**ShareIt - An Intra College Rental Platform**

**Contents :**

ShareIt Rental Platform : Project Specification Document

1. Introduction
2. Objectives
3. Technical Specifications

Backend ( C++ )

Native Integration with Java Middleware (JNI)

1. Development Setup:

1. Important Files and Folders:

**Introduction :**

*ShareIt* is a web-based tool that allows students to share and borrow items within their college community. It helps students list items they own for sharing, request items they need, and facilitates easy, sustainable resource exchange.

**Objectives :**

The code provides a comprehensive system for managing an item sharing and borrowing platform. The main goals of this system are:

* **User Management:** Allow users to register, log in, log out, and manage their accounts, including maintaining coin balances for transactions.
* **Item and Listing Management:** Enable users to list items for sharing, manage their availability, set prices, and specify conditions.
* **Request Management:** Let users request items, search for suitable listings, and manage their requested items.
* **Notifications:** Facilitate communication between users via notifications, such as borrowing requests or responses to requests.
* **Transactions and Borrowing:** Manage item borrowing, track borrowers, and handle returns.
* **Search and Discovery:** Provide mechanisms to search for items and requests based on criteria.
* **Interaction and Negotiation:** Establish a system where users can interact, such as through notifications or potential negotiation features.

**Implementation :**

Here’s how the functionality is implemented based on your class definitions:

1. **User Management**

* **Class: User**
  + **Registration:** Manager::registerUser allows users to register with a username and password.
  + **Authentication:** User::authenticate checks credentials for login via Manager::login.
  + **Coins:** Users start with 100 coins. Functions like acceptCoins and spendCoins manage their coin balances.

1. **Item and Listing Management**

* **Classes: Item and Listing**
  + **Item Details:** Item contains attributes like name, category, condition, quantity, and availability dates. Accessor and mutator

methods are provided for updates.

* + **Listing:** Listing encapsulates an item with additional details such as price and condition, as well as methods to manage availability (bookItem, freeItem).

1. **Request Management**

* **Classes: User and Manager**
  + Users can request items via User::requestItem or Manager::addRequest.
  + Requests are maintained in therequested\_items vector.
  + Requests can be removed with User::removeRequest.

1. **Notifications**

* **Classes: Notification**
  + Notifications represent communication between users about borrowing requests or responses.
  + The Manager::notifyUser function sends notifications, and User::printNotifications displays them.

1. **Transactions and Borrowing**

* **Classes: User, Manager, and Listing** 
  + Borrowing is facilitated by Manager::borrowItem, which updates ownership and listings.
  + Borrowed items are tracked in the borrowed\_items vector.

1. **Search and Discovery**

* **Classes: Manager** 
  + Searches for items (Manager::searchListingsForRequest) or requests (Manager::searchRequestsForListing).

**Technical Specifications :**

**Backend ( C++ ) :**

The C++ portion of the project will handle the core backend functionalities, including user authentication, item and listing management, request processing, notification handling, and data validation. It ensures efficient data storage, accurate processing, and smooth interaction between different components of the application.

**Middleware (Java) and Java Native Interface (JNI) :**

Create a Java Wrapper to bridge the Java frontend with the C++ backend.

* Declaring native methods to call C++ functions.
* Loading the C++ dynamic link library (DLL).