GitHub Username: Fulmen545

Meal Tracker

Description

Have you ever wonder when was the last time you ate some specific food? Or do you need to track what you eat for fitness or health purpose? Then Meal Tracker is app for you. With Meal Tracker you can write down what you ate and the app will store it in your Google calendar so you can check it even on devices where you don't have the app installed.

Intended User

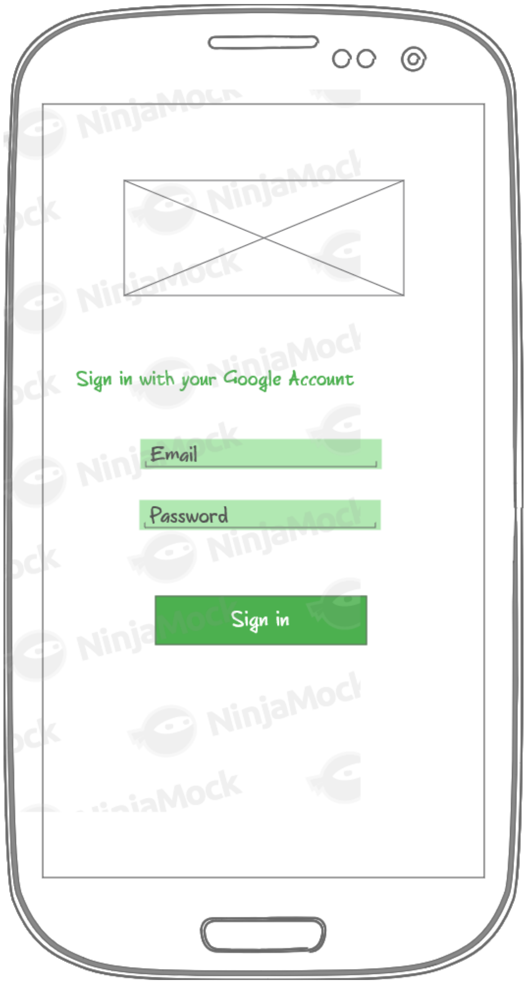
Fitness nuts or people with stomach issues who need to track what they eat.

Features

- Saves meal information in Google Calendar
- Offers history of recorded meals
- Shows last meal eaten in a widget
- Offers customizable fields (user can name new fields to describe food - location, calories, vegan, ….)
- Color customization for different meals

User Interface Mocks

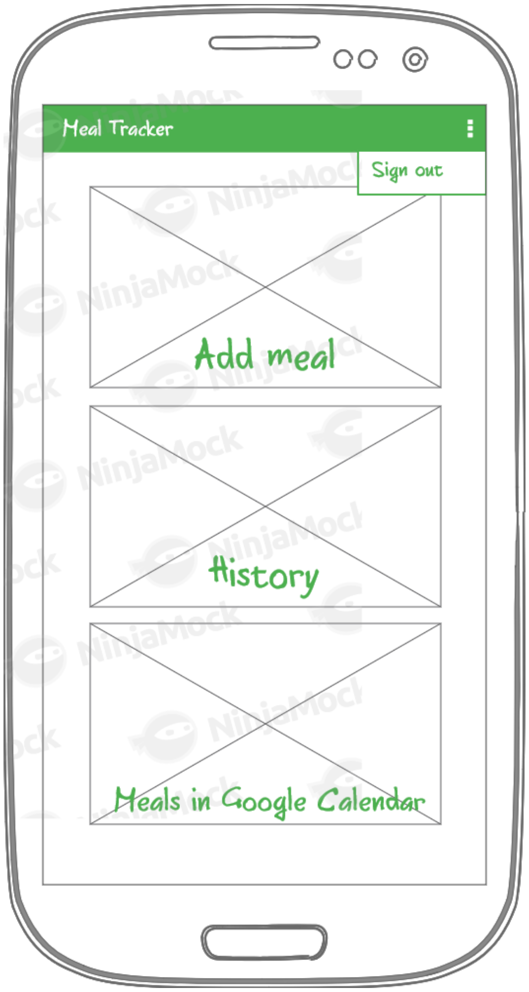
Screen 1



Start screen

Screen contains Logo (currently placeholder) and two input fields for Google email and password. After user fills credentials and clicks on Sign In button app will redirect user to Main screen

Screen 2



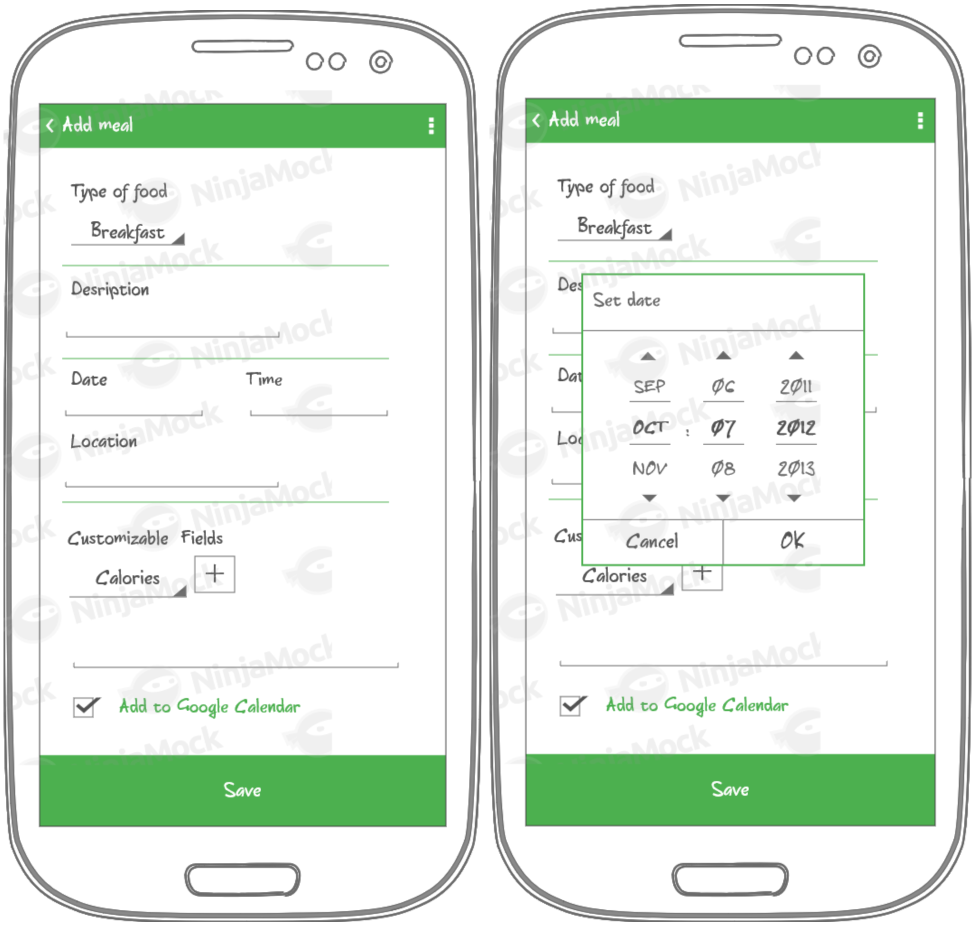
Main Screen

On Main Screen will be displayed three options.

1. Add new meal to the app
2. Check all meals he added to the app
3. Check meals user decided to add to Google Calendar

Every option will have some stock picture (currently placeholder) and list will be displayed as receipt names were in Baking app.

Screen 3



Add meal screen

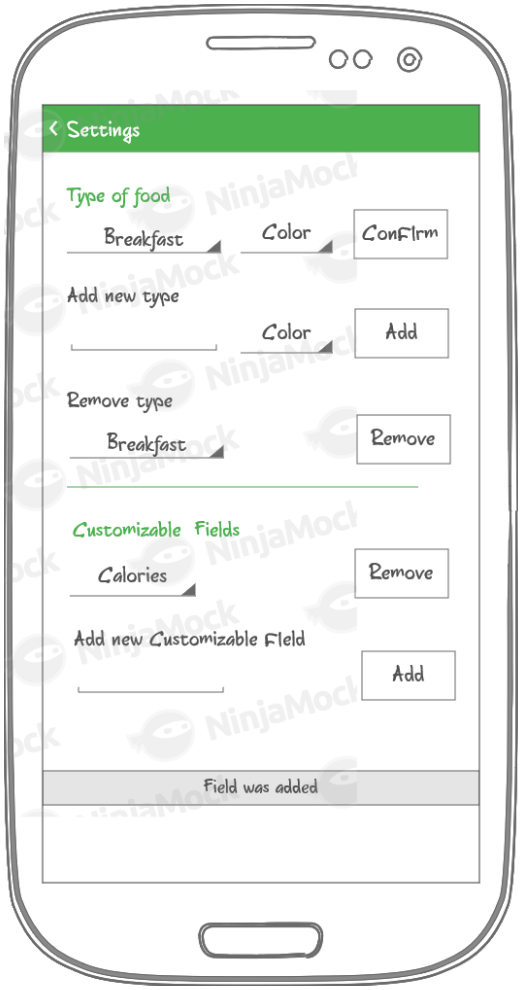
On Add meal screen user will fill all the details to for saving meal like Type of food, Description, Date, Time and user can also choose to add customizable field. When user clicks on + button next to Cust. fields spinner, Input field for chosen field will be displayed.

At the bottom of the screen is checkbox. When checked meal will be stored to Google Calendar as well (not just to internal database)

When user click on Date/Time input fields Date/Time picker dialog will be displayed.

Action bar will also offer option Settings where user can add/remove customizable fields.

Screen 4



Field was added

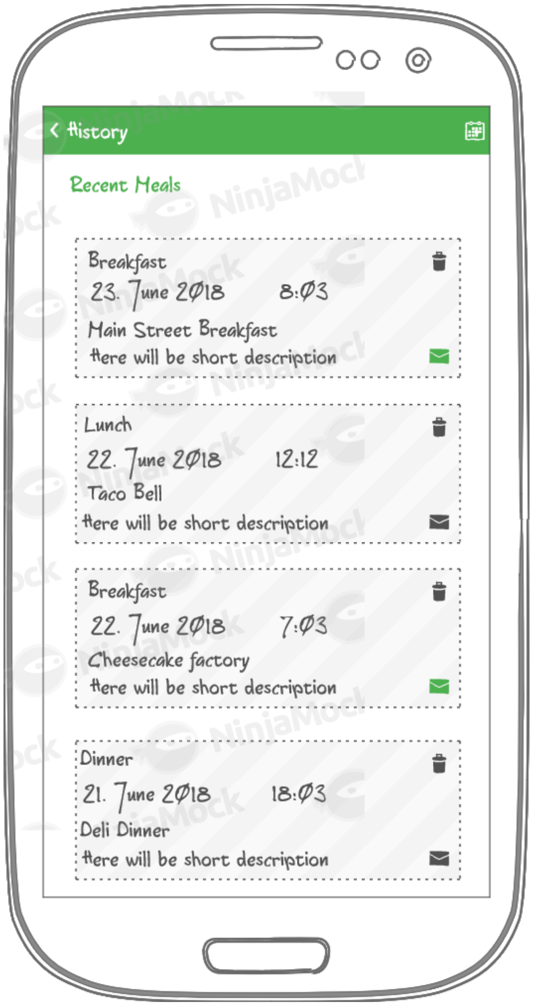
Settings screen

In this screen user can customize fields. Add specific color to the Type of food, remove add completely new type.

User can also add or remove Customizable fields.

When field is added/removed Toast message is displayed.

Screen 5



Recent Meals

Breakfast
23. June 2018 8:03
Main Street Breakfast
Here will be short description

Lunch
22. June 2018 12:12
Taco Bell
Here will be short description

Breakfast
22. June 2018 7:03
Cheesecake factory
Here will be short description

Dinner
21. June 2018 18:03
Deli Dinner
Here will be short description

History screen

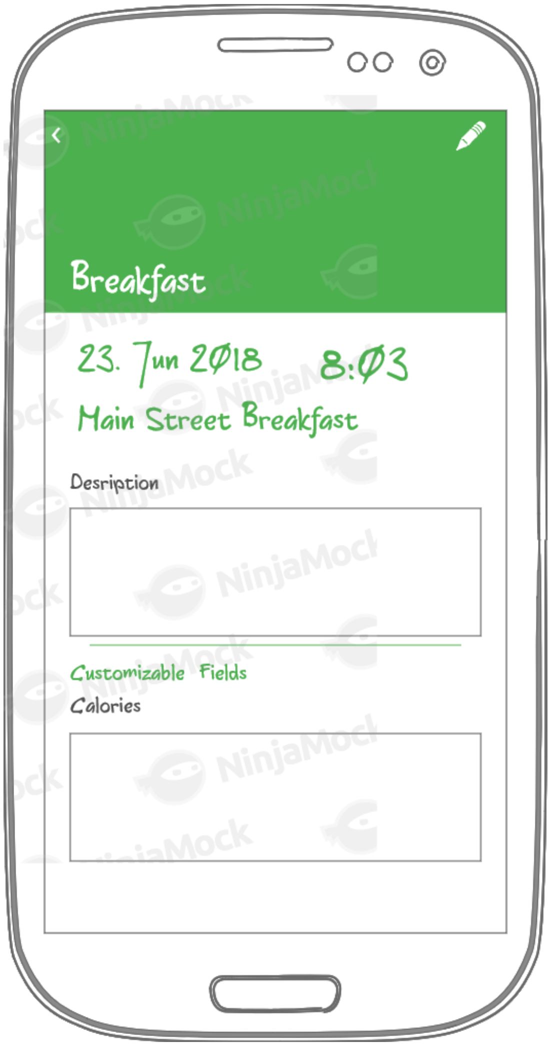
User gets here when he clicks on History option in the Main Screen. In this screen user will see his recent meals. Every item will have different color depending on type of food (couldn't figure out how to change color of clickable field in ninjamock). Item will have on 4 details since this is just overview - Type of food, date, time and description.

User will also have option to remove the item from the database by clicking on Trash icon in top right corner of the item. Envelope (icon in bottom right corner) indicates if meal was sent to Google calendar (Primary color - sent; Gray color - not sent). After clicking on Envelope icon meal will be sent/removed from the Google calendar

At the right side of action toolbar is calendar icon. After clicking on it date picker dialog is displayed. This gives user option to display meals for specific day.

After clicking on the item user is redirected to Meal detail screen

Screen 6



Breakfast

23. Jun 2Ø18 8:Ø3

Main Street Breakfast

Desription

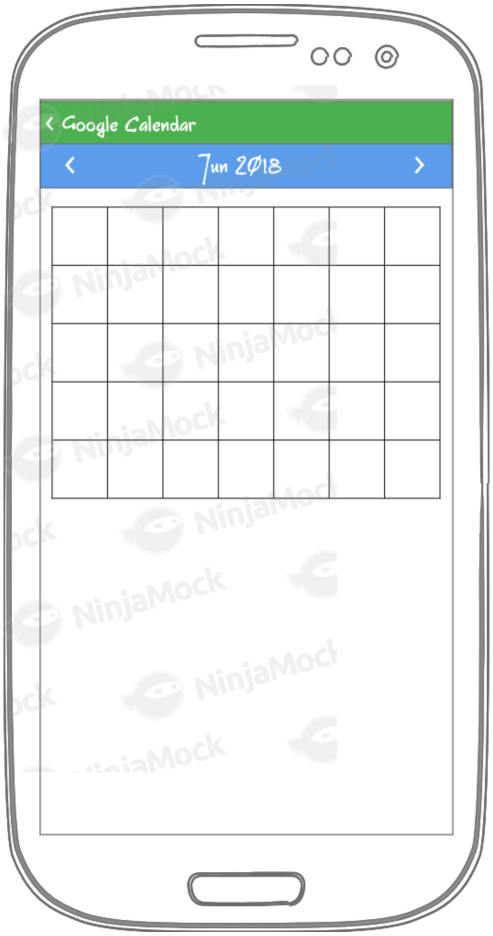
Customizable Fields

Calories

Meal Detail screen

In this screen will be displayed all the data user entered for the meal. Screen will have collapsing toolbar. User can also edit the meal. After he clicks on pen icon (top right corner on the toolbar) app will redirect user to Edit meal screen which is actually Add meal screen but with prefilled fields with data for the edited meal. User will be able to change data or add/remove fields.

Screen 7

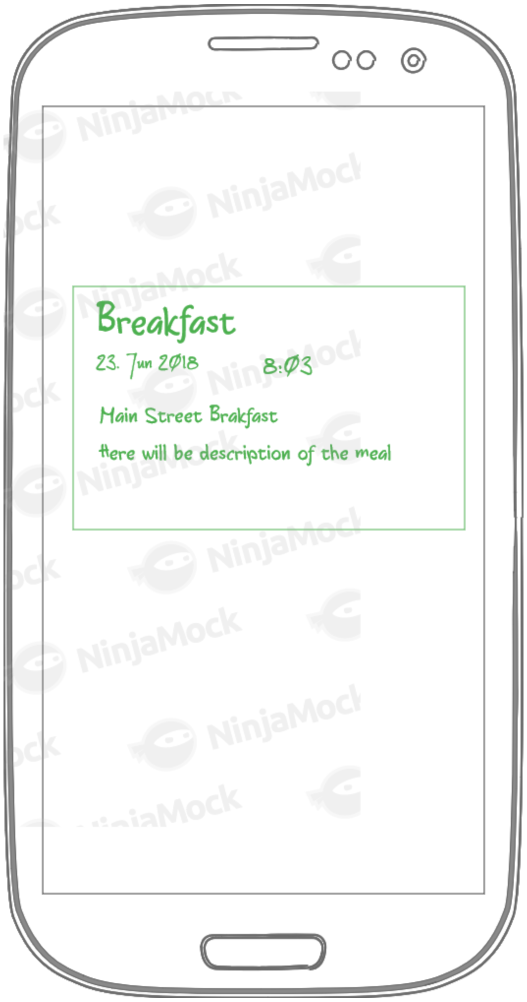


Google Calendar screen

User gets here when he clicks on Meals in Google Calendar option in the Main Screen. This screen displays calendar element (couldn’t find one in ninjamock so I used table for the mock)

When user click on the date under action toolbar date picker dialog will be displayed and user can choose specific date. This will redirect user to History screen but only Meal in Google calendar will be displayed. The same happens when user click on the day in calendar element.

Widget



Breakfast

23. Jun 2018 8:03

Main Street Brakfast

Here will be description of the meal

In the widget will be only 4 details - Type of food, date, time and description. It will be details of the most recent food. After clicking on the widget user will be redirected to the Meal Detail screen

Key Considerations

How will your app handle data persistence?

The app will have local database to store customized fields and meals. To retrieve this data Content provider will be used and also Loader to move the data to its views.
Data from/to Google will be retrieved/inserted via Get/Post Requests using an AsyncTask.
Application will be written solely in the Java Programming Language and will enable RTL layout switching on all layouts.

Resources like colors, strings, themes will be stored in XML resource files (colors.xml, styles.xml, ...). The data user input for meals will be stored in local database.

For accessibility purposes all important elements will have content descriptions so when users with vision impairments will have enabled TalkBack (Sony) function in their device, they will be able to navigate through the app.

Describe any edge or corner cases in the UX.

In the Main screen when user click two times on Android back button the app will close. Only way to get to Start (Login) screen is to Sign out.

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

1. Google Calendar api com.google.apis:google-api-services-calendar
   <version>v3-rev87-1.19.0</version>
   - So the app can work with Google Calendar
2. Butter Knife 'com.jakewharton:butterknife:8.8.1'
   - Field and method binding for Android views
3. Android studio 3.0.1
4. Com.android.tools.build:gradle:2.3.0
5. Com.android.support:recyclerview-v7:26.+
6. com.android.support:support-v4:26.+

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

Google Account Login com.google.android.gms:play-services-auth:15.0.1

- This will allow user to get Calendar ID

Google Location com.google.android.gms:play-services-location:15.0.1

- This will allow user to add location to meal item

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

Subtasks:

- Create project in Android studio
- Configure libraries
- Set up project in github

Task 2: Start (Login) screen

Subtasks:

- Build UI for Start screen
- Set up Google Account Login service
- Handle error cases

Task 3: Main screen

Subtasks:

- Find free stock pictures for the list
- Create layout for Main screen
- Use RecyclerView for the list
- Finish rest of functionality for the screen

Task 4: Add Meal screen

Subtasks:

- Create layout for Add Meal screen
- Set up database with content provider
- Finish rest of functionality for the screen
- Handle error cases

Task 5: Settings screen

Subtasks:

- Create layout for Settings screen
- Set up insert, delete and update for settings in the database through content provider
- Finish rest of functionality for the screen
- Handle error cases

Task 6: Google Calendar screen

Subtasks:

- Create layout for Google calendar screen
- Set up Google calendar functionality with Google Calendar API
- Finish rest of functionality for the screen

Task 7: History screen

Subtasks:

- Create layout for History screen
- Set up select and delete for meals in the database through content provider
- Finish rest of functionality for the screen

Task 8: Detail screen

Subtasks:

- Create layout for Detail screen
- Set up select and update for specific meals in the database through content provider
- Add functionality for Add meal screen so it can be used for editing meals as well
- Finish rest of functionality for the screens

Task 9: Widget

Subtasks:

- Create layout for Widget
- Implement functionality for Widget

Task 10: Overall functionality refinement

Subtasks:

- Handle error cases
- Handle screen resolution changes (Portrait, Landscape)
- Checks for internet connection