Programming 2B
POE Part 1

St10275468 Module: PROG6212

Thomas Knox Group: 3

Table of Contents

1. Documentation

Design choices:

Users-Experience(UX) / User-Interaction design choices:

Feedback:

I have designed the solution for the Contract Monthly Claim System to provide feedback to the user where necessary. This is an important feature as it gives the user guidance or clarity on how the system works. The feedback will also prompt the user if they have left out any necessary input fields when they are working with information. (ux4sight, 2024)

Combo-box:

I have decided to make use of combo-boxes throughout the design of the system. Combo-boxes make the user-experience and interaction with the system much more pleasant while reducing the chance of data validation errors. This is because there will be preset values in the combo-boxes which the user will just have to choose from allowing for more efficient and smooth interaction.

Filter feature:

The application will have a “Filter by status” feature allowing the Academic Managers and Program Coordinators to sort through all the claims submitted to the system. This will allow the claims to be shown based on their status which can be “Pending”, “Approved” or “Rejected”. This will enhance the experience for the administrators of the system.

Color palette choice:

Teal(#FF008080) – Primary color

White(#FFFFFFFF) – Accent color

The color palette I have chosen to use in the Contract Monthly Claim System I have designed consists of Teal and White. I have specifically chosen to use Teal as the background color as it gives the user a sense of welcoming and ease while using the application. When the color White is used with Teal it not only looks pleasing to the eye, but it also makes important features stand out extremely well. This is why I have used White for all the headings and naming of the important features like buttons. (ux4sight, 2024) (UXPin, 2023)

Database structure:

Lecturers table:

This table holds all the data and information about the Lecturers who are wanting to submit their claims.

- lecturer
  - lecturer_ID – Primary key (INTEGER), auto incremental
  - name – VARCHAR(100)
  - surname – VARCHAR(100)
  - email – VARCHAR(100)
  - password – VARCHAR(100)

Administrators table:

This table holds all the data and information about the Administrators who can review, approve or reject the claims submitted by the Lectures. The Administrators of the system include the Program Coordinators and the Academic Managers.

- admin
  - admin_ID – Primary key (INTEGER), auto incremental
  - name – VARCHAR(100)
  - surname – VARCHAR(100)
  - email – VARCHAR(100)
  - role – VARCHAR(100)
  - password – VARCHAR(100)

Claims table:

The Claims table will store all the data and information on each claim that the lecturer submits to the system.

- claim
  - claim_ID - Primary key (INTEGER), auto incremental
  - lecturer_ID – Foreign key referenced from lecturer(lecturer_ID)
  - dateSubmitted – DATE
  - claimMonth – VARCHAR(100)
  - amount – DECIMAL
  - claimStatus – VARCHAR(100)

Documents table:

This table stores the supporting documents data that correspond specific claims using the claim_ID as a Foreign Key.

- document
  - document_ID - Primary key (INTEGER), auto incremental
  - claim_ID – Foreign Key referenced from claim(claim_ID)
  - lecturer_ID Foreign Key referenced from lecturer(lecturer_ID)
  - path – VARCHAR(255)

Reviews table:

This table stores all the data and information on the reviews made by the administrators.

- claimReview
  - claimReview_ID - Primary key (INTEGER), auto incremental
  - claim_ID – Foreign key referenced from claim(claim_ID)
  - admin_ID - Foreign key referenced from admin(admin_ID)
  - date - DATE
  - reviewStatus – VARCHAR(100)

Table relationships:

Lecturer → Claims (One lecturer can submit multiple claims, however a claim can only be submitted by one lecturer) One-to-Many relationship.

Administrators → Reviews (One Administrator can make many reviews, however a review can only be made by one Administrator) One-to-Many relationship.

Reviews → Claims (One Claim can be reviewed many times, however a review can only have one claim) One-to-Many relationship.

Claims → Documents (One claim can have many documents, however a document can only belong to one claim) One-to-Many relationship.

Lecturer → Documents (One lecturer can upload many documents, however a document can only be uploaded by one lecturer) One-to-Many relationship.

GUI Layout:

Home page:

- Navigation bar:
    - Located at the top of the page.
    - Includes headings stating how the navigation works.
    - Color = White with Teal headings.
- The Login feature:
    - Located between the left side and the center of the page.
    - Includes elements like labels, input boxes and a button which are aligned left.
- The Create account:
    - Located between the right side and center of the page.
    - Includes elements like labels, input boxes, a button and a combo-box which are all aligned to the left.

Submit Claims page:

- Navigation bar:
    - Located at the top of the page.
    - Has headings stating how the navigation works.
    - Color = White with Teal headings.
- Submit Claims feature:
    - Located between the left side and the center of the page.
    - Includes elements like labels, input boxes, a button, a drop-down menu and a choose file element which are all aligned to the left.
- View Claims feature:
    - Located between the right side and center of the page.
    - Only includes a table and label elements which are aligned to the left.

Manage Claims page:

- Navigation bar:
    - Located at the top of the page.
    - Includes headings stating how the navigation works.
    - Color = White with Teal headings.
- Filter Status feature:
    - Located just below the page heading and aligned vertically between the left and center of the page.
    - Includes a label and a comb-box element.
- Submitted Claims feature:
    - Located just below the Filter Status feature and aligned in the center of the page. It is a big feature so goes onto both sides of the page.
    - Includes elements like a table, labels and buttons which are all aligned left.

Assumptions & Constraints:

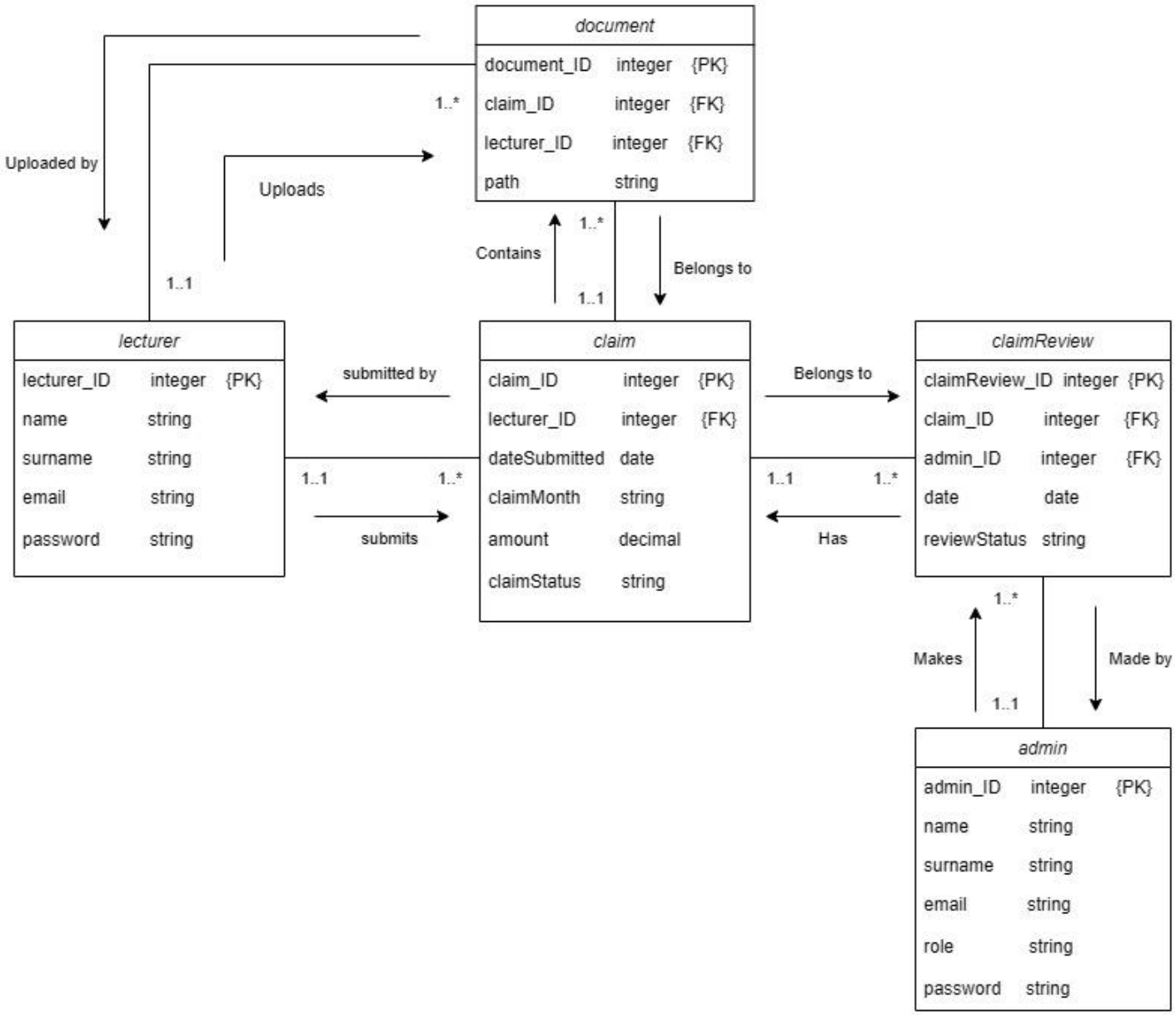
Constraints:

- Only the administrators(Program Coordinators and Academic Managers) can have access to the “Manage Claims” page. This means only the administrators can accept and reject the claims.
- This means that only the lecturers will have access to the “Submit Claims” page as this will have no use to the administrators.
- The system must allow for documentation supporting claims to be uploaded.

Assumptions:

- I am assuming that none of the administrators of the system are also lecturers.
- I assume that the administrators and lecturers have a basic understanding of how to use an application.

2. UML Class Diagram:



3. Project Plan

Planning Phase: (6 days)

During this phase I aim to understand the problem well and describe the requirements for the solution.

1. Understand the case study in detail. (1 day)
2. Identify stakeholders and any requirements from the case study. (1 day)
3. Interview the stakeholders to get any further requirements needed. (3 days)
4. Create a list of all the requirements and specifications for the system and sort them from the most important to the least important requirements. (1days)

(Team Kissflow, 2024)

Design Phase: (15 days)

During this phase I aim to design how the prototype will look and how it will function based on the requirements collected in the planning phase.

1. Create a color scheme for the prototype. (1 day)
2. Design the flow of the prototype. (2 days)
3. Design the layout of the home page user interface (2 days)
4. Design how the login feature will look and function.(1 day)
5. Design how the create account feature will look and function.(1 day)
6. Design the layout of the submit claims user interface (2 days)
7. Design how the submit claims feature will look and function. (1 day)
8. Design how the view claims feature will look and function. (1 day)
9. Design the layout of the manage claims user interface (2 days)
10. Design how the review claims feature will look and function. (1 day)
11. Design the database structure and create a UML class diagram. (1 day)

(Team Kissflow, 2024)

Develop Phase: (17 days)

1. Develop the database based on the database structure designed. (1 day)
2. Develop the home page GUI and add functionality to the features
    - Develop the navigation bar that will be used throughout the system. (1 day)
    - Develop the home page GUI. (2 days)
    - Add functionality to the create account feature. (2 days)
    - Add functionality to the login feature (2 days)
3. Develop the Submit claims GUI and add functionality to the features
    - Develop the submit claims GUI. (2 days)
    - Add functionality to the submit claims feature. (2 days)
    - Add functionality to the view claims feature (2 days)
4. Develop the manage claims GUI and add functionality to the features.
    - Develop the manage claims GUI (1 day)
    - Add functionality to the review claims feature. (1 day)
5. Connect the database to the system. (1 day)

(Team Kissflow, 2024)

Testing Phase: (5 days)

During this phase we will test the prototype and look for any errors or bugs that will need to be fixed.

1. Create and run appropriate unit tests. (2 days)
2. Allow users to test the prototype and give feedback. (2 days)
3. If any errors are to occur, fix the errors. (1 day)

(Team Kissflow, 2024)

4. GUI IU

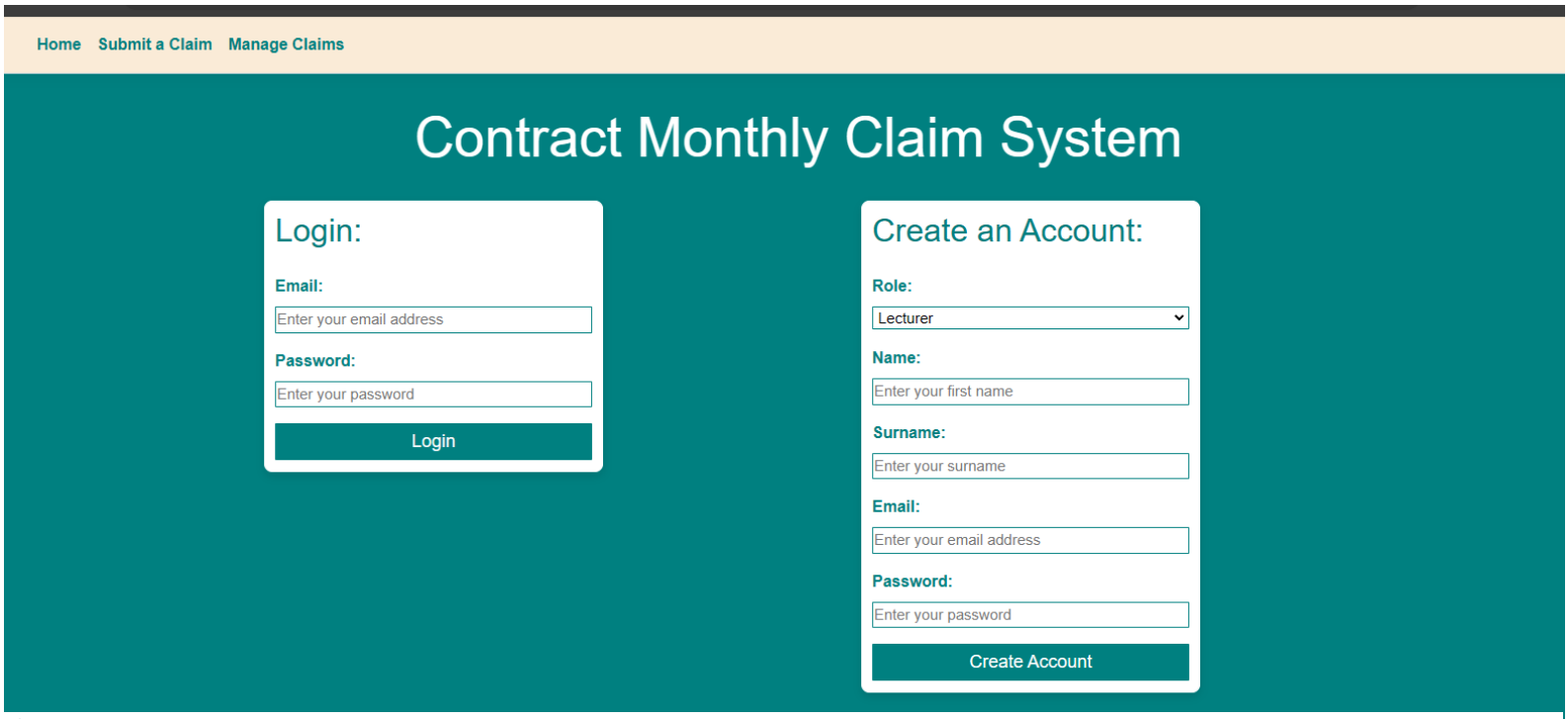


Figure 1: Home page

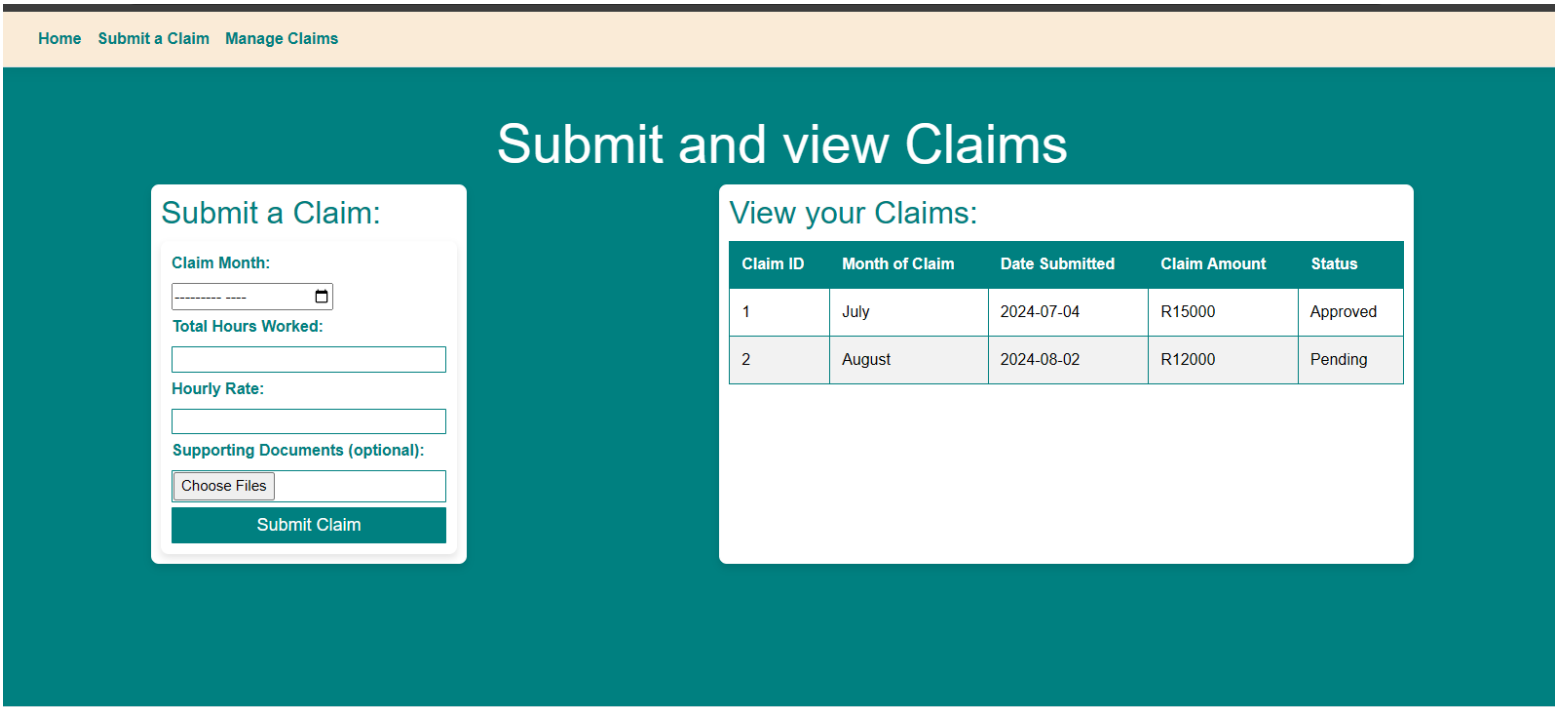


Figure 2: Submit a claim page

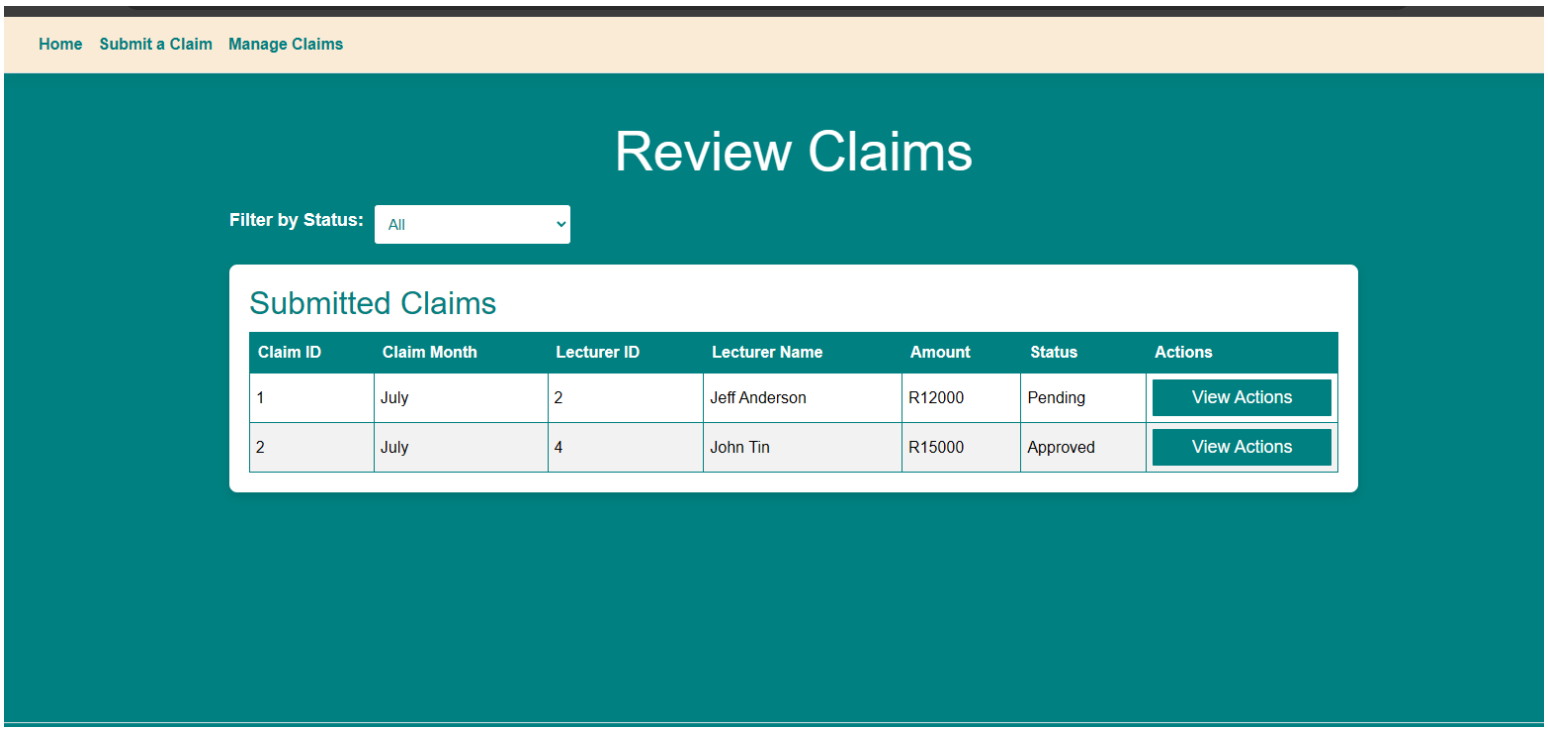


Figure 3: Review Claims page

(OpenAI.2024, 2024)

GitHub Link:

https://github.com/st10275468/st10275468_PROG6212_POE_ThomasK_gr03

References

Team Kissflow. (2024, September 3). *Application Development Lifecycle: Phases, Steps and Process Explained*. Retrieved from Kissflow: https://kissflow.com/application-development/application-development-lifecycle/

ux4sight. (2024, September 9). *Mobile Application Design: Key Considerations For Mobile Apps*. Retrieved from UX4Sight: https://ux4sight.com/blog/9-factors-to-consider-when-creating-mobile-app-designs

UXPin. (2023, February 16). *10 Examples of Design Decisions and How to Make them*. Retrieved from Studio by UXpin: https://www.uxpin.com/studio/blog/design-decisions/

OpenAI.2024. Chat-GPT(Version 3.5).[Large language model]. Available at: https://chat.openai.com/[Accessed: 5 September 2024].