

KMK 3323 HUMAN-COMPUTER INTERACTION

SEMESTER 1 2025/2026 GROUP ASSIGNMENT (40%)

INCLUSIVE DIGITAL SOLUTIONS FOR A RESILIENT FUTURE (SMARTPHONE APPS)

OVERVIEW

Students will participate in a group project, with **six students per team**, which the instructor will assign. The project aims to give students experience in **applying HCI concepts and techniques to design and build a fully functional smartphone application (with a simple scope)** that addresses a pressing real-world issue in a way that encourages inclusivity and resilience.

The project follows the iterative **HCI design lifecycle** across four phases: **User Research → Conceptual Design → Low-Fidelity Evaluation → High-Fidelity Prototyping and Usability Testing**.

As the semester progresses, your project development will evolve from a paper description to a simple prototype to a high-level physical prototype (**FULLY FUNCTIONAL PROTOTYPE**).

Since this is a project-based class, we will discuss project progress on a weekly basis during class sessions. Your instructors will comment on each project as the semester progresses. Therefore, it is essential to complete the tasks before class, allowing you to receive input during the discussions. Furthermore, computer labs are equipped with new computers and Unity 3D software, as needed.

Key Requirements

1. **Inclusive Design Focus:** All projects **must** explicitly address the needs of diverse user groups, including those with varying abilities, ages, or contexts of use.
2. **Real User Engagement:** Teams **must** engage with the targeted users from the beginning. **Proof of engagement (e.g., video clips, photos with consent)** is a required deliverable for **Tasks 1 and 4**.
3. **AI Disclosure:** The fair use of AI tools is permitted, but it must be disclosed in the final report, detailing the specific tools used and their intended purpose.

4. **Fully Functional Prototype:** The final deliverable will be a **coded, working application** covering the core features defined in the **Minimum Viable Product (MVP) Scope**.

PHASE 1: Ideation, Research, and Scoping (Task 1)

Weight: 5% | Due: Friday, Week 4, before 5 pm

Aim: To rigorously define the problem, the users, and the narrow scope of the Minimum Viable Product (MVP).

Deliverable Component	Description and Requirements
Problem & Solution Definition	Clearly define the practical problem and why it requires a digital solution. Justify how your proposed smartphone app will solve this problem.
User & Context Analysis	Identify the target user group(s) (e.g., elderly farmers, sight-impaired tourists) and their context of use. Define any special needs or abilities your app must accommodate.
Requirements Gathering	Must use a minimum of two distinct user research methods: one qualitative (e.g., 3 semi-structured interviews, contextual inquiry) and one quantitative (e.g., survey, simple prioritisation card sort, from 15-20 samples). Document the methods and synthesised findings.
Competitive Analysis	Analyse 2-3 existing competitor or analogous applications, documenting their strengths, weaknesses, and key feature sets.
Core User Benefit Statement	State the single most important benefit your app provides that existing solutions do not offer, especially concerning accessibility or context of use. This is the design thesis for your project.
MVP Scope Document	Define the Minimum Viable Product (MVP) by categorising all potential features into: (1) MVP Must-Haves (features that will be coded for Task 4), (2) Should-Haves (features that will be designed/prototyped but not coded), and (3) Future Nice-to-Haves.
Success Metrics (DRAFT)	Define 3-5 specific, measurable usability metrics that you will use to evaluate your final coded prototype in Task 4 (e.g., "First-time users will complete Task X with an $\leq 10\%$ error rate").
Proof of Engagement	Submit a brief (≈ 1 minute) User-Interview Highlight Reel (simple phone video is acceptable) showcasing key quotes or interactions from your user research sessions.
Submission	Maximum 8-page report (excluding appendices) + Peer Review Form + Hard Copy.

Grading Rubric (Task 1: 5%)

Criteria (Total 5 Marks)	Excellent (4-5)	Good (3)	Acceptable (2)	Needs Improvement (0-1)
Problem Definition & Core Benefit	The problem is compelling; the Core Benefit is clearly	The problem is clear; competitive analysis is present, but the Core Benefit could be stronger.	The problem is general; the competitive analysis is weak or missing.	The problem is vague, or the solution is not justified.

	justified against strong competitive analysis.			
User Research & Data Quality	Two distinct methods were used. Findings are rich, synthesised well, and directly inform MVP. Video Proof is high quality.	Two methods were used. Findings are present and generally useful. Video Proof is present.	One method used /data is superficial/lacks synthesis. Video Proof is of poor quality.	Research methods are inappropriate or not documented.
MVP Scoping & Metrics	MVP is tight, realistic, and clearly prioritised. Success metrics are measurable and specific.	MVP is defined with some clarity. Metrics are defined, but could be more specific.	MVP is overly broad, or the features are not clearly categorised. Metrics are too vague.	No clear MVP or success metrics are defined.

PHASE 2: Conceptual Design and Mid-Fidelity Prototyping (Task 2)

Weight: 5% | Due: Friday, Week 7, before 5 pm

Aim: To develop a structured conceptual design and visually map the user flow digitally, justifying decisions with HCI principles.

Deliverable Component	Description and Requirements
Personas and Scenarios	Create 2-3 detailed personas representing your primary user groups. Develop a critical-path scenario for each persona demonstrating the use of the MVP features.
Mid-Fidelity Wireflow	Submit a clickable Mid-Fidelity Digital Wireflow (using Figma, Axure, or similar). This must clearly show the full navigation and content flow for your MVP features without final visual styling (i.e., minimal colour, simple shapes).
Design Rationale Matrix (DRM)	Justify key design decisions (e.g., button placement, information organisation) using an explicit matrix that links Feature/Design Element → Problem Solved/User Need → HCI Principle Used (e.g., Nielsen's Heuristics or Principles of Universal Design).
Simple Design System/Style Guide	Document your initial design standards: define the colour palette (including WCAG contrast ratios), primary font(s), and key interactive components (such as buttons and navigation bars).
Submission	Maximum 12-page report (excluding appendices) + Peer Review Form + Hard Copy.

GRADING RUBRIC (TASK 2: 5%)

Criteria (Total 5 Marks)	Excellent (4-5)	Good (3)	Acceptable (2)	Needs Improvement (0-1)
Personas & Scenarios	Personas are realistic and detailed. Scenarios effectively drive the core MVP features.	Personas are present but lack depth. Scenarios cover the basic use case.	Personas are generic or inconsistent. Scenarios are too brief or absent.	Personas/Scenarios not submitted or are irrelevant.
Mid-Fidelity Wireflow	Wireflow is complete, consistent, and logically maps the critical user path digitally.	Wireflow covers most screens and is generally easy to follow.	Wireflow is inconsistent, missing key screens, or still low-fidelity sketches.	Wireflow is not submitted or is unusable.
Design Rationale Matrix (DRM)	All key features are justified using specific, appropriate HCI principles. Clear link between need and principle.	Most features are justified using relevant HCI principles.	Justifications are vague, general, or use incorrect principles.	No systematic justification for design decisions.
Design System/Consistency	Comprehensive style guide including WCAG contrast. High internal consistency in the wireflow.	A basic style guide is present, and consistency is generally maintained.	Minimal documentation or visual elements are inconsistent.	No documentation or high inconsistency.

PHASE 3: Low-Fidelity Evaluation and Iteration (Task 3)

Weight: 10% | Due: Friday, Week 9, before 5 pm

Aim: To identify and fix early usability problems using a quick, inexpensive evaluation method.

Deliverable Component	Description and Requirements
Paper Prototype	A complete paper prototype with all the main features of your MVP interface.
Evaluation Execution	Select ONE method: Cognitive Walkthrough OR Heuristic Evaluation OR Cooperative Evaluation OR Wizard of OZ. Clearly document the procedure, participants/evaluators, and tasks.
Data Collection	Record your test sessions (audio, photo, or video) after obtaining participant consent. The recordings serve as proof and analytical evidence.
Findings and Iteration Plan	Analyse the findings, documenting 3-5 critical usability problems identified. For each problem, propose the specific redesign action and explain how it addresses the problem and which HCI principle supports the change.
Submission	Maximum 10-page report + Set of illustrations/recordings (appendices) + Peer Review Form + Hard Copy.

Grading Rubric (Task 3: 10%)

Criteria (Total 10 Marks)	Excellent (9-10)	Good (7-8)	Acceptable (5-6)	Needs Improvement (0-4)
Prototype Quality	The paper prototype is complete, detailed, and facilitates effective testing.	The paper prototype is functional but may lack some detail.	The prototype is incomplete or difficult to test due to a lack of detail.	The prototype is unusable for the chosen method.
Method Execution & Proof	The chosen method is appropriate, executed rigorously, and clearly documented. The video and photo proof are of high quality and clear.	The method is appropriate and executed well. Proof of evaluation is present.	The method is poorly executed or inappropriate for the prototype. The proof is weak.	Evaluation proof is missing, or the steps are not documented.
Problem Analysis	Identifies 5+ critical usability problems, clearly categorised and prioritised (e.g., severity ratings).	Identifies 3-4 significant usability problems with adequate categorisation.	Identifies only minor, superficial problems or lacks categorisation.	Problems identified are vague or not linked to prototype issues.
Redesign/Iteration Plan	Proposed redesigns are specific, feasible, and strongly justified by HCI principles, showing a clear path to Task 4.	Proposed redesigns are clear and generally justified by the findings.	Redesigns are vague or do not fully address the severity of the problems.	There is no clear plan for incorporating the findings into the next prototype.

PHASE 4: High-Fidelity Implementation and Final Usability Test (Task 4)

Weight: 20% (10% Prototype, 10% Testing) | Due: Week 13 during the CLASS

Aim: To deliver a functional application and quantitatively validate its usability against the success metrics.

Deliverable Component	Description and Requirements
Fully Functional Prototype (10 Marks)	A coded, working smartphone application (built using Android Studio, Unity, etc.) that implements all features categorised as MVP Must-Haves in Task 1. Must demonstrate high internal consistency (from Task 2 Style Guide).
Usability Test Execution	Conduct a formal usability test on the coded application using ANY appropriate usability testing method. Document the setup, tasks, and participants. Proof of engagement/testing is required.
Goal and Metric Validation	Use the success metrics defined in Task 1. Collect and analyse data to validate whether your app meets the predefined usability goals (e.g., success rate, time-on-task, error rate).
Heuristic Compliance Check	Perform a final internal check and document how the coded application adheres to at least 3-5 critical Nielsen's Heuristics or Shneiderman's Golden Rules, verifying that implementation didn't introduce new flaws.

Final Documentation	Analyse and report the results. Include a dedicated 'Future Work and Scalability' section that discusses technical limitations, additional features, and broader implementation challenges.
Final Presentation	A 10-minute presentation comprising: (1) A professional presentation showcasing the working MVP, and (2) A Live Discussion/Q&A/Q&A focusing on the usability test results and key design rationale.
Submission	Report + Source Code Repository Link (GitHub, etc.) + Evaluation recordings + Peer Review Form + Hard Copy.

Grading Rubric (Task 4: 20%)

Criteria (Total 20 Marks)	Excellent (18-20)	Good (14-17)	Acceptable (11-13)	Needs Improvement (0-10)
Prototype Functionality & Polish (10 Marks)	Fully Functional MVP. Code works reliably for all MVP features. UI/UX is consistent, professional, and directly reflects Task 2 iteration. High adherence to inclusive design standards.	Most MVP features function reliably. UI/UX is generally consistent but has minor flaws. Good effort on inclusive design.	Key MVP features are missing or buggy. UI/UX is inconsistent, showing poor adherence to the style guide.	The prototype does not function or is severely limited in scope.
Usability Test Execution (5 Marks)	The method is rigorous and appropriate for a coded app. Tasks align with MVP. Data is collected systematically. Clear proof of testing.	The method is suitable and executed well. Data collection is organised. Proof of testing is present.	The method is weak, or only superficial testing was conducted. Proof is present but unconvincing.	No usability test conducted or insufficient data.
Results & Validation (5 Marks)	Results are clearly analysed and presented with excellent data visualisation. Measures directly validate/invalidate metrics from Task 1. The heuristic check is thorough.	Results are clear and address the success metrics. Basic data analysis. Heuristic check is present.	The results are superficial, or the analysis is flawed. Metrics are not clearly addressed. Heuristic check is incomplete.	No data analysis or clear findings were presented.

PROJECT GUIDELINES

Task 1 Guidelines: Ideation, Research, and Scoping

1. Problem Definition and Core Benefit

- **Specificity is Key:** The problem must be narrow and solvable by a simple app. Avoid tackling global issues. For example, "Lack of accessible, real-time bus arrival information for visually impaired users in our university campus" is more specific than "Improving public transportation."
- **The Core User Benefit Statement:** This statement must clearly link to the problem and specifically highlight the inclusive design advantage your app offers over current solutions (e.g., "Our app is the most reliable tool for non-verbal users to report emergencies.").

2. User Research Guidelines

- **Mandatory Research Mix:** You must conduct at least one qualitative method and at least one simplified quantitative method to triangulate your requirements.
 - **Qualitative (Depth):** Requires a minimum of 3 semi-structured interviews with target users (or their proxy/caretaker). Focus on the *context* and *narratives* of their experience with the problem.
 - **Quantitative (Breadth/Validation):** A brief survey is required to gather data from 15-20 relevant participants to validate the prevalence of the problem or prioritise features.
- **Proof of Engagement:** The Video Highlight Reel (≈approximately 1 minute) must show your team interacting with users. Ensure consent is obtained.

3. MVP Scope Document Guidelines

- **Definition:** The MVP Must-Haves must represent the *absolute minimum* set of features required to deliver the Core User Benefit Statement.
- **Prioritisation Justification:** Briefly justify *why* a feature was placed in the Must-Have category versus a Should-Have. (e.g., "Login is a Must-Have because it allows for secure data storage, which is required for WCAG AA data security compliance.")

When filling out the table, you must be ruthless in cutting features that aren't essential:

Feature Name / Function	User Need Addressed	Priority Category (MVP)	Justification (Why this priority?)
Example: One-Click Emergency Dial	Visually impaired users need immediate assistance in an unfamiliar location.	MVP Must-Have	Direct tie to safety, which is a primary component of our Core Benefit Statement. Essential functionality.
Example: Integrated QR Code Scanner	User needs to scan tickets/vouchers for entry.	Should-Have	Useful, but the user can manually enter the code if necessary. Not essential for core benefit delivery.

4. System, Accessibility Principles, and Metrics Guidelines

- **Universal Design Principles:** Instead of WCAG, your requirements must include a section outlining how your proposed app adheres to at least 3 of the 7 Principles of Universal Design (e.g.,

Equitable Use, Simple and Intuitive Use, Perceptible Information). Explain which features address these principles.

- **Draft Success Metrics (CRITICAL):**
 - Define 3-5 specific, measurable goals for your final prototype evaluation in Task 4.
 - These metrics must be quantitative and must relate directly to the MVP's core tasks.
 - Examples: Task Completion Rate (TCR), Time on Task (ToT), or User Error Rate.

Task 2 Guidelines: Conceptual Design and Mid-Fidelity Prototyping

1. Personas and Scenarios

- **Go Deep, Not Wide:** Develop 2-3 detailed personas maximum. Each persona should clearly articulate user behaviours, motivations, goals, and especially the special needs or accessibility requirements relevant to your app.
- **Critical Path Scenario:** Create a detailed Scenario for each persona. The scenario must demonstrate the user completing the primary tasks related to the MVP Must-Haves defined in Task 1, showcasing the app's core benefit.

2. Mid-Fidelity Wireflow

- **Digital, Not Paper:** The deliverable must be a mid-Fidelity Digital Wireframe using tools such as Figma, Adobe XD, or Axure. No hand-drawn sketches are permitted for this task.
- **Focus on Structure:** The wireflow should prioritise Information Architecture (IA) and flow, not aesthetics. Use black/white or simple grayscale, generic fonts, and placeholder text/images.
- **Clickable Requirement:** The prototype must be clickable to allow navigation through the entire critical path scenario (from app launch to task completion). This prepares you for the flow testing in Task 3.
- **Content and Navigation:** Clearly display all necessary navigation elements, input fields, and the content structure.

3. Design Rationale Matrix (DRM)

- **Justify the Why:** Select 5-7 key design elements from your wireflow (e.g., a specific button, a navigation pattern, a colour choice) and complete the DRM template.
- **Source Your Theory:** Each justification column must cite a specific HCI Principle, Heuristic (e.g., Nielsen's 1-10), or Design Law (e.g., Fitt's Law, Hicks Law), etc. Avoid using vague statements like "It looks modern."
- **EXAMPLE OF DRM**

Feature / Element	User Goal Supported	Design Decision/UI Pattern Used	HCI Principle/Heuristic Justification
Primary Action (e.g., "Book Ride" button)	Users need quick and reliable access to the app's main function, especially when multitasking.	The button is sticky (always visible at the bottom of the screen) and uses a high-contrast colour palette defined in the style guide.	Visibility (Nielsen's 1) and Fitt's Law (Efficiency). The sticky placement minimises travel distance for the thumb.

Input Feedback (e.g., Password field)	User needs security validation without visual obstruction.	A small, silent haptic vibration confirms each character input when the user is wearing gloves or has visual impairment.	Feedback (Shneiderman's Golden Rule 3). Accessibility (WCAG 2.1 Success Criterion 2.1.1 - Keyboard Accessibility for haptic confirmation).
Data Entry (e.g., Date/Time selection)	User needs to easily select a future date/time without errors.	Implemented a calendar picker with large, tappable areas instead of requiring manual text input.	Prevent Errors (Nielsen's 5). The restricted input method (calendar) prevents common date format errors and is easier for users with dexterity issues.

4. Simple Design System/Style Guide

- **Consistency Standard:** Document the visual standards you will use for the high-fidelity prototype (Task 4). This document must ensure visual consistency.
- **Accessibility Mandate:** Specifically document the contrast ratios for your primary text and background colours.
- **Key Elements:** Define the standard look for your primary button, link styles, and the chosen typography (font and size). The minimum readable size should be considered for inclusive design.

By following these guidelines, students will produce a mature, justifiable design that is ready for formal evaluation in the next phase.

Task 3 Guidelines: Low-Fidelity Prototype (Paper-based) & Evaluation

1. Prototype Requirement

- **Design Tool Allowed, Material Mandated:** Students are permitted to create interface elements (screens, menus, buttons, etc.) using digital tools such as Figma, Adobe XD, or Moqups. This allows for a clean look, strong consistency (using the Task 2 style guide), and a better "look and feel."
- **The Low-Fidelity Rule:** Regardless of the design tool used, the prototype must be physically represented using paper, printouts, or index cards. This keeps the design in a form that is cheap and easy to physically manipulate and discard during testing.
 - *The benefit:* This approach maximises the aesthetic quality for the user without increasing the difficulty of changing the underlying design logic (a core goal of low-fidelity evaluation).
- **Completeness:** The prototype must cover the full critical user scenario and include all necessary screens and interface elements required for a user to complete the MVP Must-Have tasks defined in Task 1.

2. Evaluation Execution

Method Choice: You must select only ONE evaluation method: Cognitive Walkthrough, Heuristic Evaluation, Cooperative Evaluation, Think Aloud Protocol or Wizard of OZ.

Clear Procedure: Clearly document the protocol for your chosen method, including the number of participants and evaluators, as well as the specific tasks performed.

Proof: You must submit a recording (audio, photographic, or video) of the test sessions to prove the evaluation was successfully conducted.

3. Findings and Iteration Plan

- **Focus on Critical Issues:** Analyse your data and document a list of **critical usability problems** (High severity). Avoid reporting minor, cosmetic issues.
- **Systematic Analysis:** For **each** of the critical problems, you must provide the following:
 1. **Observation:** A brief description of what revealed the problem.
 2. **Severity Rating:** Assign a high, medium, or low severity rating (justified).
 3. **Redesign Action:** A concrete plan (e.g., sketch, description) of **how you will change the design** for the high-fidelity prototype in Task 4.
 4. **HCI Justification:** Explicitly state the **HCI Principle or Heuristic** that the original design violated and that the new design will adhere to.

Example:

Usability Problem Found	Severity (High/Med/Low)	Observation/Evidence (What happened?)	Redesign Action for Task 4	HCI Principle Justification for Fix
Example: User struggled to find the setting to change font size (P1, P3).	High	The settings icon was buried in a secondary menu, and P3 never found it under pressure.	Move the accessibility settings directly to the main navigation bar, labelled "A11y Options."	Recognition Over Recall (Nielsen's 6). Reduce memory load by making crucial options visible.
Example: Two users accidentally triggered the "Log Out" button while aiming for "Save."	Medium	The two buttons were too close together and similar in size on the final screen of the task.	Increase the spacing between "Log Out" and "Save." Change the "Log Out" colour to a neutral grey to demphasize it.	Fitts' Law (Efficiency/Error Prevention). Increase the target distance and use visual coding to distinguish between destructive and non-destructive actions.
[Problem 1:...]				
[Problem 2:...]				

Task 4 Guidelines: High-Fidelity Prototype & Usability Testing

1. Fully Functional Prototype (Implementation)

- **Technology Choice (Flexible):** You may build a working application using traditional coding (e.g., Android Studio, React Native) or a drag-and-drop/low-code platform (e.g., AppyPie, Bubble, Figma/Protopie with high-fidelity features) that allows for data and interaction logic.
- **Scope Compliance:** The prototype must fully implement and function for all features defined as MVP Must-Haves in your Task 1 Scope Document.
 - *Functionality Check:* The app must be able to perform the core tasks (e.g., save data, show customised results, navigate through the full flow) as a real application would.
- **Design Consistency:** The final app's user interface (UI) must maintain visual consistency with the Style Guide (Task 2) and incorporate all critical design improvements identified in the Task 3 evaluation.

- **Source Submission:** You must submit a link to your source code repository, project file, or the public link to the functioning app/platform project (e.g., GitHub, Bubble project link) alongside the report for verification.

2. Mandatory Evaluation Protocol

A: Mandatory Core Usability Testing (Performance Focus)

- **Method:** Conduct a formal usability test on the **coded application** using **ANY suitable usability testing method** (e.g., Think-Aloud Protocol, Performance Testing, A/B Testing, Remote Moderated Testing).
- **Benchmark Tasks:** Use a set of **benchmark tasks** that align directly with the **MVP Must-Haves** and allow you to measure the **Success Metrics** defined in Task 1.
- **Participants:** Test with a minimum of **5 participants** who align with your target user groups.
- **Proof:** You **must** make a recording (audio, photographic, or video) of the test sessions to analyse and prove that the testing was conducted. Obtain consent from all participants before recording.

B. Subjective Feedback (Choice Required)

Choose **ONE** of the following methods to collect subjective data from your participants:

- **Option A: Post-Test Interview:** Conduct a structured interview (5-10 minutes) with each participant to gather in-depth, qualitative insights, focusing on the reasons behind their performance and the effectiveness of inclusive design features.
- **Option B: Standardised Questionnaire:** Administer a recognised, standardised questionnaire (e.g., **System Usability Scale (SUS)**, **User Experience Questionnaire (UEQ)**, or **Usability Metric for User Experience-Lite (UMUX-Lite)** or other relevant questionnaire) and report the calculated score.

3. Analysis and Reporting

- **Metrics Validation:** Analyse the test data to formally validate or invalidate the **Draft Success Metrics** defined in Task 1 (e.g., "The team achieved a 75% Task Completion Rate, falling short of the 90% goal").
- **Heuristic Compliance Check:** Conduct an internal team review of the final coded application and document how the implemented features adhere to specific Nielsen's Heuristics (e.g., did the implementation compromise 'Error Prevention?').
- **Findings and Future Work:** Document a final list of **High-Severity usability flaws** that remain in the coded app. For each, propose a **specific redesign action** for the next version.
- **AI Disclosure:** Include a mandatory section disclosing any AI tools used in the coding, design, or analysis phase.

4. Final Showcase and Deliverables

- **Final Report:** Compile all documentation, analysis, and findings into the final report.
- **Presentation:** Deliver a **10-minute presentation** to showcase your work:
 - A high-quality demonstration of the **fully functional MVP app** being used to complete the core benchmark tasks.

- **7-minute Live Discussion/Q&A/Q&A:** Focus on the **Usability Test Results** and your team's design rationale.
- **Submission:** Submit the report, code link, and individual contributions form (PEER REVIEW FORM).

1. Usability Goals Validation Table

This table validates the quantitative goals set back in Task 1.

Success Metric Type	Benchmark Goal (From Task 1)	Test Result (Actual Data)	Goal Met? (Yes/No)	Discussion/Reason for Variance
Effectiveness (TCR)	≥90% of users complete the core task.	7/10 users completed the task (70%).	No	Variance caused by confusion over the payment method selection screen (Problem B, below).
Efficiency (ToT)	Median time for task completion ≤45 seconds.	Median time was 52 seconds.	No	The app's animation load time added unnecessary cognitive friction.
Satisfaction (Error Rate)	User error rate during primary input ≤10%.	Observed error rate was 5%.	Yes	Success due to the effective implementation of the Task 3 paper prototype fixes.

2. Final Heuristic Compliance Check (Template for Task 4 Report)

This table verifies the quality of the final *coded* application.

Nielsen Heuristic / Principle of UD	Element/Feature Checked	Compliant? (Yes/No)	Evidence/Explanation (Coded App)
Heuristic 4: Consistency & Standards	Navigation bar placement and iconography.	Yes	Placement and icons are uniform across all 5 MVP screens, adhering to the Task 2 style guide.
Heuristic 5: Error Prevention	User input on the date selector.	No	The coded date field allows the user to manually type an invalid date format, crashing the input (Bug 2).
Principle of UD: Simple and Intuitive Use	The process for initiating a new service request.	Yes	The process is a single-screen form with clear, minimal required inputs.
[Heuristic/Principle 4:...]			

Final Prototype Source Code Disclosure

Component	Status/Link	Notes
Fully Functional MVP App	[Live App Link/APK download]	Tested on Android 12 and 14.
Source Code Repository	[GitHub Link]	Repository includes all commits and documentation.
AI Tool Disclosure	ChatGPT (v4.0)	Used for brainstorming descriptive error message text in the final code.

REMINDER

1. The report format should use a font type Century Schoolbook, size 11, and a spacing of 1.5 lines. In addition, you must include **references, such as the list of books, journals, online articles, etc.,** that you have used to research the topic and prepare the report. Cite your references appropriately using the **APA style** of reporting.
2. Your group must include an individual contribution form (a summary of individual contributions for each task signed by all team members) together with every project submission.
3. All submissions **MUST FOLLOW THE DEADLINES** scheduled in the course planning. You can get input from your instructor during the class session and redo your work before the deadline.