MANİSA CELAL BAYAR ÜNİVERSİTESİ

FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMANT

FALL 2021

SOFTWARE ENGİNEERİNG

Customized Visual Bookshelf from Gutenberg Project

28.12.2021

170316036 – Kardelen ÇETİN

170316044 – Muzaffer ÖZEN

170316006 – Ayseli Erem BATI

170316053 – Tuğrul Can MERCAN

200315006 – Ömer BENEK

Manisa Celal Bayar University Computer Engineering Department

CSE4123 Software Engineering Fall 2021

Project Documentation

Customized Visual Bookshelf from Gutenberg Project

TABLE OF CONTENTS

INTRODUCTION

The name of our application is Libros. The main purpose of our application is to be able to search for and read books and collect selected books in a visual library for a specific user of the system. Each user should be able to see their own library when they log in and change the books in their own shelf by making new searches. The user can remove a book from the shelf when the book is finished. The user should be able to see the page in the book he/she has stopped reading the next time he/she logs into the system. We wanted to use the GitHub tool in our project, you can see our stages from the GitHub link of our project. GitHub tool has been the most suitable tool for us both in terms of manageability and in terms of our business life. We wanted to develop our project with the Scrum methodology. The advantages that Scrum provided us were as follows; We thought that having the project open and clear would both save time and ensure the successful outcome of the project. We thought that breaking the project into understandable and manageable parts would save time in quickly identifying and correcting potential problems. We thought that the whole team being aware of the whole flow of the project would increase the communication within the team. We expected that it would build trust between the customer and the developers, and thus the project would be successful.

Our Project GitHub Link: https://github.com/kardelencetin/SoftwareEngProject

The time to determine and run our Scrum Sprints is as follows:

27.10.2021 – 10.11.2021 First Sprint => Requirements Elication Phase(Introduction, Proposed System, Overview, User Requirements, System Requirements, System Stakeholders, Functional Requirements, Non-Functional Requirements, System Model includes Scenarios)

11.11.2021 – 25.11.2021 Second Sprint => System Model includes Use Cases and Use Case Diagram

26.11.2021 – 10.12.2021 Third Sprint => System Model includes Object Design, Dynamic Design, Architectural Design

11.12.2021 – 25.12.2021 Fourth Sprint => Implementation and Testing

PROPOSED SYSTEM

1- The feature that distinguishes the system we propose from other systems is that each user should see their own library when they log in and be able to change the books on their shelf by making new searches, or they can remove the book from the shelf when the book is finished.
2- In addition, another feature that distinguishes it from other systems is that the user can see the page on which he paused reading. In addition to the visualized library, the user is also shown a pop-up screen showing which book is paused on which page.
3- This application, which simulates the virtual library, can make many people gain the habit of reading books.
4- In addition, we take the burden of books off our shoulders, and we can easily access thousands of books or our virtual library that we have created by phone or via the web. In this way, we can easily access our library and read books at every moment of our lives.

OVERVIEW

The purpose of this system is to provide features such as reading and viewing a book. The user can see the page on which you are left in the book, create a library and remove a book from the shelf when the book is finished.

USER REQIUREMENTS

1- Each user must have their own account and be able to log in to the system with this account.
2- The application shall provide the ability to search and read books for the user.
3- The application shall show the user where the user has left in their boks.
4- The application shall the user to add the books he likes to his/her library.

SYSTEM REQUIREMENTS

1. Each user must have their own account and be able to log in to the system with this account.
   - 1.1- The information of the user's account should be able to be stored in the system.
   - 1.2- The system should be able to remember the user in all logins after the first login to the system.
   - 1.3- Each user must have an individual account in the system.
2. The application shall provide the ability to search and read books for the user.
   - 2.1- Books should be able to be searched by filtering in the system.
   - 2.2- The user should be able to access the books registered in the system.
   - 2.3- A list of the books that the user has read in the system should be able to be kept.
3. The application shall show the user where the user has left in their boks.
   - 3.1- The system should be able to show the user's last stay in the books.
   - 3.2- The system must refresh the updated information each time the user logs in.
   - 3.3- When the user logs out or closes the application, the system should still be able to show the page the user was left on.
4. The application shall the user to add the books he likes to his library.
   - 4.1- The system should be able to add the books that the user likes to her/his library.
   - 4.2- The system should be able to delete the books that the user does not want to be in her library.
   - 4.3- The system should be able to update the user's library and store the updated data.

SYSTEM STAKEHOLDERS

- Users whose have an account in the system.
- Software Developers who are responsible for installing and updating the system.
- The company that provides the books data to the application.

FUNCTIONAL REQUIREMENTS

- The user must be able to login to the system.
- The user must be able to register in the system.
- If the user forgets her/his password, she/he should be able to change her password.
- The user should be able to create a library on the system.
- The user should be able to view the books in the library.
- User should be able to delete books in library.
- The user should be able to see where she/he left off in her books.
- User should be able to search for books.
- The user should be able to add books to the library.
- The user should be able to search for another book while reading a book.

NON-FUNCTIONAL REQUIREMENTS

->Performance:

-Users can access books all day long.

->Usability:

-Users should have access to all 60,000 books available in the app.

-Users can access the books in the system with the name of the author and the name of the book.

-When the user continues to read a book, he will be able to see which page he has left.

->Functionalty:

-The system should provide more than one functions and services to the user.

->Reliability:

-Only the user who created the account can access the account created by the user.

->Supportability:

-The application can support up to 10000 users.

-Users can only read the books in the application online.

->Rapid Develeopment:

-The system should be developed quickly.

->Low-Coast:

-While designing the system, the coast should be kept as low as possible.

->Fault Tolerance:

-Fault tolerance should be as low as possible.

SYSTEM MODEL

SCENARIOS

Scenario Name ömerMeetsLibrasApplication

Participating Actor Instances Ömer, Muzo : Reader

Flow of Events

1. Omer loves reading books, but his family doesn't like to buy books from the library because he is in the pandemic period.
2. That's why he's doing research on how to read a book. As a result of his research, he sees the Libras application and installs it with joy.
3. In order to log in to the application, it is necessary to register and register by email. After confirming the message sent to his mail, he can log in to the system. Now he had access to the books.
4. He immediately searches if the Sapiens book he has wanted to read for a long time is there, and he immediately finds it.
5. Later, he learned that he could add this book to his library by liking it, and immediately added a few more books.
6. He started reading Sapiens and really liked it. But suddenly his phone rang and his friend Muzo invited him to work on his graduation project.
7. When Omer goes to Muzo, he talks about the application and downloads it because Muzo loves to read books.
8. Omer, who left the book halfway, realized that he had forgotten the page he left off. He opened the application in a hurry, but the application was showing where he left off and Omer was very happy.
9. Now Omer reads his books hygienically and does not forget the place where he stayed.

Scenario Name: rachelSearchesForBook

Participating Actor Instances Rachel: Reader

Flow of Events

1. Rachel is majoring in English Language and Literature and needs to be able to purchase a book to research her teacher's assignment.
2. She logs into the Libros app she's using and sees if she can find the book she's looking for there.
3. This book is War and Peace by Fyodor Dostoyevsky. Since Rachel can't remember the name of the book, she wants to search for the author's name.
4. When Rachel writes the author's name, she thinks of which book to read when she sees the books that come before her.
5. She adds this book to the library she created for her account in the application so that she can access this book frequently.
6. She then finishes reading this book and removes it from her library.

USE CASE MODEL

Use Case Name: SearchBooks

Participating Actors: Initiated by Reader

Flow of Events:

1. The reader typed the author of the book she/he needed in the search bar.

    2. The system brought the author's books in front of the user.

Entry Condition: The reader must want to search for a book.

Exit Condition: The reader should be able to find the book.

Quality Requirements: The reader can search for 60,000 books.

Exceptions:

1. The reader typed the author of the book she/he needed in the search bar.[bookNotFound]

Use Case Name: RemoveBooks

Participating Actors: Initiated by Reader

Flow of Events:

1. The reader deletes the book she/he no longer needs from her/his library.

    2. The system removes the book from the library.

3. The reader saves the library and closes the application.

    4. The system saves the history information of the reader.

Entry Condition: The reader must want to delete a book.

Exit Condition: The reader saves and closes the application.

Quality Requirements: When the reader enters her library, she should not see the deleted book.

Use Case Name: AddBookToTheShell

Participating Actors: User

Flow of Events:

1.The user types the name of the book she wants to search or the name of the author of the book in the search bar.

    2.The system presents the book with that name or the books belonging to that author to the user in line with the information entered by the user.

3.As a result of the information entered, the user likes the book that she will choose from the books that come before her by pressing the like button.

    4.The system adds the book that the user likes to the library in the user's account.

Entry Condition: User must search for a book or author name.

Exit Condition: The book that the user likes should be added to the user's library.

Quality Requirements: When the user searches for an author's name, she should be able to see all the books belonging to that author.

Use Case Name: RegisterToLibros

Participating Actors: User

Flow of Events:

1. After the user downloads the application, she enters the name , password and e-mail information to be registered.

2. The system sends a confirmation message to the e-mail address entered by the user.

3- The user approves the e-mail sent to the e-mail address.

4- When the system e-mail address is approved, it saves the user's account to the application in line with the information entered by the user.

Entry Condition: To register, the user must enter her e-mail address and other necessary information and confirm her e-mail address.

Exit Condition: The user must register in the system with the username and e-mail address she has written.

Quality Requirements: If the user does not receive the confirmation mail while trying to register, she should be able to request the confirmation mail again.

Use Case Name: LoginToLibros

Participating Actors: User

Flow of Events:

1. The user enters the user name and password used when registering to the system to log in.

2. The system directs the user to the account according to the information entered by the user.

Entry Condition: To login, the user must enter the correct username and password.

Exit Condition: The user should be able to see their own account and their own library.

Quality Requirements: If the user has forgotten her password while logging in, she should be able to receive a password renewal e-mail.

Use Case Name: Continue reading the book where you left off

Participating Actors: Initiated by Reader

Flow of Events:

1. The reader enters the user name and password.

2. The system directs the reader to the account according to the information entered by the reader.

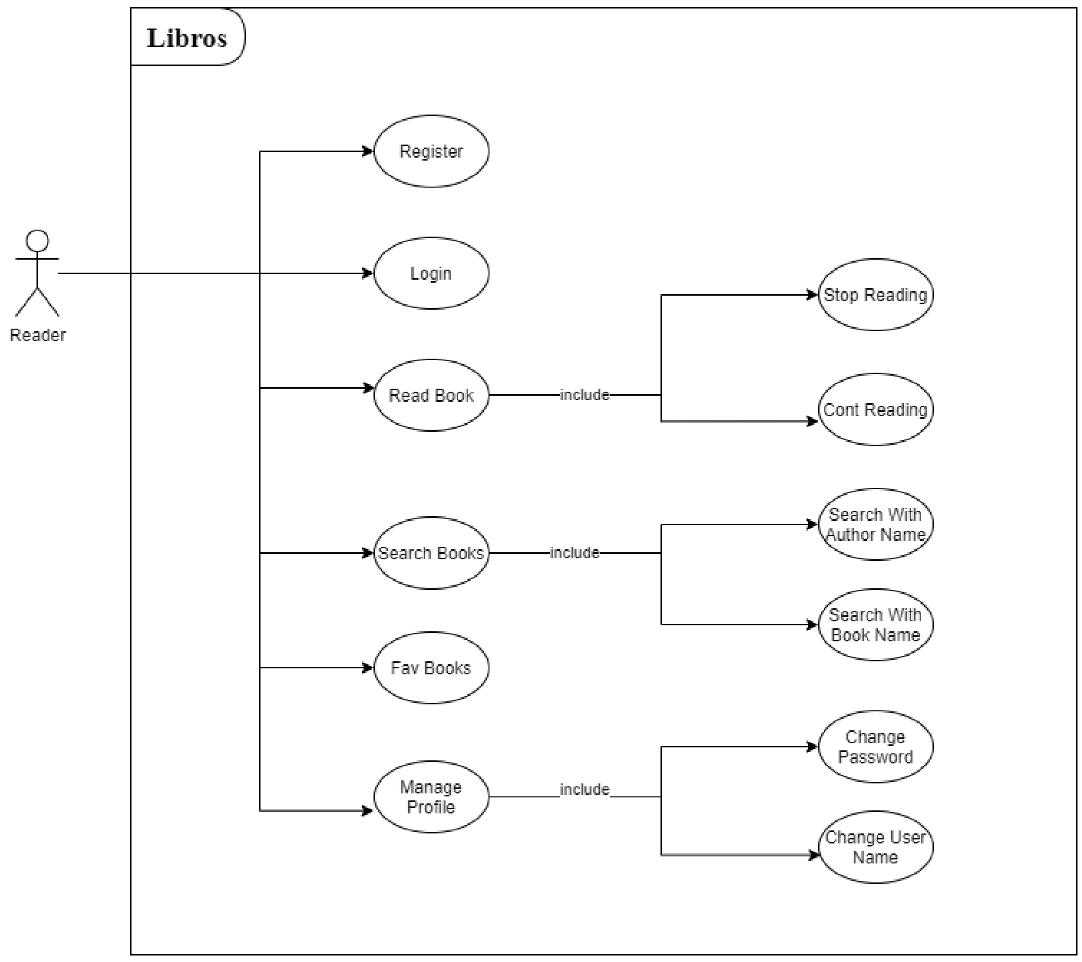
3. The reader selects a book in the library in her account, which she/he has already started reading, to continue reading.

4. The system directs the reader to the page of the book that the reader left while reading before.

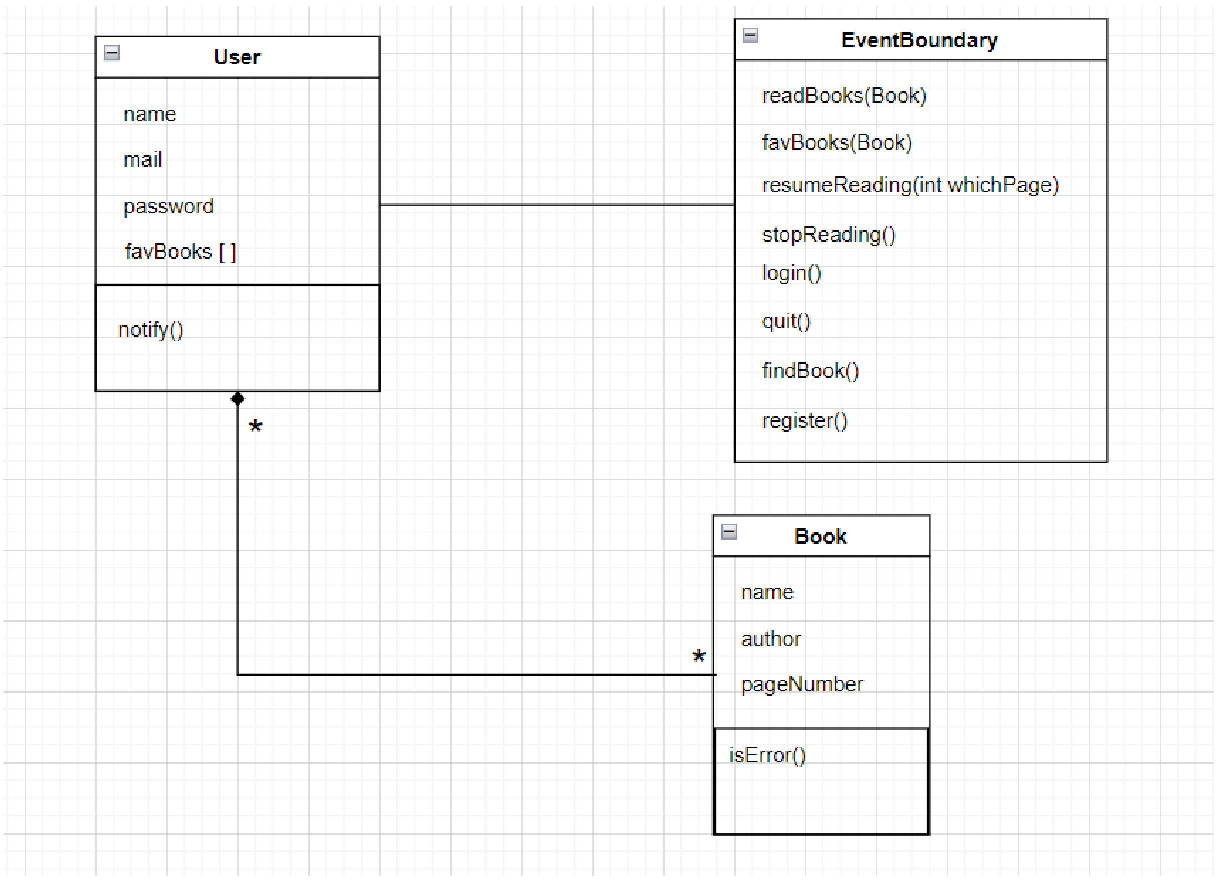
Entry Condition: The reader selects the book they want to continue reading from their library.

Exit Condition: The reader can continue reading the book she/he chooses from where she/he left off.

USE CASE DIAGRAM

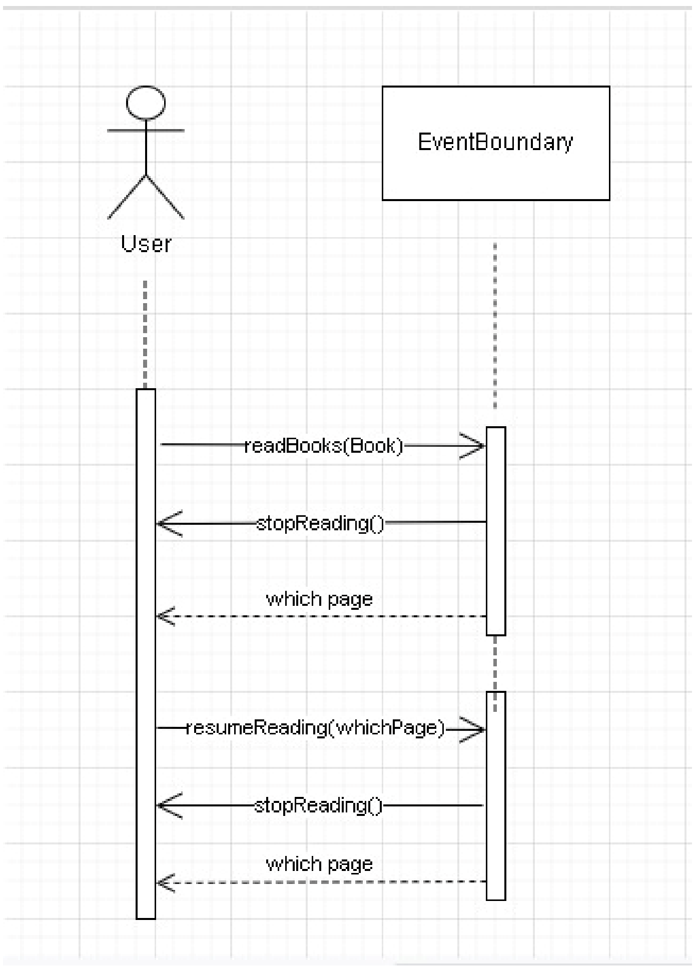


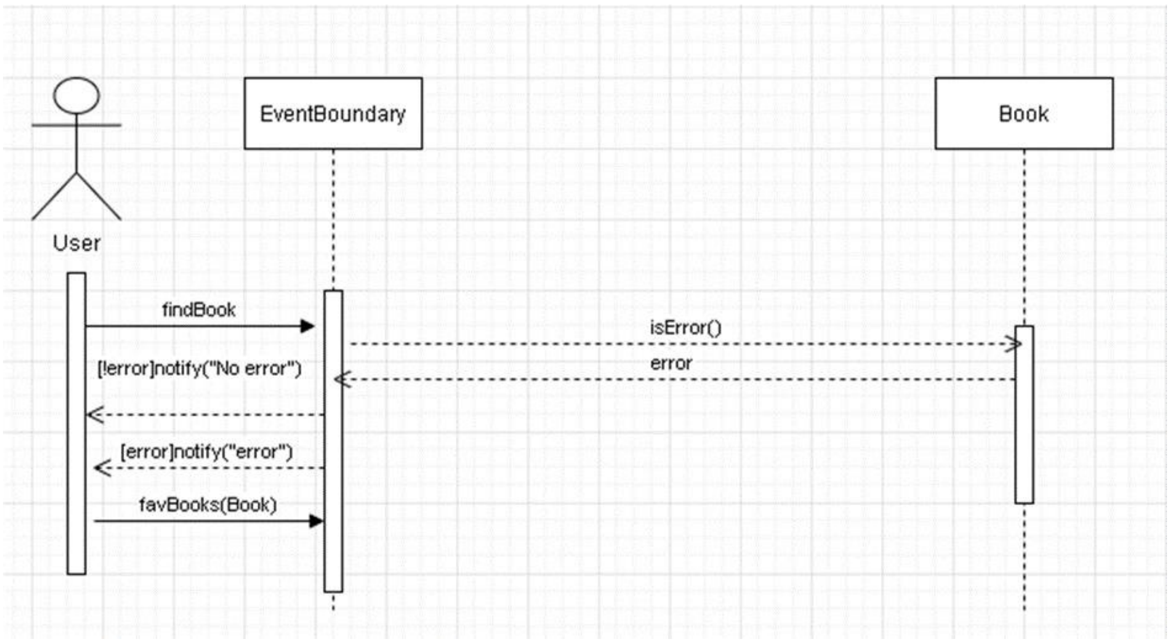
OBJECT MODEL



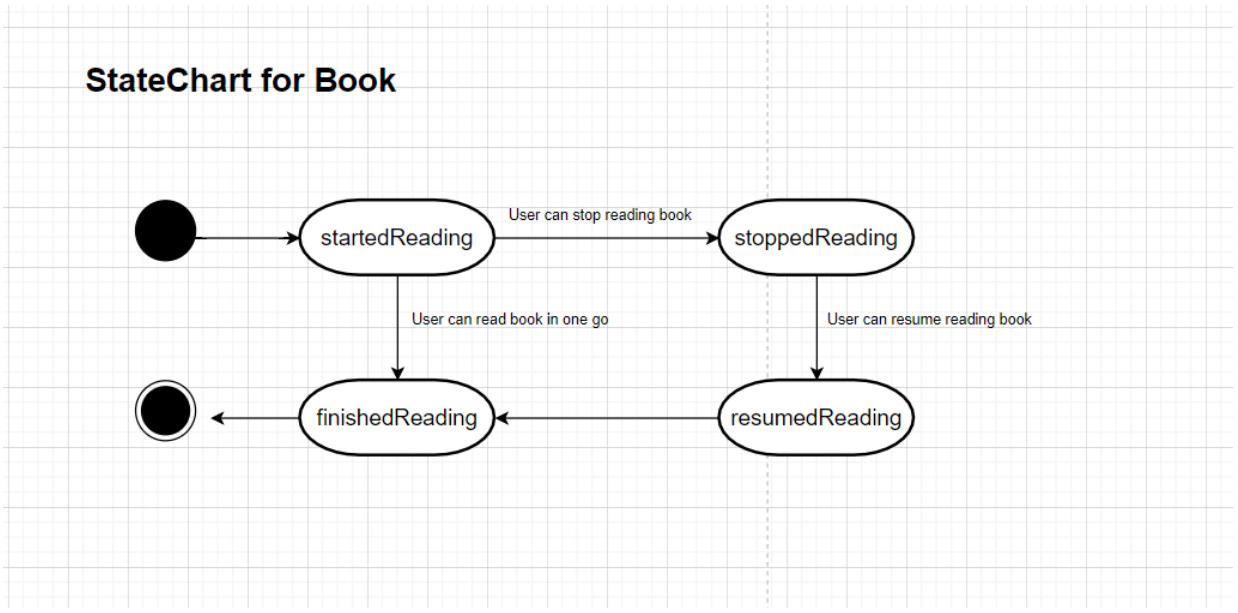
DYNAMIC MODEL

SEQUENCE DIAGRAM

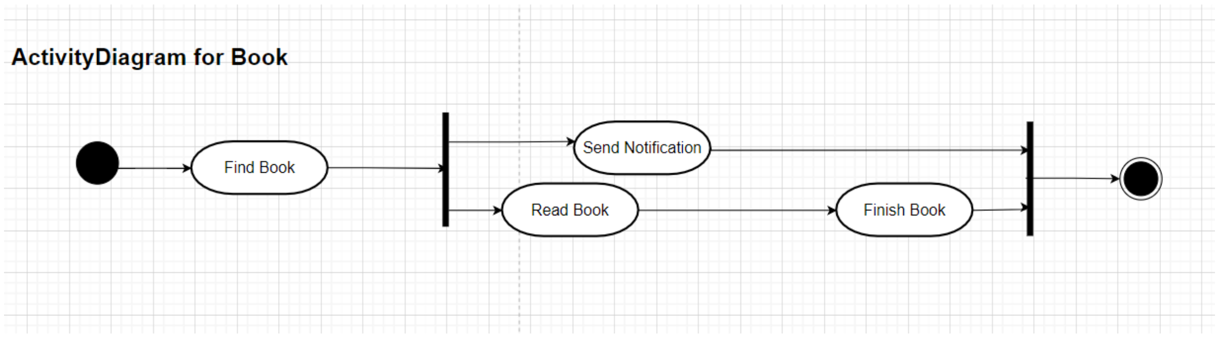




STATECHART DIAGRAM

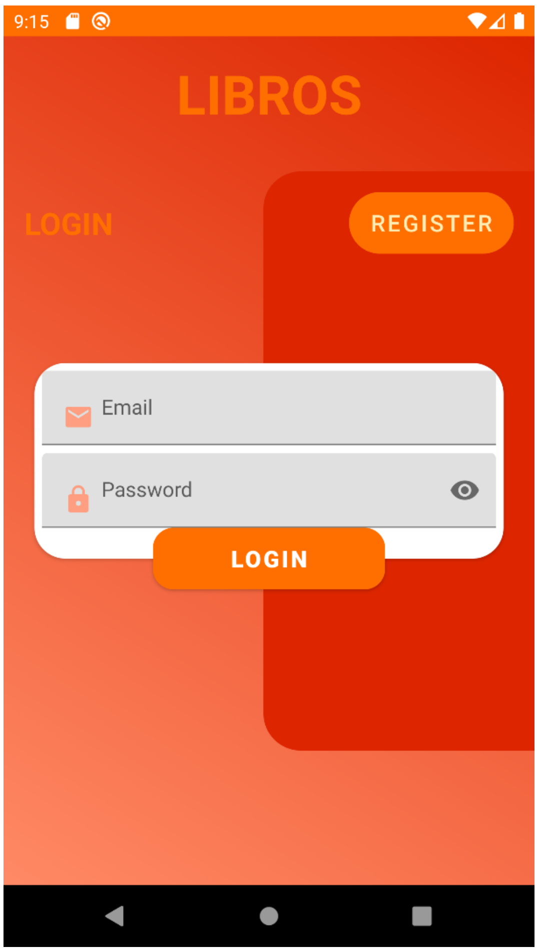

StateChart for Book
startedReading
User can stop reading book
stoppedReading
User can read book in one go
User can resume reading book
finishedReading
resumedReading

ACTIVITY DIAGRAM

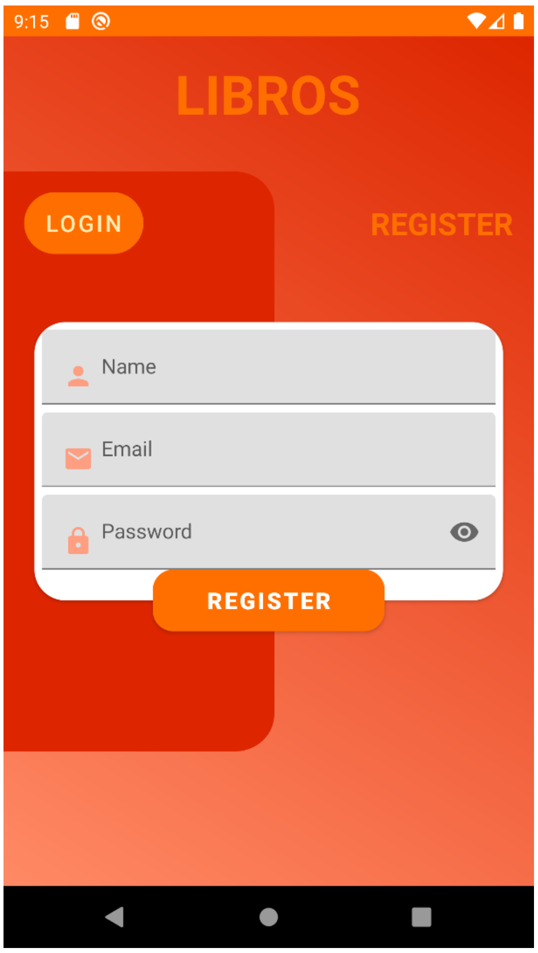


USER INTERFACE MOCK-UPS

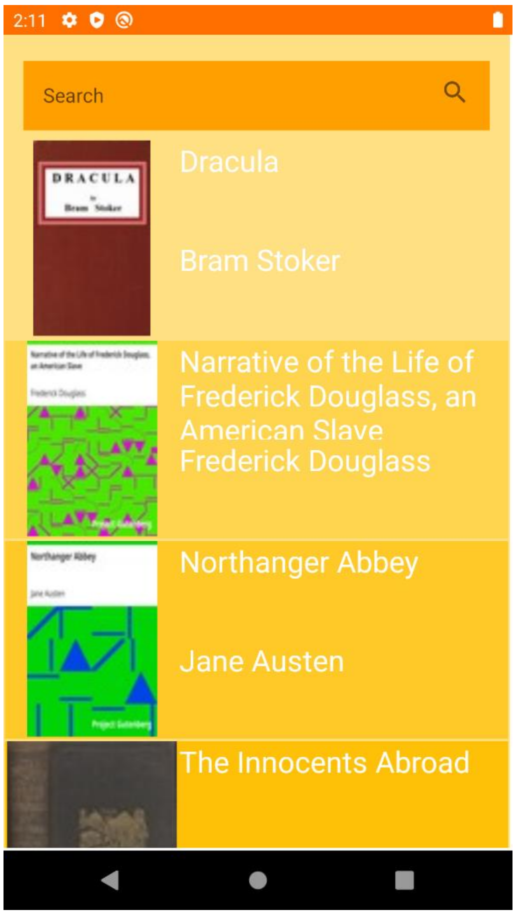
Android UI



Login Fragment



Register Fragment



Dracula

Bram Stoker

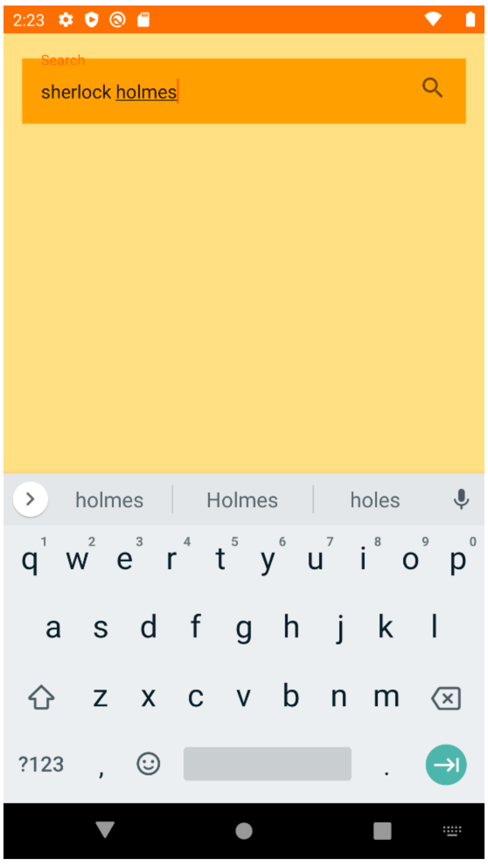
Narrative of the Life of Frederick Douglass, an American Slave
Frederick Douglass

Northanger Abbey

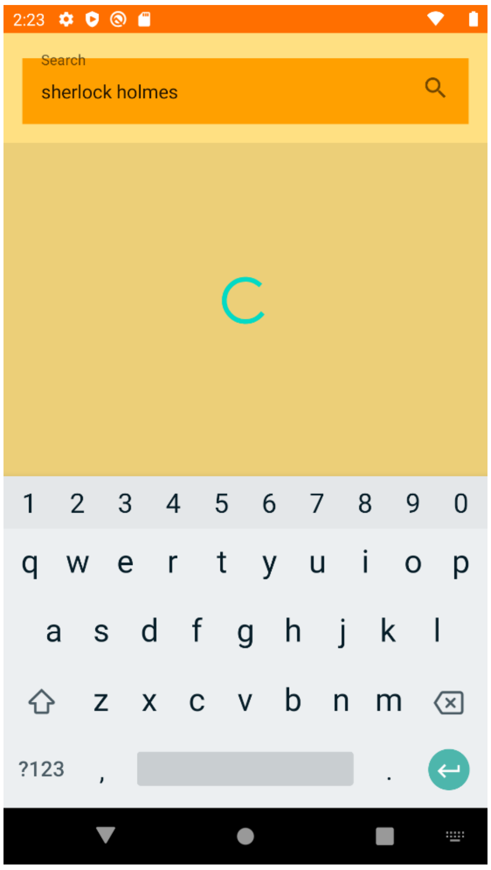
Jane Austen

The Innocents Abroad

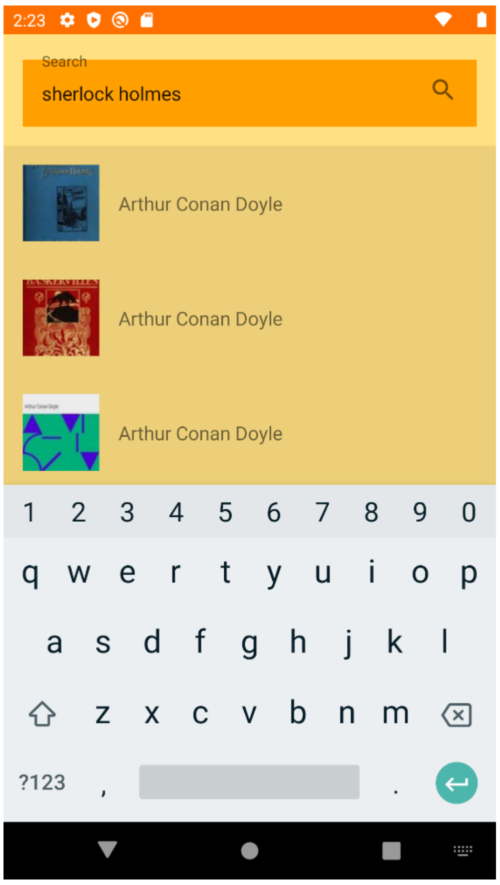
Incomplete BookRecyclerViewFragment



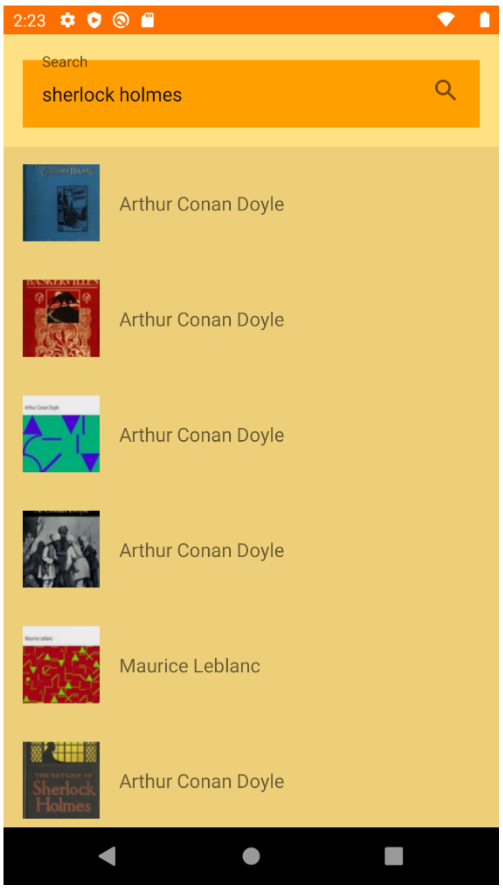
Search Sherlock Holmes Books



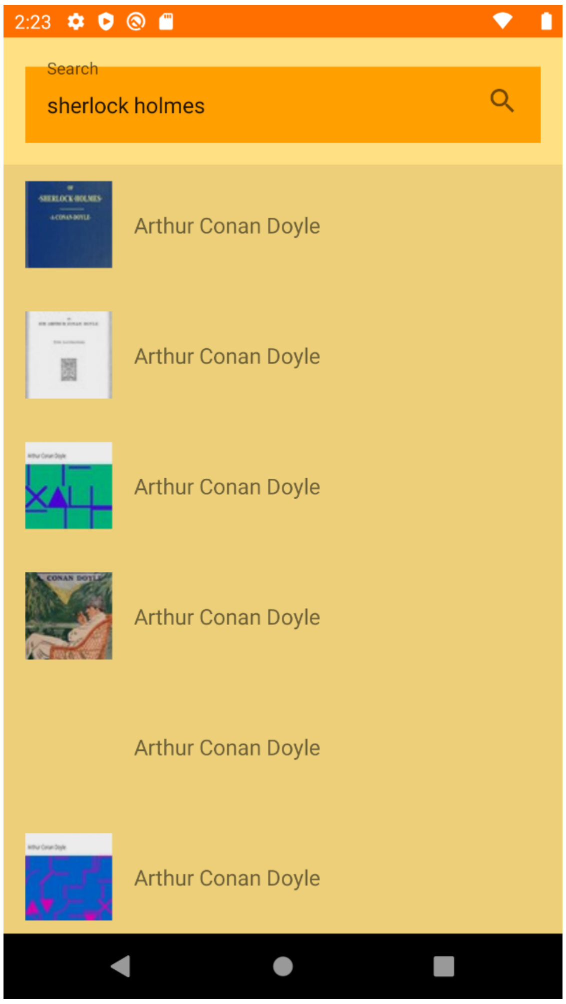
Loading Sherlock Holmes Books



Filtering Sherlock Holmes Books v1



Filtering Sherlock Holmes Books v2


2:23

Search

sherlock holmes

Arthur Conan Doyle

Arthur Conan Doyle

Arthur Conan Doyle

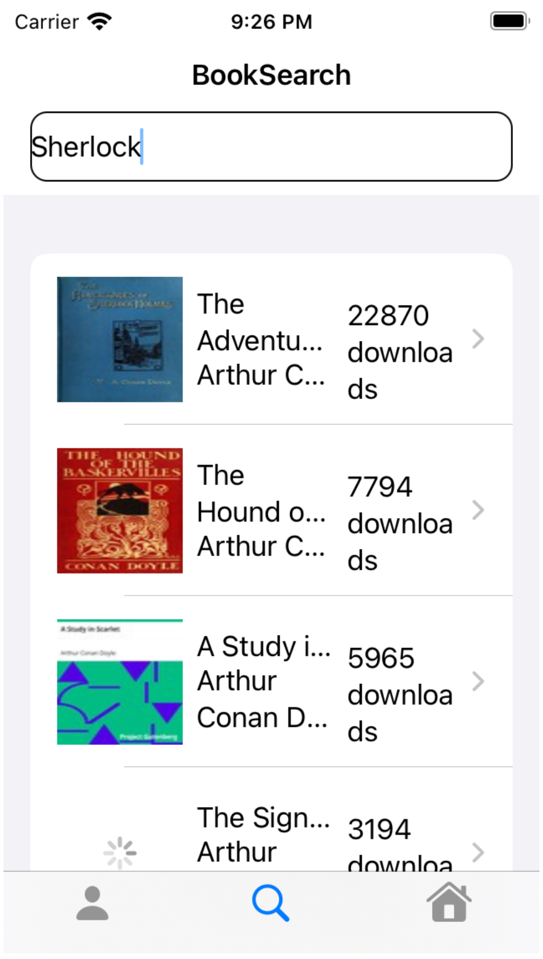
Arthur Conan Doyle

Arthur Conan Doyle

Arthur Conan Doyle

Filtering Sherlock Holmes Books v3

IOS UI


Carrier 9:26 PM

BookSearch

Sherlock

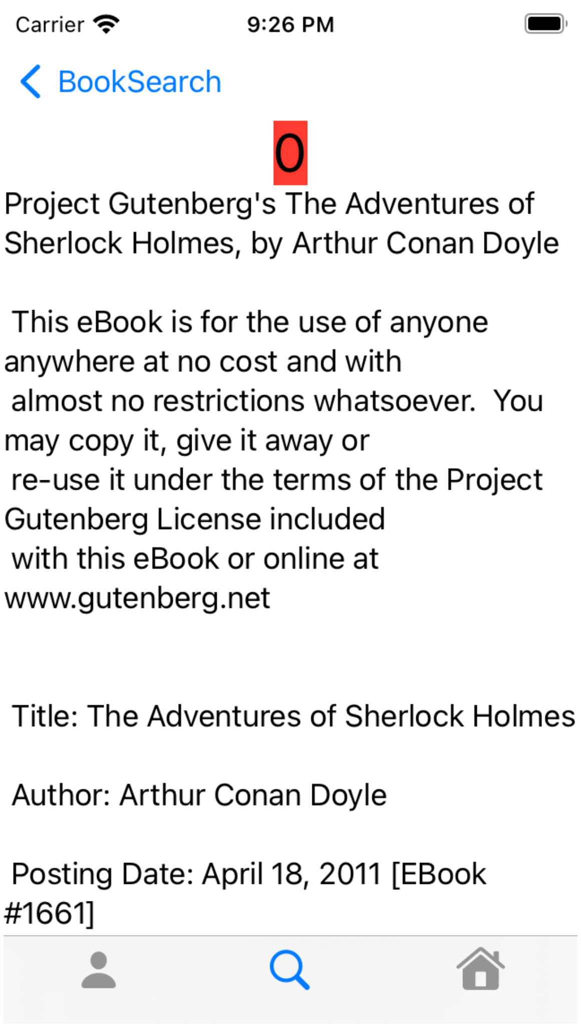
The Adventu... Arthur C... 22870 downloads

The Hound o... Arthur C... 7794 downloads

A Study i... Arthur Conan D... 5965 downloads

The Sign... Arthur 3194 downloa

Search Books



Project Gutenberg's The Adventures of Sherlock Holmes, by Arthur Conan Doyle

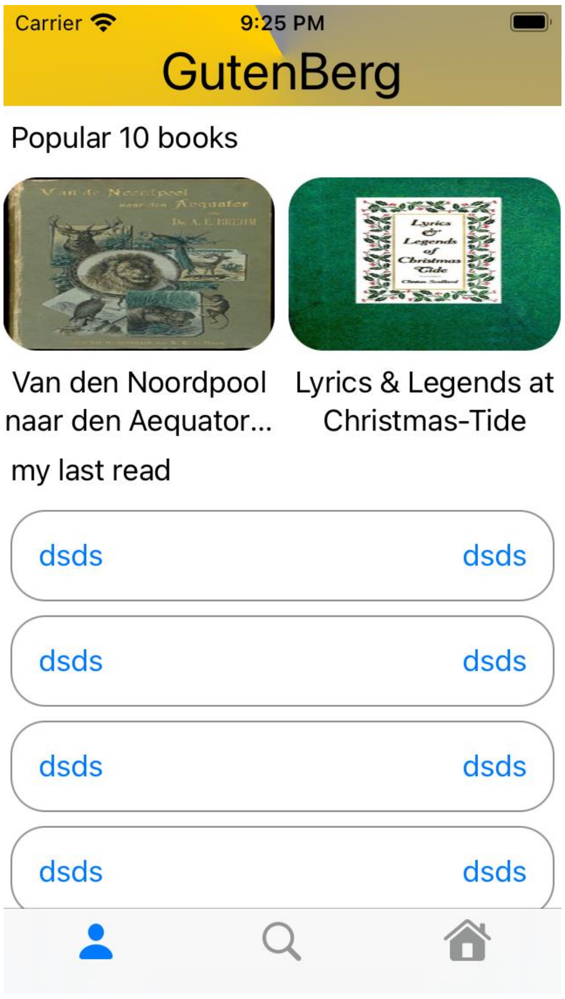
 This eBook is for the use of anyone anywhere at no cost and with
 almost no restrictions whatsoever. You may copy it, give it away or
 re-use it under the terms of the Project Gutenberg License included
 with this eBook or online at www.gutenberg.net

 Title: The Adventures of Sherlock Holmes

 Author: Arthur Conan Doyle

 Posting Date: April 18, 2011 [EBook #1661]

The page where we can read my books



Popular 10 books

Van den Noordpool naar den Aequator...

Lyrics & Legends at Christmas-Tide

my last read

dsds dsds

dsds dsds

dsds dsds

dsds dsds

Popular Books Lists

SYSTEM ARCHITECTURE

First of all, we had some difficulties in understanding the structural design of our project. We have tried to overcome these problems by conducting detailed research. As a result, we took care that the client server architecture, as well as the architecture that we will use in the content of the mobile application, is MVVM, that is, model-view-viewmodel, at the stage where we will write a service to read the books from the Gutenberg site in our application, and we have benefited from these architectural patterns in the implementation and testing stages.

What MVVM provides us, we want to talk a little bit about it. MVVM offers a design pattern based on the development of a project on a "decomposition of responsibilities" basis. Separation of responsibilities, namely Separation of Concerns(SoC), as we use it in our professional life. MVVM recommends that we develop a software in three different structures named Model, View and ViewModel. MVVM consists of three basic structures: Model, View, and ViewModel. Let's talk briefly about what these structures are:

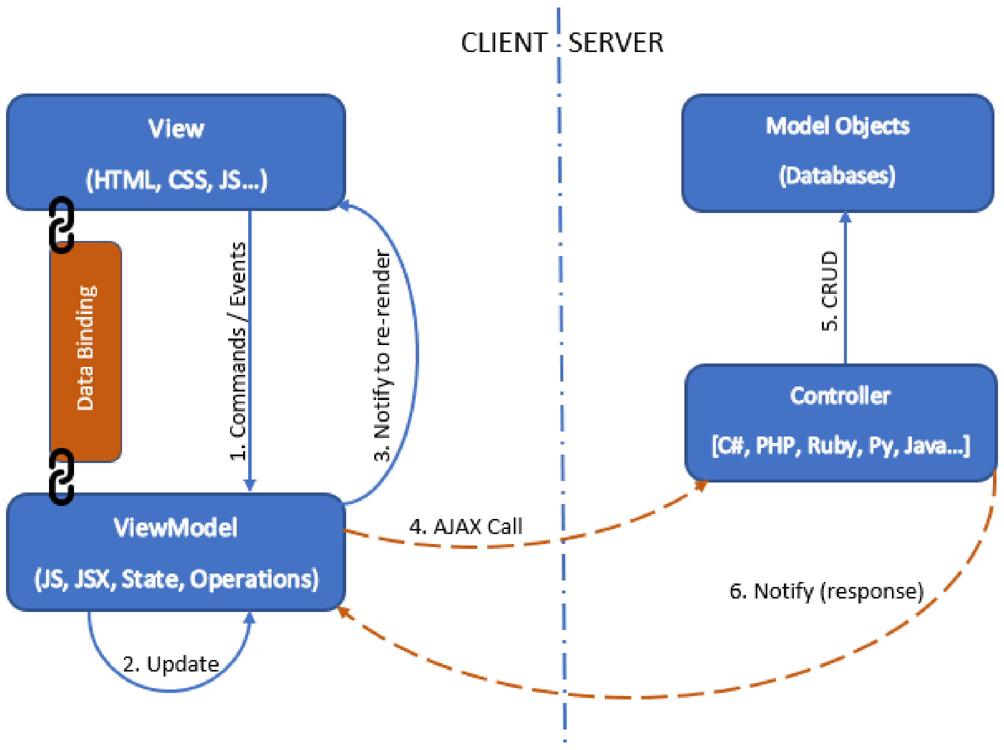
Model: It consists of POCO or entity classes used to represent our data coming from database, web services or any data source. In addition, business rules that control data consistency and accuracy are also included here.

View: This is the visual interface where we transfer our data to end users. It acts as a bridge between the end user and the application.

ViewModel: On the other hand, ViewModel acts as a bridge between the visual interface and the model, that is, it is the structure that connects the Model to the View. There is no direct interaction between View and Model. View performs related operations via ViewModel. The ViewModel does not have direct access to the View and does not know anything about the View.

By separating the structures in our application from each other, it enables us to construct a less connected structure, and separates components such as visual interface and business structures in the background, data access structures. In this way, software developers and interface designers have the opportunity to work independently on the same project simultaneously. The separation of the visual interface and the code from each other increases the testability of the application and provides convenience in terms of TDD (Test Driven Development). At the same time, these separations make our application easier to maintain. When it is desired to make a change on the model side, improvements can be made easily without the need to make any changes on the UI side. In fact, most of the features that we describe as the advantage of MVVM are what the Separation of Concerns approach has given us.

According to our research on the client-server model, we can mention the following; It is a network architecture that separates the client from the server. All resources in the network depend on the server and the server is the only responsible for network resources. If the server wants, it can give this responsibility to other machines in the network or use them all by itself. Other machines in the network can only connect to the network through the server and can benefit from network resources within the access permissions given by the server. What we call a client is a structure that requests from the server and can use the data on the server. For example, we can give a web page. The web page in the client position makes a request from the server and as a result, the server delivers the requested data to the client. Servers are computers that hold this information. They need to be equipped and have high-performance work. If we make the definition in a nutshell, Server is computers that share information on a network to users (computers), run many software on it, and have high performance. Information is kept on the server in banks and similar large companies. For companies, this takes an important place in terms of both time and cost.



Architectural Design

IMPLEMENTATION & TESTING

->IMPLEMENTATION

While developing our project, we preferred to develop android applications. While developing our application, we preferred the widely used Android Studio platform for android. If we list the reasons why we prefer Android Studio; Google Android Studio is based on IntelliJ IDEA, which offers powerful code editor and developer tools. It supports all Android platforms. It has a smart code editor that provides convenience to users. Its instant-run feature enables rapid reflection of your application running on the emulator, speeding up editing, compiling, and running processes. It offers comprehensive testing tools and frameworks. In addition to developing applications on Android, we wanted to deal with the IOS side and work on cross platforms. We used the XCODE platform while working on the IOS platform. XCODE is an excellent tool for professionals and beginners alike. With Xcode you can develop software for the amazing iOS, iPadOS, watchOS, macOS and tvOS operating systems. With support for Swift packages, Xcode lets you share the code across all your apps or use community-created packages. Xcode is the most basic development tool for someone who develops with Swift.

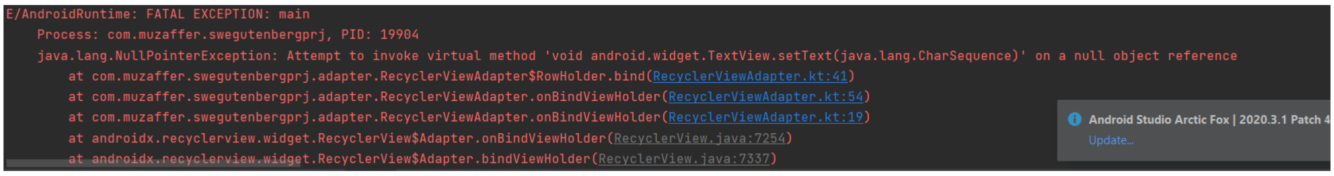
As the development language, we carried out our work in Kotlin. The purpose of using Kotlin is Much Less Lines of Code: We can say that the lines of code used in Kotlin language are much less when compared to Java, thus providing a significant advantage in the coding part. Compiled to Bytecode: Its overall performance is quite high. Plain and Understandable: The language used in coding has a much simpler and more understandable structure. Commands are clearer. Easy to Learn: It is a slightly easier language to learn because its scope is clear and the order of its commands is clear. It is Faster: Shorter codes are shown as a factor that increases compilation speed. A higher compilation speed also results in higher overall application performance and faster applications delivery. Using this platform for the transitions between the front of our application and each other has provided us with the conveniences we mentioned above. For ease of use, we used fragments instead of working with activities. You can think of fragments as post-its pasted over activities. We used navigation host to link the fragments together. Navigation host allows us to switch between fragments in the application. While we preferred Kotlini as the development language for the Android platform, we carried out our work in Swift for IOS. Swift is really fast, very powerful and very understandable programming language. That's why learning Swift is really easy. Swift is such an understandable language that even a non-coding person can understand a Swift code when they see it. We can understand how fast Swift is by comparing Swift with other programming languages: According to Apple, Swift is 2.6 times faster than Objective-C and 8.4 times faster than Python.

We used typescript with node.js while writing my service. If we talk about the advantages of Node.js; Being Fast: Node.js library built on Google Chrome's V8 JavaScript Engine is very fast at executing code. Asynchronous and Computational: All APIs of the Node.js library are asynchronous, that is, non-blocking. This means that a Node.js-based server never expects an API to return data. License: Node.js is open source and released under the MIT license. Buffering Process: Node.js apps do not buffer any data.

These applications extract data in chunks. Advantages of TypeScript; JavaScript is TypeScript: the codes written in TS are converted to their JS language counterparts when compiled, output JS code and the JS code is executed. All features that apply to JS apply to TS. Knowing JS in order to write TS code will enable you to dominate a large proportion of the work. TypeScript is an extended version of JavaScript. Every JS code is a TS code. But TS code is not JS code unless it is compiled and run. Can use all JS libraries: All JS libraries can also be used on TS. JS output of all code written as TS can use all JS frameworks, tools and libraries. Portability: TypeScript is a platform-free language and can run on different browsers, devices, operating systems. It can run in any environment where JavaScript runs. Unlike its counterparts (CoffeScript, Dart, etc.), TypeScript does not need a special virtual machine or a special run-execution environment for its execution, as the written codes are converted to JS code and the operations are executed over JS code. Creating a service provided the following convenience, we were able to use json data as we wanted, and we had no problem accessing the service.

Errors we encounter and their types

->> NullPointerException (RunTimeError)



```
E/AndroidRuntime: FATAL EXCEPTION: main
    Process: com.muzaffer.swegutenbergprj, PID: 19904
    java.lang.NullPointerException: Attempt to invoke virtual method 'void android.widget.TextView.setText(java.lang.CharSequence)' on a null object reference
        at com.muzaffer.swegutenbergprj.adapter.RecyclerViewAdapter$RowHolder.bind(RecyclerViewAdapter.kt:41)
        at com.muzaffer.swegutenbergprj.adapter.RecyclerViewAdapter.onBindViewHolder(RecyclerViewAdapter.kt:54)
        at com.muzaffer.swegutenbergprj.adapter.RecyclerViewAdapter.onBindViewHolder(RecyclerViewAdapter.kt:19)
        at androidx.recyclerview.widget.RecyclerView$Adapter.onBindViewHolder(RecyclerView.java:7254)
        at androidx.recyclerview.widget.RecyclerView$Adapter.bindViewHolder(RecyclerView.java:7337)
```

->> Cleartext HTTP traffic not permitted (IOExceptionError)

->> 'ts-node' is not recognized as an internal or external command, operable program or batch file (RunTimeError)

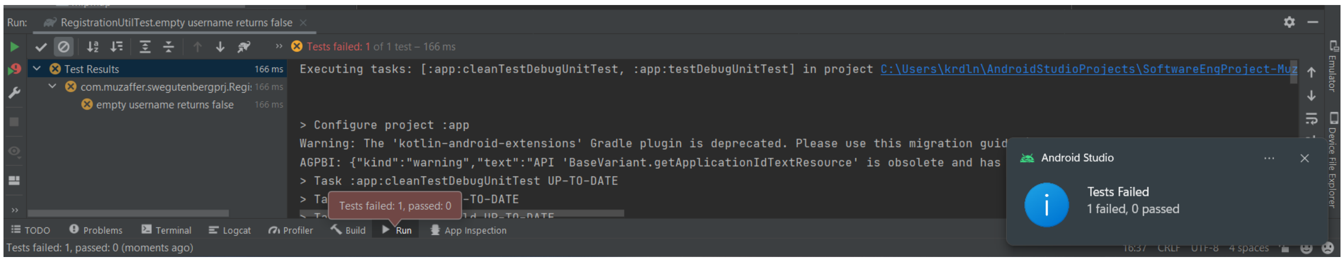
->> AlertDialog.Builder with custom layout and EditText; cannot access view (LogicError)

->> E/SpannableStringBuilder: SPAN_EXCLUSIVE_EXCLUSIVE spans cannot have a zero length.

SPAN_EXCLUSIVE_EXCLUSIVE spans cannot have a zero length (RunTimeError)

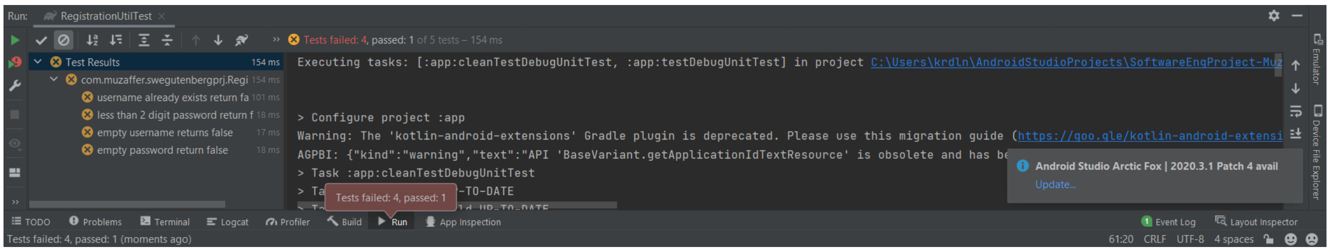
->TESTING

While developing our project, we once again understood the importance of test driven. The information we have obtained about Test Driven Development; TDD (Test-Driven Development) is a technique that can be translated into Turkish in the form of test-driven development, which argues that after determining which codes will be written for what purpose in the project, first the tests and then the code should be written. Although at first confrontation, writing tests before writing code can be difficult and complex. However, before starting to write the code of a project, its content must be determined and the code design must already be done. This is also necessary so that there is no difficulty during coding and the next step is predictable. It is normal for the TDD method to be difficult and complex as this step is usually not detailed. Writing a test of a designed coding is both healthier and helps to minimize errors. From this point of view, the purpose of TDD can be thought of as a directive that aims to prevent codes from falling into repetition, to be written in a simpler, non-vulnerable and understandable way. Since we did not have much time while writing the test and it was our last sprint, we were able to write a test for registration at this stage. Some of the tests we performed in Registration are as follows; If empty data is sent first, that is, if mail, name and password are not written, it returns an error. Returns an error if digits less than two characters are used in the password. Returns an error when sending a blank password. If username has been created before, it returns an error. If valid values are entered, it does not return an error.



```
Run:  RegistrationUtilTest.empty username returns false

Tests failed: 1 of 1 test – 166 ms

Test Results                                  166 ms
  com.muzaffer.swegutenbergprj.Regi:          166 ms
    empty username returns false              166 ms

Executing tasks: [:app:cleanTestDebugUnitTest, :app:testDebugUnitTest] in project C:\Users\krdln\AndroidStudioProjects\SoftwareEngProject-Muz


> Configure project :app
Warning: The 'kotlin-android-extensions' Gradle plugin is deprecated. Please use this migration guid
AGPBI: {"kind":"warning","text":"API 'BaseVariant.getApplicationIdTextResource' is obsolete and has
> Task :app:cleanTestDebugUnitTest UP-TO-DATE
> Ta                                -TO-DATE

Tests failed: 1, passed: 0

Tests failed: 1, passed: 0 (moments ago)

Android Studio
Tests Failed
1 failed, 0 passed
```

The test that failed when we first wrote it at the beginning



Tests failed: 4, passed: 1 of 5 tests – 154 ms

```
Test Results                                  154 ms
  com.muzaffer.swegutenbergprj.Regi           154 ms
    username already exists return fa         101 ms
    less than 2 digit password return f        18 ms
    empty username returns false               17 ms
    empty password return false                18 ms
```

```
Executing tasks: [:app:cleanTestDebugUnitTest, :app:testDebugUnitTest] in project C:\Users\krdln\AndroidStudioProjects\SoftwareEngProject-Muz


> Configure project :app
Warning: The 'kotlin-android-extensions' Gradle plugin is deprecated. Please use this migration guide (https://goo.gle/kotlin-android-extensi
AGPBI: {"kind":"warning","text":"API 'BaseVariant.getApplicationIdTextResource' is obsolete and has be
> Task :app:cleanTestDebugUnitTest
> Ta                  -TO-DATE
```

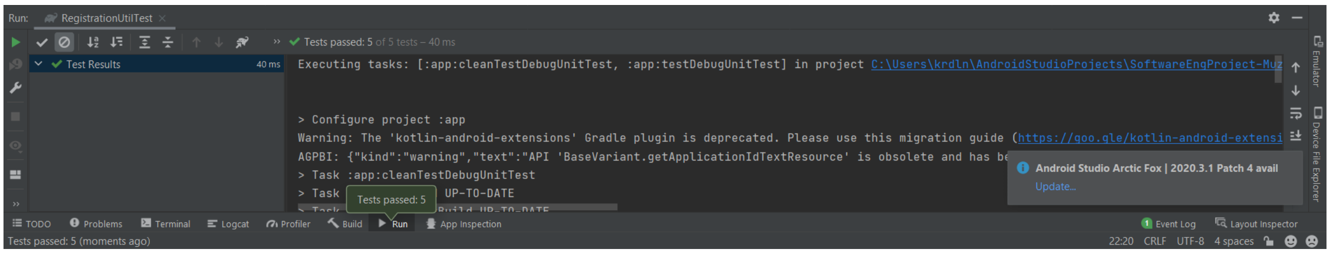
Tests failed: 4, passed: 1

Android Studio Arctic Fox | 2020.3.1 Patch 4 avail
Update...

Tests failed: 4, passed: 1 (moments ago)

61:20 CRLF UTF-8 4 spaces

4 failed, 1 passed



Tests passed: 5 of 5 tests – 40 ms

```
Test Results                                   40 ms
```

```
Executing tasks: [:app:cleanTestDebugUnitTest, :app:testDebugUnitTest] in project C:\Users\krdln\AndroidStudioProjects\SoftwareEngProject-Muz


> Configure project :app
Warning: The 'kotlin-android-extensions' Gradle plugin is deprecated. Please use this migration guide (https://goo.gle/kotlin-android-extensi
AGPBI: {"kind":"warning","text":"API 'BaseVariant.getApplicationIdTextResource' is obsolete and has be
> Task :app:cleanTestDebugUnitTest
> Task                UP-TO-DATE
```

Tests passed: 5

Android Studio Arctic Fox | 2020.3.1 Patch 4 avail
Update...

Tests passed: 5 (moments ago)

22:20 CRLF UTF-8 4 spaces

5 passed

REFERENCES

[1] https://kodlayarakhayat.com/yazilim-muhendisligi/tdd-test-driven-development-nedir/

[2] https://www.mshowto.org/agile-ve-scrum-nedir.html

[3] https://stackoverflow.com/questions/

[4] https://devnot.com/2019/typescript-nedir/

[5] https://www.mshowto.org/node-js-nedir-sagladigi-avantajlar-ve-ornek-uygulama.html

[6] https://www.kocakademi.com/egitim/android-komponentleri-ile-calismak/8633/fragment-nedir/

[7] https://medium.com/turkishkit/swift-nedir-b7ee4a283daf

[8] https://webmaster.kitchen/neden-swift-ve-avantajlari-nelerdir/