CS2102 Database Systems

Project Report

Topic A, Task-sourcing

Web application name: **VDXN**

GROUP 10

Derek Nam
A0139115L

Li Xiaowei
A0142325R

Natasha Koh Sze Sze
A0130894B

Nguyen Tuong Van
A0133387B

Table of Contents

1. Overview of Application

Problem Statement (Topic A, Task sourcing)

VDXN connects a user with someone that can get the task done for him/her. Whether it is to move large appliances or to do mundane chores, one can find a suitable fit for the job on VDXN. With our comprehensive task catalogue and bidding system, task creators can list their own tasks and task doers looking for a job can search through hundreds of jobs to find the perfect one to bid for.

Web Application Design

Web Server

Our team decided to adopt a PHP framework called Mini3 (https://github.com/panique/mini3) that provided a simple Model, View & Controller (MVC) abstraction. We chose MINI3 partly because it came as a vagrant package. It was thus very easy to get started developing even on computers that did not have the Linux, Apache, MySQL and PHP (LAMP) stack installed as vagrant could easily spin up an instance.

We used phpmyadmin as an interface to access the database.

Languages Used

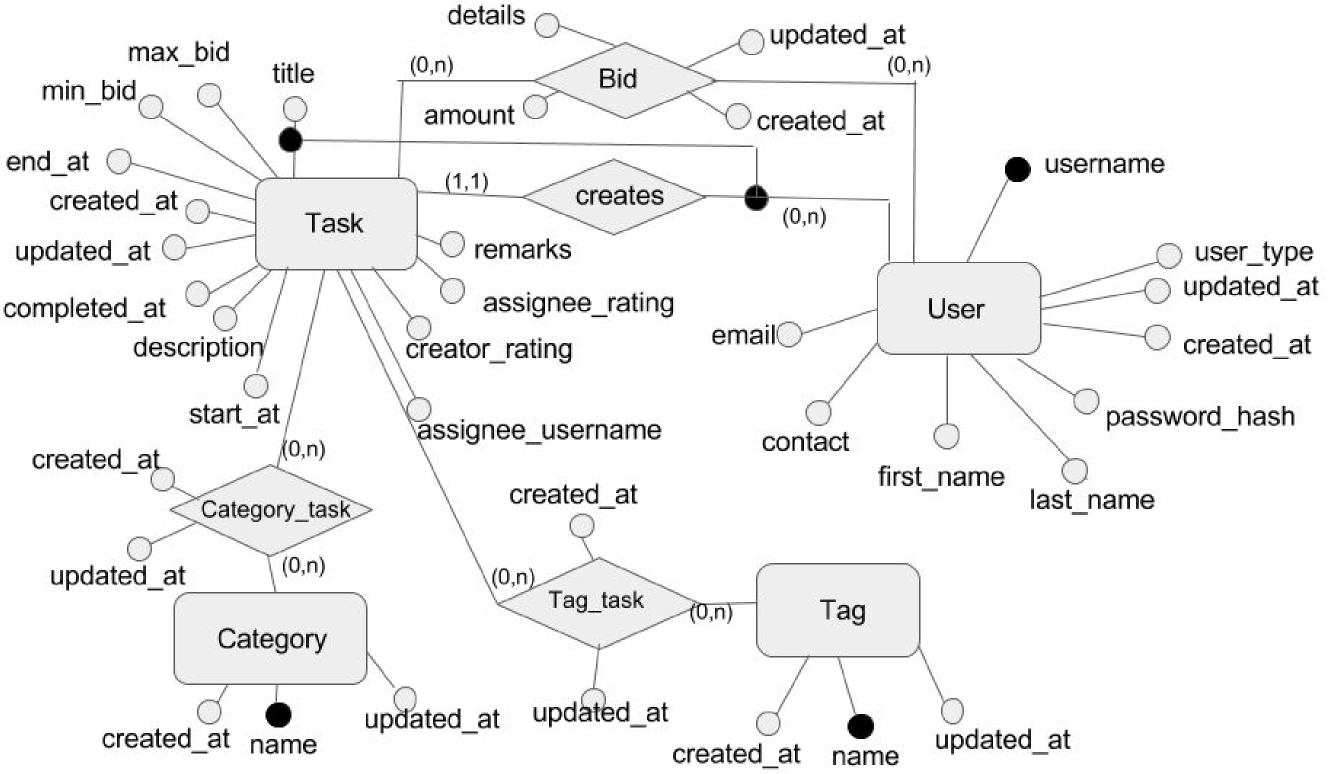
A combination of HTML, CSS, PHP and Javascript was used for the application. Most of the logic was written in PHP, with some Javascript in the front end.

Main Code File Structure

```
.
├─── /vdxn/application/Controller # Contains routing logic, retrieves data from the Model & updates
view
├─── /vdxn/application/Model       # Contains .SQL files that implement our Model methods.
                                    You may find all of our SQL queries here
```

```
├─── /vdxn/application/view       # UI template files and implementation
├─── /vdxn/public                 # Contains public image assets, and necessary CSS and JS files
├─── /vdxn/schema.sql             # Contains the SQL DDL code that represents our schema
└─── seed.sql                     # Database seed code and schema
```

2. Entity-Relationship Diagram



3. Relational Schema in SQL DDL code

```
CREATE TABLE User (
	username varchar(100),
	first_name varchar(100),
	last_name varchar(100),
	password_hash varchar(1000) NOT NULL,
	contact varchar(100),
	email varchar(100),
	created_at DATETIME NOT NULL,
	updated_at DATETIME,
	user_type ENUM('Admin', 'User'),
	PRIMARY KEY (username)
);

CREATE TABLE Task (
	title varchar(100),
	description varchar(1000),
	created_at DATETIME NOT NULL,
        updated_at DATETIME,
        start_at DATETIME NOT NULL,
        end_at DATETIME,
        min_bid numeric,
        max_bid numeric,
        creator_username varchar(100),
        assignee_username varchar(100),
        creator_rating numeric,
        assignee_rating numeric,
        completed_at DATETIME,
        remarks varchar(1000),
        CONSTRAINT min_bid CHECK(min_bid < max_bid),
        CONSTRAINT end_at CHECK(start_at < end_at),
        PRIMARY KEY (creator_username, title),
        FOREIGN KEY (creator_username) REFERENCES User(username) ON DELETE CASCADE ON UPDATE
CASCADE,
        FOREIGN KEY (assignee_username) REFERENCES User(username) ON DELETE CASCADE ON UPDATE
CASCADE
);

CREATE TABLE Category (
        name varchar(100),
        created_at DATETIME NOT NULL,
        updated_at DATETIME,
        PRIMARY KEY (name)
);

CREATE TABLE Category_task (
        category_name varchar(100) REFERENCES Category(name) ON DELETE CASCADE ON UPDATE
CASCADE,
        task_title varchar(100) REFERENCES Task(title) ON DELETE CASCADE,
        task_creator_username varchar(100) REFERENCES Task(creator_username) ON DELETE CASCADE
ON UPDATE CASCADE,
        created_at DATETIME NOT NULL,
        updated_at DATETIME,
        PRIMARY KEY (category_name, task_title, task_creator_username)
);

CREATE TABLE Bid (
        task_title varchar(100) REFERENCES Task(title) ON DELETE CASCADE ON UPDATE CASCADE,
        task_creator_username varchar(100) REFERENCES Task(creator_username) ON DELETE CASCADE
ON UPDATE CASCADE,
        bidder_username varchar(100) REFERENCES User(username) ON DELETE CASCADE ON UPDATE
CASCADE,
        details varchar(200),
        amount numeric NOT NULL,
        created_at DATETIME NOT NULL,
        updated_at DATETIME,
        PRIMARY KEY (task_title, task_creator_username, bidder_username)
);
```

```
CREATE TABLE Tag (
      name varchar(100),
      created_at DATETIME NOT NULL,
      updated_at DATETIME,
      PRIMARY KEY (name)
);

CREATE TABLE Tag_task (
      tag_name varchar(100) REFERENCES Tag(name) ON DELETE CASCADE ON UPDATE CASCADE,
      task_creator_username varchar(100) REFERENCES Task(creator_username) ON DELETE CASCADE
ON UPDATE CASCADE,
      task_title varchar(100) REFERENCES Task(title) ON DELETE CASCADE ON UPDATE CASCADE,
      created_at DATETIME NOT NULL,
      updated_at DATETIME,
      PRIMARY KEY (tag_name, task_title, task_creator_username)
);

CREATE TABLE Months(
  value varchar(3),
  name varchar(100) PRIMARY KEY
);
```

4. Database Seeding

To test the various functionalities in VDXN, we have generated some seed data. The code is located in `seed.sql` our submitted code folder. Here is a summary of the generated seed data:

1) 115 Users (With and without admin privileges)
2) 150 Tasks
    a) 50 Open tasks
    b) 50 Tasks that the creator has declared a winning bid
    c) 50 Tasks that have been completed and both creator and assignee rated each other
3) 7 Categories
4) 50 Tags
5) 1-5 Tags per Task
6) 1 Category per Task

Seed Generation

Here is a sample of the .php code used in our seed generation.

```
for($i = 0; $i < 150; $i++) {
  $time = date_format($faker->dateTimeThisDecade($max="now"), "Y-m-d H:i:s");
  $biasedminbid = $faker->biasedNumberBetween($min = 1, $max = 40, $function = 'sqrt');
  $biasedmaxbid = $faker->biasedNumberBetween($min = 41, $max = 120, $function = 'sqrt');

  $creator_username = $usernames[rand(0, count($usernames) - 1)];
  $assignee_username = $creator_username;
  while($assignee_username == $creator_username) {
      $assignee_username = $usernames[rand(0, count($usernames) - 1)];
  }

  $creator_rating = rand(3, 5);
  $assignee_rating = rand(3, 5);
  $completed_at = '2017-10-17 00:00:00';
  $remarks = 'Good job!';

  $title = $faker->jobTitle." ".($i + 1);

  $template = "INSERT INTO `Task` (`title`, `description`, `created_at`, `updated_at`,
`start_at`, `end_at`, `min_bid`, `max_bid` , `creator_username`, `assignee_username`,
`deleted_at`, `creator_rating`, `assignee_rating`, `completed_at`,`remarks`)VALUES ('$title',
'$faker->paragraph', '$time', '$time', '$timeStamp', '$timeStamp', '$biasedminbid',
'$biasedmaxbid', '$creator_username', NULL, NULL, '$creator_rating', '$assignee_rating',
'$completed_at', '$remarks');";

  echo $template."\n";
}
```

5. Implemented Database Functions

Overview of User Flow

There are 2 main types of users in VDXN - **task creators** and **task doers/bidders**.

Task creators create a task on their “My Tasks” dashboard and can subsequently view, edit or delete that task from that same dashboard. A task and bidding history can also be found on the user’s “My Tasks” dashboard. The created task is automatically listed on our public task catalogue that shows all the tasks that task doers can bid for. An interested bidder can click on any task listed on the “Task” page and view its details, place a bid for it, or browse through the list of bidders who are competing for that task. At any time during the bidding stage, bidders can edit or retract their bids.

After the bidding stage, an official “task doer” will be chosen by the task creator and the contact details of that doer will be displayed. Task creators and task doers can then liaise with each other to further arrange a meeting and rate each other when the task is done.

Administrators also get a comprehensive system dashboard where they can view useful statistics about the VDXN system, e.g. number of tasks and bidders in the system, what are the most popular tasks.

The following sections outline the various functionalities mentioned above (and more) and provide the SQL database statements used to implement them.

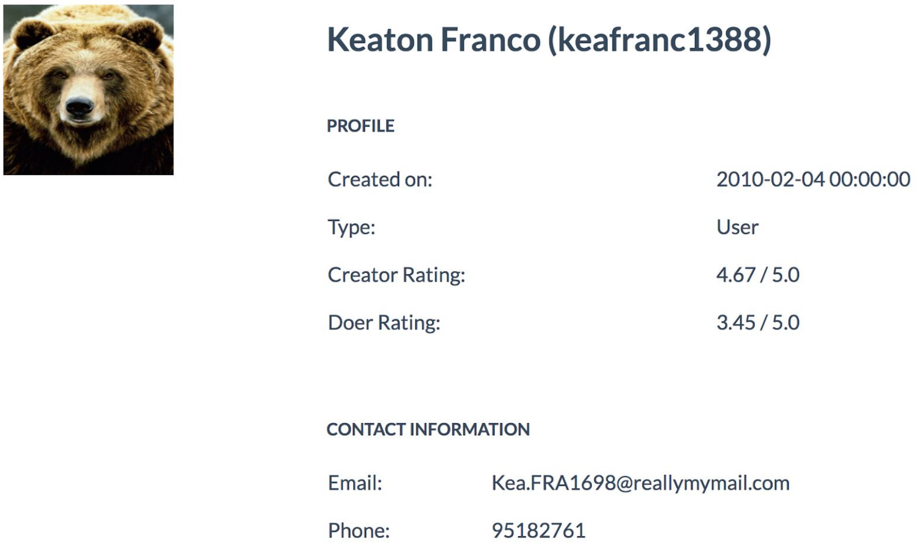
Application Functionality with SQL statements

Our SQL statements are executed in our database controller and are prepared statements / functions (located in `./application/Model`). PHP code, e.g. variables and pre-defined constants can be injected into those statements. Throughout this project, we refrained from using any database abstraction layers that would simplify our SQL code.

Legend:
`$[variable_name]` stands for variables used in our prepared SQL statements.

5.1 User CRUD and Profile (“My Profile” page)

Feature summary & Page Screenshot

Features: ● View user attributes, e.g. first name, last name, contact information, profile picture, user type (admin or normal user) ● Compute aggregated user rating ● Create a user (sign-up) ● Change user password	**Page Screenshot**  Keaton Franco (keafranc1388) PROFILE Created on: 2010-02-04 00:00:00 Type: User Creator Rating: 4.67 / 5.0 Doer Rating: 3.45 / 5.0 CONTACT INFORMATION Email: Kea.FRA1698@reallymymail.com Phone: 95182761

SQL Statements

View user attributes

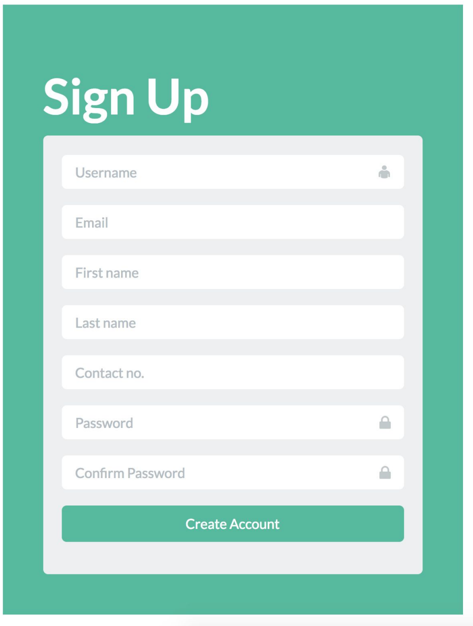
```
SELECT username, password_hash,
contact, email, user_type
FROM User
WHERE username=$username;
```

Compute aggregated user rating

```
SELECT
  ROUND(SUM(assignee_rating) /
  COUNT(assignee_rating), 2)
  AS rating
  FROM Task
  WHERE assignee_username = $username;

SELECT
  ROUND(SUM(creator_rating) /
  COUNT(creator_rating), 2)
  AS rating
  FROM Task
  WHERE creator_username = $username;
```

Sign up

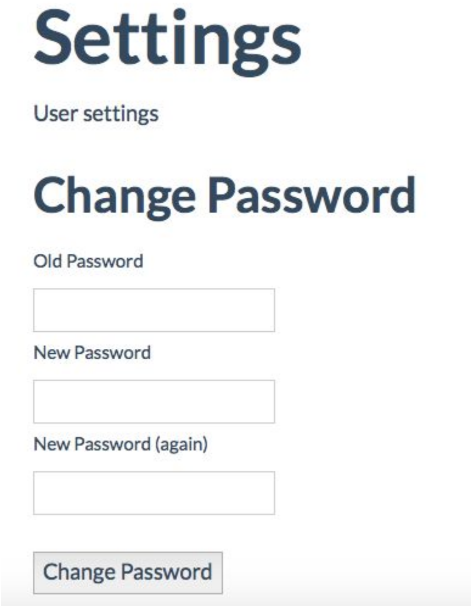
```
INSERT INTO User
    (`username`, `first_name`,
`last_name`, `password_hash`,
`contact`, `email`, `created_at`,
`updated_at`, `user_type`)
    VALUES (
      '$username',
      '$firstName',
      '$lastName',
      '$passwordHash',
      '$contactNumber',
      '$email',
      '$time',
      '$time',
      'User'
    );
```



Sign Up

Change User Password

```
UPDATE User SET
`password_hash`='$password_new_hash'
 WHERE `username`='$username';
```



Settings

User settings

Change Password

5.2 User Dashboard for Tasks & Bids ("MyTasks" page)

Feature summary & Page Screenshot

Features:

Task CRUD for the user's owned tasks

- Create a task
- View a task
- Edit a task
- Delete a task

"Created by me" tab

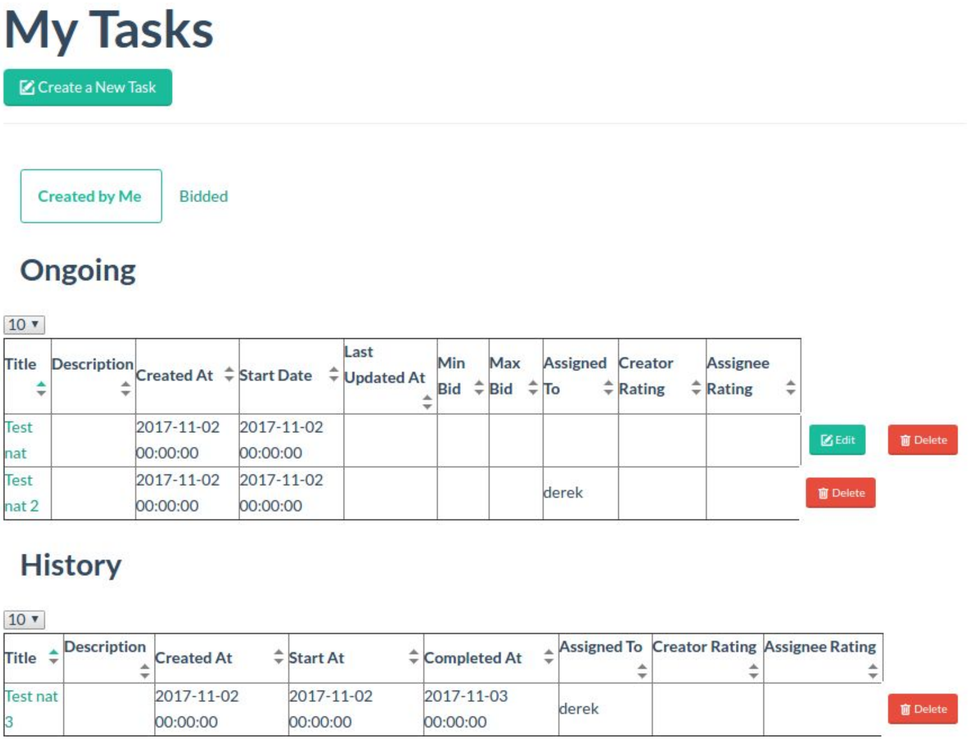
- View all tasks created by user
- View a history of all tasks that were completed
- Pagination for long tables

"Bidded" tab

- View all tasks bidded by user
- View a history of all bids that had been placed before, for which a task doer was selected
- Pagination for long tables

Page Screenshots

When selecting the "Created by me" Tab



My Tasks

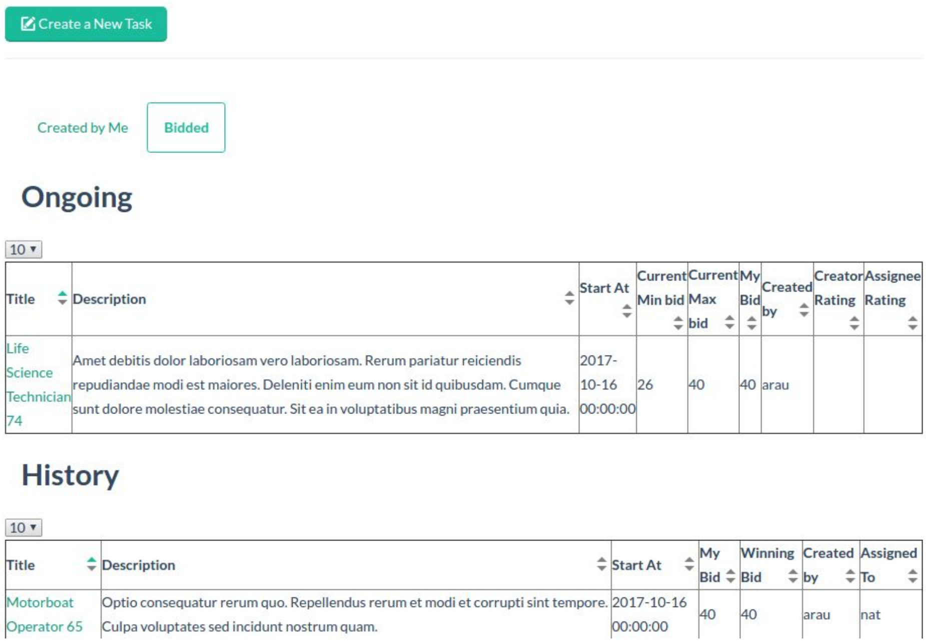
Ongoing

Title	Description	Created At	Start Date	Last Updated At	Min Bid	Max Bid	Assigned To	Creator Rating	Assignee Rating
Test nat		2017-11-02 00:00:00	2017-11-02 00:00:00						
Test nat 2		2017-11-02 00:00:00	2017-11-02 00:00:00				derek		

History

Title	Description	Created At	Start At	Completed At	Assigned To	Creator Rating	Assignee Rating
Test nat 3		2017-11-02 00:00:00	2017-11-02 00:00:00	2017-11-03 00:00:00	derek		

When selecting the "Bidded" Tab



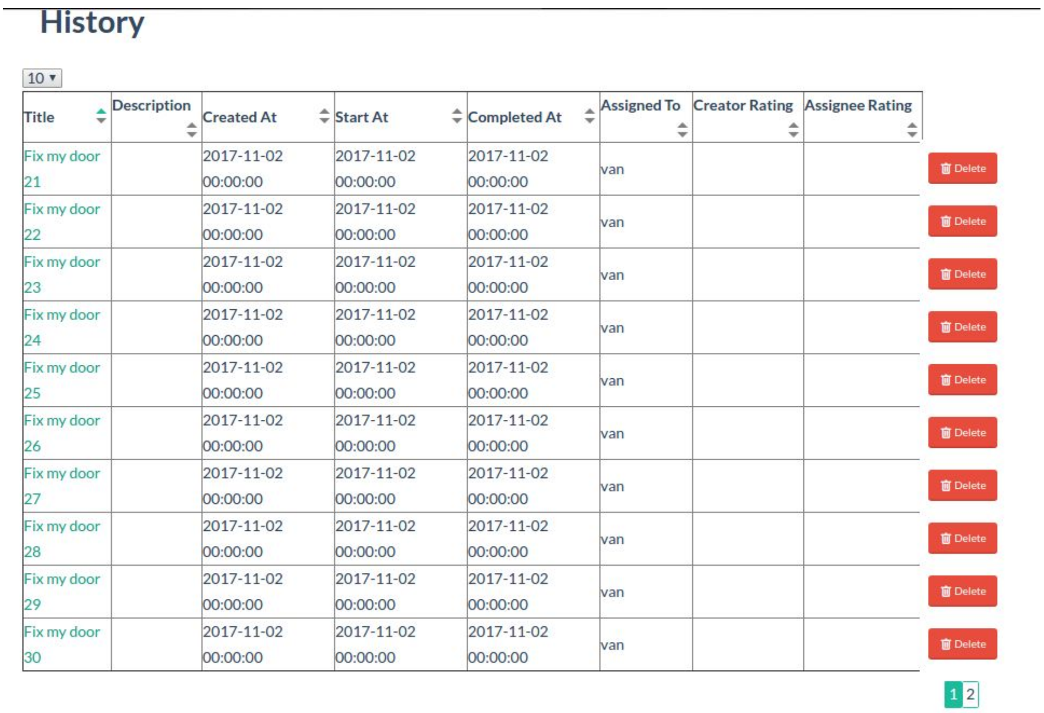
Ongoing

Title	Description	Start At	Current Min bid	Current Max bid	My Bid	Created by	Creator Rating	Assignee Rating
Life Science Technician 74	Amet debitis dolor laboriosam vero laboriosam. Rerum pariatur reiciendis repudiandae modi est maiores. Deleniti enim eum non sit id quibusdam. Cumque sunt dolore molestiae consequatur. Sit ea in voluptatibus magni praesentium quia.	2017-10-16 00:00:00	26	40	40	arau		

History

Title	Description	Start At	My Bid	Winning Bid	Created by	Assigned To
Motorboat Operator 65	Optio consequatur rerum quo. Repellendus rerum et modi et corrupti sint tempore. Culpa voluptates sed incidunt nostrum quam.	2017-10-16 00:00:00	40	40	arau	nat

Pagination when there are many records



History

10 ▾

Title	Description	Created At	Start At	Completed At	Assigned To	Creator Rating	Assignee Rating	
Fix my door 21		2017-11-02 00:00:00	2017-11-02 00:00:00	2017-11-02 00:00:00	van			Delete
Fix my door 22		2017-11-02 00:00:00	2017-11-02 00:00:00	2017-11-02 00:00:00	van			Delete
Fix my door 23		2017-11-02 00:00:00	2017-11-02 00:00:00	2017-11-02 00:00:00	van			Delete
Fix my door 24		2017-11-02 00:00:00	2017-11-02 00:00:00	2017-11-02 00:00:00	van			Delete
Fix my door 25		2017-11-02 00:00:00	2017-11-02 00:00:00	2017-11-02 00:00:00	van			Delete
Fix my door 26		2017-11-02 00:00:00	2017-11-02 00:00:00	2017-11-02 00:00:00	van			Delete
Fix my door 27		2017-11-02 00:00:00	2017-11-02 00:00:00	2017-11-02 00:00:00	van			Delete
Fix my door 28		2017-11-02 00:00:00	2017-11-02 00:00:00	2017-11-02 00:00:00	van			Delete
Fix my door 29		2017-11-02 00:00:00	2017-11-02 00:00:00	2017-11-02 00:00:00	van			Delete
Fix my door 30		2017-11-02 00:00:00	2017-11-02 00:00:00	2017-11-02 00:00:00	van			Delete

1 2

SQL Statements

View all ongoing tasks

```
SELECT title, description, created_at,
start_at, updated_at, min_bid, max_bid,
assignee_username, creator_rating,
assignee_rating
FROM Task WHERE
creator_username='$creator_username’ AND
completed_at IS NULL
ORDER BY ASC
LIMIT $limit
OFFSET $offset;
```

View a history of all tasks that were completed before

```
SELECT title, description, created_at, start_at,
updated_at, min_bid, max_bid, assignee_username,
creator_rating, assignee_rating
FROM Task WHERE
creator_username='$creator_username’ AND
completed_at IS NOT NULL
ORDER BY ASC
LIMIT $limit
OFFSET $offset;
```

View all tasks user has bidded for and for which no one has been assigned

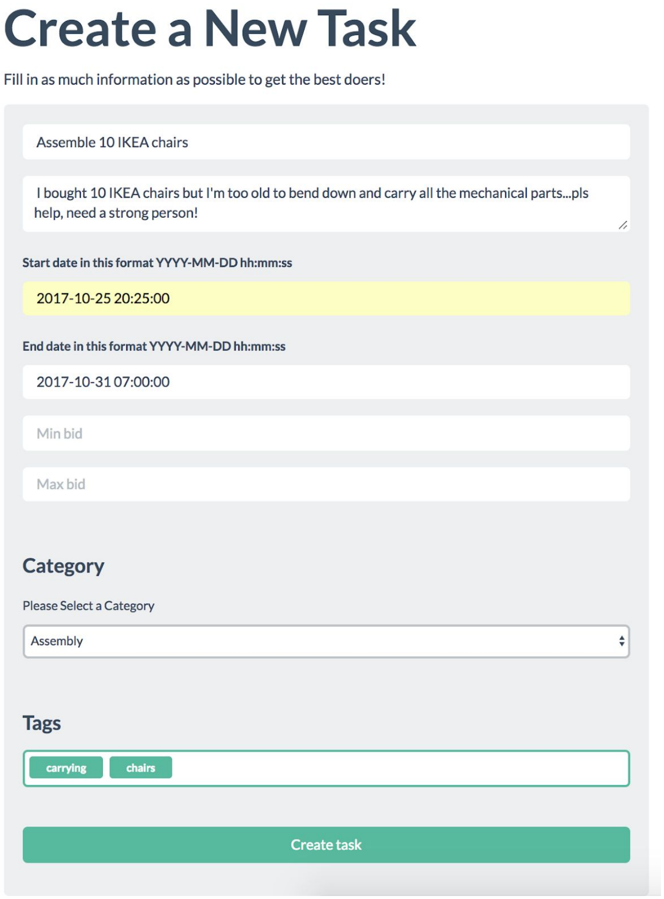
```
SELECT title, description,
      start_at, s.curr_min_bid as
minB,s.curr_max_bid as maxB, myBid,
      creator_username, creator_rating,
      Assignee_rating
FROM
(
SELECT title, description, start_at,
b1.amount AS myBid, creator_username,
creator_rating, assignee_rating
FROM Task
INNER JOIN Bid b1

ON Task.creator_username =
b1.task_creator_username
AND Task.title = b1.task_title
WHERE b1.bidder_username =  '$bdername'
AND Task.assignee_username IS NULL
)t

INNER JOIN
(SELECT b2.task_title as b2_title,
b2.task_creator_username as b2_creator,
MAX(b2.amount) as curr_max_bid,
MIN(b2.amount) as curr_min_bid
FROM Bid b2
GROUP BY b2.task_title)s

ON s.b2_title = t.title
AND s.b2_creator = t.creator_username

ORDER BY $order_by ASC LIMIT $limit
OFFSET $offset;
```

View a history of all bids that had been placed before, for which a task doer was selected

```
SELECT title, description, start_at, myBid,
winningBid.amount as winning_bid,
creator_username, assignee_username
FROM (
    (
    SELECT title, description,start_at,
myBid.amount as myBid, creator_username,
assignee_username
    FROM Task
      INNER JOIN Bid myBid
ON Task.creator_username=task_creator_username
AND Task.title = myBid.task_title
      INNER JOIN User
ON username=bidder_username
      WHERE bidder_username='$bdername' AND
assignee_username IS NOT NULL
      ) t
      INNER JOIN Bid winningBid
      ON
winningBid.bidder_username=t.assignee_username
AND t.title=winningBid.task_title
      INNER JOIN User winner
ON winner.username = t.assignee_username);
```

Creating a Task

```
INSERT INTO Task (
`title`,
`description`,
`created_at`,
`updated_at`,
`start_at`,
`end_at`,
`min_bid`,
`max_bid`,
`creator_username`,
`assignee_username`,
`completed_at`,
`creator_rating`,
`assignee_rating`)
VALUES (
  '".$task_params['title']."',
  '".$task_params['description']."',
  '$time',
  NULL,
  '$start_at_date',
  '$end_at_date',
  '$min_bid',
  '$max_bid',
  '".$_SESSION['user']->username."',
  NULL,
  NULL,
  NULL,
  NULL
);
```



Create a New Task

Fill in as much information as possible to get the best doers!

Assemble 10 IKEA chairs

I bought 10 IKEA chairs but I'm too old to bend down and carry all the mechanical parts...pls help, need a strong person!

Start date in this format YYYY-MM-DD hh:mm:ss

2017-10-25 20:25:00

End date in this format YYYY-MM-DD hh:mm:ss

2017-10-31 07:00:00

Min bid

Max bid

Category

Please Select a Category

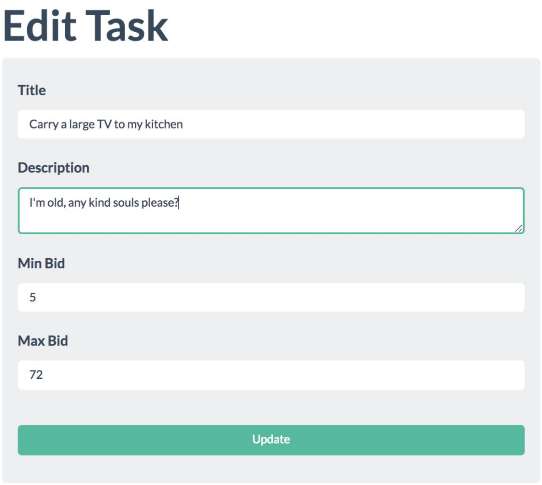
Assembly

Tags

carrying chairs

Edit a Task

```
UPDATE Task SET
  title=$title
  description='".$params['title'].
   "', updated_at='".$time.
   "', min_bid='".$params['min_bid'].
   "', max_bid='".$params['max_bid'].
   "' WHERE title='".$task_title."' AND
creator_username='".$task_creator_userna
me."';
```



Edit Task

Title

Carry a large TV to my kitchen

Description

I'm old, any kind souls please?

Min Bid

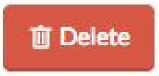
5

Max Bid

72

Delete a Task

```
DELETE FROM Task WHERE title=$title AND
creator_username=$creator_username;
```



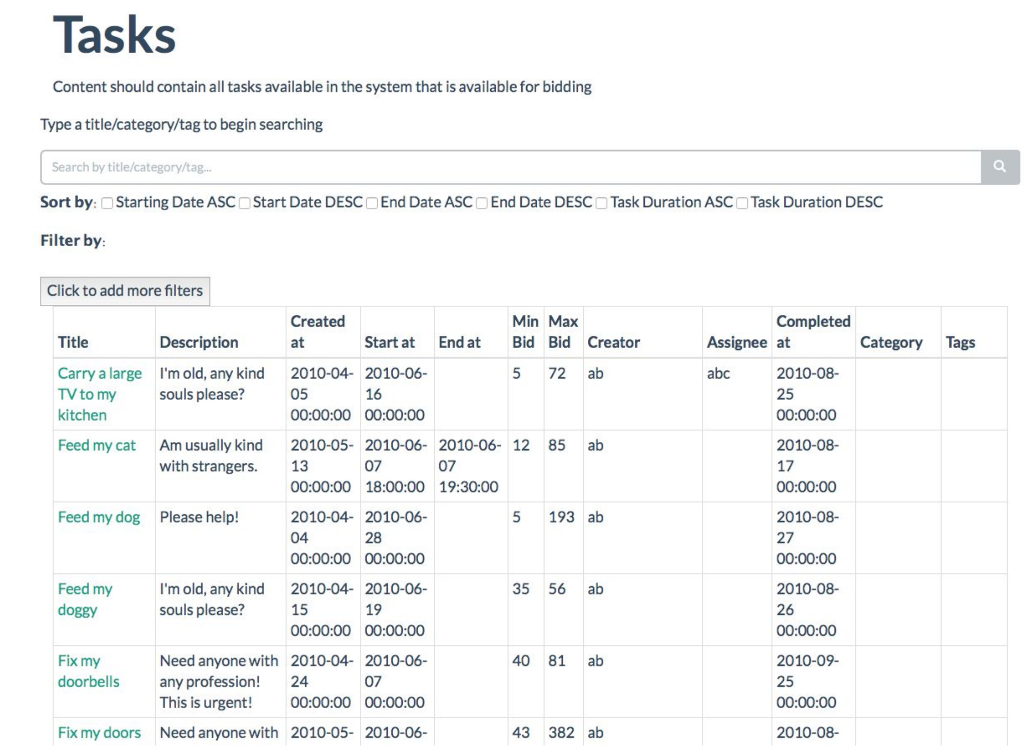
5.3 Task Search & Catalogue ("Task" page)

Feature summary & Page Screenshot

Features:

- View all tasks
- Search by title/category/tag
- Sort tasks by start date, end date and duration
- Filter tasks by category, creator rating, duration and tags

Page Screenshot



Tasks

Content should contain all tasks available in the system that is available for bidding

Type a title/category/tag to begin searching

Search by title/category/tag...

Sort by: □ Starting Date ASC □ Start Date DESC □ End Date ASC □ End Date DESC □ Task Duration ASC □ Task Duration DESC

Filter by:

Click to add more filters

Title	Description	Created at	Start at	End at	Min Bid	Max Bid	Creator	Assignee	Completed at	Category	Tags
Carry a large TV to my kitchen	I'm old, any kind souls please?	2010-04-05 00:00:00	2010-06-16 00:00:00		5	72	ab	abc	2010-08-25 00:00:00		
Feed my cat	Am usually kind with strangers.	2010-05-13 00:00:00	2010-06-07 18:00:00	2010-06-07 19:30:00	12	85	ab		2010-08-17 00:00:00		
Feed my dog	Please help!	2010-04-04 00:00:00	2010-06-28 00:00:00		5	193	ab		2010-08-27 00:00:00		
Feed my doggy	I'm old, any kind souls please?	2010-04-15 00:00:00	2010-06-19 00:00:00		35	56	ab		2010-08-26 00:00:00		
Fix my doorbells	Need anyone with any profession! This is urgent!	2010-04-24 00:00:00	2010-06-07 00:00:00		40	81	ab		2010-09-25 00:00:00		
Fix my doors	Need anyone with	2010-05-	2010-06-		43	382	ab		2010-08-		

SQL Statements

Features:

View all tasks

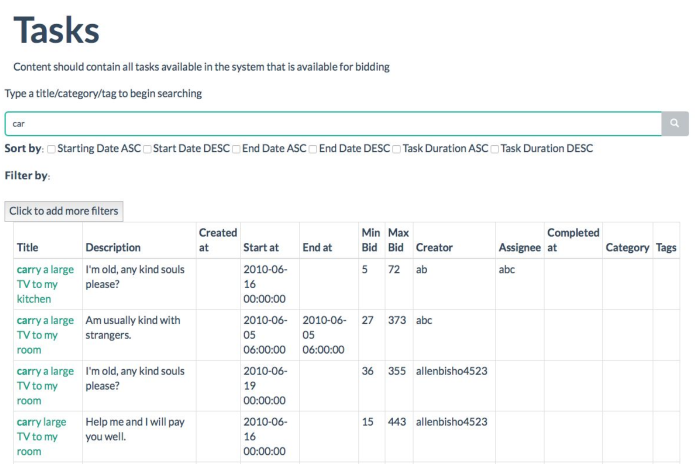
```
SELECT * FROM Task;
```

Search by title/category/tag

```
SELECT * FROM (
Task t LEFT JOIN Tag_task g ON t.title =
g.task_title AND t.creator_username =
g.task_creator_username) LEFT JOIN
Category_task c ON t.title = c.task_title
AND t.creator_username =
c.task_creator_username
WHERE t.title LIKE "%' . $search_string .
'%"  OR g.tag_name LIKE "%' .
$search_string . '%" OR c.category_name
LIKE "%' . $search_string . '%";
```

Find all tags containing specific pattern

```
SELECT * FROM Tag t
WHERE t.name LIKE "%'.$search_string.'%";
```



Tasks

Content should contain all tasks available in the system that is available for bidding

Type a title/category/tag to begin searching

car

Sort by: □ Starting Date ASC □ Start Date DESC □ End Date ASC □ End Date DESC □ Task Duration ASC □ Task Duration DESC

Filter by:

Click to add more filters

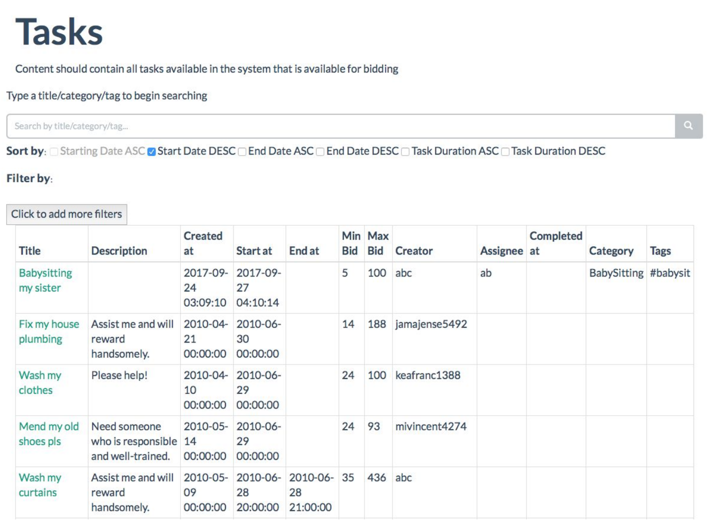
Title	Description	Created at	Start at	End at	Min Bid	Max Bid	Creator	Assignee	Completed at	Category	Tags
carry a large TV to my kitchen	I'm old, any kind souls please?		2010-06-16 00:00:00		5	72	ab	abc			
carry a large TV to my room	Am usually kind with strangers.		2010-06-05 06:00:00	2010-06-05 06:00:00	27	373	abc				
carry a large TV to my room	I'm old, any kind souls please?		2010-06-19 00:00:00		36	355	allenbisho4523				
carry large TV to my room	Help me and I will pay you well.		2010-06-16 00:00:00		15	443	allenbisho4523				

Sort tasks by start date, end date and duration

```
SELECT Task.title, Task.description,
Task.created_at, Task.updated_at,
Task.start_at, Task.end_at,
Task.min_bid, Task.max_bid,
Task.creator_username,
Task.assignee_username,
Task.completed_at,
Task.remarks, TIMESTAMPDIFF(SECOND,
Task.end_at, Task.start_at) AS duration
FROM Task
ORDER BY $attribute_str;
```

Filter tasks by category, creator rating, duration and tags

```
SELECT t.title, t.description,
t.created_at, t.updated_at, t.start_at,
t.end_at,
t.min_bid, t.max_bid, t.creator_username,
t.assignee_username, t.completed_at,
t.remarks, TIMESTAMPDIFF(SECOND,
t.end_at, t.start_at) AS duration
FROM (Task t LEFT JOIN Tag_task g ON
t.title = g.task_title AND
t.creator_username =
g.task_creator_username) LEFT JOIN
Category_task c ON t.title = c.task_title
AND t.creator_username =
c.task_creator_username
WHERE $str;
```



Tasks

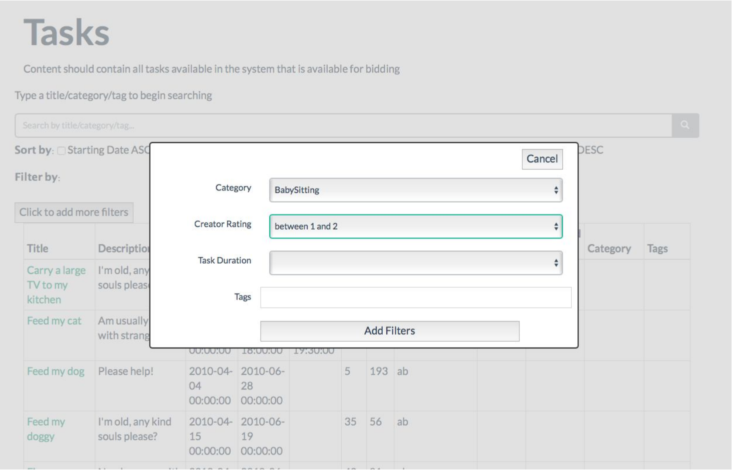
Content should contain all tasks available in the system that is available for bidding

Type a title/category/tag to begin searching

Sort by: □ Starting Date ASC ☑ Start Date DESC □ End Date ASC □ End Date DESC □ Task Duration ASC □ Task Duration DESC

Filter by:

Title	Description	Created at	Start at	End at	Min Bid	Max Bid	Creator	Assignee	Completed at	Category	Tags
Babysitting my sister		2017-09-24 03:09:10	2017-09-27 04:10:14		5	100	abc	ab		BabySitting	#babysit
Fix my house plumbing	Assist me and will reward handsomely.	2010-04-21 00:00:00	2010-06-30 00:00:00		14	188	jamajense5492				
Wash my clothes	Please help!	2010-04-10 00:00:00	2010-06-29 00:00:00		24	100	keafranc1388				
Mend my old shoes pls	Need someone who is responsible and well-trained.	2010-05-14 00:00:00	2010-06-29 00:00:00		24	93	mivincent4274				
Wash my curtains	Assist me and will reward handsomely.	2010-05-09 00:00:00	2010-06-28 20:00:00	2010-06-28 21:00:00	35	436	abc				



Tasks

Content should contain all tasks available in the system that is available for bidding

Type a title/category/tag to begin searching

Sort by: □ Starting Date ASC DESC

Filter by:

Category: BabySitting

Creator Rating: between 1 and 2

Task Duration:

Tags:

Title	Description									Category	Tags
Carry a large TV to my kitchen	I'm old, any souls pleas										
Feed my cat	Am usually with strang	00:00:00	18:00:00	19:30:00							
Feed my dog	Please help!	2010-04-04 00:00:00	2010-06-28 00:00:00		5	193	ab				
Feed my doggy	I'm old, any kind souls please?	2010-04-15 00:00:00	2010-06-19 00:00:00		35	56	ab				

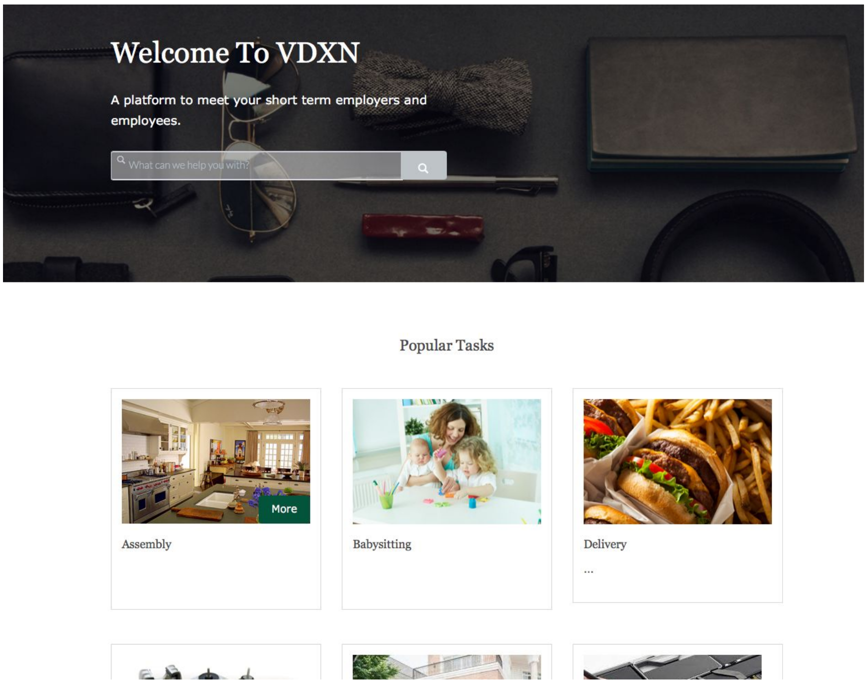
5.4 Tags & Categories

Feature summary & Page Screenshot

Features:

- Create a task with tags
- Create a task with category
- Dynamic search by tags and category
- Search for all tasks within the same category

Page Screenshot



Welcome To VDXN

A platform to meet your short term employers and employees.

SQL Statements

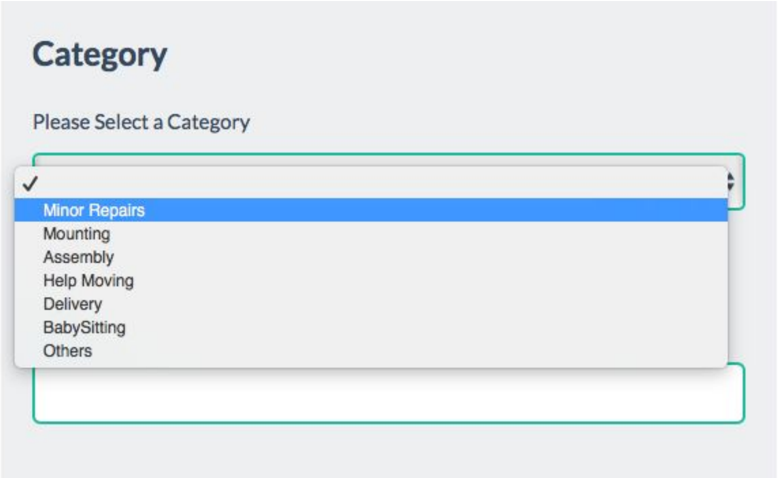
Create Tags

```
INSERT INTO `mini`.`Tag`
(`name`,
`created_at`,
`updated_at`)
VALUES (
'".$tag_name."',
'".$created_at."',
NULL
);
```

Create An Instance of Tag-Task Relation

```
INSERT INTO `mini`.`Tag_task`
(`tag_name`,
`task_creator_username`,
`task_title`,
`created_at`,
`updated_at`)
```

Setting a category when creating a task



Category

Please Select a Category

```
VALUES (
'".$tag_name."',
'".$creator_name."',
'".$task_title."',
'".$created_at."',
NULL
);
```

Create An Instance of Category-Task Relation

```
INSERT INTO `mini`.`Category_task`
(`category_name`,
`task_title`,
`task_creator_username`,
`created_at`,
`updated_at`)
VALUES (
'".$task_category."',
'".$task_title."',
'".$creator_name."',
'".$created_at."',
NULL
);
```

Dynamic Search of Tags

```
SELECT *
FROM Tag t
WHERE t.name LIKE "%' .
$search_string . '%";
```

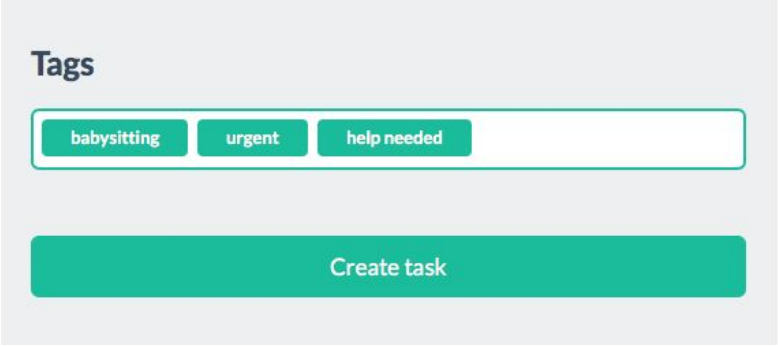
Get Tags of A Task

```
SELECT t.tag_name AS tags
FROM Tag_task t
WHERE t.task_title = '$task_title'
AND t.task_creator_username =
'$creator_name';
```

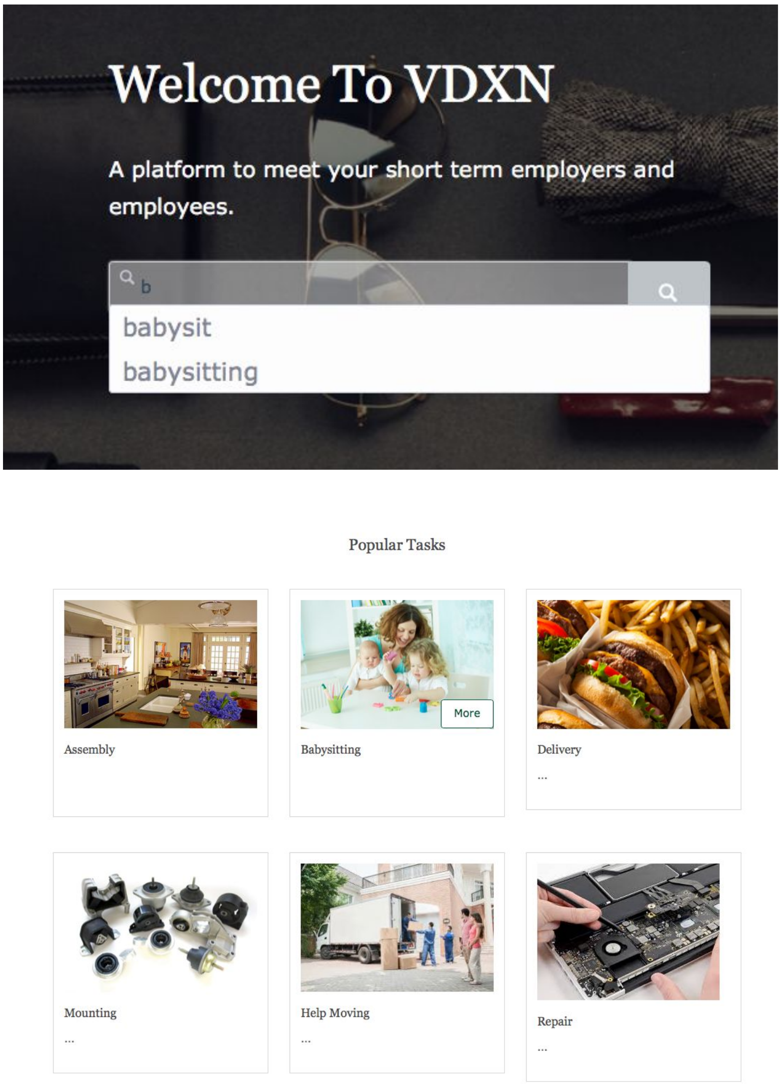
Get Number of Occurrences of Tag_Task Relation

```
SELECT COUNT(*) AS count
FROM Tag
WHERE Tag.name = '$tag_name';
```

Setting a category when creating a task



Searching for a category in the Homepage


Welcome To VDXN

A platform to meet your short term employers and employees.

b

babysit

babysitting

5.5 Task Information & Bidding ("Tasks" page)

Feature summary & Page Screenshot

Features:

For a Task

- View a task's details, e.g. creator username, description, category & tags
- Edit a task (only for verified the task creator and all admins)
- Delete a task (only for verified the task creator and all admins)

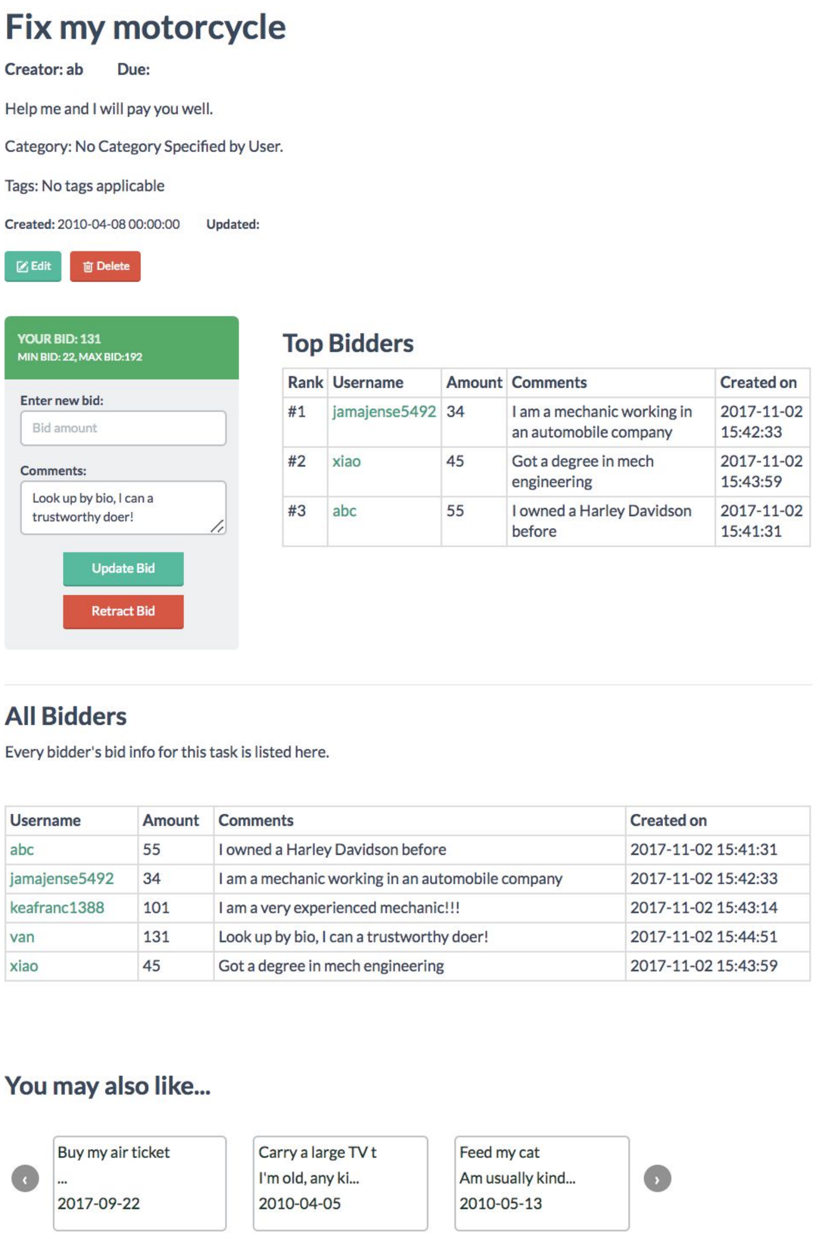
For Bidding

- List the top 3 bidders and their comments
- List all bidders for this task
- Update a bid (only for bidders and admins)
- Retract a bid (only for bidders and admins)

For Task Recommendations

- List all tasks similar to this task in the "You may also like" section

Page Screenshot



Fix my motorcycle

Creator: ab **Due:**

Help me and I will pay you well.

Category: No Category Specified by User.

Tags: No tags applicable

Created: 2010-04-08 00:00:00 **Updated:**

YOUR BID: 131
MIN BID: 22, MAX BID:192

Enter new bid:

Bid amount

Comments:

Look up by bio, I can a trustworthy doer!

Top Bidders

Rank	Username	Amount	Comments	Created on
#1	jamajense5492	34	I am a mechanic working in an automobile company	2017-11-02 15:42:33
#2	xiao	45	Got a degree in mech engineering	2017-11-02 15:43:59
#3	abc	55	I owned a Harley Davidson before	2017-11-02 15:41:31

All Bidders

Every bidder's bid info for this task is listed here.

Username	Amount	Comments	Created on
abc	55	I owned a Harley Davidson before	2017-11-02 15:41:31
jamajense5492	34	I am a mechanic working in an automobile company	2017-11-02 15:42:33
keafranc1388	101	I am a very experienced mechanic!!!	2017-11-02 15:43:14
van	131	Look up by bio, I can a trustworthy doer!	2017-11-02 15:44:51
xiao	45	Got a degree in mech engineering	2017-11-02 15:43:59

You may also like...

Buy my air ticket
...
2017-09-22

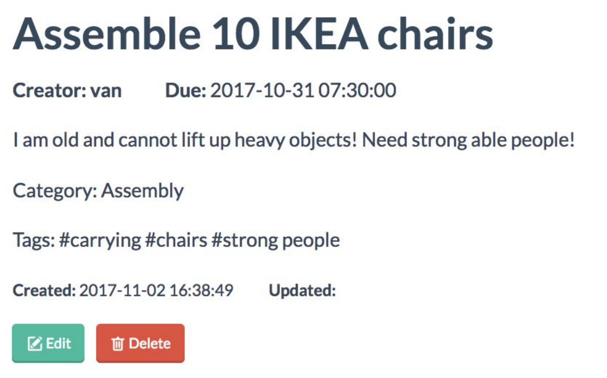
Carry a large TV t
I'm old, any ki...
2010-04-05

Feed my cat
Am usually kind...
2010-05-13

SQL Statements

View a Task and all its attributes

```
SELECT *
FROM Task WHERE
title='$title' AND
creator_username='$creator_username';
```



Assemble 10 IKEA chairs

Creator: van **Due:** 2017-10-31 07:30:00

I am old and cannot lift up heavy objects! Need strong able people!

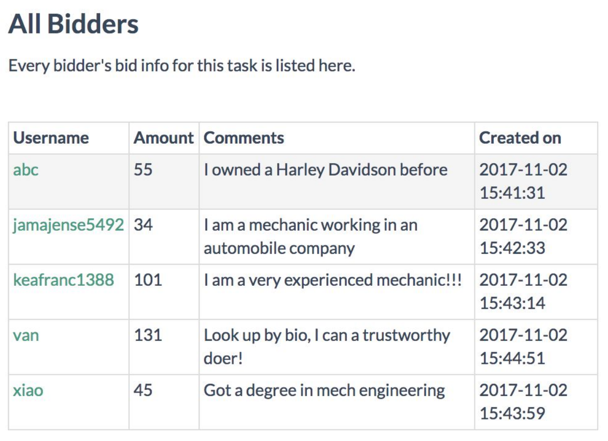
Category: Assembly

Tags: #carrying #chairs #strong people

Created: 2017-11-02 16:38:49 **Updated:**

View all Bids

```
SELECT
  `task_title`,
  `task_creator_username`,
  `bidder_username`,
  `amount`,
  `details`,
  `created_at`,
  `updated_at`
  FROM Bid
  WHERE task_title=$task_title AND
task_creator_username=$task_creator_username;
```



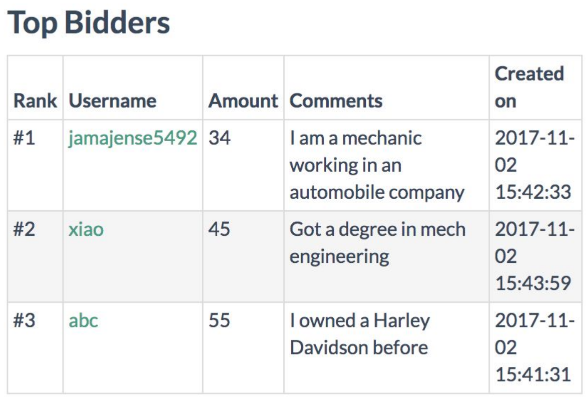
All Bidders

Every bidder's bid info for this task is listed here.

Username	Amount	Comments	Created on
abc	55	I owned a Harley Davidson before	2017-11-02 15:41:31
jamajense5492	34	I am a mechanic working in an automobile company	2017-11-02 15:42:33
keafranc1388	101	I am a very experienced mechanic!!!	2017-11-02 15:43:14
van	131	Look up by bio, I can a trustworthy doer!	2017-11-02 15:44:51
xiao	45	Got a degree in mech engineering	2017-11-02 15:43:59

Get the top N bids

```
-- $topN refers to no. of bidders from the 1st position to
the Nth position, ordered by their bid amounts. N is an
arbitrary integer.

SELECT
  `task_title`,
  `task_creator_username`,
  `bidder_username`,
  `amount`,
  `details`,
  `created_at`,
  `updated_at`
  FROM Bid
  WHERE task_title=$task_title AND
task_creator_username=$task_creator_username
  ORDER BY Bid.amount $orderByAscOrDesc
  LIMIT $topN;
```

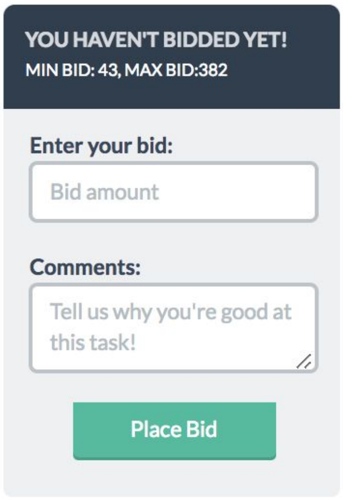


Top Bidders

Rank	Username	Amount	Comments	Created on
#1	jamajense5492	34	I am a mechanic working in an automobile company	2017-11-02 15:42:33
#2	xiao	45	Got a degree in mech engineering	2017-11-02 15:43:59
#3	abc	55	I owned a Harley Davidson before	2017-11-02 15:41:31

Create a Bid

```
INSERT INTO Bid(
  `task_title`,
  `task_creator_username`,
  `bidder_username`,
  `details`,
  `amount`,
  `created_at`,
  `updated_at`)
  VALUES (
    $task_title,
    $task_creator_username,
    $_SESSION['user']->username,
    $details,
    $amount,
    $time,
    NULL);
```



YOU HAVEN'T BIDDED YET!
MIN BID: 43, MAX BID:382

Enter your bid: Bid amount

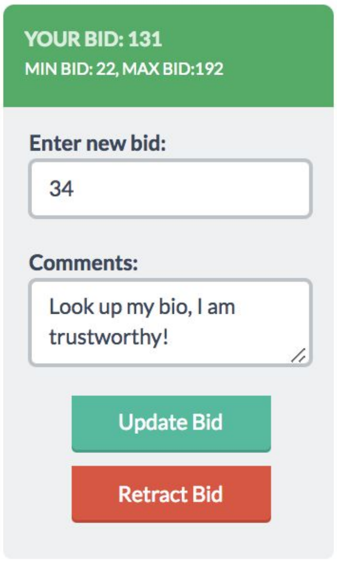
Comments: Tell us why you're good at this task!

Edit a Bid

```
UPDATE Bid
SET
  amount=$amount,
  details=$details,
  updated_at=$time,
  WHERE task_title=$task_title AND
task_creator_username=$task_creator_username AND
bidder_username=$bidder_username;
```

Delete a Bid

```
DELETE FROM Bid
WHERE task_title=$task_title AND
task_creator_username=$task_creator_username AND
bidder_username=$bidder_username;
```



YOUR BID: 131
MIN BID: 22, MAX BID:192

Enter new bid: 34

Comments: Look up my bio, I am trustworthy!

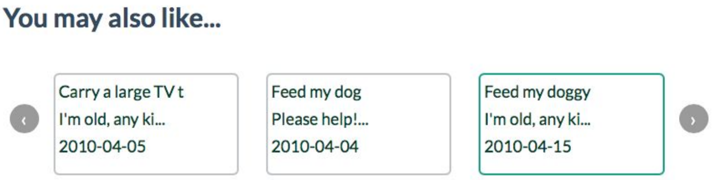
5.6 Task Recommendations ("Home" page & "Tasks" page)

Feature summary & Page Screenshot

Features:

- Similar Task Recommendation
  - Decision criteria based on other tasks with same tag, same category, or created by the same user.

Page Screenshot



You may also like...

Carry a large TV t I'm old, any ki... 2010-04-05	Feed my dog Please help!... 2010-04-04	Feed my doggy I'm old, any ki... 2010-04-15

SQL Statements

Find Category of A Task

```
SELECT g.category_name AS category
FROM Category_task g
WHERE g.task_title = $task_title AND g.task_creator_username = $creator_name ;
```

Find Tags of A Task

```
SELECT t.tag_name AS tags
FROM Tag_task t
WHERE t.task_title = $task_title AND t.task_creator_username = $creator_name ;
```

Find Similar Tasks

```
SELECT t.title, t.description, t.created_at, t.updated_at, t.start_at, t.end_at,
t.min_bid, t.max_bid, t.creator_username, t.assignee_username, t.completed_at,
t.remarks, TIMESTAMPDIFF(SECOND, t.end_at, t.start_at) AS duration
  FROM (Task t LEFT JOIN Tag_task g ON t.title = g.task_title AND t.creator_username =
        g.task_creator_username) LEFT JOIN Category_task c ON t.title = c.task_title AND
        t.creator_username = c.task_creator_username
  WHERE c.category_name = $category AND t.title <> $task->{'title'} AND t.creator_username
<> $task->{'creator_username'};

SELECT t.title, t.description, t.created_at, t.updated_at, t.start_at, t.end_at,
t.min_bid, t.max_bid, t.creator_username, t.assignee_username, t.completed_at,
t.remarks, TIMESTAMPDIFF(SECOND, t.end_at, t.start_at) AS duration
  FROM (Task t LEFT JOIN Tag_task g ON t.title = g.task_title AND t.creator_username =
g.task_creator_username)LEFT JOIN Category_task c ON t.title = c.task_title AND
t.creator_username = c.task_creator_username
  WHERE $tags_query AND t.title <> $task->{'title'} AND t.creator_username <>
$task->{'creator_username'};

SELECT *
FROM Task t
WHERE t.title <> $task_name AND t.creator_username = $creator_name ;
```

5.7 Admin Data Analysis (“System Stats” page)

Feature summary & Page Screenshot

Features:

- Specify a date range for all SQL queries

User Stats

- Number of user sign-ups
- Number of users who bidded for at least 1 task
- Get all users who never bidded

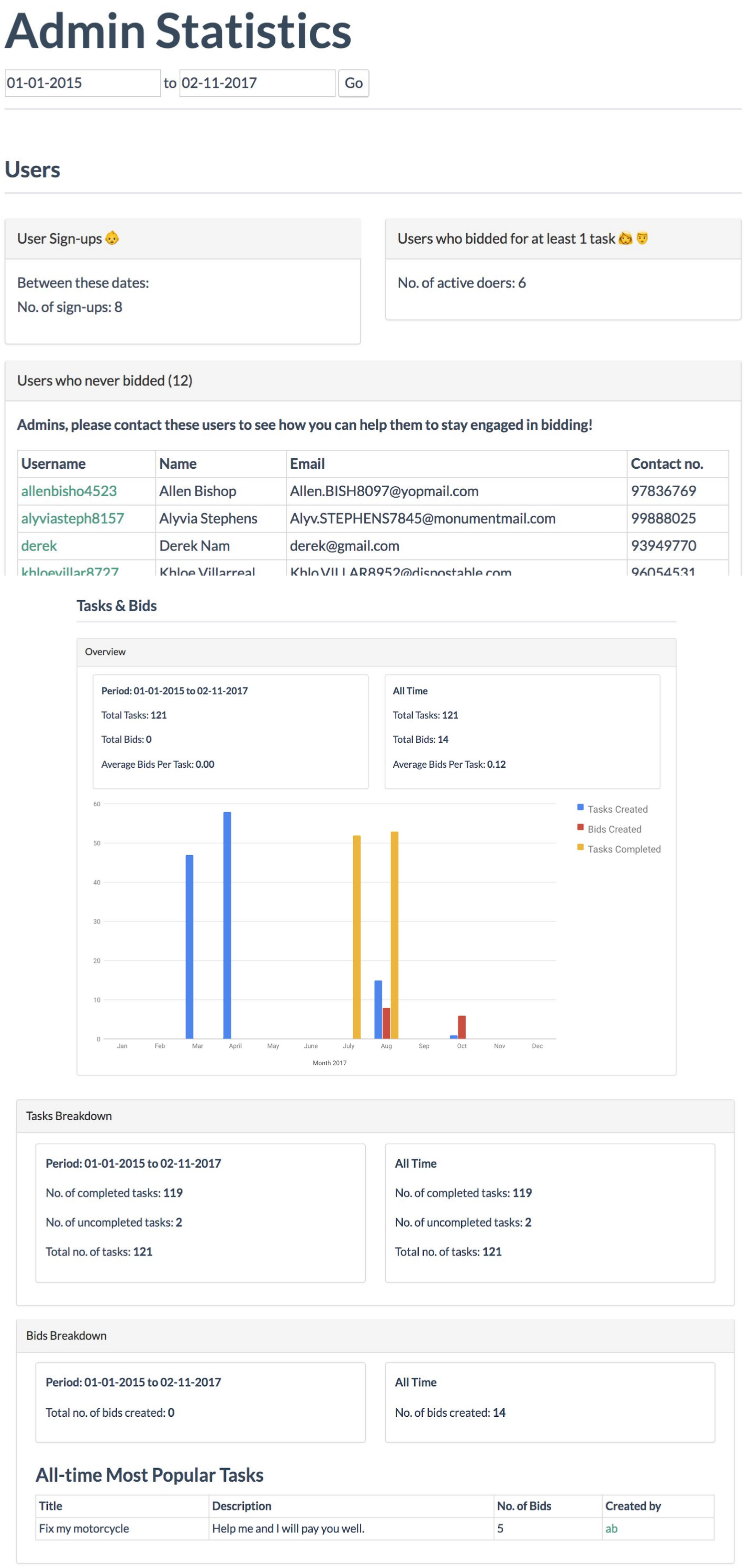
Task Stats

- No. of tasks in the system
- Average no. of bids per task
- No. of tasks created per month as a bar chart
- No. of bids created per month as a bar chart
- No. of tasks completed per month as a bar chart

Bidding Stats

- No. of bids created
- List of all-time most popular tasks (largest number of bids)

Page Screenshot



Admin Statistics

01-01-2015 to 02-11-2017 Go

Users

User Sign-ups 👶

Between these dates:

No. of sign-ups: 8

Users who bidded for at least 1 task 👩 👨

No. of active doers: 6

Users who never bidded (12)

Admins, please contact these users to see how you can help them to stay engaged in bidding!

Username	Name	Email	Contact no.
allenbisho4523	Allen Bishop	Allen.BISH8097@yopmail.com	97836769
alyviasteph8157	Alyvia Stephens	Alyv.STEPHENS7845@monumentmail.com	99888025
derek	Derek Nam	derek@gmail.com	93949770
khloevillar8727	Khloe Villarreal	Khlo.VILLAR8952@dispostable.com	96054531

Tasks & Bids

Overview

Period: 01-01-2015 to 02-11-2017

Total Tasks: **121**

Total Bids: **0**

Average Bids Per Task: **0.00**

All Time

Total Tasks: **121**

Total Bids: **14**

Average Bids Per Task: **0.12**

Tasks Breakdown

Period: 01-01-2015 to 02-11-2017

No. of completed tasks: **119**

No. of uncompleted tasks: **2**

Total no. of tasks: **121**

All Time

No. of completed tasks: **119**

No. of uncompleted tasks: **2**

Total no. of tasks: **121**

Bids Breakdown

Period: 01-01-2015 to 02-11-2017

Total no. of bids created: **0**

All Time

No. of bids created: **14**

All-time Most Popular Tasks

Title	Description	No. of Bids	Created by
Fix my motorcycle	Help me and I will pay you well.	5	ab

SQL Statements

Count no. of user sign-ups between dates

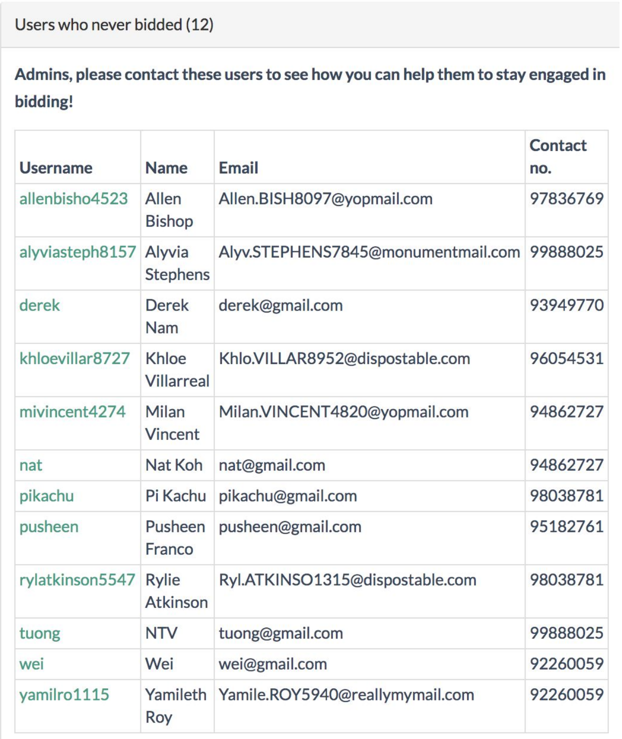
```
SELECT COUNT(*) AS num_users
  FROM User
  WHERE user_type <> 'Admin' AND
  created_at BETWEEN
  $from_date AND $to_date;
```

Count no. of users with at least 1 bid

```
SELECT COUNT(*) AS num_users FROM (
  SELECT b.bidder_username, COUNT(*)
  AS num_bids
  FROM Bid b
  GROUP BY b.bidder_username
  HAVING COUNT(*) >= 1
  ORDER BY COUNT(*) DESC
) AS NumBidsPerUserTable;
```

Get all users who never bidded

```
SELECT u.username, u.first_name,
u.last_name, u.email, u.contact
  FROM User u
  WHERE NOT EXISTS (
    SELECT b.bidder_username
    FROM Bid b
    WHERE u.username = b.bidder_username
    GROUP BY b.bidder_username
    HAVING COUNT(*) >= 1
);
```



Users who never bidded (12)

Admins, please contact these users to see how you can help them to stay engaged in bidding!

Username	Name	Email	Contact no.
allenbisho4523	Allen Bishop	Allen.BISH8097@yopmail.com	97836769
alyviasteph8157	Alyvia Stephens	Alyv.STEPHENS7845@monumentmail.com	99888025
derek	Derek Nam	derek@gmail.com	93949770
khloevillar8727	Khloe Villarreal	Khlo.VILLAR8952@dispostable.com	96054531
mivincent4274	Milan Vincent	Milan.VINCENT4820@yopmail.com	94862727
nat	Nat Koh	nat@gmail.com	94862727
pikachu	Pi Kachu	pikachu@gmail.com	98038781
pusheen	Pusheen Franco	pusheen@gmail.com	95182761
rylatkinson5547	Rylie Atkinson	Ryl.ATKINSO1315@dispostable.com	98038781
tuong	NTV	tuong@gmail.com	99888025
wei	Wei	wei@gmail.com	92260059
yamilro1115	Yamileth Roy	Yamile.ROY5940@reallymymail.com	92260059

Get no. of tasks between dates

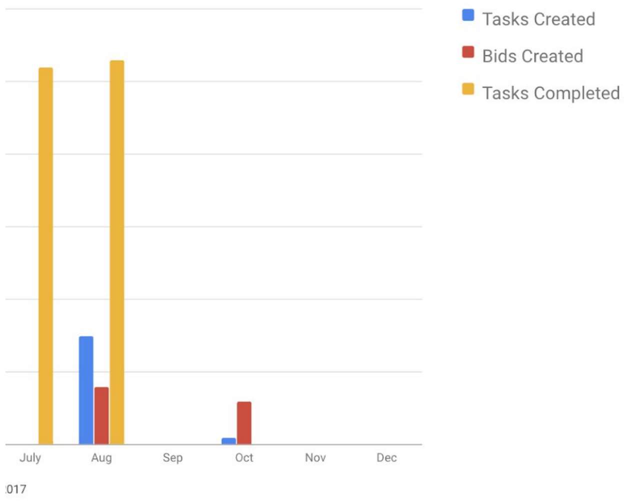
```
SELECT COUNT(*) AS num_tasks
  FROM Task
  WHERE created_at
  BETWEEN $from_date AND $to_date;
```

Get no. of bids between dates

```
SELECT COUNT(*) AS num_bids
  FROM Bid
  WHERE created_at
    BETWEEN $from_date AND $to_date;
```

Get no. of tasks created between dates grouped by month

```
DROP VIEW IF EXISTS
count_tasks_created_by_month;

CREATE VIEW count_tasks_created_by_month AS
SELECT MONTH(t.created_at) AS month,
COUNT(*) AS num_tasks_created
FROM Task t
GROUP BY MONTH(t.created_at)
ORDER BY MONTH(t.created_at) ASC;

SELECT m.value, c.num_tasks_created
FROM Months m LEFT JOIN
count_tasks_created_by_month c
ON m.value = c.month
GROUP BY m.value
ORDER BY CAST(m.value AS UNSIGNED) ASC;
```



Get no. of bids created between dates grouped by month

```
DROP VIEW IF EXISTS
count_bids_created_by_month;

CREATE VIEW count_bids_created_by_month AS
SELECT MONTH(b.created_at) AS month,
COUNT(*) AS num_bids_created
FROM Bid b
GROUP BY MONTH(b.created_at)
ORDER BY MONTH(b.created_at) ASC;

SELECT m.value, b.num_bids_created
FROM Months m LEFT JOIN
count_bids_created_by_month b
ON m.value = b.month
GROUP BY m.value
ORDER BY CAST(m.value AS UNSIGNED) ASC;
```

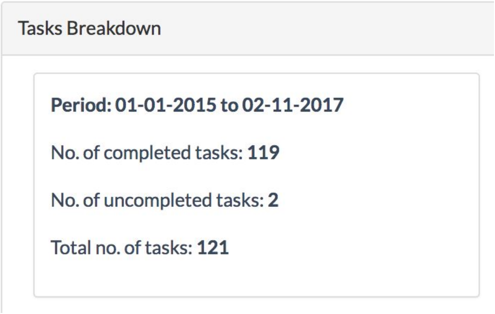
Get no. of tasks completed between dates grouped by month

```
DROP VIEW IF EXISTS
count_tasks_completed_by_month;

CREATE VIEW count_tasks_completed_by_month
AS
SELECT MONTH(t.completed_at) AS month,
COUNT(*) AS num_tasks_completed
FROM Task t
GROUP BY MONTH(t.completed_at)
ORDER BY MONTH(t.completed_at) ASC;

SELECT m.value, t.num_tasks_completed
```

```
FROM Months m LEFT JOIN
count_tasks_completed_by_month t
ON m.value = t.month
GROUP BY m.value
ORDER BY CAST(m.value AS UNSIGNED) ASC;
```

Get no. of completed and uncompleted tasks between dates

```
SELECT COUNT(completed_at)
AS num_tasks_completed,
  (COUNT(*) - COUNT(completed_at)) AS
num_tasks_uncompleted
FROM Task
WHERE completed_at
  BETWEEN $from_date AND $to_date;
```



Tasks Breakdown

Period: 01-01-2015 to 02-11-2017

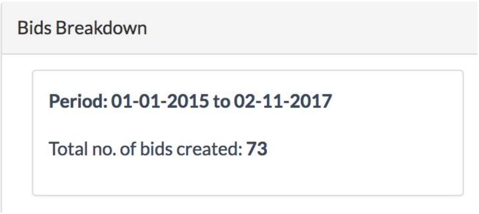
No. of completed tasks: **119**

No. of uncompleted tasks: **2**

Total no. of tasks: **121**

Count no. of bids between dates

```
SELECT COUNT(*) AS num_bids
  FROM Bid
  WHERE created_at
  BETWEEN $from_date AND $to_date;
```



Bids Breakdown

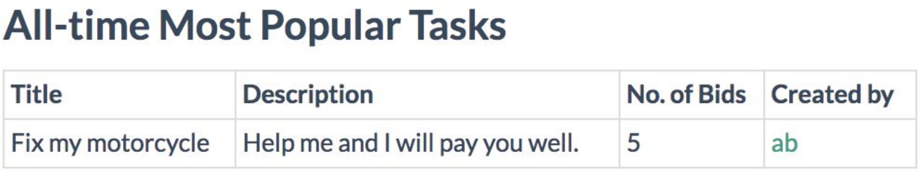
Period: 01-01-2015 to 02-11-2017

Total no. of bids created: **73**

Get all-time most popular tasks

```
SELECT
  t.title AS title, t.description AS description, t.creator_username AS username, COUNT(*)
AS num_bids
  FROM Task t, Bid b
    WHERE t.title = b.task_title
    AND t.creator_username = b.task_creator_username
    GROUP BY
      t.title,
      t.creator_username
      HAVING COUNT(*) >= ALL (
        SELECT COUNT(*)
        FROM Task t1, Bid b1
        WHERE
          t1.title = b1.task_title AND
          t1.creator_username = b1.task_creator_username
          GROUP BY t1.title, t1.creator_username);
```

Screenshot of above query given below:



All-time Most Popular Tasks

Title	Description	No. of Bids	Created by
Fix my motorcycle	Help me and I will pay you well.	5	ab

- END OF REPORT-