W1 PRACTICE

*MOBILE UX - **CASE STUDY***

In this practice, you will step into the shoes of a UX designer and critically evaluate a mobile application.

Usability and **design** are key factors that influence how users interact with an app, and understanding these aspects is essential for creating effective and enjoyable digital experiences.

Learning objectives

- ✓ Recognize inefficiencies or potential **usability issues** in **user flows**
- ✓ Evaluate a mobile application based on **Nielsen's 10 Usability Heuristics**
- ✓ Critically analyze the **app's design** strengths and weaknesses
- ✓ Identify **UX pain point**

How to submit?

- ✓ Attach your **report t**o the MS Team assignment and **turn it in**

Some references about UX/UI!

USABILITY THEOERY

Jacob 10 heuristics
Don't Make Me Think Book
Introduction To Usability
https://www.kickassux.com/ux-library/ux-process

CASE STUDY EXAMPLES

Talabat Case Study
ContractCar Case Study

STEP 1 –SELECT THE **PRODUCT**

Select and install an existing mobile application.

WARNING: Choose an application related to 1 of following business types:

- **Heath Appointment booking**
- **Educational & University Systems**
- **Transportation**

App Name	**Microsoft Teams**
Business Type	*Select the relevant option:* ☐ Heath Appointment booking ☐ **Educational & University Systems** ☐ Transportation

STEP2 – IDENTIFY THE **USER STORIES**

Identify **2 relevant user stories** - Analyze their frequency, importance

WARNING: The task should be **complex enough** (take multiple steps to complete)

TASK 1

Task	As a user, I want to open the app to quickly top up my phone, but I have to wait for the home screen to load and then search for the button among a cluttered mess of shortcuts and promotional news.
Targeted user	Any bank customer looking to do payments.
Frequency	*Select the relevant option:* ☐ The task is performed **daily or multiple** times per day ☐ **The task is performed weekly or occasionally** ☐ The task is performed **rarely**, perhaps once a month or less
Importance	*Select the relevant option*: ☐ **The task is essential for the primary purpose of the application.** ☐ The task **supports main functions** but not directly critical. ☐ The task enhances experience but is **not necessary** for application

TASK 2

Task	As a user, I need to transfer money to a friend, but I find the "Verify" screen confusing because it looks identical to a final receipt, and I am worried about accidental transfers since there is no final PIN or FaceID check after clicking confirm.
Targeted user	Any bank customer sending money.
Frequency	*Select the relevant option:* ☐ **The task is performed daily or multiple times per day** ☐ The task is performed **weekly** or occasionally ☐ The task is performed **rarely**, perhaps once a month or less
Importance	*Select the relevant option*: ☐ **The task is essential for the primary purpose of the application.** ☐ The task **supports main functions** but not directly critical. ☐ The task enhances experience but is **not necessary** for application

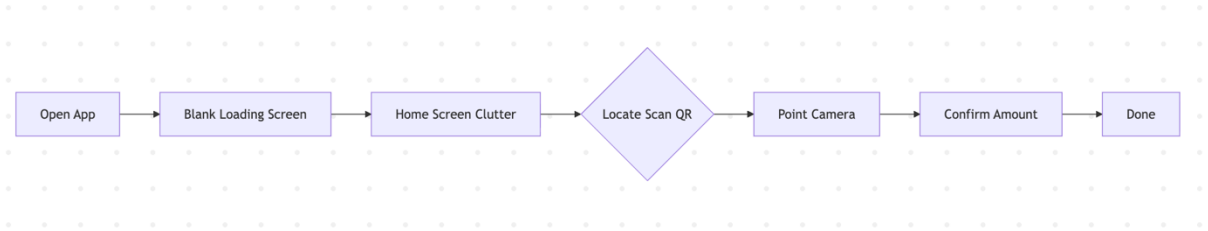
STEP 3 – **USER FLOW** TO PERFORM THE TASKS

Write the user flows (sequence of low-definition wireframes) to complete the chosen tasks.

- How to write a user flow?

USER FLOW 1 - Quick QR Payment (Home Screen Experience)

INSERT FIGMA LINK or SNAPSHOT HERE



Analyze the user flow

- Is the **user flow intuitive** and easy to follow for a first-time user?
- Are there **unnecessary** steps or redundant actions in the process?
- Are there points in the flow where the user **might feel confused** or encounter difficulties?

- **Intuition:** The flow is **not intuitive** for a "quick" task. The home screen contains a "mess" of shortcuts, promotions, and news that forces the user to scan visually for the correct button.

- **Unnecessary Steps:** The promotional news and "Cambodia Tourism" sections act as **visual hurdles**. While not a functional "step," they create a "mental step" where the user must filter out ads before performing a banking task.

- **Confusing Points:** The lack of **skeleton loading UI** on the home screen makes the app feel unresponsive or slow. Users may encounter difficulties knowing if the app is still loading or has frozen.

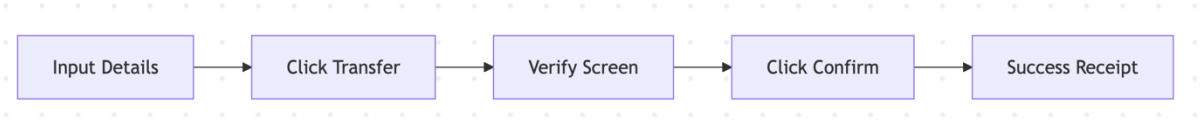
How **many steps** are needed to complete this task?
Do you think the **user flow** is relevant, regarding the **importance and the frequency** of this task?

Approximately **5 to 6 steps**, depending on whether the user needs to scroll past promotions to find the button.

Yes, this is a **highly relevant** flow because QR payments are a **daily necessity** in Cambodia. The current clutter directly interferes with the most frequent and important action a user takes in the app.

USER FLOW 2 - Transfer Confirmation (The "Verify" Trap)

INSERT FIGMA LINK or SNAPSHOT HERE



Analyze the user flow

- Is the **user flow intuitive** and easy to follow for a first-time user?
- Are there **unnecessary** steps or redundant actions in the process?
- Are there points in the flow where the user **might feel confused** or encounter difficulties?

- **Intuition:** This flow is **poorly designed** at the end. The "Verify" screen and the "Success" screen use an almost identical "white card" component.

- **Unnecessary/Redundant Actions:** There is a lack of a **final security checkpoint** (like FaceID or a Swipe). The "Confirm" button is too simple for a high-risk transaction, leading to potential accidental transfers.

- **Confusing Points:** Because the Verify screen looks like a receipt, a user might think the transaction is **already finished** and close the app before actually clicking "Confirm". Alternatively, they might click "Confirm" accidentally without checking details because the screen doesn't "look" like a final warning.

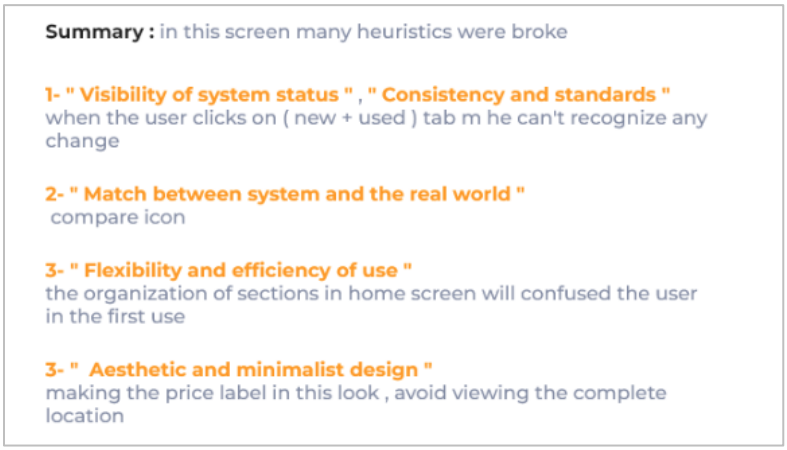
How **many steps** are needed to complete this task?
Do you think the **user flow** is relevant, regarding the **importance and the frequency** of this task?

4 steps (from input to success). While short, the lack of a distinct security step (PIN/Biometric) makes it "too fast" and unsafe.

Yes, this is an **essential task**. While performed less often than scanning QR codes (weekly), the **consequences of an error** (sending money to the wrong person) are very high. The UI fails to prevent these high-stakes errors.

STEP 4 – USABILITY HEURISTICS

Apply recognized usability principles, such as Nielsen's 10 Usability Heuristics, to identify potential issues.



To guide your analysis:

- See below the check list points for each heuristic

- You can also get inspirations on other case studies:
  - Kotak Case Study
  - ContractCar Case Study

WARNING: You need to identify at **least 3 relevant UX/UI issues** and refer to the right Heuristic.

Usability issue	**Home Screen Clutter:** Promotions and news are mixed in with bank tools, making it hard to find shortcuts.
Related Heuristic	minimalist
Proposed remedial (optional)	Remove promotional news from the main dashboard. Place only the top 4 essential shortcuts (Scan, Transfer, Top-up, Balance) on top.

Usability issue	**Confusing Verification:** The "Please verify transaction" screen looks almost exactly like the final "Success" receipt.
Related Heuristic	Consistency and standards
Proposed remedial (optional)	Use distinct colors (e.g., a blue header for "Verify" and a green header for "Success") to clearly signal the status change.

Usability issue	**Accidental Transfer Risk:** Clicking "Confirm" finishes the payment immediately without a PIN or FaceID check.
Related Heuristic	Error prevention
Proposed remedial (optional)	Add a **"Slide to Pay"** button or require a **FaceID/Fingerprint** scan after clicking Confirm to ensure the user intended to send the money.

Usability issue	**Lack of Visual Feedback:** The app remains blank while loading high-resolution ads and news.
Related Heuristic	Visibility of system status
Proposed remedial (optional)	Use **Skeleton Loading UI** (grey placeholders) so the user knows the content is loading and the app hasn't frozen.

STEP 5 – SUMMARIZE

- Highlight the **app's strengths and weaknesses** based on your analysis.

- **Strengths:** The app provides a wide variety of services in one place, and the icons used for banking tasks are standard and easy to recognize.

- **Weaknesses:** The **home screen is messy** because it tries to show ads and tourism news alongside important bank functions. Additionally, the transaction flow is **dangerous** because the "Verify" and "Success" screens are too similar, and there is no final security step like a PIN to prevent accidental transfers. The lack of **skeleton loading UI** also makes the app feel slow or frozen when opening.

- Suggest **specific improvements** to address identified issues.

- **Simplify the Home Screen:** Prioritize the most frequent tasks (Scan QR, Transfer, Top-up) and hide the promotional ads and news in a "Discovery" section.
- **Improve Loading Feedback:** Add grey skeleton boxes while the app loads data so the user knows the app is still working.
- **Visual Status Cues:** Use distinct colors and graphics for different transaction stages (e.g., Blue for "Verify" and Green for "Success") to clearly show when a task is finished.
- **Add a Safety "Speed Bump":** Implement a mandatory biometric scan or a swipe gesture for final confirmations to stop accidental taps from sending money.

APPENDIX – USABILITY HEURISTICS GUIDE (7)

HEURISTIC	**Visibility of System Status**
GOAL	The design should always keep users informed about what is going on
CHECK LIST	Are system status **updates** (e.g., success or error) provided **in real-time**? Is the user **informed of ongoing processes** (e.g., uploads, downloads, or syncing)? Do icons, animations, or visual cues **provide feedback** for user interactions (e.g., button presses)? Is there a clear **loading indicator** for actions like content loading?

HEURISTIC	**Match Between System and the Real World**
GOAL	The design should speak the users' language
CHECK LIST	Are labels, icons, and terminology **familiar** and aligned with the user's expectations? Is the user **familiar** with the terminology used in the design?

HEURISTIC	**User Control and Freedom**
GOAL	Users need a clearly marked way to leave unwanted action
CHECK LIST	Can users easily **undo actions**, such as deleting or modifying an entry? Is it easy to **exit or abandon** tasks without unintended consequences? Can users easily **interact with the inputs**, **buttons** using their **thumb** (mobile usage)?

HEURISTIC	**Consistency and Standards**
GOAL	Users should not have to wonder if different words, situations, actions mean the same thing.
CHECK LIST	Are design patterns (e.g., button styles, fonts, colors) **consistent across screens**? Are standard **platform conventions** followed (e.g., iOS vs. Android design guidelines)? Are icons**, gestures, and interactions** consistent throughout the app? Do **repeated actions behave the same** way every time (e.g., swiping left to delete)? Are similar features in the app named or **structured identically**?

HEURISTIC	**Error Prevention**
GOAL	Best designs carefully prevent problems from occurring in the first place.
CHECK LIST	Are potentially harmful actions (e.g., deleting an account) **confirmed** with a dialog or warning? Are **inactive or disabled elements visually** distinct to prevent misuse? Are actions **logically sequenced** to minimize mistakes (e.g., confirming details before proceeding)?

HEURISTIC	**Minimalist**
GOAL	Interfaces should not contain information that is **irrelevant** or rarely needed.
CHECK LIST	Are **primary actions prominently** displayed, with secondary options de-emphasized? Are **visual hierarchies clear**, guiding users toward their goals efficiently Are colors, fonts, and images used purposefully and consistently to **avoid distractions**? Is **negative space** effectively used to improve readability and focus?

HEURISTIC	**Help Users Recognize, Diagnose, and Recover from Errors**
GOAL	Error messages should **indicate the problem**, and suggest a **solution**
CHECK LIST	Are **error messages clear**, specific, and written in plain language? Do error messages suggest **actionable steps to** resolve the issue? Is the error's location **visually highlighted** for easy identification? Are users notified of errors immediately and not after they've progressed further? Can users easily **retry or correct** an action without starting over?