How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

GitHub Username: Yasmine Vaz (ygss3614)

Color Wallet Alert

Description

An easy way to update and visualize your financial state through the colors.

Intended User

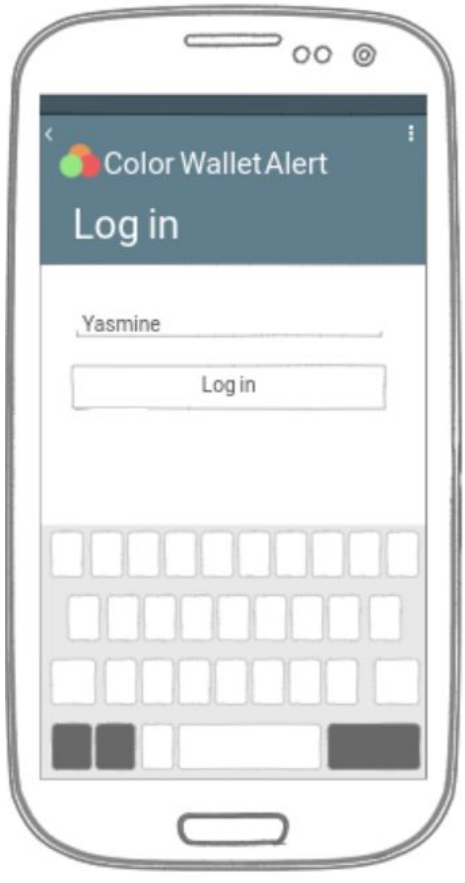
For people who lose track of your financial state every month and never knows how much more can spend

Features

- Show a financial state board
- Records spends by categories
- Alert user when it’s time to stop spending

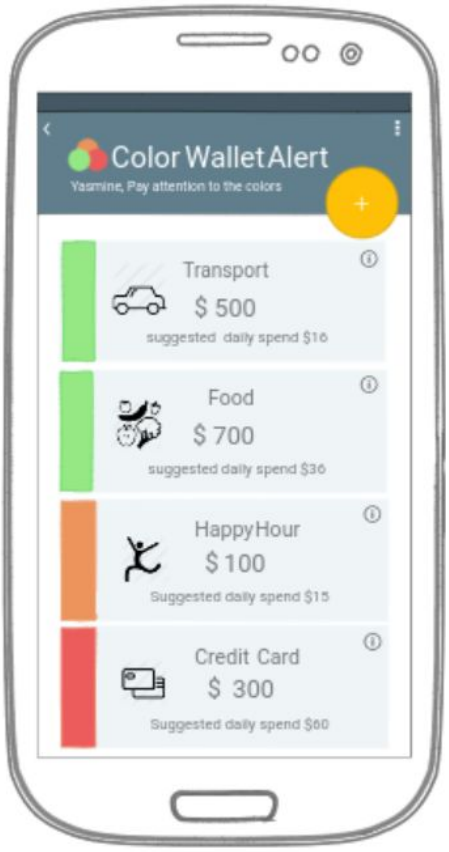
User Interface Mocks

Screen 1



A login screen to identify user by name. The user needs to fill the field name and click on Log in to access the main screen.

Screen 2



This is the main app screen, where are listed all registered categories by user. Each card category shows:

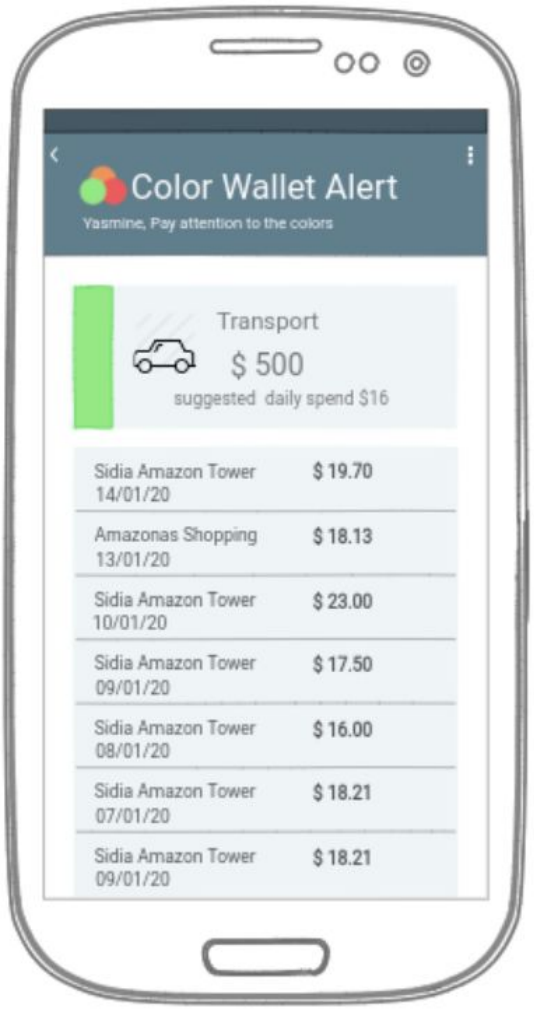
- The category name
- The value available to use
- The suggested daily spend and
- The category state (green, orange or red)

The categories state colors mean:

- Green: you have spent less than 50% of the goal
- Orange: you have spent between 50% and 80% of the goal
- Red: you have exceeded 80% of the goal and need to reduce spend

Suggested daily spend is calculated based on the available amount and number of days to month end.

Screen 3



When user clicks on detail icon in CWA board in the main screen, he/she is redirects to the category category detail, where he/she can see all spend ordered by date.

Screen 4

When the user clicks on the FAB in the main screen, he/she is redirected to this New Category screen.
The user needs to fill the fields with:

- category name,
- the target value and
- choose one icon.

After save the category the user returns to the main screen and the board is updates to show new category.

Screen 5

Every time the user makes a new spend, he/she can easily add a new spend click on the category. To add a new spend the user needs fill:

- Value of spend
- check/uncheck the save location radio button

The location is used to add information from where the spend was made without user fill one more field.

Screen 6

The widget will show the category that is closest to exceeding the goal.

Key Considerations

The app will be developed in which programming language?

App is written solely in the Java Programming Language.

How will your app handle data persistence?

The Color Wallet Alert will use Firebase Realtime Database to persists data.

Describe any edge or corner cases in the UX.

To add a new spend the user must click on the spend category and just fill the value field. Behind the scenes the app will record the date and location. This approach was adopted to decrease the amount of fields that financial app usually requires to register a new spend.

Describe any libraries you'll be using and share your reasoning for including them.

Firebase libraries to persist data and send messages in real time.
'com.google.firebase:firebase-core:16.0.8'
'com.google.firebase:firebase-database:16.0.4'
'com.google.firebase:firebase-messaging:17.3.4'

Google Play Service location to get the spend location
'com.google.android.gms:play-services-location:17.0.0'
'com.google.android.libraries.places:places:2.0.0'

Picasso to load icon categories
'com.squareup.picasso:picasso:2.5.2'

Gradle build tool
'com.android.tools.build:gradle:3.5.1'

Describe how you will implement Google Play Services or other external services.

Using the Firebase and location service as specified in section above. The Google Services libraries:
‘com.google.android.gms:play-services:11.6.0’
‘com.google.android.gms:play-services-location:17.0.0’
‘com.google.android.libraries.places:places:2.0.0’

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

The CWA project uses Java 8 as program language and gradle to store dependencies. To download dependencies you will need internet access, once dependencies will be downloaded automatically the first time you build the project.
Once dependencies are ready you can proceed to development following the next steps.
Suggested IDE: Android Studio 3.5 or higher.

Task 2: Add user permission

Internet, location

Task 2: Create a firebase project

Create and add support to firebase in CWA project follow the steps:

- Create CWA project
- Get the json config
- Add dependencies to gradle
- Grant required permission
- Create a Google Firebase Clien
-
- Implement the retrieve in real time from firebase for CWA board Activity
- Implement write firebase database to add new category and add new category spend

Task 2: Create a Google Places API Project

Create and add support to Google Places API in CWA project follow the steps:

- Create CWA project
- Get the API key and add to manifest
- Restrict the API key
- Add dependencies to gradle
- Grant required permission
- Create a Google APIclient
- Connect and use Google Play Services

Task 3: Implement UI for Each Activity and Fragment

Login Activity:

- Create an Empty Activity called LoginActivity and layout file
- In layout add field name and button
- Store login name in Shared Preferences
- Call the CWABoard Activity

CWABoard Activity

- Create an Empty Activity called CWABoardActivity and layout file
- Add a Collapsing ToolBar
- Add a RecyclerView to show all categories
- Retrieve user name from Shared Preferences
- Retrieve the Categories from Firebase
- Create a CWABoard item
- Add an action button to add a new category

CWA Board Item Fragment

- Create an Empty Fragment called CWABoardItemFragment and layout file
- Add a CardView to show category information
- Handle the New Spend Activity

New Spend Activity

- Create an Empty Activity called NewSpendActivity and layout file
- Add value fields and button
- Get User Location Information
- Save values in Firebase
- Return to CWABoard Activity

New CategoryActivity

- Add the fields and button
- Save values in Firebase
- Return to CWABoard Activity

Task 4: New Spend Value Validation

The New Spend Activity should prevent wrong input values, such as strings, special characters or negative value to avoid exception:

- Make sure use input number field in new_spend_activity.xml

Task 5: Update Category Information

To each category in CWA board the app should do some calculates the show the available amount, the suggested daily spend and the color status.

Available amount:

- Retrieve from firebase all spends of a current month grouped by category and store in **sumCategorySpend** variable
- Retrieve the category value goal and store in **categoryGoal** variable
- Subtract **sumCategorySpend** from **categoryGoal** and store in **categoryAvailableAmount** variable
- Update textView with **categoryAvailableAmount** variable value

Suggested daily spend

- Get the number of days until the month ends and store in **daysToMonthEnds** variable
- Divide **categoryAvailableAmount** variable by **daysToMonthEnds** and store in **suggestedDailySpend** variable
- Update textView with **suggestedDailySpend** variable value

Color Status

- Calculates the percentage of category goal was consumed and the set color according to:
    - Green: 50% of the goal was reached
    - Orange: 50% and 80% of the goal was reached
    - Red: 80% of the goal was exceeded

Task 6: Internationalization

Provide support to English and Portugues languages creating the correspondent string files.

- Create the string.xml default in English
- Create the string.xml pt_BR in Portuguese

Task 7: Handle Accessibility

To provide some accessibility features the app should implement some tasks:

- Add Content Description for images
- Create landscape layout
- Create tablets layout
- Adjust the size according to device size

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"