

Table of Content

[Table of Content](#)

[Data Types](#)

[Entity - User](#)

[Entity - Address](#)

[Entity - Item](#)

[Entity - Trade](#)

[Business Logic Constraints](#)

[Task Decomposition & Abstract Code](#)

[Login](#)

[Register](#)

[Main Menu](#)

[Add Listing](#)

[My Items](#)

[Search](#)

[View Item](#)

[Propose Trade](#)

[Accept / Reject trades](#)

[View Trade History](#)

[View Trade Details](#)

Data Types

Entity - User

Attribute	Data Type	Nullable
Nickname	string	No
Email	string	No
Password	string	No
First Name	string	No
Last Name	string	No
Postal Code	string	No

Entity - Address

Attribute	Data Type	Nullable
Longitude	float	No
Latitude	float	No
City	string	No
Postal Code	string	No
State	string	No

Entity - Item

Attribute	Data Type	Nullable
Item Number	integer	No
Name/title	string	No
Condition	string	No
Description	string	Yes
Game Type	string	No
Platform	string	Yes
No of Cards	string	Yes
Media	string	Yes

Entity - Trade

Attribute	Data Type	Nullable
Proposed Date	datetime	No
Trade Status	string	No
Accept/Reject Date	datetime	No
Proposed Item	Integer	No
Desired Item	Integer	No

Business Logic Constraints

1. Users who are new to TradePlaza must register first
2. User email can only be registered once
3. Users are not allowed to update their details after registering
4. Error will be shown on login screen if username/nickname is registered but password is incorrect
5. Email should have a valid email format
6. If a user has more than two unaccepted trades they cannot list a new item and application should throw an appropriate error message
7. If a user has more than two unaccepted trades, they cannot propose a trade and should not be allowed to visit propose trade form
8. Users who already have a TradePlaza account will not be able to register
9. A user cannot propose a trade with themselves
10. A user must list an item before proposing a trade
11. If a trade is rejected, the same trade proposal (proposer-counterparty-item) cannot be made again
12. Items that have been traded are not allowed for trade again
13. User can select only one game type for a listing
14. Items associated with a pending trade (a proposed trade not yet accepted or rejected) are not available for trading
15. When listing an item, application should only show fields related to the item type
16. When searching for an item, the parameters used to search are exclusive
17. Only items available for trading should be displayed

Task Decomposition & Abstract Code

Login

Task decomposition

Lock Type - Read only on USER Table
Number of Locks - single
Enabling Conditions - none
Consistency (ACID) - Consistency is not critical; order is not critical
Frequency - 150 logins
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

Task decomposition

Lock Type - Write on USER Table, read on ADDRESS table
Number of Locks - Single
Enabling Conditions - None
Consistency (ACID) - Consistency is not critical; order is not critical; Atomicity is needed.
Frequency - 20 registrations
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

Main Menu

Task decomposition

Lock Type - Read-only on USER, TRADE, ITEM LISTING Tables
Number of Locks - Three
Enabling Conditions - Log In / Register
Consistency (ACID) - Consistency is not critical; order is not critical
Frequency - 250 views
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
-

Add Listing

Task decomposition

Lock Type - Read-only on TRADE. Read & Write lock on ITEM LISTING
Number of Locks - Two
Enabling Conditions - Click on List Item, Logged In user, less than 2 unaccepted trades
Consistency (ACID) - Order critical
Frequency - 150 item listings
Subtasks - Mother task needed. Decomposition not 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
 - Return to the screen with a success message pop up.
- Else ok is pressed and fields are not valid
 - Display error message – highlight invalid fields
- END

My Items

Task decomposition

Lock Type - read on Item table
Number of Locks - Single
Enabling Conditions - Logged in User, click on my items
Consistency (ACID) - Consistency is not critical; ascending order on Item #.
Frequency - 50 My Items listings

Subtasks - Mother Task is not needed. No decomposition needed.



Abstract code

- Begin
 - Display Item count form
 - Count the number of items in each item category, and total number of listings.
 - Display available My Items form
 - On click Detail, show Item detail form
 - End
-

Search

Task decomposition

Lock Type - Read-only on USER, ITEM LISTING, TRADE Tables
Number of Locks - Three
Enabling Conditions - Log In / Register
Consistency (ACID) - Consistency is not critical. Order is not critical.
Frequency - 300 searches



Abstract code

- Begin
- Show a rundown of radio buttons which permit to look by watchword, in clients postal code, inside X miles, in determined postal code
- On Click Search!:
- Assuming that results are found:
- Then get things' subtleties from ITEM LISTING
- Get client postcode from ADDRESS
- Ascertain distance with Calculate User Distance
- Show results with Display Search Details
- Show thing number, game sort, thing name/title, the condition and the initial 100 characters of the depiction (in the event that the portrayal is more noteworthy than 100 characters, place an ellipsis (...) toward the finish to demonstrate it has been shortened) , normal reaction time and dealer position of other client
- In the event that hunt by catchphrase chose:
- Then, at that point, feature the fields that match the catchphrase in blue
- Assuming that pursuit by postcode is chosen AND the postcode is invalid:
- Then show a mistake message
- On click Detail:
- Divert to a matching View Item page
- ELSE
- Show a message "Sorry, no outcomes found!"
- END

View Item

Task decomposition

Lock Type - Read-only on USER, ITEM LISTING, TRADE Tables
Number of Locks - Three
Enabling Conditions - Log In / Register
Consistency (ACID) - Consistency is not critical. Order is not critical.
Frequency - 500 views
Subtasks - Get Item Details, Get User Details



Abstract code

- BEGIN
- Obtain item information from the ITEM LISTING
- Obtain user information from USER
- Get the user's address information from ADDRESS.
- Retrieve a user's trades from Trades
- Show the following information about an item:
 - #, game title, platform, media, and item condition
 - IF the item description is not null, then show it; otherwise, hide it.

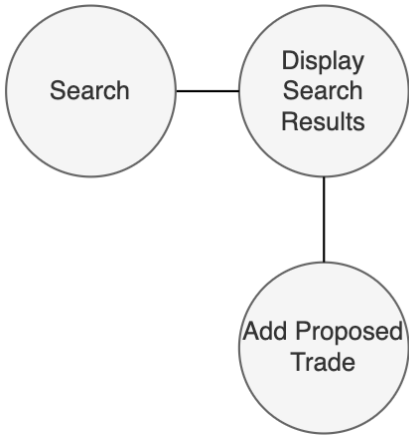
- Show the following information about a user: nickname, city, state, postal code, and response time.
- use the "Calculate User Distance" subtask to calculate the distance between the user and the item's owner.
- use the "Calculate User Distance" subtask to calculate the distance between the user and the item's owner.
- If the user's postal code!= the item's owner postal code,
- Then display distance
- IF $0.0 < \text{distance} < 25.0$ miles
- Then add green background
- IF $25.00 < \text{distance} < 50.0$ miles
- Then add yellow background
- IF $50.00 < \text{distance} < 100.0$ miles
- Then add orange background
- IF $\text{distance} > 100.0$ miles
- Then add red background
- OTHERWISE Hide distance
- Depending on the context, different elements will be displayed.
- If the user has not accepted < 2 trades, then display Propose Trade.
- OTHERWISE, Propose Trade is not displayed.
- When you click Propose Trade, you will be redirected to Add Trade Proposal

Propose Trade

Task decomposition

Lock Type - Read-only on USER and Read and Write lock on ITEM LISTING Tables
Number of Locks - Two
Enabling Conditions - Login and User has less than 2 unaccepted Trades and has at least 1 Item listed in his/her login
Consistency (ACID) - We need to ensure, desired item is locked once identified for trade by a user
Frequency - 150 proposals

Subtasks - Mother task needed. Decomposition is needed. Search, Display Search Results, Propose Trade



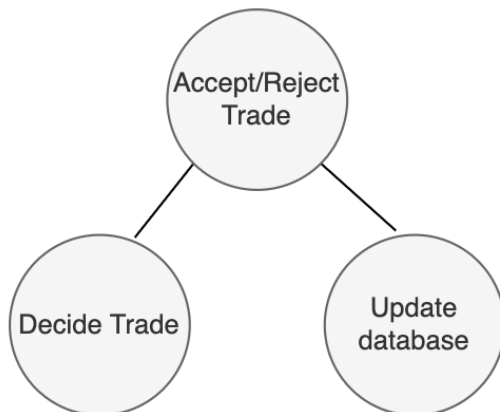
Abstract code

- BEGIN
- If the counterparty distance exceeds 100.0, a warning message with the distance displayed at the top should be displayed.
- Only items that are available for trading must be displayed (self).
- Provide an appropriate mechanism for selecting the desired item.
- Should be displayed Number of items, game type, title, and condition
- Items should be ordered alphabetically.
- Once the proposed item has been selected, the user must be given a confirm Button to confirm the proposal.
- Once confirmed, the proposed date is saved/written into the Trade table.
- The user should be able to return to the main menu after receiving a confirmation message.
- END

Accept / Reject trades

Task decomposition

Lock Type - Read and Write Lock on Trade table, Read lock on USER and ITEM LISTING table
Number of Locks - Three
Enabling Conditions - Login and Has pending Proposals
Consistency (ACID) - Not needed as it is available only to the user
Frequency - 250 decisions
Subtasks - Mother task needed. Decomposition is needed. Decide Trade, Update Database



Abstract code

- BEGIN
- Results of a Search
- List proposed trades in which the user is the counterparty and can accept or reject a trade.
- List proposed trades in which the user is the counterparty and can accept or reject a trade. For each proposal
- Show the date proposed, the desired item's title, the proposer's nickname, their rank, distance from the user (rounded to the hundredths), and the proposed item title, all ordered by proposal date.
- Both item fields should contain a link to the item's detail page.

- A mechanism for accepting/rejecting should be provided: Accept, Reject
- If the trade is accepted, query the USER table and display a dialog containing the proposer's email address and first name.
- The trade's acceptance or rejection date is recorded in the database as part of the trade table
- If the proposed trade is turned down, a new trade for the same proposed item and the same desired item cannot be proposed.
- Accepting a trade removes the item from the listing (Search and Display Search Results), and the user should be returned to the main menu if no more trades need to be accepted or rejected.
- END

View Trade History

Task decomposition

Lock Type - Read-only on USER, ITEM LISTING Read and Write lock on Trade Table
Number of Locks - Three
Enabling Conditions - Log In
Consistency (ACID) - Not needed as it is available only to the user
Frequency - 150 views
Subtasks - Get Item Details, Get User Details



Abstract code

- BEGIN
- Obtain information about users from USER
- Obtain item information from the ITEM LISTING
- Get trade information by USER or ITEM LISTING.
- Calculate a trade summary for the user and display it as a view:
 - SUM(Accepted, Rejected) AS Total, COUNT(IF TRADE STATUS = 'Accepted') COUNT(CASE WHEN TRADE STATUS IS 'REJECTED') AS ACCEPTED AS Rejected, (Total Rejected) AS Rejected percentage
- GROUP USER E-MAIL, ROLE
- IF THE REJECTED PERCENTAGE EXCEEDS 50%
- 'Red' should be highlighted.
- Display the following information about a trade, sorted by acceptance/rejection date in ascending order and trade proposed date in descending order:
 - Proposed date, accepted/rejected date, Trade status, Response Time in days, my role, proposed item, desired item, nickname of other user, link to trade details
- END

View Trade Details

Task decomposition

Lock Type - Read-only on USER, ITEM LISTING Read and Write lock on Trade Table
Number of Locks - Three
Enabling Conditions - Log In, View Trade History
Consistency (ACID) - Not needed as it is available only to the user
Frequency - 50 views
Subtasks - Mother task needed. Decomposition not needed.



Abstract code

- BEGIN
 - 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
 - Display proposed item: Item Number, name/title, game type, condition,
 - IF description, piece count, platform, media is not null, display them
 - Display desired item: Item Number, name/title, game type, condition description
 - IF description, piece count, platform, media is not

null,display them

- END