INITIAL DESIGN

PROJECT DOCUMENTATION

StuPass

Student-to-Student Second-hand Marketplace Platform

C2C Mobile Application Development

Flutter | Rust/Axum/SeaORM

Extracurricular Project

January 2025

Note: This Table of Contents is generated via field codes. To ensure page number accuracy after editing, please right-click the TOC and select "Update Field."

1. Executive Summary

StuPass represents a comprehensive peer-to-peer marketplace platform designed exclusively for university students to buy, sell, and exchange pre-owned items within their campus communities. Drawing inspiration from the success of established C2C platforms while addressing the unique needs and constraints of the student demographic, StuPass creates a trusted ecosystem where students can safely transact with verified peers from their own academic institutions. The platform integrates mobile-first design principles with real-time communication capabilities, creating an end-to-end transaction experience that emphasizes safety, convenience, and community building.

The core value proposition of StuPass lies in its student-exclusive verification system, which ensures that all platform participants are legitimate university students with valid academic credentials. By requiring student ID verification during registration and maintaining institutional affiliation throughout the user lifecycle, the platform creates a naturally trusted environment that significantly reduces the fraud and safety concerns inherent in open marketplaces. This approach transforms the abstract trust challenges of anonymous C2C transactions into concrete, verifiable relationships within a bounded academic community.

The platform incorporates sophisticated features including multi-image product listings with detailed condition assessments, an integrated Storage system for personal inventory management and streamlined transaction history, real-time instant messaging with location sharing capabilities, integrated video calling for remote product inspection, and a comprehensive reputation system built on transaction-based reviews and ratings. The technical architecture leverages Flutter for cross-platform mobile development and Rust with Axum for backend services, utilizing SeaORM for robust database management. This technology stack was selected to deliver exceptional performance, type safety, and maintainability while supporting the real-time communication requirements essential to the platform's value proposition.

2. Project Background and Context

2.1 The Student Marketplace Landscape

University students represent a unique demographic with distinct consumption patterns, financial constraints, and lifestyle needs that create natural opportunities for second-hand commerce. Each academic year, students acquire textbooks, electronics, furniture, clothing, and various supplies that often become surplus as courses conclude, living arrangements change, or graduation approaches. Traditional channels for disposing of these items—including social media groups, bulletin board postings, and word-of-mouth arrangements—suffer from fragmentation, limited reach, and significant trust barriers that impede efficient market matching between buyers and sellers.

The emergence of general-purpose C2C platforms such as Facebook Marketplace, Carousell, and OfferUp has demonstrated strong consumer demand for peer-to-peer commerce, yet these platforms present specific challenges for student users. The open nature of these marketplaces means students must transact with unknown individuals who may not share their community values or geographic proximity. Safety concerns, scheduling difficulties, and the absence of institutional context create friction that discourages participation or leads to suboptimal transaction experiences. Students frequently report anxiety about meeting strangers, uncertainty about product authenticity, and frustration with no-show appointments.

Campus-specific solutions have emerged at various institutions, typically taking the form of Facebook groups, Discord servers, or simple web forums. While these approaches succeed in constraining participation to a specific community, they lack the infrastructure for safe transactions, efficient discovery, and reputation building that characterize mature marketplace platforms. The absence of dedicated features for product cataloguing, secure communication, and transaction tracking leaves users to manage these aspects through ad-hoc methods, resulting in lost messages, forgotten commitments, and unresolved disputes that undermine trust in the community marketplace concept.

2.2 Project Objectives

The primary objectives of the StuPass project encompass both technical achievement and practical utility in addressing the genuine needs of student communities:

1. Design and implement a robust client-server architecture that delivers a seamless mobile application experience for student users while maintaining secure backend services for authentication, data management, and real-time communication. The architecture must support both iOS and Android platforms through a unified Flutter codebase while leveraging Rust's performance characteristics and memory safety guarantees for server-side operations.
2. Establish a comprehensive student verification system that validates academic affiliation during registration and maintains verified status throughout the user lifecycle. This verification layer forms the foundation of platform trust by ensuring that all participants are legitimate members of recognized academic institutions, dramatically reducing the fraud risks inherent in open marketplaces.
3. Develop a feature-rich marketplace experience that supports the complete transaction lifecycle from product discovery through final exchange. Core capabilities include multi-image product listings, category-based browsing, keyword search, condition assessment, pricing negotiation, and transaction completion tracking. The interface must balance comprehensive functionality with intuitive navigation suitable for mobile-first usage patterns.

4. Implement integrated communication features that enable safe and convenient coordination between transacting parties. Real-time messaging, location sharing, and video calling capabilities allow buyers and sellers to communicate effectively without exposing personal contact information, addressing privacy concerns while facilitating the coordination essential to completing in-person exchanges.
5. Create a reputation and trust system that rewards positive transaction behaviour and provides visibility into user reliability. Star ratings, written reviews, and transaction history combine to create reputation profiles that help users make informed decisions about potential trading partners, building community trust through transparent accountability mechanisms.

3. Problem Statement

Students seeking to buy or sell pre-owned items within their campus communities face significant challenges that discourage participation in second-hand commerce and result in substantial value destruction as usable items are discarded rather than transferred to new owners. These challenges span trust and safety concerns, communication friction, discovery inefficiencies, and accountability gaps that collectively create a hostile environment for the peer-to-peer transactions that should flourish within tightly-knit academic communities.

3.1 Trust and Safety Concerns

The fundamental challenge facing student marketplace participation is the trust deficit that emerges when transacting with unknown individuals. Unlike commercial retail transactions where institutional reputation provides assurance, peer-to-peer exchanges involve personal meetings between strangers whose identities, intentions, and reliability are unknown to each other. Students, particularly those new to campus or unfamiliar with local areas, may feel uncomfortable arranging meetings with unknown parties, leading to avoidance of marketplace participation entirely. This trust gap is particularly acute for high-value items where the financial stakes amplify concerns about fraud, theft, or misrepresentation.

General-purpose marketplace platforms attempt to address trust through user profiles, ratings, and messaging features, but these mechanisms prove insufficient for the student context. Profile information on open platforms may be fabricated or irrelevant to transaction reliability, ratings may reflect transactions outside the student community, and messaging systems cannot verify that the person on the other end is who they claim to be. The absence of institutional anchoring means there is no shared context, no common accountability structure, and no recourse mechanism when transactions go wrong. Students who experience negative interactions on open platforms often withdraw from second-hand commerce entirely rather than risk repeated negative experiences.

3.2 Discovery and Matching Inefficiencies

Students attempting to sell items face significant challenges in reaching potential buyers within their campus communities. Traditional channels including bulletin boards, word-of-mouth, and social media posts suffer from limited reach, short visibility windows, and poor targeting. A student selling a statistics textbook, for example, must somehow connect with another student who needs that specific textbook, is willing to pay the asking price, and is available for exchange—a matching problem that becomes increasingly complex as the number of variables increases. The result is often that items go unsold despite genuine demand existing within the community.

Buyers face complementary challenges in discovering available items that match their needs. Campus bulletin boards are limited by physical space and visibility duration, social media posts quickly disappear from feeds, and general marketplace platforms flood results with irrelevant listings from outside the campus community. Without a centralized, searchable repository of campus-specific listings, buyers must invest significant effort in checking multiple sources, often missing opportunities or giving up the search entirely and purchasing new items despite used alternatives being available nearby.

3.3 Communication and Coordination Friction

Even when buyers and sellers successfully identify each other, the process of coordinating the actual exchange presents substantial friction. Communication must transition from whatever discovery channel-initiated contact to a medium suitable for real-time coordination, often requiring the exchange of personal contact information that parties may be reluctant to share. Scheduling complications arise as students balance academic commitments, and the absence of integrated tools for proposing and confirming meeting times leads to extended back-and-forth exchanges that frequently fail to produce concrete agreements.

The physical exchange itself introduces additional complications. Students must arrange to meet at mutually convenient locations and times, often without clear tools for sharing precise locations or confirming arrival. No-shows waste time and create frustration, while uncertainty about whether the counterparty will appear discourages commitment to the exchange process. When items require inspection before purchase, the absence of remote viewing capabilities means buyers must commit to meetings before knowing whether the item meets their expectations, leading to awkward refusals and wasted trips.

3.4 Transaction History and Item Tracking Challenges

A frequently overlooked challenge in C2C marketplaces is the difficulty users face in tracking their transaction history and maintaining awareness of items they have bought or sold through the platform. Traditional transaction history interfaces present information as a chronologically ordered linear list, displaying timestamps, counterparty names, and transaction amounts in a format optimized

for accounting rather than user recollection. As users accumulate transaction history over months or years, these lists become unwieldy, requiring extensive scrolling or imprecise filtering to locate specific past transactions.

The cognitive burden of navigating lengthy transaction histories is compounded by the presentation of excessive metadata upfront. Users attempting to recall a particular previous purchase must sift through entries displaying information they did not request and do not need—transaction IDs, payment methods, and administrative timestamps—while the information they seek, namely the product itself and its visual representation, is buried in expandable details or absent entirely. This disconnection between interface design and user mental models creates friction that discourages engagement with transaction history features, undermining their utility for dispute resolution, re-listing decisions, and repeat purchase identification.

4. Target Users and Roles

StuPass implements a streamlined role model that recognizes the fluid nature of C2C marketplace participation. Unlike platforms that enforce rigid buyer-seller distinctions, StuPass acknowledges that student users naturally transition between buying and selling roles based on their current needs, resulting in a unified user model where a single account can perform both buyer and seller actions interchangeably. This design philosophy simplifies the user experience while accurately reflecting the actual usage patterns of student marketplace participants.

4.1 Student User Role

The student user represents the sole participant type in the StuPass ecosystem, with all platform functionality accessible through this unified role. Students authenticate using their student ID credentials combined with email verification, establishing their verified academic status that forms the foundation of platform trust. Once registered, students can immediately engage in both buying and selling activities without additional role transitions or permission requests, creating a frictionless marketplace experience that encourages active participation.

As sellers, students can create product listings that include multiple photographs, detailed descriptions, condition assessments, and pricing information. The listing interface guides students through providing comprehensive information that helps buyers make informed decisions, reducing the back-and-forth communication that often plagues incomplete listings. Sellers maintain control over their listings throughout the lifecycle, with the ability to modify details, adjust pricing, mark items as sold, or remove listings entirely when circumstances change. The seller dashboard provides visibility into listing performance, buyer inquiries, and transaction history that supports effective marketplace participation.

As buyers, students can browse the marketplace through multiple discovery mechanisms including category navigation, keyword search, and filtered feeds. The browsing experience surfaces relevant listings while providing the information needed to assess product suitability without requiring immediate seller contact. When a listing warrants further investigation, buyers can initiate real-time communication with sellers and coordinate meeting logistics through integrated messaging, request video calls for remote inspection. The purchase process culminates in transaction confirmation and optional review submission that contributes to the seller's reputation profile.

4.2 User Capability Matrix

The following table summarizes the capabilities available to student users within the StuPass platform:

Capability Category	Available Actions
Account Management	Profile creation/editing, avatar upload, contact info management
Listing Management	Create, edit, list, delist, delete listings; upload images; set pricing; mark as sold
Storage Management	Personal inventory tracking, private tagging, purchased item history
Discovery & Browsing	Search by keyword; filter by category, price, location; browse feeds
Communication	Real-time messaging; image sharing; video calls
Transaction Management	Purchase history; mark transactions complete; write reviews
Reputation System	View own ratings/reviews; receive ratings from transactions; view others' reputation

Table 1: Student User Capability Summary

5. Core System Features

StuPass incorporates a comprehensive suite of features designed to address the complete C2C transaction lifecycle within the student community context. Each feature has been designed to integrate seamlessly with other system components, creating a cohesive platform that transforms the typically fragmented second-hand commerce experience into a streamlined, trustworthy process. The following sections detail the core features and their implementation approaches.

5.1 Authentication and Student Verification

The authentication system implements a multi-stage verification process designed to establish and maintain student identity throughout the user lifecycle. Initial registration captures essential information including student ID number, institutional email address, full name, and verified phone number. Email verification through a confirmation link/one-time code establishes communication validity and prevents automated registration abuse.

Student ID verification represents a critical trust-building component that distinguishes StuPass from open marketplace platforms. The system validates the institutional email domain against a database of recognized academic institutions, ensuring that registration is limited to legitimate students at participating universities. Verified users gain full platform access, while unverified users receive limited functionality that encourages completing the verification process. The verification status is prominently displayed on user profiles, providing transparency that enables informed transaction decisions.

Session management implements secure token-based authentication with configurable expiration periods. Password reset functionality through email verification ensures account recovery capabilities while maintaining security standards. The system supports multiple authentication providers including traditional email/password combinations and institutional single sign-on where available, providing flexibility that accommodates diverse university authentication infrastructures. All authentication events are logged for security auditing and fraud detection purposes.

5.2 Storage System

The Storage system provides users with a personal inventory management capability that serves dual purposes within the platform ecosystem. Primarily, Storage functions as a private catalogue where users can record items in their possession, complete with descriptions, photographs, condition assessments, and private organizational tags. This inventory management aspect enables users to maintain awareness of their belongings independent of marketplace activity, supporting personal organization and simplifying the transition from ownership to listing when users decide to sell.

The secondary and arguably more impactful purpose of Storage is its role as an intuitive transaction history interface. When a marketplace transaction completes, the sold item automatically transfers to the buyer's Storage under a designated "Purchased" category, complete with a backlink to the original order information. This approach transforms the traditionally unwieldy chronological transaction list into a browsable collection of owned items, where users can visually scan their purchases organized by category rather than mentally reconstructing their history from timestamped entries. The product-centric view aligns with how users naturally think about their possessions, making transaction history genuinely useful for recall and reference.

The Storage interface closely mirrors the product listing interface, providing visual consistency that reduces learning curve and cognitive overhead. Items display with thumbnail images, names, and category badges in a grid or list layout familiar from marketplace browsing. Private tags allow users to apply custom organizational labels that remain visible only to the item owner, supporting personalized categorization schemes without imposing platform-defined taxonomies. When items transfer ownership through completed transactions, these private tags are automatically stripped, ensuring the new owner receives a clean inventory entry without the previous owner's organizational context.

5.2.1 Listing Integration

Storage integrates tightly with the listing creation workflow, enabling users to create marketplace listings from existing Storage entries with minimal data re-entry. When initiating a new listing, users are presented with two options: select a pre-existing item from their Storage inventory or enter item information manually as in traditional listing flows. Selecting a Storage item pre-populates the listing form with stored data including photographs, description, and condition assessment, which the user can then modify or supplement before publishing.

This integration provides tangible benefits for user workflow efficiency. Students who maintain their Storage inventory proactively can convert items to listings in seconds rather than minutes, lowering the friction that might otherwise discourage listing creation. The connection between Storage and listings also supports data consistency, as a single Storage entry can be listed, delisted, and re-listed multiple times without requiring information re-entry, accommodating scenarios where timing or pricing adjustments necessitate temporary marketplace absence.

The system maintains a clear distinction between Storage entries and active listings. An item in Storage can exist without any associated listing, representing personal property not currently for sale. When a Storage item is used to create a listing, the relationship is tracked but the Storage entry remains independent, allowing users to continue managing their inventory even while items are actively listed. Upon successful sale, the Storage entry transfers to the buyer's inventory, maintaining continuity of the product record across the ownership transition.

5.3 Product Listing and Marketplace

The product listing system enables comprehensive item documentation that supports informed purchase decisions while minimizing the communication overhead typically required to gather missing information. As detailed in the Storage section, sellers may create listings either from existing Storage entries or by entering information manually. In either case, sellers upload multiple photographs that collectively showcase the item from various angles, with the first image designated as the primary listing thumbnail visible in browse views. The image upload interface provides guidance on effective product photography and enforces minimum quality standards to ensure listings present items accurately and attractively.

Listing metadata captures essential product attributes including name, category, original price, asking price, and condition assessment. The condition system uses a standardized scale (New, Like New, Good, Fair, Poor) with accompanying text descriptions that help sellers accurately characterize item state. A free-form description field allows sellers to document specific defects, modifications, or other details that affect item value or usability. This structured yet flexible approach ensures consistent information availability while accommodating the diverse item types present in student marketplaces.

The marketplace feed presents listings through multiple view modes optimized for different browsing patterns. Grid view maximizes visual scanning efficiency, displaying thumbnail images with price and condition badges for rapid assessment. List view provides expanded information including description previews, seller reputation indicators, and distance calculations for users prioritizing detailed comparison. Filtering capabilities constrain results by category, price range, condition, and campus location, enabling targeted discovery that surfaces relevant listings without overwhelming users with irrelevant options.

5.4 Real-Time Communication System

The integrated messaging system enables real-time text communication between buyers and sellers without requiring exchange of personal contact information. Messages are transmitted through WebSocket/MQTT connections that provide instant delivery with visual indicators for sent, delivered, and read states. The conversation interface displays message history with participant avatars and timestamps, creating a familiar chat experience that requires no learning curve for users accustomed to modern messaging applications. Image attachments can be shared within conversations, allowing buyers to request additional photos or sellers to provide documentation.

Video calling extends the communication capabilities to support remote product inspection, addressing a critical need in second-hand commerce where item condition significantly impacts value. The integrated video call interface includes camera switching between front and rear lenses, enabling sellers to provide detailed product tours while buyers observe in real-time. Mute and camera disable controls allow participants to manage their presence during calls. This capability dramatically reduces the uncertainty that leads to abandoned transactions, as buyers can verify item condition before committing to in-person meetings.

5.5 Transaction Management and Reputation

Transaction management features support the complete lifecycle from initial contact through final exchange and follow-up. Sellers can mark listings as sold to remove them from active marketplace visibility while preserving the listing record for transaction history purposes. The system captures the buyer identity, transaction date, and final price, creating a permanent record that contributes to both parties' transaction histories. This documentation supports the reputation system while providing users with a personal transaction log for their own reference.

The reputation system aggregates transaction outcomes into reputation profiles that inform future transaction decisions. After each completed transaction, buyers can submit star ratings and optional written reviews describing their experience. These reviews appear on the seller's profile, providing visibility into historical transaction behaviour that helps potential buyers assess reliability. The system calculates aggregate reputation scores that summarize overall performance while preserving access to individual reviews for users who want detailed context.

Review content is moderated through a combination of automated filtering and user reporting mechanisms. Inappropriate content, spam, and retaliatory reviews can be flagged for automated administrative review, with actions ranging from content removal to account suspension for serious violations. This moderation framework maintains the usefulness of the reputation system while protecting users from abuse. The reputation system also supports transaction resolution by providing a structured channel for feedback that might otherwise escalate to public complaints or formal disputes.

5.6 Content Moderation and Reporting

The reporting system enables community-driven content moderation that maintains marketplace quality at scale. Users can report listings or other users for various concerns including inappropriate content, suspected fraud, harassment, or counterfeit items. Reports are categorized by reason type and routed for automated administrative review, with priority given to serious safety concerns. The reporting interface captures relevant context including optional message attachments that support investigation.

Reported content undergoes review against platform guidelines, with outcomes including content removal, user warnings, temporary suspensions, and permanent bans for severe or repeated violations. The enforcement system tracks violation history to enable progressive discipline that appropriately addresses different severity levels. Users who submit reports receive notification of resolution outcomes, closing the feedback loop that encourages continued community participation in content moderation.

6. Data Model and Entity Relationships

The StuPass data model implements a normalized relational schema designed for the SeaORM framework, emphasizing referential integrity, query efficiency, and maintainability.

6.1 Core Entities

The User entity serves as the central entity around which most other entities revolve, reflecting the user-centric nature of the marketplace platform. Core required fields include a unique identifier, full name, institutional email, and student ID number—the minimum information necessary to establish

verified student identity. Optional fields capture supplementary profile information including avatar URL, biography, and phone number that enhance user profiles without being essential to platform functionality. The reputation score and verification status fields track user standing within the platform.

The Product entity represents items available for sale, with required fields capturing essential listing information including name, price, category, condition, and owner reference. The separation of Product from Post entities reflects the design pattern where Product captures relatively stable item attributes while Post captures the mutable marketplace presence including visibility state, view count, and publication timing. This separation supports scenarios where products might be relisted with modified attributes while maintaining historical records.

Communication entities include ChatBox, ChatParticipant, and Message, implementing a flexible messaging architecture that supports both one-on-one conversations and potential future group chat scenarios. The ChatBox entity represents a conversation container, ChatParticipant links users to conversations with participation metadata, and Message captures individual communication events. This design enables efficient message retrieval through conversation-scoped queries while maintaining the relationship structure needed for features like unread message counts and conversation hiding.

6.2 Entity Relationship Summary

The following table summarizes the primary entity relationships within the StuPass data model:

Entity	Related Entity	Relationship	Cardinality
User	Credential	Authentication provider	1:N
User	Product	Item ownership	1:N
User	Storage	Personal inventory	1:1
Storage	Product	Listing source (optional)/Purchased products	1:N
Product	Post	Marketplace listing	1:0..1
Post	PurchaseHistory	Transaction record	1:0..1
PurchaseHistory	Storage	Purchase history backlink	1:1

Entity	Related Entity	Relationship	Cardinality
User	ChatParticipant	Conversation membership	1:N
ChatBox	Message	Contains messages	1:N
User	UserReport	Reports submitted	1:N

Table 2: Primary Entity Relationships

7. Technical Architecture

7.1 Architecture Overview

StuPass implements a client-server architecture optimized for mobile-first marketplace operations with real-time communication requirements. The architecture separates concerns between the Flutter-based mobile client and the Rust-based backend services, enabling independent development, testing, and scaling of each component. The backend exposes RESTful APIs for standard CRUD operations while leveraging WebSocket connections for real-time messaging and notification delivery. This hybrid approach balances the simplicity of request-response patterns for data operations with the responsiveness of persistent connections for time-sensitive communication.

The server infrastructure is designed for horizontal scalability, with stateless API servers that can be provisioned elastically based on demand. Database operations are handled through SeaORM, providing type-safe query construction and connection pooling that maximizes throughput while protecting against common database access vulnerabilities. The architecture supports containerized deployment with orchestration for reliability and operational efficiency, enabling straightforward scaling as the user base grows.

7.2 Mobile Application Stack (Flutter)

The mobile application is built using Flutter, enabling deployment to both iOS and Android platforms from a single Dart codebase. Flutter's widget-based architecture supports rapid UI development with consistent rendering across platforms, while its compiled-to-native approach delivers performance suitable for the image-heavy, real-time communication features central to the StuPass experience. The framework's hot reload capability accelerates development iteration, and its comprehensive widget library provides building blocks for the marketplace interface.

- State Management: Provider or Riverpod pattern for reactive state propagation across the widget tree

- HTTP Client: Dio package for RESTful API communication with interceptors for authentication and error handling
- WebSocket: web_socket_channel package for real-time messaging connections with automatic reconnection logic
- Image Handling: cached_network_image for efficient remote image loading with local caching
- Local Storage: SharedPreferences for user preferences and Hive for offline data persistence
- Camera & Media: image_picker and camera packages for photo capture and video calling integration

7.3 Backend Server Stack (Rust/Axum/SeaORM)

The backend server is implemented in Rust using the Axum web framework, leveraging Rust's memory safety guarantees and zero-cost abstractions for high-performance, reliable service delivery. Axum's tower-based middleware architecture enables clean separation of concerns for authentication, logging, and error handling while its async-first design efficiently handles the concurrent connections required for real-time communication features.

- Web Framework: Axum for HTTP routing, request extraction, and response serialization
- ORM: SeaORM for type-safe database operations with async support and migration tooling
- Database: PostgreSQL for relational data storage with JSON support for flexible document fields
- WebSocket: axum-extra WebSocket support for real-time messaging connections
- Authentication: JWT tokens with bcrypt password hashing for secure credential management
- File Storage: S3-compatible object storage for user-uploaded images and media

7.4 Infrastructure Components

The supporting infrastructure enables the core application services to operate reliably at scale:

- API Gateway: Request routing, rate limiting, and API version management for client communication
- Message Queue: Background task processing for image processing, notification delivery, and report handling
- Cache Layer: Redis for session storage, frequently accessed data, and real-time presence indicators

- Search Engine: Full-text search capabilities for product discovery with typo tolerance and relevance ranking

8. Security Considerations

Security represents a foundational concern for StuPass given the personal information handled during student verification and the trust implications of facilitating in-person meetings between platform users. The security architecture implements defence-in-depth principles, with multiple layers of protection addressing different threat vectors from network-level attacks to application-level vulnerabilities and social engineering attempts.

8.1 Data Protection

All data in transit is protected using TLS 1.3 encryption, ensuring that communications between mobile clients and backend servers cannot be intercepted or modified by malicious actors. Data at rest in the PostgreSQL database is encrypted using platform-native encryption capabilities, with backup encryption for disaster recovery. Sensitive fields including password hashes and authentication tokens receive additional protection through dedicated hashing and encryption mechanisms that limit exposure even if database access is compromised.

Student ID card images submitted during verification require special handling given their sensitive nature as identity documents. These images are stored in encrypted object storage with access controls that restrict retrieval to verification processes and administrative review. Retention policies ensure that verification documents are deleted after verification completion or expiration, minimizing the privacy exposure from stored identity documents.

8.2 Access Control

Authentication requires verified student credentials, with JWT tokens managing session state for authenticated requests. Token expiration periods balance security (shorter sessions reduce exposure from stolen tokens) with usability (frequent re-authentication creates friction). The system implements device tracking that enables detection of suspicious login patterns, with additional verification challenges triggered for logins from unrecognized devices or locations.

Authorization enforces the principle of least privilege, with users able to access only their own data and the public profile information of other users. API endpoints validate not just authentication status but also resource ownership for operations that modify data. The separation of Product and Post entities enables fine-grained access control where users can modify their own listings while viewing others' listings as read-only.

8.3 Trust and Safety Features

Platform trust is enhanced through features designed to reduce risk during in-person transactions. The reputation system provides visibility into transaction history, enabling users to assess counterparty reliability before committing to meetings. The messaging system's privacy-preserving design allows communication coordination without exposing personal contact information. Video call inspection capabilities reduce the need for physical meetings with unknown parties by enabling remote product verification.

The reporting system provides a channel for flagging concerning behaviour, with administrative review processes for investigating and responding to reports. User blocking capabilities allow individuals to prevent communication from specific users without requiring administrative intervention. These features collectively create multiple mechanisms for addressing trust and safety concerns before they escalate to serious incidents.

9. Conclusion

StuPass represents a comprehensive approach to solving the trust, discovery, and coordination challenges that impede efficient secondhand commerce within student communities. By creating a verified, student-exclusive marketplace platform with integrated communication and reputation systems, the project addresses the fundamental barriers that prevent students from realizing the economic and environmental benefits of peer-to-peer item exchange. The technology choices of Flutter for cross-platform mobile development and Rust with Axum for backend services provide the performance, reliability, and maintainability foundation necessary for a production-quality platform.

The Storage system introduces a novel approach to transaction history that aligns with how users naturally think about their possessions. By presenting past purchases as a browsable collection rather than a chronological ledger, the platform transforms transaction history from an administrative afterthought into a genuinely useful feature that users will engage with voluntarily. The integration between Storage and listing creation further enhances user experience by reducing friction in the seller workflow, encouraging marketplace participation through ease of use.

The core features outlined in this document directly address the pain points identified in the problem statement: student verification establishes the trust foundation missing from open marketplaces, comprehensive product listings reduce discovery friction, integrated communication tools streamline coordination, the Storage system provides intuitive transaction history, and the reputation system creates accountability that encourages positive transaction behaviour. The data model has been carefully designed to avoid the optional field proliferation that complicates business logic, ensuring clean data states that support reliable application behaviour.

This initial design establishes the foundation for detailed requirements specification, technical implementation, and iterative refinement. The modular architecture enables phased delivery with demonstrable value at each milestone, supporting an agile development approach that can

incorporate user feedback as the platform evolves. We look forward to advancing this project through collaborative development and committed execution, ultimately delivering a student marketplace platform that transforms how campus communities conduct second-hand commerce.