Bachelor Thesis

Scrap Book App

Authored by
Vasilije Mirković

academic degree
Bachelor of Science (BSc)

May 7, 2025

Study code according to study sheet:	UA 521
Field of study:	Informatics
Supervisor:	Univ.-Prof. Dipl.-Ing. Dr. Helmut Hlavacs

Contents

1 Introduction

This Bachelor thesis presents the functional extension of an existing Android application, the Scrap Book App. The project focuses on implementing a cloud-based backup mechanism using Firebase Storage and introducing administrator functionalities. These enhancements allow users to securely store their data both locally and in the cloud, ensuring that important user-generated content can be restored even after uninstalling the app or changing devices.

Special emphasis was placed on ensuring a seamless and intuitive user experience, given that the app is primarily designed for children recovering from surgery or serious illness. The simplicity of the interface, combined with robust backend support, improves the app's long-term usability, resilience, and suitability for real-world healthcare environments. The developed solution can serve as a foundation for future research and development in the area of mobile healthcare support for young patients.

In recent years, mobile applications have become integral tools in supporting patient recovery, particularly in pediatric healthcare. The Scrap Book App was originally developed to assist children during home healing process after surgery or serious illness by providing a creative and interactive outlet for emotional expression. By allowing children to draw, write diary entries, take photographs, and track their moods, the application offers a simple yet effective way for them to communicate their emotional states, helping parents and healthcare professionals monitor recovery progress.

Despite the successful implementation of its core functionalities, the original version of the app faced a significant limitation: the absence of any data persistence mechanism beyond local storage. When a user uninstalled the app or switched to a new device, all previously saved data — including drawings, diary entries, and mood inputs — was irretrievably lost. This lack of long-term data storage significantly hindered the app's utility, especially when emotional progress over time was essential for evaluating a child's recovery.

This thesis addresses that limitation by integrating cloud-based storage using Firebase Storage, thereby enabling secure and user-specific backups. Users can now upload their files to a cloud repository and restore them as needed, even after reinstalling the app or transitioning to a different device. In addition, an administrator interface was developed to enable authorized personnel, such as doctors or therapists, to manage user accounts and delete inactive profiles. These enhancements contribute to the app's resilience and practical applicability in healthcare settings.

The extension of the Scrap Book App presented in this thesis thus offers an example of how cloud technologies and user-centered design can be combined to create more robust mobile healthcare solutions. Furthermore, it lays the groundwork for future improvements in areas such as data encryption, authentication, and automated synchronization. The motivation behind this extension is further explained in the following section.

The motivation behind this Bachelor thesis stemmed from the desire to create a meaningful and technically sound contribution to healthcare-focused mobile applications. Children undergoing healing process from medical procedures often struggle to articulate their feelings through traditional means, such as verbal communication or written questionnaires. Tools that allow for alternative forms of psychological resilience are therefore crucial in supporting both the psychological and physical aspects of recovery.

The original Scrap Book App addressed this need by providing a child-friendly platform for drawing, journaling, and emotional tracking. However, its dependence on local storage posed a critical limitation: the risk of losing all valuable user data in the event of app uninstallation, device loss, or system failure. Considering the emotional and therapeutic value embedded in the children's content, it became clear that the application required a more durable solution for data preservation.

The decision to integrate Firebase Storage was driven by its ease of use, security features, and seamless compatibility with Android applications. By enabling cloud-based backup and restore functionalities, the Scrap Book App can now offer continuity of care and emotional tracking, supporting long-term therapeutic goals. Furthermore, the addition of an administrator module enhances the app's maintainability, making it better suited for clinical environments where user management is essential.

On a personal level, this project represented an opportunity to deepen technical skills in mobile development, cloud integration, and user experience design, while simultaneously contributing to a socially meaningful cause. The potential to improve the emotional well-being of young patients provided

a strong incentive throughout the development process, reinforcing the belief that technology can play an important role in healthcare beyond physical treatment alone.

2 Related Work

This section provides a structured overview of the academic and technological foundations relevant to the Scrap Book App. It summarizes key developments in mobile health (mHealth) applications, the role of cloud storage in healthcare, data privacy concerns, and the use of Firebase in academic prototyping projects.

Mobile Health Applications (mHealth) Overview. Mobile health (mHealth) applications have become an essential part of modern healthcare, providing support for patient monitoring, chronic disease management, and mental health interventions. Especially in pediatric settings, digital tools have shown potential for improving emotional well-being during recovery processes.

Studies such as Ahtinen et al. (2018) emphasize that digital platforms allowing emotional self-expression — including drawings, diaries, and interactive games — positively influence children's psychological resilience. However, many existing mHealth tools tend to rely on structured data input, such as standardized mood scales, limiting the flexibility of emotional expression. These findings highlight the importance of designing mobile health applications that prioritize emotional engagement and flexible forms of self-expression.

Cloud Storage Solutions in mHealth. The integration of cloud technologies has addressed one of the critical challenges in mobile health applications — ensuring data continuity across devices and preventing data loss. Mahmoud et al. (2019) found that cloud-enabled applications exhibit higher user retention rates and satisfaction, primarily due to seamless data backup and recovery.

Firebase Storage, a widely used Backend-as-a-Service (BaaS) platform, provides scalable and secure cloud storage options for mobile applications. Its straightforward integration with Android Studio makes it particularly suitable for lightweight healthcare prototypes like the Scrap Book App. The integration of scalable cloud storage systems thus represents a crucial advancement in ensuring both data resilience and user satisfaction in mHealth environments.

Data Privacy and Ethical Considerations. Handling sensitive health-related user data requires strict adherence to data protection regulations such as GDPR and COPPA. Research by Livingstone and Stoilova (2020) stresses that applications for minors must prioritize transparency, parental consent, and minimal data collection.

Although the Scrap Book App does not collect clinical medical records, it handles emotionally sensitive data, such as mood entries and diary notes. This underlines the need for privacy-aware design, encrypted data storage, and secure authentication mechanisms in future iterations. These considerations form the ethical backbone of any application dealing with sensitive user data, reinforcing the importance of privacy-focused design in future healthcare technologies.

Firebase for Academic Prototyping. Firebase has emerged as a leading platform for academic mobile application development due to its ease of use, robust free tier, and fast deployment capabilities. Comparative studies (e.g., J. Li et al., 2022) show that Firebase-based projects can reduce development time by up to 35% compared to traditional backend setups.

Its integration with Android applications through well-documented SDKs makes Firebase an ideal choice for educational prototypes where rapid iteration, scalability, and minimal server maintenance are priorities. In the context of the Scrap Book App, Firebase enabled the seamless implementation of cloud-based backup and restore functionalities within a limited timeframe. Firebase thus serves as an effective platform for bridging the gap between theoretical healthcare concepts and practical, deployable mobile applications.

In summary, the literature and technological review conducted in this section highlights the growing importance of mobile health applications in pediatric care and the crucial role that cloud storage technologies play in preserving emotionally valuable user data. While existing mHealth solutions provide a solid foundation, many fall short when it comes to offering flexible emotional expression tools, seamless cross-device data continuity, and strong privacy protections — areas that the Scrap Book App specifically aims to address through its functional extensions. The findings discussed here

validate the architectural choices made during the development of the app, particularly the integration of Firebase as a lightweight yet powerful backend infrastructure. Furthermore, the critical importance of ethical considerations, especially when dealing with minors' emotional data, has been underscored throughout the review. These insights provided a clear roadmap for extending the Scrap Book App not only as a technical upgrade but also as an ethical and user-centered healthcare tool. In the following sections, the practical implementation of these concepts will be examined in detail, demonstrating how theoretical best practices were translated into a functional and reliable mobile application for supporting children's emotional recovery. In the next section we show how these lessons informed our design.

3 Basic Technologies

In this section, we present the core technologies underlying the Scrap Book App extension.

Mobile applications often need to strike a balance between local availability and long-term data reliability. In the first part of this section, we review the original architecture of the Scrap Book App, where data persistence was handled via the Room library on top of an SQLite database. We describe the advantages of this approach—fast and straightforward implementation, offline access, and integration with Android lifecycle components—as well as its key drawbacks: data loss upon app uninstallation or device change.

Next, we demonstrate how a hybrid model combining local and cloud storage significantly enhanced the app's functionality and security. We explain in detail the process of introducing Firebase Storage, from initial project registration in the Firebase console to the implementation of file upload and download procedures via the SDK's 'putFile' and 'getFile' methods. Additionally, we discuss the organization of user folders, considerations for secure data transmission, and the impact of these technological choices on scalability and user experience.

3.1 Technological Background

The Scrap Book App was initially developed as an Android mobile application using the Java programming language within the Android Studio integrated development environment (IDE). The main functionalities of the app included features such as drawing, diary writing, photographing important moments, and tracking emotional states. The overall purpose was to offer a child-friendly platform to support emotional recovery after serious illnesses or surgical procedures. From a technical standpoint, data persistence was managed locally through the use of Room, an object-mapping library that provides an abstraction layer over SQLite databases in Android applications. Room facilitated structured storage of user-generated content, such as diary entries and mood inputs, directly on the user's device. It offered compile-time validation of SQL queries, simplified database access, and integrated well with Android's lifecycle components. While Room successfully provided local storage and offline access to user data, it introduced significant limitations in the context of long-term data security. All content was confined to the device's internal storage directory under the path '/data/data/packageName/', which is only accessible to the application itself. Although this approach ensured a basic level of security by isolating data from other apps, it had the major drawback that all data would be lost if the app were uninstalled or if the device were damaged or replaced. As a result, important emotional and therapeutic records created by children could easily be irretrievably lost. Recognizing these limitations, the extension presented in this thesis introduced cloud-based storage using Firebase Storage. Firebase, a platform developed by Google, offers a suite of cloud services designed for mobile and web development. Firebase Storage, in particular, provides a secure and scalable solution for storing and retrieving user-generated files, such as images and documents, through seamless SDK integration with Android applications. The introduction of Firebase into the Scrap Book App's technology stack required several technical adjustments. Firstly, the Android application had to be registered within the Firebase console and configured with the appropriate project credentials. Secondly, specific upload and download procedures were implemented to enable the transfer of files between the device and the Firebase cloud environment. These procedures relied on Firebase's 'putFile' and 'getFile' methods, leveraging secure HTTPS connections to ensure data integrity during transmission. An important design consideration

was the organization of user files within Firebase. Each user's drawings and database files were stored within a uniquely identifiable folder structure, ensuring that user data remained logically separated and could be efficiently restored when needed. This design also supported scalability, allowing the system to manage multiple users simultaneously without file conflicts. In addition to file storage, Firebase Authentication was considered as a potential enhancement for user login and security. However, for the scope of this thesis, a simplified local authentication mechanism based on Room was maintained to avoid the additional complexity of integrating online authentication workflows. Nonetheless, Firebase Authentication remains a recommended avenue for future development, particularly in the context of secure user verification and role-based access control. The technological improvements introduced by the integration of Firebase Storage significantly enhanced the reliability, usability, and real-world applicability of the Scrap Book App. Cloud storage not only safeguarded important user data against device failure or app uninstallation but also paved the way for future features such as background synchronization, encrypted backups, and multi-device support. Overall, the extended technology stack, combining Room for local persistence and Firebase for cloud-based backups, reflects modern best practices in mobile application development. It aligns with broader industry trends favoring hybrid storage solutions to achieve both offline availability and cross-device data continuity, particularly in domains like healthcare where data resilience and security are of paramount importance.

Data Management Strategy. Efficient data management is a core aspect of modern mobile application development, especially for applications that handle user-generated content such as diaries, emotional tracking records, and multimedia files. In Android applications, two primary strategies for data storage are commonly used: local device storage and cloud-based storage solutions. Each method has its advantages, challenges, and appropriate use cases depending on the application's goals and the sensitivity of the data involved.

In the context of the Scrap Book App, initial data management was performed exclusively on the local device using internal app storage in combination with a Room database. Room, an abstraction layer over SQLite, offers a robust, type-safe way to handle structured data like mood inputs, diary entries, and metadata about user sessions. Storing data locally ensures fast access times, offline availability, and straightforward implementation. However, it also poses significant risks related to data loss. Uninstallation of the application or device replacement typically results in the loss of all stored files unless a backup mechanism is explicitly implemented.

Recognizing these risks, modern Android applications increasingly integrate cloud-based storage solutions, either as a primary or complementary storage mechanism. In this project, Firebase Storage was selected as the cloud backend for managing media files and database backups. Firebase provides scalable, secure, and highly available storage infrastructure, allowing users to upload files that persist beyond the lifespan of a single device. Integration with Android SDKs and support for asynchronous operations make Firebase particularly suitable for mobile environments where network connectivity may be intermittent.

A significant architectural decision in implementing cloud storage is determining what type of data to upload and when. In the Scrap Book App, two categories of files are targeted for cloud backup: user-created drawings and the local Room database file (database.db). Uploads are designed to be manual, triggered by the user through the interface, thereby granting users full control over which data is sent to the cloud. Similarly, restoring data is an intentional action initiated by the user, ensuring that accidental overwrites or synchronization conflicts are minimized.

Security considerations are paramount when dealing with cloud storage in healthcare-related applications. Although Firebase ensures that data is transmitted securely via HTTPS and can be isolated per user folder, the Scrap Book App currently relies on a basic local login system, without full integration of Firebase Authentication. This presents an area for future enhancement, as true end-to-end security would require authenticated and encrypted transactions on both client and server sides.

Another challenge lies in handling synchronization between the local and remote datasets. Unlike enterprise-level applications with real-time database replication, the Scrap Book App employs a simple model where the most recent upload serves as the authoritative source. No automated conflict resolution or versioning system is yet implemented, primarily due to the project's scope as a prototype for educational purposes.

In summary, data management strategies in the Scrap Book App evolved from a purely local model

to a hybrid model combining local storage for fast, offline access and cloud storage for persistence and backup. This design choice reflects broader trends in mobile application development, where balancing performance, security, and user autonomy remains a complex but essential task. The hybrid model not only enhances the resilience of the app but also prepares it for future scalability, where more sophisticated synchronization and security mechanisms could be integrated without disrupting the existing user experience.

3.2 The Existing Application

The initial version of the Scrap Book App was developed as part of a previous bachelor thesis. Its primary aim was to support children during their recovery at home after surgery or illness by providing an interactive platform where they could express emotions through drawings, diary entries, and photos. The application also allowed doctors and parents to monitor emotional progress via visual feedback and statistical summaries.

The original version of the application offered several core features. Children could use a drawing area to express emotions with colors and text, while camera integration enabled them to capture meaningful moments. A dedicated diary section allowed for daily journaling, and mood tracking using emojis provided a simple way to document emotional states. The app also included a calendar view to enable users to review their historical entries.

From a technical standpoint, the Scrap Book App was built using Java within the Android Studio development environment. Data was stored locally on the device using Room, an abstraction layer over SQLite that simplifies database management while ensuring strong compile-time validation. All user-generated content, such as drawings, photos, feelings, and diary entries, was stored exclusively on the user's device without any cloud backup.

Despite its strengths, the application exhibited several critical limitations. The most significant issue was the lack of data persistence across sessions: uninstalling the app or switching to a new device would result in the permanent loss of all user data. Furthermore, no backup or restore functionality was implemented, exposing the risk of data loss in real-world scenarios. All user data remained local-only, which limited the application's utility in medical tracking and continuity of care. Additionally, the absence of administrative access prevented medical professionals from managing or supervising user accounts.

These shortcomings significantly impacted the app's robustness and long-term usability, particularly in healthcare environments where data continuity is crucial. The aim of this thesis is to address these limitations by introducing functional extensions that preserve the original purpose of the application while significantly enhancing its resilience, usability, and maintainability. The integration of cloud storage via Firebase and the addition of administrator login functionality represent the key improvements proposed and implemented in this work.

Moreover, it became evident during the initial analysis that modern healthcare applications increasingly demand features such as cross-device synchronization, real-time data access, and robust security mechanisms — requirements that the original Scrap Book App could not fulfill in its existing form. In environments where patient data needs to be consistently available and securely managed, the absence of cloud capabilities posed a critical barrier to the app's potential adoption in clinical or therapeutic settings. Recognizing these evolving expectations in mobile health technology provided strong justification for undertaking a comprehensive extension of the application, ensuring that it could meet both current and future demands for digital tools in pediatric emotional support.

4 Implemented Extensions

In this section, we describe how the Scrap Book App was systematically extended with cloud backup and administrative modules. We first present the overall architecture and then zoom in on key implementation details, UI changes, and testing strategies. This section presents the architectural decisions, technical steps, and software components used to extend the functionality of the Scrap Book App. The main objective of this extension was to enable cloud-based data persistence through Firebase

Storage, while maintaining the application's child-friendly design and introducing administrative controls. Expanding the functionality of an existing application, particularly one intended for vulnerable user groups like children, required a careful and methodical development strategy. Unlike building a new app from scratch, extending the Scrap Book App involved respecting and preserving the existing user experience while seamlessly introducing new cloud-based features. It was essential to maintain the application's intuitive interface and emotional engagement qualities, ensuring that enhancements such as backup, restoration, and administrative functions integrated naturally without overwhelming or confusing the young target audience. This balancing act between technical innovation and user-centered design became a guiding principle throughout the implementation phase. To achieve this, an incremental development approach was adopted, where each new functionality was prototyped and tested individually before integration into the main app. This iterative process helped minimize the risk of introducing regressions or negatively impacting the existing user workflows. In particular, the introduction of asynchronous cloud operations required careful handling of user sessions and activity lifecycle events to prevent inconsistencies or crashes. Each network-dependent feature was extensively validated under different connectivity scenarios, including intermittent Wi-Fi and mobile data environments. This rigorous testing ensured that the Scrap Book App could provide a stable and predictable experience even in less-than-ideal technical conditions, which is crucial for maintaining user trust, especially when targeting sensitive user groups like pediatric patients.

The updated source code for the Scrap Book App is available at:

`https://git01lab.cs.univie.ac.at/vasilijem98/scrap-book-app-vasilije-mirkovic`

Architectural Integration Strategy. The architectural approach built upon the original local-only model by introducing a layered structure. It includes a local persistence layer (Room database), a remote cloud storage layer (Firebase Storage), and a modular application logic layer. This strategy improves scalability and maintainability and aligns with modern Android architecture guidelines.

Cloud Storage Mechanism. Firebase Storage was selected due to its scalability and seamless integration with Android Studio. Each user's drawings and local database are stored under individualized folders in the cloud. Uploads and downloads occur asynchronously to preserve the responsiveness of the app.

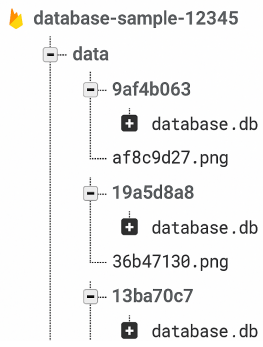


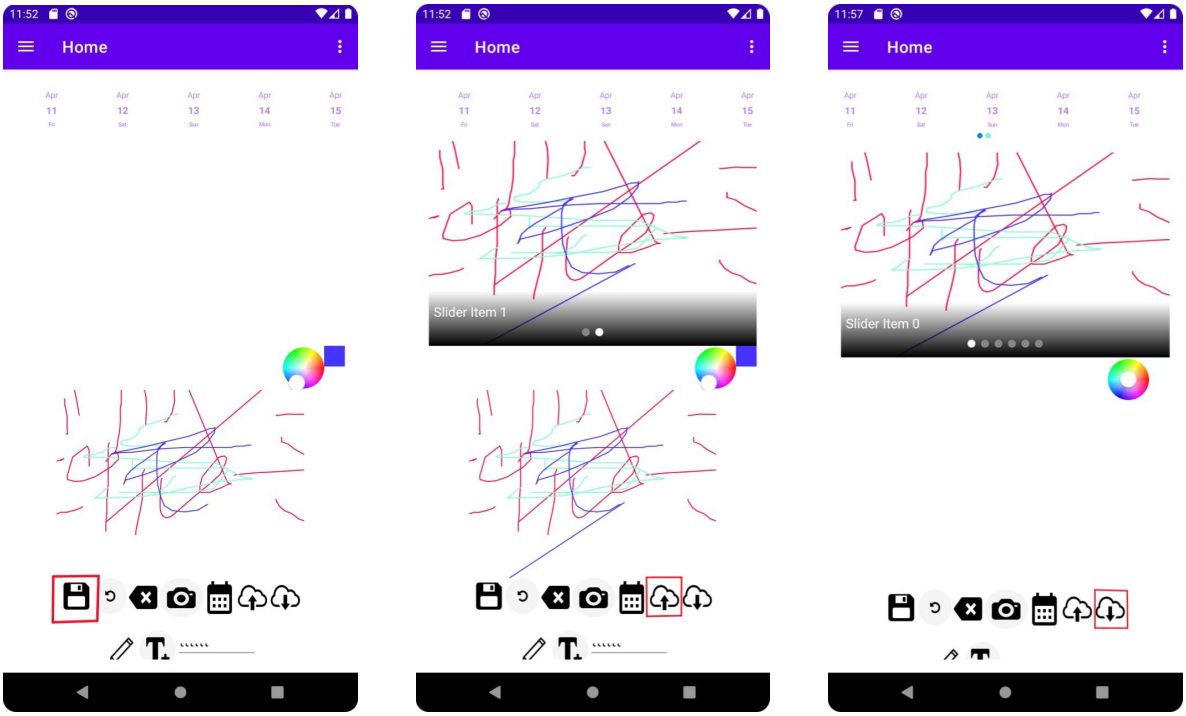
Figure 1: Firebase folder structure for storing user-specific files.

Secure File Upload and Download. The application utilizes Android's `Uri.fromFile()` method to securely reference internal files before uploading or downloading them through Firebase's APIs. Great care was taken to avoid security vulnerabilities by restricting file access within the app's private storage.

In order to initiate the cloud storage process, a dedicated upload button was integrated into the main UI. The button provides immediate user feedback upon pressing, confirming successful file transfer to Firebase Storage.

When the user selects the upload option, the application internally collects the necessary files — including user-created drawings and the local Room database — and triggers asynchronous upload tasks. Progress feedback is provided through Toast messages to inform the user about the success or

failure of the operation. By isolating upload logic into a dedicated service class, the app ensures that these background tasks do not block or slow down the main UI thread. This design not only improves performance but also preserves a smooth and responsive user experience, even on older or lower-specification devices. In addition to simple feedback, error handling mechanisms were implemented to catch potential issues such as upload failures due to network interruptions or permission errors. If an upload attempt fails, the user is immediately notified and encouraged to retry once connectivity is restored. These precautions ensure that the upload process remains reliable and user-friendly, which is crucial when dealing with emotionally significant content created by children.



(a) Locally save button (b) Upload button (c) Restore button

Figure 2: Key UI elements for local saving, uploading to Firebase, and restoring user data.

Once the restore operation is initiated, the application connects to the appropriate user-specific folder in Firebase Storage and downloads the latest available files, including both image content and the local Room database backup. Clear Toast messages inform the user about the progress and outcome of the restoration process, ensuring that the experience remains transparent and reassuring. In case of network interruptions or missing files, appropriate error messages are displayed to maintain user trust and guide further actions.

Before initiating the cloud upload, the Scrap Book App offers users the option to locally save their current session data. A dedicated ”Save Locally” button was implemented to allow children to securely store their drawings and diary entries within the app’s internal storage. This intermediate step ensures that any modifications made by the user are safely written to disk before an upload attempt is made. It also helps prevent accidental data loss in scenarios where the user chooses to postpone uploading or experiences an abrupt interruption, such as a loss of internet connection or unexpected app closure. By introducing this feature, the application enhances reliability and gives users more control over the backup process.

Local Database Management. User data is managed locally using the Room persistence library, ensuring structured and efficient data handling. Room provides an abstraction layer over SQLite, allowing the application to perform complex queries and data manipulations with less risk of errors and easier maintainability. This architecture simplifies the implementation of relationships between entities, such as linking diary entries to usernames or timestamps, and enables the use of LiveData to

reflect database changes in real time across the UI.

Backup operations were implemented to periodically upload the Room database to Firebase, minimizing potential data loss even in cases of unexpected app uninstallations or device malfunctions. These backups include the full .db file, which encapsulates all user-created entries, mood logs, and activity metadata in a compact format.

To ensure data consistency, uploads are performed during moments of application inactivity, reducing the chance of concurrency issues or database locking errors. Synchronization logic checks for ongoing write operations before initiating the backup, and upload tasks are queued to avoid race conditions. Additionally, by preserving the original Room database structure during both local operations and cloud backups, the app ensures that restored data remains coherent and accurately reflects the user's historical interactions within the application.

This dual-layer persistence strategy — local access for fast performance and cloud backups for resilience — creates a reliable and scalable foundation for pediatric health use cases, where accidental data loss could diminish therapeutic value and user trust. In future iterations, the Room schema could be expanded to support additional data types or logging fields, all while preserving backward compatibility and upgrade paths using Room's migration tools.

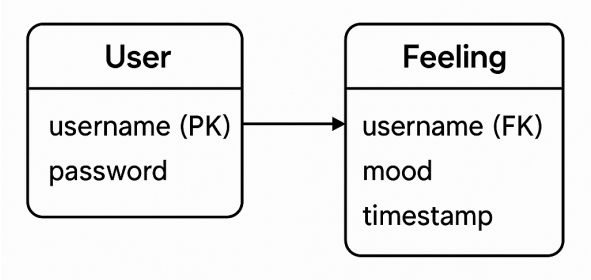
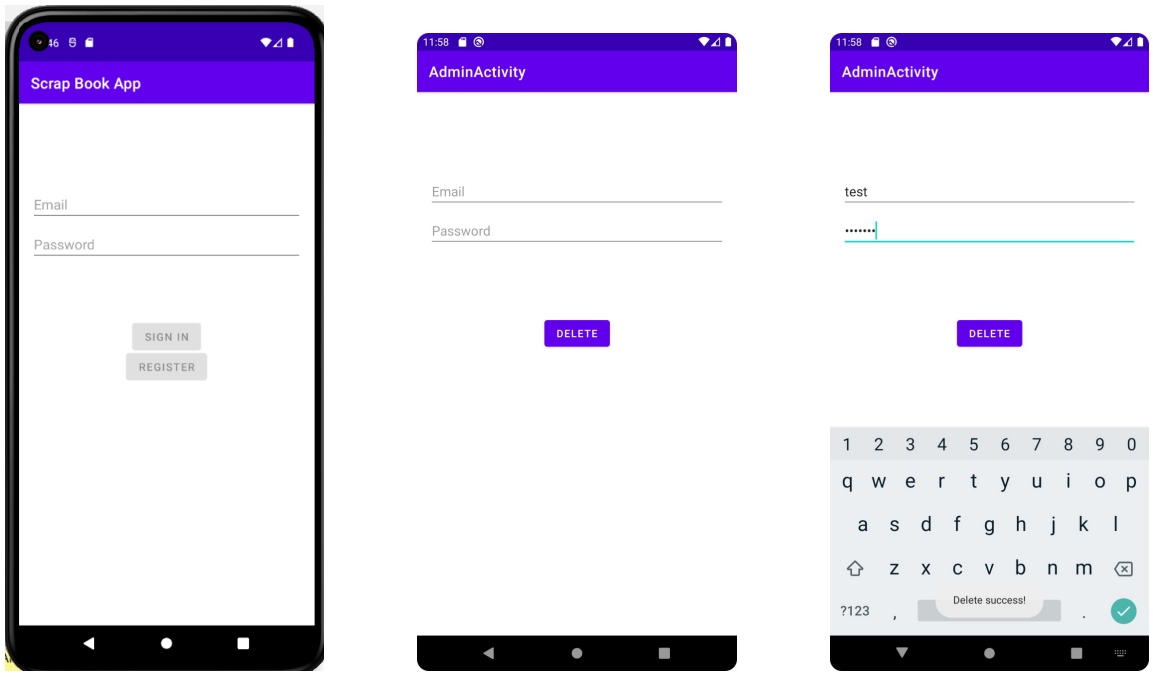


Figure 3: Structure of the local Room database. The username acts as the primary key for user data.

Login and Administrative Features. A simple local authentication system was introduced using Room. Users can register and log in to personalized sessions, allowing the application to associate diary entries, mood records, and drawings with specific user accounts. This personalized approach ensures that each child's data remains isolated and retrievable across sessions, enhancing both privacy and usability. Session tokens are stored securely and automatically expired after a period of inactivity, reducing the risk of unauthorized access on shared devices. Additionally, a "Remember Me" option was implemented to streamline repeat logins while still preserving security controls. An administrator account with static credentials (`admin/admin123`) was implemented to allow privileged access for managing and deleting inactive user accounts. Although the hardcoded login information is sufficient for demonstration purposes, it highlights an important area for future improvement, such as migrating to Firebase Authentication to support more secure, scalable, and dynamic account management. Introducing multi-factor authentication and audit logging for admin actions would further strengthen the security posture and provide traceability of critical operations. To facilitate personalized access and data organization, a simple login and registration screen was implemented. Upon first launching the application, users are prompted to either register a new account or log into an existing one. The registration process requires minimal input—primarily a username and password—to accommodate the cognitive and attention span limitations of young users. Basic input validation ensures that usernames are unique, preventing conflicts during cloud synchronization operations. Clear error messages and inline guidance assist users in correcting invalid inputs without frustration. In addition to regular user access, administrative control was introduced to allow supervisors, therapists, or medical staff to monitor and manage user accounts efficiently. The administrator interface is accessible only through the special credentials and enables functionalities such as listing all users, selecting inactive accounts, and initiating deletion processes. This administrative oversight supports better data hygiene, prevents unnecessary storage consumption in Firebase, and aligns with data minimization principles important for healthcare applications. Future enhancements could include role-based dashboards, usage analytics,

and bulk user operations to further streamline administrative workflows.

The following figures illustrate the login and registration screen, administrator interface and the user deletion workflow within the administrator panel:



(a) Login and registration screen

(b) Administrator interface

(c) Account deletion panel

Figure 4: Administrative and user access screens: registration, account management, and deletion workflow.

This separation between user and administrator roles ensures that sensitive management operations are protected from accidental access by regular users. Moreover, by enabling the deletion of outdated or inactive accounts, the system helps maintain an organized and privacy-conscious data environment.

Administrators can take corrective actions when necessary, such as cleaning up obsolete records that are no longer needed, thereby strengthening compliance with privacy regulations like GDPR. Furthermore, limiting administrative privileges to a secured login ensures that operational control remains firmly in the hands of authorized personnel, safeguarding both the integrity and confidentiality of user-generated data. This approach reflects best practices in mobile health application design, where accountability and transparency are crucial for fostering trust among users and healthcare stakeholders.

The administrator panel offers a minimalistic yet functional interface tailored specifically for ease of use in healthcare or supervised environments. Upon successful login, administrators are presented with a list of available user accounts, including relevant metadata such as registration dates or last activity timestamps. This overview facilitates the quick identification of outdated or inactive accounts, supporting efficient database management and proactive data maintenance strategies.

Administrative actions are intentionally streamlined to reduce the risk of human error. Critical operations, such as account deletions, require explicit confirmation steps to prevent accidental loss of user data. In the current prototype version, the administrator panel is restricted to essential maintenance features; however, it has been architected in a modular fashion, allowing future extensions to include additional functionalities such as password resets, account deactivation/reactivation, or even reporting features to monitor user engagement trends over time.

The importance of a dedicated administrative module lies not only in operational convenience but also in its contribution to broader ethical obligations regarding data stewardship. In pediatric health-

care applications, maintaining tight control over data access and retention is essential for protecting vulnerable users and preserving the therapeutic value of digital interventions. By embedding these controls directly into the app's architecture, the Scrap Book App demonstrates a commitment to user privacy, security, and long-term system maintainability.

Within the administrator interface, a specific function was implemented to remove inactive users from the system. Administrators can input the username of a target account into a dedicated text field and initiate the deletion process with a simple button press.

Upon confirmation, the app executes a two-step cleanup operation: it first deletes the corresponding local database entries to maintain device hygiene and then connects to Firebase Storage to remove the user's folder and associated files. This dual-layer deletion mechanism guarantees that no orphaned records or residual personal data remain either locally or in the cloud, adhering to principles of complete and secure data erasure.

Such robust account deletion processes are particularly important in health-related apps, where data minimization and the user's "right to be forgotten" are key aspects of legal compliance and ethical responsibility.

Main Screen Overview. The main screen was redesigned to place critical functions such as saving, resetting, uploading, and restoring within easy reach. The child-oriented drawing tools remained separated to preserve the app's simplicity. Clear icons and color cues were introduced to help young users quickly identify each action, while tooltips provide brief instructions on first use. A responsive layout adjusts button size and spacing based on device orientation and screen size, ensuring consistent accessibility across smartphones and tablets.

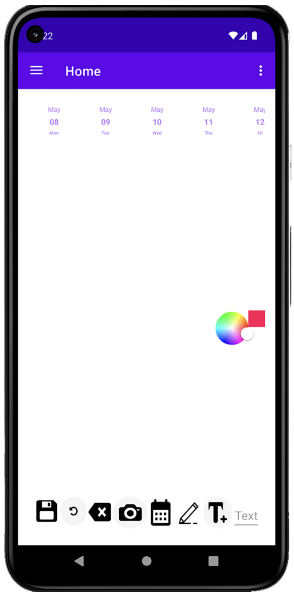


Figure 5: Main screen with core functionality grouped logically.

Cross-Device Data Consistency and Testing. Testing across different Android devices confirmed the robustness of the cloud integration. Restoration operations were successful across devices with different screen sizes and Android versions. End-to-end trials included legacy phones running Android 8.1 up to the latest Android 13 builds, ensuring backward compatibility. Performance remained consistent under varied conditions, with no data corruption observed when switching between devices or during rapid account logins. Network resilience tests—simulating handovers between Wi-Fi and cellular data—demonstrated that upload and restore processes resumed seamlessly after brief connectivity interruptions.

Error Handling and Limitations. All cloud interactions implement success and failure callbacks to handle different runtime scenarios. However, limitations such as the lack of encryption, absence of Firebase Authentication, and reliance on manual file management were identified as areas for future

improvement. Current error handling displays generic Toast messages on failure, which could be enhanced with context-sensitive alerts and retry prompts. Network timeouts and large file uploads may still result in unresponsive behavior; implementing exponential backoff, background job scheduling, and offline queuing would improve reliability. Additionally, detailed error logging and user-friendly diagnostics could assist both developers and caregivers in quickly resolving issues without technical support.

Summary. The technical improvements implemented in this project, including modular design and cloud-based data backup, significantly increased the robustness and usability of the Scrap Book App. Cloud integration required careful consideration of data integrity under variable network conditions, secure access rules, and user-friendly interactions for the young audience. While challenges such as handling unstable connections and ensuring strong authentication were identified during the development phase, the current version successfully addresses the key limitations of the original app. The groundwork laid here can serve as a strong foundation for future enhancements, including advanced security measures, background synchronization, and role-based access management.

Evaluation of Firebase as a Cloud Platform for Health Apps. Firebase, developed by Google, has established itself as a leading backend-as-a-service (BaaS) platform for mobile and web applications. Its ease of use, comprehensive documentation, and seamless integration with Android Studio make it an attractive option for student projects, prototypes, and even small-scale production deployments. In the context of the Scrap Book App, Firebase provided a relatively quick and straightforward solution for implementing essential features such as cloud storage and, potentially, authentication services. Despite its numerous advantages, an in-depth evaluation of Firebase reveals several limitations that must be considered, particularly when the platform is used in healthcare-oriented applications involving sensitive user data. One of the major strengths of Firebase is its high availability and global scalability. The platform automatically manages server resources, allowing developers to focus on frontend and application logic rather than infrastructure management. This characteristic was crucial during the development of the Scrap Book App, as it allowed the project to remain within the scope and timeline constraints of a bachelor thesis. However, Firebase's pricing model can become cost-prohibitive at scale, especially when storing large volumes of multimedia data or serving high traffic. Moreover, the lack of built-in support for advanced compliance frameworks (such as HIPAA or ISO 27001) means that additional configurations or third-party services are required for true enterprise readiness. Vendor lock-in is another concern: migrating away from Firebase to an alternative backend can involve significant effort, including rewriting authentication flows, storage rules, and API interactions. Finally, while Firebase excels at rapid prototyping, production deployments in regulated environments should incorporate rigorous security audits, custom access controls, and data residency configurations to meet strict healthcare standards. Moreover, Firebase Storage offers simple APIs for file upload and retrieval, ensuring a minimal learning curve for developers who may not have extensive experience with cloud technologies. The use of Google's authentication, security rules, and data hosting further alleviates the need for maintaining separate servers, which is a substantial benefit for academic and prototype projects. However, when considering Firebase for applications dealing with healthcare data, several concerns arise. While Firebase provides strong baseline security measures such as HTTPS transmission and role-based security rules, it does not inherently offer client-side encryption or compliance certifications like HIPAA (Health Insurance Portability and Accountability Act) unless developers use the enterprise-grade Firebase plans coupled with strict configurations. This limitation is critical because healthcare-related apps must guarantee the confidentiality, integrity, and availability of user data beyond the general practices of commercial apps. Additionally, Firebase Storage is optimized for scalability and developer productivity, but it provides limited options for complex data queries or advanced server-side operations without integrating other Google Cloud services such as Firestore or Cloud Functions. For applications like the Scrap Book App, which currently require only basic file storage, this is sufficient. Nevertheless, future expansions involving real-time collaboration, advanced analytics, or integration with external healthcare databases would require careful architectural rethinking.

Another practical concern is vendor lock-in. While Firebase accelerates development in the early stages, it tightly couples the application backend to Google's infrastructure. Migrating to another platform later would involve substantial effort, including rewriting authentication logic, reconfigur-

ing storage rules, and adapting APIs. In a healthcare context where long-term sustainability, data portability, and regulatory compliance are crucial, this dependency can be a serious disadvantage.

Finally, while Firebase supports GDPR compliance when properly configured, it places the burden of configuration and user data management largely on the developers. As such, deploying a health-focused app built on Firebase in a real-world medical setting would necessitate an extensive legal review, secure configuration, and likely the implementation of additional privacy-enhancing technologies beyond what Firebase natively offers. Developers should consider integrating client-side encryption libraries, automated data retention and deletion policies, and anonymization techniques for analytics data to further mitigate privacy risks. Regular security audits, penetration testing, and third-party compliance assessments are also recommended to ensure that all potential vulnerabilities are addressed. Moreover, fine-grained access controls and comprehensive consent management workflows must be implemented to capture, record, and enforce parental or guardian permissions throughout the application lifecycle.

In conclusion, Firebase serves as an excellent platform for prototyping healthcare applications like the Scrap Book App, especially when project timelines, budget constraints, and developer experience are limited. However, for production deployment in sensitive environments, developers must be aware of its security limitations, compliance challenges, and long-term sustainability issues. The platform's strengths in scalability and ease of use must be carefully balanced against the heightened requirements for data protection, user consent, and regulatory compliance in the healthcare sector. The Scrap Book App follows a modular architecture designed to separate core functionalities into independent layers. This separation improves maintainability, scalability, and facilitates the integration of new features, such as cloud-based backup mechanisms and administrator functionality, without disrupting existing workflows. The architecture was purposefully designed to adhere to the principles of clean architecture and separation of concerns, ensuring that each logical part of the system has a well-defined role and interface.

At the highest level, the system is divided into three main layers:

- **Presentation Layer (UI)**: Contains user-facing components, including activities and fragments for drawing, diary input, camera usage, and login. It handles user input and visual feedback, translating user actions into signals that the logic layer can process. In this layer, special attention is given to usability and accessibility, ensuring that even very young users can interact with the application intuitively.

- **Application Logic Layer (ViewModels and Controllers)**: Manages the business logic, processes user interactions, validates data, and coordinates communication between the UI and data storage. This layer ensures that all interactions follow application rules, decoupling the data management logic from the interface, which greatly simplifies maintenance and testing.

- **Data Layer**: Responsible for storing and retrieving data using Room for local persistence and Firebase Storage for cloud backup operations. It exposes a clean API to the logic layer, hiding implementation details and enabling future migration to different storage backends, such as Firestore or other cloud databases, with minimal effort.

Within these layers, specialized modules further encapsulate critical functionalities. The **LoginManager** handles user authentication and session management, the **FirebaseService** abstracts file upload and restore operations to and from the cloud, and the **AdminActivity** provides an interface for managing user accounts, including the deletion of inactive users. This modular approach also enhances the app's maintainability, as each layer operates independently and can be modified or extended without introducing unintended side effects in other parts of the system. For example, updates to the cloud storage mechanisms would primarily affect the FirebaseService module, leaving the UI and local storage untouched. Similarly, future improvements to user authentication can be localized within the LoginManager, enabling a clean upgrade path without major restructuring of the application. Moreover, the architectural design supports scalability, allowing the app to adapt to increased user demands or the integration of more advanced healthcare features over time. By isolating key responsibilities within specific modules, the Scrap Book App achieves greater robustness, flexibility, and readiness for

future developments—qualities that are particularly important when dealing with sensitive user data in dynamic healthcare environments.

In addition, this structured organization simplifies collaborative development, as different developers can work on individual components without causing conflicts or inconsistencies in the system. Such modularity not only accelerates development cycles but also improves long-term sustainability, ensuring that the app can evolve alongside technological advancements and evolving user needs. This approach also facilitates easier testing, as individual modules can be independently verified using unit tests or mocked in integration scenarios. Automated CI/CD pipelines can further leverage this separation to run targeted regression tests, reducing feedback loops and supporting rapid iteration.

Clear module boundaries encourage better documentation practices and enable domain-driven development, where teams can focus on specific features—such as authentication, data storage, or UI components—without needing detailed knowledge of unrelated parts of the codebase. Additionally, modularity fosters reuse: well-designed components (e.g., the FirebaseService or LoginManager) can be extracted into shared libraries for use across multiple projects or teams. This not only saves development effort but also standardizes best practices across the organization. It also enhances fault isolation, since bugs in one module can be contained and resolved without risking the stability of the entire application. Moreover, a modular structure supports parallel feature development, allowing multiple teams to build and deploy new functionality simultaneously. Furthermore, modular designs facilitate versioned API contracts between modules, enabling backward-compatible upgrades and smoother migrations. The clear separation of responsibilities simplifies maintenance, as deprecated or obsolete modules can be replaced independently without major code rewrites. Ultimately, this modular architecture not only accelerates development and testing but also ensures that the Scrap Book App remains adaptable, resilient, and easy to extend as new requirements emerge.

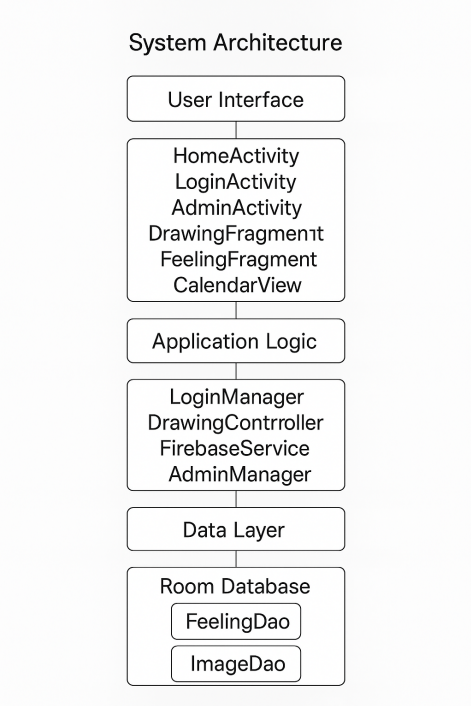


Figure 6: System Architecture of the Scrap Book App

This modular structure ensures a clean separation of concerns, facilitating easier testing, debugging, and future expansion. Additionally, it allows for potential upgrades, such as replacing the Room database with Firestore or integrating Firebase Authentication for enhanced user security, thereby providing a robust foundation for the application's long-term evolution. As the Scrap Book App handles sensitive user-generated content — such as children's drawings, diary entries, and mood-related

data — special attention was devoted to ensuring data security and user privacy at both the local and cloud storage levels. All user data is stored locally through the Room database, utilizing the app's private storage directory (‘/data/data/¡packageName¿/‘). This approach ensures that data is isolated from other applications and is automatically cleared upon app uninstallation, unless externally backed up. By relying solely on private app storage, exposure to external threats or unauthorized access is minimized.

The integration of Firebase Storage introduces additional privacy considerations, particularly related to data transmission and remote storage. Several measures were implemented to address these concerns. All data transmissions are secured using HTTPS protocols, safeguarding against man-in-the-middle attacks. Access to cloud-stored files is restricted through user-specific folder structures, leveraging Firebase Storage's built-in access control rules. Currently, user authentication is handled locally through the Room database. However, for production deployments, integrating Firebase Authentication is strongly recommended to ensure scalable and secure identity management. Importantly, no sensitive medical records or personal identifiers are stored in plaintext, either locally or in the cloud.

To further bolster security, client-side encryption of user data before upload can be introduced, ensuring that even in the unlikely event of a breach, the files remain unintelligible without the correct decryption keys. Additionally, implementing short-lived, refreshable tokens for session management will minimize the risk associated with credential theft or reuse. Comprehensive audit logs should be maintained—capturing both successful and failed access attempts—which can feed into an anomaly-detection system to flag suspicious activities in real time. Automated periodic reviews of Firebase security rules and simulated penetration tests would help detect misconfigurations before they can be exploited. Finally, a formal data lifecycle policy, leveraging Firebase's built-in lifecycle management APIs, should be enforced to automatically purge inactive or outdated user data in compliance with regulations like GDPR.

Administrator functionality is currently protected by hardcoded credentials (admin/admin123), enabling privileged users to manage account lifecycles, such as removing inactive users. While sufficient for demonstration purposes, this method lacks robust security features and is unsuitable for production environments. Future enhancements should include the integration of Firebase Authentication for secure credential verification, role-based access control (RBAC) to distinguish between user types, and the implementation of password hashing mechanisms. Furthermore, introducing multi-factor authentication (MFA) for administrative logins will significantly reduce the risk of unauthorized access, while enabling audit-grade logging of all admin operations will create a verifiable trail of critical system changes. A dedicated admin portal with IP whitelisting or VPN access requirements could add another layer of protection, ensuring that only trusted networks can perform sensitive maintenance tasks.

The Scrap Book App enables administrators to remove inactive user accounts, automatically deleting associated files from Firebase Storage. This functionality supports compliance with data minimization principles and reduces risks tied to prolonged storage of obsolete or unused user data.

To achieve full compliance with GDPR-like data protection standards and enhance the overall security posture, several improvements are proposed. These include implementing client-side encrypted backups to protect sensitive content, introducing opt-in consent mechanisms during user registration, and maintaining audit logs to monitor access and modification activities within the application.

Overall, while the current implementation provides a solid foundation for secure and private data handling, further enhancements are necessary to meet the rigorous standards expected in clinical and healthcare environments.

In addition to technical safeguards, aligning the Scrap Book App with international legal frameworks is essential for any potential deployment in clinical or educational settings. Regulations such as the General Data Protection Regulation (GDPR) in the European Union and the Children's Online Privacy Protection Act (COPPA) in the United States establish clear obligations when processing personal data, particularly for minors. These laws emphasize principles such as transparency, data minimization, limited retention, and the right to be forgotten. As such, future versions of the Scrap Book App must provide explicit, verifiable parental consent flows and clear, age-appropriate explanations of what data is collected, how it is used, and how users can request its deletion.

Moreover, implementing a basic threat model during development would help identify and mitigate security risks early. Potential threats in a mobile health context include unauthorized access to cloud

data, misconfigured access permissions in Firebase, or insecure data transmission during upload and restore operations. To address these, the app must continue to enforce secure HTTPS connections, apply strict Firebase storage rules that isolate user data, and explore integrating logging mechanisms for failed login attempts or suspicious cloud operations. Such defensive programming practices are crucial for establishing trustworthiness in health-related software.

From an ethical standpoint, developers must also consider the psychological and emotional consequences of digital self-tracking in children. Although the app is designed as a supportive tool, continuous emotional monitoring—even with the best intentions—can lead to unintended outcomes, such as over-fixation on negative feelings or feelings of being observed. To minimize this risk, emotional data should remain entirely under user control, and visual feedback within the app should emphasize encouragement rather than judgment. Working closely with child psychologists and UX designers experienced in pediatric health applications can ensure that these ethical concerns are integrated into future development cycles. Lastly, attention must be paid to long-term stewardship of emotional data. While the current app allows deletion of user accounts and associated files, future versions should introduce functionality for exporting personal records in accessible formats, enabling families to preserve meaningful content outside the application. This empowers users while supporting regulatory demands for data portability and user autonomy. Embedding these ethical values into the app's foundation ensures that technical innovation remains aligned with the emotional needs and rights of its most vulnerable users. Although the Scrap Book App significantly improves upon the original version, important technical and ethical limitations remain. Addressing these is crucial for future development, especially if the app is intended for clinical or healthcare deployment. While the extended Scrap Book App introduces significant improvements compared to its original version, it is important to acknowledge the technical limitations that still exist. Understanding these boundaries provides a realistic foundation for future development and highlights areas where additional work will be necessary to make the application suitable for real-world deployment, especially in sensitive healthcare environments. From a technical standpoint, one of the major limitations of the current version is the reliance on local authentication without external validation mechanisms. Although user login and registration are implemented through a Room database, the absence of a robust authentication service like Firebase Authentication introduces vulnerabilities, particularly in environments where device sharing is common. Without proper encryption and centralized identity management, the risk of unauthorized access to sensitive emotional records cannot be entirely mitigated.

Another significant technical constraint lies in the manual nature of cloud synchronization. The current implementation requires users to actively trigger upload and restore actions. This approach, while suitable for a prototype or academic project, may not meet the expectations of end-users in a clinical or therapeutic setting where data continuity should occur automatically in the background without explicit user intervention. Moreover, the lack of background synchronization increases the chances of data inconsistency if a user forgets to upload their latest changes.

From an ethical perspective, the Scrap Book App deals with vulnerable populations, namely children undergoing medical treatment. Although no explicit medical records are collected, the storage of mood inputs, drawings, and diary entries represents a form of sensitive psychological data. The absence of end-to-end encryption and comprehensive user consent management mechanisms introduces risks related to privacy and data protection. In a real-world healthcare context, applications that process children's data must adhere to strict legal frameworks such as GDPR in the European Union or COPPA in the United States, which require informed consent, data minimization, transparency, and the right to data erasure.

Furthermore, the hardcoded administrative credentials in the current app version, while practical for demonstration purposes, pose a substantial ethical issue regarding access control. In production systems, administrative privileges must be securely managed and audited to prevent misuse and unauthorized access to user data. A future implementation would need to include dynamic role-based access control systems and proper encryption of sensitive authentication credentials.

Lastly, the collection and analysis of user-generated emotional data introduce questions around the potential for unintended psychological impacts. Although the Scrap Book App is designed to offer a safe space for expression, any tool that captures emotional states must be carefully monitored to ensure that it does not inadvertently reinforce negative emotions or become a substitute for professional

mental health support. Addressing these technical and ethical limitations will be crucial if the Scrap Book App is to evolve from a proof-of-concept prototype to a reliable and trusted tool within pediatric healthcare environments. Future work must prioritize not only technological enhancements but also rigorous ethical evaluations to ensure that the app remains safe, supportive, and compliant with the highest standards of medical data governance. During the development of the extended Scrap Book App, multiple critical design decisions had to be made, each involving trade-offs between competing goals such as usability, security, scalability, and simplicity. This section outlines key decisions and the rationale behind them. One of the primary decisions concerned the implementation of user authentication. Initially, integrating Firebase Authentication was considered, as it would have provided a scalable and secure solution. However, due to project time constraints and the scope of a bachelor thesis, a local authentication system based on the Room database was selected. This choice enabled faster development and testing, at the cost of reduced security and scalability. Nevertheless, the architecture was designed to allow for future migration to Firebase Authentication with minimal disruption.

Another important decision involved the synchronization strategy for cloud storage. Implementing automatic background synchronization would have improved usability by ensuring continuous backups without user intervention. However, background services in Android introduce significant complexity, especially when dealing with battery optimizations and permission management across different OS versions. To maintain simplicity and stability within the project timeline, a manual upload and restore system was implemented. Users are required to explicitly trigger backup and recovery actions, which reduces complexity but places more responsibility on users to manage their data. The choice of Firebase Storage as the cloud backend also reflects a pragmatic trade-off. Although Firebase offers quick integration and high availability, it also introduces a degree of vendor lock-in and potential limitations regarding advanced compliance requirements, such as HIPAA certification for healthcare apps. Nevertheless, considering the project's academic and prototype nature, Firebase was deemed an appropriate and effective solution. From a user interface (UI) perspective, simplicity and child-friendliness were prioritized over the inclusion of advanced features. All essential functions — such as saving drawings, uploading files, and restoring data — were grouped logically on the main screen, using intuitive icons. Administrative functions were placed behind dedicated login screens to avoid confusing young users. While this separation maintains ease of use, it also limits discoverability for power users or medical staff without clear documentation. Finally, security decisions had to balance protection with feasibility. Client-side encryption was not implemented in this version, primarily to avoid excessive technical complexity. Instead, secure HTTPS transmission and Firebase security rules were relied upon to protect user data during upload and download operations. Future improvements will aim to incorporate stronger encryption techniques to meet healthcare compliance standards. In conclusion, each design decision was the result of careful evaluation of trade-offs between development time, technical risk, usability, and security. While certain compromises were necessary, the overall approach prioritized creating a reliable, user-friendly, and extensible platform suitable for further academic and clinical development. Throughout the development and extension of the Scrap Book App, several technical, design, and organizational challenges emerged. Successfully addressing these obstacles contributed significantly to both the robustness of the final product and my personal and professional development as a software engineer.

Technical Integration Challenges. Integrating Firebase Storage into a legacy local-only application was a significant technical hurdle. Adapting the file management logic for asynchronous cloud operations, handling secure `Uri` objects, and configuring access permissions demanded thorough understanding of both Firebase's architecture and modern Android security policies. Managing the Room database posed additional complexities. Uploading and restoring the database required careful synchronization to prevent data corruption, particularly when the app was active. Testing database interactions across updates also highlighted the importance of consistent schema design and safe concurrency handling.

Design and Usability Challenges. Introducing administrative features while preserving a child-friendly user interface required careful design decisions. Administrative functionalities were hidden behind a secure login to prevent accidental access by children, while cloud backup and restore options were integrated seamlessly alongside traditional actions like saving and resetting drawings. Multiple UI iterations were necessary to balance feature richness with simplicity, ensuring that the target audience

— young children — could engage with the app independently and intuitively. Icons, layout spacing, and feedback messages were refined iteratively to improve clarity and accessibility for the intended user base.

Cross-Device Testing and Real-World Validation. Testing on various Android devices and network conditions revealed critical insights. Emulators often failed to accurately simulate real-world performance issues such as permission prompts, network interruptions, and UI scaling inconsistencies. Testing on physical devices running Android 10 to 13 exposed edge cases that were addressed through improved error handling and adaptive layouts. Real-world testing also brought attention to scenarios such as intermittent connectivity and device storage limitations, prompting the inclusion of informative error messages and fallback behaviors that preserved user trust.

Lessons Learned and Personal Development. This project reinforced key principles of modular software design, security-by-design thinking, and user-centric development. Building on a layered architecture made it easier to extend the application without compromising existing functionality.

On a personal level, the project strengthened my technical expertise in cloud integration, mobile security, and database management. Equally importantly, it deepened my appreciation for the ethical and social dimensions of technology, particularly when designing for vulnerable populations such as recovering children.

The experience also honed my project management skills, teaching me how to organize complex tasks, adapt to unexpected challenges, and balance technical, usability, and ethical requirements within a constrained timeline. Through iterative testing, collaborative feedback, and continuous improvement, I learned how to build sustainable and impactful software in a real-world context.

In summary, this Bachelor thesis was not just an exercise in software development but a transformative learning experience that prepared me for the professional demands of the software engineering field. While the extension of the Scrap Book App demonstrated technical and functional success, it also raised important questions regarding the ethical use of technology, particularly in sensitive areas like pediatric emotional health. This section critically reflects on some of the broader challenges and ethical considerations identified during the project.

Emotional Data and Child Privacy. The Scrap Book App collects emotionally significant data from children, including mood inputs, drawings, and diary entries. Although no explicit medical data is gathered, the emotional profiles created through repeated use can reveal sensitive psychological information. In real-world deployment, strict consent management would be necessary to ensure that both children and their legal guardians understand what data is collected, how it is used, and what rights they have regarding access and deletion. Ethical guidelines such as the General Data Protection Regulation (GDPR) and the Children's Online Privacy Protection Act (COPPA) emphasize the importance of clear, age-appropriate communication of privacy policies. The current implementation assumes that children and parents understand the implications of cloud storage, which may not always be the case. Future versions of the app should integrate detailed consent flows, simplified privacy explanations for minors, and opt-in data sharing policies.

Potential Psychological Impacts. Although the Scrap Book App is intended to support emotional expression and well-being, the regular tracking of emotional states can also have unintended psychological effects. For instance, children might over-focus on their negative emotions, leading to reinforcement of distress rather than its alleviation. Similarly, constant external monitoring of emotional outputs by parents or therapists could diminish a child's sense of autonomy or privacy. Additionally, frequent prompts to record moods may induce self-monitoring fatigue, causing children to view the process as a chore rather than a therapeutic activity. There is also a risk that comparing one's emotional entries with normative trends or peer data could foster negative self-comparison or anxiety. It is crucial that digital emotional support tools are positioned as complementary to, rather than replacements for, professional psychological care. Healthcare professionals should be involved in the deployment and interpretation of app-collected data to avoid misdiagnosis or inappropriate responses based solely on digital records.

Data Ownership and Long-Term Responsibility. In educational prototypes like this project, data ownership often remains implicitly tied to the user device or Firebase account. However, in production systems, clear policies must define who owns the emotional data — the child, their parents, or the healthcare institution. The long-term storage of such sensitive information carries ethical

obligations regarding retention periods, access rights, and secure deletion protocols. The current implementation does not address long-term storage limits or data portability (e.g., allowing families to export a child's emotional history). In the future, implementing export and delete functionalities that empower users to control their own data would be essential.

Balancing Simplicity and Security. The design of the Scrap Book App favors simplicity to support independent use by young children. However, in professional healthcare environments, simplicity must be balanced against robust security mechanisms. Future iterations should explore how authentication, encryption, and role-based access control can be implemented without compromising the user experience for children. For example, biometric authentication (such as fingerprint unlocking) could offer a child-friendly and secure method for identity verification without burdening users with passwords. Such technologies, when properly designed, could bridge the gap between usability and data protection.

Conclusion of Critical Reflection. Overall, the Scrap Book App extension project highlights the inherent tensions between user-centered design, emotional sensitivity, and technical robustness in pediatric digital health solutions. Future work must not only refine the application's technical features but also deepen its ethical framework to ensure that the app remains supportive, empowering, and respectful of its vulnerable user base. Ethical development is not an optional add-on but a fundamental component of designing technology for children. This project demonstrates the importance of embedding ethical thinking at every stage of the software development lifecycle, particularly when dealing with emotional or health-related data.

Potential Clinical Applications and Future Research Directions. The Scrap Book App, although initially designed as a prototype for academic purposes, presents promising potential for real-world clinical applications, especially in the field of pediatric recovery, psychological therapy, and family-centered care.

Use in Pediatric Recovery and Emotional Therapy. The app could serve as a valuable support tool for children recovering from surgeries, chronic illnesses, or trauma. By providing a non-verbal medium to express feelings through drawings and diary entries, it can assist healthcare providers in gaining insights into a child's emotional well-being over time.

In therapeutic contexts, the Scrap Book App could complement existing treatment plans by:

- Helping psychologists and therapists monitor emotional patterns and behavior.
- Supporting early detection of anxiety, depression, or emotional withdrawal during long-term treatment.
- Encouraging children to externalize thoughts and emotions that might otherwise remain hidden.

Furthermore, the app's simple design makes it suitable for home use, allowing parents and caregivers to track their child's emotional journey during recovery phases.

Future Research Opportunities. Several future research directions can build upon the current project:

- **Clinical Trials**: Conducting pilot studies with real patients to validate the app's effectiveness in supporting emotional recovery and improving communication between children, parents, and medical staff.
- **Integration of Advanced Analytics**: Implementing machine learning algorithms to detect mood trends or emotional fluctuations based on children's drawings and diary entries.
- **Gamification Elements**: Adding interactive, reward-based elements to increase children's motivation to use the app regularly.
- **Multi-User Environments**: Extending the application to allow family members or therapists to collaboratively monitor a child's progress via linked accounts.
- **Wearable Device Integration**: Combining physiological data (e.g., heart rate, sleep patterns) with self-reported emotional states for a holistic view of well-being.

- **Privacy-Preserving Techniques**: Exploring federated learning and homomorphic encryption to protect sensitive user data while enabling intelligent feature development.

In summary, future research has the potential to transform the Scrap Book App into a dynamic, intelligent, and ethically responsible companion for pediatric emotional support. Strategic development across clinical validation, intelligent features, and enhanced data protection will be essential in realizing its full potential within healthcare environments.

5 Evaluation

In this section, we assess the effectiveness, reliability, and usability of the extended Scrap Book App. First, we describe our manual testing methodology and the specific scenarios under which cloud backup and restore operations were validated. We then report on cross-device compatibility tests, hardware performance observations, and user feedback mechanisms. Finally, we compare our solution against existing pediatric mHealth applications to highlight its unique advantages in terms of emotional expressiveness and data resilience.

Testing and comparison. To assess the reliability and robustness of the newly implemented functionalities, a comprehensive set of manual tests was conducted. The primary focus was on verifying the cloud integration and restore mechanisms, ensuring that users could securely store and recover their data under real-world conditions. Testing scenarios encompassed various typical use cases. Firstly, the application was uninstalled and reinstalled to validate that previously uploaded files could be successfully restored from Firebase Storage upon re-login. Secondly, device migration was simulated by installing the app on multiple Android devices and confirming that user-specific data was retrievable using the same credentials. Furthermore, the app's resilience was tested by repeatedly closing and restarting it to verify session continuity and data integrity. Although offline usage was not a primary focus, the upload and restore functionalities were found to operate reliably under standard Wi-Fi conditions.

Tests were performed on a range of hardware platforms, including the author's personal laptop and smartphone, as well as a friend's device to simulate a typical end-user experience. Across all configurations, the system demonstrated consistent behavior and reliability.

The results of the testing phase were highly positive. Both the upload and restore processes performed as expected, without encountering any critical bugs. In rare instances, minor performance issues were observed on older laptop hardware; however, these were attributed to device limitations rather than flaws in the application's logic.

User feedback within the app was effectively communicated through Toast messages, providing clear status updates for key actions such as successful uploads of *.png* files and *database.db* backups, confirmation of restore completions, and successful login or registration events. These immediate notifications contributed to an intuitive and user-friendly experience during critical interactions. Regarding cross-device compatibility, the application performed reliably on various Android smartphones with different screen sizes and operating system versions. Thanks to the cloud-based architecture, users could seamlessly retrieve their stored data across different devices, provided they logged in with the correct account credentials. This outcome confirms the effectiveness of the implemented cloud backup and restoration functionalities in achieving platform independence and enhancing user confidence in data persistence. Overall, the testing phase confirmed that the extended Scrap Book App meets its intended functional requirements, providing robust data persistence, intuitive user interaction, and reliable cross-device performance. These results validate the feasibility of cloud-based backup integration within a child-focused healthcare application and provide a strong foundation for future enhancements.

Testing Methodology and Future User Studies. The manual testing performed for this project primarily focused on technical correctness, cross-device compatibility, and cloud synchronization reliability. While these tests confirmed the robustness of core functionalities, they represent only the first phase of a complete validation process for a healthcare-related mobile application. Future user studies are recommended to systematically evaluate the usability, emotional impact, and therapeutic value of the Scrap Book App. These studies should involve structured observational sessions with pediatric participants, complemented by caregiver and clinician interviews to capture qualitative insights.

Quantitative metrics—such as task completion times, error rates, and engagement frequency—will provide objective measures of usability, while validated psychological scales can assess changes in emotional well-being over time. Additionally, A/B testing of different interface designs and feature sets could help optimize the app's effectiveness and ensure that the final product meets both clinical and user-centered goals.

- **Participant Recruitment**: Recruiting a diverse group of child participants recovering from various medical conditions, alongside parents and healthcare providers. - **Controlled Environments**: Deploying the application in clinical or therapeutic settings to simulate real-world usage conditions. - **Usability Testing**: Observing how children interact with the app, identifying usability barriers, and collecting qualitative feedback through interviews and surveys. - **Emotional Impact Measurement**: Using validated psychological scales, such as the Pediatric Quality of Life Inventory (PedsQL) or the Strengths and Difficulties Questionnaire (SDQ), to assess changes in emotional well-being before and after prolonged app usage. - **Engagement Metrics**: Tracking frequency and patterns of app usage to evaluate sustained engagement and identify features that drive or hinder continued use. - **Feedback from Healthcare Professionals**: Collecting input from therapists and doctors to understand how app-generated emotional data can be meaningfully integrated into existing treatment workflows.

Such user-centered evaluations would not only validate the app's effectiveness but also guide further refinements in user interface design, feature prioritization, and data management practices.

In the future, longitudinal studies could be conducted to observe whether consistent use of the Scrap Book App correlates with improved emotional resilience, better communication with caregivers, or faster recovery times in pediatric patients. Such evidence-based validation would greatly strengthen the app's position as a serious tool within digital health ecosystems.

Overall, while technical testing confirmed that the Scrap Book App meets its functional requirements, true validation must involve its intended end-users — children — and the healthcare professionals who support them.

Comparison with Related Applications. Mobile health (mHealth) applications aimed at supporting children during recovery are becoming increasingly common. However, a closer comparison reveals that the Scrap Book App provides a unique combination of emotional expressiveness, usability, and data resilience compared to other available tools.

Many pediatric mHealth applications, such as *Pain Squad* and *My Hospital Diary*, focus primarily on collecting structured feedback through mood scales, questionnaires, or daily check-ins. While effective for gathering quantitative health data, they often offer limited opportunities for free emotional expression. In contrast, the Scrap Book App emphasizes creativity by allowing children to draw, write diary entries, and take photos, enabling them to express complex emotions in a natural and personalized way.

An illustrative example is the mobile health application *Pain Squad*, developed for pediatric cancer patients to help them track pain levels and medication use. While the app employs a gamified interface and includes reminders to encourage daily reporting, its primary focus is on quantitative data collection, without options for free-form emotional expression or multimedia journaling.

Meanwhile, the Scrap Book App avoids rigid structures by encouraging spontaneous drawing, photo capture, and diary writing, aligning closely with therapeutic strategies in pediatric psychology. This open-ended design fosters richer emotional insights and promotes self-motivated engagement, critical elements for supporting mental health during recovery.

Compared to many existing applications that store user data only locally, the Scrap Book App integrates Firebase Storage by default, ensuring that drawings, diary entries, and mood inputs can be securely backed up and restored across devices. This significantly reduces the risk of data loss — an area often neglected by similar apps.

Key comparative advantages of the Scrap Book App include:

- Native integration of cloud-based backup and restore functionalities.
- User-specific cloud storage with secure access rules.
- Child-centered design prioritizing ease of use and emotional engagement.
- Basic administrative control for user account management.

Another relevant example is *My Hospital Diary*, an app providing structured templates for documenting hospital experiences. While helpful for standardizing feedback, it lacks personalized emotional tracking and robust cloud backup options.

Furthermore, many commercial emotional health apps, such as *MoodMeter* and *Daylio*, primarily target adolescents or adults and rely heavily on written input or mood scales. In contrast, the Scrap Book App's intuitive drawing tools, emoji-based mood tracking, and minimal text entry requirements are specifically optimized for young children's cognitive development and interaction styles.

From a technical standpoint, the Scrap Book App adopts a hybrid model of local and cloud storage, ensuring offline functionality and enabling secure backup without burdening users with constant internet dependency. While some advanced features like end-to-end encryption and dynamic role-based access are planned for future iterations, the current version already offers a thoughtful balance between security, usability, and emotional expressiveness.

In conclusion, the Scrap Book App successfully bridges gaps present in existing pediatric mHealth solutions by combining free emotional expression, robust data management, and a child-friendly interface. These thoughtful design choices distinguish it from conventional emotional tracking apps, making it uniquely suited for supporting the emotional recovery journey of young patients.

6 Conclusion and Future Work

This section summarizes the key findings of the thesis, reflects on the implications of the implemented extensions, and outlines directions for future development. This Bachelor thesis presented the development and functional extension of the existing Scrap Book App, focusing on improving data persistence, usability, and long-term applicability in healthcare-related scenarios. The integration of a robust cloud-based backup and restore mechanism using **Firebase Storage** enabled the application to support secure file storage and retrieval across devices, even after app uninstallation or device replacement.

In addition, a dedicated **admin interface** was implemented, allowing authorized personnel — such as doctors, therapists, or caregivers — to manage user accounts and remove inactive users. Particular emphasis was placed on integrating these new features seamlessly into the child-oriented user interface, ensuring that usability for young users remained unaffected by the increased complexity.

The extended version of the application was subjected to comprehensive manual testing across multiple Android devices and operating system versions. These tests confirmed that the implemented functionalities perform reliably under various real-world conditions, including unstable network environments and cross-device data recovery.

Overall, the enhanced Scrap Book App fulfills its intended role as a digital companion supporting emotional expression and post-operative care for children. By preserving children's diary entries, drawings, and emotional states in a cloud-secured environment, the app facilitates a more holistic and continuous monitoring of pediatric recovery processes.

Outlook. While the current version represents a substantial improvement over the original implementation, several areas for future enhancement have been identified, based on both the development experience and emerging industry standards:

- **Migration to Firebase Authentication**: Integrating Firebase Authentication would enable secure, scalable, and centralized identity management, reducing reliance on local login systems and improving overall security.
- **Background synchronization**: Enabling automatic upload and restore processes, rather than relying solely on manual user actions, would further enhance data safety and user convenience.
- **End-to-end encryption**: Introducing client-side encryption before uploading files to Firebase would strengthen compliance with healthcare data protection regulations (such as GDPR and HIPAA).

- **Advanced logging and monitoring**: Implementing enhanced error handling, crash reporting (e.g., Firebase Crashlytics), and offline data caching mechanisms would improve system robustness, particularly under poor connectivity or storage failures.
- **Role-based access control (RBAC)**: Future versions should implement differentiated user roles, allowing finer-grained access management between child users, parents, and healthcare providers.
- **Integration with medical platforms**: Expanding the app's capabilities to integrate with electronic health record (EHR) systems or pediatric monitoring tools could increase its clinical value.

Beyond technical improvements, future work will explore conducting usability studies with real pediatric patients and medical staff to validate the app's impact on emotional well-being and therapeutic outcomes.

Moreover, lessons learned during this project — including modular software design, data privacy considerations, and human-centered UX principles — provide a solid foundation for tackling more complex healthcare software challenges in future academic or professional work.

In conclusion, this thesis illustrates how cloud technologies, when thoughtfully integrated with mobile applications, can contribute meaningfully to sensitive domains such as pediatric healthcare. It also demonstrates how even small-scale academic projects can be designed with scalability, security, and long-term sustainability in mind.

The full implementation and codebase of the extended version is publicly available via GitLab:

`https://git01lab.cs.univie.ac.at/vasilijem98/scrap-book-app-vasilije-mirkovic`

While the current version of the Scrap Book App successfully addresses major challenges such as data persistence, child-centered interaction, and administrative account management, several opportunities for further improvement and research remain. One of the most immediate enhancements would be the full integration of Firebase Authentication. Replacing the local login system with a scalable and secure authentication method would not only protect user accounts more effectively but also open possibilities for additional user roles, password recovery options, and access control levels. This transition would align the app more closely with professional standards for mHealth solutions, where data protection and user verification are essential. Another promising area for development is the implementation of background synchronization. Allowing automatic and periodic backups of user data without manual intervention would greatly improve usability, particularly for young users who might forget to trigger uploads themselves. Such a feature would also enhance data consistency and minimize the risk of loss during unexpected app closures or device malfunctions. From a user experience (UX) perspective, the app could benefit from a more dynamic emotional tracking interface. For example, instead of relying solely on emoji-based inputs or free drawing, future versions could explore storytelling templates, mood-based journaling prompts, or even gamification elements that reward regular participation and reflection. These additions could increase engagement, making emotional tracking an enjoyable daily habit for children rather than a clinical task. In terms of research, further studies could investigate the therapeutic value of the Scrap Book App in controlled environments, such as pediatric hospital wards or rehabilitation centers. Clinical trials or longitudinal studies could measure its impact on psychological resilience, treatment adherence, or psychological recovery compared to traditional methods. Data encryption, both in transit and at rest, should also be a focus of future work. Although Firebase provides secure transmission via HTTPS, client-side encryption before uploading sensitive files (such as diary entries) would offer an additional layer of protection, especially under GDPR and COPPA regulations. Implementing encryption libraries tailored for mobile apps would not significantly degrade performance while enhancing privacy guarantees. Finally, broader interoperability with other healthcare systems could be explored. Integrating the Scrap Book App with electronic health record (EHR) systems via standardized APIs (e.g., FHIR) would enable seamless sharing of relevant emotional tracking data with authorized healthcare providers. This would turn the app into not just a standalone tool but part of a larger digital healthcare ecosystem. In conclusion, the Scrap Book App holds significant potential beyond its current implementation. Through continued technical

enhancement, expanded user engagement strategies, and empirical research validation, it can evolve from a promising prototype into a vital tool supporting emotional well-being in pediatric care.

Potential Clinical Applications and Future Research Directions. The Scrap Book App, although initially designed as a prototype for academic purposes, presents promising potential for real-world clinical applications, especially in the field of pediatric recovery, psychological therapy, and family-centered care. The app could serve as a valuable support tool for children recovering from surgeries, chronic illnesses, or trauma. By providing a non-verbal medium to express feelings through drawings and diary entries, it can assist healthcare providers in gaining insights into a child's emotional well-being over time.

In therapeutic contexts, the Scrap Book App could complement existing treatment plans by:

- Helping psychologists and therapists monitor emotional patterns and behavior.
- Supporting early detection of anxiety, depression, or emotional withdrawal during long-term treatment.
- Encouraging children to externalize thoughts and emotions that might otherwise remain hidden.

Furthermore, the app's simple design makes it suitable for home use, allowing parents and caregivers to track their child's emotional journey during recovery phases.

Future Research Opportunities. Several future research directions can build upon the current project:

- **Clinical Trials**: Conducting pilot studies with real patients to validate the app's effectiveness in supporting emotional recovery and improving communication between children, parents, and medical staff.
- **Integration of Advanced Analytics**: Implementing machine learning algorithms to detect mood trends or emotional fluctuations based on children's drawings and diary entries.
- **Gamification Elements**: Adding interactive, reward-based elements to increase children's motivation to use the app regularly.
- **Multi-User Environments**: Extending the application to allow family members or therapists to collaboratively monitor a child's progress via linked accounts.
- **Wearable Device Integration**: Combining physiological data (e.g., heart rate, sleep patterns) with self-reported emotional states for a holistic view of well-being.
- **Privacy-Preserving Techniques**: Exploring federated learning and homomorphic encryption to protect sensitive user data while enabling intelligent feature development.

In summary, future research has the potential to transform the Scrap Book App into a dynamic, intelligent, and ethically responsible companion for pediatric emotional support. By incorporating machine learning algorithms to analyze drawing patterns and diary entries, the app could proactively identify early signs of distress or mood fluctuations, enabling timely interventions. Integrating real-time analytics and personalized feedback loops would further empower caregivers and clinicians to tailor therapeutic strategies based on objective emotional metrics. At the same time, rigorous clinical validation—through randomized controlled trials and longitudinal observational studies—will be essential to quantify the app's impact on psychological resilience and treatment adherence. Equally important is the ongoing enhancement of data protection measures: future work should explore homomorphic encryption, federated learning, and blockchain-based audit trails to ensure that sensitive emotional records remain secure and private, even in multi-stakeholder environments. Finally, establishing clear ethical frameworks and regulatory compliance processes—aligned with GDPR, HIPAA, and emerging pediatric data protection guidelines—will safeguard user rights and foster trust among families and healthcare institutions. Strategic development across these dimensions will be critical in realizing the Scrap Book App's full potential as a transformative tool in pediatric healthcare ecosystems.

7 Bibliography

1. V. Mirkovic. Source code of the extended Scrap Book App (2025). Available online at: `https://git01lab.cs.univie.ac.at/vasilijem98/scrap-book-app-vasilije-mirkovic`.

2. A. Ahtinen, M. Kinnula, and J. Häkkilä (2018). "Meaningful digital technology for children in hospital settings: Supporting communication and emotional expression." In *Proceedings of the 17th ACM Conference on Interaction Design and Children*, pp. 207–218.

3. A. Martínez-Millana, C. Fernàndez-Llatas, and V. Traver (2016). "Mobile health applications for the most prevalent conditions: A systematic review." In *Journal of Medical Systems*, 40(6), 178.

4. B. S. Fjeldsoe, Y. D. Miller, J. O'Brien, and A. L. Marshall (2020). "Behavioral responses to a mobile phone delivered health intervention for adolescent girls." *mHealth*, 6, 50.

5. A. N. Antle, C. Frauenberger, and W. Odom (2017). "Designing interactive systems for children with special needs: Beyond accessibility." In *Proceedings of the 2017 Conference on Interaction Design and Children*, pp. 1–12.

6. Stinson, J. N., Jibb, L. A., Nguyen, C., Nathan, P. C., Maloney, A. M., Dupuis, L. L., ... & Gupta, A. (2013). Development and testing of a multidimensional iPhone pain assessment application for adolescents with cancer. Journal of Medical Internet Research, 15(3), e51.

7. Hall, C. L., Newell, K., Taylor, J., Sayal, K., & Hollis, C. (2019). Service user and provider perspectives of a prototype psychotherapeutic mobile app for adolescent depression: A qualitative study. BMC Psychiatry, 19(1), 1-11.