**Project 1**

**Option 1: Inventory App**

William Brandow

Southern New Hampshire University

CS-360-T4233: Mobile Architecture & Programming

Professor Matthew Parker

2023-03-19

**Contents**

Application Name . . . . . . . . . . . .

Application Goals . . . . . . . . . . . .

Target User . . . . . . . . . . . . . . .

Major Design Features . . . . . . . . . .

Functional Requirements . . . . . . . . .

References . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . 3

. . . . . . . . . . . . . . . . . . 3

. . . . . . . . . . . . . . . . . . 3

. . . . . . . . . . . . . . . . . . 4

. . . . . . . . . . . . . . . . . . 6

. . . . . . . . . . . . . . . . . . 8

**Application Name**

Mobile2App InventoryApp

**Application Goals**

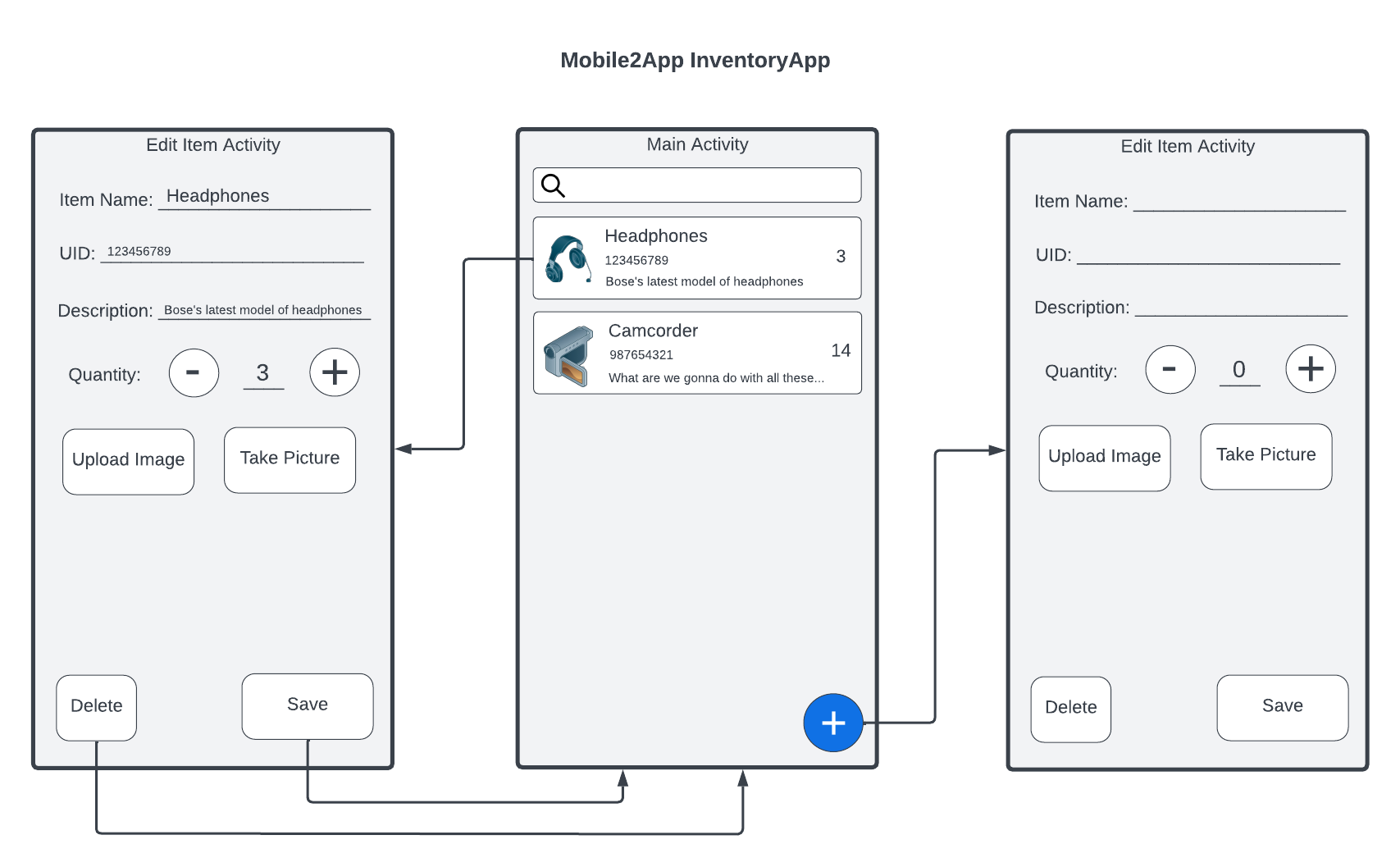
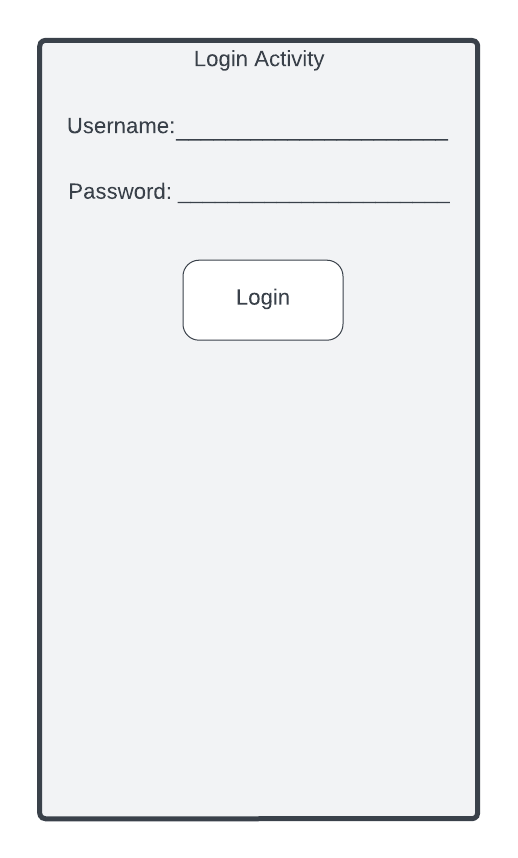
Mobile2App’s InventoryApp is intended to assist users in managing inventory items in a warehouse environment. The application will allow users to create and delete items and edit item quantities. The application will also notify users when an item’s quantity reaches zero.

**Target Users**

The intended user for Mobile2App’s InventoryApp is a warehouse worker. The worker will be receiving and shipping orders and will need a way to conveniently and efficiently keep track of items coming in and going out as they work.

Another user of the application could be a warehouse manager. One use case a manager might need to use the app for could be reconciling inventory numbers. A manager would want to make sure the number of items in the warehouse was equal to the number they should have according to their accounting books. In this case the manager would need to quickly see the quantity for items. If the numbers do not match, they would need to go do a physical inventory count and adjust the numbers accordingly.

Both of these users would be made more efficient by the ability to search for a particular item they need to view or modify.

**Major Design Features**

The first iteration of Mobile2App InventoryApp will consist of 2 screens, a Main Activity screen and an Edit Item Activity screen. The Main Activity screen will list all items currently in inventory, allow the user to search for a particular item, allow the user to select an item to edit, and allow the user to create a new item. To accomplish this the Main Activity will consist of a GridLayout with 3 columns. The first row will have a search icon in the first column and an EditText that spans column 2 and 3. After that, each item will occupy its own row with the first column containing an ImageView. The second column will contain a LinearLayout with a vertical orientation with three TextViews for name, a unique identifier, and description. The third column will contain a TextView with the item’s quantity. The application will also include a floating action button in the bottom right that the user will be able to use to create a new item.

Clicking on an item or on the FAB will open the Edit Item Activity screen. The Edit Item Activity will allow users to edit the name, unique identifier, description, and quantity of the item as well as add a photo for the item. The user will then be able to save or delete the item. To accomplish these goals the Edit Item Activity screen will consist of a GridLayout with 2 columns. Item Name, UID, and Description will each occupy their own row with a TextView in column 1 and an EditText in column 2. Quantity will occupy the next row with a TextView in column 1 and a LinearLayout in column 2. The LinearLayout will contain a Button to decrement, and EditText, and a Button to increment. In this way a user can choose how they would like to edit the quantity to increase their efficiency. The next row will contain a button to delete the item in the first column and a button to save the item in the second column. The save button will be much larger than the delete button to decrease the chances of accidentally hitting delete. When a user navigates to the Edit Item Activity screen from the floating action button there will be no default values for any of the EditTexts except for Quantity which will have a default of 0. When a user navigates to the Edit Item Activity screen by clicking on an item the EditTexts will use data from that item as default values.

The Login Activity screen will be simple. It will consist of a GridLayout with 2 columns. The first row will have a TextView in the first column and an EditText in the second column for the user to enter a username. The second row will be the same as the first for password. The third row will contain a Button that spans both columns to submit the info.

This application will take care to adhere to standard Android design and quality guidelines. For example, every EditText will contain a minHeight value of 48dp to prevent any clickable fields from being too small. Also, the Login Activity will provide hints to autofill account credentials. The layout of all the screens will be designed for both portrait and landscape orientation and tested in each.

**Functional Requirements**

Mobile2App InventoryApp will store data in an SQLLite database. This relational database will have two tables. The user table will store username and password data and the item table will store item data. When the user logs in the application will need to search the user table for a matching username and check to make sure the provided password matches the stored password. If successful it will allow the user to log in. This will move the application into the Main Activity Screen which will iterate through the items to populate our layout. This will be equivalent to querying our database with SELECT \* FROM items. Each item will populate one row of our GridLayout with an image in column 1, the name, UID, and description in column 2, and the quantity in column 3.

When a user searches for a particular item it will affect what the database query is. For example, a user might search for ‘123456789’. The corresponding query will return items where either the name or UID matches this value. The application will then populate the Main Activity layout with the results of the query.

When a user clicks on an item to edit it the data from its name, UID, description, and quantity TextViews will be used for the default values in the corresponding EditText fields of the Edit Item Activity screen. This will help to ensure faster load times that if that data needed to be pulled from the database again. When a user clicks the save button an update query will be called on the database with the entered information. If a user clicks the delete button, the application will use the UID from the Main Activity screen, which was used to populate a default value, to perform a delete query on the database. If the user navigates to the Edit Item Activity screen from the create item FAB, then the save button will cause the application to perform a create query with the provided information. If the user clicks the delete button after navigating to the Edit Item Activity screen via the FAB, the application will simply return to the Main Activity screen without performing any query on the database.

**References**

*Android mobile App Developer tools*. Android Developers. (n.d.). Retrieved March 17, 2023, from https://developer.android.com/