Introduction

🏆 A2.2 Learning activity

- Documentation of system architecture based on 4+1 model

📘 Instructions

- Based on the documentation provided by the consultant, develop diagrams that can be used to design the 4+1 model-based system architecture for the case study.
- All activity or challenge must be done using the **MarkDown style with .md** extension and the VSCode development environment, or you can use any platform for example **Notion**, and it must be elaborated as a **single page** document, that is to say if the document has images, links or any external document it must be accessed from tags and links, and it must be named with the nomenclature **A2.2_ActivityName_StudentName.pdf.**.
- It is required that the .MD contains a tag of the link to the repository of your document in GITHUB, for example **Link to my GitHub** and at the conclusion of the challenge it should be uploaded to github.
- From the **.md** file export a **.pdf** file that should be uploaded to classroom within its corresponding section, serving as evidence of your delivery, since being the **official** platform here you will receive the qualification of your activity.
- Considering that the .PDF file, which was obtained from the .MD file, both must be identical.
- Your repository, in addition to having a **readme**.md file in its root directory, with information such as student data, work team, subject, career, advisor data, and even logo or images, must have a contents section or index, which are actually links or **links to your .md** documents, *avoid using text* to indicate internal or external links.
- We propose a structure such as this one indicated below, however you can use any other that supports you to organize your repository.

```
| readme.md
| | blog
| | | | Cx.1_ActivityName.md
| | | | Ax.1_ActivityName.md
| | diagrams
| | | docs
| | html
| | img
| | | pdf
```

✏️ Development

1. Develop diagrams for each of the views set out in the 4+1 architecture model.

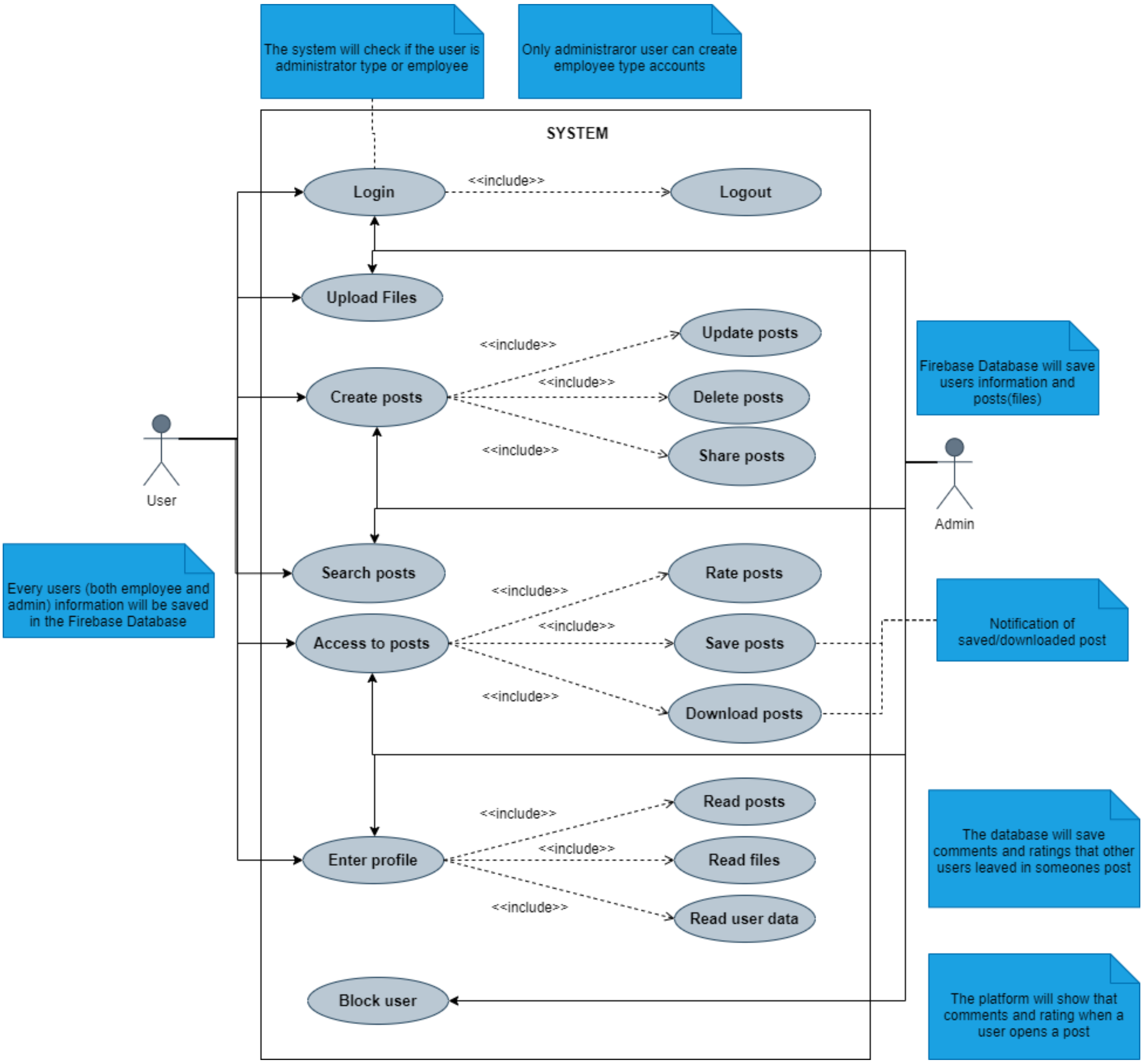
    - ☑ Scenario View: User Case Diagram
    - ☑ Logic View: Class Diagram

  - ☑ Process View: Sequence Diagram
  - ☑ Developer view: Component Diagram
  - ☑ Physical View: Distribution Diagram

2. Each diagram shall contain at least 3 elements within its representation.

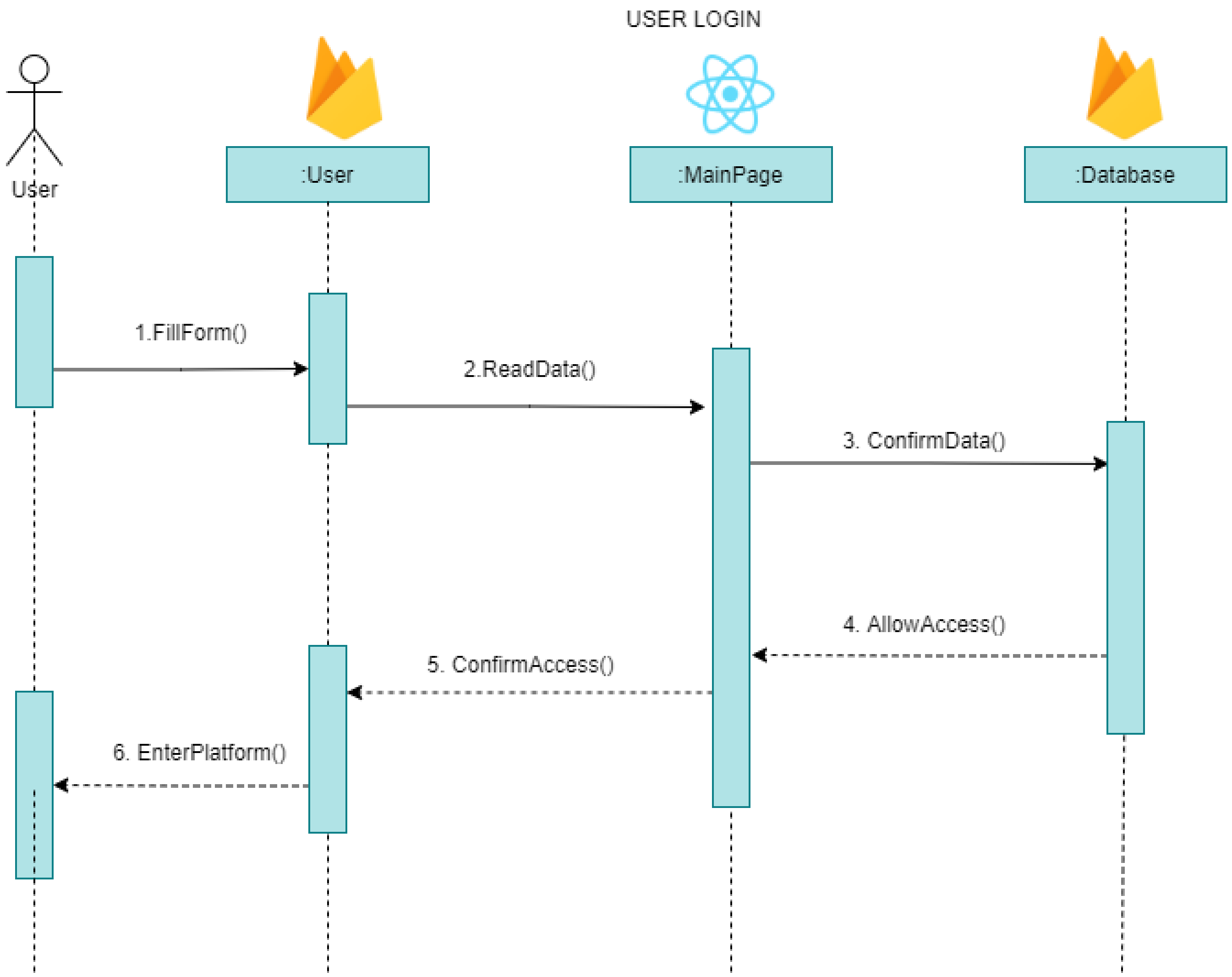
   - ☑ Use cases (Include at least 5 elements of the diagram)

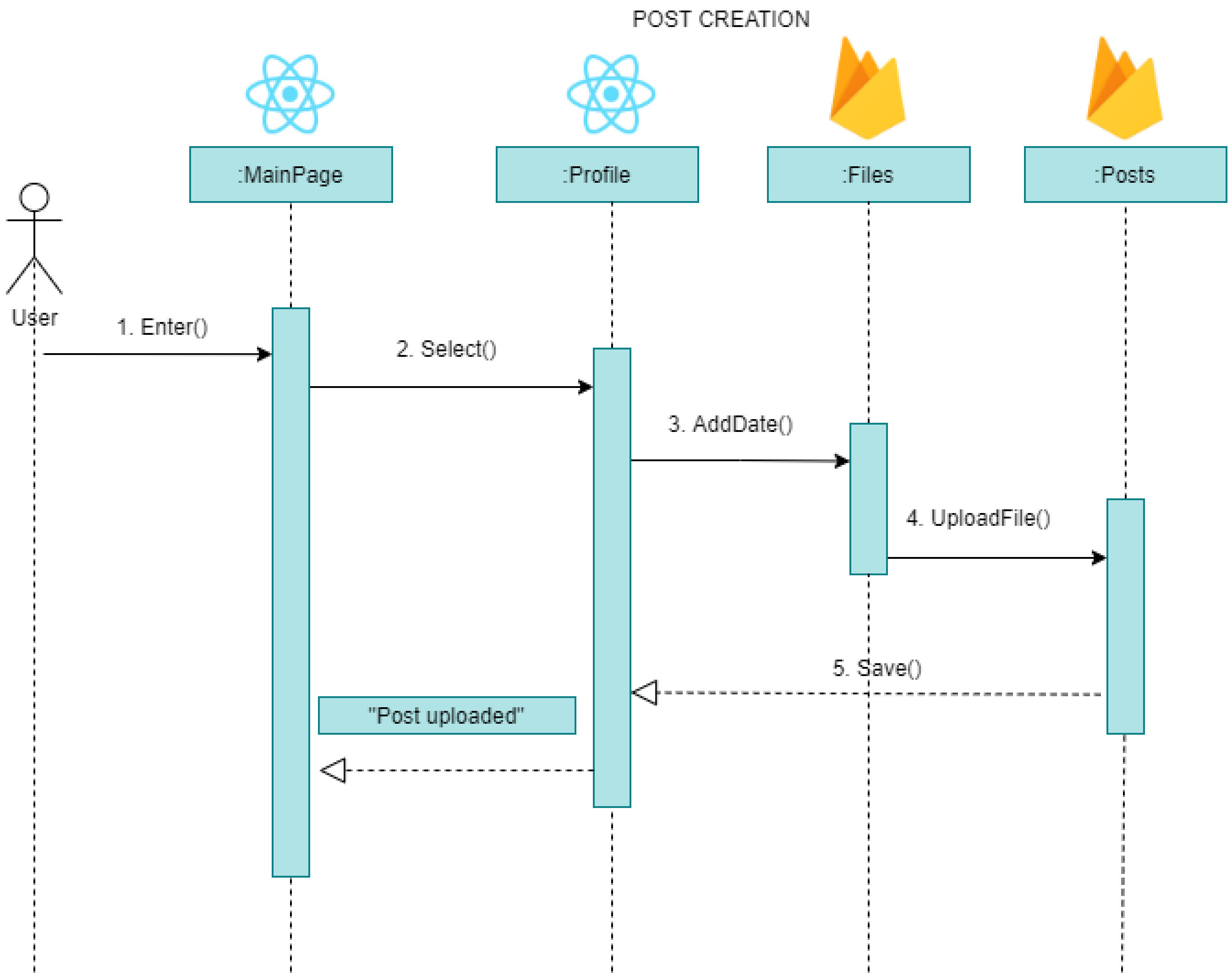
SCENARIO VIEW

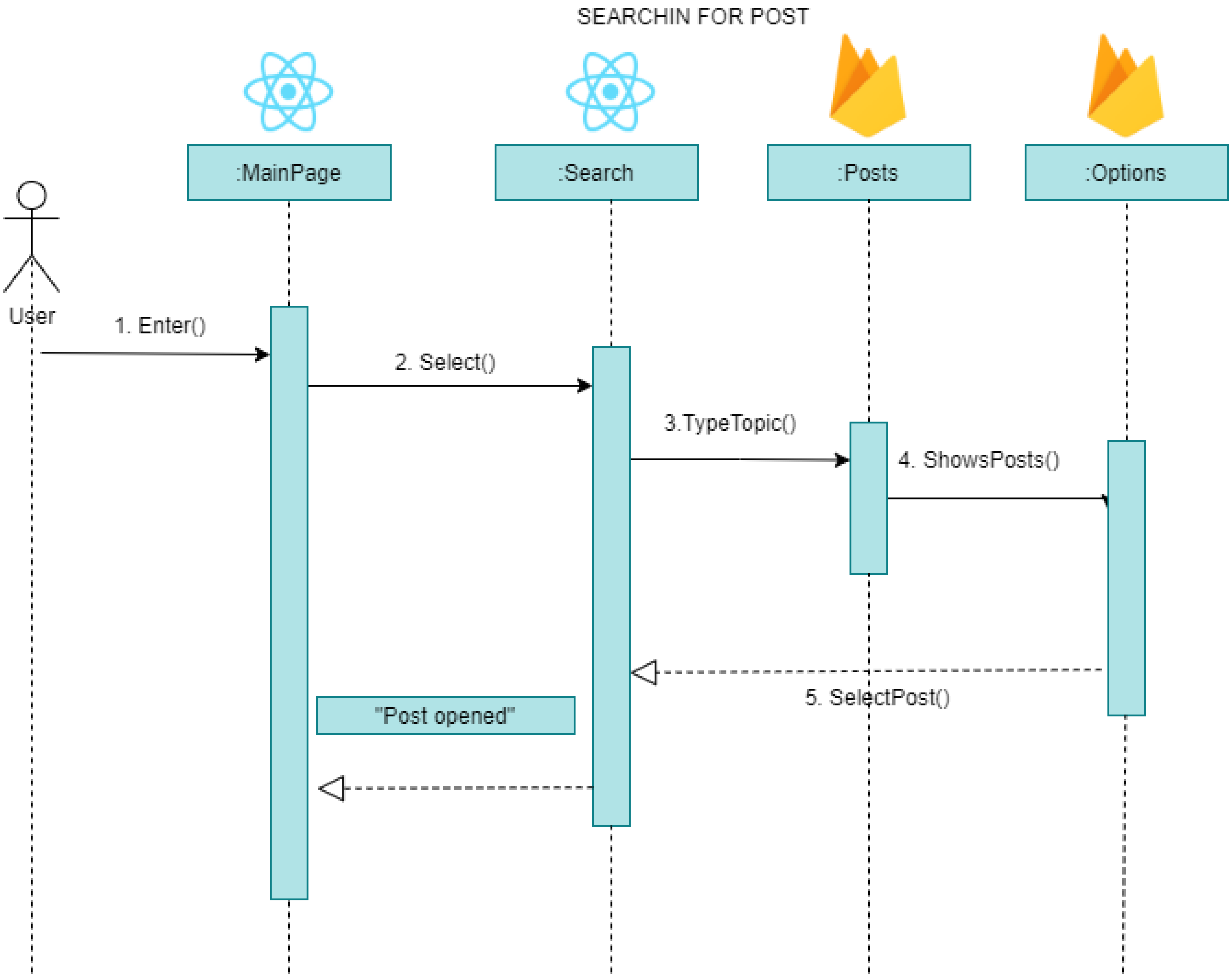

The system will check if the user is administrator type or employee
Only administraror user can create employee type accounts
SYSTEM
Login
<<include>>
Logout
Upload Files
Create posts
<<include>>
Update posts
<<include>>
Delete posts
<<include>>
Share posts
Firebase Database will save users information and posts(files)
User
Admin
Every users (both employee and admin) information will be saved in the Firebase Database
Search posts
Rate posts
<<include>>
Access to posts
<<include>>
Save posts
<<include>>
Download posts
Notification of saved/downloaded post
Read posts
<<include>>
Enter profile
<<include>>
Read files
<<include>>
Read user data
The database will save comments and ratings that other users leaved in someones post
Block user
The platform will show that comments and rating when a user opens a post

- ☑ Sequence Diagram (Include at least 5 diagram elements)

PROCESS VIEW

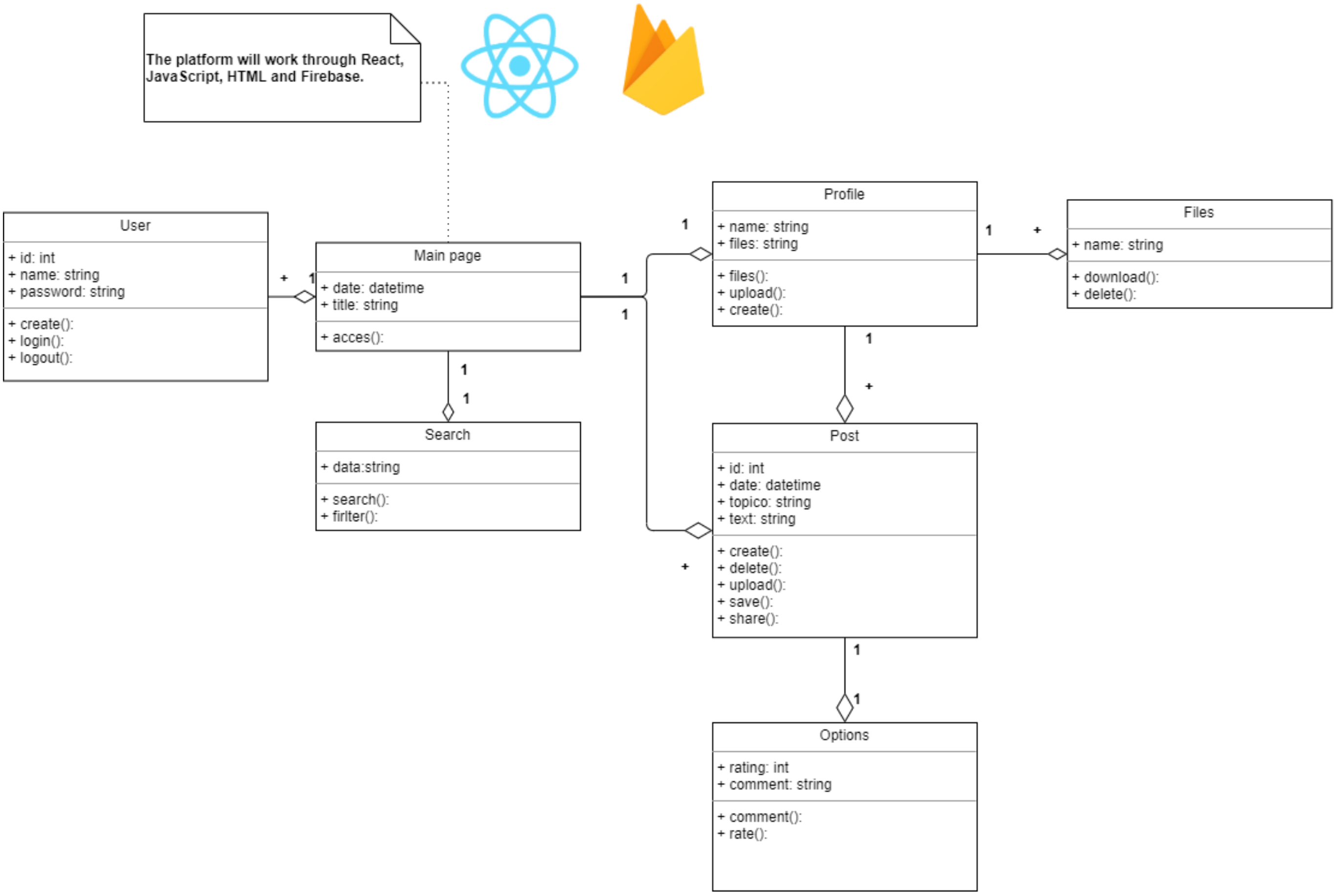

USER LOGIN
User
:User
:MainPage
:Database
1.FillForm()
2.ReadData()
3. ConfirmData()
4. AllowAccess()
5. ConfirmAccess()
6. EnterPlatform()


POST CREATION
:MainPage
:Profile
:Files
:Posts
User
1. Enter()
2. Select()
3. AddDate()
4. UploadFile()
5. Save()
"Post uploaded"



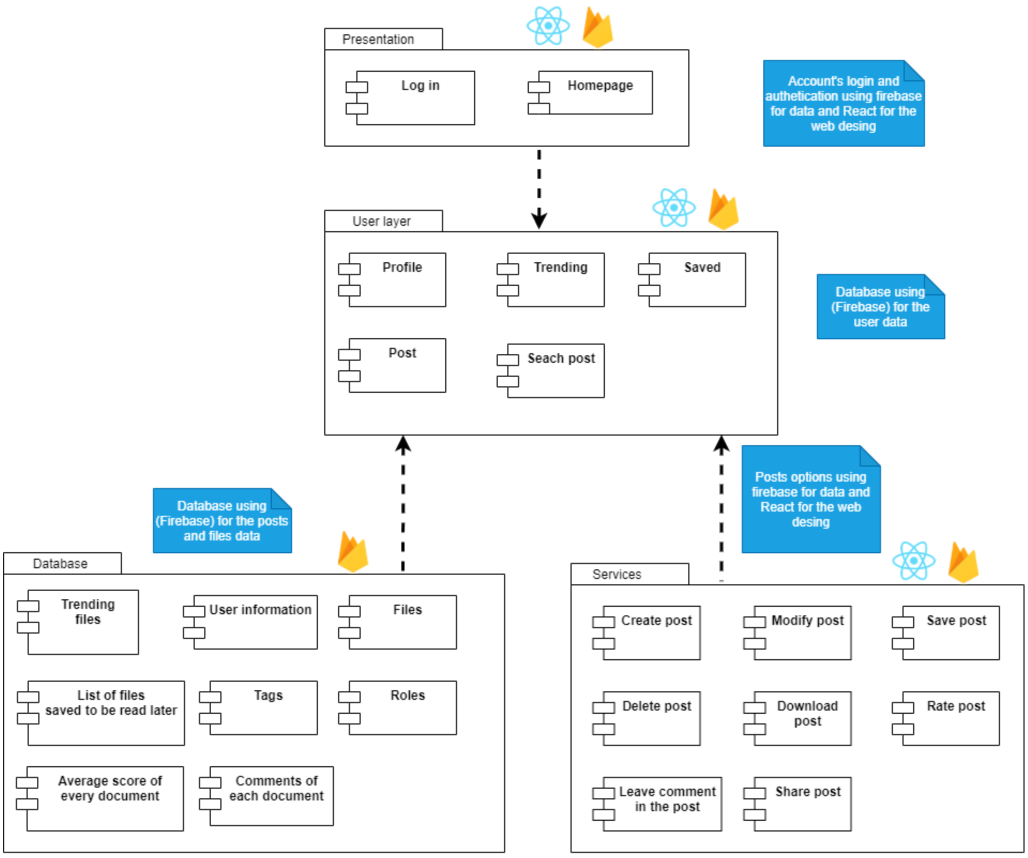
- ☑ Class Diagrams (Include at least 5 diagram elements)

LOGIC VIEW



- ☑ Package diagram containing component diagrams (Include at least 3 - diagram elements)

DEVELOPER VIEW



Account's login and authetication using firebase for data and React for the web desing

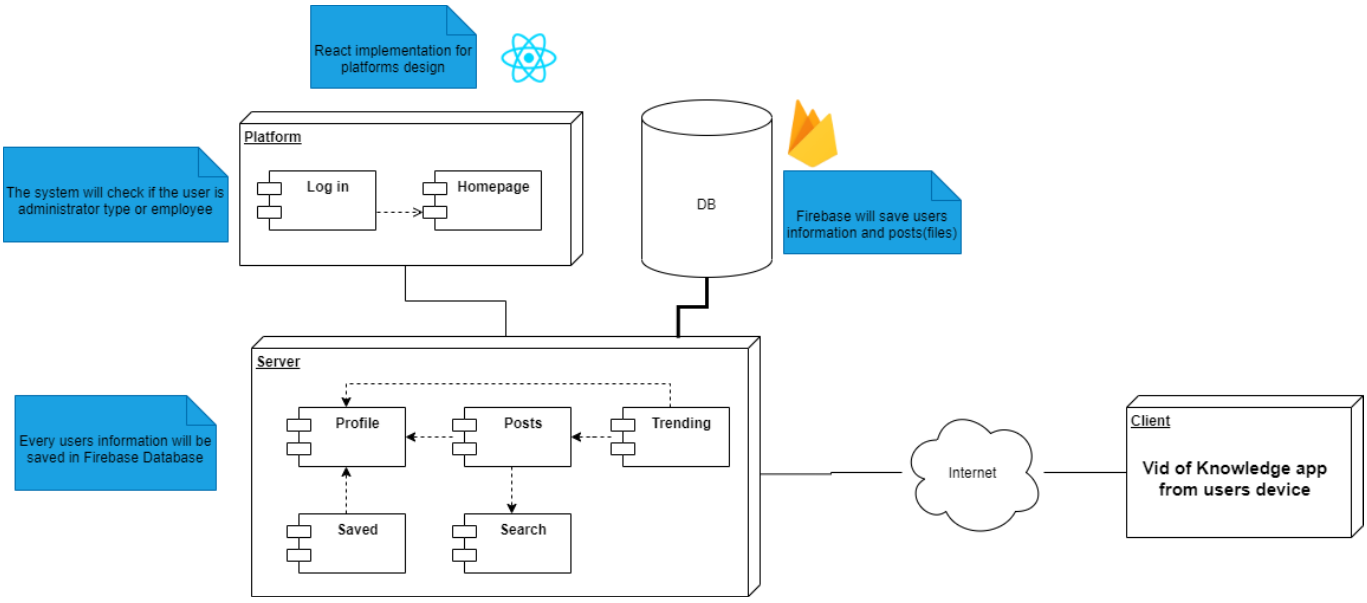
Database using (Firebase) for the user data

Database using (Firebase) for the posts and files data

Posts options using firebase for data and React for the web desing

- ☑ Distribution diagrams (Include at least 5 diagram elements)

PHYSICAL VIEW



React implementation for platforms design

The system will check if the user is administrator type or employee

Firebase will save users information and posts(files)

Every users information will be saved in Firebase Database

3. Indicate by annotations the proprietary technologies to be used, supported by pictures or illustrations depicting them.

Conclusions

- **Bernal Arellano Roberto:** Krutchen's 4+1 Architecture describes the software systems architecture with the help of multiple views that describes, in our case, the plaform from points of view for our target users, developers and interested people. The explanation of our platform becomes more easy to understand thanks to this, all this because here are explanated the process, development, interactions (of users with the platform), the communications of every systems element with another and the way our platfom works.

- **Cisneros Acosta Jose Enrique:** Thanks to the 4 + 1 architecture, we can understand in a different and exact way the diagrams that are made when creating a project and know its function.

- **Pimienta Castillo Kevin Aryam Cristopher:** The Kruchten 4 + 1 model allowed us to see a simplified description of the system through its four views and the stage view.

- **Sandoval Salazar Esmeralda:** In conclusion, the use of 4 +1 architecture are the same diagrams, but based on the use of multiple concurrent views, which are used to illustrate from any selected point or view the four views the important thing about this architecture is to resolve the errors and find the right solutions.

💣 Rubric

Criteria	Description	Scoring
Instructions	Are each of the items listed in the Instructions section met?	10
Development	Did you respond to each of the points requested in the development of the activity?	60
Demonstration	Is the student presented during the explanation of the functionality of the activity?	20
Conclusions	Is a personal opinion of the activity included for each of the team members?	10

🏫 **Github repository**