Modifications based on phase1 feedback:

- IFD
  - Added “Search” task on Home Page Form
  - Reversed “Password” task arrow on Password Form
  - Changed “View CorkBoard” arrow to one way on CorkBoard Form
  - Added “Watch CorkBoard” Task
- EER
  - Converted “comment” from a relation to a weak entity
  - Deleted “Follow CorkBoard” Task
  - Changed “create time” to “last updated time”
- DataTypes
  - Deleted “Datetime”
  - Added Comment Type “String, Not Null”
- Business Logic Constraints
  - Added “Users can only like a pushpin once" constraint
- Task Decomposition
  - Updated "Follow users, Like/Unlike task diagrams”
  - Updated “My CorkBoard and CorkBoard Updates” tasks
- Abstract Code
  - Added code to prevent people from "Watching" their own CorkBoards

Table of Contents

Abstract Code + SQL

Logging In

Abstract Code

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

```
SELECT Pin FROM `User` WHERE email='$Email';
```

        - If User record is found but User.Pin != '$Pin':
            - Go back to **Logging In** form, with error message.
        - Else:
            - Store login information as session variable '$Email'.
            - Go to **View Profile** form.
- Else *email* and *pin* input fields are invalid, display **Logging In** form, with error message.

Home Screen

Abstract Code

- Show "***CorkBoard Updates***", "***My CorkBoards***", "***Add CorkBoard***", "***Popular Tags***", and "***PushPin Search***" tabs.
- Upon:
    - Click ***CorkBoard Updates*** button- Jump to the ***CorkBoard Updates*** task.
    - Click ***My CorkBoards*** button- Jump to the ***My CorkBoards*** Profile task.
    - Click ***Add CorkBoard*** button- Jump to the ***Add CorkBoard*** task.
    - Click ***Popular Tags*** button- Jump to the ***Popular Tags*** task.
    - Click ***PushPin Search*** button- Jump to the ***PushPin Search*** task.

CorkBoard Updates

Abstract Code

- User clicked on ***CorkBoard Updates*** button from **Home Screen**:
- Run the **CorkBoard Updates** task: query for updates about the user and other users that the logged-in user watched or followed.
    - Find the last four CorkBoard updates; DisplayTitle, Owner, Date, Time of the last Four PushPin.

```
SELECT name, title, date, time FROM `User` LFET JOIN CorkBoard ON `User`.ID =
CorkBoard.userID WHERE `User`.email='$Email' ORDER BY date DESC, time DESC LIMIT
4;
```

When ready, user selects next action from choices in **Home Screen**

My CorkBoards

Abstract Code

- User clicked on ***My CorkBoards*** button from **Home Screen**:
- Run the ***My CorkBoards*** task: query for CorkBoards owned by the logged in user ($Email).
    - Sort the title of CorkBoards alphabetically;

```
SELECT title From CorkBoards RIGHT JOIN ‘User’ ON `User`.ID = CorkBoard.userID
WHERE email='$Email' ORDER BY title;
```

-
    - Display the number of PushPin on each board.

```
SELECT title, COUNT(PushPin.ID) FROM CorkBoards LEFT JOIN PushiPin ON
CorkBoard.ID= PushPin.corkboardID WHERE CorkBoard.email='$Email' GROUP BY title;
```

Add CorkBoard

Abstract Code

- User clicked on ***Add CorkBoard*** button from **Home Screen**:
- Run the **Add CorkBoard Task**:
    - User enters *Title* ($Title)
    - Find the *Category* ($Category) for the corkboard.
        - Display initial list of categories supported by CorkBoardIt

```
SELECT category_type FROM Category ORDER BY category_type;
```

-
    -
        - Find the *Category* ($Category)
    - Select *Visibility: Public or Private.*
        - If CorkBoard.visibility == ‘Public’, do nothing;
        - Else user *enters password* ($password).
- User Clicked on ***Add*** button.

View CorkBoard

Abstract Code

- User clicked on any public CorkBoard:
- Run the **View CorkBoard** Task:
    - Display *Category, Owner, Title* and *last updated date/time of the CorkBoard*.

```
SELECT category_type, name, title, data, time FROM `User` JOIN CorkBoard ON `User`.ID =
CorkBoard.userID JOIN Category ON CorkBoard.categoryID = Category.ID WHERE
`User`.email='$Email';
```

-
    - If logged in user owns the CorkBoard: `User`.email==CorkBoard.email, then:
        - Enable ***Add PushPin*** button.
    - Else:
        - Enable ***Follow*** button besides the owner’s name.

- If the current user clicks ***Follow*** button, then run **Follow Users** task.
- Else: do nothing.
- If the CorkBoard.Visibility == 'Public', then
  - If the current user clicks ***Watch*** button, then the CorkBoard’s owner is added to the current user’s watch list.

```
INSERT INTO Watch
VALUES (‘$WatchID’, ‘$Watch.userID’, ‘$CorkBoardID’)
```

  - Else: do nothing.
- Else: do nothing.
- Display *PushPins* in this CorkBoard collection

```
SELECT URL FROM CorkBoard JOIN PushiPins ON CorkBoard.ID=PushiPins.corkboardID
JOIN `User` ON `User`.ID = CorkBoard.userID WHERE `User`.email='$Email';
```

  - The thumbnail images are hyperlinked to **View PushPin**.
- Display the number of watchers of this CorkBoard.

```
SELECT COUNT (userID) FROM PublicCorkBoard LEFT JOIN WatchCorkboard ON
PublicCorkBoard.corkboardID = Watch. corkboardID JOIN `User` ON `User`.ID =
CorkBoard.userID WHERE email='$Email';
```

Follow Users

Abstract Code

- User clicked on ***Follow*** button:
- Run the **Follow Users Task**: add the CorkBoard’s owner to the current user’s *follow list*.

```
INSERT INTO Follow
VALUES (‘$FollowerUserID’, ‘$FolloweeUserID’)
```

View PushPin

Abstract Code

- User clicked on ***View PushPin*** button:
- Run the **View *PushPin*** Task:
  - Display *Owner, Title* and *first pinned date/time of the CorkBoard*.

```
SELECT name, title, update_date, updated_time FROM CorkBoard JOIN PushiPins ON
CorkBoard.ID=PushiPins.corkboardID JOIN `User` ON `User`.ID = CorkBoard.userID
WHERE `User`.email='$Email';
```

- Title is hyperlinked to CorkBoard Screen.
- If logged in user owns the CorkBoard: User.email==CorkBoard.email, then:
    - Do nothing.
- Else:
    - Enable ***Follow*** button besides the owner's name.
        - If the current user clicks ***Follow*** button, then run **Follow Users** task.
        - Else: do nothing.
- Display *PushPins* in this CorkBoard.

```
SELECT url FROM CorkBoard JOIN PushiPins ON CorkBoard.ID=PushiPins.corkboardID
JOIN `User` ON `User`.ID = CorkBoard.userID WHERE `User`.email='$Email';
```

-
    - The thumbnail images are hyperlinked to **View PushPin**.
- Display Tags: sorted alphabetically.

```
SELECT pushipin_tags FROM CorkBoard JOIN PushiPins ON
CorkBoard.ID=PushiPins.corkboardID JOIN `User` ON `User`.ID = CorkBoard.userID JOIN
Tags ON Tags.pushipinID = PushiPins. ID WHERE `User`.email='$Email' ORDER BY
pushipin_tags;
```

- Display the users who liked this Pushpin

```
SELECT name FROM `User` JOIN Like_Unlike ON `User`.ID= Like_Unlike.userID WHERE
Like_Unlike.ID = '$ID';
```

-
    - If the PushPin is owned by another user: `User`.email != '$Email', then:
        - If the current user clicks ***Like*** Button, then run **Like** Task.
        - Else: do nothing.
        - If the current user clicks ***Unlike*** Button, then run **Unlike** Task.
        - Else: do nothing.
    - Else: Disable ***Like*** or ***Unlike*** Button.
- Display the comments: comments sorted with latest comment on top.

```
SELECT Content FROM PushPin JOIN Comment ON PushPin.ID=Comment.pushpinID
WHERE PushPin.ID='$ID' ORDY BY date_time DESC;
```

-
    - If the current user enter a comment and clicks ***Post Comment*** Button, display the comment.

```
SELECT content FROM `User` JOIN Comment ON `User`.ID=Comment.userID WHERE
`User`.ID='$ID' ORDY BY date_time DESC LIMIT 1;
```

        - Else: do nothing.

Like/Unlike PushPin

Abstract Code

- If the user clicked on ***Like*** button:
    - Run the **Like/Unlike PushPin Task**: Display the logged in user's name under the Pushpin; change the button to ***Unlike***.

```
SELECT Name FROM `User` JOIN Like_Unlike ON `User`.ID= Like_Unlike.userID JOIN
PushPin ON PushPin.ID = Like_Unlike.pushpinID WHERE `User`.email='$Email' AND
PushPin.ID = '$pushipinID';
```

- If the user clicked on ***Unlike*** button:
    - Run the **Like/Unlike PushPin Task**: Remove the logged in user's name under the Pushpin; change the button to ***Like***.

```
DELETE FROM Like_Unlike WHERE LikerEmail='$Email';
```

Add Comments

Abstract Code

●User clicked on ***View PushPin*** from CorkBoard or ***Search PushPin***:

●Run the **View PushPin** task: query for information about the PushPin either from image or search results of which hyperlink is clicked

●Find and display the user's *Name* under which the PushPin is saved

```
SELECT name, title FROM CorkBoard JOIN PushiPins ON CorkBoard.ID =
PushiPins.corkboardID JOIN `User` ON `User`.ID = CorkBoard.userID WHERE
`User`.email='$Email';
```

●Find and display the *Date* and *Time* the PushPin was added and the title of the associated CorkBoard

```
SELECT url, date, time FROM CorkBoard JOIN PushiPins ON CorkBoard.ID =
PushiPins.corkboardID WHERE PushPin.ID='$ID';
```

●Display the image and the website addresses it comes from

```
SELECT url FROM PushPin WHERE ID ='$ID';
```

●For each image, enable the collaboration between users by letting users ***like/unlike*** and ***comment*** PushPin with limitations as following:

If the user already likes the PushPin, the button should read "unlike"

A user cannot like his or her own PushPin

Comments cannot be deleted and should be displayed with the latest comment appearing first

View Private CorkBoards

Abstract Code

- User clicked on a private CorkBoard indicated on the home page:
- Run the **View Private CorkBoard** Task:
  - Ask the user to type in password ('$Password')
  - If password is correct and log in is successful PrivateCorkBoard.Password == '$Password', then
    - Display *Category, Owner, Title* and *last updated date/time of the CorkBoard*

```
SELECT Type, Name, PrivateCorkBoard.Title, Date, Time FROM CorkBoard JOIN `User` ON
`User`.Email = CorkBoard.Email JOIN PrivateCorkBoard ON CorkBoard.Email
=  PrivateCorkBoard.Email AND CorkBoard.Title =  PrivateCorkBoard.Title WHERE
CorkBoard.Email = '$Email' AND CorkBoard.Title = '$Title';
```

    - Else: Display an error message and return to the **Password Protected** form
  - If logged in user owns the CorkBoard, then:
    - Enable ***Add PushPin*** button if this CorkBoard is owned by the current user.

```
INSERT INTO PushPin
VALUES ('$Email', '$Title', '$URL', '$Date', '$Time', '$Description') ;
```

  - Else:
    - Enable ***Follow*** button besides the owner's name.
      - If the current user clicks ***Follow*** button, then run **Follow Users** task.

```
INSERT INTO Follow
VALUES ('$FollowerEmail', '$FolloweeEmail') ;
```

      - Else: do nothing.
    - Display *PushPins* in this CorkBoard ($ID) collection, Enable ***Add PushPin*** button if this CorkBoard is owned by the current user.

```
SELECT email, title, uRL, date, time, Description FROM CorkBoard JOIN PushiPins ON
CorkBoard.ID = PushiPins.corkboardID WHERE CorkBoard.ID = '$ID';
```

    - The thumbnail images are hyperlinked to **View PushPin**.

Add PushPin

Abstract Code

- Check if the user is the owner of current CorkBoard,
- If yes User.Email == CorkBoard.Email
  - User is able to click on ***Add PushPin*** button from **View CorkBoard** form:
  - Run the **Add PushPin Task**:
    - User enters *URL of an image on the Web, a brief description (less than 200 characters) and a set of tags (optional)*

```
INSERT INTO PushPin
VALUES (‘$ID’, ‘$url’, ‘$date’, ‘$time’, ‘$description’, ‘$corkboardID’) ;
```

  - Else User.Email != CorkBoard.Email: Disable ***Add PushPin*** button
- *After URL of an image, a brief description (less than 200 characters) and* an optional set of *tags* are typed in:
  - Strings need to be parsed, tags(case-sensitive) need to be extracted and they need to be stored individually in the database
  - After the ***Add*** button is clicked, validate the form and add the PushPin to the CorkBoard, then the user should be returned to the **View CorkBoard** Screen

Search PushPin

Abstract Code

- User enters some text in the search box and clicked ***Pushpin Search*** button from **Home Screen**
- Run the **Search Pushpin** task:
  - Find all the Pushpins in database whose description, tags or category match the entered text ‘$Text’.
    - If the PushPin Search Results are public, then:
      - Display *PushPin Description, the associated CorkBoard title and the owner of the CorkBoard*.
      - Sort the results alphabetically by Pushpin description.
      - Hyperlink the description to display the **View PushPin**.

```
SELECT url, Description, title, Name FROM CorkBoard JOIN PushiPins ON CorkBoard.ID =
PushiPins.corkboardID JOIN `User` ON `User`.ID = CorkBoard.userID JOIN Tags ON
Tags.pushpinID = PushPin.ID WHERE PushPin.Description LIKE ‘$Text’ OR
Tags.pushpinTags LIKE ‘$Text’;
```

    - Else: Do not show the search results.

View Popular Tags

Abstract Code

- User clicked ***View Popular Tags*** button from homepage **MainMenu**
- Run the **View Popular Tags** task:
  - Find all the tags in database

- Count the PushPins each tag has
- Sort above findings by:
    - the number of PushPins with highest appearing first
- Display three columns: Tag, PushPins and Unique CorkBoards
    - Unique CorkBoards list the total number of CorkBoards that have a PushPin with the given tag
    - Limitation: Only count one time if multiple PushPins on the CorkBoard have the same tag

```
SELECT Description, COUNT(CONCAT(email, title, url)), COUNT(DISTINCT CONCAT(email,
title)) FROM CorkBoard JOIN PushiPins ON CorkBoard.ID = PushiPins.corkboardID JOIN
`User` ON `User`.ID = CorkBoard.userID JOIN Tags ON Tags.pushpinID = PushPin.ID
GROUP BY Tags.Description ORDER BY COUNT(CONCAT(email, title, url)) DESC;
```

View Popular Sites

Abstract Code

- User clicked ***View Popular Sites*** button from homepage **MainMenu**
- Run the **View Popular Sites** task:
    - Find the sites that have the most PushPin links
    - Count the number of the PushPins
    - Sort above findings by:
        - Number of PushPins with the highest appearing first

```
SELECT url, COUNT(CONCAT(email, title)) FROM CorkBoard JOIN PushiPins ON
CorkBoard.ID = PushiPins.corkboardID JOIN `User` ON `User`.ID = CorkBoard.userID
GROUP BY url ORDER BY COUNT(CONCAT(email, title)) LIMIT 4;
```

View CorkBoard Statistics

Abstract Code

- User clicked ***View CorkBoard Statistics*** button from **MainMenu**
- Run the **View CorkBoard Statistics** task: query for information about the user where user ID is the ID of the current user using the system from the HTTP Session
    - Find the number of public CorkBoards the user has
    - Find the number of private CorkBoards the user has
    - Find the count of PushPins across all public CorkBoards the user has
    - Find the count of PushPins across all private CorkBoards the user has
    - Sort above findings by:
        - the number of public CorkBoards each user has
        - second by the number of private CorkBoards each user has
        - print the current user's row in red text so the rank could be easily seen

```
SELECT name, totalPublicCB, totalPublicPP, totalPrivateCB, totalPrivatePP
FROM `User` JOIN
(SELECT `User`.ID AS ID, COUNT(PublicCorkBoard.ID) AS totalPublicCB,
COUNT(PublicPushPinID) AS totalPublicPP, COUNT(PrivateCorkBoard.ID) AS
totalPrivateCB, COUNT(PrivatePushPinID) totalPrivatePP
FROM
CorkBoard JOIN PushiPins
ON CorkBoard.ID = PushiPins.corkboardID
JOIN `User`
ON `User`.ID = CorkBoard.userID
LEFT JOIN PublicCorkBoard
ON PublicCorkBoard.corkboardID = CorkBoard.ID
LEFT JOIN PrivateCorkBoard
ON PrivateCorkBoard.corkboardID = CorkBoard.ID
LEFT JOIN
(SELECT PublicCorkBoard. corkboardID, PushPin.ID AS PublicPushPinID FROM
PublicCorkBoard LEFT JOIN PushiPin
ON PublicCorkBoard. corkboardID = PushPin.corkboardID
) AS PublicPushPin
ON CorkBoard.ID = PublicPushPin. corkboardID
LEFT JOIN
(SELECT PrivateCorkBoard. corkboardID, PushPin.ID AS PublicPushPinID FROM
PrivateCorkBoard LEFT JOIN PushiPin
ON PrivateCorkBoard. corkboardID = PushPin.corkboardID
) AS PrivatePushPin
ON CorkBoard.ID = PrivatePushPin. corkboardID) AS StatisticTable
ON StatisticTable.ID = `User`.ID;
```