

**Gurabo Campus**

**Mini Facebook Project**

Professor: Idalides Vergara

Student: Genesis Resto

Student: Carlos Roque

**Database Schema:**

Graphical user interface

Description automatically generated

**Data Dictionary:**

**Graphical user interface, application

Description automatically generatedTable UserInformation:** It is used to store information about Mini-Facebook users. The UserInformation table has a one-to-many relationship with the RoleForUser, AddressInformation and PicturesForUser tables.

*UserName:* The username, which one chooses while joining the Mini-Facebook, is a name and is defined as a text string (VARCHAR) that can contain up to 20 characters. The UserInformation table's primary key, UserName, also enables each user to have a distinct username.

*PasswordHash:* This is a medium-sized text that is specified as the hash value of the user's password. (MEDIUMTEXT). Passwords for users are kept in this field.

*Name:* This is the complete name of the user, and it can contain up to 20 characters. This field is required and cannot be left blank.

*Telephone:* A text string with up to 20 characters that represents the user's phone number. This field must be filled in and cannot be left empty.

*dob*: is a text string with a maximum of 20 characters that represents the user's date of birth. This field must be filled in and cannot be left empty.

*gender:* A text string with up to 10 characters that identifies the user's gender. This field must be filled in and cannot be left empty.

*email:* Refers to the user's email address and is a text string with a maximum length of 100 characters. This field must be filled in and cannot be left empty.

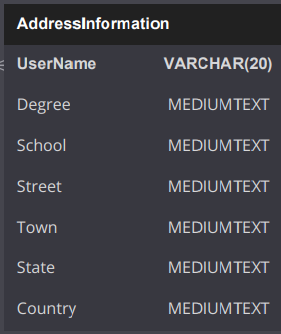
**Table picturesForUser:** The picturesForUser table is used to store the pictures associated with Mini-Facebook users. This table has a many-to-one relationship with the UserInformation table.

Graphical user interface, text, application

Description automatically generated

*PicturePath*: Defined as a mediumtext, this is the path to the image in the file system. This field is required and acts as the table's main key to guarantee that each image has a distinct route.

*UserName*: This is a text string with a maximum of 50 characters that serves as the user name linked to the image. The UserName column in the UserInformation table is referenced by this field, which is required and functions as a foreign key. This makes it possible to link the image to the relevant user in the other table. A foreign key constraint that ensures that images can only be inserted for users who are present in the UserInformation database is defined by the line "FOREIGN KEY (UserName) REFERENCES UserInformation(UserName)".

**Table AddressInformation:** Is used to store the address information of the users registered in the Mini-Facebook. It has a many-to-one relationship with the UserInformation table.

*UserName*: This field serves as a foreign key that points to the UserName field in the UserInformation table. It is a text string with up to 20 characters that is used to specify the user to whom the address belongs. This field serves as the table's primary key as well.

*Degree:* This field, which is designated as a medium size text, is used to store the user's academic standing. (MEDIUMTEXT). This field cannot be left empty. (NOT NULL).

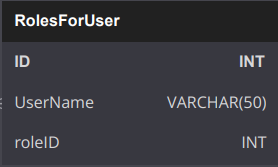
*School:* This field, which is designated as a medium-sized text field, is used to store the name of the educational establishment where the user completed his schooling. (MEDIUMTEXT). This field cannot be left empty. (NOT NULL).

*Street:* This field, which is designated as a medium-sized text field (MEDIUMTEXT), is used to store the street address of the user's home. This field can't be left blank (NOT NULL).

*Town:* This field, which is designated as a medium-sized text field (MEDIUMTEXT), is used to store the user's hometown’s name. This field can't be left blank (NOT NULL).

*State:* This field is designated as a medium size text (MEDIUMTEXT) and is used to hold the name of the state or province where the user resides. This field can't be left blank (NOT NULL).

*Country:* This field, which is designated as a medium-sized text field (MEDIUMTEXT), is used to hold the name of the nation where the user resides. This field can't be empty (NOT NULL).

**Table RolesForUser:** The RolesForUser table is used to assign roles to Mini-Facebook users. It has a many-to-one relationship with the UserInformation table and Roles table.

*ID:* Every record in the database has a unique identification column called ID. It is an auto-incrementing, non-null integer. The primary key of the table is this field.

*UserName:* This is the user name connected to the role, and it can contain up to 50 characters. This field must be present and functions as a foreign key to the UserName column of the UserInformation table. This enables linking the role to the appropriate user in the other table. The foreign key constraint "FOREIGN KEY (UserName) REFERENCES UserInformation(UserName)" guarantees that only users who are present in the UserInformation table can be allocated with roles.

*roleID:* is a non-null integer that serves as the user's given role's unique identifier. The roleID field in the Roles table is referenced by this field, which serves as a foreign key. Only roles that are present in the Roles database may be assigned thanks to this safeguard. Only roles that are present in the Roles table may be allocated, thanks to the foreign key restriction defined by the line "FOREIGN KEY (roleID) REFERENCES Roles(roleID)".

A screenshot of a computer

Description automatically generated with low confidence**Table Roles:** Is used to store information about the different roles that exist in the system. It has a one-to-many relationship with the RoleForUser table and roleForWebPage table.

*roleID:* Each role in the table is uniquely identified by its column roleID, which is a non-null, auto-incremental integer. This field serves as the table's primary key.

*roleName:* The role name can be up to 50 characters long and is supplied as a text string. The job is described in this area, which is necessary.

*description:* is a text string of up to 255 characters that contains a succinct description of the role. This compulsory field is used to provide information about the role.

**Graphical user interface, application

Description automatically generatedTable rolesForWebPage:** The table roleForWebPage is used to assign roles to web pages in the system.

*ID:* Each record in the table has a unique identification field called ID, which is defined as a non-null, auto-incrementing integer. The table's primary key is this field.

*RoleID:* This is an integer that is not null and serves as the role identification for the web page. The roleID field in the Roles table is referenced by this field, which serves as a foreign key. This makes it possible to link the role to the relevant web page in the other table. Only roles that already exist in the Roles database can be assigned thanks to the foreign key constraint defined by the line "FOREIGN KEY (RoleID) REFERENCES Roles(roleID)".

*Page:* A text string with up to 40 characters is designated as the name of the web page connected to the role. This field must be present and functions as a foreign key to the webPages table's Page field. This makes it possible to tie the web page to the role that corresponds to it in the other table. The foreign key constraint "FOREIGN KEY (Page) REFERENCES webPages(Page)" ensures that only web pages that are present in the webPages database can be allocated.

**A screenshot of a computer

Description automatically generated with low confidenceTable webPages:** Is used to store information about web pages in the system.

*Page:* The web page's name is defined as a text string with a maximum of 40 characters. This field is required and serves as the table's primary key.

*Description:* It is defined as a medium-length text string (MEDIUMTEXT) that cannot be null. It is the description of the web page. This field is employed to provide more details about the page.

*menuID:* is a numeric value that serves as the identification of the menu item that corresponds to the page. It serves as a foreign key to the menuID field in the menuElement table and is an optional field. As a result, you can connect the website to the comparable menu item on the other website.

**A screenshot of a computer

Description automatically generated with low confidenceTable menuElement:** The menuElement table is used to store information about the Mini-Facebook menu items, depending on the type of Role a user has.

*menuID*: The identifier for the menu item is a number value. This field, which acts as the table's main key, is necessary.

*title:* is contained in this text string, which has a maximum character count of 40. This essential property contains the name of the menu item.

*Description:* It acts as a menu item description and is classified as a medium text string (MEDIUMTEXT) that cannot be null. In this box, you can add further information about the menu item*.*

Graphical user interface, text, application

Description automatically generated**Table webPageFlow:** The webPageFlow table is used to record the user's navigation flow in the web system.

*id:* This is the identifier of the page flow record and is defined as an auto-incrementing numeric value. This field is mandatory and is used as the primary key of the table.

*currentPage:* This is the name of the current page being viewed by the user and is defined as a text string of up to 255 characters. This field is mandatory and is used to record the current page.

*previousPage:* This is the name of the previous page that the user has visited and is defined as a text string of up to 255 characters. This field is optional and is used to record the previous page.

**UML Diagrams:**

**A picture containing diagram

Description automatically generated**

**Graphical user interface, text

Description automatically generated**

**Text

Description automatically generated**

**Activities Flow Diagram for Android App**

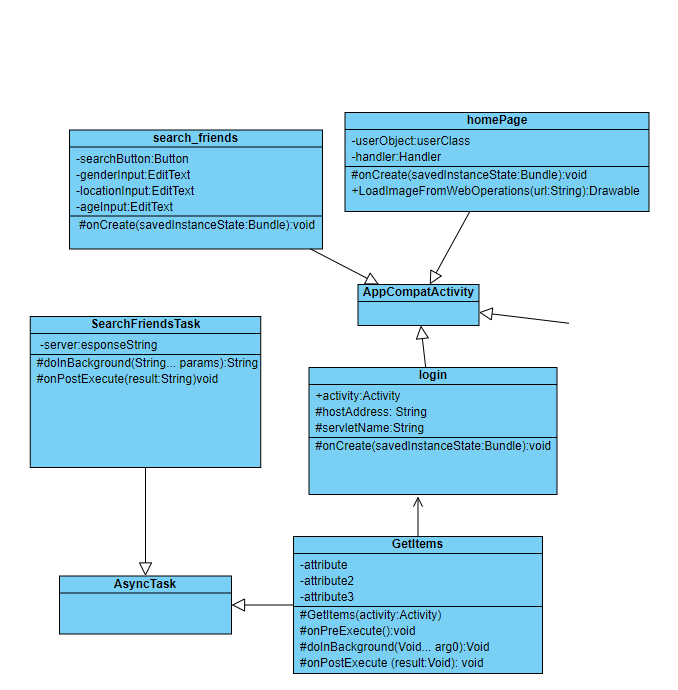
**A picture containing text, screenshot, font, line

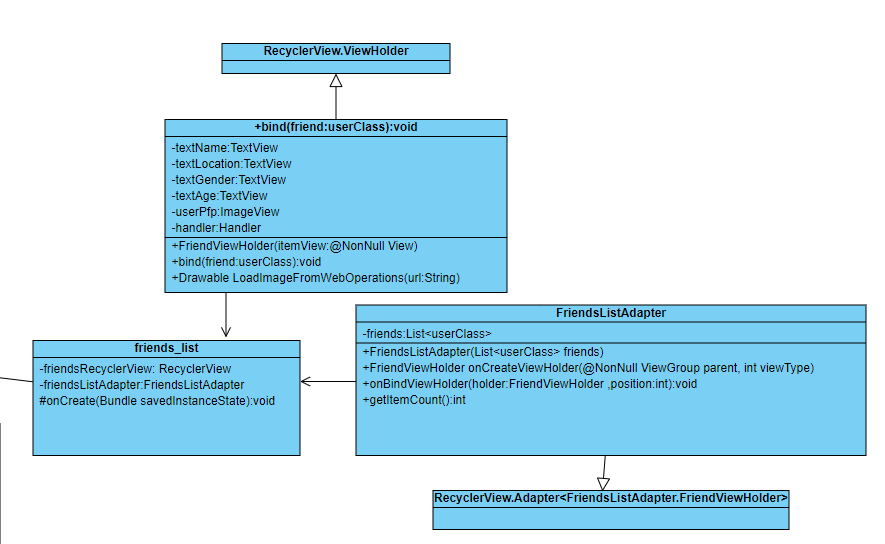
Description automatically generated**

**UML for Android App**

**A picture containing text, screenshot, font, line

Description automatically generated**

****

****

**A picture containing text, screenshot, font, software

Description automatically generated**

**Page Dataflow**

If Admins enter to Mini-Facebook:

**Diagram

Description automatically generated**

If a normal user enters to Mini-Facebook:

Diagram

Description automatically generated

**Web Server**

**login.html**

In this page, the user will find some textboxes to put the required information. In this case, the information will be username and password. This data is handled with the "POST" method so that the data is kept secure and is not part of the URL. You will also find a hyperlink in the word create account, which will take you to a page to register.

**signUp.html**

On this page, the user will be able to register to access the Mini-Facebook. You will find several textboxes that must be filled in. Among the fields to be filled in are userName, password, name, date of birth, gender, email address, profile picture. Also, a user can add his address (including street, town, state, and country) and his education including degree and school.

**uploadProfilePicure.jsp**

This page will only take care that the user has uploaded the picture, but it will only store the path to where the picture is stored.

**homePage.jsp**

This is the page that the user will see once he validates his userName and password, this page will show a menu which are the options that the user has. If he is an Admin, he will be able to see more options than a normal user.

**searchUser.jsp**

This page is in charge of searching for any user already registered in Mini-Facebook. It shows several textboxes that must be filled to search for a registered user. Depending on the fields that are filled, it will search for a user that has the information that was put in the textboxes.

**AddUser.jsp**

This page allows only the admin to add a new user to the Mini-Facebook.

**removeUser.jsp**

This page allows only the admin to remove a user that is registered in the Mini-Facebook.

**editProfile.jsp**

This page allows you to modify some aspects of the profile of the user that is registered in the Mini-Facebook.

**modifyUser.jsp**

This page allows only the admin to modify a user that is registered in the Mini-Facebook.

**Android Activities**

**activity\_login.xml**

Allows users to enter their credentials (such as username and password) to access the Mini-Facebook application and use its functions.

**activity\_search\_friends.xml**

provides users with an easy way to search for friends online. The features of “activity\_search\_friends” include finding someone by their location, age, and/or gender.

**third\_activity.xml**

This activity is to show the user's personal information in his profile. It is rather considered the "home page" of the user and it shows the following information: name, date of birth, gender, email address, the profile picture, and her address (including street, town, state, and country).

**activity\_friends\_list.xml**

This activity was created for the purpose of indicating to the list the format in which it should display the information.