NDI: TAP Technical Assessment Submission – Rachel Gina Abelarde

**Problem Statement**

In Singapore's dynamic e-commerce landscape, preserving the integrity and trustworthiness of marketplaces, while ensuring peace of mind for all participants, is a formidable challenge. A critical concern is the need to effectively mitigate the escalating threats of fraudulent activities, including the widespread illegal resale of items such as concert tickets.

**Proposed Solution**

To enhance the security and integrity of Singapore's e-commerce ecosystem, I propose the integration of SingPass, NDI's Verify system, and the implementation of biometric authentication, specifically Identiface, into e-commerce platforms (especially B2B ones). For now let’s give it an arbitrary name – SGMarket, just for ease of reference.

User registration will be simplified by utilizing SingPass. If any suspicious activities are detected in connection with a user, our system will generate notifications and flag the user's account. Subsequently, flagged users will undergo biometric verification during their login attempts. Verify will be used during transactions to confirm the exchange of goods before the release of payment to minimise remote scams/scams due to remote purchases.

In compliance with government regulations, our platform will adhere to strict privacy standards and won't access users' personal information. Instead, both users and the application team will be prompted to report suspicious activities to NDI for further investigation. The application team will also have access to logs of suspicious activities associated with each user. Furthermore, the added security not only discourages potential scammers from registering, but in the event they do, NDI can access their actual personal data for identification purposes.

This solution is vital for establishing a secure and trustworthy digital marketplace, promoting the digitization of trade. It allows citizens to transact seamlessly using their digital identities, with reduced concerns about scams, all backed by a reliable government ecosystem.

Furthermore, I recommend expanding our biometric verification options beyond facial recognition to include voice and thumbprint recognition. This flexibility will enable us to apply different verification methods based on the level of suspicion associated with user activities

**[A diagram of a computer

Description automatically generated](https://drive.google.com/file/d/14TMFhzgqdTwZbdScGy7TXjuZKzaCvfRh/view?usp=sharing)User Flows­**

Illustration 1.1 User flow - User Registration and Login

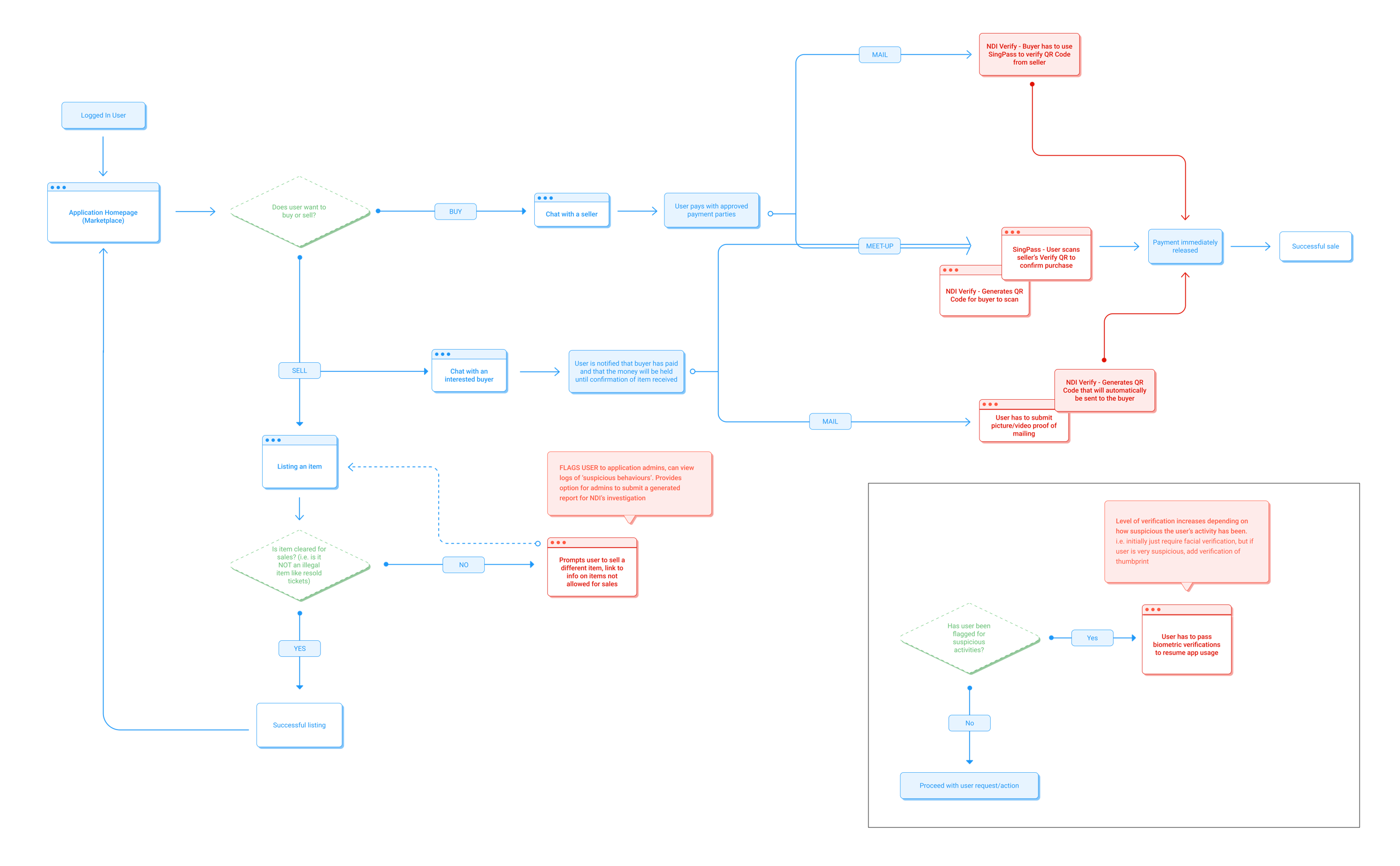
[](https://drive.google.com/file/d/1PF-vbdRUxrbVCNauJujcBgzN9qJ-IMju/view?usp=sharing)

Illustration 2.2 Anti-scam feature that can happen anytime

Illustration 2.1 User flow – User Features

**Database**

I have chosen to utilise MongoDB and structure the database in such a manner:

**Users Collection:** This collection stores information about the users of the application.

Attributes:

* \_id: Automatically generated unique identifier.
* uuid: Unique identifier provided by SingPass for user registration.
* username: The chosen username of the user.
* email: The email address associated with the user.
* biometricVerified: Indicates whether biometric verification is required for the user.
* isAdmin: Indicates whether the user has admin privileges.
* flags: A numeric value representing the number of flags associated with the user for suspicious activities.
* address: The residential address associated with the user.

**Items Collection\*:** This collection contains details about items listed by users for sale or trade.

Attributes:

* \_id: Automatically generated unique identifier.
* userId: Reference to the user who listed the item.
* itemName: The name or title of the item.
* description: A description of the item.
* status: The current status of the item (e.g., valid, reserved, sold).
* price: The price of the item.
* reservations: An array of references to reservation documents associated with the item.

**Reservations Collection:** This collection stores reservation information related to items.

Attributes:

* \_id: Automatically generated unique identifier.
* itemId: Reference to the item being reserved.
* buyerId: Reference to the user who reserved the item.
* approvalStatus: The status of the reservation (e.g., pending, approved, rejected).
* meetupLocation: The location for in-person meetups, if applicable.
* isMailing: Indicates whether mailing is chosen.
* paymentStatus: The status of payment for the item (e.g., pending, completed).
* priceOffer: The price offered by the buyer.
* isMailingProven: Indicates whether the item’s mailing evidence (media\*\*) has been sent.
* isReceived: Indicates whether the item has been received.

**Chats Collection:** This collection represents chat conversations between users.

Attributes:

* \_id: Automatically generated unique identifier.
* participants: An array of references to the users participating in the chat.
* itemId: Reference to the item associated with the chat.

**Messages Collection:** This collection contains individual messages exchanged within chat conversations.

Attributes:

* \_id: Automatically generated unique identifier.
* chatId: Reference to the chat associated with the message.
* senderId: Reference to the user who sent the message.
* content: The content or text of the message.
* timestamp: The timestamp indicating when the message was sent.

**Suspicious Activity Reports Collection:** This collection stores reports of suspicious activities associated with users.

Attributes:

* \_id: Automatically generated unique identifier.
* userId: Reference to the user being reported.
* reportedBy: Reference to the user who reported the suspicious activity.
* reason: A description of the reason for the report.
* datetime: The date and time when the report was submitted.

\*Deletion of item will happen once sold for this prototype, for future iterations, soft delete can be done where a CRON job can migrate items that are sold to another collection for logging purposes. Alternatively, once a purchase is completed, we can do a POST to a new ‘sold items’ collection and DELETE the item off the items collection

\*\*For now I will leave media out of the database and not include the schema in my database as well as prototype development.

**Architecture**

**[A diagram of a company

Description automatically generated](https://drive.google.com/file/d/1sQgP5r36Xnz8war63E1c4P82ZhhPxC9B/view?usp=drive_link)**

**Appendix**

Brainstorming:

(Customer Problem Statements) Potential Issues that can be tackled

A white rectangular sign on a yellow surface

Description automatically generatedA screen shot of a phone

Description automatically generated

A yellow box with a white rectangle on it

Description automatically generated