



FocusFox

CMP309

Tia Cotton: 1602119



Overview of FocusFox



FocusFox is a lightweight all-in-one study tool.



It has a journal, planner and study timer.



There are various customisable features within the app.

FocusFox

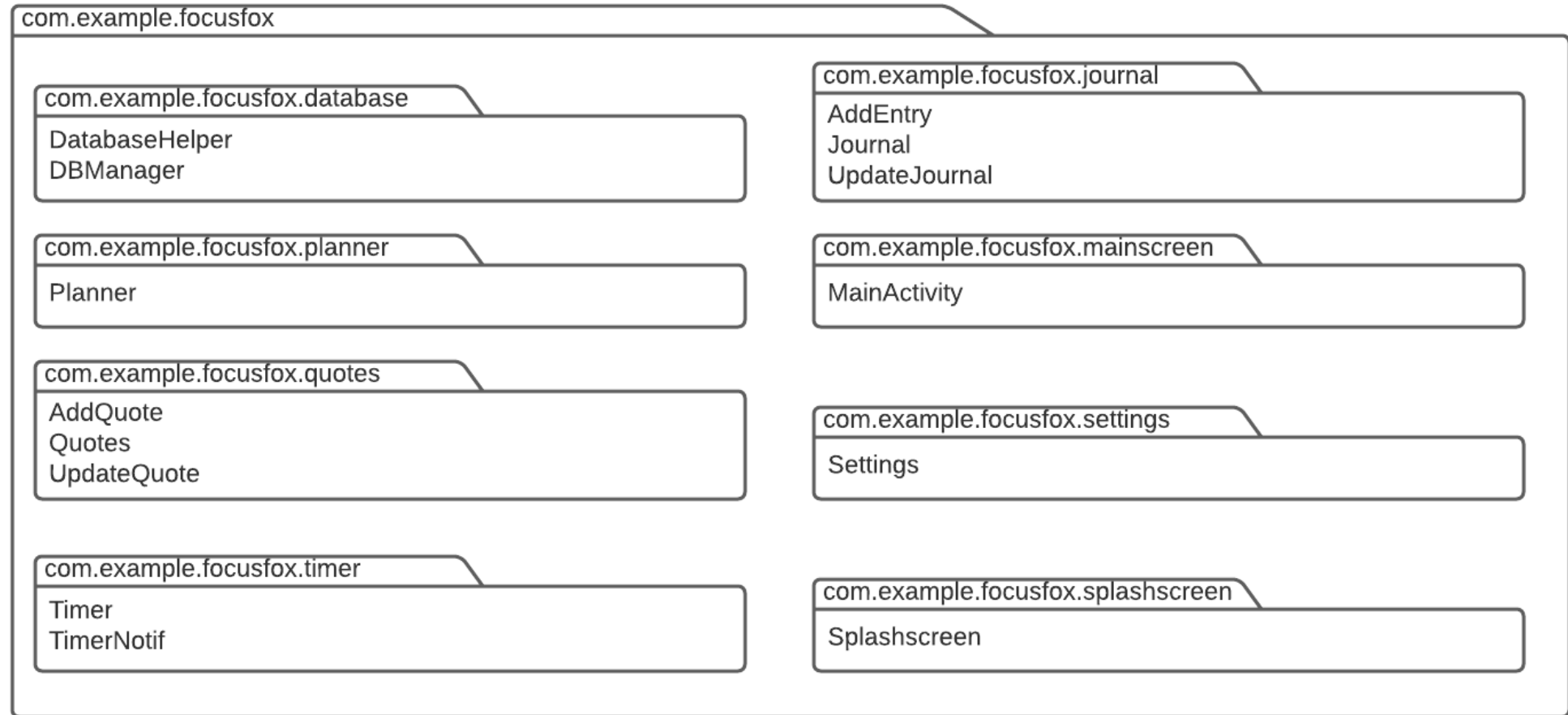
Focusfox

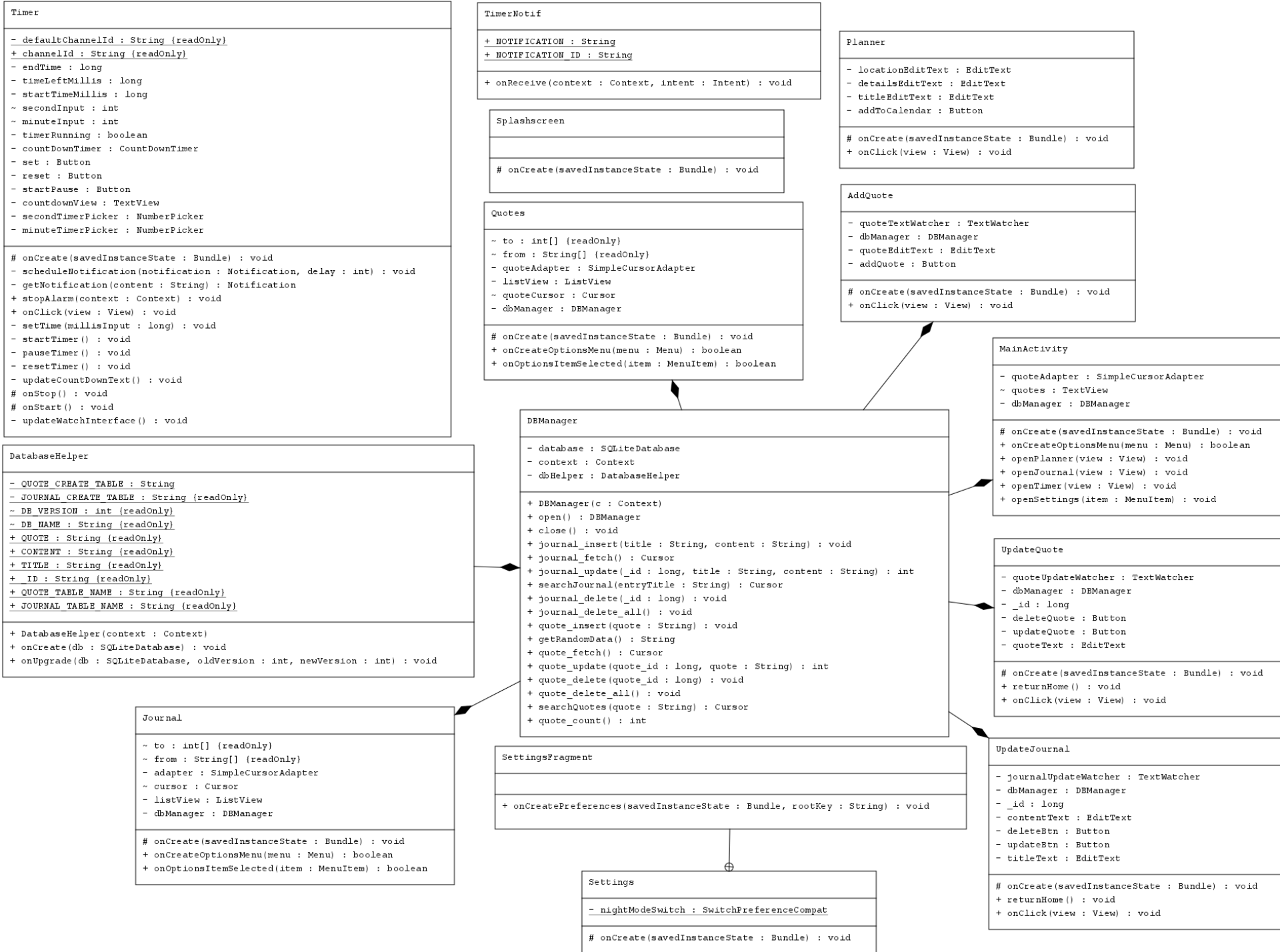


Focusfox



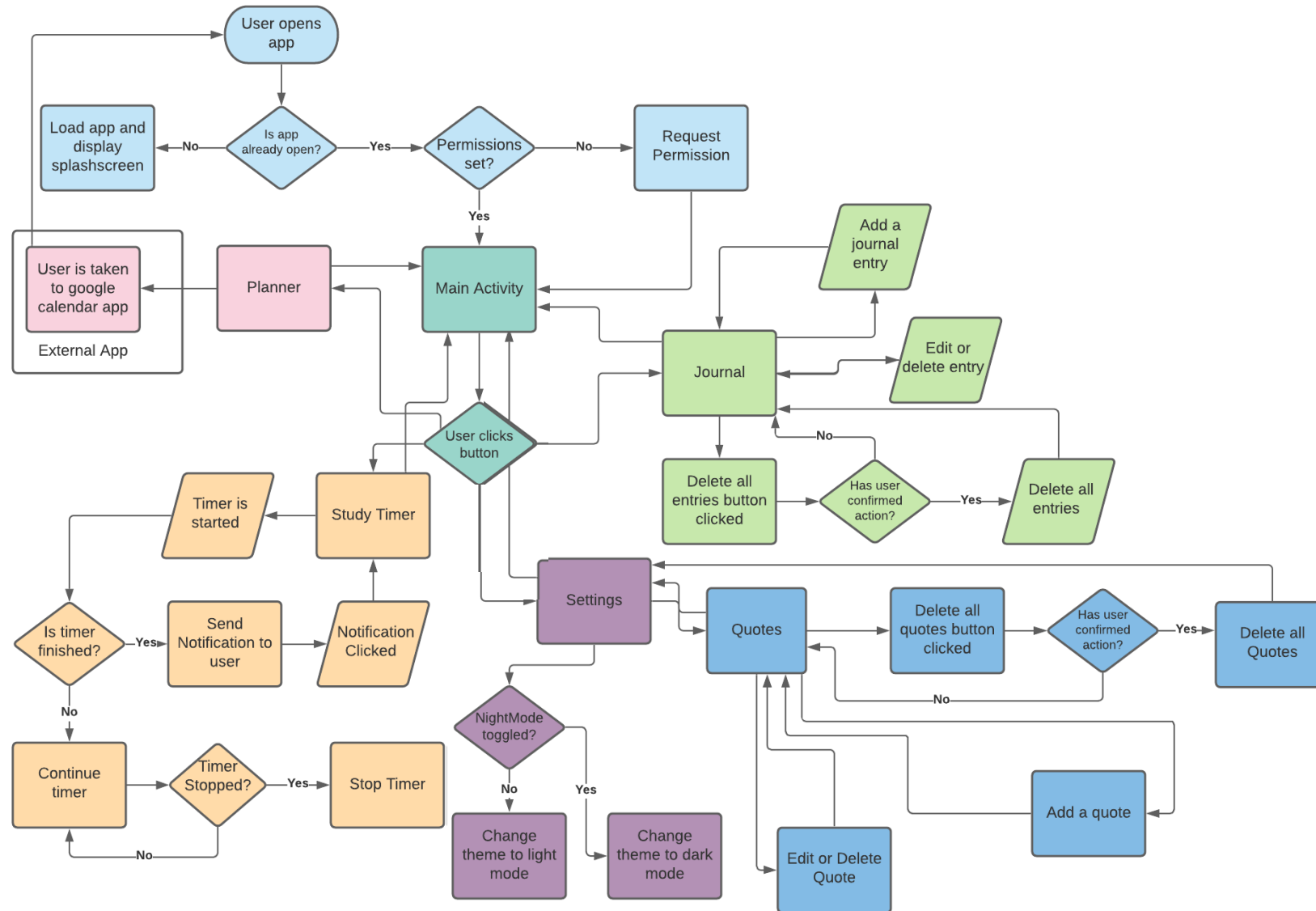
FocusFox - Package Structure





FocusFox - Package Structure

FocusFox – Flow Chart



FocusFox - Main Features

SQLite Database

Search Ability

Google Calendar
Intents

Splashscreen

Dark Mode –
Shared
Preferences

Notifications

Alert Dialogs

Disabling/Enabling
buttons

FocusFox - SQLite Database

- Used to store journal entries and quotes
- Serverless, which allows for offline functionality
- One database, two tables – JournalEntries and QuoteEntries



FocusFox - Search Ability

```
//This function is used to allow the user to search for quotes.
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar
    getMenuInflater().inflate(R.menu.quote_menu, menu);

    //Initialising the searchmanager and searchview for the quote activity
    SearchManager manager = (SearchManager) getSystemService(Context.SEARCH_SERVICE);
    SearchView searchView = (SearchView) menu.findItem(R.id.searchView).getActionView();
    searchView.setSearchableInfo(manager.getSearchableInfo(getComponentName()));

    //Creating the onQueryListener for the search view
    searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String quote) {
            quoteCursor = dbManager.searchQuotes(quote);
            //If the cursor is empty, display a no record toast
            if (quoteCursor == null) {
                Toast.makeText(context, Quotes.this, text: "No records found!", Toast.LENGTH_LONG).show();
            }
            //if the cursor has quotes, display the number of quotes found
            else {
                Toast.makeText(context, Quotes.this, text: quoteCursor.getCount() + " records found!", Toast.LENGTH_LONG).show();
            }
            //Set the listView to the quote(s) within the cursor
            quoteAdapter.swapCursor(quoteCursor);
            return false;
        }

        //Each time the searchView is changed, check for quotes with current value
        @Override
        public boolean onQueryTextChange(String entryTitle) {
            quoteCursor = dbManager.searchQuotes(entryTitle);
            //if the cursor contains quotes, display them in the listView
            if (quoteCursor != null) {
                quoteAdapter.swapCursor(quoteCursor);
            }
            return false;
        }
    });
    return true;
}
```

```
<meta-data
    android:name="android.app.searchable"
    android:resource="@xml/searchable" />
```



FocusFox - Google calendar intents

```
//When the addToCalendar button is clicked, send the user and entered information to google calendars
@Override
public void onClick(View view) {
    //Using an intent to send the entered information to the calendar app with the user
    Intent intent = new Intent(Intent.ACTION_INSERT)
        .setData(CalendarContract.Events.CONTENT_URI)
        .putExtra(CalendarContract.Events.TITLE, titleEditText.getText().toString())
        .putExtra(CalendarContract.Events.DESRIPTION, detailsEditText.getText().toString())
        .putExtra(CalendarContract.Events.EVENT_LOCATION, locationEditText.getText().toString())
        .putExtra(CalendarContract.Events.AVAILABILITY, CalendarContract.Events.AVAILABILITY_BUSY);
    startActivity(intent);
}
```

Add gif of adding item to
calendar here

FocusFox - Splash screen

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:drawable="@android:color/black"/>
  <item
    android:drawable="@drawable/focusfox_removebg_preview"
    android:gravity="center" />
</layer-list>
```

```
<activity android:name=".Splashscreen"
  android:theme="@style/FocusFoxSplash">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

```
public class Splashscreen extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        //This launches the main activity from the splash screen when the 2 seconds is complete
        //The actual splash screen may be longer due the splashscreen covering a cold start
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent splash = new Intent( packageContext: Splashscreen.this, MainActivity.class);
                startActivity(splash);
                finish();
            }
        }, delayMillis: 2000);
    }
}
```

5:36



FocusFox - Dark Mode

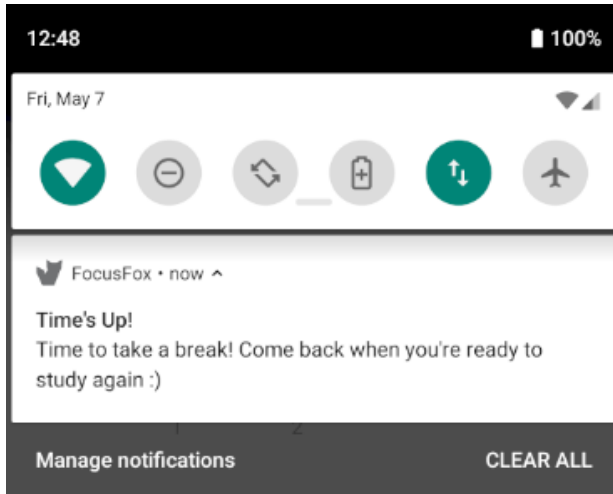
```
//SharedPreferences are used to set app in light or night mode depending on option chosen in settings.
SharedPreferences sharedPreferences = PreferenceManager.getDefaultSharedPreferences( context: this);
boolean nightMode = sharedPreferences.getBoolean("Toggle Night Mode", defValue: false);
if (nightMode)
    AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_YES);
else {
    if (Build.VERSION.SDK_INT <= Build.VERSION_CODES.P)
        AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_AUTO_BATTERY);
    else
        AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_FOLLOW_SYSTEM);
}
```

```
//Setting nightModeSwitch's preference to nightMode within preference fragment
nightModeSwitch = findPreference( key: "nightMode");

if (nightModeSwitch != null) {
    nightModeSwitch.setOnPreferenceChangeListener((preference, newValue) -> {
        //set isChecked boolean to false so night mode is off by default
        boolean isChecked = false;
        //if nightMode toggled check the state of boolean
        if (newValue instanceof Boolean)
            isChecked = (Boolean) newValue;
        //If boolean is true/preference has been toggled on, set theme to night mode
        if (isChecked) {
            getPreferenceManager().getSharedPreferences().edit().putBoolean("Toggle Night Mode", true).apply();
            AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_YES);
        }//If boolean is false/preference has been toggled off, set theme to light mode
        else {
            getPreferenceManager().getSharedPreferences().edit().putBoolean("Toggle Night Mode", false).apply();
            AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_NO);
        }
        return true;
    });
}
```



FocusFox - Notifications



```
public class TimerNotif extends BroadcastReceiver {

    //Declaring the variables to be used for the notification
    public static String NOTIFICATION_ID = "notificationId";
    public static String NOTIFICATION = "notification";

    //This function is used when the Broadcast Receiver is receiving a broadcast intent
    public void onReceive(Context context, Intent intent){
        //When an intent is received, the notification channel is created for the Timer
        NotificationManager notificationManager = (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
        Notification notification = intent.getParcelableExtra(NOTIFICATION);
        if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O){
            int importance = NotificationManager.IMPORTANCE_HIGH;
            NotificationChannel notificationChannel = new NotificationChannel(channelId, name: "Timer", importance);
            assert notificationManager != null;
            notificationManager.createNotificationChannel(notificationChannel);
        }
        int id = intent.getIntExtra(NOTIFICATION_ID, defaultValue: 0);
        notificationManager.notify(id, notification);
    }
}
```

```
//Function to set alarmManager to schedule notification for timer running out
private void scheduleNotification (Notification notification, int delay){
    Intent notificationIntent = new Intent( packageContext: this, TimerNotif.class);
    notificationIntent.putExtra(TimerNotif.NOTIFICATION_ID, value: 1 );
    notificationIntent.putExtra(TimerNotif.NOTIFICATION, notification);
    PendingIntent pendingIntent = PendingIntent.getBroadcast( context: this, requestCode: 0, notificationIntent, PendingIntent.FLAG_UPDATE_CURRENT);
    long futureInMillis = SystemClock.elapsedRealtime () + delay;
    AlarmManager alarmManager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
    assert alarmManager != null;
    alarmManager.set(AlarmManager.ELAPSED_REALTIME_WAKEUP, futureInMillis, pendingIntent);
}

//Function containing the information to be displayed to user in notification.
private Notification getNotification (String content) {
    //Intent for user to open study timer from notification
    Intent alarmIntent = new Intent( packageContext: this, Timer.class);
    PendingIntent alarmPendingIntent = PendingIntent.getActivity( context: this, requestCode: 0, alarmIntent, PendingIntent.FLAG_UPDATE_CURRENT);

    //building the notification with the required information
    NotificationCompat.Builder builder = new NotificationCompat.Builder( context: this, defaultChannelId )
        .setContentTitle("Time's Up!")
        .setContentText(content)
        .setStyle(new NotificationCompat.BigTextStyle()
            .bigText(content))
        .setSmallIcon(R.drawable.focusfox_transparent)
        .setVisibility(NotificationCompat.VISIBILITY_PUBLIC)
        .setAutoCancel(true)
        .setChannelId(channelId)
        .setContentIntent(alarmPendingIntent)
        .setPriority(NotificationCompat.PRIORITY_HIGH);
    return builder.build();
}

//When timer is reset, alarm is cancelled
public void stopAlarm(Context context) {
    AlarmManager alarmManager = (AlarmManager) context.getSystemService(Context.ALARM_SERVICE);
    Intent alarmIntent = new Intent( packageContext: this, TimerNotif.class);
    PendingIntent alarmPendingIntent = PendingIntent.getBroadcast(context, requestCode: 0, alarmIntent, flags: 0);
    alarmManager.cancel(alarmPendingIntent);
}
```

FocusFox - Alert Dialogs

```
//if Delete All Entries is selected, display an alert dialog to confirm deletion
else if (id == R.id.deleteEntries){
    AlertDialog.Builder builder = new AlertDialog.Builder(context this);
    builder.setCancelable(true);
    builder.setTitle("Delete All Entries?");
    builder.setMessage("This cannot be undone");

    //If user confirms deletion of entries, delete all entries within table and confirm with toast message.
    builder.setPositiveButton( text: "Confirm",
        (dialog, which) -> {
            dbManager.journal_delete_all();
            Intent home_intent = new Intent(getApplicationContext(), Journal.class)
                .setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(home_intent);
            Toast.makeText(getApplicationContext(), text: "Entries have been successfully deleted", Toast.LENGTH_SHORT).show();
        });

    //If user clicks cancel, close dialog and display toast to confirm that no entries were deleted.
    builder.setNegativeButton( text: "Cancel", (dialog, which) -> Toast.makeText(getApplicationContext(), text: "No entries were deleted.", Toast.LENGTH_SHORT).show());

    AlertDialog dialog = builder.create();
    dialog.show();
}
```

Delete All Entries?

This cannot be undone

CANCEL CONFIRM

No entries were deleted.

Entries have been successfully deleted

FocusFox – Disabling/Enabling Buttons


```
//Setting the TextWatcher listener to both editText fields
titleEditText.addTextChangedListener(journalTextWatcher);
contentEditText.addTextChangedListener(journalTextWatcher);
}

//A text watcher is used to ensure that the fields are not null
//If the fields are empty, the button is disabled and is enabled when there is text in the field
private TextWatcher journalTextWatcher = new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
        //Required function for TextWatcher to work
    }

    //When the editText fields aren't empty, the button is re-enabled
    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        String titleInput = titleEditText.getText().toString().trim();
        String contentInput = contentEditText.getText().toString().trim();

        addEntryButton.setEnabled(!titleInput.isEmpty() && !contentInput.isEmpty());
    }


    @Override
    public void afterTextChanged(Editable s) {
        //Required function for TextWatcher to work
    }
};
```

 Edit This Entry

Today was a good day!

Enter Content

UPDATEDELETE

 Edit This Entry

Today was a good day!

I had a great day today! It was super, super fun and I can't wait to see what tomorrow brings!

UPDATEDELETE

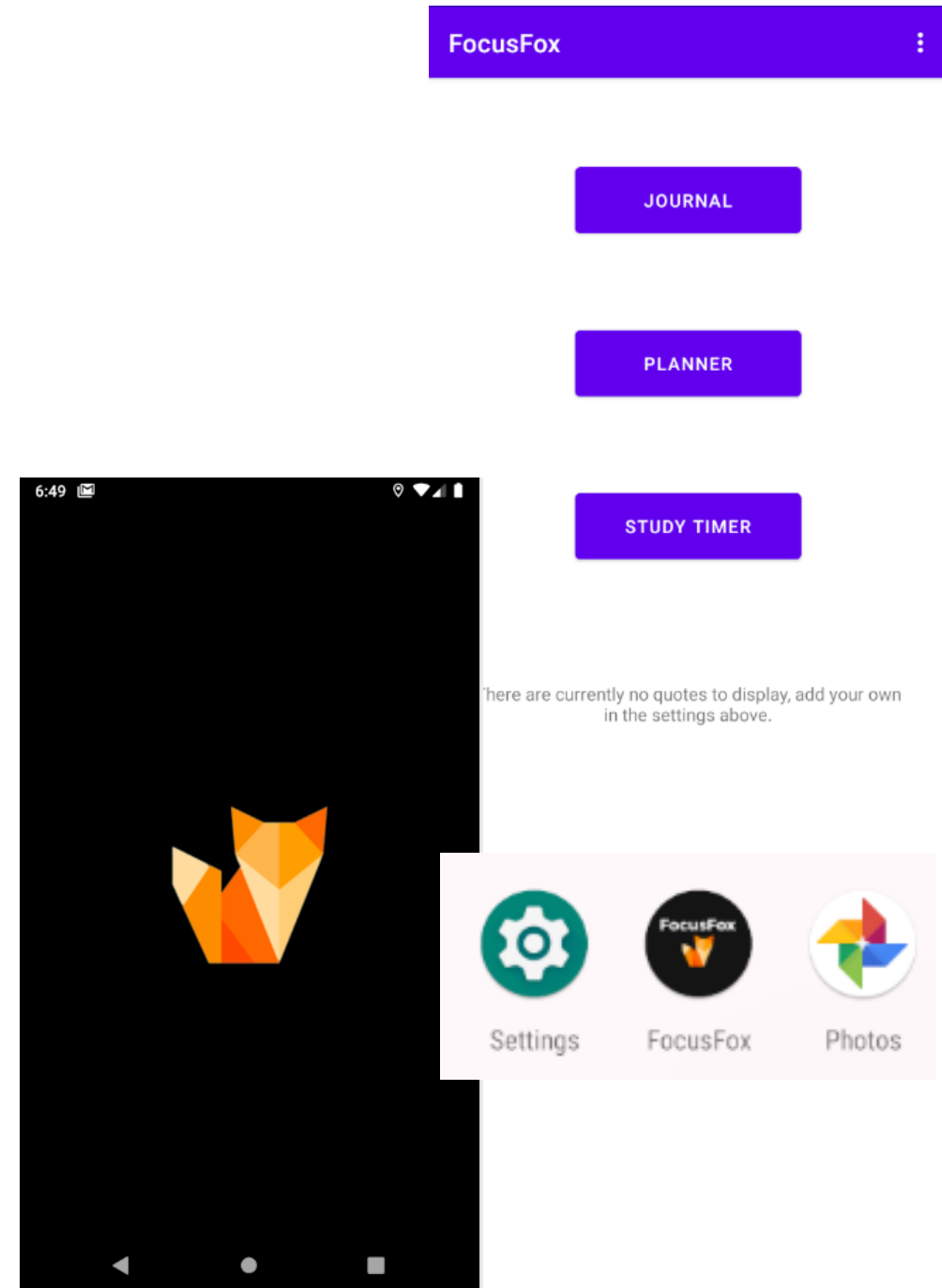


FocusFox – Critical Analysis

Critical analysis of the overall design and implementation of the FocusFox App.

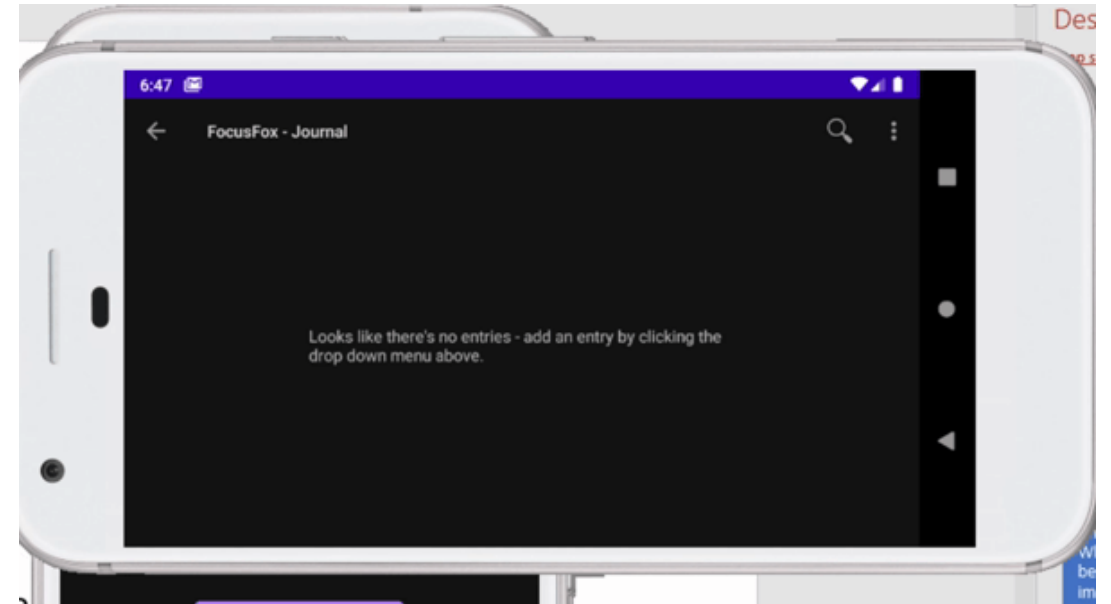
Critical Analysis – What went well?

- Light-weight, sleek and attractive design
- All activities work correctly and are suitable for purpose
- Usability considered throughout
- Using google calendar provides notifications and cloud backup



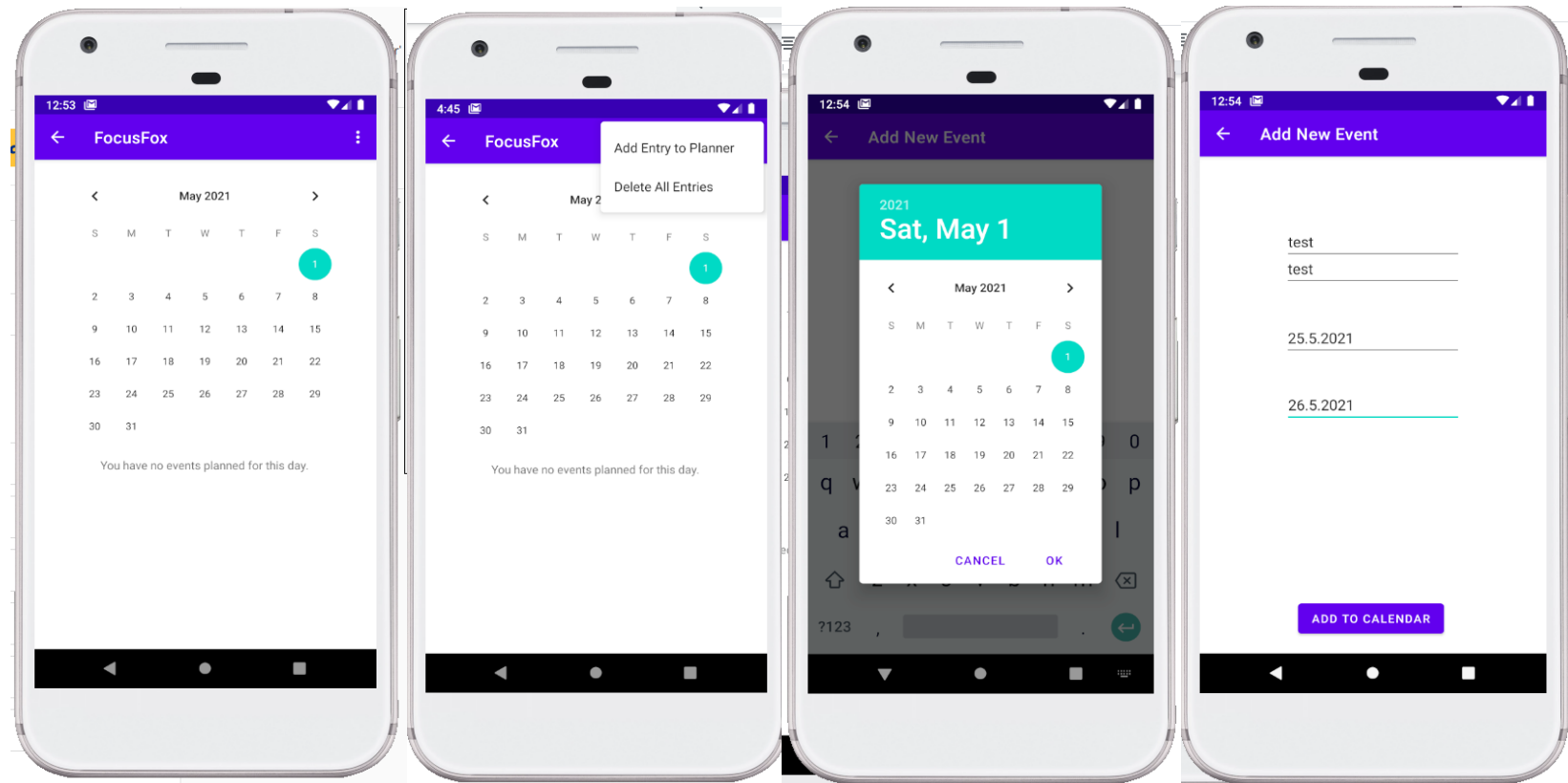
Critical Analysis – What could be improved?

- Ability to add images to journal entries
- Ability to toggle quotes on and off
- Keyboard in full screen causes theme to change
- Timer would have been more effective as a background service



Critical Analysis - What could be improved?

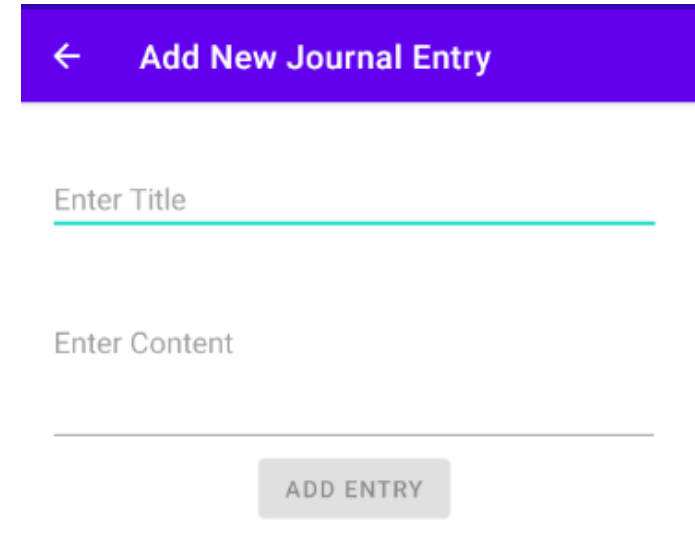
- Hoped to have planner built-in, with option to add to Google Calendar
- Would have been multithreaded too.
- Was attempted, unfortunately did not work as planned



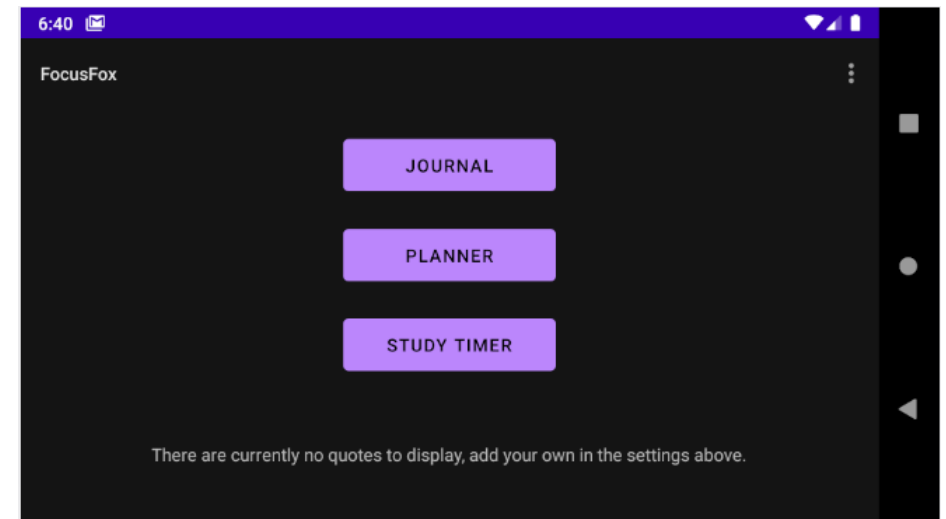
Original Planner activity, with date picker to select and enter dates of event

Usability – What went well?

- User is able to leave app/lock phone when study timer is on
- Landscape/portrait and light/dark mode
- Entries cannot be added until the editText field has a value
- Search utility for journal and quotes



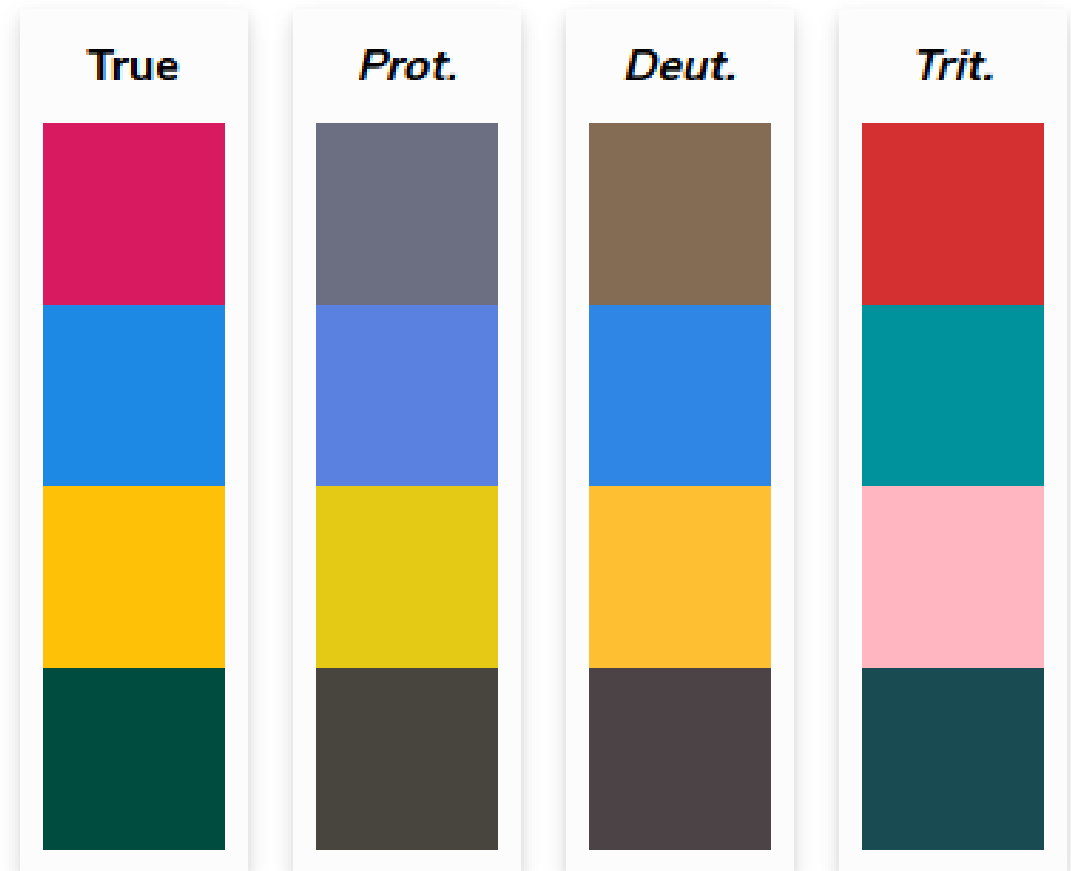
A screenshot of a mobile application screen titled "Add New Journal Entry". The screen has a light blue header bar with a back arrow and the title. Below the header, there are two text input fields: "Enter Title" and "Enter Content". The "Enter Title" field is currently active, indicated by a blue underline. Below the "Enter Content" field, there is a grey button labeled "ADD ENTRY".



Usability – What could be improved?

- Colour Blindness mode/different colour theme
- Change splash screen background to theme

Color Palette



Application Performance – What went well?

- Prefer External Storage to save space in local memory.
- Lazy First concept was considered when building the app.
(Optimize for battery life | Android Developers, 2021)

```
<application  
    android:installLocation="preferExternal"
```

Application Performance – What could be improved?

- Room would have been used instead of SQLite
- Study Timer would have been built as a background service.
- If planner had been built as planned, multithreading would have been used



Security – What went well?

User data is stored locally, rather than being uploaded to cloud database



Libraries and coding practices are up-to-date

No personal data is taken

Security – What could be improved?

- Rex Scanner was used to test the security of FocusFox
- Issues found were relating to the raw queries used in the DBManager class.

Issues & Vulnerabilities

Issue	Severity	Confidence	Summary	
Potential Android SQL Injection	12: Medium	2: Normal	This use of android/database/sqlite/SQLiteDatabase.rawQuery(Ljava/lang/String;[Ljava/lang/String;)Landroid/database/Cursor; can be vulnerable to SQL injection	View Details 
Potential Android SQL Injection	12: Medium	2: Normal	This use of android/database/sqlite/SQLiteDatabase.execSQL(Ljava/lang/String;)V can be vulnerable to SQL injection	View Details 

References

- Android Developers. 2021. *Add and handle actions* | *Android Developers*. [online] Available at: <<https://developer.android.com/training/appbar/actions>> [Accessed 13 March 2021].
- Android Developers. 2021. *AppCompatActivity* | *Android Developers*. [online] Available at: <<https://developer.android.com/reference/androidx/appcompat/app/AppCompatActivity>> [Accessed 11 March 2021].
- Android Developers. 2021. *Calendar provider overview* | *Android Developers*. [online] Available at: <<https://developer.android.com/guide/topics/providers/calendar-provider>> [Accessed 19 April 2021].
- Android Developers. 2021. *Creating a Search Interface* | *Android Developers*. [online] Available at: <<https://developer.android.com/guide/topics/search/search-dialog>> [Accessed 13 April 2021].
- Android Developers. 2021. *Data and file storage overview* | *Android Developers*. [online] Available at: <<https://developer.android.com/training/data-storage>> [Accessed 6 May 2021].
- Android Developers. 2021. *Optimize for battery life* | *Android Developers*. [online] Available at: <<https://developer.android.com/topic/performance/power>> [Accessed 16 March 2021].
- App.rexscan.com. 2021. *REX*. [online] Available at: <<https://app.rexscan.com>> [Accessed 11 May 2021].
- Coding in Flow. 2021. *Part 1 - Countdown Timer* - *Coding in Flow*. [online] Available at: <<https://codinginflow.com/tutorials/android/countdowntimer/part-1-countdown-timer>> [Accessed 27 April 2021].
- Ghori, C., 2021. *How to disable Button if EditText is empty ?*. [online] Stack Overflow. Available at: <<https://stackoverflow.com/questions/22680106/how-to-disable-button-if-edittext-is-empty>> [Accessed 7 May 2021].
- JetBrains Marketplace. 2021. *UML Generator - Plugins* | *JetBrains*. [online] Available at: <<https://plugins.jetbrains.com/plugin/15124-uml-generator>> [Accessed 12 May 2021].
- JournalDev. 2021. *Android Splash Screen* - *JournalDev*. [online] Available at: <<https://www.journaldev.com/17831/android-splash-screen>> [Accessed 4 May 2021].
- JournalDev. 2021. *Android SQLite Database Example Tutorial* - *JournalDev*. [online] Available at: <<https://www.journaldev.com/9438/android-sqlite-database-example-tutorial>> [Accessed 18 March 2021].
- Medium. 2021. *Splash Screen in Android*. [online] Available at: <<https://medium.com/swlh/splash-screen-in-android-8ab250e40190>> [Accessed 4 May 2021].
- Nichols, D., 2021. *Coloring for Colorblindness*. [online] Davidmathlogic.com. Available at: <<https://davidmathlogic.com/colorblind/#%23D81B60-%231E88E5-%23FFC107-%23004D40>> [Accessed 5 May 2021].
- programmerworld. 2021. *How to switch Night Mode/ Dark Mode ON and OFF programmatically in your Android App?*. [online] Available at: <<https://programmerworld.co/android/how-to-switch-night-mode-dark-mode-on-and-off-programmatically-in-your-android-app/>> [Accessed 10 March 2021].
- Studio, A. and Hristov, D., 2021. *App freezes after PreferenceSwitch in settings Android Studio*. [online] Stack Overflow. Available at: <<https://stackoverflow.com/questions/61463103/app-freezes-after-preferenceswitch-in-settings-android-studio?noredirect=1&1q=1>> [Accessed 11 March 2021].

Thank you for listening.

If you have any questions, please
email myself at 1602119@uad.ac.uk
or message me on Teams 😊