

CORKBOARDIT PROJECT

PHASE 2 REPORT

CS 6400

Fall 2018

Team 04

Table of Contents

CorkBoardIt Task Abstract Code	2
View Home Screen	2
Add CorkBoard	4
Login to Private CorkBoard	4
Login	5
Add PushPin	5
View Corkboard Stats	5
View Popular Tags	6
Follow	6
View Corkboard	6
Watch Corkboard	7
Search PushPins	8
View Popular Sites	8
View PushPin	8
Like/Unlike	10
Post Comment	10

CorkBoardIt Task Abstract Code

View Home Screen

- Upon user login from the **Login** form:
- Run **Home Screen** task for logged in **User**
- Assumed that current **User** \$Email is available
- Find **User** full name (first and last) and display it:

```
SELECT CONCAT(fname, ' ', lname) AS full_name FROM `User` WHERE email = '$Email';
```

- Find *Corkboards* the current **User** owns and display them in the My Corkboards section:
 - Should be ordered alphabetically, indicated if the *Corkboard* is private, and the number of pushpins associated with the *Corkboard*

```
SELECT title, 'private' AS type,  
(SELECT COUNT(*) FROM Pushpin  
WHERE owner_email = prv.owner_email  
AND corkboardID = prv.corkboardID) AS pushpin_count  
FROM PrivateCorkboard as prv  
WHERE prv.owner_email = '$Email'  
UNION ALL  
SELECT title, 'public',  
(SELECT COUNT(*) FROM Pushpin  
WHERE owner_email = pub.owner_email  
AND corkboardID = pub.corkboardID) AS pushpin_count  
FROM PublicCorkboard as pub  
WHERE pub.owner_email = '$Email'  
ORDER BY title;
```

- Find who the **User** *follows*, *watches*, and their *owned CorkBoards* and display them in Recent Corkboard Updates section
 - Should be ordered by most recently updated and restricted to (4) total corkboards displayed

```
SELECT user_cork_push.full_name, user_cork_push.title, user_cork_push.recent_date,  
user_cork_push.type  
FROM  
(SELECT CONCAT(`User`.fname, ' ', `User`.lname) AS full_name, `User`.email,  
pub.corkboardID, pub.title, MAX(pin.pushpin_date) AS recent_date, 'public' AS type  
FROM `User`  
INNER JOIN PublicCorkboard AS pub ON pub.owner_email = `User`.email  
INNER JOIN Pushpin AS pin ON pin.owner_email = pub.owner_email  
AND pin.corkboardID = pub.corkboardID  
GROUP BY full_name, `User`.email, pub.corkboardID, pub.title, type  
UNION ALL  
SELECT CONCAT(`User`.fname, ' ', `User`.lname) AS full_name, `User`.email,  
prv.corkboardID, prv.title, MAX(pin.pushpin_date) AS recent_date, 'private' as type  
FROM `User`  
INNER JOIN PrivateCorkboard AS prv ON prv.owner_email = `User`.email  
INNER JOIN Pushpin AS pin ON pin.owner_email = prv.owner_email  
AND pin.corkboardID = prv.corkboardID  
GROUP BY full_name, `User`.email, prv.corkboardID, prv.title, type)  
AS user_cork_push  
WHERE user_cork_push.email IN (
```

```
SELECT Follows.followee_email FROM Follows
INNER JOIN `User` ON `User`.email = Follows.followee_email
WHERE Follows.follower_email = '$Email')
OR (user_cork_push.email, user_cork_push.corkboardID) IN (
SELECT `User`.email, Corkboard.corkboardID FROM Watches
INNER JOIN Corkboard ON Corkboard.corkboardID = Watches.corkboardID
INNER JOIN `User` ON `User`.email = Corkboard.owner_email
WHERE Watches.watcher_email = '$Email')
OR user_cork_push.email = '$Email'
ORDER BY user_cork_push.recent_date DESC
LIMIT 4;
```

- 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

- User clicked on **Add CorkBoard** button from **Home Screen**:
- The current User is identified by \$Email
- 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

```
SELECT category FROM Category;
```

- User enters *title* (\$Title), chooses a *category* (\$Category), and selects radio button of *visibility*
 - If ‘private’ *visibility* is chosen:
 - User must enter a *password* (\$Password)
 - If data validation is successful for *title* and *visibility* and potentially *password*:
 - When **Add** button is clicked:
 - Create a corkboard and get its associated ID

```
INSERT INTO Corkboard(owner_email) VALUES ('$Email');
```

```
SET @createdCorkboardID = (
SELECT corkboardID FROM Corkboard
WHERE owner_email = '$Email'
ORDER BY corkboardID DESC
LIMIT 1);
```

- If radio button for ‘private’ was chosen make Corkboard a Private Corkboard and write appropriate information to database

```
INSERT INTO PrivateCorkboard(corkboardID, owner_email, title, category, password)
VALUES (@createdCorkboardID, '$Email', '$Title', '$Category', '$Password');
```

Phase 2 Report CS 6400 – Fall 2018 Team 04

- Else make Corkboard a Public Corkboard and write appropriate information to database

```
INSERT INTO PublicCorkboard(corkboardID, owner_email, title, category)
VALUES (@createdCorkboardID, '$Email', '$Title', '$Category');
```

- Run **View CorkBoard** task
- Else:
 - Remain on **Add CorkBoard** form displaying error message regarding invalid input

Login to Private CorkBoard

- User clicked on a private **CorkBoard** from the **Home Screen** form:
- Run the **Login to Private Corkboard** task:
- Assuming current Corkboards '\$CorkboardID' is accessible
- User enters *password* (\$Password) into input field of **Private CorkBoard Login** form
- If data validation is successful for *password* field, then:
 - When **Enter** button is clicked:

```
SELECT password FROM PrivateCorkboard WHERE corkboardID = '$CorkboardID';
```

- If PrivateCorkboard.password != '\$Password':
 - Return to **Home Screen** form
- Else:
 - Go to **Corkboard** form
- When **Cancel** button is clicked:
 - Jump to **View Home Screen** task

Login

- User enters \$email and \$PIN into **Login Screen**
- When **User** clicks **Login**:

```
SELECT pin, email FROM 'User' WHERE email = '$email';
```

- If query has no result:
 - Warn **User** they've entered invalid *Email*
- If email = \$email:
 - If *PIN* != \$PIN: warn **User** they've entered an invalid *PIN*
 - If *PIN* = \$PIN: **User** is directed to **Home Screen**

Add PushPin

- Assumed that current **User** \$Email is available
- **User** enters *url* (\$URL), *description* (\$Description), and *tags* into **Add PushPin Screen**
- When **User** clicks **Add**:
 - Sub-task will separate string of *tags* (\$Tag) by the comma delimiter into a list of strings
 - Sub-task will change all *tags* to only uppercase for consistency
 - Add user information to database

```
INSERT INTO Pushpin(owner_email, corkboardID, url, description)
VALUES ('$Email', '$CorkboardID', '$URL', '$Description');
```

Phase 2 Report CS 6400 – Fall 2018 Team 04

```
SET @CurrentPushpinID =  
(SELECT pushpinID FROM Pushpin  
ORDER BY pushpinID DESC LIMIT 1);
```

- o Repeat script for each tag that was inputted

```
INSERT INTO Tags(pushpinID, tag) VALUES (@CurrentPushpinID, '$Tag');
```

View Corkboard Stats

- When **User** clicks **View Corkboard Statistics** on **Home Screen**
 - o Query database for relevant information
 - o Present this information on all **Users** in a table format
 - Primary Sort by Most Public **CorkBoards**
 - Secondary Sort by Most Private **CorkBoards**

```
SELECT c.owner_email,  
(SELECT COUNT(*) FROM PublicCorkboard  
WHERE owner_email = c.owner_email) AS pub_cork_count,  
(SELECT COUNT(*) FROM PrivateCorkboard  
WHERE owner_email = c.owner_email) AS prv_cork_count,  
(SELECT COUNT(*) FROM PublicCorkboard  
NATURAL JOIN Pushpin  
WHERE owner_email = c.owner_email) AS pub_push_count,  
(SELECT COUNT(*) FROM PrivateCorkboard  
NATURAL JOIN Pushpin  
WHERE owner_email = c.owner_email) AS prv_push_count  
FROM Corkboard AS c  
GROUP BY owner_email  
ORDER BY pub_cork_count, prv_cork_count DESC;
```

- Display current **Users** row in red

View Popular Tags

- When **User** presses **View Popular Tags** on **Home Screen**
- Read from database for All **PushPin Tags**:

```
SELECT tag, COUNT(Pushpin.pushpinID) AS pushpin_count,  
COUNT(DISTINCT Pushpin.corkboardID) AS corkboard_count  
FROM Pushpin  
INNER JOIN Tags ON Pushpin.pushpinID = Tags.pushpinID  
INNER JOIN Corkboard ON Corkboard.corkboardID = Pushpin.corkboardID  
GROUP BY tag  
ORDER BY pushpin_count DESC;
```

- 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

Phase 2 Report CS 6400 – Fall 2018 Team 04

Follow

- **User** clicks on the **Follow** button from the **View Corkboard** or **View Pushpin** screen
- Assume \$Followee_email is already available from the **View Corkboard** or **View Pushpin** task
- Run the **Follow** task:
 - This will add the **User** who owns the **CorkBoard** or **PushPin** to the current **User's** follow list

```
INSERT INTO Follows(follower_email, followee_email)
VALUES ('$Email', '$Followee_email');
```

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

View Corkboard

- When **User** clicks the **title** of a Corkboard on the **Home screen**
- The **View Corkboard** task is run:
 - Assume we have the corkboards owner (\$Email) and ID (\$CorkboardID)
 - Read the following from database:
 - **PushPins**: Collection of pushpins associated with Corkboard
 - Web url for image

```
SELECT pushpinID, url FROM Pushpins
WHERE owner_email = '$Email' AND corkboardID = '$CorkboardID';
```

- **User**:
 - Owners name

```
SELECT fname, lname FROM `User` WHERE `User`.Email = '$Email';
```

- **CorkBoard**:
 - Title of **CorkBoard**
 - Date last updated (when last PushPin was added)
 - Category of **CorkBoard**
 - Number of **Users** watching the board

```
SELECT title, cork_date, category FROM Corkboard
NATURAL JOIN PrivateCorkboard
WHERE owner_email = '$Email' AND corkboardID = '$CorkboardID'
UNION
SELECT title, cork_date, category FROM Corkboard
NATURAL JOIN PublicCorkboard
WHERE owner_email = '$Email' AND corkboardID = '$CorkboardID';
```

- If not Owner of CorkBoard:
 - If following, grey out **Follow** button
- Get number of watchers

```
SELECT COUNT(*) FROM Watches
WHERE corkboardID = '$CorkboardID';
```

- 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

- **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

```
INSERT INTO Watches(watcher_email, corkboardID)
VALUES ('$Email', '$CorkboardID');
```

Search PushPins

- **User** enters *search text* (\$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 for Public Corkboards:
 - **PushPin** description
 - **PushPin** tags
 - Category of **CorkBoard** associated

```
SELECT DISTINCT description, category, CONCAT(fname, ' ', lname) AS
fullname
FROM `User`
INNER JOIN PublicCorkboard AS pub ON pub.owner_email = `User`.email
INNER JOIN Pushpin AS pin ON pin.corkboardID = pub.corkboardID
LEFT JOIN Tags ON Tags.pushpinID = pin.pushpinID
WHERE description LIKE '%$Search_text%' OR
tag LIKE '%$Search_text%' OR
category LIKE '%$Search_text%'
ORDER BY description DESC;
```

- If **User** clicks **PushPin description** run **View PushPin** task

Phase 2 Report CS 6400 – Fall 2018 Team 04

View Popular Sites

- When **User** clicks **View Popular Sites** on **Home Screen** form
- Read from database for all **PushPins**
 - Collect all domain website of *url* 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

```
SELECT DISTINCT SUBSTRING(  
    url,  
    LOCATE('/', url) + 2,  
    LOCATE('/', url, 8) - LOCATE('/', url) - 2  
) AS PopularSite,  
COUNT(*) AS CountPopularSite  
FROM Pushpin  
GROUP BY PopularSite  
ORDER BY CountPopularSite DESC;
```

View PushPin

- **User** clicks image hyperlink in **CorkBoard form** which runs the **View PushPin** task:
 - Assume PushpinID (\$PushpinID) of pushpin is managed by application
 - Assume Email (\$Email) is current user
 - Find **User** Owner of **PushPin**; Display **User** Owner First and Last Name

```
SELECT fname, lname  
FROM Pushpin  
INNER JOIN `User`  
ON Pushpin.owner_email = `User`.email  
WHERE pushpinID = '$PushpinID';
```

- If not Owner of **PushPin**, Read Follow Information:
 - If Following, grey out **Follow** button

```
SELECT *  
FROM Follows  
WHERE followee_email = '$Email'  
AND follower_email = (  
    SELECT owner_email  
    FROM Pushpin  
    WHERE pushpinID='$PushpinID'  
);
```

- Find and Display Title of **PushPins CorkBoard**;
 - If **User** clicks **Title**, display the associated **CorkBoard** view

```
SELECT title, corkboardID  
FROM Pushpin NATURAL JOIN PrivateCorkboard  
WHERE pushpinID='$PushpinID'  
UNION  
SELECT title, corkboardID  
FROM Pushpin NATURAL JOIN PublicCorkboard  
WHERE pushpinID='$PushpinID';
```

- Find the image *url* of **PushPin** and display the image

Phase 2 Report CS 6400 – Fall 2018 Team 04

- Find *url* of **PushPin** Image and Display the domain name of *url*
- Find and display the *Description* of **PushPin**
- Find and Display Date/Time of **PushPin**

```
SELECT url, SUBSTRING(  
    url,  
    LOCATE('/', url) + 2,  
    LOCATE('/', url, 8) - LOCATE('/', url) - 2  
) AS domain,  
description, pushpin_date  
FROM Pushpin  
WHERE pushpinID='$PushpinID';
```

- Find and display the *Tag* of **PushPin**, split by comma

```
SELECT tag  
FROM Pushpin  
LEFT OUTER JOIN Tags ON Pushpin.pushpinID = Tags.pushpinID  
WHERE pushpinID='$PushpinID';
```

- Look up current users likes:
 - If already liked:
 - Display “**Unlike!**” button
 - Else:
 - Display “**Like!**” button

```
SELECT *  
FROM Pushpin  
LEFT OUTER JOIN Likes  
ON Pushpin.pushpinID = Likes.pushpinID  
WHERE liker_email='$Email' AND pushpinID='$PushpinID';
```

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

```
SELECT CONCAT(fname, ' ', lname) AS fullname, comment, comment_date  
FROM Pushpin  
NATURAL JOIN `Comment`  
LEFT OUTER JOIN `User` ON Comment.commenter_email = `User`.email  
WHERE Pushpin.pushpinID = '$PushpinID'  
ORDER BY comment_date DESC;
```

Like/Unlike

- Run **Like/Unlike** Task on **Pushpin** form
 - Assume Email (\$Email) is current user
 - Assume PushpinID (\$PushpinID) of pushpin is managed by application
 - Display all **Users** first and last name who have liked the **Pushpin**

Phase 2 Report CS 6400 – Fall 2018 Team 04

```
SELECT CONCAT(fname, ' ', lname) AS fullname
FROM (Pushpin
LEFT OUTER JOIN Likes
ON Pushpin.pushpinID = Likes.pushpinID)
LEFT OUTER JOIN `User` ON liker_email = email
WHERE pushpinID='$PushpinID';
```

- Find if current User has liked the PushPin

```
SELECT *
FROM Likes
WHERE pushpinID = '$PushpinID' AND liker_email = '$Email';
```

- If User has unliked the PushPin,
 - If User click “Like!” button, write the User into database

```
INSERT INTO Likes (pushpinID, liker_email)
VALUES ('$PushpinID', '$Email');
```

- If User has liked the PushPin,
 - If User click “Unlike!” button, remove the User from database

```
DELETE FROM Likes
WHERE liker_email = '$Email' AND pushpinID='$PushpinID';
```

- Run View PushPin task

Post Comment

- Run Post Comment Task on Pushpin form
 - Assume PushpinID (\$PushpinID) of pushpin is managed by application
 - Assume Comment (\$Comment) is the input provided by User (\$Email)
 - User 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

```
INSERT INTO `Comment` (commenter_email, pushpinID, comment)
VALUES('$Email', '$PushpinID', '$Comment');
```

- Run View PushPin task