Table Of Contents

Data Types

User

Attribute	**Data Type**	**Allow Null**
Pin	Int	Not NULL
Email	String	Not NULL
FirstName	String	Not NULL
LastName	String	Not NULL
Follower	List <String>	NULL
OwnedCorkBoard	List <String>	NULL
WatchedCorkBoard	List <String>	NULL

CorkBoard

Attribute	**Data Type**	**Allow Null**
DateTime	Date	Not NULL
Title	String<String<50 characters or less>	Not NULL
Visibility	Boolean	Not NULL

Category	String	Not NULL
Password	String	NULL
WatchedByEmail	List <String>	NULL
OwnedByEmail	String	Not NULL
PushPin	List <Date>	NULL

PushPin

Attribute	**Data Type**	**Allow Null**
DateTime	Date	Not NULL
Tag	List <String<20 characters or less>>	NULL
URL	String	Not NULL
Description	String<200 characters or less>	Not NULL
LikedByUser	List <String>	NULL
Comment	List <Date>	Not NULL
CorkBoard	String	Not NULL

Comment

Attribute	**Data Type**	**Allow Null**
DateTime	Date	Not NULL
Text	String	Not NULL
Posted By	String	Not NULL
PushPin	Date	Not NULL

Constraints

1. All types of users must use password to watch private CorkBoard
2. Users cannot follow themselves
3. Comments cannot be deleted

4. Users cannot like their own post
5. Owner cannot watch their own CorkBoard
6. Categories must initially have the 14 categories: Education, Architecture, Home & Garden, People, Travel, Photography, Sports, Pets, Technology, Other, Food & Drink, Art
7. A user cannot change the CorkBoard categories but a database admin can
8. Url is publicly accessible, is correct, and contains a known image format extension
9. DateTime of CorkBoard represents when CorkBoard is updated/created whichever is most current
10. When user adds new push pin to CorkBoard, CorkBoard date & time is updated
11. DateTime on PushPin represents when it was added to the CorkBoard
12. Date & time on comment represent when it is first added
13. When the user adds a CorkBoard, the user becomes the owner of the CorkBoard
14. A tag cannot contain a comma but can have any other characters
15. A CorkBoard with 'False' as visibility cannot have an empty password (null)
16. Only an owner of a CorkBoard can add a PushPin
17. Newly created CorkBoards are empty

Variable Representations for Abstract Code

1. '$VariableName' represents the ID of the variable that is being queried by or passed through in navigations between screen
2. '**BoldedVariableName**' represent datatypes or instance of datatype
3. '*ItalicizedVariableName'* represent buttons or ui with possible interactions (such as hyperlinks)
4. 'UnderlinedVariableName' represent screens of the requirement

Logging In

Task Decomposition

1. Locktype: Read Only on users
2. Number of locks: Single
3. Enabling Condition: None
4. Frequency: 200 a day
5. Consistency (Acid): Not critical, order is not critical
6. Subtasks: Mothertask is not needed

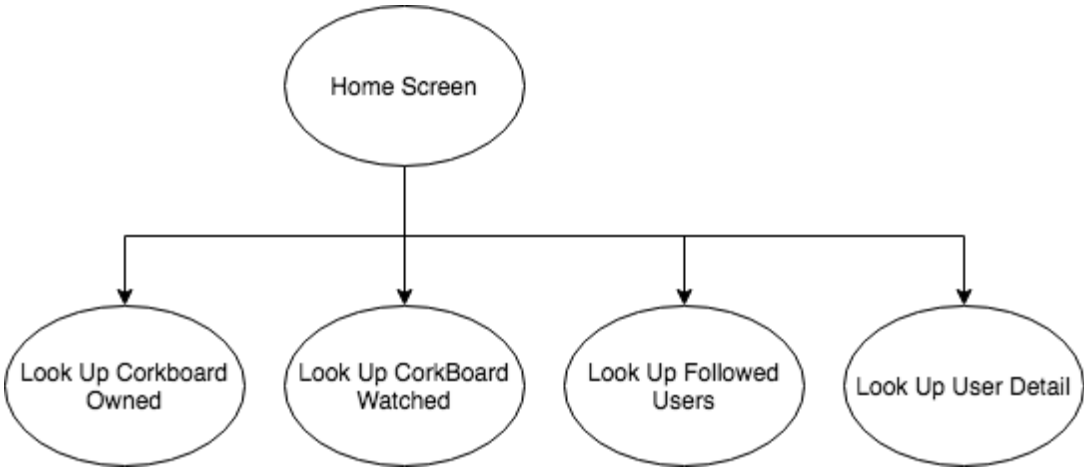
Abstract Code

1. User enters email and pin into the input fields
2. If validation is successful for both email and pin, then
    a. When *Login* is clicked:
        i. If **User** record is not found or pin is incorrect:
            1. An error message is displayed and user is returned to the login screen
        ii. Else:
            1. Go to Home Page

Home Screen

Task Decomposition

1. Locktype: Read only on Users, Read only on CorkBoard
2. Number of locks: Several different schema constructs are needed
3. Enabling Conditions:
    a. Updates and its subtasks/User lookup/My Corkboard are enabled when users log in or view home screen
    b. Search is enabled when user enters text and presses *PushPin Search* button
4. Frequency:
    a. High frequency for Followed Users Corkboard and Watched Corkboard
    b. Low frequency for Owned CorkBoard updates, My CorkBoard, and User
    c. Medium frequency for Search
5. Consistency (ACID): Not critical
6. Subtasks: All tasks must be done, but can be done in parallel, mother task is needed for update and home screen. Order does not matter



Abstract Code

1. **User** logged in gets taken to Home Screen, where $UserID is the id of the current user using the system from the HTTP session/cookie
2. Run the Home Screen task:
    a. Run **User** subtask:
        i. Query information about **User** of $UserID and their profile
            1. Find the current **User** using **User.email**; Display **User.FirstName** and **User.LastName**
    b. Run Updates subtask
        i. Query information about the **CorkBoards** that are updated through: owned by **User** of $UserID, watched by **User** of $UserID, and owned by **Users** followed by **User** of $UserID, where $CorkBoardIDs represent the queried **CorkBoards**
            1. Find the current owned **Corkboards** using **User.OwnedCorkBoard**
                a. Display on My Corkboard
            2. Find the current watched **Corkboards** using **User.WatchedCorkBoard**
            3. Find the current owned **Corkboards** of **Users** followed by **User** of $UserID using foreach **User** of **User.Followed,** return **User.OwnedCorkBoard**
        ii. Filter **Corkboards** of $CorkBoardIDs by top 4 most recent date
            1. Display top 4 **Corkboard** in descending order using **Corkboard.DateTime**
    c. If *PushPin Search* is pressed and searchbox contains text:
            1. Go to PushPin Search Results
    d. If *Popular Tags* button is pressed:
        i. Go to Popular Tags
    e. If *Add CorkBoard* button is pressed:
        i. Go to Add CorkBoard
    f. If *CorkBoard hyperlink* is pressed:
        i. If CorkBoard is private:
            1. Go to Password Protected
        ii. Else:
            1. Go to View CorkBoard

Password Protected

Task Decomposition

1. Locktype: Read Only on Corkboard
2. Number of locks: Single
3. Enabling Condition: When user clicks on *Corkboard hyperlink*
4. Frequency: 20 a day; Few

5. Consistency (Acid): Not critical, order is not critical
6. Subtasks: Mothertask is not needed

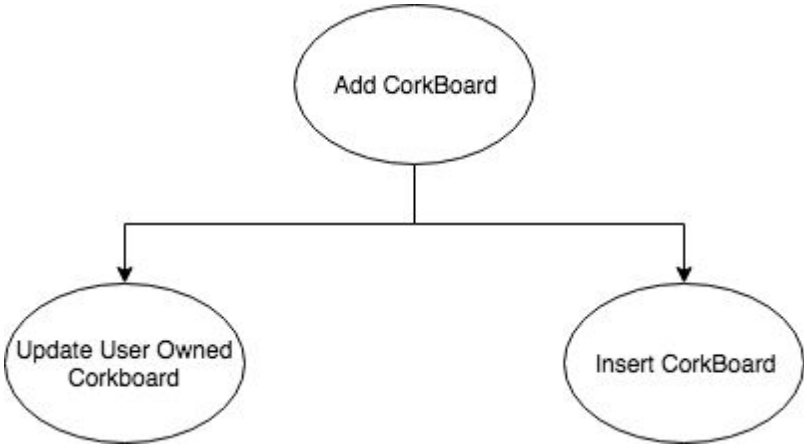
Abstract Code

1. User enters password into the input field
2. If validation is successful, for password for CorkBoard of $CorkBoardID the user is trying to view, then
    a. When *OK* is clicked:
        i. If password is incorrect:
            1. Go to Home Screen
        ii. Else:
            1. Go to View CorkBoard
3. If *Cancel* is clicked:
    a. Go to Home Screen

Add CorkBoard

Task Decomposition

1. Locktype: Write on User, Write on Corkboard
2. Number of Locks: Several different schema constructs are needed
3. Enabling Condition: When user clicks on *Add CorkBoard* button
4. Frequency: 10 a day; Few
5. Consistency (Acid): Not critical
6. Subtask: All tasks must be done but can be done in parallel, Mothertask is needed



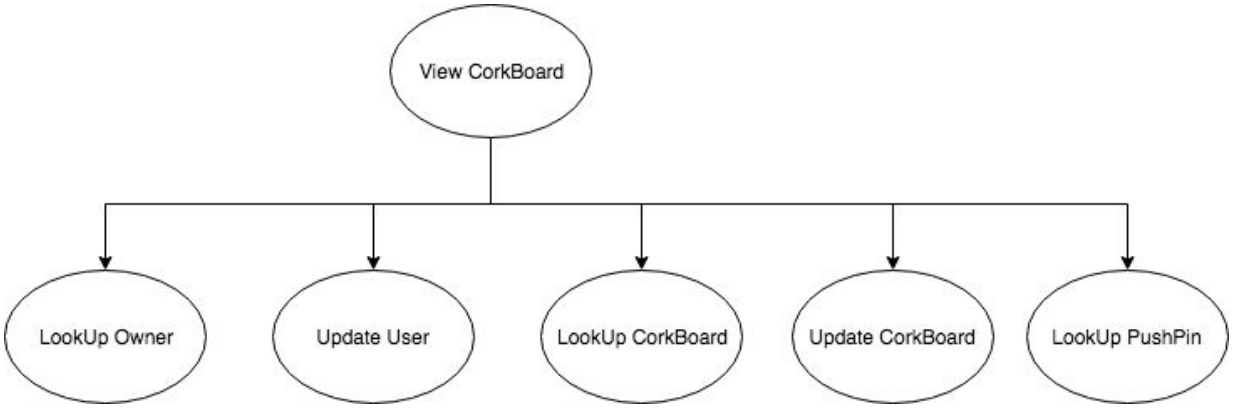
Abstract Code

1. User of $UserID enters Add CorkBoard form from clicking on *Add CorkBoard* button
2. If *Add* button is selected and title, category, public/private with password is filled:
    a. If Title input is greater than 50 characters:
        i. Disable the *Add* button
    b. Else:
        i. Insert **CorkBoard,** where $CorkBoardID represents ID of **CorkBoard :**
            1. Set **CorkBoard.Title** with title from title textbox
            2. Set **CorkBoard.Category** with category from category selection
            3. If public is selected:
                a. Set **CorkBoard.Visibility** to **True**
            4. Else:
                a. Set **CorkBoard.Visibility** to **False**
                b. Set **CorkBoard.Password** to password from password textbox
        ii. Update $UserID's **User.OwnedCorkBoard** with $CorkBoardID's **CorkBoard.Title**
        iii. Go to View CorkBoard screen of the new CorkBoard of $CorkBoardID

View CorkBoard

Task Decomposition

1. Locktype: Read & Write on User, Read and Write on Corkboard, Read on PushPin
2. Number of Locks: Several different schema constructs are needed
3. Enabling Condition: When user clicks on *Add* button from *Add CorkBoard* or when user clicks on *CorkBoard hyperlink*
4. Frequency: 200+ a day
5. Consistency (Acid): Not critical
6. Subtask: All tasks must be done but can be done in parallel, Mothertask is needed



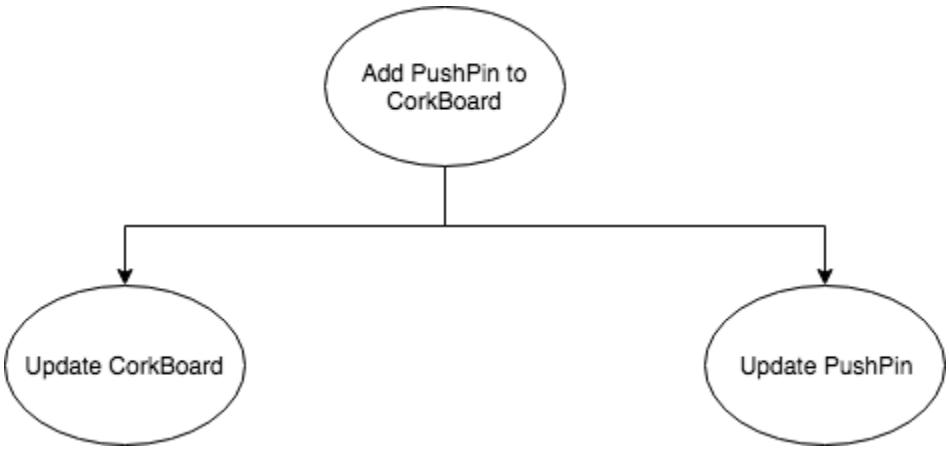
Abstract Code

1. **User** of $UserID enters View CorkBoard on **CorkBoard** of $CorkBoardID from clicking *Add* on Add CorkBoard screen, or clicking on *CorkBoard hyperlink* from Home Screen
2. Query $UserID's **User.FirstName** and **User.LastName**
    a. Find **User** from **CorkBoard.OwnedByEmail**
    b. Display First and Last Name as owner
3. Query $CorkBoardID's details
    a. Find title from **CorkBoard.Title;** Display title
    b. Find category from **CorkBoard.Category;** Display category
    c. Find date & time from **CorkBoard.DateTime;** Display date & time
    d. Find number of watchers by counting number of **CorkBoard.WatchedByEmail**; Display count of watchers
4. Query $CorkBoardID's **PushPin** images
    a. Find **PushPins** in **CorkBoard** by finding foreach **PushPin date** in **CorkBoard.PushPin** matching to **PushPin.Date** and **PushPin.CorkBoard** matching to **CorkBoard.Title**; Display **PushPin.URL** as image
5. If **PushPin** image is clicked:
    a. Go to View PushPin screen
6. If *Follow* button is clicked and **User.Email** is not the same as **CorkBoard.OwnedByEmail**:
    a. Get **User.Email** from owner query, Insert **User.Email**(owner) into **Followed** of **User**
7. If **User.Email** is equal to **CorkBoard.OwnedByEmail**:
    a. *Watch* button is disabled
8. If *Watch* button is clicked:
    a. Insert **CorkBoard.Title** into **User.WatchedCorkBoard**
    b. Insert **User.Email** into **CorkBoard.WatchedByEmail**
9. If *Add PushPin* button is clicked and **User.Email** equals **CorkBoard.OwnedByEmail**:
    a. Go to Add PushPin to CorkBoards form

Add PushPin to CorkBoard

Task Decomposition

1. Locktype: Write on Corkboard, Write on PushPin
2. Number of Locks: Several different schema constructs are needed
3. Enabling Condition: When owner presses *Add PushPin* button
4. Frequency: 10 a day; Few
5. Consistency (Acid): Not critical; Even if PushPin is being written to before CorkBoard is queried
6. Subtask: All tasks must be done but can be done in parallel, Mothertask is needed



Abstract Code

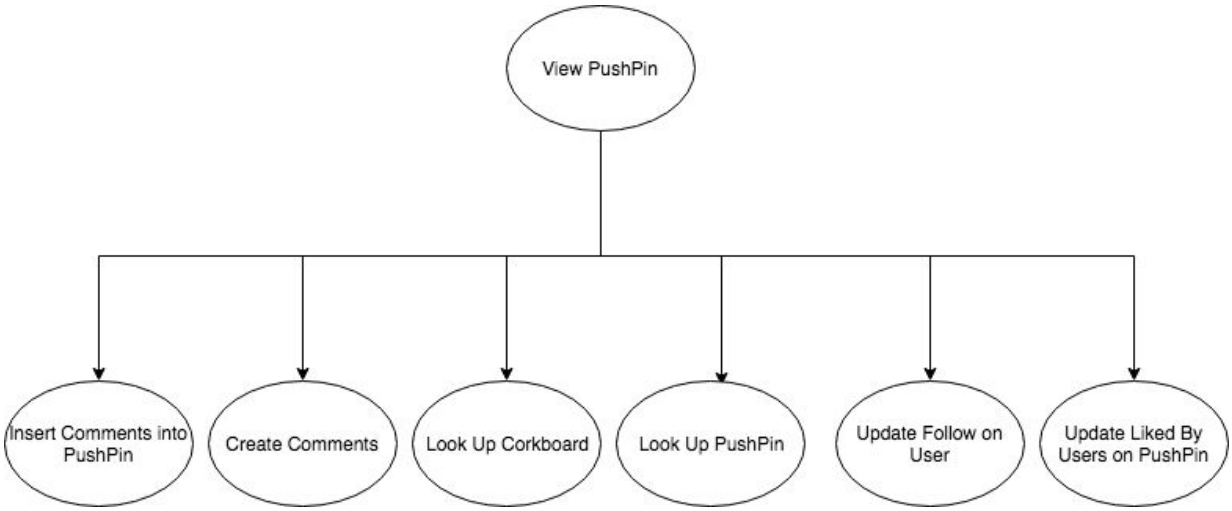
1. User of $UserID enters Add PushPin form from clicking on *Add PushPin* button
2. Display **CorkBoard.Title** from **CorkBoard** of $CorkBoardID for the **CorkBoard** where *Add PushPin* was clicked on
3. User clicks *Add PushPin* on
4. If *Add* button is selected and valid url, description, and tags (comma separated) are filled:
    a. If description is greater than 200 characters && tag(s) are less than 20 characters each:
        i. Disable the *Add* button
    b. Else:
        i. Insert **PushPin:**
            1. Set **PushPin.URL** with url textbox
            2. Separate tags by comma; Foreach tag
                a. Set **PushPin.Tag** by tag
            3. Set **PushPin.Description** with description from textbox
            4. Set **PushPin.CorkBoard** with **CorkBoard.Title** of $CorkBoardID
        ii. Update **CorkBoard** of $CorkBoardID**:**
            1. Set **CorkBoard.PushPin** with **PushPin.DateTime**
            2. Set **CorkBoard.DateTime** with **PushPin.DateTime**

iii. **User** is returned to View CorkBoard Screen

View PushPin

Task Decomposition

1. Locktype: Read & Write on PushPin, Read & Write on Comment, Read & Write on User, Read on CorkBoard
2. Number of Locks: Several different schema constructs are needed
3. Enabling Condition: When user clicks on PushPin image from CorkBoard
4. Frequency: 200 a day
5. Consistency (Acid): Not critical; Even if PushPin is being written to before CorkBoard is queried
6. Subtask: All tasks must be done but can be done in parallel, Mothertask is needed



Abstract Code

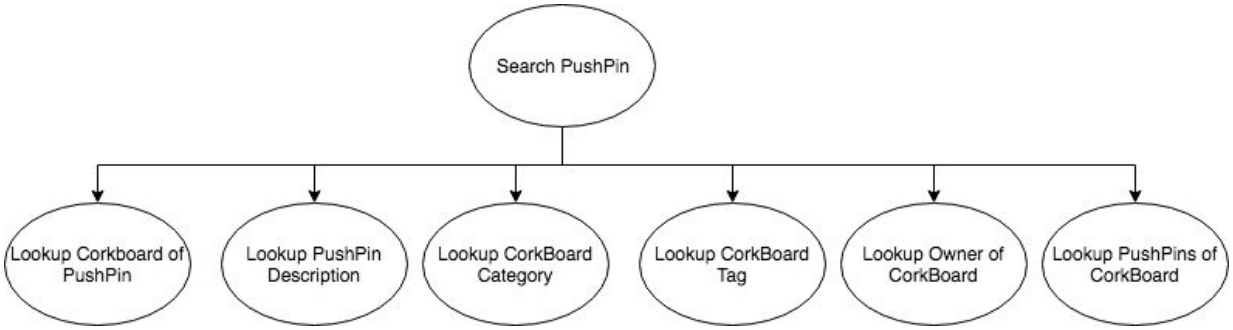
1. **User** of $UserID enters **View PushPin** screen for $PushPinID from clicking on the image in **View CorkBoard** of $CorkBoardID
2. Query owner's User.FirstName and User.LastName from $CorkBoardID's Owner
    a. Find **User** from **Corkboard.OwnedByEmail** of $CorkBoardID where image was clicked on
    b. Display **User.FirstName** and U**ser.LastName**
3. Query PushPin details
    a. Find **DateTime** from **PushPin.DateTime** for the image that was clicked on
    b. Display **PushPin.DateTime**
    c. Find the **CorkBoard** the **PushPin** belongs to, where **PushPin.Corkboard** matches **CorkBoard.Title** of $CorkBoardID
        i. *Hyperlink* back to the **CorkBoard** of $CorkBoardID
    d. Find image URL from **PushPin.URL**

    e. Display root **PushPin.URL**
    f. Find image from **PushPin.URL**
    g. Display image from **PushPin.URL** via http GET request
        i. *Hyperlink* image to the original site
    h. Find description from **Pushpin.Description**
    i. Display **Pushpin.Description**
    j. Find tags from **Pushpin.Tag**
    k. Display all **Pushpin.Tag**
        i. Sort tags alphabetically
    l. Find **Users** who liked the PushPin from **User.Email** that matches **PushPin.LikedByUser**
    m. Display **User.FirstName** and **User.LastName**
    n. Find comments on PushPin from **Comment.Date Pushpin.Comment**
        i. Display **Comment.PostedBy** and **Comment.Text**
        ii. Sort comments in descending order
4. If *Follow* is click and **CorkBoard.OwnedByEmail** of $CorkBoardID is not equal to **User.Email** of $UserID
    a. Add **Corkboard.OwnedByEmail** of to **User.Follower** of $UserID
5. If *Like!/Unlike* Is clicked and **CorkBoard.OwnedByEmail** of $CorkBaordIDis not equal to **User.Email** of $UserID
    a. If User clicks *Like!*
        i. Add **User.FirstName** and **User.LastName** into **Pushpin.LikedByUser**
        ii. Change *Like!* Button to *Unlike*
    b. If User clicks *Unlike*
        i. Remove **User.FirstName** and **User.LastName** into **Pushpin.LikedByUser**
        ii. Change *Unlike* button to *Like!*
6. If *Post Comment* is clicked
    a. Insert new Comment in **PushPin.Comment**
        i. Insert the text entered in the *Enter Comment* box as **Comment.Text**
        ii. Insert the **User.FirstName** and **User.LastName** as **Comment.PostedBy**
        iii. Insert the current Date and Time as **Comment.Date&Time**

Searching For PushPins

Task Decomposition

1. Locktype: Read on PushPin, Read on CorkBoard
2. Number of Locks: Several different schema constructs are needed
3. Enabling Condition: When user enters text in the search box and clicks *PushPin Search*
4. Frequency: 200 a day
5. Consistency (Acid): Not critical
6. Subtask: All tasks must be done but can be done in parallel, Mothertask is needed



Abstract Code

1. User of $UserID enters Searching For PushPins results from click *PushPin Search* on *HomeScreen*
2. Find matched **PushPin** of $PushPinID through **PushPin.Description** or **PushPin.Tag** containing searched text
    a. Find associated **CorkBoard** of $CorkBoardID from where **CorkBoard.PushPin** matches **PushPin.Date** and **PushPin.CorkBoard** matches **CorkBoard.Title**
    b. If **CorkBoard.Visibility** is **False**:
        i. Do not return result
    c. Find associated **User** from where **User.email** matches **CorkBoard.OwnedByEmail**
    d. Sort results by **PushPin.Description** alphabetically
        i. Display **PushPin.Description** under PushPin Description Column
            1. *Hyperlink* to View PushPin screen
        ii. Display **CorkBoard.Title** under CorkBoard Column
        iii. Display **User.FirstName, User.LastName** under Owner Column
3. Find matched **CorkBoard** through **CorkBoard.Category** containing searched text and **CorkBoard.Visibility** is **True**
    a. Find associated **User** from where **User.email** matches **CorkBoard.OwnedByEmail**
    b. Find associated **PushPin** from where **CorkBoard.PushPin** matches **PushPin.Date** and **PushPin.CorkBoard** matches **CorkBoard.Title**
    c. Sort results by **PushPin.Description** alphabetically
        i. Display **PushPin.Description** under PushPin Description Column
            1. *Hyperlink* to View PushPin screen
        ii. Display **CorkBoard.Title** under CorkBoard Column
        iii. Display **User.FirstName, User.LastName** under Owner Column

Popular Tags

Task Decomposition

1. Locktype: Read on PushPins
2. Number of Locks: Single
3. Enabling Condition: When User clicks *Popular Tags* on Main Menu
4. Frequency: 10 a day; Few
5. Consistency (Acid): Not critical, read only
6. Subtask: Mothertask is not needed

Abstract Code

1. User enters Popular Tags from clicking *Popular Tags* on Main Menu
2. Query all PushPins
    a. Find count of all **Pushpin.Tag**
        i. Limit to top five highest counts
        ii. Order in descending order
    b. Display each **Pushpin.Tag**
    c. Display count for each **Pushpin.Tag**
    d. Display count for unique **Pushin.Corkboard** containing that tag
3. If specific tag is clicked
    a. Go to Searching For PushPins for that specific tag

Popular Sites

Task Decomposition

1. Locktype: Read on Pushpins
2. Number of Locks: Single
3. Enabling Condition: When User clicks *Popular Sites* on Main Menu
4. Frequency: 10 a day; Few
5. Consistency (Acid): Not critical, read only
6. Subtask: Mothertask is not needed

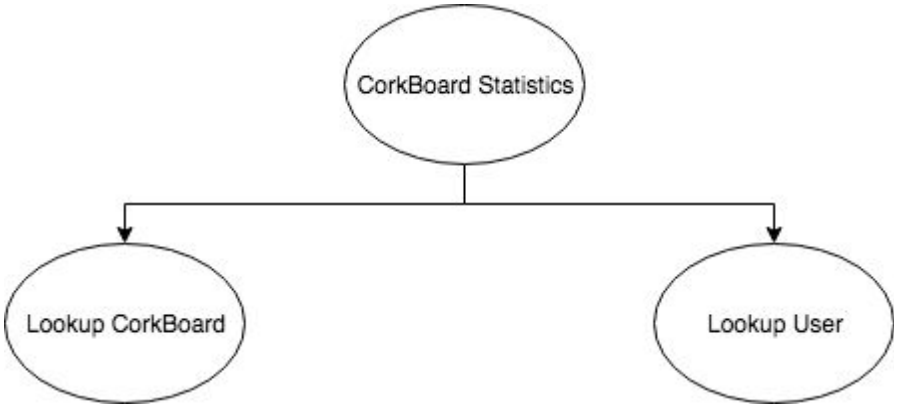
Abstract Code

4. User enters Popular Sites from clicking *Popular Sites* on Home Screen
5. Query all PushPins
    a. Find Count of all **Pushpin.URL**
        i. Limit to top four highest counts
        ii. Order in descending order
        iii. Site is defined as anything after "http://" or "https://" upto the first "/" in the address
    b. Display each **Pushpin.URL**
    c. Display Count for each **Pushpin.URL**

CorkBoard Statistics

Task Decomposition

1. Locktype: Readonly on CorkBoard, Readonly on Users
2. Number of Locks: Several different schema constructs are needed
3. Enabling Condition: When user presses *CorkBoard Statistics* from Main Menu
4. Frequency: 10 a day; Few
5. Consistency (Acid): Not critical.
6. Subtask: All tasks must be done but can be done in parallel, Mothertask is needed



Abstract Code

1. User of $UserID enters CorkBoard Statistics by clicking *CorkBoard Statistics* from Home Screen
2. Query information on all **CorkBoards** for all **Users**:

a. For each **User**:
    i. Get count of **User.OwnedCorkBoard**
    ii. For each **CorkBoard** in **User.OwnedCorkBoard**
        1. Get count of **CorkBoard.Visibility** equals False
            a. For each **CorkBoard.Visibility** equals False
                i. Get count of **CorkBoard.PushPins**
        2. Get count of **CorkBoard.Visibility** equals True
            a. For each **CorkBoard.Visibility** equals True
                i. Get count of **CorkBoard.PushPins**
    iii. Display **User.Firstname** and **User.Lastname** under User column
        1. If **User.Email** matches $UserID's **User.Email**:
            a. Highlight **User.FirstName** and **User.LastName** in red
    iv. Display count of **CorkBoards** where **CorkBoard.Visibility** is **True** under Public CorkBoards column
    v. Display count of **PushPins** belonging to **CorkBoard** where **CorkBoard**.**visibility** is **True** under Public PushPins column
    vi. Display count of **CorkBoards** where **CorkBoard.Visibility** is False under Private CorkBoards column
    vii. Display count of **PushPins** belonging to CorkBoard where **CorkBoard.visibility** is False under Private **PushPins** column