Table of Contents

[Data Types 1](#_Toc106452958)

[User 1](#_Toc106452959)

[Location 1](#_Toc106452960)

[Item 1](#_Toc106452961)

[Computer Game 1](#_Toc106452962)

[Video Game 2](#_Toc106452963)

[Collectible Game 2](#_Toc106452964)

[Condition 2](#_Toc106452965)

[Type 2](#_Toc106452966)

[Trade 2](#_Toc106452967)

# Data Types

## User

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Nullable |
| Email | String | Not Null |
| First Name | String | Not Null |
| Last Name | String | Not Null |
| Password | String | Not Null |
| Postal Code | String | Not Null |

## Location

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Nullable |
| State | String | Not Null |
| Longitude | String | Not Null |
| Latitude | String | Not Null |
| Postal Code | String | Not Null |
| City | String | Not Null |

## Item

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Nullable |
| Condition | String | Not Null |
| Description | String | Nullable |
| Title/Name | String | Not Null |
| Type | String | Not Null |
| City | String | Not Null |

### Computer Game

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Nullable |
| Platform | String | Not Null |

### Video Game

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Nullable |
| Media Type | String | Not Null |
| Platform | String | Not Null |

### Collectible Game

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Nullable |
| # Offered | Int | Not Null |

## Condition

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Nullable |
| Name | String | Not Null |

## Type

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Nullable |
| Name | String | Not Null |

## Trade

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Nullable |
| Date Proposed | Date | Not Null |
| Trade Status | String | Not Null |
| Date Processed | String | Not Null |

# Business Logic Constraints

* Users who are new to TradePlaza must register first
* Users who already have a TradePlaza account will not be able to register
* Users cannot make edits after registration
* User must live in one of the accepted area codes
* If a user has more than two unaccepted trades, they must first accept/reject one before listing a new item
* If a user has more than two unaccepted trades, they must first accept/reject one before proposing a new trade
* Users cannot trade with themselves
* User cannot propose a trade is he/she has not listed items
* Items in an accepted trade cannot be traded again
* If a trade is rejected, the same trade proposal (proposer-counterparty-item) cannot be made again

# Task Decomposition & Abstract Code

## Login

Lock Types: Read-only on User table

Number of Locks: Single

Enabling Conditions: None

Frequency: High

Consistency (ACID): Not critical, order is not critical

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

* User enters $email and $password
* If data validation is successful for both $email and $password, then:
  + When **enter** button is clicked:
    - If User record is found and user.password != $password
      * Return to **login** form and display error message
    - Else
      * Store login information as session variable $UserID
      * Go to **view profile** form
* Else $email and/or $password fields invalid, display error message

## Register

Lock Types: Write on User table, read Location table

Number of Locks: Single

Enabling Conditions: None

Frequency: Low-Med

Consistency (ACID): Not critical, order is not critical

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

* User enters $email, $password, $Nickname, $First Name $Last Name and $Postal code
* If data validation is successful for all fields, then:
  + When **register** button is clicked:
    - If User $email is not found in the User table **and** $Nickname not found in User table **and** postal code is found in the location table
      * Add new user to the User table
      * Redirect new user to the main menu
    - Else
      * Display error message

## View Main Menu

Lock Types: Write on **User, trade, items** tables

Number of Locks: Three

Enabling Conditions: Successful Login In/ Registration

Frequency: High

Consistency (ACID): Not critical, order is not critical

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

* Display welcome message ($first, $last name ($Nickname))
* Compute and display average response time based on the $accept/reject date attribute of the trade table entries for which the user is the counterparty
  + Color the response time label’s background based on the average response time
  + None 🡪 Black
  + 0-0.7 🡪 Green
  + 7.1-14.0 🡪 Yellow
  + 14.1-20.9 🡪 Orange
  + 21-27.9 🡪 Red
  + Else 🡪 Red (Bold)
* For each proposed trade, provide link to accept/reject
* If number of unaccepted trades > 2 then print the number of trades in bold and red
* Compute and display user’s trade rank based on the number of entries into the trade table for which the user has participated
  + 0 🡪 None
  + 1-2 🡪 Aluminum
  + 3 🡪 Bronze
  + 4-5 🡪 Silver
  + 6-7 🡪 Gold
  + 8-9 🡪 Platinum
  + 10+ 🡪 Alexandinium
* Show button to add new listing
* Show button to view the current user’s items
* Show button for searching items
* Show button to view trade history
* Show button for logging user out
* When the **logout** button is pressed
  + Redirect user to **login** screen

## View Trade Details

Lock Types: Read on User, items, R/W on trade table

Number of Locks: Three

Enabling Conditions: Log in, View Trade History

Frequency: Med

Consistency (ACID): Not critical, order is not critical

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

* Get details of proposer and counterparty
* Get/display details on items for trade
* Display trade proposed date, accepted/rejected date, response time, trade status, user’s role
* If the trade status is accepted, then display first name, email of the other party

## List New Item

Lock Types: R/W on items table, R on trades table

Number of Locks: Three

Enabling Conditions: successful login, Main menu 🡪 list item, user has less than 2 pending trades

Frequency: High

Consistency (ACID): Order critical

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

* Get pending trades for which the current user is participating
* If #trades > 2, display error message; return
* Display form and take user input - $game type, $title, $condition, $platform, $Media, $description
* if ok is pressed and fields are all valid
  + Add item to items table
* Else ok is pressed and fields are not valid
  + Display error message – highlight invalid fields