

Contents

Contents.....	1
Table of Figures.....	2
Objective of Application.....	5
Intended Users.....	5
Application DFD	5
Use Cases	Error! Bookmark not defined.
Sequence Diagram	6
Software Design Decisions	6
Libraries Used and Code Screenshots.....	7
Social, Legal, Ethical and Security Issue	13
Social Issues and Concerns	14
Legal Issues and Concerns	14
Ethical Issues and Concerns	15
Security Issues and Concerns.....	15
Development Strategy	16
Phases	16
Sprint Product Backlogs	16
Phase-1.....	17
Phase-2.....	20
Phase-3.....	21
Phase-4.....	23
Burn-down Chart.....	24
For Sprint 1.....	24
For Sprint 2.....	24
For Sprint 3.....	25
For Sprint 4.....	25
Firebase DB at backend and regarding API authentication and searching.....	26
Digital Library Application (UI).....	29
GIT	37
1 st commit	37
References	41

Table of Figures

Figure 1. Data Flow Diagram.....	5
Figure 2 Sequence Diagram	6

Software Design Decisions Libraries and Code

Figure 3 AppConfig Code and Libraries	7
Figure 4 routing Navigation Code and Libraries	8
Figure 5 BooksScreen UI Code and Libraries	9
Figure 6 DetailScreen UI Code and Libraries	10
Figure 7 LoginScreen UI Code and Libraries	11
Figure 8 utils Extension Code and Libraries	12
Figure 9 LibraryBorderField utils Code and Libraries	12
Figure 10 RoundedButton utils Code and Libraries	13
Figure 11 Social, Legal, Ethical and Security Issue	14

Development Strategy

Figure 12 Product backlog in trello	16
Figure 13 Sprint task in trello	17
Figure 14 Sprint card extended in trello	18
Figure 15 View of the sprint 1 and sprint 2	19
Figure 16 View of the sprint 3 and sprint 4	20
Figure 17 Continuation task with card in trello	21
Figure 18 Testing card in trello board	22
Figure 19 Done task in trello board	23
Figure 20 View of ignored and unable to complete tasks	23

Burn-down Chart

Figure 21 Burn down for sprint task 1	24
Figure 22 Burn down for sprint task 2	25
Figure 23 Burn down for sprint task 3	25
Figure 24 Burn down for sprint task 4	26

Firestore DB at backend and regarding API authentication and searching

Figure 25 Firebase Project page	26
Figure 26 Project overview in firebase	27
Figure 27 Authentication in firebase	27
Figure 28 Sign in methods in firebase	27
Figure 29 Authentication Sign in approach	28
Figure 30 Cloud firestore in DB.....	28
Figure 31 Start a collection pop up in DB	28

Digital Library Application (UI)

Figure 32 Welcome page of application.....	29
Figure 33 Registration page of application	29
Figure 34 Login Page of application	30
Figure 35 Home Page and Books Page of application-1.....	30
Figure 36 Home Page and Books Page of application-2.....	31
Figure 37 Home Page and Books Page of application-3.....	31
Figure 38 Home Page and Books Page of application-4.....	32
Figure 39 Home Page and Books Page of application-5.....	32
Figure 40 Home Page and Books Page of application-6.....	32
Figure 41 Home Page and Books Page of application-7.....	32
Figure 42 Home Page and Books Page of application-8.....	33
Figure 43 Home Page and Books Page of application-8.....	33
Figure 44 Home Page and Books Page of application-9.....	34
Figure 45 Home Page and Books Page of application-10.....	34
Figure 46 Detail page of application	35
Figure 47 Enquiry Page of application	35
Figure 48 Contact Us Page of application.....	36

Figure 49 About Us Page of application	36
Figure 50 Log out page of application	37

GIT

Figure 51 First Git commit part 1.....	37
Figure 52 First Git commit part 2.....	37
Figure 53 First Git commit part 3.....	38
Figure 54 First Git commit part 4.....	38
Figure 55 First Git commit part 5.....	38
Figure 56 Digital library application Git	38
Figure 57 Digital library application Git overview.....	39
Figure 58 Digital library application Git overview 2.....	39

Digital Library Mobile Application

Objective of Application

This online application's primary purpose was to give users an easy-to-use platform for managing and accessing library materials. The application provides features including searching the library's catalogue, checking out and retrieving eBooks, and making inquiries. Users can choose their preferred titles, browse, search, and ask questions about books that are available, and turn pages with ease for a captivating reading experience thanks to user-friendly interfaces.

Intended Users

- User

Application DFD

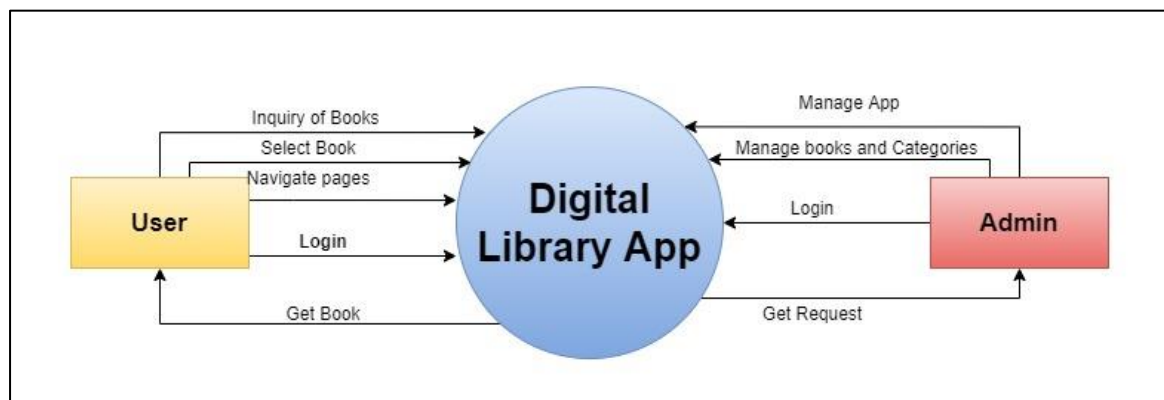


Figure 1. Data Flow Diagram

Sequence Diagram

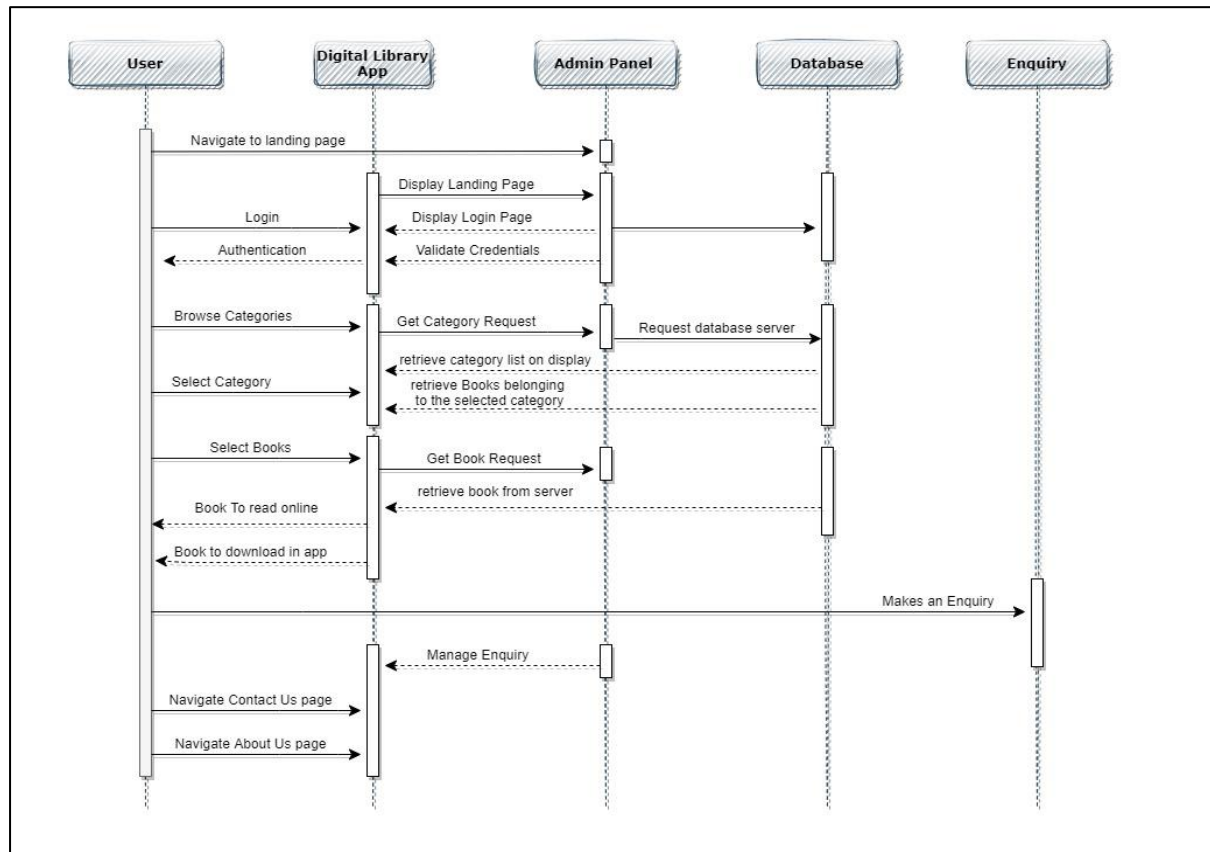


Figure 2 Sequence Diagram

Software Design Decisions

Tool used - Android Studio

Programming language - Java programming language

User Interface

- The majority of instructor capabilities may be built after those for students, with the former group given priority due to the fact that students have access to more features inside the app.
- If additional time were available, it was planned to provide a calendar and notes area for tutors.

Libraries Used and Code Screenshots

```
package com.digitallibrary.application

import android.app.Application
import com.digitallibrary.R
import com.digitallibrary.ui.model.BooksModel
import com.digitallibrary.ui.model.CategoryModel

class AppConfig : Application() {
    companion object {
        val categoryList = ArrayList<CategoryModel>().apply {
            add(
                CategoryModel(
                    "Action and adventure",
                    image = R.drawable.ic_action_and_adventure,
                    booksList = ArrayList<BooksModel>().apply {
                        add(
                            BooksModel(
                                name = "Jurassic Park (Jurassic Park, #1)",
                                image = R.drawable.ic_jurassic,
                                detail = "An astonishing technique for recovering and cloning dinosaur DNA
has been discovered. Now humankind's most thrilling fantasies have come true. Creatures extinct for eons
roam Jurassic Park with their awesome presence and profound mystery, and all the world can visit them—for a
price.\nUntil something goes wrong. . . \nIn Jurassic Park, Michael Crichton taps all his mesmerizing
talent and scientific brilliance to create his most electrifying technothriller."
                            )
                        )
                    }
                )
            add(
                BooksModel(
                    name = "The Hobbit",
                    image = R.drawable.ic_hobbit,
                    detail = "In a hole in the ground there lived a hobbit. Not a nasty, dirty,
wet hole, filled with the ends of worms and an oozy smell, nor yet a dry, bare, sandy hole with nothing in
it to sit down on or to eat: it was a hobbit-hole, and that means comfort.\nWritten for J.R.R. Tolkien's own
children, The Hobbit met with instant critical acclaim when it was first published in 1937. Now recognized
as a timeless classic, this introduction to the hobbit Bilbo Baggins, the wizard Gandalf, Gollum, and the
spectacular world of Middle-earth recounts of the adventures of a reluctant hero, a powerful and dangerous
```

Figure 3AppConfig Code and Libraries

```

package com.digitallibrary.routing

import androidx.compose.runtime.Composable
import androidx.navigation.compose.NavHost
import androidx.navigation.compose.composable
import androidx.navigation.compose.rememberNavController
import com.digitallibrary.ui.about_us.AboutUsScreen
import com.digitallibrary.ui.booksScreen.BooksScreen
import com.digitallibrary.ui.contact_us.ContactUsScreen
import com.digitallibrary.ui.detail.DetailScreen
import com.digitallibrary.ui.enquiryScreen.EnquiryScreen
import com.digitallibrary.ui.login.LoginScreen
import com.digitallibrary.ui.main.MainScreen
import com.digitallibrary.ui.register.RegisterScreen
import com.digitallibrary.ui.splash.SplashScreen

@Composable
fun Navigation() {
    val navController = rememberNavController()

    NavHost(
        navController = navController,
        startDestination = Screen.SplashScreen.route
    ) {
        composable(route = Screen.SplashScreen.route) {
            SplashScreen(navController = navController)
        }
        composable(route = Screen.LoginScreen.route) {
            LoginScreen(navController = navController)
        }
        composable(route = Screen.RegisterScreen.route) {
            RegisterScreen(navController = navController)
        }
        composable(route = Screen.MainScreen.route) {
            MainScreen(navController = navController)
        }
    }
}

```

Figure 4 routingNavigation Code and Libraries

```

package com.digitallibrary.ui.booksScreen

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.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.layout.ContentScale
import androidx.compose.ui.platform.LocalContext
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.text.input.KeyboardType
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import com.digitallibrary.R
import com.digitallibrary.application.AppConfig
import com.digitallibrary.routing.Screen
import com.digitallibrary.ui.model.BooksModel
import com.digitallibrary.ui.model.CategoryModel
import com.digitallibrary.ui.theme.DigitalLibraryAppTheme
import com.digitallibrary.ui.theme.black
import com.digitallibrary.ui.theme.white
import com.digitallibrary.ui.theme.brown
import com.digitallibrary.utils.LibraryBorderField
import com.digitallibrary.utils.RoundedButton
import com.digitallibrary.utils.isValidEmail

@OptIn(ExperimentalMaterial3Api::class, ExperimentalComposeUiApi::class)
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@Composable
fun BooksScreen(navController: NavController, pos: Int) {
    val context = LocalContext.current
    val scrollState = rememberScrollState()
    val bookList = ArrayList<BooksModel>().apply {
        addAll(AppConfig.categoryList[pos].booksList)
    }

```

Figure 5 BooksScreen UI Code and Libraries

```

package com.digitallibrary.ui.detail

import android.annotation.SuppressLint
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.icons.Icons
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.layout.ContentScale
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.painterResource
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 com.digitallibrary.routing.Screen
import com.digitallibrary.ui.theme.DigitalLibraryAppTheme
import com.digitallibrary.ui.theme.brown
import com.digitallibrary.ui.theme.white
import com.digitallibrary.utils.RoundedButton

@OptIn(ExperimentalMaterial3Api::class, ExperimentalComposeUiApi::class)
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@Composable
fun DetailScreen(navController: NavController, name: String, image: Int, details: String) {
    val context = LocalContext.current
    val scrollState = rememberScrollState()
    DigitalLibraryAppTheme {
        Scaffold {
            Column(
                modifier = Modifier
                    .fillMaxSize()
                    .background(color = brown)
                    .padding(top = 40.dp)
                    .height(IntrinsicSize.Max)
                    .verticalScroll(scrollState)
            ) {
                SmallTopAppBar(
                    title = {
                        Text(

```

Figure 6DetailScreen UI Code and Libraries

```

package com.digitallibrary.ui.login

import android.annotation.SuppressLint
import android.util.Log
import android.widget.Toast
import androidx.compose.foundation.*
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.CircleShape
import androidx.compose.foundation.shape.RoundedCornerShape
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.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.painterResource
import androidx.compose.ui.text.TextStyle
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.firestore.ktx.firestore
import com.google.firebase.ktx.Firebase
import com.digitallibrary.R
import com.digitallibrary.routing.Screen
import com.digitallibrary.ui.restaurantPreference.RestaurantPreference
import com.digitallibrary.ui.theme.*
import com.digitallibrary.utils.LibraryBorderField
import com.digitallibrary.utils.RoundedButton
import com.digitallibrary.utils.isValidEmail
import com.google.firebase.auth.FirebaseAuth

@OptIn(ExperimentalMaterial3Api::class, ExperimentalComposeUiApi::class)
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@Composable
fun LoginScreen(navController: NavController) {
    val context = LocalContext.current
    val preference = remember {
        RestaurantPreference(context)
    }
    var isLibraryLogin by remember { mutableStateOf(false) }
    var email by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    val firebaseAuth = FirebaseAuth.getInstance()
    DigitalLibraryAppTheme {
        Scaffold {

```

Figure 7 LoginScreen UI Code and Libraries

```

package com.digitallibrary.utils

import java.util.regex.Pattern

fun isValidEmail(s: String): Boolean {
    val emailPattern = ("^((\\w-|\\.)+\\w-|([a-zA-Z]{1}|[\\w-]{2,}))@"
        + "((((\\d-|\\d{1,2}|25[0-5]|2[0-4][0-9]))\\.([0-1]?"
        + "[0-9]{1,2}|25[0-5]|2[0-4][0-9]))\\.([0-1]?"
        + "[0-9]{1,2}|25[0-5]|2[0-4][0-9]))\\.([0-1]?"
        + "[0-9]{1,2}|25[0-5]|2[0-4][0-9])){1}|"
        + "([a-zA-Z]+[\\w-|\\.)+\\w-|([a-zA-Z]{2,4})$")
    val pattern = Pattern.compile(emailPattern)
    val matcher = pattern.matcher(s)

    return !matcher.matches()
}

```

Figure 8utilsExtension Code and Libraries

```

package com.digitallibrary.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.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.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.digitallibrary.ui.theme.black

@OptIn(ExperimentalMaterial3Api::class, ExperimentalComposeUiApi::class)
@Composable
fun LibraryBorderField(
    value: String,
    placeholder: String,
    onChange: (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,
) {
    OutlinedTextField(
        value = value,
        shape = RoundedCornerShape(10.dp),
        keyboardOptions = KeyboardOptions(keyboardType = keyboardType, capitalization = keyboardCapitalization),
        visualTransformation = visualTransformation,
        enabled = isEnabled,
        readOnly = readOnly,
        label = label,
        maxLines = 1,
    )
}

```

Figure 9LibraryBorderFieldutils Code and Libraries

```

package com.digitallibrary.utils

import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.PaddingValues
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonDefaults
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Brush
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.digitallibrary.ui.theme.black
import com.digitallibrary.ui.theme.brown

@Composable
fun RoundedButton(text: String, isEnabled : Boolean = true, onClick: () -> Unit, textColor : Color = Color.White) {
    GradientButton(
        onClick = onClick,
        modifier = Modifier.padding(vertical = 5.dp).fillMaxWidth(),
        textColor = textColor,
        isEnabled = isEnabled,
        gradient = Brush.horizontalGradient(listOf(brown, brown)),
        text = text
    )
}

@Composable
fun GradientButton(
    text: String,
    gradient : Brush,
    modifier: Modifier = Modifier,
    onClick: () -> Unit = { },
    textColor: Color,
    isEnabled: Boolean
) {
    Button(
        modifier = modifier,
        colors = ButtonDefaults.buttonColors(containerColor = Color.Transparent, contentColor = textColor),
        contentPadding = PaddingValues(),
        shape = RoundedCornerShape(30.dp),
        enabled = isEnabled,
        onClick = { onClick() },
    ) {
        Box(

```

Figure 10 RoundedButtonutils Code and Libraries

Social, Legal, Ethical and Security Issue

An Android application called Digital Library give users fascinating new options for accessing knowledge and provide online book access. However, when to developing an app, some social, legal, ethical, and security consideration must be made. Through the resolution of problems, the Digital Library app can advance literacy, knowledge exchanging, and access to information responsibility.

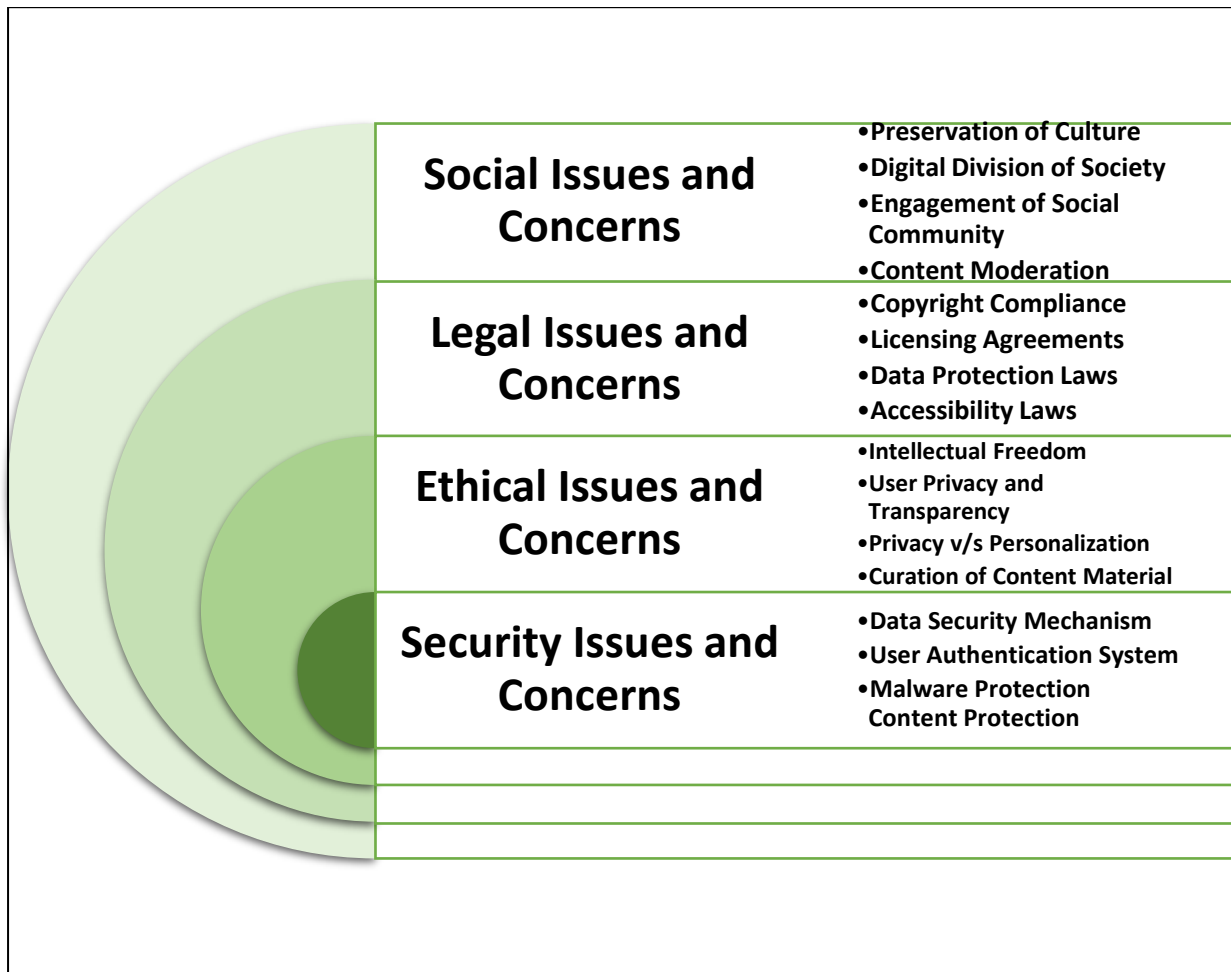


Figure 11 Social, Legal, Ethical and Security Issue

Social Issues and Concerns

- **Engagement of Social Community**→ For readers to have a feeling "of" community and "cooperation," the Digital Library Android app should encourage community "interaction" with characteristics like user discussion boards, reviews, and online book clubs.
- **Content Moderation**→ The Android app for the Digital Library should weigh the "openness" of knowledge against "possibility" "offensive" or dangerous content.

Legal Issues and Concerns

- **Copyright Compliance**→ In order to avoid legal "troubles" and maintain the rights of writers, the Digital Library Android app must abide by "copyrights" regulations "and" secure the required licensing and "permits for" displaying copyrighted materials in the form of digital book "distribution."

- **Licensing Agreements**→ For the diffusion of copyrighted goods "within" the digital library app, the developer of this digital library android app must secure compliance with license agreements with publishers "and" writers. The useful laws are-

Open-source software (OSS) licenses Act

Third-party SDKs (Software Development Kits) Act

- **Data Protection Laws**→To preserve user's "privacy" and stop unwanted access to "or" misuse of personal data, the Digital Library Android app must safeguard user data "in" compliance with data protection laws.
- **Accessibility Laws**→ For "screen" readers and text-to-speech functionality to be accessible to people with "impairments," the Digital Library Android app must "adhere towards" accessibility laws. The need act are-

Equality Act 2010

Public Sector Bodies (Websites and Mobile Applications) (Accessibility Standards) 2018

WCAG (Web Content Accessibility Guidelines) 2.1 AA

Ethical Issues and Concerns

- **Intellectual Freedom**→ By granting unrestricted access to a variety of opinions and content, the Digital Library Android app should maintain the values "of" intellectual freedom while "consistently" adhering to legal "and" community norms.
- **User Privacy and Transparency**→ In addition to attempting to give users "controls" "over" their personal information, the Digital Library Android app should be "open" "and" honest about its data gathering methods and how it "uses" user information.
- **Privacy v/s Personalization**→ In order to "better" user experience, the Digital Library Android app should be balanced to the user "privacy" with tailored "recommendation."

Security Issues and Concerns

- **Data Security Mechanism**→ To "safeguarded" user data while it is being transmitted and stored, the Digital Library Android app should employ

strong encryption techniques. Additionally, it can lessen the chance of illegal access or data leaks.

- **User Authentication System**→ Strong authentication methods, like email authentication, should be "could utilize" by the Digital Library Android app to confirm user "identify" "and" stop illegal access to accounts.
- **Malware Protection**→ To "secure" against malware "that" could infect user devices, the Digital "Librarying" app should use the Safeguard technique.

Development Strategy

Agile methodology is use to make sure that the app is improved continuously and that it is adaptable enough to allowing for easily adding of new data or modifications, which finally results in the best application development!

Phases

Sprint Product Backlogs

In this Sprint Product Backlogs, the goal that need to be achieved while doing development of the Digital Library application, are mentioned-

- Seek for and prepare the Digital Library Application requirements.
- Development of Digital Library Application
- Code Design and Database Architecture for the Digital Library Program
- GIT Repository Deployment, Testing and UAT
- Documentations of planning, architecture and methods

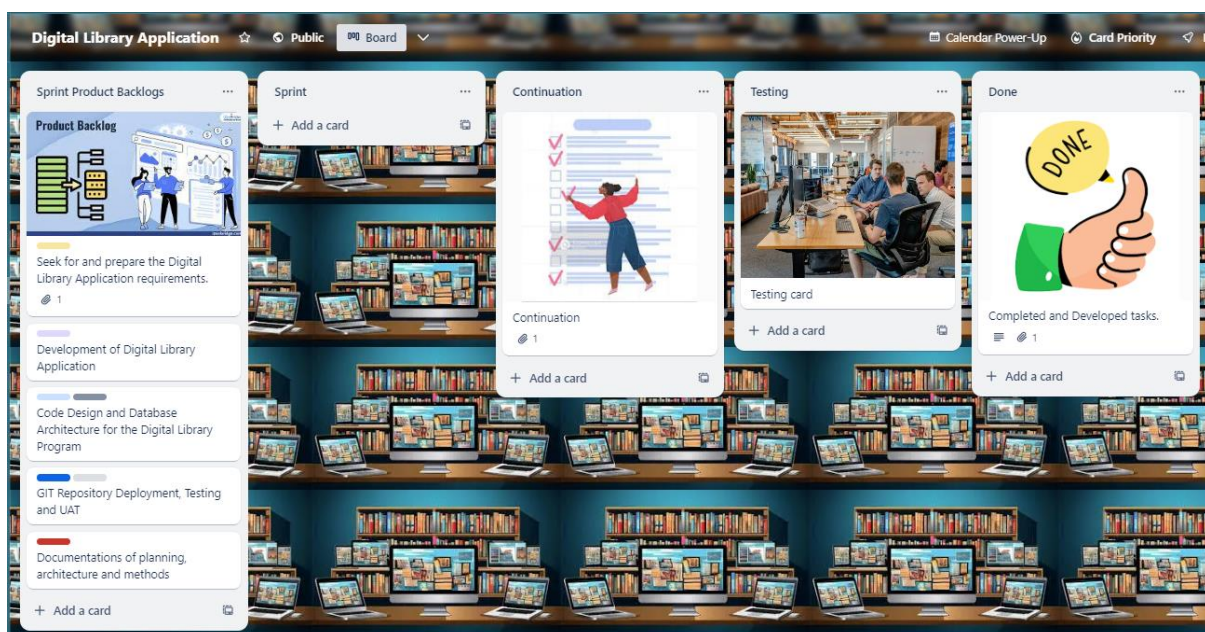


Figure 12 Product backlog in trello

Phase-1

The development of this Digital Library application is divided into four sprints.

1. Sprint-1: Research and Bring Apart the Requirements for Digital Library Application
2. Sprint-2: Designing of the Wire Frame with implementation of UI
3. Sprint-3: Coding, Git Repository for Version Control of UI design, Database schema
4. Sprint-4: Testing of mobile App, Deployment, UAT & Documentation

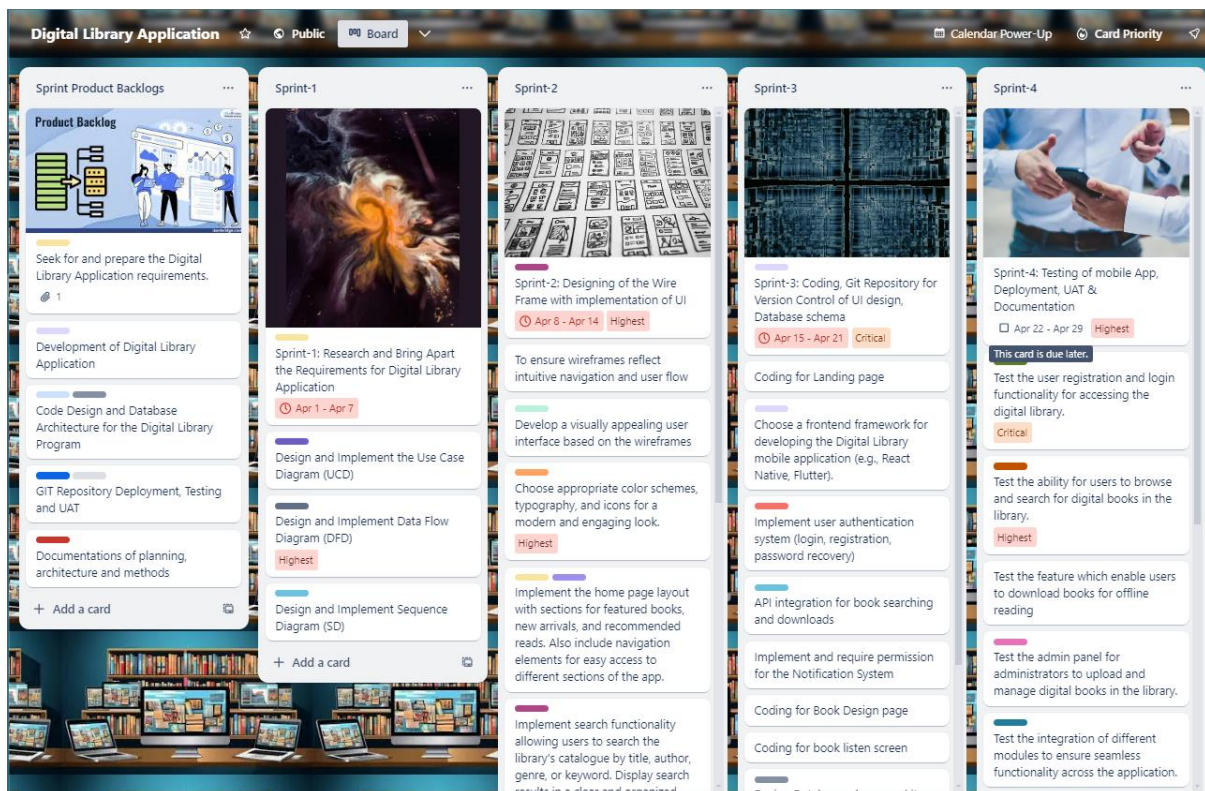


Figure 13 Sprint task in trello

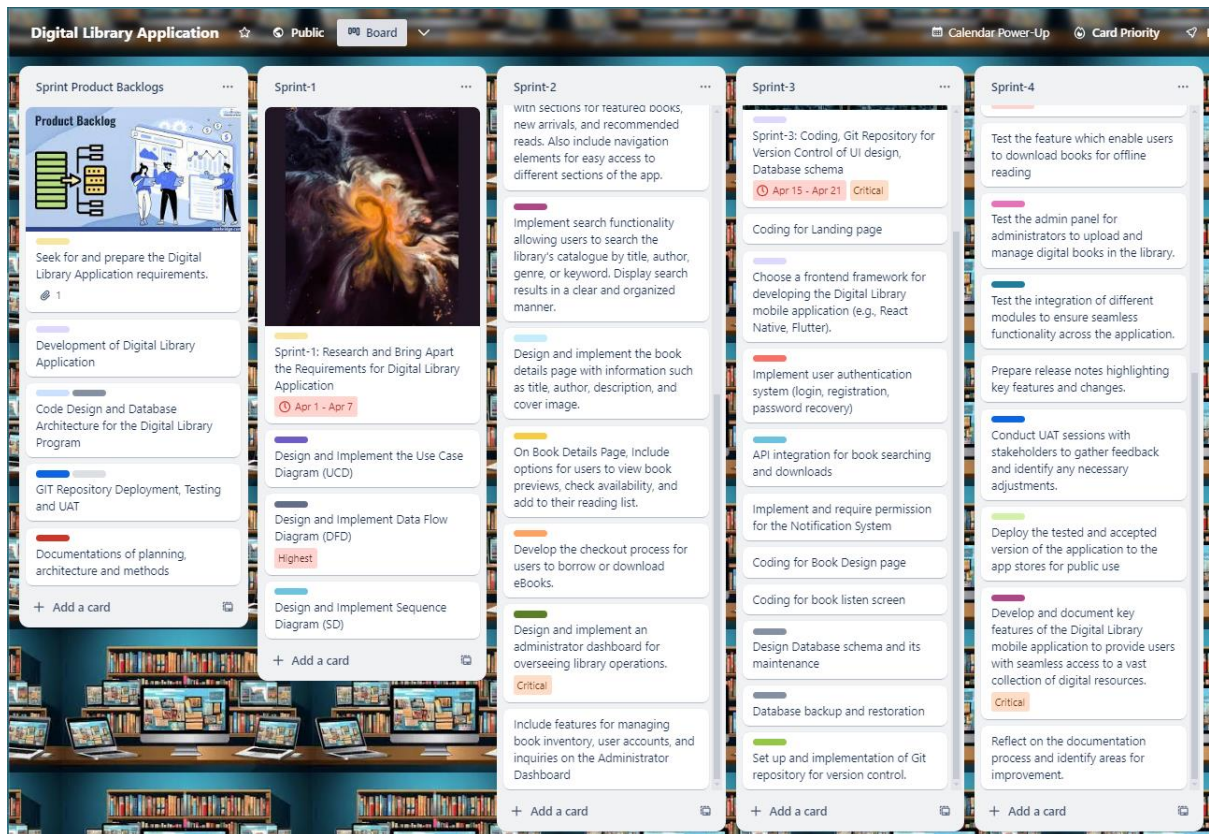


Figure 14 Sprint card extended in trello

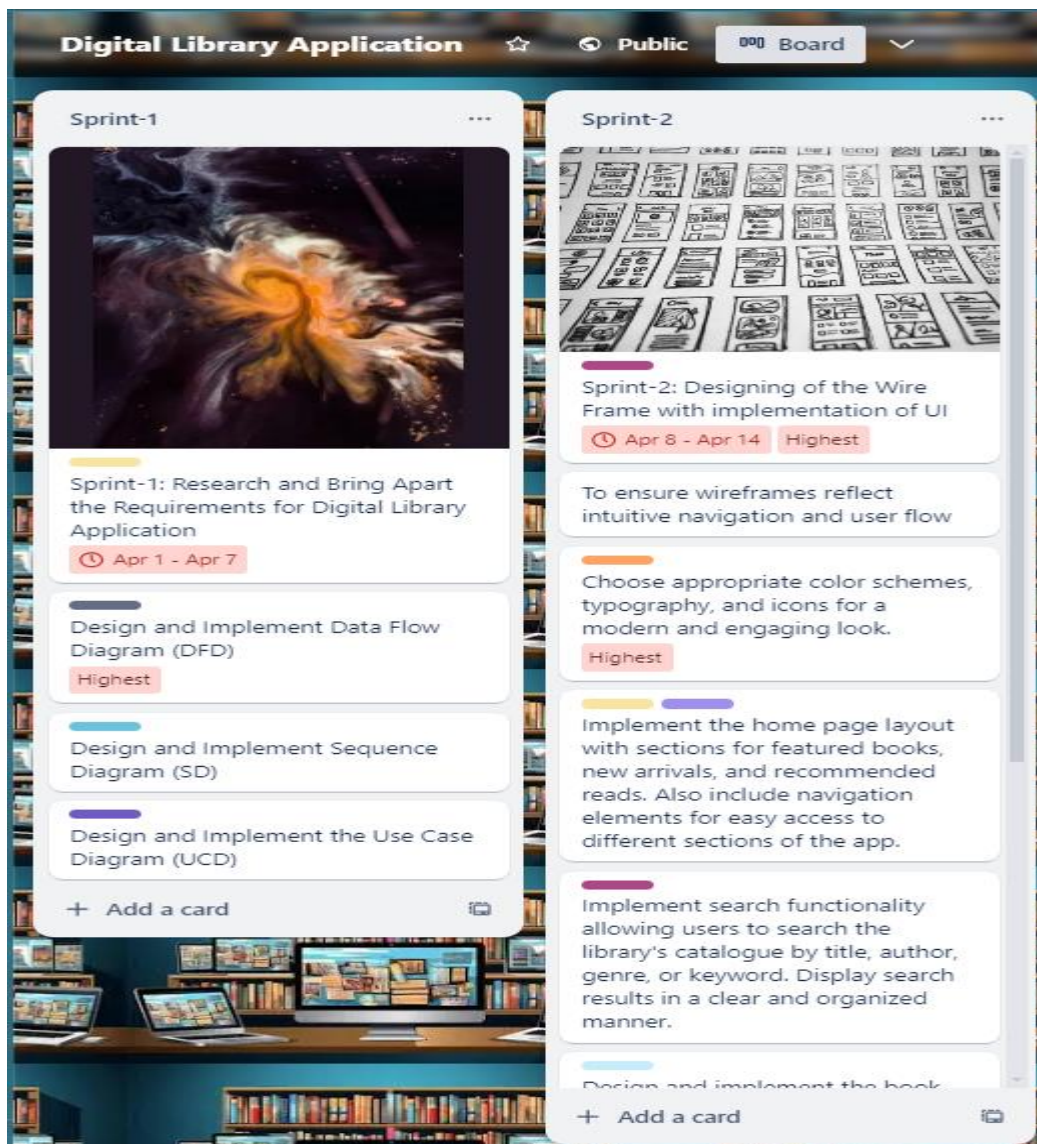


Figure 15 View of the sprint 1 and sprint 2

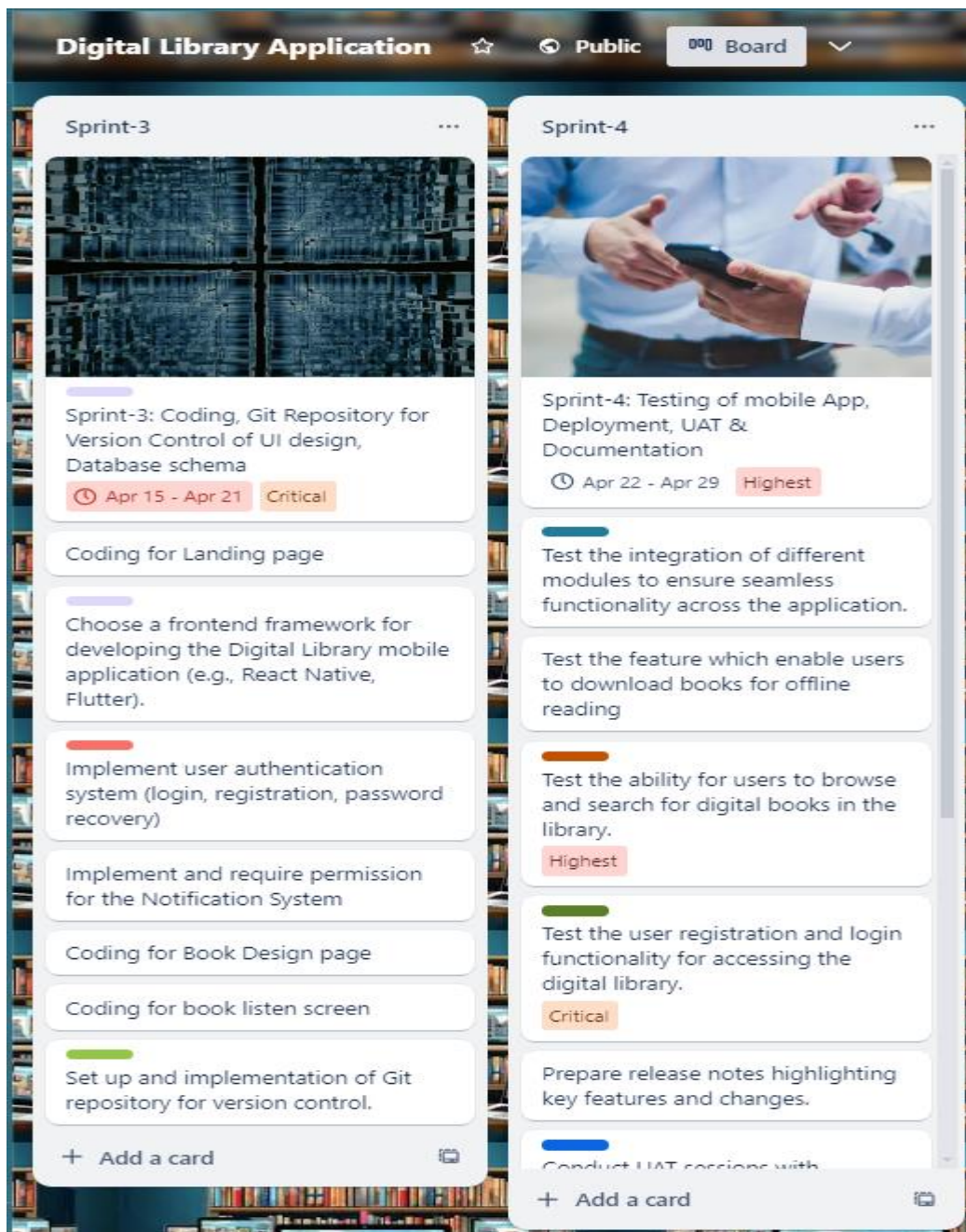


Figure 16 View of the sprint 3 and sprint 4

Phase-2

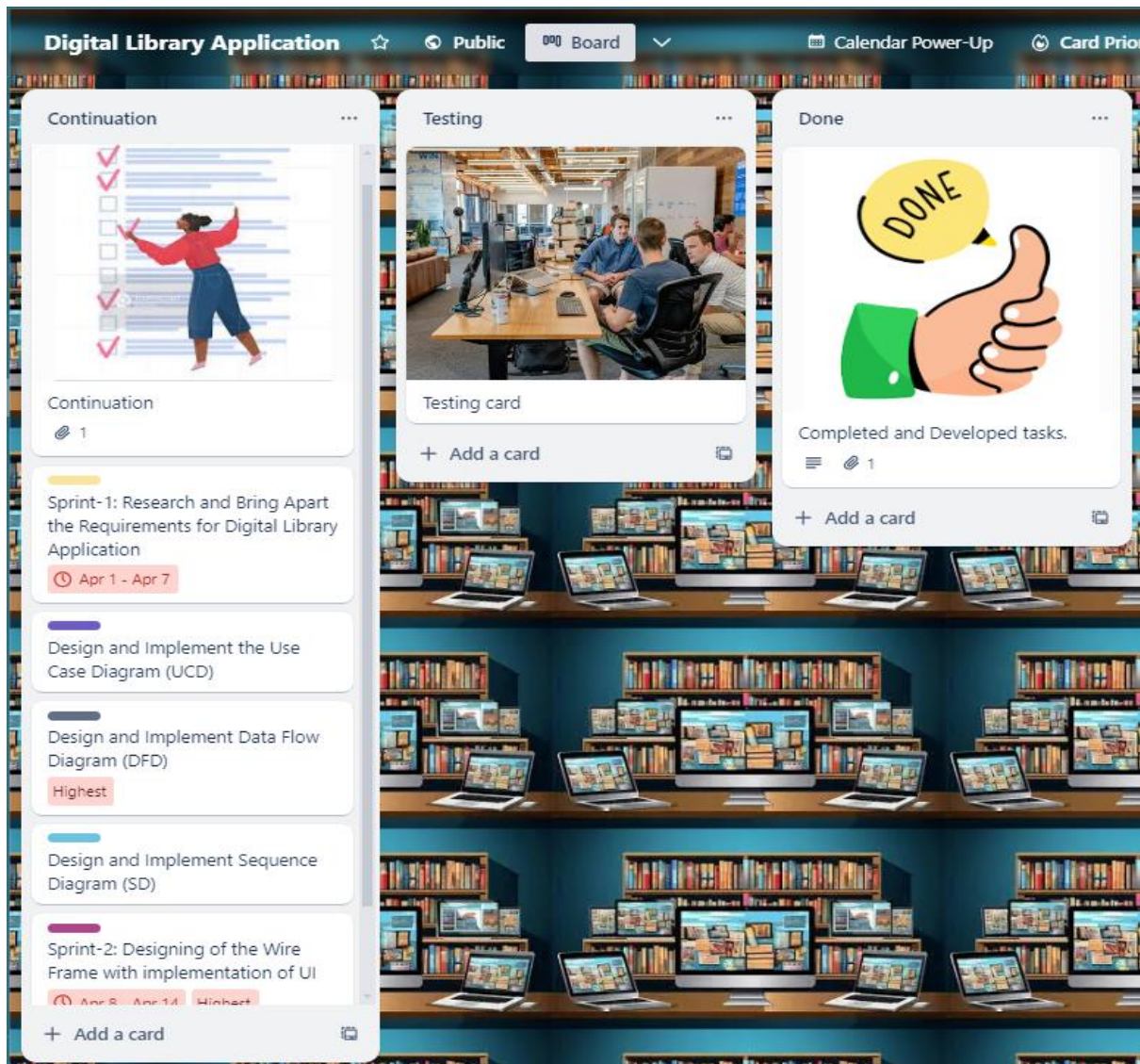


Figure 17 Continuation task with card in trello

Phase-3

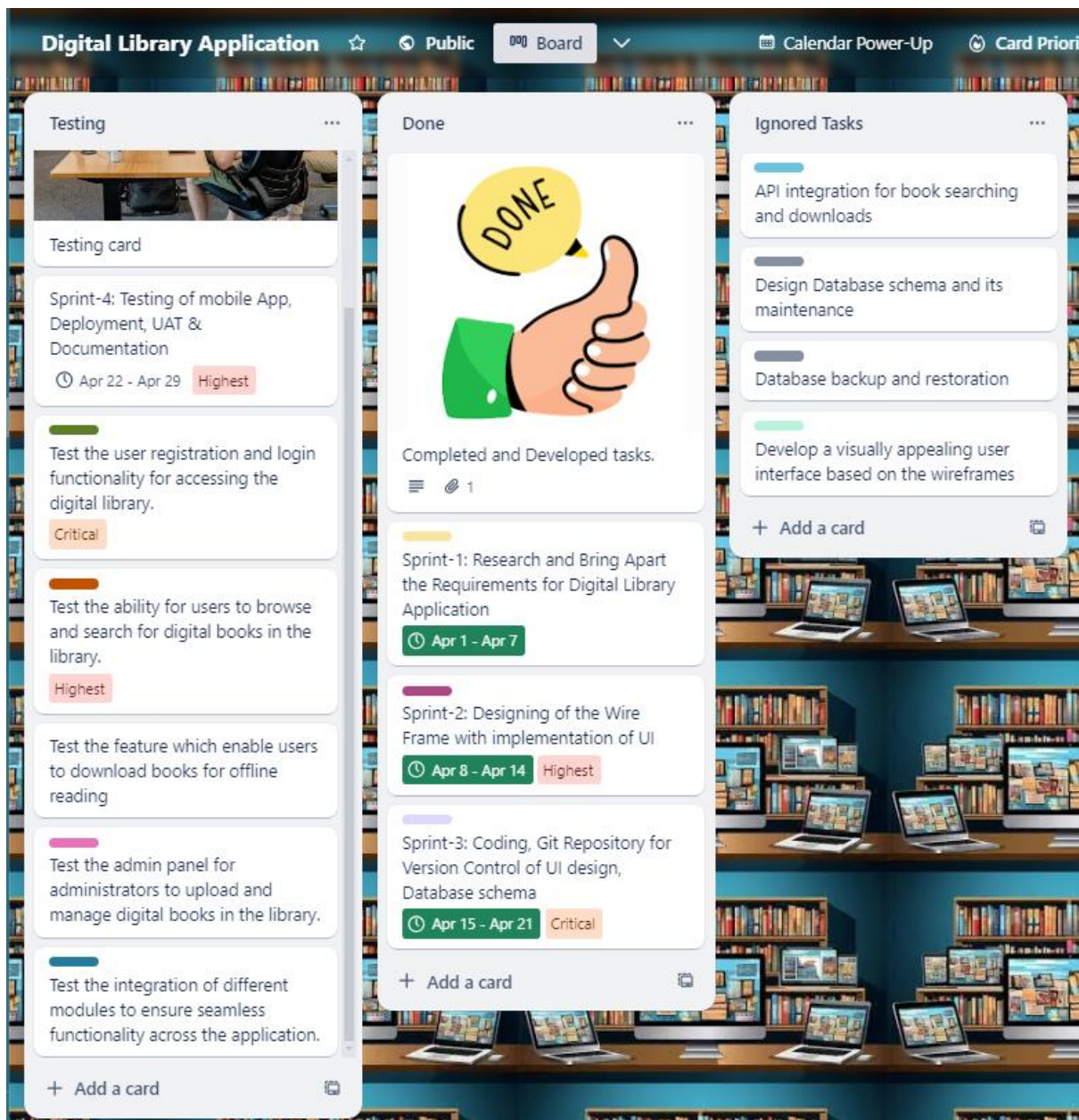


Figure 18 Testing card in trello board

Phase-4

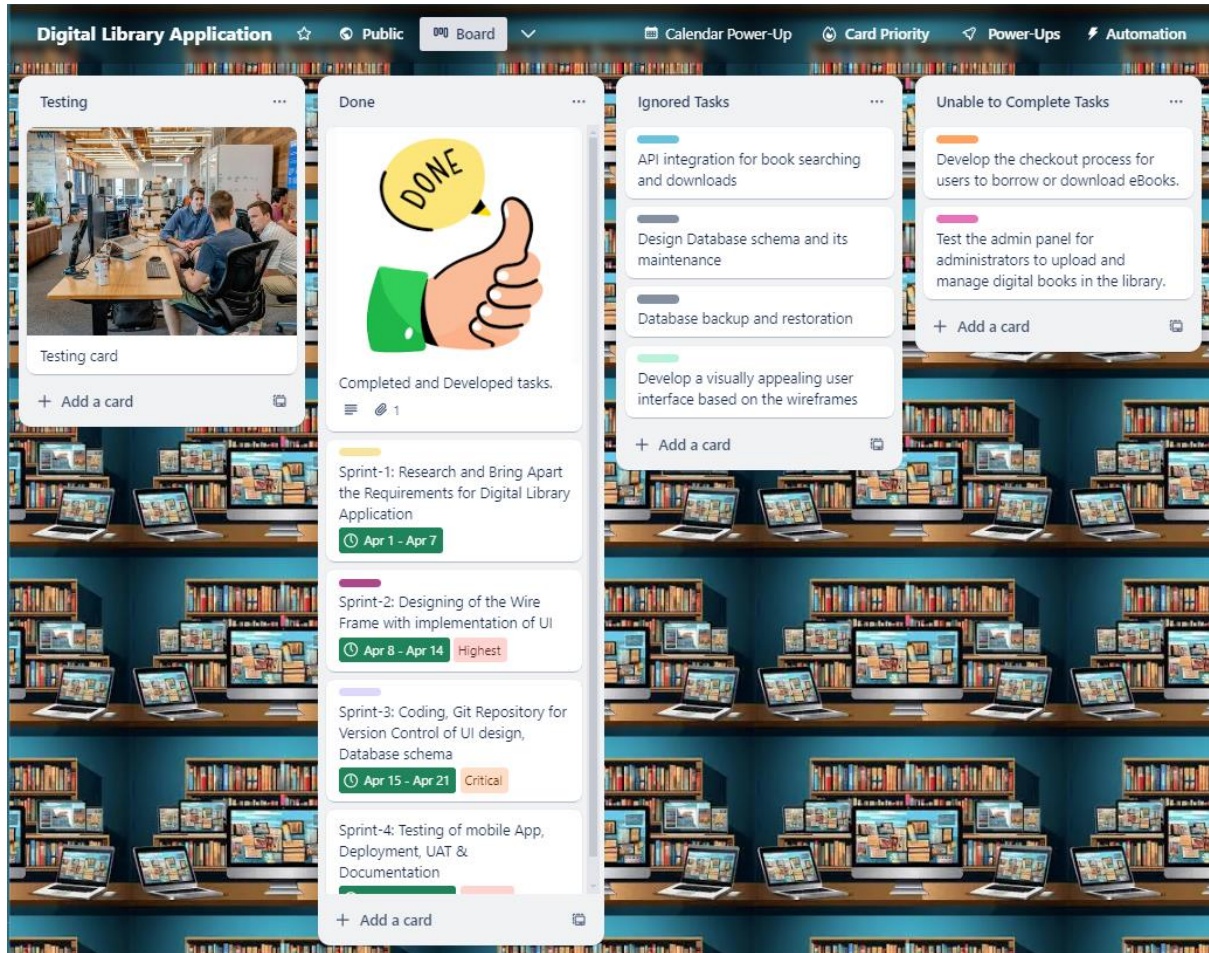


Figure 19 Done task in trello board

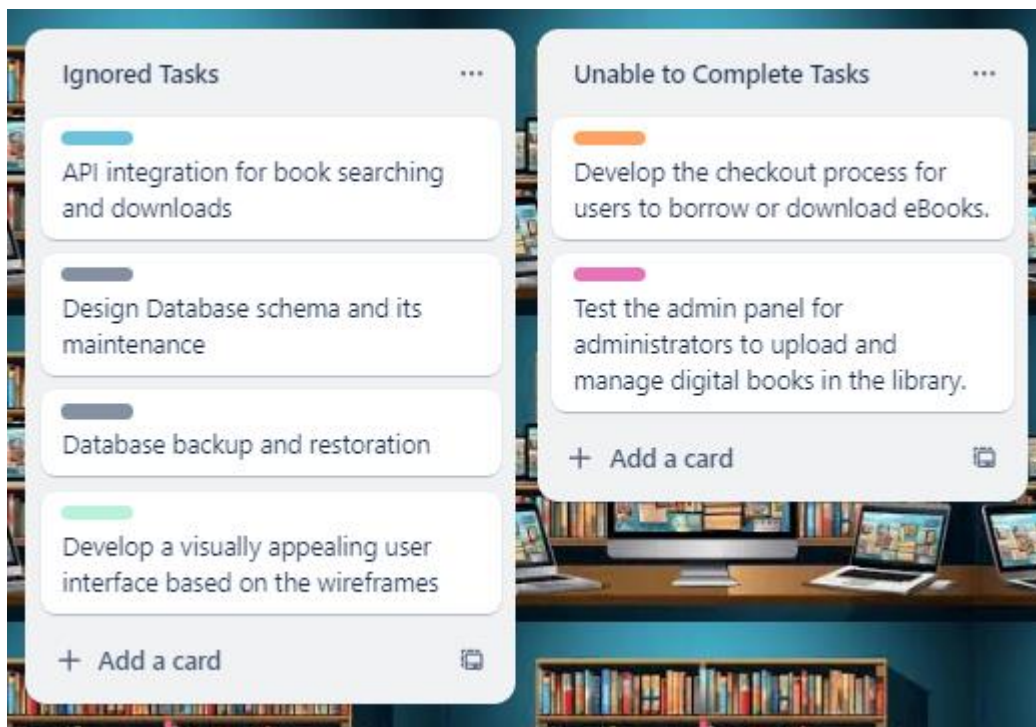


Figure 20 View of ignored and unable to complete tasks

Burn-down Chart

Burnt-down charts provide an easy-to-see overview of a project's status and indicate if the team is on schedule to finish the job within the allocated time. They aid in the early detection of possible issues, such as scope creep or jobs taking longer than anticipated, enabling teams to make adjustments as necessary to stay on schedule.

For Sprint 1

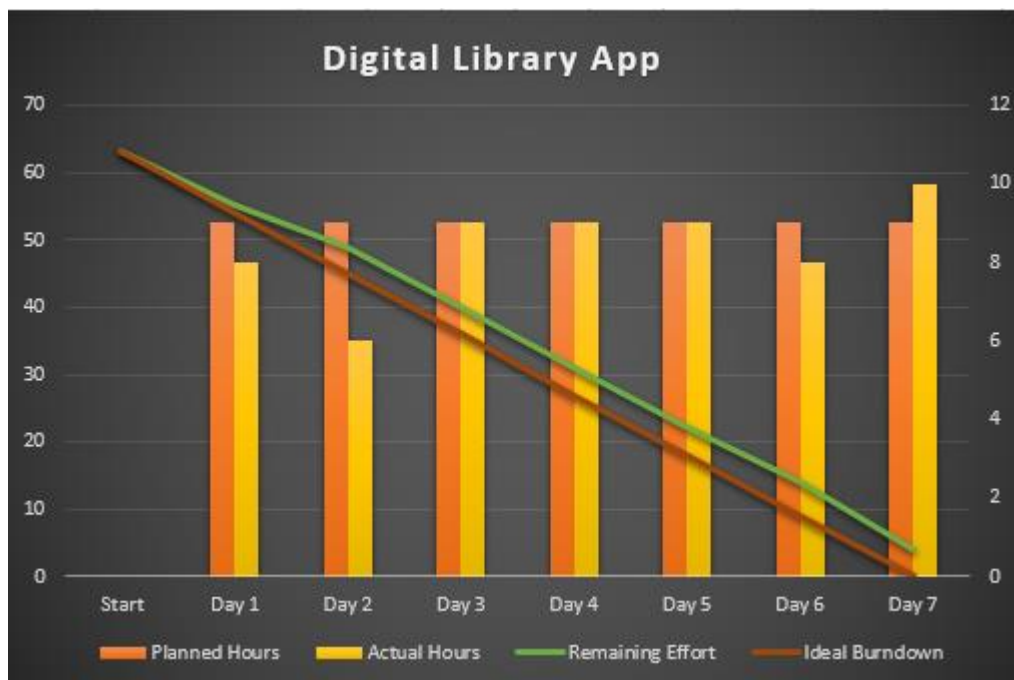


Figure 21 Burn down for sprint task 1

For Sprint 2

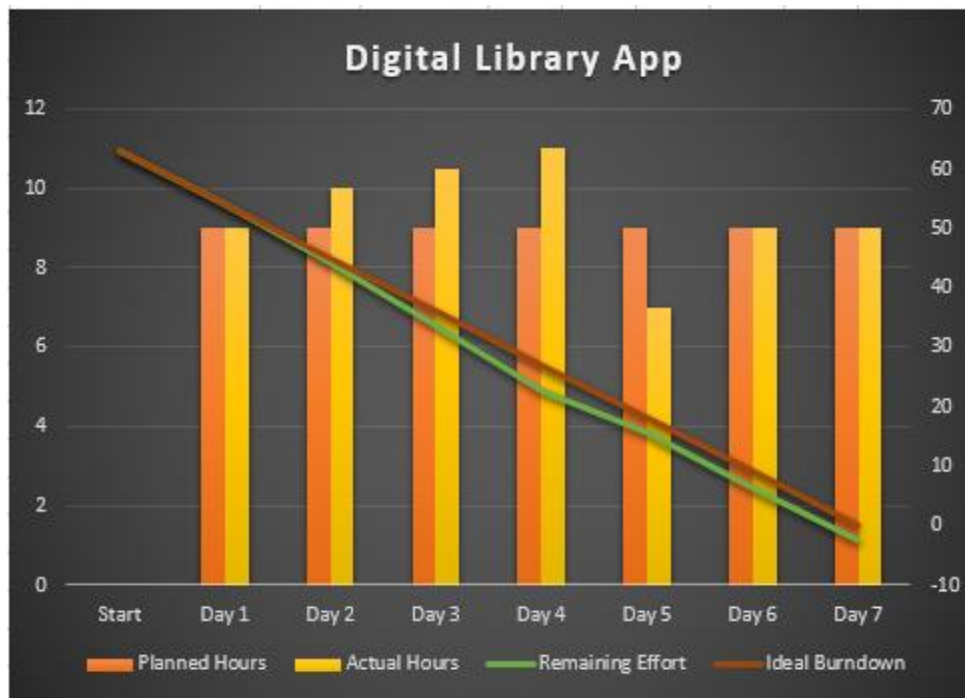


Figure 22 Burn down for sprint task 2

For Sprint 3

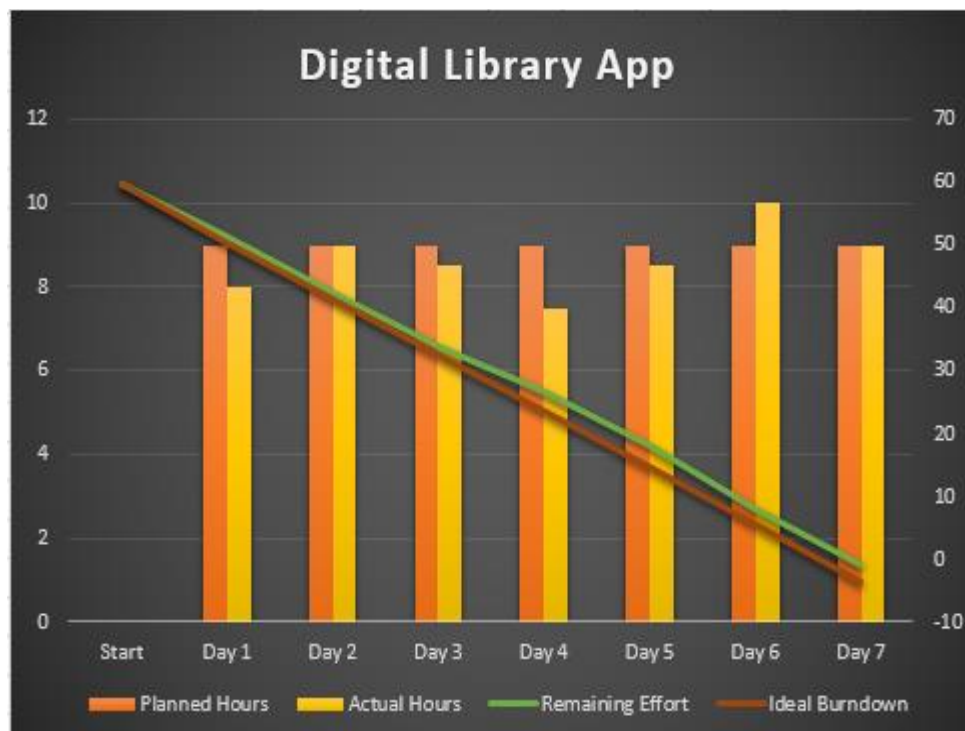


Figure 23 Burn down for sprint task 3

For Sprint 4

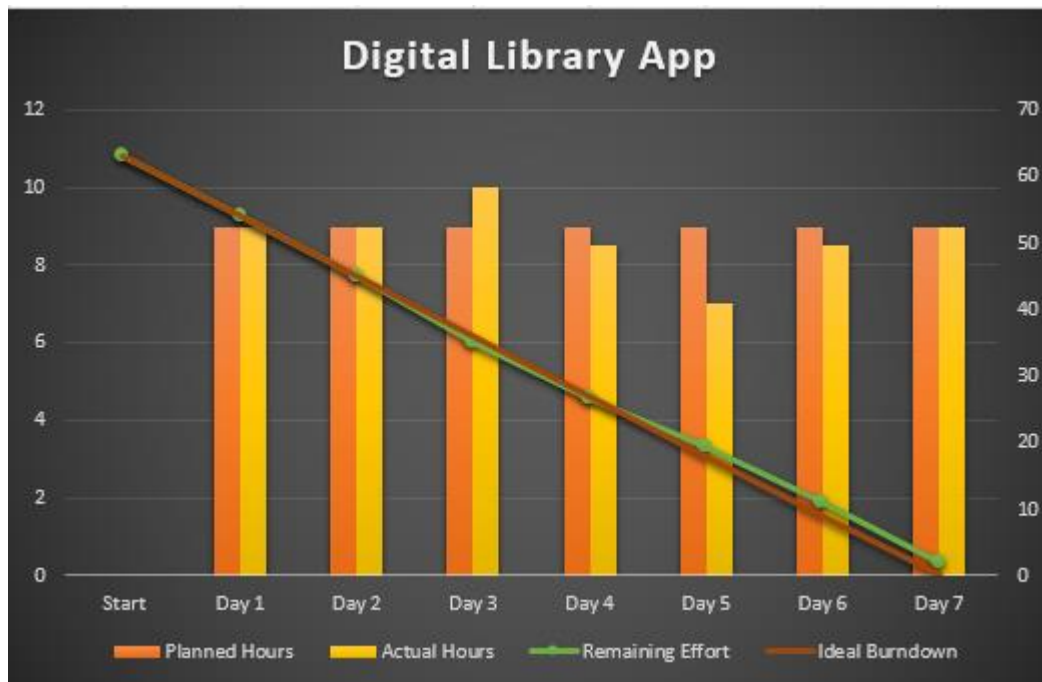


Figure 24 Burn down for sprint task 4

Firestore DB at backend and regarding API authentication and searching

Digital Library:

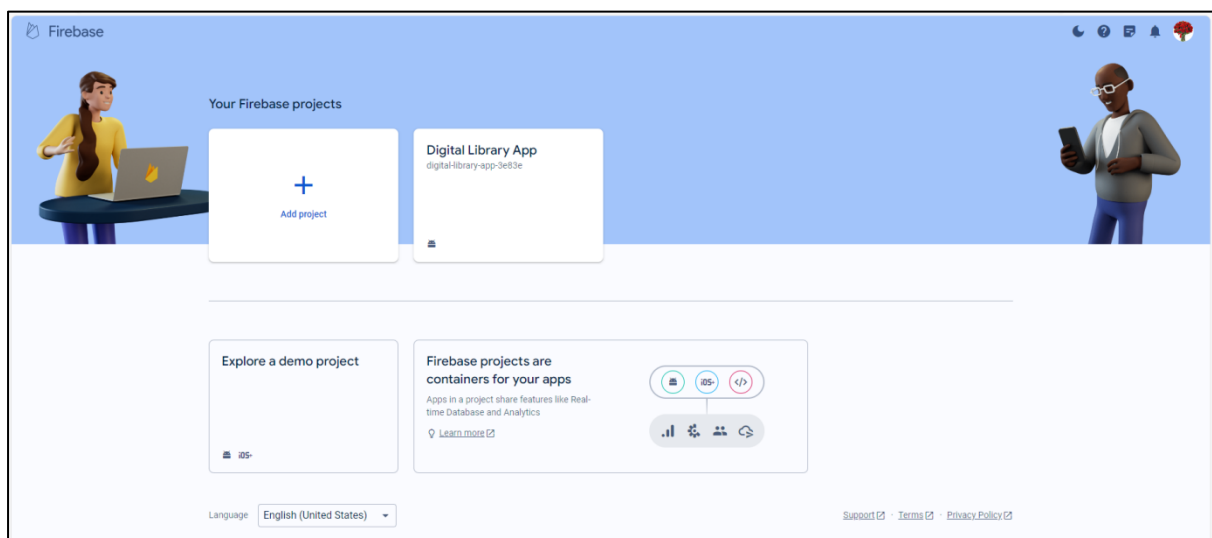


Figure 25 Firebase Project page

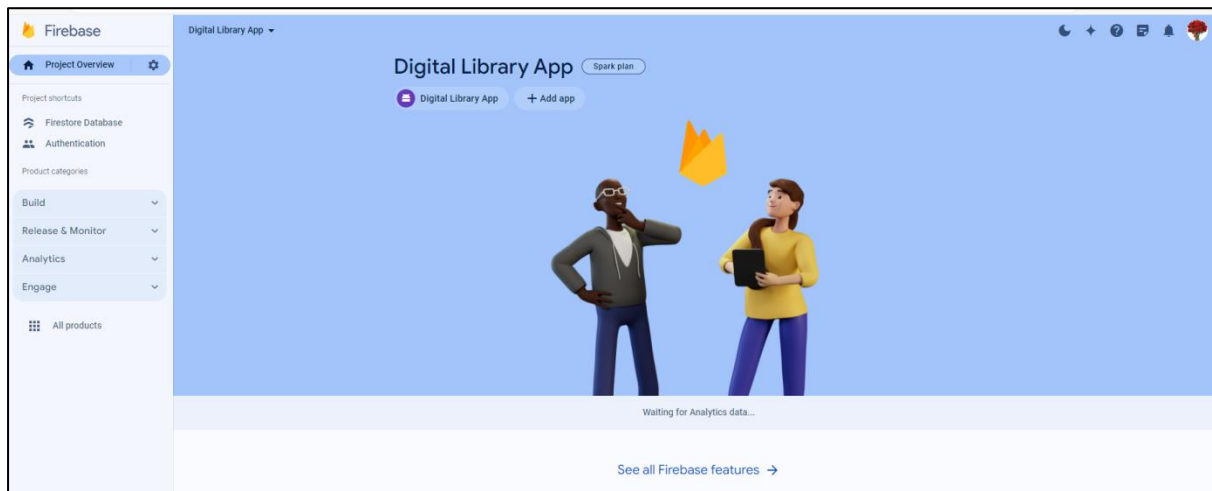


Figure 26 Project overview in firebase

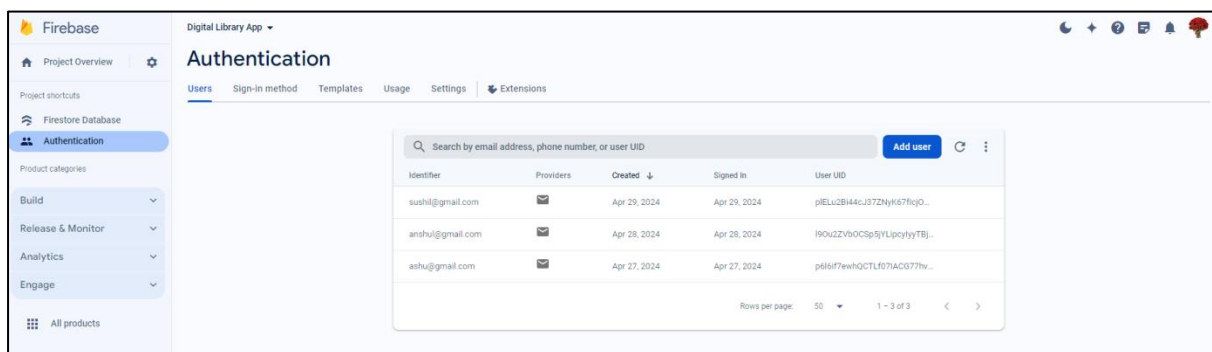


Figure 27 Authentication in firebase

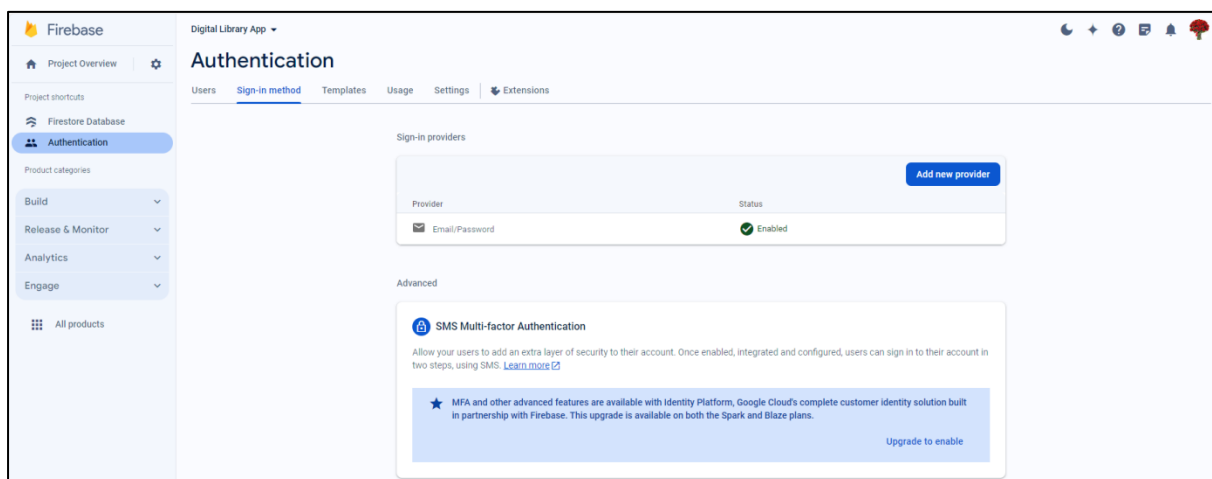


Figure 28 Sign in methods in firebase

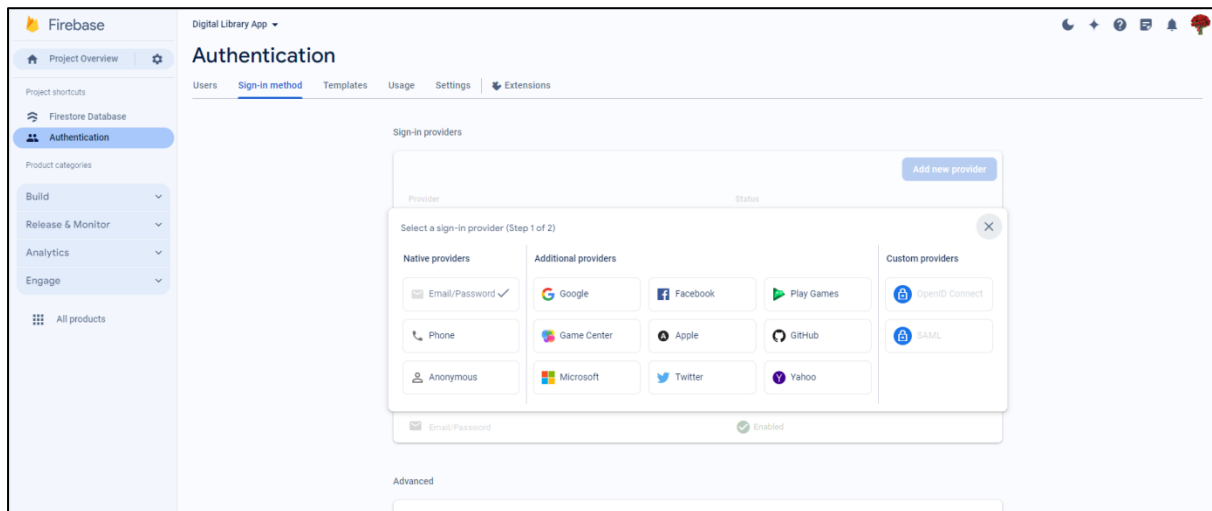


Figure 29 Authentication Sign in approach

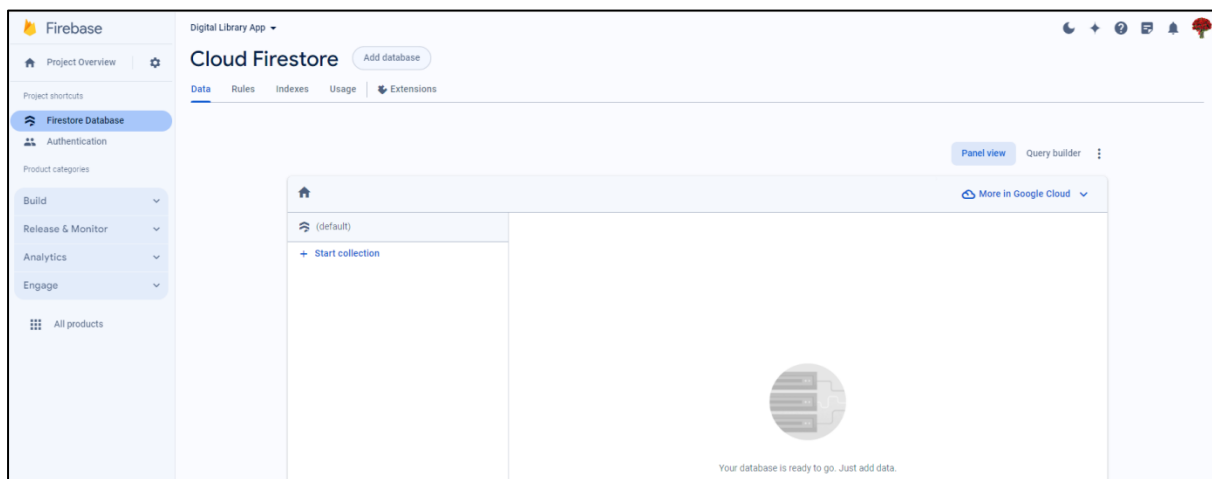


Figure 30 Cloud firestore in DB

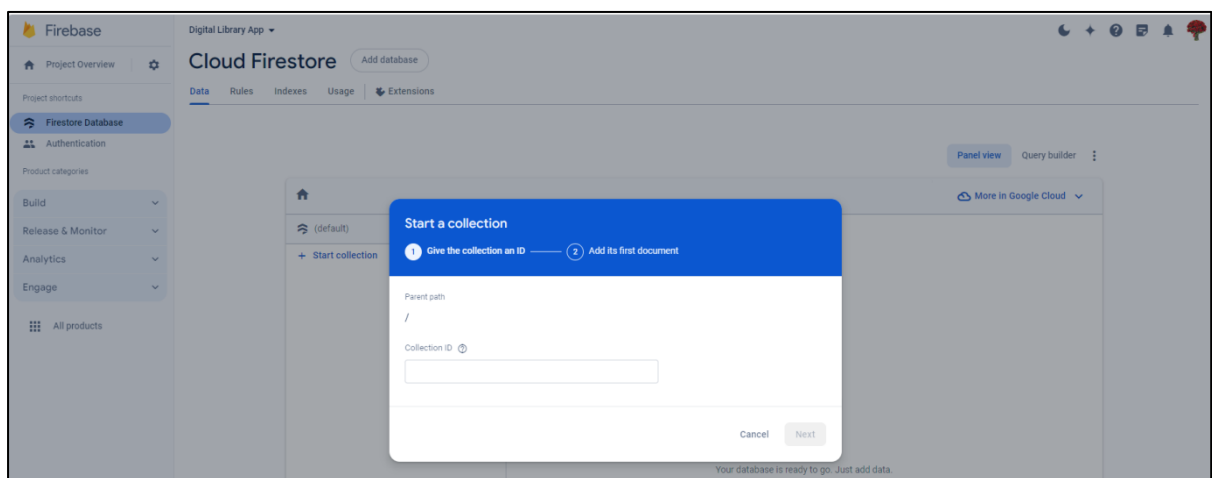


Figure 31 Start a collection pop up in DB

Digital Library Application (UI)

1. Welcome Page –

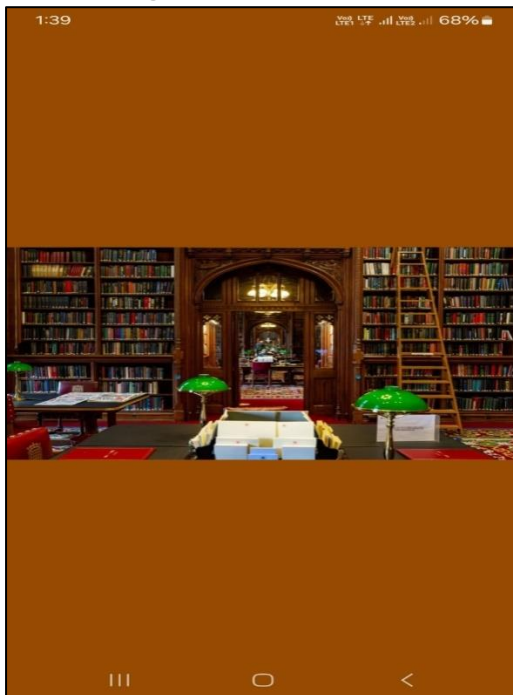


Figure 32 Welcome page of application

2. Registration Page –

A screenshot of a mobile application's registration page. The page has a white background. At the top, there is a brown header bar containing a circular image of the library interior. Below the header, there are three white input fields with rounded corners, each containing a placeholder text: "Enter name", "Enter email", and "Enter Password". Below these fields is a large, rounded brown button with the text "Register" in white. Underneath the button, there is a line of text: "Already have an account? Login". At the bottom of the screen, there are three small white icons: a list, a circle, and a left arrow.

Figure 33Registration page of application

3. Login Page –

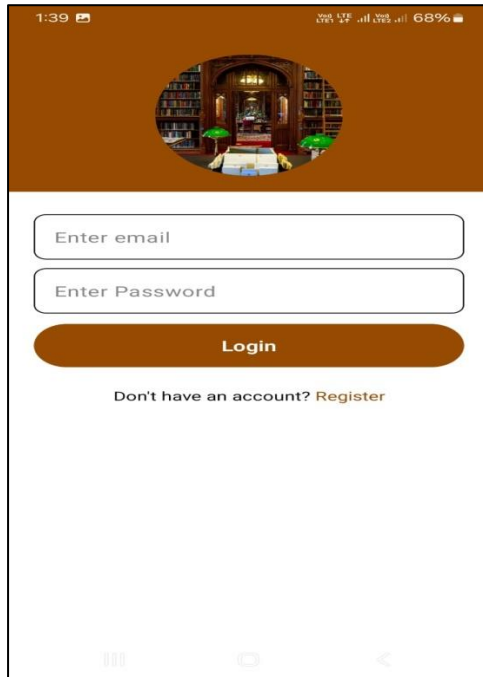


Figure 34 Login Page of application

4. Home Page and Books Page –

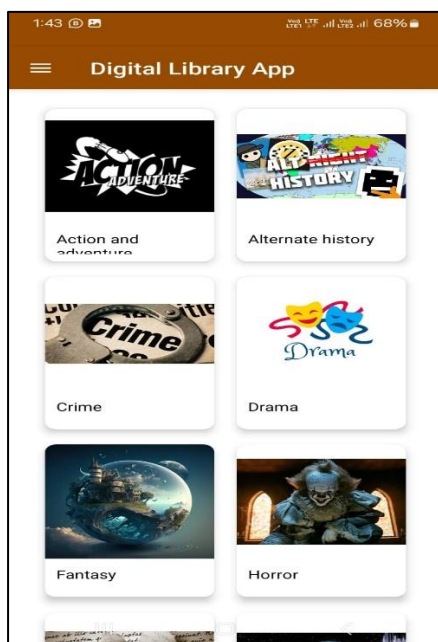


Figure 35 Home Page and Books Page of application-1

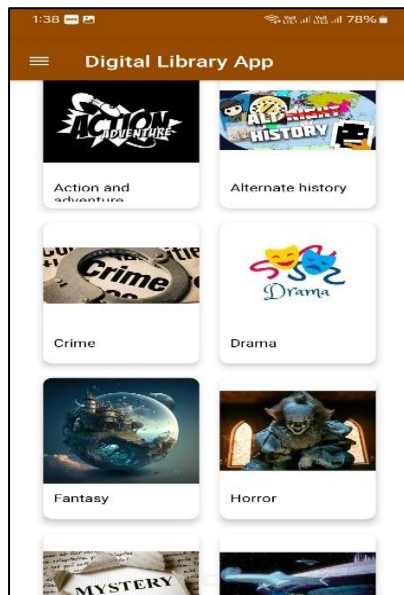


Figure 36 Home Page and Books Page of application-2

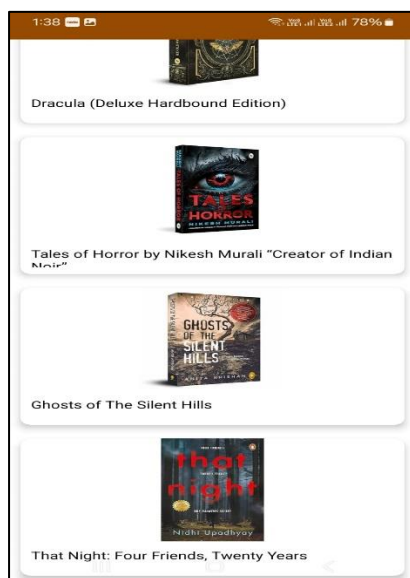


Figure 37 Home Page and Books Page of application-3



Figure 38 Home Page and Books Page of application-4



Figure 39 Home Page and Books Page of application-5

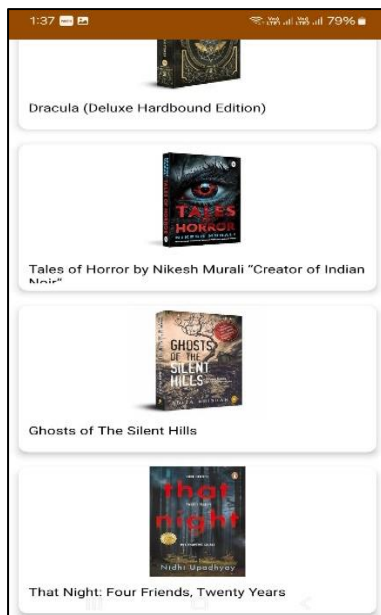


Figure 40 Home Page and Books Page of application-6



Figure 41 Home Page and Books Page of application-7



Figure 42 Home Page and Books Page of application-8

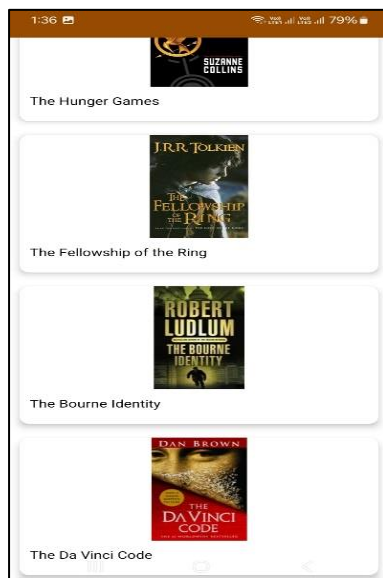


Figure 43 Home Page and Books Page of application-8



Figure 44 Home Page and Books Page of application-9

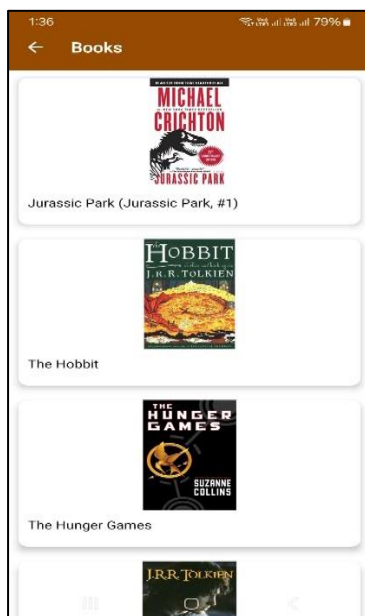


Figure 45 Home Page and Books Page of application-10

5. Details Page –

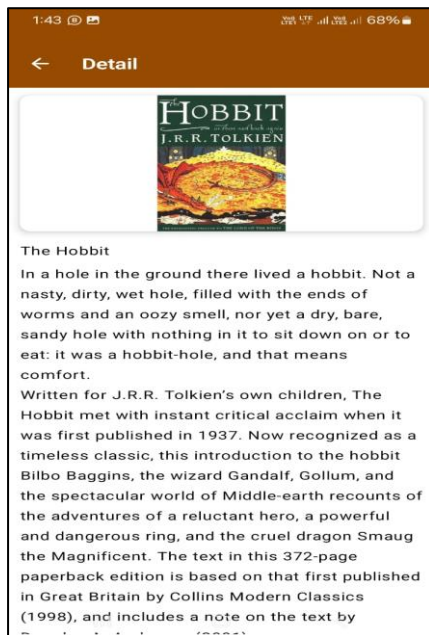


Figure 46 Detail page of application

6. Enquiry Page –

Figure 47 Enquiry Page of application

7. Contact Us Page –

Figure 48Contact Us Page of application

8. About Us Page –

Figure 49About Us Page of application

9. Log Out Page –

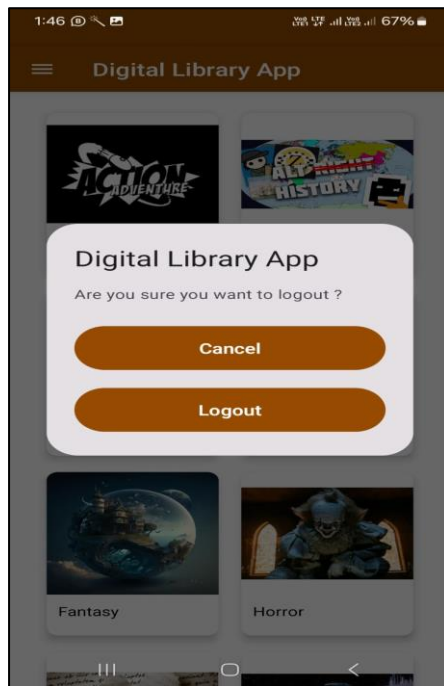


Figure 50 Log out page of application

GIT

1st commit

```
Ramesh@LAPTOP-1AR0D29K MINGW64 ~/Downloads/1st Commit/DigitalLibraryApp (master)
$ git remote add origin https://github.com/Bharathim999/DigitalLibraryApp.git

Ramesh@LAPTOP-1AR0D29K MINGW64 ~/Downloads/1st Commit/DigitalLibraryApp (master)
$ git push origin master
```

Figure 51 First Git commit part 1

```
Ramesh@LAPTOP-1AR0D29K MINGW64 ~/Downloads/1st Commit/DigitalLibraryApp (master)
$ git commit -m "First commit"
[master (root-commit) de986a] first commit
92 files changed, 3248 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/digitallibrary/ExampleInstrumentedTest.kt
create mode 100644 app/src/main/AndroidManifest.xml
create mode 100644 app/src/main/java/com/digitallibrary/MainActivity.kt
create mode 100644 app/src/main/java/com/digitallibrary/application/AppConfig.kt
create mode 100644 app/src/main/java/com/digitallibrary/routing/Navigation.kt
create mode 100644 app/src/main/java/com/digitallibrary/routing/Screen.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/about_us/AboutUsScreen.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/booksScreen/BooksScreen.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/contact_us/ContactUsScreen.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/detail/DetailScreen.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/drawer/NavDrawer.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/drawer/TopBar.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/enquiryScreen/EnquiryScreen.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/login/LoginScreen.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/main/MainScreen.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/model/BooksModel.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/model/CategoryModel.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/register/RegisterScreen.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/restaurantPreference/RestaurantPreference.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/splash/SplashScreen.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/theme/Color.kt
create mode 100644 app/src/main/java/com/digitallibrary/ui/theme/Theme.kt
create mode 100644 app/src/main/java/com/digitallibrary/Utils/Extension.kt
create mode 100644 app/src/main/java/com/digitallibrary/Utils/LibraryBorderField.kt
create mode 100644 app/src/main/java/com/digitallibrary/Utils/RoundedButton.kt
create mode 100644 app/src/main/res/drawable-v24/ic_launcher_foreground.xml
create mode 100644 app/src/main/res/drawable/ic_action_and_adventure.png
create mode 100644 app/src/main/res/drawable/ic_alt.jpg
create mode 100644 app/src/main/res/drawable/ic_bourne.jpg
create mode 100644 app/src/main/res/drawable/ic_boy.jpg
```

Figure 52 First Git commit part 2

```

Ramesh@LAPTOP-1AR0D29K MINGW64 ~/Downloads/1st Commit/DigitalLibraryApp (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/digitallibrary/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/digitallibrary/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/digitallibrary/ui/drawer/Drawer.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/digitallibrary/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/digitallibrary/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/digitallibrary/utils/RoundedButton.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/drawable-v24/ic_launcher_foreground.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/drawable/ic_launcher_background.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/mipmap-anydpi-v26/ic_launcher.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/mipmap-anydpi-v26/ic_launcher_round.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/mipmap-anydpi-v33/ic_launcher.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/values/colors.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/values/strings.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/values/themes.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/xml/backup_rules.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/xml/data_extraction_rules.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/test/java/com/digitallibrary/ExampleUnitTest.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build.gradle', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'gradle.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'gradlew', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'settings.gradle', LF will be replaced by CRLF the next time Git touches it
Ramesh@LAPTOP-1AR0D29K MINGW64 ~/Downloads/1st Commit/DigitalLibraryApp (master)

```

Figure 53 First Git commit part 3

```

Ramesh@LAPTOP-1AR0D29K MINGW64 ~/Downloads/1st Commit/DigitalLibraryApp
$ 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 <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in C:/Users/Ramesh/Downloads/1st Commit/DigitalLibraryApp/.git/
Ramesh@LAPTOP-1AR0D29K MINGW64 ~/Downloads/1st Commit/DigitalLibraryApp (master)

```

Figure 54 First Git commit part 4

```

Ramesh@LAPTOP-1AR0D29K MINGW64 ~/Downloads/1st Commit/DigitalLibraryApp
$ git config --global user.name "Bharathim999"

Ramesh@LAPTOP-1AR0D29K MINGW64 ~/Downloads/1st Commit/DigitalLibraryApp
$ git config --global user.email "bharathisweet999@gmail.com"

Ramesh@LAPTOP-1AR0D29K MINGW64 ~/Downloads/1st Commit/DigitalLibraryApp
$ git init

```

Figure 55 First Git commit part 5

The screenshot shows the GitHub web interface for the repository 'Bharathim999 / DigitalLibraryApp'. The repository is public and has 1 commit. The commit history table lists the following files and their commit times:

File	Commit	Time
app	first commit	5 minutes ago
gradle/wrapper	first commit	5 minutes ago
.gitignore	first commit	5 minutes ago
build.gradle	first commit	5 minutes ago
gradle.properties	first commit	5 minutes ago
gradlew	first commit	5 minutes ago
gradlew.bat	first commit	5 minutes ago

The right sidebar shows repository statistics: 0 stars, 1 watching, and 0 forks. There are no releases or packages published yet.

Figure 56 Digital library application Git

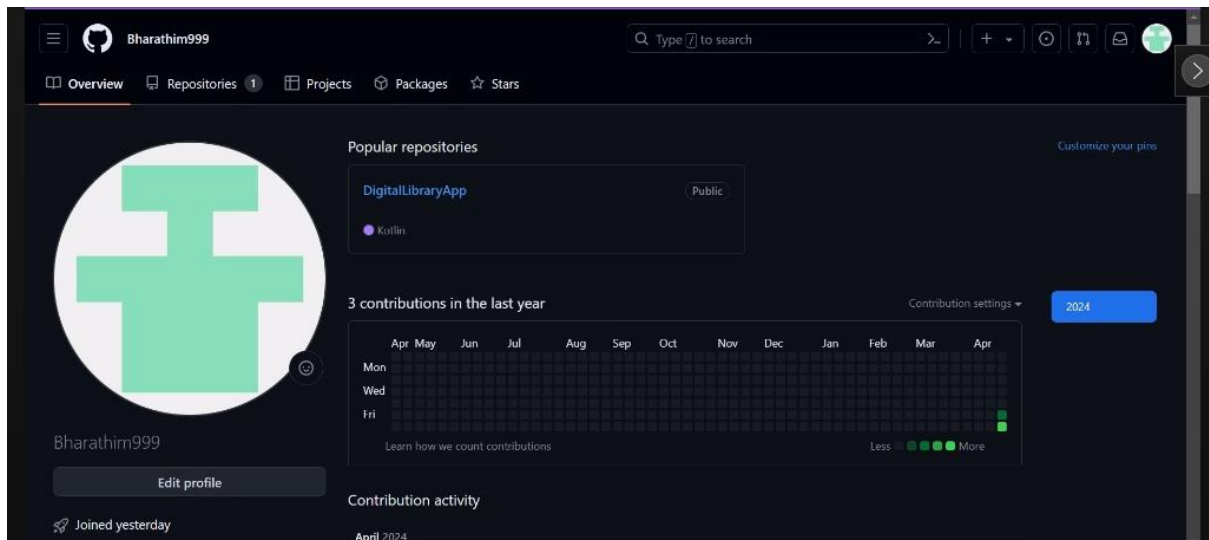


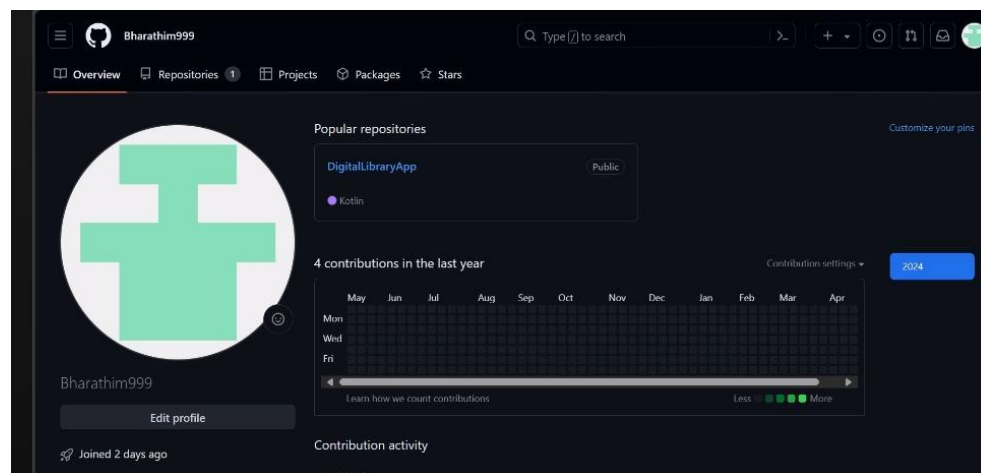
Figure 57 Digital library application Git overview

```
Ramesh@LAPTOP-1AR0D29K MINGW64 ~/Downloads/1st Commit/DigitalLibraryApp (master)
$ git push origin master
Enumerating objects: 138, done.
Counting objects: 100% (138/138), done.
Delta compression using up to 8 threads
Compressing objects: 100% (116/116), done.
Writing objects: 100% (138/138), 3.05 MiB | 911.00 KiB/s, done.
Total 138 (delta 9), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (9/9), done.
To https://github.com/Bharathim999/DigitalLibraryApp.git
 * [new branch]      master -> master

Ramesh@LAPTOP-1AR0D29K MINGW64 ~/Downloads/1st Commit/DigitalLibraryApp (master)
$ |
```

Figure 58 Digital library application Git overview 2

2nd commit



```
Ramesh@LAPTOP-IAROD29K MINGW64 ~/Downloads/third commit/clone/DigitalLibraryApp (master)
$ git push
Enumerating objects: 35, done.
Counting objects: 100% (35/35), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (18/18), 2.18 KiB | 1.09 MiB/s, done.
Total 18 (delta 10), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (10/10), completed with 10 local objects.
To https://github.com/Bharathim999/DigitalLibraryApp.git
dce986a..89a9615 master -> master

Ramesh@LAPTOP-IAROD29K MINGW64 ~/Downloads/third commit/clone/DigitalLibraryApp (master)
$
```

```
Ramesh@LAPTOP-IAROD29K MINGW64 ~/Downloads/third commit/clone/DigitalLibraryApp (master)
$ git commit -m "second commit"
[master 89a9615] second commit
6 files changed, 62 insertions(+), 155 deletions(-)

Ramesh@LAPTOP-IAROD29K MINGW64 ~/Downloads/third commit/clone/DigitalLibraryApp (master)
$ git push
Enumerating objects: 35, done.
Counting objects: 100% (35/35), done.
Delta compression using up to 8 threads
```

```
Ramesh@LAPTOP-IAROD29K MINGW64 ~/Downloads/third commit/clone/DigitalLibraryApp (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/digitallibrary/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/digitallibrary/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/digitallibrary/ui/drawer/NavDrawer.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/java/com/digitallibrary/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/digitallibrary/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/digitallibrary/utils/RoundedButton.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/drawable-v24/ic_launcher_foreground.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/drawable/ic_launcher_background.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/mipmap-anydpi-v26/ic_launcher.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/mipmap-anydpi-v26/ic_launcher_round.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/mipmap-anydpi-v33/ic_launcher.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/values/colors.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/values/strings.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/values/themes.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/xml/backup_rules.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/main/res/xml/data_extraction_rules.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'app/src/test/java/com/digitallibrary/ExampleUnitTest.kt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build.gradle', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'gradle.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'gradlew', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'settings.gradle', LF will be replaced by CRLF the next time Git touches it

Ramesh@LAPTOP-IAROD29K MINGW64 ~/Downloads/third commit/clone/DigitalLibraryApp (master)
$ git commit -m "second commit"
[master 89a9615] second commit
```

Bharathim999 / DigitalLibraryApp

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security 1

Insights

Settings

DigitalLibraryApp

Public

Pin

Unwatch 1

Fork 0

Star 0

master

1 Branch

0 Tags

Go to file

Code

About

Bharathim999

second commit

89a9615 · now

2 Commits

app	second commit	now
gradle/wrapper	first commit	yesterday
gitignore	first commit	yesterday
build.gradle	first commit	yesterday
gradle.properties	first commit	yesterday
gradlew	first commit	yesterday
gradlew.bat	first commit	yesterday

No description, website, or topics provided.

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

References

1. Uebe Mansur, A.F., Alves, A.C. and Torres, R.B., 2019. *Trello as virtual learning environment and active learning organiser for PBL classes: An analysis under Bloom's Taxonomy*.
2. Bergmann, T. and Karwowski, W., 2019. *Agile project management and project success: A literature review*. In *Advances in Human Factors, Business Management and Society: Proceedings of the AHFE 2018 International Conference on Human Factors, Business Management and Society, July 21-25, 2018, Loews Sapphire Falls Resort at Universal Studios, Orlando, Florida, USA 9* (pp. 405-414). Springer International Publishing.
3. Crowder, J.A. and Friess, S., 2015. *Agile project management: managing for success*.
4. Abd El-Migid, M.A., Cai, D., Niven, T., Vo, J., Madampe, K., Grundy, J. and Hoda, R., 2022. *Emotimonitor: A trello power-up to capture and monitor emotions of agile teams*. *Journal of Systems and Software*, 186, p.111206.
5. Hron, M. and Obwegeser, N., 2022. *Why and how is Scrum being adapted in practice: A systematic review*. *Journal of Systems and Software*, 183, p.111110.
6. 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.