Log In

Abstract Code

- User enters email ($email) and *pin ($pin)*
- User clicks <**Login**> button:
    - if form is validated, then:

```
SELECT pin FROM `User` WHERE email=’$email
```

        - if *email* does not have a known record, login page is redisplayed with an error message
        - else if *email* is found, and *pin* does not match, then login page is redisplayed with error message
        - Else create a session and store record of the user in $UserName and proceed to Home Screen
    - Else show an error message indicating the form was not validated

---

View Home Page

Abstract Code

- Find the current USER using the User.*email*

```
SELECT name FROM `User` WHERE User.email=’$email’
```

- Find the Corkboards followed by the USER

```
SELECT title, visibility, category type FROM Corkboard WHERE Corkboard.owner_email =’$email’
```

- Find users followed by USER

```
SELECT email FROM Follows WHERE Follows.follower_email=’$email’
```

- Display the USERs username
- Find each update to boards that are owned or watched by USER, or owned by a user followed by USER

```
SELECT title, date_time FROM Corkboard INNER JOIN PushPin ON Corkboard.owner_email =
PushPin.owner_email ORDER BY date_time DESC
```

    - Sort updates and display the 4 newest updates
    - If no updates
        - Display message “No Updates”
- Display all Corkboards Sorted alphabetically that are owned by USER

```
SELECT  title FROM Corkboard  INNER JOIN Private_Corkboard ON Corkboard.owner_email  =
Private_Corkboard.owner_email WHERE Corkboard.owner_email =’$email’
```

- While no buttons are pushed, do nothing

- When a button is push, then do:
  - if POPULAR_TAGS, **View Popular Tags**
  - if ADD_CORKBOARD, **Add Corkboard**
  - if PUSHPIN_SEARCH, **Search PushPins**
  - if CORKBOARD_UPDATE, Write to Corkboard, **View Corkboard**
  - if MY_CORKBOARD, **View Corkboard**
  - if CORKBOARD STATS, **View Stats**
  - if POPULAR_SITES, **View Popular Sites**

Add Corkboard

Abstract Code

- Display window with input forms and a dropdown containing Corkboard Categories alphabetically
- if USER selects Private
  - USER must enter password ($Password)

```
INSERT INTO Private_Corkboard ( title, owner_email, password )
VALUES ( $Title, $Email, $Pin
```

-
- While no buttons are pushed, do nothing
- If a button is pushed , then do:
  - if ADD, Write to Corkboard, **View Corkboard**

View Corkboard

Abstract Code

- Find the OWNER, CATEGORY, and IMAGES of the CORKBOARD ($Title)

```
SELECT owner_email, category_type, url FROM Corkboard c INNER JOIN Private_Corkboard pc
ON c.owner_email = pc.owner_email INNER JOIN PushPin pp ON c.owner_email = pp.owner_email
WHERE c.title = $Title
```

- Display OWNER, CATEGORY, and IMAGES
- If OWNER is current USER, Enable ‘Add PushPin’ button
- If current USER is not OWNER, display ‘follow’ button
- If current USER is not OWNER and CORKBOARD is public, display ‘watch’ button
- Display number of watchers

```
SELECT COUNT(*) FROM (SELECT email FROM Corkboard c INNER JOIN Watch w ON c.title =
w.title WHERE c.title = $Title)
```

- While no buttons are pushed, do nothing
- If a button is pushed, then do:
  - if FOLLOW, Write to USER, **View Corkboard**

  - if ADD_PUSHPIN, Write to Corkboard, **Add Pushpin to Corkboard**
  - if WATCH, Write to USER, **View Corkboard**

```
INSERT INTO Watch ( email, title, owner_email)
VALUES ( $Email, $Title, $Owner_Email)
```

  - if <image>, **View PushPin**

Add PushPin to CorkBoard

Abstract Code

- Display window with input forms
- Display CATEGORY
- While no buttons are pushed, do nothing
- If a button is pushed, then do:
  - if ADD, Validate Form, Write to Corkboard, **View Corkboard**

```
INSERT INTO PushPin ( url, owner_email, date_time, title)
VALUES ($url, $Email, $DateTime, $Title)
```

View PushPins

Abstract Code

- Find OWNER of PushPin ($PushPin)

```
SELECT owner_email FROM PushPin WHERE PushPin.title = $PushPin
```

- Display OWNER
- Find DATE and TIME of Pin

```
SELECT date_time FROM PushPin WHERE PushPin.
```

- Display DATE and TIME of Pin
- Find CATEGORY of Pin

```
SELECT category_type FROM PushPin pp INNER JOIN Category c ON pp.title = c.title WHERE
```

- Display CATEGORY
- Find COMMENTS and LIKES

```
SELECT text, date_time FROM `Comment` WHERE url = $PushPin.url
GROUP BY text
ORDER BY date_time ASC;
```

```
SELECT email, COUNT(email) FROM `Like` WHERE url = $PushPin.url
GROUP BY email;
```

- Display LIKES
- Display COMMENTS sorted by date/time
- Find TAGS of Pin

```
SELECT tags FROM PushPin pp INNER JOIN Category c ON pp.title = c.title WHERE pp.title =
$Title AND pp.url = $url AND pp.owner_email = $Email AND pp.date_time = $DateTime
```

- Display TAGS sorted alphabetically
- While no buttons are pushed, do nothing
- If a button is pushed, then do:
    - if FOLLOW, Write to USER, **View PushPin**

```
INSERT INTO Follows ( email, follower_email )
VALUES ( $Email, $Follower_Email )
```

    - if *<hyperlinked Category>*, **View Corkboard**
    - if *<image>*, GOTO Hyperlinked Site
    - if POST_COMMENT, Write to PushPin, **View PushPin**

```
INSERT INTO `Comment` ( date_time, text, email, url, title, owner_email )
VALUES ( $DateTime, $Text, $Email, $Url, $Title, $Owner_Email )
```

    - if THUMBS_UP, Write to PushPin, **View PushPin**

```
INSERT INTO `Like` ( email, date_time, url, title, owner_email )
VALES ($Email, $DateTime, $Url, $Title, $Owner_Email )
```

Search PushPins

Abstract Code

- Find related PushPins

```
SELECT title, url, description FROM PushPin pp INNER JOIN Tags t ON t.name = pp.tags WHERE
pp.title = $Title AND pp.date_time = $Date_time AND pp.owner_email = $Owner_email AND
pp.title = $Title;
```

- For Each PushPin in related PushPins:
    - Find OWNER, and CORKBOARD the Pin is on
    - If CORKBOARD is public
        - Display alphabetically PUSHPIN{hyperlink}, CORKBOARD, OWNER

```
SELECT owner_email, title, description FROM (SELECT title, url, description FROM PushPin pp
INNER JOIN Tags t ON t.name = pp.tags WHERE pp.title = $Title AND pp.date_time = $Date_time
AND pp.owner_email = $Owner_email AND pp.title = $Title)
ORDER BY description ASC;
```

- While not button is pushed, do nothing
- When button is pushed, then do:

  - ○ if PUSHPIN_DESCRIPTION, **View PushPin**

View Popular Tags

Abstract Code

- Find Popular Tags
- For Each Tag in related Popular Tags:
  - ○ Find #PUSHPINS, and #CORKBOARD the Tag is on
  - ○ Display TAG{hyperlink}, #PUSHPINS, and #CORKBOARDs sorted by most Pushpins followed by most Corkboards

```
SELECT name, COUNT (name), url, title FROM Tags INNER JOIN PushPin pp ON
Tags.name = pp.tag
GROUP BY name
ORDER BY COUNT(url) DESC, COUNT(title) DESC
LIMIT 5;
```

- While not button is pushed, do nothing
- When button is pushed, then do:
  - ○ if TAG, Search Tags

View Popular Sites

Abstract Code

- Find Popular Sites
- For Each Site in related Popular Sites:
- Find #PUSHPINS the Site is on
- Display SITE, #PUSHPINS sorted by most PushPins

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

- While not button is pushed, do nothing
- When button is pushed, then do:
  - ○ if TAG, Search Tags

View Stats

Abstract Code

- Find all USERS
- For Each USER within USERS

- Find PUBLIC CORKBOARDS
- Find PUBLIC PUSHPINS
- Find PRIVATE CORKBOARDS
- Find PRIVATE PUSHPINS

```
SELECT email, COUNT(title), COUNT(url), COUNT(password), COUNT(title, owner_email)
FROM `User` INNER JOIN Corkboard cb ON `User`.email = cb.email INNER JOIN
Private_Corkboard pc ON `User`.email = pc.email INNER JOIN PushPin pp ON `User`.email =
pp.email
GROUP BY email
ORDER BY COUNT(title) DESC, COUNT(title, owner_email) DESC;
```

- Display number of PUBLIC CORKBOARDS→ {Sorted by most: Wt=1}
- Display number of PUBLIC PUSHPINS
- Display number of PRIVATE CORKBOARDS → {Sorted by most: Wt=2}
- Display number of PRIVATE PUSHPINS