CORKBOARDIT PROJECT

PHASE 1 REPORT

CS 6400

Fall 2018

Team 004

Table of Contents

CorkBoardIt Data Types

User

Attribute	Data type	Allow Null
Email	String	Not Null
First_Name	String	Not Null
Last_Name	String	Not Null
Pin	String	Not Null

CorkBoard

Attribute	Data type	Allow Null
Date/Time	Datetime	Not Null
Category	String	Not Null
Title	String	Not Null
Visibility	Boolean	Not Null
Password	String	Not Null

PushPin

Attribute	Data type	Allow Null
Date/Time	Datetime	Not Null
URL	String	Not Null
Description	String	Not Null
Tags	list<string>	NULL

Comments

Attribute	Data type	Allow Null
Date/Time	Datetime	Not Null
Comment	String	Not Null

Business Logic Constraints

- Users are not allowed to register for an account for CorkBoardIt
    - Users can only be added by the Database Administrators
- Home Screen should show only Corkboards that the user owns, is watching, and corkboards owned by another user that the logged-in user is following
    - CorkBoard updates only occur when a new PushPin is added to that CorkBoard
    - Recent CorkBoard updates may only display (4) CorkBoards MAX
    - If there are no recent updates, display a message that says "No Updates"
    - Corkboards listed in Recent Updates should be sorted with the most recently updated Corkboard appearing first
    - Corkboard list should be sorted alphabetically by the Corkboard title
    - If user does not own corkboards, display message that says "You have no Corkboards"
    - Hyperlinks are on titles of CorkBoards on MyCorkBoards and Recent CorkBoard Updates
    - Private Corkboards should be indicated in the list
- All passwords do not need to be encrypted
- CorkBoard title must be constrained to a reasonable length
- CorkBoard images are hyperlinked to their respective PushPins
- When adding a CorkBoard, one MUST have a title, category, and visibility (public/private)
- The list of categories is set by the DBA
    - Users can neither delete or add categories to this list
    - Should be ordered alphabetically
- Private CorkBoard
    - No one can ‘watch’
    - PushPins never appear in search results
- Only current owner of CorkBoard can add a PushPin
- When adding a PushPin, one MUST have a valid URL (image), and description
    - URL must link to an image with a valid format (.png, .jpg, .gif)
    - Description must be less than 200 characters
    - Tags must be <= 20 characters
        - Tags cannot contain commas
        - Tags are case insensitive
- PushPins
    - Comments cannot be DELETED
    - List of comments is sorted by the date and time that the comment was added
        - The latest comment appearing first
- Search
    - User can search by 3 fields: tags, description, category of Corkboard
        - Sorted alphabetically by PushPin Description
    - Listed PushPin Descriptions are hyperlinked to their respective pushpins
- Popular Tags
    - Sorted by number of Pushpins associated with tag
    - Listed tags are hyperlinked to their respective search
- CorkBoard Statistics
    - Current users stats should be in RED

CorkBoardIt Task Decomposition with Abstract Code

Login to Private CorkBoard

Task Decomposition

Lock Types: Read-only on CorkBoard
Number of Locks: Single
Enabling Conditions: Enabled when trying to view a private CorkBoard
Frequency: Low
Consistency (ACID): Not critical
Subtasks: Mother task is not needed. No decomposition needed

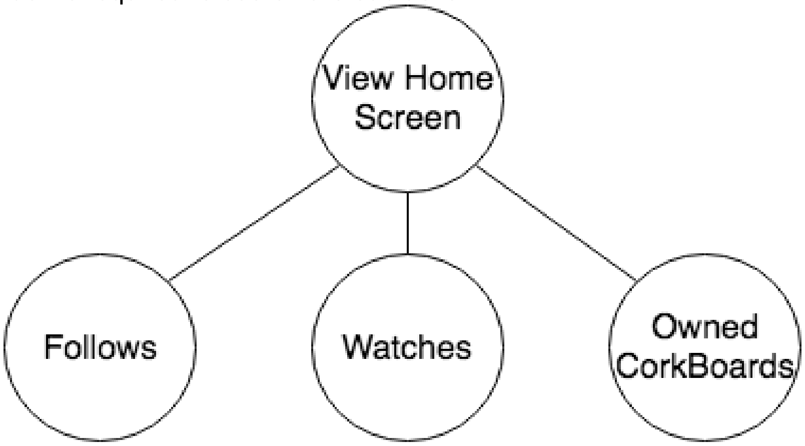
Abstract Code

- User clicked on a private CorkBoard from the **Home Screen** form:
- User enters *password* into input field
- If data validation is successful for *password* field, then:
    - When ***Enter*** button is clicked:
        - Check input *password* against stored *password*
            - If private CorkBoard *password* is incorrect:
                - Return to **Login to Private CorkBoard** form
            - Else:
                - Go to **Corkboard** form
    - When ***Cancel*** button is clicked:
        - Jump to **View Home Screen** task

View Home Screen

Task Decomposition

Lock Types: Read-only lookups on Followed, Watched, and Owned CorkBoards of User
Number of Locks: Several different schema constructs are needed
Enabling Conditions: Enabled upon User login
Frequency: Every login by a User; all three subtasks have the same frequency
Consistency (ACID): Not critical
Subtasks: All tasks must be done but can be done in parallel although order is not necessary. Mother task is required to coordinate subtasks.

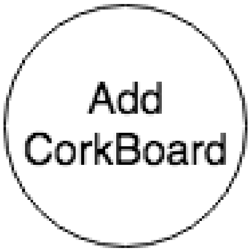


Abstract Code

- Upon user login from the **Login** form:
- Find the current User using their *email*
- Display the User *name* (first and last)
- Find who the User *follows* (by other User email), *watches* (by other User email), and their *owned CorkBoards*
- Order *owned CorkBoards* alphabetically
    - Populate **“*My CorkBoards*”** section with *owned CorkBoards*
- Order *owned CorkBoards*, *follows* CorkBoards, and *watches* CorkBoards by most recently updated and restrict to (4) total corkboards and then populate **“*Recent CorkBoard Updates*”** section
- Display **“*Popular Tags*”**, **“*Popular Sites*”**, **“*CorkBoard Statistics*”**, **“*Add CorkBoard*”**, and **“*PushPin Search*”**
- Upon:
    - Click ***Popular Tags*** button - Jump to **View Popular Tags** task
    - Click ***Popular Sites*** button - Jump to **View Popular Sites** task
    - Click ***CorkBoard Statistics*** button - Jump to **View CorkBoard Statistics** task
    - Click ***Add CorkBoard*** button - Jump to **Add CorkBoard** task
    - Click ***PushPin Search*** button - Jump to **Search for PushPins** task
    - Click any corkboard title in ***My CorkBoards*** section - Jump to **View CorkBoard** task
    - Click any corkboard title of ***Recent CorkBoard Updates*** section - Jump to **View CorkBoard** task

Add CorkBoard

Task Decomposition



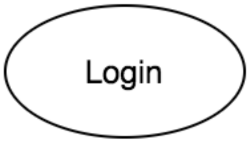
Lock Types: Read-only lookup on *category*
Number of Locks: Single
Enabling Conditions: Enabled when adding a new CorkBoard from **Home Screen** form
Frequency: Low
Consistency (ACID): Not critical
Subtasks: Mother task is not needed. No decomposition needed.

Abstract Code

- User clicked on ***Add CorkBoard*** button from **Home Screen**:
- Run the **Add CorkBoard** task:
    - The **Add CorkBoard** form will be the new form for the current user
    - *Category* list is read from the database and populates the *category* drop down menu
    - User enters *title*, chooses a *category*, and selects radio button of *visibility*
        - If ‘private’ *visibility* is chosen:
            - User must enter a *password*
    - If data validation is successful for *title, visibility*, and potentially *password* then:
        - When ***Add*** button is clicked:
            - If input information is valid:
                - Write information to database
                - Run **View CorkBoard** task
            - Else:
                - Remain on **Add CorkBoard** form displaying error message regarding invalid input

Login

Task Decomposition



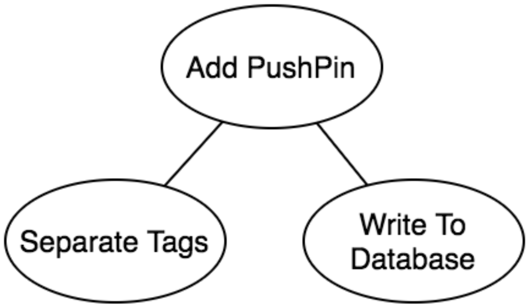
Lock Types: Read-Only on User
Number of Locks: Single
Enabling Conditions: None
Frequency: Low
Consistency (ACID): Not critical
Subtasks: None

Abstract Code

- User enters *Email* and *PIN* into **<u>Login Screen</u>**
- When User clicks ***Login:***
  - Lookup *Email* in database and confirm it's valid
  - If not-valid:
    - Warn User they've entered invalid *Email*
  - If valid:
    - Lookup *PIN* for *Email* and compare to User's entered *PIN*
      - If *PIN* does not match, warn user they've entered an invalid PIN
      - If *PIN* matches User's provided *PIN,* User is directed to **<u>Home Screen</u>**

Add PushPin

Task Decomposition



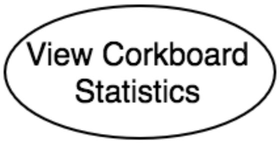
Lock Types: Write-only to PushPin
Number of Locks: Single
Enabling Conditions: User input
Frequency: Medium
Consistency (ACID): Not critical
Subtasks: Separate *Tags*, Write to Database

Abstract Code

- User enters *URL*, *Description*, and *Tags* into **<u>Add PushPin Screen</u>**
- When User clicks ***Add:***
  - Sub-task will separate string of *Tags* by the comma delimiter into a list of strings
  - Sub-task will change all *Tags* to only uppercase for consistency
  - Write *URL*, *Description*, and *List of Tags* to database

View Corkboard Stats

Task Decomposition



Lock Types: Read-Only on CorkBoard
Number of Locks: Single
Enabling Conditions: None
Frequency: Low
Consistency (ACID): Not critical
Subtasks: None

Abstract Code

- When User clicks ***View Corkboard Statistics*** on **Home Screen**
- Read from database for All Users:
  - User Name
  - Count of User's Public CorkBoards
  - Count of User's Public PushPins
  - Count of User's Private CorkBoards
  - Count of User's Private PushPins
- Present this information on all Users in a table format
  - Primary Sort by Most Public CorkBoards
  - Secondary Sort by Most Private CorkBoards
- Display current Users row in red

View Popular Tags

Task Decomposition

Lock Types: Read-Only on PushPin
Number of Locks: Single
Enabling Conditions: None
Frequency: Low
Consistency (ACID): Not critical
Subtasks: None

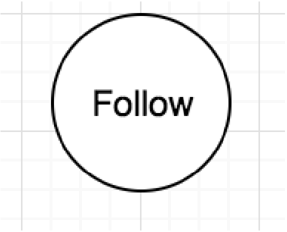
Abstract Code

- When User presses **View Popular Tags** on **Home Screen**
- Read from database for All PushPin *Tags:*
  - Get Count of each *Tag*
  - Get Count of unique CorkBoards
- If CorkBoard contains same *Tag* two or more times, count once
- Display this data in order of *Tag* count, highest to lowest
- Display *Tags* as hyperlinks to search results for *Tag*
- If User presses *Tag* hyperlink, execute **Search for PushPins** task with that *Tag* as keyword

Follow

Task Decomposition

Lock Types: Write-only to Follow Schema
Number of Locks: Single
Enabling Conditions: None
Frequency: High
Consistency (ACID): Consistency is not critical
Subtasks: Mother task is not needed, no decomposition needed, no subtasks



Abstract Code

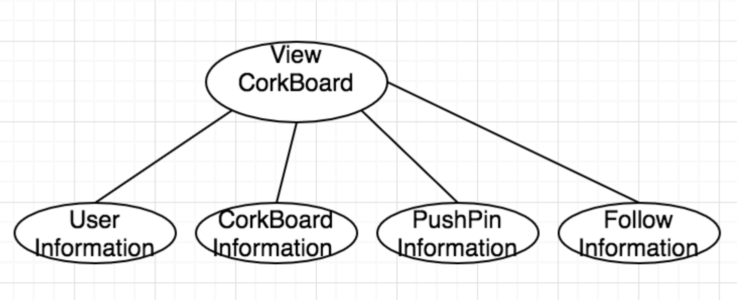
- User clicks on the ***Follow*** button from the **View Corkboard** or **View Pushpin** screen
- Run the **Follow** task:
  - This will add the User who owns the CorkBoard or PushPin to the current User's follow list
  - If on **View Corkboard** Screen:

        - Run **View Corkboard** Task
    - If on **View PushPin** Screen
        - Run **View PushPin** Task

View Corkboard

Task Decomposition

Lock Types: Read from CorkBoard
Number of Locks: 3 different schema constructs are needed
Enabling Conditions: All 3 are enabled by Users login, or successful Private CorkBoard *password* validation
Frequency: High – All 3 frequencies have the same frequency
Consistency (ACID): Consistency is not critical
Subtasks: All tasks must be done but can be done in parallel. Mother task is required to coordinate subtasks. Order is not necessary

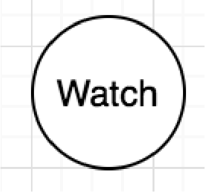


Abstract Code

- When User clicks the ***title*** of a Corkboard on the **Home screen**
- The **View Corkboard** task is run:
    - Read the following from database:
        - PushPins: Collection of pushpins associated with Corkboard
            - Web URL for image
            - Thumbnail for PushPin image
        - User:
            - Owners name
        - CorkBoard:
            - Title of CorkBoard
            - Date last updated (when last PushPin was added)
            - Category of CorkBoard
            - Number of Users watching the board
        - If not Owner of CorkBoard:
            - If following, grey out ***Follow*** button
- Show ***Follow, Add PushPin***, and ***Watch*** tabs/buttons
- Upon:
    - Click ***Follow*** button, jump to **Follow** task
    - Click ***Add PushPin*** button, jump to **Add PushPin** task
    - Click ***Watch*** button, jump to **Watch** task
    - Click ***PushPin*** image, jump to **View PushPin** task

Watch Corkboard

Task Decomposition



Lock Types: Writes to CorkBoard
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, no subtasks

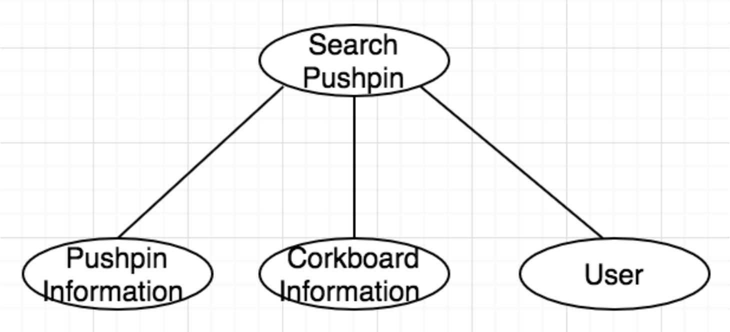
Abstract Code

- User clicks on the ***Watch*** button from the **View Corkboard** screen
- Run the **Watch** task:
    - This will add the corkboard to the current Users watch list

Search PushPins

Task Decomposition

Lock Types: Read only on PushPin, CorkBoard and User entities
Number of Locks: Several different schema constructs are needed
Enabling Conditions: None
Frequency: High – All tasks have the same frequency
Consistency (ACID): Not critical
Subtasks: All tasks must be done mother task is required to coordinate subtasks. Order is necessary

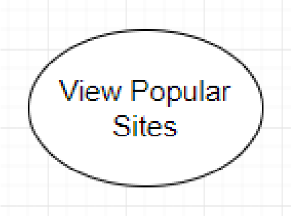


Abstract Code

- User enters *search text* in the text box on the **Home screen**
- When User clicks the ***Pushpin Search*** button on the **Home screen**, **Search Pushpin** task is run
    - Read the following from database:
        - PushPin:
            - Description
            - Tags
        - CorkBoard (associated with PushPin):
            - Category
    - Launch **Results Screen**
    - If *search text* is found in any of the following fields:
        - PushPin description
        - PushPin tags
        - Category of CorkBoard associated
            - List PushPins on results screen with the following read from database:
                - PushPin Description
                - Associated CorkBoard title
                - Owner of CorkBoard
                    - Sorted alphabetically by ***Pushpin description***
                - If User clicks PushPin *description* run **View PushPin** task
            - Except if**:**
                - Associated CorkBoard is private

View Popular Sites

Task Decomposition



Lock Types: Read-only on PushPin
Number of Locks: Single
Enabling Conditions: It is enabled by a Users login
Frequency: Low
Consistency (ACID): Not critical, order is not critical
Subtasks: Mother Task is not needed. No decomposition needed

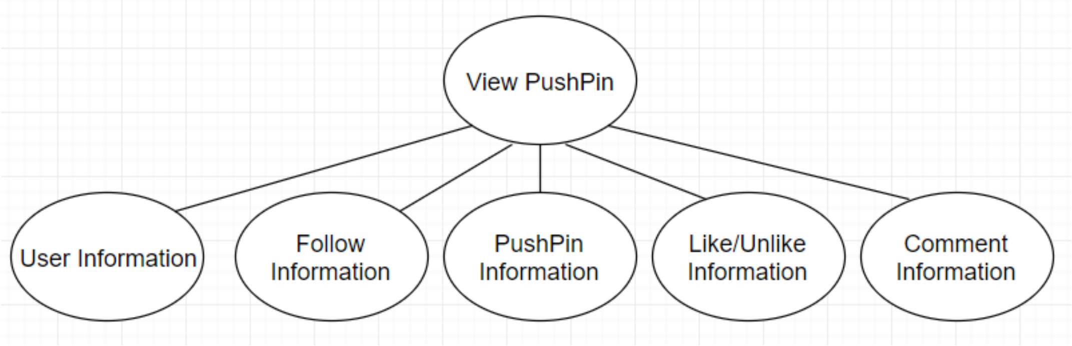
Abstract Code

- When User clicks ***View Popular Sites*** on **<u>Home Screen</u> form**
- Read from database for all PushPins
    - Collect all domain website of *URLs* from all PushPins
        - Count up each unique domain website
- Display simplified *URL* as Site and PushPin Count in order of Count from highest to lowest

View PushPin

Task Decomposition

Lock Types: Read-only on PushPin
Number of Locks: Several different schema constructs are needed
Enabling Conditions: It is enabled by a User's login or PushPin search
Frequency: High
Consistency (ACID): Not critical, order is not critical
Subtasks: Mother Task is needed. Decomposition is needed



Abstract Code

- User clicks image hyperlink in **<u>CorkBoard form</u>** which runs the **View PushPin** task:
    - Find User Owner of PushPin; Display User Owner First and Last Name
    - If not Owner of PushPin, Read Follow Information:
        - If Following, grey out "***Follow***" button
    - Find and Display Date/Time of PushPin
    - Find and Display Title of PushPins CorkBoard;
        - If User clicks ***Title***, display the associated CorkBoard view
    - Find *URL* of PushPin Image and Display the domain name of *URL*
    - Find the image *URL* of PushPin and display the image
    - Find and display the *Description* of PushPin
    - Find and display the *Tag* of PushPin, split by comma
    - Look up current users likes:
        - If already liked:
            - Display "***Unlike!"*** button
        - Else:
            - Display "***Like!"*** button

- Find and display the Comments of PushPin
    - Display Comments in order of date/time posted; most recent first
    - Display post comment button

Like/Unlike

Task Decomposition

Lock Types: Read and Write from Likes Schema

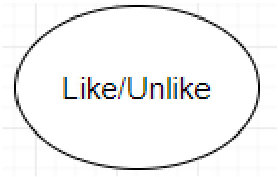
Number of Locks: Single

Enabling Conditions: It is enabled by a User's login and Like/Unlike request.

Frequency: High

Consistency (ACID): Consistency is critical. It is possible for the user to change his or her mind multiple times and the application should not prevent this.

Subtasks: No subtasks



Abstract Code

- Run **Like/Unlike** Task on **<u>Pushpin</u>** form
    - Find the User Watcher's *Email* who Like the PushPin; Display User Watcher's First and Last Name;
    - Find if User Watcher has liked the PushPin
        - If User Watcher has unliked the PushPin,
            - If User Watcher click "***Like!***" button, Write the User Watcher into database
        - If User Watcher has liked the PushPin,
            - If User Watcher click "***Unlike!***" button, remove the User Watcher from database
    - Run **View PushPin** task

Post Comment

Task Decomposition

Lock Types: Write only on Comments schema

Number of Locks: Single

Enabling Conditions: None

Frequency: High

Consistency (ACID): Consistency is not critical.

Subtasks: Mother task is not needed; no decomposition is needed. Order is not critical.

Abstract Code

- Run **Post Comment** Task on **<u>Pushpin</u>** form
    - Users enter the text of *Comment* into input field
        - If "***Post Comment***" is clicked, Write the text of *Comment*, User's Email and Date/Time into database
    - Run **View PushPin** task