

Contents

Table of Figures	2
Introduction	4
Unique screens	4
Aimed Users	5
Data Flow Diagram (DFD)	5
Sequence Diagram	5
User Opens Landing Page:	Error! Bookmark not defined.
Login/Registration:	Error! Bookmark not defined.
Login:	Error! Bookmark not defined.
Initiate Product Search:	Error! Bookmark not defined.
Get Product Category Info:	Error! Bookmark not defined.
Checkout:	Error! Bookmark not defined.
Use Cases	Error! Bookmark not defined.
Software Design Decisions	6
Libraries Used and Code Screenshots	6
Social, Legal, Ethical and Security Issues	12
Social Issues:	12
Legal Issues:	13
Ethical Issues:	14
Security Issues:	15
Recommendations for better Social, Legal, Ethical and Security Issues:	Error! Bookmark not defined.
Usage of Application	Error! Bookmark not defined.
Development Strategy	15
Phases	16
Sprint Goal:	16
Phase-1:	16
Phase-2:	18
Phase-3:	19
Phase-4:	19
Burn Down Charts	20

Sprint 1 Burn-down.....	20
Sprint 2 Burn-down.....	21
Sprint 3 Burn-down:	21
Sprint 4 Burn-down:	22
Firebase DB at backend and regarding API authentication and searching	22
GIT	25
1 st commit	25
2 nd commit	28
3 rd commit.....	30
Grocery Delivery Mobile Application (UI).....	32
Reference.....	38

Table of Figures

Figure 1 Data Flow Diagram	5
Figure 2 Sequence Diagram	6

Libraries Used and Code Screenshots

Figure 3 AppConfig Code & Libraries.....	6
Figure 4 Screen routing Code & Libraries	7
Figure 5 LoginScreen UI Code & Libraries.....	8
Figure 6 ReviewScreen UI Code & Libraries	8
Figure 7 SplashScreen UI Code & Libraries	9
Figure 8 WishlistScreen UI Code & Libraries.....	10
Figure 9 ApiClass utils Code & Libraries	11
Figure 10 ApiService utils Code & Libraries	11
Figure 11 OutlineFormField utils Code & Libraries	12

Development Strategy

Figure 12 Sprit Goal	16
Figure 13 Phase 1 (i).....	17
Figure 14 Phase 1 (ii)	17
Figure 15 To Do Sprint-3.....	18

Figure 16 Phase 2.....	18
Figure 17 Phase 3.....	19
Figure 18 Phase 4.....	20

Burn Down Charts

Figure 19 Sprint 1 Burn-down	20
Figure 20 Sprint 2 Burn-down	21
Figure 21 Sprint 3 Burn-down	21
Figure 22 Sprint 4 Burn-down	22

Firebase DB at backend and regarding API authentication and searching

Figure 23 Firebase Project home page	22
Figure 24 Grocery application read and write graph in firebase	23
Figure 25 Authentication screen in firebase	23
Figure 26 Authentication option in firebase	23
Figure 27 user list in firebase log in	24
Figure 28 Cloud Firestore	24
Figure 29 Users and Add document in DB Cloud firestore	24
Figure 30 Start collection option	25

GIT

Figure 31 1st commit.....	25
Figure 32 1st commit I	26
Figure 33 1st commit II	26
Figure 34 1st commit III.....	27
Figure 35 1st commit III.....	27
Figure 36 2nd commit I	28
Figure 37 2nd commit II.....	28
Figure 38 2nd commit III	28
Figure 39 2nd commit IV	29
Figure 40 2nd commit V	29
Figure 41 2nd commit VI	30
Figure 42 3 rd commit I.....	30
Figure 43 3rd commit II	31
Figure 44 3rd commit III	31
Figure 45 3rd commit IV.....	31

Grocery Delivery Mobile Application (UI)

Figure 46 Welcome Page	32
Figure 47 Registration Page	32

Figure 48 Login Page	33
Figure 49 Home Page	33
Figure 50 Product Detail Page	34
Figure 51 Product added to Cart	34
Figure 52 Product added to Cart	35
Figure 53 Payment and Placed Order	35
Figure 54 Wish List	36
Figure 55 Review and Feedback.....	36
Figure 56 Thanks for Feedback	37
Figure 57 Log out Screen	37

Grocery Delivery Application

Introduction

The main goal of the report on the design of a mobile grocery delivery application is to offer a simple and easy-to-use platform that lets consumers purchase groceries online with just a few clicks, pay for them online, and track their deliveries from there.

Unique screens

Home screen

Grocery details screen

In Stock and Arriving Soon Item Details

Error handling screen for any error

Aimed Users

- Customer

Data Flow Diagram (DFD)

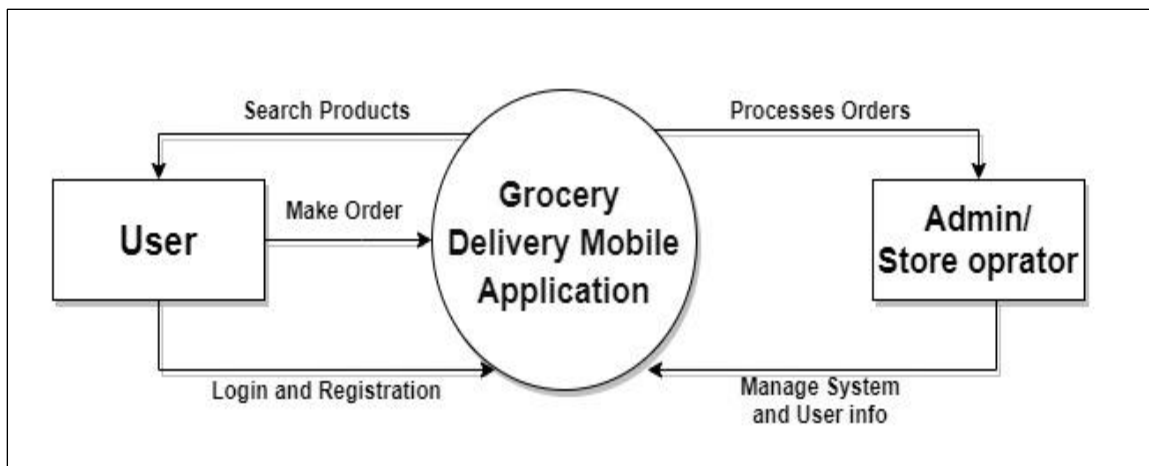


Figure 1 Data Flow Diagram

Sequence Diagram

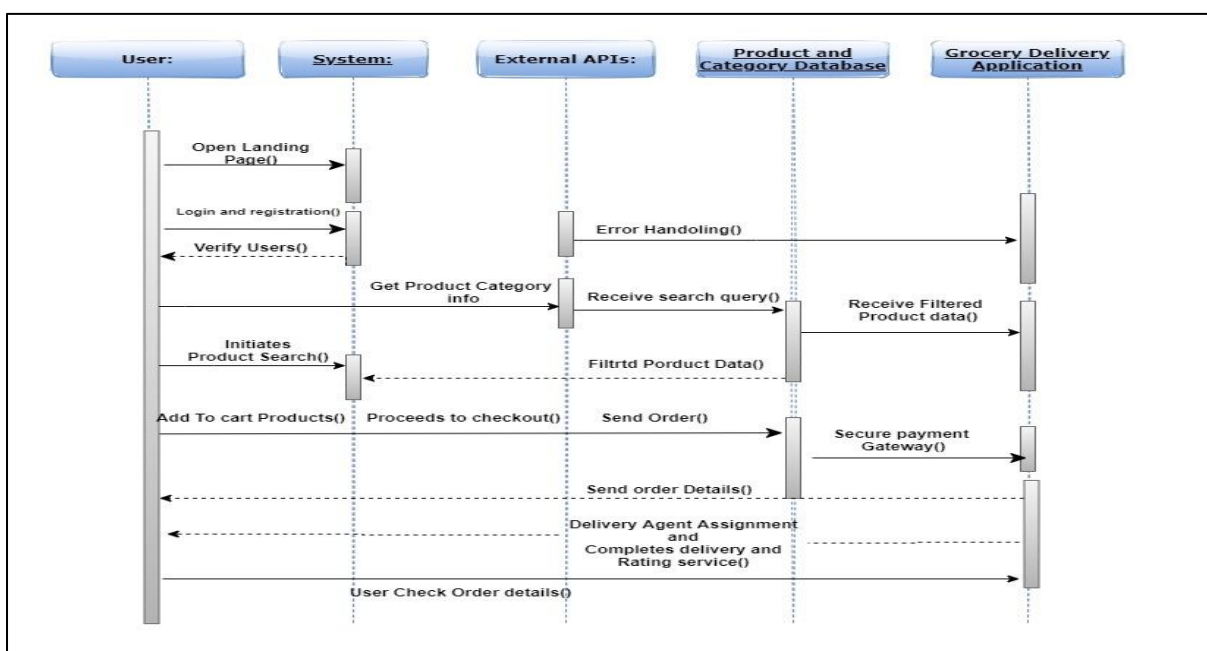


Figure 2 Sequence Diagram

Software Design Decisions

The application was built using Android Studio, and subsequently tested on an emulator. It's being developed using Java as the primary programming language.

User Experience:

The app prioritizes student users due to the breadth of functionalities available to them. Features for tutors will be rolled out subsequent to those for students. If additional time permits, plans include adding a calendar and notes section tailored for tutors.

Integration of 3rd Party Libraries:

The app seamlessly incorporates the Google Books API, empowering users to search for subject-relevant books.

Data Persistence:

Profile pictures of users are securely stored using Firebase Storage.

Libraries Used and Code Screenshots

```
package com.groceryapp.application

import android.app.Application
import com.groceryapp.ui.model.CartModel

class AppConfig : Application() {
    companion object {
        val cartList = ArrayList<CartModel>()
    }

    override fun onCreate() {
        super.onCreate()
    }
}
```

Figure 3 AppConfig Code & Libraries

```

package com.groceryapp.routing

sealed class Screen(val route: String) {

    object SplashScreen: Screen("splash_screen")
    object LoginScreen: Screen("login_screen")
    object RegisterScreen: Screen("register_screen")
    object MainScreen: Screen("main_screen")
    object WishlistScreen: Screen("wishlist_screen")
    object DetailScreen: Screen("detail_screen")
    object CartScreen: Screen("cart_screen")
    object ReviewScreen: Screen("review_screen")

}

```

Figure 4 Screen routing Code & Libraries

```

package com.groceryapp.ui.login

import android.annotation.SuppressLint
import android.util.Log
import android.widget.Toast
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Alignment.Companion.BottomCenter
import androidx.compose.ui.ExperimentalComposeUiApi
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.colorResource
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.TextStyle
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.compose.ui.window.Dialog
import androidx.compose.ui.window.DialogProperties
import androidx.navigation.NavController
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.firestore.ktx.firestore
import com.google.firebase.ktx.Firebase
import com.groceryapp.R
import com.groceryapp.routing.Screen
import com.groceryapp.ui.grocery_database.GroceryDatabase
import com.groceryapp.ui.theme.GroceryAppTheme
import com.groceryapp.utils.OutlineFormField
import com.groceryapp.utils.RoundedButton
import com.groceryapp.utils.isValidEmail

@OptIn(ExperimentalMaterial3Api::class, ExperimentalComposeUiApi::class)
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@Composable
fun LoginScreen(navController: NavController) {
    val context = LocalContext.current
    val preferenceManager = remember {
        GroceryDatabase(context)
    }
}

```

Figure 5 LoginScreen UI Code & Libraries

```
package com.groceryapp.ui.reviewScreen

import android.annotation.SuppressLint
import android.widget.Toast
import androidx.compose.foundation.*
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.selection.selectable
import androidx.compose.foundation.selection.selectableGroup
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Star
import androidx.compose.material.icons.rounded.ArrowBack
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.ExperimentalComposeUiApi
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.platform.LocalDensity
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.TextStyle
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.ImeAction
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import com.groceryapp.R
import com.groceryapp.ui.theme.GroceryAppTheme
import com.groceryapp.ui.theme.orange
import com.groceryapp.ui.theme.white
import com.groceryapp.utils.RoundedButton

@OptIn(ExperimentalMaterial3Api::class, ExperimentalComposeUiApi::class)
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@Composable
fun ReviewScreen(navController: NavController) {
    val context = LocalContext.current
    val scrollState = rememberScrollState()
    var rating by remember { mutableStateOf(0F) }

    var review by remember { mutableStateOf("") }
    var feedback by remember { mutableStateOf("") }
    var isReviewed by remember { mutableStateOf(false) }
    GroceryAppTheme {
        Scaffold {
            Column(
                modifier = Modifier
                    .fillMaxSize()
                    .background(color = orange)
                    .padding(top = 40.dp)
            )
        }
    }
}
```

Figure 6 ReviewScreen UI Code & Libraries

```

package com.groceryapp.ui.splash

import android.annotation.SuppressLint
import android.content.Context
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.Scaffold
import androidx.compose.runtime.Composable
import androidx.compose.runtime.LaunchedEffect
import androidx.compose.runtime.remember
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.colorResource
import androidx.compose.ui.res.painterResource
import androidx.navigation.NavController
import com.groceryapp.R
import com.groceryapp.routing.Screen
import com.groceryapp.ui.grocery_database.GroceryDatabase
import com.groceryapp.ui.theme.GroceryAppTheme
import kotlinx.coroutines.delay
import kotlin.time.Duration.Companion.seconds

@OptIn(ExperimentalMaterial3Api::class)
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@Composable
fun SplashScreen(navController: NavController) {
    val context = LocalContext.current
    val preferenceManager = remember {
        GroceryDatabase(context)
    }
    LaunchedEffect(Unit) {
        delay(3.seconds)
        if(preferenceManager.getData("isLogin")) {
            navController.navigate(Screen.MainScreen.route) {
                popUpTo(Screen.SplashScreen.route) {
                    inclusive = true
                }
            }
        }else{
            navController.navigate(Screen.LoginScreen.route) {
                popUpTo(Screen.SplashScreen.route) {
                    inclusive = true
                }
            }
        }
    }
}

```

Figure 7 SplashScreen UI Code & Libraries

```

package com.groceryapp.ui.wishlist

import android.annotation.SuppressLint
import android.widget.Toast
import androidx.compose.foundation.*
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.rounded.ArrowBack
import androidx.compose.material.icons.rounded.Home
import androidx.compose.material.icons.rounded.Settings
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.ExperimentalComposeUiApi
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.graphics.vector.ImageVector
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.colorResource
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.TextStyle
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import androidx.navigation.compose.currentBackStackEntryAsState
import com.groceryapp.R
import com.groceryapp.application.AppConfig
import com.groceryapp.routing.Screen
import com.groceryapp.ui.model.CartModel
import com.groceryapp.ui.model.ProductModel
import com.groceryapp.ui.theme.GroceryAppTheme
import com.groceryapp.ui.theme.orange
import com.groceryapp.ui.theme.white
import com.groceryapp.utils.RoundedButton

@OptIn(ExperimentalMaterial3Api::class, ExperimentalComposeUiApi::class)
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter", "MutableCollectionMutableState")
@Composable
fun WishlistScreen(navController: NavController) {
    val context = LocalContext.current
    val scrollState = rememberScrollState()
    var isAdded by remember { mutableStateOf(false) }

```

Figure 8 WishlistScreen UI Code & Libraries

```

package com.groceryapp.utils

import okhttp3.OkHttpClient
import retrofit2.Retrofit
import retrofit2.adapter.rxjava2.RxJava2CallAdapterFactory
import retrofit2.converter.gson.GsonConverterFactory
import retrofit2.converter.scalars.ScalarsConverterFactory

object ApiClass {
    fun createClient(): Retrofit {
        return Retrofit.Builder()
            .baseUrl("https://fakestoreapi.com/")
            .client(OkHttpClient.Builder().build())
            .addConverterFactory(GsonConverterFactory.create())
            .addConverterFactory(ScalarsConverterFactory.create())
            .addCallAdapterFactory(RxJava2CallAdapterFactory.create())
            .build()
    }
}

```

Figure 9 ApiClass utils Code & Libraries

```

package com.groceryapp.utils

import retrofit2.Call
import retrofit2.Response
import retrofit2.http.GET
import retrofit2.http.POST

interface ApiService {
    @get:POST("products")
    val data : Response<String>
}

```

Figure 10 ApiService utils Code & Libraries

```

package com.groceryapp.utils

import androidx.compose.foundation.clickable
import androidx.compose.foundation.focusable
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.text.KeyboardActions
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material3.*
import androidx.compose.runtime.Composable
import androidx.compose.ui.ExperimentalComposeUiApi
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.LocalSoftwareKeyboardController
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.KeyboardCapitalization
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.text.input.VisualTransformation
import androidx.compose.ui.unit.Dp
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.groceryapp.ui.theme.orange

@OptIn(ExperimentalMaterial3Api::class, ExperimentalComposeUiApi::class)
@Composable
fun OutlineFormField(
    value: String,
    placeholder: String,
    onValueChange: (String) -> Unit,
    onClick: () -> Unit = {},
    isEnabled: Boolean = true,
    readOnly: Boolean = false,
    keyboardType: KeyboardType = KeyboardType.Text,
    keyboardCapitalization: KeyboardCapitalization = KeyboardCapitalization.Sentences,
    visualTransformation: VisualTransformation = VisualTransformation.None,
    leadingIcon: @Composable (() -> Unit)? = null,
    trailingIcon: @Composable (() -> Unit)? = null,
    horizontalPadding : Dp = 0.dp,
    verticalPadding : Dp = 3.dp,
    colors : TextFieldColors? = null,
    label : @Composable (() -> Unit)? = null,
) {
    val keyboardController = LocalSoftwareKeyboardController.current
    OutlinedTextField(
        value = value,
        shape = RoundedCornerShape(10.dp),
        keyboardOptions = KeyboardOptions(keyboardType = keyboardType, capitalization = keyboardCapitalization),
        keyboardActions = KeyboardActions(

```

Figure 11 OutlineFormField utils Code & Libraries

Social, Legal, Ethical and Security Issues

Social Issues:

Social Issues

Availability and Accessibility

Digital Division of the Technology

Policy for Labour Practices

Effect on Small-Local Businesses

Privacy Concerns

Digital Division of the Technology:

In today's world, still not everyone has access to the smartphones and reliable internet technology, so the Grocery Delivery Mobile Application potentially creating an unequal access to groceries.

Policy for Labour Practices:

The salaries and working conditions of the delivery workers must be taken into consideration by the grocery delivery mobile app. Fair pay and conditions at work are essential, and they should adhere to labour laws.

Effect on Small-Local Businesses:

If a large portion of the social community shifts to online grocery delivery then Grocery Delivery Mobile Application's convenience may affect local and small grocery stores, potentially leading to closure and loss of jobs in the community.

Privacy Concerns:

The users may have concerns about sharing personal data, such as their location, shopping habits, and payment information, with the application.

Legal Issues:

Legal Issues

Data Protection Laws

Terms and Conditions of the services

Intellectual Property Rights

Observance of Food Safety Regulations

Data Privacy Policy

Intellectual Property Rights:

The Grocery Delivery Mobile Application must need to follow the intellectual property rights to ensure its customers that this application does not violate the trademarks, copyrights, and patents policies of others.

Observance of Food Safety Regulations:

The grocery delivery mobile app should always comply **The Food Safety Act 1990** and laws regarding storage and transportation of groceries strictly for the perishables items. The Grocery Delivery Mobile Application must strictly follow the basic hygiene standards for two reasons, the first one is to prevent the

risk of contamination and illness, and the other one is to build the trust of the users.

Data Privacy Policy:

The Grocery Delivery Mobile Application must guarantee to the users about the data privacy guidelines. It is important for the application administrator to state that they never miss-use the user's personnel information by providing it to an external host or server.

Ethical Issues:

Ethical Issues	Algorithmic Bias
	Environmental Impact
	Worker Treatment and Labour Practices
	Transparency

Algorithmic Bias:

The Grocery Delivery Mobile Application's administration need to avoid biasing in recommendation algorithms that may disproportionately benefit certain products or exclude others based on factors like price or brand, potentially influencing user choices unfairly.

Environmental Impact:

Increased use of packaging paper and plastic for deliveries may contribute to packaging waste and increase carbon emissions from transportation. The app could explore ways to promote eco-friendly packaging or encourage users to opt-out of unnecessary packaging.

Worker Treatment and Labour Practices:

For the better performance of the Grocery Delivery Mobile Application, delivery workers other employees should be treated fairly. They should have the opportunities for feedback and grievance redress.

Security Issues:

security Issues	Secure Payment Processing
	Data Breaches and its Security
	Account Security and Authentication
	Secure Communication

Data Breaches and its Security:

The Grocery Delivery Mobile Application should have safeguarded feature against data breaches that could expose users' personal information, including credit card details and addresses from unauthorized access.

Account Security and Authentication:

The Grocery Delivery Mobile Application must be implemented the two-factor authentication and other security measures to prevent unauthorized account access. The app administrator should try to preventing identity theft by verifying the identity of users and protecting their accounts from unauthorized access.

Secure Communication:

The Grocery Delivery Mobile Application must ensure that communication between users, the application, and delivery personnel is encrypted to prevent interception or manipulation of sensitive information.

Development Strategy

In the development of the app, Agile Methodology is used. In this process we use 5 steps like:

Sprint Goal, To- do sprint, Doing sprint, Finally, Testing and Completed sprints.

Phases

Sprint Goal:

We define sprint goals as requirement gathering, development of Grocery Delivery Mobile Application, GIT deployment, Testing.

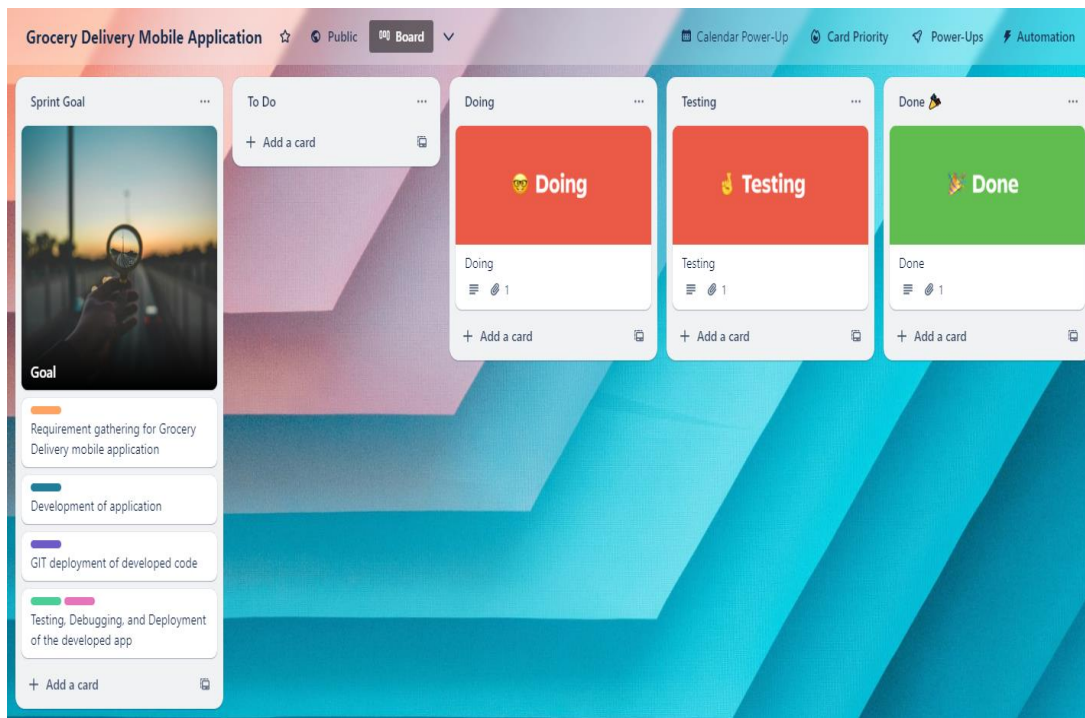


Figure 12 Sprit Goal

Phase-1:

This phase contains a list of works to do for the development of grocery delivery application.

Each sprint deadline is fixed. Based on this, tasks for each sprint is given in cards.

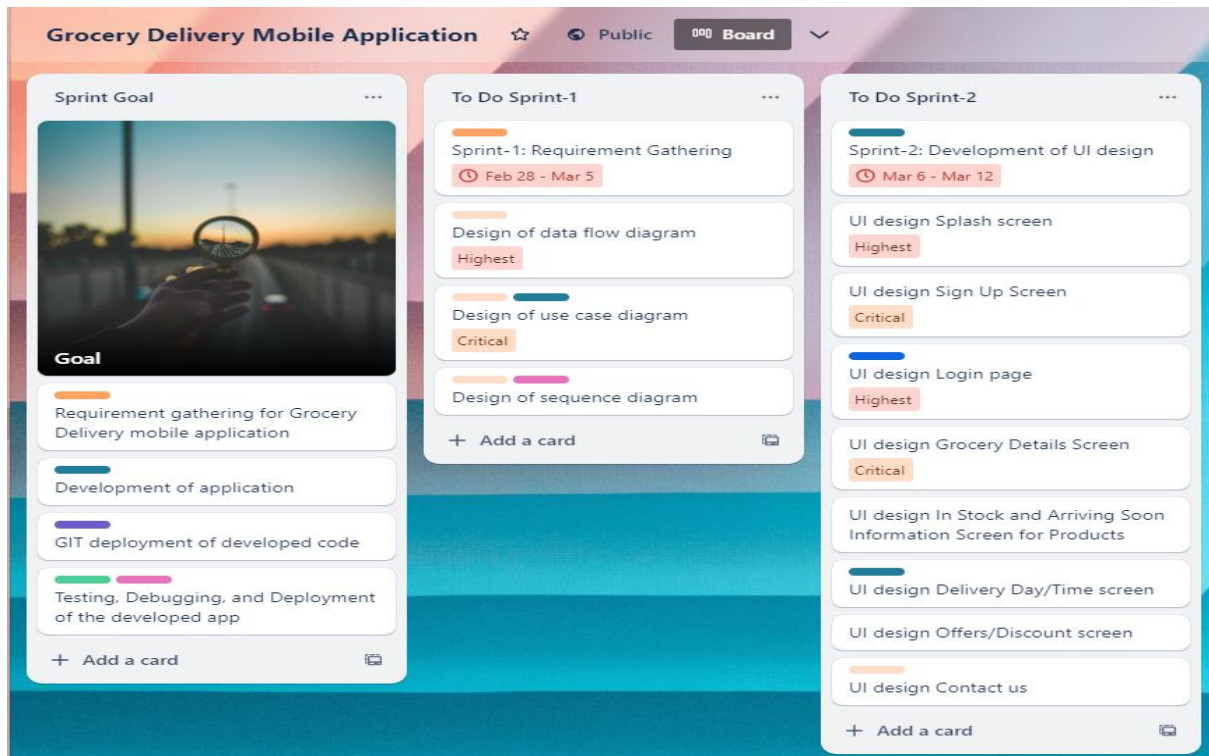


Figure 13 Phase 1 (i)

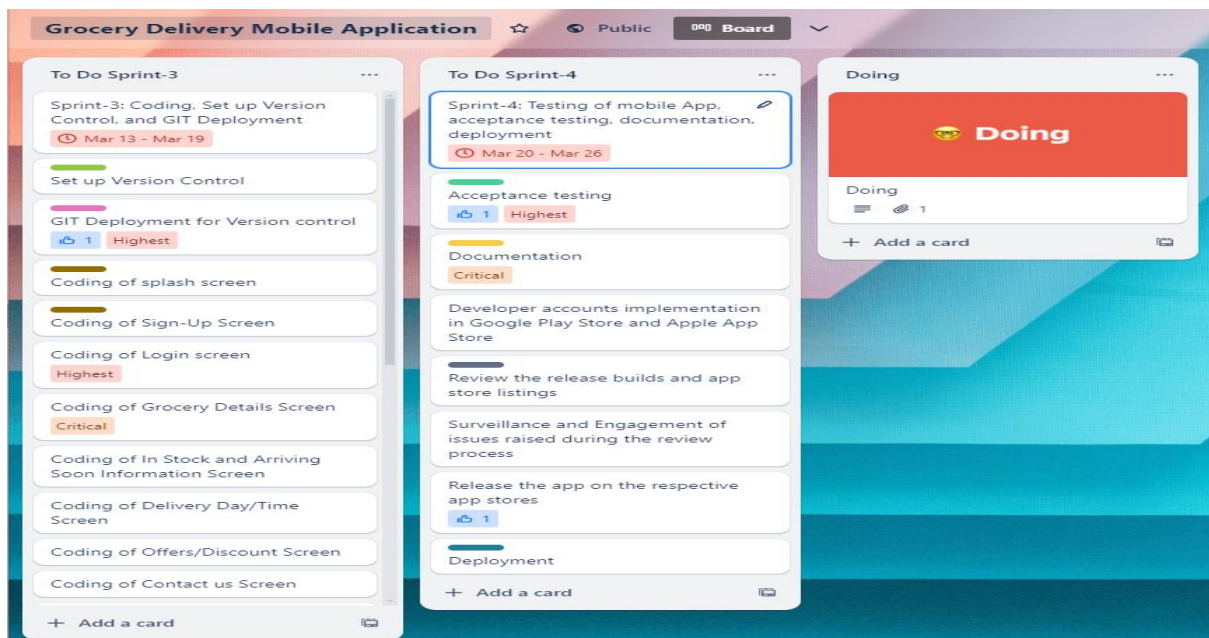


Figure 14 Phase 1 (ii)

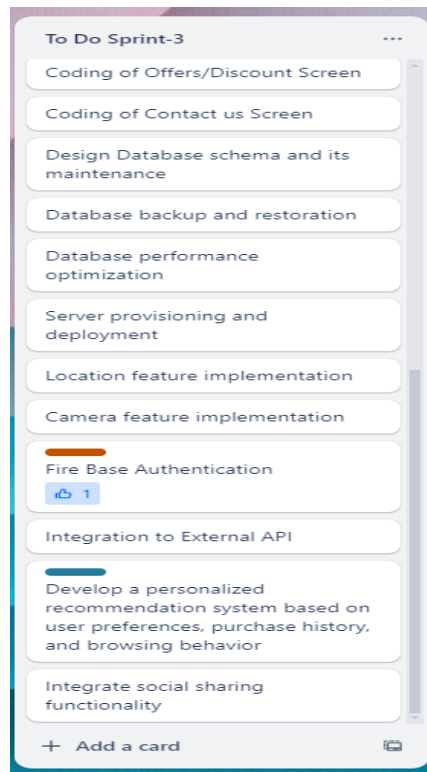


Figure 15 To Do Sprint-3

Phase-2: This phase shows the ongoing work. We start working on each sprint one by one. However, for Sprint 3, certain task is complicated to complete. Hence, they are not performed and kept in Abandoned Tasks.

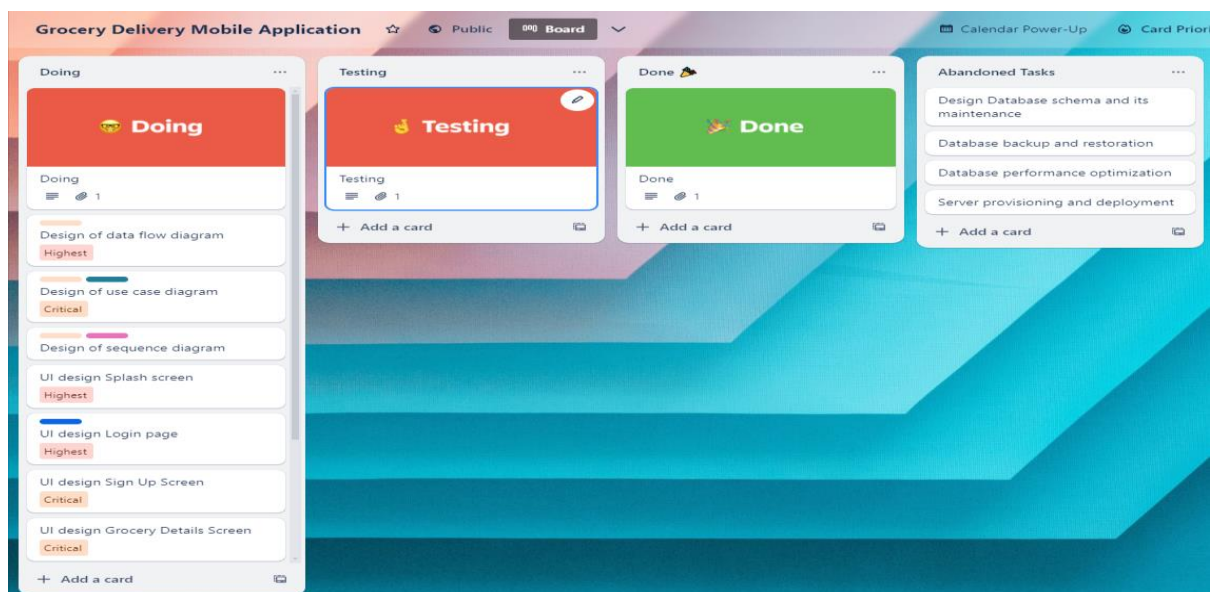


Figure 16 Phase 2

Phase-3: Once all tasks are finished, the whole card is moved to Done. On the other hand, we start testing of the developed app. Therefore, we move the Testing app card to Testing list. Whereas, certain tasks could not be completed. They are moved to Could Not Reach List.

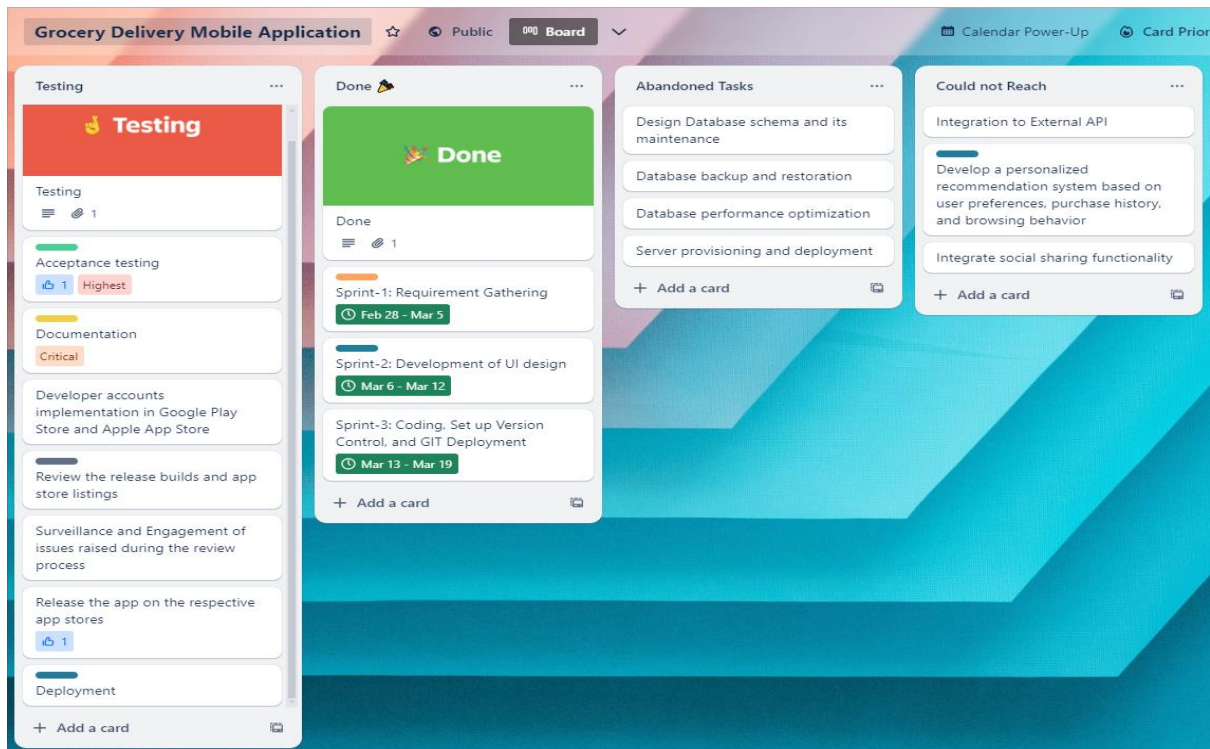


Figure 17 Phase 3

Phase-4:

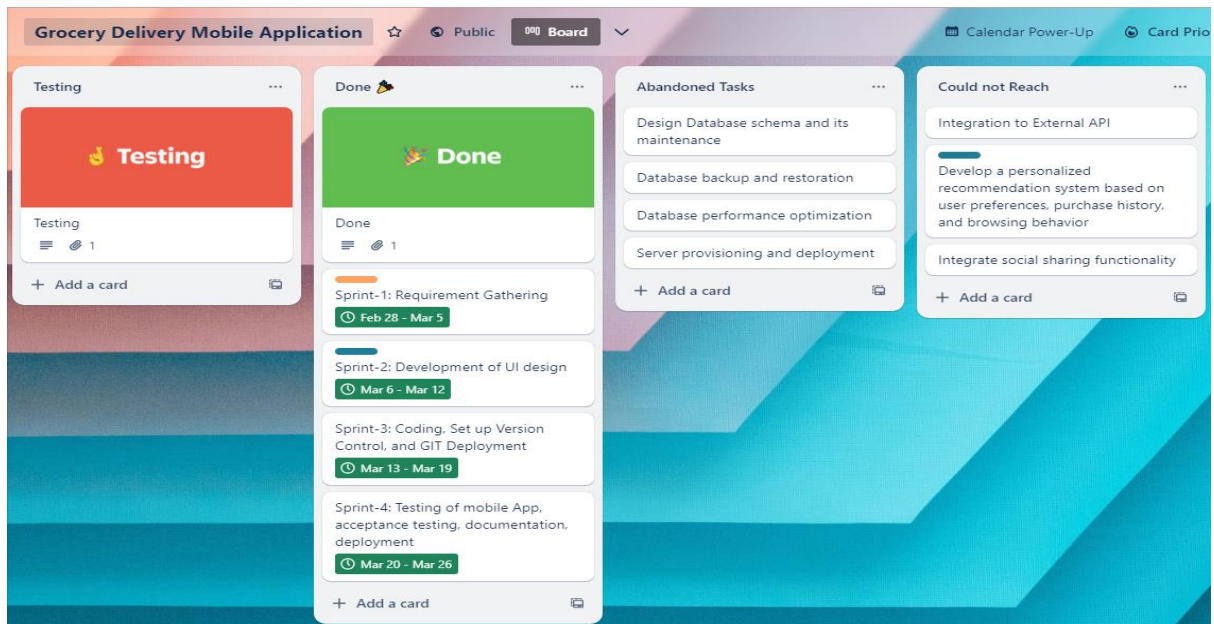


Figure 18 Phase 4

Burn Down Charts

Sprint 1 Burn-down

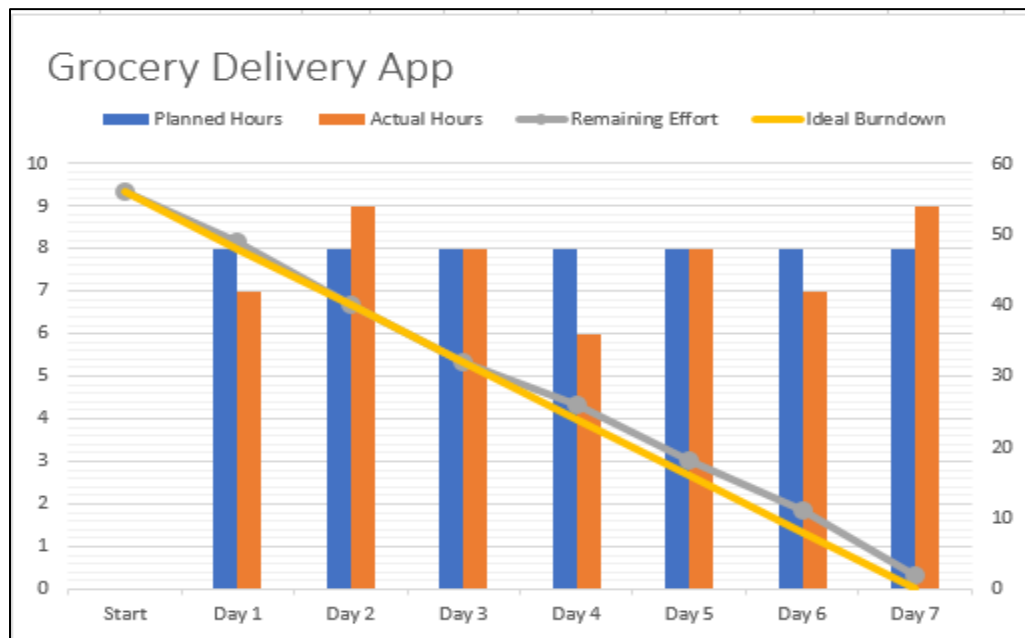


Figure 19 Sprint 1 Burn-down

Sprint 2 Burn-down

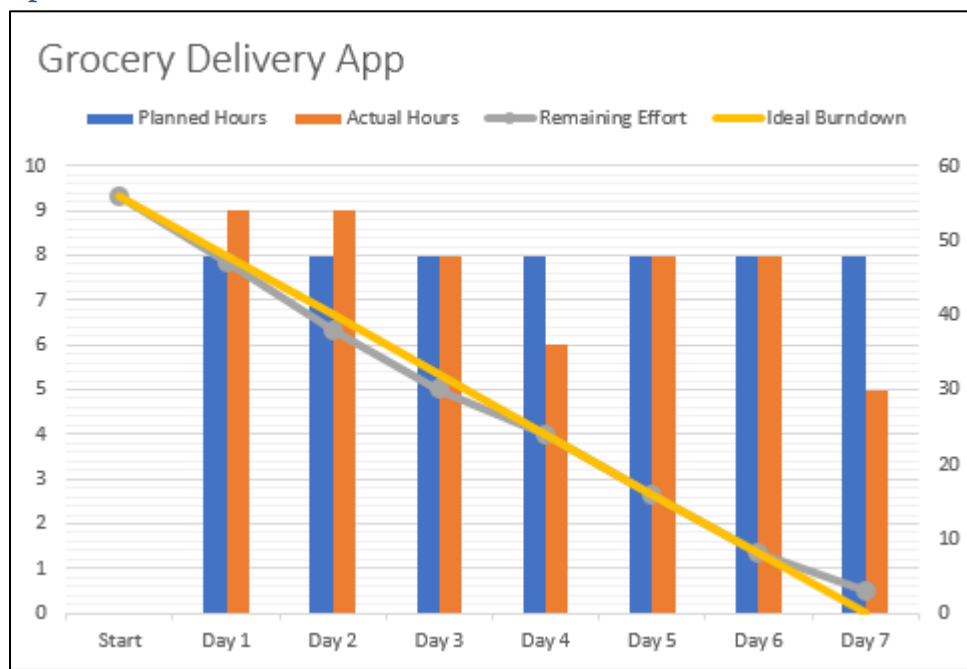


Figure 20 Sprint 2 Burn-down

Sprint 3 Burn-down:

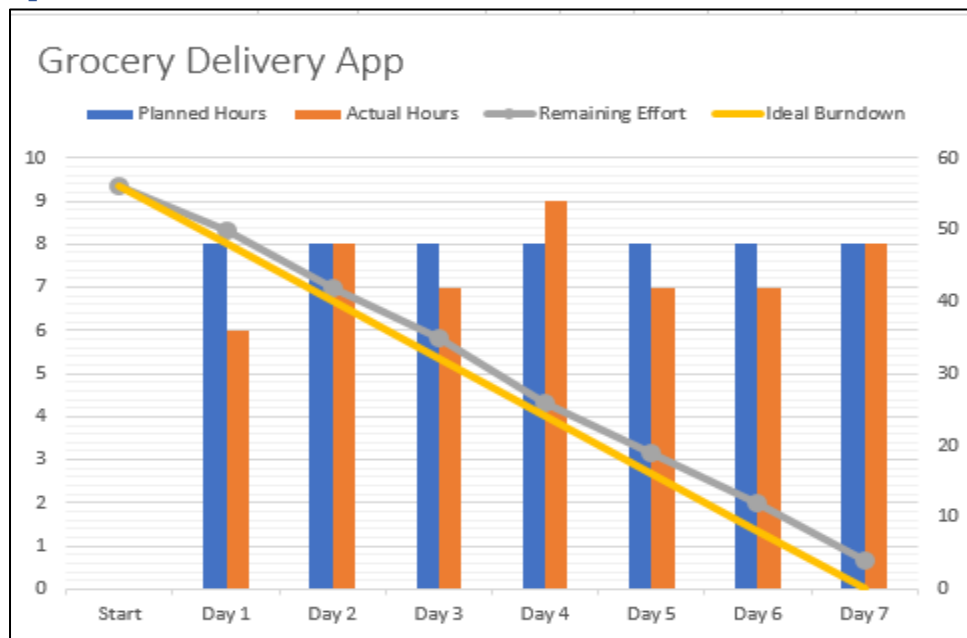


Figure 21 Sprint 3 Burn-down

Sprint 4 Burn-down:

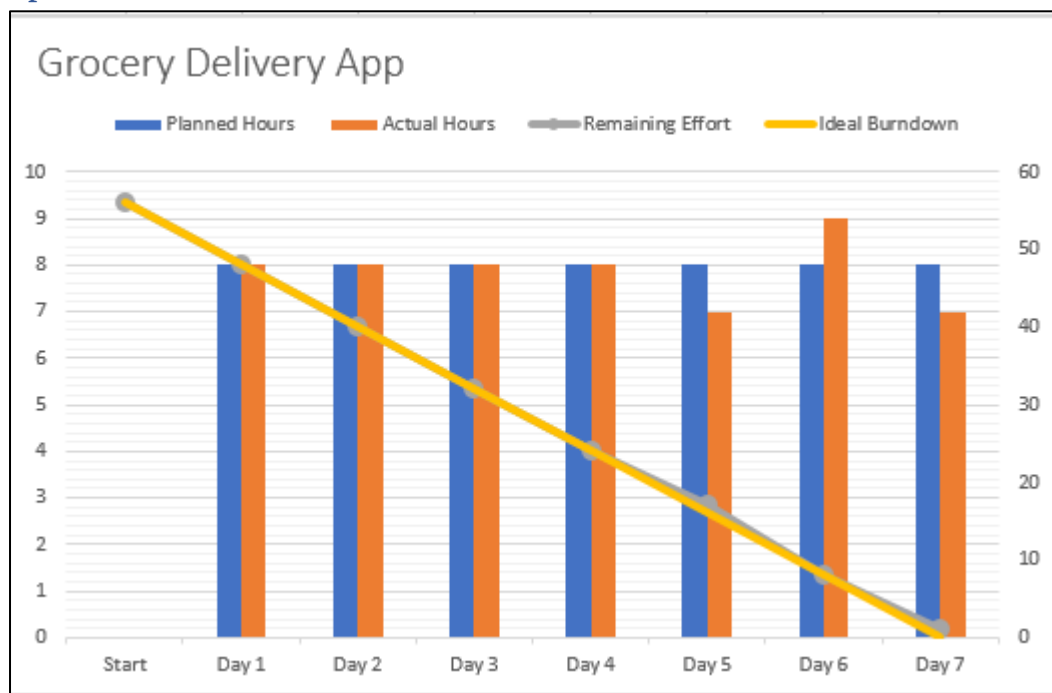


Figure 22 Sprint 4 Burn-down

Firestore DB at backend and regarding API authentication and searching

Grocery Delivery Application

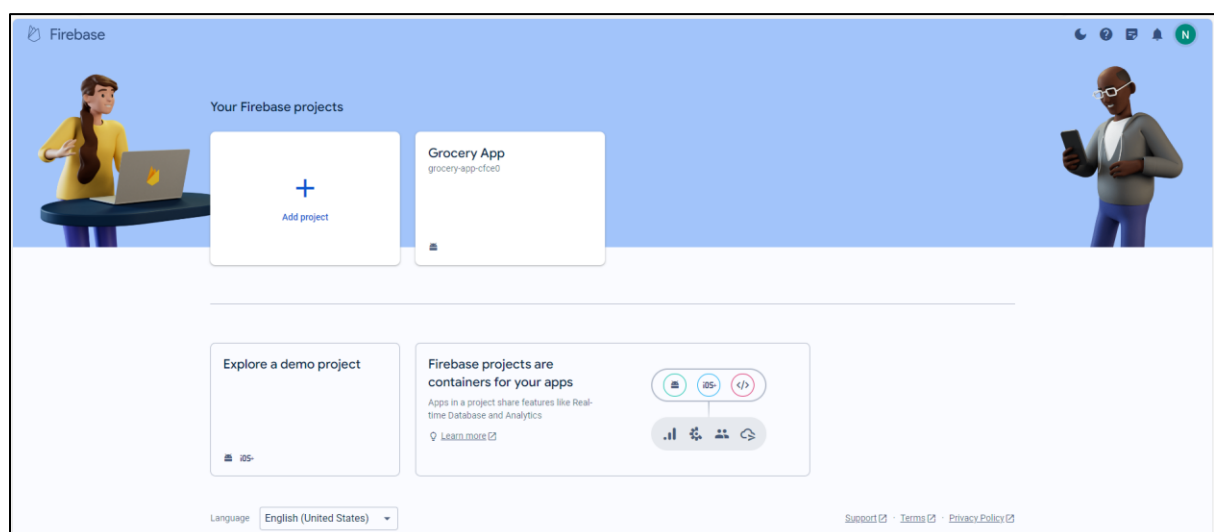


Figure 23 Firebase Project home page

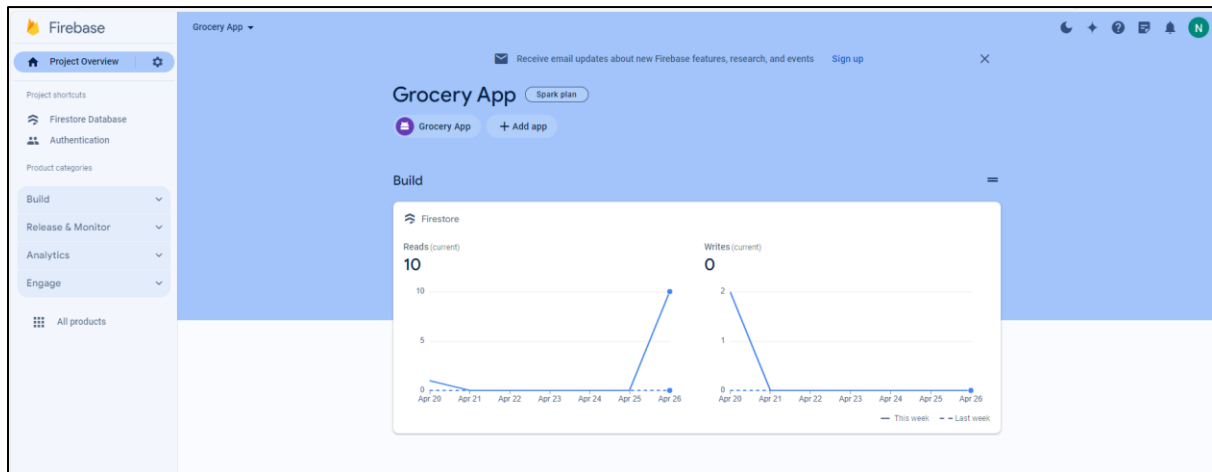


Figure 24 Grocery application read and write graph in firebase

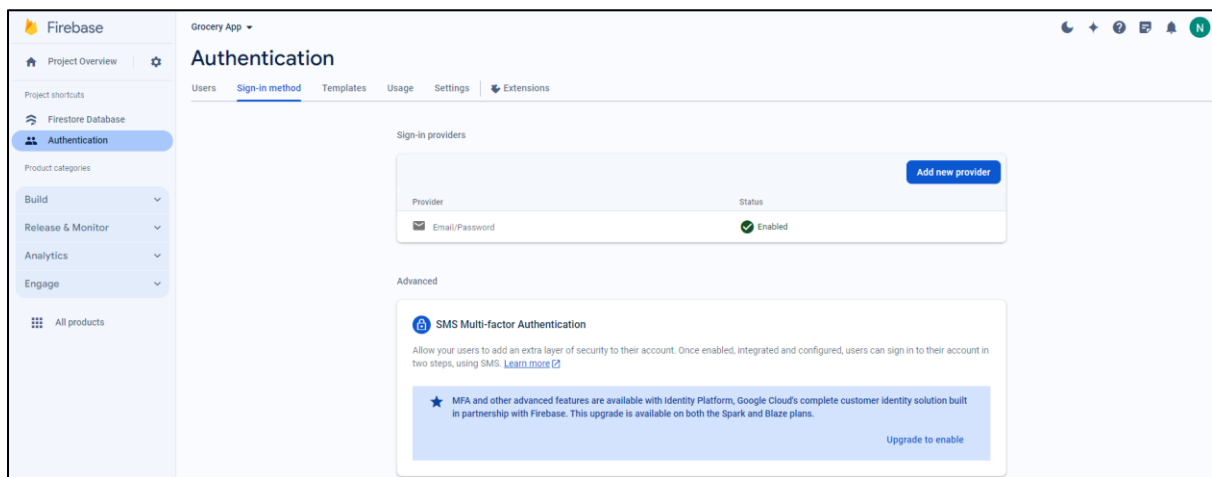


Figure 25 Authentication screen in firebase

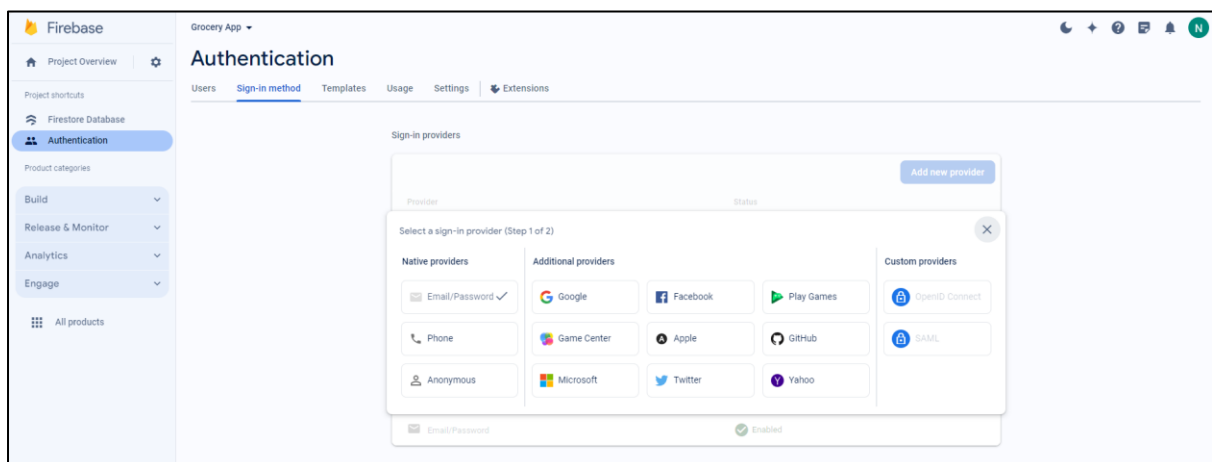


Figure 26 Authentication option in firebase

The screenshot shows the Firebase Authentication console for a project named 'Grocery App'. The 'Users' tab is selected, displaying a table of registered users. The table has columns for Identifier, Providers, Created, Signed In, and User UID. There are 5 users listed, all created on April 26 or 27, 2024. A search bar at the top allows filtering by email, phone number, or user UID. An 'Add user' button is in the top right corner.

Identifier	Providers	Created	Signed In	User UID
arjun@gmail.com	Google	Apr 26, 2024	Apr 26, 2024	ZoCsrbdavHOecuZfZdyFZOBn...
sushil@gmail.com	Google	Apr 26, 2024	Apr 26, 2024	tuVFDa3Bdes3twB44dQar0L...
naveen.chintala1708@...	Google	Apr 26, 2024	Apr 26, 2024	K6iO4JvcvFgdbVhUbc9gh8P...
anshul@gmail.com	Google	Apr 26, 2024	Apr 26, 2024	kixvS9Ny6T5ARbcqdmek7Vh...
anu@gmail.com	Google	Apr 27, 2024	Apr 27, 2024	8ppOfpwcVmfPQd4i6iPS7PH...

Figure 27 user list in firebase log in

The screenshot shows the Firebase Cloud Firestore console for the 'Grocery App' project. The 'Data' tab is selected, showing the 'users' collection under the 'OZas209rSWa2...' database. The collection is currently empty. A 'Panel view' button is visible. A warning message at the top states: 'Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing. Configure App Check'.

Figure 28 Cloud Firestore

This screenshot is a closer view of the 'users' collection in the Cloud Firestore console. It shows the 'Add document' button and the document ID 'OZas209rSWa2rePdCD5C'. Below the ID, the document content is visible: 'email: "9999999999"', 'name: "MUKESH"', and 'password: "1234"'. The breadcrumb navigation at the top shows 'home > users'.

Figure 29 Users and Add document in DB Cloud firestore

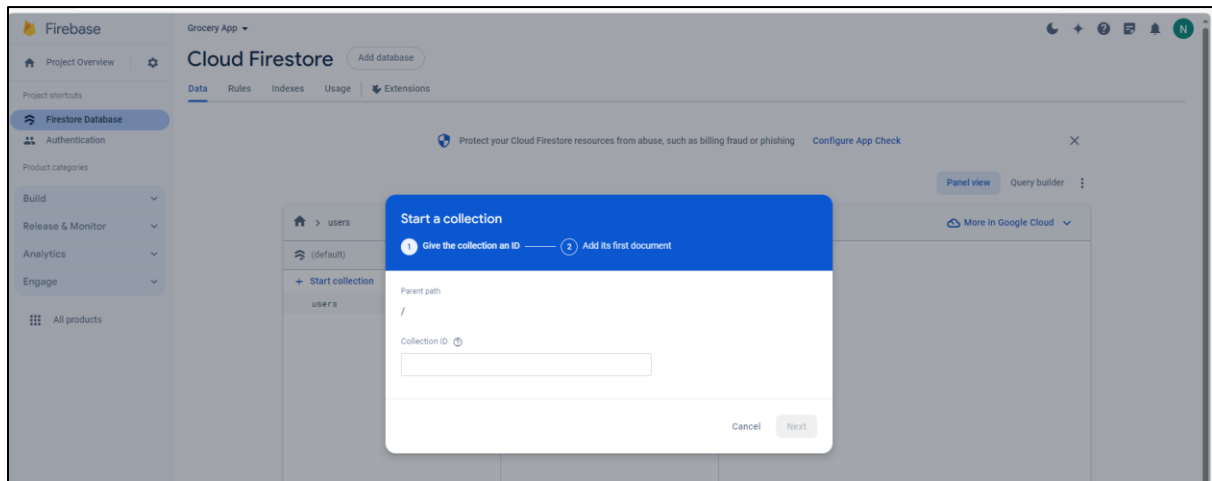


Figure 30 Start collection option

GIT

1st commit

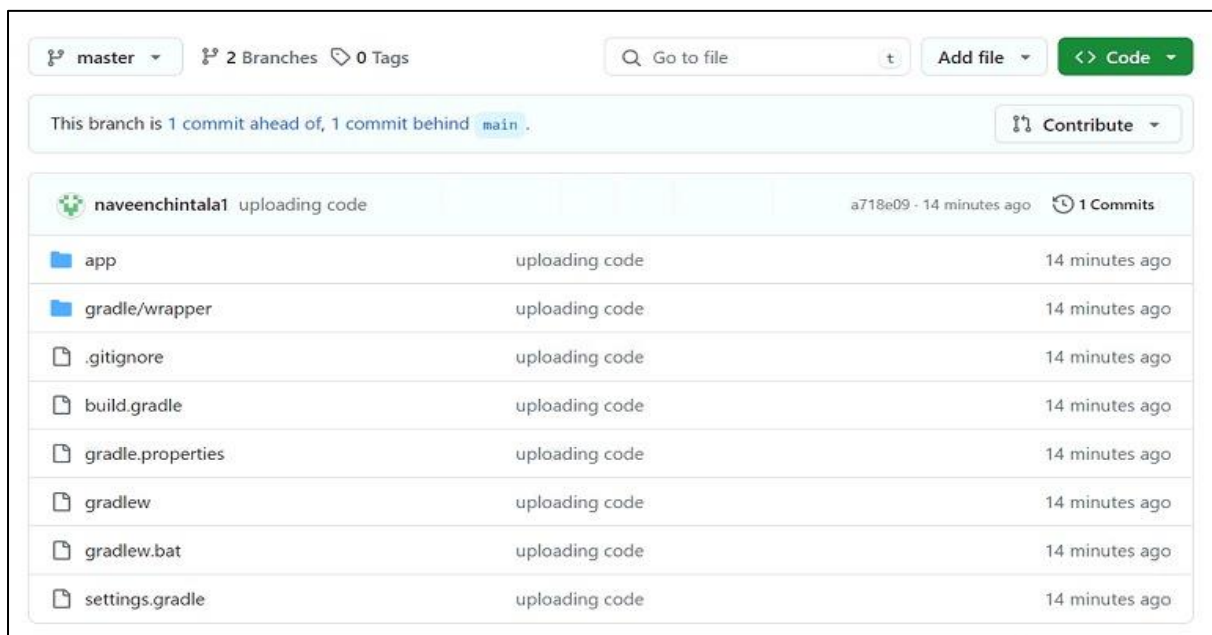


Figure 31 1st commit

```

prave@HP-000 MINGW64 ~/Downloads/Grocery App/GroceryApp (master)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/build.gradle', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/google-services.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/proguard-rules.pro', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/androidTest/java/com/groceryapp/ExampleInstrumentedTest.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/groceryapp/MainActivity.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/groceryapp/ui/model/ProductModel.kt', LF will be replaced by CRLF the next time Git touches it

```

Figure 32 1st commit I

```

prave@HP-000 MINGW64 ~/Downloads/Grocery App/GroceryApp (master)
$ git commit -m "uploading code"
[master (root-commit) a718e09] uploading code
59 files changed, 2248 insertions(+)
create mode 100644 .gitignore
create mode 100644 app/.gitignore
create mode 100644 app/build.gradle
create mode 100644 app/google-services.json
create mode 100644 app/proguard-rules.pro
create mode 100644 app/src/androidTest/java/com/groceryapp/ExampleInstrumentedTest.kt
create mode 100644 app/src/main/AndroidManifest.xml
create mode 100644 app/src/main/java/com/groceryapp/MainActivity.kt
create mode 100644 app/src/main/java/com/groceryapp/routing/Navigation.kt
create mode 100644 app/src/main/java/com/groceryapp/routing/Screen.kt
create mode 100644 app/src/main/java/com/groceryapp/ui/detail/DetailsScreen.kt
create mode 100644 app/src/main/java/com/groceryapp/ui/login/LoginScreen.kt
create mode 100644 app/src/main/java/com/groceryapp/ui/main/MainScreen.kt

```

```

prave@HP-000 MINGW64 ~/Downloads/Grocery App/GroceryApp (master)
$ git remote add origin https://github.com/naveen-chintala1/GroceryApp.git

```

Figure 33 1st commit II

```

prave@HP-000 MINGW64 ~/Downloads/Grocery App/GroceryApp (master)
$ git push origin master
Enumerating objects: 99, done.
Counting objects: 100% (99/99), done.
Delta compression using up to 16 threads
Compressing objects: 100% (80/80), done.
Writing objects: 100% (99/99), 119.05 KiB | 4.76 MiB/s, done.
Total 99 (delta 11), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (11/11), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/naveenchintala1/GroceryApp/pull/new/master
remote:
To https://github.com/naveenchintala1/GroceryApp.git
 * [new branch]      master -> master

prave@HP-000 MINGW64 ~/Downloads/Grocery App/GroceryApp (master)
$

```

Figure 34 1st commit III

```

prave@HP-000 MINGW64 ~/Downloads/Grocery App/GroceryApp
$ git config --global user.name "naveenchintala1"

prave@HP-000 MINGW64 ~/Downloads/Grocery App/GroceryApp
$ git config --global user.email "d3919253@live.tees.ac.uk"

prave@HP-000 MINGW64 ~/Downloads/Grocery App/GroceryApp
$ pwd
/c/Users/prave/Downloads/Grocery App/GroceryApp

prave@HP-000 MINGW64 ~/Downloads/Grocery App/GroceryApp
$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <

```

Figure 35 1st commit III

2nd commit

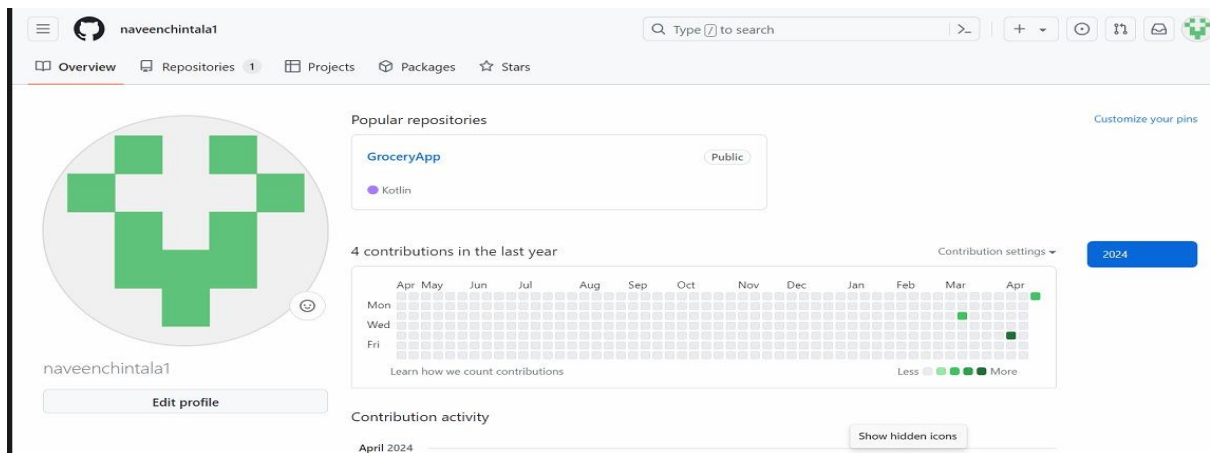


Figure 36 2nd commit I

```
prave@HP-000 MINGW64 ~/Downloads/second commit/cloned/GroceryApp (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        app/
        build.gradle
        gradle.properties
        gradle/
        gradlew
        gradlew.bat
        settings.gradle

nothing added to commit but untracked files present (use "git add" to track)
```

Figure 37 2nd commit II

```
prave@HP-000 MINGW64 ~/Downloads/second commit/cloned/GroceryApp (main)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CR
the next time Git touches it
warning: in the working copy of 'app/build.gradle', LF will be replaced
the next time Git touches it
warning: in the working copy of 'app/google-services.json', LF will be
replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/proguard-rules.pro', LF will be re
placed by CRLF the next time Git touches it
warning: in the working copy of 'app/src/androidTest/java/com/groceryapp
InstrumentedTest.kt', LF will be replaced by CRLF the next time Git tou
warning: in the working copy of 'app/src/main/java/com/groceryapp/MainA
t', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/groceryapp/ui/dr
rawer.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/groceryapp/ui/mo
ctModel.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/groceryapp/ui/th
.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/groceryapp/ui/th
.kt', LF will be replaced by CRLF the next time Git touches it
```

Figure 38 2nd commit III

```

prave@HP-000 MINGW64 ~/Downloads/second commit/cloned/GroceryApp (main)
$ git commit -m "second commit"
[main 695c432] second commit
64 files changed, 2477 insertions(+)
create mode 100644 .gitignore
create mode 100644 app/.gitignore
create mode 100644 app/build.gradle
create mode 100644 app/google-services.json
create mode 100644 app/proguard-rules.pro
create mode 100644 app/src/androidTest/java/com/groceryapp/ExampleInstrumentedT
est.kt
create mode 100644 app/src/main/AndroidManifest.xml
create mode 100644 app/src/main/java/com/groceryapp/MainActivity.kt
create mode 100644 app/src/main/java/com/groceryapp/routing/Navigation.kt
create mode 100644 app/src/main/java/com/groceryapp/routing/Screen.kt
create mode 100644 app/src/main/java/com/groceryapp/ui/detail/DetailScreen.kt
create mode 100644 app/src/main/java/com/groceryapp/ui/drawer/NavDrawer.kt
create mode 100644 app/src/main/java/com/groceryapp/ui/drawer/TopBar.kt
create mode 100644 app/src/main/java/com/groceryapp/ui/grocery_database/Grocery
Database.kt
create mode 100644 app/src/main/java/com/groceryapp/ui/login/LoginsScreen.kt
create mode 100644 app/src/main/java/com/groceryapp/ui/main/MainScreen.kt
create mode 100644 app/src/main/java/com/groceryapp/ui/model/ProductModel.kt
create mode 100644 app/src/main/java/com/groceryapp/ui/sign_up/SignUpScreen.kt
create mode 100644 app/src/main/java/com/groceryapp/ui/splash/SplashS
create mode 100644 app/src/main/java/com/groceryapp/ui/theme/Color.kt

```

Figure 39 2nd commit IV

```

prave@HP-000 MINGW64 ~/Downloads/second commit/cloned/GroceryApp (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

prave@HP-000 MINGW64 ~/Downloads/second commit/cloned/GroceryApp (main)
$ git push
Enumerating objects: 107, done.
Counting objects: 100% (107/107), done.
Delta compression using up to 16 threads
Compressing objects: 100% (78/78), done.
Writing objects: 100% (106/106), 123.40 KiB | 15.42 MiB/s, done.
Total 106 (delta 11), reused 80 (delta 8), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (11/11), done.
To https://github.com/naveenchintalal/GroceryApp.git
1f7240a..695c432 main -> main

```

Figure 40 2nd commit V

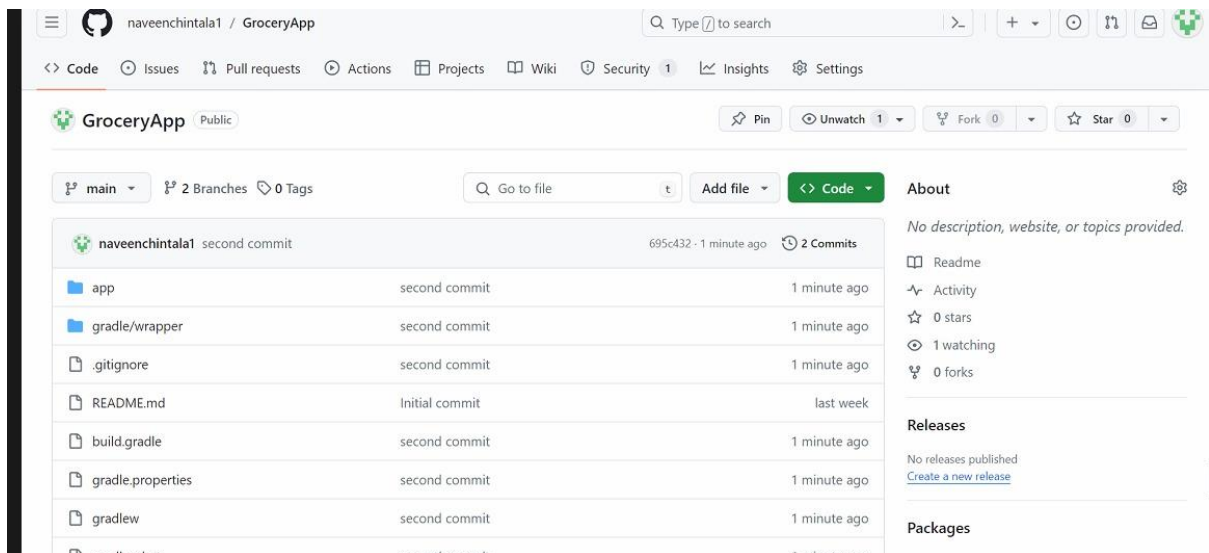


Figure 41 2nd commit VI

3rd commit

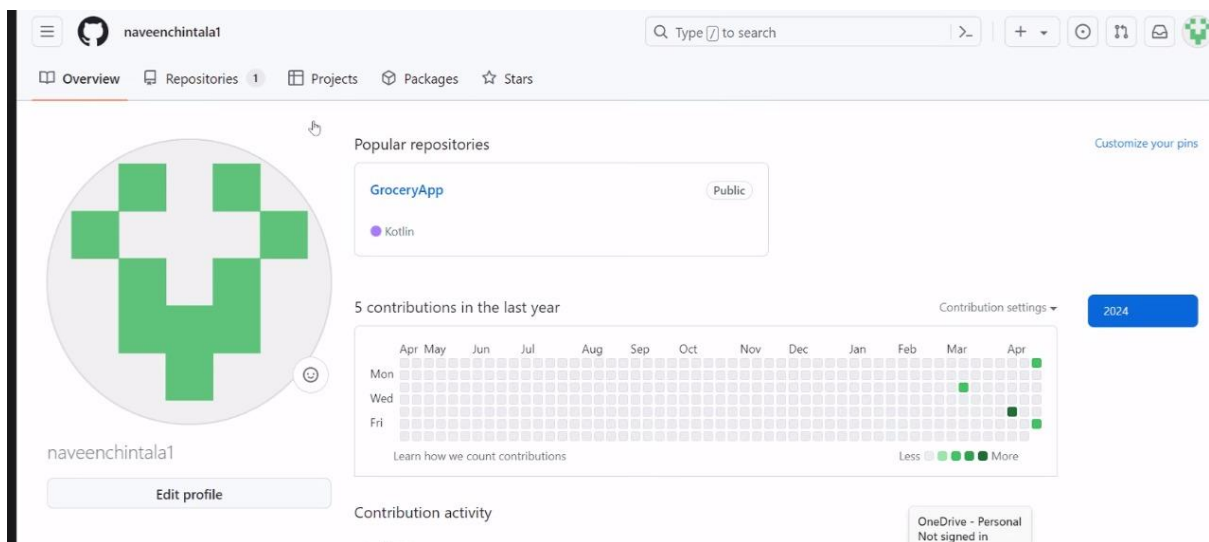


Figure 42 3rd commit I

```
prave@HP-000 MINGW64 ~/Downloads/third commit/clone/GroceryApp (main)
$ git push
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 16 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 777 bytes | 388.00 KiB/s, done.
Total 10 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/naveenchintala1/GroceryApp.git
695c432..37cac33  main -> main
```

Figure 43 3rd commit II

```
prave@HP-000 MINGW64 ~/Downloads/third commit/clone/GroceryApp (main)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the ne
xt time Git touches it
warning: in the working copy of 'app/build.gradle', LF will be replaced by CRLF
the next time Git touches it
warning: in the working copy of 'app/google-services.json', LF will be replaced
by CRLF the next time Git touches it
warning: in the working copy of 'app/proguard-rules.pro', LF will be replaced by
CRLF the next time Git touches it
warning: in the working copy of 'app/src/androidTest/java/com/groceryapp/Example
InstrumentedTest.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/groceryapp/MainActivity.k
t', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/groceryapp/ui/drawer/NavD
rawer.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/groceryapp/ui/model/Produ
ctModel.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/groceryapp/ui/theme/Color
.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/groceryapp/ui/theme/Theme
.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/groceryapp/ui/theme/Type.
```

Figure 44 3rd commit III

The screenshot shows the GitHub repository page for 'naveenchintala1 / GroceryApp'. The repository is public and has 2 branches and 0 tags. The commit history shows the third commit by naveenchintala1, which includes files like app, gradle/wrapper, .gitignore, README.md, build.gradle, gradle.properties, and gradlew. The file list on the right shows the commit history for each file, with the 'app' directory being the most recent commit.

File	Commit	Time
app	third commit	1 minute ago
gradle/wrapper	second commit	5 days ago
.gitignore	second commit	5 days ago
README.md	Initial commit	2 weeks ago
build.gradle	second commit	5 days ago
gradle.properties	second commit	5 days ago
gradlew	second commit	5 days ago

Figure 45 3rd commit IV

Grocery Delivery Mobile Application (UI)

1. Welcome Page –



Figure 46 Welcome Page

2. Registration Page –

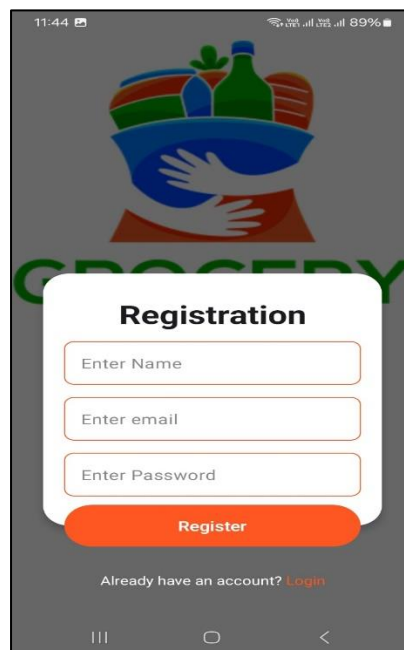


Figure 47 Registration Page

3. Login Page –

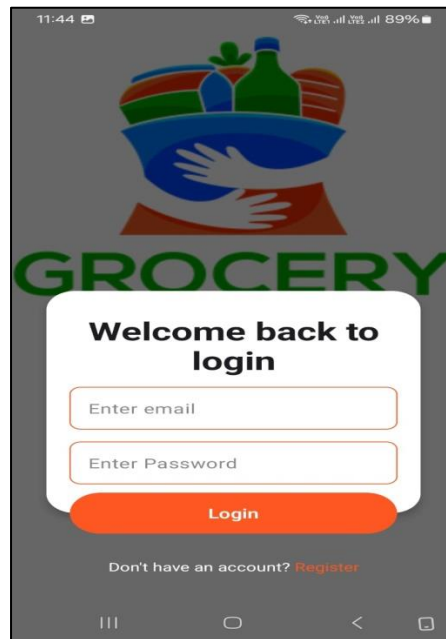


Figure 48 Login Page

4. Home Page –

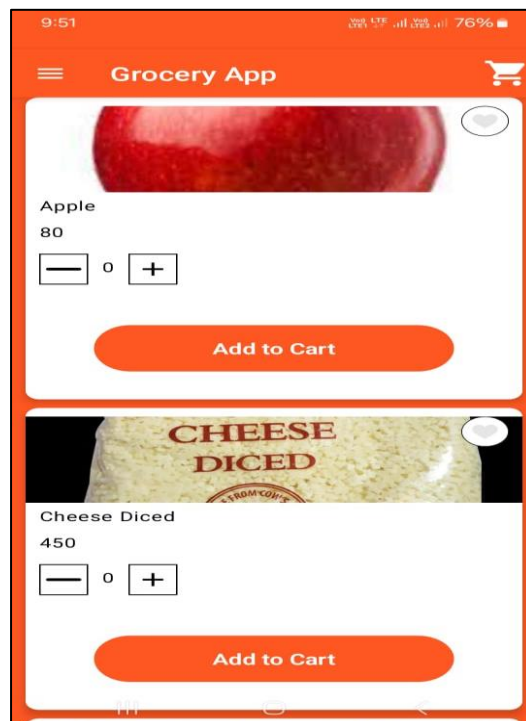


Figure 49 Home Page

5. Product Detail Page –

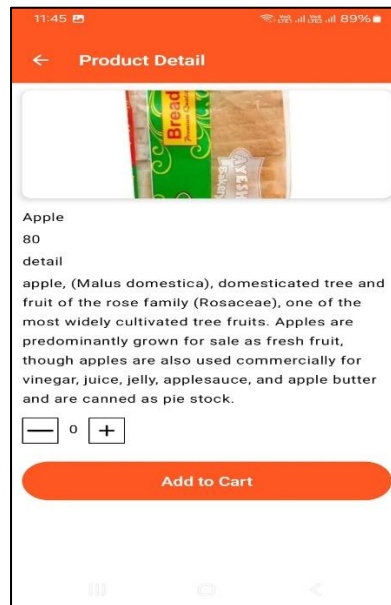


Figure 50 Product Detail Page

6. Add Product to cart –

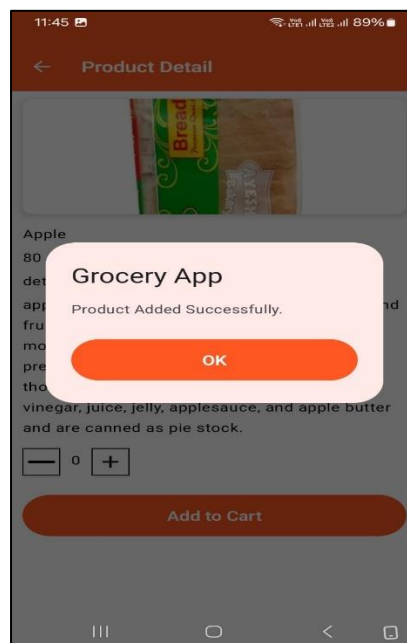


Figure 51 Product added to Cart

7. Cart Screen –

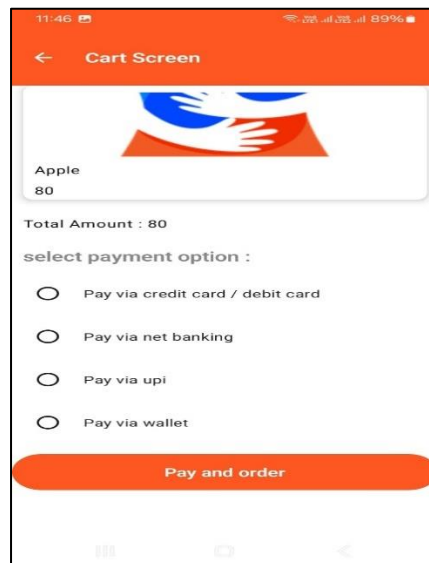


Figure 52 Product added to Cart

8. Payment and placed order –

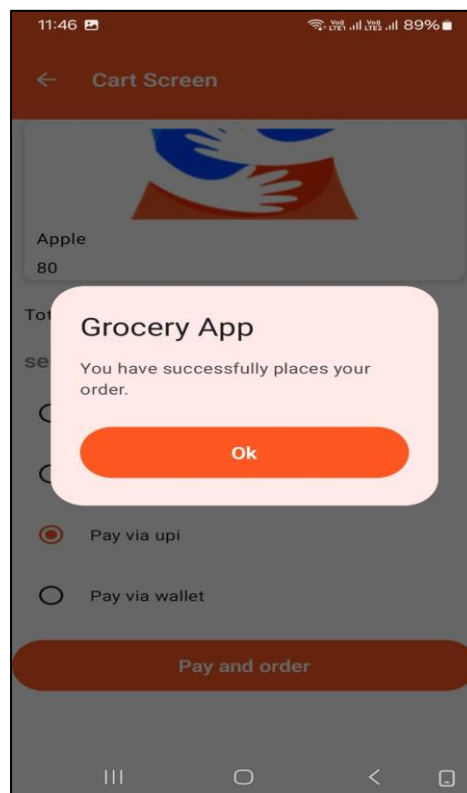


Figure 53 Payment and Placed Order

9. Wish list –

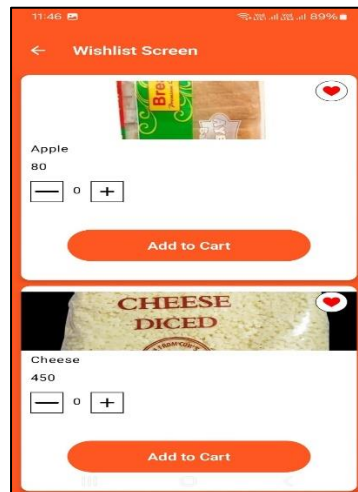


Figure 54 Wish List

10. Review and Feedback –

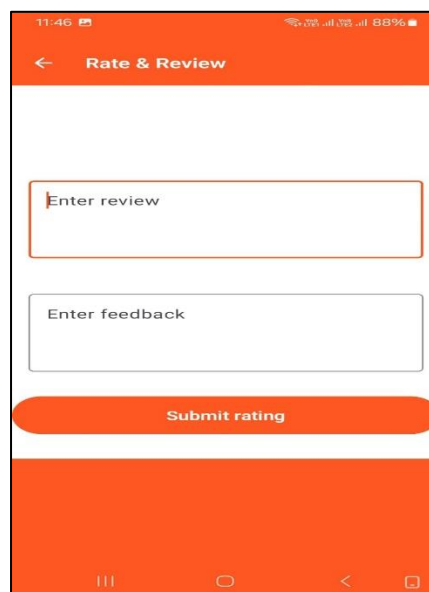


Figure 55 Review and Feedback

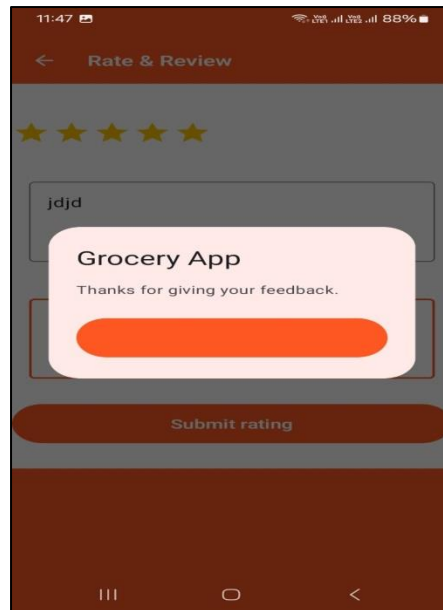


Figure 56 Thanks for Feedback

11. Log out Screen –

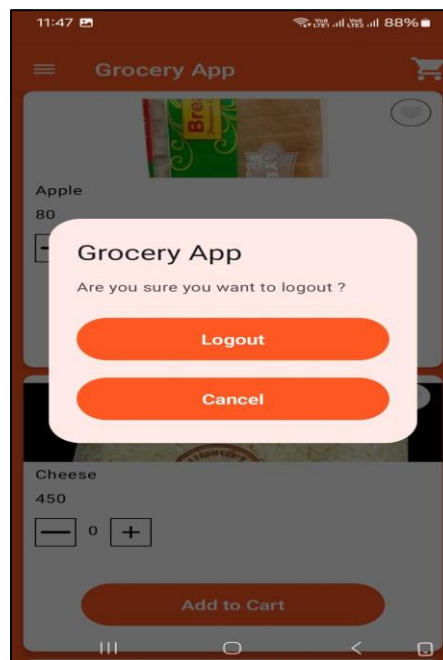


Figure 57 Log out Screen

Reference

Srivastava, A., Bhardwaj, S. and Saraswat, S., 2017, May. *SCRUM model for agile methodology*. In *2017 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 864-869). IEEE.

Cervone, H.F., 2011. *Understanding agile project management methods using Scrum*. *OCLC Systems & Services: International digital library perspectives*, 27(1), pp.18-22.

Kochhar, P.S., Thung, F., Nagappan, N., Zimmermann, T. and Lo, D., 2015, April. *Understanding the test automation culture of app developers*. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)* (pp. 1-10). IEEE.

McCann, T., 2011. *The art of the app store: The business of Apple development*. John Wiley & Sons.

Lai, S.T., Susanto, H. and Leu, F.Y., 2022. *Project management mechanism based on burndown chart to reduce the risk of software project failure*. In *Advances on Broad-Band Wireless Computing, Communication and Applications: Proceedings of the 16th International Conference on Broad-Band Wireless Computing, Communication and Applications (BWCCA-2021)* (pp. 197-205). Springer International Publishing.

Ault, A.C., Krogmeier, J.V. and Buckmaster, D., 2013. *Mobile, cloud-based farm management: a case study with trello on my farm*. In *2013 Kansas City, Missouri, July 21-July 24, 2013* (p. 1). American Society of Agricultural and Biological Engineers.