SD6501 – Mobile Application Development

Assignment 2

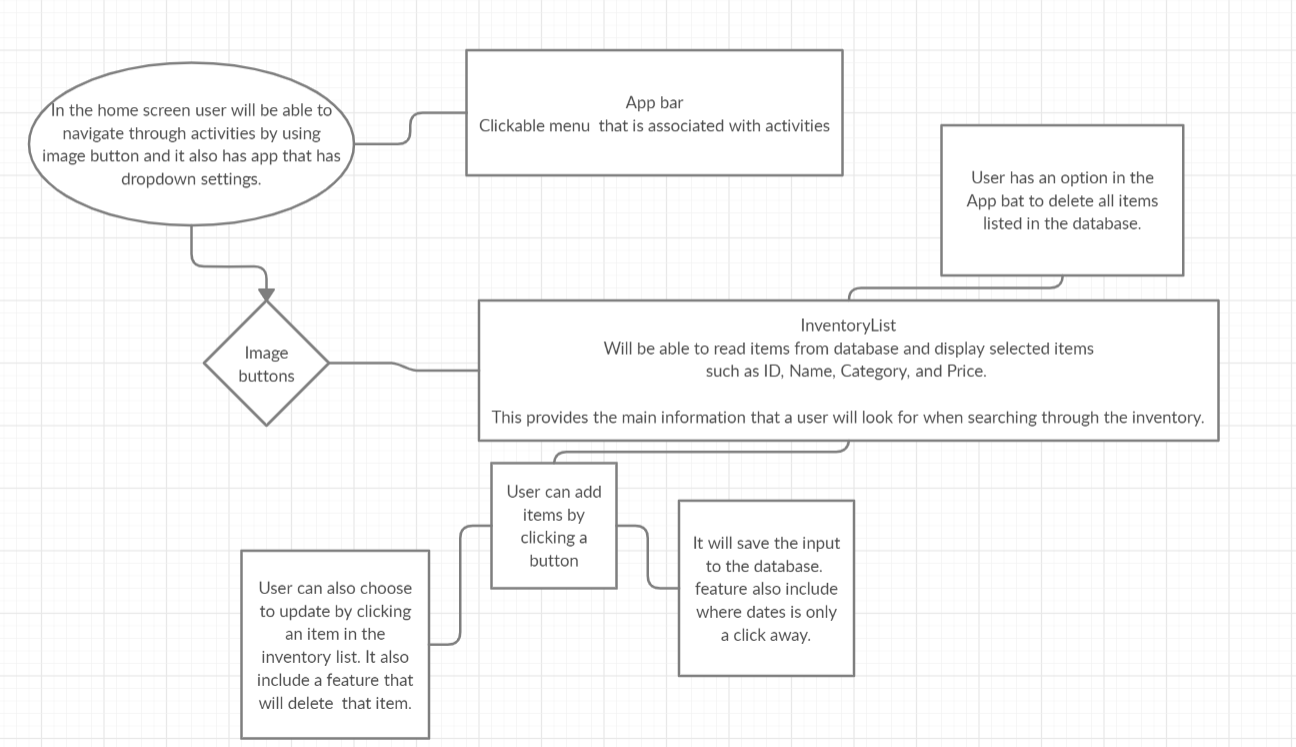
Due Date: Sunday, 20th September (11:55pm)

**Pocket Tracer App**

In this assignment I am building a new components for my existing app from assignment 1. I will be adding SQLite Database and an app bar where it’ll provide settings that a user can navigate along the app. The database features include read and update the details of InventoryItemDetails that is chosen by the user. The user will be able to either delete all at once or specific item. User will be able to add items in the InventoryList.

Some improvements I will be doing is to add animation InventoryList where it’ll smoothen the interface for the user. I will be replacing the buttons with Image button that has a transparent background which will result a friendly interface for the HomeScreen.

**Conceptual framework**



**List of features and concepts**

Login Screen – Username and password with log in button. ( *admin and pass123)*

Home Screen – Image buttons(Calendar, Inventory list, and details of the Property)

* + - Settings menu – logout, account information, dashboard and property tab.
    - It include back button which reverts back to home screen.

Add item screen – saves user input to the database.

Update item screen – The data is collected from the database and displayed where it is editable and may be updated by clicking update button. The data can also be deleted.

Delete button in the settings menu that will delete everything in the database.

A list of classes that are used in the app for the activities.

ItemDetails – id, name, category, brand, price, purchaseddate, and warrantydate.

Property – address, garage, backyardsize, frontyardsize, bedroom, bathroom, and propertyname.

DatabaseHelper – includes the features of read, create, delete, and add.

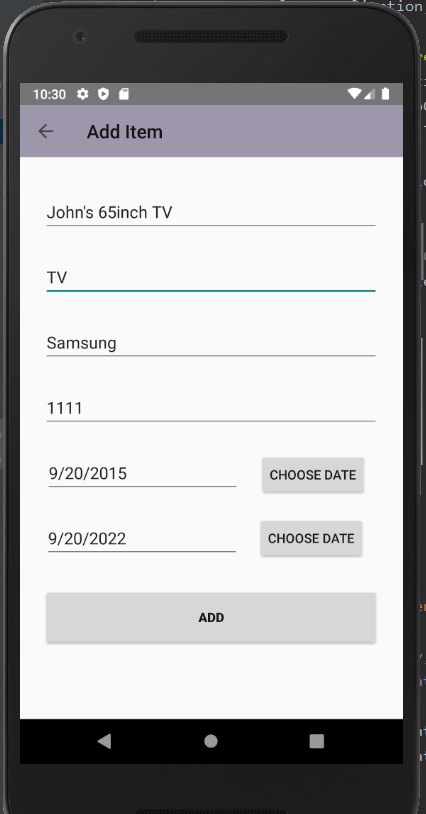
CustomAdapter – this feature allows to hold data that is passed from database, which displays in a textview.

**Constraints and Strategies applied**

The Items takes a date format feature that includes “/” between numbers. This wasn’t directly accepted in the code to be stored in the database. The strategy that I applied is to turn the date in a string and convert it into milliseconds. This is then formatted back to a date that how it was originally entered by the user.

I’ve also come where it had small typos but did not affect the run time of the app but causes crashes. I started debugging each line going through the data in the console and debugging console. Using the debugging feature in the android studio allowed to me to fix the fault.

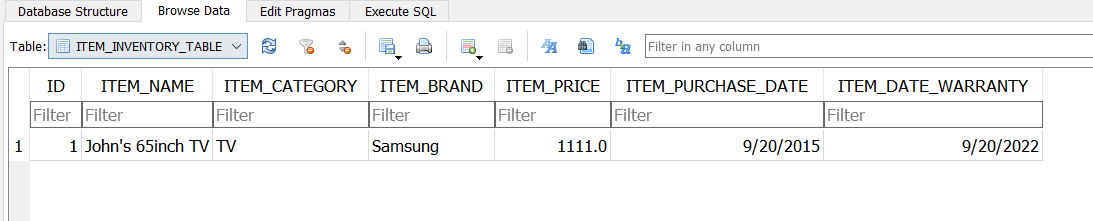
**Updated debugging evidence performed**



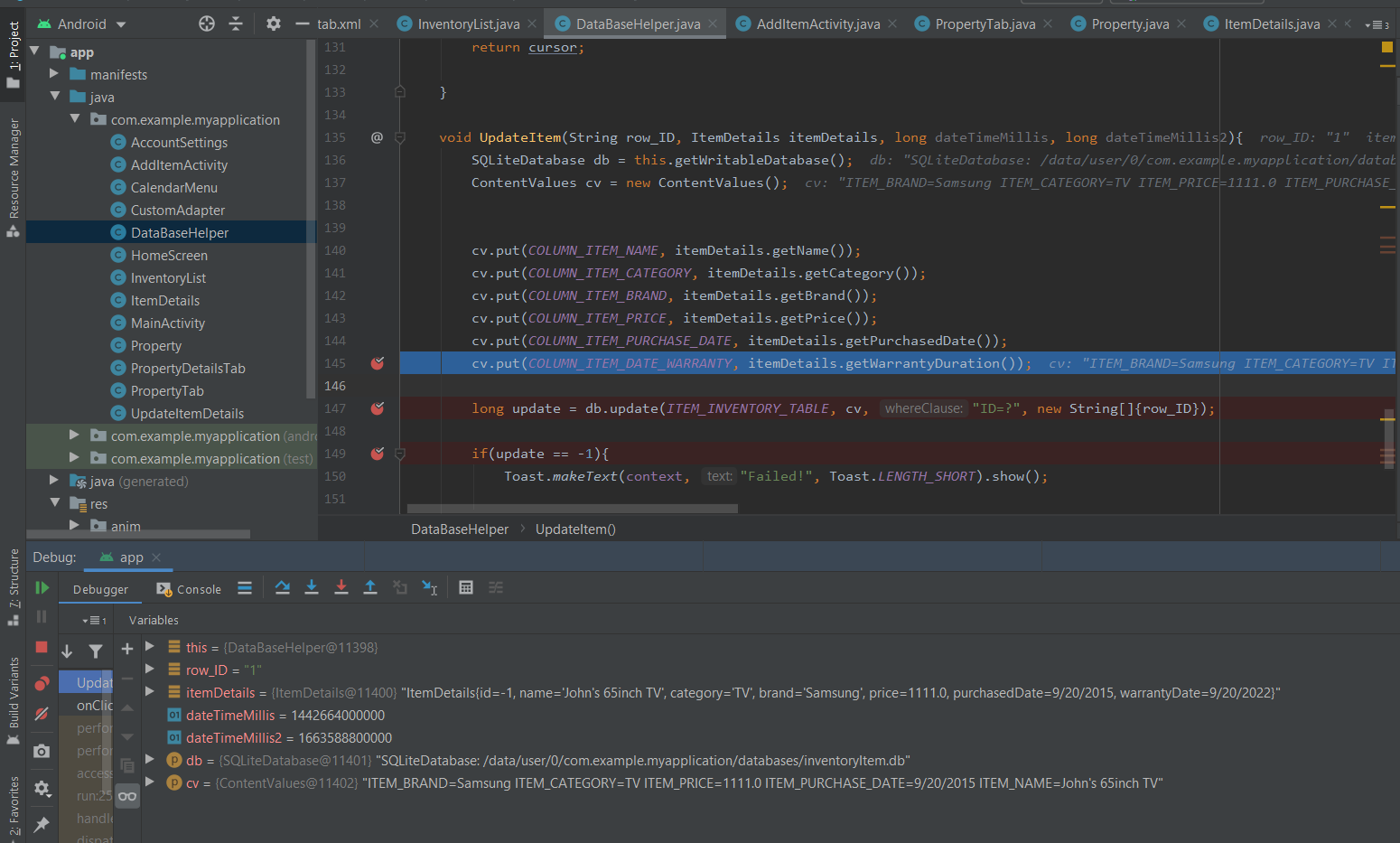
@RequiresApi(api = Build.VERSION\_CODES.*N*)  
public static long getTimeMillis(String dateString) throws ParseException{  
 String myDate = dateString;  
 SimpleDateFormat sdf = new SimpleDateFormat("MM/dd/yyyy");  
 Date date = sdf.parse(myDate);  
 long millis = date.getTime();  
  
 return millis;  
}

btnPurchasedDate.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Calendar cal = Calendar.*getInstance*();  
 int year = cal.get(Calendar.*YEAR*);  
 int month = cal.get(Calendar.*MONTH*);  
 int day = cal.get(Calendar.*DAY\_OF\_MONTH*);  
  
 DatePickerDialog dialog = new DatePickerDialog(AddItemActivity.this,  
 android.R.style.*Theme\_Light\_WallpaperSettings*, btnPurchasedDateSetListener, year, month, day);  
  
 //dialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));  
 dialog.show();  
 }  
});

btnPurchasedDateSetListener = new DatePickerDialog.OnDateSetListener() {  
 @Override  
 public void onDateSet(DatePicker datePicker, int year, int month, int day) {  
 month = month + 1;  
  
 String date = month + "/" + day + "/" + year;  
 et\_PurchasedDate.setText(date);  
  
 }  
};



**Test case results**



**Final app components screenshot**

