



**NUS**  
National University  
of Singapore

**CS2102**

**Database Systems**

**Task Sourcing**

<i>Sl.No.</i>	<i>Matric No.</i>	<i>Name</i>
1.	A0146738U	Lee Jun Han, Bryan
2.	A0192743Y	Bark HyeHyeon
3.	A0144843A	Tehseen Taha
4.	A0144848U	Tan Ping Min, Jasmine

## 1. Web Server, Server Page Language, DBMS

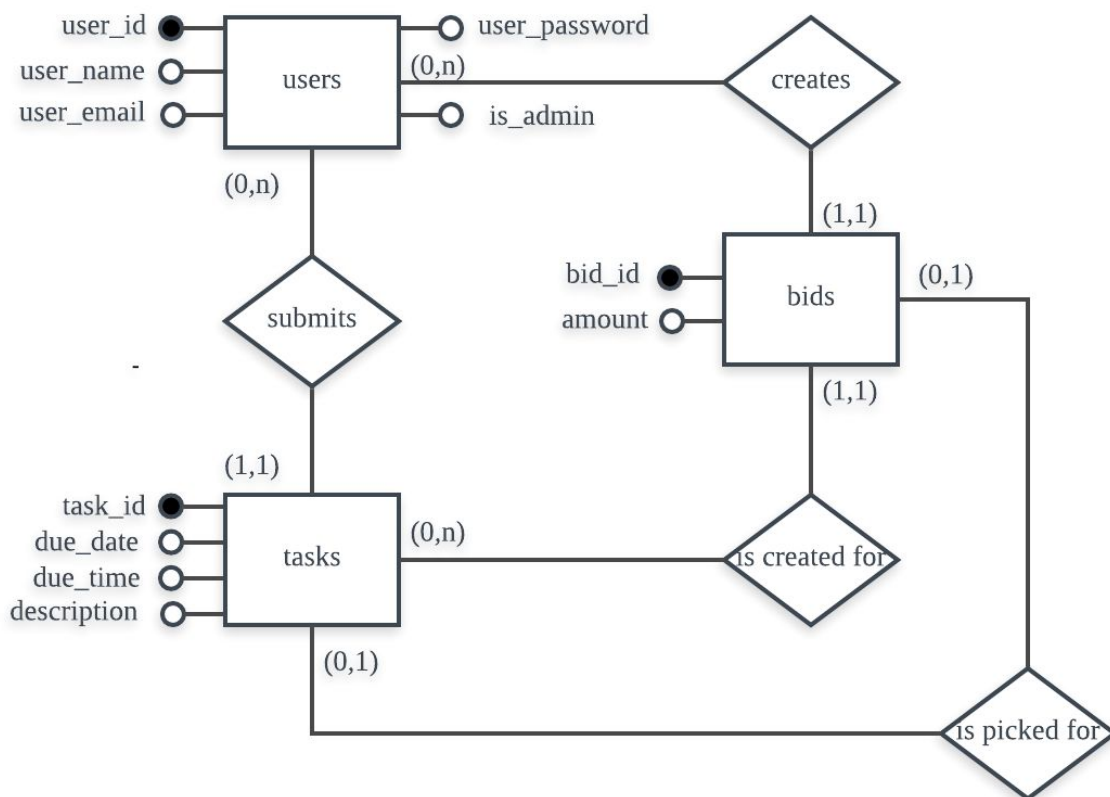
We used the recommended web server, server page language and DBMS stack, namely Bitnami MAPP stack, PostgreSQL, Apache and PHP.

Access our application at <http://localhost:8080/cs2102/login.php>

I.e. the application starts with the login page at login.php.

Make sure that you change the Postgres dbname and password as required in the php files before using the application.

## 2. Entity Relationship Diagram for Task Sourcing



### 3. SQL codes

#### 3.1. SQL DDL code for relational schema

```
CREATE TABLE users (  
  user_id SERIAL PRIMARY KEY,  
  user_name VARCHAR(64) NOT NULL,  
  user_password VARCHAR(64) NOT NULL,  
  user_email VARCHAR(64) NOT NULL,  
  is_admin BOOLEAN NOT NULL DEFAULT FALSE,  
  CHECK (user_email LIKE "%@%.%"),  
  CHECK (user_name NOT LIKE "%[^A-Z0-9]*%" AND LEN(user_name)>=5),  
  CHECK (user_password LIKE "%[0-9]%" AND user_password LIKE "%[A-Z]%"  
    AND user_password LIKE "%[!@#$$%^&*()-_+=.,;:~]%" AND  
  LEN(user_password)>=8);
```

```
CREATE TABLE tasks (  
  task_id SERIAL PRIMARY KEY,  
  owner_id INTEGER NOT NULL,  
  due_date DATE NOT NULL,  
  due_time TIME NOT NULL,  
  description VARCHAR(256) NOT NULL,  
  FOREIGN KEY owner_id REFERENCES users(user_id) ON DELETE CASCADE),  
  CHECK (due_date >= GetDate()));
```

```
CREATE TABLE bids (  
  bid_id SERIAL PRIMARY KEY,  
  bidder_id INTEGER NOT NULL,  
  task_id INTEGER NOT NULL,  
  amount INTEGER NOT NULL,  
  FOREIGN KEY bidder_id REFERENCES users(user_id) ON DELETE CASCADE,
```

```
FOREIGN KEY task_id REFERENCES task(task_id) ON DELETE CASCADE)  
CHECK (amount < 100000);
```

```
CREATE TABLE is_picked_for (  
task_id INTEGER PRIMARY KEY,  
bid_id INTEGER NOT NULL,  
FOREIGN KEY task_id REFERENCES task(task_id) ON DELETE CASCADE,  
FOREIGN KEY bid_id REFERENCES bid(bid_id) ON DELETE CASCADE);
```

### **3-2. SQL codes used to implement functions**

#### **1. Log In**

- **To find user information from users table based on user name**

```
SELECT user_id, user_name, is_admin FROM users WHERE user_name =  
'$_POST[username]'
```

- **To find user information from users table based on user name and password**

```
SELECT user_id, user_name FROM users WHERE user_password =  
'$_POST[password]' AND user_name = '$_POST[username]'
```

#### **2. Register**

- **To check if user name and user email already exist in users table**

```
SELECT user_name FROM users WHERE user_name = '$_POST[username]'
```

```
SELECT user_email FROM users WHERE user_email = '$_POST[email]'
```

- **To register user by inserting necessary information in users table**

```
INSERT INTO users (user_name ,user_password,user_email) VALUES  
( '$_POST[username]', '$_POST[password]', '$_POST[email]')
```

#### **3. All Tasks**

- **To display all available tasks that are not assigned yet**

```
SELECT t.description, t.due_date, t.due_time, u.user_name FROM tasks t,  
users u WHERE t.owner_id = u.user_id AND t.task_id NOT IN (SELECT  
p.task_id FROM is_picked_for p) AND t.owner_id <> $userid ORDER BY  
t.due_date
```

- **To search for tasks with task description**

```
SELECT t.task_id, t.description, t.due_date, t.due_time, u.user_name  
  
FROM tasks t, users u WHERE t.owner_id = u.user_id AND  
t.task_id
```

```
NOT IN (SELECT p.task_id FROM is_picked_for p) AND t.owner_id
<> $userid AND t.description LIKE '%$_POST[search_bar]%' ORDER BY
t.due_date
```

- **To bid for a certain task**

```
SELECT t.task_id FROM tasks t, users u WHERE t.owner_id = u.user_id AND
t.task_id NOT IN (SELECT p.task_id FROM is_picked_for p) AND t.owner_id <>
$userid ORDER BY t.due_date LIMIT 1 OFFSET $rownumber;
```

```
INSERT INTO bids(bidder_id, task_id, amount) VALUES ($userid, $taskno,
$amount)
```

#### 4. To Do

- **To display the tasks that the user is assigned to do**

```
SELECT u.user_name, t.due_date, t.due_time, t.description FROM
is_picked_for p, tasks t, users u, bids b WHERE b.bidder_id = $userid and
p.bid_id = b.bid_id and p.task_id = t.task_id and t.owner_id = u.user_id
```

- **To display the tasks that the user has bidded for and pending for assignment**

```
SELECT u.user_name, t.description, t.due_date, t.due_time, b.amount FROM
bids b, tasks t, users u WHERE t.owner_id = u.user_id and b.task_id =
t.task_id and b.bidder_id = $userid and t.task_id NOT IN (SELECT p.task_id
FROM is_picked_for p) ORDER BY t.due_date
```

- **To delete the tasks that are done by the user**

```
SELECT t.task_id FROM is_picked_for p, tasks t, users u, bids b WHERE
b.bidder_id = $userid and p.bid_id = b.bid_id and p.task_id = t.task_id
and t.owner_id = u.user_id ORDER BY t.due_date LIMIT 1 OFFSET $rownumber
```

```
DELETE FROM tasks WHERE task_id = $done_with
```

#### 5. Assign Task

- **To display bids for the task that user owns and did not assigned yet**

```
SELECT u.user_name, t.description, t.due_date, t.due_time, b.amount FROM
bids b, tasks t, users u WHERE b.bidder_id = u.user_id and b.task_id =
```

```
t.task_id and t.owner_id = $userid and t.task_id NOT IN (SELECT p.task_id
FROM is_picked_for p)
```

- **To pick a certain bid to assign the task**

```
SELECT t.task_id, b.bid_id FROM bids b, tasks t, users u WHERE b.bidder_id
= u.user_id and b.task_id = t.task_id and t.owner_id = $userid and
t.task_id NOT IN (SELECT p.task_id FROM is_picked_for p) ORDER BY user_id
LIMIT 1 OFFSET $rownumber
```

```
INSERT INTO is_picked_for(task_id, bid_id) VALUES ($task, $bidder)
```

## **6. Create Task**

- **To create a new task**

```
INSERT INTO tasks (owner_id, due_date, due_time, description) VALUES
($userid, '$_POST[due_date]', '$_POST[due_time]', '$_POST[description]')
```

## **7. My Tasks**

- **To display tasks that the user owns but has not assigned yet.**

```
SELECT t.description, t.due_date, t.due_time FROM users o JOIN tasks t on
o.user_id = t.owner_id WHERE t.task_id NOT IN (SELECT task_id FROM
is_picked_for) AND o.user_id = $userid ORDER BY t.due_date")
```

- **To display tasks that the user owns and has assigned**

```
SELECT t.description, bd.user_name as bidder, t.due_date, t.due_time FROM
users o JOIN tasks t on o.user_id = t.owner_id JOIN is_picked_for i on
i.task_id = t.task_id JOIN bids b on b.bid_id = i.bid_id JOIN users bd on
b.bidder_id = bd.user_id WHERE o.user_id = $userid ORDER BY t.due_date
```

- **To delete a task that the user owns**

```
SELECT t.task_id FROM tasks t, users u WHERE t.owner_id = u.user_id AND
t.owner_id = $userid ORDER BY t.due_date LIMIT 1 OFFSET $rownumber
```

```
DELETE FROM tasks WHERE task_id = $taskno
```

#### 4. Assertions

##### - users table

```
CHECK (user_email LIKE "%@%.%"),  
CHECK (user_name NOT LIKE "%[^A-Z0-9*]%" AND LEN(user_name)>=5),  
CHECK (user_password LIKE "%[0-9]%" AND user_password LIKE "%[A-Z]%"  
AND user_password LIKE "%[!@#$%^&*()-_+=.~;:~]%" AND  
LEN(user_password)>=8);
```

We check that the email provided is in a proper email format. We also check that the username only consists of lowercase alphabet and that the username is of at least length 5. We also check that the password the user creates contains at least one number, uppercase letter and special character, as per secure password standard, as well as check that the password is of at least length 8.

##### - tasks table

```
CHECK (due_date >= GetDate());
```

We check that the due date of the created task is not earlier than today's date.

##### - bids table

```
CHECK (amount < 100000);
```

We check that the amount submitted for the bid is not too extravagant.

#### 5. Screenshots of the web interface

- All Tasks





Work Lah

## MY TASKS

All Tasks To Do Assign Task Create Task My Tasks Sign Out

### All Available Tasks

- Search For Tasks With Description
- 
- 

Tasks you can bid for

Description	Due Date	Due Time	Owner Name	Make a bid
Celebrate my Birthday!	2018-12-25	23:59:00	jasmine	0
Help me get laundry	2018-12-06	12:12:12	jasmine	1
lalala	2018-12-12	12:12:12	jasmine	2
Help me with groceries	2018-12-12	12:12:13	jasmine	3

## ● My Tasks



Work Lah

## MY TASKS

All Tasks To Do Assign Task Create Task My Tasks Sign Out

### My Tasks

- Search For Tasks With Description
- 
- 

Tasks you have created

Description	Due Date	Due Time	Delete
babysitting	2018-10-28	20:30:00	0
make lanterns	2018-10-28	23:00:00	1
shift furniture	2018-10-28	23:00:00	2
do laundry	2018-10-28	23:00:00	3
clean my house	2018-10-29	23:00:00	4

## ● Make a bid for a task



Work Lah

## MY TASKS

All Tasks To Do Assign Task Create Task My Tasks Sign Out

### All Available Tasks

- Search For Tasks With Description
- 
- 

Tasks you can bid for

Description	Due Date	Due Time	Owner Name	Make a bid
Celebrate my Birthday!	2018-12-25	23:59:00	jasmine	5000
Help me get laundry	2018-12-06	12:12:12	jasmine	1
lalala	2018-12-12	12:12:12	jasmine	2
Help me with groceries	2018-12-12	12:12:13	jasmine	3

## ● Assign Task



## MY TASKS

[All Tasks](#) [To Do](#) [Assign Task](#) [Create Task](#) [My Tasks](#) [Sign Out](#)

### Assign Task

Tasks To Assign

Bidder Name	Description	Due Date	Due Time	Amount	Assign to this Bidder
johnappleseed	babysitting	2018-10-28	20:30:00	19	0
rachelhoo	clean my house	2018-10-29	23:00:00	19	1
rachelhoo	make lanterns	2018-10-29	23:00:00	19	2
rachelhoo	shift furniture	2018-10-28	23:00:00	19	3
rachelhoo	do laundry	2018-10-28	23:00:00	19	4

- **To Do**



## MY TASKS

[All Tasks](#) [To Do](#) [Assign Task](#) [Create Task](#) [My Tasks](#) [Sign Out](#)

### To Do

Tasks To Do

Owner Name	Due Date	Due Time	Description	Mark as Completed
barkhyehyeon	2018-10-29	23:00:00	clean my house	0
barkhyehyeon	2018-10-28	23:00:00	do laundry	1

Tasks Bidded for, Pending for Assignment

Owner Name	Description	Due Date	Due Time	Amount
johnappleseed	carwash	2018-10-28	19:40:00	30
barkhyehyeon	make lanterns	2018-10-28	23:00:00	19
barkhyehyeon	shift furniture	2018-10-28	23:00:00	19

- **Admin User**



## MY TASKS

[All Tasks](#) [To Do](#) [Assign Task](#) [Create Task](#) [My Tasks](#) [Sign Out](#)

Supply user data and enter to create

User ID:

User Name:

Password:

Email:

Supply userid and enter to search

User ID:

Supply task data and enter to create

Task ID:

Owner ID:

Due Date (e.g.: yyyy-mm-dd):

Due Time (e.g.: hh:mm:ss):

Description:

- **Admin User updating bids**

# Supply bidid and enter to search

Bid ID:

Submit

Bid ID:

Bidder ID:

Task ID:

Amount:

Submit

# Supply bidid and enter to search

Bid ID:

Submit

Update bid successful!