

Table Of Contents

Table Of Contents	0
Variable Representations for Abstract Code	1
Logging In	1
Abstract Code	1
Home Screen	1
Abstract Code	1
Password Protected	3
Abstract Code	3
Add CorkBoard	3
Abstract Code	3
View CorkBoard	4
Abstract Code	4
Add PushPin	6
Abstract Code	6
View PushPin	8
Abstract Code	8
Searching For PushPins	11
Abstract Code	11
Popular Tags	12
Abstract Code	12
Popular Sites	12
Abstract Code	12
CorkBoard Statistics	13
Abstract Code	13

Variable Representations for Abstract Code

1. **Bold Underline: Form** Example: **Main Menu** form
2. **Bold Italics: Buttons** Example: *Save* button
3. **Bold: Task** Example: **Login** task
4. *Italics*: Form Input Fields: Example: *username, model, cost*, etc.

Logging In

Abstract Code

1. User enters email (*\$Email*) and pin (*\$Pin*) into the input fields
 - a. If pin not 4 digits:
 - i. An error message is displayed and user is returned to the login screen
2. If validation is successful for both email and pin, then
 - a. When **Login** is clicked:

```
SELECT user_id, pin FROM CorkBoardItUser WHERE  
CorkBoardItUser.email=$Email
```

- i. If *CorkBoardItUser* record is found but *CorkBoardItUser.password* != *\$Pin*:
 1. An error message is displayed and user is returned to **Logging In**
- Else:
 2. Store login *CorkBoardItUser.user_id* as session variable '*\$UserID*'
 3. Go to **Home Page**

Home Screen

Abstract Code

1. Show '*Popular Tags*', '*Add Corkboard*', '*Pushpin Search*' on the screen
2. Show **Look Up CorkBoard Owned** task
3. Show **Look Up Recent CorkBoard Updates** task
4. Upon:
 - a. Click *Popular Tags* button - Jump to the **Popular Tags** form
 - b. Click *Add CorkBoard* button - Jump to the **Add CorkBoard** form
 - c. Click *PushPin Search* button - Jump to the **Searching for PushPins** form

Look Up CorkBoard Owned

Abstract Code

[Table of Contents](#)

1. Find the current CorkBoards owned and PushPins each CorkBoard has using [CorkBoardItUser.user_id](#), [CorkBoard.corkBoard_id](#), and [PushPin.pushpin_id](#);
2. Display [CorkBoard.title](#) and number of pushpins each corkboard has;
3. Display [CorkBoard.visibility](#); If visibility is False show '(private)' next to title
4. If no result is returned, display "You have no CorkBoards"

```
SELECT board.title, board.visibility, pins_count.pins_count
FROM (SELECT owned_corkboard.corkboard_id, COUNT(owned_corkboard.corkboard_id) as
pins_count
FROM (SELECT board.title, board.visibility, board.corkboard_id
      FROM CorkBoard board
      WHERE board.fk_user_id = $UserID) owned_corkboard
LEFT OUTER JOIN PushPin pin
ON pin.fk_corkboard_id = owned_corkboard.corkboard_id
GROUP BY owned_corkboard.corkboard_id) pins_count
INNER JOIN CorkBoard board
ON board.corkboard_id = pins_count.corkboard_id
ORDER BY board.title ASC;
```

Look Up Recent CorkBoard Updates

Abstract Code

1. Find
 - a. All [CorkBoards](#) watched by \$UserID through [Watch.fk_public_corkboard_id](#)
 - b. All [CorkBoardItUsers](#) followed by \$UserID through [Follow.fk_user_follower_id](#)
 - c. All [CorkBoards](#) owned by \$UserID
2. Display [CorkBoard.title](#), [CorkBoard.fk_user_id](#), [CorkBoard.date_time](#), [CorkBoard.privacy](#)
3. Limit maximum size to 4
4. Sort by [CorkBoard.date_time](#)
5. Upon clicking on [CorkBoard.title](#) - Jump to **Viewing CorkBoards** form
6. If no result is returned, display "No Updates"

```
SELECT fcorkboard.title, fcorkboard.date_time, fcorkboard.visibility
FROM
  (SELECT Follow.fk_user_follower_id
   FROM Follow
   WHERE Follow.fk_user_follower_id = $UserID) fid
INNER JOIN CorkBoard fcorkboard
ON fcorkboard.fk_user_id = fid.fk_user_follower_id
```

```

UNION
SELECT wcorkboard.title, wcorkboard.date_time, wcorkboard.visibility
  FROM
  (SELECT w.fk_public_corkboard_id
    FROM Watch w
   WHERE w.fk_user_id = $UserID) wid
INNER JOIN CorkBoard wcorkboard
ON wcorkboard.corkboard_id = wid.fk_public_corkboard_id
UNION
SELECT CorkBoard.title, CorkBoard.date_time, CorkBoard.visibility
  FROM CorkBoard
 WHERE CorkBoard.fk_user_id = $UserID
ORDER BY date_time DESC
LIMIT 4;

```

Password Protected

Abstract Code

1. User enters password into the input field for **Corkboard** of \$CorkboardID
2. When **OK** is clicked:
 - a. If \$Password is not the same as PrivateCorkBoard.password:
 - i. Go to **Home Screen**
 - b. Else:

```

SELECT password
FROM PrivateCorkBoard AS private
WHERE private.password = $Password AND private.fk_corkboard_id = $CorkBoardID;

```

- i. Go to **View CorkBoard**
3. If **Cancel** is clicked:
 - a. Go to **Home Screen**

Add CorkBoard

Abstract Code

1. User of \$UserID and \$Email enters **Add CorkBoard** form from clicking on **Add CorkBoard** button
2. If **Add** button is selected and \$Title, \$Category, \$Visibility with \$Password is filled:
 - a. If \$Title input is greater than 50 characters:

[Table of Contents](#)

- i. Disable the *Add* button
- b. Else:
 - i. Insert **CorkBoard**, where **\$CorkBoardID** represents ID of **CorkBoard** :
 - 1. Set **CorkBoard**.Title to **\$Title** from title textbox
 - 2. Set **CorkBoard**.Category to **\$Category** from category selection
 - a. If **Category Dropdown** is clicked:
 - i. Display categories in sorted manner

```
INSERT INTO CorkBoard (corkboard_id, fk_user_id, email, date_time, title, category, visibility)
VALUES ( $CorkBoardID, $UserID, $Email, $DateTime, $Category, $Visibility);
```

- 3. If public is selected:
 - a. Set **CorkBoard**.Visibility to **True** (**\$Visibility** = True)

```
INSERT INTO PublicCorkBoard (public_corkboard_id, fk_corkboard_id)
VALUES ($PublicCorkBoardID, $CorkBoardID);
```

- 4. Else:
 - a. Set **CorkBoard**.Visibility to **False** (**\$Visibility** = False)
 - b. Set **CorkBoard**.Password to **\$Password** from password textbox

```
INSERT INTO PrivateCorkBoard (private_corkboard_id, fk_corkboard_id, password)
VALUES ($PrivateCorkBoardID, $CorkBoardID, $Password);
```

- ii. Go to **View CorkBoard** screen of the new **CorkBoard** of **\$CorkBoardID**

View CorkBoard

Abstract Code

1. **CorkboardItUser** 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 **CorkBoardItUser** **\$UserID**'s **CorkBoardItUser.first_name** and **CorkBoardItUser.last_name**
 - a. Display First and Last Name as owner

[Table of Contents](#)

```
SELECT first_name, last_name
FROM CorkBoard AS cb
    INNER JOIN CorkBoardItUser AS u
        ON u.user_id = cb.fk_user_id
WHERE cb.corkboard_id = $CorkBoardID;
```

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.date_time`; Display date & time
 - d. Find number of watchers by counting number of `Watch.user_id`; Display count of watchers

```
SELECT title, category, date_time, cb.email, COUNT(w.fk_user_id)
FROM CorkBoard AS cb
    LEFT OUTER JOIN PublicCorkBoard AS public
        ON cb.corkboard_id = public.fk_corkboard_id
    LEFT OUTER JOIN Watch AS w
        ON public.public_corkboard_id = w.fk_public_corkboard_id
WHERE cb.corkboard_id = $CorkBoardID
GROUP BY title, cb.category, cb.date_time, cb.email;
```

4. Query `PushPins` with `PushPin.corkboard_id` equal to `$CorkBoardID`
 - a. Find `PushPins` related to `CorkBoard`
 - b. Display `PushPin.url` as image

```
SELECT url
FROM Pushpin AS p
    INNER JOIN CorkBoard AS cb
        ON p.fk_corkboard_id = cb.corkboard_id
WHERE p.fk_corkboard_id = $CorkBoardID;
```

5. If **PushPin** image is clicked:
 - a. Go to **View PushPin** screen
6. If `CorkboardItUser.user_id` is not the same as `CorkBoard.user_id`:

```
SELECT u.user_id
FROM CorkBoardItUser AS u
    INNER JOIN Corkboard AS cb
        ON u.user_id = cb.fk_user_id
    INNER JOIN PublicCorkBoard AS public
```

```
ON cb.corkboard_id = public.fk_corkboard_id
WHERE cb.corkboard_id = $CorkBoardID;
```

- a. Enable **Follow** button
- b. If button is clicked:
 - i. Insert \$UserID of current user and Corkboard.user_ID of Corkboard with \$CorkboardID into Follow

```
INSERT INTO Follow (fk_user_follower_id, fk_user_followee_id)
VALUES ( $UserID,
        ( SELECT user_id
          FROM CorkBoardItUser AS u
            INNER JOIN CorkBoard AS cb
                  ON u.user_id = cb.fk_user_id
          WHERE cb.corkboard_id = $CorkBoardID ));
```

7. Else:
 - a. **Follow** button is not shown
8. If CorkboardItUser.user_id is equal to CorkBoard.user_id
 - a. **Watch** button is not shown
9. If **Watch** button is clicked:
 - a. Insert \$UserID and PublicCorkBoard.public_corkboard_id where PublicCorkBoard.corkboard_id is equal to \$CorkboardID

```
INSERT INTO Watch(fk_user_id, fk_public_corkboard_id)
VALUES ($UserID,
        ( SELECT public_corkboard_id
          FROM PublicCorkBoard AS public
          WHERE public.fk_corkboard_id = $CorkboardID ));
```

10. If **Add PushPin** button is clicked and CorkboardItUser.user_id is equal to CorkBoard.user_id:
 - a. Go to Add PushPin to CorkBoards form

Add PushPin to CorkBoard

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

[Table of Contents](#)

```
SELECT title
FROM CorkBoard AS cb
WHERE cb.corkboard_id = $CorkBoardID;
```

3. User clicks **Add PushPin** button
4. If **Add** button is selected and valid url, description, and tags (comma separated) are filled:
 - a. If description is greater than 200 characters or tag(s) are greater than 20 characters or tag(s) contain comma does not contain commas each:
 - i. Disable the **Add** button
 - b. Else If url does not contain '.jpg', '.png', '.gif' extension:
 - i. Disable the **Add** button
 - c. Else:
 - i. Insert **PushPin**:
 1. Set **PushPin**.URL to \$URL from url textbox
 2. Set **PushPin**.Description with \$Description from description textbox
 3. Set **PushPin**.corkboard_id with \$CorkBoardID of current **CorkBoard**.corkboard_id
 4. Set **PushPin**.pushpin_id to \$PushPinID
 5. Set **PushPin**.date_time to \$DateTime
 6. Separate tags by comma; For each tag and insert **Tag**
 - a. Set **Tag**.tag to \$Tag from tag box

```
INSERT INTO PushPin (pushpin_id, fk_user_id, fk_corkboard_id, date_time, url, description)
VALUES ($PushPinID, $UserID, $CorkBoardID, $DateTime, $URL, $Description);
```

```
INSERT INTO Tag (fk_pushpin_id, tag)
VALUES ($PushPinID, $Tag);
```

- ii. Update **CorkBoard** of \$CorkBoardID:
 1. Set **CorkBoard**.date_time with **PushPin**.date_time

```
UPDATE CorkBoard
SET date_time = dt.date_time
FROM (SELECT date_time
      FROM PushPin AS p
      WHERE p.pushpin_id = $PushPinID) AS dt
WHERE corkboard_id = $CorkBoardID;
```

- iii. User of \$UserID is returned to View CorkBoard Screen

View PushPin

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 CorkBoardItUser.first_name and CorkBoardItUser.last_name through current CorkBoard.corkboard_id
 - a. Find CorkBoardItUser.user_id from CorkBoard.fk_user_id of \$CorkBoardID where image was clicked on
 - b. Display CorkBoardItUser.first_name and CorkBoardItUser.last_name

```
SELECT first_name, last_name
FROM CorkBoard AS cb
     INNER JOIN CorkBoardItUser AS u
           ON u.user_id = cb.fk_user_id
WHERE cb.corkboard_id = $CorkBoardID ;
```

3. Query PushPin details
 - a. Find \$DateTime from PushPin.date_time for the image that was clicked on
 - b. Display PushPin.date_time

```
SELECT pp.date_time
FROM PushPin AS pp
     INNER JOIN CorkBoard AS cb
           ON pp.fk_corkboard_id = cb.corkboard_id
WHERE cb.corkboard_id = $CorkBoardID;
```

- c. Find the CorkBoard.title the PushPin belongs to, where PushPin.corkboard_id matches CorkBoard.corkboard_id of \$CorkBoardID
 - i. **Hyperlink** back to the CorkBoard of \$CorkBoardID

```
SELECT title
FROM PushPin AS pp
     INNER JOIN CorkBoard AS cb
           ON cb.corkboard_id = pp.fk_corkboard_id
WHERE cb.corkboard_id = $CorkBoardID;
```

- d. Find image URL from PushPin.URL

```
SELECT url
FROM PushPin as pp
WHERE pp.pushpin_id = $PushPinID;
```

- 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 of \$PushPinID
- i. Display PushPin.Description of \$PushPinID

```
SELECT description
FROM PushPin as pp
WHERE pp.pushpin_id = $PushPinID;
```

- j. Find tags from Tag.tag
- k. Display all Tag.tag from \$PushPinID
 - i. Sort tags alphabetically

```
SELECT tag
FROM Tag AS t
      INNER JOIN PushPin AS pp
            ON pp.pushpin_id = t.fk_pushpin_id
WHERE t.fk_pushpin_id = $PushPinID;
```

- l. Find CorkBoardItUser.user_id who liked the PushPin from Liked.user_id that matches PushPin.user_id
- m. Display CorkBoardItUser.first_name and CorkBoardItUser.last_name

```
SELECT first_name, last_name
FROM CorkBoardItUser AS u
      INNER JOIN Liked AS ld
            ON ld.fk_user_id = u.user_id
      INNER JOIN PushPin as pp
            ON pp.pushpin_id = ld.fk_pushpin_id
WHERE pp.pushpin_id = $PushPinID;
```

- n. Find comments on PushPin from Comment.date_time
 - i. Display Comment.user_id and Comment.text
 - ii. Sort comments in descending order

```
SELECT c.fk_user_id, c.text, c.date_time
FROM Comment AS c
    INNER JOIN CorkBoardItUser AS u
        ON u.user_id = c.fk_user_id
    INNER JOIN PushPin as pp
        ON pp.pushpin_id = c.fk_pushpin_id
WHERE c.fk_pushpin_id = $PushPinID
ORDER BY c.date_time DESC;
```

4. User of \$UserID clicks **Follow** and check if the CorkBoardItUser.user_id is equal to \$UserID
 - a. If the email is not the same then we insert CorkBoardItUser.email to Follow.fk_user_follower_id

```
INSERT INTO Follow (fk_user_follower_id, fk_user_followee_id)
VALUES ((SELECT user_id FROM CorkBoardItUser AS u WHERE u.user_id =
$UserID), (SELECT fk_user_id FROM CorkBoard WHERE corkboard.corkboard_id
= $CorkBoardID AND corkboard.fk_user_id != $UserID));
```

5. User of \$UserID clicks **Like!/Unlike** and CorkBoardItUser.user_id is not equal to \$UserID
 - a. If User clicks **Like!**
 - i. Insert \$UserID to Liked.fk_user_id
 - ii. Change **Like!** Button to **Unlike**

```
INSERT INTO Liked (fk_user_id, fk_pushpin_id)
VALUES ((SELECT user_id FROM CorkBoardItUser AS u WHERE u.user_id =
$UserID), (SELECT fk_user_id FROM CorkBoard WHERE corkboard.corkboard_id
= $CorkBoardID AND corkboard.fk_user_id != $UserID));
```

- b. If User of \$UserID clicks **Unlike**
 - i. Remove \$UserID from Liked.fk_user_id and Liked.fk_pushpin_id
 - ii. Change **Unlike** button to **Like!**

```
DELETE FROM Liked
WHERE Liked.fk_user_id = $UserID AND Liked.fk_pushpin_id = $PushPinID;
```

6. Else if User of \$UserID equals CorkBoardItUser.user_id
 - a. Disable **Like!** button
7. If User of \$UserID clicks **Post Comment**
 - a. Insert the text entered, \$Text, in the Enter Comment box as Comment.Text
 - b. Insert \$UserID into Comment.user_id

- c. Insert the current Date and Time, `$DateTime`, as `Comment.date_time`

```
INSERT INTO Comment (comment_id, date_time, text, fk_user_id, fk_pushpin_id)
VALUES ($comment_id, $date_time, $text, (SELECT user_id FROM
CorkBoardItUser AS u WHERE u.user_id = $UserID), (SELECT fk_user_id FROM
PushPin WHERE pushpin.pushpin_id = $PushPinID));
```

Searching For PushPins

Abstract Code

1. User of `$UserID` enters **PushPin Search Results** screen by click *PushPin Search* on **HomeScreen**, if some text is entered in `$search_textbox`.
2. Find all `PushPin.description` containing `$search_textbox.text`
3. Find all `PushPin.tag` containing `$search_textbox.text`
4. Find all `PushPin.category` containing `$search_textbox.text`
5. Once found:
 - a. Find all `PushPin` descriptions, and all associated `CorkBoardItUser.first_name` and `CorkBoardItUser.last_name` and `CorkBoard.title`
6. Sort `PushPins` alphabetically using `PushPin.description`
7. For all found `PushPins`
 - a. Display `PushPin.description` under PushPin Description column
 - b. Hyperlink `PushPin.description` to **View PushPin** Screen
 - c. Display `PushPin` associated `CorkBoard.title` under CorkBoard column
 - d. Display `PushPin` associated `CorkBoardItUser.first_name` and `CorkBoard.last_name` under Owner column

```
SELECT DISTINCT ON (search.description) search.description, search.title, search.first_name,
search.last_name
FROM(
SELECT PushPin.description, CorkBoard.title, CorkBoard.category, CorkBoardItUser.first_name,
CorkBoardItUser.last_name, Tag.tag
FROM CorkBoard INNER JOIN CorkBoardItUser ON CorkBoard.fk_user_id =
CorkBoardItUser.user_id INNER JOIN PushPin ON CorkBoard.corkboard_id =
PushPin.fk_corkboard_id FULL OUTER JOIN Tag ON Tag.fk_pushpin_id = PushPin.pushpin_id
NATURAL JOIN PublicCorkBoard
WHERE
description LIKE '%$search_textbox.text%' OR category LIKE '%$search_textbox.text%' OR tag
LIKE '%$search_textbox.text%'
)search
ORDER BY description ASC;
```

Popular Tags

Abstract Code

1. User enters **Popular Tags** from clicking *Popular Tags* on Main Menu
2. Query all PushPins
 - a. Find count of all **Tag**.tags
 - b. Find count of all distinct **CorkBoard**.corkboard_id associated with each tag
 - c. Display each **Tag**.tag
 - d. Display count for each **Tag**.tag
 - e. Display count for unique **CorkBoard**.corkboard_id containing that tag

```
SELECT Tag.tag, COUNT(Tag.Tag) AS pushpins, COUNT(DISTINCT
Corkboard.corkboard_id) as unique_cb
FROM Tag INNER JOIN PushPin ON Tag.fk_pushpin_id = PushPin.pushpin_id
INNER JOIN CorkBoard ON CorkBoard.corkboard_id = PushPin.fk_corkboard_id
GROUP BY tag
ORDER BY pushpins DESC
LIMIT 5;
```

3. If specific tag is clicked
 - a. Go to **Searching For PushPins** for that specific tag

Popular Sites

Abstract Code

4. User enters **Popular Sites** from clicking *Popular Sites* on **Home Screen**

```
SELECT PushPin.url as site, COUNT(PushPin.url) as pushpins
FROM PushPin
GROUP BY site
ORDER BY pushpins DESC
LIMIT 4;
```

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

Abstract Code

1. User of [\\$UserID](#) enters **CorkBoard Statistics** by clicking *CorkBoard Statistics* from **Home Screen**
2. Query information on all [CorkBoards](#) for all [CorkBoardItUser](#):
 - a. For each [CorkBoardItUser](#):
 - i. Get count of [PublicCorkBoards](#) associated with [CorkBoardItUser.user_id](#)
 - ii. Get count of [PushPins](#) associated with [CorkBoardItUser's PublicCorkBoards](#)
 - iii. Get count of [PrivateCorkBoards](#) associated with [CorkBoardItUser.user_id](#)
 - iv. Get count of [PushPins](#) associated with [CorkBoardItUser's PrivateCorkBoards](#)
 - v. Display count of [PublicCorkBoards](#) under Public CorkBoards column
 - vi. Display count of [PushPins](#) belonging to [PublicCorkBoards](#) under Public PushPins column
 - vii. Display count of [PrivateCorkBoards](#) under Private CorkBoards column
 - viii. Display count of [PushPins](#) belonging to [PrivateCorkBoards](#) under Private PushPins column

```
SELECT CorkBoardItUser.first_name, CorkBoardItUser.last_name, COUNT(DISTINCT
PublicCorkBoard.fk_corkboard_id) AS public_cb, COUNT(DISTINCT pub_pin.pushpin_id) AS
pub_pushpins, COUNT(DISTINCT PrivateCorkBoard.fk_corkboard_id) AS private_cb,
COUNT(DISTINCT pr_pin.pushpin_id) AS private_pushpins
FROM CorkBoardItUser FULL OUTER JOIN CorkBoard ON CorkBoardItUser.user_id =
CorkBoard.fk_user_id
FULL OUTER JOIN PublicCorkBoard ON CorkBoard.corkboard_id =
PublicCorkBoard.fk_corkboard_id
LEFT OUTER JOIN Pushpin pub_pin ON PublicCorkBoard.fk_corkboard_id =
pub_pin.fk_corkboard_id
FULL OUTER JOIN PrivateCorkBoard ON CorkBoard.corkboard_id =
PrivateCorkBoard.fk_corkboard_id
LEFT OUTER JOIN Pushpin pr_pin ON PrivateCorkBoard.fk_corkboard_id = pr_pin.fk_corkboard_id
GROUP BY first_name, last_name;
```