

PART 3.1: System Prototype Report

Team Unit 3 - TAYABA / PEPITO

1. System Prototype Description

Our group has implemented a detailed prototype of the Student Assisted Studying and Homework Application (SASHA) interface using **Figma**. Figma was chosen for its interactive prototyping capabilities, ease of distribution to testers via links, and collaborative features, allowing team members to edit the prototype concurrently.

The prototype focuses on delivering a fully functional interface, simulating much of the front-end user experience without needing to implement complex backend application functionality. It integrates and refines aspects from our Part 2 design alternatives, aiming for a modern-minimalistic aesthetic while addressing the identified user needs for pacing and task management.

Key Screens and Interactions:

- **Splash Screen:** The initial screen displaying the SASHA app's logo, visible for a brief 2 seconds on startup. (As seen in Figure 2 of the sample document).
- **Main Menu:** The central hub where users navigate to core functionalities. It features prominent buttons for "FOLDERS" and "QUIZ," providing clear pathways for user choice. A "Welcome!" message is also displayed. (As seen in Figure 2 of the sample document).
- **Navigation Drawer:** Accessible from the Main Menu, this drawer allows users to navigate through app settings or exit the application completely. (As seen in Figure 2 of the sample document).
- **Folders Screen:** Displays a list of user-created folders, typically representing different course subjects (e.g., "Computer Programming 2," "Human Computer Interaction"). Users can access a dropdown menu for options like editing, sorting, or creating new folders. (As seen in Figure 3.1 of the sample document).
- **Delete Overlay:** Activated by holding down a folder or file, this overlay provides a confirmation prompt ("Delete X Folders?") with "Cancel" and "Delete" options, ensuring user control and error prevention. (As seen in Figure 3.2 of the sample document).
- **Keyboard Overlay:** This overlay appears automatically when the user needs to input text, such as when creating or renaming a folder or file, providing a standard input interface. (As seen in Figure 3.3 of the sample document).
- **Folder-Files Screen:** Accessed by clicking on a specific folder, this screen displays the files and tasks within that folder. It features tabs to switch between "Files" and "Task" views. (As seen in Figure 3.5 of the sample document).

- **Import Screen:** Allows users to import external PDF or image files into a selected folder. (As seen in Figure 3.4 of the sample document).
- **Creating Text Files:** Within a folder, users can create new text files via a Floating Action Button (FAB) or a dropdown option, which then activates the keyboard overlay for naming. (As seen in Figure 3.5 of the sample document).
- **Folder-Task Screen:** Displays a list of tasks associated with a folder. Tasks can be marked as "Complete" or "Ongoing" by tapping on them, reflecting their progress. (As seen in Figure 3.7 of the sample document).
- **Sorter Overlay:** Accessed via a "sort" button, this overlay provides options to sort files/tasks by "Name" or "Progress" and in "Ascending" or "Descending" order. (As seen in Figure 3.8 of the sample document).
- **Quiz Screens:**
 - **Quiz Main Screen:** Shows a list of created quizzes.
 - **Quiz Creation:** Users can create new quizzes, defining quiz type (e.g., multiple choice, word-based), questions, and answers.
 - **Quiz Formatting Screen:** Allows users to set quiz settings like timers, number of tries, and shuffling options.
 - **Quiz Taking Screen:** The interface where users actively take the quiz.
 - **Result Screen:** Displays the user's score and indicates correct/incorrect answers upon completion. (As seen in Figures 4.1, 4.2, 4.3, and the Result Screen in the sample document).

Prototype Flow (Figures 2, 3.1-3.8, 4.1-4.3 in sample document):

The prototype simulates common user flows, including:

- **Entering and Exiting the Prototype:** From the Home Screen through the Splash Screen to the Main Menu, and back via the Navigation Drawer or home button.
- **Folders-Files/Tasks:** Illustrates renaming files/folders (via a "pen" icon), deleting (holding down or dropdown), adding new folders/files (dropdown), importing PDFs/images, creating text files, creating and marking tasks.
- **Quiz Functionality:** Demonstrates creating new quiz files, adding and editing quiz questions, setting quiz formats (timer, tries, shuffle), taking the quiz, and viewing results.

Adaptive Design:

The Figma prototype has been designed to adapt to different screen sizes, showcasing how it would look on a wider screen (laptops/tablets) versus a smaller phone screen, ensuring responsiveness as per the initial design considerations. (As seen in Figure 5 of the sample document).

2. Implementation Challenges

Implementing this prototype in Figma presented several challenges, primarily inherent limitations of a high-fidelity prototyping tool that is not a full-fledged development environment:

- **Backend Simulation:** The core challenge was simulating backend application functionality. While the interface appears functional, the actual logic for features like real-time notifications, robust file synchronization, complex quiz grading algorithms, or persistent data storage (beyond Figma's linking capabilities) cannot be truly implemented. This means functionalities like the "Alarm feature" (from Part 2's requirements changes) could only be represented conceptually rather than being actively triggered by real-time events.
- **Internet Dependency for Saving:** Figma requires an internet connection to save edits, which can be problematic if team members face connectivity issues, potentially hindering collaboration and progress as noted in the rationale.
- **Responsiveness for Smaller Elements:** While Figma allows for adaptive design, ensuring that smaller interactive elements (buttons, icons) remain easily tappable and accessible on larger screens without appearing "clustered or compressed" (a concern from Part 1) required careful attention to spacing and sizing.
- **Dynamic Content and User-Generated Data:** Simulating dynamic content, such as a perpetually growing list of tasks or files, or allowing users to truly "type" and save arbitrary content, is limited. The prototype uses placeholder text and predefined interaction flows.
- **External Integrations:** Features like "music can be searched with the Internet or can be played from the files" (from Part 2's requirements changes) could only be conceptualized as a toggle to ambient noises. Actual integration with music libraries or local file systems is beyond Figma's scope.
- **Performance Testing:** Figma cannot simulate the actual performance of the application on various devices, including load times, animation smoothness, or battery consumption, which are crucial for a mobile app.

Aspects we wanted to build but could not fully implement:

- Live, real-time notification system integrated with actual course platforms.
- Full-fledged search and playback of music from the internet or local device files.
- Sophisticated quiz question types beyond multiple-choice/word-based, such as drag-and-drop or complex calculations.
- User account management, login/logout, and data synchronization across devices.
- Sharing quizzes or files with other users.

3. Justification of Prototype Design

The design of the SASHA prototype is an empathetic response to the multifaceted challenges faced by online college and senior high school students in the Philippines, as detailed in Part 1. Our design choices prioritize simplicity, intuition, and direct problem-solving, aiming to reduce the "unhealthy amount of stress" caused by unmanaged academic demands.

- **Addressing Pacing and Task Management:** The central problem of students struggling with self-pacing and accumulating unfinished tasks is directly addressed through the clear "Folders" and "Task" sections. The ability to categorize notes and tasks within folders, coupled with the "Complete" / "Ongoing" status for tasks, provides a tangible way for students to track progress and stay on pace. The newly integrated **sorter feature** (from Part 2 feedback) further enhances task management efficiency, allowing users to filter and sort based on course, importance, or progress.
- **Mitigating Notification Deficiencies:** While full real-time integration is a backend challenge, the prototype's logical flow accounts for the need for better alerts. The explicit inclusion of an **"Alarm feature to alert you if the user is out for awhile"** (a key change from Part 2 requirements) demonstrates our commitment to proactive information delivery, reducing the need for constant manual checks.
- **Enhancing Self-Assessment and Review:** The dedicated "Quiz" module empowers students to create personalized study tools. This directly addresses the need for self-assessment, crucial for students who may lack timely communication with teachers or have limited access to comprehensive review materials.
- **User-Centered Simplicity and Accessibility:** Our adherence to a **modern-minimalistic aesthetic** (a core design philosophy from Part 2) ensures the interface remains uncluttered and intuitive. Features like the readily available **Keyboard Overlay** and simple tap/hold interactions for CRUD operations align with the need for an easy-to-learn interface that doesn't add to cognitive load. Furthermore, the focus on relying on **signs and words over color-dependent cues** (from Part 1's accessibility criteria) ensures the prototype is more inclusive for users with color blindness.
- **Responding to User Feedback:** The design iterated based on feedback from Part 2. The shift from Lo-Fi to **ambient noises for the music toggle** reflects user preference for choice and a universally acceptable study environment. The integration of the **task sorter** and the emphasis on **clear navigation** were direct responses to user suggestions for enhanced convenience.
- **Competitive Parity:** By analyzing competitors like Notion, Quizlet, and StudyBlue, our prototype strives for similar levels of convenience and efficiency in task management, note-taking, and self-assessment, offering a competitive solution within the Android mobile ecosystem.

In essence, the SASHA prototype is not merely a visual representation; it's a carefully considered design that translates the identified problems and user needs into a practical, intuitive, and supportive digital tool for online students.

4. Usability Specifications

Our initial usability specifications for the SASHA system are designed to evaluate its effectiveness in addressing the outlined problems and user needs. These specifications align with standard usability measures and our project objectives:

- **Effectiveness:**

- **Criterion:** Users should successfully complete core tasks (e.g., creating a folder, marking a task complete, taking a quiz) without requiring external assistance.
- **Measurement:** Task completion rate (percentage of tasks completed correctly).
- **Efficiency:**
 - **Criterion:** Users should be able to complete common tasks within specified time limits.
 - **Measurement:** Task completion time (time taken to complete a task from start to finish).
 - Main Menu Navigation: Within 1 minute or Below.
 - Folder Management (Create/Edit/Delete): Within 5 minutes or Below.
 - Quiz Interaction (Create/Take): Within 5 minutes or Below.
- **Utility:**
 - **Criterion:** The prototype should support functions and alternatives that meet the user's needs for pacing, task management, and self-assessment.
 - **Measurement:** Qualitative feedback from user surveys regarding satisfaction with features and their perceived usefulness for academic management.
- **Learnability (Ease of Learning):**
 - **Criterion:** New users should be able to learn the interface and successfully perform basic operations quickly.
 - **Measurement:** Number of errors made during initial task attempts, and self-reported ease of learning from a subjective questionnaire.
- **Memorization:**
 - **Criterion:** Users should easily remember steps and interactions for tasks after an initial learning period.
 - **Measurement:** Success rate and time taken for tasks after a short break, and qualitative feedback on recall from surveys.
- **Aesthetic and Non-Distracting Design:**
 - **Criterion:** The visual design should be appealing and not distract the user from their academic tasks.
 - **Measurement:** User ratings on survey questions related to visual appeal, clarity, and distraction levels.
- **Accessibility:**
 - **Criterion:** Information and functionality should be discernible and usable by users with color blindness (relying on signs and words, not just color).
 - **Measurement:** Observational notes during testing for any difficulties related to color-dependent cues.

5. Initial Evaluation Plan

Due to the ongoing quarantine and online class setup, our evaluation will leverage online social media platforms such as Microsoft Teams and Discord to conduct live observations. The plan is segmented into three parts:

5.1. Data Gathering Methods:

- **Survey (Quantitative):** Post-test survey using a 5-point Likert scale (referencing Table 5 in the sample document). This will gather data on user experience and UI design.
- **Feedback (Qualitative):** An open-ended feedback section in the survey for users to voice concerns or issues.

5.2. Population:

- We will select **10-20 participants** from the Senior High School and College populations.
- Participants will be required to perform a set of outlined tasks using the prototype.

5.3. Prototype Tasks (Benchmark Tasks):

The tasks are divided into three sections, focusing on core functionalities:

- **Main Menu Tasks:**
 - Enter and Exit the Prototype.
 - Navigate through various sections of the Prototype.
- **Folder Tasks:**
 - Create new files and folders.
 - Delete existing files/folders.
 - Edit files (e.g., renaming).
 - Import PDF or IMG files.
 - Create text files.
 - Create and mark tasks as complete/ongoing.
 - Sort files and folders.
- **Quiz Tasks:**
 - Create quiz files.
 - Create quiz questions.
 - Take quizzes.
 - Utilize quiz formatting options (timer, tries, shuffle).

These tasks were selected as they directly reflect the CRUD (Create, Read, Update, Delete) operations and navigation measures designed into the prototype.

5.4. Roles and Responsibilities (Team Member Tasks):

- **Neil Angelo Almanzor:** Record time users interact with a task section, take notes of the user's experience, and relay the task to the participant.
- **Jeschee Jay Tabuno:** Record time users interact with a task section, take notes of the user's experience, and relay the task to the participant.

5.5. Time Interpretation (Table 3 in sample document):

This table serves as a guideline to interpret the success of the design based on time spent:

| Task Section | Time to Accomplish Tasks | Interpretation