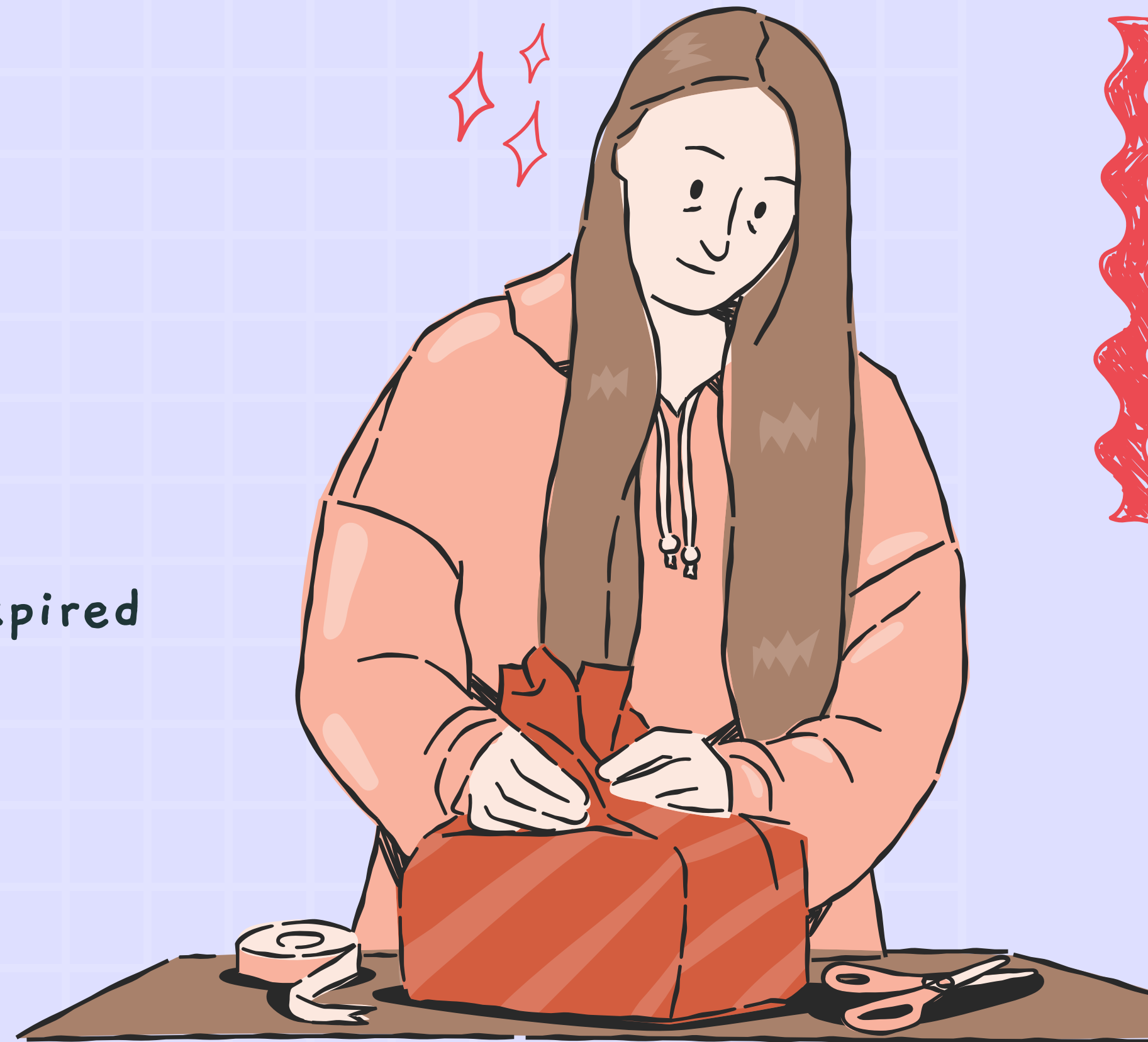


# RYFABE



Remind Your Foods And Beverages Expired

Angeline Felicia  
Kelas A - Flutter



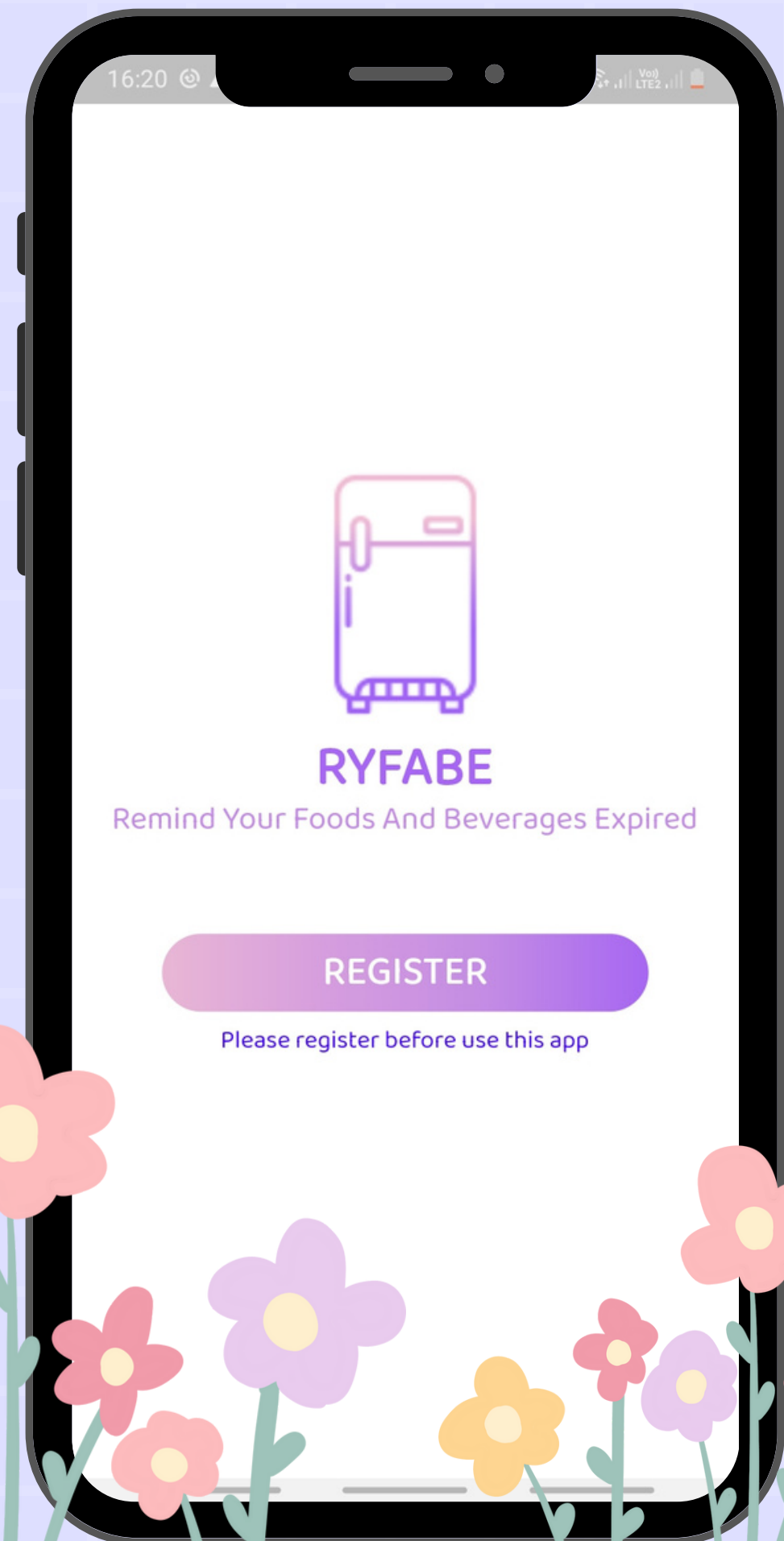
# Latar Belakang Pembuatan Proyek

- Sering lupa tanggal expired makanan yang ada di rumah.
- Berujung banyak makanan yang dibuang ke tempat sampah dikarenakan sudah tidak layak dikonsumsi.



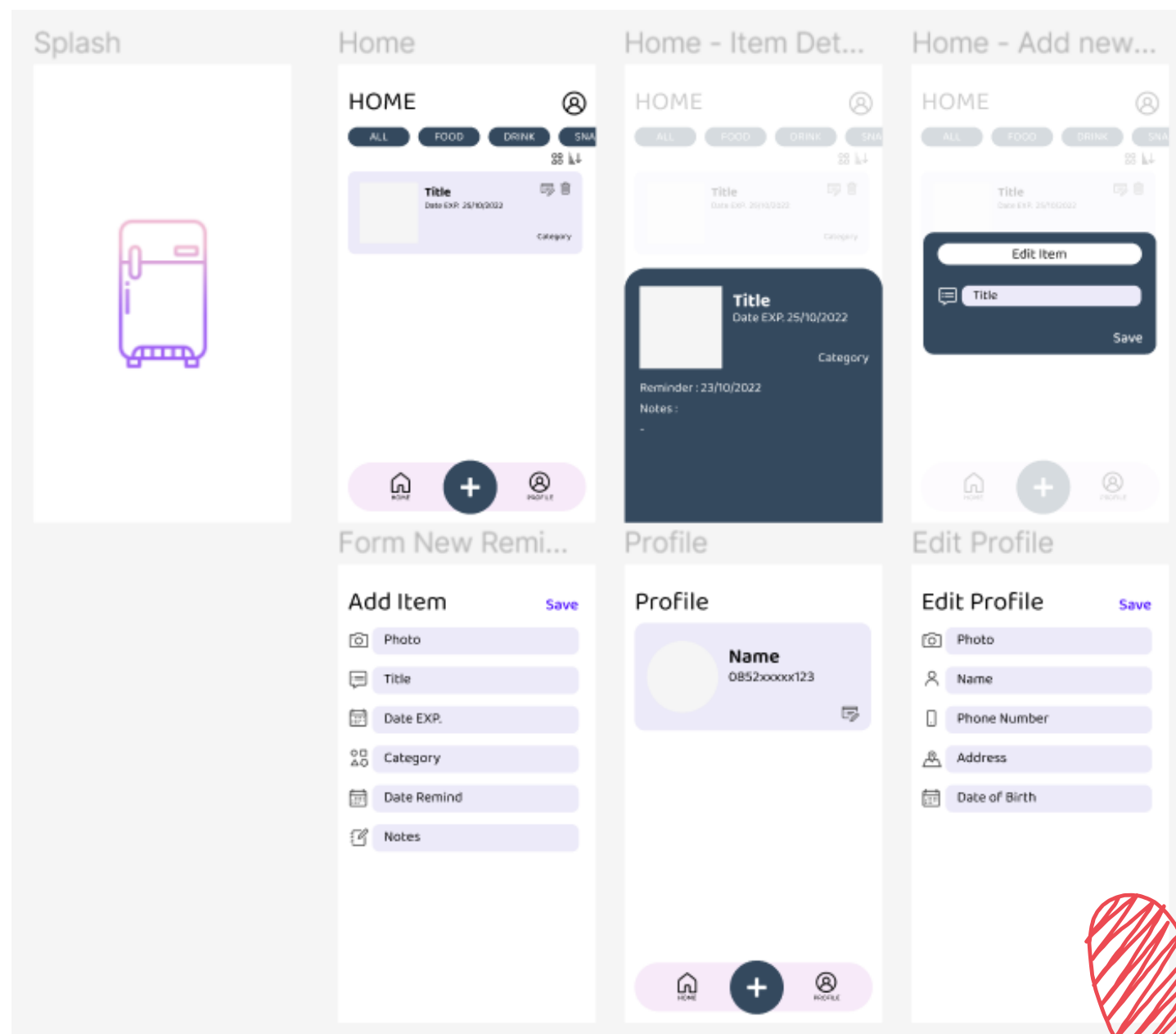
# RYFABE

Merupakan aplikasi pengingat tanggal expired makanan dan minuman.



# Desain Figma

<https://www.figma.com/file/64QrEl6ZOBS4bVQI8g2hR/RyFABE?node-id=0%3A1>



## Color Palette

FFFFFF

F7EAF9

ECEAF9

34495E

4200FF

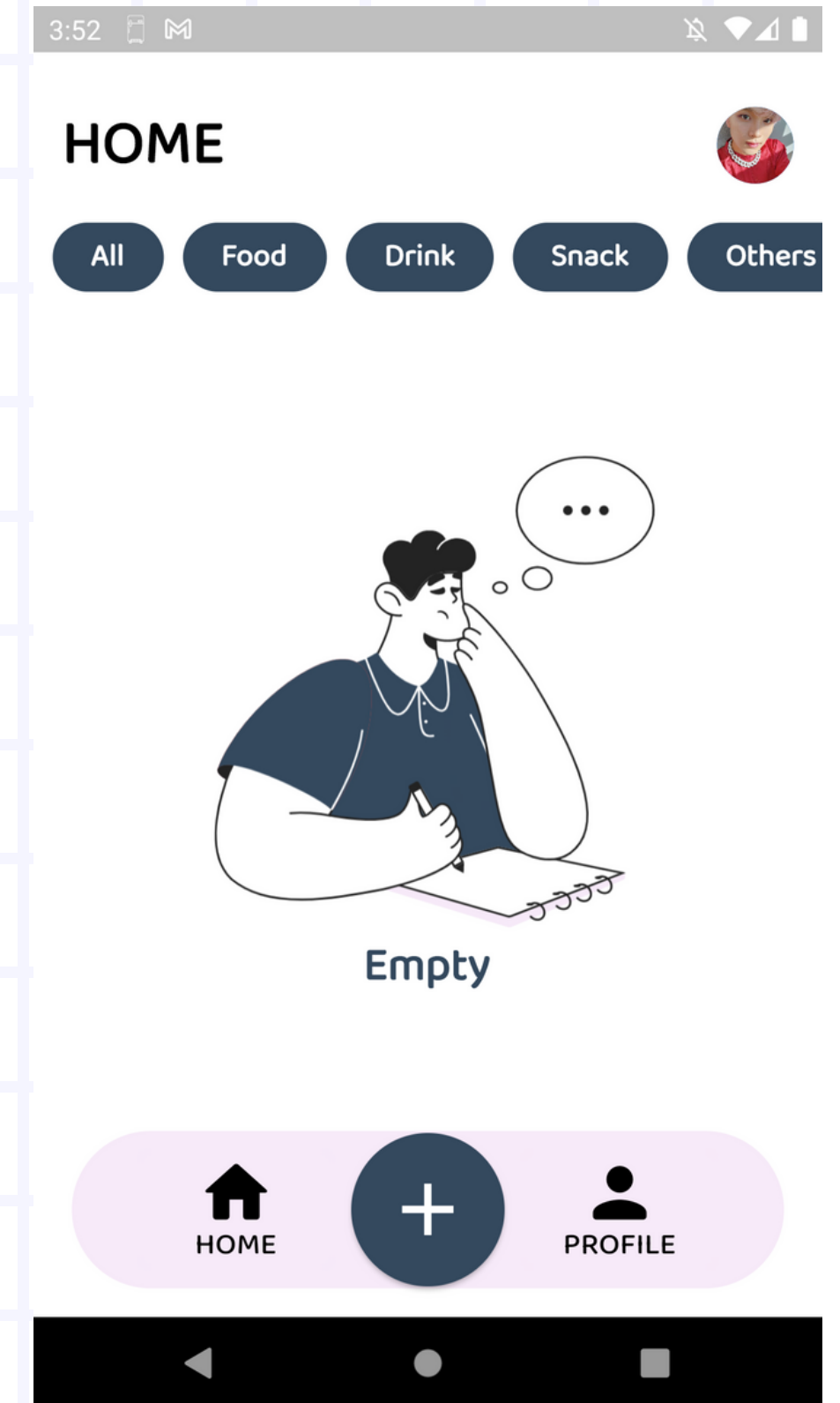
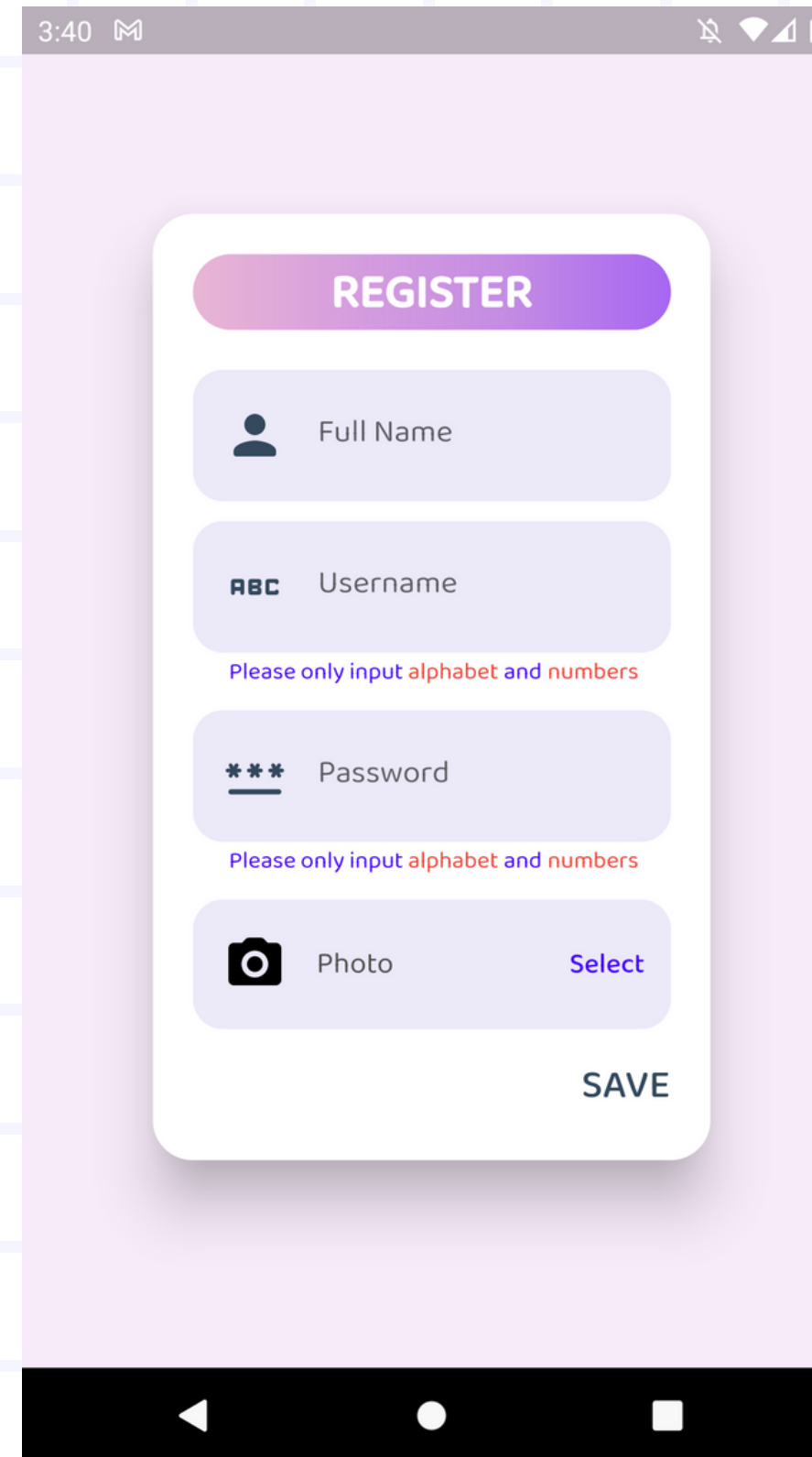
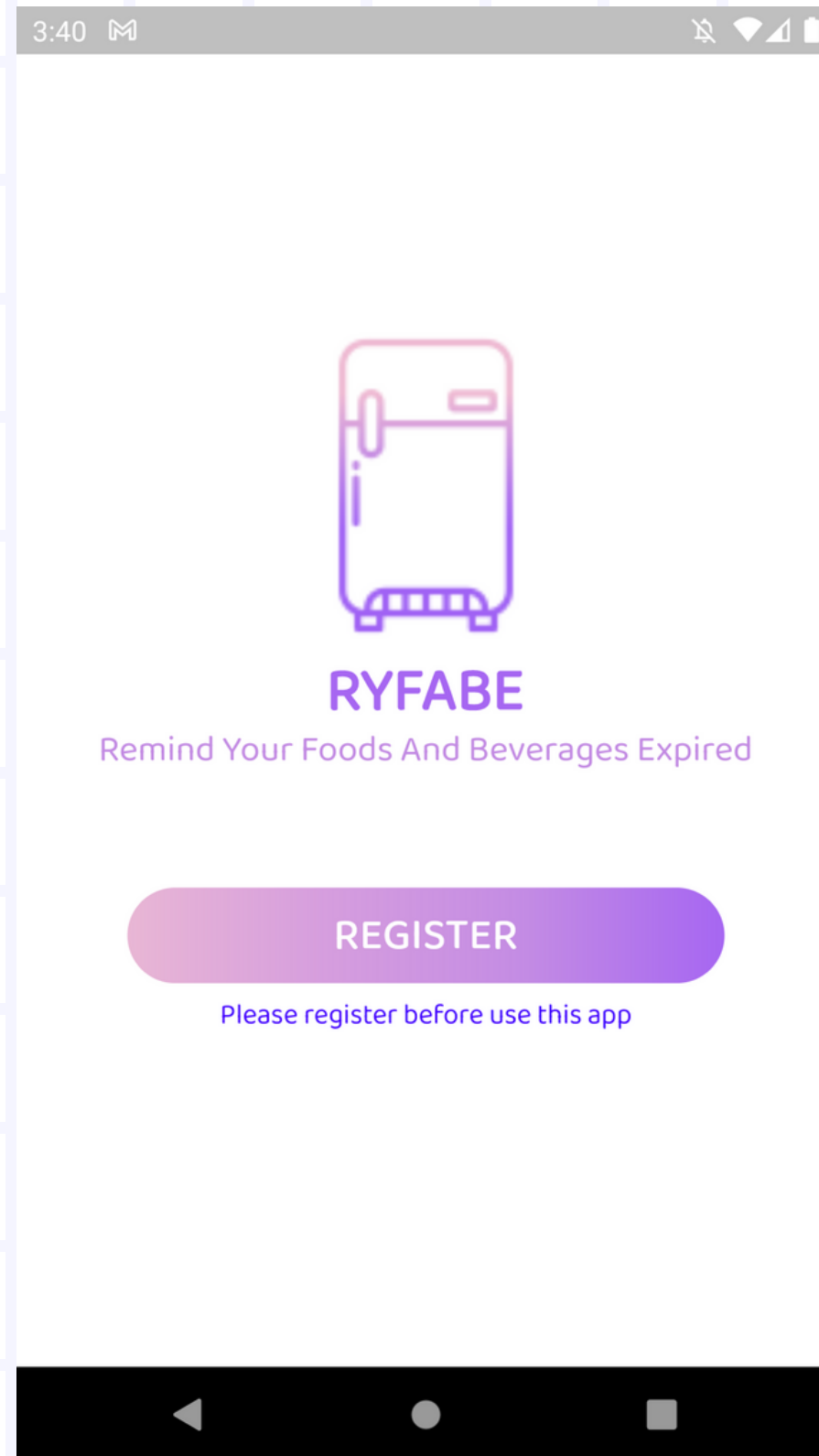
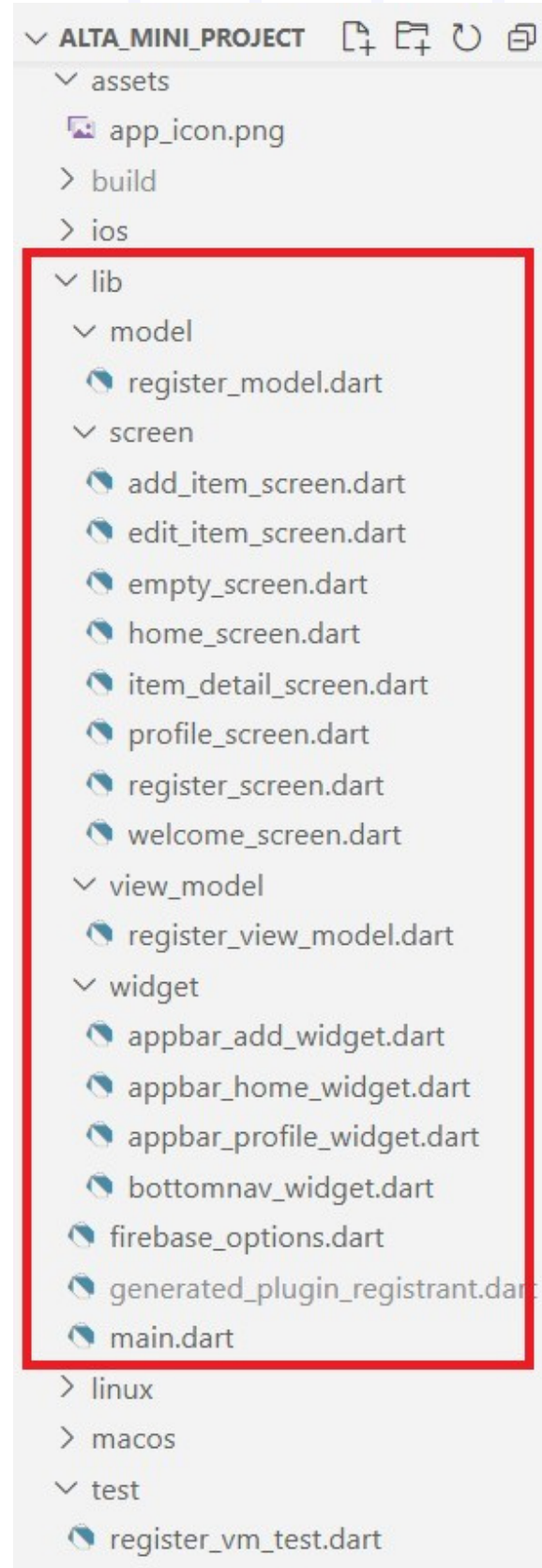




# Tools yang digunakan



# Struktur Program



# Struktur Program

3:42

< Add Item

SAVE

Photo Select

Title

Date Expired 07/11/2022 Select

Category Food

Date Reminder 07/11/2022 3:42 PM

Notes

3:51

HOME

All Food Drink Snack Others

Biskuit Coklat  
Date EXP. 08/11/2022  
Food

Susu Bendera  
Date EXP. 23/11/2023  
Drink

HOME + PROFILE

3:52

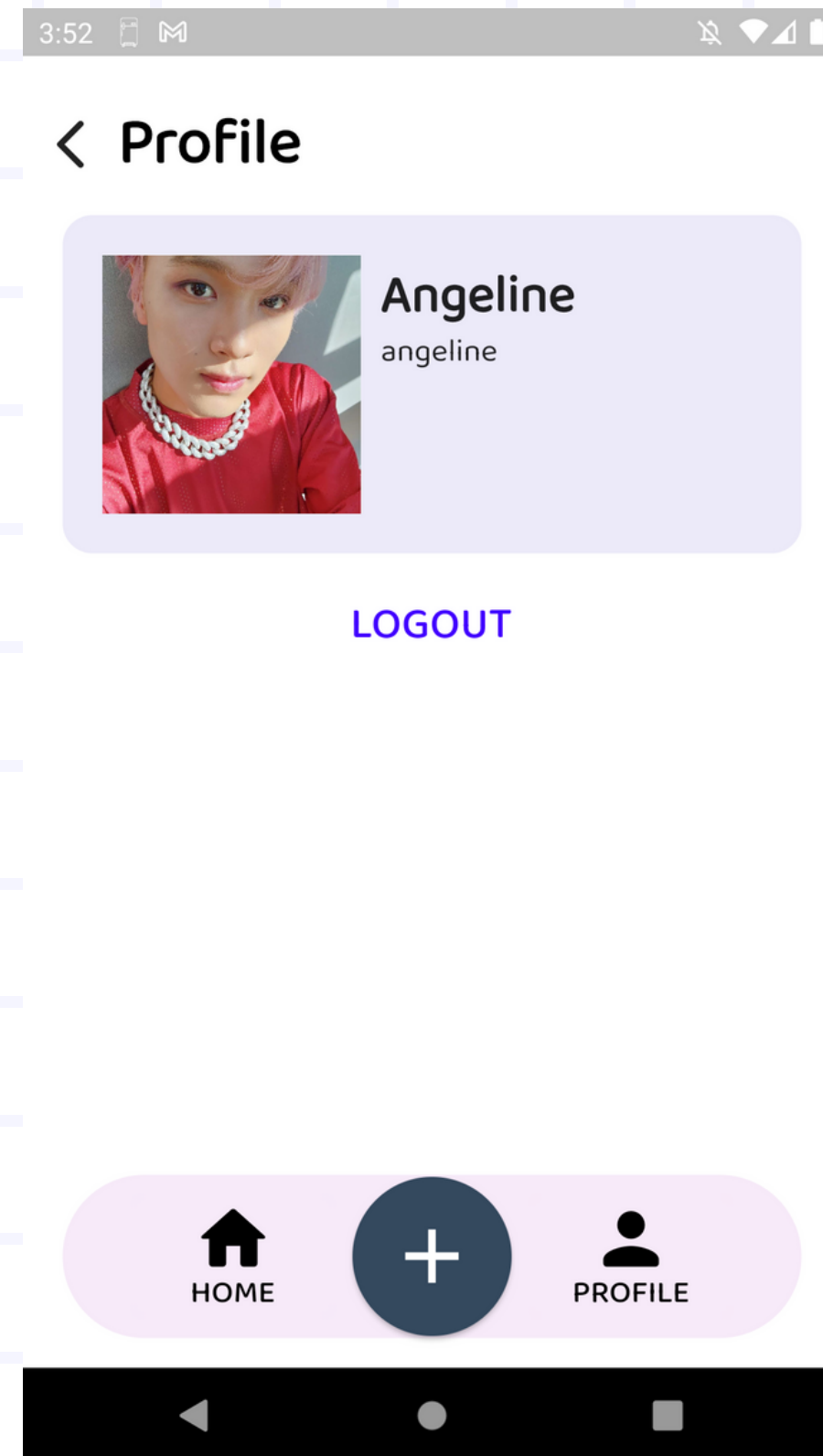
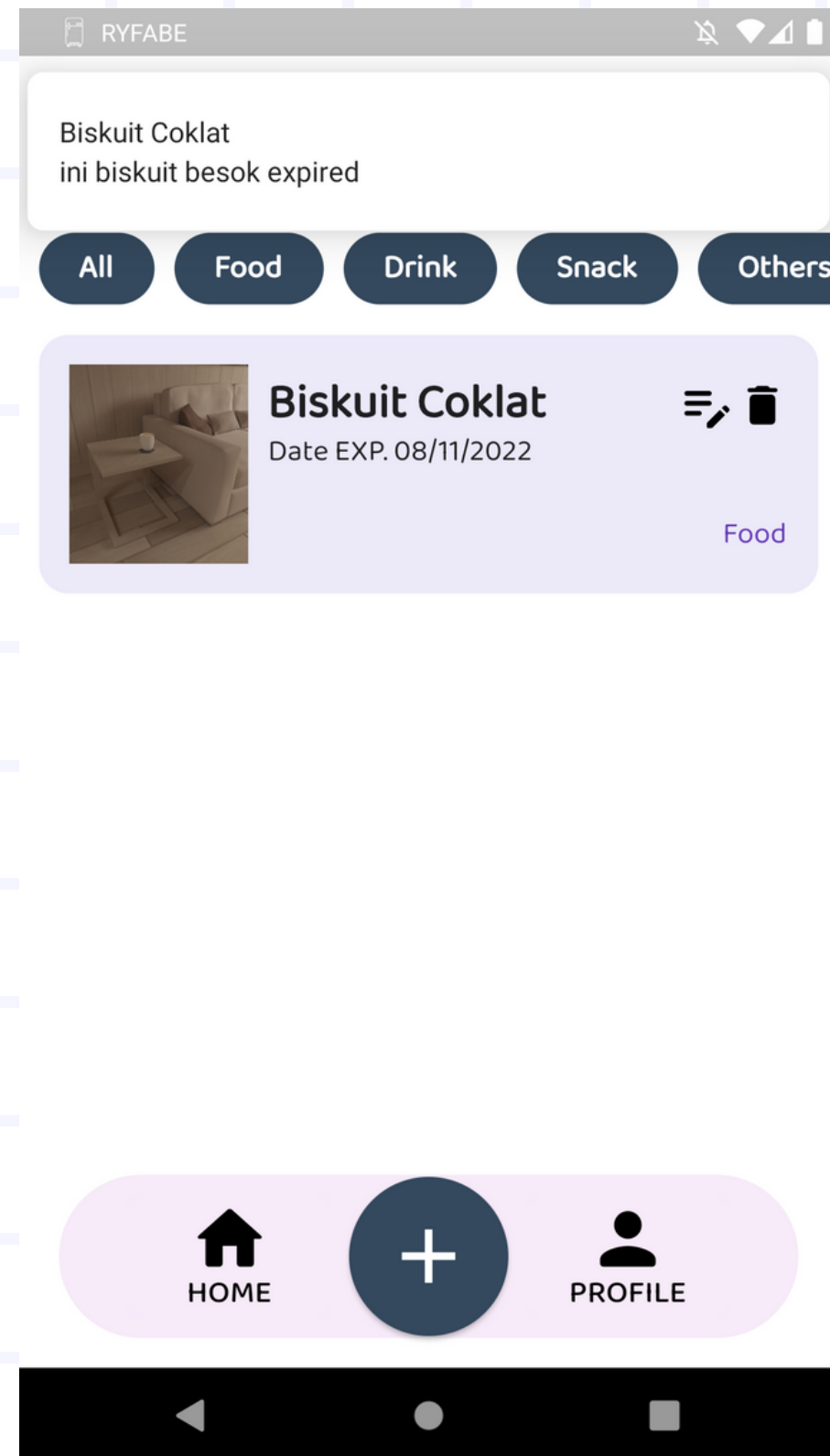
HOME

All Food Drink Snack Others

Susu Bendera  
Date EXP. 23/11/2023  
Drink

HOME + PROFILE

# Struktur Program





# Flow Program

```
Widget btnSave() {
  return Row(
    mainAxisAlignment: MainAxisAlignment.end,
    children: [
      TextButton(
        onPressed: () async {
          if (imageUrl.isEmpty) {
            ScaffoldMessenger.of(context).showSnackBar(
              const SnackBar(content: Text("Photo Required")));
          }
          return;
        },
      ),
      if (formKey.currentState!.validate()) {
        // push notification
        int uniqueId =
          DateTime.now().millisecondsSinceEpoch.reminder(100000);

        await AwesomeNotifications().createNotification(---
          print("notification id: $uniqueId");

        // make item
        final item = <String, dynamic>{
          "photo": imageUrl,
          "title": _titleController.text,
          "date_exp": _expDate,
          "category": categoryValue,
          "date_remind": _remindDate,
          "time_remind": _remindTime.format(context).toString(),
          "notification_id": uniqueId,
          "notes": _notesController.text,
        };

        // add to firebase with random unique id
        db.collection(dbCollection).add(item).then(
          (DocumentReference doc) =>
            print('DocumentSnapshot added with ID: ${doc.id}'));

        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => const HomeScreen(),
          ), // MaterialPageRoute
        );
      },
      child: const Text(
        "SAVE",
        style: TextStyle(
          color: darkBlue,
          fontSize: 18,
        ), // TextStyle
      ), // Text
    ], // TextButton
  ); // Row
}
```

```
// =====
body: Consumer<RegisterViewModel>(
  builder: (context, RegisterViewModel data, child) {
    if (data.getDatas.isNotEmpty) {
      var registerData = data.getDatas[0];
      nameProfile = registerData.name;
      usernameProfile = registerData.username;
      passwordProfile = registerData.password;

      dbCollection = 'items_${usernameProfile}_${passwordProfile}';
    }

    return StreamBuilder<QuerySnapshot<Map<String, dynamic>>>(
      // change order + category
      stream: streamDatabase(),
      builder: (context, snapshot) {
        // loading data..
        if (snapshot.connectionState == ConnectionState.waiting) {
          return const Center(
            child: CircularProgressIndicator(),
          ); // Center
        }

        // data error
        if (snapshot.hasError) {
          return const Center(
            child: Text("Error"),
          ); // Center
        }

        // take data
        var data = snapshot.data!.docs;

        if (data.isEmpty) {
          return const EmptyScreen();
        }

        return ListView.builder(
          itemCount: data.length,
          itemBuilder: (context, index) {
            // change timestamp to string (date exp.)
            Timestamp exptime = data[index].data()['date_exp'];
            DateTime expDate = exptime.toDate();

            // change timestamp to string (date remind.)
            Timestamp remindtime = data[index].data()['date_remind'];
            DateTime remindDate = remindtime.toDate();

            return GestureDetector(
              child: Container(
                margin: const EdgeInsets.only(
                  left: 10,
                  right: 10,
                  bottom: 10,
                ), // EdgeInsets.only
                padding: const EdgeInsets.all(15),
                decoration: const BoxDecoration(
                  borderRadius: BorderRadius.all(Radius.circular(15)),
                  color: lilac,
                ), // BoxDecoration
            );
          },
        );
      },
    );
  },
);
```

```
// third section
Column(
  mainAxisAlignment: MainAxisAlignment.end,
  crossAxisAlignment: CrossAxisAlignment.end,
  children: [
    Row(
      children: [
        // edit
        MaterialButton(
          onPressed: () {
            showDialog(
              context: context,
              builder: (context) {
                return EditItemScreen(
                  idData: data[index].id,
                  imageUrlData:
                    data[index].data()['photo'],
                  titleData:
                    data[index].data()['title'],
                  expdateData: data[index]
                    .data()['date_exp'],
                  categoryData: data[index]
                    .data()['category'],
                  reminddateData: data[index]
                    .data()['date_remind'],
                  remindtimeData: data[index]
                    .data()['time_remind'],
                  notifIdData: data[index]
                    .data()['notification_id'],
                  notesData:
                    data[index].data()['notes'],
                ); // EditItemScreen
              },
            );
          },
          padding: EdgeInsets.zero,
          materialTapTargetSize:
            MaterialTapTargetSize.shrinkWrap,
          minWidth: 10,
          child: const Icon(
            Icons.edit_note_rounded,
            color: black,
            size: 30,
          ), // Icon
        ), // MaterialButton
      ],
    ),
  ],
);
```

```
// delete
MaterialButton(
  onPressed: () async {
    // delete notification
    await AwesomeNotifications()
      .cancelSchedule(data[index]
        .data()['notification_id'])
      .then((value) =>
        print("Schedule deleted"));

    // delete firebase
    db
      .collection(dbCollection)
      .doc(data[index].id)
      .delete()
      .then(
        (value) =>
          print("Document deleted"),
        onError: (e) => print(
          "error delete document: $e"),
      );
  },
  padding: EdgeInsets.zero,
  materialTapTargetSize:
    MaterialTapTargetSize.shrinkWrap,
  minWidth: 10,
  child: const Icon(
    Icons.delete_rounded,
    color: black,
    size: 26,
  ), // Icon
), // MaterialButton
,
```



# Product Success

- Push Notification
- Firebase CRUD
- Sort by Category

## Saran

- Login / Register



# Thank You

Berikutnya penjelasan kodingan dan running app secara langsung