
Wenzhou University

Lost & Found Management System

Major: Computer Science & Artificial Intelligence

Name: Franck Davy Tchienkoua

Student Number: 19511160012

Due Date: 21/10/2021

Abstract:

Universities are busy places. There is a lot of coming and going. People are moving from one class to the next. They also frequent the hangouts between classes and are all over the campus. When people are this busy some of their personal items may get lost. A person may take their coat off and forget it. Students lose their briefcases and bags all of the time. There can be some personal information or items of sentimental value that accidentally get left behind. Many universities have lost and found boxes but people may not know where they are located. Lost and found software make it easy to post and manage items that have been lost or found and turned to the university.

Introduction

The proposed project “Lost & Found Management System” has been developed to overcome the absence of any specific channel to gather the lost and found information on campus. This software is built to serve the teachers and students get around these obstacles.

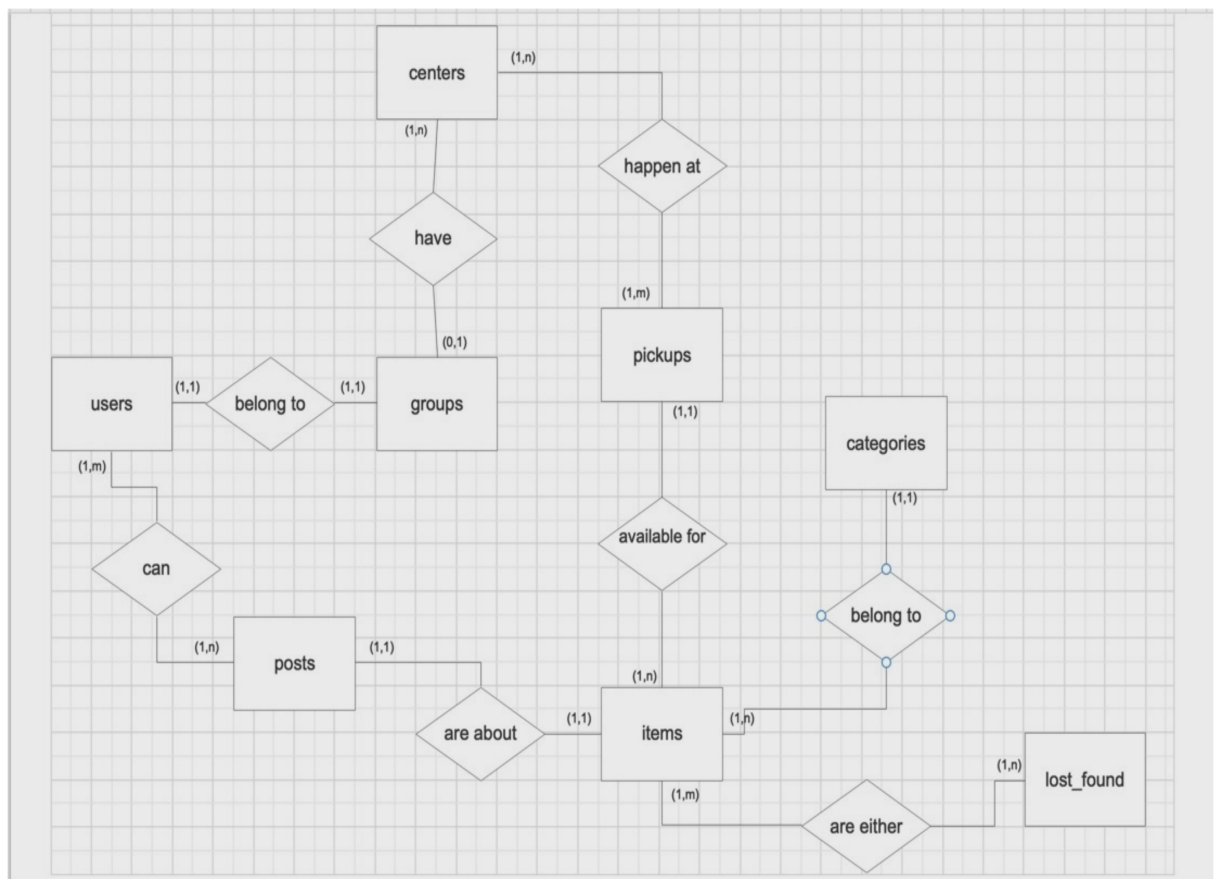
This application is built with a privacy-first approach and user-friendliness to eliminate any tasks requiring more knowledge than the bare minimum. It uses the schemas and views available in PostgreSQL to achieve such.

The software by design, uses a connection to the university student information system in order to let students post without the need of registering. They can use already existing school accounts. A unique ID and additional information are matched with each claim. This effectively reduce the chance of the wrong person picking up the lost item. Everything that is found will be posted with the location that the item was found and the location on the campus where it is being kept. This program will increase the rates at which the items are returned to their owner. The data will be constantly updated. When an item is returned to its owner, the system it is updated and the post is taken down.

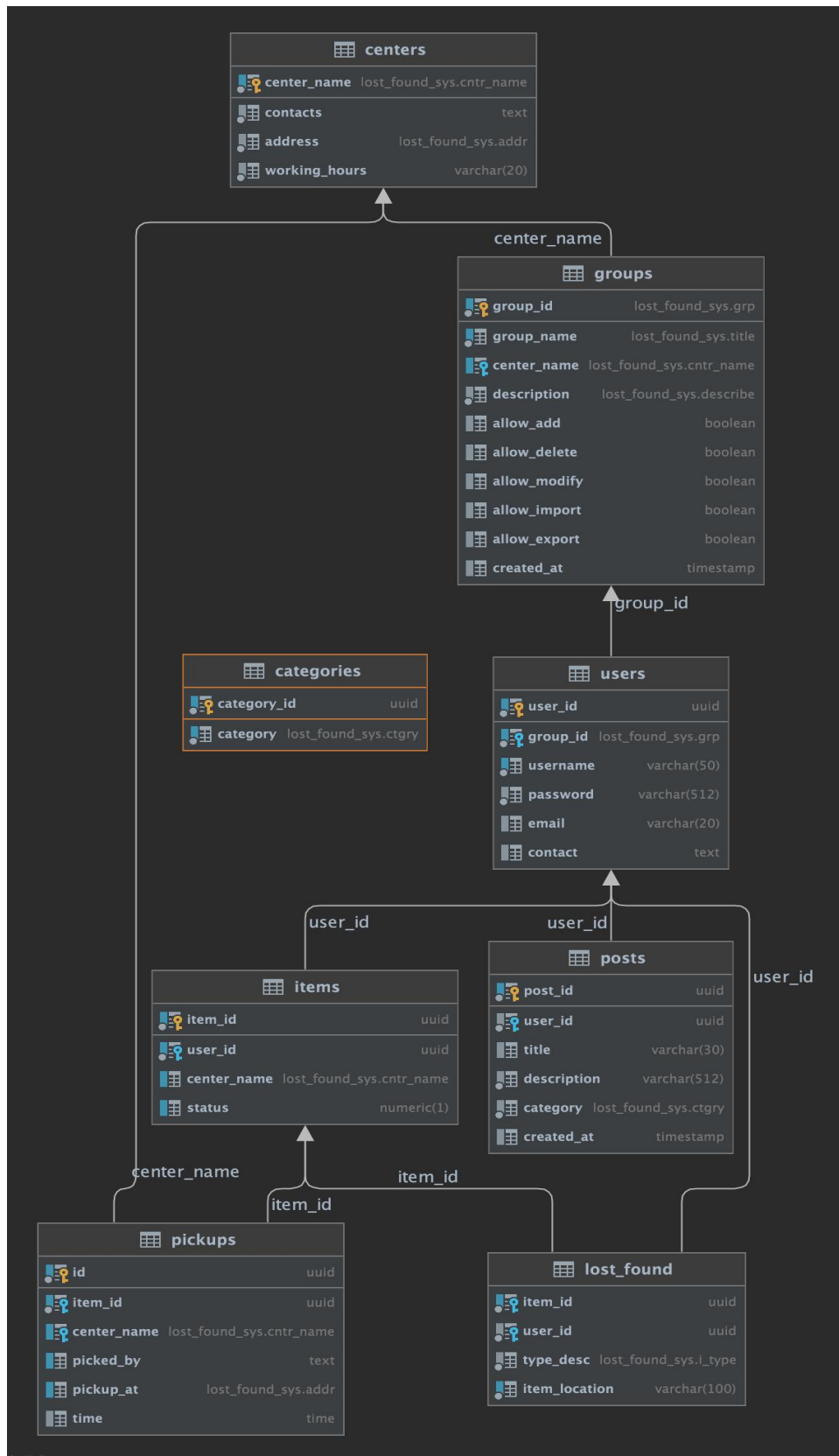
An administrator oversees the entire system and is responsible for adding each center manager. Every center is then given specific privileges. They can all communicate with each other, but a center isn't allowed to view data owned by another unless given the right privileges by the administrator. This approach is taken to maintain privacy and allow the administrator to better manage privileges for each center, and the entire system.

DATABASE DESIGN

ERD Diagram



Relational Schema



Normalization

Database normalization is a technique of organizing the data in the database. Normalization is a formal approach that applies a set of rules to associate attributes with entities. Normalization is used when designing a database. This database has been normalized to the 3rd Normal Form.

1NF: No Repeating Groups

The first form sets the very basic rules for our database:

- We eliminate any duplicate columns from the same table
- We create separate tables for each set of related data
- We identify each row with a unique column (the primary key)

2NF: Elimination of Redundant Data

In this form we further address the concept of removing duplicate data:

- We have met all the requirements of the first normal form
- We remove subsets of data that apply to multiple rows of a table and place them in separate tables
- We create relationships between these new tables and their predecessors through the use of foreign keys

3NF: Transitive Dependency Elimination

- We have met all the requirements of the second normal form
- We ensure that there is no transitive functional dependency. A better way to phrase this rule may be that all parts of the primary key must depend on the characteristics of an entity form.

Optimization

Database optimization refers to a variety of strategies for reducing database system response time. Databases can store tremendous amounts of information. Specific bits of data are accessed by queries written in a particular interface language, such as SQL (we use PostgreSQL in our case). Database optimization involves maximizing the speed and efficiency with which data is retrieved.

Indexes

An index is used to retrieve data in clusters of the database.

```

157  --- INDEXES (QUERY OPTIMIZATION)
158  --- =====
159  --- indexes on groups table
160  DROP INDEX IF EXISTS idx_groups_id, idx_groups_id_name;
161  CREATE INDEX idx_groups_id ON lost_found_sys.groups(group_id);
162  CREATE INDEX idx_groups_id_name ON lost_found_sys.groups(group_id, group_name);
163  ---
164  --- indexes on centers table
165  DROP INDEX IF EXISTS idx_centers_name;
166  CREATE INDEX idx_centers_name ON lost_found_sys.centers(center_name);
167  ---
168  --- indexes on users table
169  DROP INDEX IF EXISTS idx_users_id, idx_users_ctid, idx_users_name;
170  CREATE INDEX idx_users_id ON lost_found_sys.users(user_id);
171  CREATE INDEX idx_users_ctid ON lost_found_sys.users(group_id);
172  CREATE INDEX idx_users_name ON lost_found_sys.users(username);
173  ---
174  --- indexes on logs table
175  DROP INDEX IF EXISTS idx_logs_lid, idx_logs_gid, idx_logs_uid;
176  CREATE INDEX idx_logs_lid ON lost_found_sys.logs(log_id);
177  CREATE INDEX idx_logs_gid ON lost_found_sys.logs(group_id);
178  CREATE INDEX idx_logs_uid ON lost_found_sys.logs(user_id);
179  ---
180  --- indexes on categories table
181  DROP INDEX IF EXISTS idx_categories_cat, idx_categories_catid;
182  CREATE INDEX idx_categories_cat ON lost_found_sys.categories(category);
183  CREATE INDEX idx_categories_catid ON lost_found_sys.categories(category_id, category);
184  ---
185  --- indexes on items table
186  DROP INDEX IF EXISTS idx_items_id, idx_items_uid, idx_items_stat, idx_items_ustat;
187  CREATE INDEX idx_items_id ON lost_found_sys.items(item_id);
188  CREATE INDEX idx_items_uid ON lost_found_sys.items(user_id);
189  CREATE INDEX idx_items_stat ON lost_found_sys.items(status);
190  CREATE INDEX idx_items_ustat ON lost_found_sys.items(user_id, status) ;
191  ---
192  --- indexes on lost_found table
193  DROP INDEX IF EXISTS idx_lost_found_id, idx_lost_found_loc;
194  CREATE INDEX idx_lost_found_id ON lost_found_sys.lost_found(item_id);
195  CREATE INDEX idx_lost_found_loc ON lost_found_sys.lost_found(item_location);
196  ---
197  --- indexes on pickups table
198  DROP INDEX IF EXISTS idx_pickups_id, idx_pickups_ctid, idx_pickups_by, idx_pickups_at, idx_pickups_byat;
199  CREATE INDEX idx_pickups_id ON lost_found_sys.pickups(item_id);
200  CREATE INDEX idx_pickups_ctid ON lost_found_sys.pickups(center_name);
201  CREATE INDEX idx_pickups_by ON lost_found_sys.pickups(picked_by);
202  CREATE INDEX idx_pickups_at ON lost_found_sys.pickups(pickup_at);
203  CREATE INDEX idx_pickups_byat ON lost_found_sys.pickups(picked_by, pickup_at);

```

FEASIBILITY STUDY

After identifying the scope of the project, the feasibility study is needed to be carried out. It is basically keeping the following points in mind.

Building the software for meeting the scope of the project

This software meets the scope of the project. As it successfully solves the problem at hand, and uses quite simple processing for that.

Technically feasible

The database specifications will successfully satisfy almost every admin's requirements. It uses the most modern and used techniques in the industry for ease of use, and large support in case a problem arise.

Functions of the Database :

T-SQL Commands

After the database and all its tables are set, we begin by creating the different centers, groups the users belong to (This group table serve to differentiate between the administrator, center managers and users of the software), users and categories in which the items fall for better accuracy.

Inserting a center

```
INSERT INTO lost_found_sys.centers VALUES ('ZHOU',  
'(463)–056554', 'North Campus', '8:20 AM to 4:40 PM');
```

Inserting a group

```
INSERT INTO lost_found_sys.groups VALUES ('grp_003@stu',  
'student', NULL, 'can only post lost or found stuff');
```

Inserting a new user

```
INSERT INTO lost_found_sys.users VALUES (uuid_generate_v4(),  
'grp_003@stu', 'uglydavy', 'password1', 'uglydavy@mail.com',  
'(000)–00000000');
```

Inserting a category

```
INSERT INTO lost_found_sys.categories VALUES  
(uuid_generate_v4(), 'category1');
```


Now that the which is considered the core part of our database is set, a user can now post about items they have misplaced or found. This process is comprised as a procedure which takes certain information, all related to the the item whether lost or found and put them into different tables such as the posts table to identify an individual's post; The items table keeping track of each individual items posted on the system; The lost_found which serves to differentiate between lost and found items; And lastly, the pickups table which keeps track of items status at a specified center (Whether items have already been fetched or not).

Inserting into posts, items, lost_found and pickups table by calling a stored procedure

1) Calling the stored procedure:

```
call lost_found_sys.usp_posts ('uglydavy', 'Lost Airpods 2', 'I  
lost my airpods yesterday around the business center.',  
'category1', found, 'ZHOU', 'Business Center');
```

2) Inside the stored procedure:

```
INSERT INTO lost_found_sys.posts VALUES (postID, userID,  
usp_title, usp_desc, usp_category);  
  
INSERT INTO lost_found_sys.items VALUES (itemID, userID,  
usp_center_name);  
  
INSERT INTO lost_found_sys.lost_found VALUES (itemID, userID,  
usp_type_desc, usp_location);  
  
INSERT INTO lost_found_sys.pickups VALUES (pickupID, itemID,  
usp_center_name);
```

After items have been posted to be lost or found, we expect the relevant parties to claim them at their specified storing location. This process is also carried out by a specific stored procedure. After the item is collected, the pickups table is updated with the name of the individual who picked up the item, and the specific time. The items table status is also updated to collected and the lost or found object post is taken down.

Picking up the lost or found item and updating its status

1) Calling the procedure:

```
call lost_found_sys.usp_pickups  
( 'c7aa0b71-ff52-46c2-ae24-4cdd35b6e661', 'Michael', 'North  
Campus' );
```

2) Functions inside the procedure:

```
SELECT id INTO STRICT _ID FROM lost_found_sys.pickups  
WHERE item_id = itemID;  
  
UPDATE lost_found_sys.pickups SET picked_by = pickedBY,  
pickup_at = pickedAT, time = pickedTIME WHERE id = _ID;  
  
UPDATE lost_found_sys.items SET status = 1 WHERE item_id  
= itemID;
```

Conclusion

In conclusion, a database is a far more efficient mechanism to store and organize data than spreadsheets; it allows for a centralized facility that can easily be modified and quickly shared among multiple users.

This project is built keeping in mind privacy and that it is going to be use by several parties, and at different levels (admin, centers, students..). It is designed to be easily extendable, according the the administrator needs. The main objective of this framework is to save time, make the system cost effective and efficient records management.