4STUDY

Ivan Giacomoni 1796069

Marco Carfora 1794568

Andrei Baciu 1796858

Daniele Bufalieri 1794424

Index

1. Introduction

This project borns for the need of find different students in the same **study** field, in order to build some **study groups**, for which all these students can join them for sharing their **knowledge** and **notes** about the courses.

During the pandemic period, relationships between students decreased because of remote lectures, and so retrieve some informations also became difficult.

So the purpose of **4Study** is to support and increase relationships between students, also remotely.
In particular, 4Study will be developed as a **web application**, in which we will find a lot of features:

- **Find a study group** in your area according to a specific subject
- **Sharing notes and documents** related to specific subject
- Give the possibility to students to **improve their language skills** by using a **chat**
- **Find study rooms**
- **Create announcements** for particular requests

2. Development methodology

The methodology adopted for developing 4Study is a **Scrum methodology**: this is an Agile method based on daily meetings in which all the group members will share their opinions on the work done until that moment, focusing on what can be improved in the next iterations.

Each sprint will be of the duration of **one/two weeks**.

All the development process has been done fully remotely, by using videoconferences platforms such as **Google Meet** or **Zoom**, and **GitHub** and **Google Drive** platforms for data and content sharing.

3. Requirements' analysis

The requirements' analysis is very important to develop a product which is useful for the end user. In this phase, it is important to define the **needs** of the **users**, in order to deliver an high quality product, which is able to help them to use our product in a simple way.

3.1 User analysis

The first kind of requirement analysis is the **user analysis**, which consists identifying the target that will use our system. Given that our application is addressed for **students**, we have defined our target audience has the following features:

- Age: 14 – 25
- Gender: Male and Female
- State: Worldwide
- Technology: web application (laptop / Desktop)
- Education: high school / university
- Occupation: student

3.1.1 User personas and scenarios

The next step of the user analysis is to define several **user personas** and **related scenarios**.

User personas are useful to describe the potential users of the system, based on the user profile defined above.

A **scenario** is a story that describes how a particular persona completes a task or behaves in a given situation.

3.1.1.1 First Persona: Roberta

Roberta is a 17 years-old high school student who is studying at Liceo Scientifico Galileo Galilei in Rome. She doesn't like study a lot, she has troubles in learning and focusing both when she is at school and when she is at home. She loves spend time with other people.

It is a Wednesday afternoon and Roberta is going back home after a school morning. After lunch, she tries to study, but due to her inability to focus during study hours, she decides that it is better to find a study group, in order to improve her school marks.

3.1.1.2 Second Persona: Kevin

Kevin is a 22 years-old university student who is studying at La Sapienza University of Rome. He is strongly convinced that all the knowledge should be open and easily accessible to everyone. So, he would like a tool where he can share his notes.

It is a Tuesday afternoon and Kevin is studying. Meanwhile, a lot of his friends message him by asking for his Physics notes, but Kevin doesn't want to waste time by sending his notes to all his friends, one by one. So he would like to use a tool where the notes can be shared forever to all of them.

3.1.1.3 Third Persona: Alice
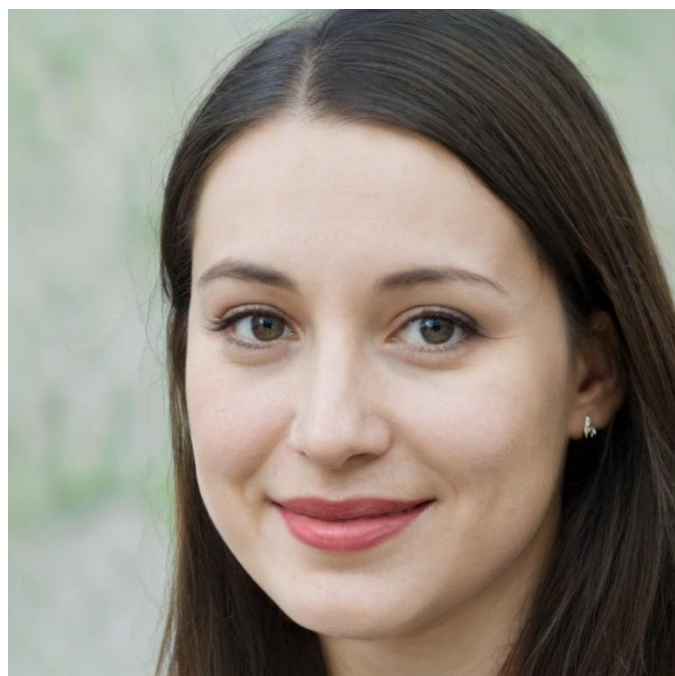
Alice is a 19 years-old high school student who is studying at Liceo Scientifico Shakespeare in Rome. She likes to study a lot and in particular, given that she is at her final year of high school education, she would like to enroll a language study course at university, and so she would like to improve their language skills after the end of the school.

It is a Monday morning and Alice is attending an extra English course, but she is not satisfied about the dialogue skills that the course gives. She would like to find a person to chat with, in order to improve her language knowledge

3.2 Competitors

The next phase of user requirements step is to identify **competitors** and to analyze the **pros** and the **drawbacks** of 4Study with respect to the other applications.

	4Study	StudyDrive	Docsity
Unique Features	All in one: it offers all services useful to students	Job market and companies search	Marketplace notes
Design strengths	Learn English Study groups by map	Mobile app	Video Private lessons
Design weakness	Location based difficulties People matching language skills	Social based approach (Post) Not worldwide availability	Only teaching material Contents not free
Customer base	Customer size unknown	2.400.000	6.600.000
Satisfaction score	Not available	80	Not available
Requirements	Accessible on all browser types	Available on all browser types and mobile OS	Accessible on all browser types
Core features:			
Sharing notes	v	v	v
Find study groups	v	v	x
Language chat	v	x	x
Job search	x	v	x
Video lessons	x	x	v
Earn money	x	x	v
Find study rooms	v	x	x
Create announcements	v	v	x

3.3 Questionnaire

There are many ways that can be used to reach real audience, get an initial **feedback** about our idea and investigate user needs, expectations, perspectives, priorities and preferences.

The most used one is the **questionnaire**, which is easy to use and gives a quick feedback from users, and furthermore it can be useful to present several choices to our users in a clear way.

In order to reach a good amount of users we used **Google Form**.

Questionnaires principal **advantages** are:

- Quickly collecting quantitative data from a large number of users
- Collect information from a large number of users simultaneously
- If designed correctly, it can be quick & easy to analyze the data
- Relatively cheap
- The evaluator must be skilled in creating unbiased surveys (this requires training)
- If it is posted to the web, little effort is required to distribute it Medium

4. Task and Dialogue Analysis

The next steps that we need to do are the **task analysis** and the **dialogue analysis**. They are both considered very important when we deal with the most relevant and representative **use cases** of our system. Obviously, the uses cases of the application can be defined if and only if we previously went through a deep analysis of the requirements, by identifying the main users, the users' needs, and by looking at the competitor's analysis and the results of the questionnaire.

Let's start by introducing the **task analysis**: this is a method used to analyze **people's jobs**, in particular what they do, what things they work with and what they must know. The system analysis' method is different from the task analysis' method, because the first one focuses more on the system design, while the second one focuses more on the **user**.

At the same time, we have differences between cognitive models and task analysis, because the first one focuses more on the internal mental state and the tasks intended as individual work units, while the second one focuses more on the **external actions** and the **whole work** behind the task itself. The general method associated to task analysis is to **observe**, **collect unstructured list of words and actions** and finally **organize using notation or diagrams**. There are several approaches related to task analysis, but we will focus on **task decomposition**, which consists into splitting a task into **subtasks**: in particular, with task decomposition we can describe the actions people do, structure them within task subtask hierarchy and describe order of subtasks. The variant of task decomposition we will use is **Hierarchical Task Analysis (HTA)**, which is the most common. For generating the hierarchy when analyzing a task, we first need to get list of tasks, then we group tasks into higher level tasks and finally we decompose lowest level tasks further.
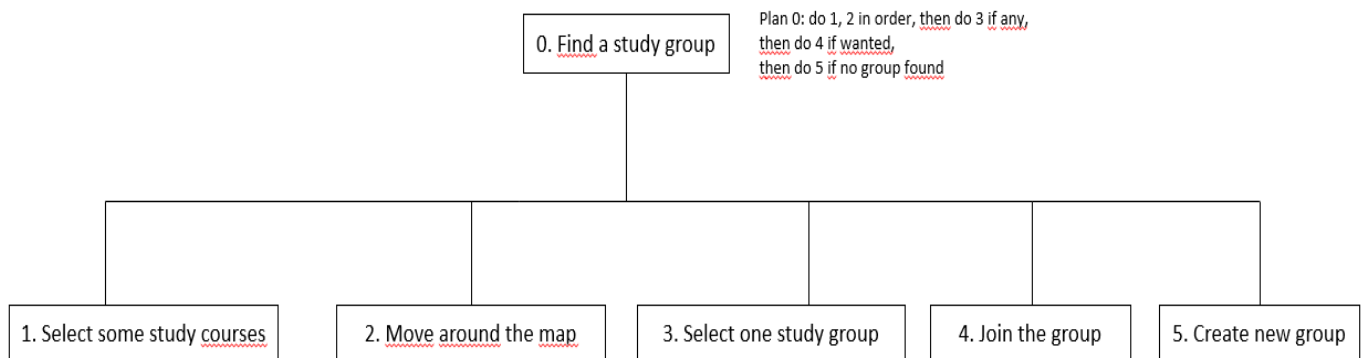
Now let's focus on **dialogue analysis**. The dialogue is also considered very important in the human computer interaction literature, especially from the point of view of **user interfaces**, when we deal with the syntactic level of **human-computer conversation**. The dialogue analysis helps us gain a better understanding of the whole system, from a lot of point of views. In order to conduct the dialogue analysis, we will use the **State Transition Network (STN)**, one for each of our most relevant tasks for which we have first defined the HTAs. A STN is a **directed graph** where each node represents a **state** in which the user will go through during the execution of the task, while each edge represents a **transition** from a state to another state, that is triggered by a **user action** and performed by the **system** in response to the user trigger.

We have identified four relevant tasks for our application, and now we will show, for each of them, their HTAs and STNs.
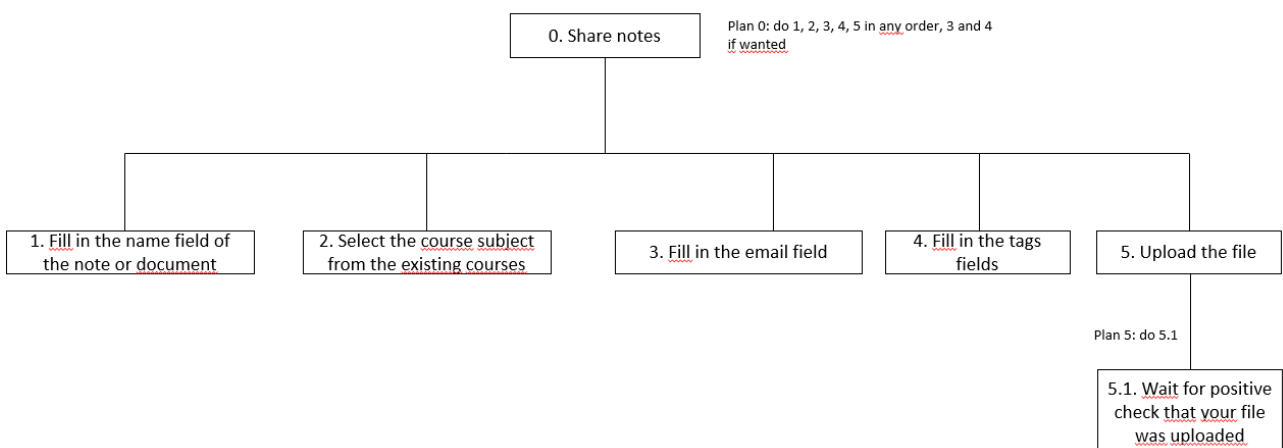
4.1 Hierarchical Task Analysis

4.1.1 Find a study group

The first task we show is the one of **finding a study group**. This task allows the user to filter possible existing study groups on an interactive map, in order to join one of them if any available or, in case no group is found, to create himself a new group. This task is representative for the application, because one of the main goal of our application is that of making possible for students to improve their knowledge and concentration by studying with other people. Below we report the HTA:

```
                        ┌─────────────────────┐   Plan 0: do 1, 2 in order, then do 3 if any,
                        │ 0. Find a study group│   then do 4 if wanted,
                        └─────────────────────┘   then do 5 if no group found

┌──────────────────┐ ┌──────────────────┐ ┌──────────────────┐ ┌────────────┐ ┌──────────────┐
│1. Select some     │ │2. Move around    │ │3. Select one     │ │4. Join the │ │5. Create new │
│  study courses    │ │   the map        │ │   study group    │ │   group    │ │   group      │
└──────────────────┘ └──────────────────┘ └──────────────────┘ └────────────┘ └──────────────┘
```
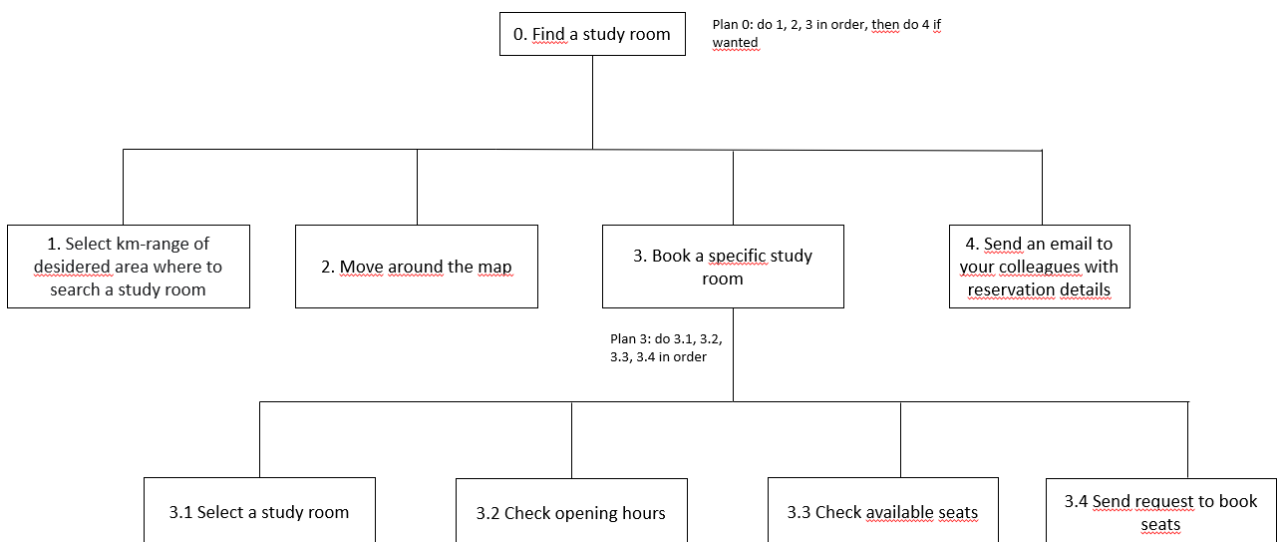
4.1.2 Share notes

The second task we show is the one of **sharing notes**. This task allows the user to upload files on the application, in order to share notes with other people: in particular, it will be required to insert the name of the document and the course subject and, if desired by the user, the tags and the email addresses of some specific users, in order to send them the document. This task is representative for the application, because another main goal of our application is that of making possible for students to share their knowledge on the application, in order for a student who wants to study at home alone to improve in one or more specific subjects. Below we report the HTA:

```
                        ┌─────────────────┐   Plan 0: do 1, 2, 3, 4, 5 in any order, 3 and 4
                        │ 0. Share notes  │   if wanted
                        └─────────────────┘

┌──────────────────┐ ┌──────────────────┐ ┌──────────────────┐ ┌────────────┐ ┌──────────────┐
│1. Fill in the name│ │2. Select the     │ │3. Fill in the    │ │4. Fill in  │ │5. Upload the │
│field of the note  │ │course subject    │ │   email field    │ │the tags    │ │   file       │
│or document        │ │from the existing │ │                  │ │fields      │ │              │
│                   │ │courses           │ │                  │ │            │ │              │
└──────────────────┘ └──────────────────┘ └──────────────────┘ └────────────┘ └──────────────┘
                                                                                Plan 5: do 5.1

                                                                              ┌──────────────┐
                                                                              │5.1. Wait for │
                                                                              │positive check│
                                                                              │that your file│
                                                                              │was uploaded  │
                                                                              └──────────────┘
```
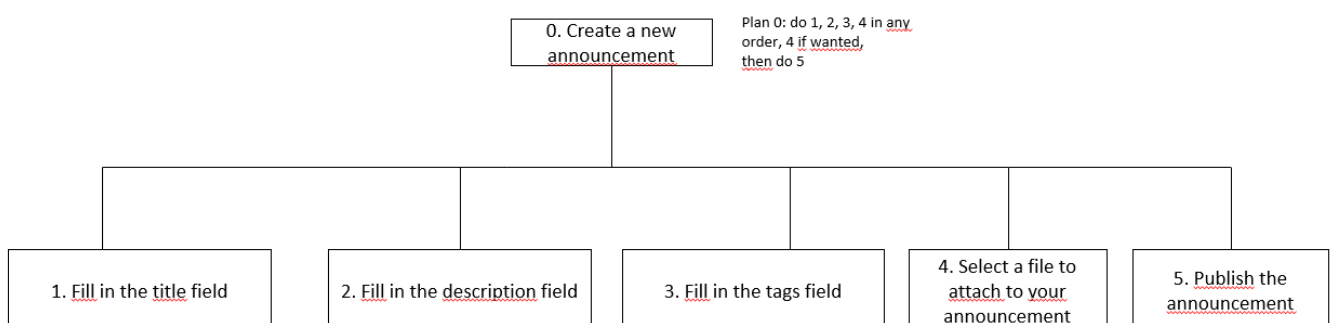
4.1.3 Find a study room

The third task we show is the one of **finding a study room**. This task allows the user to find a study room: in particular, the user will have the possibility to select the km-range of the desired area where to search a study room, and then, by using an interactive map, he will book a specific study room, if possible. Also, he will have the possibility to book the room for his colleagues, by sending an email with reservation details. This task is representative for the application, because another main goal of our application is that of making possible for students to exploit the university/school environments, in order for them to improve their study skills. Below we report the HTA:

```
                        ┌──────────────────────┐   Plan 0: do 1, 2, 3 in order, then do 4 if
                        │ 0. Find a study room │   wanted
                        └──────────────────────┘
        ┌───────────────┬──────────────┴────────────┬───────────────────┐
┌─────────────────┐ ┌──────────────────┐ ┌──────────────────┐ ┌──────────────────┐
│ 1. Select km-   │ │ 2. Move around   │ │ 3. Book a        │ │ 4. Send an email │
│ range of        │ │ the map          │ │ specific study   │ │ to your          │
│ desidered area  │ │                  │ │ room             │ │ colleagues with  │
│ where to search │ │                  │ │                  │ │ reservation      │
│ a study room    │ │                  │ │                  │ │ details          │
└─────────────────┘ └──────────────────┘ └──────────────────┘ └──────────────────┘
                              Plan 3: do 3.1, 3.2,
                              3.3, 3.4 in order
        ┌─────────────────┬──────────┴────────┬─────────────────┐
┌───────────────┐ ┌───────────────┐ ┌──────────────────┐ ┌──────────────────┐
│ 3.1 Select a  │ │ 3.2 Check     │ │ 3.3 Check        │ │ 3.4 Send request │
│ study room    │ │ opening hours │ │ available seats  │ │ to book seats    │
└───────────────┘ └───────────────┘ └──────────────────┘ └──────────────────┘
```
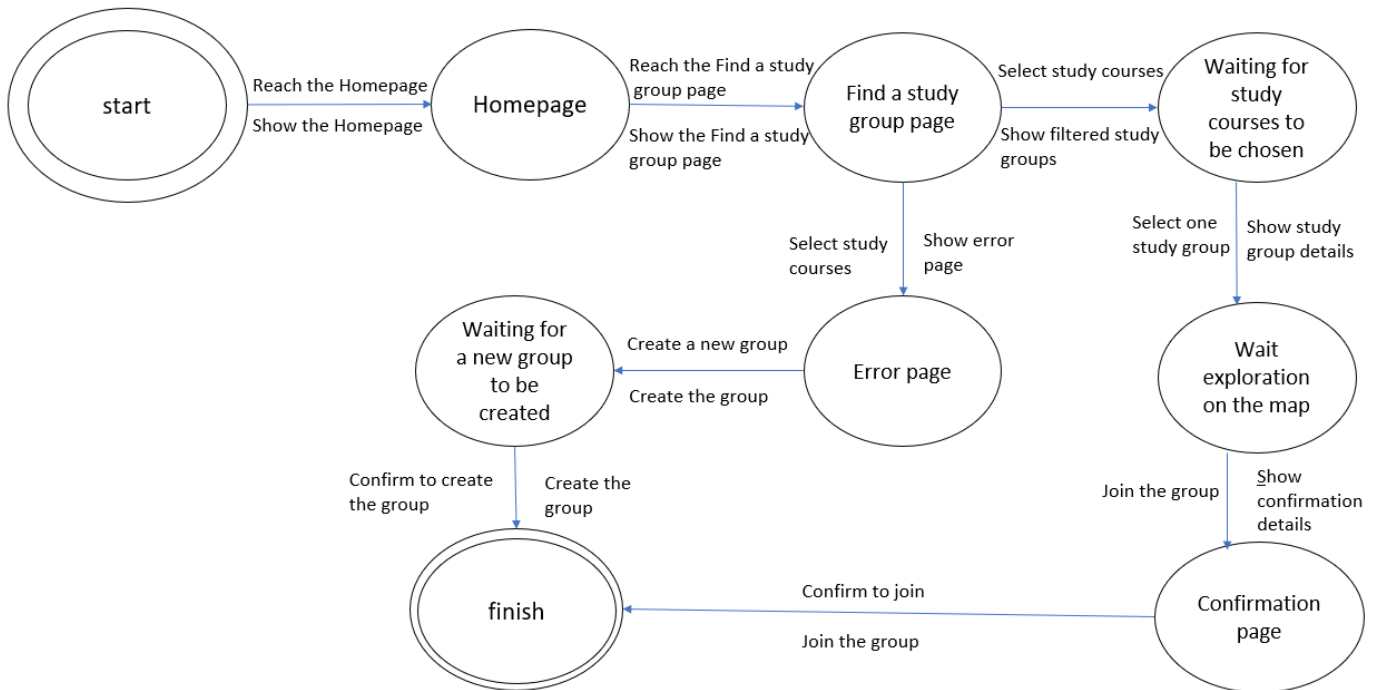
4.1.4 Create a new announcement

The fourth task we show is the one of **creating a new announcement**. This task allows the user to post an announcement on the application: in particular, the user will be required to insert a title, a description and at least one tag for the announcement, and also to attach a file to it, if he wants. This task is representative for the application, because another main goal of our application is that of making the life of students a lot easier, by giving them the possibility to open a discussion thread below each announcement, for any kind of reasons, such as sharing of personal notes, requests to find people or asking for the solution of exercises. Below we report the HTA:
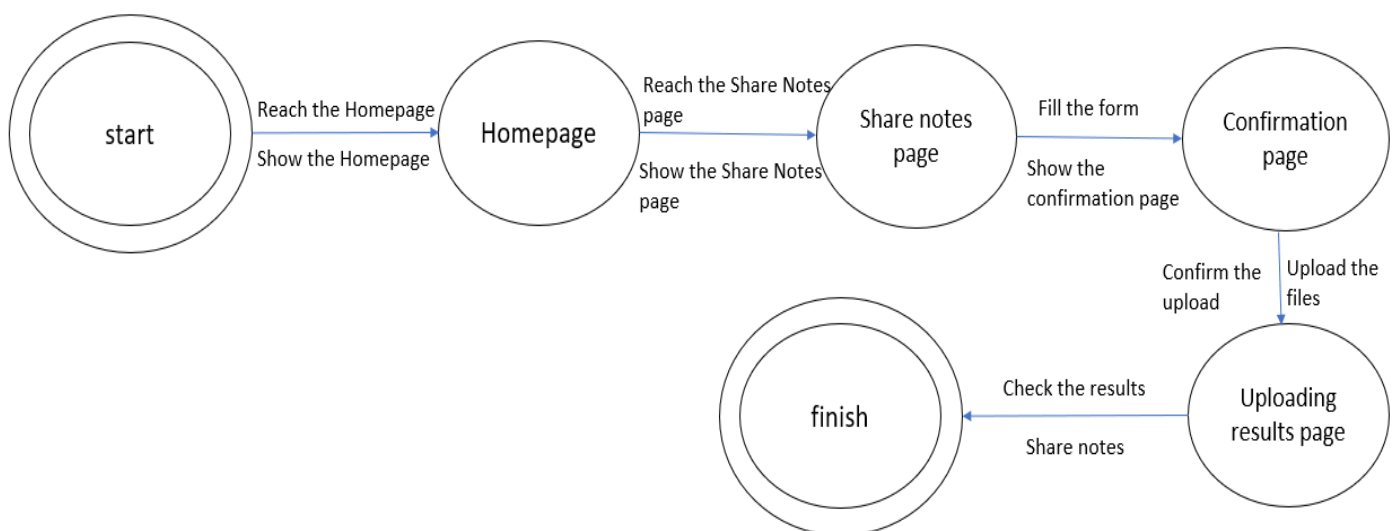
```
                    ┌──────────────────┐   Plan 0: do 1, 2, 3, 4 in any
                    │ 0. Create a new  │   order, 4 if wanted,
                    │ announcement     │   then do 5
                    └──────────────────┘
    ┌──────────────┬──────────┴───────┬──────────────┬──────────────┐
┌────────────┐ ┌──────────────┐ ┌────────────┐ ┌────────────┐ ┌────────────┐
│ 1. Fill in │ │ 2. Fill in   │ │ 3. Fill in │ │ 4. Select  │ │ 5. Publish │
│ the title  │ │ the          │ │ the tags   │ │ a file to  │ │ the        │
│ field      │ │ description  │ │ field      │ │ attach to  │ │ announce-  │
│            │ │ field        │ │            │ │ your       │ │ ment       │
│            │ │              │ │            │ │ announce-  │ │            │
│            │ │              │ │            │ │ ment       │ │            │
└────────────┘ └──────────────┘ └────────────┘ └────────────┘ └────────────┘
```

4.2 State Transition Networks

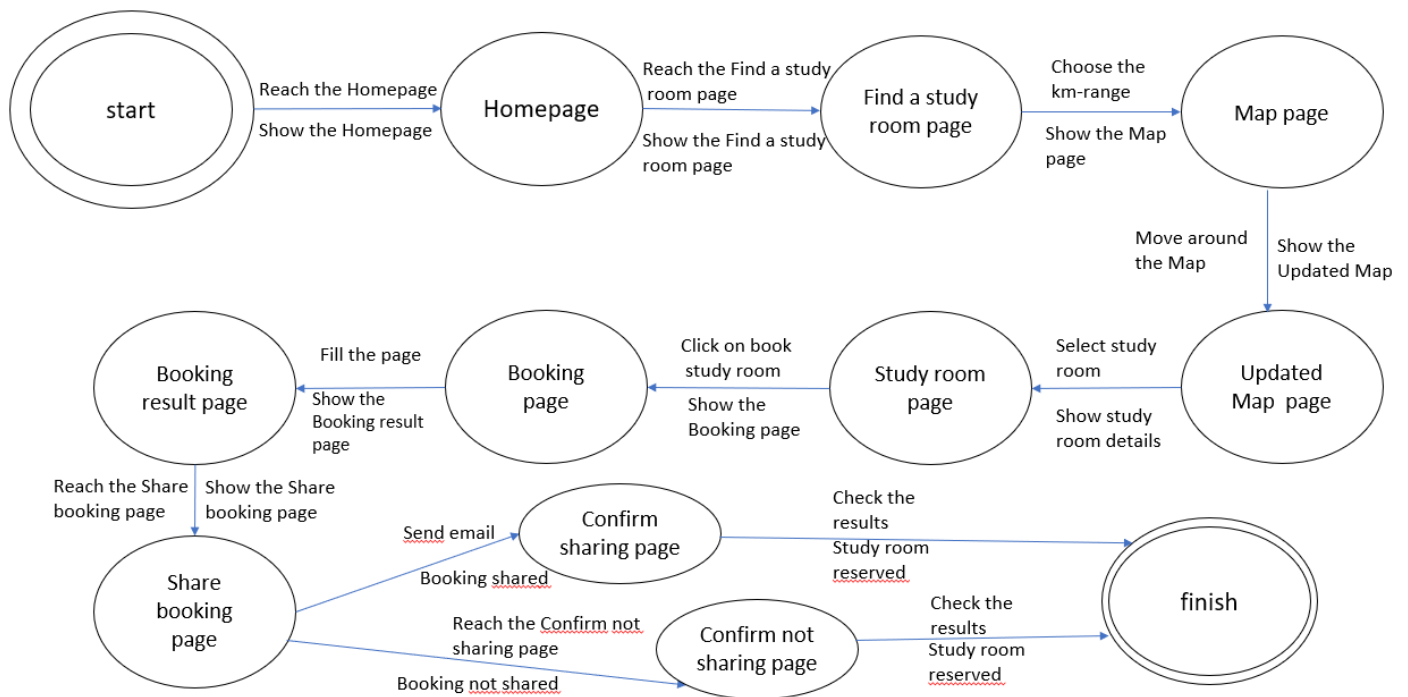Now we will show the **State Transition Network** for each of the four tasks that we have just analyzed with HTAs.
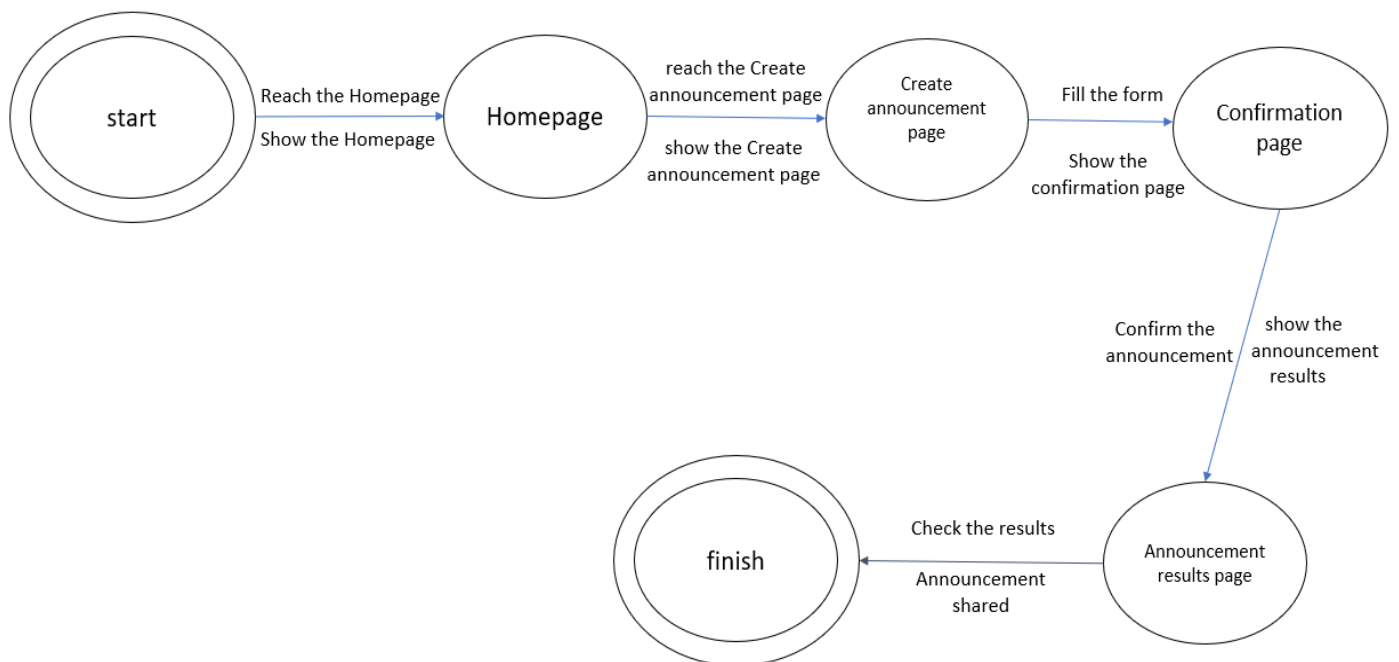
4.2.1 Find a study group

### 4.2.2 Share notes

4.2.3 Find a study room

start → Homepage
- Reach the Homepage
- Show the Homepage

Homepage → Find a study room page
- Reach the Find a study room page
- Show the Find a study room page

Find a study room page → Map page
- Choose the km-range
- Show the Map page

Map page → Updated Map page
- Move around the Map
- Show the Updated Map

Updated Map page → Study room page
- Select study room
- Show study room details

Study room page → Booking page
- Click on book study room
- Show the Booking page

Booking page → Booking result page
- Fill the page
- Show the Booking result page

Booking result page → Share booking page
- Reach the Share booking page
- Show the Share booking page

Share booking page → Confirm sharing page
- Send email
- Booking shared

Share booking page → Confirm not sharing page
- Reach the Confirm not sharing page
- Booking not shared

Confirm sharing page → finish
- Check the results
- Study room reserved

Confirm not sharing page → finish
- Check the results
- Study room reserved

4.2.4 Create a new announcement

start → Homepage
- Reach the Homepage
- Show the Homepage

Homepage → Create announcement page
- reach the Create announcement page
- show the Create announcement page

Create announcement page → Confirmation page
- Fill the form
- Show the confirmation page

Confirmation page → Announcement results page
- Confirm the announcement
- show the announcement results

Announcement results page → finish
- Check the results
- Announcement shared

14

5. Prototype Zero

In this section we will present the **first prototype** of our application. In order to build the prototype, there was first the needing to gather all the possible informations from the questionnaire, because we needed to be sure whether some of the first ideas could be interesting or not for our typical users. The realization of the first prototype has been conducted with the tool of **Figma**: Figma allowed us to create all the **mockups** of the application, thanks to his powerful and creative tools, but also allowed us to create a **dynamic prototype**, where all the interfaces where connected between them, in order to show the **interaction** between the system and the user in a clear way.

Now we will start showing the most representative interfaces of our first prototype, because the entire application will be composed by many other of them, so if you want to check the entire prototype, you can check that online on Figma, by looking at the bibliography.

The first mockup we show is that of the **Homepage**: after the user has registered or logged himself into the application, he will be redirected on the Homepage, that represents the start point from where navigate into the system; in particular, he'll have the possibility to explore the **main sections** of the application, such as **the announcement section, the study room section, the study group section and the notes section**. Below we report the Homepage:
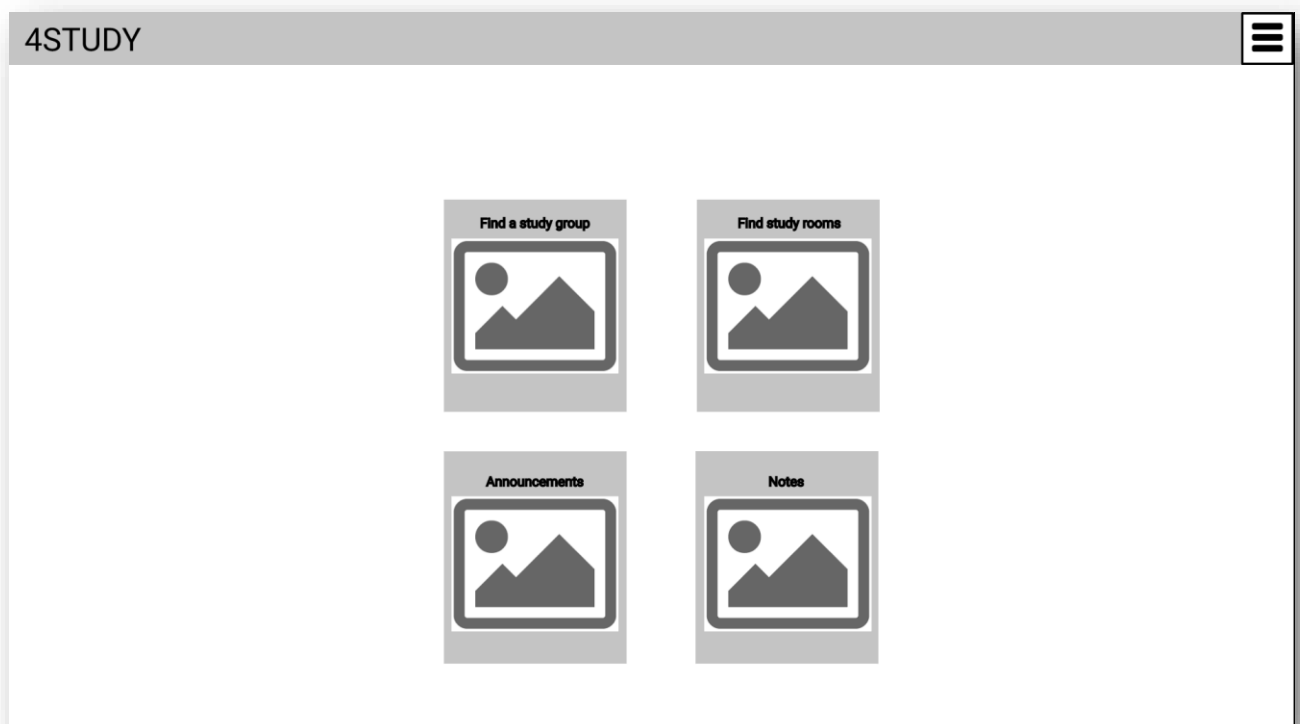
On the top left of each webpage, we can notice the **4Study logo**, while on the top right, we can see a **drop-down menu**: the user will be allowed to interact with it from every interface, in order to access the **profile section**, the **chat section** and also he will have the possibility to **logout** from the application. Below we report the drop-down menu opening after the interaction of the user on the Homepage:
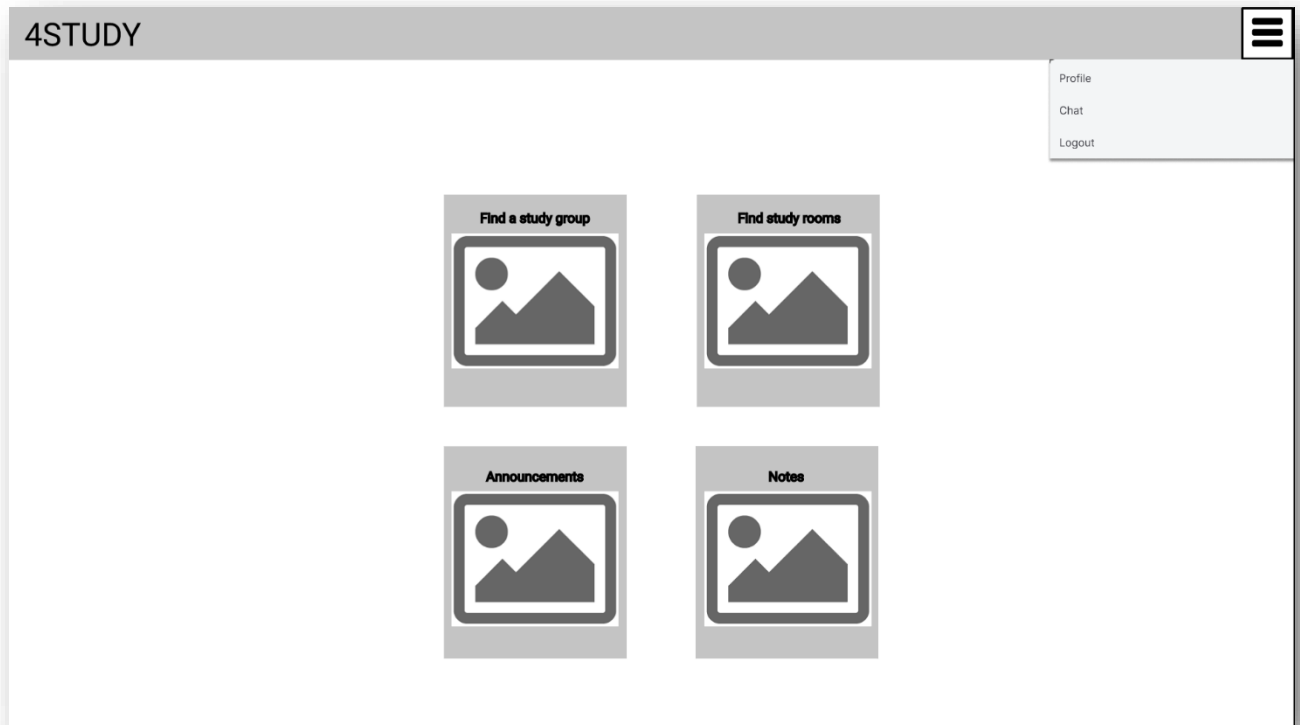
By clicking on the **'Find a study group' box**, the user will be redirected on a new page where there will be shown a **map** on the left part of screen, and a drop-down menu on the right, with **a list of courses**.
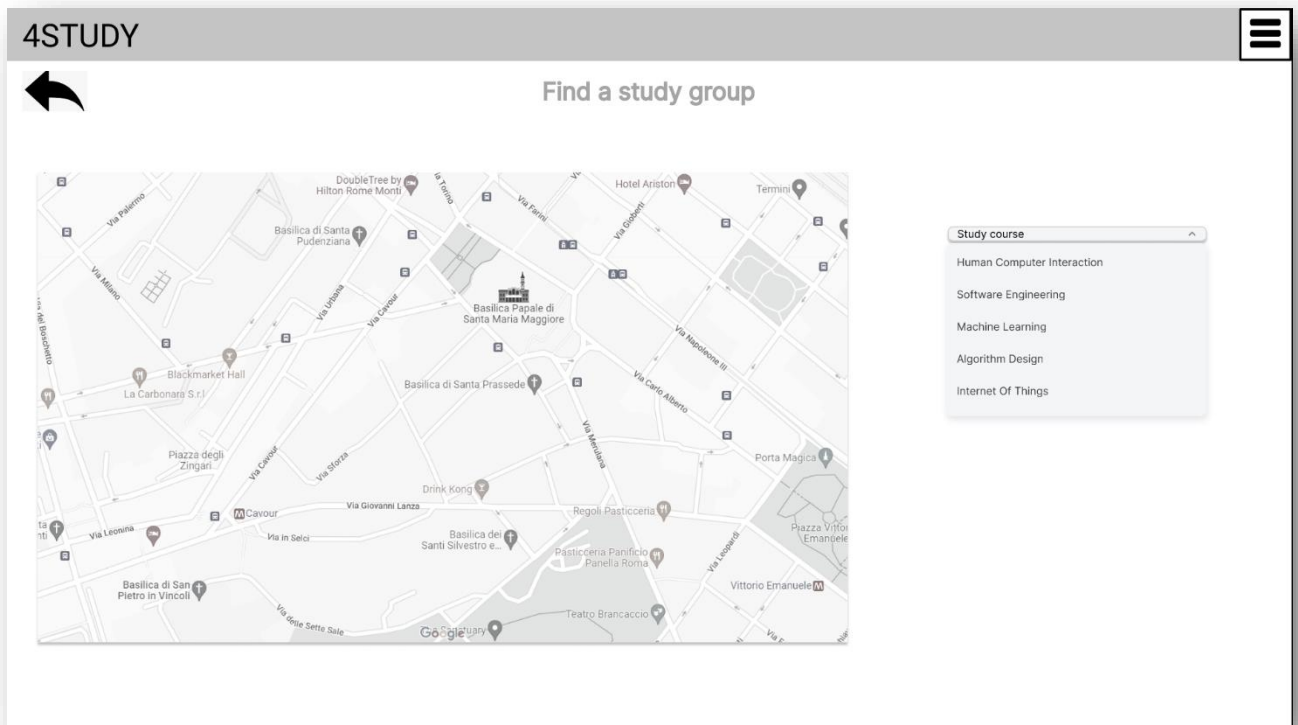
This section is very important for our users, because they will have the possibility to find one or more **study group** located on a **map** pointing on their study town, in order for them to **join** an existing study group, if they want. The main goal of this section is to allow students to easily find informations about existing study groups in their zone with the help of an **interactive map**, in order for them to **increase their study skills** and **improve their marks**.

Starting from now, we will show all steps that the user needs to complete in order to join a group. By clicking, for example, on Software Engineering, the map will change by showing to the user some existing study groups of Software Engineering located in different areas, with the help of **markers**, and below we report the related mockup:

By clicking on one of this marker, each associated to a Software Engineering group, he will see the details related to that group in a popup, in particular the group name, a short description, the address where the group is located and the number of participants. If the user is interested, there will be immediately the possibility for him for join the group. Below we report the related mockup:
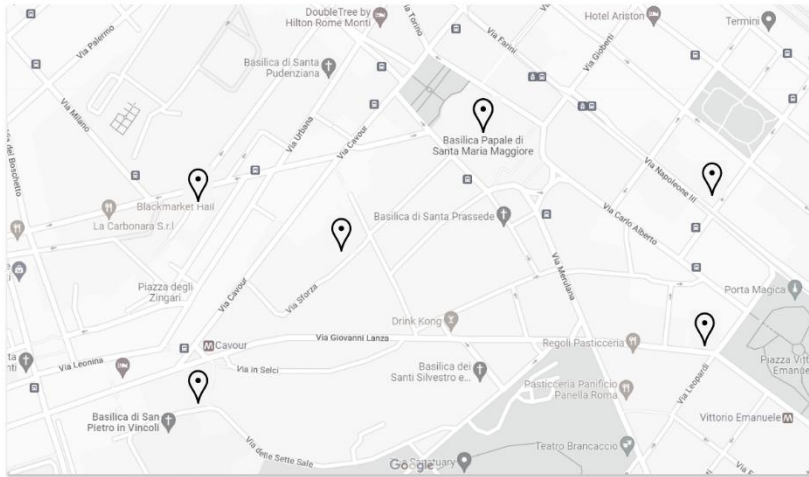
4STUDY

Find a study group

Select a study course:

Software Engineering

4STUDY

Select a study course:

Software Engineering

Study group description: this is a group for Software Engineering

Group name: Software Engineering group

Address: Via Sforza 12

Partecipants: 4

join the group
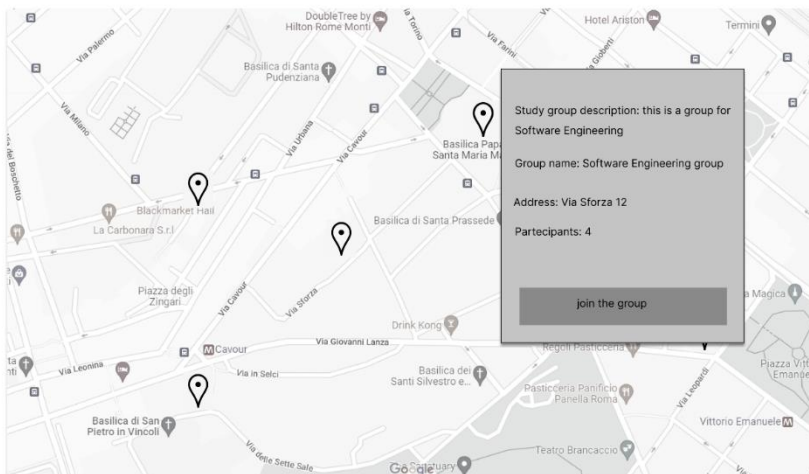
In case no group is found for a certain subject, for example Machine Learning, the user will be redirected to an **error page**, with the possibility for him to search for another group or to **create a new group** for that subject, in our case Machine Learning. Below we report the related mockup:
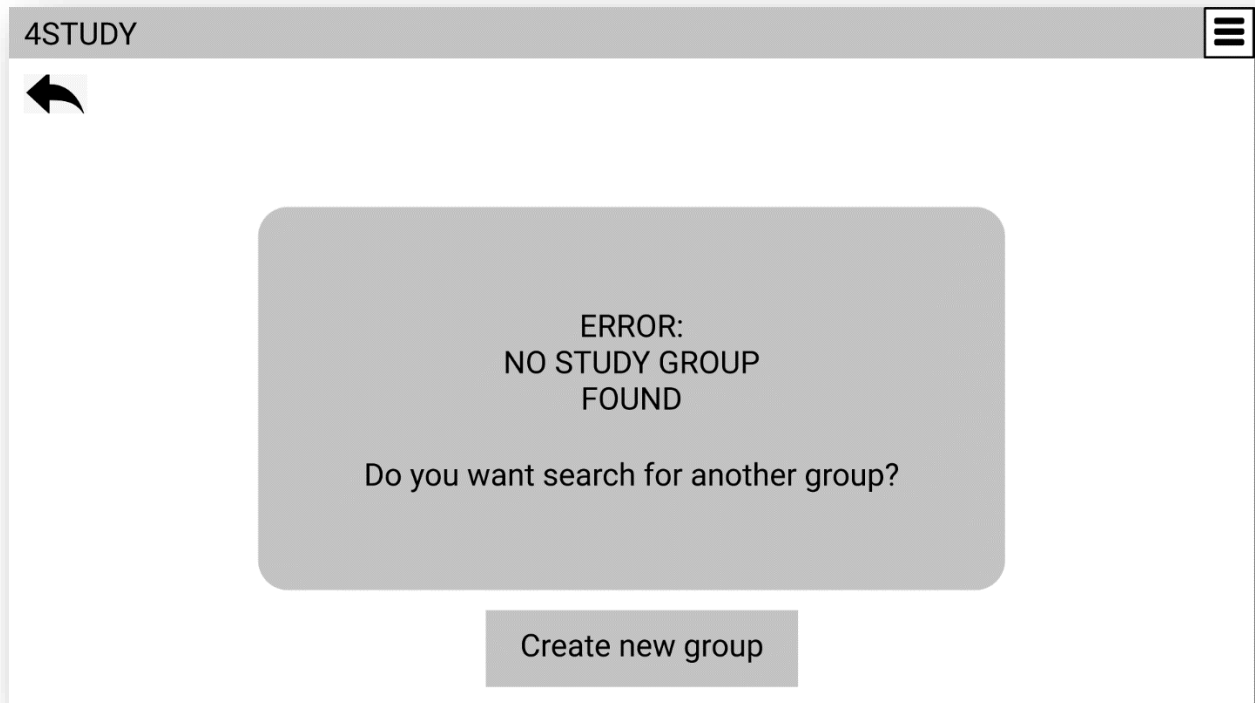
If the user decides to create a new group, he will be redirected to a **form**, where he will be asked to insert the **group name**, the **group course** and the **address**. Below we report the related mockup:

Create a new group

Group name

Insert group name

Group course

Insert group course

Address

Insert the address

Create group

Going back to the Homepage, and by clicking on the **'Find study rooms' box**, the user will be redirected on a new page where there will be shown a simple **text field** where he needs to insert the **city** where he studies. Below we report the related mockup:

Insert the city where study

Insert city

Search

The main purpose of this section is that of giving the possibility to **find one or more study rooms** located on an **interactive map** pointing on their study town, in order allow students to easily find free study rooms for studying alone or together with friends, if they want. This section has also another purpose, which is that of helping students who prefer studying in presence more than studying remotely at home, just like they did before the **Covid-19 pandemic**.

Starting from now, we will show all steps that the user needs to complete in order to book a study room. After having insert a city, by clicking on the **'Search' button**, the user will be redirected to the **interactive map** with **markers**, each representing a different **study room** in that city. Below we report the related mockup:
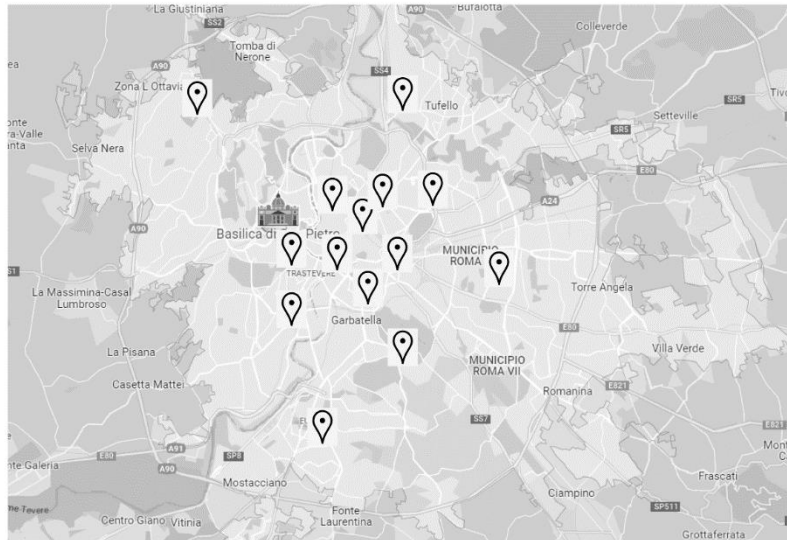
By clicking on one of these markers, a **popup** will appear on screen, by giving us some useful **details** about that study room, such as the **address**, the **opening hours** and the **current available seats**. The user will have immediately the possibility to book a seat, by clicking on the **'Book a seat'** button. Below we report the related mockup:
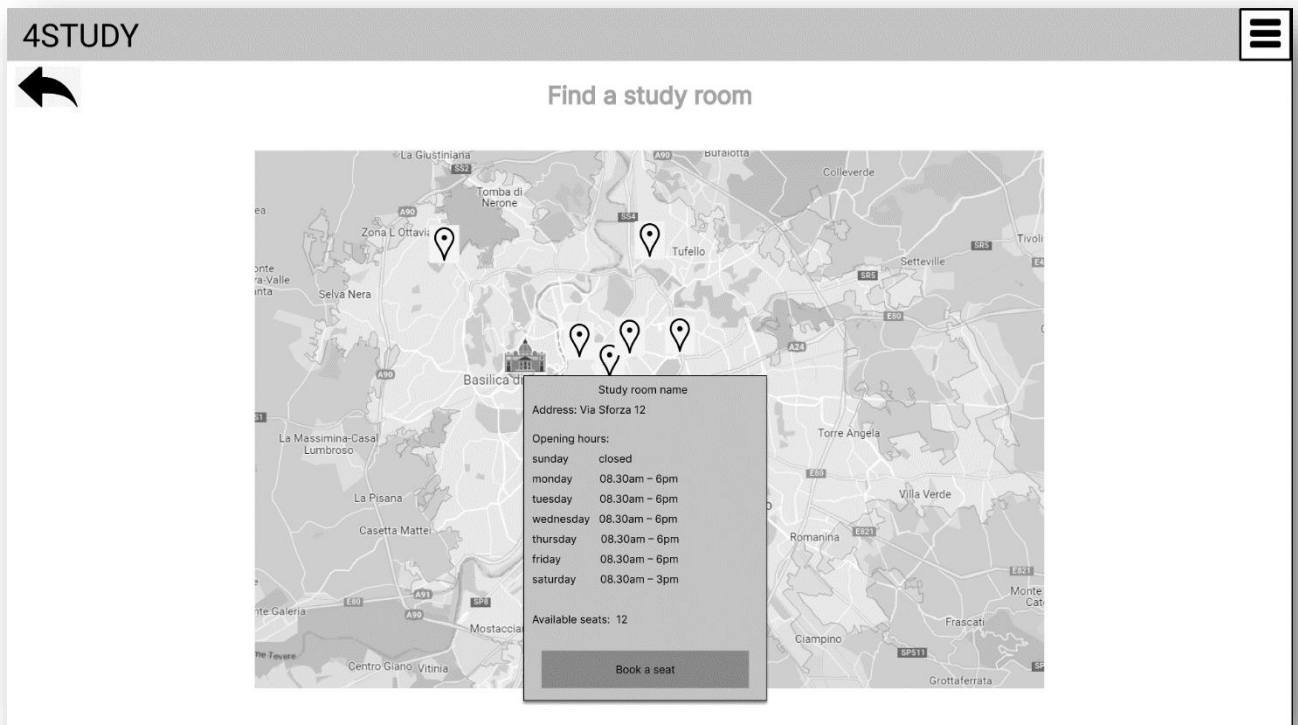
By clicking on the 'Book a seat' button, the user will be redirected to a **confirmation page**, for which we report the related mockup below:
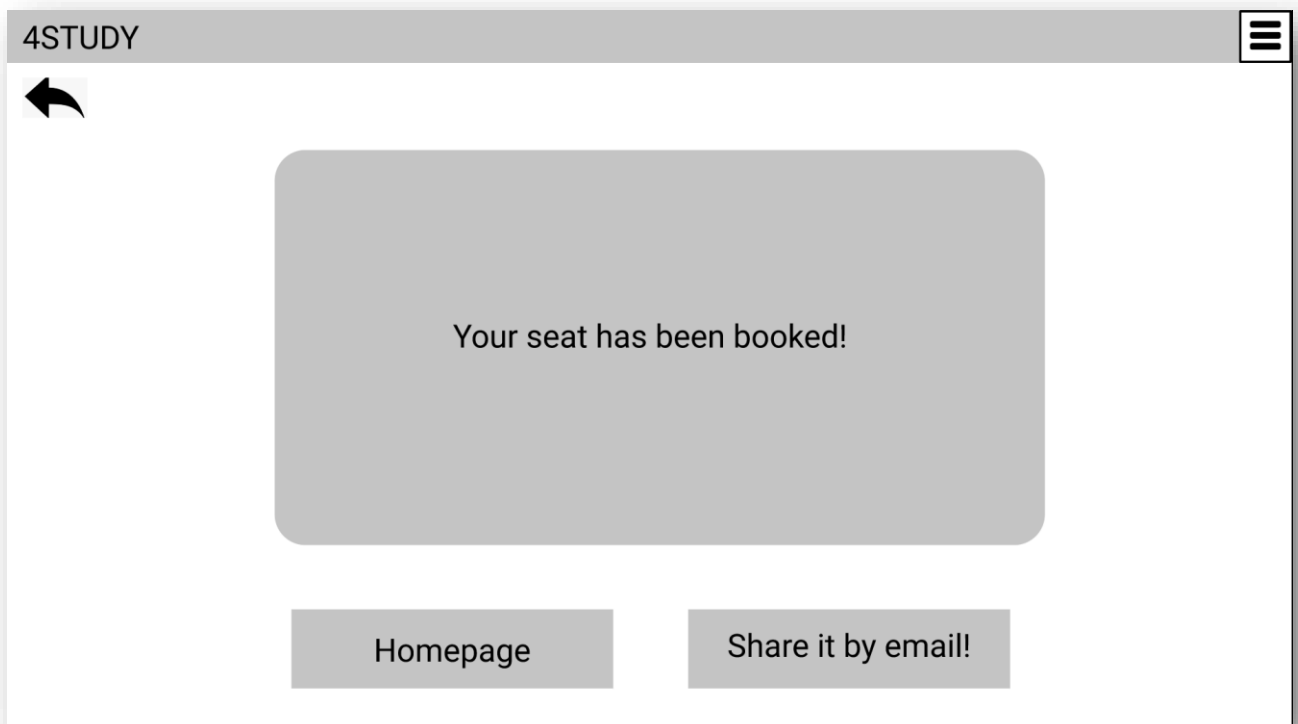
By clicking on the **'Share it by email!'** button, the user will be redirected to another webpage, in which, if he wants, he will be allowed to insert some **email addresses** of his colleagues, in order to send them by email the **reservation** for that study room. Below we report the related mockup:

Going back to the Homepage, and by clicking on the **'Announcements'** box, the user will be redirected on a new page where he will be shown a **table** containing all **announcements** published by **users** of the application. Below we report the mockup:
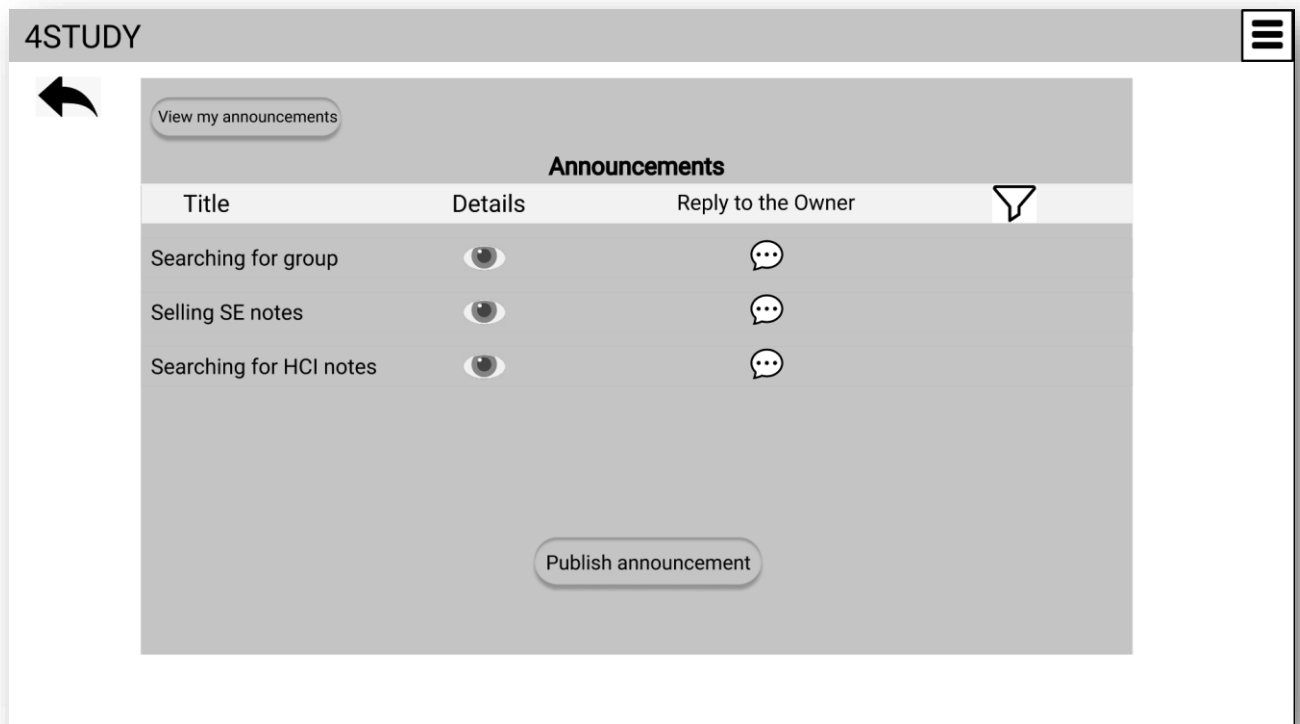
The main purpose of this section is that of allowing users to quickly find some useful **information** and **news** about the main topics of our app, such as **study groups**, **notes**, **suggestions on how to solve some exercise**, and so on.

Starting from now, we will show all actions users can do when dealing with announcements. By clicking on the **'Details'** icon of an announcement, for example the first one, the user will be redirected to a webpage in which all **details** related to that announcement will be shown, such as the **title**, the **description**, **tags** and also a **document,** if available. Below we report the related mockup:

Announcement detail

Title	Searching for group
Description	I'm searching for a group of at least 3 people
Tags	#group #people #3
Document	open

Going back to the previous page, by clicking on the **'Publish announcement'** button, the user will be redirected to a **form**, in which he will have to specify all **informations** related to a new announcement that he wants to **publish**, such as the **title**, the **description**, the **tags** and, if he wants, a **document** with additional details. Below we report the related mockup:

Going back to the previous page, we can see that there is also the possibility, for each announcement, to **write** directly to the **owner** of that specific announcement by using our **chat**, but we will talk later about that. In the same way, we will show later the **filtering** option for the announcements, accessible from the related icon in the table header.

Let's now focus on the **'View my announcements'** button: by clicking on it, the user will be redirected to a different webpage, in which he will have the possibility to view in a table all **his announcements**. Below we report the related mockup:
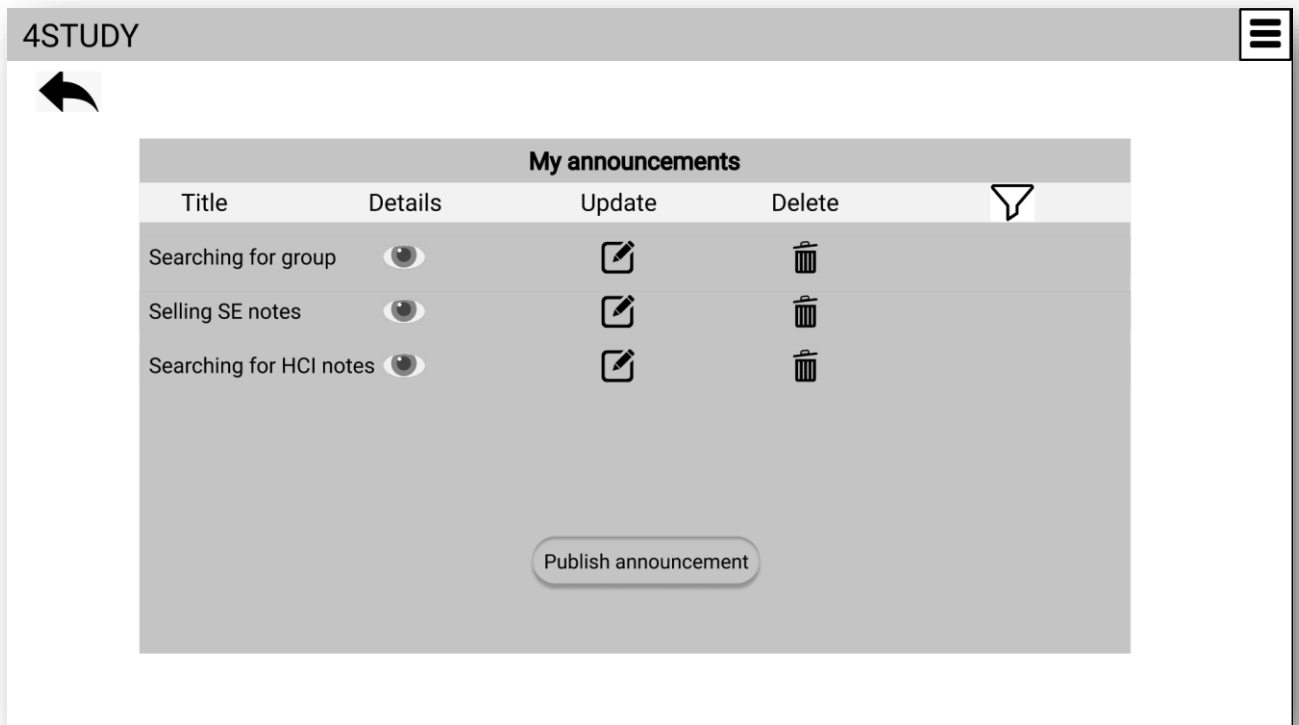
Starting from here, the user can **view the details**, **publish**, **update**, **remove** or **filter** his announcements. The visualization of the details and the publication of an announcement is analogue as before, so let's see the **update** of an announcement first, which will be possible by clicking on the related icon. The user will be redirected to a **form**, which will be already filled up with the current details of the announcement, and he will have the possibility to **update** each field of the announcement. Below we show the related mockup:

Update announcement

Title

Searching for group

Description

I'm searching for a group of at least 3 people

Add tags

#group #people #3

Choose file

Browse

Update announcement
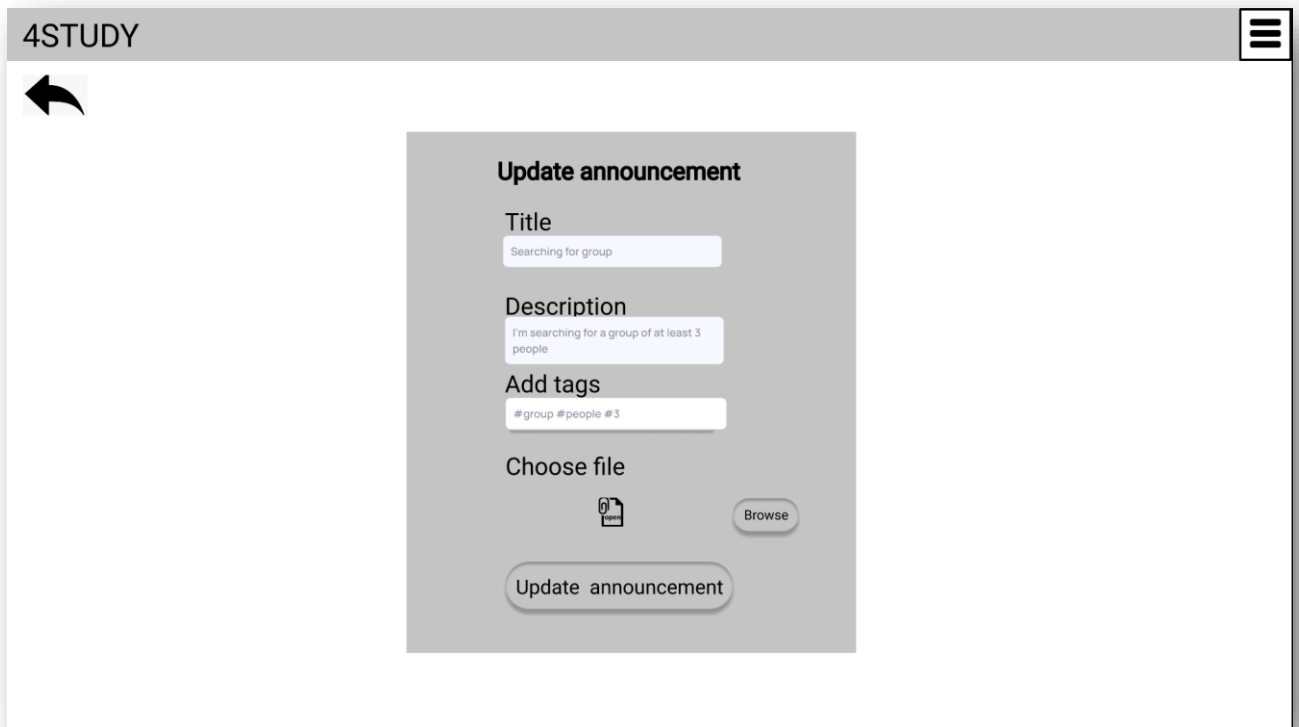
Going back to the previous page, the user will also have the possibility to **delete** an announcement by clicking on the corresponding icon. This interaction will make a **popup** appear on the screen, by asking to the user a **confirmation**. Below we report the related mockup:

4STUDY

My announcements

Title	Details	Update	Delete	

Are you sure that you want to delete this announcement ?

YES   NO

Publish announcement

Finally, the user will also have the possibility to **filter** the announcements, by clicking on the related icon in the table header. He will be redirected to a **form**, in which he will be able to insert some **tags**, in order to quickly filter announcements associated to those tags. For example, in our case we are filtering announcements who are associated with the tag **#group**. Below we report the related mockup:
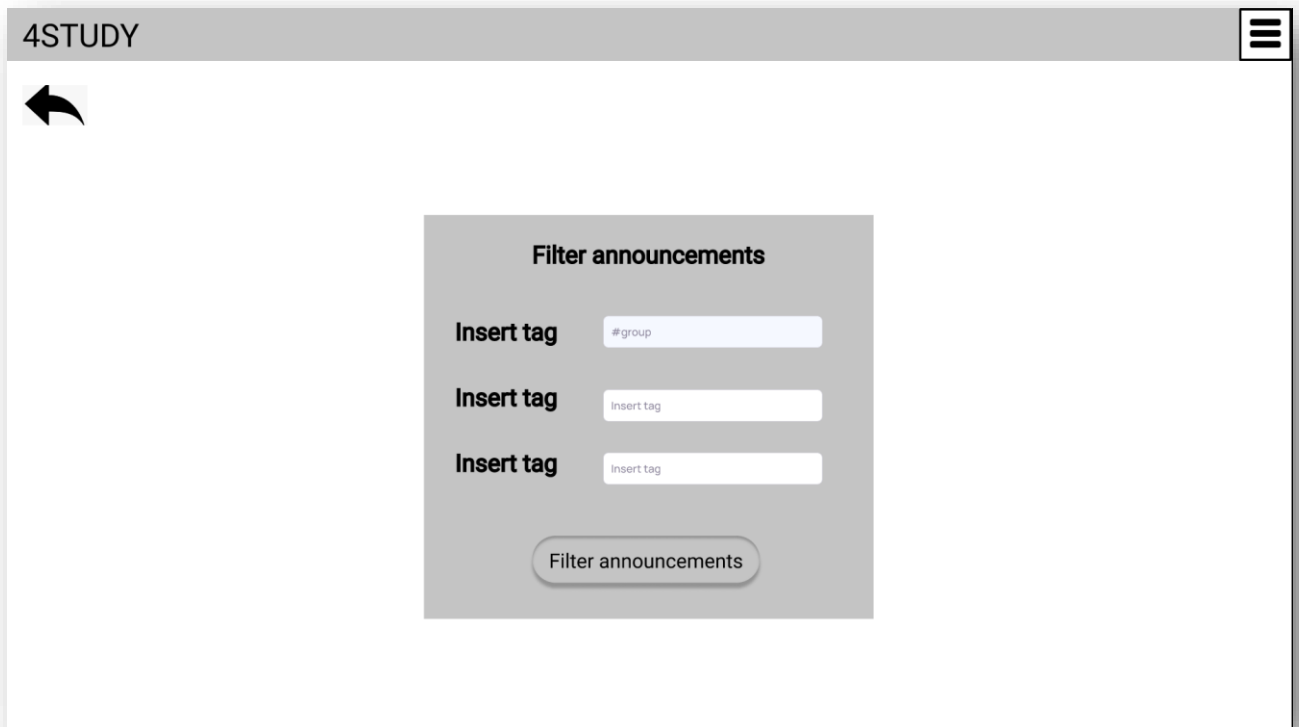
By clicking on the **'Filter announcements'** button, the application will show only announcements associated with the tag **#group**. Below we report the related mockup:

| My filtered announcements				
Title	Details	Update	Delete	✈
Searching for group	◉	✎	🗑	
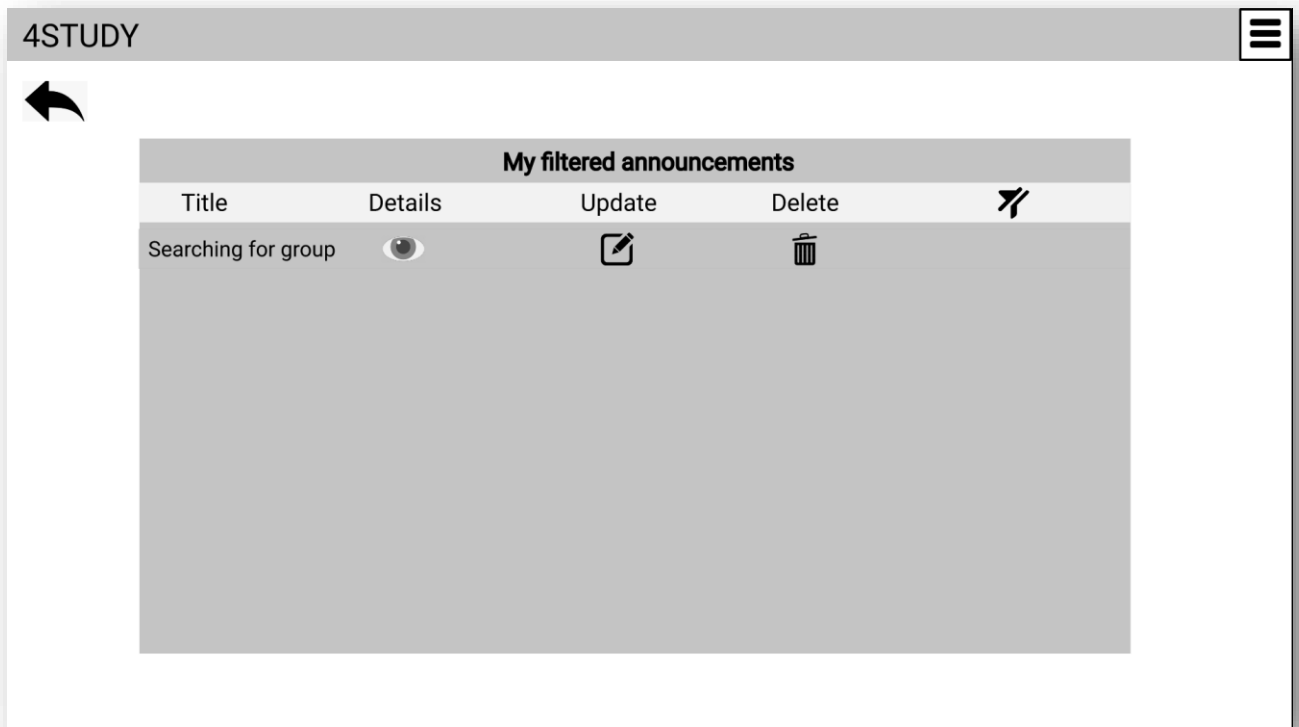
Going back to the Homepage, and by clicking on the **'Notes'** box, the user will be redirected on a new page where he will be shown a **table** containing all **notes** shared by **users** of the application. The main purpose of the 'Notes' section is that of giving the possibility to students to **share their notes** for other students that could get an easy and quick knowledge for preparing a subject or a test. For simplicity, we only report the webpage related to the **personal notes of the user currently logged in the application**, because the management of the notes is analogue to that of announcements.
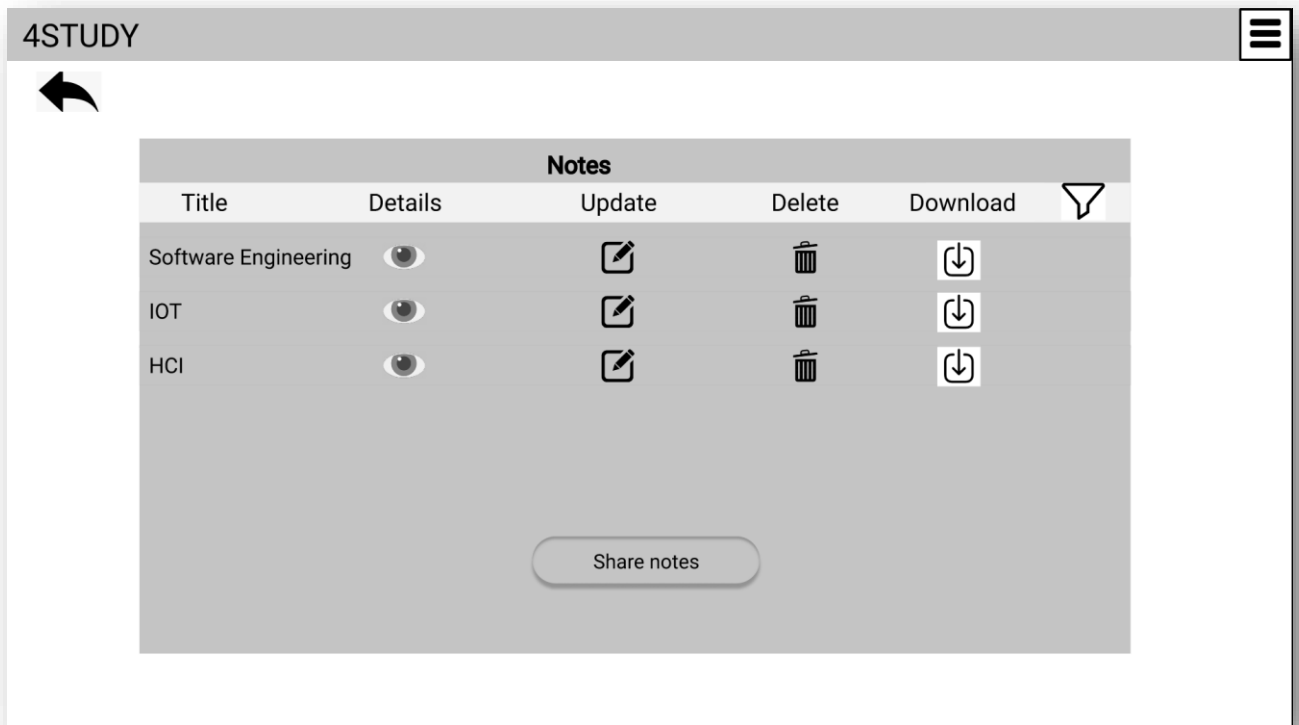
In particular, from the above mockup, we can notice how the user has the possibility to **download** all his notes, by clicking on the related icon.

Now, by opening the **drop-down menu** on the top right of the page, and by selecting **'Chat'**, the user will be redirected to the **chat** of our application. We report below the related mockup:
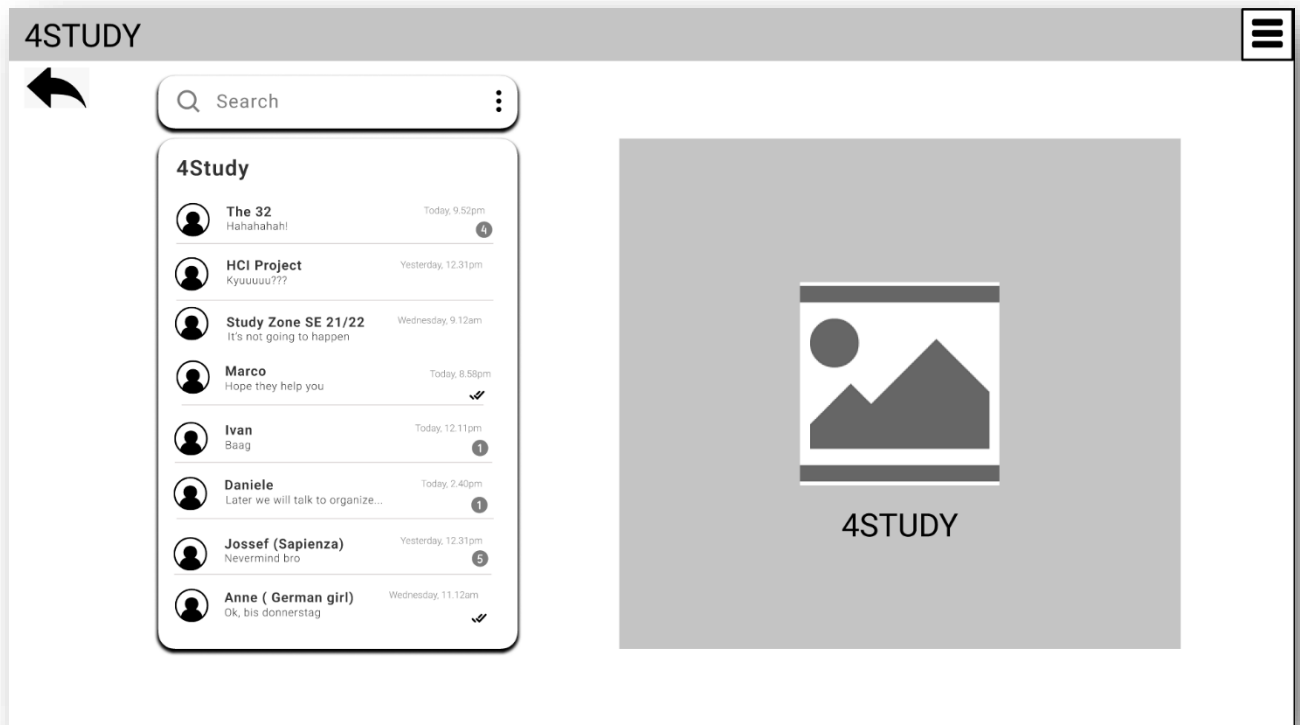
The main purpose of the chat is that of **encourage** students to **communicate** and **share opinions** about their study courses and subjects, in order also to **improve** their communication skills. In the mockup above, we can see all the **recent chats** of the user, both private and groups, on the left, while on the right there will be the **4Study logo**, and on top of the chats there is a **search bar**.

By clicking on the 'three dots' icon located in the search bar a **drop-down menu** will appear on screen, by showing us the possibility to create a **new group**. Below we report the related mockup:
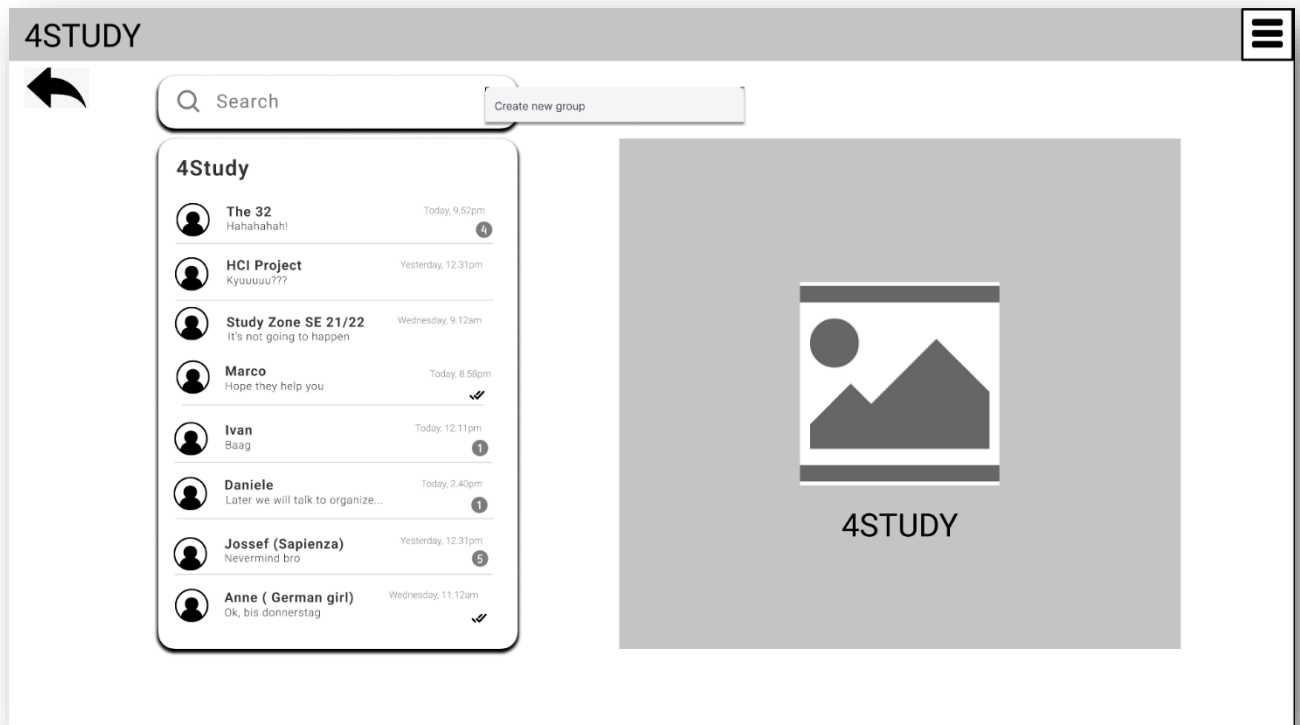
If the user will click on the **'Create a new group' option**, he will be redirected to the form related to the creation of a new group, that we already saw when dealing with the search of a study group.

By clicking on the **Marco private chat**, the chat with Marco will appear on the right, by showing **the last messages** of the conversation, giving also the possibility to the users to continue talking by **sending messages or files**. Below we report the related mockup:
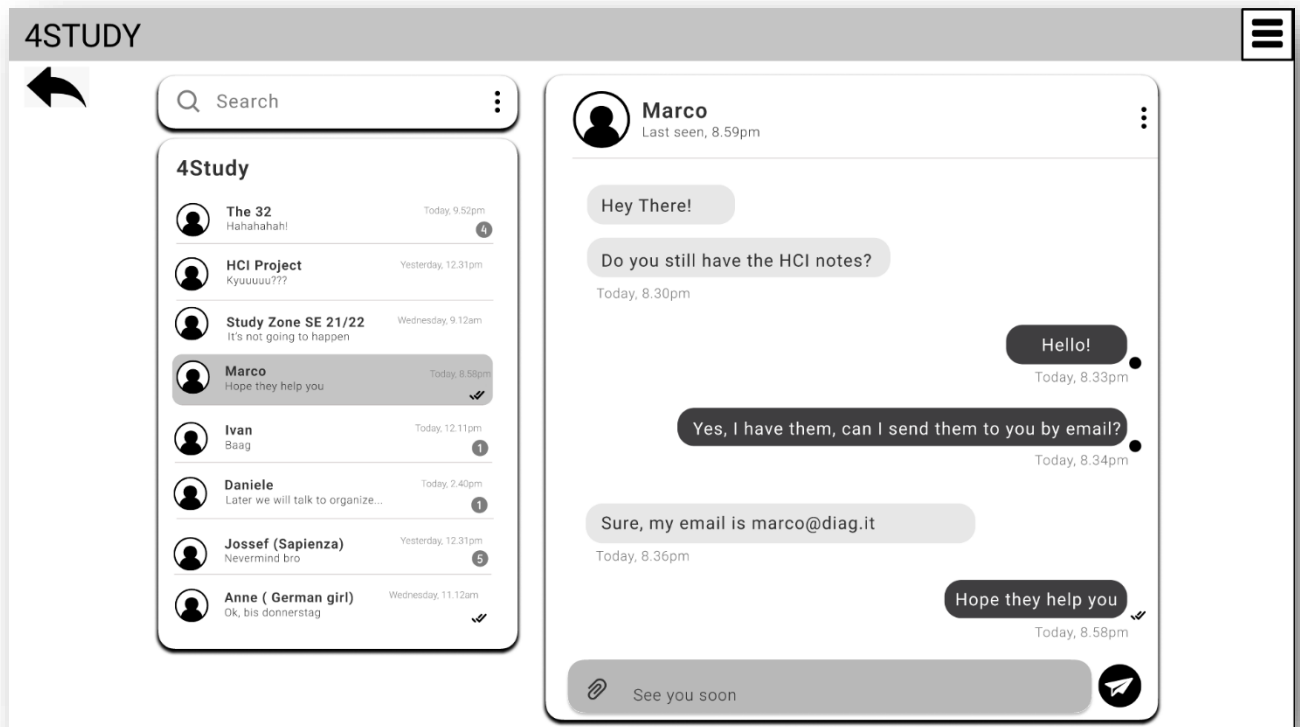
On the top right of the chat, the user can interact with the 'three dots' icon by having the possibility to **delete** the chat, in particular a **popup** will appear on screen, asking for a confirmation from the user. Below report the related mockup:
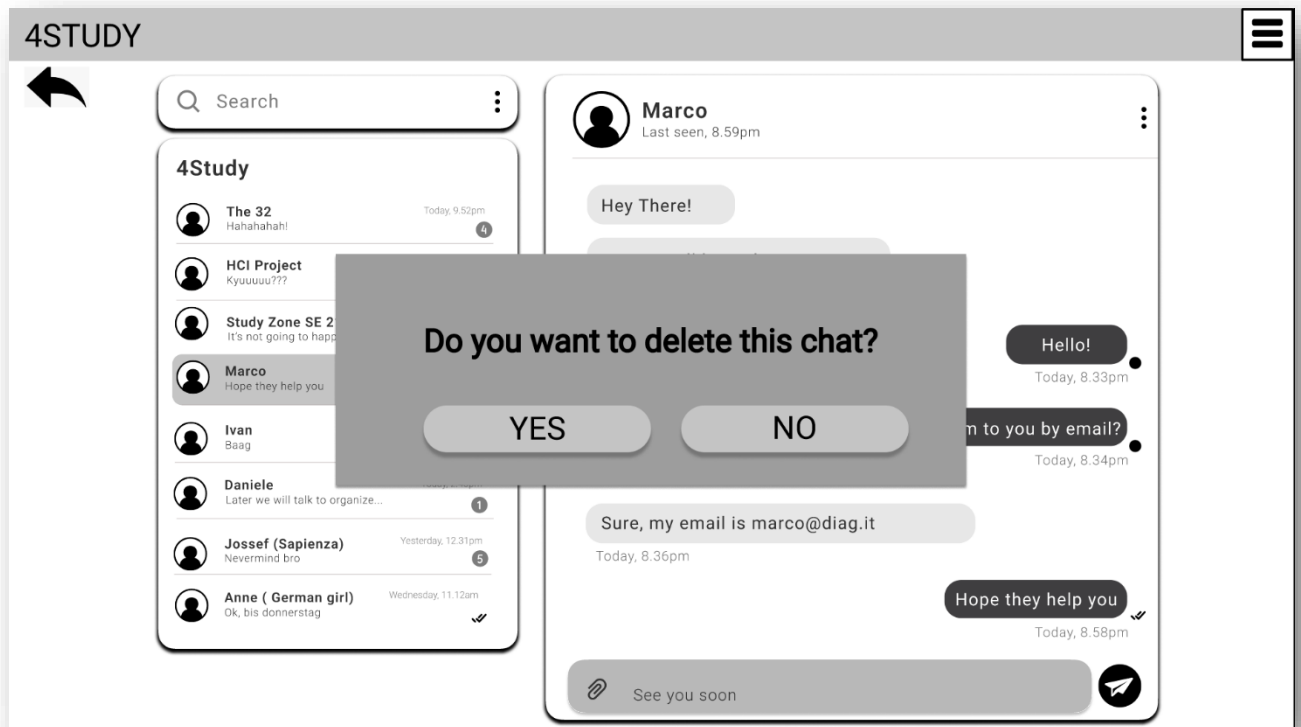
By clicking on the search bar, the user will have to possibility to quickly **search** for a **chat**, by inserting the **name of the private chat or of the group** he is looking for. Below we report the related mockup:
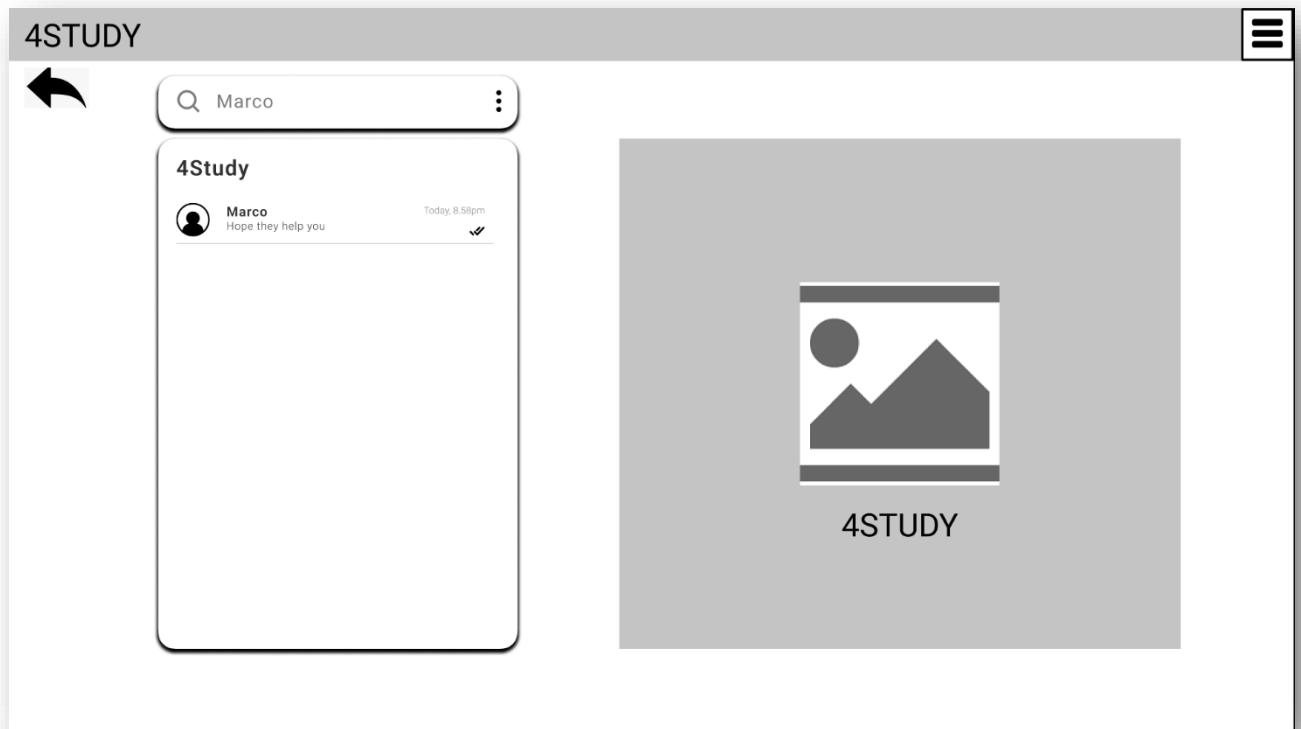
To conclude this section, we want to remember that there are other pages to explore in our system, but we considered them less representative than those reported above, and they are all available for consultation online on the dynamic prototype on Figma.

6. Expert Based Evaluation

6.1 Cognitive Walkthrough

6.2 Heuristic Evaluation

6.3 Results and Gatherings

Bibliography

[1] Official website of GitHub: https://github.com/

[2] Official website of Google Drive: http://drive.google.com/

[3] Official website of Google Meet: https://meet.google.com/

[4] Official website of Zoom: https://zoom.us/

[5] Official website of Google Form: https://www.google.it/intl/it/forms/about/

[6] Official website of Figma: https://figma.com/

[7] Link to the dynamic Figma prototype:
https://www.figma.com/file/fy0RDLkNnwnfH2ZrYcIGBf/4Study-Project?node-id=0%3A1

[?] Official website of Bootstrap: https://getbootstrap.com/

[?] Official website of Vue.js: https://vuejs.org/