



TOUDOOM

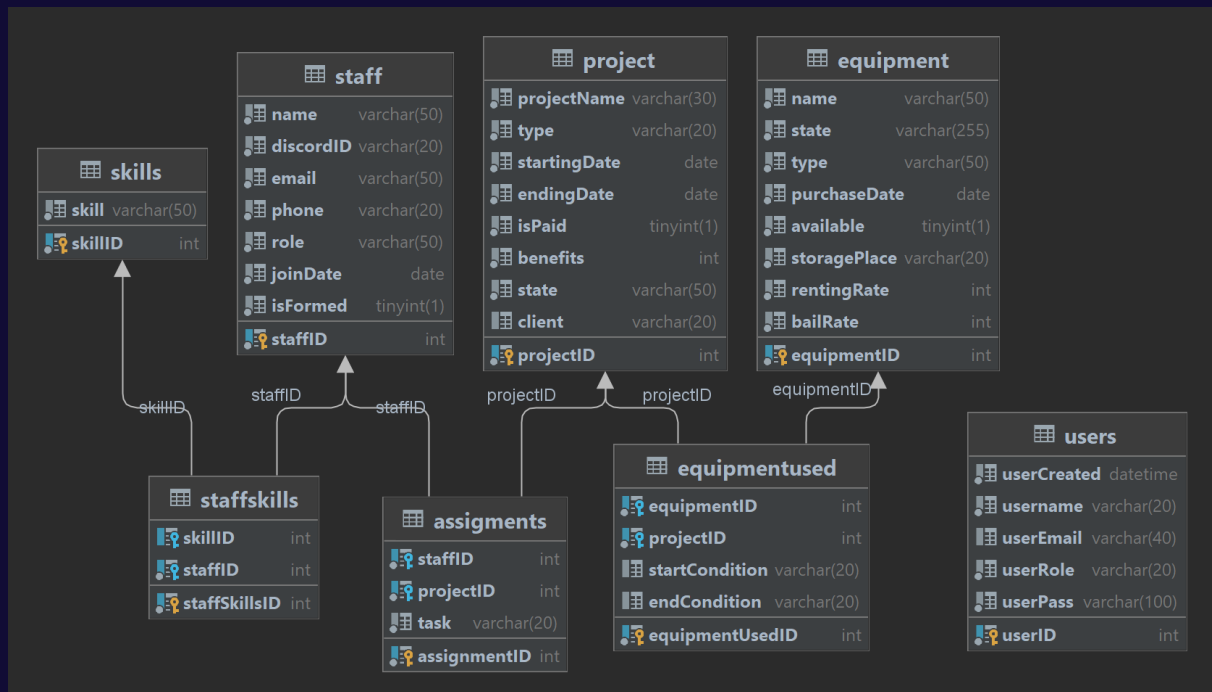
WEB/UML PROJECT
DIAGRAMS

MADE BY CEDRIC YOGANATHAN,
JULIEN STARCK & ROMAIN DUBOIS

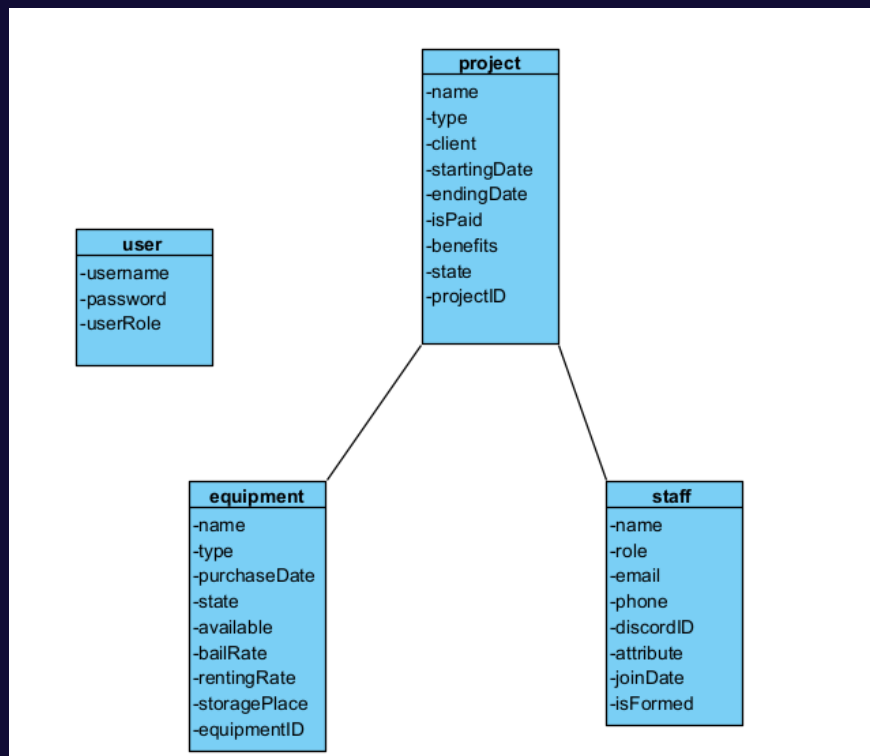
VIDEO PRODUCTION COMPANY MANAGEMENT WEBAPP

For the WEB/UML Project we will do a WebApp for the management in intern of a video production company called Toudoom. With this app we will be able to manage the inventory of the company, the staff, projects and clients. In the inventory section we will be able to track all of the equipment, It's condition, it's availability, the storage place, check the bail rate, and is renting rate in case of a renting for another company. In the Project section we will be able to check the progression of the project, assigns people to a project, assigns the equipment needed. In the team section you will be able to manage your team find their contacts, check their skills, check if they are formed, find the projects that they assigned on.

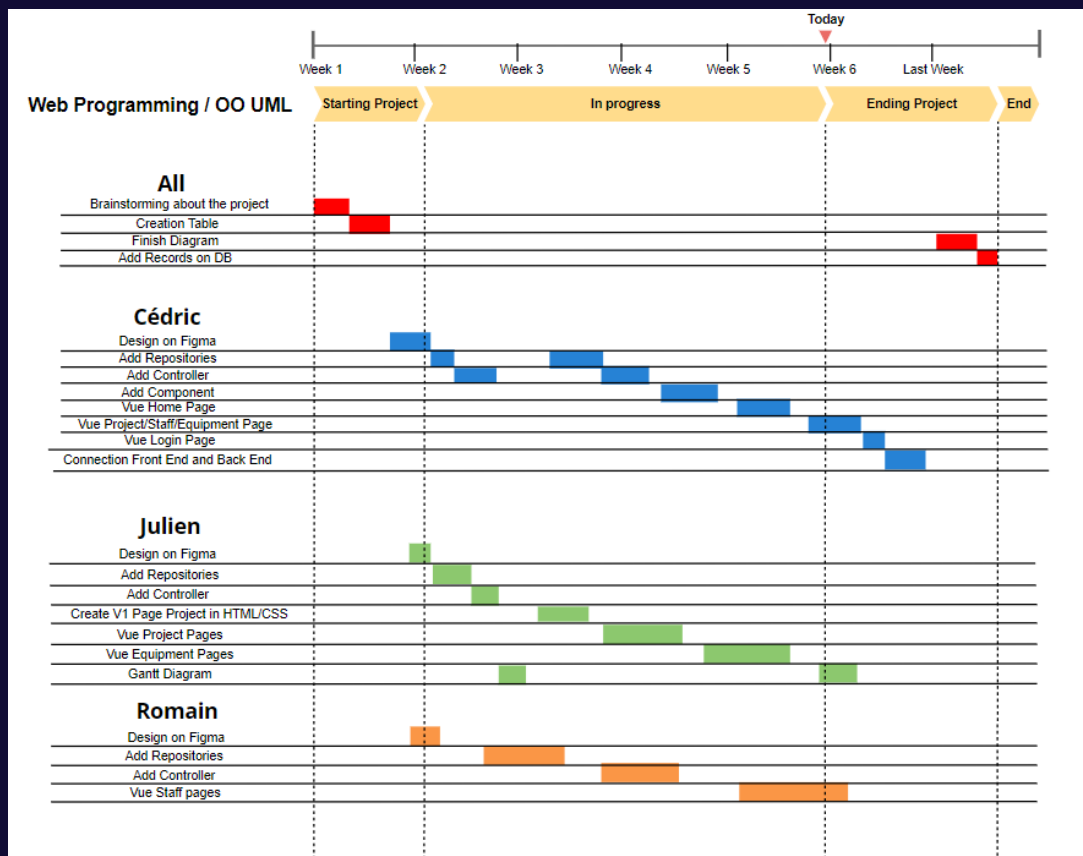
THERE IS THE DIAGRAM OF THE DATABASE OF OUR APP WHO SHOW ALL TABLE (WITH THEIR PARAMETERS) WE WILL USE AND CONNECTION THEY HAVE EACH OTHER (MADE BY ALL)



THERE IS THE CLASS DIAGRAM OF THE VUE FRONT



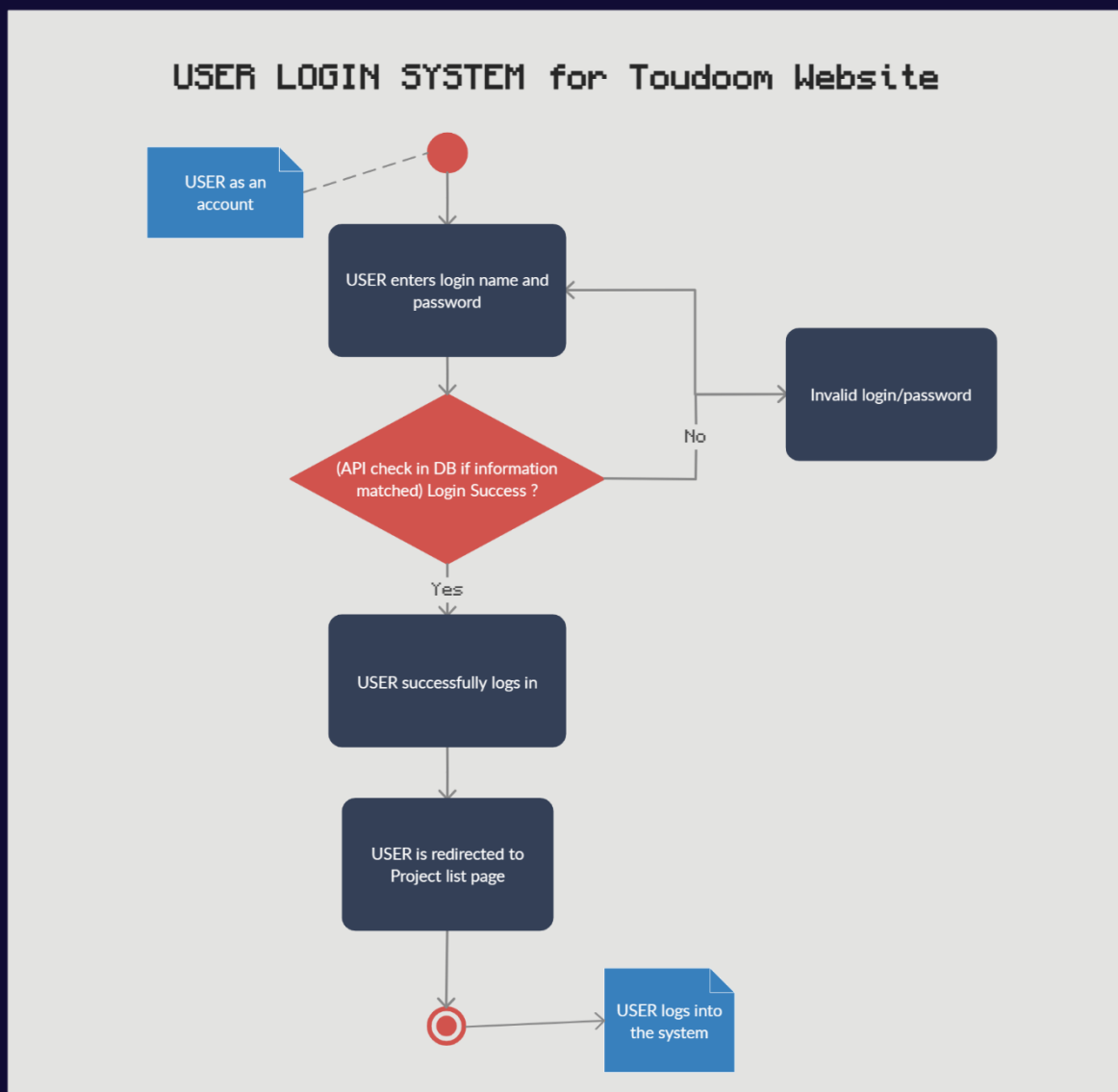
THE FINAL GANTT DIAGRAM REPRESENTING REPARTITION OF TASKS TO FINISH THE PROJECT ON TIME, WE CAN SEE ALL GLOBAL TASK WE DO AND WHO DOING IT AND AT WHAT POINT IN THE PROJECT IT WAS DONE




**DIAGRAMS MADE
BY
JULIEN STARCK**

DIAGRAMS MADE BY JULIEN STARCK

THIS ACTIVITY DIAGRAM REPRESENTS THE WORKING OF THE AUTHENTICATION SYSTEM OF AN USER WHO HAS ALREADY AN ACCOUNT ON TOODOOM. FIRST HE ENTERS HIS MAIL AND PASSWORD, SEND THIS INFORMATION TO THE API THAT WILL CHECK IN DATABASE IF INFORMATION ARE CORRECT AND GIVE ACCESS IF ITS CORRECT



THIS IS THE WIREFRAMES FOR "PROJECT" PAGES. WE CAN SEE THE SIDEBAR WITH THREE TYPES OF PAGES AND ON LEFT THE MAIN PAGE THAT SHOW THE PURPOSE OF THIS PAGE. THEY ARE ACCESSIBLE ONLY LOGGED IN USER

Logout

MENU

PROJECTS

STAFFS

EQUIPMENT

PROJECT

EDIT

INFORMATION

Name
Type
Client
Start Date
End Date
Benefits
State

STAFF

Lastname

Firstname

Rôle

EQUIPMENT

ID


Name

Type

Condition

Storage place

Available

Logout

MENU

PROJECTS

STAFFS

EQUIPMENT

PROJECTS

+ ADD

ID

Name


Type

Start date

End date

Client

State

Logout

MENU

PROJECTS

STAFFS

EQUIPMENT

PROJECT

SAVE

INFORMATION

Name
Type
Client
Start Date
End Date
Benefits
State

STAFF

Nom

Prénom

Rôle

EQUIPMENT

ID

Name

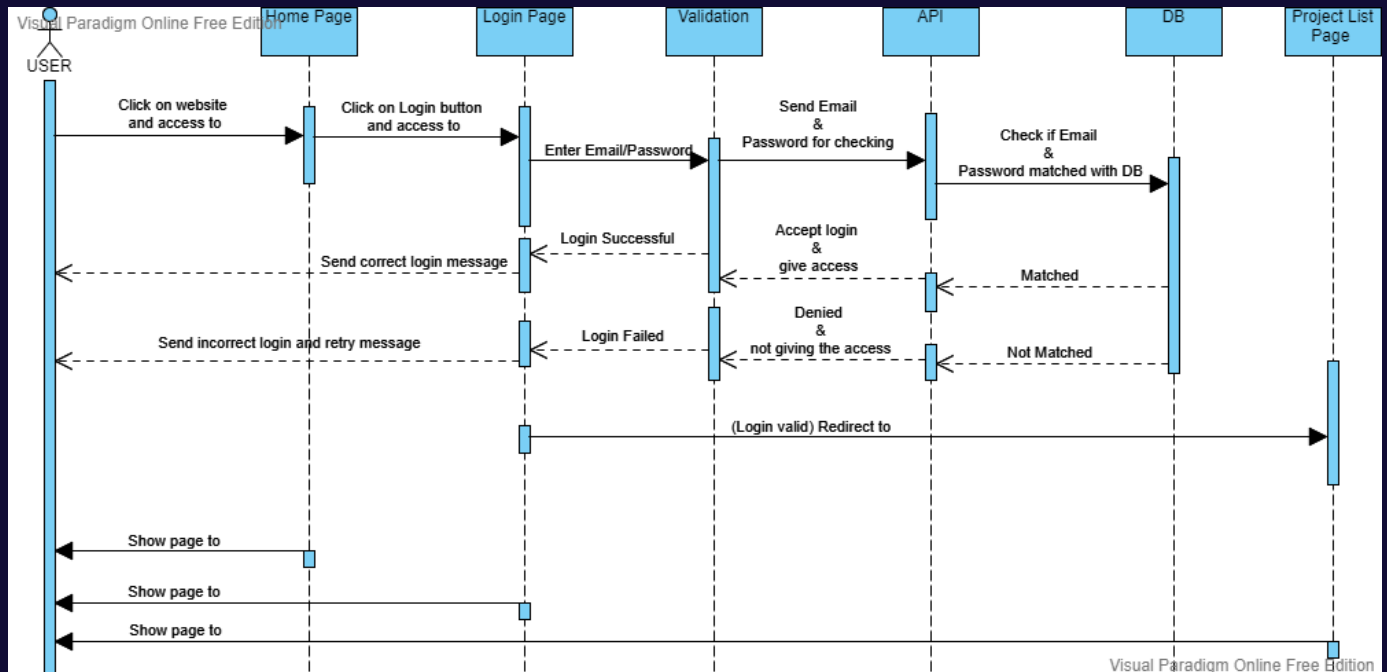
Type

Condition

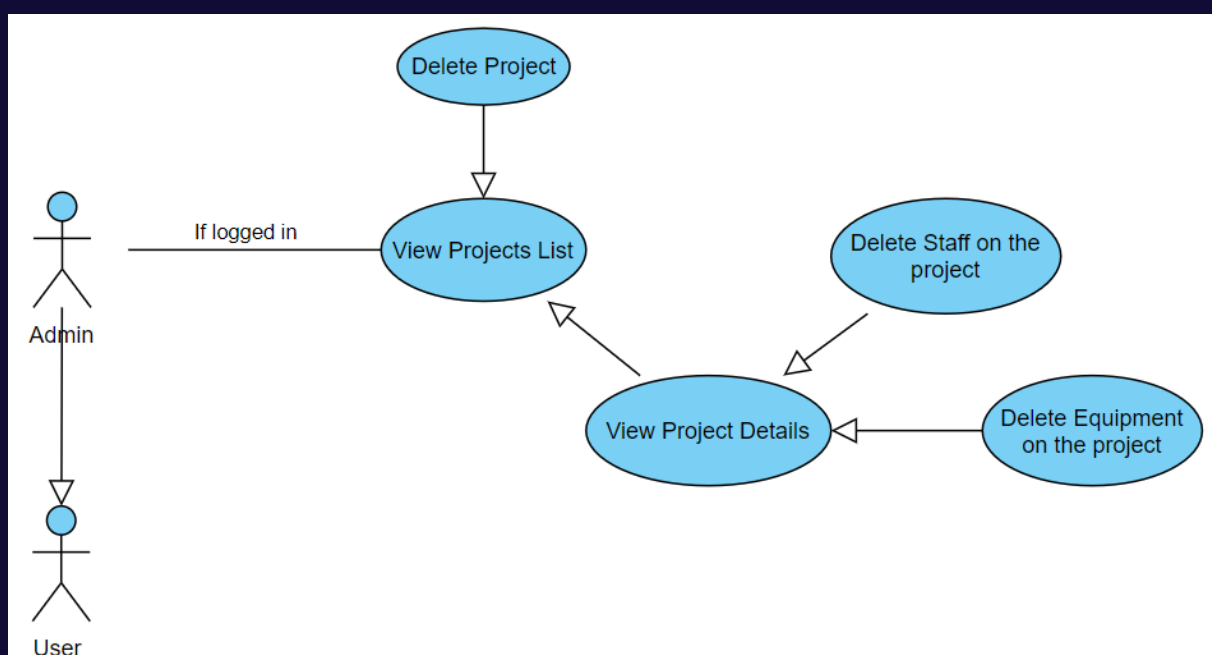
Storage place

Available

THIS IS THE SEQUENCE DIAGRAM FOR AN USER WHO WANT TO LOG IN HIS ACCOUNT AND ACCESS ON PROJECT PAGES (THAT ONLY MEMBER OF TOUDOOM CAN ACCESS), WE CAN SEE ALL ACTION MADE BY USER AND BY THE WEBAPP TO ACCESS IT



THIS IS THE USE CASE DIAGRAM WHO SHOW WHAT ADMIN CAN DO WHEN HE IS LOG AND ON PROJECT LIST PAGE. LIKE DELETE A PROJECT, UNASSIGN MEMBER ON A PROJECT TO WORK ON IT OR EQUIPMENT, HE ALSO CAN DO EVERYTHING ALLOWED FOR USER. LIKE CREATE PROJECT, ADD STAFF, EQUIPMENT, AND UPDATE DETAILS ON IT.

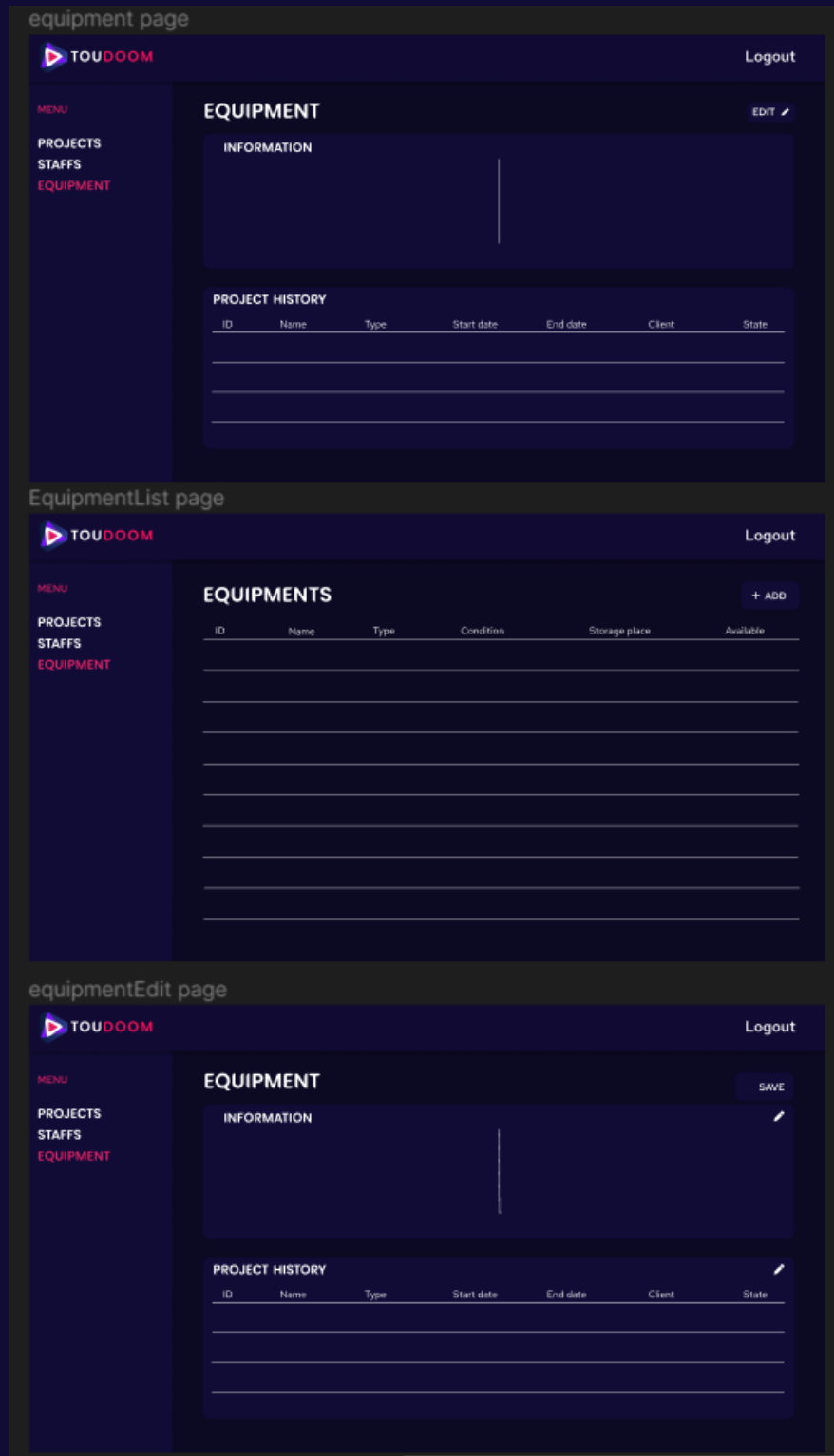


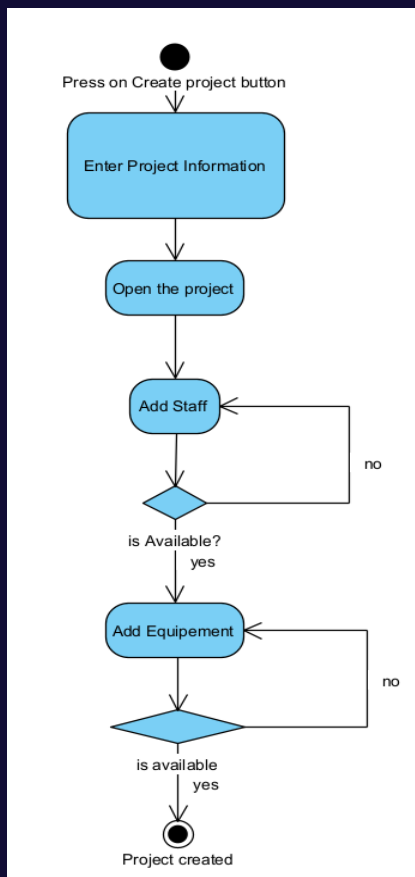
**DIAGRAMS MADE
BY
CEDRIC YOGANATHAN**

THIS IS THE WIREFRAMES OF THE LANDING PAGE ACCESSIBLE BY EVERYONE. THIS PAGE PROVIDES ALL NECESSARY INFORMATION ABOUT TOUDOOM ASSOCIATION.



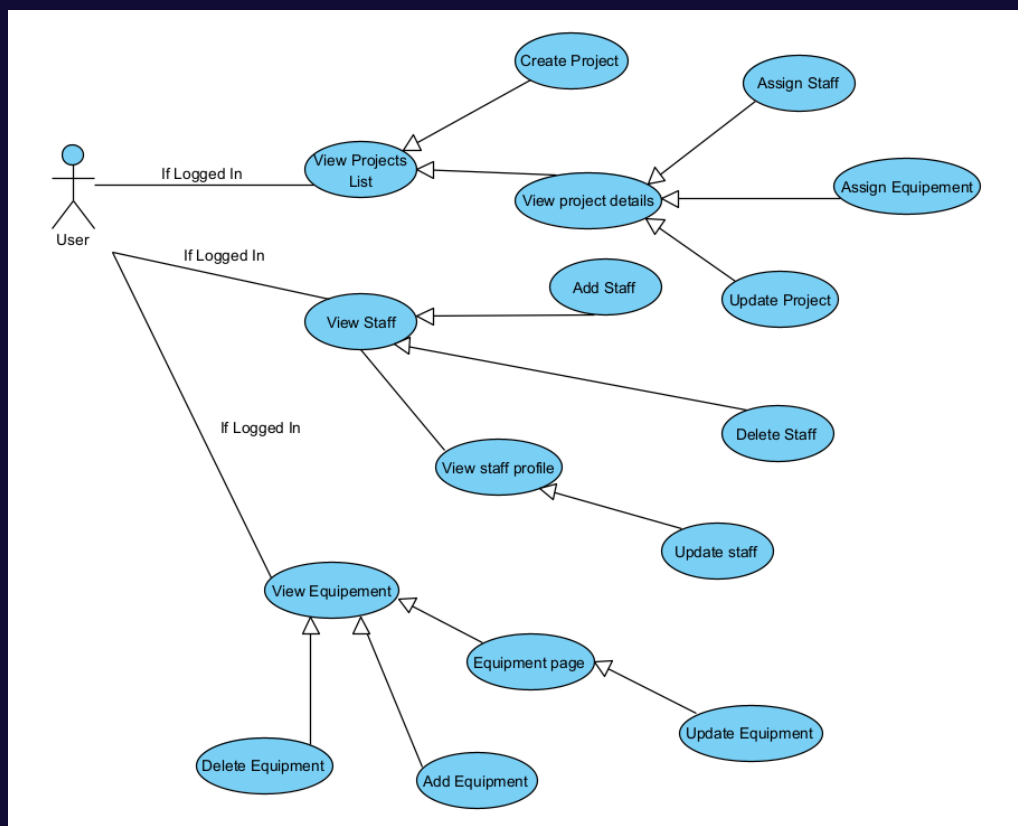
THIS IS THE WIREFRAMES OF THE EQUIPMENT PAGES ACCESSIBLE BY ONLY LOGGED IN USERS. THE FIRST PAGE SHOWS ALL NECESSARY INFORMATION ABOUT EQUIPMENT AND PROJECT THAT WE USE IT BEFORE, THE SECOND PAGE SHOW THE FULL LIST OF EQUIPMENT AND THE LAST ONE SHOW THE PLACE TO EDIT A SELECTED EQUIPMENT



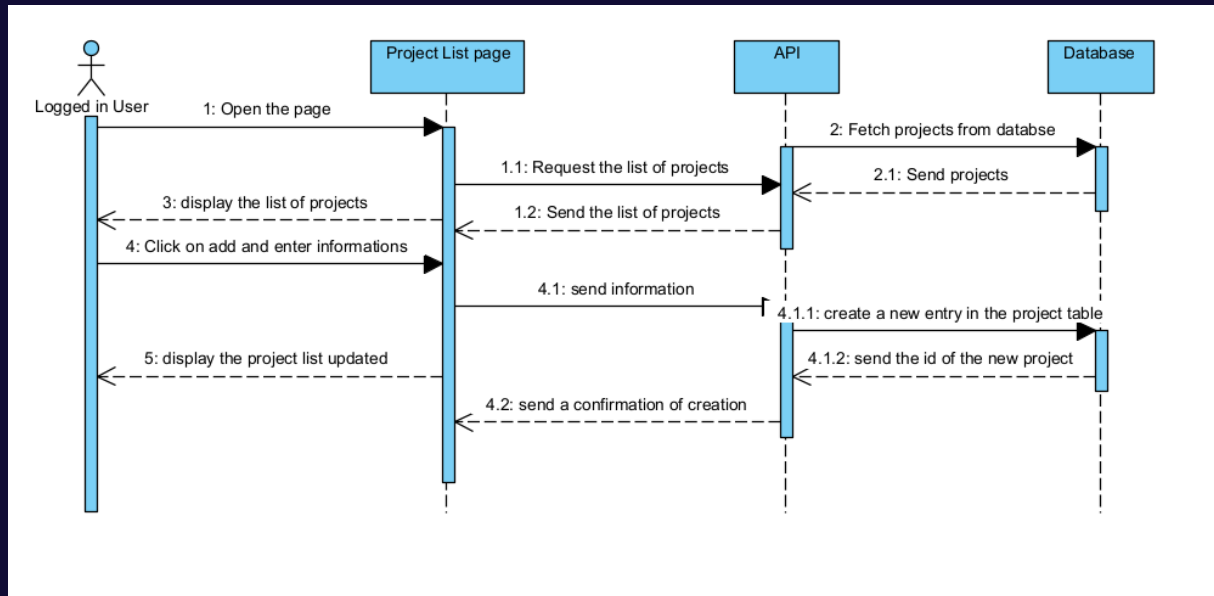


**ACTIVITY DIAGRAM OF
THE PROJECT CREATION PROCESS**

USER USE CASE REPRESENTING ALL THE POSSIBLE ACTION FOR A USER. HE CAN CREATE A PROJECT ASSIGN AND UPDATE ALL STAFF AND EQUIPMENT ON A PROJECT, HE CAN ALSO ADD AND DELETE A STAFF IN STAFF PAGES, AND FINALLY, MANAGE ALL EQUIPMENT IN EQUIPMENT PAGES.

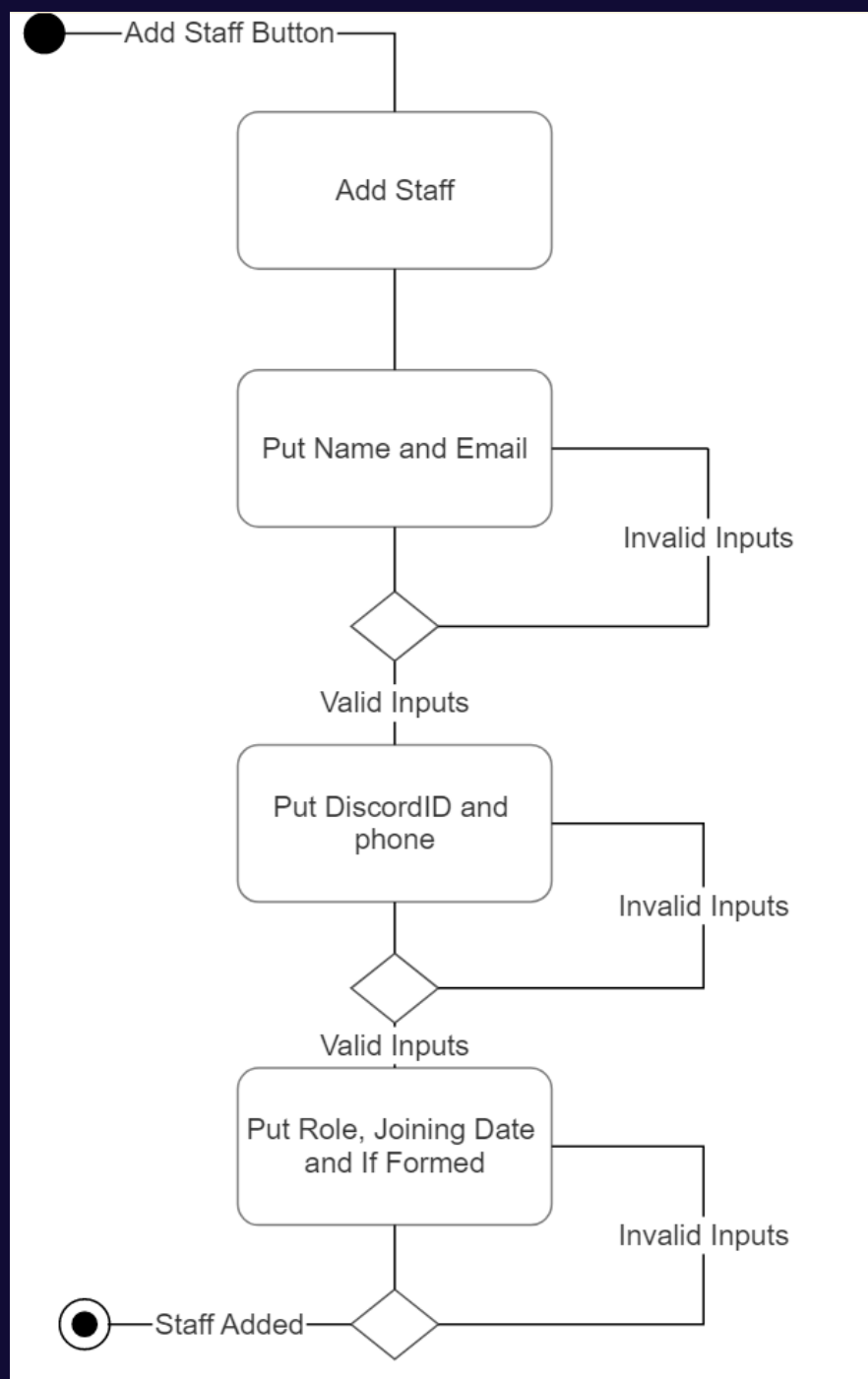


SEQUENCE DIAGRAM REPRESENTING THE PROJECT CREATION PROCESS AND THE PROJECT LIST DISPLAY, WE CAN SEE REQUEST AND RESPONSE WITH API THAT MAKE A LINK OF FRONT END AND DATABASE

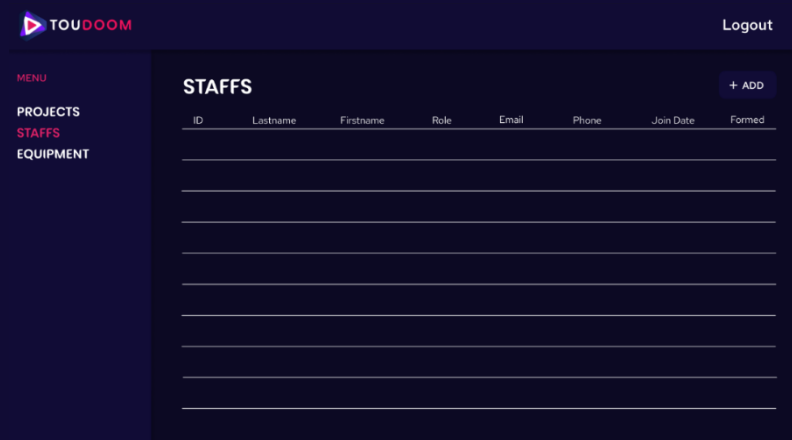
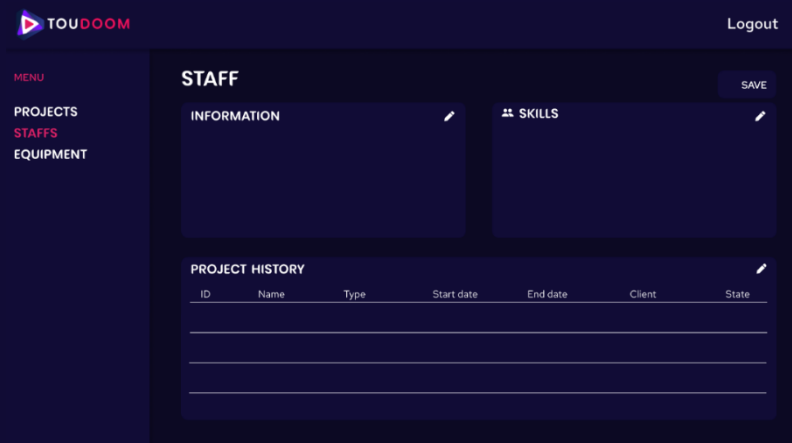
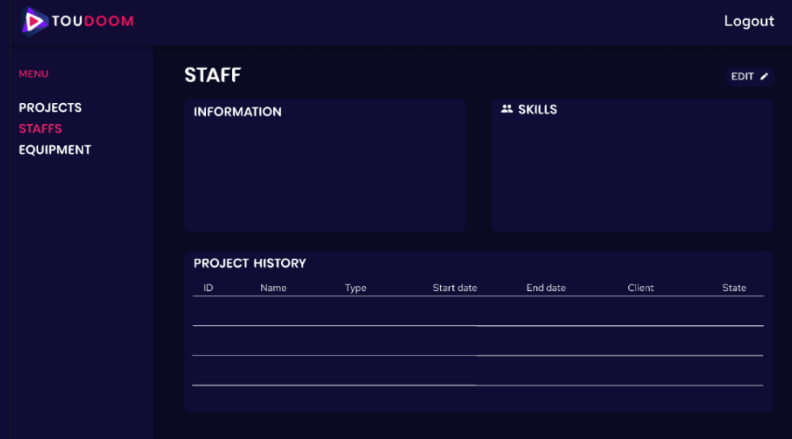


**DIAGRAMS MADE
BY
ROMAIN DUBOIS**

FIRST, FOR THE ACTIVITY DIAGRAM, WE HAVE A REPRESENTATION ON HOW THE STAFF ADDING PROCESSES WORKS. AS IT'S A RATHER SIMPLE TASK, WHEN YOU HIT THE ADD BUTTON ON THE STAFF LIST PAGE, YOU JUST NEED TO GIVE ALL THE DATA, AND IF THEY ARE ON THE RIGHT FORMAT, WHEN YOU VALID IT, IT ENTERS IN YOUR DATABASE.



WIREFRAMES ARE SIMILAR TO OTHERS BECAUSE OF THE DESIRE TO HAVE A UNIFIED DESIGN FROM ONE PAGE TO ANOTHER. IN A FORM THAT DOES NOT DESTABILIZE FROM ONE PAGE TO ANOTHER, THEY SHOW LIST OF STAFF AND IF WE CLICK ON A SPECIFIED STAFF MEMBER, IT SHOWS STAFF INFORMATION, SKILLS, AND PROJECT HE DO, IT ALSO POSSIBLE TO EDIT THE STAFF MEMBER.



```

sequenceDiagram
    actor User
    participant Login as Login Page
    participant Project as Project List Page
    participant StaffList as Staff List Page
    participant StaffDetails as Staff Details Page
    participant StaffEdit as Staff Edit Page
    participant API
    participant DB as DataBase

    User->>Login: Access the site
    Login->>Project: Pushed to if Login as Admin or Staff Valid
    Project->>StaffList: Chose to go to
    StaffList->>API: Send Identifiers
    API->>DB: Check if identifiers match
    DB-->>API: Responds
    API-->>StaffList: Get response
    StaffList->>API: Send Request for Staff Data
    API->>DB: Access to Database
    DB-->>API: Get Staff Data
    API-->>StaffList: Recieving All Staff Data
    StaffList->>API: Send Staff Data to Add
    API->>DB: Adding Data in Staff
    DB-->>API: Get Staff Data
    API-->>StaffList: Recieving All Staff Data
    StaffList->>API: Send Staff Member's ID to Delete (Only if Logged as Admin)
    API->>DB: Delete Targeted Staff Member
    DB-->>API: Get Staff Data
    API-->>StaffList: Recieving All Staff Data
    StaffList->>StaffDetails: Chose to go to
    StaffDetails->>API: Send Request for Staff Member's Data
    API->>DB: Access to Database
    DB-->>API: Get Staff Member's Data
    API-->>StaffDetails: Recieving Staff Member's Data
    StaffDetails->>StaffEdit: Chose to go to
    StaffEdit->>API: Send modification for Staff Member's Data
    API->>DB: Editing Staff Member's Data
    DB-->>API: Get Staff Member's Data
    API-->>StaffEdit: Recieving Staff Member's Data
    StaffEdit->>StaffList: Want to go back to
    StaffList->>User: Show Page
    StaffDetails->>User: Show Page
    StaffEdit->>User: Show Page
  
```

```
graph TD
    User((User)) -- "If logged as Staff or Admin" --> ViewStaffList((View Staff List))
    DeleteStaffMember((Delete Staff Member)) -- "If logged as Admin" --> ViewStaffList
    AddStaffMember((Add Staff Member)) --> ViewStaffList
    EditingStaffMember((Editing Staff Member)) --> SeeOneStaffMember((See One Staff Member))
    SeeOneStaffMember --> ViewStaffList
```

The diagram illustrates the following use cases and relationships:

- User** (Actor) interacts with **View Staff List** (Use Case) with the condition "If logged as Staff or Admin".
- Delete Staff Member** (Use Case) interacts with **View Staff List** (Use Case) with the condition "If logged as Admin".
- Add Staff Member** (Use Case) interacts with **View Staff List** (Use Case).
- Editing Staff Member** (Use Case) interacts with **See One Staff Member** (Use Case).
- See One Staff Member** (Use Case) interacts with **View Staff List** (Use Case).