SIT305 Assignment 2 Task 1 Semester 1 2020

RKWebb 215168986

**Late comment:**

Due to an inability to determine how to extract the individual row data from the “album” and “owners” table once the search was successful I was unable to implement the “activity\_record\_found” and “activity\_owner\_found” screens.

**Overview:**

This app will provide participants with a rich collection of LP records owned by themselves and their friends. As each participant contributes to the database they will be able to search for any LP record of interest and the app will return who possesses the LP record, how far away they are, what are their hours to allow LP records to be borrowed and what is the allowed borrowing time.

**Product Purpose:**

**a. Target audience**

This app is targeted to those users who enjoy the experience of listening to analog rather than digital music. Of those who currently purchase LP records, 50% are aged 35 years or less while most household’s own LP records either purchased by themselves or purchased by their parents or grandparents. I am 70 years of age and possess over 400 records from the 60’s, 70’s and 80’s.

The current data from the music industry indicates that for the first time since the 80’s that more LP’s are being purchased on a weekly basis than CD’s.

**b. Reasoning of how your project demonstrates creativity:**

I am not aware of any app that facilitates a group of friends having the ability to search, and then borrow LP’s from each other. It is planned that using the Google maps API will enable each user to ascertain the directions to each other user and travel time required. They will also have access to information regarding the availability of the owner to lend the album.

**Features:**

The access to the database will only be available to registered users.

The user will be able to enter each of their LPs and associate their name with each album.

Users will be able to search for the presence of an album in the database.

This searching will initially be by Album name and Artist name.

Users will be able to search for Owner’s details by Owner name.

The app will significantly widen the availability of users to experience analogue sound for albums they may not have.

**Design:**

**Wireframes:**

**![A close up of text on a white background

Description automatically generated]()**

![A close up of text on a white background

Description automatically generated]()

![A close up of text on a white background

Description automatically generated]()

**![A close up of text on a black background

Description automatically generated]()**

**![A close up of text on a white background

Description automatically generated]()**

**![A close up of text on a white background

Description automatically generated]()**

**![A close up of text on a white background

Description automatically generated]()**

**![A close up of text on a white background

Description automatically generated]()**

**![A close up of text on a white background

Description automatically generated]()**

**Data:**

The app consists of one database:

lp.db.

The app consists of three tables:

“users”, “album” and “owners”

The “users” table consists of the following columns:

“user\_id” Integer Autogenerated

“username” Text Name of registered user

“Password” Text Password of registered user

The “album” table consists of the following columns:

“record\_id” Integer Autogenerated

“albumName” Text The name of the album/record/LP

“artistname” Text The name of the artist/band

“labelName” Text The name of the company producing the LP

“albumYear” Text The year the LP was released

“ownerName” Text The name of the user who possesses the LP

The “owners” table consists of the following columns:

“owner\_id” Integer Autogenerated

“owner2Name” Text The name of an owner

“ownerStreet” Text The address of the owner

“ownerSuburb” Text The suburb of the owner

“ownerPostcode” Text The postcode of the owner

“ownerState” Text The state in which the owner lives

“ownerPhone” Text The phone number of the owner

“ownerDays” Text The days the owner is available for a user to pick up an

Album

“ownerTimes” Text The times the owner is available for a user to pick up an

Album

The data will be permanently stored on the device. A future version may make use of a central database solution so that all users have access to the most up-to-date information. If stored on the cloud security will need to be considered to protect the addresses of the users and owners.

![A screenshot of a cell phone

Description automatically generated]()

**A screenshot of a cell phone

Description automatically generated**

**![A screenshot of a cell phone

Description automatically generated]()**

**API/CLASS Structure**:

MainActivity.java

|

|

------------------ MainActivity2.java --------------------------------------------------

| | | |

| | | |

AddRecord FindRecord AddOwner FindOwner

| |

| |

ShowAlbum ShowOwner

+ DatabaseHeleper.java class

This top level class is the MainActivity.java class and contains the code to login a user if they exist in the database:

If the login is ok then this class calls the MainActivity2.java class.

If the user is not in the database the user can click on the “REGISTER” button to call the RegisterActivity.java class. This call the “insert1” method from the Databasehelper.java class.

In the MainActivity2.class there is the option to call four other classes.

These are the:

AddRecord.java class – To add a new album details – this calls the “insert2” method from the Databasehelper.java class.

FindRecord.java class – To find an album – this calls the “fetchAlbum” method from the DatabaseHelper.java class.

AddOwner.java class – To add new owner details – this call the “insert3” method from the DatabaseHelper.java class.

FindOwner.java class – To find an owner – this calls the “fetchOwner” method from the DatabaseHelper.java class.

In the FindRecord.java class, if the record is found then the ShowAlbum.java class is called.

In the FindOwner.java class, if the owner is found then the ShowOwner.java class is called.

The DatabaseHelper.java class initialises the “lp” database and the “users”, “album” and “owners” tables. It also contains the methods to search all three tables and to insert new data into each of the tables.