circle	Boolean	<ul style="list-style-type: none"> <li>• isMember ( \$user )</li> <li>• 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.</li> </ul>
	Void	<ul style="list-style-type: none"> <li>• initialize(\$name , \$thumbnail)</li> <li>• This method takes name of the circle and the thumbnail as a parameter and it works as to initialize a new circle.</li> </ul>
		init(\$name, \$thumbnail, \$user)

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Model Name	Modifier Type	Method and Description
	Array	<ul style="list-style-type: none"> <li>getGroupKnocksRegularMax(\$max)</li> <li>This method takes min object as a parameter and it works as to retrieve the group knocks with the given maximum limit.</li> </ul>
hashtags	Void	<ul style="list-style-type: none"> <li>createOrUpdate(\$q)</li> <li>This method works as to create a hashtag if it is the first hashtag or to update its count if it already exists</li> </ul>
	Array	<ul style="list-style-type: none"> <li>retrieveOlderKnocks(\$min)</li> <li>This method takes min object as a parameter and it works as to retrieve the older knocks</li> </ul>
	Array	<ul style="list-style-type: none"> <li>retrieveNewerKnocks(\$max)</li> <li>This method takes min object as a parameter and it works as to retrieve the newer knocks</li> </ul>
	Array	<ul style="list-style-type: none"> <li>retrieveKnocks()</li> <li>This method takes min object as a parameter and it works as to retrieve the knocks</li> </ul>
High_education	Void	<ul style="list-style-type: none"> <li>initialize(\$study_at , \$study_what, \$study_since , \$study_to , \$grade)</li> <li>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.</li> </ul>
Hobby	Void	<ul style="list-style-type: none"> <li>initialize(\$name)</li> <li>This method takes name of the hobby as a parameter and it works as to initialize hobby instance.</li> </ul>
ignore_object		
Jam_circle		
Knock	Void	<ul style="list-style-type: none"> <li>addFollower(\$user)</li> <li>This method works as to add a follower for a specific knock</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>canEdit(\$user)</li> <li>This method works as to check if a user can edit a knock or not</li> </ul>
	Void	<ul style="list-style-type: none"> <li>deleteKnock()</li> <li>This method works as to delete a knock</li> </ul>
	Object	<ul style="list-style-type: none"> <li>isIgnored(\$user)</li> <li>This method works as to check if a knock is being notified with or not</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Model Name	Modifier Type	Method and Description
	Object	<ul style="list-style-type: none"> <li>• addFollower(\$user)</li> <li>• This method works as to add a follower to the object with specific user</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• knockIndex()</li> <li>• This method works as to get the knock index</li> </ul>
		index()
		updatePublicPresetNum(\$num)
	Void	<ul style="list-style-type: none"> <li>• initialize(\$type)</li> <li>• This method takes the object type as a parameter and it works as to initialize a new object.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• isAvailable (\$requestMaker)</li> <li>• 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.</li> </ul>
Privacy_preset		
Privacy_set_circle		
Privacy_set_user		
Public_figure	Void	<ul style="list-style-type: none"> <li>• initialize(\$name,\$type,\$verified)</li> <li>• 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.</li> </ul>
Question		
Reaction	Void	<ul style="list-style-type: none"> <li>• init(\$user, \$object, \$reaction, \$type)</li> <li>• This method takes reaction object and reaction type and user as a parameter and it works as to initialize new reaction instance.</li> </ul>
	Void	<ul style="list-style-type: none"> <li>• initialize(\$object)</li> <li>• This method takes reaction object as a parameter and it works as to initialize new reaction instance.</li> </ul>
	String	<ul style="list-style-type: none"> <li>• initializeReaction(\$parent_id,\$type)</li> <li>• 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.</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Model Name	Modifier Type	Method and Description
Reply	Object	<ul style="list-style-type: none"> <li>• replyReplies()</li> <li>• This method works as to get the replies for a specific reply.</li> </ul>
	Void	<ul style="list-style-type: none"> <li>• initialize(\$object)</li> <li>• This method takes reply object as a parameter and it works as to initialize new reply.</li> </ul>
saved_object		
Saved_presets		
Sport	Void	<ul style="list-style-type: none"> <li>• initialize(\$name)</li> <li>• This method takes sport name as a parameter and it works as to initialize new sport.</li> </ul>
Stage_member		
Stage		
Static_message	Void	<ul style="list-style-type: none"> <li>• initialize(\$object)</li> <li>• This method takes static message object as a parameter and it works as to initialize new static message.</li> </ul>
	Void	<ul style="list-style-type: none"> <li>• translate(\$object)</li> <li>• This method takes static message object as a parameter and it works as to translate it.</li> </ul>
Talent_category		
Talent_object		
Talent		
user_blocks	Void	<ul style="list-style-type: none"> <li>• initialize(\$blocked_user_id)</li> <li>• This method takes the blocked user id and initialize the block for him</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• index()</li> <li>• This method returns the Index of the block</li> </ul>
User_checkins		
User_hashtags		
User_keywords		
User_log	Void	<ul style="list-style-type: none"> <li>• autoLog(\$url , \$ip , \$method)</li> <li>• 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,</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Model Name	Modifier Type	Method and Description
	Void	<ul style="list-style-type: none"> <li>• addUserLog(\$user , \$url , \$ip)</li> <li>• This method takes user id and the url and the ip of the user as a parameter and it works as to log in.</li> </ul>
	Void	<ul style="list-style-type: none"> <li>• addAnonymousLog(\$url , \$ip)</li> <li>• 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.</li> </ul>
<b>user_reports</b>		
User_request	Void	<ul style="list-style-type: none"> <li>• initialize(\$sender_id , \$receiver_id)</li> <li>• 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.</li> </ul>
	Void	<ul style="list-style-type: none"> <li>• initializeForGroups(\$sender_id , \$receiver_id)</li> <li>• 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.</li> </ul>
User	Boolean	<ul style="list-style-type: none"> <li>• isMemberIn(\$circleToCheck , \$userToCheck)</li> <li>• 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.</li> </ul>
	Array	<ul style="list-style-type: none"> <li>• entryStatus()</li> <li>• This method works as to get the entry for each user</li> </ul>
	Array	<ul style="list-style-type: none"> <li>• getAddresses(\$requester)</li> <li>• This method works as to get the address for each user</li> </ul>
	Array	<ul style="list-style-type: none"> <li>• getSuggestions()</li> <li>• This method works as to get the suggestion for each user</li> </ul>
	Array	<ul style="list-style-type: none"> <li>• getSuggestionsAvoid(\$arr)</li> <li>• This method works as to get the suggestion that are avoided for each user</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• isSuggestableInList(\$arr, \$user)</li> <li>• This method works as to get the suggestions list</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• isSuggestable(\$user)</li> <li>• This method works as to check if the user is suggestible or not</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Model Name	Modifier Type	Method and Description
	Object	<ul style="list-style-type: none"> <li>• passFriends(\$friend, \$prev)</li> <li>• This method works as to check if the user is suggestible or not</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• fullName()</li> <li>• This method works as to get the full name of the user</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• friendsByWeight()</li> <li>• This method works as to set weight for friends</li> </ul>
	Array	<ul style="list-style-type: none"> <li>• mutualFriends(\$oth)</li> <li>• This method works as to get the mutual friends between two users</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• isBlockedBy(\$user)</li> <li>• This method works as to check if the user is being blocked by other user</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• isBlocking(\$user)</li> <li>• This method works as to check if the user is blocking other users or not</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• hasNoBlocks(\$user)</li> <li>• This method works as to check if the user has no blocks</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• friends()</li> <li>• This method works as to get the user friends</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• userAddresses()</li> <li>• This method works as to get the user address</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• knocks()</li> <li>• This method works as to get the user knocks</li> </ul>
	Void	<ul style="list-style-type: none"> <li>• updateFriendshipWeight(\$fr, \$weight)</li> <li>• This method works as to update the friendship weight</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• updateLastSeen()</li> <li>• This method works as to update the last seen for a user.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• turnOffChat()</li> <li>• This method works as to turn off the chat for the user and indicate that this user is offline.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• cog()</li> <li>• This method works as to retrieve the user configurations.</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Model Name	Modifier Type	Method and Description
	Boolean	<ul style="list-style-type: none"> <li>• isFriend(\$user)</li> <li>• 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.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• mainCircle()</li> <li>• This method works as to retrieve the main circle of the account owner.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• getCircleId(\$name)</li> <li>• This method takes the circle name as a parameter and it works as to get the circle id of the given circle name.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• hasCircle(\$circle)</li> <li>• This method takes the circle object as a parameter and it works as to check if this user has circles or not.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• hasCircleById(\$circle)</li> <li>• 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.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• hasReceivedRequest(\$id)</li> <li>• This method takes the request id as a parameter and it works as to check if this user has received request or not.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• has_career(\$works_at, \$works_since)</li> <li>• This method takes works at and works since as a parameter and it works as to check if this user has career or not.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• has_hobby(\$name, \$url)</li> <li>• This method takes hobby name and hobby url as a parameter and it works as to check if this user has hobby or not.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• has_public_figure(\$name, \$url)</li> <li>• 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.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• has_Education(\$uniname, \$study_since)</li> <li>• 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.</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Model Name	Modifier Type	Method and Description
User	Boolean	<ul style="list-style-type: none"> <li>• hasHighEducation(\$uname, \$study_since)</li> <li>• 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.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• hasSport(\$sportname, \$url)</li> <li>• This method takes sport name and sport url as a parameter and it works as to check if this user has sport or not.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• userLanguage()</li> <li>• This method works as to retrieve the user languages.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• userLanguageFont()</li> <li>• This method works as to retrieve the user font family.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• userLanguageAlignment()</li> <li>• This method works as to retrieve the user language alignment.</li> </ul>
	String	<ul style="list-style-type: none"> <li>• profilePicture()</li> <li>• This method works as to retrieve the user profile picture .</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• profilePictures()</li> <li>• This method works as to retrieve the user profile pictures .</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• coverPictures()</li> <li>• This method works as to retrieve the user cover pictures .</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• profilePictureBlob()</li> <li>• This method works as to retrieve the user profile picture blob .</li> </ul>
	Array	<ul style="list-style-type: none"> <li>• profilePictureBlobObject()</li> <li>• This method works as to retrieve the user profile picture blob as object .</li> </ul>
	Array	<ul style="list-style-type: none"> <li>• coverPictureBlob()</li> <li>• This method works as to retrieve the user cover picture blob.</li> </ul>
	String	<ul style="list-style-type: none"> <li>• nextProfilePicture()</li> <li>• This method works as to retrieve the user's next profile picture .</li> </ul>
	String	<ul style="list-style-type: none"> <li>• currentProfilePicture()</li> <li>• This method works as to retrieve the user current profile picture .</li> </ul>

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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

## CHAPTER 5 : SYSTEM IMPLEMENTATION

Model Name	Modifier Type	Method and Description
User	Object	<ul style="list-style-type: none"> <li>• soundsLike(\$q)</li> <li>• 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.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• soundsLikeID(\$q)</li> <li>• 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.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• FriendsSoundsLikeID(\$q)</li> <li>• 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.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• friendsObjects()</li> <li>• This method works as to retrieve the user friends from the main circle.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• friendsToChat()</li> <li>• This method works as to retrieve the user friends and their last seen and their status.</li> </ul>
	Boolean	<ul style="list-style-type: none"> <li>• isMatched(\$q)</li> <li>• This method takes q object as a parameter and it works as to get the similarity between users accounts.</li> </ul>
	String	<ul style="list-style-type: none"> <li>• generateRandomString(\$length)</li> <li>• This method takes length object as a parameter and it works as to generate random strings</li> </ul>
	Void	<ul style="list-style-type: none"> <li>• pairAsFriend(\$friend)</li> <li>• 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.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• devices()</li> <li>• This method works as to get the user devices</li> </ul>
	Array	<ul style="list-style-type: none"> <li>• deviceInfo(\$device)</li> <li>• This method works as to get the user devices information</li> </ul>

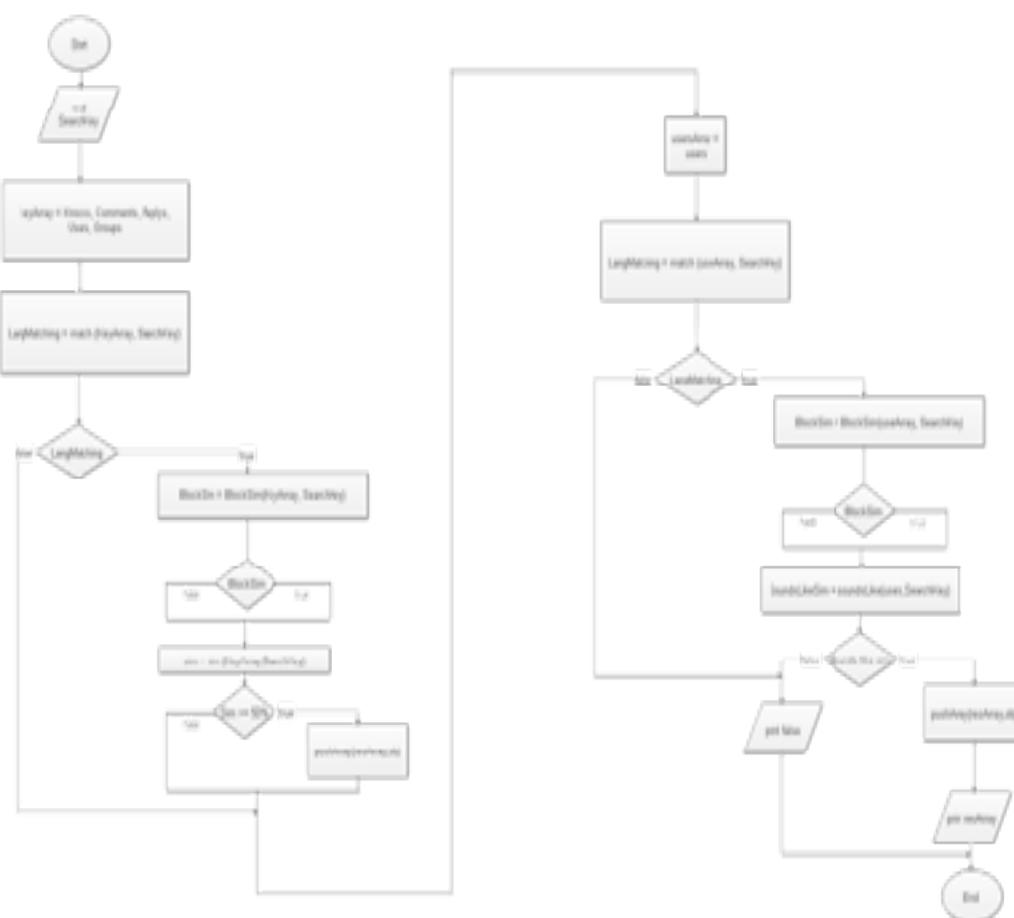
## CHAPTER 5 : SYSTEM IMPLEMENTATION

Model Name	Modifier Type	Method and Description
	Void	<ul style="list-style-type: none"> <li>• createCirclesMembership(\$circles)</li> <li>• This method takes circle object as a parameter and it works for the user to be a circle member.</li> </ul>
	Object	<ul style="list-style-type: none"> <li>• isBlockedAccount()</li> <li>• This method works as to check if the account is blocked</li> </ul>
UserAddress	Void	<ul style="list-style-type: none"> <li>• init(\$object)</li> <li>• This method works as to initialize user address</li> </ul>

### 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.

### 5.3.2.Voice Commands Algorithm

#### 5.3.2.1.Algorithm Flowchart

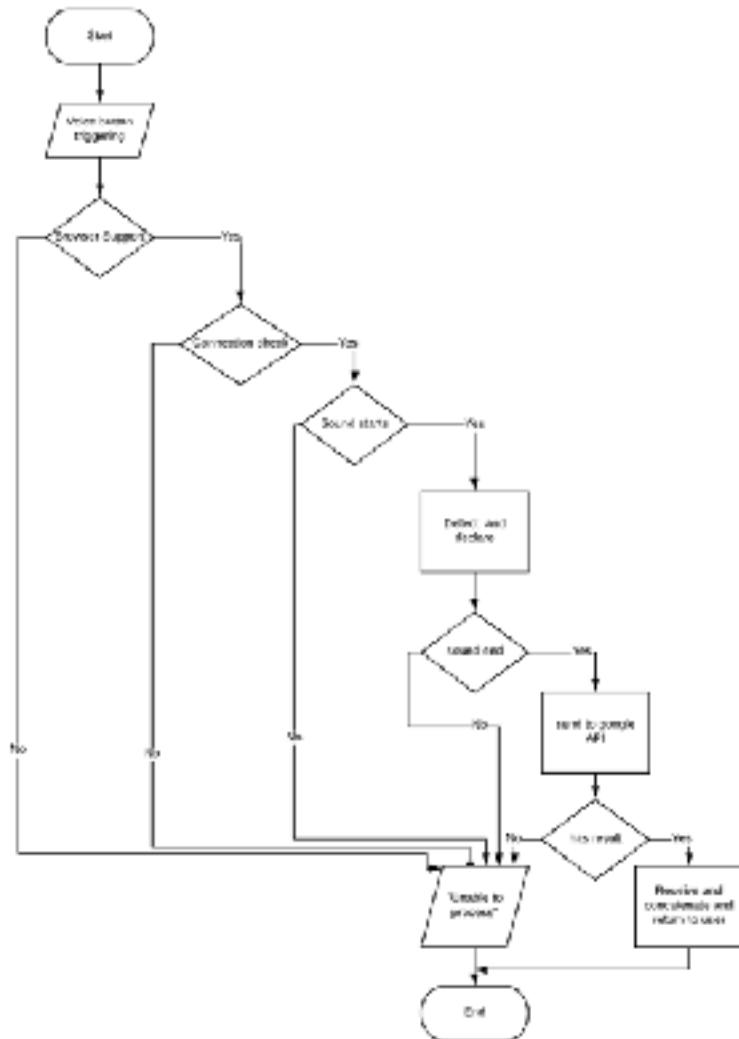


### 5.3.2.2.Algorithm Description

This feature produces by knocks app is for processing the user voice into commands to process function the user need to provide the voice or text first and then the input must be matching with the existing key words and if there is no matching then the system cannot recognize , else if matching appearing then calling back the requested function should be triggered and if there is any data writing or redirecting, the system give time out for calling off the requested function, else the system will execute immediately.

### 5.3.3.Voice to text conversion Algorithm

#### 5.3.3.1.Algorithm Flowchart



### 5.3.3.2.Algorithm Description

First of all, when the user clicks on the voice recording button, the system should first check if the browser supports this feature or not and if it does not support it then it will be unable to process the voice, else the system will check for internet connection if there is no connection then it will be unable to process the voice, else the for sound start it will detected and declare for voice and when the sound ends, it sends to google and for the result failure it won't be able to process the function, else it receives and concatenates and return to the user the speech in text.

## 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  
knockssimg  
knockssimgframeless  
knocksviewcircle  
knocksreactor  
knock  
knocksmultipleuploader  
knockssimageeditor  
knockscirclechip  
knocksmultipleswitch  
knockssprivacysetter  
knockspopover

## CHAPTER 5 : SYSTEM IMPLEMENTATION

knocksimagewriter

knocksaddcircle

knockscircleseditor

knocksknock

knockscomment

knockscchildreply

knocksreply

knocksreactionstats

knockstaps

imagequote

knocksfilewriter

knocksballon

knockssuseractions

knockssquickaddcircle

knockselbutton

knocksknockinjector

knockssusercareers

knockssusereducation

knockssuserhigheducation

knockssuserhobby

knockssusersport

knockssuserabout

knocksrtriver

knocksvoicerecognition

knockscroppie

knockssuserabouthedit

knockssusersportedit

knockssuserhobbyedit

knockssusereducationedit

## CHAPTER 5 : SYSTEM IMPLEMENTATION

knocksusercareeredit

knocksuseraboutdelete

knocksgroupcreation

knocksgroupslist

knockspagesearch

knocksgroupmembers

knocksimg

knockschattingzone

knocksconversation

knocksmessagesender

knocksgroupshortcut

knocksgroupjoining

knocksgroupmemberdelete

knocksgrouppictures

knocksgroupfiles

knocksgroupvoices

knocksrightbar

knocksprivacyadjustments

knocksgroupsettings

knocksuserinfo

knocksgroupmemberposition

knockshashhtag

knocksusercircles

knocksqickcircleadder

knockshashtagchip

knocksimagestates

knocksuserinfodelete

knocksphotocomments

knocksusergenralinfo

## CHAPTER 5 : SYSTEM IMPLEMENTATION

knocksdepimgviewer

knockscirclemembers

knockscollapse

knocksmediaobject

knocksshowkeys

knocksrecordermpeg

knocksvoicecommands.vue

knocksquickaddress.vue

knocksaddressviewer.vue

knocksdatepicker.vue

knockselect.vue

knocksusersuggestions.vue

knocksusersettings.vue

knockswatchmywindow.vue

knocksblockuser.vue

knocksblockuserlist.vue

knocksvetifyuser.vue

knockstips.vue

knocksshowdevice.vue

knocksdateviewer.vue

knockseditcirclename.vue

knockswelcomeslider.vue

knocksforgotmypassword.vue

knockssitemap.vue

knocksprivacytips.vue

knocksmeganumber.vue

knockstrendslist.vue

knocksgenderviewer.vue

## 5.4.2.Vue Component specification

### 1. knocksfileinput.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

### 2. 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

### 3. knocksdatepicker.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 the add and subtract dates for determining the minimum and the maximum values, it also can finally return the database format.

### 4. 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).

## 5. 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 it's id), message (and it translates this message from the database with it's 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 provid 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 invlove 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 replacments as much as you want, as you define your replacements as an array you can define any number of replacments 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.

**6. loader.vue**

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

**7. 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

**8. knocksprogressbarindecator.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

**9. 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, knockssueractions, knockscollapse, knockssaddressviewer, knocksshowkeys, staticmessage, and eltooltip.

### 10. 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, knockssplayer 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 record 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.

## 11. 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.

## 12. 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 : knocksmimageeditor, knocks popover.

## 13. knocksmimageeditor.vue

- Brief Description

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

- Props

gid, crops

## 14. knockscirclechip.vue

- Brief Description

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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 : knocks\_rebound\_circle (payloads : circle and if it matched with the circle id it bents the circle data), knocks\_circle\_key\_update ( it does the same function of knocks\_user\_key\_update), knocks\_circle\_add\_member ( payloads : circle, member and updates the circle model to add a member programmatically), knocks\_circle\_remove\_member ( the same as add member but it removes).

### 15. 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).

### 16. 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)

### 17. 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 ).

#### 18. 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, untoolipped, CSS customizations.
- API Events  
knocks\_submit (explained before ) kncocks\_changed\_taps\_value ( payload : scope and value and it changes it's value ).
- Used Components  
None

#### 19. knocksfleviewer.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

#### 20. knocksballon.vue

- Brief Description  
This component is used for representing the balons 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 stating 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.

## 21. Knocksimageviewer.vue

### 22. 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 ).

## 23. knocksusercareers.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

- Used Components

using the previously made component knocks selector.

## 24. 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

- Used Components

using the previously made component knocks selector

## 25. 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

- Used Components
  - using the previously made component knocks selector

## 26. knocksuserhobby.vue

- Brief Description

This component is for viewing the user's hobby
- Used Components
  - using the previously made component knocks selector

## 27. knocksusersport.vue

- Brief Description

This component is for viewing the user's sport
- Used Components
  - using the previously made component knocks selector

## 28. 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.

## 29. 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

**30. knocksvoicerecognition.vue**

**31. 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.

**32. knocksuseraboutedit.vue**

- Brief Description

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

**33. knocksusersportedit.vue**

- Brief Description

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

**34. knocksuserhobbyedit.vue**

- Brief Description

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

**35. knocksusereducationedit.vue**

- Brief Description

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

**36. knocksusercareeredit.vue**

- Brief Description

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

**37. 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 .

**38. knocksgroupcreation.vue**

- Brief Description

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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 .

### 39. 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 knocks show keys

### 40. 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

it has a previously made components knocks user, knock knock, knocks group shortcut .

### 41. 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.

- Used Components

it uses the previously made components knockssusershortcut, knocksgroupmemberdelete, knockstriver,knocksfileviewer, knockssusershortcut, knockssplayer, knockssimg and knocksgroupjoining .

#### 42. 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), knockssimg ( to show group image ) and knocksgroupjoining ( to show the button of showing the group or leaving the group or get into the group).

#### 43. 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).

- 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.

#### 44. knocksgroupmemberdelete.vue

- Brief Description

## CHAPTER 5 : SYSTEM IMPLEMENTATION

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.

### 45. knocksgrouppictures.vue

- Brief Description

This component is for viewing the group pictures.

- Props

Props : group\_id (number).

- Used Components

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

### 46. knocksgroupfiles.vue

- Brief Description

This component is for viewing the group files.

- Props

Props : group\_id (number).

- Used Components

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

### 47. knocksgroupvoices.vue

- Brief Description

This component is for viewing the group voices.

- Props

Props : group\_id (number).

- Used Components

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

#### **48. 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 )

- Used Components

It also uses the previously made component knocksrretiver ( retrieves the group members and the group requests), knockscroppie ( 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 ).

#### **49. knockssuserinfo.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 wrote freely by the user to be added by him as an option.

- Used Components

date picker component

#### **50. 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 intents 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).

- **Used Components**

It also uses the previously made component knocksretriver ( to retrieve the group member positions )

### 51. 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.

### 52. 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.

# Chapter 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

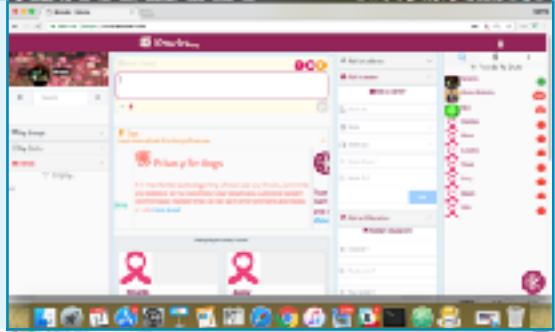
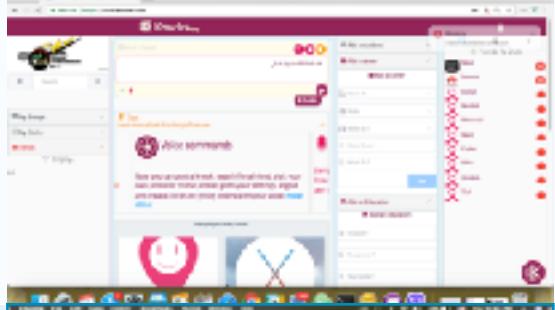
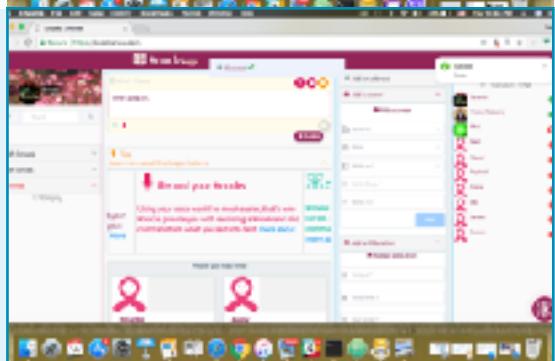
Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

### 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.

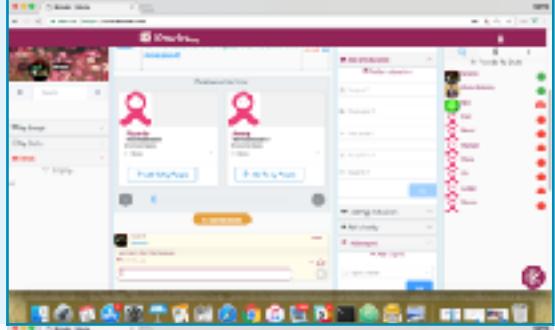
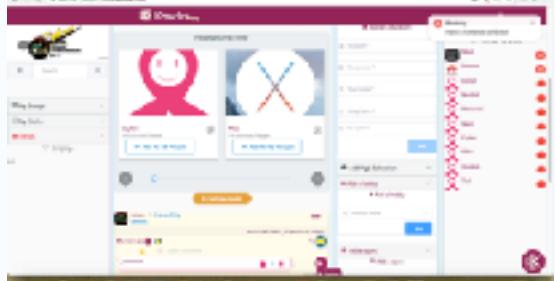
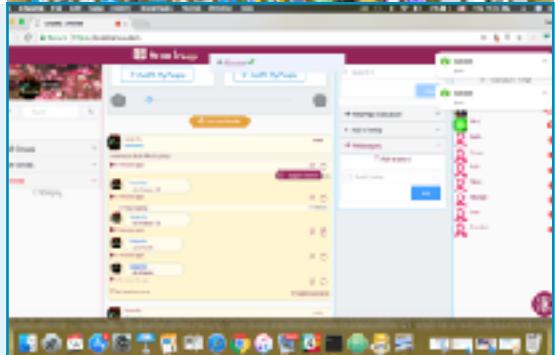
## 6.2. Test Cases

- 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

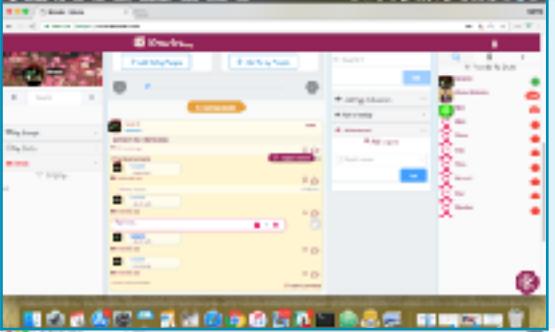
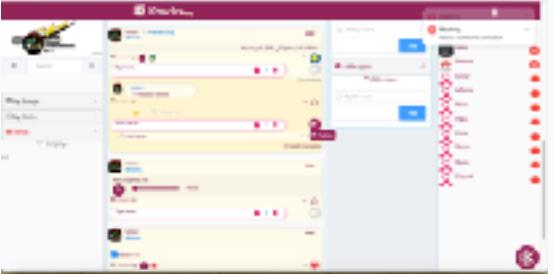
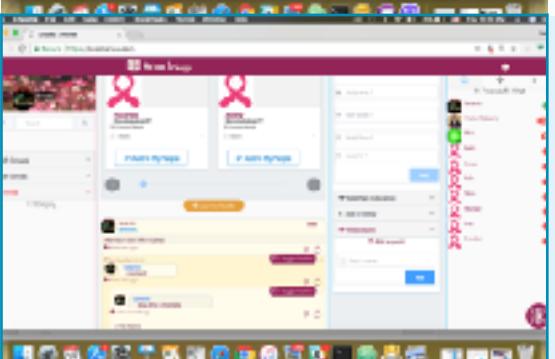
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Write a comment

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

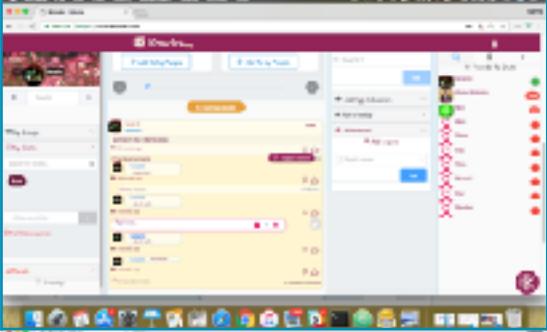
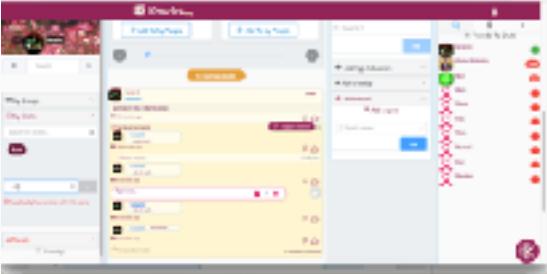
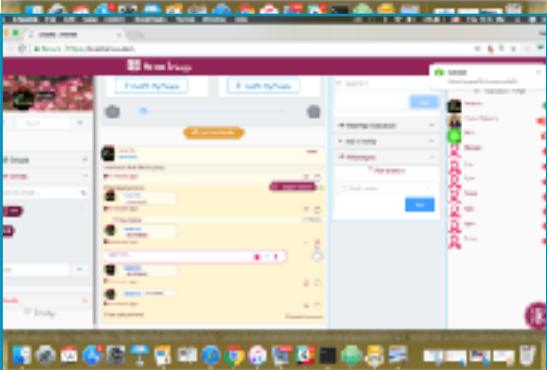
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- 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

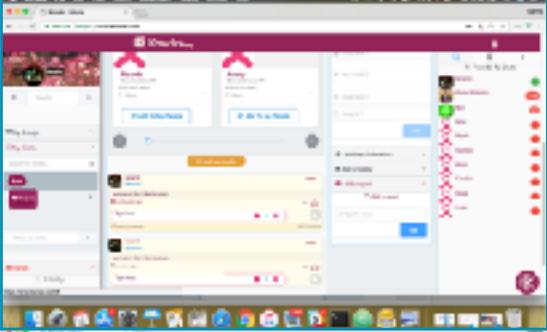
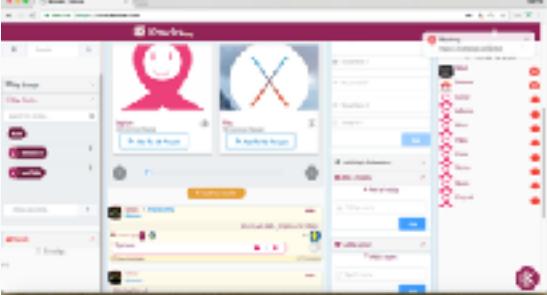
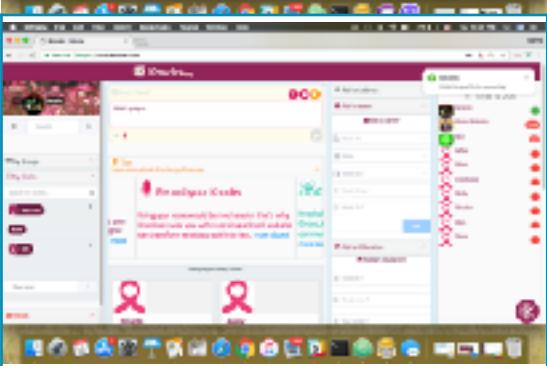
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- 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

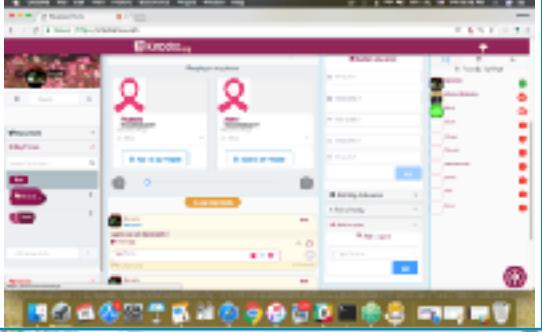
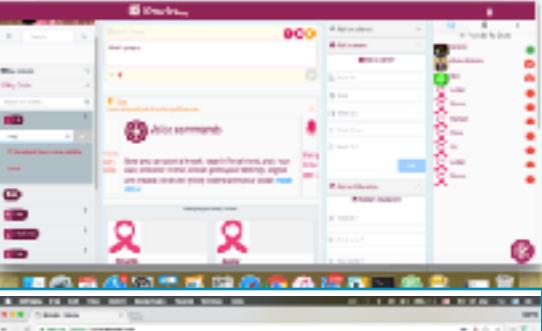
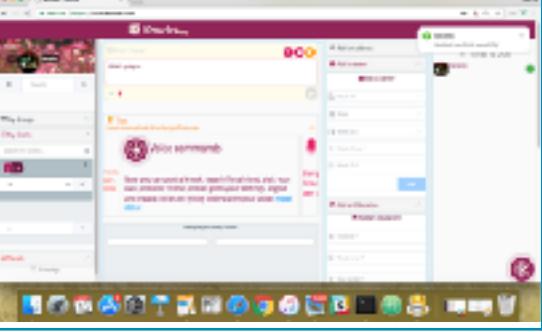
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- **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 circle that has been created before except for all circle, but there is no internet connection )	"There is no Internet Connection"		Success
Selected Circle : a circle that has been created before successfully	"Selected Circle has been deleted successfully"		Success

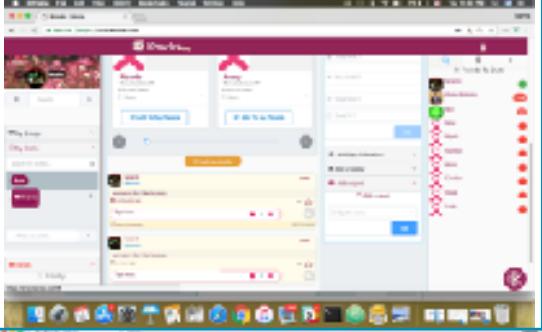
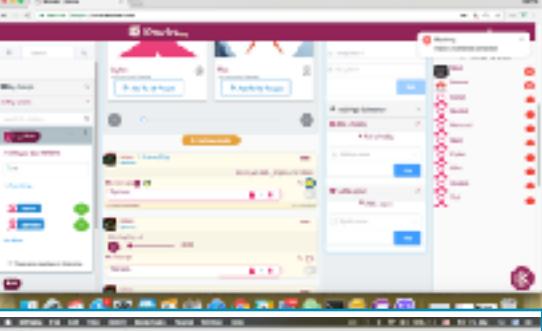
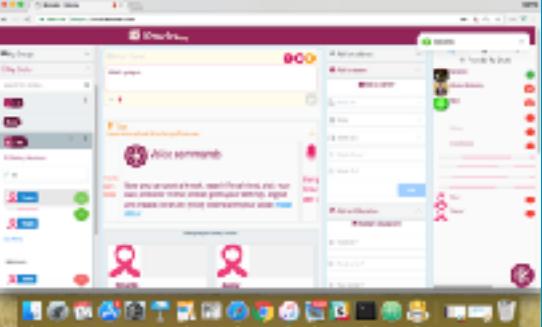
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- 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 created before except for all circle)	"Updated your circle successfully"		Success

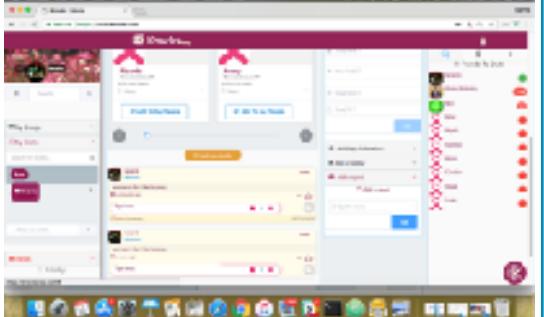
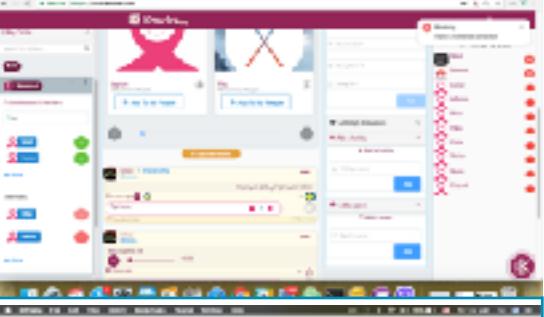
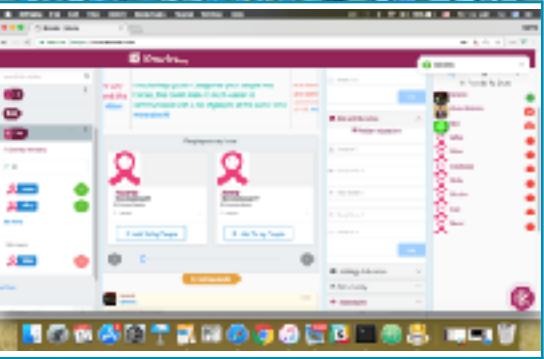
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- **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

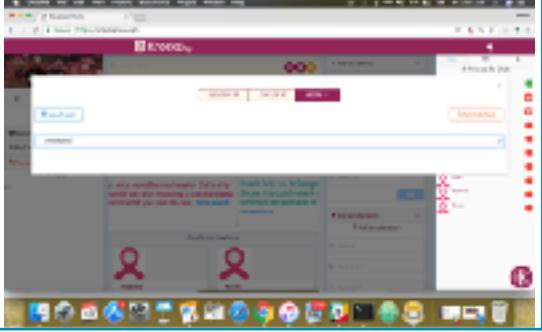
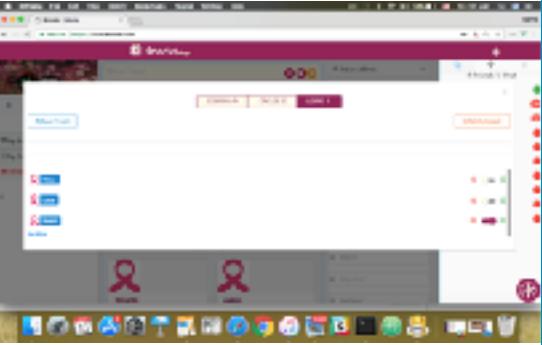
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- **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 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 circle that has been created before except for all circle) Selected User : valid	"Success"		Success

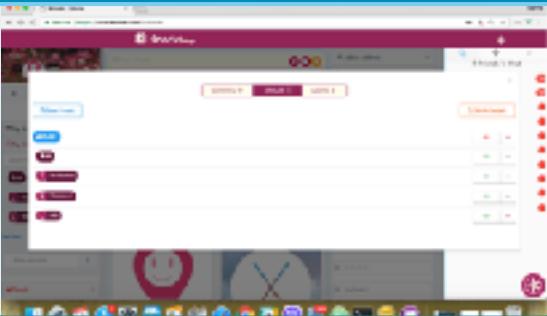
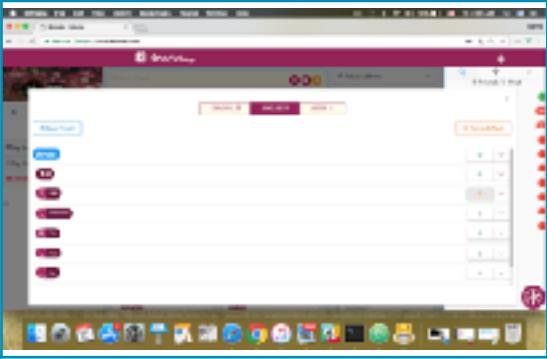
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- 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 : Invalid 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

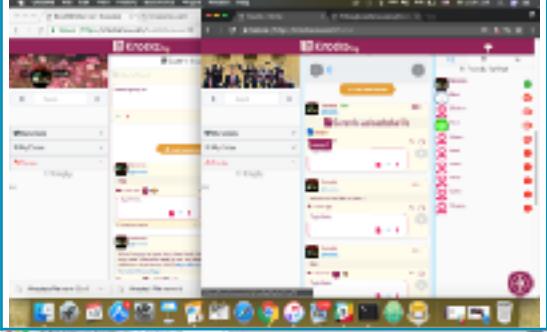
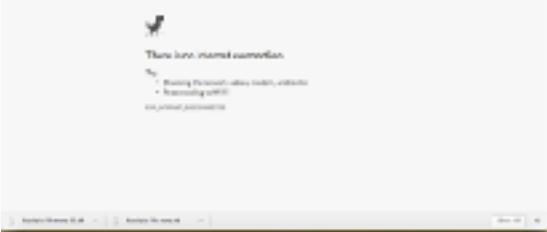
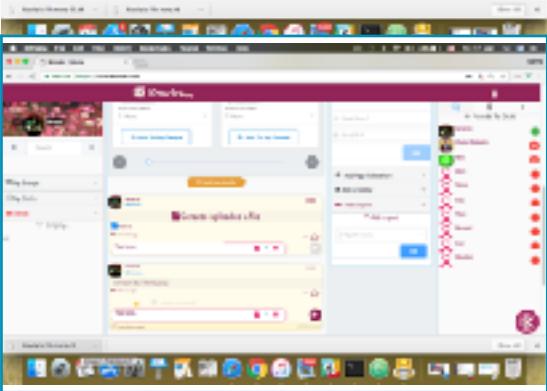
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Regulate a privacy set for circles

Input	Expected Output	Actual Output	Comment
Selected Circle :valid Privacy switch : Valid but there is no internet connection	Not saving the new privacy set		Success
Selected Circle :valid Privacy switch : Valid	Changed privacy set for the selected circle with the selected privacy set		Success

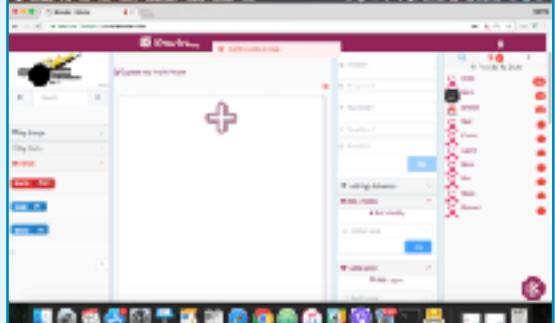
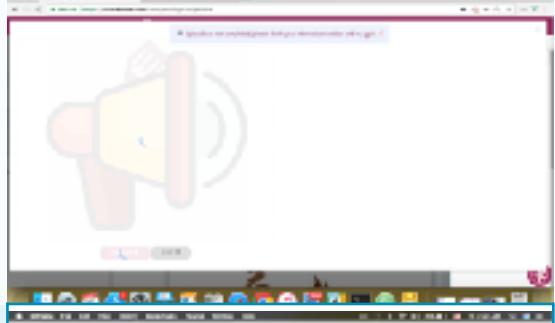
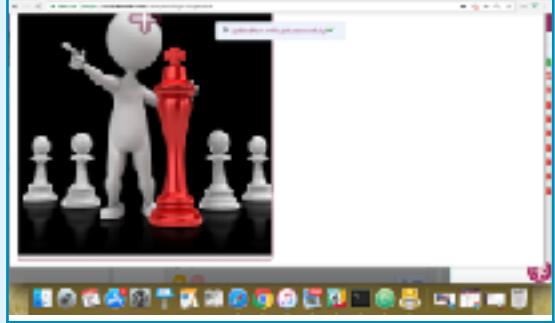
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- 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

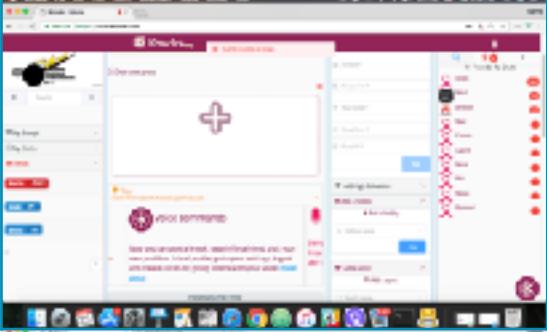
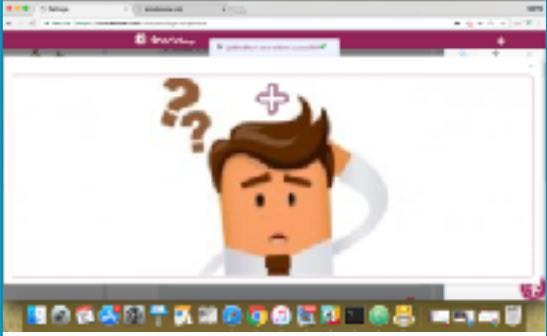
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- **Upload profile picture**

Input	Expected Output	Actual Output	Comment
Selected file : invalid	“Your file must be an image”		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

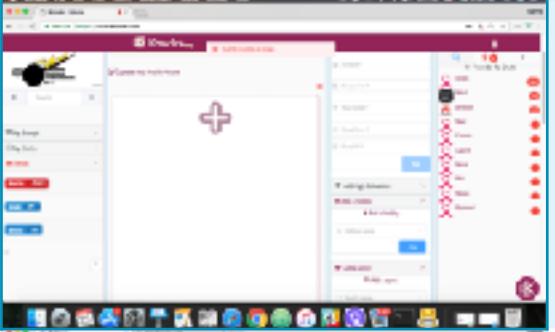
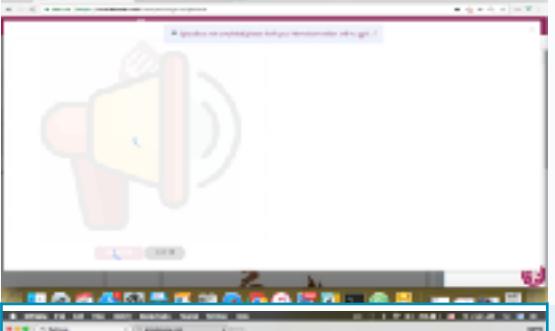
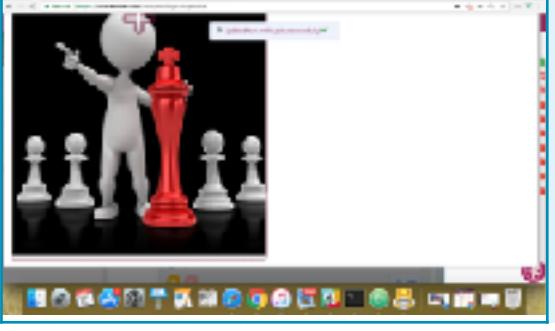
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Upload cover picture

Input	Expected Output	Actual Output	Comment
Selected file : invalid	“Your file must be an image”		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

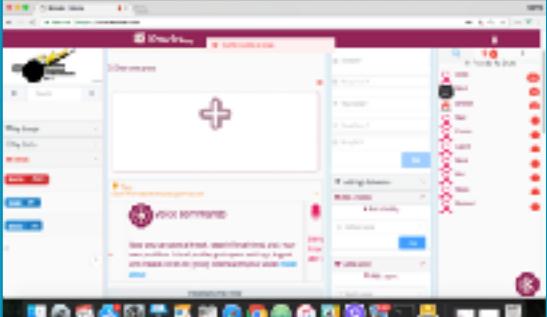
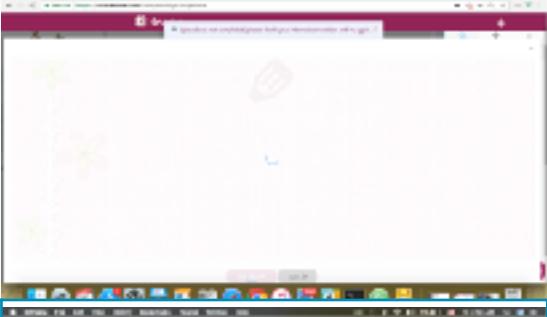
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Update profile picture

Input	Expected Output	Actual Output	Comment
Selected file : invalid	“Your file must be an image”		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

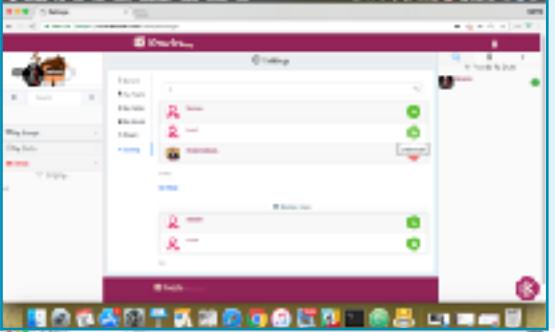
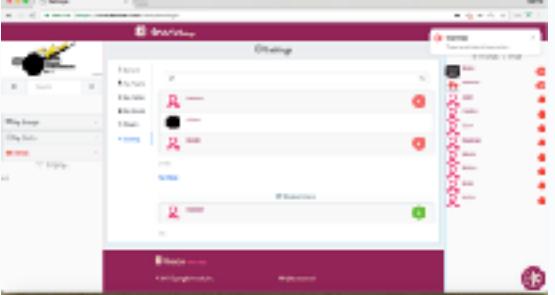
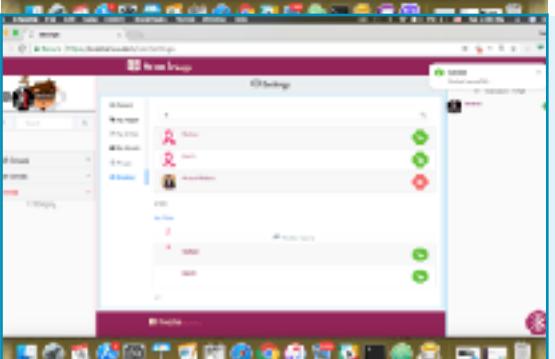
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Update cover picture

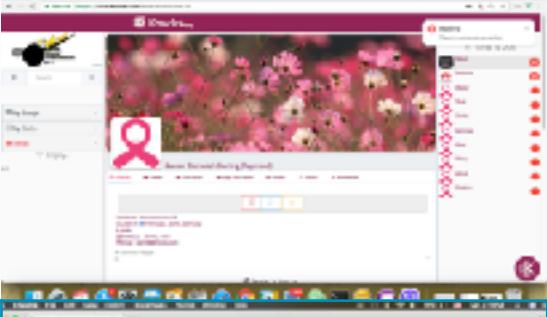
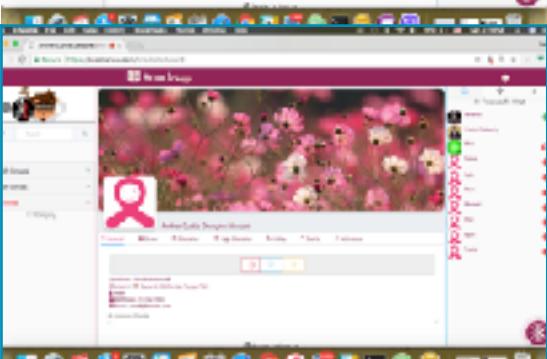
Input	Expected Output	Actual Output	Comment
Selected file : invalid	“Your file must be an image”		Success
Selected file : valid But there is no internet connection	“Upload was not completed, Please check your internet connection and try again.”		Success
Selected file :valid	“updated your cover picture successfully”		Success

## CHAPTER 5 : SYSTEM IMPLEMENTATION

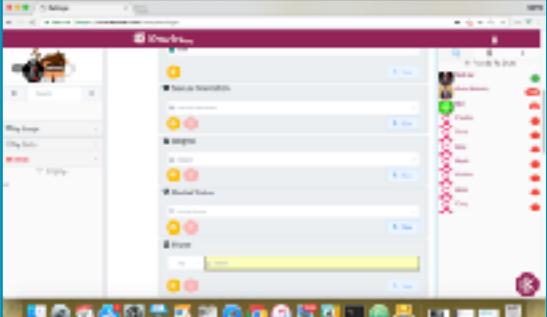
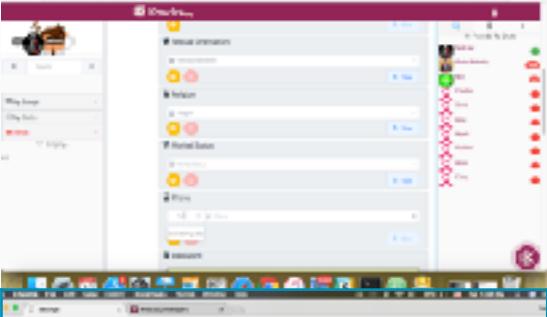
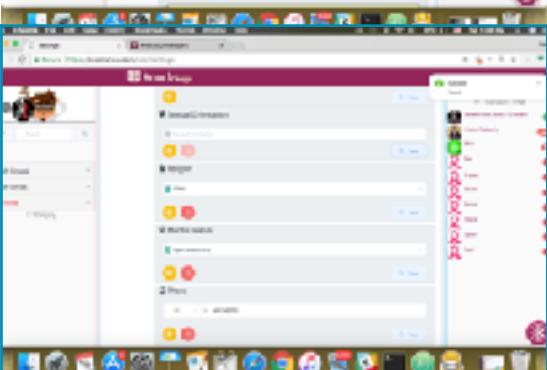
- **Block a friend**

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

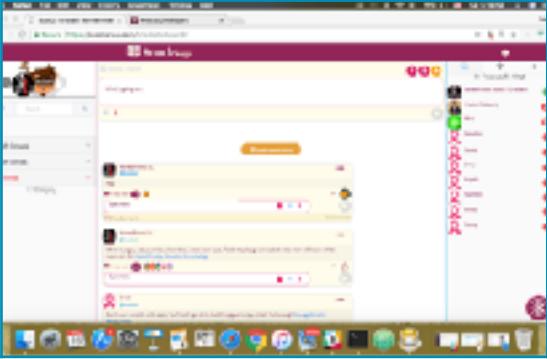
- **Unfriend a friend**

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

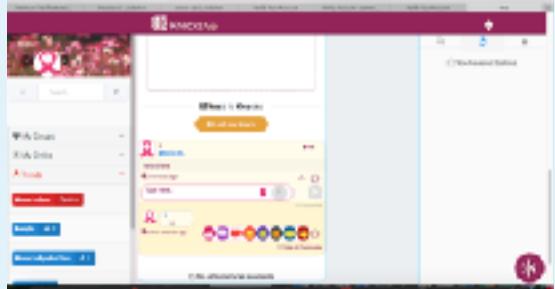
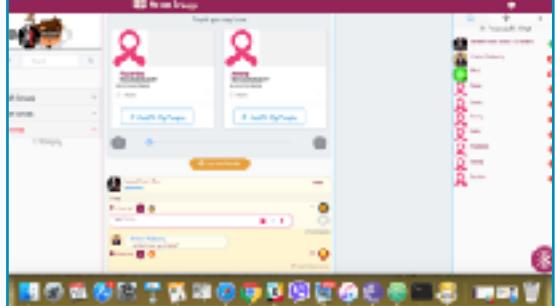
- Update personal information (Eg :Phone number)

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**

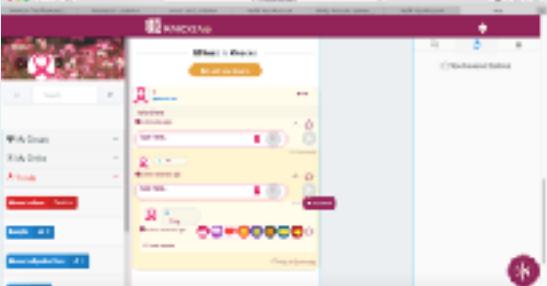
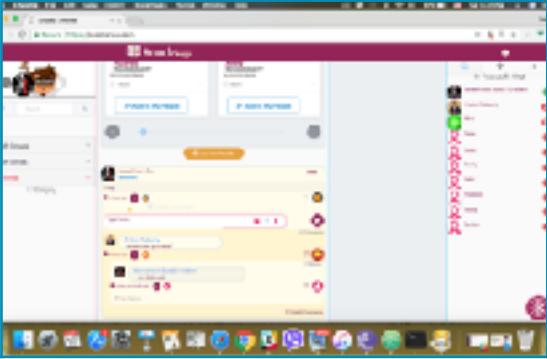
Input	Expected Output	Actual Output	Comment
Selected Post : valid Reaction Type : valid But there is no internet connection	Not saving reaction		Success
Selected Post : invalid Reaction Type : valid	Displays your reaction across the post		Success

- **React to comment**

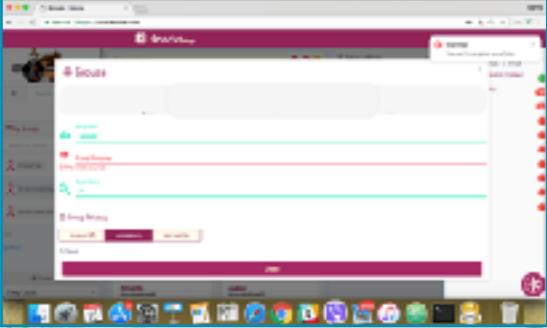
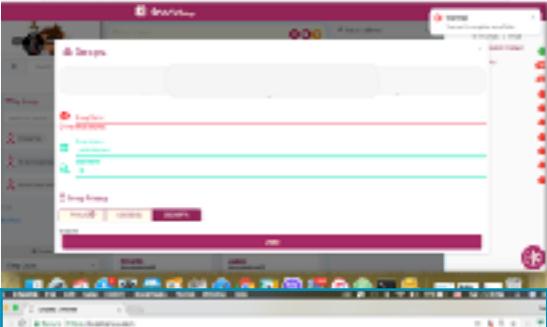
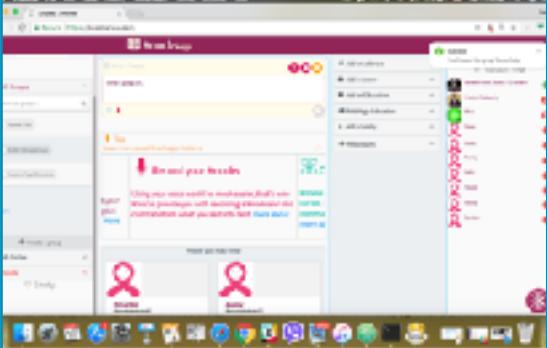
Input	Expected Output	Actual Output	Comment
Selected Comment : valid Reaction Type : valid But there is no internet connection	Not saving reaction		Success
Selected Comment : valid Reaction Type : valid	“Your reaction is added to this comment”		Success

## CHAPTER 5 : SYSTEM IMPLEMENTATION

- React to reply

Input	Expected Output	Actual Output	Comment
Selected Reply : valid Reaction Type : valid But there is no internet connection	Not saving reaction		Success
Selected Reply : valid Reaction Type : valid	“Your reaction is added to this reply”		Success

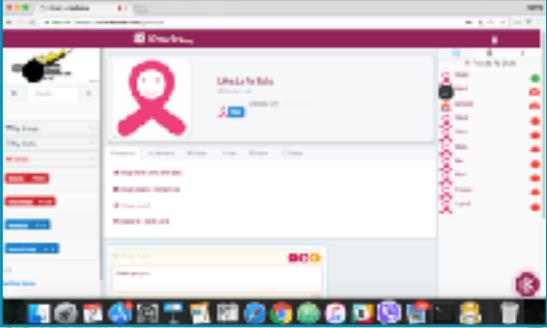
- **Create a group**

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

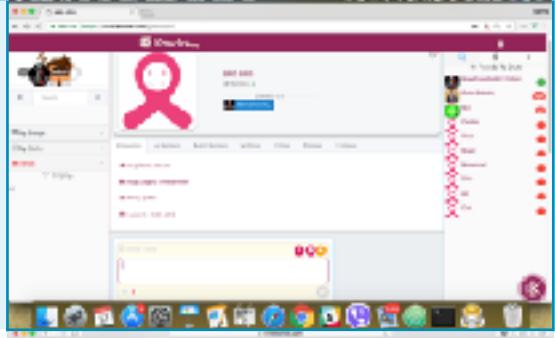
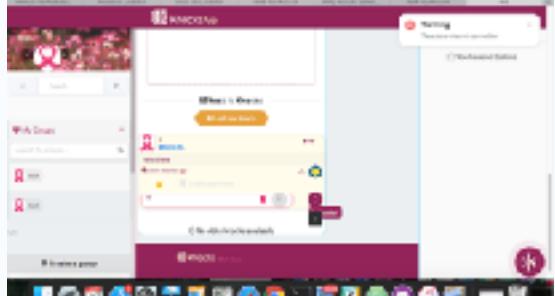
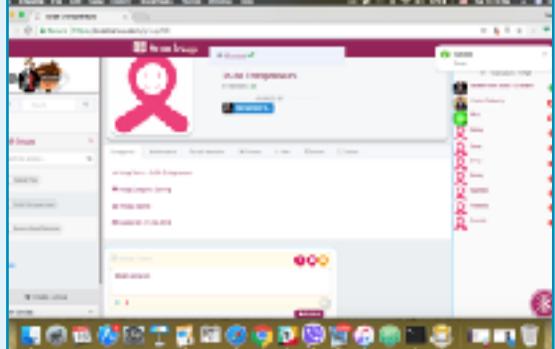
- **Add friends to group**

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 added to group ‘group name’ successfully”		Success

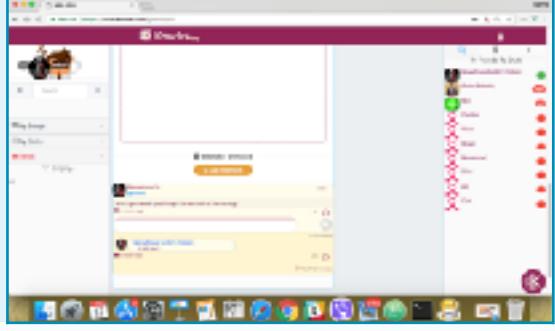
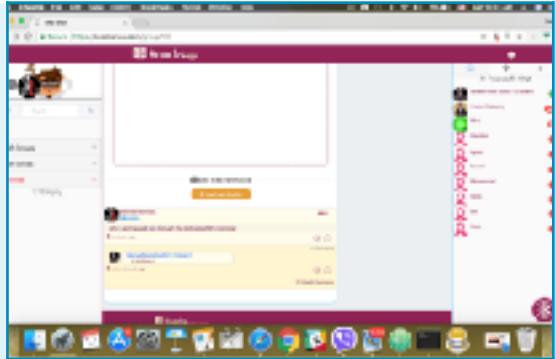
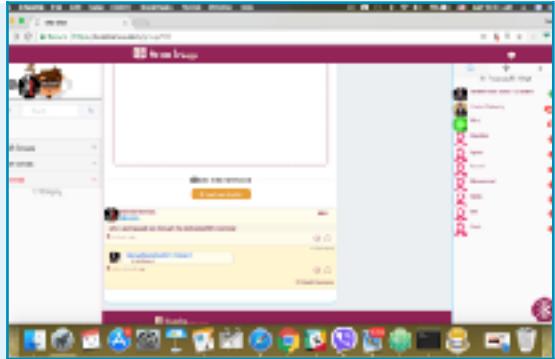
- **Update a group**

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	“You need to complete some fields”		Success
Group Name : valid Group Category : valid User Position : valid	“Group information is updated successfully”		Success

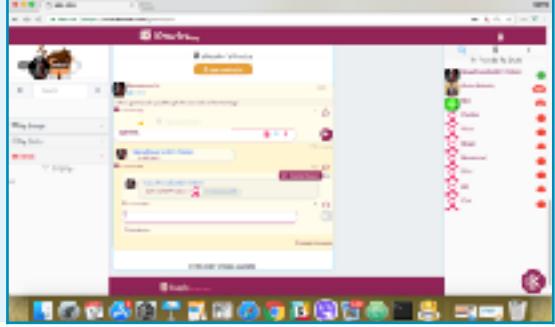
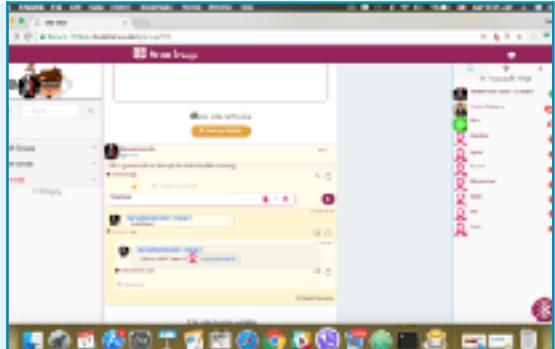
- Post in a group

Input	Expected Output	Actual Output	Comment
Selected Group : Valid Knock Content : Empty User Position : Valid	Disabled “Publish” Button		Success
Selected Group : Valid Knock content : Valid User Position : valid But there is no internet connection	“There is no Internet Connection”		Success
Selected Group : Valid Knock content : Valid User Position : valid	Enabled Publish Button, “Success”.		Success

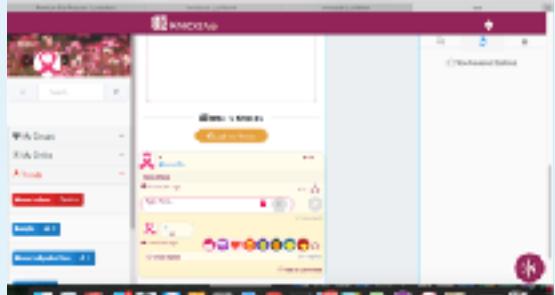
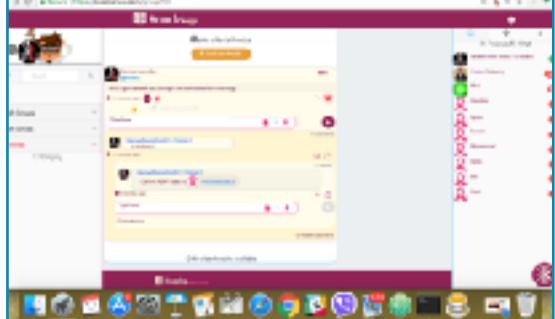
- **Comment in a group**

Input	Expected Output	Actual Output	Comment
Selected Group : Valid Comment Content : Empty User Position : Valid	Disabled “Publish” Button		Success
Selected Group : Valid Comment content : Valid User Position : valid But there is not internet connection	“There is no Internet Connection”		Success
Selected Group : Valid Comment content : Valid User Position : valid	Enabled Publish Button, “Success”.		Success

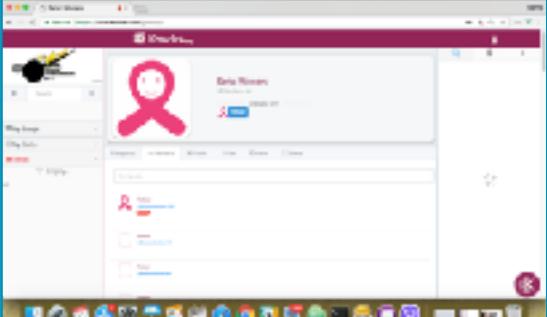
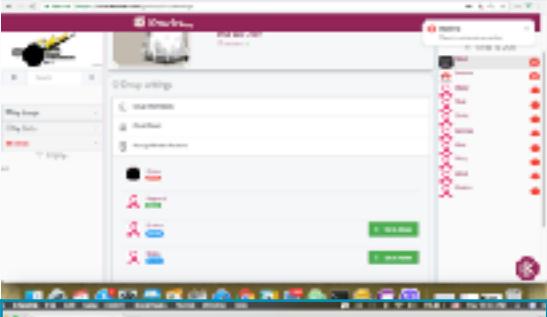
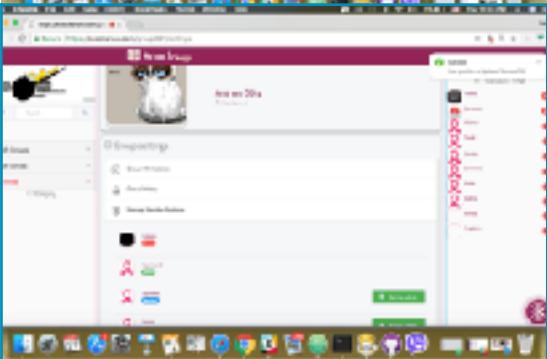
- **Reply in a group**

Input	Expected Output	Actual Output	Comment
Selected Group : Valid Reply Content : Empty User Position : Valid	Disabled “Publish” Button		Success
Selected Group : Valid Reply content : Valid User Position : valid But there is not internet connection	“There is no Internet Connection”		Success
Selected Group : Valid Reply content : Valid User Position : valid	Enabled Publish Button, “Success”.		Success

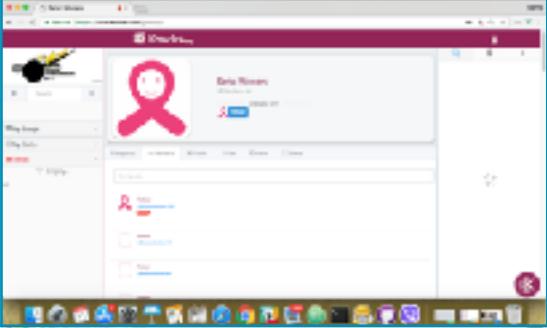
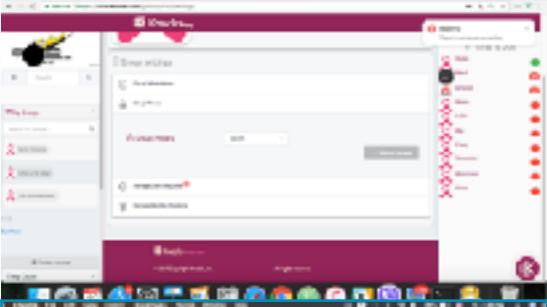
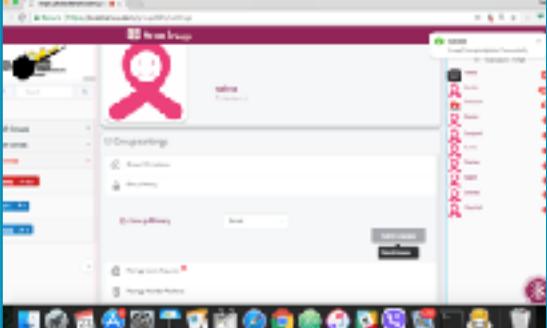
- React to an object a group (Eg : Post in a group )

Input	Expected Output	Actual Output	Comment
Selected Group : Valid User Position : Valid Selected Post : Invalid Reaction Type : Valid But there is no internet connection	not saving the reaction		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

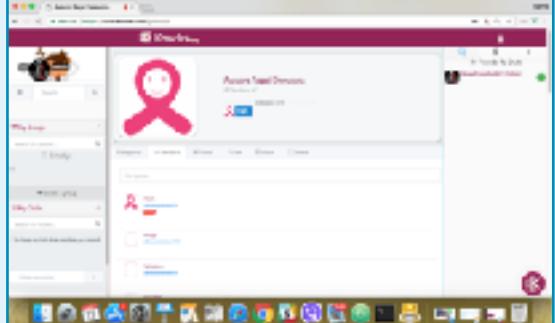
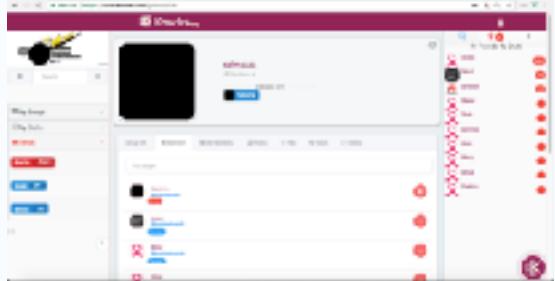
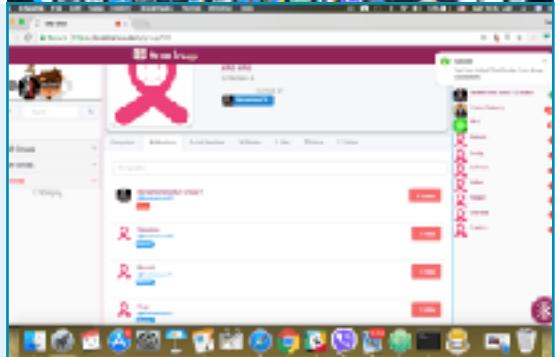
Input	Expected Output	Actual Output	Comment
Selected Group : Valid User Position : Invalid New Position : Valid	Not displaying the “Manage positions” Section		Success
Group Name : valid Group Category : empty User Position : Valid But there is no internet connection	“There is no Internet Connection”		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

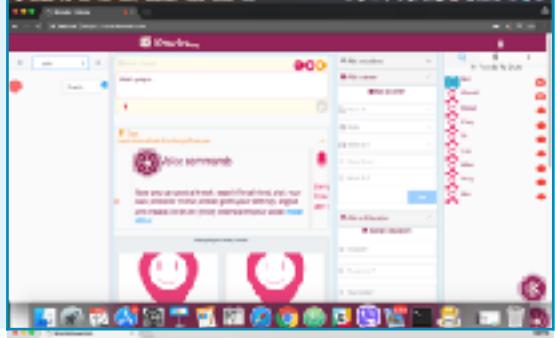
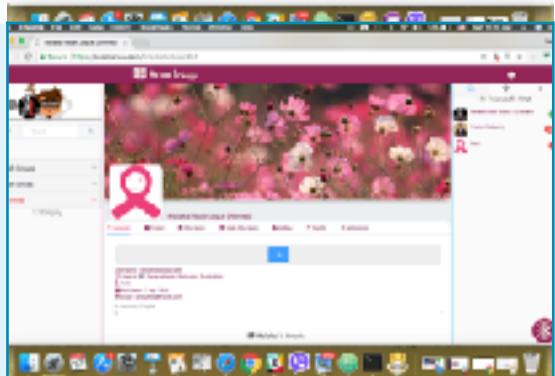
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- **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	Giving no response for kicking the member		Success
User Position : Valid Group Name : Valid Selected Member : Valid	“You have kicked this member from group successfully”		Success

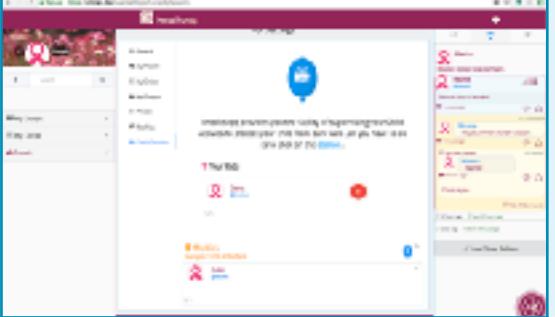
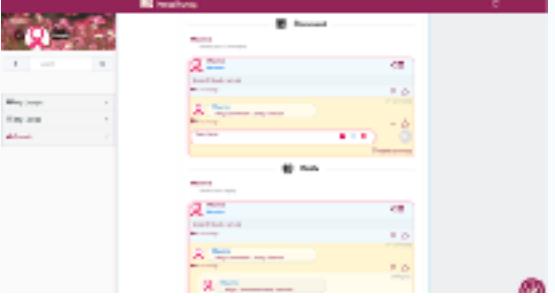
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- View a user profile

Input	Expected Output	Actual Output	Comment
Selected User : invalid (blocking friend)	Not displaying the user in the search bar		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

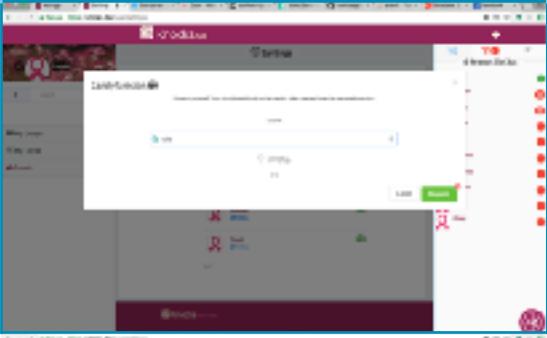
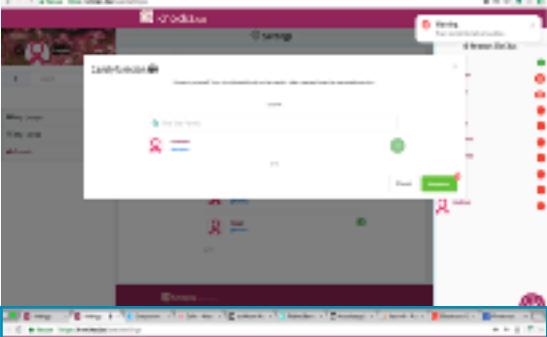
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Monitor objects for kid

Input	Expected Output	Actual Output	Comment
Parent : Valid Monitor mode : off	Not displaying the child's log		Success
Parent : Valid Monitor mode : off	Displaying child log (including friends, friend requests, groups, knocks, comment, and replies)		Success

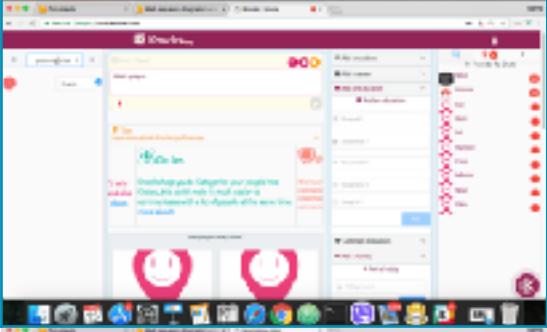
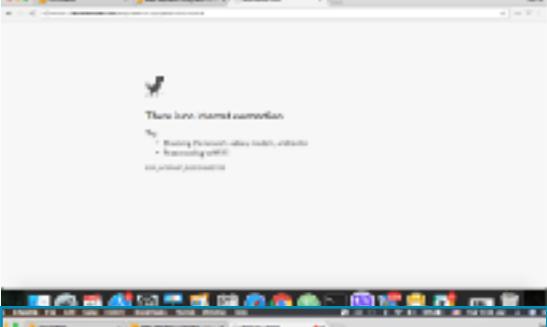
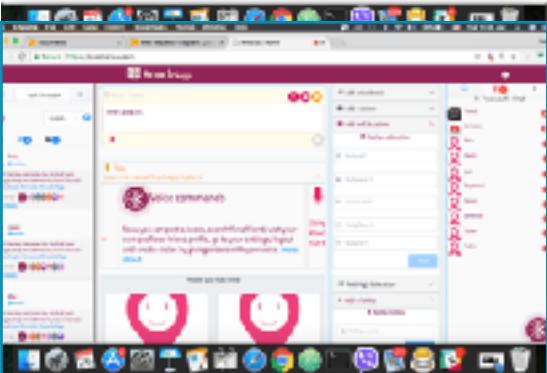
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Set a parent for a child

Input	Expected Output	Actual Output	Comment
Child : valid Parent : invalid ( not a friend )	Not displaying the parent in the search bar		Success
Child : valid Parent : valid, but there is no internet connection	"There is no Internet Connection"		Success
Child : valid Parent : valid	"Successfully", Sending request for the parent		Success

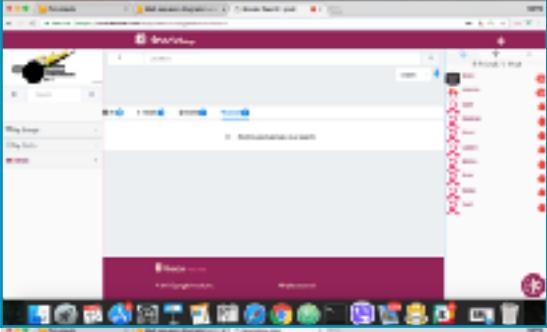
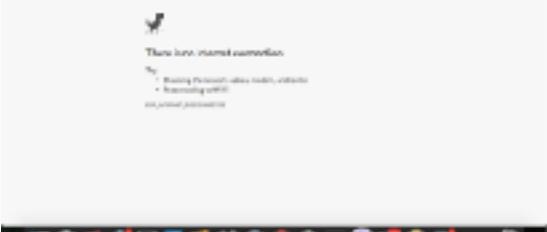
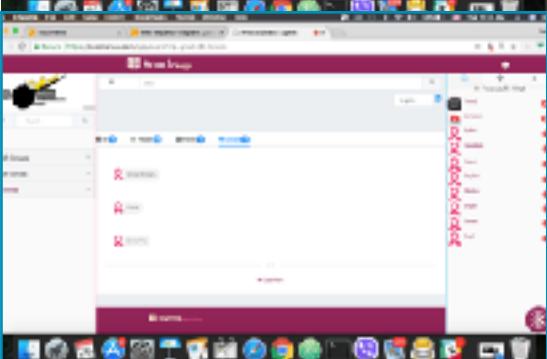
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- **Search for knock**

Input	Expected Output	Actual Output	Comment
Search key : Invalid ( Not having a knock with this key )	Not displaying any result for the user		Success
Search key : Valid But there is no internet connection	"There is no Internet Connection"		Success
Search key : Valid	Viewing the knocks that matched the search key		Success

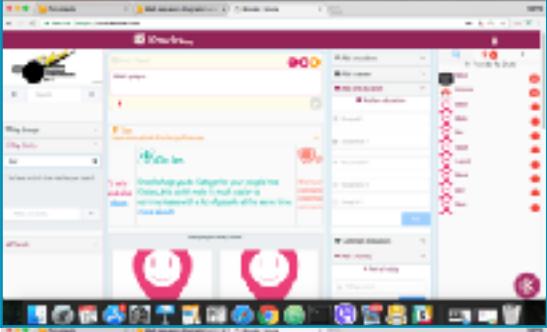
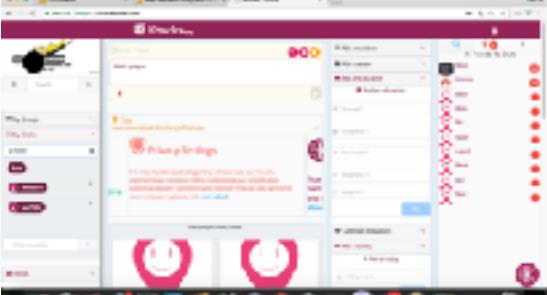
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- **Search for group**

Input	Expected Output	Actual Output	Comment
Search key : Invalid ( Not having a group with this key )	Not displaying any result for the user		Success
Search key : Valid But there is no internet connection	"There is no Internet Connection"		Success
Search key : Valid	Viewing the groups that matched the search key		Success

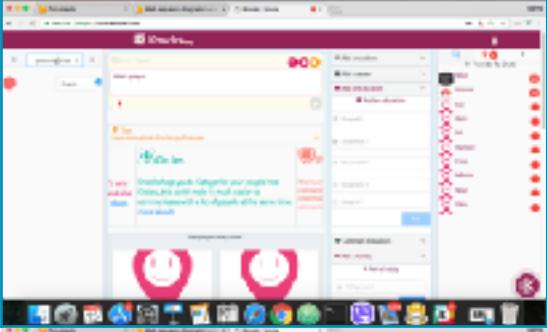
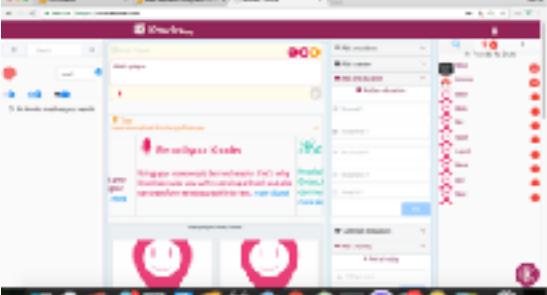
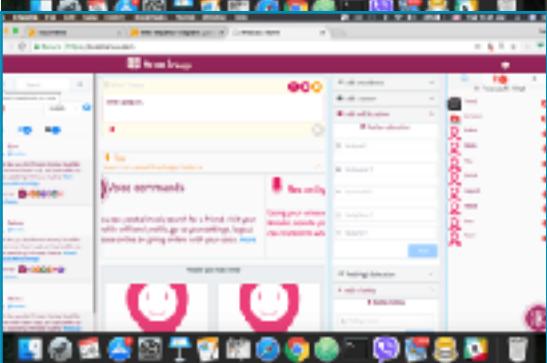
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- **Search for circle**

Input	Expected Output	Actual Output	Comment
Search key : Invalid ( Not having a circle with this result for the user key )	Not displaying any result for the user key		Success
Search key : Valid But there is no internet connection	not filtering the result or gives any response		Success
Search key : Valid	Viewing the circles that matched the search key		Success

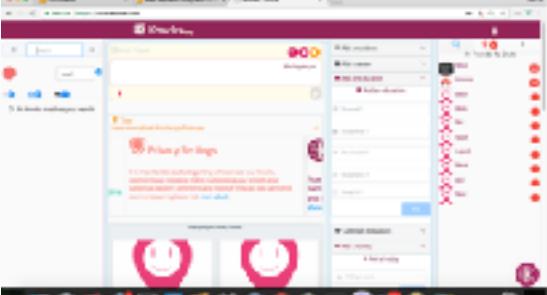
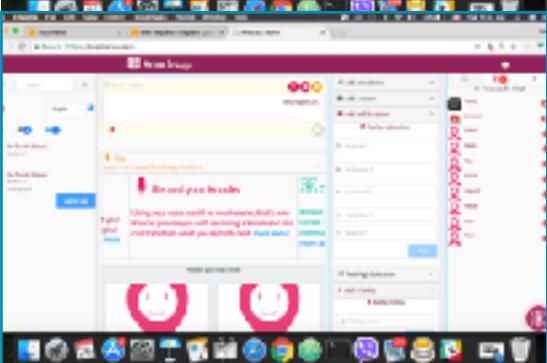
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Search by voice for knock

Input	Expected Output	Actual Output	Comment
Search key (voice) : Invalid ( Not having a knock with this key ) Language : valid	Not displaying any result for the user		Success
Search key : valid Language : invalid ( speaking in a language which is not the same as the selected )	"No knocks matches your search"		Success
Search key : Valid Language : valid	Viewing the knocks that matched the search key		Success

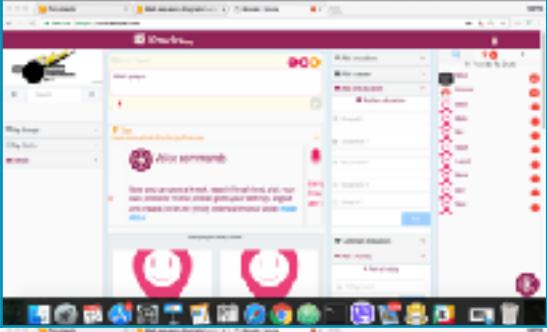
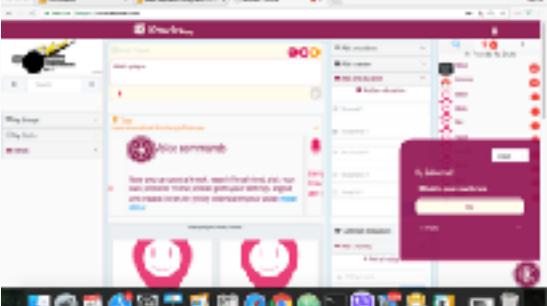
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Search by voice for group

Input	Expected Output	Actual Output	Comment
Search key (voice) : Invalid ( Not having a group with your search ) this key ) Language : valid	"No groups matches ( Not having a group with your search )"		Success
Search key : valid Language : invalid ( speaking in a language which is not the same as the selected )	"No groups matches your search"		Success
Search key : Valid Language : valid	Viewing the groups that matched the search key		Success

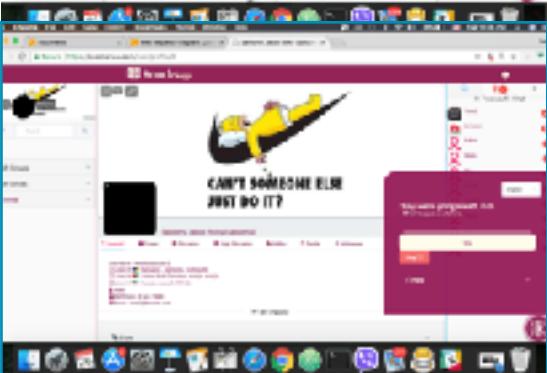
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Enabling and disabling voice command

Input	Expected Output	Actual Output	Comment
Command : invalid	Not responding to the speech		Success
Command : valid	enables and disable the voice commands depending on input		Success

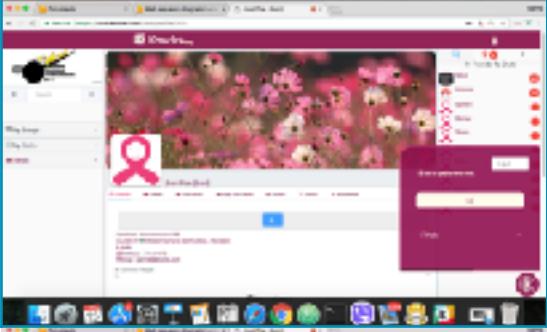
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- visiting profiles by voice commands

Input	Expected Output	Actual Output	Comment
Command : invalid ( in non understandable language ) User name : valid	Not responding to the speech		Success
Command : valid User name : invalid ( there is no user with such a name )	Speaking : "Excuse me !" For trying with correct name		Success
Command : valid User name : invalid	Viewing the user's profile		Success

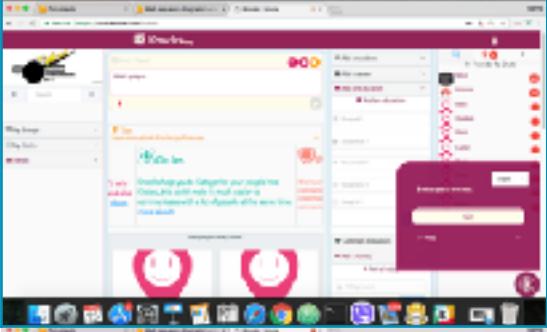
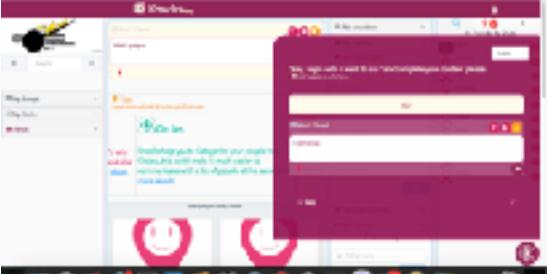
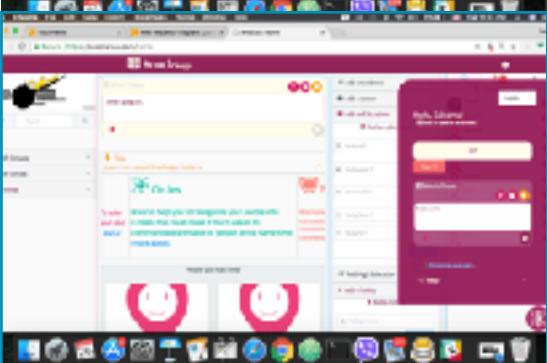
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Log out using voice commands

Input	Expected Output	Actual Output	Comment
Command : invalid	Not responding to the speech		Success
Command : valid	Logging out from account		Success

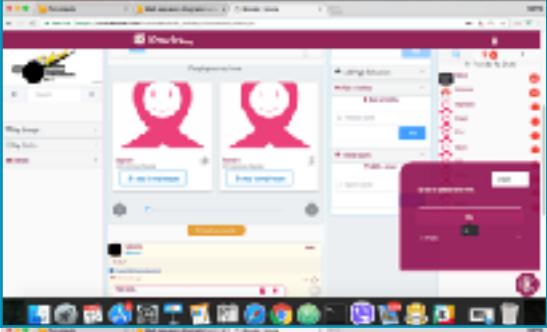
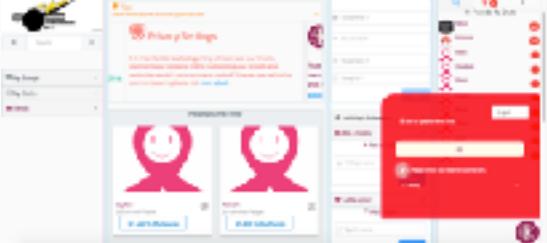
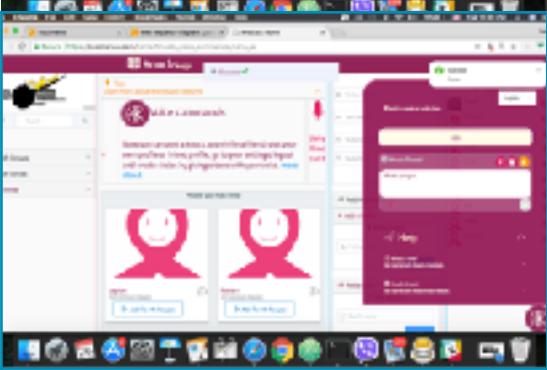
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Publishing a knock by voice commands

Input	Expected Output	Actual Output	Comment
Command : invalid ( in non understandable language ) Knock content : valid	Not responding to the speech		Success
Command : valid Knock content : invalid (Empty )	It will fill the knock with "i want to say " and wait for user confirmation		Success
Command : valid Knock content : valid	Publishing the knock		Success

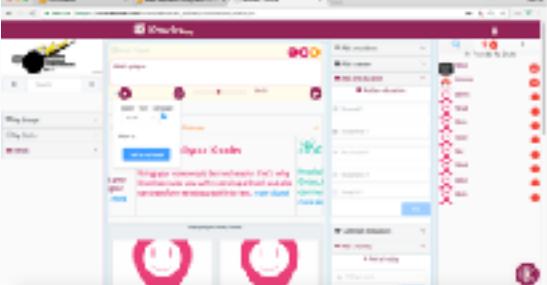
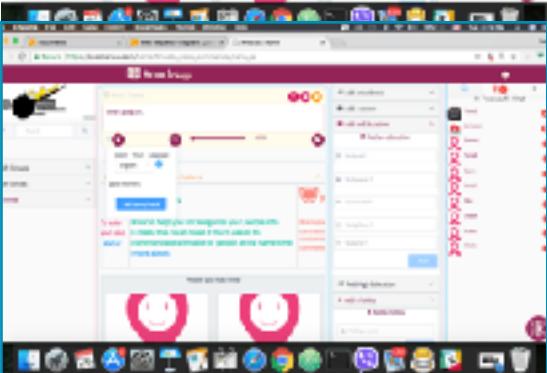
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Assisting the voice commands with text (Publish a knock )

Input	Expected Output	Actual Output	Comment
Command : invalid	Speaking : “Excuse me !” For trying with correct command		Success
Command : valid But there is not internet connection	“Please check your internet connection”		Success
Command : valid	Publishing the knock		Success

## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Converting the spoken voice into text

Input	Expected Output	Actual Output	Comment
Speech : invalid ( not matching with the selected language ) Language : valid	Giving wrong text based on the selected language		Success
Speech : valid Language : valid But there is not internet connection	It will not process on the listened sound		Success
Speech : valid Language : valid	Giving the right text based on speech		Success

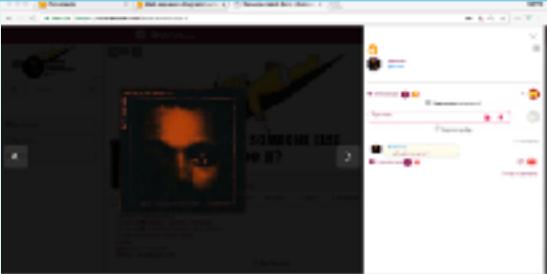
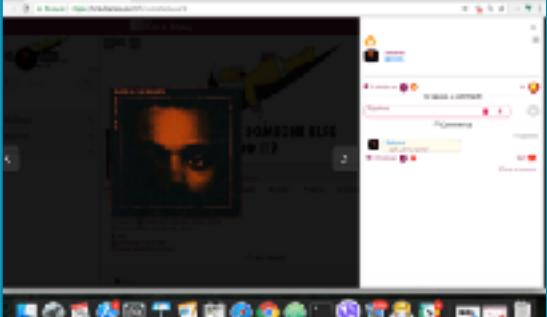
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- **Comment on photos**

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

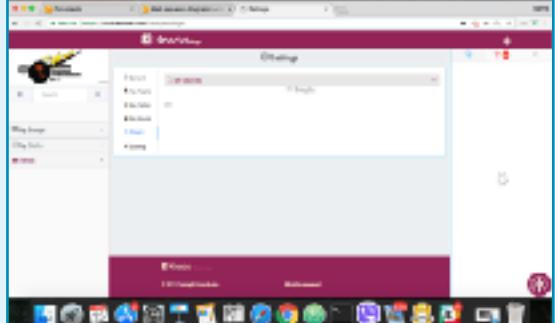
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- React on photos

Input	Expected Output	Actual Output	Comment
Selected photo : valid Reaction Type : valid But there is no internet connection	"There is no Internet Connection", not saving the chosen reaction		Success
Selected photo : invalid Reaction Type : valid	Displays your reaction across the photo		Success

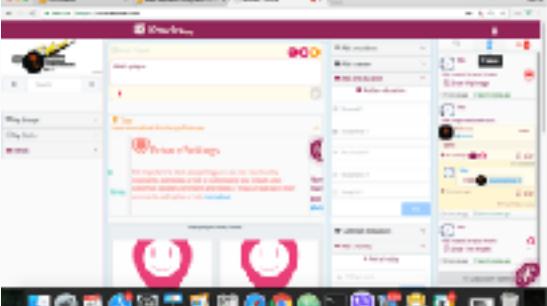
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Generating report for user's activities

Input	Expected Output	Actual Output	Comment
Internet connection : invalid	Not showing any details		Success
Internet connection : valid	Providing a user activity specifications		Success

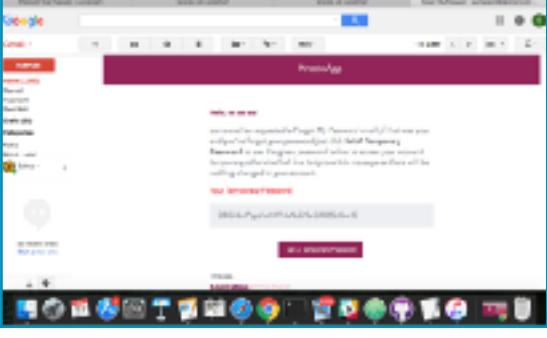
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Notifying the users

Input	Expected Output	Actual Output	Comment
Internet connection : invalid Object of notification : valid	Not showing any details		Success
Internet connection : valid Object of notification : valid	Providing a user activity specifications		Success

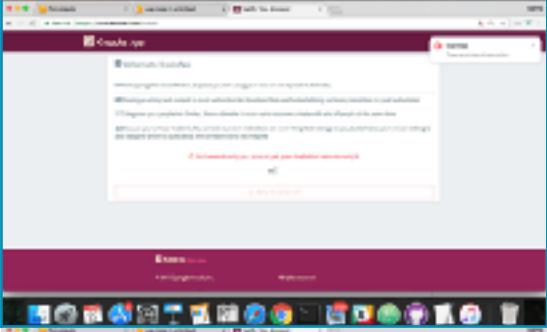
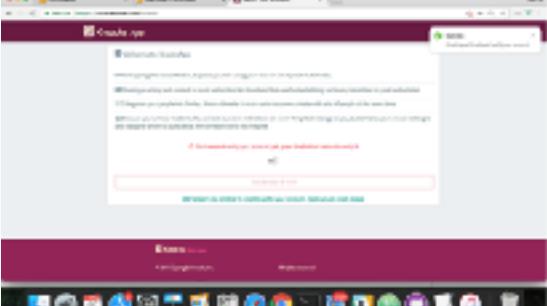
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Enabling Forgot my password mode

Input	Expected Output	Actual Output	Comment
e-mail : invalid	“another formula is expected, please try again”		Success
e-mail :valid But there is no internet connection	“There is no Internet Connection”		Success
e-mail :valid	“Success”, sending the email		Success

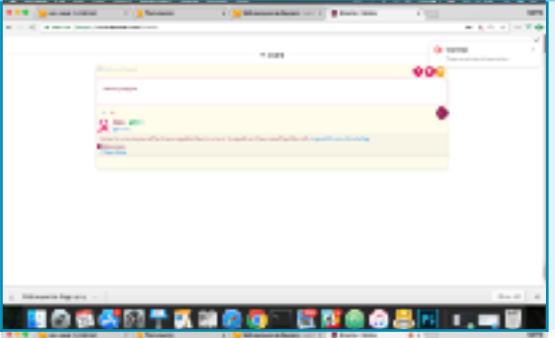
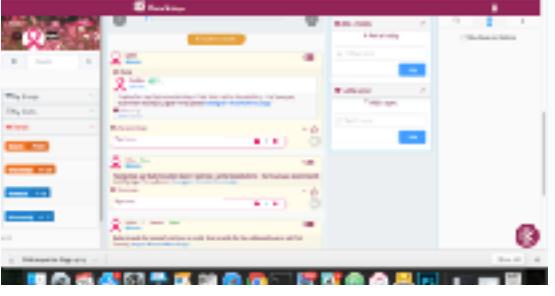
## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Request for e-mail verification

Input	Expected Output	Actual Output	Comment
No Internet connection	“There is no Internet Connection”		Success
Having internet connection	“Success”		Success

## CHAPTER 5 : SYSTEM IMPLEMENTATION

- Share Posts

Input	Expected Output	Actual Output	Comment
Knock : Available No Internet connection	“There is no Internet Connection”		Success
Knock : Available	“Success”		Success

# Chapter 7 : System Deployment

## 7.1.Tools Setup

### 7.1.1.Network Authentication

Once the system installation is done then the next step is to begin installing the packages

that server needs to run, installing such packages needs an internet connection.

Fedora server is coming with a console mode, which means that the admin needs to connect manually.

So first we need to check if there is a connected wireless devices attached to our server incase we desire to connect using wi-fi.

First we need to list our attached devices.

**IFCONFIG**

To show us the current status for our server connections and display any associated networks if existed.

As we still don't know what is the name of the server wifi device we need to list all the network devices that connected to our sever.

To list them we need to use the command:

**IFCONFIG -A**

You can find your wifi device is taking the name of:

wlan0 which is common for the Debian based systems (Debian , Kali Linux, Ubuntu).

wlp2s0 which is common for the redHat based systems (Fedora , CentOS , RHEL).

After finding your wireless interface you need to make sure whether the device is currently up or not.

You can do this using iwconfig library.

**IWCONFIG YOURINTERFACE**

In our case:

**IWCONFIG WLP2S0**

Then you need to know how to switch on and off your network devices So back to ifconfig library and use ifconfig youInterface desiredStatus. In our case:

**IFCONFIG WLP2S0 UP**

That is turning the device up, you can also make sure using iwconfig as we've mentioned, then you will need to find which network to connect.

You can do this using a several libraries.

## CHAPTER 7 : SYSTEM DEPLOYMENT

iwconfig offers listing and connection modules that is more specific than ever, it gives you a list for a specific device.

You can figure it out using:

**MAN -K IW**

That will list a lot of commands and stuff you can do using iw library.

As a result you will find iwlist which you can use to list all the seen networks for some device.

**IWLST WLP2S0**

That will return a lot of details about the network such as ESSID, signal strength, security protocols, and many more.

Then you can use the same library to connect to the network;

Just hit man iwlist and that will let you see all the possible things that you can do

So for scanning you can use:

**IWLST WLP2S0 SCANNING**

Then we can connect to any network using:

**IWCONFIG WLP2S0 NWESSID NETWORKNAME KEY NWPASSWORD**

Another simple way is by using NMCLI library which is making everything easier but showing less details.

Scan for the available networks using :

**NMCLI DEV WIFI**

And no need to mention what device name you need it to scan. And also  
nmcli dev wifi connect nwESSID password nwPassword

For associating with any network

The good news is that Fedora is including both NMCLI and iw as a built in module so  
need to install it as it already exists.

### 7.1.2.DNF Packages Manager

A dependency manager that validate installing packages, uninstalling and updating them, it also makes sure the the installed packages does not require something that is not exists.

As we need to install many external packages, we need a good dependency manager, Fedora offers 2 dependency managers, YUM and DNF.

## CHAPTER 7 : SYSTEM DEPLOYMENT

Both of them is great but DNF is being updated frequently more than YUM so DNF can guarantee installing the newest versions always as he is having a newer repositories.

So our very first thing will be updating the DNF packages and repositories.

DNF is a very powerful dependency manager that allows the user to update every single application on the whole system using only one command!

But before we begin updating our system there's something we need to focus on.

The server admin need to have the privilege to install or remove packages, as we've just made a fresh install we can login as 'root', this is a user that you must create while setting up the system and the root user is having all the privileges, but normally it's not recommended to work as a root, that means that you allow lots of things to happen without a permission, so using another created user will require the admin permission lots of actions which can make the admin always aware of what's going on.

So First Login as root, or use another user and in this case you need to begin your session with:

**SU**

which opens a root session or you can use 'sudo' as a prefix for most of commands. Then let's update our Fedora Server packages.

**DNF UPDATE**

You can hit dnf and it will list you many options such as upgrading the whole distro version and many more.

After updating the system it's better to restart the system as you can begin a fresh session with the new updates.

After restarting keep in mind to login as root or begin a session with 'su', DNF will not allow you to install any application without satisfying the administration privilege which is powerful to have one platform for packages installations and secure your system in the same time.

You'll also need to keep in mind your network status, hit 'ifconfig' to see whether you're connected to any network or not, if you're connected you will also be able to see the assigned IP address that your device takes in the local network and many more details, otherwise you'll need to reconnect using 'nmcli' or 'iwconfig'.

Before we start installing those packages we need to know that most of those packages creates its own configuration files on '/etc/' directory which makes it possible to edit in their settings from one directory.

Also Developers upload the code on GitHub and the server fetches these files on github then the composer web pack and npm builds the code of the developers.

### 7.1.3.Knocks Packages

- Apache HTTP Server And Routes Manager
- Composer PHP Dependency Manager
- MongoDB NoSQL Database
- Maria Database Server/Client MySQL Relational Database
- Laravel PHP MVC Framework
- GNOME GUI and Desktop environment
- PHP 7.1.8 Server side compiler working with Zend Engine
- NPM Javascript package manager
- phpMyAdmin
- Graphical database manager for MySQL
- Cockpit Server Resource Manager
- Mozilla Firefox Web browser includes development and testing tools
- Compass Graphical database for MongoDB

### 7.1.4.Packages Installation

First of all we need to focus on our installation sequence as we don't want to stuck in dependency flags.

So we'll begin with installing a packages that don't require any missing dependencies.

#### 7.1.4.1.HTTPD Apache Server

- ▶ DNF INSTALL HTTPD
- ▶ DNF SYSTEMCTL ENABLE HTTPD.SERVICE
- ▶ SYSTEMCTL START HTTPD.SERVICE

First we need to request the package from DNF.

After installing the package we need to turn it to a service so we can easily control when to start or stop it.

We do this by enabling httpd as a service then we can start it and also start the service with every server boot.

After enabling httpd as a service we can start/ stop/restart it normally with those commands

- ▶ SYSTEMCTL START HTTPD
- ▶ SYSTEMCTL STOP HTTPD
- ▶ SYSTEMCTL RESTART HTTPD

### 7.1.4.2.Maria DB (MySQL)

Installing MariaDB is very similar to httpd, that's because MariaDB is a service too.

MariaDB is having 2 main packages one for the server and the other's client package.

It could be enough for us to install the server package only but currently Knocks server is a platform for development too, so we need both of them.

- ▶ //CLIENT PACKAGE
- ▶ SUDO DNF INSTALL MARIADB
- ▶ //SERVER PACKAGE
- ▶ DNF INSTALL MARIADB-SERVER
- ▶ SYSTEMCTL ENABLE MARIADB
- ▶ SYSTEMCTL START MARIADB

So we did the same as we did with httpd for enabling and starting mariadb as a service, but for MariaDB this is not everything, we still need more configurations, as a database we can't use it without having a users, So we need to add a root user at least, and configure other things such as enabling or disabling the remote access and the anonymous users position and many more, MariaDB offers a script that assist you to configure those settings without going to the configuration file, so we need to run this script, luckily we don't even need to search for it and run it manually, DNF is making an alias for this script so you can execute it using the following command

- ▶ MYSQL\_SECURE\_INSTALLATION

Then You can test your connection by joining the database, you can call mysql globally from the terminal such as the following:

- ▶ MYSQL -UROOT -PYOURPASSWORD
- ▶ //THE OUTPUT SHOULD BE 'MYSQL >' OPENING A NEW SESSION

### 7.1.4.3.PHP

PHP is the 3rd in our installation sequence because each of the previous packages

(httpd, mariadb) are needed to make php works.

httpd is required for both MariaDB and PHP, for PHP both of them are required, because PHP is working with MySQL database and includes a modules for it already.

As a compiler, installing PHP is much easier than mariadb and httpd all we need is to install the package from DNF and he'll do the rest and copy the compiler binaries to the '/usr/bin' directory which usually includes the binaries for most of compilers.

So all we need to do is installing from DNF and specifying the last version as we need it for upcoming packages such as Composer and Laravel.

So we can do this using only one command and then its done!

#### ► DNF INSTALL PHP-CLI

That means that DNF will install the last stable version automatically for me.

Usually PHP is installing many modules automatically, so we don't need to install any modules for the moment.

Make sure That the installed version is 7.1.8 or greater using the following command:

#### ► PHP --VERSION ► //OR JUST PHP -V

This command is telling which version of PHP is installed on your system.

#### 7.1.4.4.Composer

Composer is a PHP dependency manager, it makes it easier than ever to install plugins and external modules for PHP.

Composer's documentation is pretty good to understand what can Composer do for your PHP, they also offers a packages gallery that we can easily search and browse for packages on [getcomposer.org](https://getcomposer.org).

As a dependency manager its very similar to DNF but only works for PHP, which means that we can install or remove PHP packages through Composer.

Obviously Composer is requiring PHP installed and lately its requiring a higher versions from PHP (  $\leq 7$ ), and as we've installed version 7.1.8 so we have no problem for this.

For installing Composer we can do it through DNF or we can use cURL project to do it, luckily cURL is already installed on Fedora so we can just use it.

if you are not familiar with cURL its a data transferring tool like a download manager but works extremely good with the command line environment, hit ‘curl -h’ for more.

The reason why we'll download it and install it manually instead of using DNF is that we want make sure that we're installing Composer globally.

- ▶ CURL -S [HTTPS://GETCOMPOSER.ORG/INSTALLER](https://getcomposer.org/installer) | PHP
- ▶ MV COMPOSER.PHAR /USR/LOCAL/BIN/COMPOSER
- ▶ CHMOD +X /USR/LOCAL/BIN/COMPOSER

What we've done here is downloading ‘composer.phar’ file from Composer's website, then we've moved it to lay on the local binaries directory so we can join composer globally, the last command is giving composer directory the executing privilege.

You can test your installation by running Composer's alias ‘composer’ or checking Composer's version.

- ▶ COMPOSER
- ▶ COMPOSER -V

#### 7.1.4.5.Laravel

Laravel is PHP MVC Framework that makes it easier for the development team to work using PHP, Laravel is coming with many modules built in such as ‘Laravel/Router’.

Laravel is supported by Composer and together they are totally helpful for the development team and saves a lot of time.

Installing laravel used to be easy as we only use Composer to make it happen. all we've to do is to require Laravel Package globally from Composer.

▶ `composer global require "laravel/installer"`

Unlike DNF, Composer does not verify Laravel after installation, and as we are going to use Laravel from the Terminal we need to verify Laravel paths to the system, and also create an alias for laravel so we can join it globally.

▶ `composer`  
▶ `composer -V`

So now we've added two lines to the bashrc file that makes it possible to join Laravel.

If you want to test hit 'laravel' which is the alias that we've just created, and then the system should recognize the alias and begin executing the binaries in the mentioned directory.

#### 7.1.4.6.MongoDB

MongoDB is an open source noSQL database which is widely known.

MongoDB is storing data as a JSON documents which makes it flexible with schemas and also makes it easier for the developers to work with it as they don't have to learn a new syntax. MongoDB is great with the big data in general, Knocks Development team intends to use it for storing and processing the media blobs as an objects.

Installing MongoDB has no dependencies, you can just go ahead and install it from DNF and its totally done!

► **DNF INSTALL MONGODB MONGODB-SERVER**

As a database we need to mention MongoDB as a service, so for testing and running MongoDB all you need to do is:

► **SERVICE MONGO START**

► **MONGO**

#### 7.1.4.7.NPM

NPM is a javascript dependency manager that contains many useful packages could boast up the development process.

NPM stands for Node Package Manager, Node is meaning NodeJS which we're going to install through NPM  
Installing NPM is totally done by DNF too.

So Next we'll Install NPM and use it to install another things such as MongoDB Compass, But first we need to install NodeJS which is a runtime server side language based on javascript

► **DNF INSTALL NODEJS**

► **DNF INSTALL NPM**

The default Version From NodeJS is running on Google V8 Engine which is exactly what we need.

#### 7.1.4.8.GNOME

GNOME is a graphical desktop environment that's works on most of Linux distros.

GNOME is a brilliant interface and light spaced on the disk and processor at the same time which is not achieved in many other GUIs such as KDE, KDE is more than the double of GNOME's space and a way heavier on the processor, However GNOME is installed with a switch

which means that the admin can turn it on and off at anytime which is very flexible.

GNOME isn't actually one package, its a group package that comes with graphical directory browsers, and other regular utilities that will only run under GNOME as a process.

Installing GNOME is very easy by DNF, you don't need to mention every member in the group package, all you need to do is:

▶ **DNF GROUPINSTALL GNOME**

That means that DNF will handle installing and verifying all of those packages. And then it's time to get out from the console mode

▶ **STARTX**

And the X-Based system that we've just installed will start GNOME. After installing GNOME now we can install and test all the graphical packages that we've mentioned.

#### **7.1.4.9.Mozilla Firefox**

Mozilla Firefox is a powerful web browser that contains many development, measuring, and testing tools.

Installing Mozilla Firefox is being done by DNF as the repository of it is already mentioned.

▶ **DNF INSTALL FIREFOX**

Once the installation is complete you should see the application's icon in GNOME menu.

#### **7.1.4.10.phpMyAdmin**

phpMyAdmin is a graphical tool for the relational database MySQL that makes it easy for the developers to browse their tables and represent the schematics easier.

phpMyAdmin is a web application and it only requires:

(httpd, mariadb, PHP, php-mysqli) and all of them should be covered in the current stage.

Installing phpMyAdmin is very simple all we need to do is installing the package from DNF and DNF will do the rest.

► **DNF INSTALL PHPMYADMIN**

Then you can join it locally on <http://127.0.0.1/phpMyAdmin>

#### **7.1.4.11.MongoDB Compass**

MongoDB Compass is a graphical tool for the noSQL database MongoDB, its ranked as the best among all the alternative tools, Compass is not only a query platform, it includes many statistical terminals that assists the developer to know much a way easier.

For Compass you can head to <https://mongodb.com/compass> and download it using your browser, Then you can go to the downloaded file and just execute it.

Once the installation is complete you should see the application's icon in GNOME menu.

#### **7.1.4.12.Cockpit**

Cockpit is a resource manager for Fedora Server that includes many statistical terminals to show the admin many details about the server performance, storage, network status and many more.

Installing Cockpit is being done by DNF but Cockpit still needs some configuration.

- **DNF INSTALL COCKPIT**
- **SYSTEMCTL ENABLE COCKPIT SOCKET**
- **//--NOW FLAG COULD BE AN OPTION**
- **FIREWALL-CMD --ADD-SERVICE=COCKPIT**
- **FIREWALL-CMD --ADD-SERVICE=COCKPIT --PARMANENT**

Enabling Cockpit and verifying it to the fire wall is necessary, but its optional to determine whether you want it to start with every boot or not.

### 7.1.5.Knocks Installation Script

```
▶ SU
▶ DNF UPDATE
▶ DNF INSTALL HTTPD
▶ DNF SYSTEMCTL ENABLE HTTPD.SERVICE
▶ SYSTEMCTL START HTTPD.SERVICE
▶ SYSTEMCTL RESTART HTTPD
▶ SUDO DNF INSTALL MARIADB
▶ DNF INSTALL MARIADB-SERVER
▶ SYSTEMCTL ENABLE MARIADB
▶ SYSTEMCTL START MARIADB
▶ MYSQL_SECURE_INSTALLATION
▶ SYSTEMCTL RESTART HTTPD
▶ DNF INSTALL PHP-CLI
▶ SYSTEMCTL RESTART HTTPD
▶ CURL -Ss HTTPS://GETCOMPOSER.ORG/INSTALLER | PHP
▶ MV COMPOSER.PHAR /USR/LOCAL/BIN/COMPOSER
▶ CHMOD +X /USR/LOCAL/BIN/COMPOSER
▶ COMPOSER GLOBAL REQUIRE "LARAVEL/INSTALLER"
▶ ECHO 'EXPORT PATH="$PATH:$HOME/.CONFIG/COMPOSER/VENDOR/BIN"'>> ~/.BASHRC
▶ ECHO 'ALIAS LARAVEL="~/.COMPOSER/VENDOR/BIN/LARAVEL" >> ~/.BASHRC
▶ DNF INSTALL MONGODB MONGODB-SERVER
▶ DNF INSTALL NODEJS
▶ DNF INSTALL NPM
▶ DNF GROUPINSTALL GNOME
▶ DNF INSTALL FIREFOX
▶ DNF INSTALL PHPMYADMIN
▶ SYSTEMCTL RESTART HTTPD
▶ DNF INSTALL COCKPIT
▶ SYSTEMCTL ENABLE COCKPIT SOCKET
▶ FIREWALL-CMD --ADD-SERVICE=COCKPIT
▶ FIREWALL-CMD --ADD-SERVICE=COCKPIT --PARMANENT
▶ SYSTEMCTL RESTART HTTPD
▶ REBOOT
```

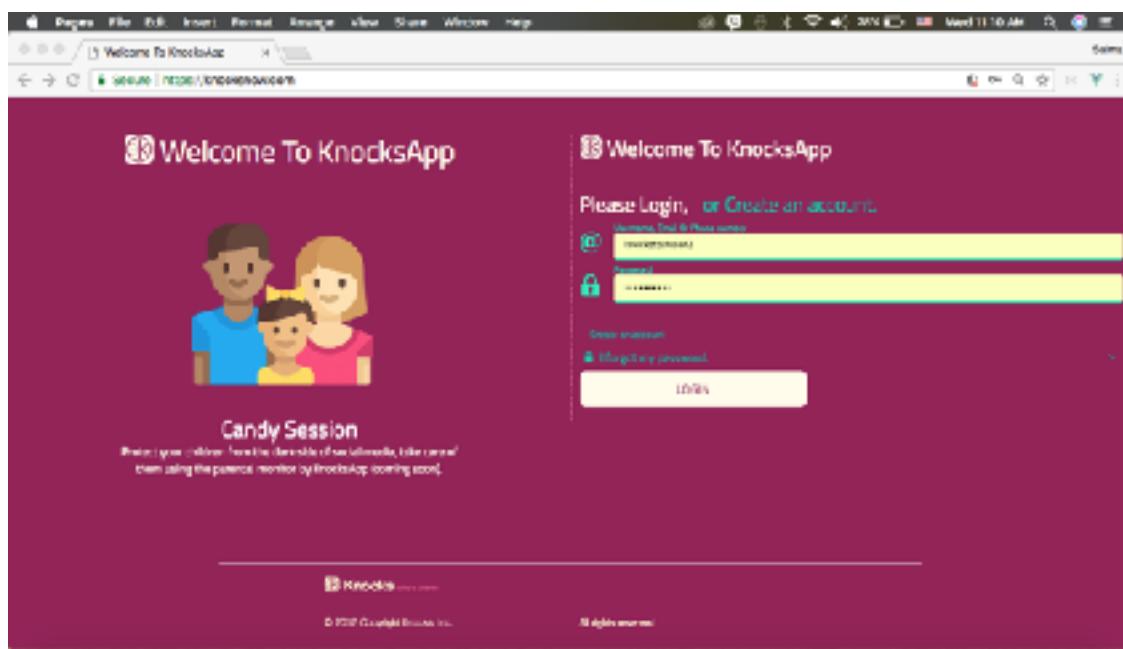
# Chapter 8 : User Manual

.

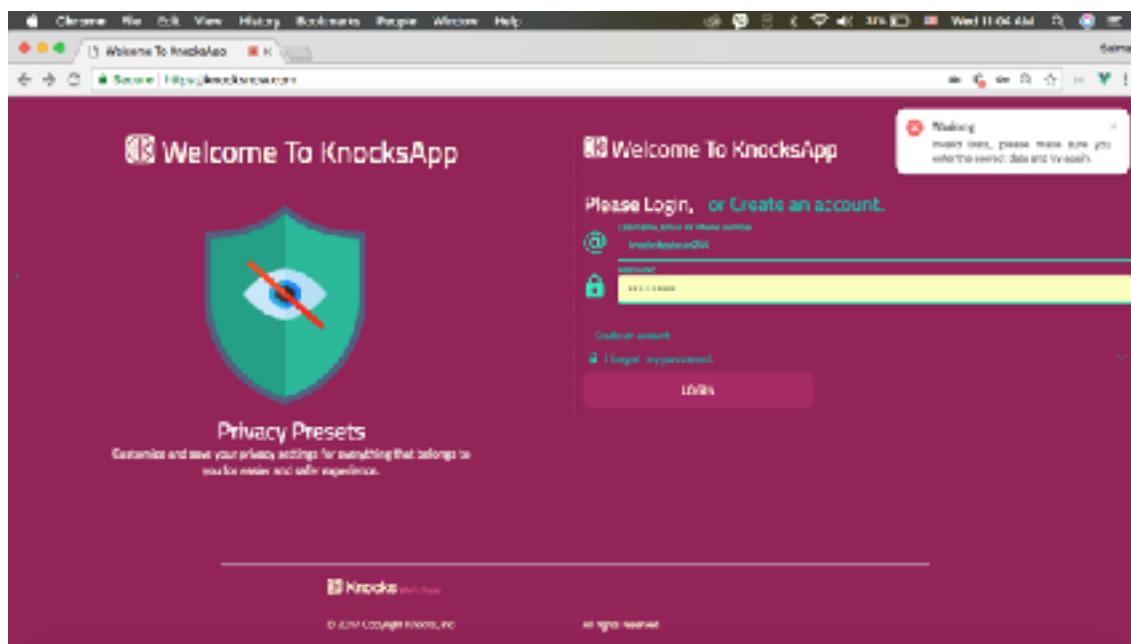
## CHAPTER 8 : USER MANUAL

### • 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”

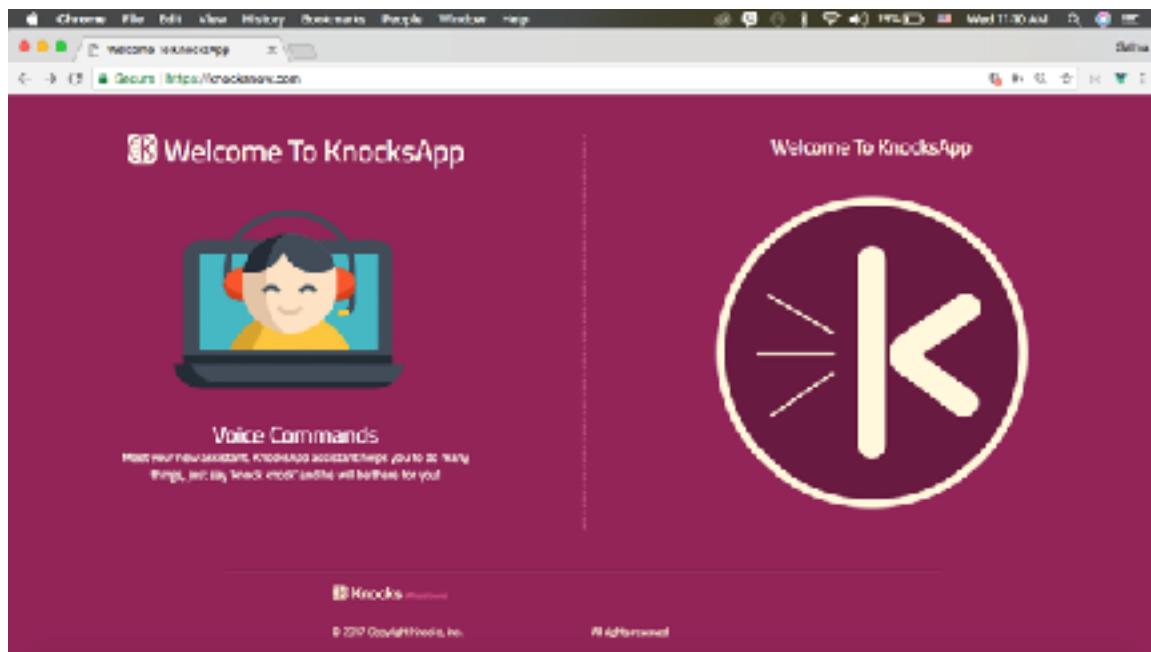


If the date are incorrect, the system will display a message “Invalid Data, Please make sure you enter the correct data and try again”



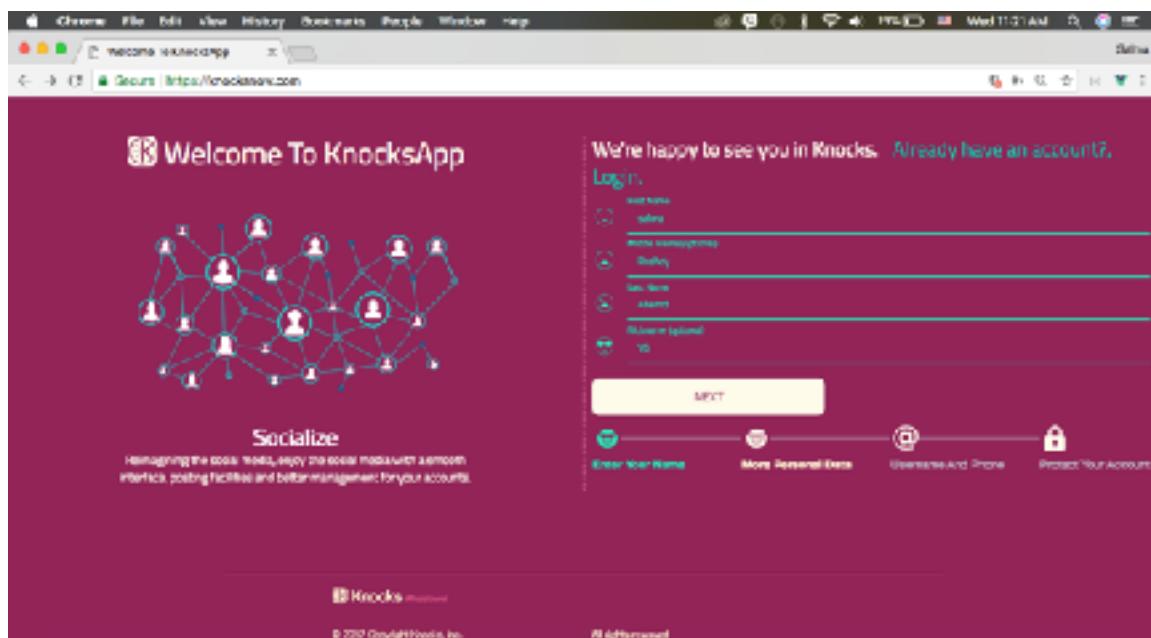
## CHAPTER 8 : USER MANUAL

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



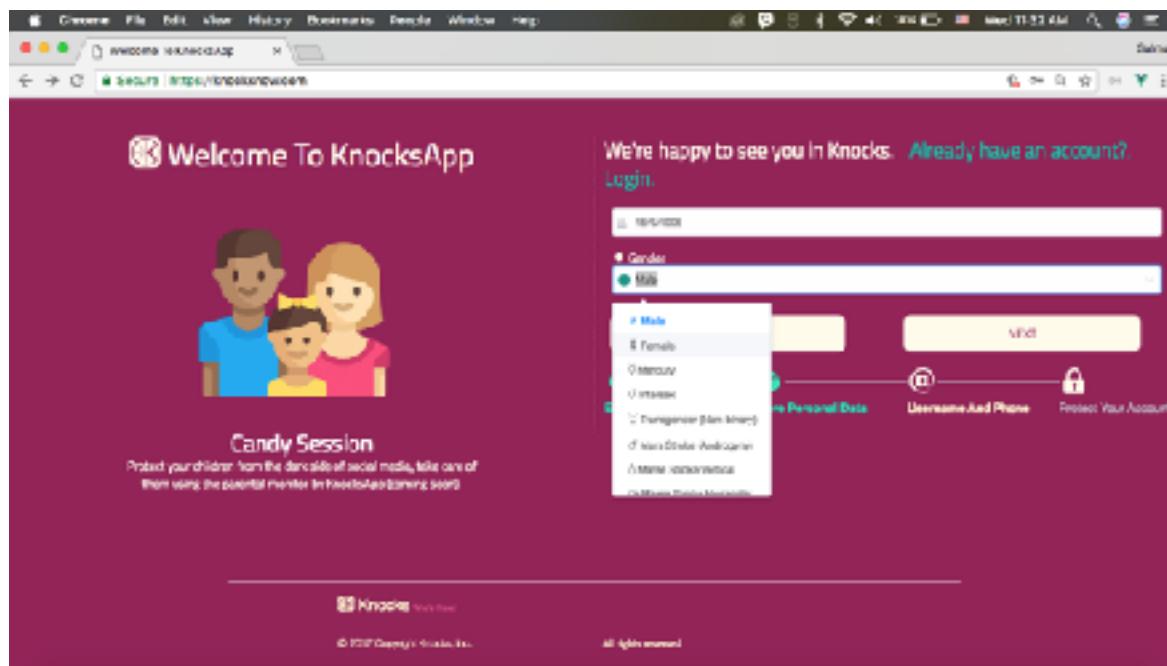
- **Sign up**

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

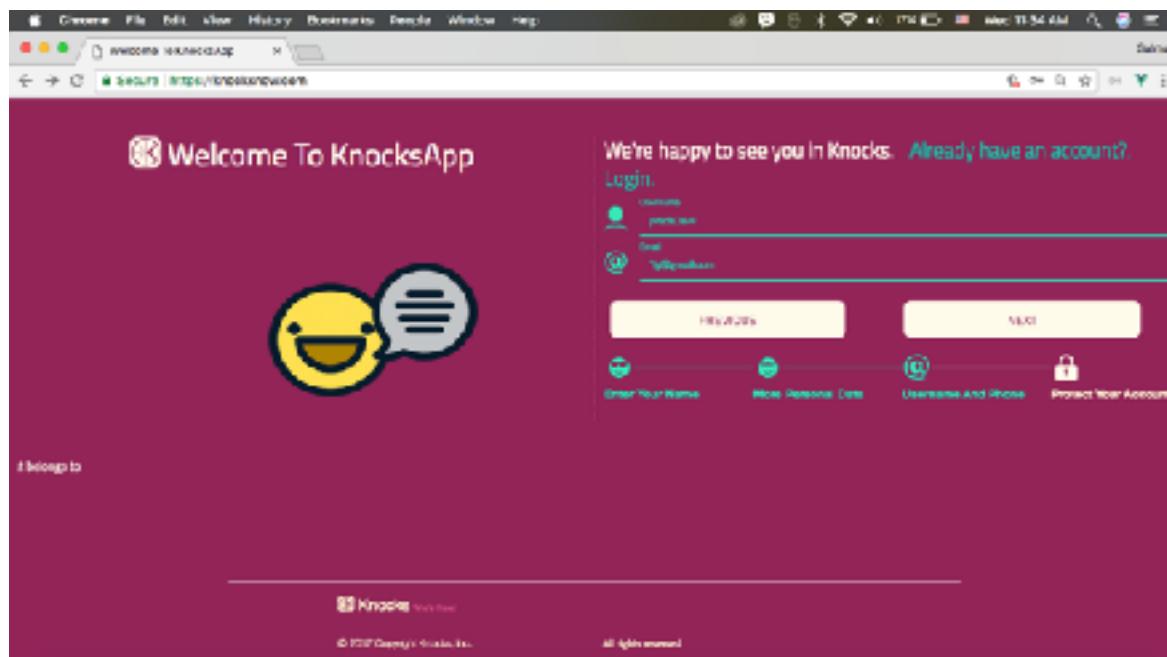


## CHAPTER 8 : USER MANUAL

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.

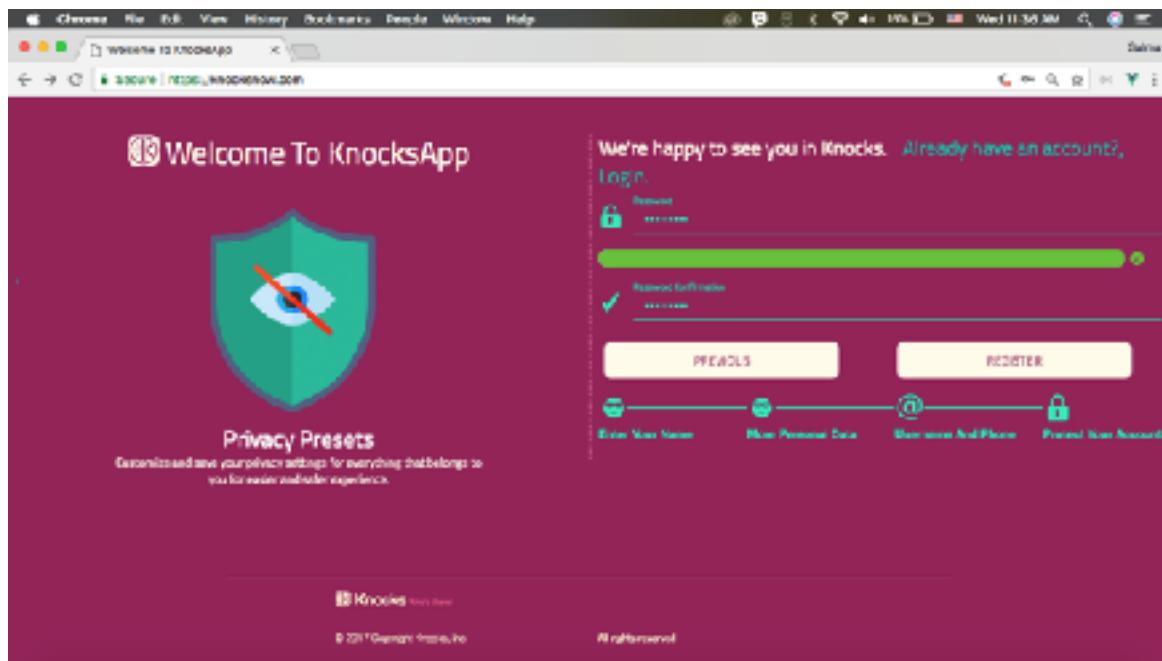


Then the user provides the user name ( which must be non used by other users ) and the e-mail which must be valid one then clicks on “NEXT” to finish third stage of sign up or “PREVIOUS” to get back to previous stage for editing.

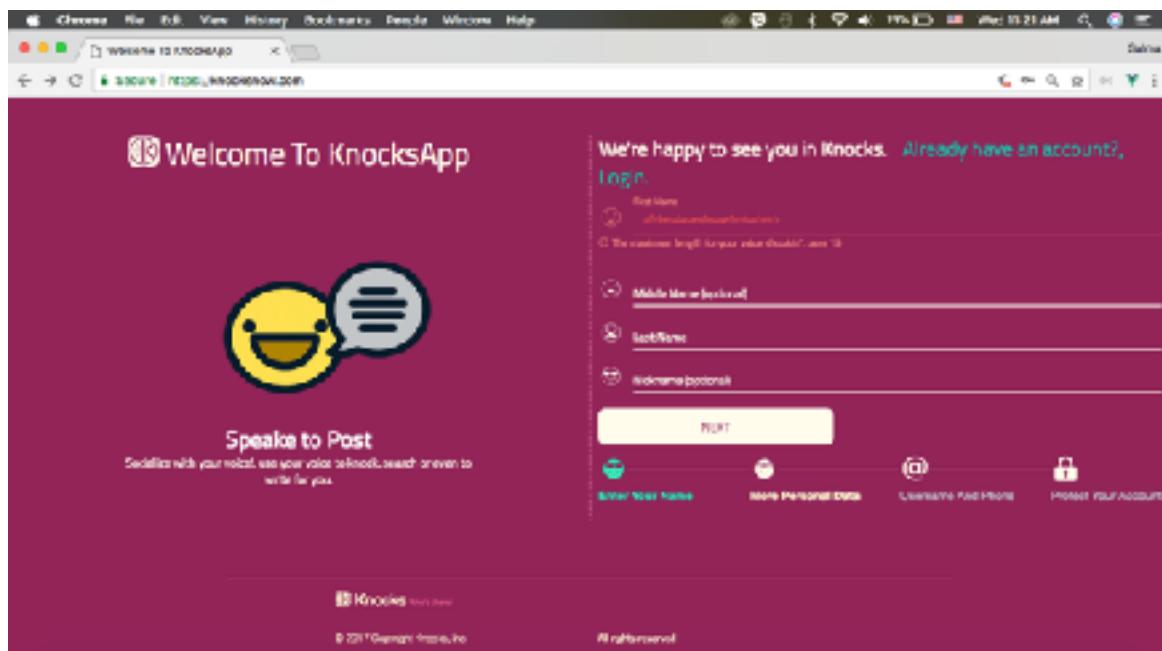


## CHAPTER 8 : USER MANUAL

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.

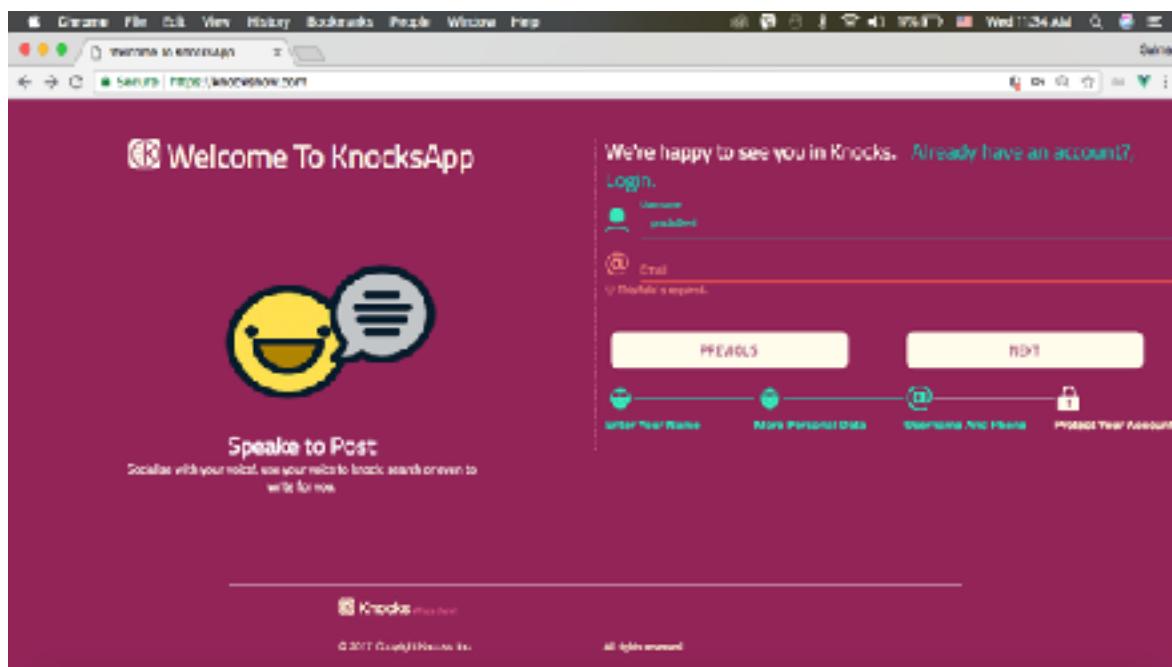


If the date are incorrect, for example providing incorrect first name, the system will display a message “The maximum length of the first name should not pass 15 characters ”

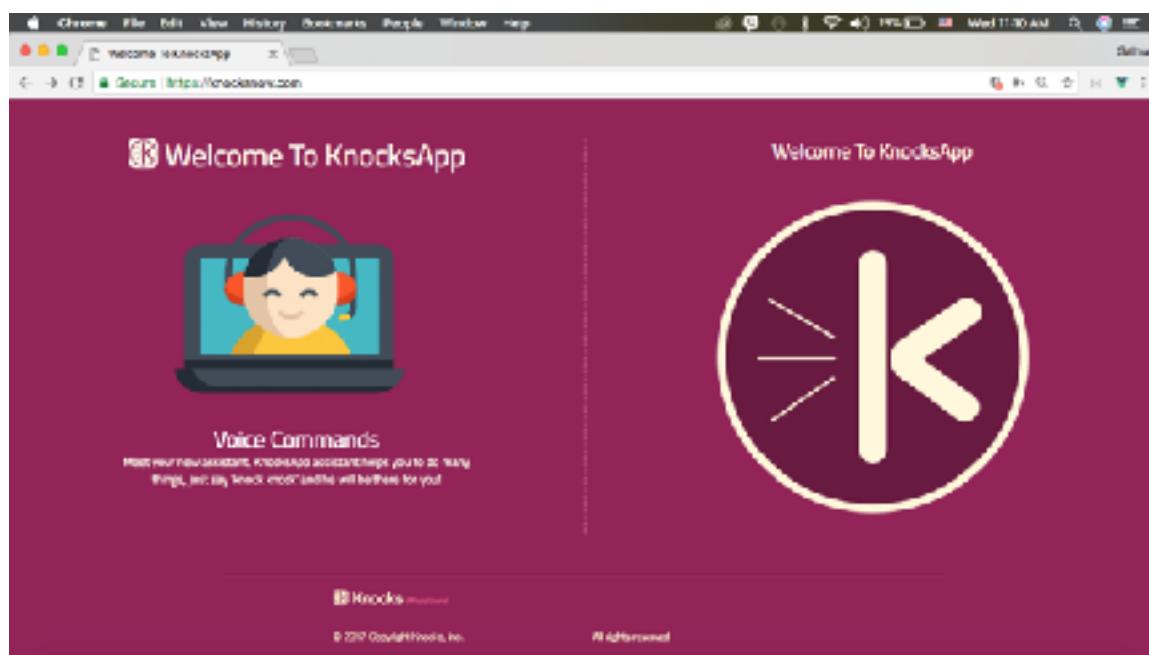


## CHAPTER 8 : USER MANUAL

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



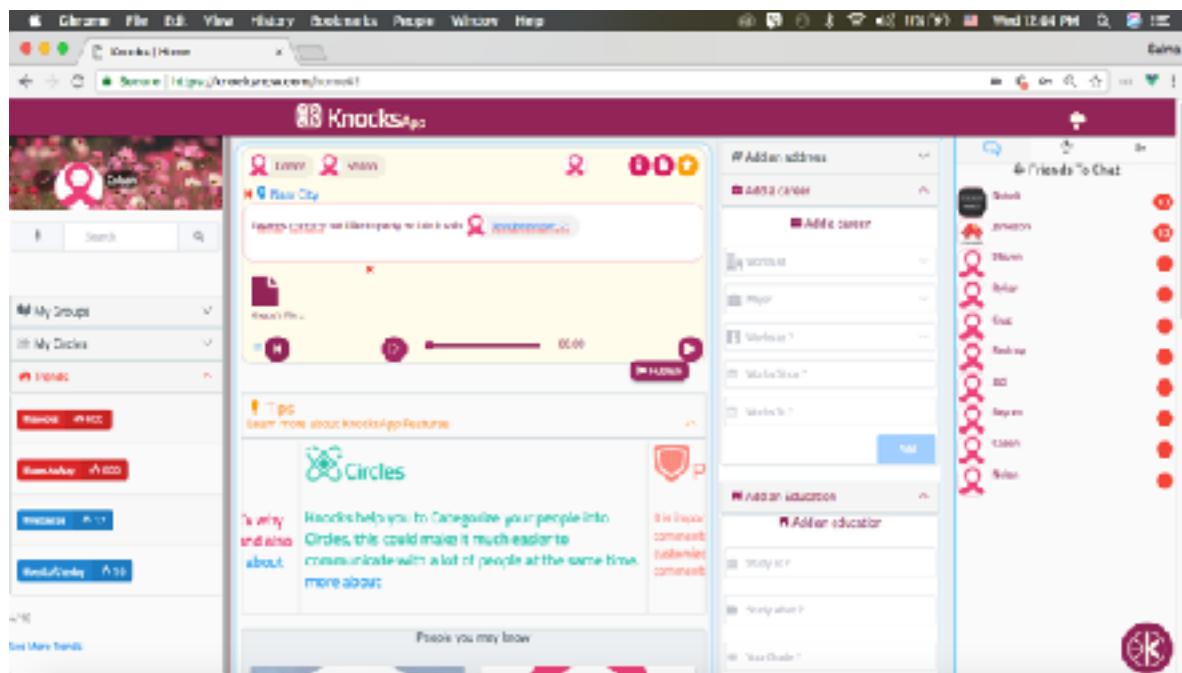
Else the system will display message “Welcome to KnockscApp” and system direct the user to his home page.



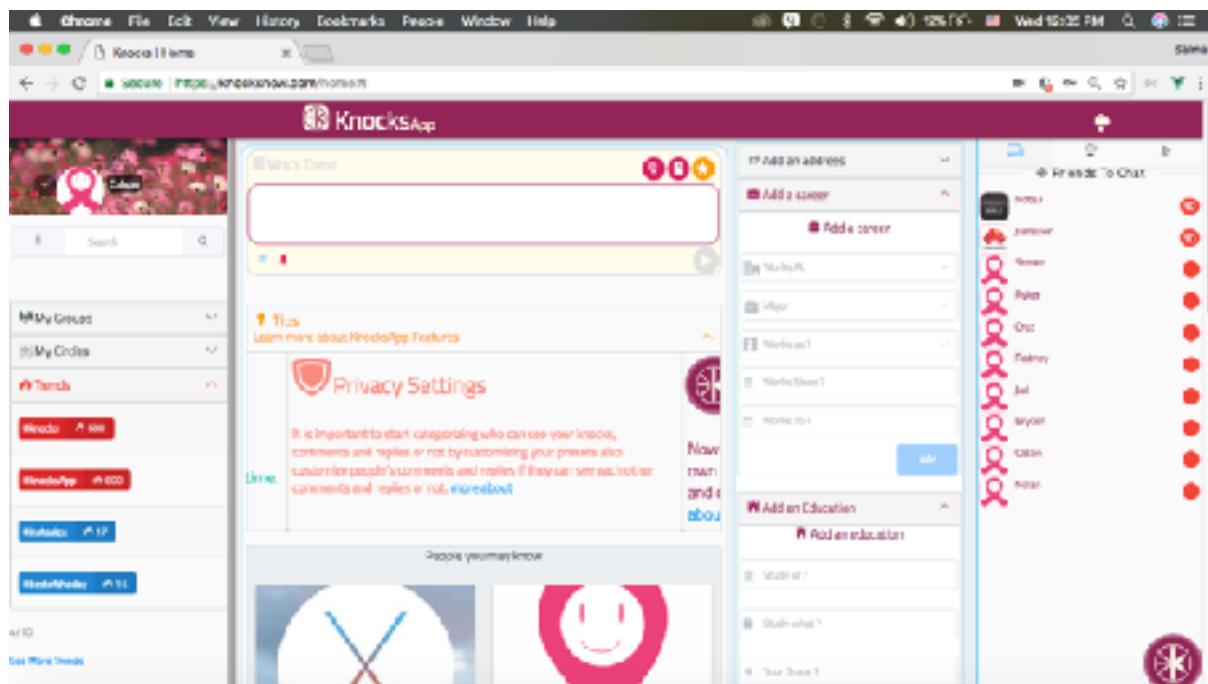
## CHAPTER 8 : USER MANUAL

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

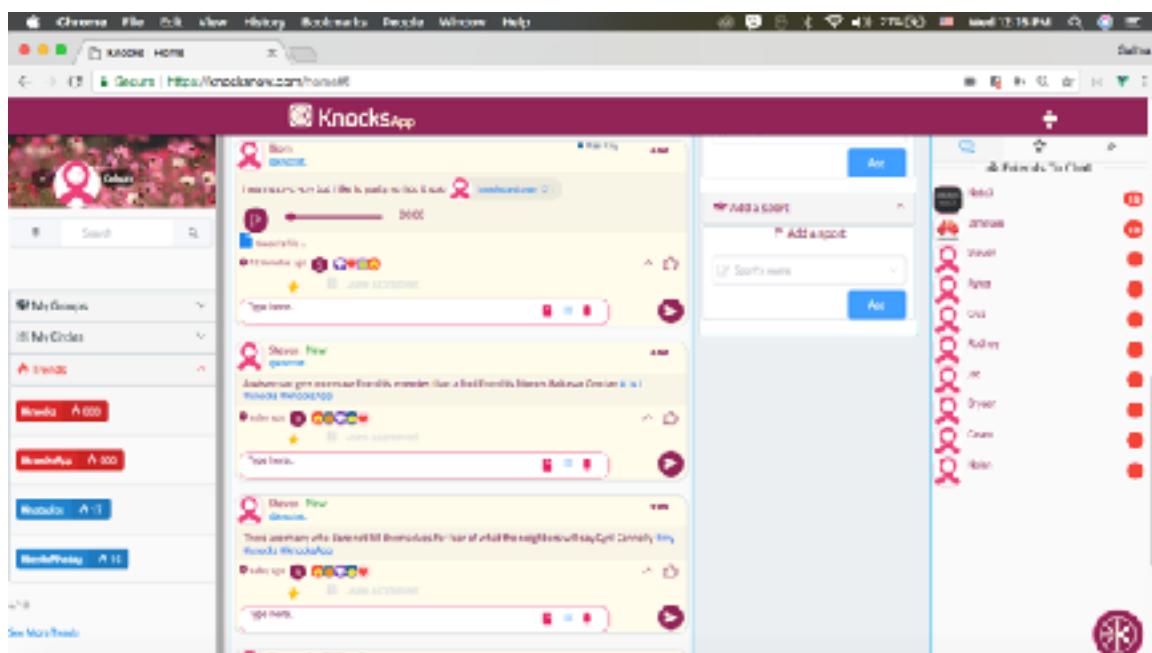
The user should provide text, check in, voice note, tagging a friend, uploading a file, voice to be converted into text or combination between them, then clicks on “Publish”



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

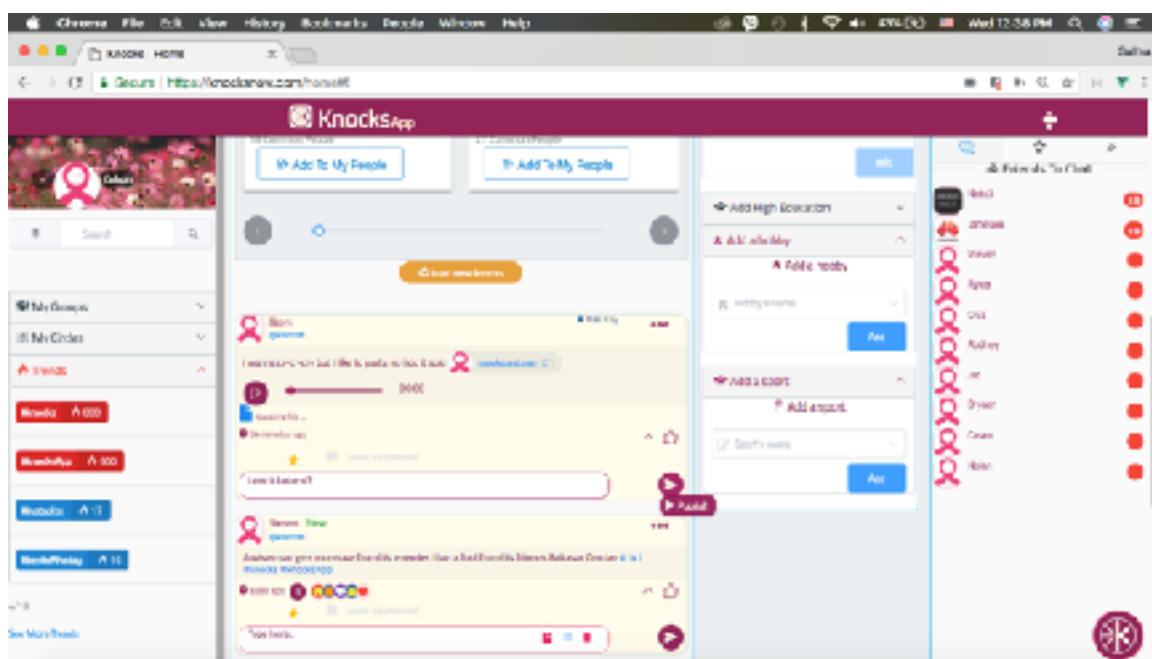


Else the system will display the knock at the user's time line



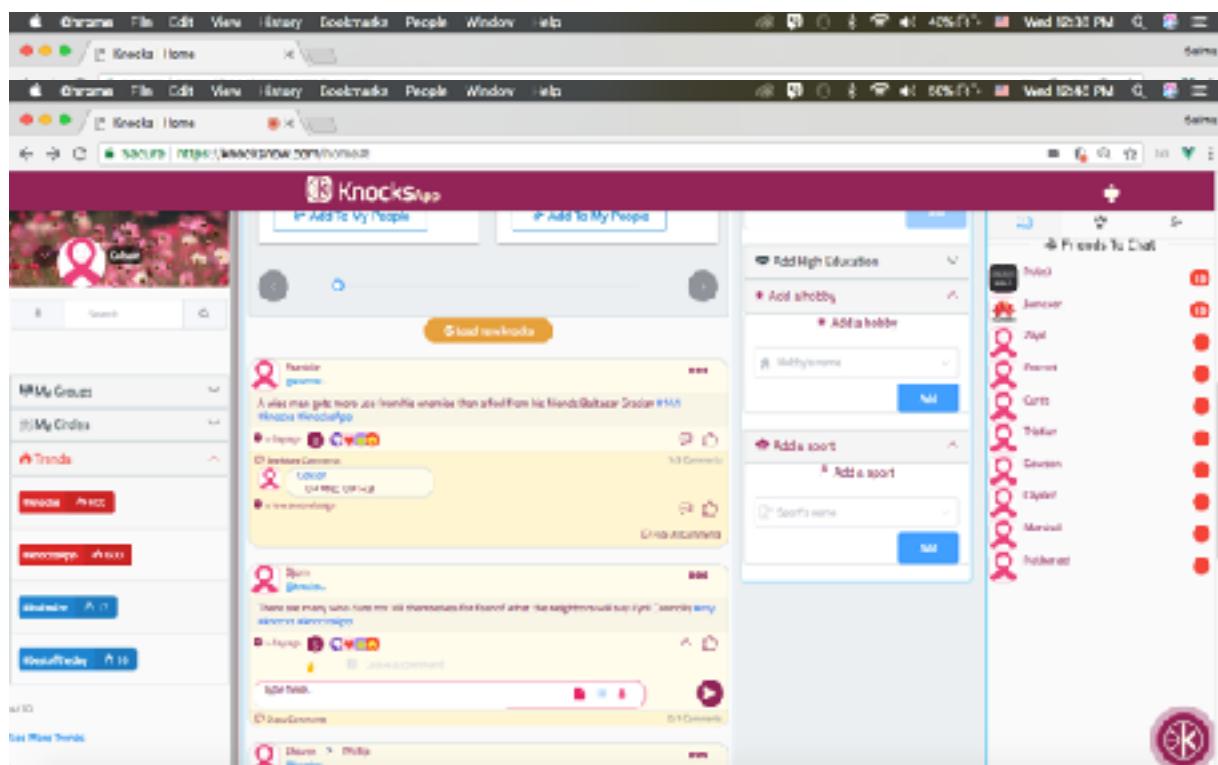
- **Write a comment**

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”

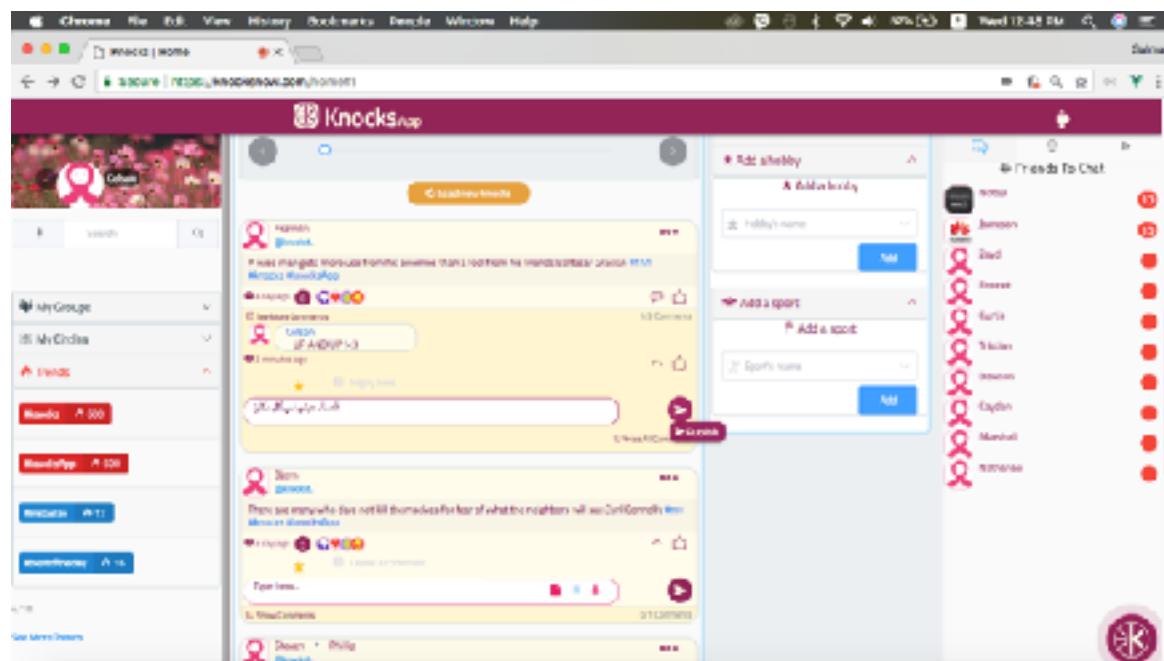
## CHAPTER 8 : USER MANUAL



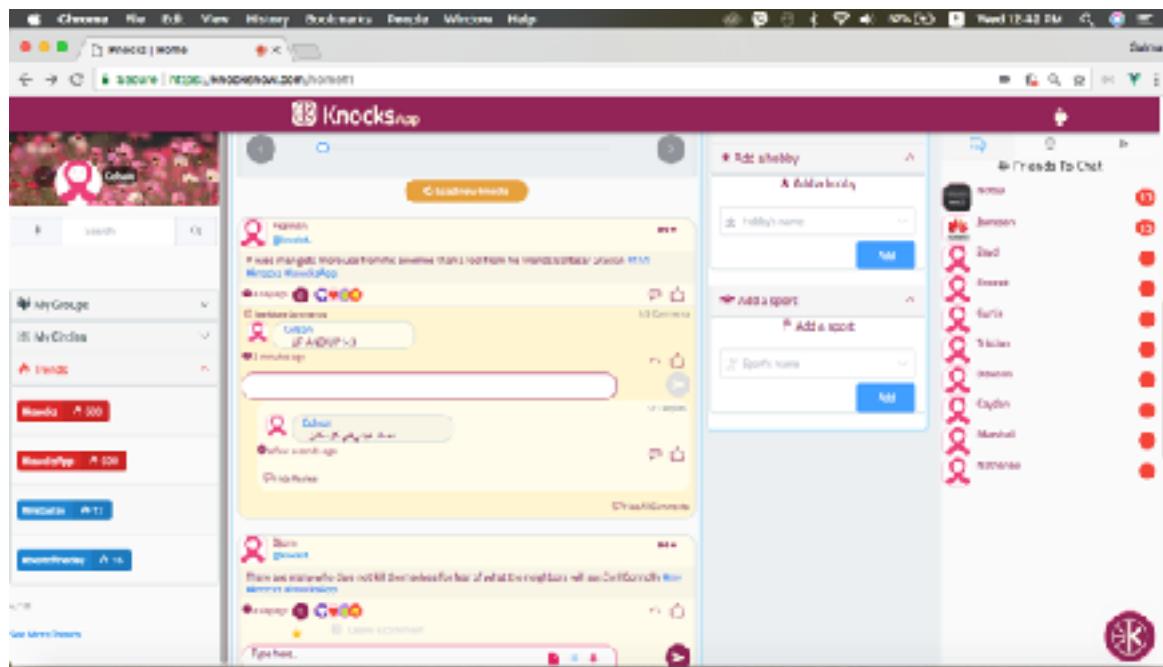
Else the system will display the Comment towards the knock at the user's time line

- **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”

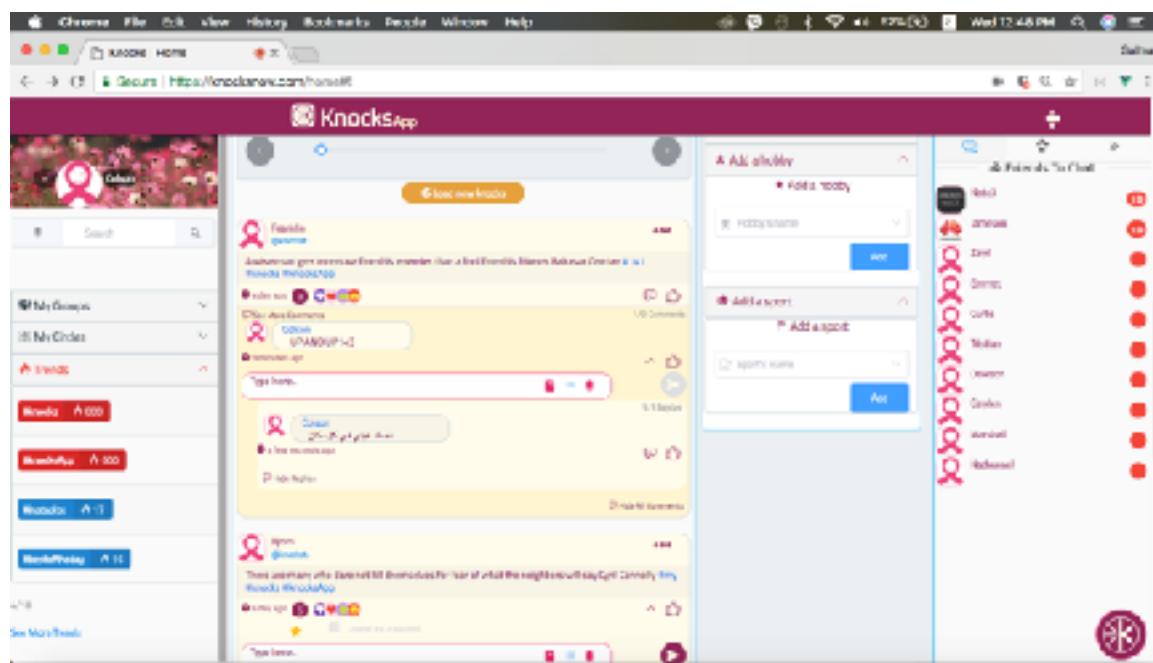


If the date are incorrect (Empty Reply), the system will not enable the publish button



Else

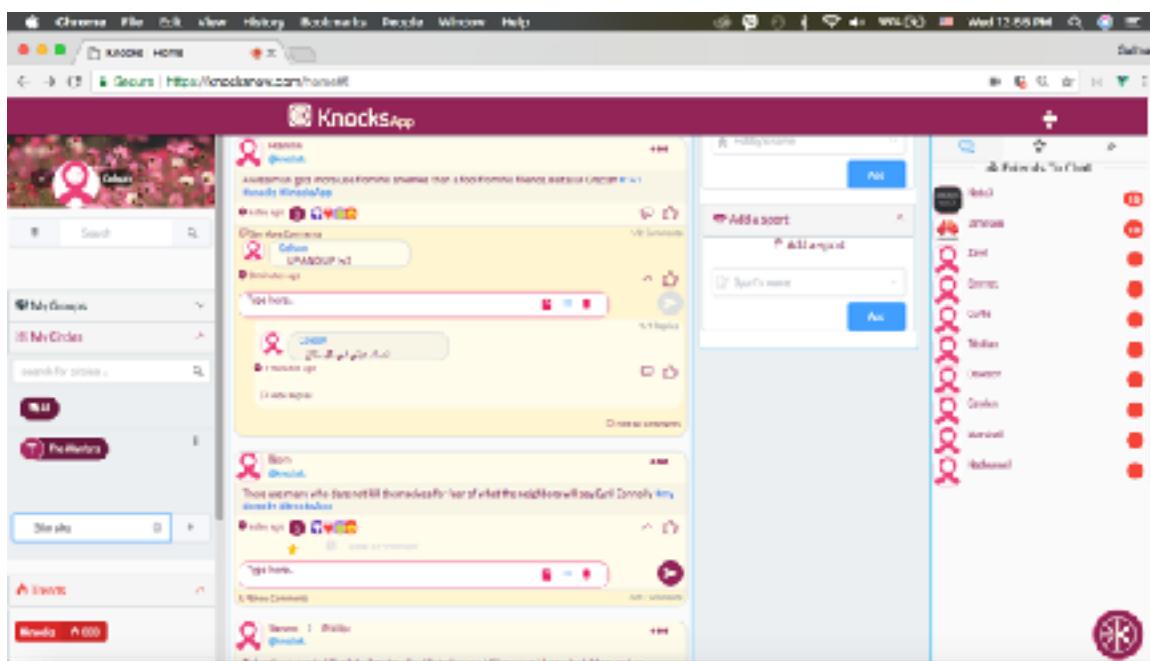
the system will display the reply towards the Comment at the user's time line



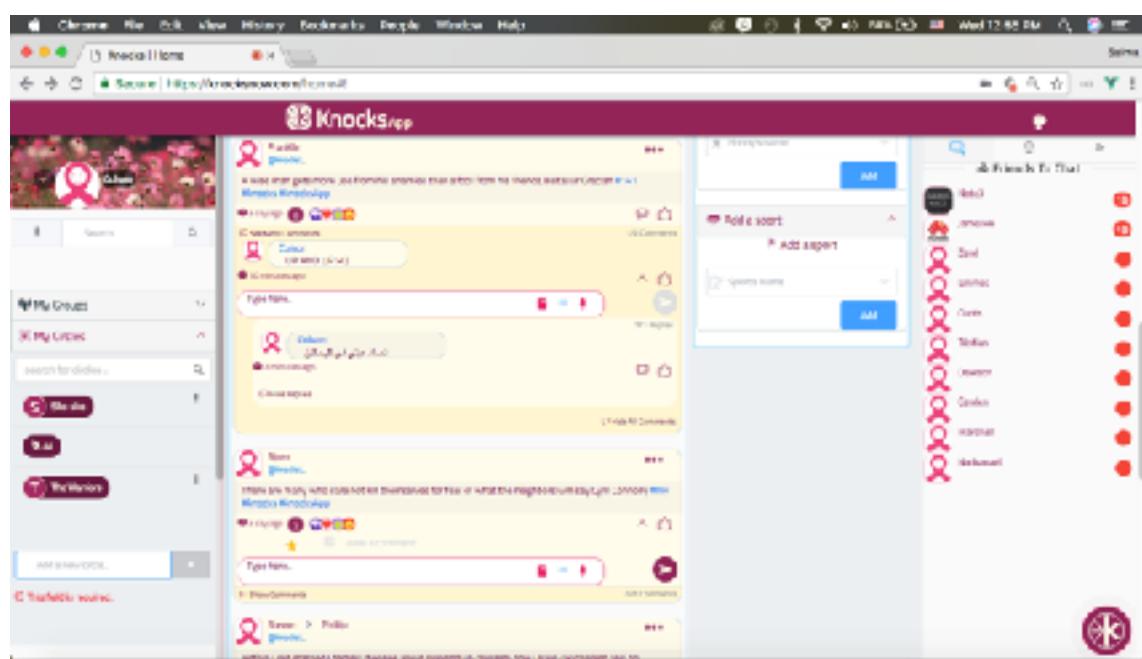
- Create a circle.**

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

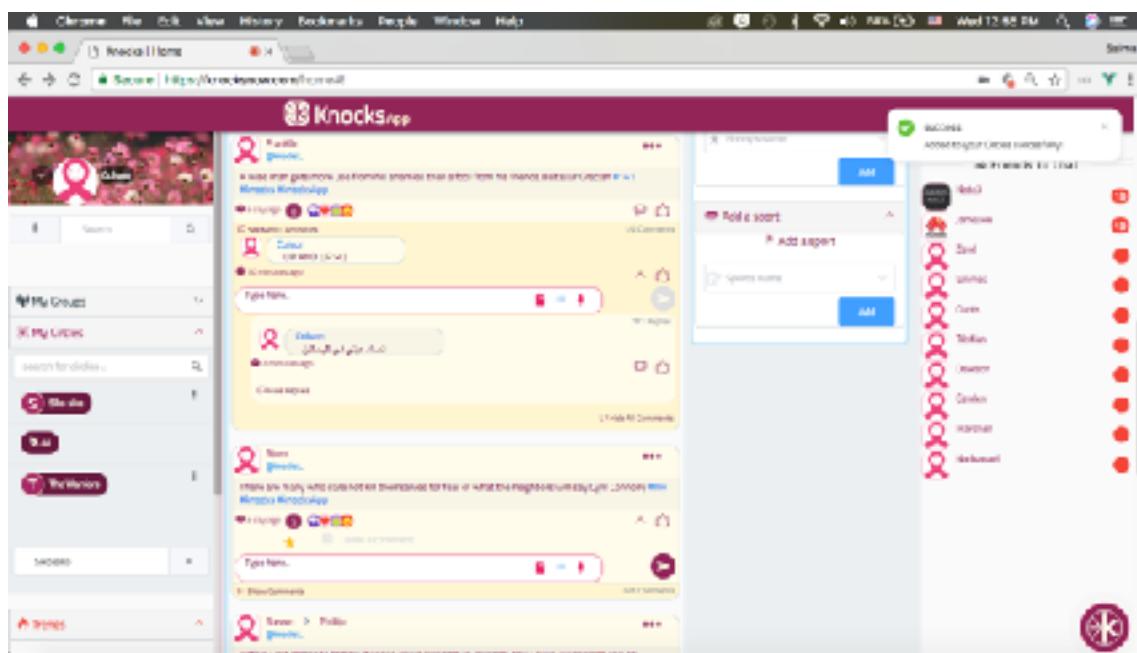
## CHAPTER 8 : USER MANUAL



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

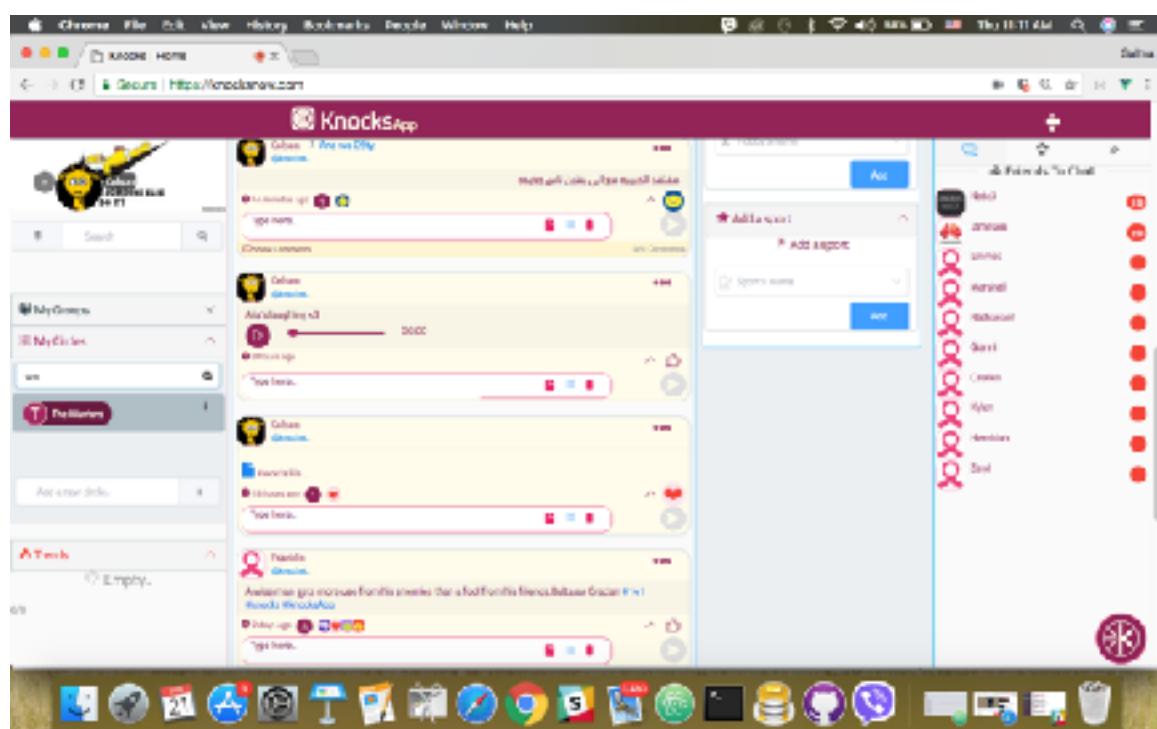


Else the system will display message “Added to your circles successfully” and displays the newly created user circle in the circles section



- **Delete a circle**

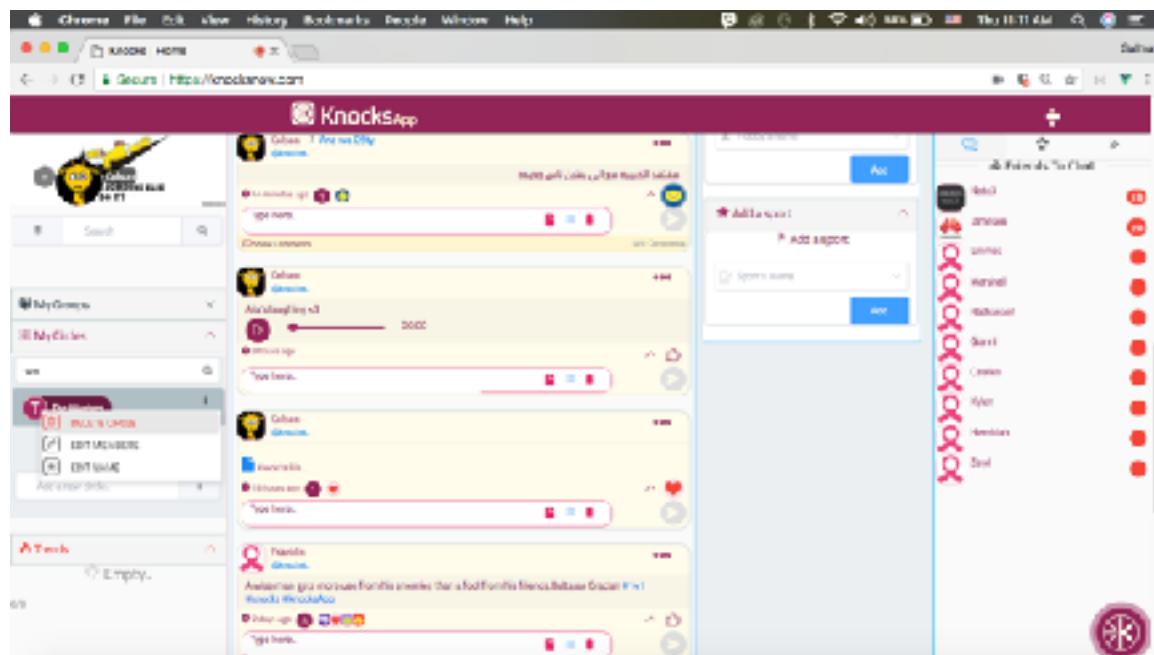
In the circles section on the middle left corner of the timeline the user should provide selection for the previously created circle by searching for it.



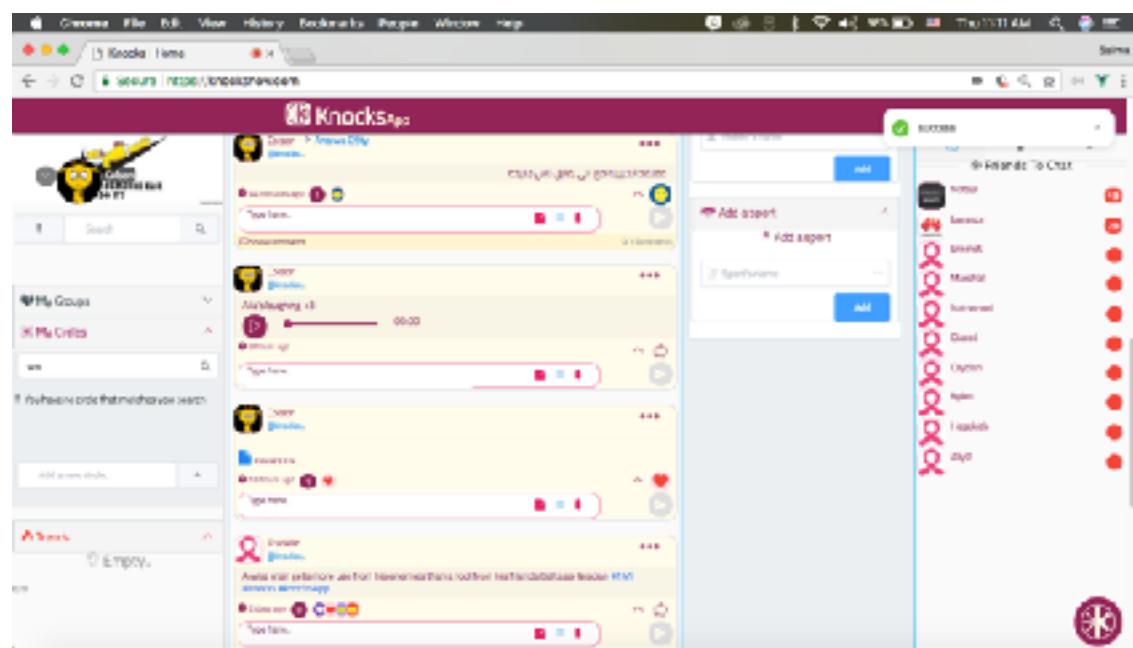
And then chooses from the dropdown list ( Delete Circle, Edit Member, Edit Name ), then he

## CHAPTER 8 : USER MANUAL

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.

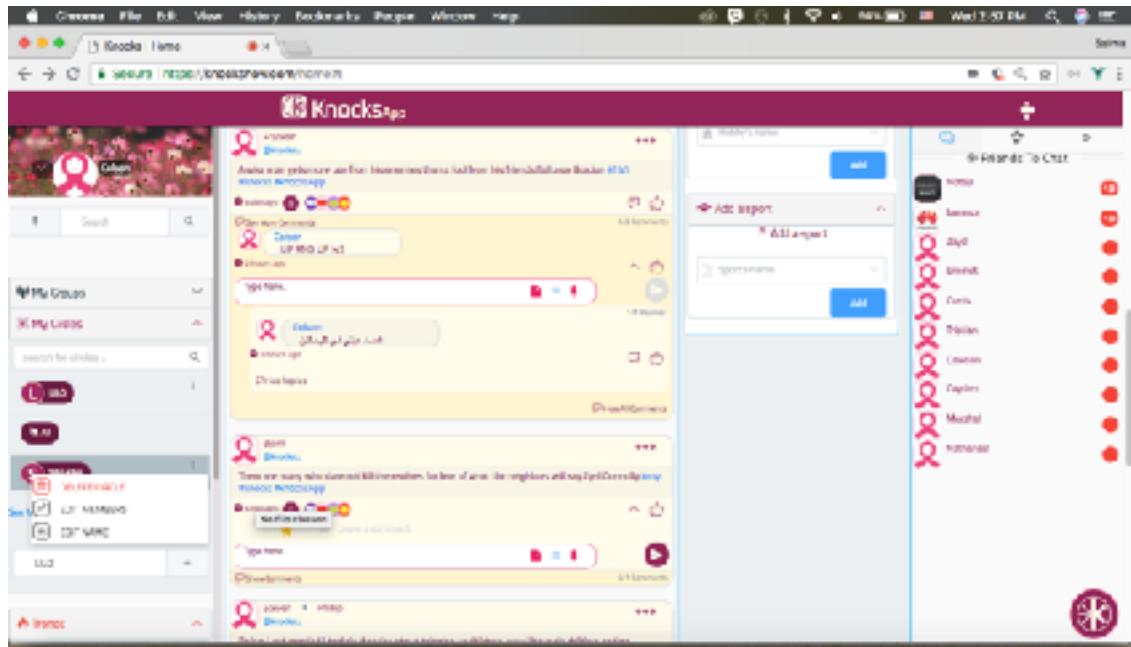


### • Update circle name

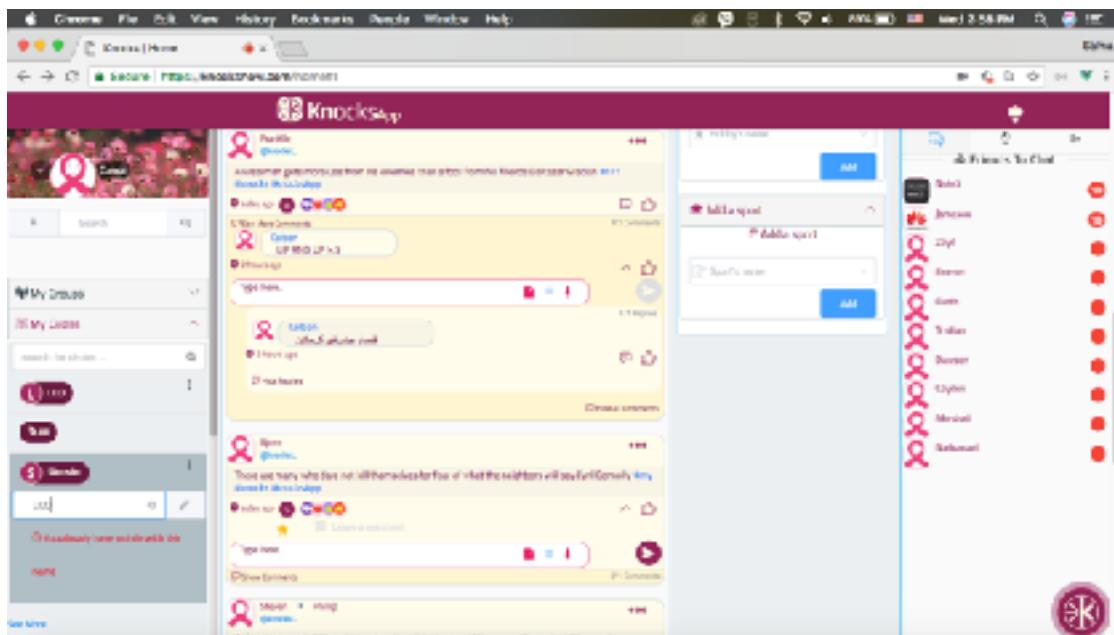
In the circles section on the middle left corner of the timeline the user should provide select the previously created circle and then chooses from the dropdown list ( Delete Circle, Edit

## CHAPTER 8 : USER MANUAL

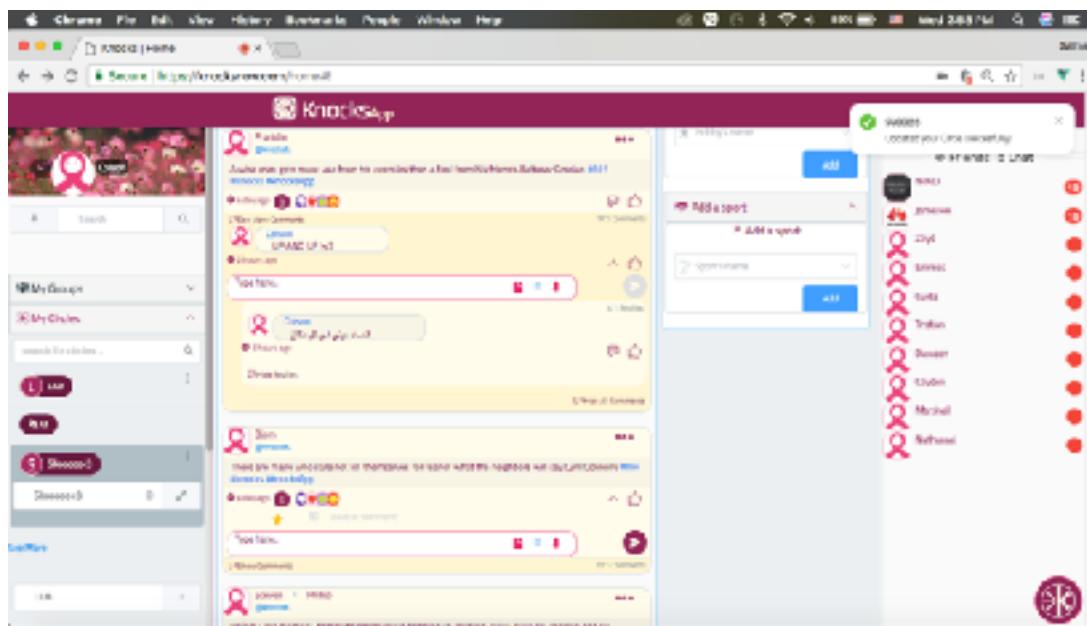
Member, Edit Name ), then he chooses “Edit Name” option then he should provide the new circle name that must be unique for this user’s circles, then clicks on Add Circle icon



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

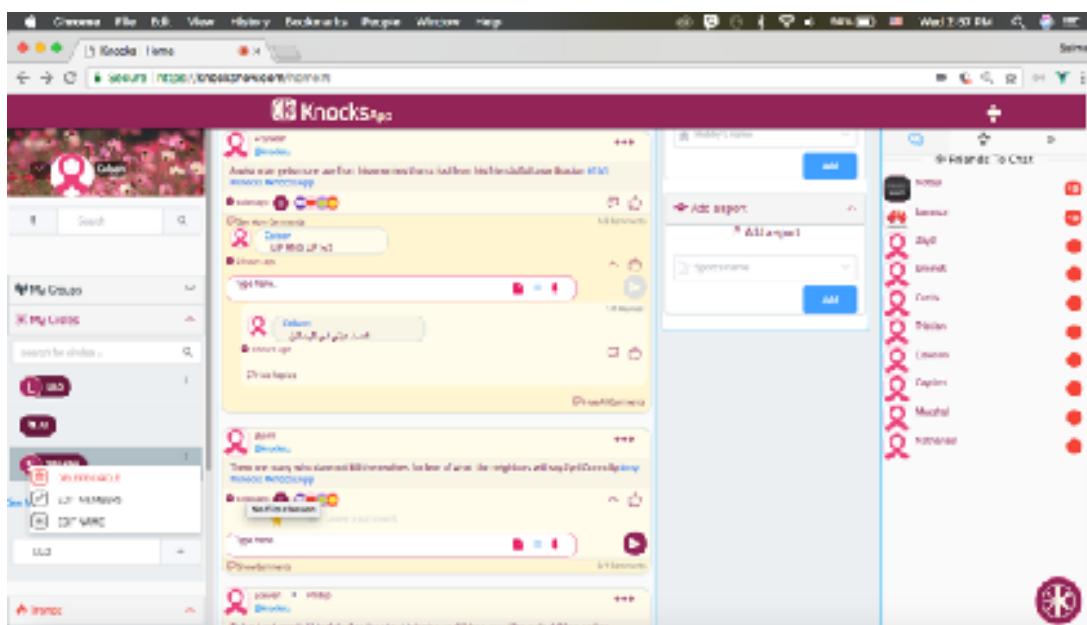


Else the system will display message “Updated your circle successfully” and displays the newly updated user circle in the circles section

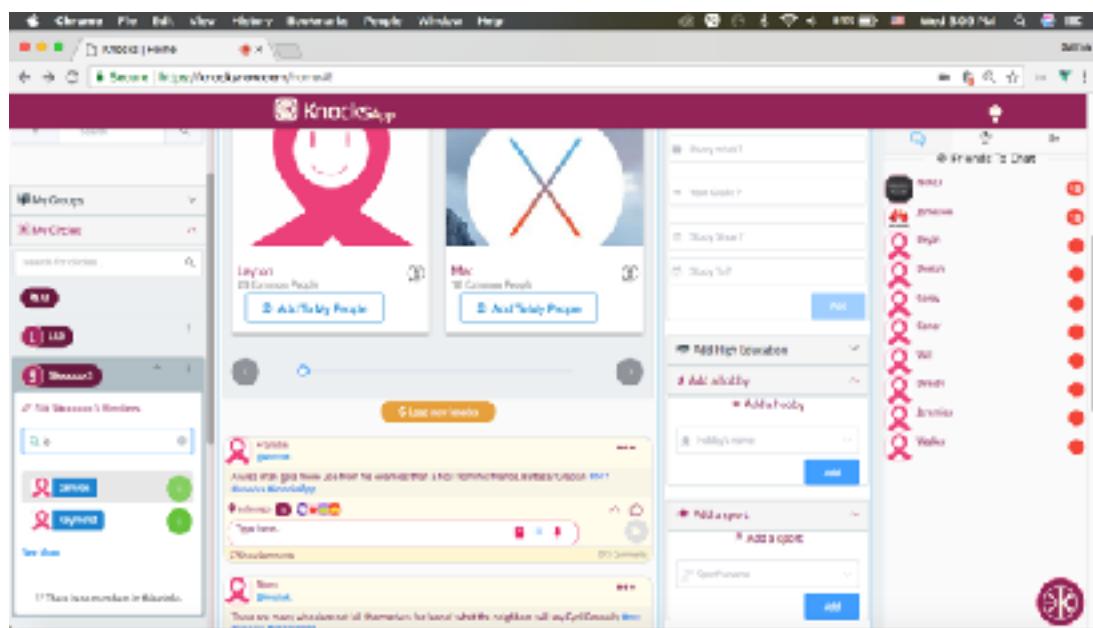


- Include a friend to a circle**

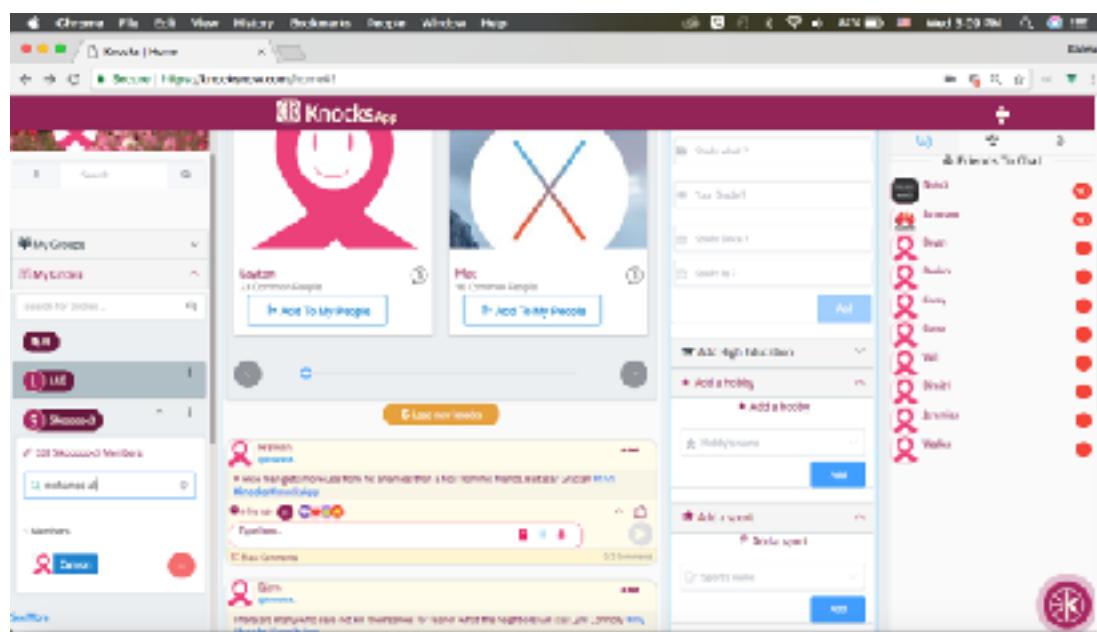
In the circles section on the middle left corner of the timeline the user should provide select the previously created circle and then chooses from the dropdown list ( Delete Circle, Edit Member, Edit Name ), then he chooses “Edit Members” option then he should search for the user that he wants to add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



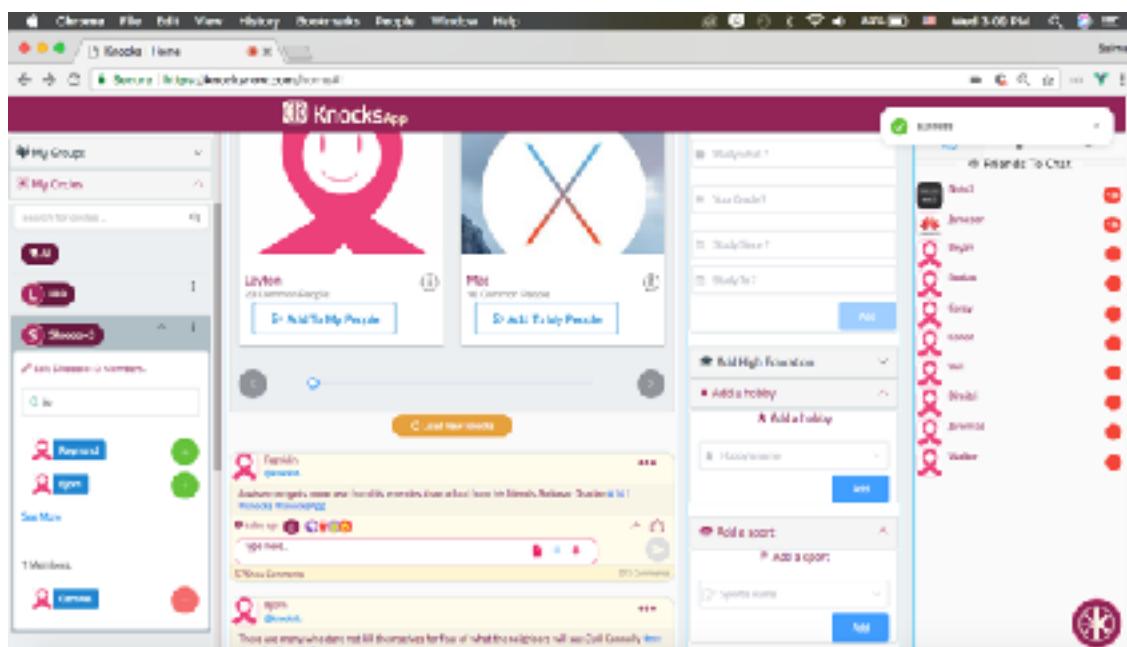
## CHAPTER 8 : USER MANUAL



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

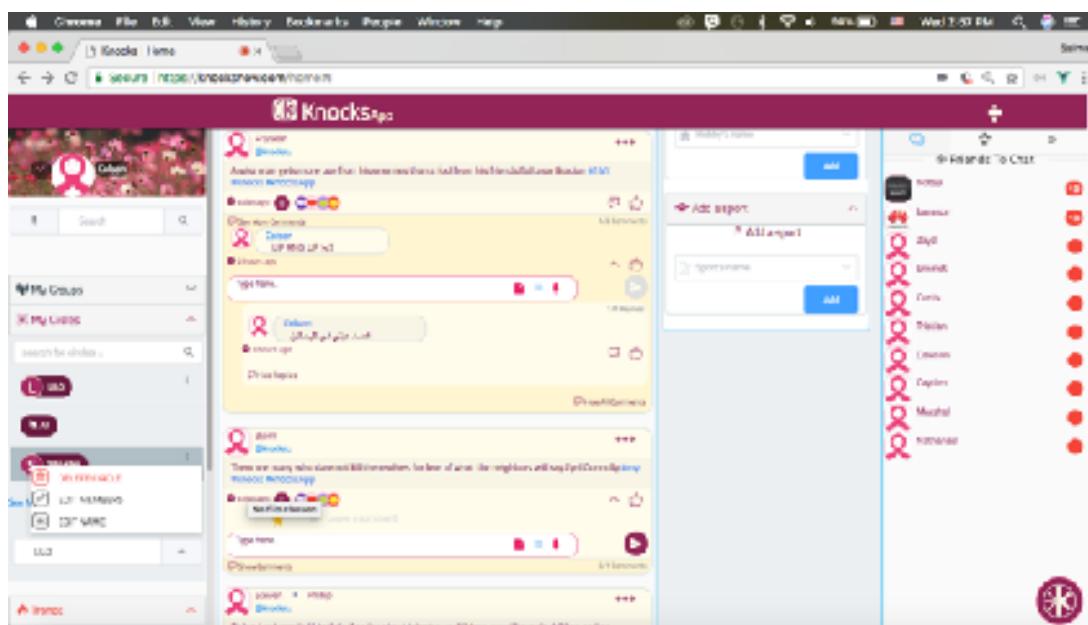


Else the system will display message “success” and displays the newly added member below the circle with red icon to indicate that this user is a member of the circle in the circles section

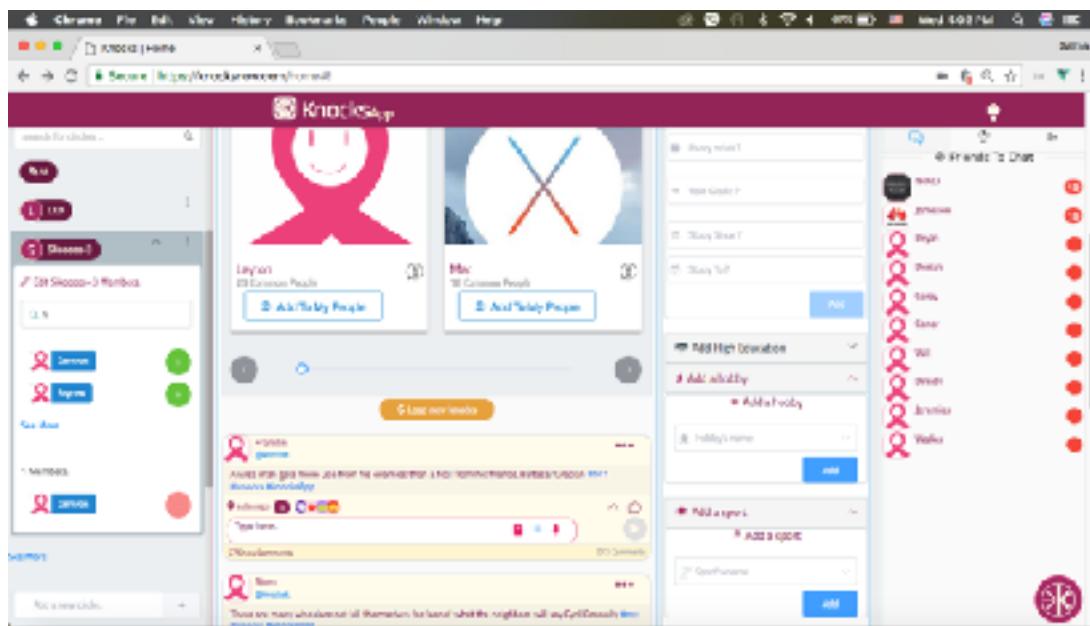


- Exclude a friend from a circle**

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



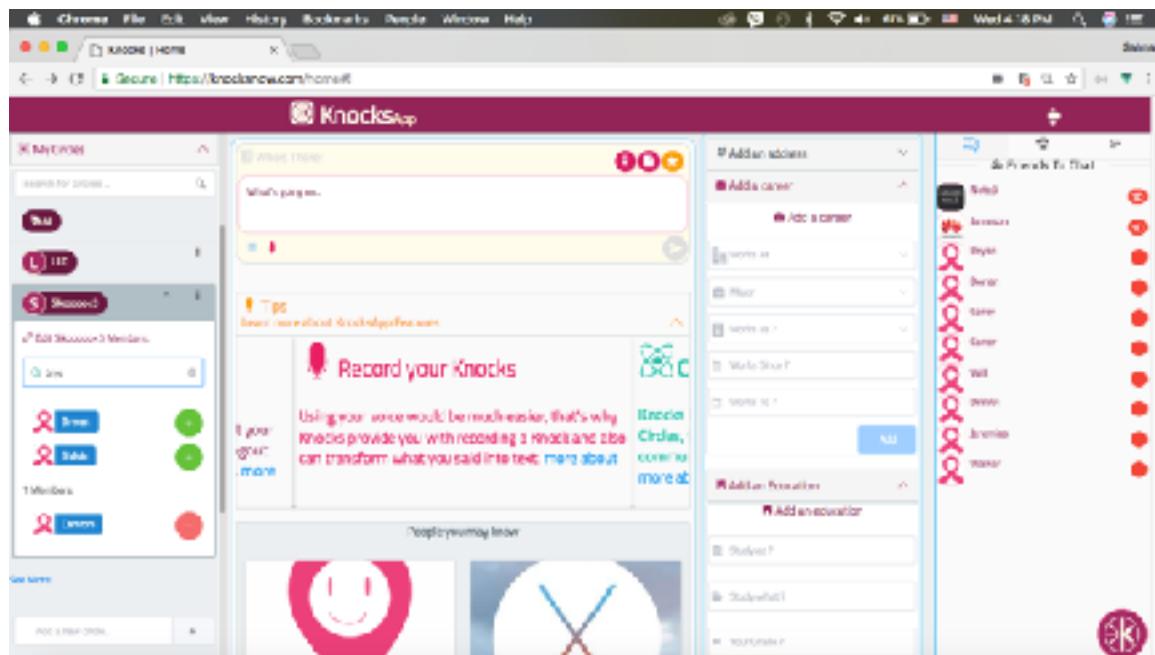
## CHAPTER 8 : USER MANUAL



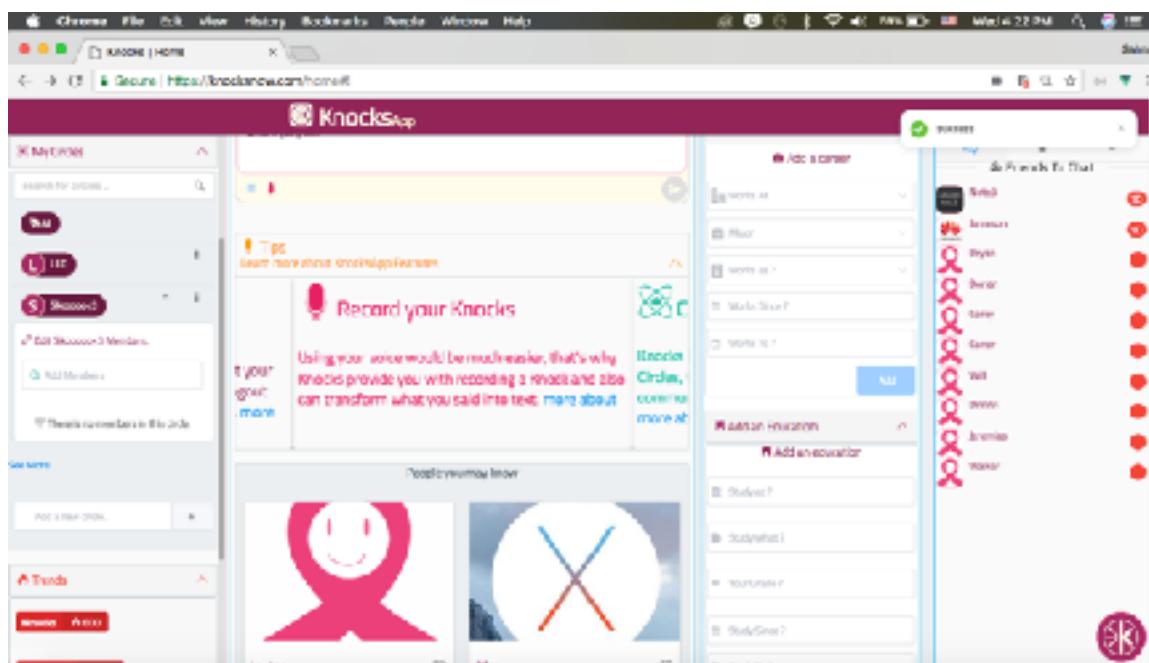
If

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

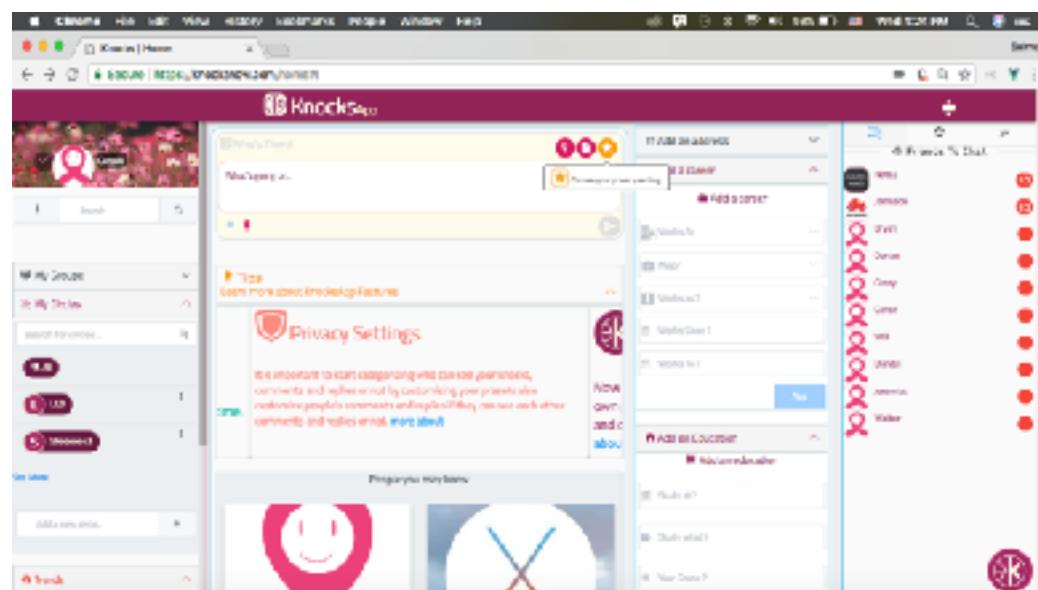


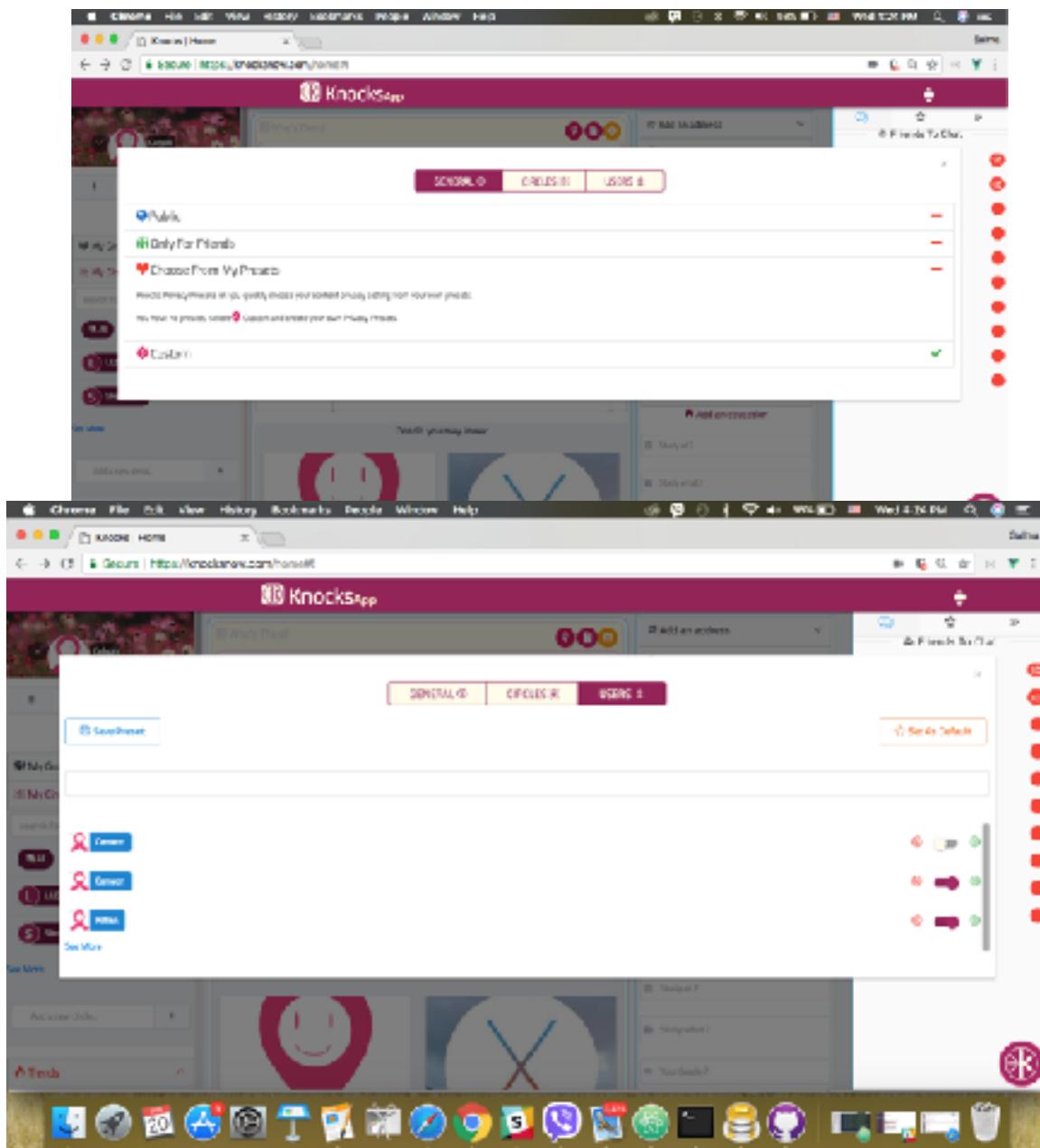
Else the system will display message “success” and displays the newly added member below the circle with red icon to indicate that this user is a member of the circle in the circles section



- **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

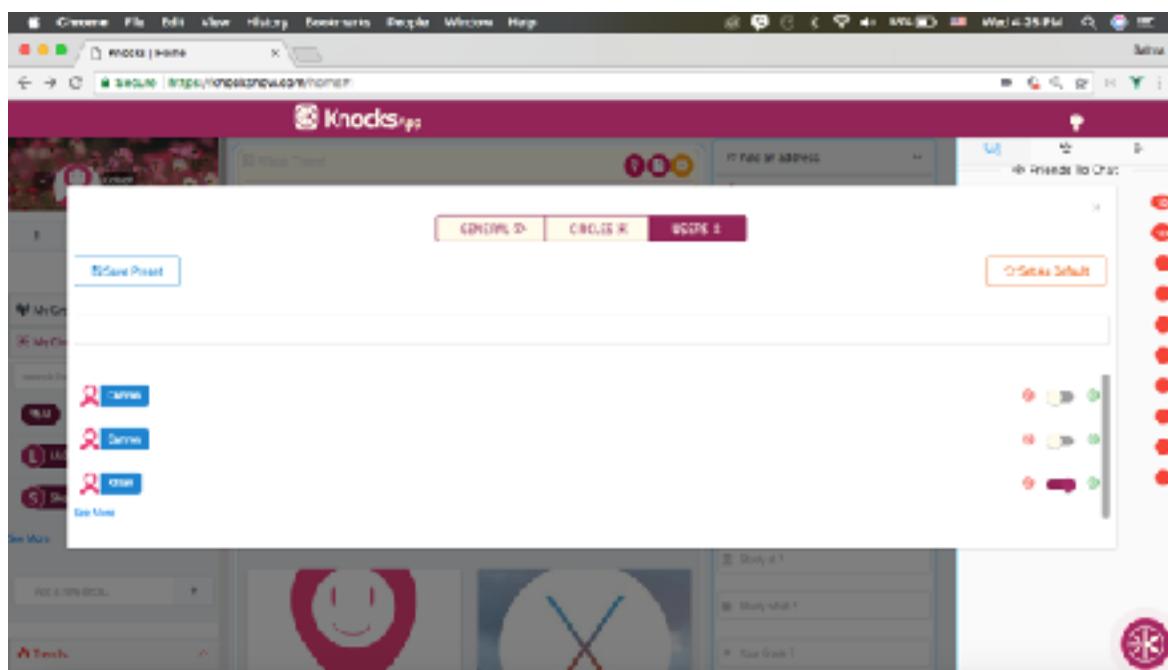
the date are incorrect (searching for a user who is not a friend), the system will not display any user to choose privacy set for him.

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

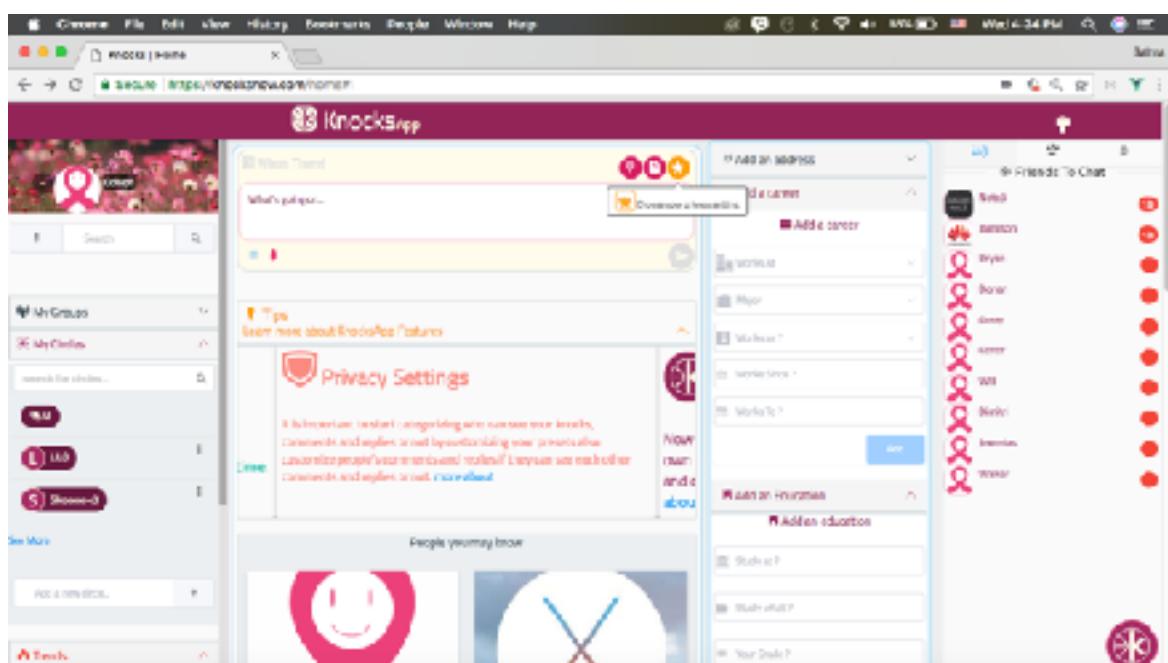
- **Regulate a privacy set for circles**

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

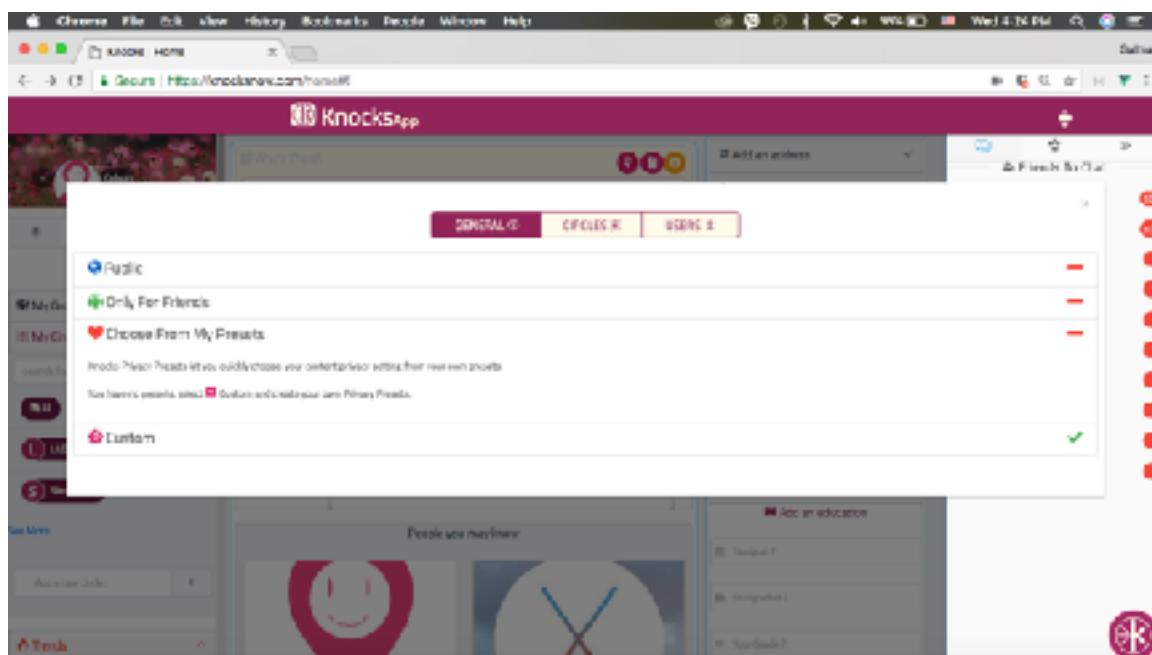
## CHAPTER 8 : USER MANUAL



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.



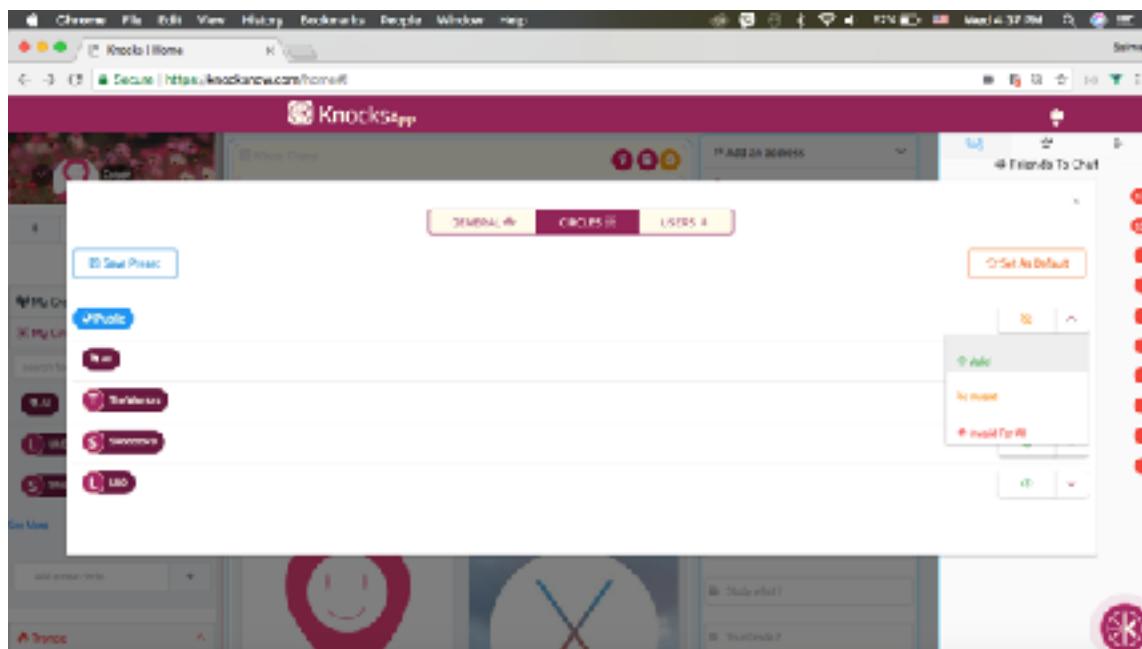
## CHAPTER 8 : USER MANUAL



Else  
the

system

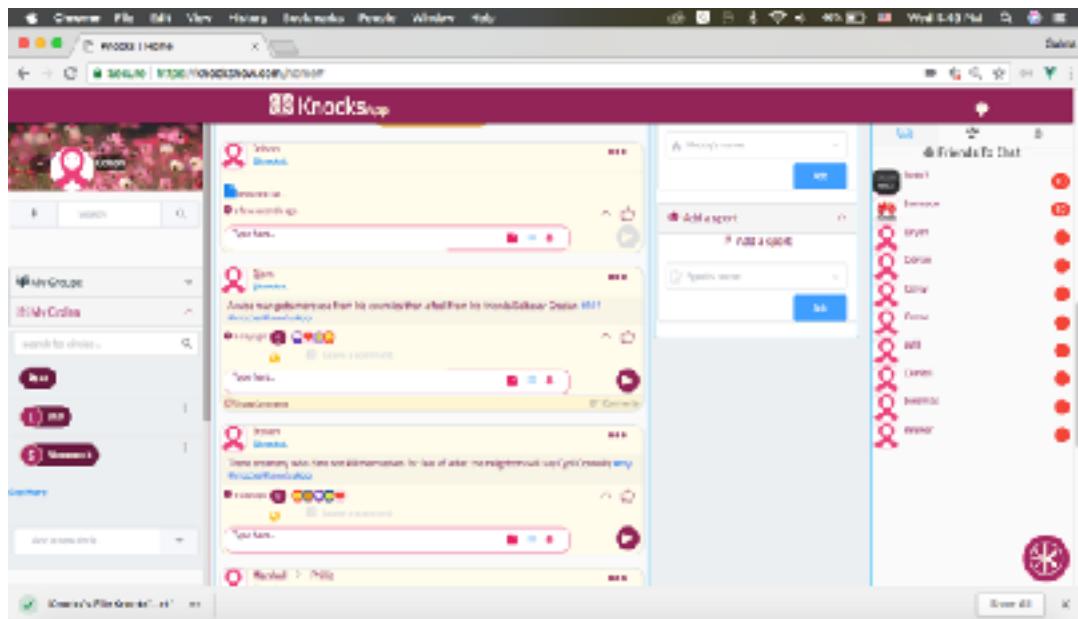
will display the new privacy set for the selected circle.



## CHAPTER 8 : USER MANUAL

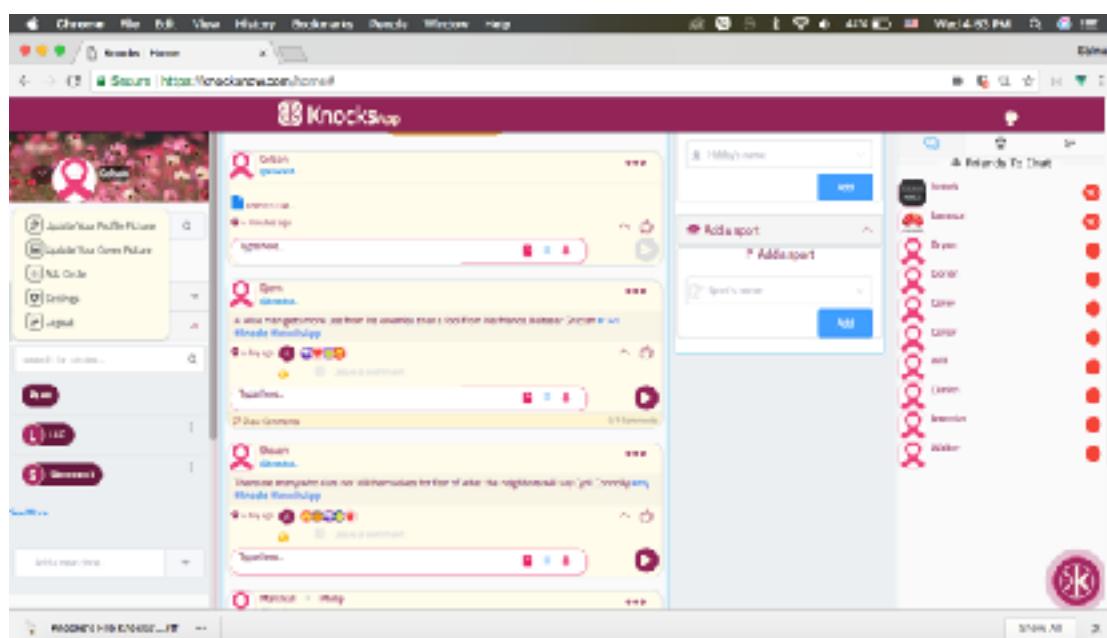
- **Download files (with any extension).**

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.

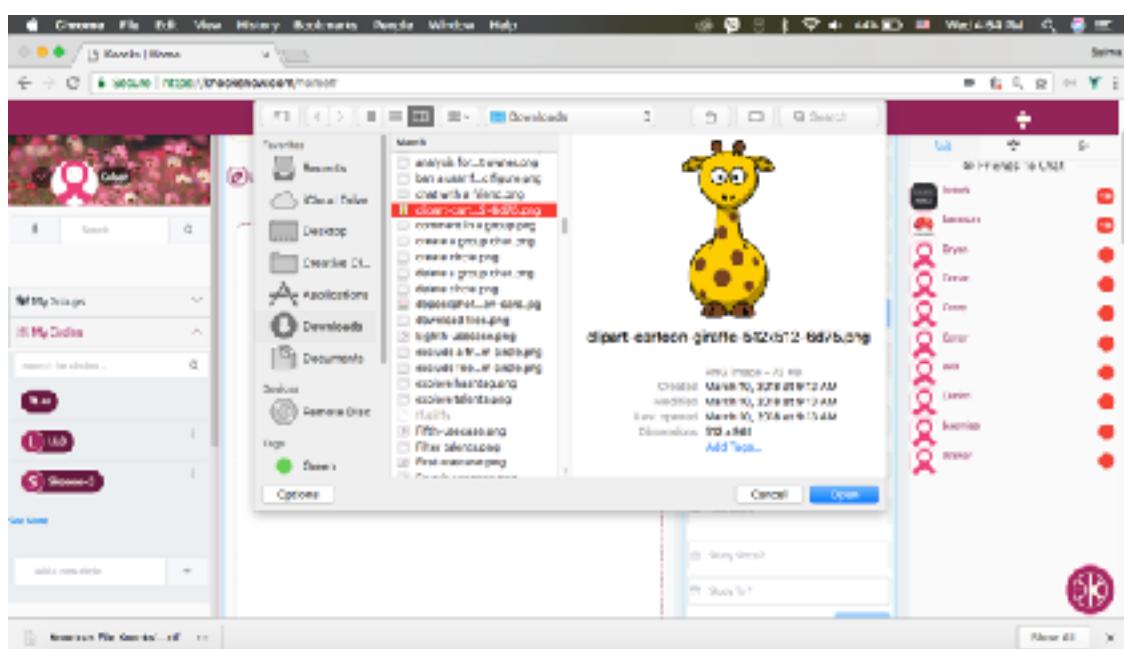
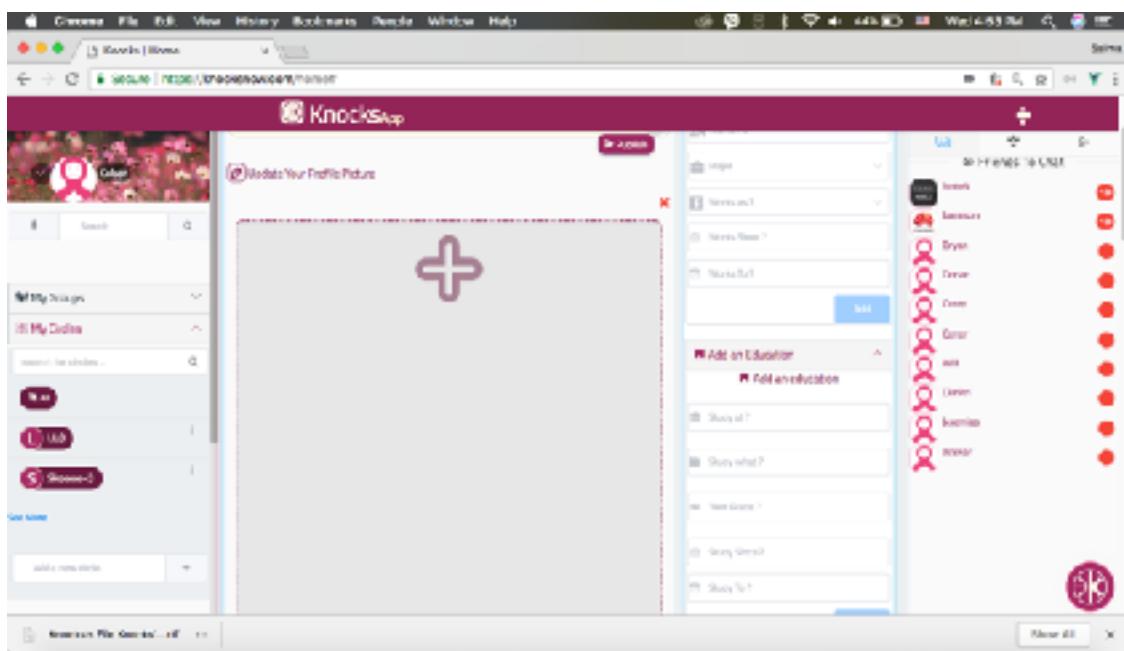


- **Upload or Update profile picture**

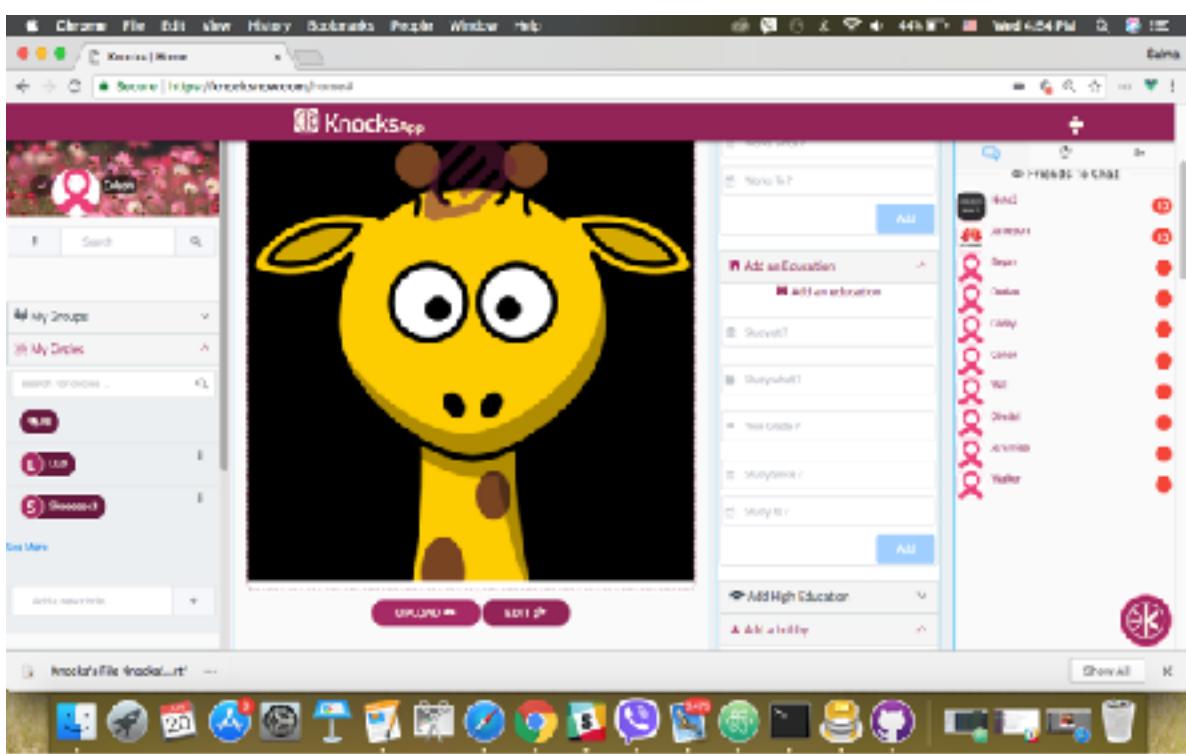
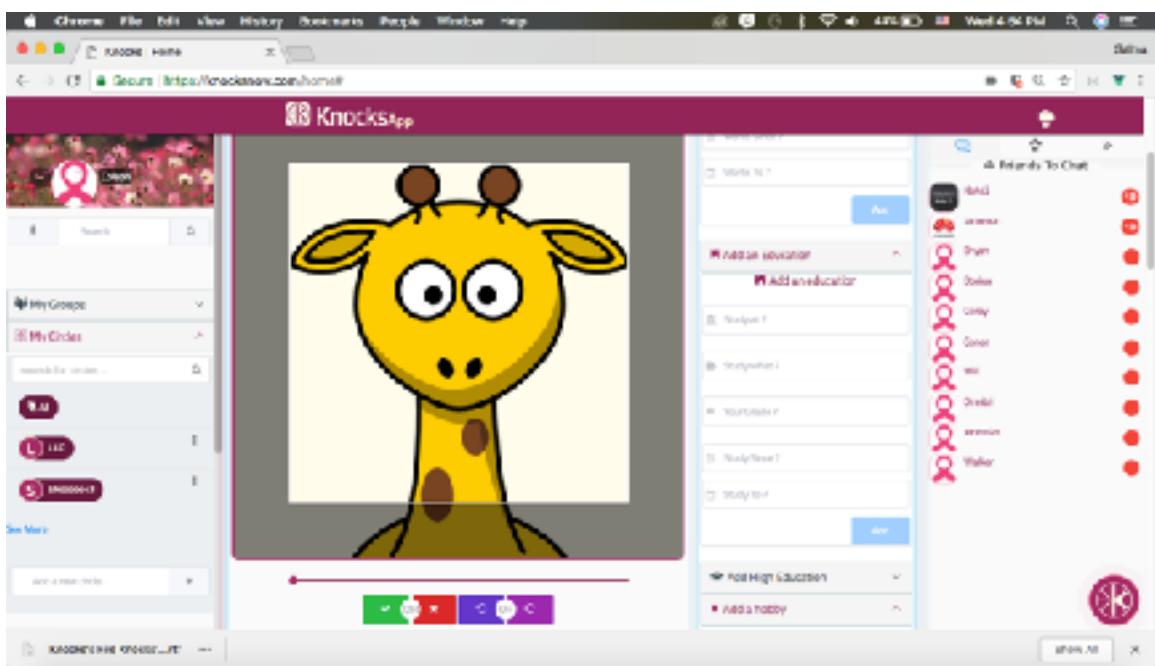
At the Profile picture section on the top left of the page, click on the arrow icon to display a dropdown list ( Update your profile picture, Update your cover picture, Add circle, Settings, Logout ) to choose “Update your profile picture”, then he clicks on the add icon on the middle and chooses a picture 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.



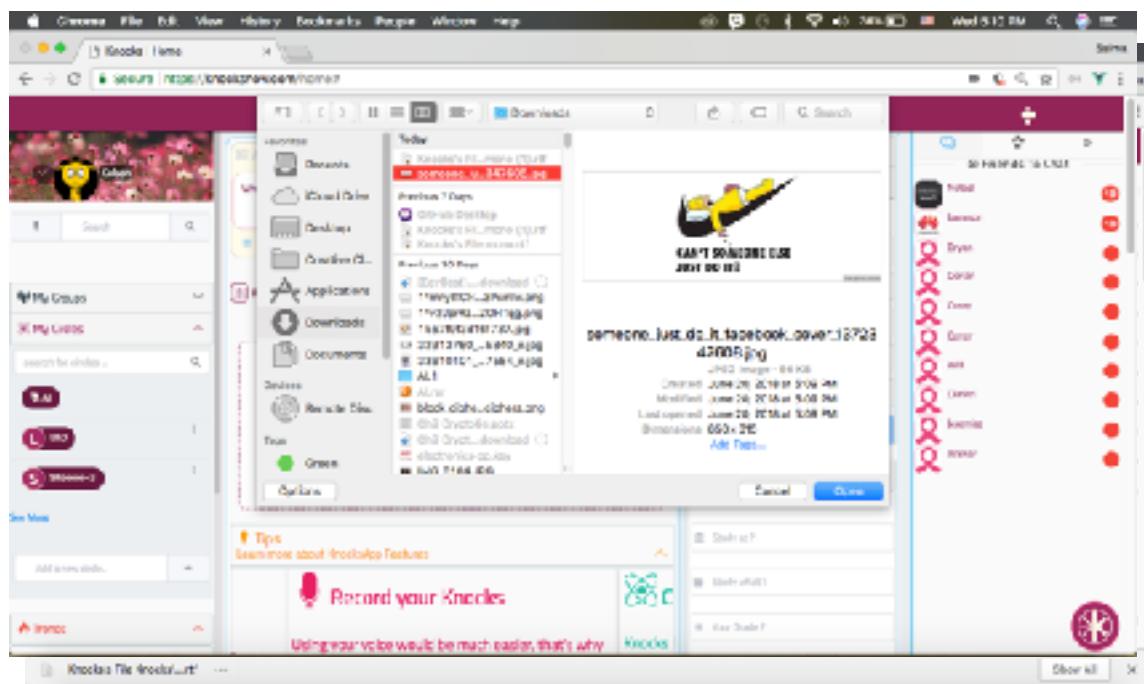
## CHAPTER 8 : USER MANUAL



## CHAPTER 8 : USER MANUAL

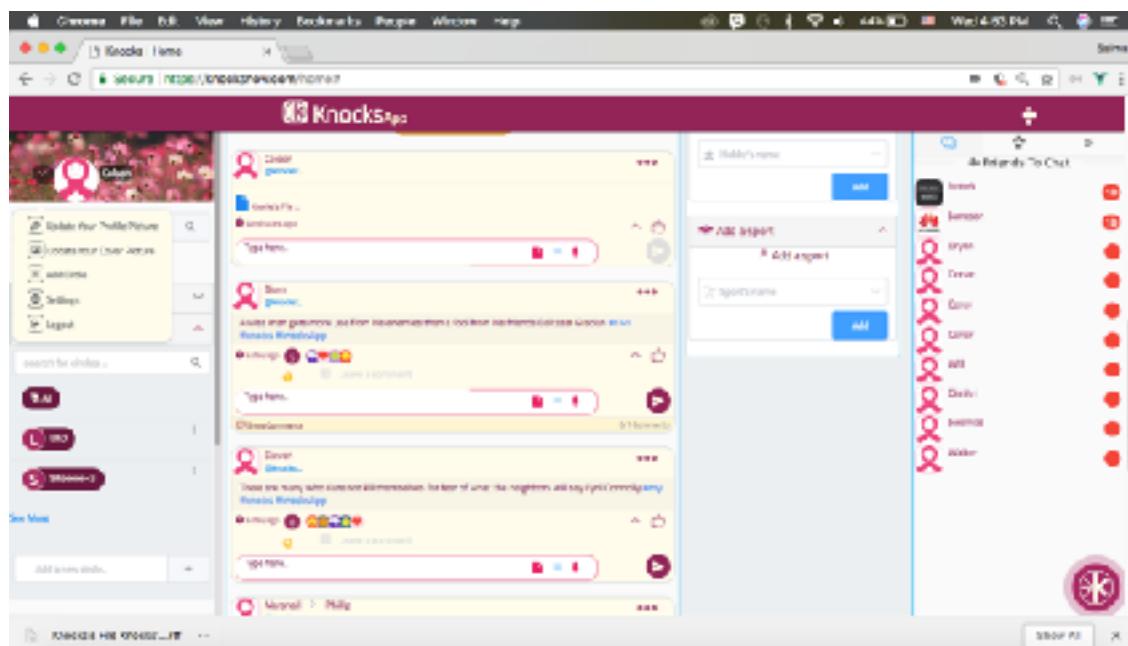


Then the system will display a message “Updated your profile picture successfully” and it will display the profile picture of the user instead of the old one.

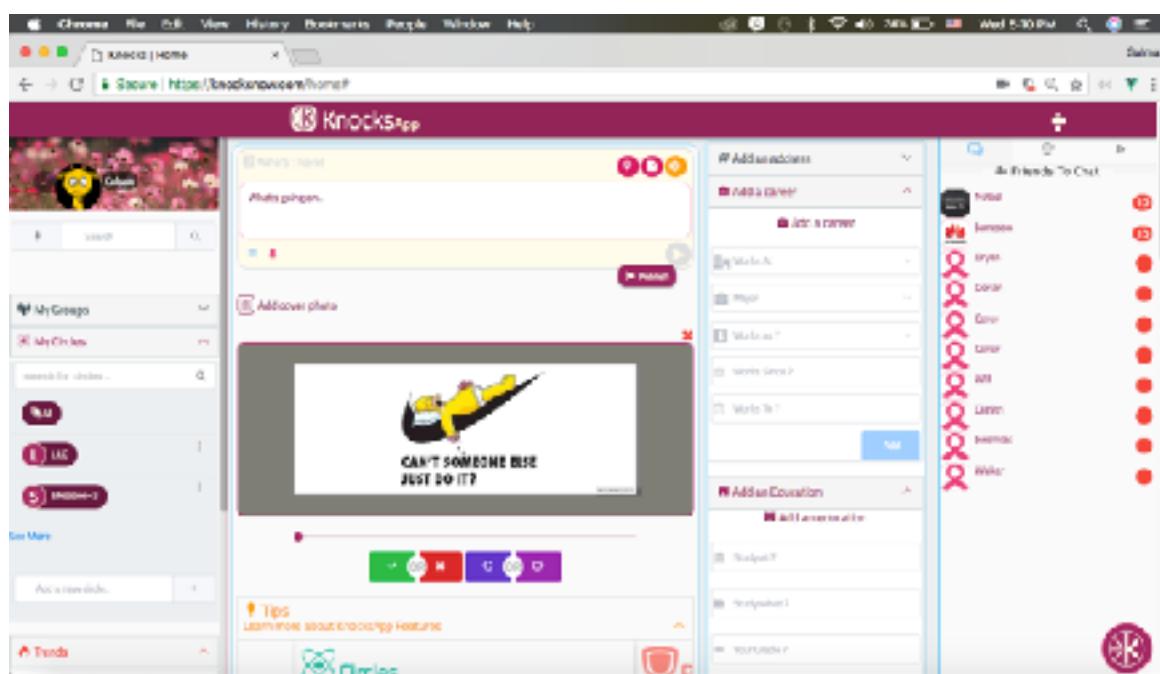
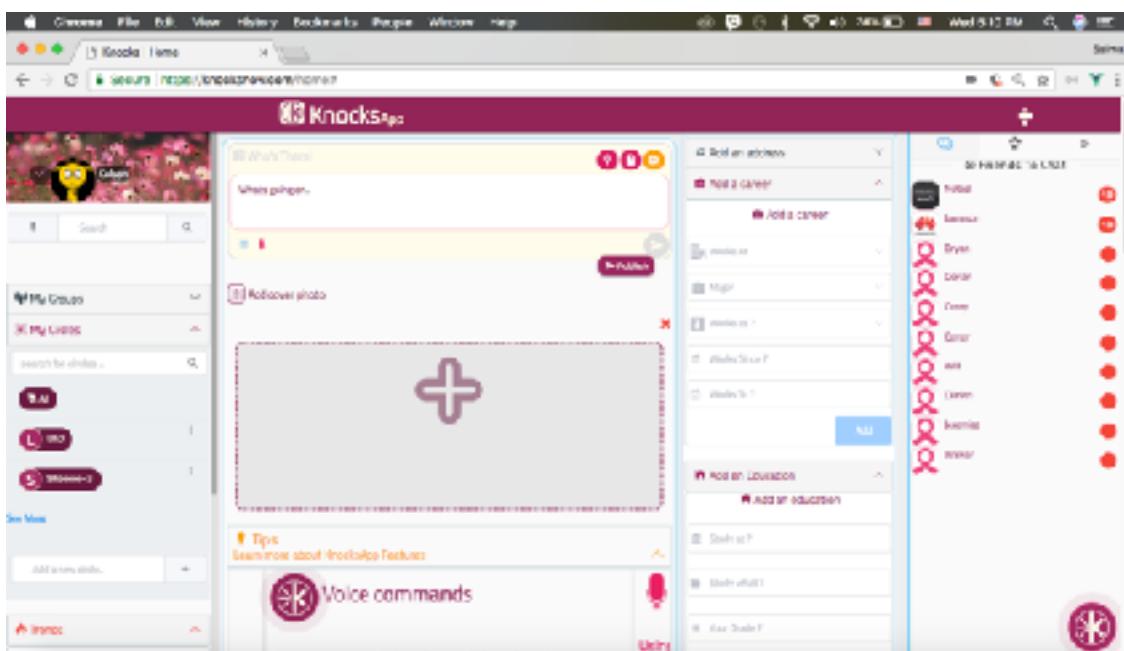


- Upload or Update cover picture**

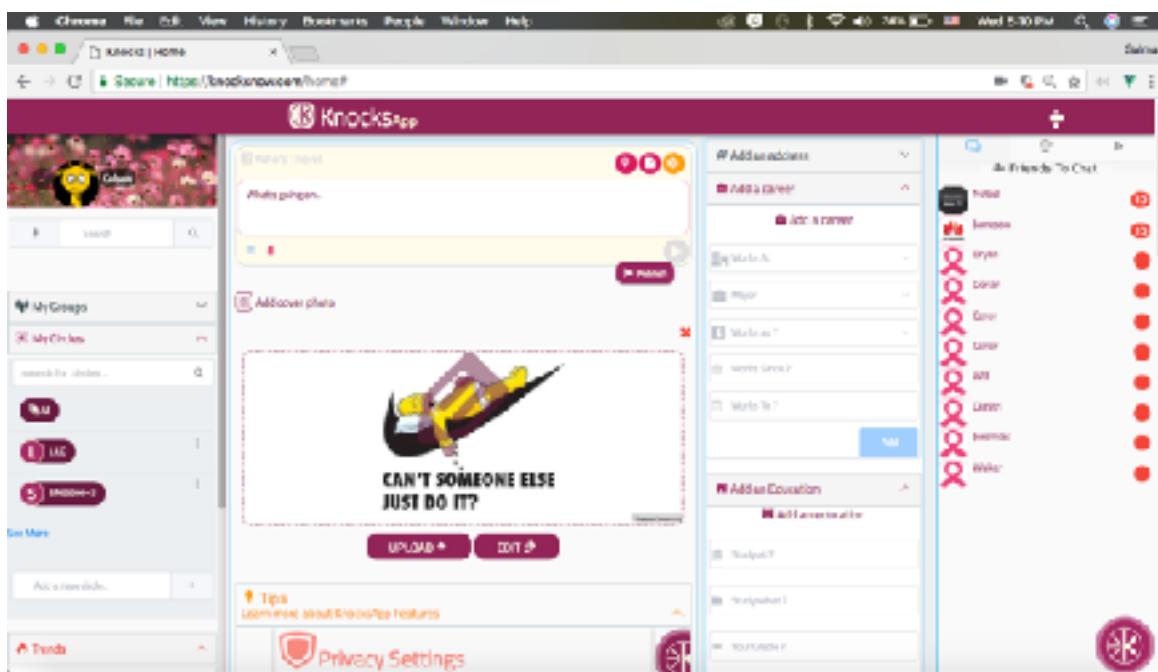
At the cover picture section on the top left of the page, click on the arrow icon to display a dropdown list ( Update your profile picture, Update your cover picture, Add circle, Settings, Logout ) to choose “Update your cover picture”, then he clicks on the add icon on the middle and chooses a picture 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.



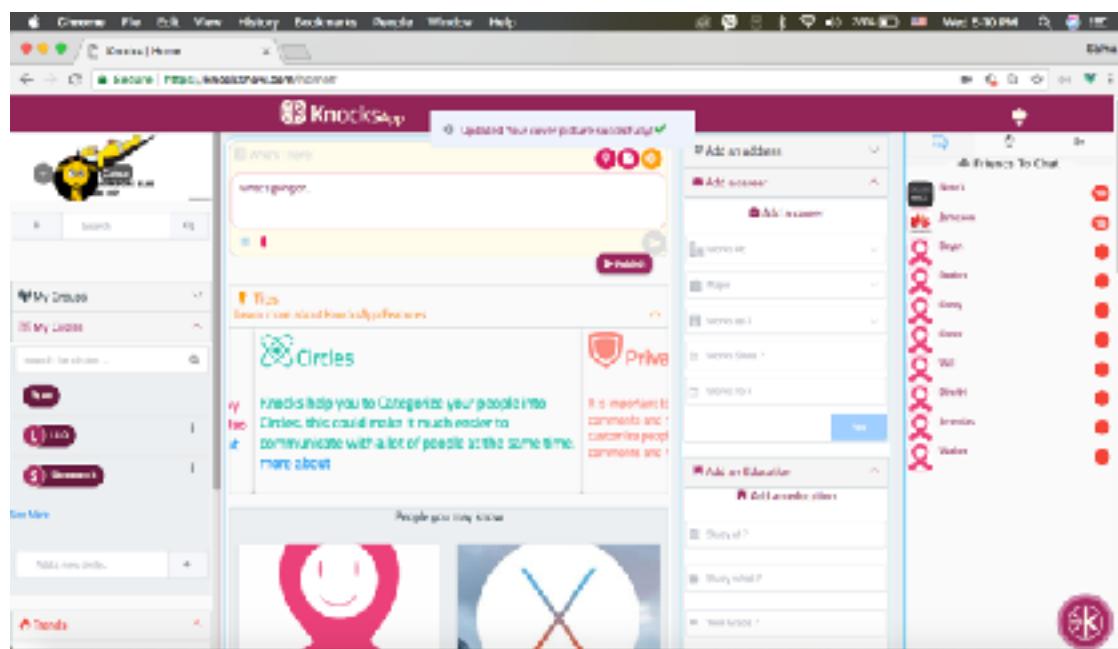
## CHAPTER 8 : USER MANUAL



## CHAPTER 8 : USER MANUAL



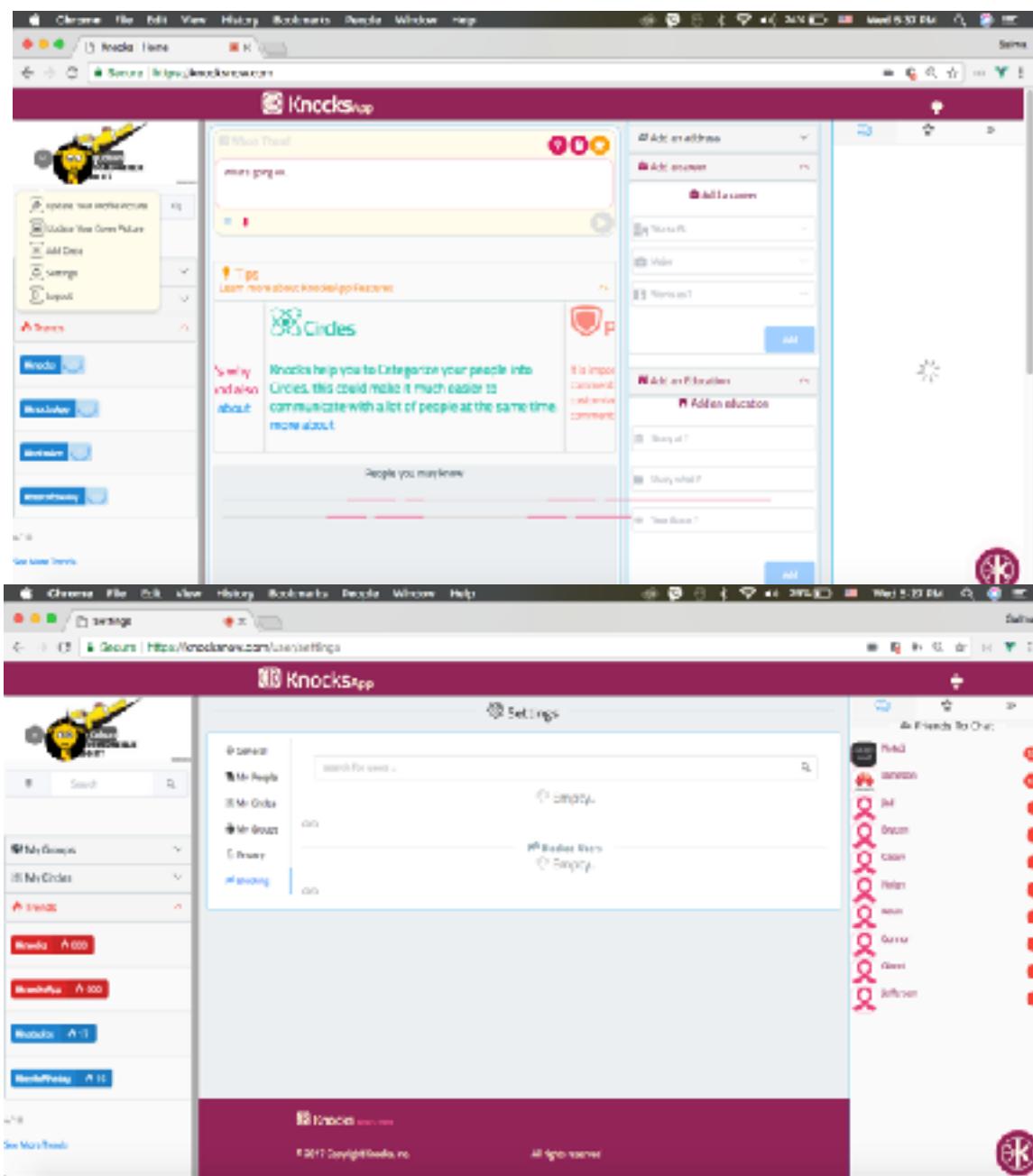
Then the system will display a message “Updated your cover picture successfully” and it will display the cover picture of the user instead of the old one.



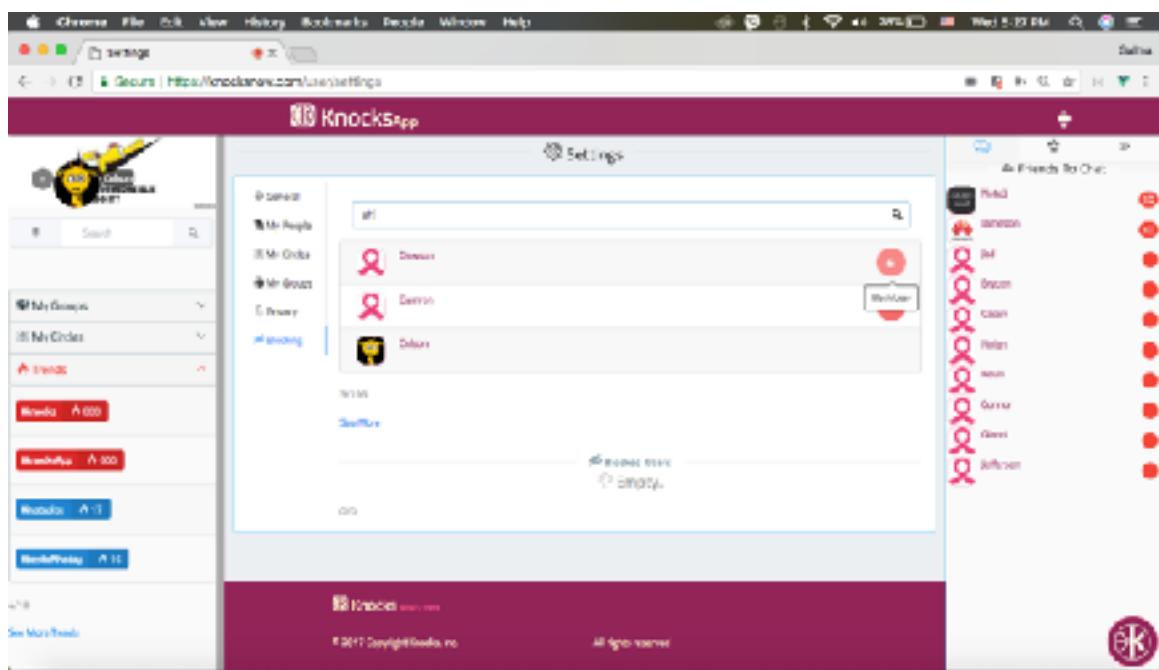
## CHAPTER 8 : USER MANUAL

- **Block a friend**

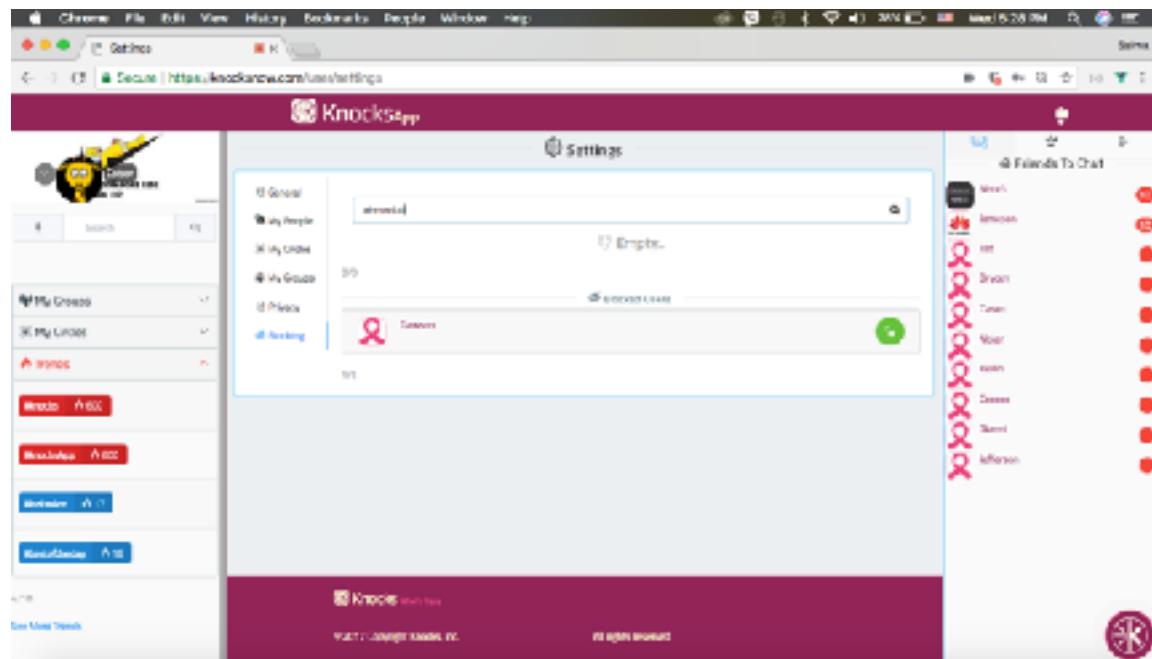
At the cover picture section on the top left of the page, click on the arrow icon to display a dropdown list ( Update your profile picture, Update your cover picture, Add circle, Settings, Logout ) to choose “Settings”, then he clicks on the lest side bar at “Blocking” tab, and search for friends then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



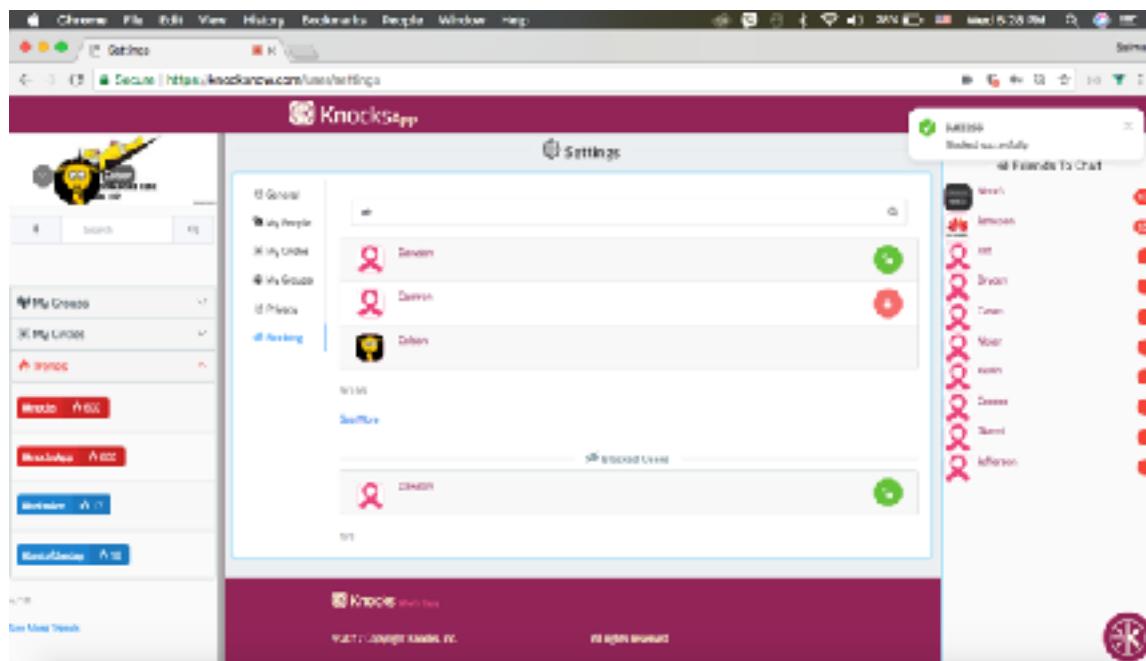
## CHAPTER 8 : USER MANUAL



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

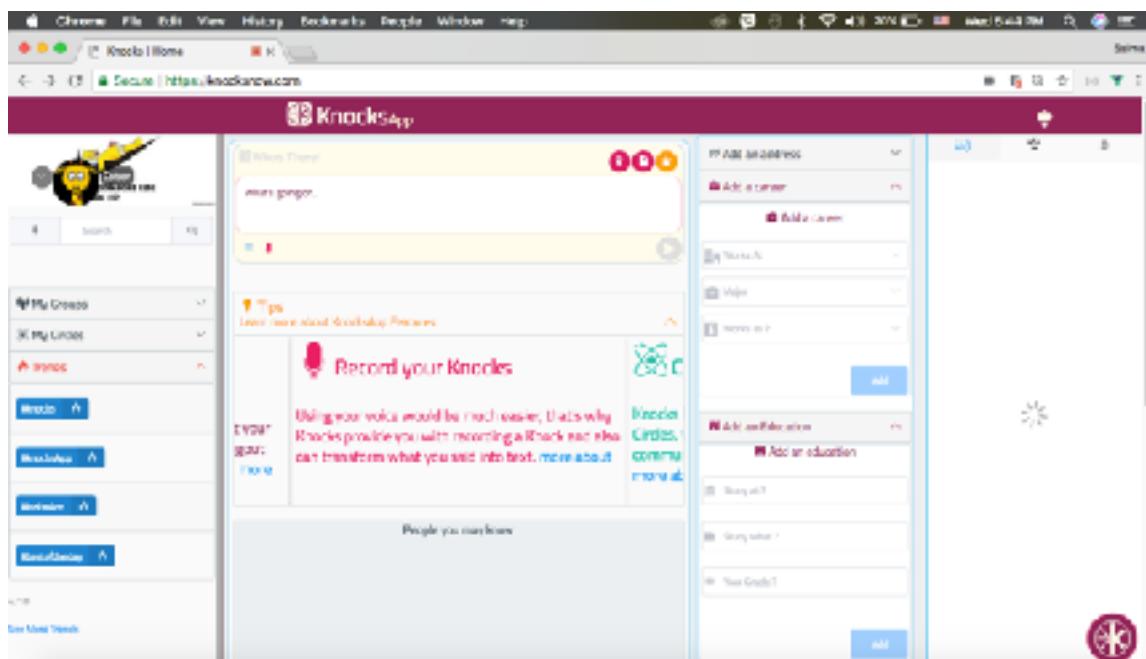


Else the system will block the selected user and display a message “Blocked successfully” and displays the green lock beside the blocked username for unblocking.

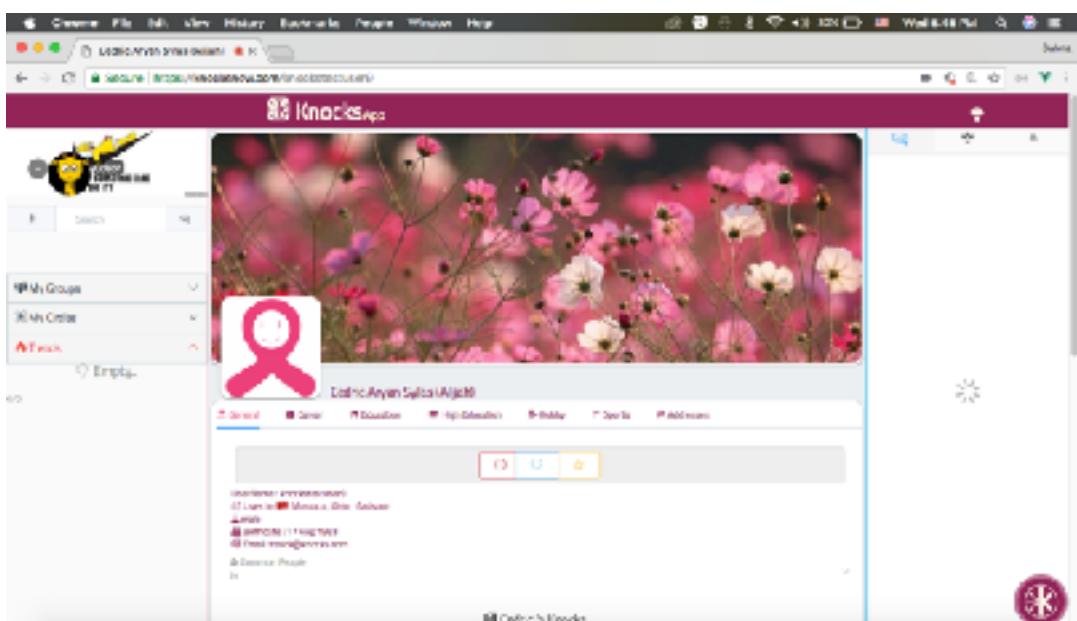
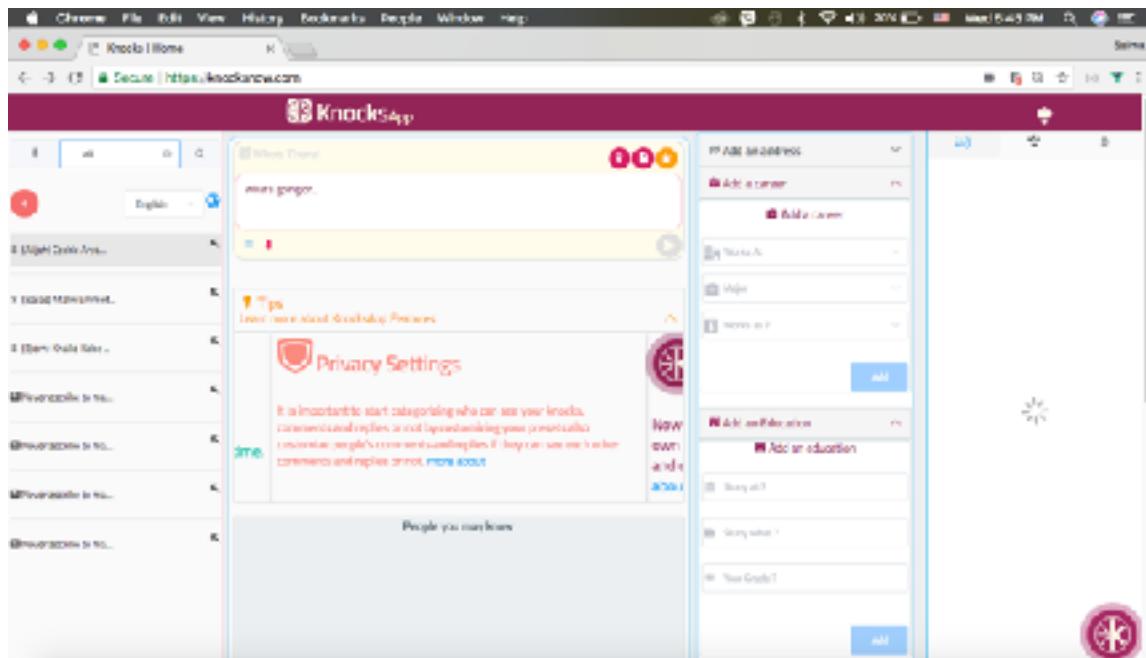


- Unfriend a friend**

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.

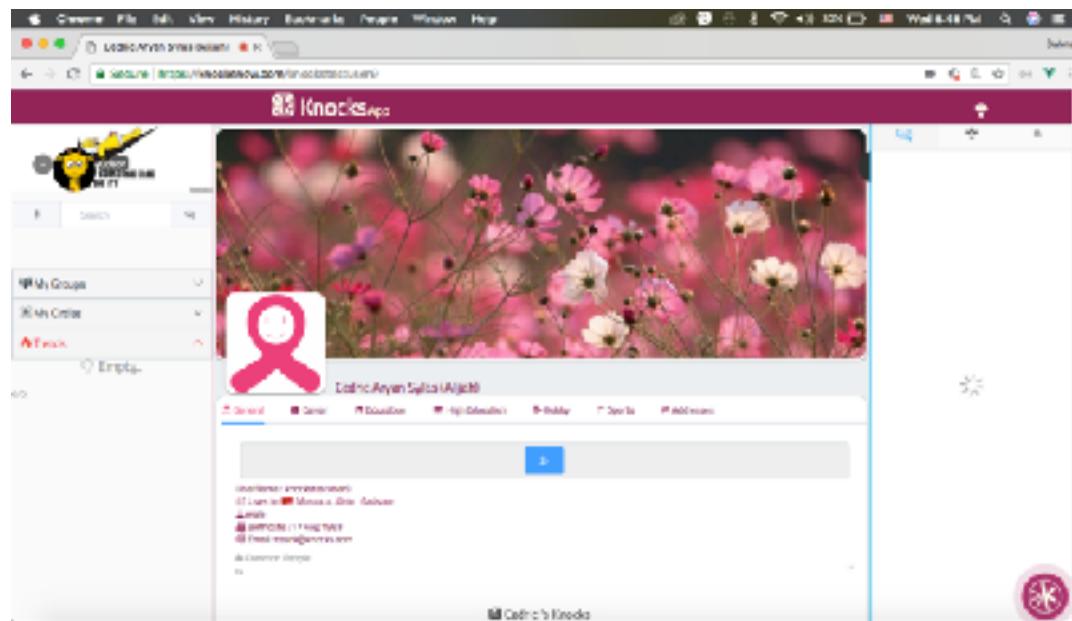


## CHAPTER 8 : USER MANUAL



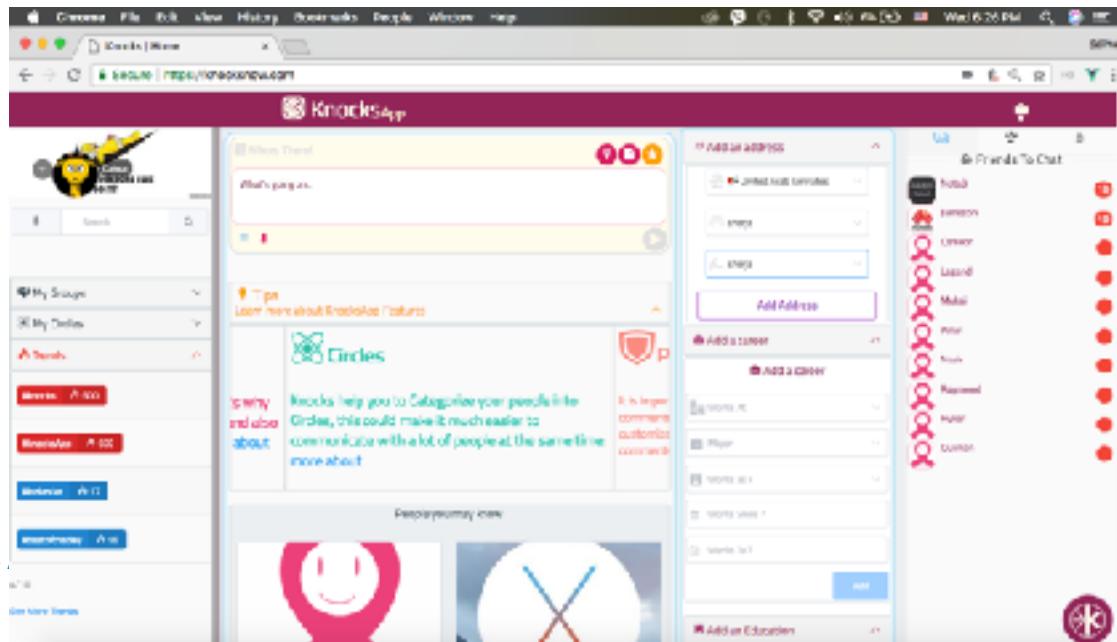
## CHAPTER 8 : USER MANUAL

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.

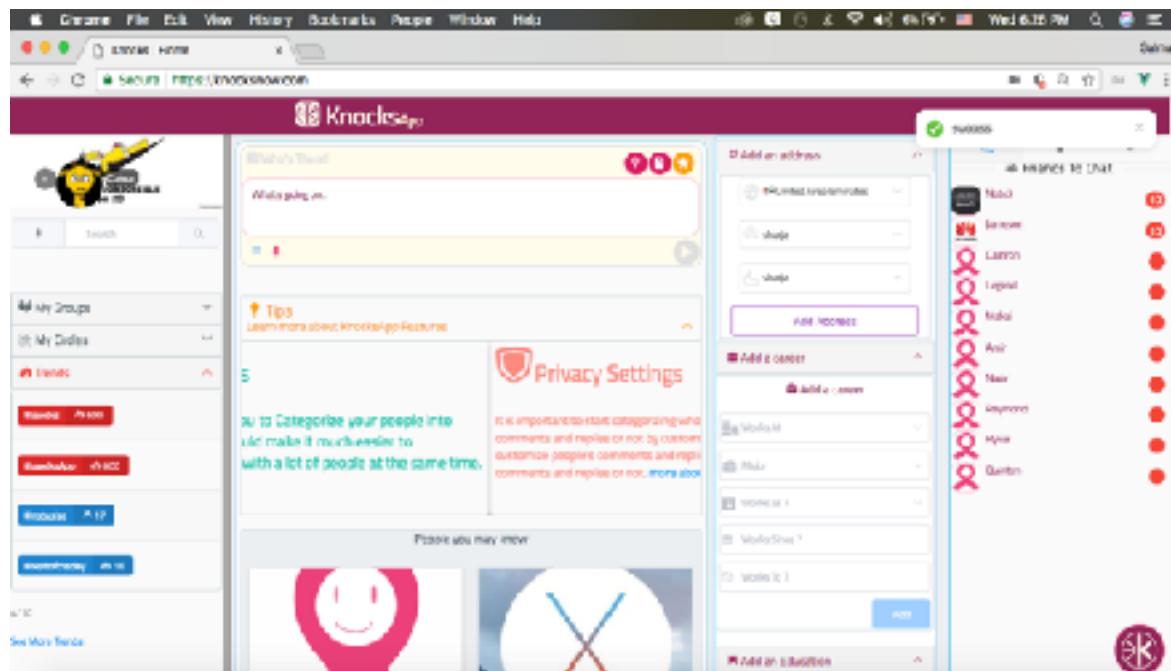


### • Add or update personal information

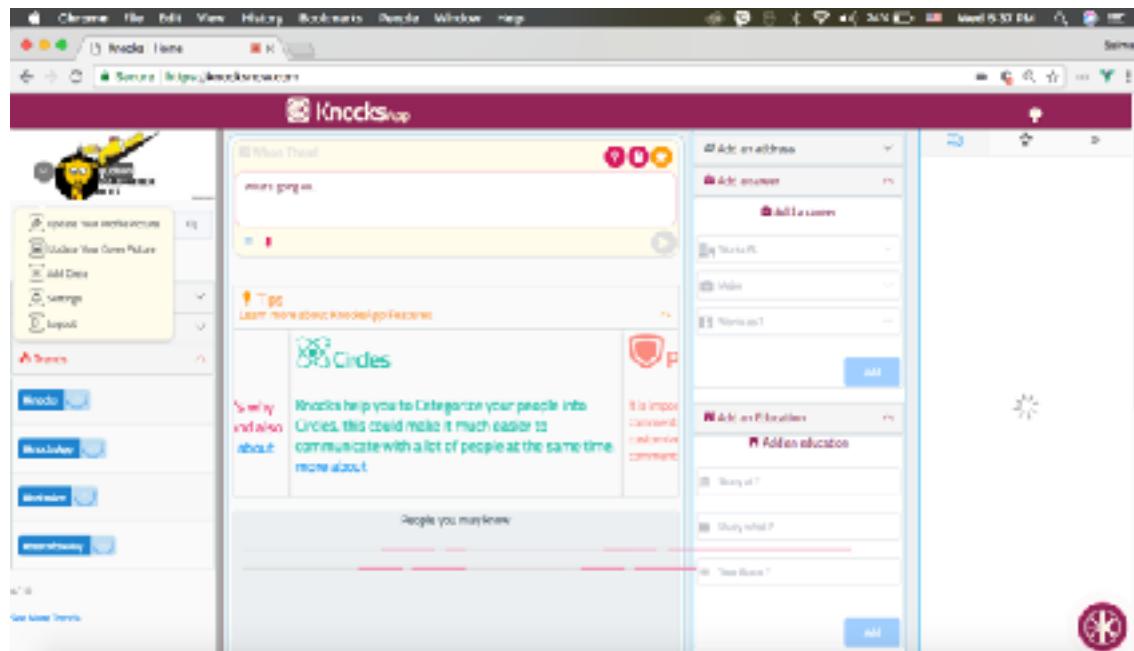
At the right corner of the home page, the user can update the personal information like the address, career, education, high education, sport and hobby, for example in adding address the user should provide the country, state and region then clicks on “Add” button.

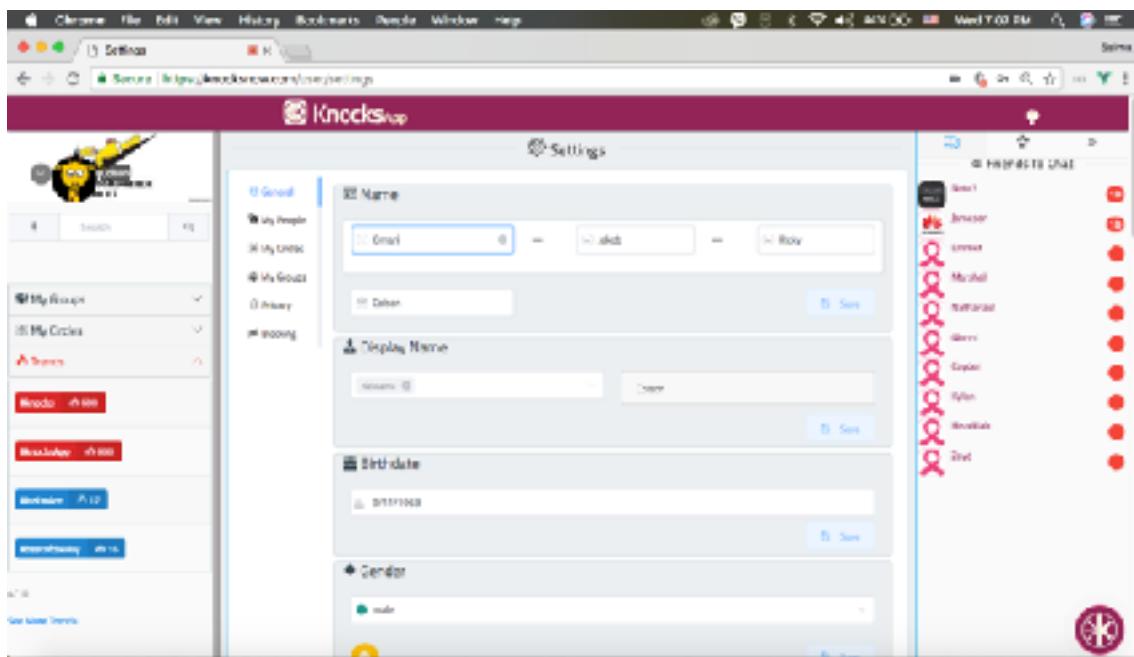


Then the system will display a message “Success” and displays the new address at the user’s profile.

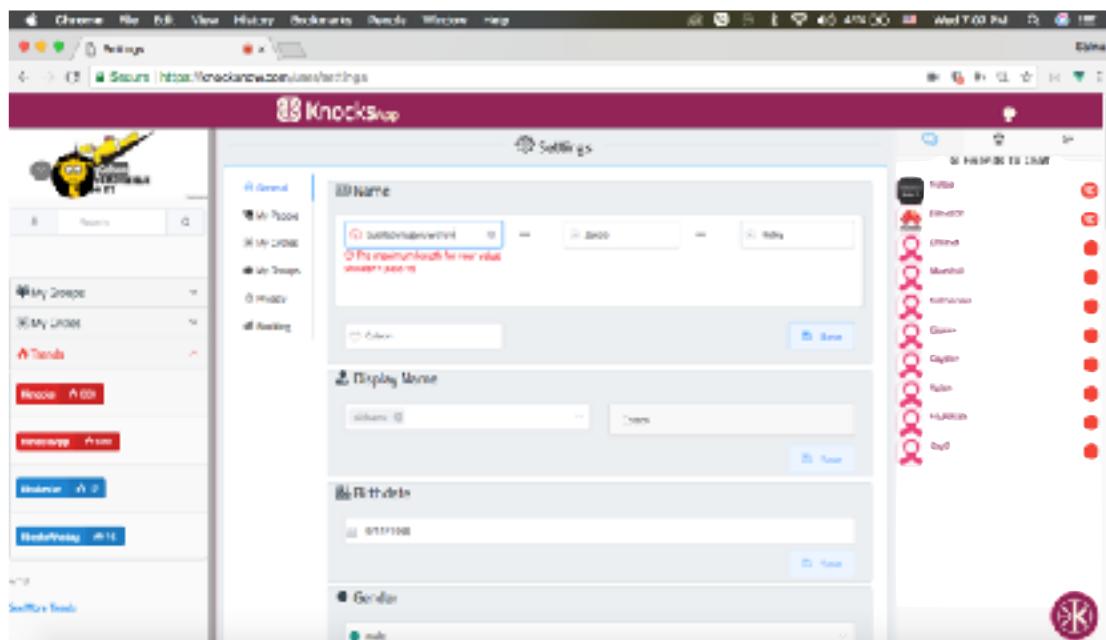


Or at the cover picture section on the top left of the page, click on the arrow icon to display a dropdown list ( Update your profile picture, Update your cover picture, Add circle, Settings, Logout ) to choose “Settings”, then he clicks on the left side bar at “General” tab, and chooses for example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



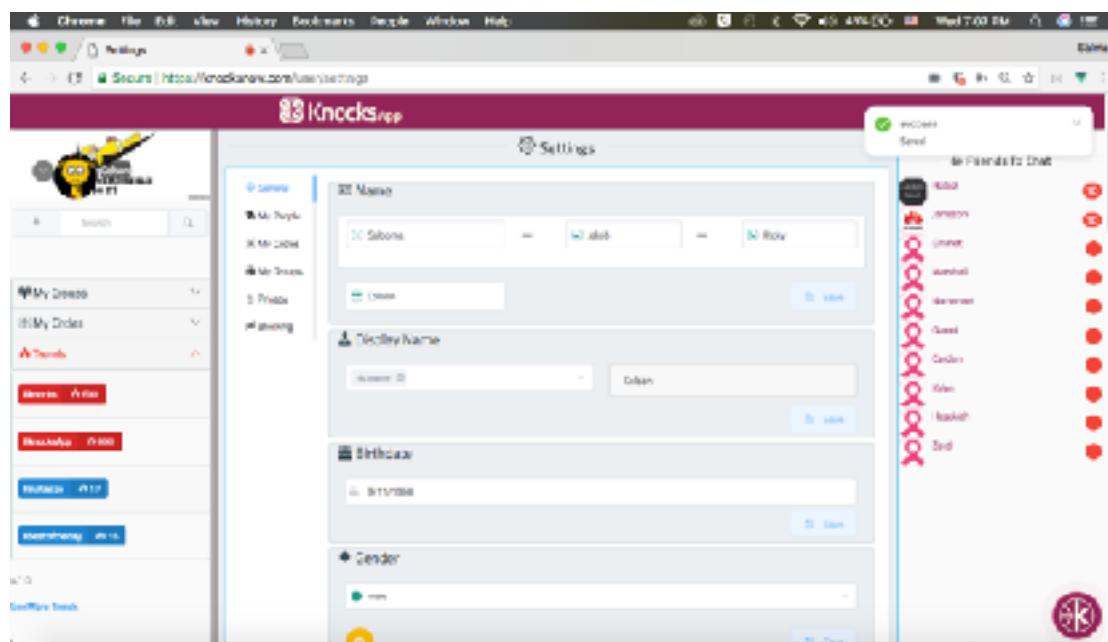


If the date are incorrect ( providing a first name that exceeds 15 characters), the system will display message “The maximum length for your value shouldn't pass 15”.



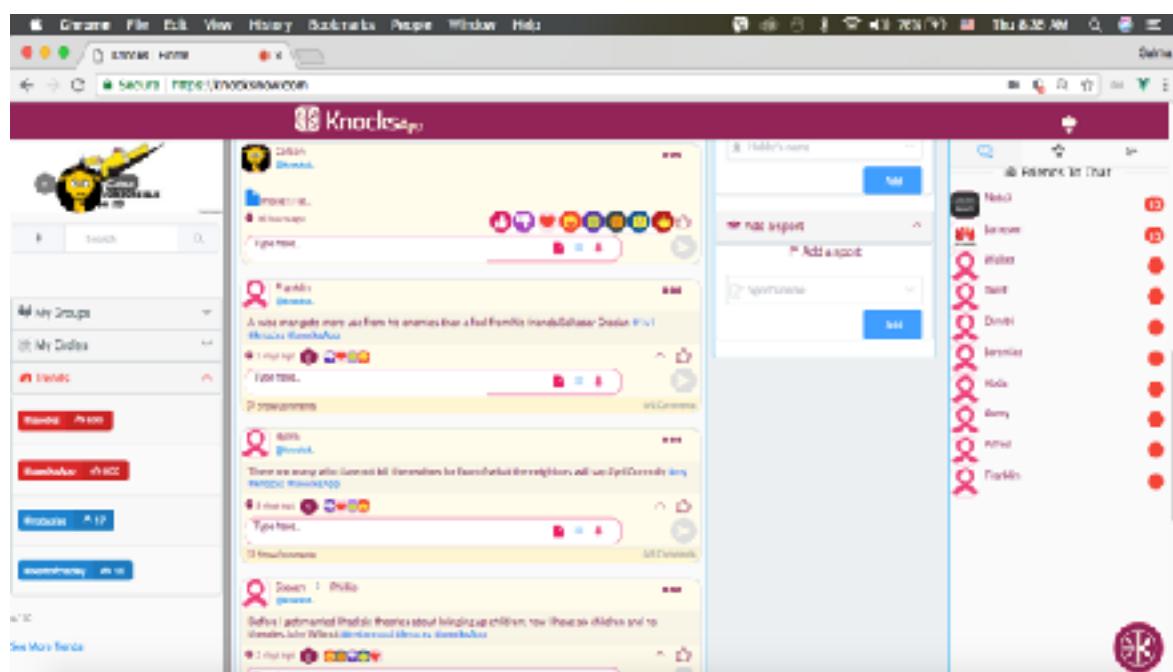
Else the system will block the selected user and display a message “Saved” and displays the the new first name for the user.

## CHAPTER 8 : USER MANUAL

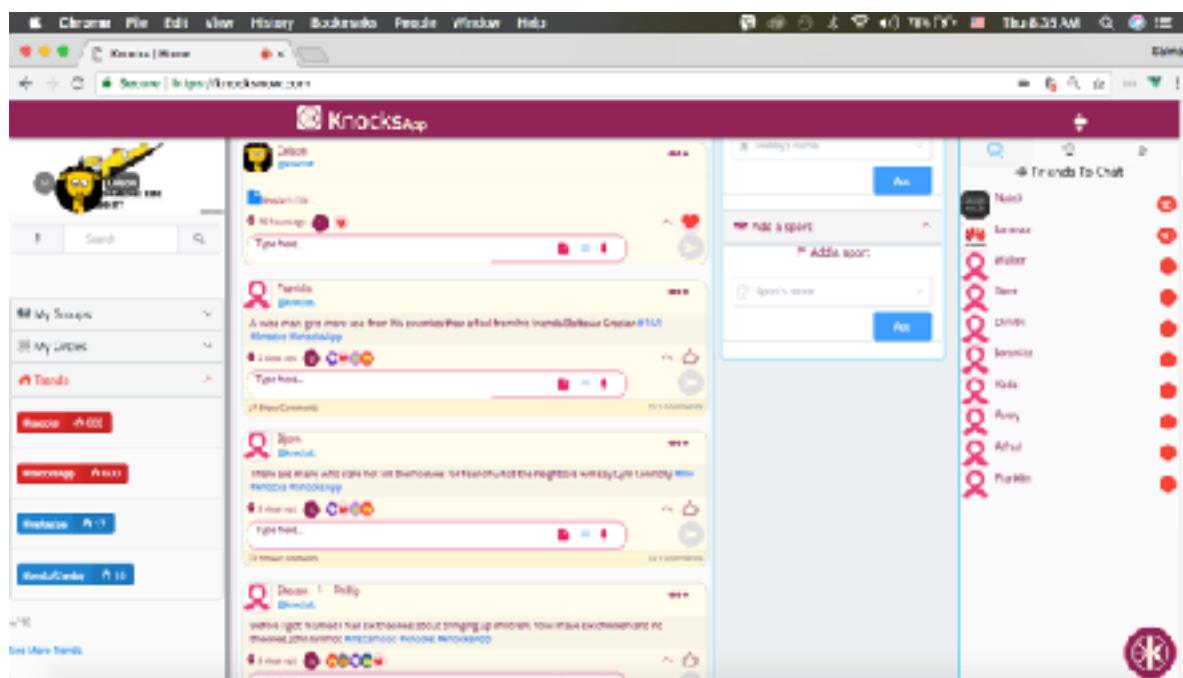


- **React to post- when allowed to do so with allowed reactions.**

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

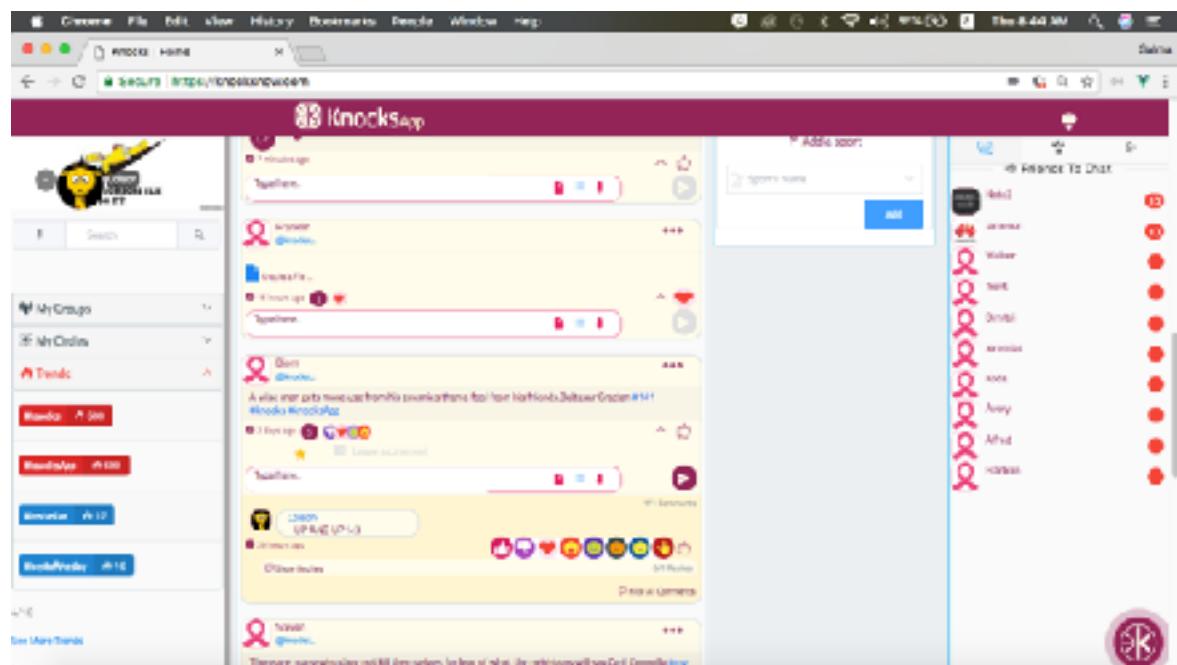


Then the system will display the new reaction towards the post and updates the reactions counter of the knock.

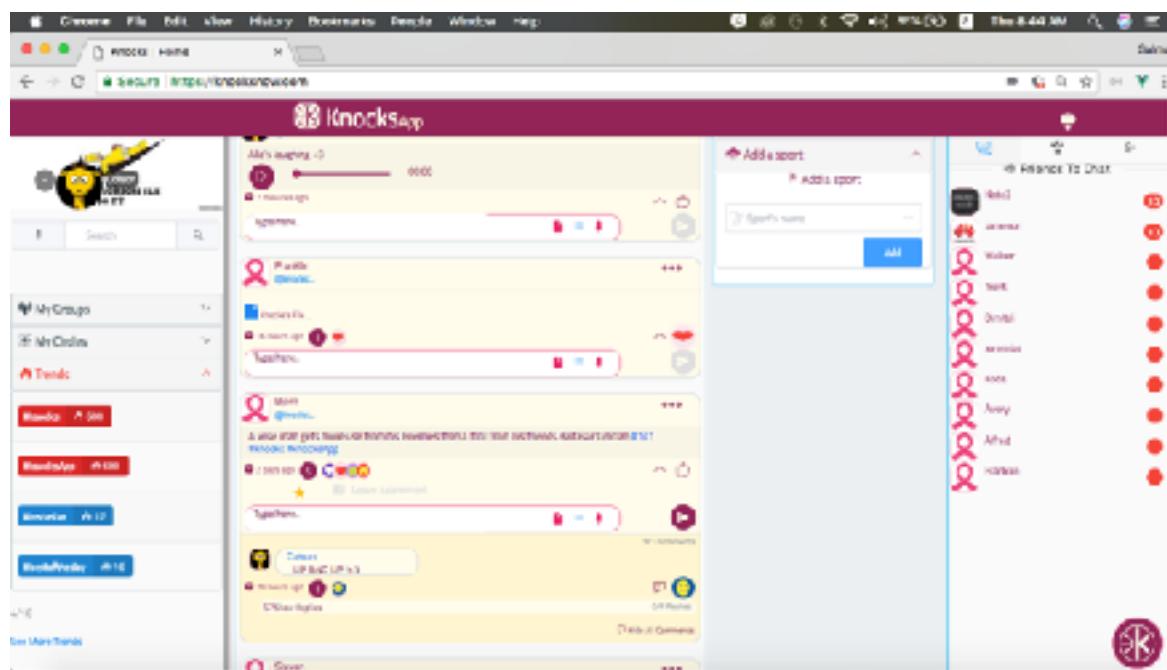


- React to comment- when allowed to do so with allowed reactions.**

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

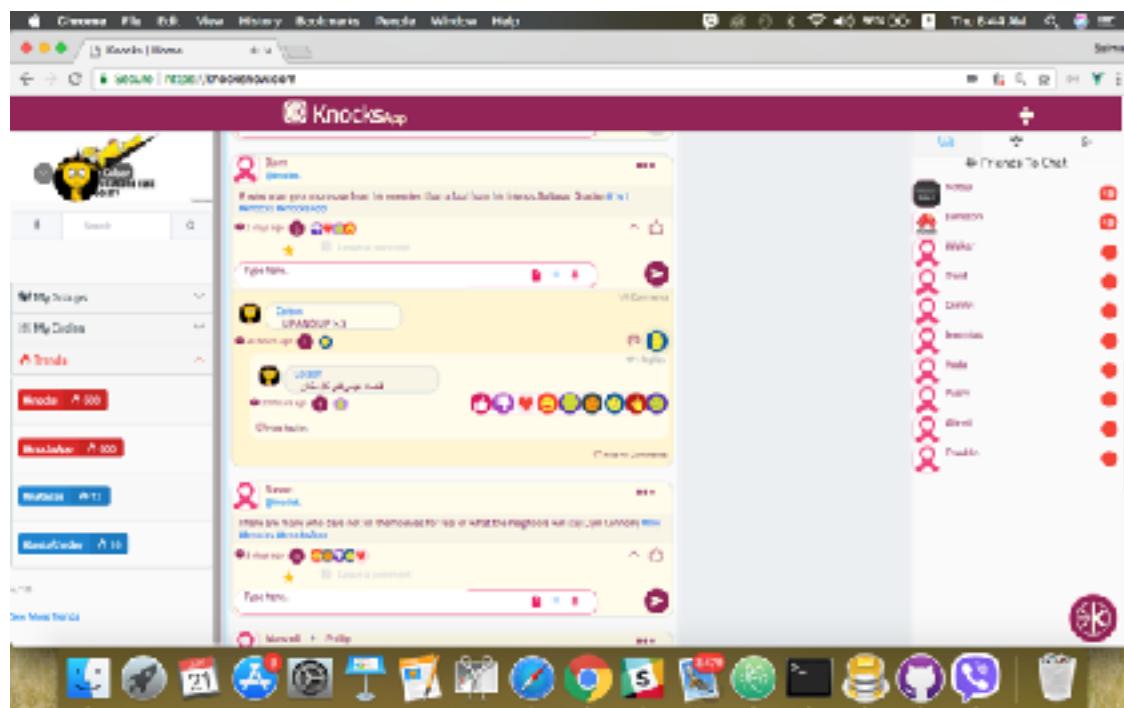


Then the system will display the new reaction towards the comment and updates the reactions counter of the comment.

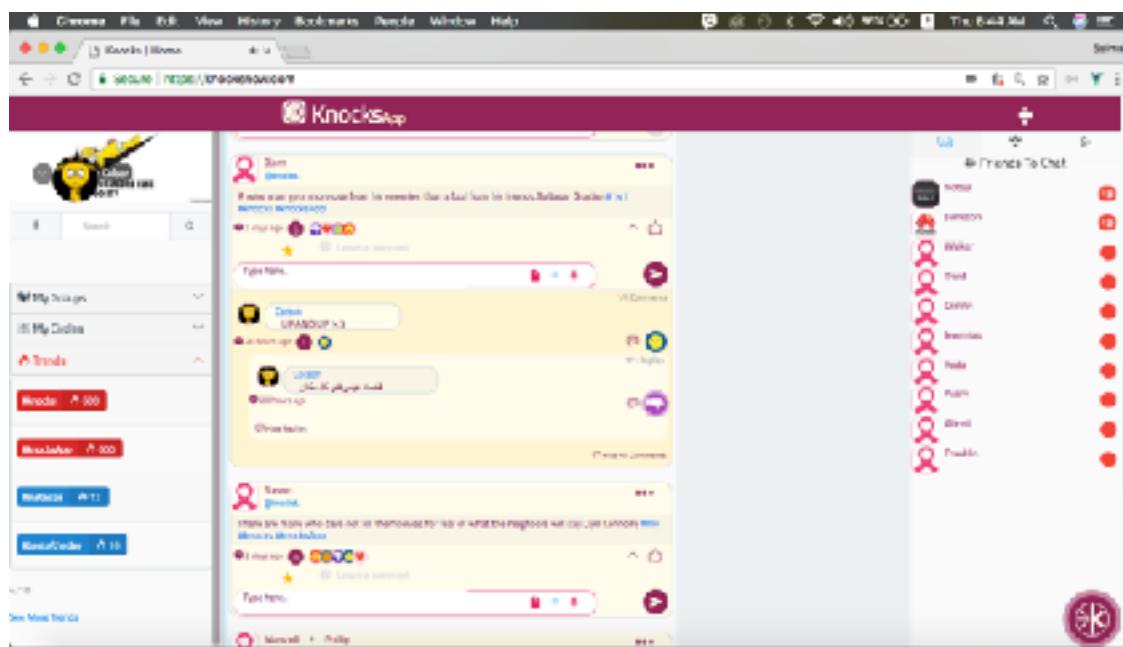


- React to reply- when allowed to do so with allowed reactions.**

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

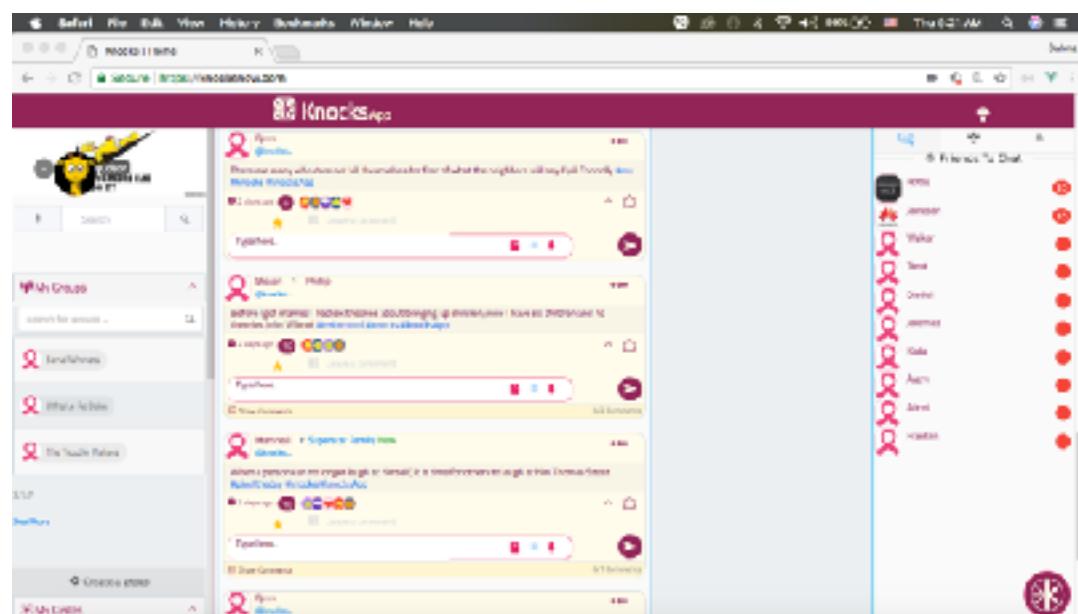


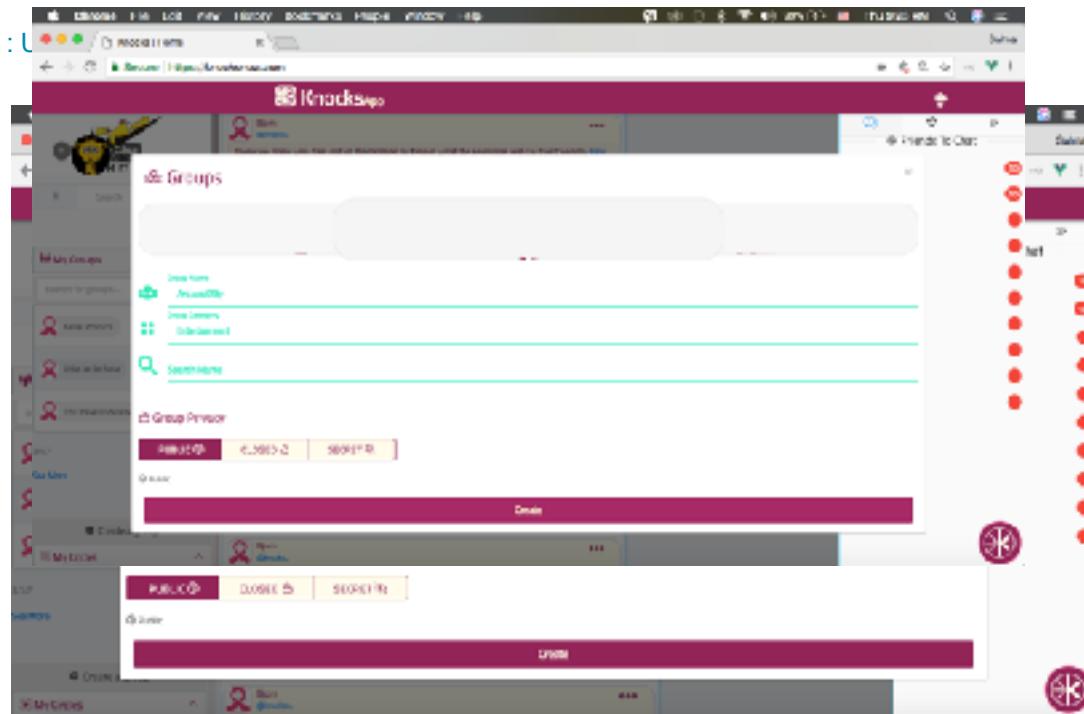
Then the system will display the new reaction towards the reply and updates the reactions counter of the reply.



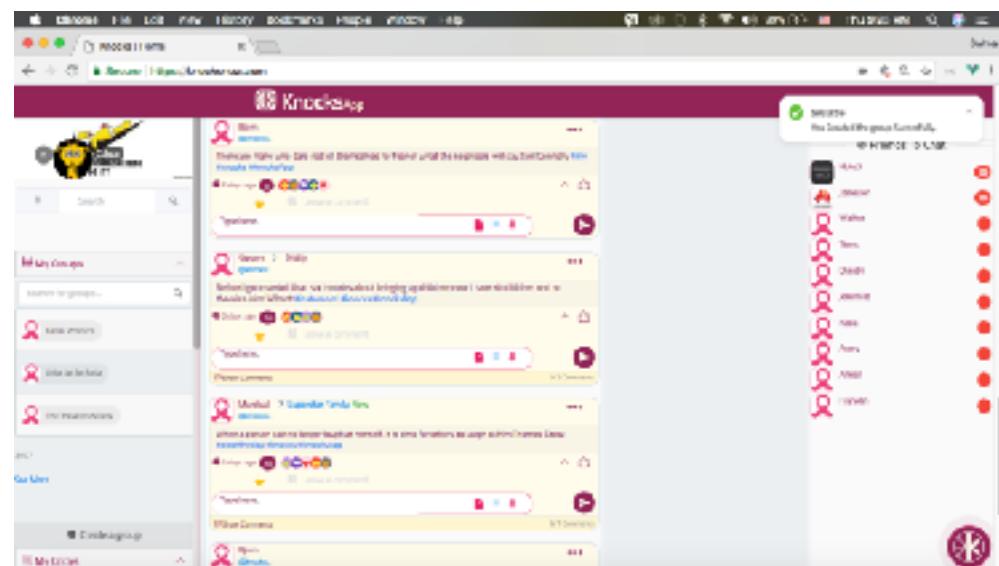
- **Create a group**

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”.



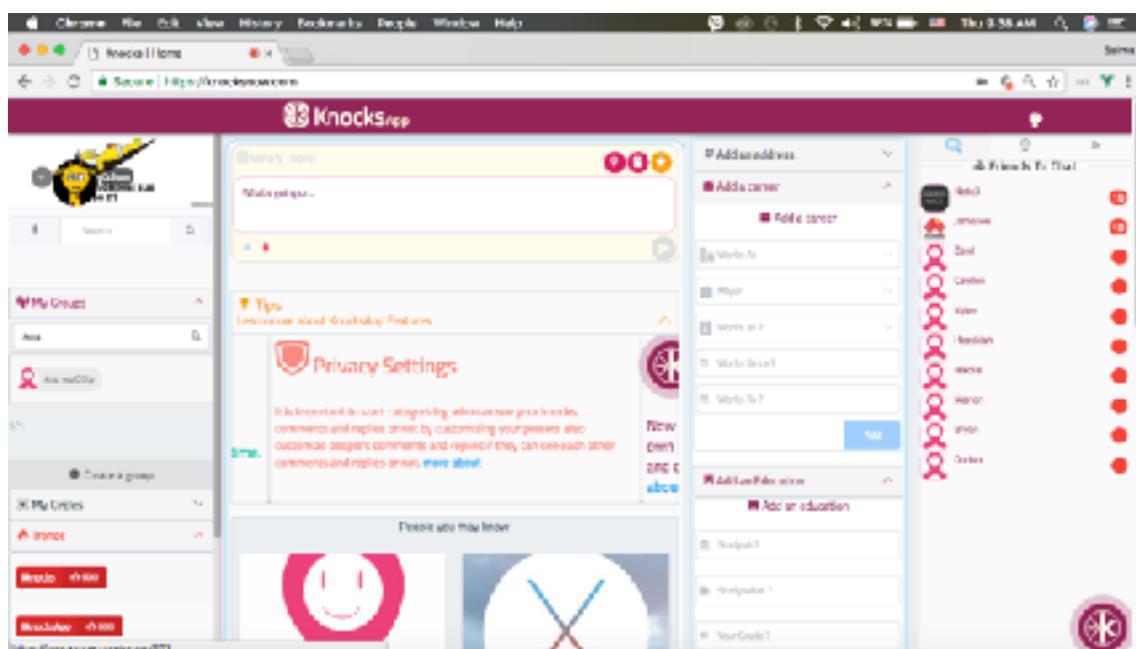


Then the system will display a message “You created the group successfully” and displays the newly created group in the groups section

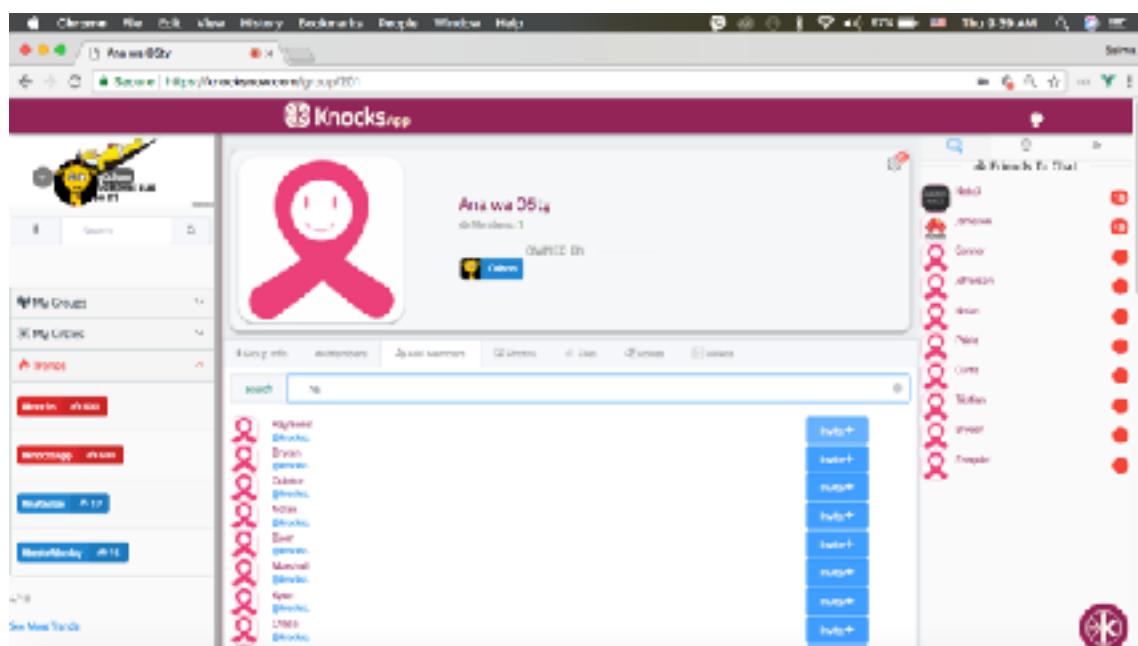


- Add friends to group**

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.

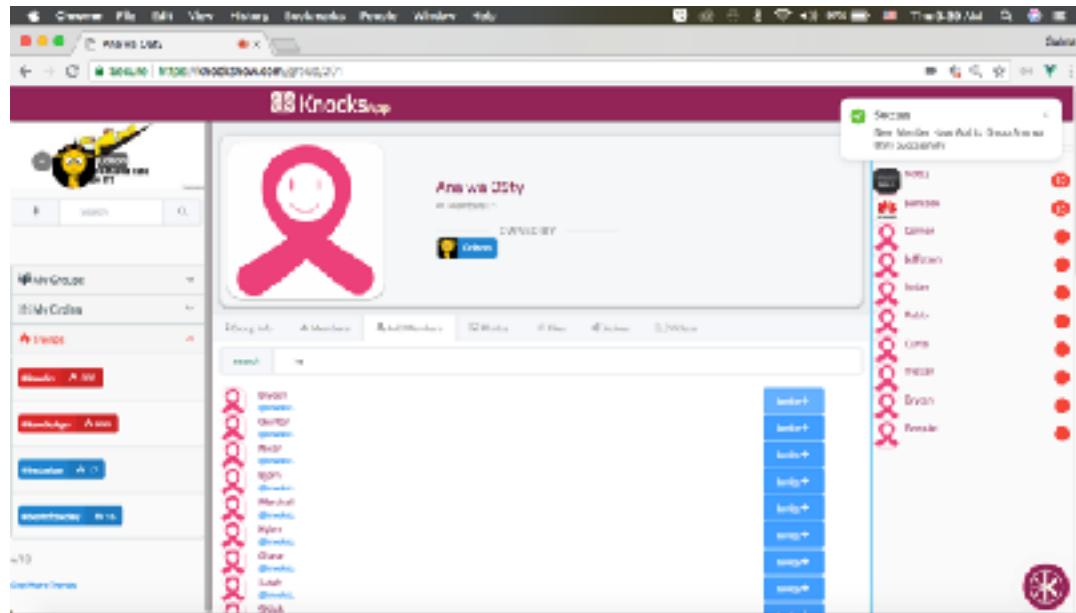


Then click on the group name chip, it will direct the user to the group page.



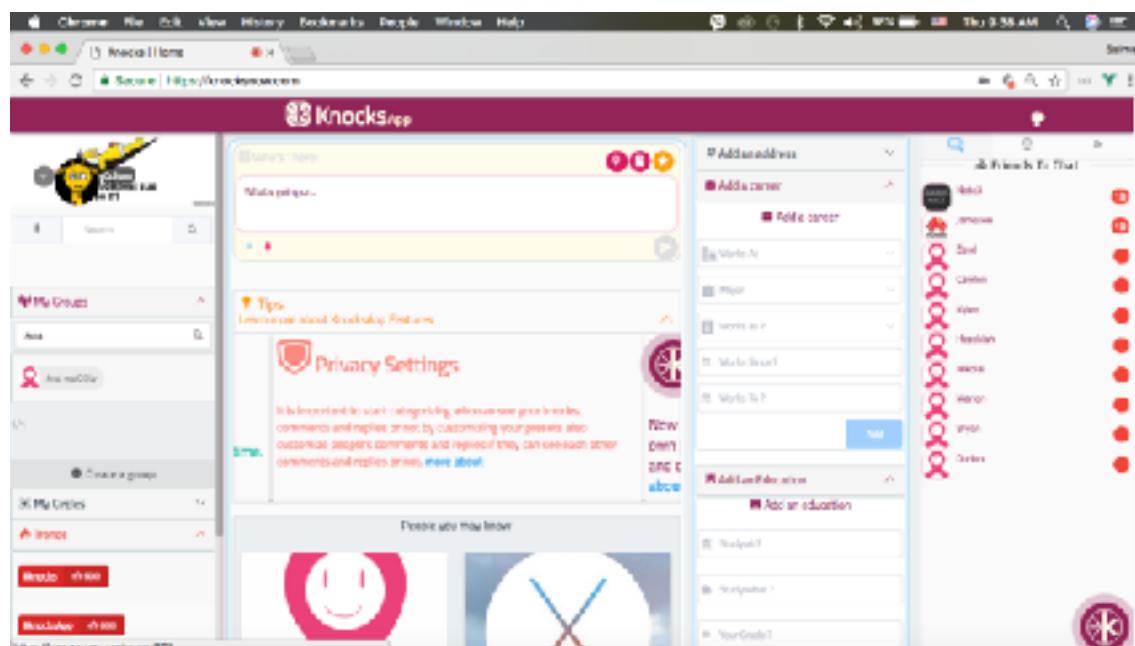
## CHAPTER 8 : USER MANUAL

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”.



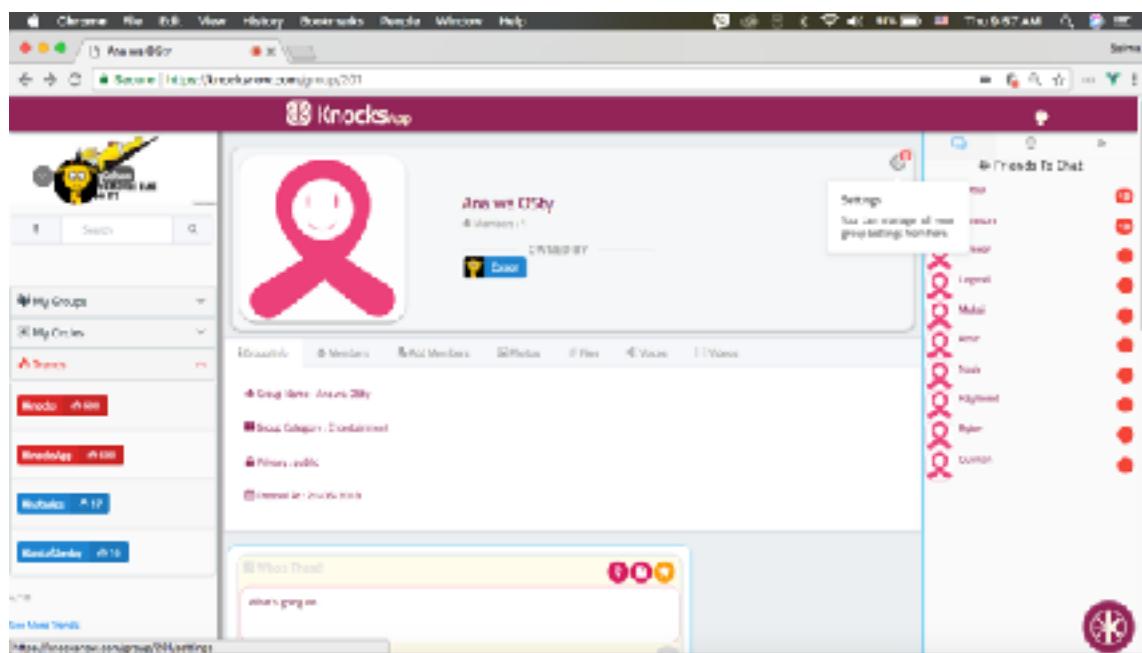
- **Update a group- when allowed to do so.**

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.

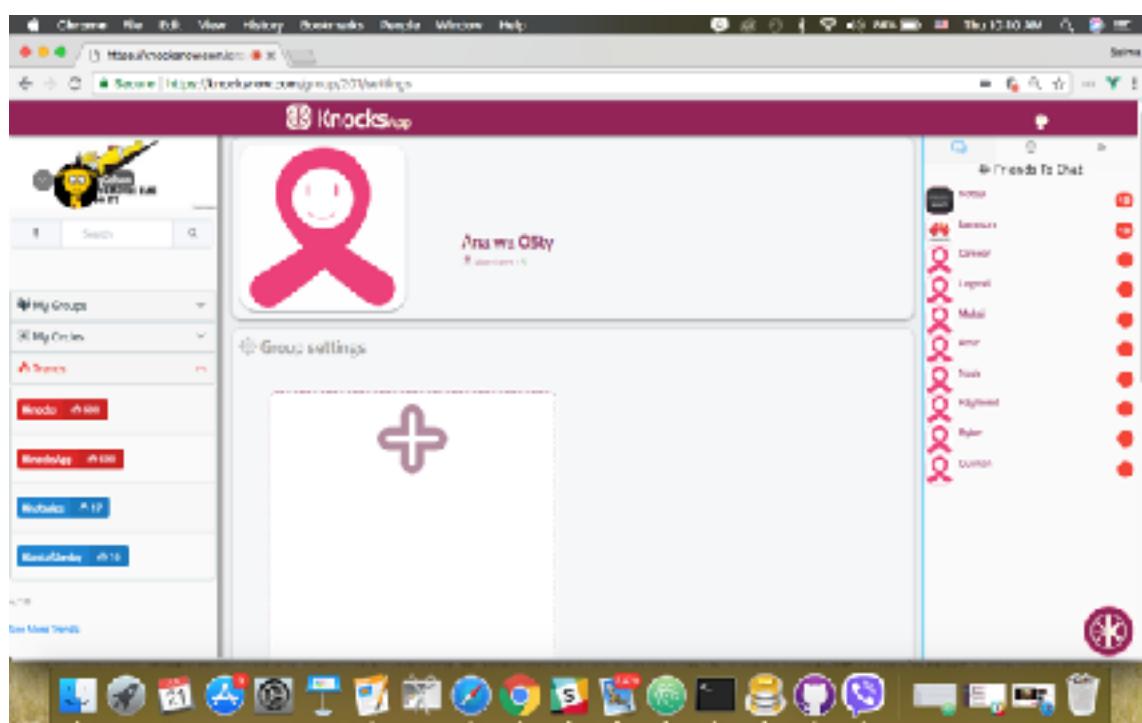


Then click on the group name chip, it will direct the user to the group page, then click on the settings icon.

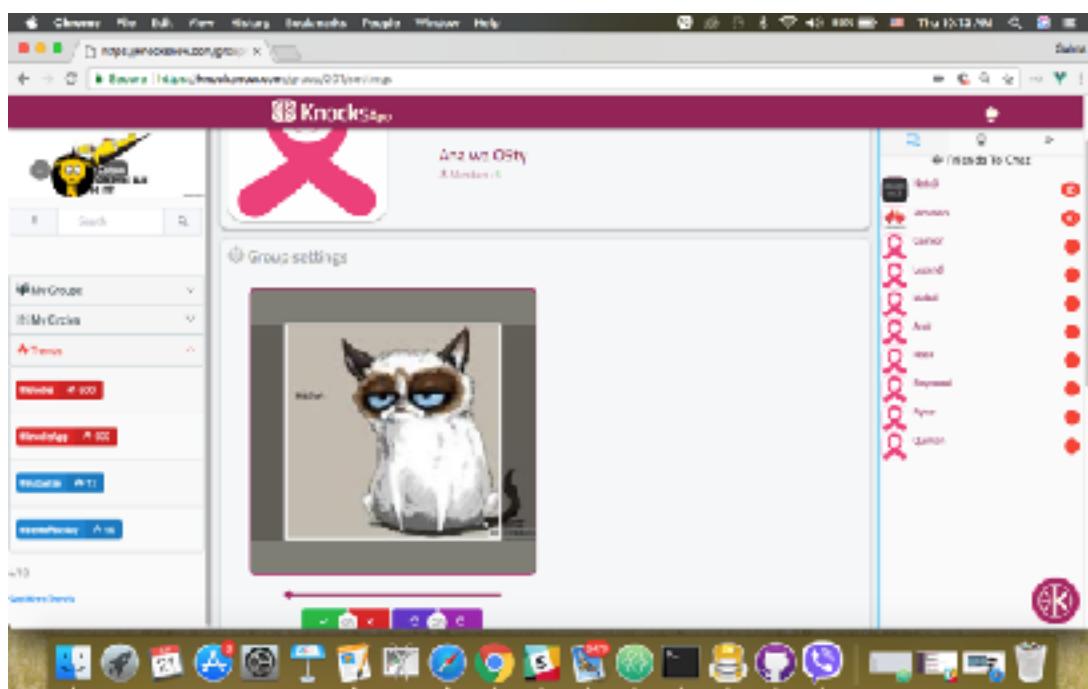
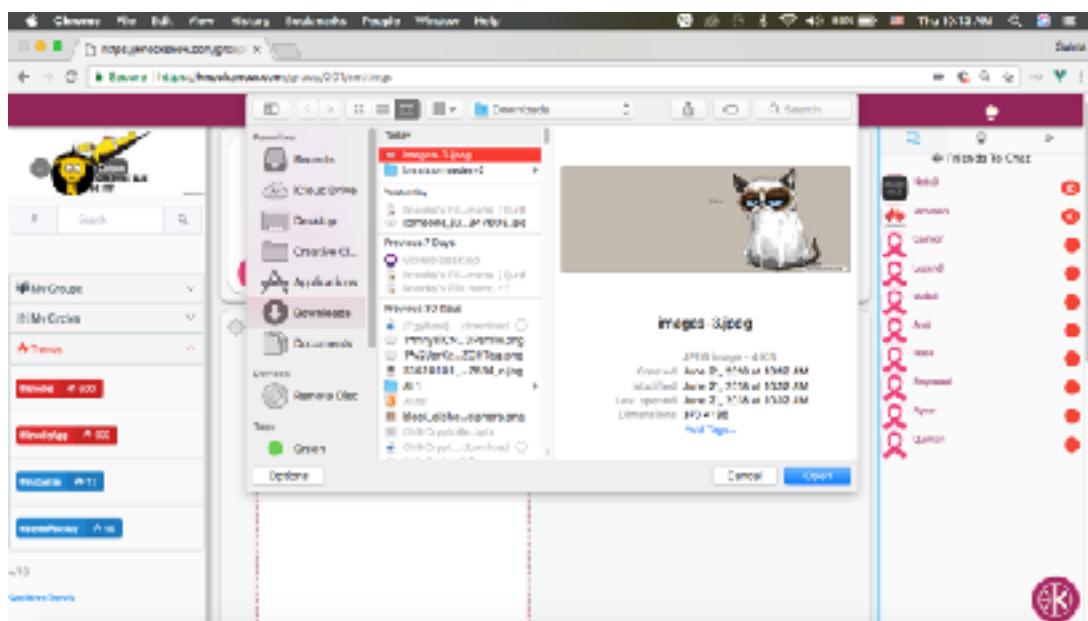
## CHAPTER 8 : USER MANUAL



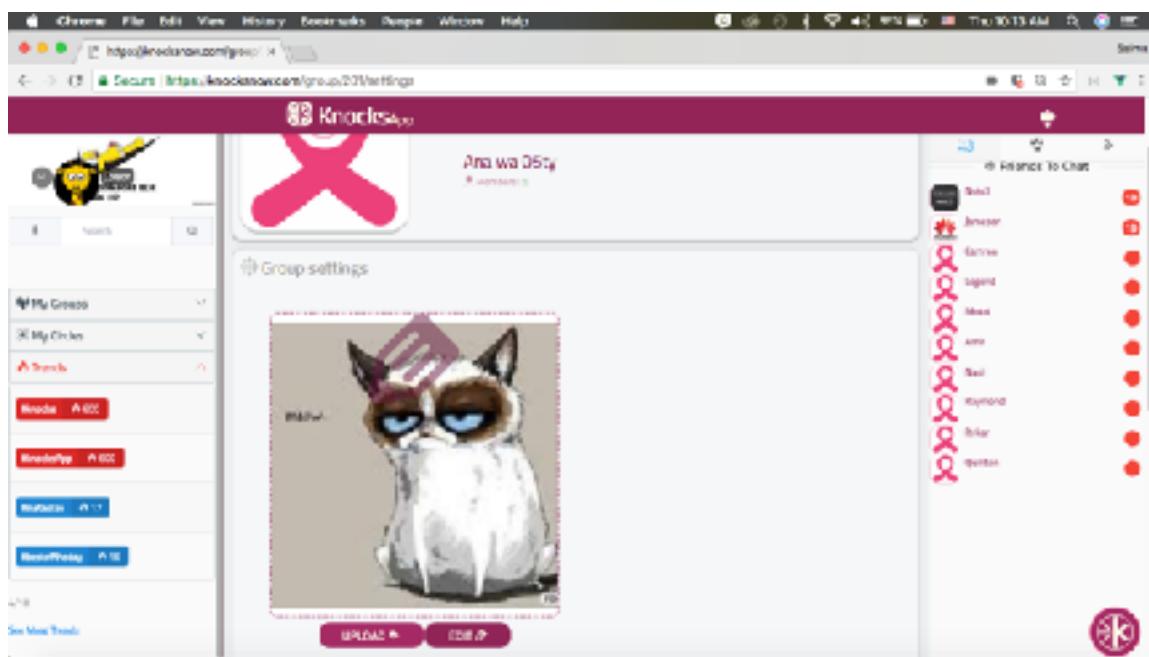
Then click on the change group picture tab and choose the picture to be updated with clicking on the add icon.



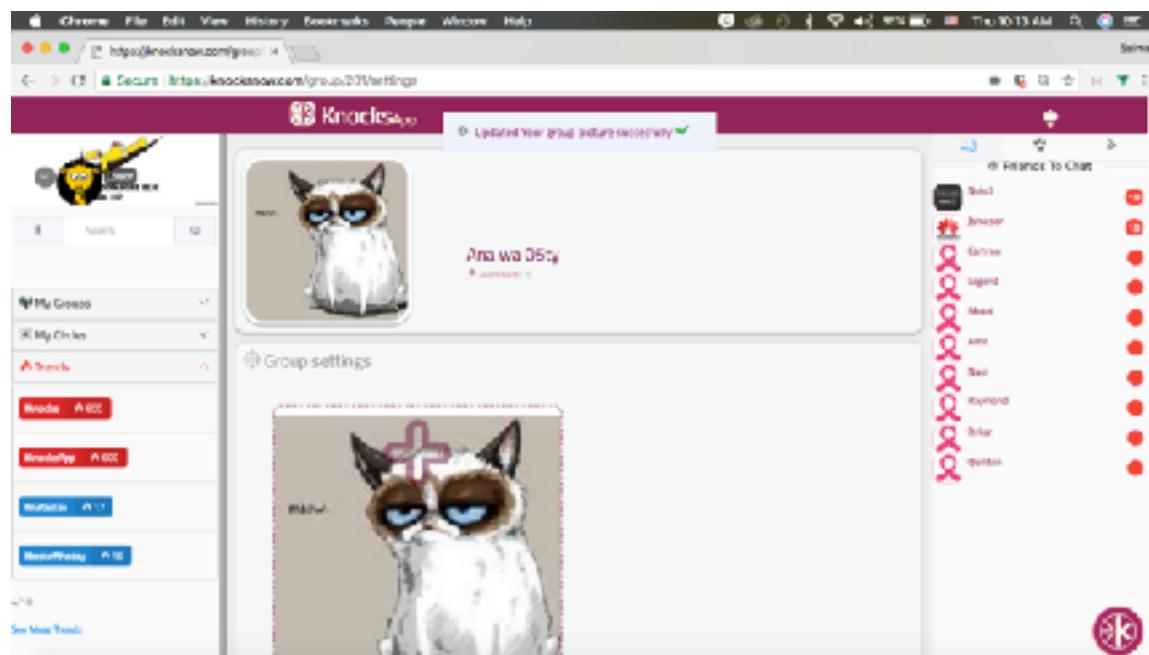
## CHAPTER 8 : USER MANUAL



## CHAPTER 8 : USER MANUAL



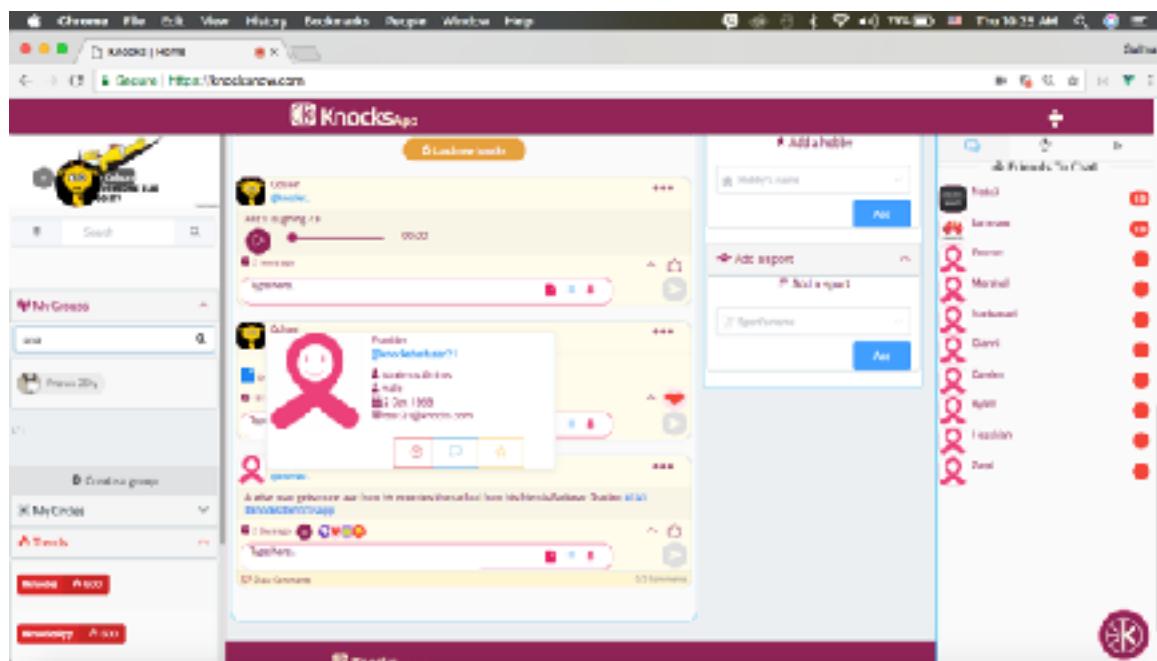
Then the System will display message “Updated your group picture successfully”, and changes the group picture into the new one.



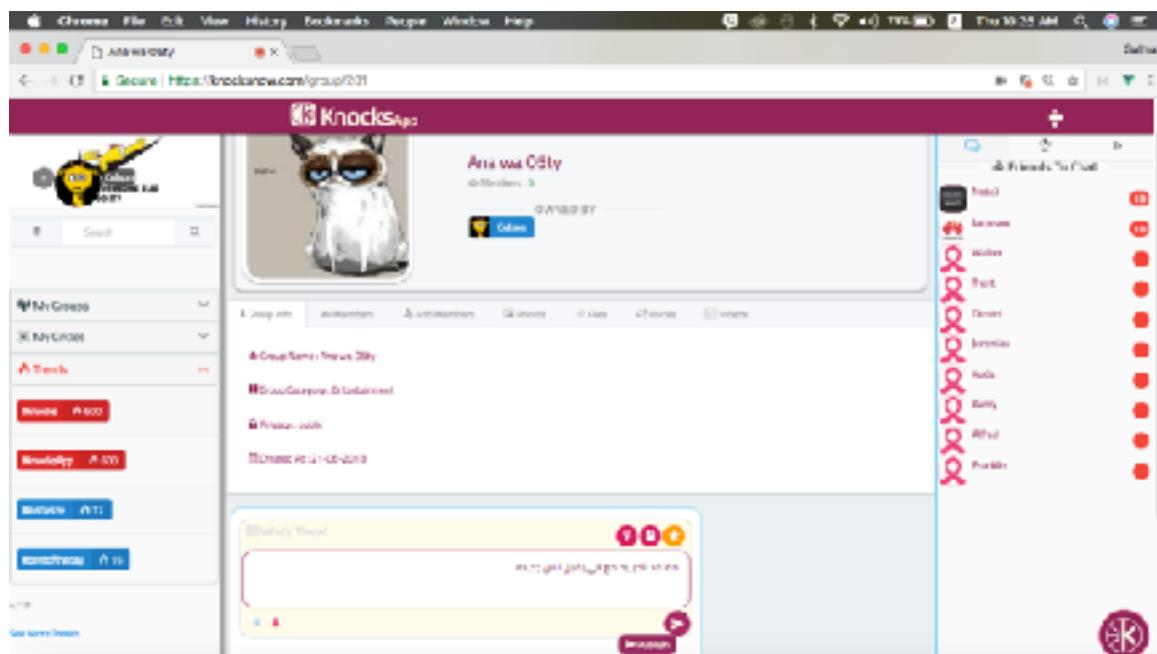
- Post in a group- when allowed to do so.

## CHAPTER 8 : USER MANUAL

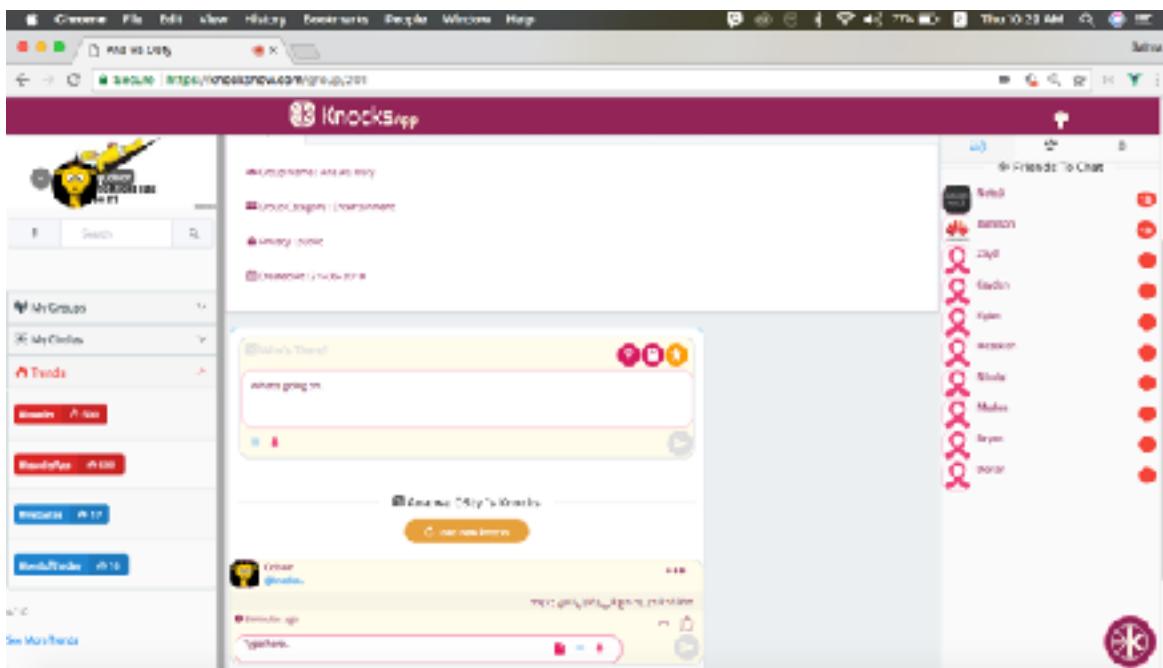
At the left side of the home page in groups section, click on “My Groups” tab and search for a group that he can post at.



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”.

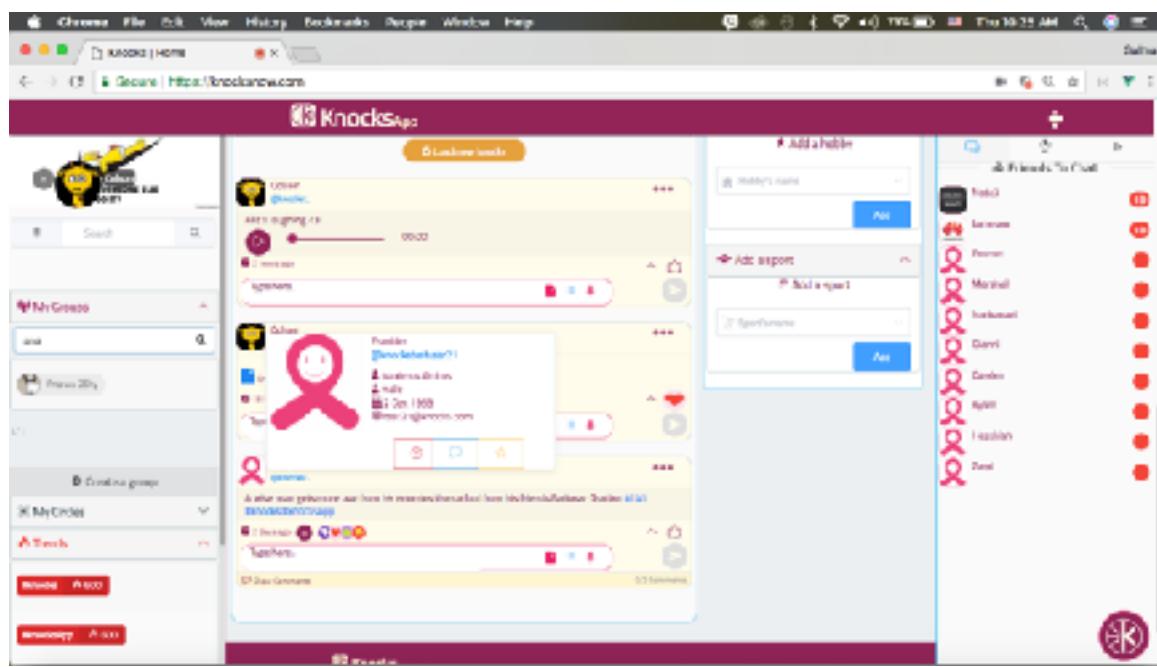


Then the System will display message “success”, and adds your post to the group knocks.



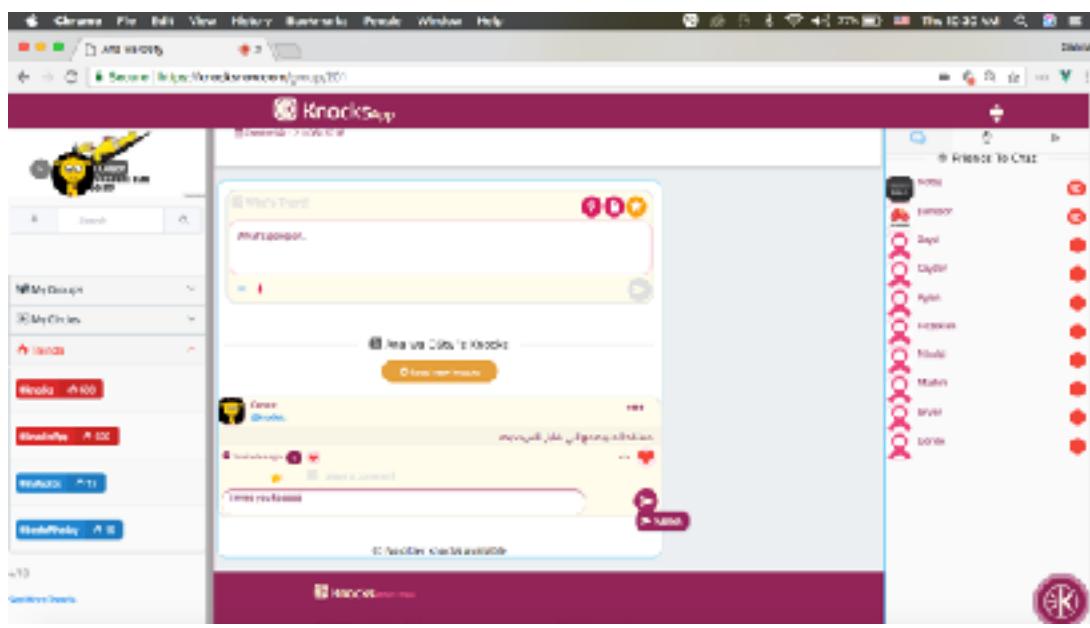
- **Comment in a group- when allowed to do so.**

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.



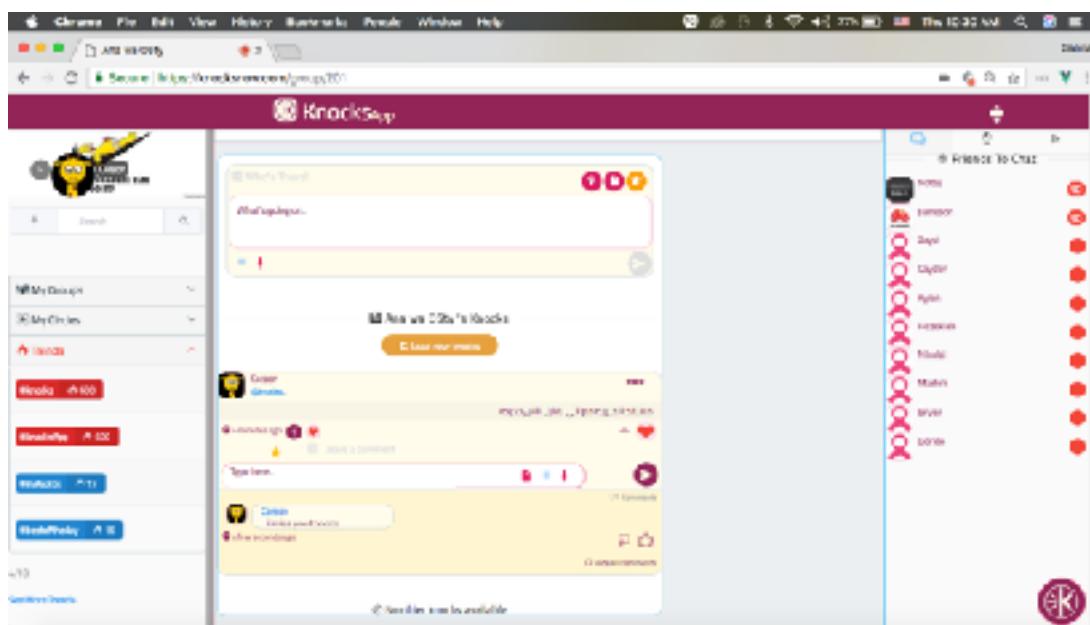
Then click on the group name chip, it will direct the user to the group page, then create a comment for a post in the group and click “Publish”.

## CHAPTER 8 : USER MANUAL



Then  
the

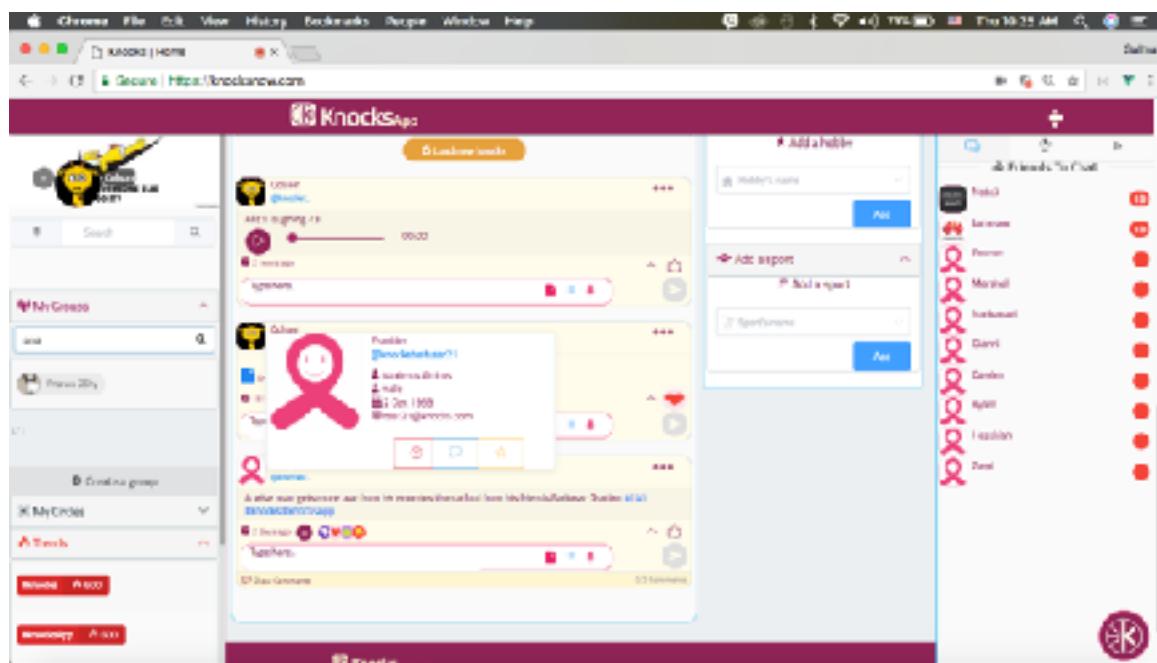
System will display message “success”, and adds your comment to the post comments in the group.



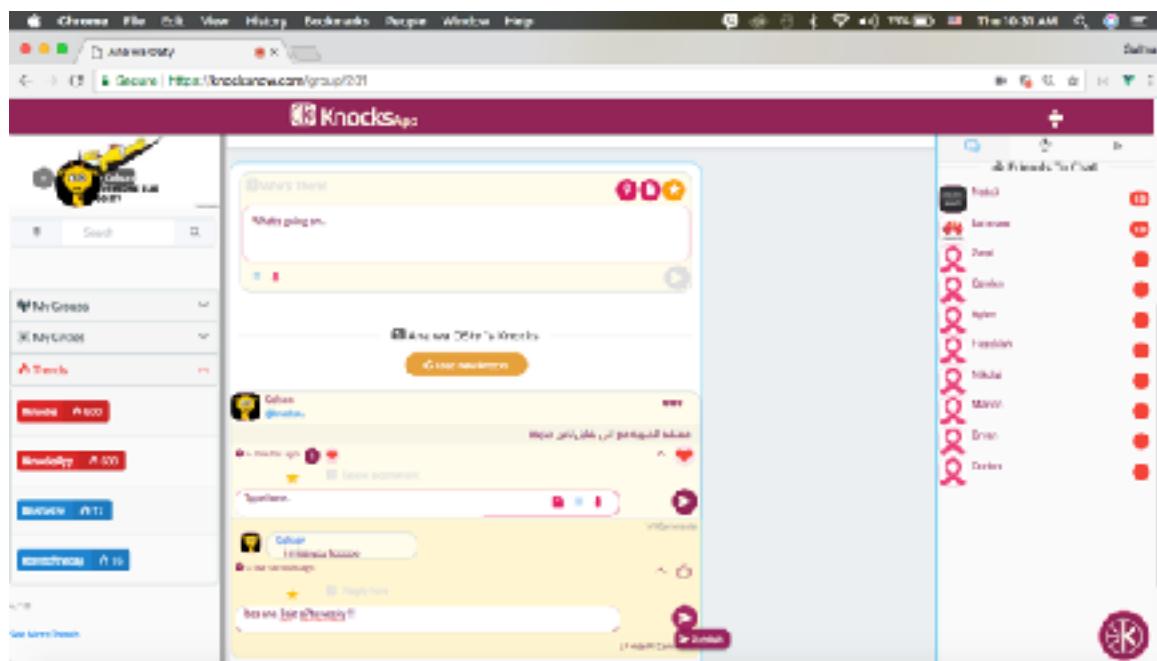
- **Reply in a group- when allowed to do so.**

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.

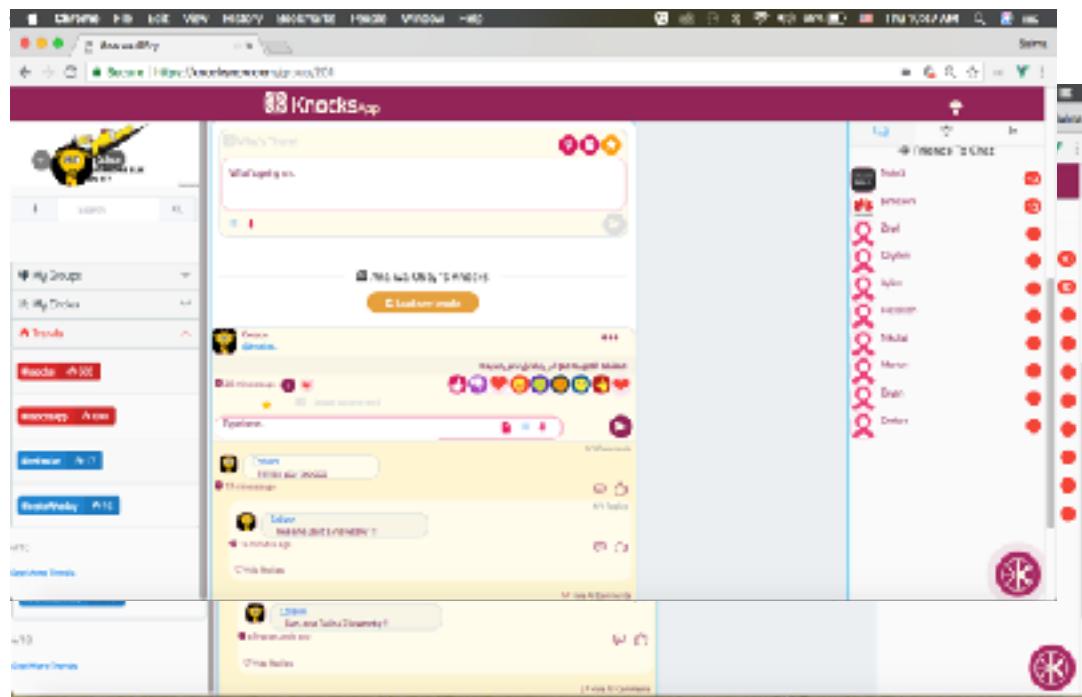
## CHAPTER 8 : USER MANUAL



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

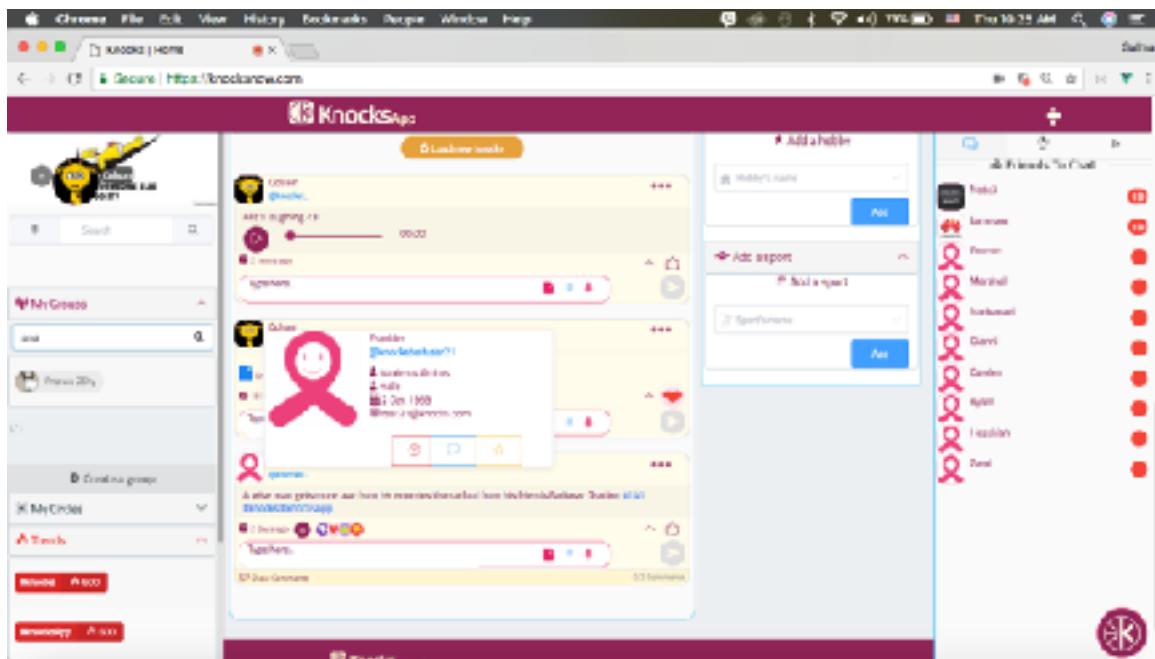


Then the System will display message “success”, and adds your reply to the comment of the post in the group.



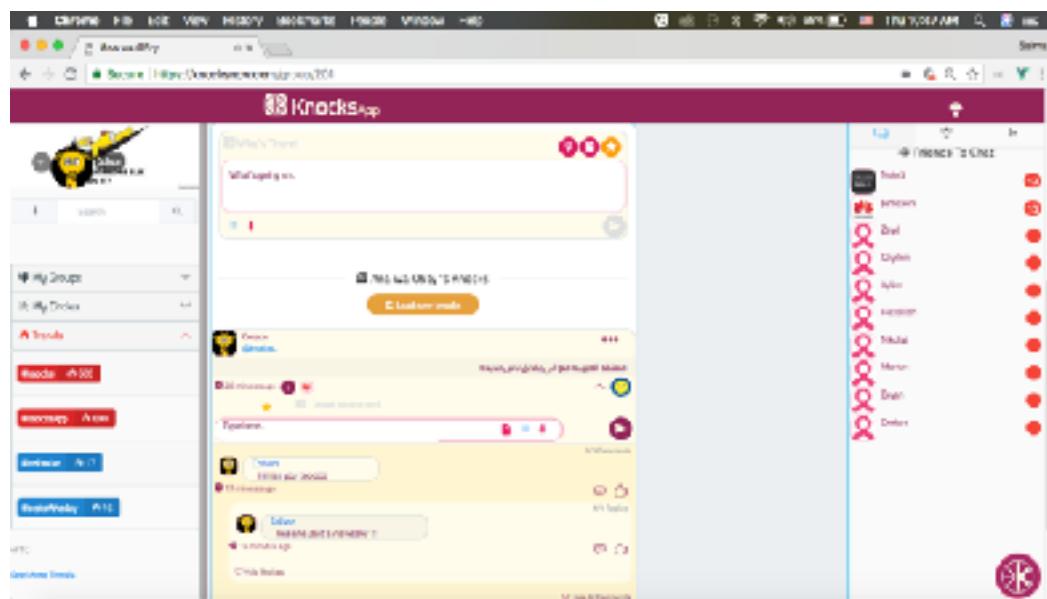
- React to an object a group- when allowed to do so ( Eg : Post Reaction ).

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.



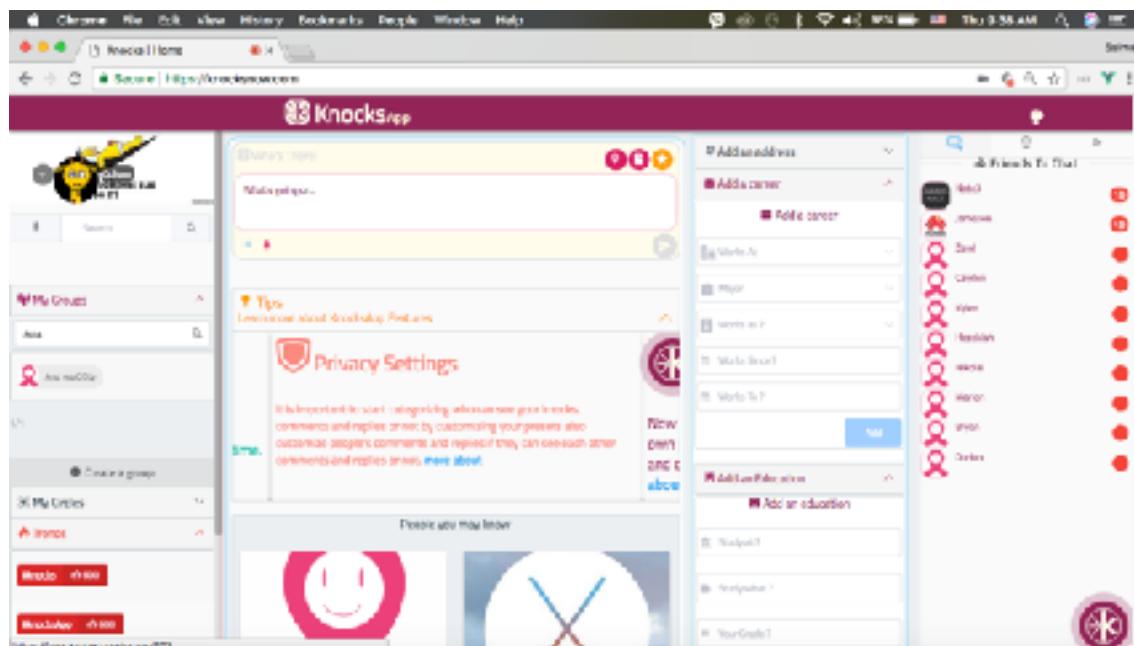
In any Knock of the group knocks where the user desires to react, at right of the the knock section he should hover or click on the reaction icon, then he clicks on the reaction

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



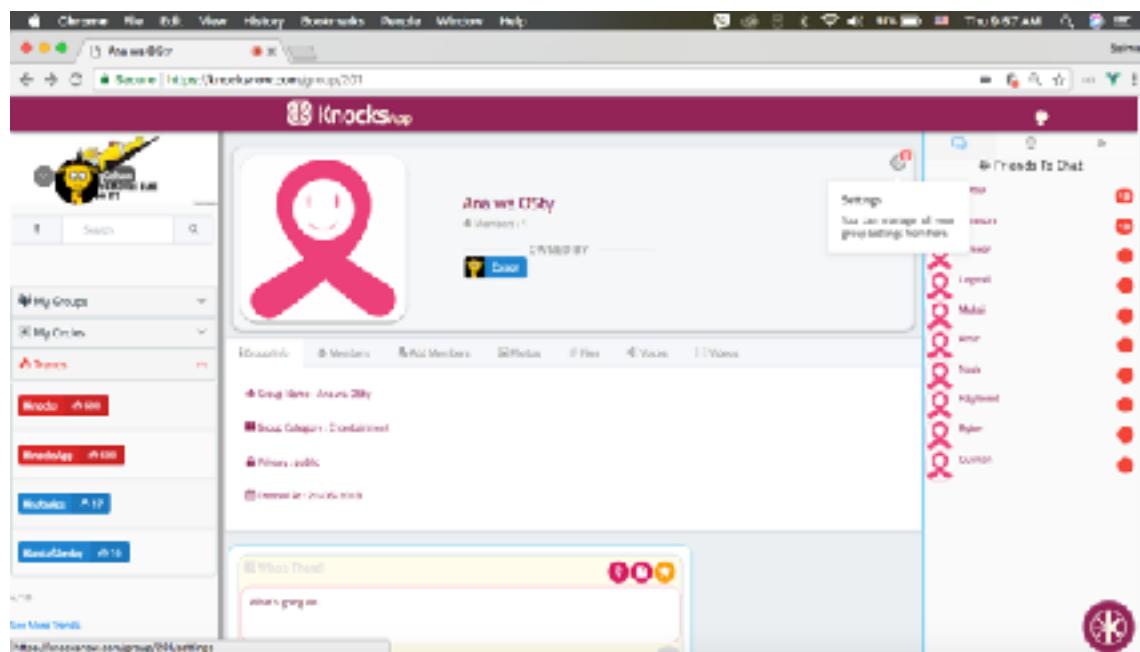
- **Change user permission- when allowed to do so.**

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.

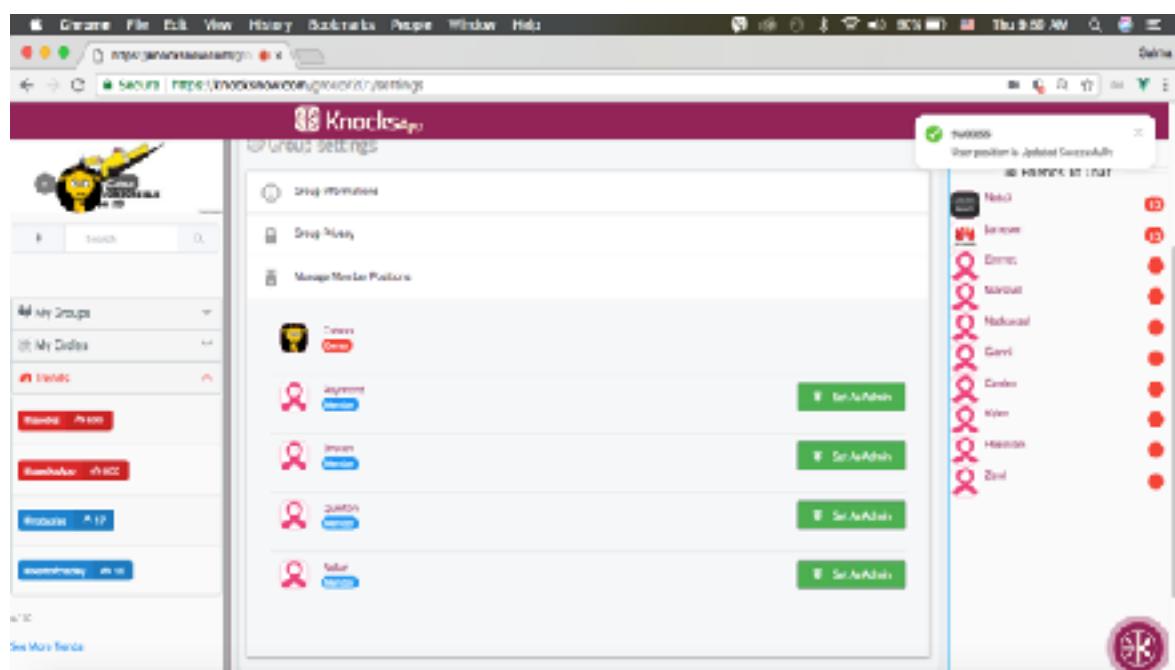


## CHAPTER 8 : USER MANUAL

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



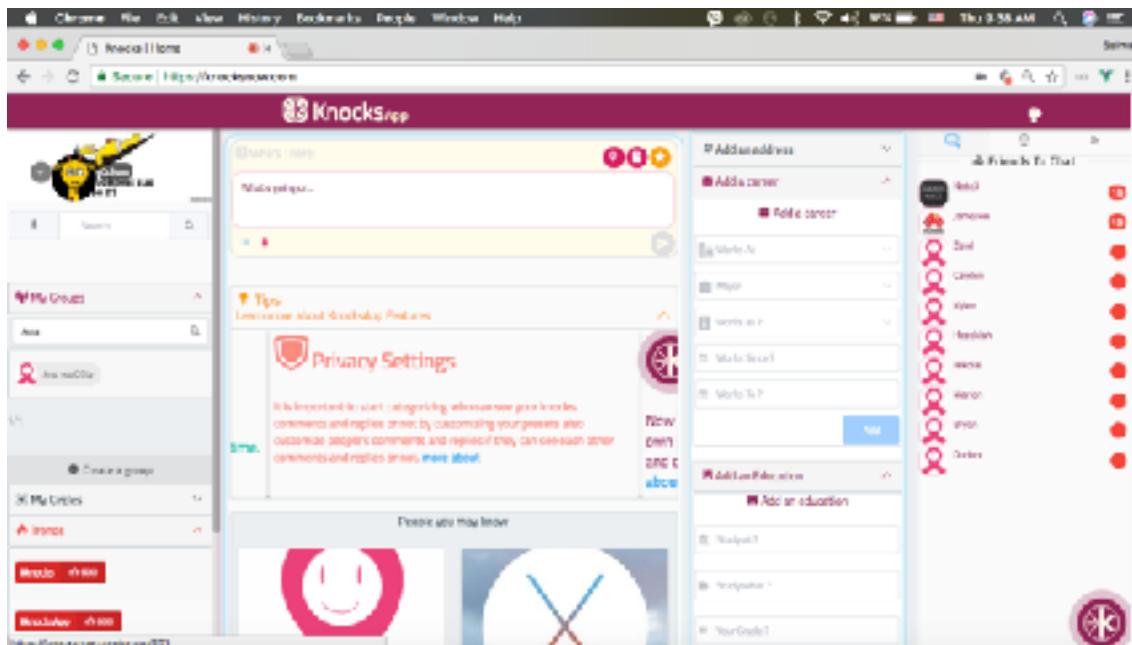
Then click on “Manage Member Positions” and click on “Set as Admin” for specific user, then the system will display a message “User Position is updated successfully”.



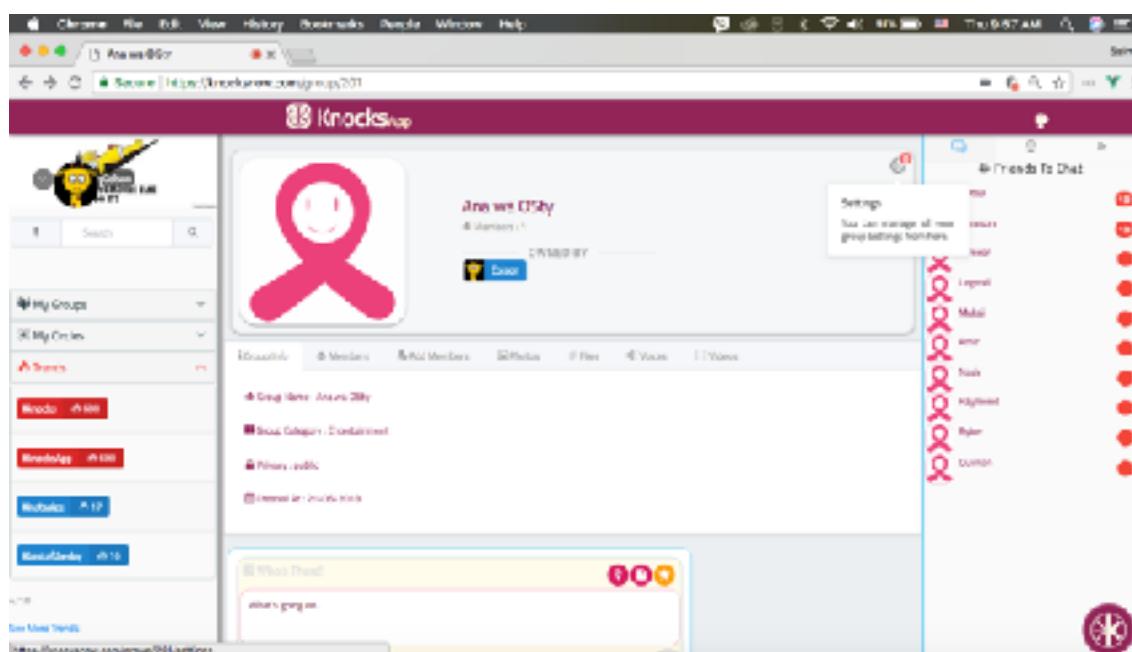
## CHAPTER 8 : USER MANUAL

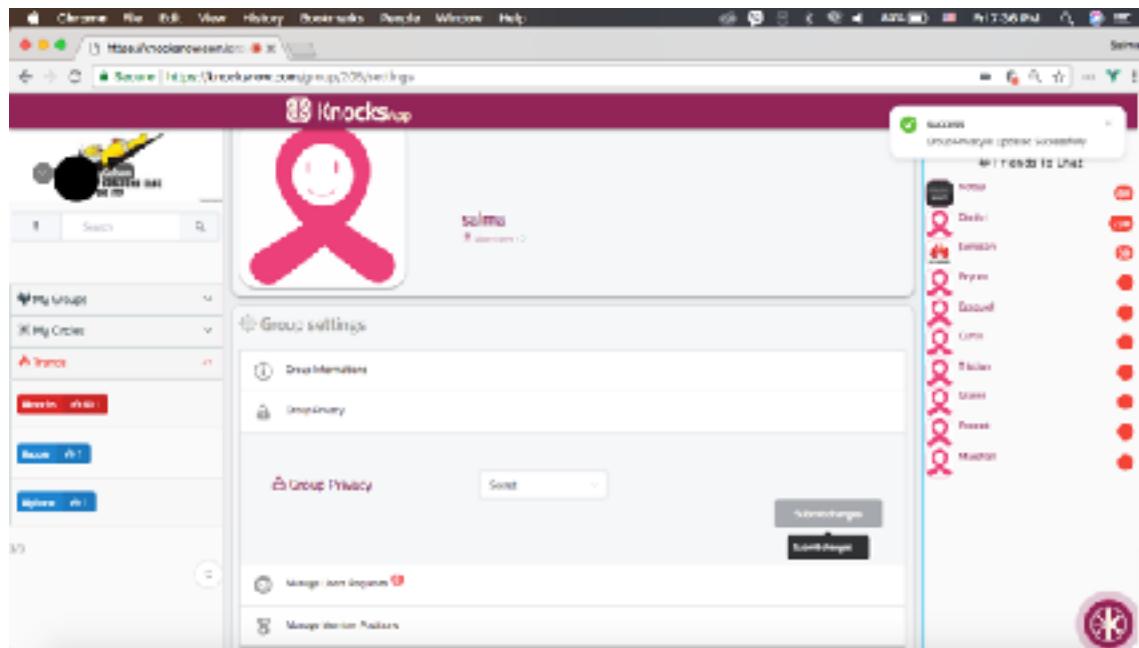
- Change the group privacy set - when allowed to do so.

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.



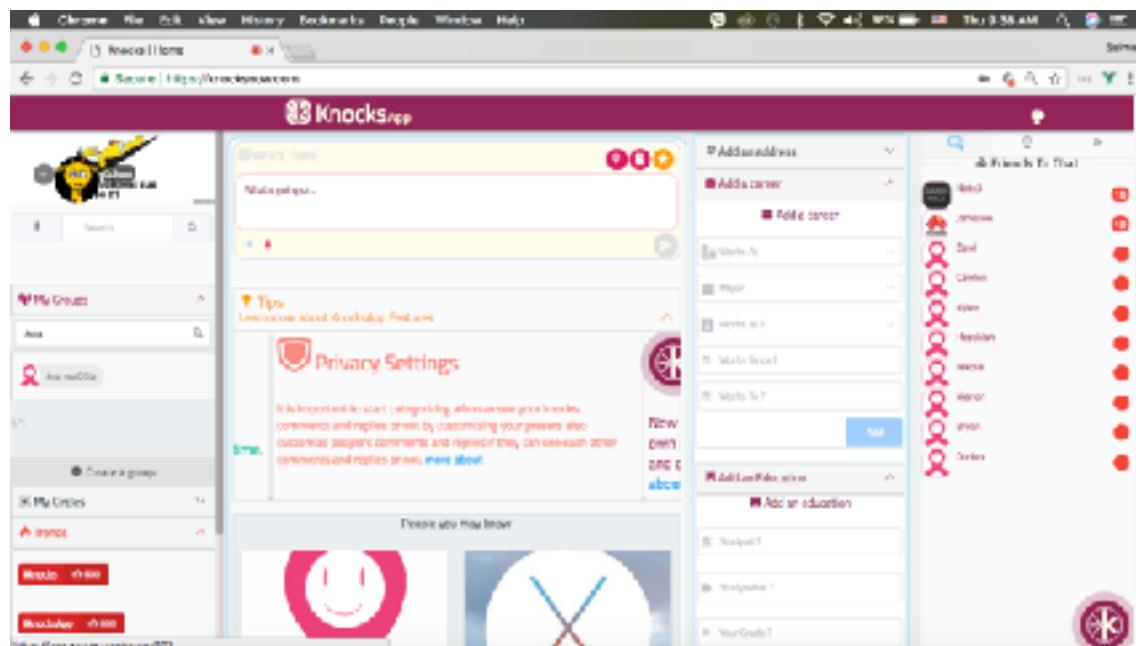
Then he would pick on settings icon and choose “Group Preset”, and then choose his desired privacy set





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

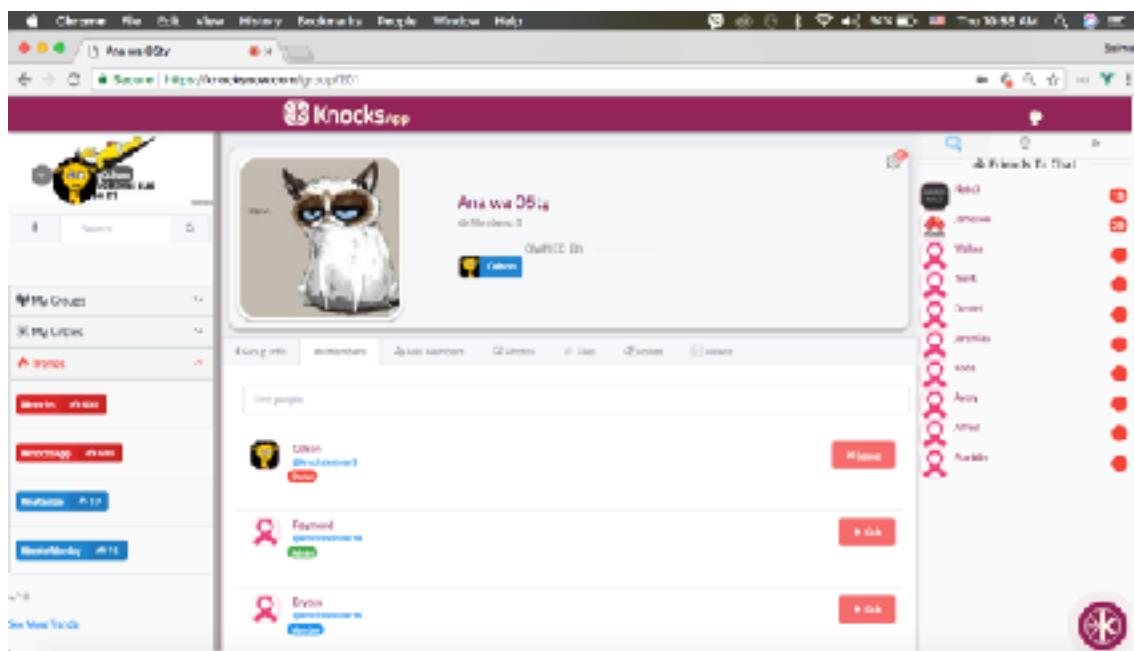
At the left side of the home page in groups section, click on “My Groups” tab and search for a group that he owns or admins.



Then he should click on the “Members” tab, then choose the user to be kicked out of the group ( if

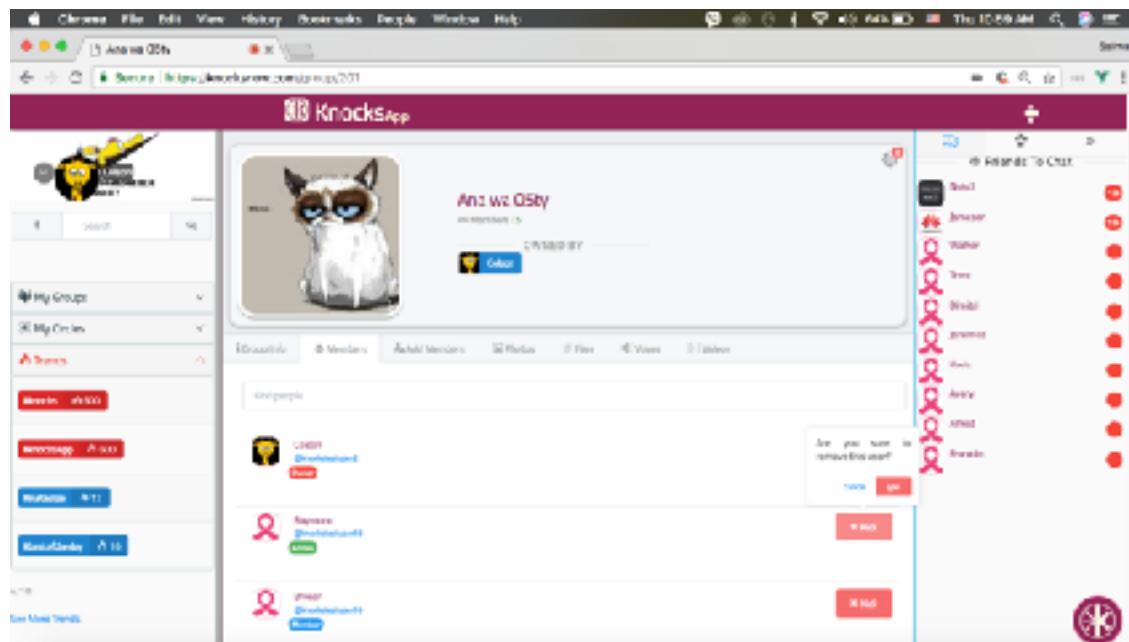
## CHAPTER 8 : USER MANUAL

you are leaving the group that you own, you should transfer the group ownership yo one of the members first), then click “Kick”



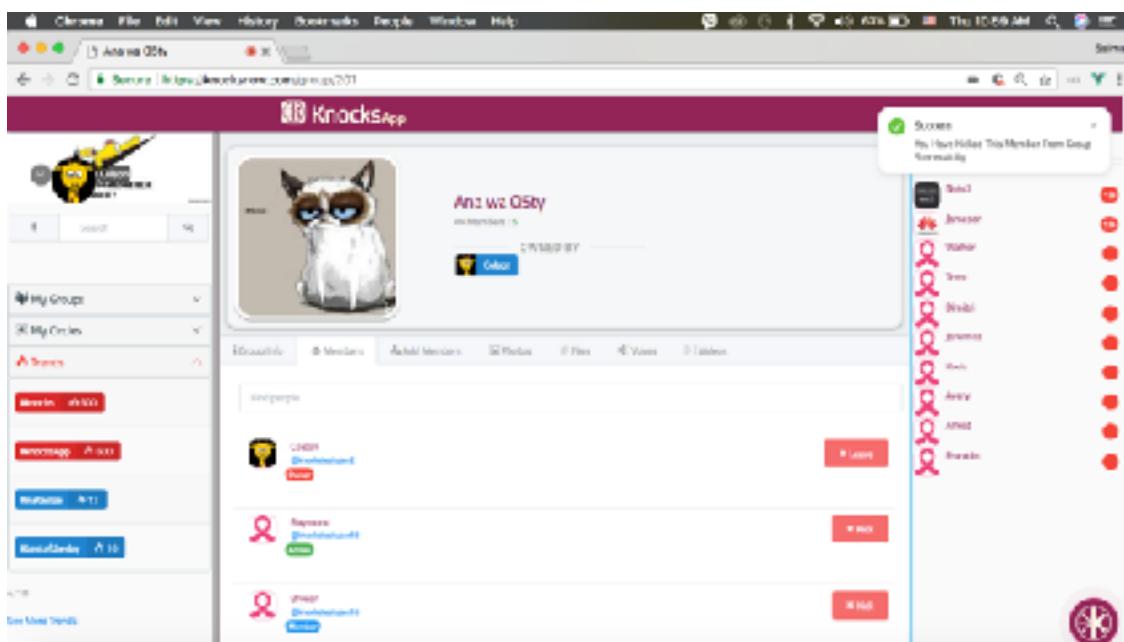
A

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



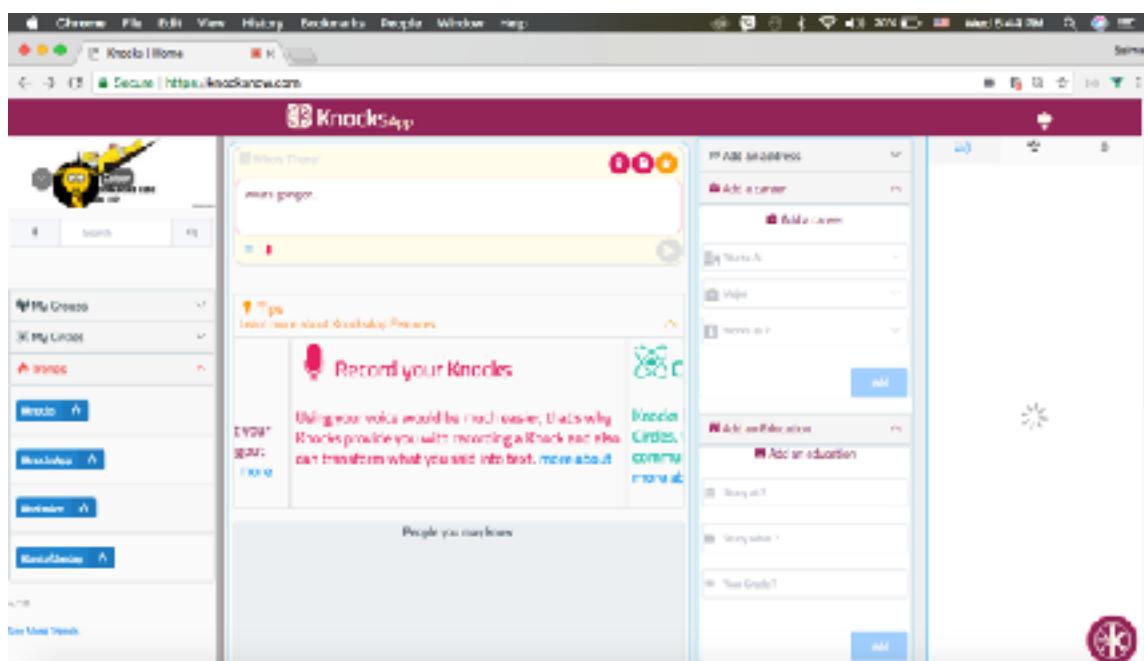
Then the system will display a message “You have kicked this member from group successfully”

## CHAPTER 8 : USER MANUAL

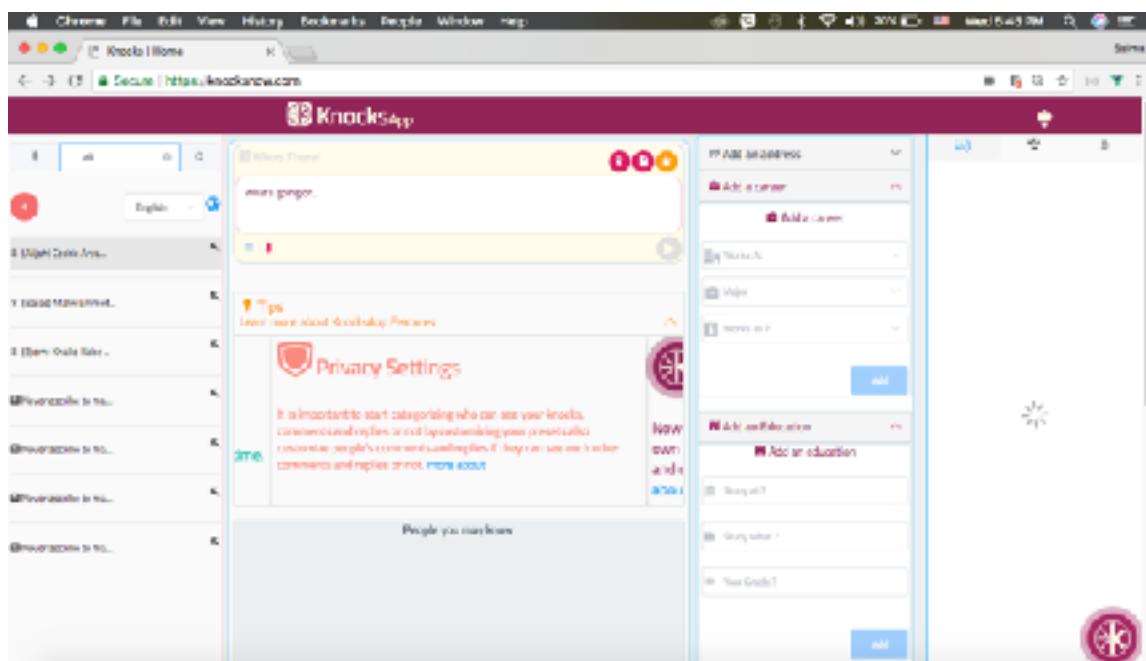


- **View a user profile - when allowed to do so.**

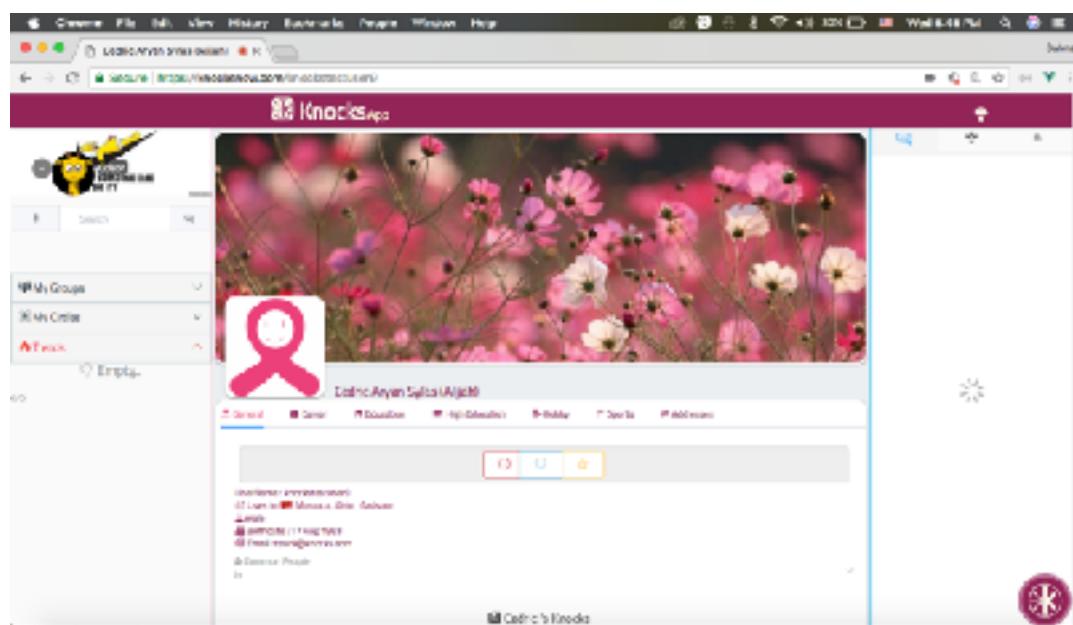
The user can search for friends in the left side bar and clicks on the user chip to be directed to his profile or just click on his username in a knock, comment, reply or any activity where his username appears.



## CHAPTER 8 : USER MANUAL



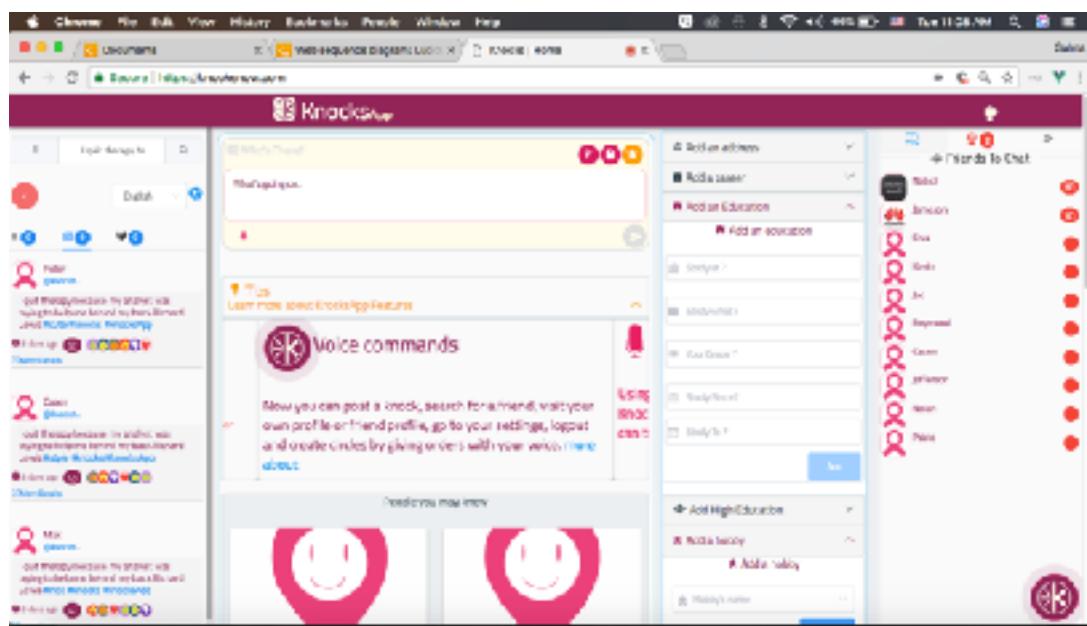
Then the system will display the user's profile according to the other user's privacy settings.



## CHAPTER 8 : USER MANUAL

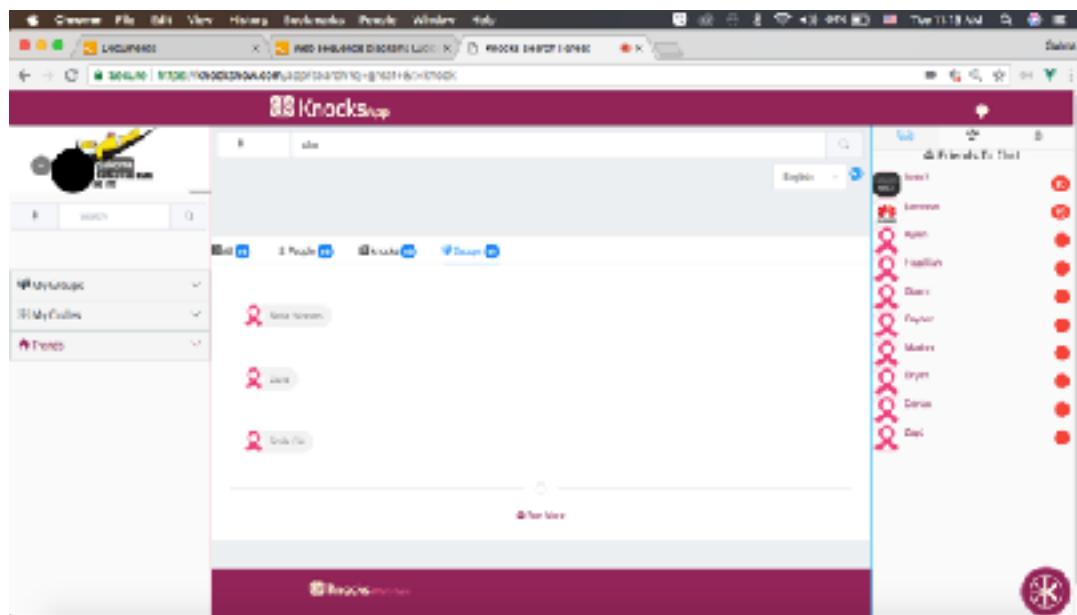
- **Search for knock.**

At the left section of the home page , the user would write a knock or any keyword that refers to a knock and chooses the language where he want to search with and clicks on button “search”



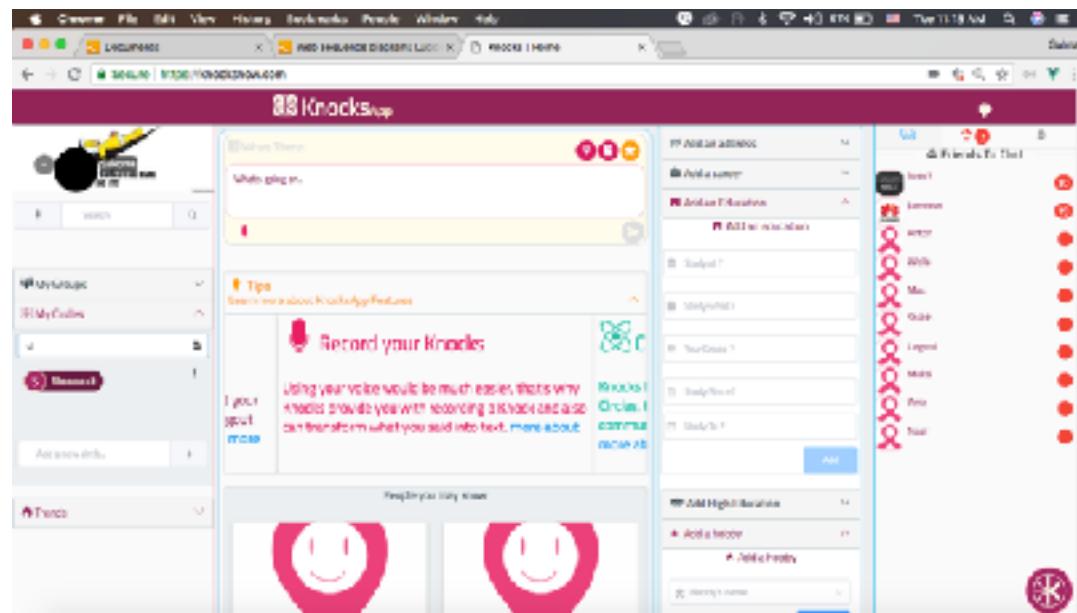
- **Search for group**

At the left section of the home page , the user would write a group name and chooses the language where he want to search with and clicks on button “search”



- **Search for circle**

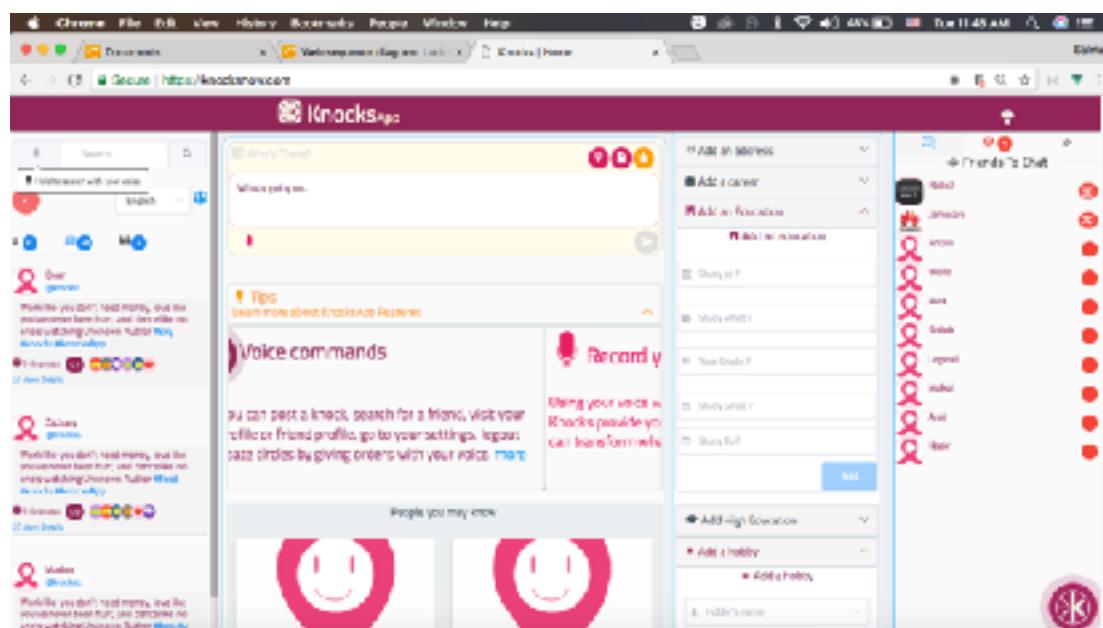
At the left section of my circles, the user would write a circle name and clicks on button “search”



- **Search voice for knock**

b y

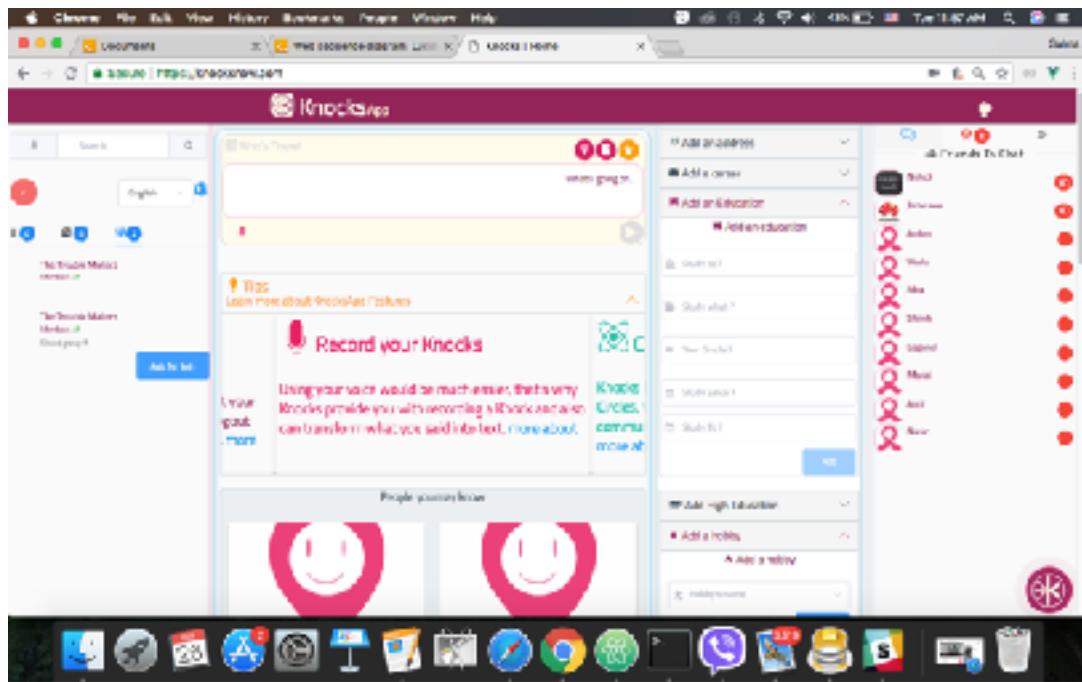
At the left section of the home page , the user would clock on mice icon and record a knock or any keyword that refers to a knock and chooses the language where he want to search with



## CHAPTER 8 : USER MANUAL

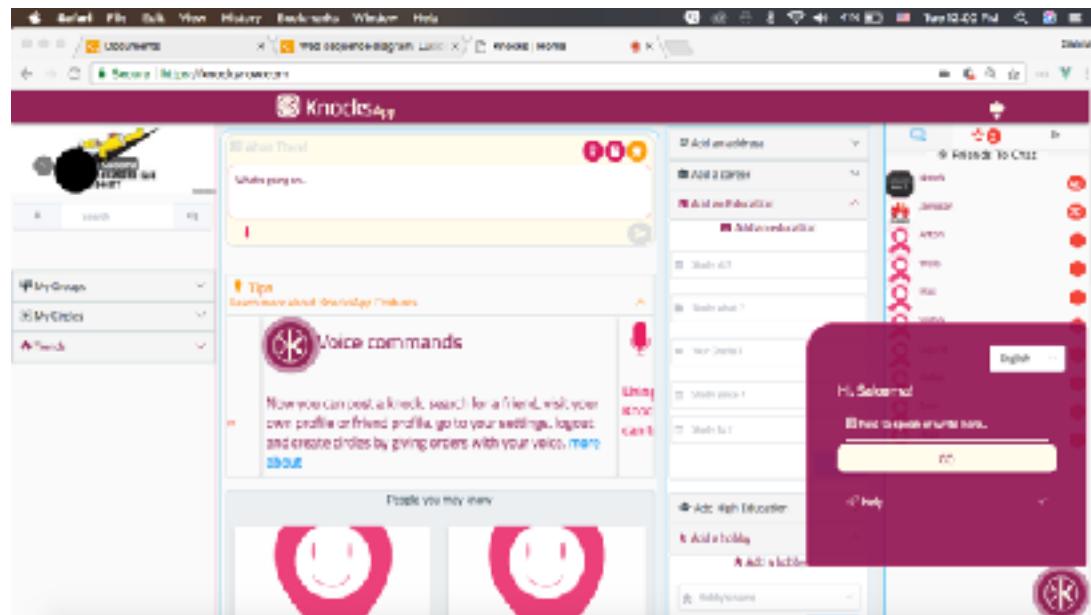
- **Search by voice for group**

At the left section of the home page , the user would clock on mice icon and record a group name and chooses the language where he want to search with



- **Enabling and disabling voice command**

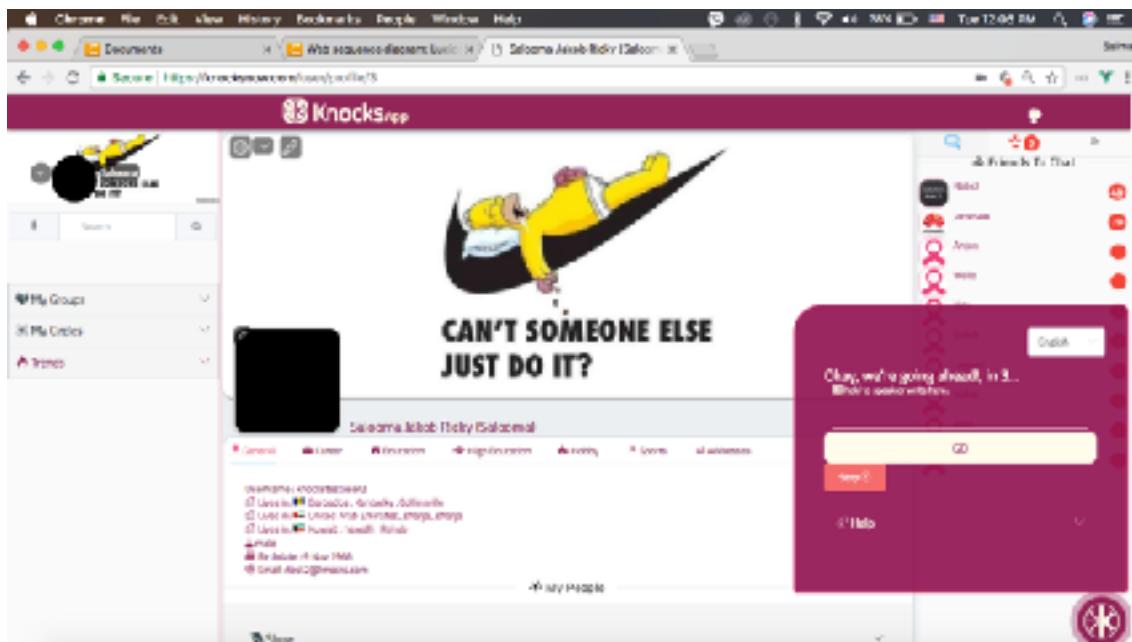
The user starts to record “Knock knock” to enable the voice commands and dialog from the bottom right will collapse to start your commands, for disabling the user should say “Close”.



## CHAPTER 8 : USER MANUAL

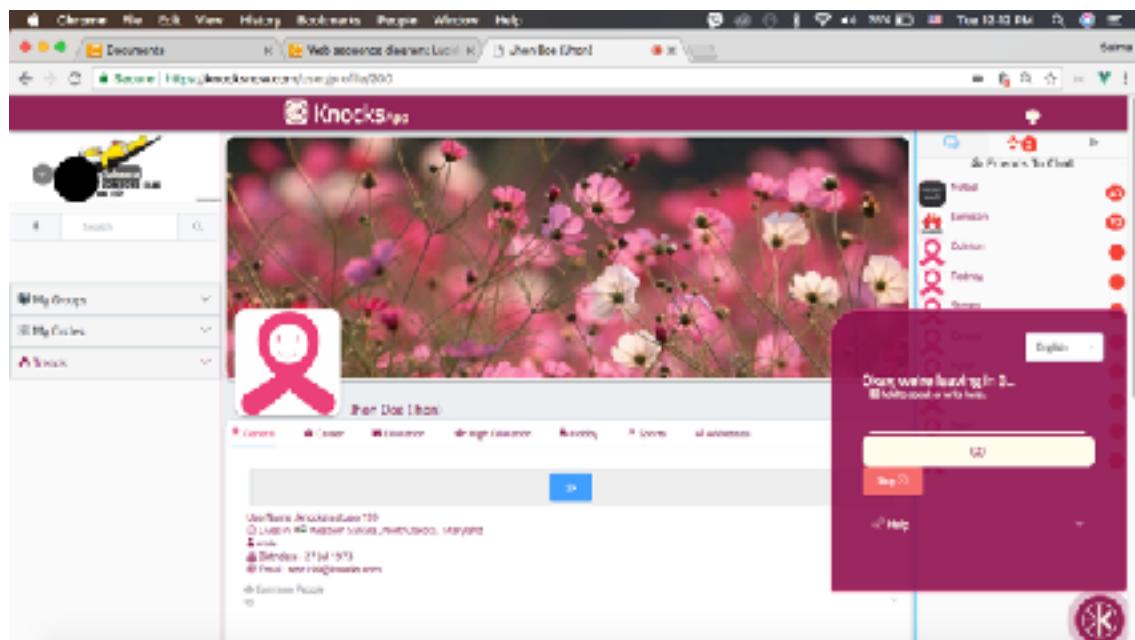
- **visiting profiles by voice commands**

The user starts to record “Knock knock” to enable the voice commands and dialog from the bottom right will collapse then the user should say “Visit username profile”,



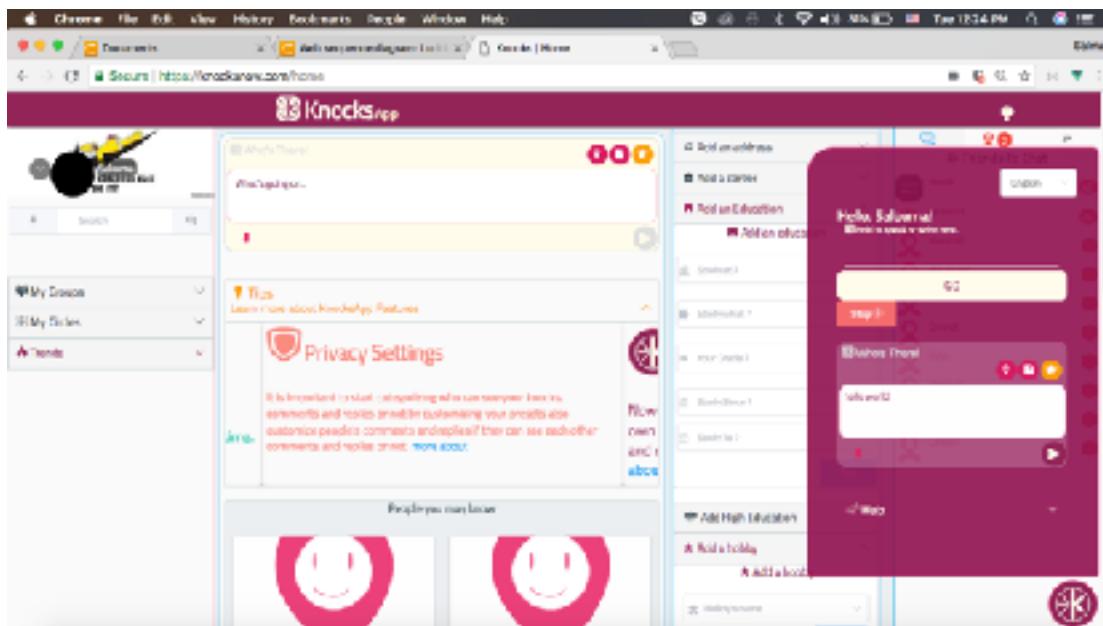
- **Log out using voice commands**

The user starts to record “Knock knock” to enable the voice commands and dialog from the bottom right will collapse then the user should say “Log out”.



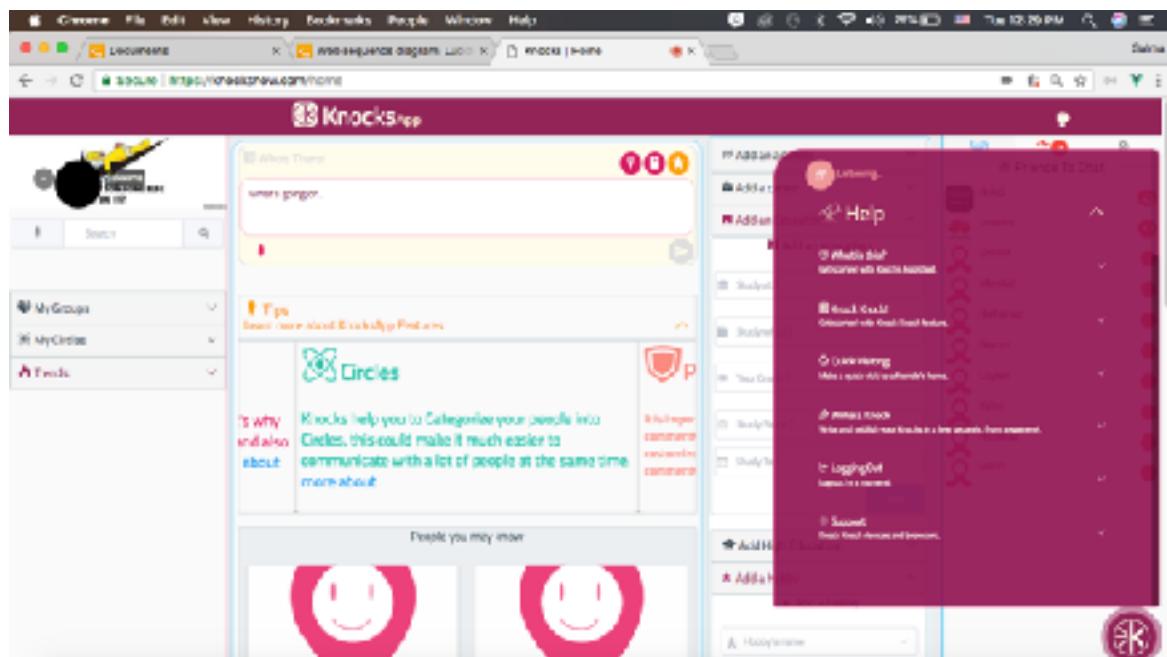
- **Publishing a knock by voice commands**

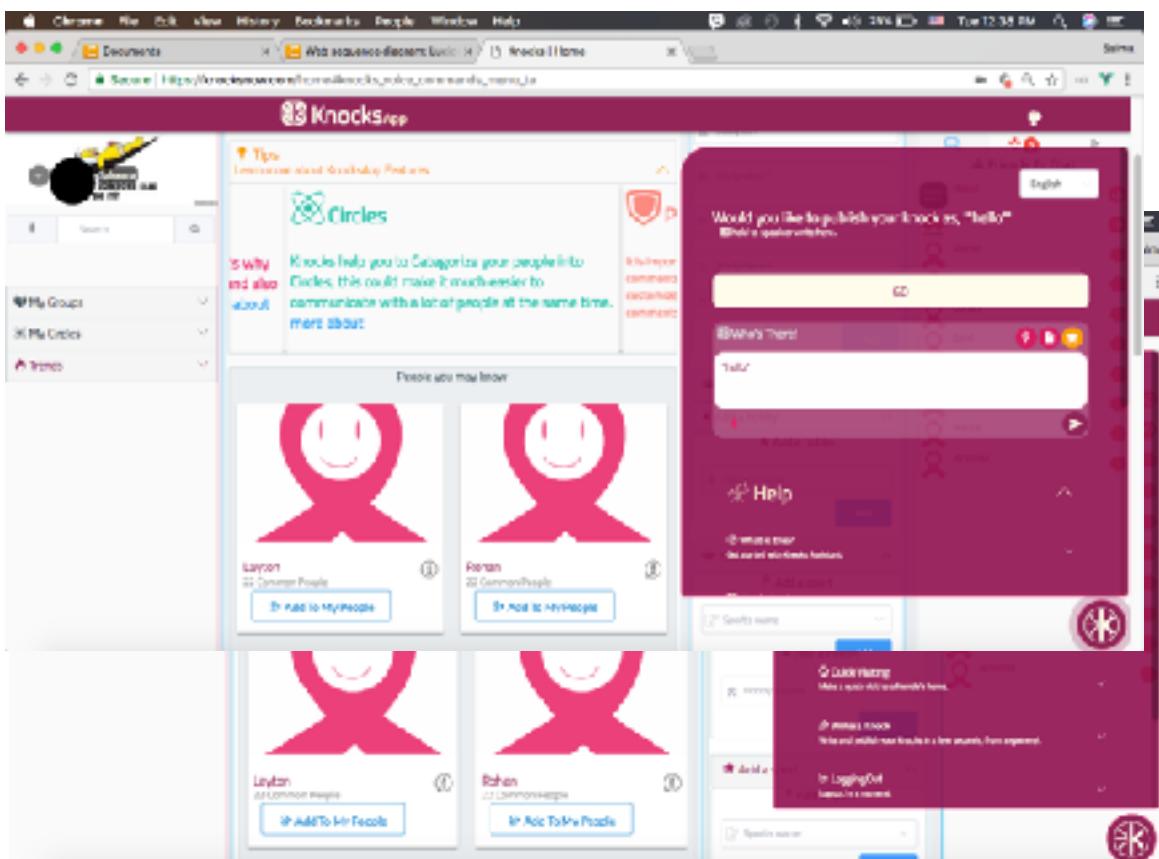
The user starts to record “Knock knock” to enable the voice commands and dialog from the bottom right will collapse then the user should say “i want to say ‘the knock content’ ” then the system asks for the user confirmation to publish in three seconds.



- **Guiding users by voice commands**

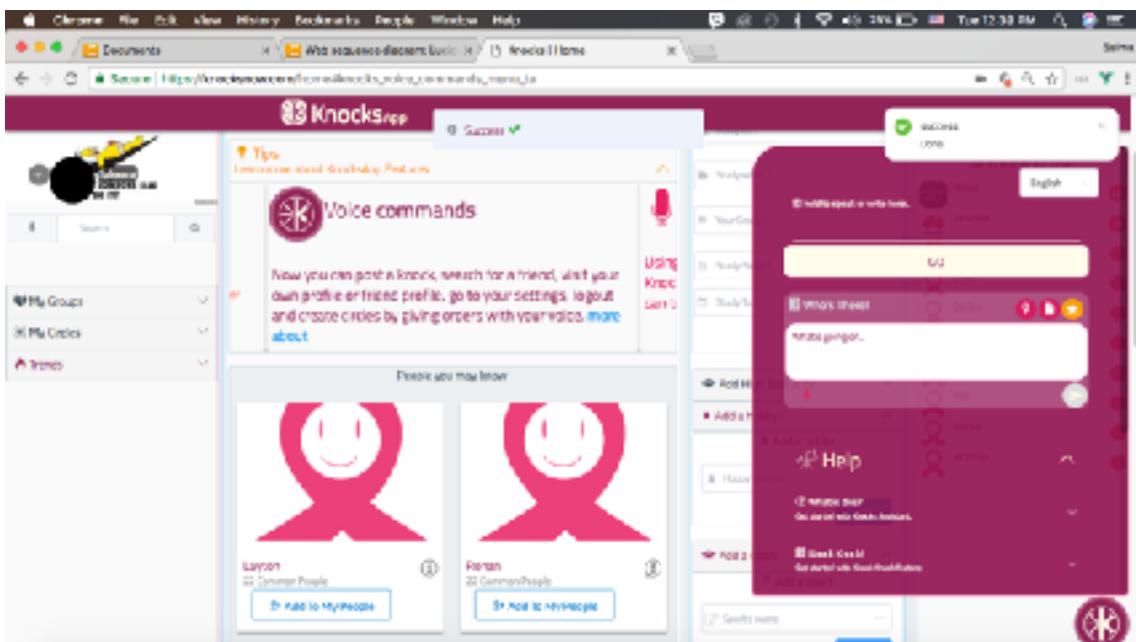
The user starts to record “Knock knock” to enable the voice commands and dialog from the bottom right will collapse then the user will click on help to view the voice commands guides.





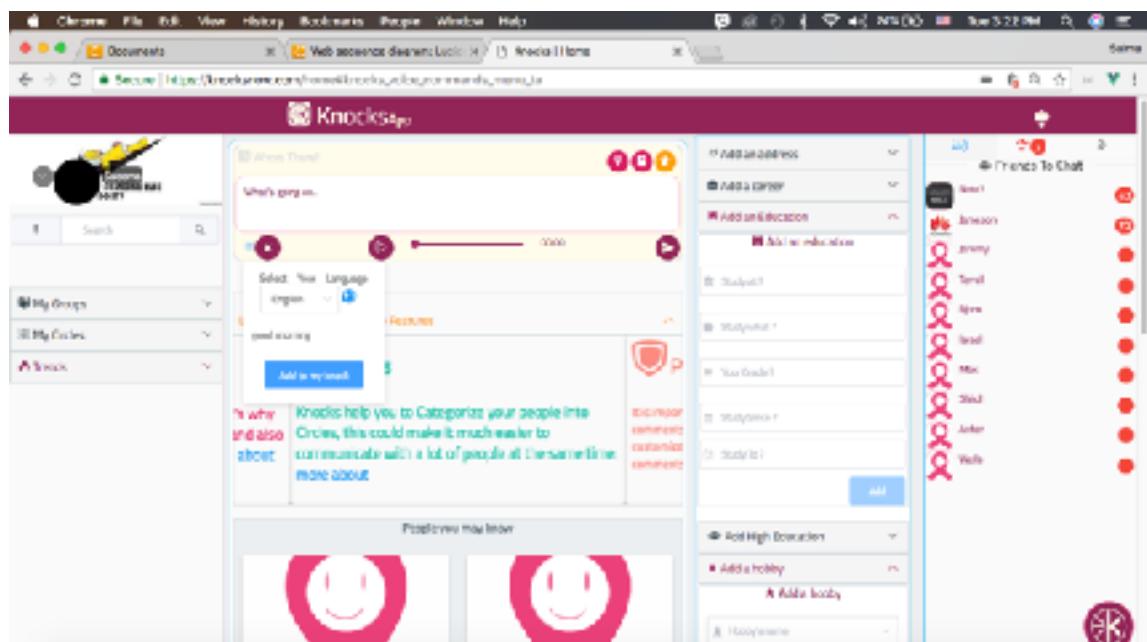
- Assisting the voice commands with text

The user starts to record “Knock knock” to enable the voice commands and dialog from the bottom right will collapse then the user will type his command instead of recording



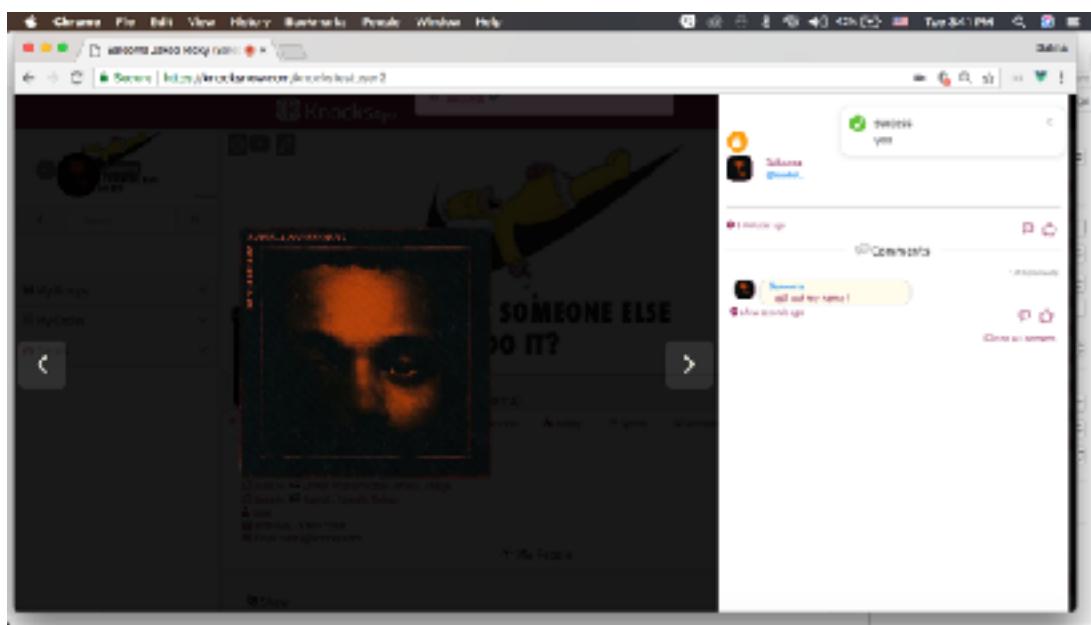
- **Converting the spoken voice into text**

At the knock section the user can record what he wants to say after choosing the language for his words to be displayed with and then he will click add to my knock.



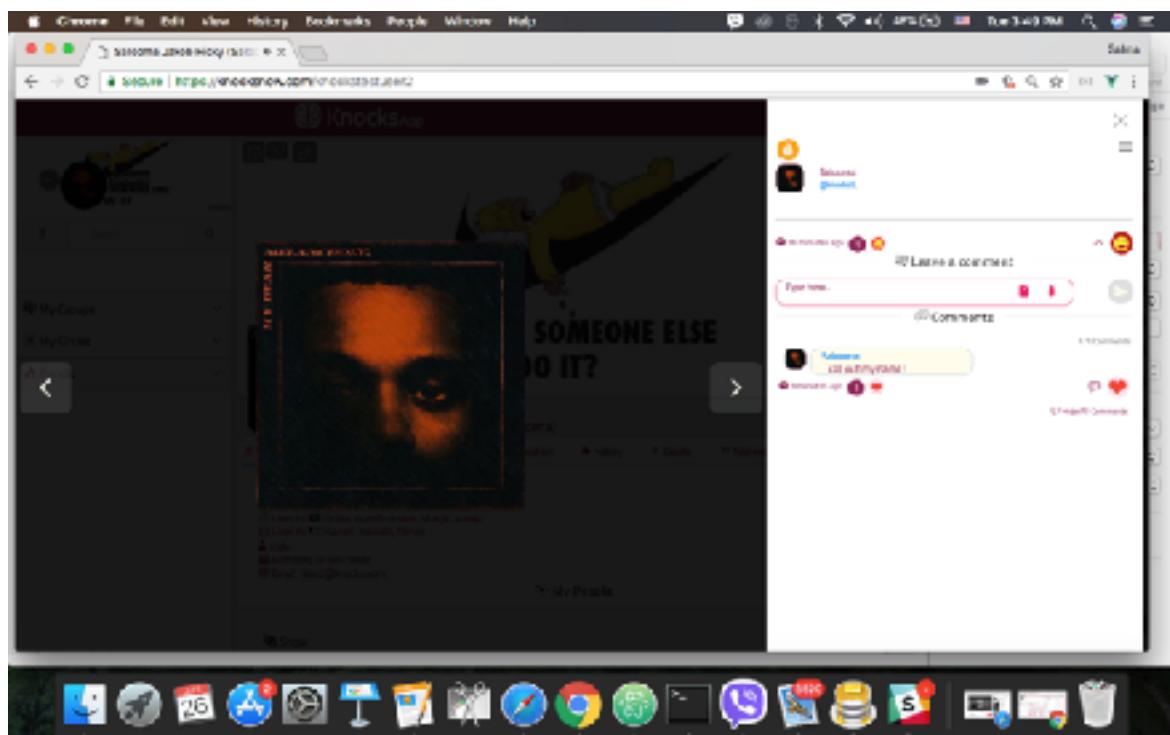
- **Comment on photos**

For any photo, the user can comment on that selected photo with a non empty content, he can comment with text, voice, or file



- **React on photos**

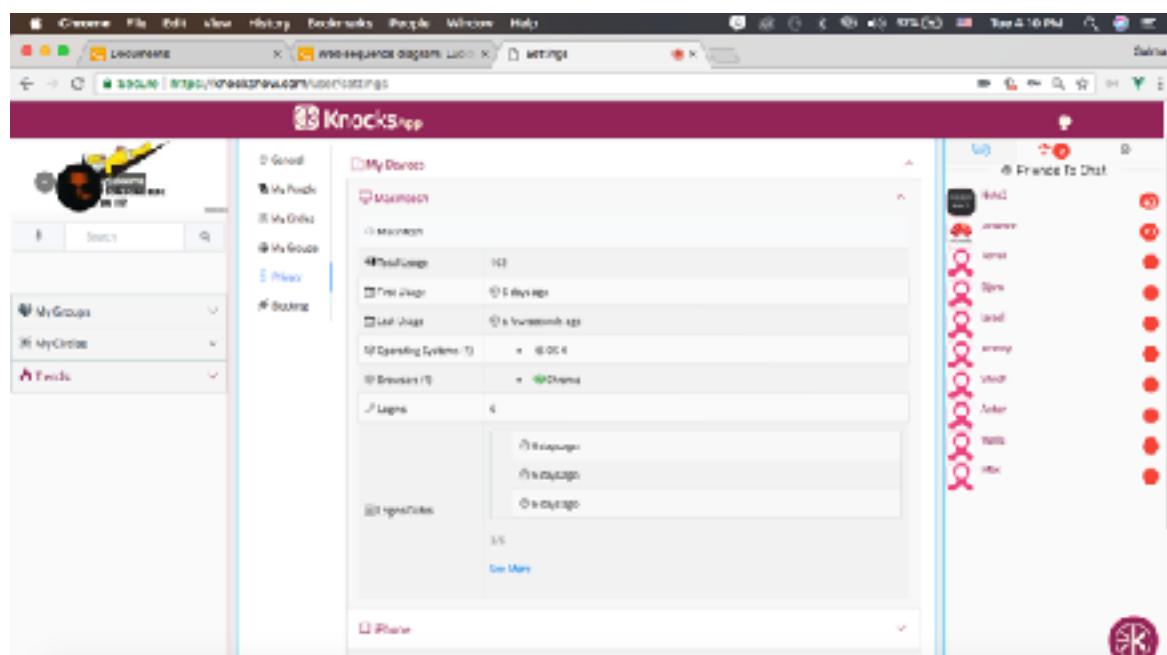
For any photo, the user can react to that selected photo by choosing from the available reactions



## CHAPTER 8 : USER MANUAL

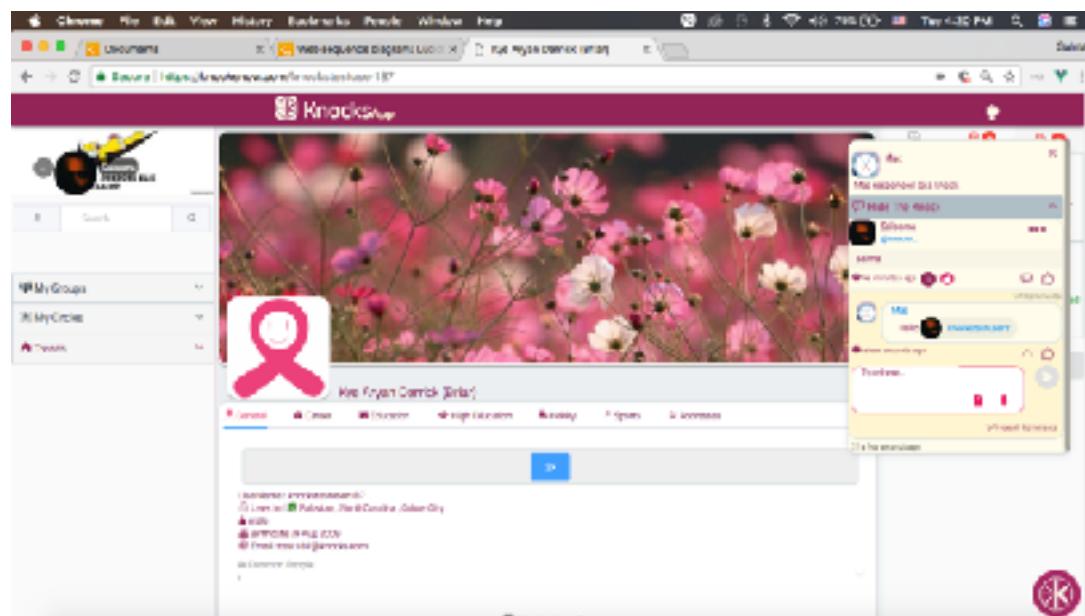
- **Generating report for user's activities**

The user can view his own activities by going to the settings and choose my privacy.



- **Notifying the users**

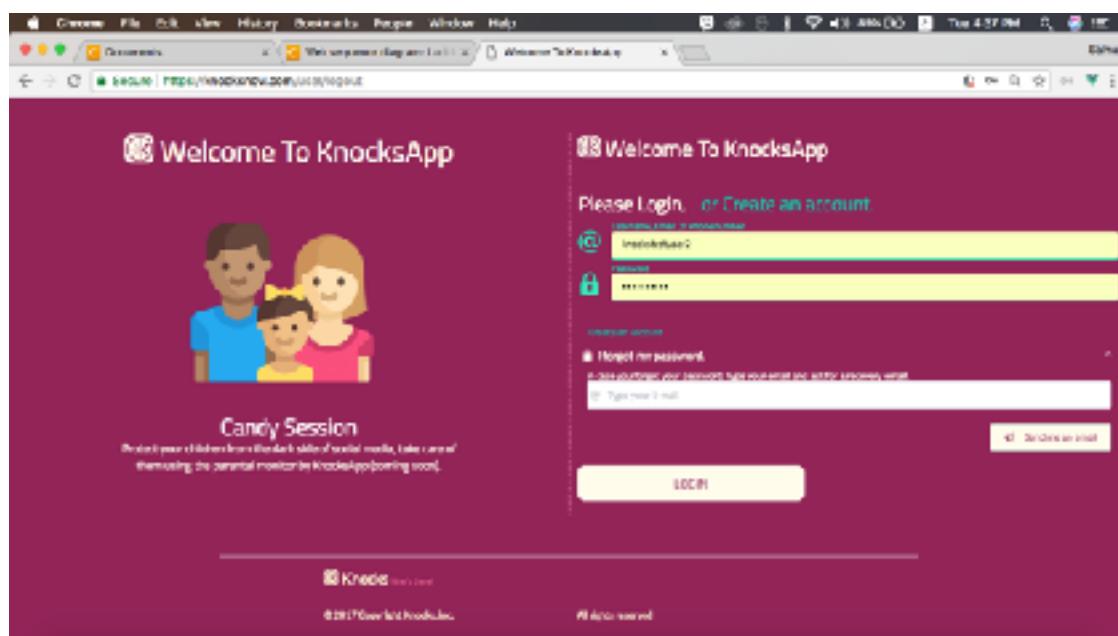
Whenever the user has any balloons, he will be notified with them by a small popover on the right showing the balloon details.



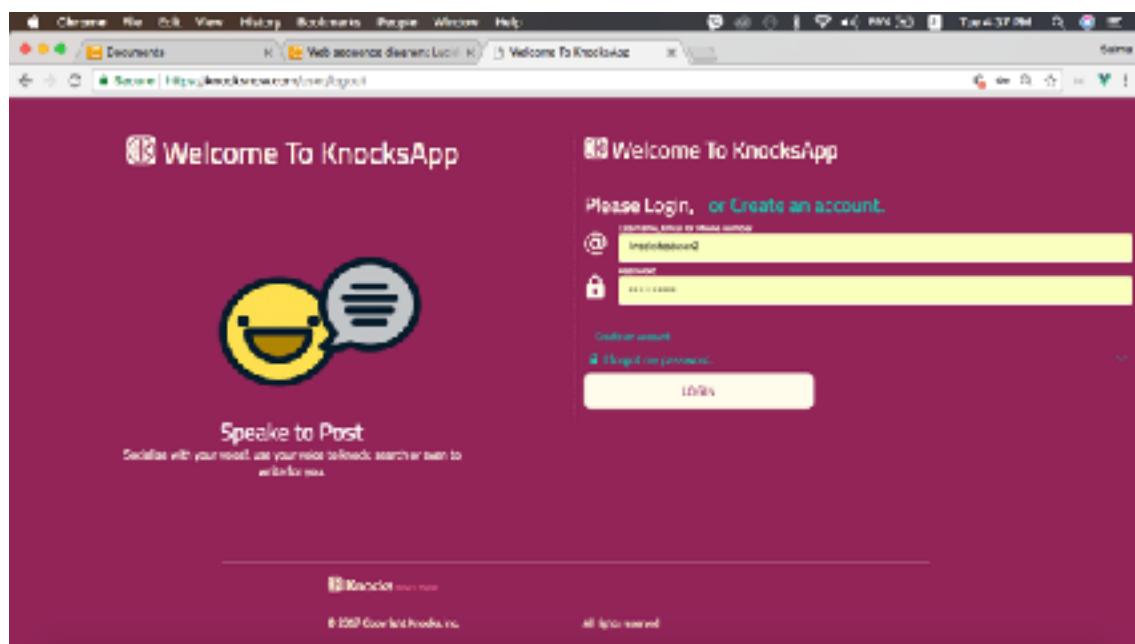
## CHAPTER 8 : USER MANUAL

- **Enabling Forgot my password mode**

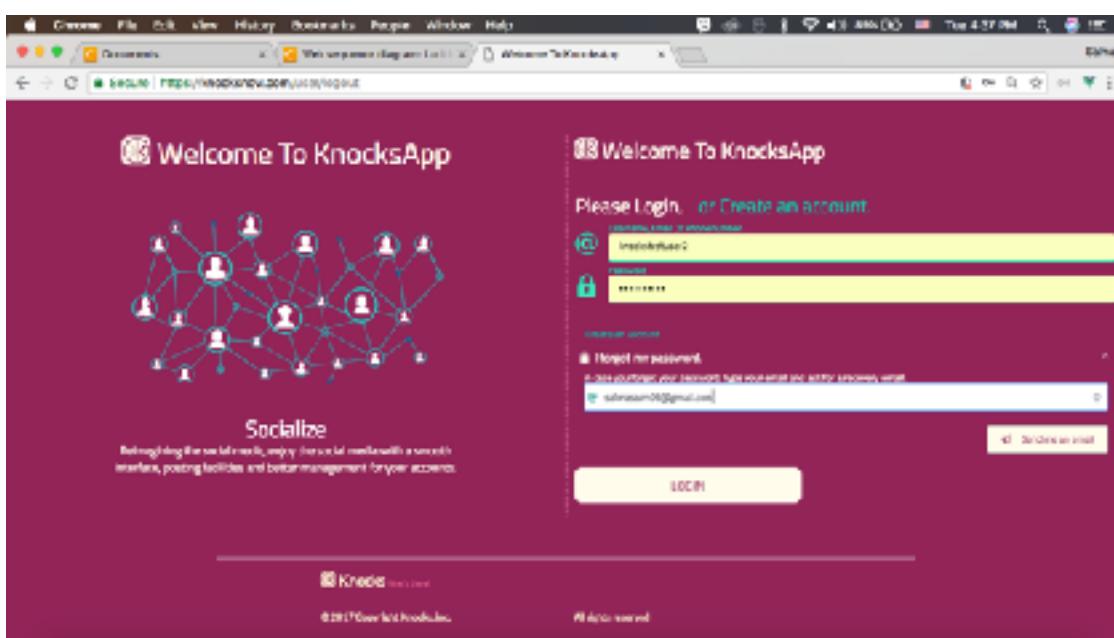
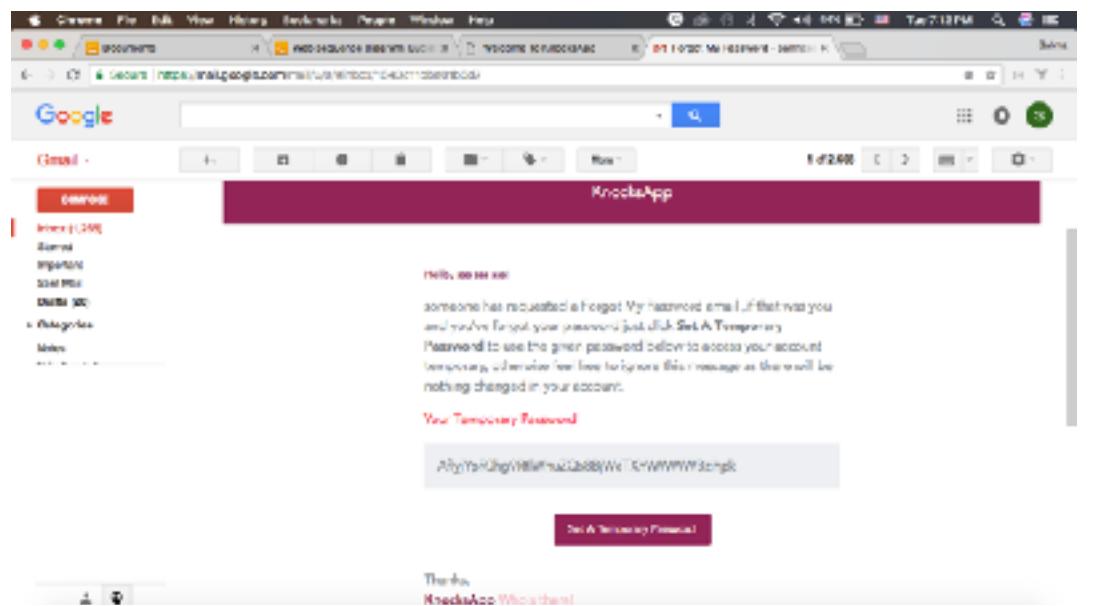
If the user forgot his password when logging in, then he can enable the forgot my password mode and rest his own password again, by clicking on “forgot my password” and then

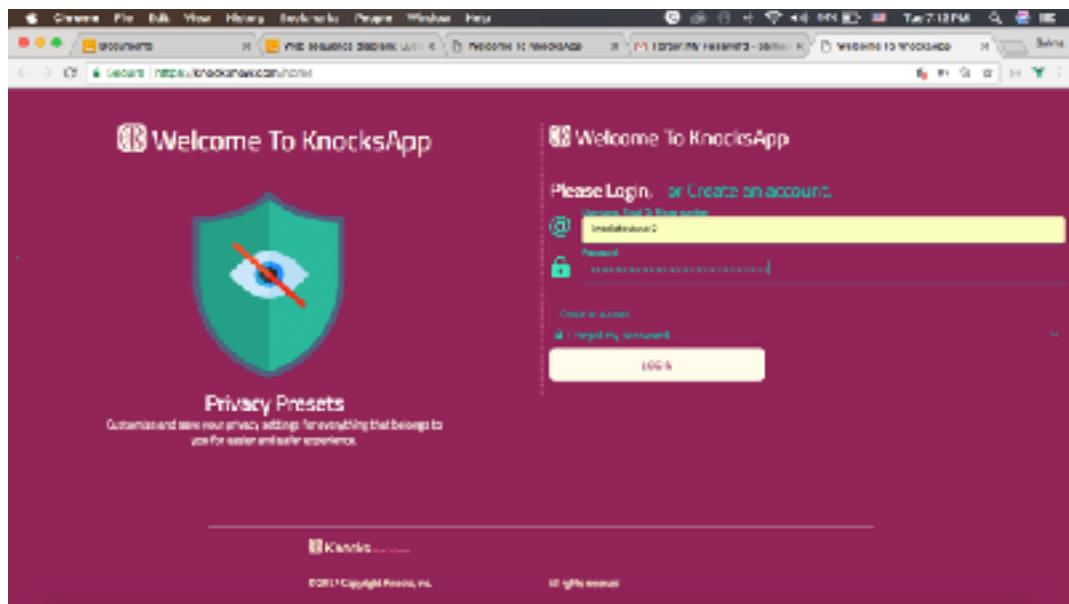


providing his e-mail where he can receive the temporary password, and then he will click on reset my password which will appear in the sent e-mail.



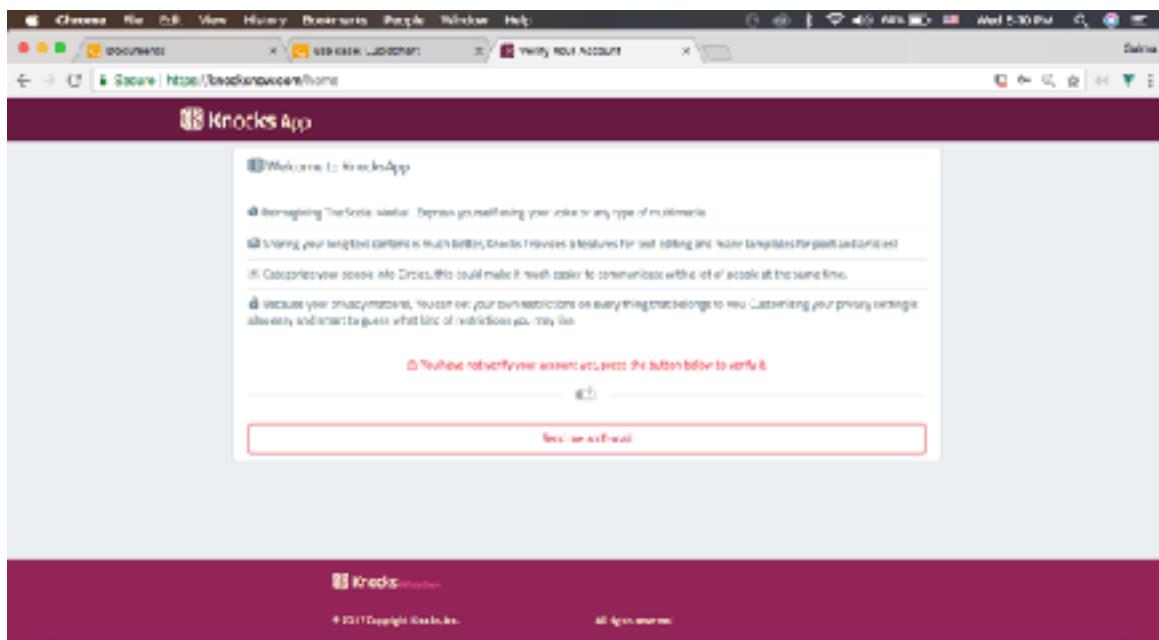
## CHAPTER 8 : USER MANUAL

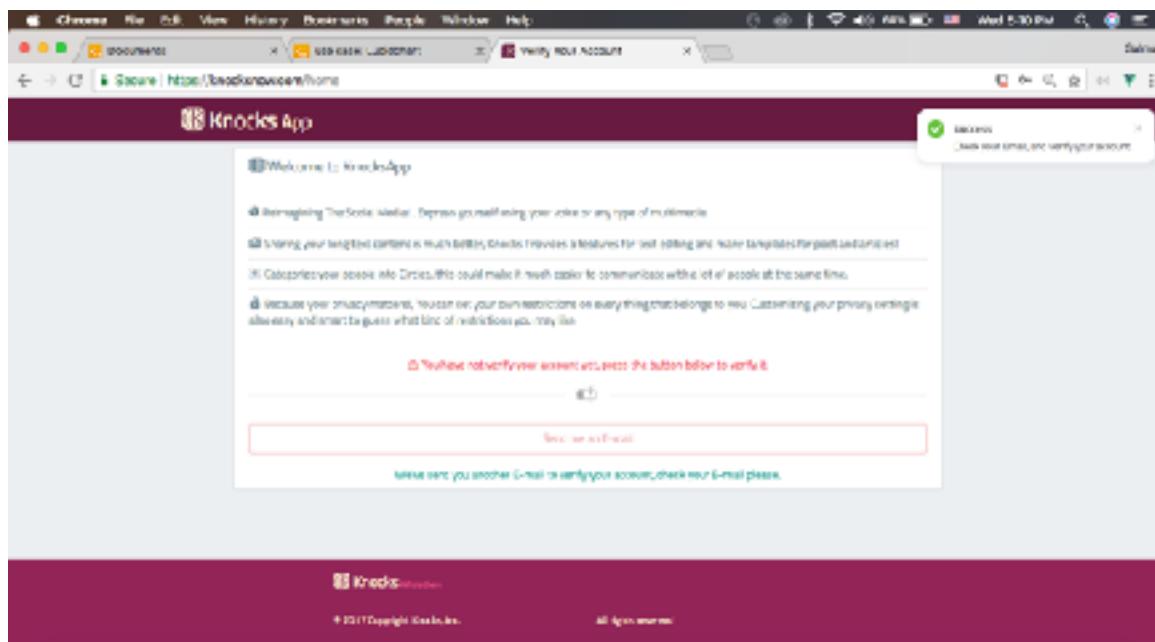




- **Request for e-mail verification**

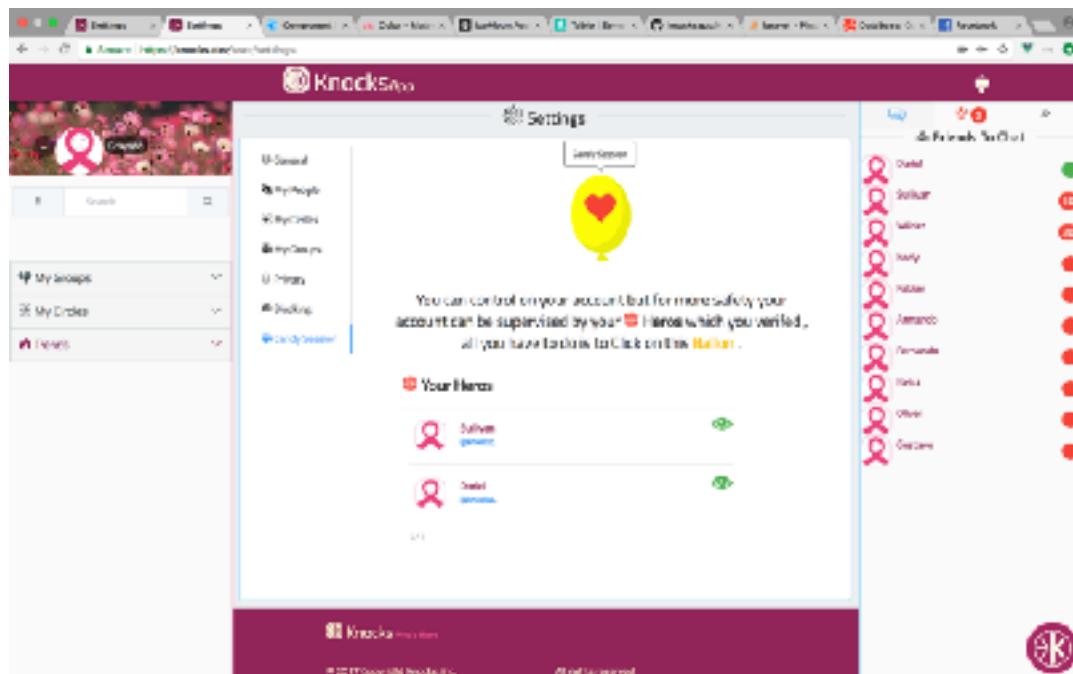
Whenever the user finishes the registration, the system asks for e-mail verification, so it will send the verification link at the provided e-mail when registering, then the user receives that e-mail and click on

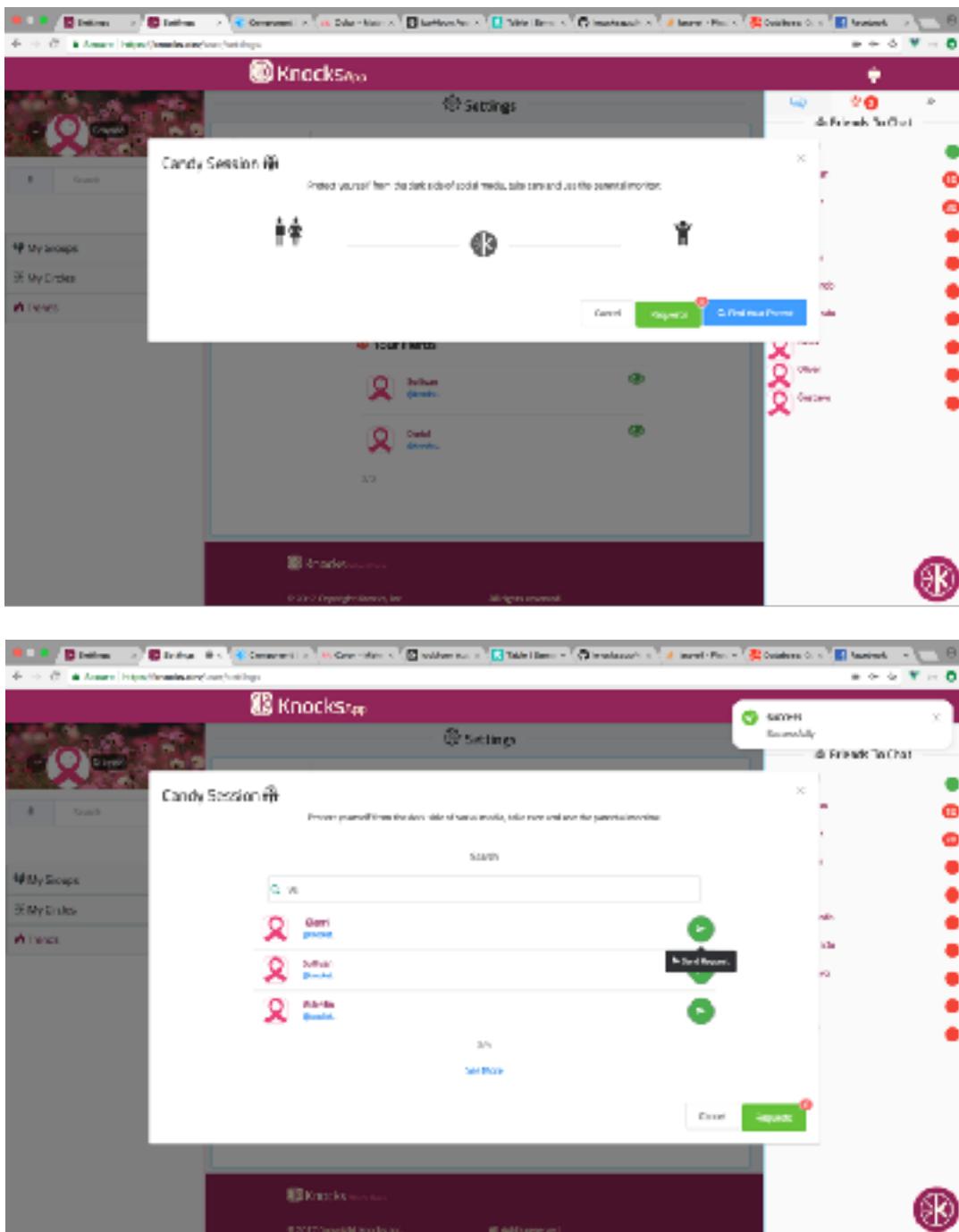




#### • Set a parent to a child

If the user is a kid ( twelve years or less ), then he is considered as a candy session member and he can add his parents to supervise his account, so he enters the settings and choose candy session tab then he clicks in “Find my parent” then enters the user name to set him as apparent, and this user has to be one of the child’s friends, then clicks on “Add”



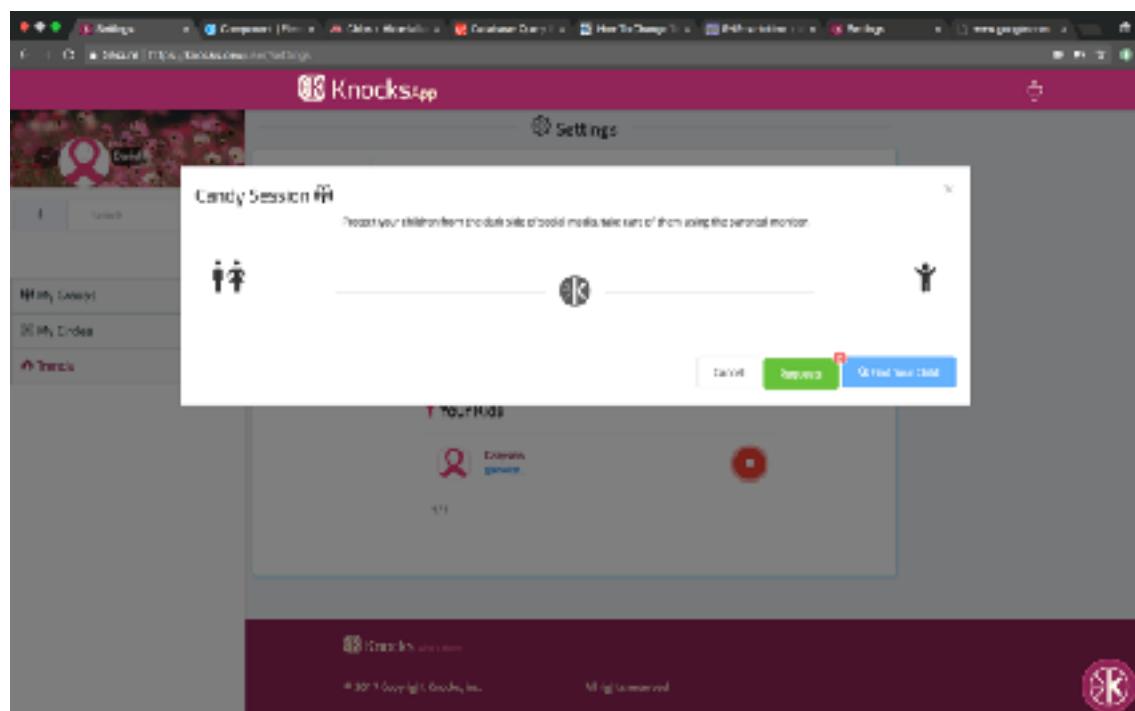
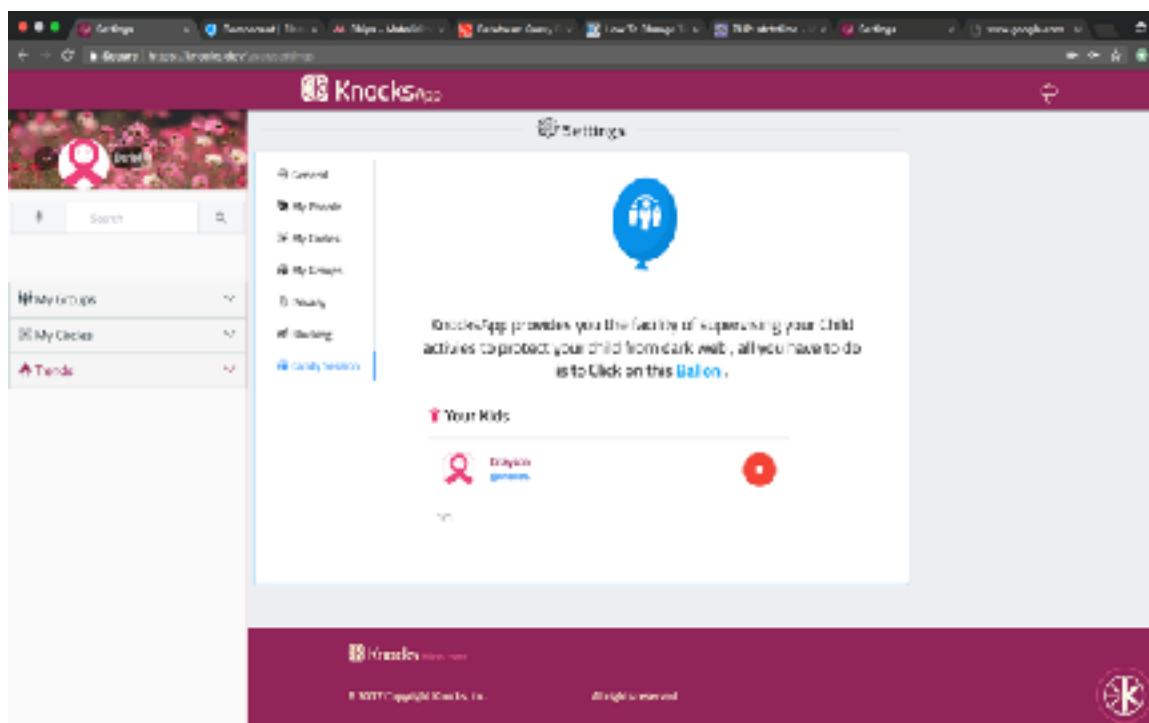


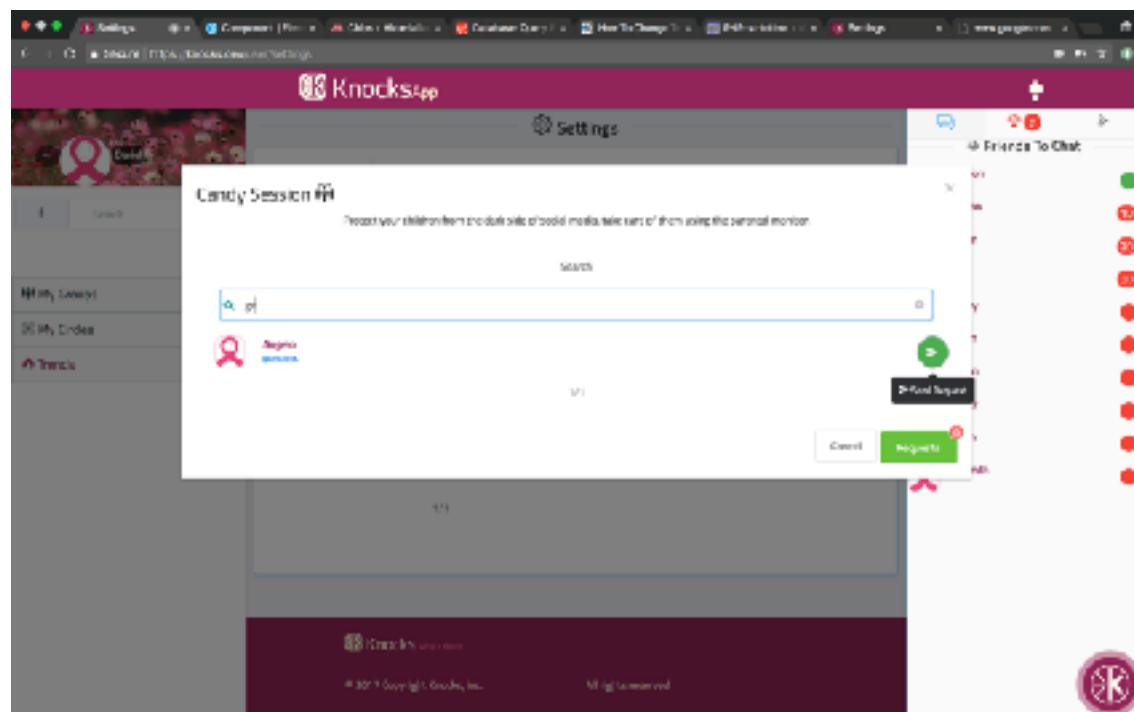
- **Set a parent to a child**

If you have a child ( twelve years or less ), the he is considered as a candy session member and you can set him a child, just enter the settings and choose candy session tab then click "Find

## CHAPTER 8 : USER MANUAL

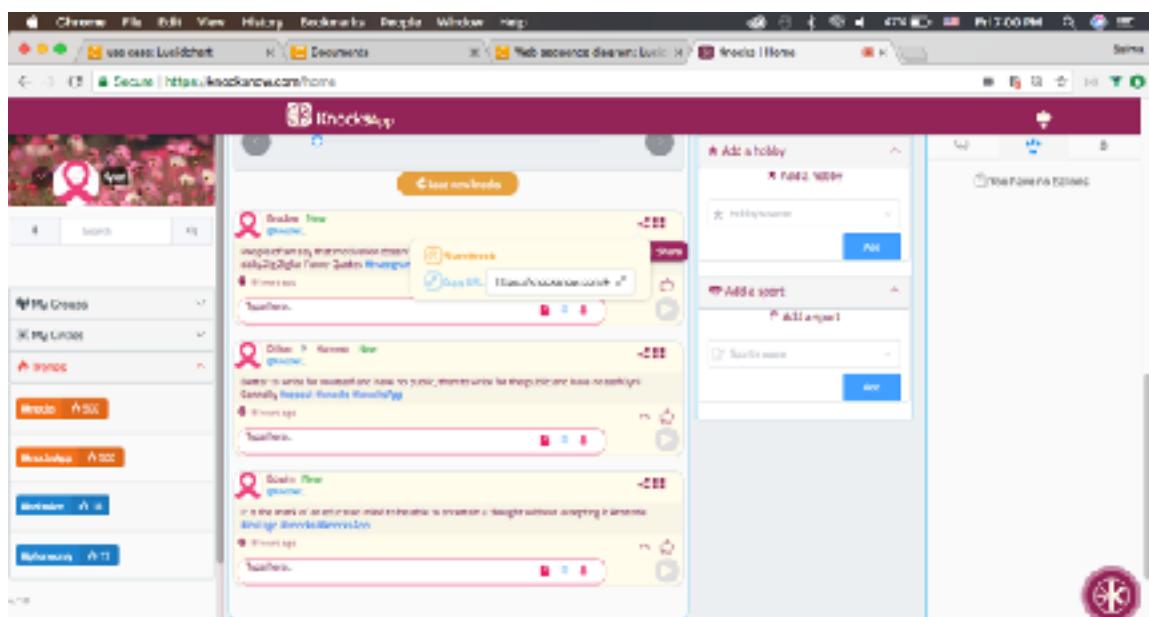
my child" then enter the user name to set him as child, and this user has to be one of the parent's friends, then clicks on "Add"



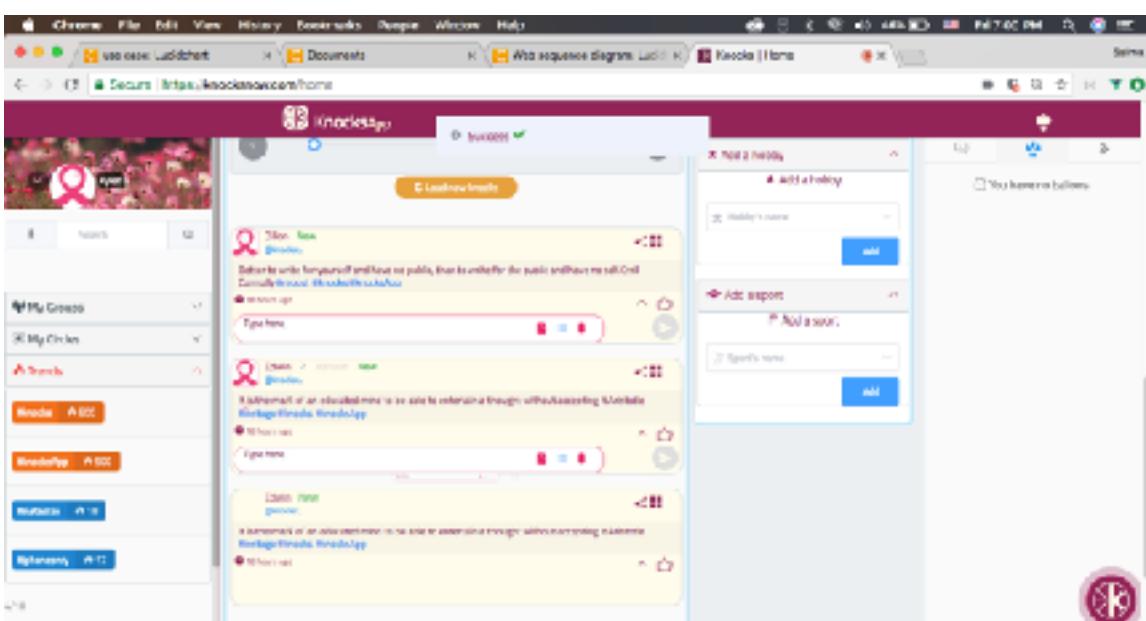
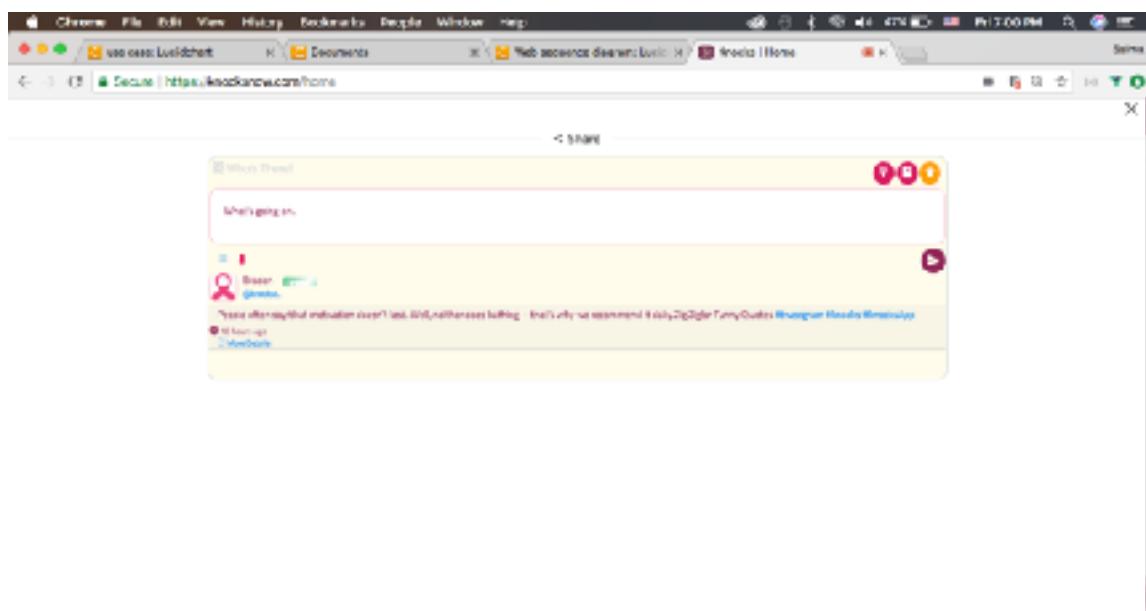


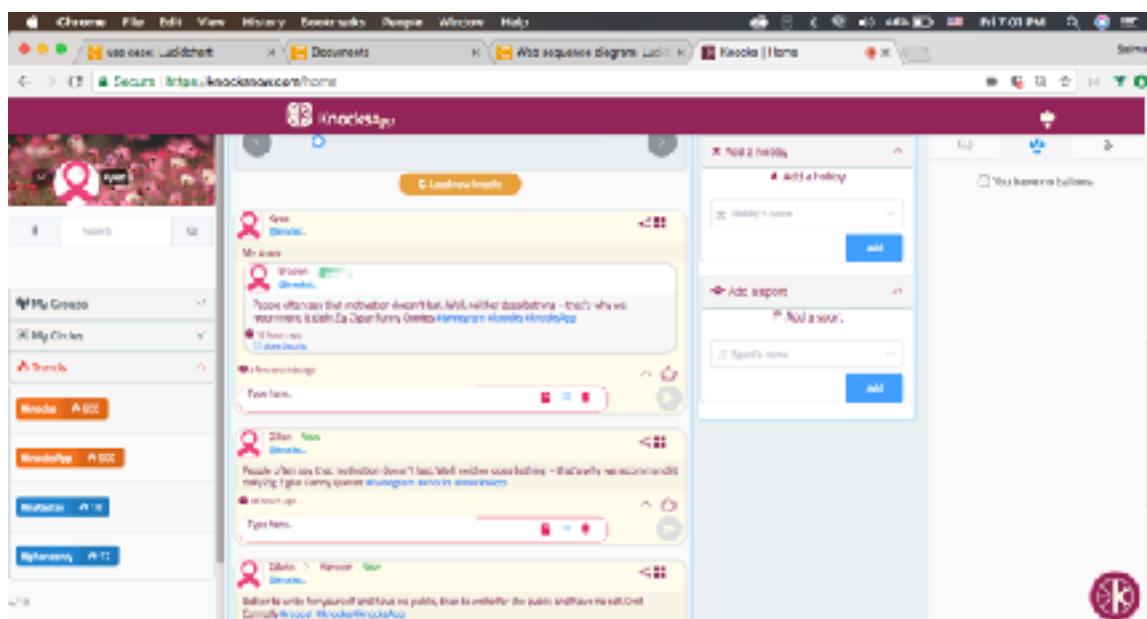
- Share Posts**

The user can share specific post by clicking in the share icon and choose if he wants to copy the link or just share the knock, and he can also write for this knock share



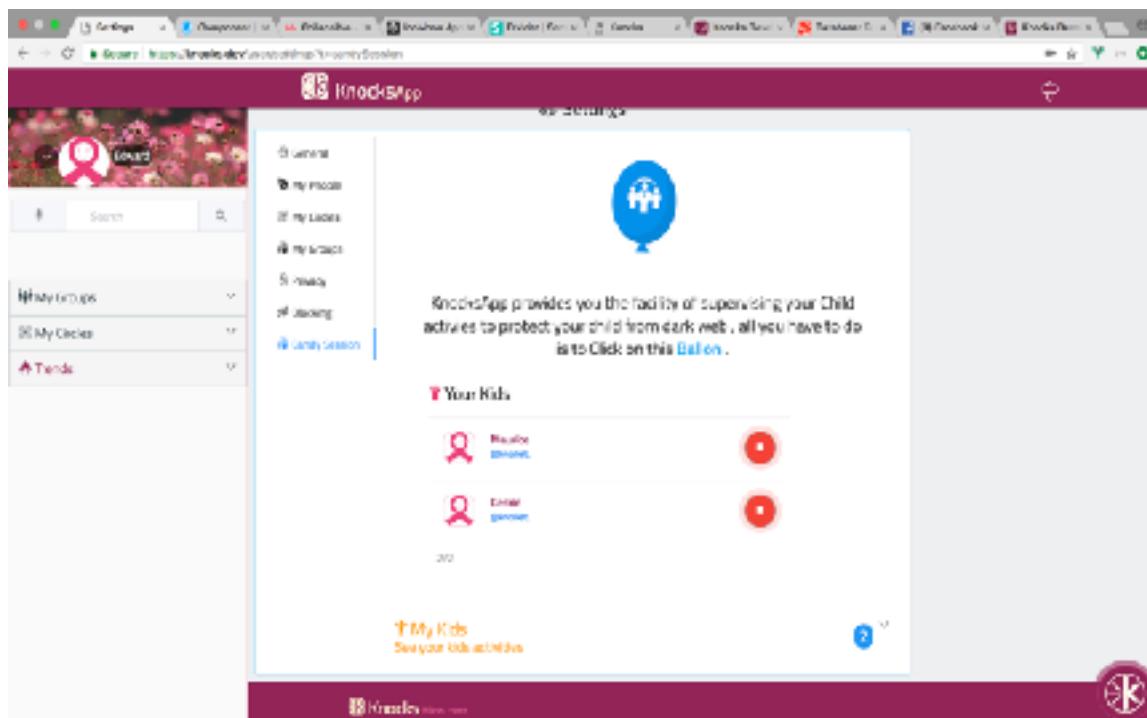
## CHAPTER 8 : USER MANUAL

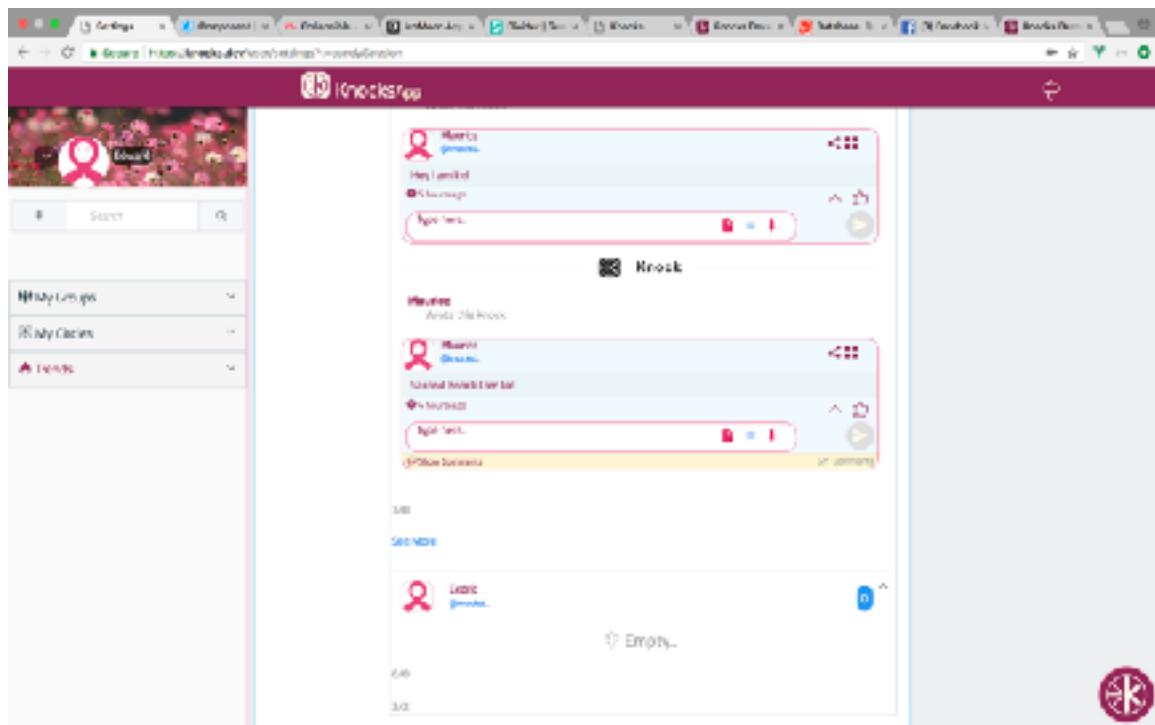
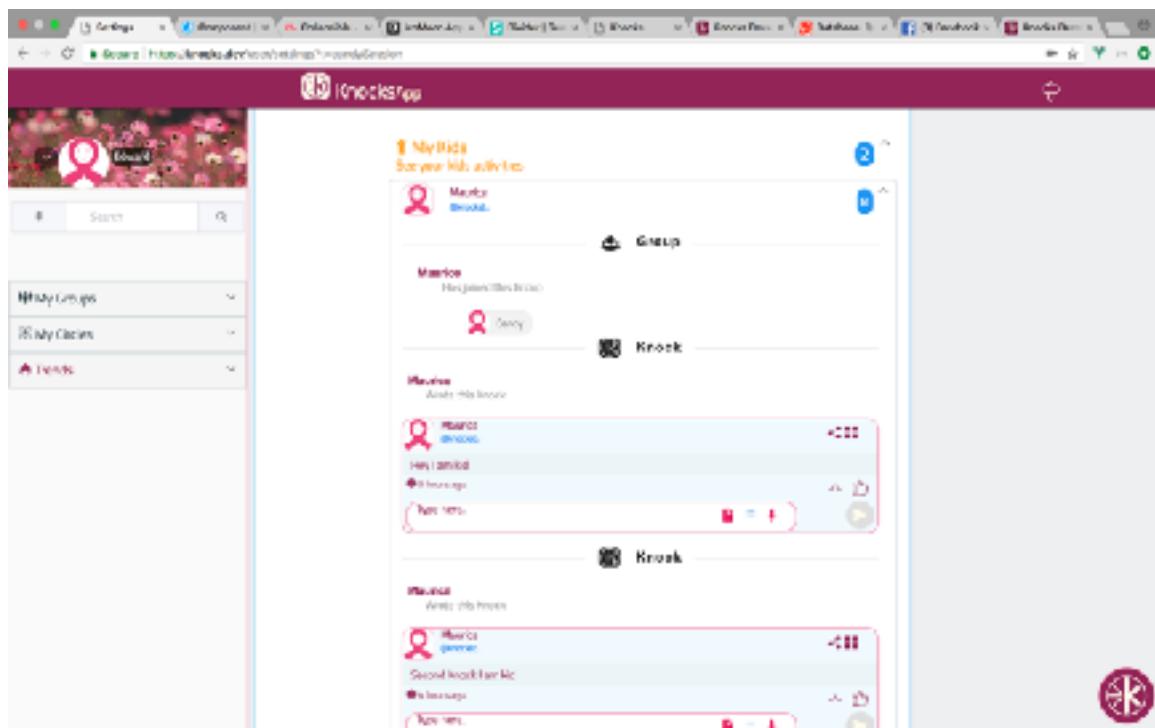


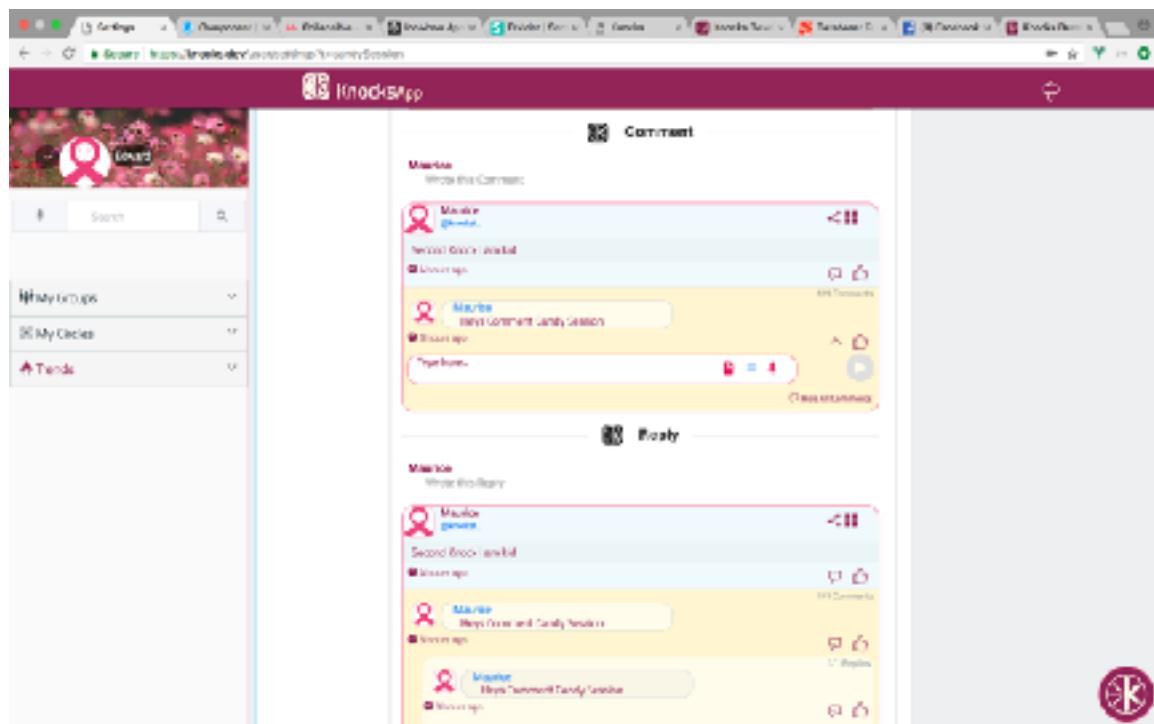


#### • Monitor child log

Verified parents can monitor their child's log and watch his child's groups, knocks, comments, replies, friend requests and friends by allowing the monitor mode.







# Chapter 9 : Conclusion and Future Work

## **9.1.Achievements**

What we could achieve from this project?

- Users can do what they want with voice commands like posting, commenting, searching, and viewing other user's profiles.
- Users can post, comment, reply, and search using voice recording
- Users can manage their privacy settings easily and precisely.
- kids are being protected against the dark web
- Users can connect with others with the ability to mange their circles and groups
- Users can convert their spoken speech into text and publish it

## **9.2.Future Works**

We propose more improvement at the system in the future, we propose to make it more intelligence, easier to access, also to make it an android and IOS mobile application so the user can access and use it anywhere and anytime.

Also we propose to more features to allow the user to monitor the child log, in case that he is a parent, also to have chatting feature with group chat, However, the application can have talents session to share talents among users, and for verified public figure they can have their own accounts admin for managing their accounts, also the privacy settings can be more enhanced.

# Chapter 10 : References

## CHAPTER 10 : REFERENCES

1. <https://getfedora.org/>
2. <https://fedoraproject.org/wiki/DNF>
3. <https://httpd.apache.org/>
4. <http://php.net/>
5. <https://getcomposer.org/>
6. <https://laravel.com/>
7. <https://laracasts.com/>
8. <https://stackoverflow.com/>
9. <https://stackexchange.com/>
10. <https://codepen.io/>
11. <https://jsfiddle.net/>
12. <https://www.npmjs.com/>
13. <https://nodejs.org/en/>
14. <https://expressjs.com/>
15. <https://vuejs.org/>
16. <https://angular.io/>
17. <https://www.mongodb.com/>
18. <https://cockpit-project.org/>
19. <https://www.gnome.org/>
20. <https://www.phpmyadmin.net/>
21. <http://docs.mongodb.com/compass/master/>
22. <https://github.com/axios/axios>
23. <https://materializecss.com/>
24. <https://foliotek.github.io/Croppie/>
25. <https://fengyuanchen.github.io/cropperjs/>
26. <https://developers.google.com/maps/documentation/>
27. <https://www.npmjs.com/package/vue-popperjs>
28. <https://www.npmjs.com/package/countries-list>
29. <https://github.com/zhuiker/lamejs>
30. <https://www.npmjs.com/package/blob-to-arraybuffer>
31. <https://momentjs.com/>
32. <https://jquery.com/>
33. <https://www.w3schools.com/>
34. <https://www.npmjs.com/package/jquery-touchswipe>
35. <https://howlerjs.com/>
36. <https://alligator.io/vuejs/raw-html-binding/>
37. <https://www.npmjs.com/package/@xkeshi/image-compressor>
38. <https://getuikit.com/>
39. <https://icomoon.io/>
40. <https://forge.laravel.com/>
41. <https://www.digitalocean.com/>
42. <https://ae.godaddy.com/>
43. <http://restclient.net/>
44. <https://slack.com/>
45. <https://github.com>
46. <https://www.getpostman.com/apps>
47. <https://daneden.github.io/animate.css/>
48. <https://semantic-ui.com/>
49. <http://element.eleme.io/>
50. <https://insomnia.rest/>
51. <https://icomoon.io/>
52. <https://beakerbrowser.com/>
53. <https://icons8.com/>