

University of Buea

Faculty of Engineering and
Technology

Department of Computer
Engineering



Universite de Buea

Faculte d'ingenieurie et de
Technologie

Genie Informatique

CEF 440: INTERNET AND MOBILE PROGRAMMING

Database Design and Implementation of ReUseEats

- TENDONGFAC GEORGE NJIMO FE20A116
- DONGHO NONGNI GUIROLE FRANCIS FE20A230
- ARVELLA CHRISTIE PETTOUN TCHOUAKWE
FE20A011
- MBUNGAI MICHAEL BERNARD FOMO FE20A064
- NGUEMGNE FOTSO VINY ANGELE FE20A078

Supervisor:

Dr. NKEMENI Valery

TABLE OF CONTENTS

Table of Contents

INTRODUCTION	3
1. Database Design.....	3
2. Implementing the database.....	5

INTRODUCTION

Database design refers to the organization of data according to a database model and the designer determines what data must be stored and how the data elements interrelate. with this information, data from our users will be fitted to our database model. Firebase will be used for our ReUseEats project due to its ability to sync data between users in real time and a cloud hosted database.

1. Database Design

In this phase, Entity relationship diagram (ER) was extracted from the class diagram. An ER diagram is one which contain entities that depicts real life situations with their different relationships and cardinalities. In the ER diagram the primary keys, foreign keys and other constrains were laid emphasis on as we designed the database for our system.

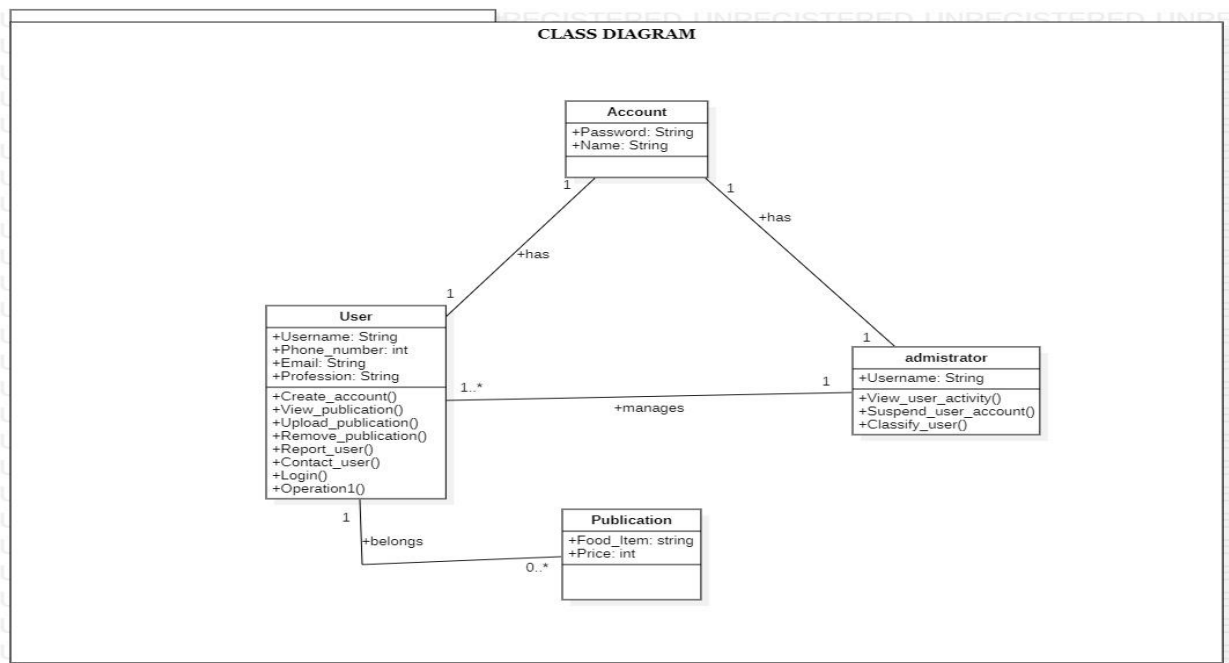


Fig 1. Class diagram

Our ER diagram was developed with the use of the class diagram in Fig 1.

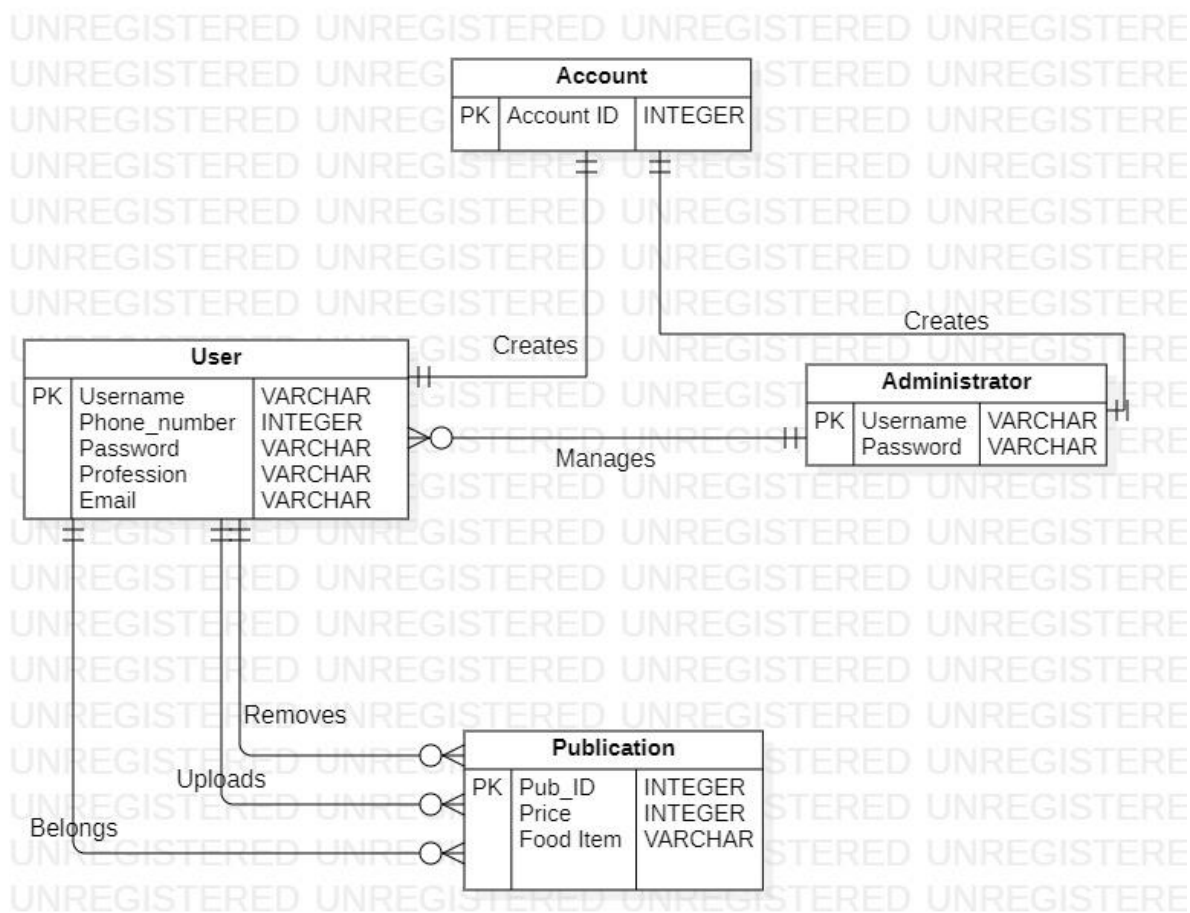


Fig 2. Entity Relational (ER) Diagram

With the help of the ER diagram, we proceeded to creating the relational schema containing the various tables with their primary keys, foreign keys as well as their various attributes and datatypes as shown below.

Account (AccountID, UserID*)

Administrator (Username, password, AccountID*)

User (user_name, password, Phone_number, Password, Profession, Email, Username*)

Upload (pub_id, Price, food_item, user_name*)

Removes (pub_id, price, food_item, user_name*)

2. Implementing the database

This involves creating the tables and the columns in the database and the necessary SQL codes to interact with the database. Firstly, we begin with the creation of database and proceed to the tables and the columns. Here below are the various tables with attributes;

To begin with, we have the account table containing an instance of a unique key value as seen below.

Account > ZsAnBUK0YqN2.

More in Google Cloud

reuseeats

Account

ZsAnBUK0YqN2jep0Y7L3

Start collection

Add document

Start collection

Account

ZsAnBUK0YqN2jep0Y7L3

Add field

Upload

Users

administrator

remove

mike

Also, there is the administrator and user tables created with unique values such as user_name as seen below.

Administrator;

User table;

The screenshot shows the Google Cloud Firestore console. The breadcrumb navigation is 'Home > reuseeats > remove > remove1'. The left sidebar shows a tree view with 'reuseeats' expanded, containing 'Account', 'Upload', 'Users', 'administrator', and 'remove'. The 'remove' collection is selected. The main area shows the 'remove1' document with fields: 'food_item' (value: 'eru'), 'price' (value: 1399), and 'pubID' (value: 100).

Collection	Document	Fields
remove	remove1	<ul style="list-style-type: none">food_item: "eru"price: 1399pubID: 100

The publication table is as follow which handles the various posts made by the various users of the system.

Publication;

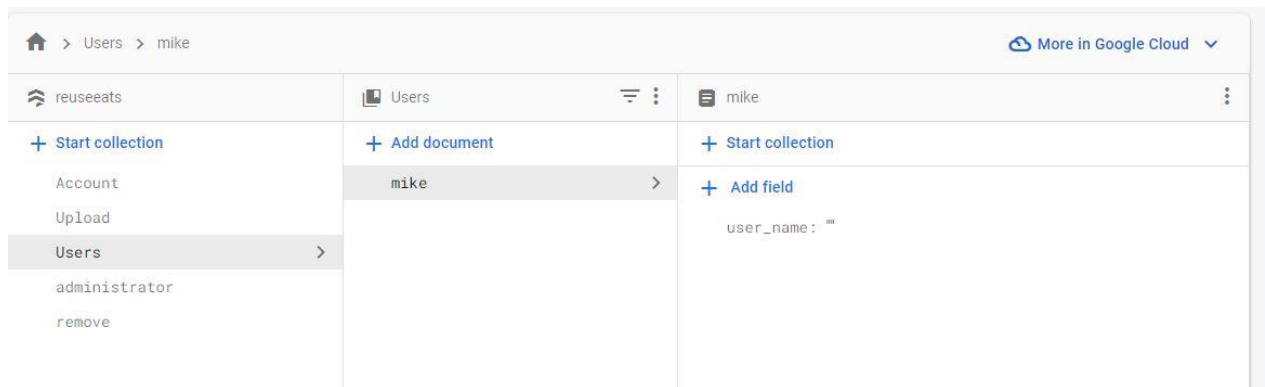
The screenshot shows the Google Cloud Firestore console. The breadcrumb navigation is 'Home > reuseeats > publication > pubID'. The left sidebar shows a tree view with 'reuseeats' expanded, containing 'Account', 'Upload', 'Users', 'administrator', 'publication', and 'remove'. The 'publication' collection is selected. The main area shows the 'pubID' document with fields: 'food_item' (value: 'fufu'), 'price' (value: 2112, type: number), and 'pubID' (value: 1).

Collection	Document	Fields
publication	pubID	<ul style="list-style-type: none">food_item: "fufu"price: 2112 (number)pubID: 1

Furthermore, the tables below are derived as a result of the relationship between publication and users that is upload and remove tables.

The screenshot shows the Google Cloud Firestore console. The breadcrumb navigation is 'Home > Upload > up1'. The left sidebar shows a tree view with 'reuseeats' expanded, containing 'Account', 'Upload', 'Users', 'administrator', and 'remove'. The 'Upload' collection is selected. The main area shows the 'up1' document with fields: 'food_item' (value: 'fufu'), 'price' (value: 1000), 'pubID' (value: 1), and 'user_name' (value: 'mike', type: string).

Collection	Document	Fields
Upload	up1	<ul style="list-style-type: none">food_item: "fufu"price: 1000pubID: 1user_name: "mike" (string)



For our database architecture, a distributed database approach will be used. Since the system will deal with a huge number of users, data access time using a distributed approach will be considerably shorter because the data is retrieved from the nearest database file.

Also, in case of faultiness in one of the databases users can still access the other database files.

Using the steps mentioned above, from extracting the ER model from the class diagram, developing the relational schema and proceeding to the implementation of the schema using sql code in firebase, our database is up and functional.