CS 353 - Database Systems Project Final Report

PURE DIGITAL LIBRARY

Group 4

Kaan Aktürk - 21401120

Pelin Elbin Günay - 21402149

Hareem Larik - 21503645

Emine Ayşe Sunar – 21502099

Contents

1. Description

Our project is a Scientific Paper Data Management System called Pure Digital Library, where users can search scientific paper, like, view, download and add comment on those papers. Our main aim is to create a user friendly digital library service where social aspects are also added to a regular digital library by adding a like, comment option. The search bar has various options to filter and sort the results. Users are able to see the paper’s publication date, place, citation number, page number, number of views, and the number of downloads. In the system, a membership is required to access library data. Users must enter their name(s), surname, e-mail address, password and a unique username during the sign up process. After the login process, user can search for papers. In this system there are different user types such as, authors, reviewers, editors, or just a regular user. Each of them will have different roles with different functionality. For instance, during the sign up process, in addition to features that are shown above, students and teachers (regular users) could enter their institutions. Moreover, some users can subscribe to scientific journals. These users are considered as journal subscribers. To distinguish journal subscribers, subscription identity numbers are assigned to them. Each type of users can make comments on papers and journals. These comments can be seen by only library members. The sources are downloadable for members and their download count are also stored. This count also will be displayed to library members. Pages of the papers are calculated. The number of views of each paper is also counted when the users open them. Users can search sources according to their titles, release year, author, and institution of the author and/or language. Users can also distinguish the sources by checking the conference that the source is submitted or published. Editors and reviewers decide the publication status of the source. In addition to these options, results can be sorted according to like number, download count, page number, comment number and view count.

2.Contributions Of Each Group Member

Kaan Aktürk

- Wrote the HTML code in Bootstrap for the website(SignUpPage, EditorProfilePage, SearchPage, SubmitPage, AssignReviewerPage)
- Wrote the SQL code in java for creation and insertion of some tables.
- Wrote the PHP code in the HTML files for data retrieval from the database.

Pelin Elbin Günay

- Wrote the HTML code in Bootstrap for the website(viewPaperPage, ReviewerProfilePage, RegularUserProfilePage)
- Wrote the SQL code in java for creation and insertion of some tables.
- Wrote the PHP code in the HTML files for data retrieval from the database.

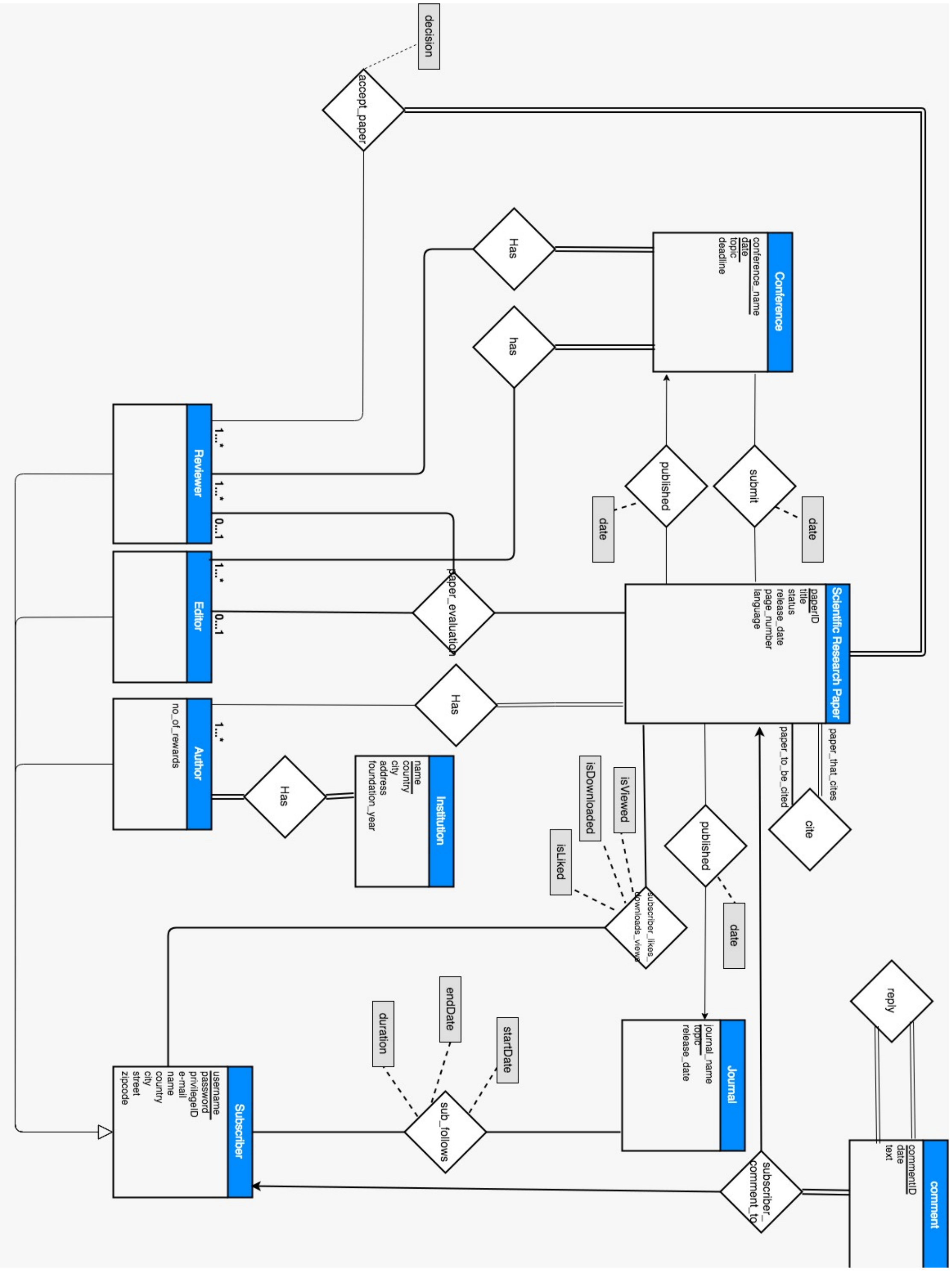
Hareem Larik

- Wrote the HTML code in Bootstrap for the website(LoginPage, ResultsPage, SearchPage, AuthorProfilePage, viewPaperPage, MyPapersPage, AssignReviewerPage, PendingPapersPage, FollowedPapersPage)
- Wrote the SQL code in java for creation and insertion of some tables.
- Wrote the PHP code in the HTML files for data retrieval from the database.

Emine Ayşe Sunar

- Wrote the SQL queries for insertion, modification and deletion from the database.
- Wrote the SQL code in java for creation and insertion of some tables.
- Wrote the PHP code in the HTML files for data retrieval from the database(LoginPage, SignUpPage, ResultsPage, SearchPage, DeletePage)

3.Final E/R


decision
accept_paper
Has
has
Conference
conference_name
date
topic
deadline
published
submit
date
date
Reviewer
1...*
1...*
0...1
Editor
1...*
0...1
Author
1...*
no_of_rewards
paper_evaluation
Scientific Research Paper
paperID
title
status
release_date
page_number
language
Has
Has
Institution
name
country
city
address
foundation_year
paper_that_cites
paper_to_be_cited
cite
published
date
isViewed
isDownloaded
isLiked
subscriber_likes_downloads_views
reply
Journal
journal_name
topic
release_date
endDate
duration
startDate
sub_follows
Subscriber
username
password
privilegeID
e-mail
name
country
city
street
zipcode
subscriber_comment_to
comment
commentID
date
text

4.Table Schemas

```
create tablesubscriber (
        username        varchar(20) primary
        key, password
                        varchar(20) not
        null, privilegedID     int,
        email           varchar(40) notnull,
        name            varchar(40) notnull,
        country         varchar(30),
        city            varchar(40),
        street          varchar(40),
        zip-code        int
);
```

```
create table scientific_research_paper (
        paperID                 int primary
        key, title              varchar(100)not
        null,
        status                  int not null,
        page_number             int ,
        language                varchar(20),
        release_date            int
);
```

```
create tablejournal (
        journal_name            varchar(40) primary
        key, topic              varchar(15),
        release_date            int
);
```

```
create table conference (
        conference_name     varchar(40),
        date                date,
        topic               varchar(15),
        deadline            date,
        primary key (conference_name, date)
);

create table reply (
        replyID             int primary key,
        commentedToID       int,
        foreign key (replyID) references comment,
        foreign key (commentedToID) references comment
);

create table paper_publish_journal (
        paperID             int primary key,
        journal_name        varchar(40),
        publication_date    int,
        foreign key (journal_name) references journal,
        foreign key (paperID) references scientific_research_paper
);



create table paper_publish_conference (
        paperID             int primary key,
        conference_name     varchar(40),
        publication_date    int,
        foreign key (conference _name) references conference,
        foreign key (paperID) references scientific_research_paper
);



create table institution (
        institution_name    varchar(40) primary key,
        foundation_year     int,
        country             varchar(30),
        city                varchar(40),
        street              varchar(40),
        zip-code            int
);


create table subscriber_comment_paper (
        username      varchar(20),
        paperID             int,
        foreign key ( username) references subscriber,
        foreign key (paperID) references scientific_research_paper
        primary key (paperID, username),

);



create table subscriber_follows_journal (
        start date          int
        end date            int
        duration            int
        username            varchar(20),
        journal_name        varchar(40),
        primary key (username, journal_name)
);



create table subscriber_likes_downloads_views_paper (
        username            varchar(20),
        paperID                     int,
        isDownloaded        int,
        isLiked             int,
        isViewed            int,
        foreign key (username) references subscriber (username)
        foreign key (paperID) references scientific_research_paper (paperID)
        primary key (username, papered)
);
```

```
create table author_institution (
        username            varchar(20),
        institution_name    varchar(40),
        primary key (username, institution_name)
);
```

```
create table author_has_paper (
        username            varchar(20),
        paperID                         int,
        primary key (username, paperID)
);
```

```
create table paper_citation (
        paper_that_cites    int,
        paper_to_be_cited   int,
        foreign key (paper_that_cites) references
scientific_research_paper,
        foreign key (paper_to_be_cited) references
scientific_research_paper,
        primary key (paper_that_cites, paper_to_be_cited)
);
```

```
create table paper_submit_conference (
        conference_name     varchar(40),
        paperID             int,
        date                date,
        foreign key (paperID ) references scientific_research_paper,
        foreign key (conference_name) references conference,
        primary key (conference_name, paperID ,date)
);
```

```
create table conference_reviewer (
        username             varchar(20),
        conference_name      varchar(40),
         foreign key (conference_name) references conference,
        foreign key (username) references subscriber
        primary key (conference_name, username)

);



create table conference_editor (
        username             varchar(20),
        conference_name      varchar(40),
        foreign key (conference_name) references conference,
        foreign key (username) references subscriber
        primary key (conference_name, username)

);



create table paper_evaluation (
        reviewer_username          varchar(20),
        editor_username            varchar(20),
        paperID                    int,
        foreign key (reviewer_username) references subscriber
        foreign key (editor_username) references subscriber
        foreign key (paperID ) references scientific_research_paper
        primary key (reviewer_username, editor_username)

);


create table paper_acceptance (
        reviewer_username      varchar(20),
        papered                 int,
        decision                int,
        foreign key (reviewer_username) references subscriber,
        foreign key (paperID ) references scientific_research_paper,
        primary key (username, paperID)
);
```

5. Implementation Details

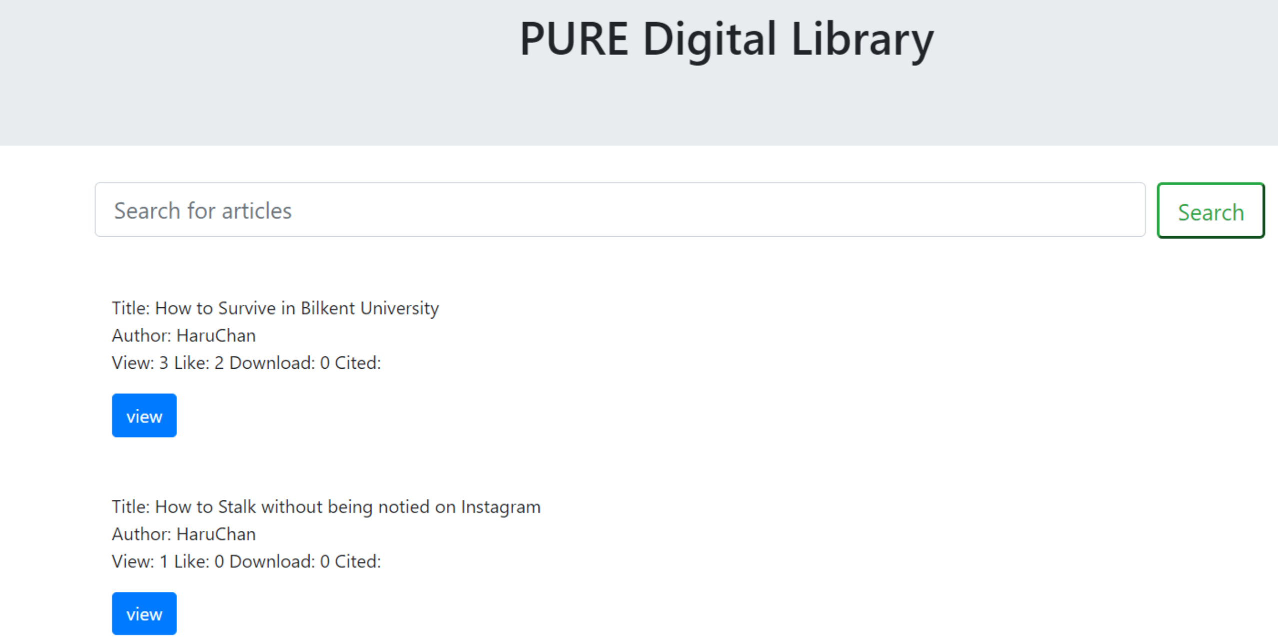
- We use Bootstrap to create the web pages. It is the HTML front-end framework that we used. Bootstrap creates a more user friendly website with better visual interface. Our implementation includes CSS and HTML. By using bootstrap we added buttons, forms, images, text fields, navigation bar, etc. to create an interactive website.
- We have finalized our tables using the feedback we have received from the design report and created them using Java. In our Java code we first connected to our database which is the MariaDB on Dijkstra server provided by Bilkent. Then we created and inserted some predefined tuples to our tables (We had predefined values to check the functionality of our website).
- We use Ajax(Asynchronous JavaScript and XML) PHP and HTML in coding. We used PHP in order to create connection with our database. After creating the connection we have retrieved data to display in our website and we have updated, inserted, deleted information from our database according to the inputs received from the user.
- Couple of problems we faced:
    - We accidently added some of our integer attributes as varchar's to our tables when inserting a tuple, for example we inserted '2' instead of adding 2 as an integer. Because of this when writing our sql queries we had problem writing the conditions in the when clause. The comparisons we made returned wrong results because we were comparing integers with varchar's.
    - We had problem transferring data between two websites. We were able to transfer data between some pages, however we had trouble transferring data between some other pages. We used different data transfer methods such as _GET[ ], _POST[ ] (get and post together), _SESSION[ ] etc.
    - We used Bootstrap to create our front design. So we had some trouble retrieving data from some of the Bootstrap components. For example, when using drop down forms, the data retrieved are usually strings, however sometimes we have noticed that when we use $_POST["language"] (language is the value of the drop down form), integers were given to us instead of the string value selected from the drop down form. So sometimes we received inetegrs and sometimes we received strings. So we had problem when comparing data.

6. Advanced Database Components Preliminary Results (Actual Outputs)

1. The Figure below shows, outputs to the input given from the search page, the input is “How to”,

SQL query:

```
SELECT DISTINCT R.title, R.username, T. tot_view, T.yoy_like, T.totdownload FROM  (SELECT DISTINCT
* FROM scientific_research_paper S NATURAL JOIN author_has_paper A WHERE S.title LIKE
'%$search_input%' AND S.language = '$language' AND S.release_date BETWEEN $year-5 AND $year+5
AND S.status = $status) T LEFT OUTER JOIN (SELECT paperID, SUM(isViewed) AS tot_view,
SUM(isLiked) AS tot_like, SUM(isDownloaded) AS tot_download FROM
subscriber_likes_downloads_views_paper GROUP BY paperID) R ON (R.paperID = T.paperID)
```

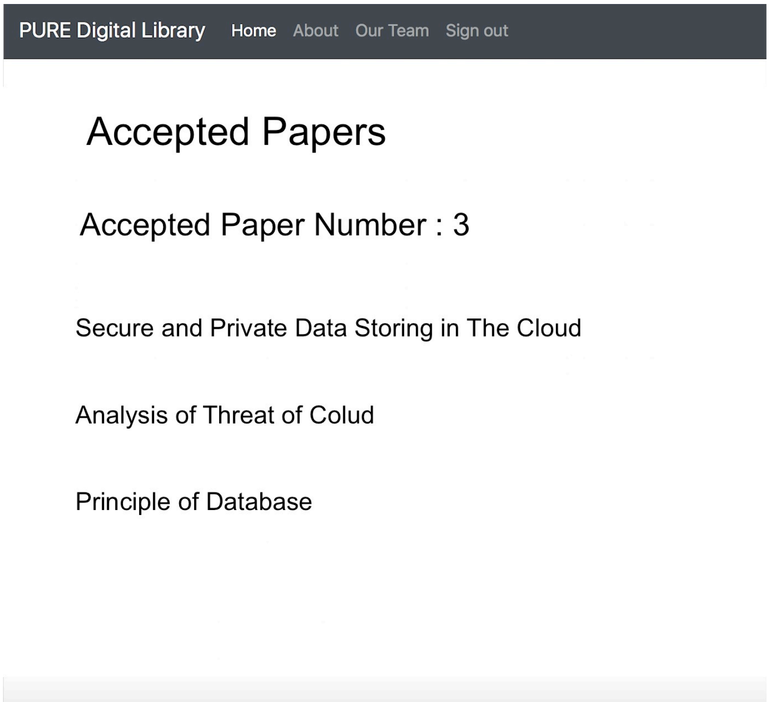


Title: How to Survive in Bilkent University
Author: HaruChan
View: 3 Like: 2 Download: 0 Cited:

Title: How to Stalk without being notied on Instagram
Author: HaruChan
View: 1 Like: 0 Download: 0 Cited:

Here, using our sql query we have enabled the user to see some attributes of the paper and disabled from seeing some other. For example the user does not have to see the paperID of the paper because that is an information considering only sql queries that we use to retrieve data.

2. The page bellow shows the Author’s Accepted Papers page. This page could be accessed only by the Author. In this page, the number of accepted papers the author has and the associated information are displayed:



Accepted Papers

Accepted Paper Number : 3

Secure and Private Data Storing in The Cloud

Analysis of Threat of Colud

Principle of Database

```
SELECT A.username, COUNT(*) AS accepted_paper_count
FROM author_has_paper A JOIN scientific_research_paper S ON (S.username = A.username)
WHERE A.status = 2 AND A.author_has_paper.username = $user_username
```

This report gives the information of how many papers the author has as accepted. Afterwards we display the name of the papers that are accepted using this report (it is very similar with the previous query but this time we don’t count the tuples, we display the title of the paper in each tuple):

```
SELECT A.username, A.title
FROM author_has_paper A JOIN scientific_research_paper S ON (S.username = A.username)
WHERE A.status = 2 AND A.author_has_paper.username = $user_username
```

3. We specified a trigger which disables liking a paper without viewing it.

```
create trigger check_like_1 after update on subscriber_likes_downloads_views_paper for each row if
isViewed = 0 and isLiked = 1 begin rollback end;
```

7.Users Manual

1. User can create account if s/he does not have any account by clicking sign-up button. User must type her/his username, password, e-mail, name, country, city, street, zip-code to create account and User must be specify her/his user type.
2. User can login into system if s/he has an account by clicking sign-in button. User must type her/his username and password to login.
3. User can search any scientific article by selecting filters(title, author, journal, institution, conference)and by selecting sort options(like, status, year) in search page.
4. By clicking into the paper in the result page, user see the full article.
5. In the view page, user can make comment by clicking the comment box and send her/his comment by clicking the send button.
6. In the view page, user can like this paper by clicking the thumbs up icon and dislike this paper by clicking the down arrow icon.
7. In the view page, user can follow this paper's journal by clicking the follow icon.
8. In the view page, user can see her/his profile page by clicking the profile photo icon after that user will be directed to profile page.
9. In the profile page user can see her/his personal information, s/he can see journals by clicking the Journals button and s/he can delete her/his account by clicking the Delete Account button.
10. After clicking My Papers button in profile page, author can see her/his papers and s/he can learn the people who like her/his paper by clicking the People who like button.
11. After clicking Submit Paper button in profile page, author is directed to the Submit Paper page. In this page, s/he can upload a file by typing her/his paper's title, page number, language, release date, conference name. After that by clicking the Upload File button, author can submit her/his paper.
12. In the profile page, editor can assign papers to reviewers by clicking the Assign Reviewer button. Editor can see the list of the reviewers according to the conference name selected and then can select a reviewer to assign papers to review. After clicking the Assign Paper button the assigning process will be done.
13. In the profile page, reviewer can see the papers assigned to her by clicking Pending Papers button and make a decision about the acceptance of the paper by clicking the thick and cancel buttons.
14. In the profile page, regular user can do only common operations described in step 9.
15. After clicking the Journals button, User can see her/his followed journals. S/he can unfollow them by clicking the unfollow button.