THE PROBLEM SOLVER: A MOBILE PLATFORM TO MEDIATE TEENAGER FAMILY RELATIONSHIP USING DART AND MACHINELEARNING

Ziheng Guan[1] and Ang Li[2]

[1]Arcadia High School, 180 Campus Dr, Arcadia, CA 91006
[2]California State University, Long Beach,
1250 Bellflower Blvd, Long Beach, CA 90840

ABSTRACT

Family conflicts between parents and their children are nothing new and are something experienced by many in such situations [1]. These conflicts can even be exacerbated by cultural differences that exist between the two parties, especially in cases where the parents and child were raised in different countries, cultures and/or generations [2]. This description illustrates my personal experiences of conflict with my parents, which is what inspired me to create this app: The Problem Solver app. The app differs from other methods that could be applied to resolve these conflicts in that it facilitates more direct communication between the two conflicting parties, which would hopefully result in a more rapid and successful conflict resolution [3]. Naturally, there were challenges I faced in the making of the app, but I was eventually able to work through these and build a working product. I will also explore some related works and research into this topic that were helpful in supporting the idea that cultural differences between differently raised generations can have an impact on familial relations [4]. Then, I give a general overview of the system of the app and finally delve into possible limitations of the app and further steps I could take in the development of the app.

KEYWORDS

Machine Learning, Communication, Cultural Differences, Flutter.

1. INTRODUCTION

The background of this topic stems from conflicts that I have experienced at home with my parents. More specifically, my parents and I were raised in very different cultures during our respective childhoods, with my parents having been raised in a traditionally “Eastern" society (China), while I have been raised in the United States (US) since middle school [5]. Thus, I have attended school in the US for much of my life and have been raised in a traditionally “Western” society. Our conflicts originated from occurrences such as disagreements over what amount of studying is an acceptable level. For example, my parents were brought up and taught that studying and working hard is the most important thing one can do, to the point that one should be studying almost anytime one has free time. However, I do not want to study all of the time, as I would prefer to also do other things (playing video games, sports or listening to music) to relax and unwind. From a Western society perspective, studying all the time and never taking a break would be frowned upon or even be seen as concerning [6]. However, in an Eastern society, that would be seen as the “standard” for what one should be doing. As a result, my parents and I have

David C. Wyld et al. (Eds): AI, AIMLNET, BIOS, BINLP, CSTY, MaVaS, SIGI - 2022
pp. 285-294, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121825

frequently come into conflict over ideas such as these; this inspired me to seek a solution so my parents and I can work out our issues and communicate better. This idea also extended into relationships outside of just parental conflict, as I realized that the app could also be applied to relationships with friends, siblings and other people in my life. Thus, this is how I envision this application being more generalizable and usable for people in different contexts.

This idea of communicating our problems between the people we are close with is important, as we need to be sure our relationships are good and healthy. More specifically, with a lack of communication, problems and issues in a relationship can build up and build up to the point where they could boil over into an even larger conflict - one that could've been avoided or resolved sooner with better communication [7]. However, even if one knows what they want to communicate regarding an issue or what they would like to say to the conflicting party, it is not always the easiest to directly speak to the conflicting party. That is, sometimes it is hard to make oneself confident enough to bring up an issue, especially when there is fear of punishment or not being listened to. This is where the app provides another unique perspective, being that it allows the users of the app to take time to think about and carefully craft what they are actually trying to communicate instead of essentially having to do it "on the spot." Also, this could potentially allow users to bring up issues that they may have difficulty discussing with someone face to face.

There are not many tools/existing methods that exist to solve issues that present themselves in close parental relationships such as this, but one possible way that a parent could solve this issue (at least through personal experience) is by taking their child to see a specialist like a psychologist. The parents may do this due to their child exhibiting behavior the parents would deem as "unnatural." For example, my parents and I have argued over the amount of time that I need to be using electronics such as iPads, a Chromebook, etc. Using electronics such as these has become increasingly necessary within the past few years, as many classes and assessments have moved online and much of the homework for classes must be either completed and/or submitted online. However, my parents tend to view being on electronics as me being distracted or unproductive, as they don't fully understand how truly everything is done on the computer these days. My parents see me using these devices frequently and seem to assume that there is something mentally wrong with me that is causing me to become very easily distracted or that is causing me to need to be on an electronic device all of the time. From my perspective, clearly, there is nothing mentally wrong, and there is no issue with being on devices so frequently, as that is just the way that the world works today, especially in a school setting.

In such a setting, the parents would likely expect the psychologist to offer a "cure" or an explanation as to why their child is behaving in such a way, believing that the psychologist could solve the issue or prescribe a medicine that would. There is a glaring problem with this "method," however, being that this method would be completely unhelpful and ineffective. That is, since there is no real psychological issue, and since all I am trying to do is to use my electronic devices to complete schoolwork, there is nothing that the psychologist would be able to do to help, as there is no issue to be solved. Perhaps the psychologist would be able to better offer insight into why these familial disagreements are occurring in the first place and offer counseling in that area, but we assume that the parents who are reaching out to the psychologist are convinced that they are firmly in the right and that they do not need any counseling themselves. So, as a result, this method of resolving the issue would be wholly ineffective and perhaps may raise even moretensions between the parents and the child.

There is a somewhat similar app that exists (called the Conflict Coach app), but this app does not offer the same set of services as my app. The conflict coach offers more along the lines of suggestions for defusing tense situations with talking points or suggesting gentle ways to ease into conversations on difficult topics. Thus, there appears to be less direct communication between the two parties that are experiencing conflict (at least less direct communication through the app itself). While this may be useful for older people or for work conflicts, the Problem Solver app that I built offers more direct communication between the conflicting parties, which may lead to a more efficient problem resolution. Another drawback of the Conflict Coach app is that it is not free, it costs $28, whereas my app is free.

My tool is called the Problem Solver app, and it is an android-based application written using Flutter that can be used to solve conflicts that arise in familial and other close relationships [8]. The app is purposed as a mobile app for android personal cell phones (hence its availability on the Google Play Store). When the app first launches, the user is greeted with three introductory screens that describe the purpose of the app; that it is for solving problems between parents and their children and can help the user set goals to make that happen. The user can skip these screens or simply scroll through them. After this, the user is prompted to login or to create an account if they don't already have one. After logging in, the user can view their profile to edit account information or they can jump right in to solving their conflicts. To start, the user clicks on the "Create a Conflict" button, which brings up the corresponding page to create a conflict. On this page, the user can give a name to the conflict, describe the conflict and who it involves, set an urgency level from a dropdown list, select a type of conflict from a dropdown list (study, computer time, etc) and select a start date and time for the issue. The Active Conflict Page holds a list of all current active conflicts (those that have not yet been resolved). On this page, one can click on the conflicts in order to edit their contents (name, who it involves, start date, urgency, etc). One can also click on the plus button in order to add a new conflict. The final page is the Old Conflicts page, where the user can view conflicts that have been resolved. Once again, the user can click on the conflicts in order to view their details. Additionally, the user can click on these conflicts to view them again and can also make the conflicts "active" again, which would put that conflict back to the Active Conflicts page.

One of the clear differences between the app and the existing methods discussed previously (e.g psychological counseling) is that this app involves the two conflicting parties communicating with one another directly instead of one of the parties (such as the parents) not really attempting to listen to their child as much and just assuming that their way is the best. As such, there is no "third party" involved in the resolution of the issue. I feel as though this would result in a more direct line of communication between the two conflicting people, and would result in more conflict resolution than if other methods had been used. In my mind, one of the greatest weaknesses of other methods (specifically therapy) is that it would not get to the root cause of the issue, that of the cultural differences that exist across generations. This is where the app becomes very useful, as it allows a safer space for everyone to communicate their issues and how they are feeling about the conflict, which in turn could be very productive towards solving the root cause of some of these issues.

The remainder of the paper will proceed as such: Section 2 details some of the challenges and difficulties I faced in the process of developing the app. Then, Section 3 will describe an overview of the components (each different page) of the app, what they look like (through screenshots), the code that makes up these segments and finally a general overview of how the app handles user data. Finally, Section 4 will explore some related works and studies on this same topic and compare these works and studies to my Problem Solver app, while Section 5 will conclude the essay and wrap up the ideas covered through this writing.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Finding Resources

One of the most frequent challenges I encountered in the undertaking of this project was that, when I would come across an issue or get stuck on something, I had a difficult time finding resources that could answer my questions. Naturally, I utilized standard resources such asGoogle search and the Flutter documentation in order to attempt to answer my questions, but those were only useful in a more general sense and did not exactly apply directly to what I was having difficulties with [9]. Specifically, the documentation was somewhat difficult to read, as it is all quite technical and more describes the functionalities of the language and offers few examples. As a result, there were times that I felt that I had hit a dead end on the project. I was able to resolve this issue by making use of other resources that were available to me, such as contacting teachers for help or even asking my friends if they could check my work in case their new perspective was able to offer helpful insight.

2.2. Learning new knowledge

Another challenge that I faced in the building of this app relates to my previous education in computer science. Specifically, I have taken computer science classes during my time in high school, so I have experience with computer science concepts, coding, etc. However, these classes that I have taken are more theoretical in their scope rather than specific to building things like apps, games or other programs. Specifically, computer science classes tend to focus on topics such as algorithms, simple computer communications or just a basic understanding of how a computer works. As a result, I had to teach myself quite a bit about using Flutter, publishing an app on the Google Play Store and Firebase (cloud based user data handling, also Google based). I was able to work my way through this issue by making use of as many online resources as I could, such as YouTube tutorials and examples of how Flutter is used. In cases where these resources were not enough, I took a similar approach as before and sought help through one-on-one time with my teachers as well as consulting with my friends in order to get their input.

2.3. Becoming Distracted

A third challenge that I faced in the development of this app relates somewhat to the issues that I encountered when I felt as though I had hit dead ends on the project. Specifically, I felt it easy to become distracted and become engaged in other things such as playing video games or watching videos on YouTube, especially at times where I felt frustrated with working on the app or felt that I was not making much progress [15]. This would also occasionally extend into my schoolwork as well, which resulted in me occasionally falling behind in classes, resulting in even less time for me to commit to the development of my app. Once I realized I was using my time poorly, I decided to solve this issue by creating a schedule for myself so that I was able to keep up with my schoolwork and set aside time to work on my app as well as time for relaxation.

3. SOLUTION

To take an overview of the entire system, one can see from Figure 1 below that a user would begin by downloading the app from the Google Play store on a supported device (an Android-based mobile phone). Once the user navigates through the three introductory screens (which are shown in interface screenshots below), the user is prompted to either log in or sign up if they do not already have an account. If the user needs to sign up, they will be taken to a separate page to do so, after which their data will be stored in the Firebase cloud database with the app creator being notified of a new user signing up. Then, the user will be taken to the home page, which is also the Active Conflicts Page (this is where the user would end up if they already had an account and had just logged in). Once the user is at this home screen, they are able to take multiple actions (which Figure 1 compresses into general "User Actions" for simplicity). These actions include: creating a new conflict, viewing/editing an active conflict, viewing/editing old conflicts and viewing/editing user account information. Upon the user making any changes to any conflicts or account information, that respective information will be updated in the Firebase cloud in order to keep user data current.

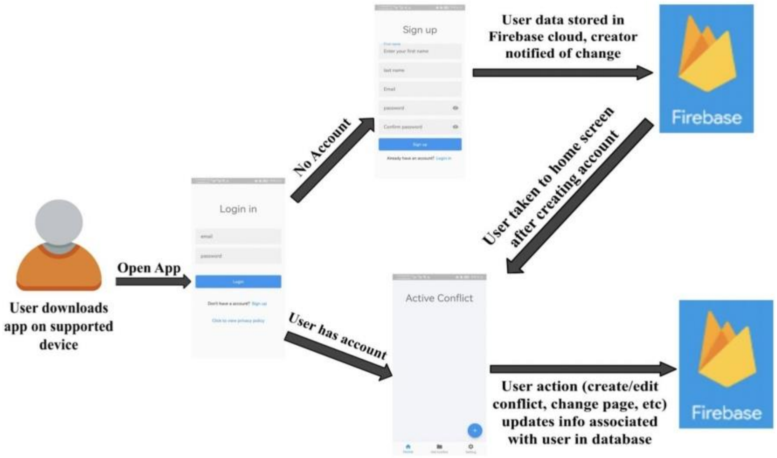
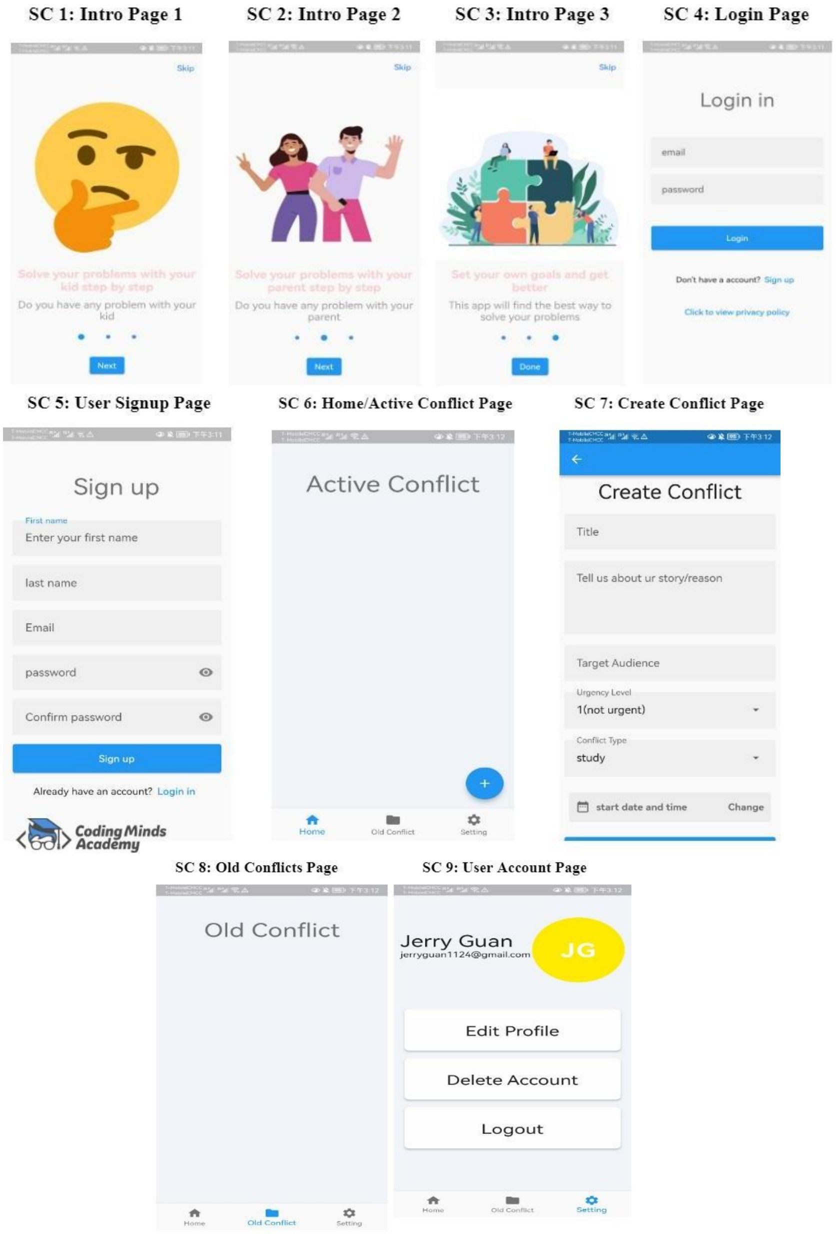

User downloads app on supported device
Open App
Login in
email
password
Login
Don't have a account? Sign up
Click to view privacy policy
No Account
Sign up
Enter your first name
last name
Email
password
Confirm password
Sign up
Already have an account? Login in
User data stored in Firebase cloud, creator notified of change
Firebase
User taken to home screen after creating account
User has account
Active Conflict
User action (create/edit conflict, change page, etc) updates info associated with user in database
Firebase

Figure 1. Overview of the solution

For the purposes of this paper, we define the "components" of the app to be each different "page"that the user has the ability to perform actions on. One can also see that, below, we have provided screenshots of each page of the app in order to further clarify what happens on each page. In Screenshots 1, 2 and 3 (SC 1-3), we can see the first three initial pages that greet the user upon the opening of the app. As is suggested by the "Skip" button and the scrolling list at the bottom, the user has the ability to either skip all of these pages or just scroll through them (SC 1-3). Upon passing these pages, the user is prompted to either log in or to create an account if they do not already have one (SC 4 & 5). After successfully logging in, the user comes upon the Home Screen/Active Conflicts screen, from which the user can navigate to other pages such as the Old Conflicts or Settings pages (SC 8 & 9). The user can also use the "Plus" button in the bottom right hand corner to create a new conflict (SC 6 -> SC 7). When there are active conflicts, they will appear on the home page (SC 6) and can be clicked on to edit their contents, bringing up a screen similar to the Create a Conflict page (SC 7) where the user can edit the contents ofthe conflict. In the Old Conflicts page (SC 8), the user is once again able to click on previously resolved conflicts to view their contents, and can even choose to "restart" the conflict if they feel that a similar issue has already been discussed. Finally, on the User Account page (SC 9), the user is able to see their name, associated email account and can click on buttons that can edit the user's profile, delete the user's account or simply log out.



SC 1: Intro Page 1

Skip

Solve your problems with your kid step by step

Do you have any problem with your kid

Next

SC 2: Intro Page 2

Skip

Solve your problems with your parent step by step

Do you have any problem with your parent

Next

SC 3: Intro Page 3

Skip

Set your own goals and get better

This app will find the best way to solve your problems

Done

SC 4: Login Page

Login in

email

password

Login

Don't have a account? Sign up

Click to view privacy policy

SC 5: User Signup Page

Sign up

First name

Enter your first name

last name

Email

password

Confirm password

Sign up

Already have an account? Login in

Coding Minds Academy

SC 6: Home/Active Conflict Page

Active Conflict

+

Home Old Conflict Setting

SC 7: Create Conflict Page

Create Conflict

Title

Tell us about ur story/reason

Target Audience

Urgency Level

1(not urgent)

Conflict Type

study

start date and time Change

SC 8: Old Conflicts Page

Old Conflict

Home Old Conflict Setting

SC 9: User Account Page

Jerry Guan

jerryguan1124@gmail.com

JG

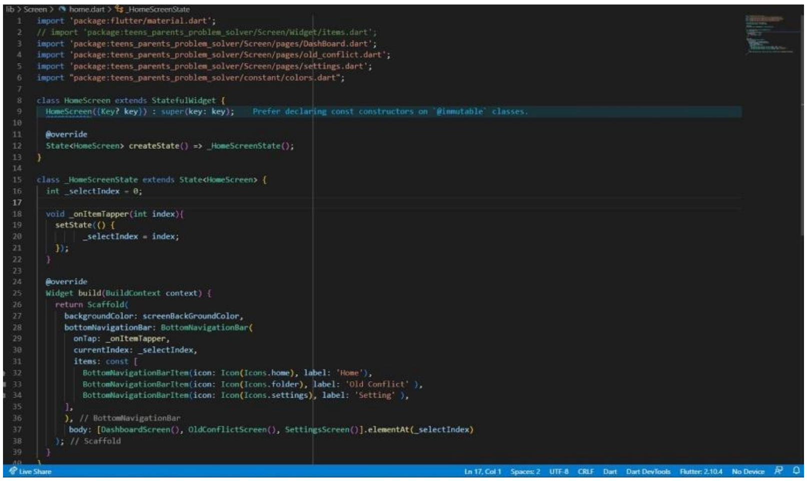
Edit Profile

Delete Account

Logout

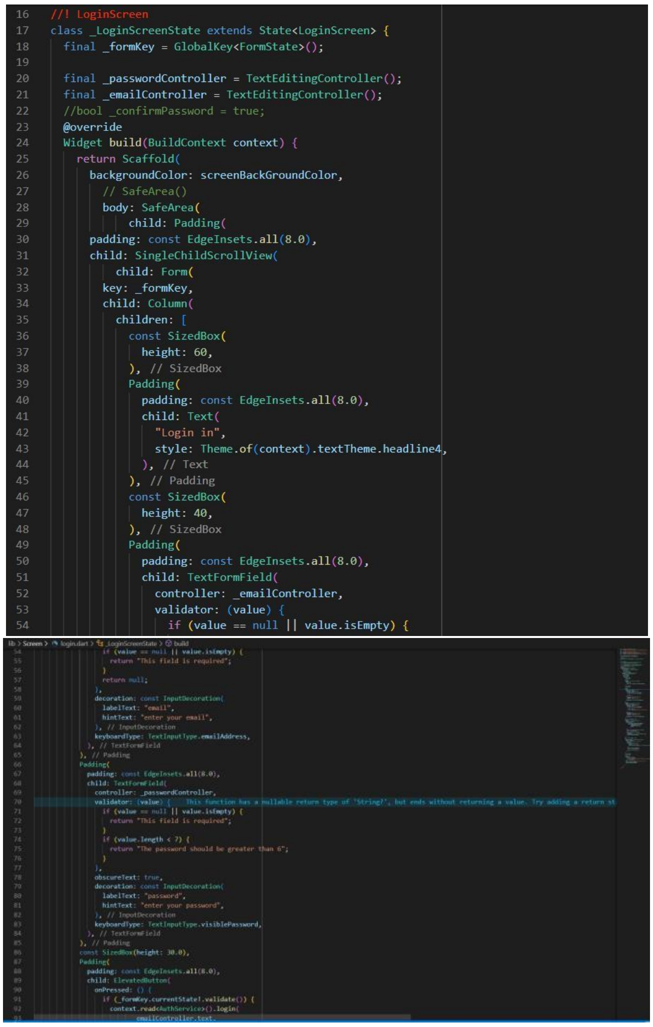
Home Old Conflict Setting

Figure 2. 3 - 4 parts of code screen



```
import 'package:flutter/material.dart';
// import 'package:teens_parents_problem_solver/Screen/Widget/items.dart';
import 'package:teens_parents_problem_solver/Screen/pages/DashBoard.dart';
import 'package:teens_parents_problem_solver/Screen/pages/old_conflict.dart';
import 'package:teens_parents_problem_solver/Screen/pages/settings.dart';
import "package:teens_parents_problem_solver/constant/colors.dart";

class HomeScreen extends StatefulWidget {
  HomeScreen({Key? key}) : super(key: key);    Prefer declaring const constructors on `@immutable` classes.

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  int _selectIndex = 0;

  void _onItemTapper(int index){
    setState(() {
          _selectIndex = index;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: screenBackGroundColor,
      bottomNavigationBar: BottomNavigationBar(
        onTap: _onItemTapper,
        currentIndex: _selectIndex,
        items: const [
          BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'),
          BottomNavigationBarItem(icon: Icon(Icons.folder), label: 'Old Conflict' ),
          BottomNavigationBarItem(icon: Icon(Icons.settings), label: 'Setting' ),
        ],
      ), // BottomNavigationBar
       body: [DashboardScreen(), OldConflictScreen(), SettingsScreen()].elementAt(_selectIndex)
    ); // Scaffold
  }
}
```

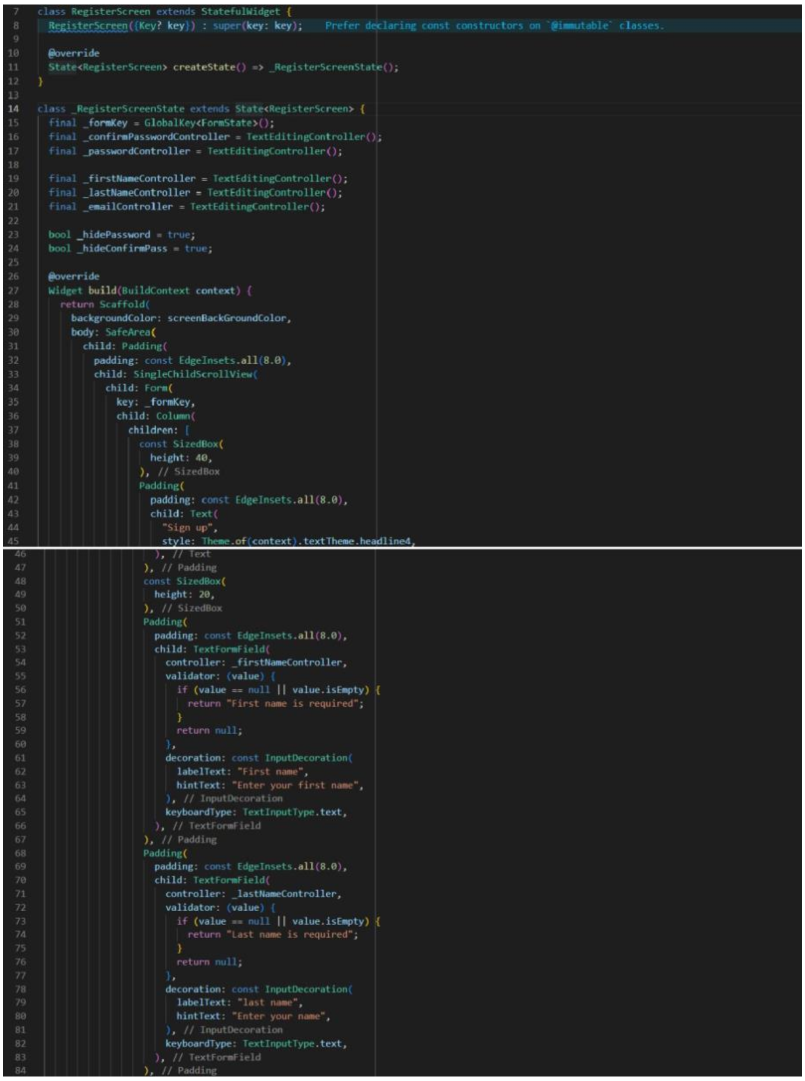
Figure 3. Code of the home screen



```
//! LoginScreen
class _LoginScreenState extends State<LoginScreen> {
  final _formKey = GlobalKey<FormState>();

  final _passwordController = TextEditingController();
  final _emailController = TextEditingController();
  //bool _confirmPassword = true;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: screenBackGroundColor,
        // SafeArea()
        body: SafeArea(
            child: Padding(
      padding: const EdgeInsets.all(8.0),
      child: SingleChildScrollView(
          child: Form(
        key: _formKey,
        child: Column(
          children: [
            const SizedBox(
              height: 60,
            ), // SizedBox
            Padding(
              padding: const EdgeInsets.all(8.0),
              child: Text(
                "Login in",
                style: Theme.of(context).textTheme.headline4,
              ), // Text
            ), // Padding
            const SizedBox(
              height: 40,
            ), // SizedBox
            Padding(
              padding: const EdgeInsets.all(8.0),
              child: TextFormField(
                controller: _emailController,
                validator: (value) {
                  if (value == null || value.isEmpty) {
```

```
                  if (value == null || value.isEmpty) {
                    return "This field is required";
                  }
                  return null;
                },
                decoration: const InputDecoration(
                  labelText: "email",
                  hintText: "enter your email",
                ), // InputDecoration
                keyboardType: TextInputType.emailAddress,
              ), // TextFormField
            ), // Padding
            Padding(
              padding: const EdgeInsets.all(8.0),
              child: TextFormField(
                controller: _passwordController,
                validator: (value) {    This function has a nullable return type of 'String?', but ends without returning a value. Try adding a return st
                  if (value == null || value.isEmpty) {
                    return "This field is required";
                  }
                  if (value.length < 7) {
                    return "The password should be greater than 6";
                  }
                },
                obscureText: true,
                decoration: const InputDecoration(
                  labelText: "password",
                  hintText: "enter your password",
                ), // InputDecoration
                keyboardType: TextInputType.visiblePassword,
              ), // TextFormField
            ), // Padding
            const SizedBox(height: 30.0),
            Padding(
              padding: const EdgeInsets.all(8.0),
              child: ElevatedButton(
                onPressed: () {
                  if (_formKey.currentState!.validate()) {
                    context.read<AuthService>().login(
                          emailController.text,
```

Figure 4. Code of the login page



```
class RegisterScreen extends StatefulWidget {
  RegisterScreen({Key? key}) : super(key: key);    Prefer declaring const constructors on `@immutable` classes.

  @override
  State<RegisterScreen> createState() => _RegisterScreenState();
}

class _RegisterScreenState extends State<RegisterScreen> {
  final _formKey = GlobalKey<FormState>();
  final _confirmPasswordController = TextEditingController();
  final _passwordController = TextEditingController();

  final _firstNameController = TextEditingController();
  final _lastNameController = TextEditingController();
  final _emailController = TextEditingController();

  bool _hidePassword = true;
  bool _hideConfirmPass = true;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: screenBackGroundColor,
      body: SafeArea(
        child: Padding(
          padding: const EdgeInsets.all(8.0),
          child: SingleChildScrollView(
            child: Form(
              key: _formKey,
              child: Column(
                children: [
                  const SizedBox(
                    height: 40,
                  ), // SizedBox
                  Padding(
                    padding: const EdgeInsets.all(8.0),
                    child: Text(
                      "Sign up",
                      style: Theme.of(context).textTheme.headline4,
                    ), // Text
                  ), // Padding
                  const SizedBox(
                    height: 20,
                  ), // SizedBox
                  Padding(
                    padding: const EdgeInsets.all(8.0),
                    child: TextFormField(
                      controller: _firstNameController,
                      validator: (value) {
                        if (value == null || value.isEmpty) {
                          return "First name is required";
                        }
                        return null;
                      },
                      decoration: const InputDecoration(
                        labelText: "First name",
                        hintText: "Enter your first name",
                      ), // InputDecoration
                      keyboardType: TextInputType.text,
                    ), // TextFormField
                  ), // Padding
                  Padding(
                    padding: const EdgeInsets.all(8.0),
                    child: TextFormField(
                      controller: _lastNameController,
                      validator: (value) {
                        if (value == null || value.isEmpty) {
                          return "Last name is required";
                        }
                        return null;
                      },
                      decoration: const InputDecoration(
                        labelText: "last name",
                        hintText: "Enter your name",
                      ), // InputDecoration
                      keyboardType: TextInputType.text,
                    ), // TextFormField
                  ), // Padding
```

Figure 5. Code of the register page

4. Related Work

One related work that the co-author of this paper compared my app to is an app called the Love Nudge app, which is something that he uses with his partner in his personal life. The Love Nudge app is based on the idea of the Five Love Languages, a book by Gary Chapman that describes 5 Love Languages as the primary way in which people receive affection from their partners (Chapman, 1992) [10]. As may be apparent from the name and focus topic of the app, however, the Love Nudge app is designed primarily with couples/romantic relationships in mind, not relations between parents and children. Thus, I would say that my app is more useful in the area of general family conflicts rather than only in the limited scope of romantic relationships. The Love Nudge app does hold some similarities to my app, however, in that the Love Nudge app also supports the ability to create conflicts/issues to be resolved, which can be seen by the other user in the app rather than necessitating that the issue be talked about verbally first.

A second related paper I explored was an essay by Preevo and Tamis-LeMonda (2017) titled “Parenting and globalization in western countries: explaining differences in parent–child interactions” [11]. This paper essentially discusses how parents who are the ethnic minority "differ from majority parents in parenting values, child rearing goals and resources—differences that affect parenting practices and children's development" (Preevo & Tamis-LeMonda, 2017). In the case of the paper, "ethnic minority parents" were defined to be parents who were of a minority group relative to the majority of other parents raising children in a similar socio-economic background and geographical location. That is, a "minority parent" could be a pair of Asian parents coming from an Asian background raising their child in a community that is primarily made up of white families and/or families coming from a non-Asian background. The most applicable portion of this paper describes traditional values in Asian families stemming back to religious foundations in the values of Confucianism, these being Qian, Chi, Yue, He and Xiao, the last of which translates to "filial piety," meaning respect for one's parents and other elders. As the authors describe, "One way that filial piety is expressed is through educational success. It is thus unsurprising that pursuing knowledge to improve educational success is an important childrearing goal in Chinese communities" (Preevo & Tamis-LeMonda, 2017). This would tend to support the idea that my parent's upbringing in a traditionally Chinese society influences their thoughts that I should be studying all of the time and devoting much of my time to schoolwork. Additionally, this article also suggests that cultural differences can cause clashes between these generations, further supporting the idea that no one party is really right or wrong, it can all be a matter of perspective.

A third related work that I explored had to do with conflict resolution in families with adolescents [12]. In the article, Smetana et. al describes how conflict resolution in families with adolescents is a commonly necessary task, and that the ways in which conflicts were resolved and the topics of those conflicts varied due to influence from variables such as age, sex and conflict topic (1991). This relates to my work, as I feel as though my app could open another possible avenue for adolescents and their parents to be able to communicate their issues. This is not to say that the Problem Solver app would be the solution to everyone's issues. Rather, I feel as though the newer perspective the app provides could play an important role in resolving some issues that were not handled well using traditional methods. Another important point that this article makes is that one of the possible variables that had not been previously accounted for in this area is the topic of the conflict, which seemed to have a significant impact on the outcomes of the conflict resolution in the study (Smetana et. al, 1991). Once again, I feel as though my app may be able to help in such situations, given that the topic of the conflict is one of the necessary components in the creation of a conflict in the app.

5. CONCLUSIONS

As has been discussed throughout this paper, I have designed an app called the Problem Solver app that aims to be of aid in the resolution of conflicts that arise between parents and their children . I wrote the app (using Flutter) to be an android-based mobile app available for free on the Google Play store. My desire to build such an app stems from personal experience, being thatI have experienced conflicts with my parents that we had difficulty resolving, inspiring me to try to create a different way to attempt to solve these issues. I also took into account other methods that have/could have been used to solve these same problems and attempted to have my app cover up the weaknesses that the other resolution methods tended to have. I also took slight inspiration from the related works that I researched, specifically the article about conflict resolution in families with adolescents. These works led me to be even more inspired to make a relatively unique product that I felt could offer something new. While we were unable to design a full experiment to determine the effectiveness of the app, I feel as though the app could be quite effective and could help families or other people who have a difficult time expressing what they are struggling with [13].

Some of the limitations that exist in the current version of the app is that we are not exactly sure how effective it would actually be in resolving conflict (at least, we haven't been able to perform an experiment to quantify such a thing). Additionally, the app is only available for android-based mobile devices, which significantly limits the pool of people able to use the app (due to the prevalence of iPhones in today's world). A third limitation one could describe would be that the app itself is somewhat "bare bones," meaning that it does work, run and perform its purpose, but it contains mainly the basic features required (create a conflict, view a conflict, edit profile, etc) without any "Quality of Life" upgrades [14].

Naturally, we would desire to resolve these limitations to create a better working app for all to use. One way we could resolve the first limitation discussed above is by getting more people to use the app for its intended purpose, resulting in more feedback about how well the app worked for people. One could expect to see more usership of the app after running ads for it on sites such as Facebook, Instagram, etc. Solving the android-only issue would be more difficult however, as it would require tinkering with and changing the code and the way the app runs in order to work well on Apple devices. Additionally, the Apple App Store is much more difficult to get an app published on than on the Google Play Store, presenting yet another hurdle to overcome for this limitation. Finally, a much more feasible correcting of a limitation that could be done would be to polish the appearance and features of the app up in order to make it aesthetically pleasing and easier for the user to interact with. This could be accomplished by doing more research into Flutter and how to make things look nicer or how to add new features that users might appreciate (such as connecting with friends who also have the app).

REFERENCES

[1] Vuchinich, Samuel. "Starting and stopping spontaneous family conflicts." Journal of Marriage and the Family (1987): 591-601.

[2] Leege, David C., et al. "The politics of cultural differences." The Politics of Cultural Differences. Princeton University Press, 2009.

[3] Burleson, Brant R., and Amanda J. Holmstrom. "Comforting communication." The international encyclopedia of communication (2008).

[4] Harker, Christopher, and Lauren L. Martin. "Familial relations: Spaces, subjects, and politics." Environment and Planning A 44.4 (2012): 768-775.

[5] Westjohn, Stanford A., Holger Roschk, and Peter Magnusson. "Eastern versus Western culture pricing strategy: Superstition, lucky numbers, and localization." Journal of International Marketing 25.1 (2017): 72-90.

[6] Boudon, Raymond. "Education, opportunity, and social inequality: Changing prospects in western society." (1974).

[7] Barnes, Howard L., and David H. Olson. "Parent-adolescent communication and the circumplex model." Child development (1985): 438-447.

[8] Holla, Suhas, and Mahima M. Katti. "Android based mobile application development and its security." International Journal of Computer Trends and Technology 3.3 (2012): 486-490.

[9] Kuzmin, Nikita, Konstantin Ignatiev, and Denis Grafov. "Experience of developing a mobile application using flutter." Information Science and Applications. Springer, Singapore, 2020. 571-575.

[10] Surijah, Edwin Adrianta, and Yashinta Levy Septiarly. "Construct validation of five love languages." Anima Indonesian Psychological Journal 31.2 (2016): 65-76.

[11] Prevoo, Mariëlle JL, and Catherine S. Tamis-LeMonda. "Parenting and globalization in western countries: explaining differences in parent–child interactions." Current opinion in psychology 15 (2017): 33-39.

[12] Smetana, Judith G., Jenny Yau, and Shirley Hanson. "Conflict resolution in families with adolescents." Journal of Research on Adolescence 1.2 (1991): 189-206.

[13] Liu, Kaifeng, Zhenzhen Xie, and Calvin Kalun Or. "Effectiveness of mobile app-assisted self-care interventions for improving patient outcomes in type 2 diabetes and/or hypertension: systematic review and meta- analysis of randomized controlled trials." JMIR mHealth and uHealth 8.8 (2020):

e15779.

[14] Omran, Mahamed GH, Andries P. Engelbrecht, and Ayed Salman. "Bare bones differential evolution." European Journal of Operational Research 196.1 (2009): 128-139.

[15] Shaffer, David Williamson, et al. "Video games and the future of learning." Phi delta kappan 87.2 (2005): 105-111.