

Login

Abstract Code

- User enters *email* (\$Email), *password* (\$PIN) input fields.
- If data validation is successful for both *email* and *password* input fields, then:
 - When **Login** button is clicked:

```
SELECT pin FROM RegulerUser WHERE user_id = '$User_ID';
```

- If User record is found but RegulerUser.pin!= '\$PIN':
 - Go back to **Login** form, with error message.
- Else
 - Store login information as session variable '\$UserID'.
 - Go to **Home Screen**
- Else *email* and *password* input fields are invalid, display **Login** form, with error message.

Home Screen

- Display user's first and last name, **Recent CorkBoard Updates** and **My CorkBoards** forms.
- Find the current RegulerUser using the User.email; Display user's first and last name;

```
SELECT first_Name, last_name FROM RegulerUser WHERE  
userid='$UserID';
```

Recent CorkBoard Updates

Abstract Code

- Display the title, owner, visibility, last update time of cork boards that user watched or owner that followed by user.

```
SELECT C.last_dtime, C.title, C.visibility, U.first_name, U.last_name  
FROM  
Corkboard as C  
INNER JOIN User as U on C.user_id = U.user_id  
INNER JOIN Watch as W on C.user_id= W.user_id and C.board_id= W.board_id  
INNER JOIN Follow as F on C.user_id= F.user_id  
INNER JOIN Follow as F on C.user_id= F.followed_user_id  
WHERE user_id='$User_ID'  
ORDER BY C.last_dtime DESC
```

LIMIT 4

- Click title of the corkboard:
 - Save CorkBoard.title as session variable '\$Title', save CorkBoard.email as '\$CorkBoard_UserID'.
 - If public cork board: Jump to **View Cork Board** task.
 - Else: Jump to **Enter CorkBoard's Password** task.

My CorkBoards

Abstract Code

- Show "Add CorkBoard" tab.
- Click **Add CorkBoard** button - Jump to **Add CorkBoard** task.
- Find current user's cork boards using the user Email; Display CorkBoard Title, visualibility, number of pins.

```
Select C.board_id, C.title, C.visualibility, count(pin)
FROM CorkBoard as C
INNER JOIN RegulerUser R ON CorkBoard.user_id = R.user_id
WHERE R.user_id = '$UserID';
Group by board_id, title, visualibility
```

- Click title of the cork board:
 - Save cork board title as session variable '\$Title', save '\$UserID' as '\$CorkBoard_UserID'.
 - If public cork board: Jump to **View Cork Board** task.
 - Else: Jump to **Enter CorkBoard's Password** task.

Enter CorkBoard's Password

- User enters *password* (\$Password) input fields.
- If data validation is successful for *password* input fields, then:
 - When **Cancel** button is clicked: Go back to **Enter CorkBoard's Password** form.
 - When **OK** button is clicked:

```
SELECT password FROM Corkboard WHERE user_id= '$UserID'
AND board_id= '$Board_ID';
```

- If CorkBoard.password = '\$Password':
 - Go to **View CorkBoard** task.
- Else:
 - Go back to **Enter CorkBoard's Password** form, with error message.

Add CorkBoard

- Populate **Category** dropdowns.

```
SELECT cat_name FROM Categorize
```

- User enters Title as variable \$Title into input field.
- User choose visibility as variable \$Visibility of the adding corkboard.
 - If the adding corkboard is Private, then enter password as variable \$Password into input field.
- Click **Add** button, add data to the **corkboard** form, jump to the View Corkboard task.

```
INSERT INTO Corkboard (board_id, user_id, Title, categoryname, visibility, Password) VALUES($Board_ID, $User_ID, $Title, $CategoryName, $Visibility, $Password);
```

View Cork Board

- Run the View Corkboard task, query information about the corkboard; Display the owner, title, and last updated date/time, Category, of the **Corkboard**.

```
SELECT R.First_Name, R.last_name, C.title, C.Last_dtime, CA.cat_name
FROM
Corkboard AS C
INNER JOIN RegularUser AS R ON U.user_id = C.user_id
INNER JOIN CATEGORIZE AS CA ON C.user_id = CA.user_id AND
C.board_id= CA.board_id
WHERE C.board_id = '$Board_ID';
```

- Query URL of all the pushpin in this **corkboard**.

```
SELECT URL_image FROM Pushpin  
INNER JOIN Corkboard ON Pushpin .board_id = Corkboard.board_id  
WHERE Corkboard .board_id = '$Board_ID';
```

- Query number of the watcher of this corkboard, display the total number of watcher to this corkboard.

```
SELECT COUNT(user_id)  
FROM Watch  
WHERE Watch.board_id='$Board_ID';
```

- Click **Follow** button, follow the user who owns the Corkboard.
- Click **Add Pushpin** button, jump to the **Add Pushpin** task.
- Click image, jump to the **View Pushpin** task.
- If the corkboard is owned by other user.
 - If the corkboard is public.
 - Click the **watch** button to watch the corkboard.

Add Pushpin

- User enters URL, Description and Tags into input fields.
 - If enter Tags, each tag should be separate by commas.
- When Click Add button
 - If input URL is valid, add data to the **Pushpin** form.

```
INSERT INTO PUSHPIN (pushpin_id, board_id, URL_image, Description,  
Pushpin_dtime)  
VALUES($Pushpin_ID, $Board_ID, $URL_image, $Description,  
$Pushpin_dtime);  
INSERT INTO PPIN_TAGS(pushpin_id, Tags) VALUES ($Pushpin_ID,  
$Tags);
```

- Jump to **View Corkboard** task.

View Pushpin

Abstract Code

- User click the hyperlink from View Corkboard.
- Run the View Pushpin task, query for information about the owner of the Corkboard which the pushpin pinned, the title of the Corkboard, data and time the first pinned.
- Query for information about the image and the tags.
- Query information about the user who comment this pushpin and the content of the comment.
- When click *Follow* button, update the Follow relation form.
- When click Hyperlink of the Corkboard title, jump to the **View Corkboard** task.
- When click *Like* button, update the user like relation form.
- When *Like* button has been clicked. There's an *UnLike* button. Click the *UnLike* button and update the user like relation form.
- When click *Post Comment* button, update the user comment pushpin form.

```

/*View Pushpin- Pushpin part*/
SELECT R.first_name, R.Last_Name, P.Pushpin_dtime, P.URL_IMAGE,
P.Description, C.title
FROM
Pushpin AS P
INNER JOIN Corkboard AS C ON P.board_id=C.board_id
INNER JOIN RegularUser AS R on C.user_id= R.user_id
WHERE pushpin_id='$Pushpin_ID';

```

```

/*View Pushpin- Tag*/
SELECT tag
FROM
Pushpin AS P
INNER JOIN Pushpin_Tag AS PT ON P.pushpin_id = PT.pushpin_id
WHERE pushpin_id='$Pushpin_ID'
ORDER BY tag ASC;

```

```

/*View Pushpin- Like*/
SELECT R.first_name, R.last_name
FROM Pushpin AS P
INNER JOIN User_Like AS UL on P.pushpin_id= UL.pushpin_id
INNER JOIN RegularUser AS R ON UL.user_id= R.user_id
WHERE pushpin_id='$Pushpin_ID' ;

```

```

/*View Pushpin- Comment*/
SELECT R.first_name, R.last_name, CO.Content
FROM
Comment AS CO
INNER JOIN

```

```
RegulerUser AS R ON CO.user_id=R.user_id  
WHERE Comment_ID='$Comment_ID'  
ORDER BY Comment_dtime DESC;
```

Watcher follow owner

Abstract Code

- When click *Follow* button, update the Follow relation form.

```
INSERT INTO Watch (user_id, board_id)  
VALUES($User_ID, $Board_ID);
```

Watcher like/ unlike owner's pushpins

Abstract Code

- When click *Like* button, update the user like relation form.
- When *Like* button has been clicked. There's an *UnLike* button. Click the *UnLike* button and update the user like relation form.

```
/*Like: */  
INSERT INTO User_Like (user_id, pushpin_id)  
VALUES($User_ID, $Pushpin_ID);  
/*Unlike: */  
Delete FROM User_Like  
WHERE user_id= $User_ID AND pushpin_id= $Pushpin_ID;
```

User comment pushpins

Abstract Code

- When click *Post Comment* button, update the user comment pushpin form.

```
INSERT INTO Comment (user_id, Comment_ID, pushpin_id,  
Comment_dtime, Content)  
VALUES('$User_ID', '$Comment_ID', '$Pushpin_ID', '$Comment_dtime',  
'$Content');
```

Search for Pushpin

Abstract Code

- Query the information of Pushpin Description, Category and owner of the Corkboard which the pushpin pinned.
- User Search '\$Search';
- Sorted the results alphabetically by Pushpin description.
- Click the hyperlink of the pushpin description, jump to **View Pushpin** task.

```
SELECT R.first_name, R.last_name, C.Title, P.Description  
FROM  
Pushpin AS P  
INNER JOIN Corkboard AS C ON P.board_id=C.board_id  
INNER JOIN RegularUser AS R on P.user_id= R.User_ID  
WHERE P.Description like '%$Search%';
```

View Popular Tags

Abstract Code

- Query the information of Tag, the number of Pushpins with this tag and the corresponding unique corkboards number.
- sorted the results alphabetically by number of Pushpins with the highest appearing first.
- Click the hyperlink of the tag, jump to **Search for Pushpin** task.

```
SELECT PT.tag, COUNT(PT.pushpin_id), COUNT(distinct P.board_id)
FROM
Pushpin AS P
INNER JOIN Pushpin_tag AS PT ON P.pushpin_id=PT.pushpin_id
GROUP BY tag
Limit 5;
```

View Popular Sites

```
Select Subquery.PouplarSite, count(Subquery.pushpin_id) AS
PushpinCount
FROM
(SELECT
pushpin_id,SUBSTRING_INDEX((SUBSTRING_INDEX(URL_image, "/",
3)), "/", -1) AS PouplarSite
FROM
Pushpin
) AS Subquery
GROUP BY Subquery.PopularSite
ORDER BY Subquery.PushpinCount;
```

Abstract Code

- Query the information of Sites that the most pushpin links.
- Validate the name of the sites and display by number of Pushpins with the highest appearing first.

View Corkboard Statistics

```
SELECT first_name, last_name, PublicCorkboardcount,
Publicpushpincount, PrivateCorkboardcount, Privatepushpincount
```



```

FROM
(
(SELECT R.user_id, R.first_name, R.last_name, count(board_id) AS
PublicCorkboardcount, COUNT(pushpin_id) AS Publicpushpincount
FROM
Corkboard AS C
INNER JOIN Pushpin AS P ON C.board_id= P.board_id
INNER JOIN RegularUser AS R on C.user_id= R.user_id
WHERE visibility='Public'
GROUP BY user_id
) AS PUB
INNER JOIN
(SELECT R.UserID, R.first_name, R.last_name, count(board_id) AS
PrivateCorkboardcount, COUNT(pushpin_id) AS Privatepushpincount
FROM
Corkboard AS C
INNER JOIN Pushpin AS P ON C.board_id= P.board_id
INNER JOIN RegularUser AS R on C.user_id= R.user_id
WHERE visibility='Private'
GROUP BY user_id
) AS PRI
ON PUB.user_id=PRI.user_id
) AS SUB
ORDER BY PublicCorkboardcount,PrivateCorkboardcount;

```

Abstract Code

- Populate the query result on screen.
- Print the current user's row to red.