CS6400: Phase 1 Report

Team65

Summer 2022

Table of Contents

1 TRADEPLAZA DATA TYPES

1.1 User

Attributes	Data Type	Unique	Nullable
Email	String	Yes	No
Nickname	String	Yes	No
Password	String	No	No
First Name	String	No	No
Last Name	String	No	No
Postal Code	String	No	No

1.2 Postal Code

Attributes	Data Type	Unique	Null
Postal Code	String	Yes	No
City	String	No	No
State	String	No	No
Latitude	float	No	No
Longitude	float	No	No

1.3 Item

Attributes	Data Type	Unique	Null
Item Number	Integer	Yes	No
Registered User	String	No	No

Attributes	Data Type	Unique	Null
(Owner)			
Name/Title	String	No	No
Description	String	No	Yes
Condition	String	No	No
Game type	String	No	No
Video Game – Platform	String	No	Yes
Video Game – Media	String	No	Yes
Computer Game – Platform	String	No	Yes
Availability	Boolean	No	No

1.4 Condition

Enum of Unopened, Like New, Lightly Used, Moderately Used, Heavily Used, Damaged/Missing parts

1.5 Game Type

Enum of Board Game, Playing Card Game, Collectible Card Game, Video Game, Computer Game

1.6 Collectible Card Game

Attributes	Data Type	Unique	Null
Number of Cards being offered	Smallint	No	No

1.7 Video Game – Platform

Enum of Nintendo, PlayStation, Xbox

1.8 Video Game – Media

Enum of optical disc, game card, cartridge

1.9 Computer Game – Platform

Enum of Linux, macOS, Windows

1.10 Trade

Attributes	**Data Type**	**Unique**	**Null**
Proposer item number	Int	Yes	No
Desired item number	Int	Yes	No
Proposal date	Date	No	No
Trade Status	String	No	No
Accept/Reject date	Date	No	Yes

2 TRADEPLAZA BUSINESS LOGIC CONSTRAINTS

2.1 User

- New user need self-register
- Email is used to uniquely identify user in DBMS
- User cannot update detailed information after registration
- Present error message when password is wrong
- Check valid email format, otherwise pop error message

2.2 Item

- If a user has two or more unaccepted trades where they are the counterparty, they cannot list a new item and an appropriate message should be displayed.
- When collectible card game is chosen for game type, display, Number of cards being offered, input option
- When video game is chosen for game type, display, platform and media drop downs
- When computer game is chosen for game type, display, platform dropdown
- Item number is ordinally auto-generated by system
- Item availability becomes FALSE, when owner propose to trade or trade completed

2.3 Search

- Search parameters are exclusive
- Search results form's title and header should indicate what type of search has been executed
- Keyword search, the field(s) that matched the keyword should be highlighted with a blue background in the search results

2.4 Trade

- Counterparty can only propose trade when item availability is TRUE
- User must list an item before proposing trade
- User cannot propose trade to him/herself
- User without item listed can only browse, but not trade

- Proposer item number and Desired item number pair has no duplicates, meaning same trade cannot be proposed again.
- If action is TRUE, meaning trade is completed, both proposed and desired item availability are FALSE and cannot be used for trading again

3 TASK DECOMPOSITION AND ABSTRACT CODE (TD/AC)

3.1 Login

3.1.1 *Task decomposition*

Lock Type: Read-only on User

Number of Locks: Single

Enabling Conditions: None

Frequency: 200 logins

Consistency (ACID): consistency not critical, order not critical.

Subtasks: Mother task is not needed. No decomposition needed.

3.1.2 *Abstract Code*

- Begin
- Display **Login** form
- User enters *email* ('$Email'), *password* ('$Password') input fields.
- On click ***Enter*** button
- If data validation is successful for both *username* and *password* input fields, then:
    - If User record is found but User.password != '$Password':
    - Go back to **Login** form, with error message.
- Else:
    - Go to Main Menu form.
    - Store login information as session variable '$UserID
- Else *email* and *password* input fields are invalid, display **Login** form, with error message.
- On click ***Registration*** button
    - Go to **User registration** form
- End

3.2 User registration

3.2.1 *Task decomposition*

Lock Type: Write on User Table, read on Postal Code table

Number of Locks: Single

Enabling Conditions: None

Frequency: 200 registrations

Consistency (ACID): consistency not critical, order not critical. Atomicity needed.

Subtasks: Mother task is not needed. No decomposition needed.

3.2.2 *Abstract Code*

- Begin
- Display **Registration** form
- On click ***Register*** button
    - Validate all fields are completed in required format
    - Look up Postal code table to validate input
- If validation error occurs
    - Display error message and stay in **Registration** form
- Else:
    - Write values to User table
    - Go to **Main menu** form
- End

3.3 Main menu

3.3.1 *Task decomposition*

Lock Type: Read-only on User, Trade, Item table

Number of Locks: three

Enabling Conditions: login/register

Frequency: 500

Consistency (ACID consistency not critical, order not critical. Atomicity needed.

Subtasks: Mother task is not needed. No decomposition needed.

3.3.2 *Abstract Code*

- Begin
- Display **Main Menu** form
    - Welcome
    - First Name and Last Name
    - Nickname
    - Unaccepted trades
        - Lookup Trade table
        - If number of proposed trades > 0 Then display link to accept/reject trades
        - If number of unaccepted trades > 2 Then print number of trades in bold and red
        - Store unaccepted trade number as session variable '$unacceptedtrades#’
    - Response time
        - Lookup Trade table
        - If response time is None Then Black
        - Else if response time in between 0 and 7.0 Then Green
        - Else if response time in between 7.1 and 14.0 Then Yellow
        - Else if response time in between 14.1 and 20.9 Then Orange
        - Else if response time in between 21.0 and 27.9 Then Red
        - Else if response time > 28.0 Then Bolded Red
        - Store response time number as session variable '$responsetime#’
    - My rank
        - Lookup Trade table
        - If no of trades == 0 Then None

        - Else if no of trades in between 1 and 2 Then Aluminium
        - Else if no of trades in between 3 and 4 Then Bronze
        - Else if no of trades in between 4 and 5 Then Silver
        - Else if no of trades in between 6 and 7 Then Gold
        - Else if no of trades in between 8 and 9 Then Platinum
        - Else if no of trades > 10 Then Alexandinium
        - Store my rank level as session variable '$myrank'
        -
    - ***List Item*** button
        - On click, go to **List Item** task
    - ***My items*** button
        - On click, go to **My Item** task
    - ***Search items*** button
        - On click, go to **Search Item** task
    - ***Trade history*** button
        - On click, go to **Trade History** task
    - ***Logout*** button
        - Invalidate login session and go back to the **Login** form
- End

3.4 List item

3.4.1 *Task decomposition*

Lock Type: Read-only on Trade table, read and write on Item table

Number of Locks: two

Enabling Conditions: click on ***List Item***, Logged in user, less than 2 unaccepted trades

Frequency: 200

Consistency (ACID): order critical

Subtasks: mother task needed, decomposition not needed

3.4.2 *Abstract Code*

- Begin
- Look up Trade table to get total unaccepted trades of user
    - If user has > 2 unaccepted trades Then display error "Cannot list item. Check pending trades"
        - Back to ***Main Menu*** button
    - Else:
        - Show **<u>New Item Listing</u>** form
        - On click Game types show Game Type in dropdown
            - If Collectable card game selected, then show number of cards being offered
            - If video game selected
                - Show dropdown for platform
                - Show dropdown for media
            - If computer game selected
                - Show dropdown for platform
        - On click Condition display Condition in dropdown
        - Description field is optional
- On click List Item button
    - Validate fields and throw appropriate messages to the user. Stay on the **<u>New Item Listing</u>** form
- If no error then write field values to Item table
- Display screen with a success message pop up.
- End

3.5 My items

3.5.1 *Task decomposition*

Lock Type: Read-only on Item table

Number of Locks: one

Enabling Conditions: click on ***My Item***, Logged in user

Frequency: 200

Consistency (ACID): ascending order in Item #

Subtasks: mother task needed, decomposition not needed

3.5.2 *Abstract Code*

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

3.6 Search items

3.6.1 *Task decomposition*

Lock Type: Read-only on User, Trade, Item table

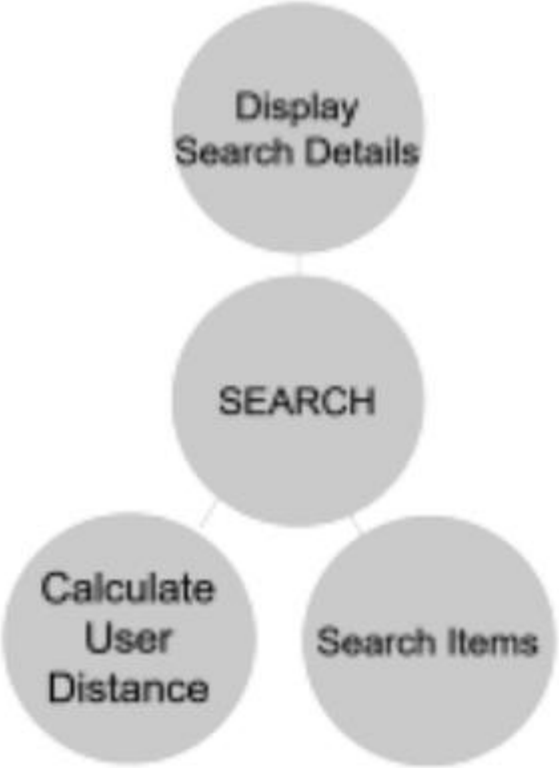
Number of Locks: three

Enabling Conditions: login/register

Frequency: 500

Consistency (ACID): Consistency is not critical. Order is not critical.

Subtasks: Calculate User Distance, Search Items, Display Search Details



3.6.2 *Abstract Code*

- BEGIN
- Display a list of radio buttons which allow to search by keyword, in users postal code, within X miles, in specified postal code
- On Click ***Search!*** Button
- If results are found:
    - get items' details from Item table
    - Get user longitude and latitude from Postal Code table
    - Calculate distance with Calculate User Distance
    - Display results with Display Search Details
    - Display item number, game type, item name/title, the condition and the first 100 characters of the description (if the description is greater than 100 characters, place an ellipsis (…) at the end to indicate it has been truncated) , average response time and trader rank of other user

        - IF search by keyword selected:
            - highlight the fields that match the keyword in blue
        - IF search by postcode is selected AND the postcode is invalid:
            - display an error message
    - On click ***Detail*** button, go to **Item Details** form
- ELSE
    - Display a message "Sorry, no results found!"
- End

3.7 View item

3.7.1 *Task decomposition*

Lock Type: read-only on User, item and trade tables

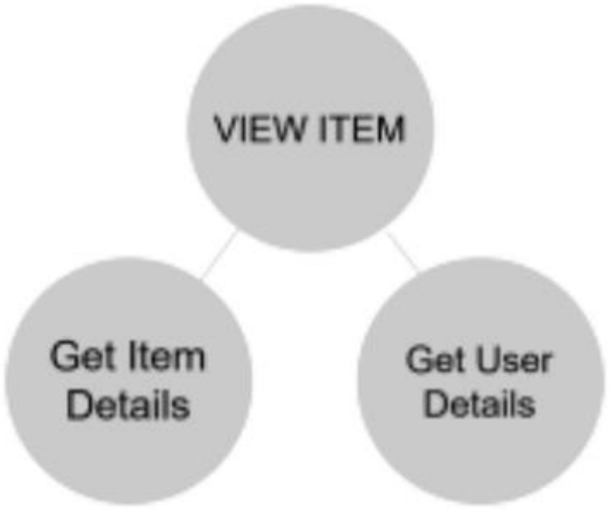
Number of Locks: three

Enabling Conditions: login/register

Frequency: 500

Consistency (ACID): Consistency is not critical. Order is not critical

Subtasks: get item details, get user details



3.7.2 *Abstract Code*

- Begin
- Get items details from Item, user details from User, address from Postal Code, get 'myrank', 'responsetime'
- Use "Calculate User Distance" subtask to get distance between the user
    - IF user's post code != item's owner post code, Then display distance
    - IF 0.0 < distance < 25.0 miles, Then add green background
    - IF 25.00 < distance < 50.0 miles, Then add yellow background
    - IF 50.00 < distance < 100.0 miles, Then add orange background
    - IF distance > 100.0 miles, Then add red background
    - ELSE, Hide distance
- Display all required fields, Different elements to be displayed based on the context
    - IF USER unaccepted trades < 2
        - Then display Propose Trade
    - ELSE
        - Do not display Propose Trade
- On click of Propose Trade, redirect to Trade Proposal
- End

3.8 Propose trade

3.8.1 *Task decomposition*

Lock Type: - Read-only on User and Read and Write lock on Item Tables

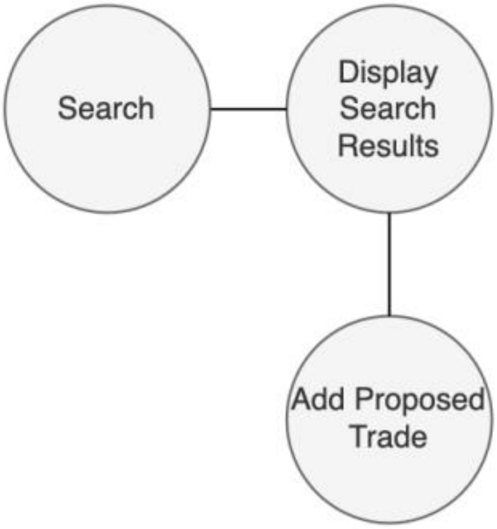
Number of Locks: Two

Enabling Conditions: Login and User has less than 2 unaccepted Trades and has at least 1 Item listed in his/her login

Frequency: 200

Consistency (ACID): We need to ensure, desired item is locked once identified for trade by a user

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



3.8.2 *Abstract Code*

- BEGIN
- Search, Calculate User Distance, Display Search Results
- If counterparty distance >=100.0, a warning message showing distance at-the top should be shown.
- Must display only items that are available for trading (self).
- Provide appropriate mechanism to choose the desired item
- Should display Item number, game type, title, and condition
- Should be ordered by item number
- Once the proposed item is chosen, the user must be provided with a confirm Button to confirm the proposal.
- Once confirmed, the date proposal was made is stored/written into the Trade table
- A confirmation message should be displayed allowing the user to return to the main menu.
- END

3.9 Accept/Reject trades

3.9.1 *Task decomposition*

Lock Type: Read and Write Lock on Trade table, Read lock on USER and ITEM table

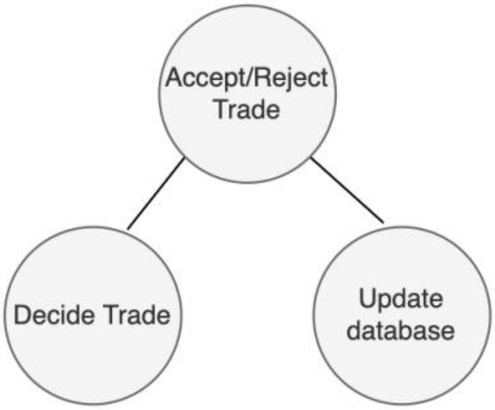
Number of Locks: three

Enabling Conditions: login and has pending proposals

Frequency: 200

Consistency (ACID): available only to the user

Subtasks: Mother task needed. Decomposition is needed. Decide Trade, Update Database



3.9.2 *Abstract Code*

- Begin
- Search, Display Search Results
- List proposed trades where user is the counterparty so they may accept or reject a Trade
- For each proposal, show the date proposed, desired item's title, proposer's nickname, their rank, distance from user (rounded to hundreths) and proposed item title, ordered by proposal date.
- Both item fields should link to the item's detail page
- Accept/Reject mechanism should be provided: (Decide Trade), Accept, Reject

- If trade is accepted, query the USER table and display a dialog with the proposer's Email and first name.
- The acceptance / rejection date of the trade is recorded into the database as part of trade table (Update Database)
    - If rejected, a new trade for the same proposed item and the same desired item cannot be proposed again by the same proposer
    - Accepting a trade removes the item from the listing (Search and Display Search Results) and if no more trades need to be accepted or rejected, the user should be returned to the main menu.
- End

3.10 Trade history

3.10.1 *Task decomposition*

Lock Type: - Read-only on USER, ITEM Read and Write lock on Trade Table

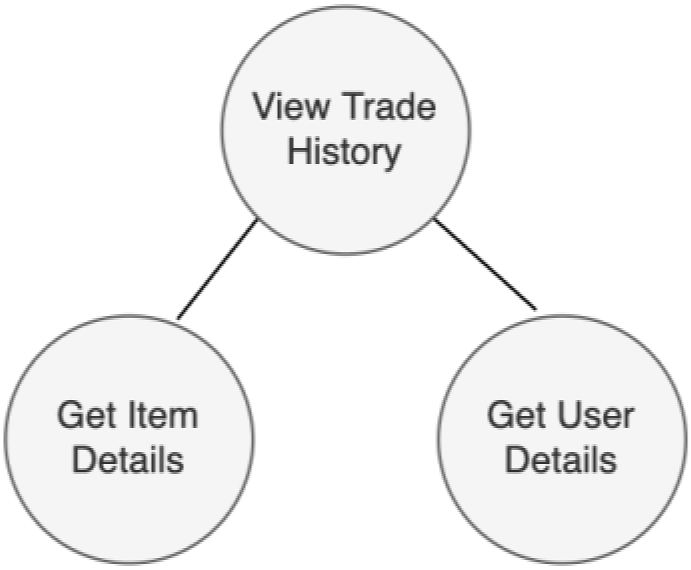
Number of Locks: three

Enabling Conditions: login

Frequency: Not needed as it is available only to the user

Consistency (ACID): 200

Subtasks: Get Item Details, Get User Details



3.10.2 *Abstract Code*

- BEGIN
  - Get users' details from USER
  - Get items details from ITEM
  - Get trades details by USER, ITEM
- Calculate trade summary for user and present as a view:
  - SUM(Accepted, Rejected) AS Total,
  - COUNT(CASE WHEN trade status = 'Accepted') AS Accepted,
  - COUNT(CASE WHEN trade status = 'Rejected') AS Rejected
  - (Rejected/Total) AS Rejected %
  - GROUP BY USER email, role
  - IF Rejected % >= 50%
    - Highlight 'Red'
- Display the following details about a trade,sorted by acceptance/rejection date descending and trade proposed date ascending :
- Proposed date, accepted/rejected date, Trade status, Response Time in days, my role, proposed item, desired item, other user's nickname, link to trade details
- END

3.11 Trade details

3.11.1 *Task decomposition*

Lock Type: Read-only on USER, ITEM Read and Write lock on Trade Table

Number of Locks: three

Enabling Conditions: Log In, View Trade History

Frequency:200

Consistency (ACID): Not needed as it is available only to the user

Subtasks: Mother task needed. Decomposition not needed.

3.11.2 *Abstract Code*

- Begin
- Get users' details from USER
- Get items details from ITEM
- Get Trade details by USER, ITEM
- Display trade details: proposed date, accepted/rejected date, response time in days, trade status, user's role
- Display user details (counterparty): nickname, (Calculate User Distance) IF trade status is accepted
    - Display first name, email
- 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