)

Inheritance Share Calculator App

Contents

Intermediate Report for "Miras Payı Hesaplayıcı" Application

Esra Kantarcı Çayır

May 2020

1 Introduction

In this paper, a detailed progress report for the "Miras Payı Hesaplayıcı" (Inheritance Share Calculator) will be given. There are several design and business choices were made different than the Project Report 1, which was written in December 2020.

2 Sprint Plans

Our project is determined to consist of 10 sprints in the plan.

2.1 Business Analysis and Preparation

Start date: 2 January 2021
End date: 15 February 2021
Total time: 45 days

In the first sprint, business analysis phase is elaborately covered. Use cases were collected, tried on the similar applications, cross checked the results with Turkish Civil Code Volume IV - Inheritance Law (written by Prof. Dr. Mustafa Dural and Prof. Dr. Turgut Öz). Then a workflow for logic was extracted from the rules. It turns out if the rank is eligible, the steps are pretty much simple and similar, in a sense, we can provide a recursive solution for the dead-but-eligible-inheritors by adding child to them.

In the similar applications, they did not provide adopted child option or hereditary portion of legal share (in Turkish: "saklı pay".) So, as a novelty item, in our application we will also provide legal share information as well.

The similar application's user interface is most likely to be used by Law Faculty students, and not easy to be used by potential inheritors. Therefore, we need to make some UI which is clean, understandable and easy to be used by average people.

First drafts for UI are hand-drawn on papers and possible screens were designed. And as the Flutter is chosen as development technology, I checked out example Material Design projects which were done in Flutter and did similar screen views in my first drafts.

One of the main aims I had while choosing the Senior Project was to get familiar with mobile programming. As I had no prior knowledge on Flutter and Dart, after collecting the necessary knowledge on Inheritance Law in 1 week, I began to enroll Flutter online certified courses. I had completed 1 course on Udemy, 2 free courses in Youtube and enrolled 1 certificate program. During the course sessions, I made similar UIs which may provide me better solutions and learned the architecture. I noted the good practices as well.

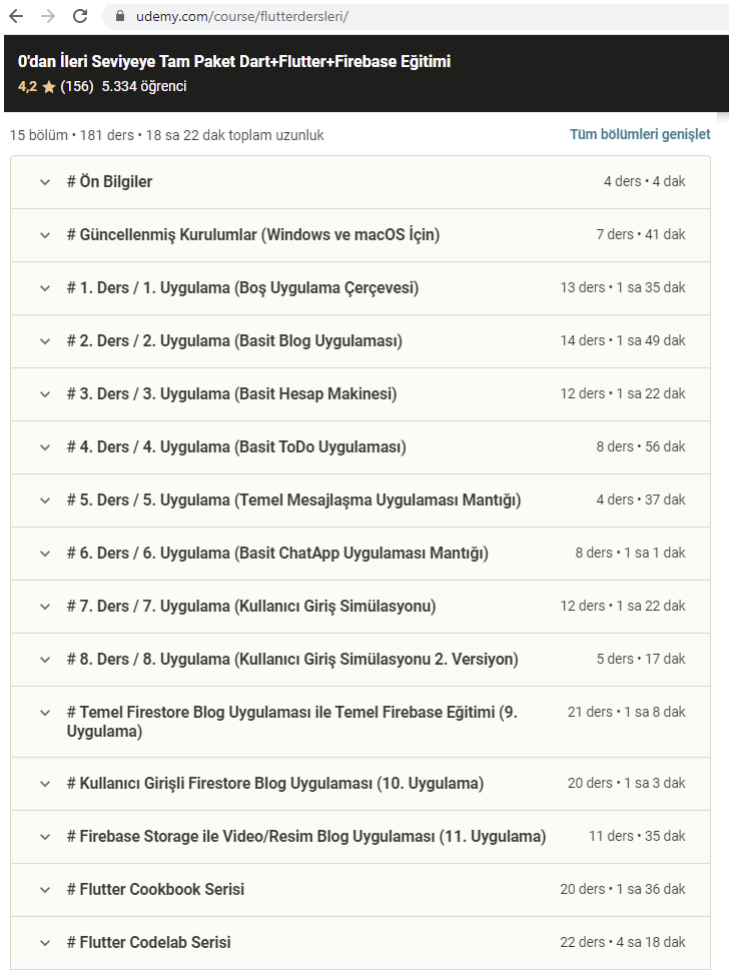

Figure 1: Completed Flutter Course Curriculum

After completing the courses, I redesigned the wireframes and used Balsamiq for the mock-ups, which I will tell about them in the next subsection.

2.2 Charts and User Interface Design

Start date: 16 February 2021
End date: 17 March 2021
Total time: 31 days

Initially, I planned this sprint to be at most 15 days long. However, as I created similar flowcharts and workflows in draw.io, I figured out that if I want to give a simple UI for the average user, I need to plan and design the interfaces better.

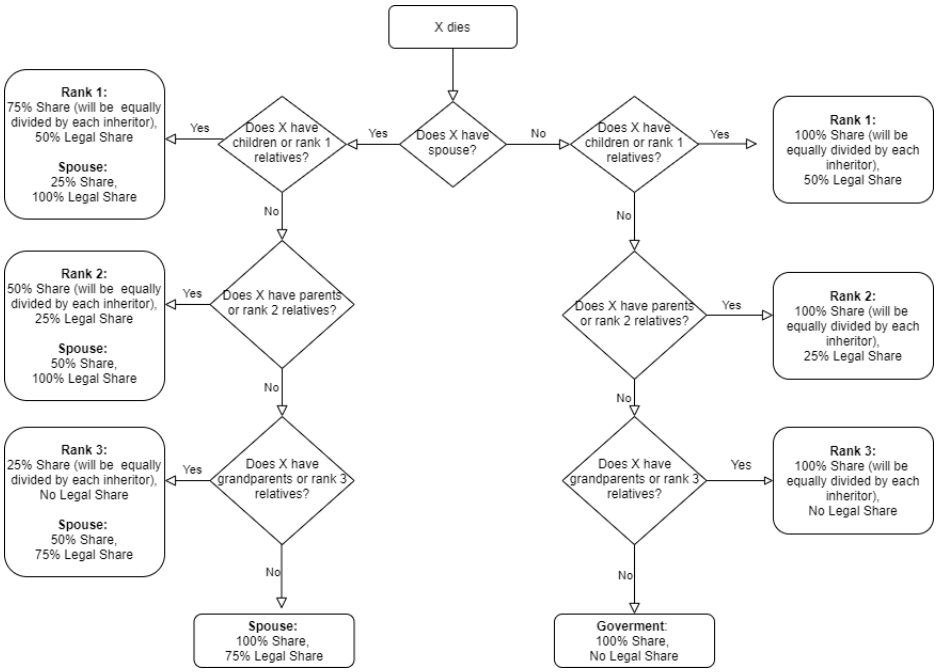

Figure 2: Share Rates Distribution by Rank

At first, this flow was quite simple. However, this is only a general rule flowchart. Adding respective children brings more and more issues, because there maybe recursive inheritance due to dead eligible ranked relatives. For example, if A had 2 children B and C. If A died when B was dead, if B has also 2 alive children called D and E, then B's shares will be inherited equally by D and E as well. So, it complicates the logic by recursively adding. Also there might be double-relative case (in Turkish "çift kan bağı hısım") which also a

problematic case and not covered by other applications as well.

If we were to design an app which reads from JSON or e-Goverment (in Turkish "e-Devlet") family documentation directly, then we had no issues with it. Logic is simple to apply. However, since we get the input from user, we need to make the interface simple and understandable.

Therefore, proposing a solution considering we are in an agile process is necessary. Time is limited. So, I decided to focus on the common cases first. I want to give details on the next sprint, in order not to repeat same design issues in this section.

2.3 Collecting Feedback and Re-designing User Interface

Start date:17 March 2021
End date: 2 April 2021
Total time: 14 days

In order to collect feedback, I had phoned my parents, who are also lawyers and asked mentorship. İlknur Devrim Kantarcı and Ahmet Kantarcı helped me on the ideas, however online meeting was not quite satisfactory since they are not really fond of the idea of Zoom/Skype/Teams. Therefore, I went to my hometown for 5 days and re-discussed with them on the issue.

İlknur Devrim Kantarcı found the UI I designed very complicated and said that she cannot keep the count of the children and would be better to see a list after adding them up. Also, she said she did not want to share all the information with an app, so she would not use e-Goverment documents on the app. This also made me realize that I should not save the data on Firebase or any other DB due to security reasons. This would also bring advantage during deployment on Play Store.

Ahmet Kantarcı told that there are many issues I did not cover, for example there might be several spouses which are died but had children from the heritage owner. Also he wanted me to place a warning that this app will not any legal impact on the courts. So I also added a disclaimer page instead of a splash screen.

Here is my UI wireframe design of the screens done in Balsamiq after the feedbacks:

Figure 3: Disclaimer Page

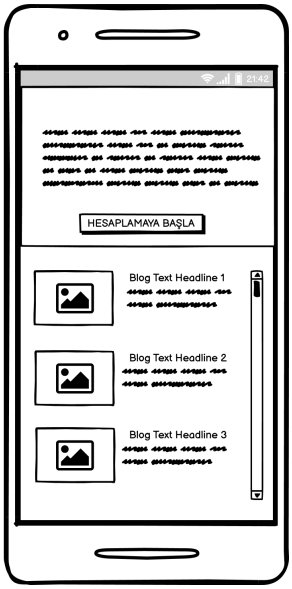

Figure 4: Home Page

After you accept the terms in Disclaimer Page (Figure 3), you are redirected to Home Page. After implementing the business logic, the improvements will be done on Disclaimer Page and the app will remember the "acceptance" state and will return only splash screen.

Home Page (Figure 4) consists of 2 main elements: Start Calculation button and Blog part. In the Blog part, we will provide a kind of knowledgebase with legal documentaries such as petitions and user manual for the app.

After hitting the button on home page, it will redirect us a common Start Page (Figure 5). In this part, we need to know the name of decedent just for detailed scheme for adding children/relatives. Then we need to check if decedent was married and if decedent had descendants such as children or grandchildren.

Radio buttons were placed because it is better to use, in the similar app drop-downs were used and it was quite tiring. The radio buttons are not selected by default so that user should always select the necessary answers. This is Rank 1 evaluation part.

If decedent had spouse (Answer 1: Yes) and descendent (Answer 2: Yes) then we can move forward to Spous eAlive - ChildrenAlive Page (Figure 6).

Here is a point I had forgotten to design as the feedback, but figured out during the implementation part: children may have different mother/father than the decedent's spouse. So, only adding child to spouse will not help. However I covered that case in the implementation by asking the name of other parent. This parent-name answer in the child can be drop-down, but have not

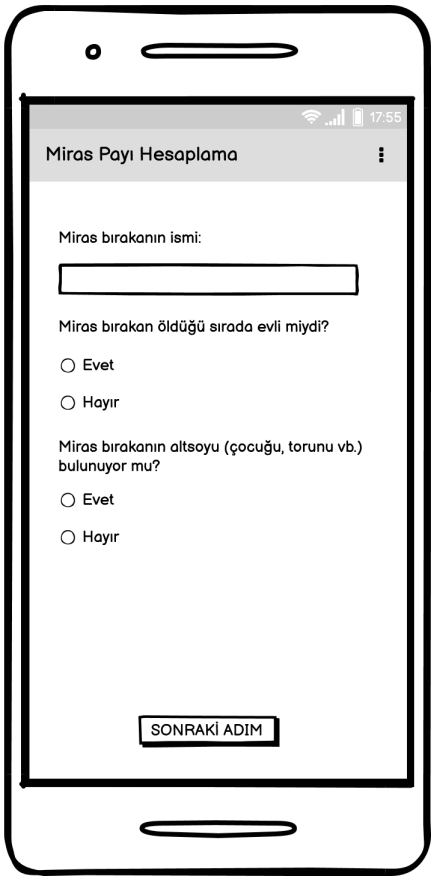

Figure 5: Start Page

had feedback on the issue yet, so I will reconsider it during the business logic implementation sprint.

Also, here is my solution for the recursive approach for the common case I stated in the previous section: Recursively adding descendant to deceased child (Figure 6). However, this is also a design mistake no-one pointed out during getting feedback session. Deceased child may not have descendants. So, sometimes you should not add child to the deceased child and close the "adding window". I added this during implementation while cross checking the use cases.

Here is a sample calculation result page for the imagination, you can check all the mockup screens in the Github Link, so let us not overcrowd this documentary with similar figures. Save button is for saving the results as PDF or PNG format, so we do not store the information on database and users do not need to give same efforts for checking out the same output.

Let us move on the user interface implementation part.

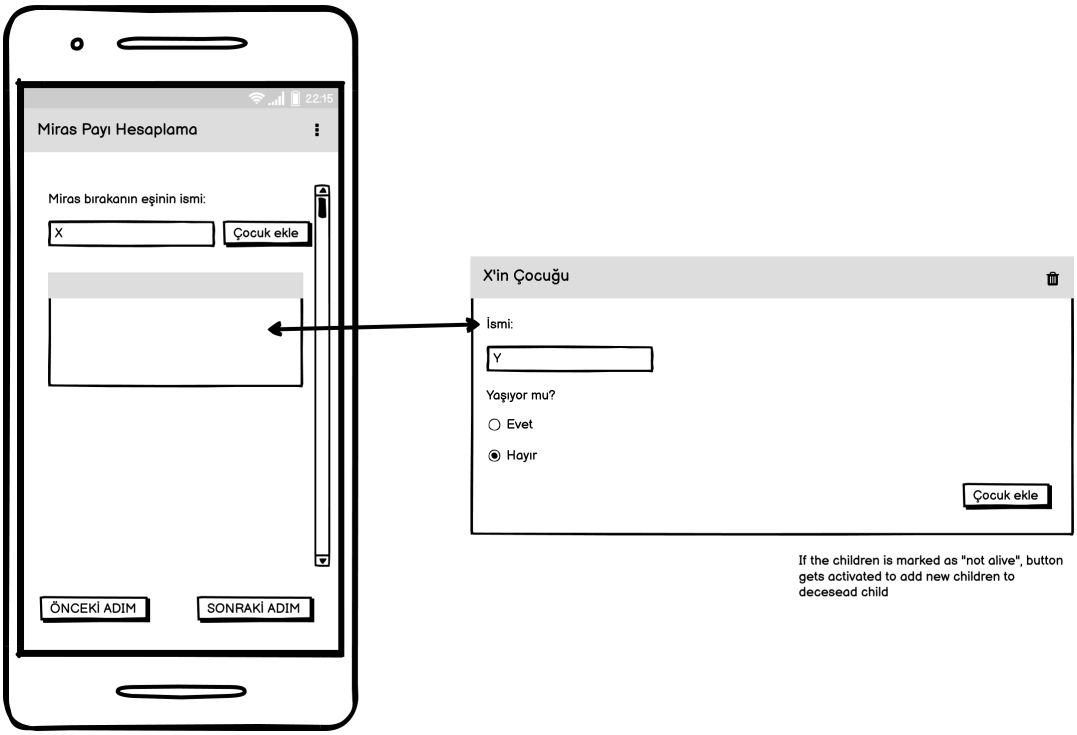

Figure 6: Spouse Alive - Children Alive Page - Adding Descendant to Dead Descendant

2.4 User Interface Implementation

Start date:3 April 2021
Expected End date: 4 May 2021
Estimated sprint time: 31 days
Actual end date: 4 June 2021
Total time: 60 days

Due to unexpected turn of events, I got COVID-19 in April 2021 and after the illness is cured, I had post-symptoms such as tachycardia, allergy and lung stiffness. My husband also got asthma-like issues with his lungs. Therefore I needed to give break on April, because developing a project by myself needs more brain activity, health and high motivation in times like these. Therefore, the implementation phase of the sprint was off the deadline. In May, I had to cool down for 1 week after the midterm projects and checking out in hospital for newly developed allergic issues after COVID-19. Therefore, the sprints were postponed for almost 1 month and 1 week in total.

As I explained in previous sections, during the implementation I had

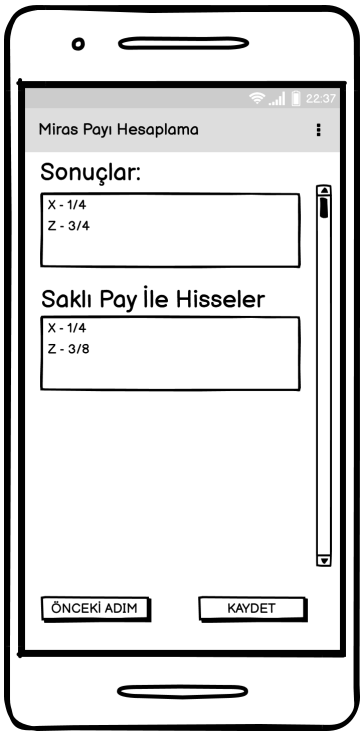

Figure 7: Result Page

2.5 Collecting Feedback (2)

Start date: 4 June 2021
End date: 8 June 2021
Total time: 4 days

2.6 Business Logic Implementation

Start date: 9 June 2021
End date: 9 July 2021
Total time: 30 days

2.7 Unit Testing

Start date: 10 July 2021
End date: 15 July 2021

Total time: 5 days

2.8 Collecting Feedback and Creating User Manual(3)

Start date: 16 July 2021
End date: 20 July 2021
Total time: 4 days

2.9 Stress Testing Before Deployment and Adding Legal Context to Blog

Start date: 21 July 2021
End date: 31 July 2021
Total time: 10 days

2.10 Deployment to Play Store and App Store

Start date: 31 July 2021
End date:
Total time: Depends on Google and iOS, possible time: 7 days

3 User Interface Implementation

You can find out the wireframe mock-ups in the Github Link here, just not to repeat all the work again. Here are design analysis for wireframe:

Major functions for User

- Calculating the share
- Saving the output as a pdf or png
- Reading the blog

Major requirements for calculation

- Getting information on living relatives by rank
- Getting information on deceased eligible ranked relatives
- Number and rank information of alive eligible relatives
- Share and legal share rates enumerated

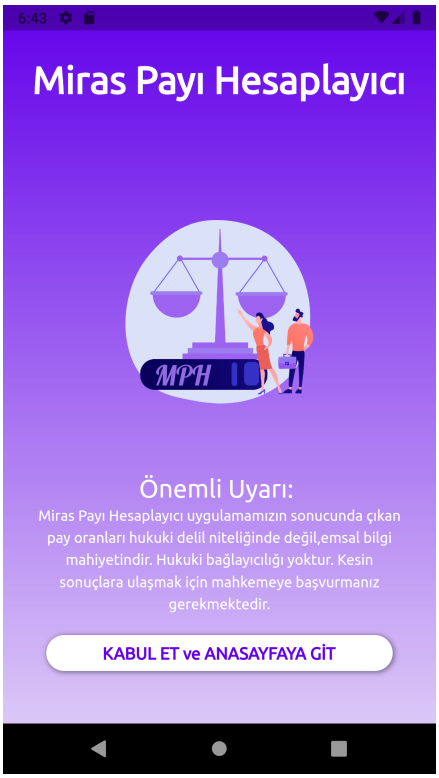

Figure 8: Disclaimer Page

Figure 9: Home Page

3.1 Decisions

3.1.1 Base Colors and Choices

As flat designs are getting more and more popular, finding free licensed icons and graphs are pretty easy. Therefore, my choice for the design will go with clean flat designs. However, during my observations, blue is too widely used in mobile applications, I wanted to go with flat design with indigo/violet color for the. As base color, violet/indigo was also chosen due to its psychological interpretation. Indigo color is widely used in funeral services and it gives intuition and integrity vibes according to several sources which are stated in the bibliography section.

3.1.2 Logo and Font

Logo is not designed as its final form. Just for now, I am using Freepik's loyalty-free design for splash screen and disclaimer screen. I am planning to use "MPH" initials for logo, but have not decided on the Font Family. I am also considering Google Fonts because they are also loyalty-free, Turkish character supportive and easy to refer from Flutter dependencies (since Flutter is also a

Google technology product.)

3.1.3 Forms and Steps

In order to make useful and easy to follow flows for the average users, I decided to use forms and especially radio buttons for simplicity. In the implementation, scroll-able view was used. In the top right corner, which step you are on is also shown. At most there are 4 steps, but the issue with the forms are not the number of the step but adding relatives respectively. So, in the next sprint, I will be adding count information on the each Add Child form's App-bars.

3.1.4 Knowledgebase

Knowledgebase is actually blog part of the application. In the future, I may use Firebase/Firestore to store, add, update, delete the written articles. However, for now since the context is somewhat stable, I aim not to use Internet Connection in the application.

In the knowledgebase, there will be example petitions on the scrollable view from left to right, because there are only limited number of petition type. Other kind of blogs and user manual will be stored as a column scrollable from top to down. I had chosen this kind of design during implementation, because many of Flutter Material Design examples had this kind of views as good practices. Therefore, this part of the implementation is different than the initial mock-ups.

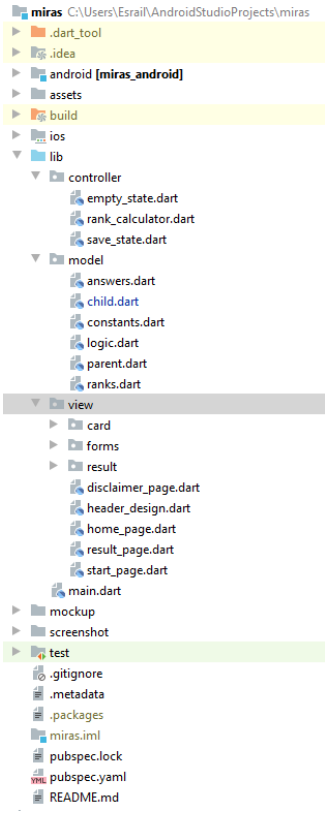

Figure 10: MVC Structure Tree

4 Codes and Commentaries

You can find all my codes and commentaries inside the Github link.

For design pattern, I went for MVC (model-view-controller) pattern which is pretty common for user interface designs and therefore Flutter apps as well. Since I am also familiar with the pattern from my older project, it was easy to decide.

The reference source codes are also stated in the commentaries inside the codes.

5 Bugs to Fix and Future Improvement List

In our demo there are many cases which we will be fixing in the next releases after we get the first approval. Here is the list of bugs to fix and features to add in the next release:

- Expanded and SingleScrollView will be applied in each screen for responsiveness.
- Adding parent by drop-down button to a child will be implemented.
- Even if the eligible ranks' relative does not have descendants, user can still add the dead relative into the list.
- User should only approve the disclaimer at the first fly.
- Hardcoded ranking should be implemented as enumerated.

5.1 What to expect from next sprints?

- Implementation of the business logic
- Adding child with numbers and parental information
- Adding child to deceased eligible ranked relative
- Blog, knowledgebase and user manual contents
- Result page
- Testing
- Deployment to Play Store

6 Progress of the Project

In the progress of the project, application development is still in sprint 4 (User Interface Implementation). Due to overdue sprints, the project completion schedule is postponed to end of the July. Therefore, as one-person-team, I collect feedback from potential users before the respective sprint time comes and change the necessary parts during implementation for faster development.

Screen views are ready to be deployed, we can say User Interface Implementation phase is almost complete. We need to cover adding children and adding parent cases and using flex factors for the better UI responsiveness. After all, Flutter was mainly chosen for the responsiveness and cross platform advantages.

I also created use case charts from Inheritance Law Practices Books in order to get more prepared before unit testing phase.

You can check the development and progress in the Github link. Working interface documentation will be ready after business logic is implemented, however the designed guide and documentation can be found in the mockup directory as *.bmpr (Balsamiq project extension) in the Github repository.

7 Conclusion

As a lawyer who is also studying Computer Science Engineering, I decided to develop an app which may inform people about the rights they have before going court or lawyer. After analyzing the similar applications and possible use cases, I decided to use Flutter for mobile responsiveness and started to get educated on the Flutter and Dart concepts, which also helped me starting a mobile development career.

The design process took longer than expected due to COVID-19. Also, complicated procedures of the business logic during the practical use cases made the design harder than expected. Pattern choice was Model-View-Controller and Flutter's Material Design was used. Mock-ups, feedback are stated in the project, code can be found in Github, demo, tests and guide will be done after business implementation sprint.Sample workflow can be found as Balsamiq Project in the Github repository.

I seek to improve the development by using real testers. I had good feedback from my old Law Faculty students and lawyer friends. Screenshots were also loved by average users who are not familiar with law as well. Therefore, I am hopeful from the future results. After adding tests, at first, the app will be deployed to Google Play Store. Then, App Store will be my next aim due to more limiting constraints of iOS App Development.

As a conclusion, getting positive feedback from people and create something by myself feels pretty good and I learned necessary technologies for my future path. Thank you.

8 Bibliography

- Github - Esra Kantarcı - MirasApp:
  https://github.com/EsraKantarci/mirasapp
- Github - Robby Rahmana - Covid19 App(Used for Scroll Design):
  https://github.com/robbyrahmana/app_covid19
- Github - Marcus NG - Flutter Travel UI Starter:
  https://github.com/MarcusNg/flutter_travel_ui_starter
- GitHub - Celal Kutluer - Dynamic Dropdown API:
  https://github.com/celalkutluer/f_drpdwn_dyn_api
- GitHub - Giancarlo - Form Bloc(Used for Dynamic Design):
  https://github.com/GiancarloCode/form_bloc
- GitHub - Putraxor - Flutter Multipage Form(Used for Child Card):
  https://github.com/putraxor/flutter_multipage_form
- GitHub - Jide Guru - Flutter Travel(Used for Material Design Examples):
  https://github.com/JideGuru/FlutterTravel
- GitHub - Pawan - Flutter UI Kit(Used for Material Design Examples):
  https://github.com/iampawan/Flutter-UI-Kit
- Color Psychology - Indigo Color in Business:
  https://www.empower-yourself-with-color-psychology.com/indigo-in-business.html
- Color Psychology - Indigo:
  https://www.colorpsychology.org/indigo/
- Türk Özel Hukuku - Miras Hukuku Cilt IV - Mustafa DuralTurgut Öz
- Academind - Flutter Crash Course for Beginners 2021 - Build a Flutter App with Google's Flutter Dart:
  https://www.youtube.com/watch?v=x0uinJvhNxI
- Udemy - Oğuz Başaran - 0'dan İleri Seviyeye Tam Paket Dart+Flutter+Firebase Eğitimi
- Free Online Courses Official - Flutter Dart - The Complete Flutter App Development Course
  https://www.youtube.com/watch?v=WPpoRkoeQVYlist=PLziVQlsc4EpCin9YcyI43QV0Jk8EQVraA