Multiplatformska aplikacija za pomoć u planiranju

Krušlin, Antonio

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:198745

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-30**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

MULTIPLATFORMSKA APLIKACIJA ZA POMOĆ U PLANIRANJU

Završni rad

Antonio Krušlin

Osijek, 2020.

SADRŽAJ

1. Uvod 3
1.1. Zadatak završnog rada 3
2. Prijedlog rješenja 4
2.1. Opis postojećih rješenja 4
2.1.1. Planner Pro 4
2.1.2. Any.do 6
3. Tehnologije korištene u izradi rješenja 9
3.1. Visual Studio Code 9
3.2. Dart 10
3.3. Flutter 11
3.4. Firebase 12
3.5. Adobe XD 13
4. Izrada aplikacije 15
4.1. Dizajniranje aplikacije 15
4.2. Izrada projekta 16
4.3. Zasloni 16
4.3.1. Zaslon za prijavu korisnika 16
4.3.2. Početni zaslon 17
4.3.3. Izbornik 19
4.3.4. Zaslon To-Do liste 19
4.3.5. Donji list za dodavanje To-Do elementa 21
4.3.6. Zaslon kalendara 21
4.3.7. Zaslon za dodavanje događaja 23
4.4. Testiranje aplikacije 24
5. Funkcionalnosti 25
6. Prednosti i nedostaci u odnosu na postojeća rješenja 34
7. Zaključak 35
Literatura 36
Popis slika 37
Sažetak 39
Abstract 40

Životopis.....41
Prilozi 42

1. UVOD

Svakim danom svijet se sve više ubrzava i način života se mijenja, a na ljudima je zadatak da se prilagode. Zbog raznih obaveza teško je popratiti ubrzani ritam modernog života i zato je potrebno dobro organizirati vrijeme, kako se neki važni događaj ne bi propustio zbog nagomilavanja raznih informacija primljenih tijekom cijelog dana. Kako danas imamo više različitih operacijskih sustava, a svaki od njih se razvija u različitoj okolini, potrebno je naći rješenje kako lako i efikasno prebroditi jaz između različitih operacijskih sustava i programskih jezika. Jedno od takvih rješenja jest korištenje multiplatformskog sučelja. U ovome radu opisat će se postupak izrade aplikacije za organiziranje vremena gdje korisnik organizira svoje vrijeme dodajući ga u aplikaciju ili dodavanjem naputka odnosno zadaće unutar popisa zadataka odnosno „*to do*“ liste. Također, bit će opisan sam proces izrade aplikacije kao i korištene tehnologije unutar aplikacije. Aplikacija bit će napisana u programskom jeziku Dart, korištenjem Flutter razvojnog okvira i pokretat će se na iOS i Android operacijskim sustavima.

1.1. Zadatak završnog rada

Opisati funkcionalnosti aplikacije koja bi služila za planiranje događaja, organizaciju vremena i "to-do" liste. Opisati tehnologije koje će se koristiti prilikom izrade aplikacije s osvrtom na specifičnosti izrade multiplatformske aplikacije. Opisati proces izrade aplikacije.

2. PRIJEDLOG RJEŠENJA

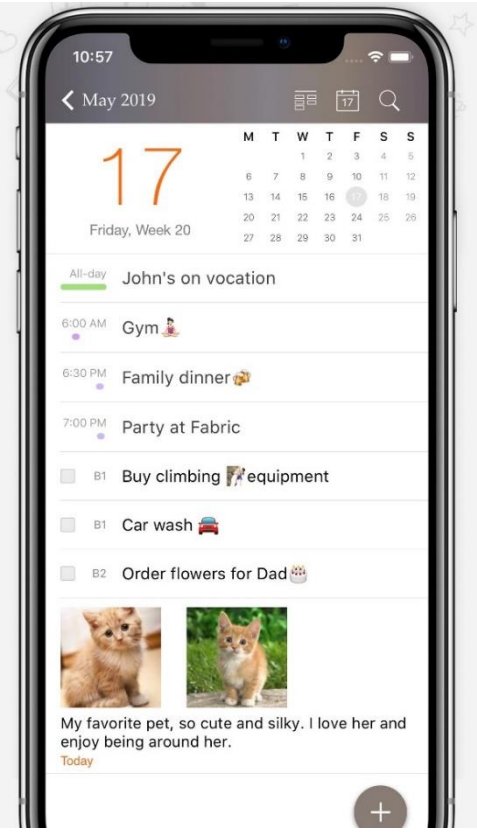
Aplikacija će se sastojati od pet zaslona gdje će se prvi zaslon sastojati od prijave korisnika u aplikaciju ili registracije korisnika ako već nema korisnički račun. Nakon toga slijedi naslovni zaslon na kojemu će biti prikazani događaji i zadaci za taj dan. Bit će moguće detaljno otvoriti kalendar kao poseban zaslon ili „*to do*“ zaslon. Kalendar prikazuje sve događaje koji se trebaju dogoditi ili su se dogodili u dnevnom, tjednom ili mjesečnom prikazu događaja. Moguće je obrisati događaje neovisno o danu, a ako se želi dodati novi događaj koristi se gumb za dodavanje. Pritiskom na tipku za dodavanje otvara se novi zaslon, gdje se upisuju svi potrebni podaci kako bi se događaj mogao uvesti u kalendar. „*to do*“ zaslon prikazuje sve zadatke, sortirajući ih prema starosti, tako da će noviji zadaci će biti prikazani na samome početku liste.

2.1. Opis postojećih rješenja

Postoje razna rješenja i aplikacije za planiranje i organiziranje vremena od kojih su neke bolje, a neka lošije napravljena. Za primjer su uzete su dvije aplikacije koje su prema količini skinutih inačica, najbolje rješenje za organizaciju vremena.

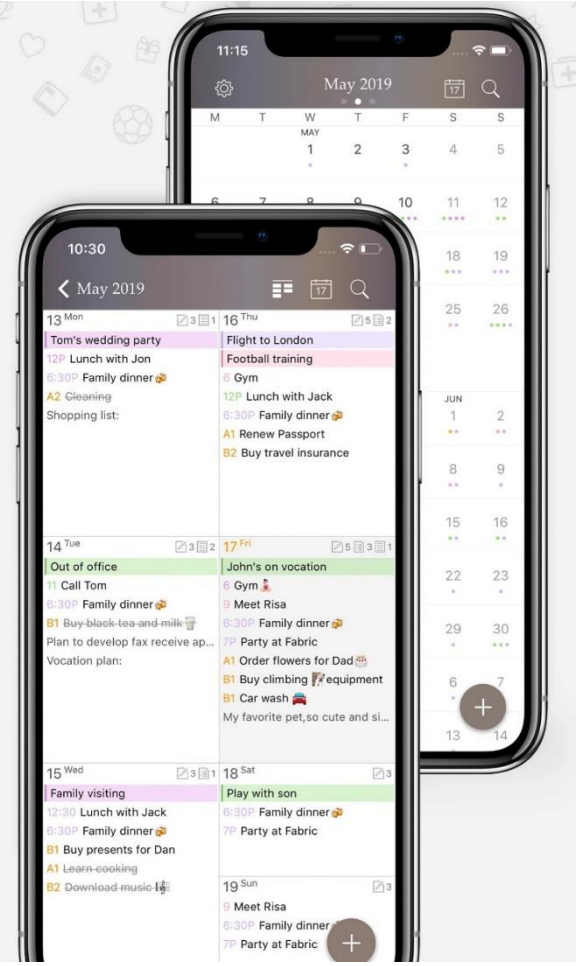
2.1.1. Planner Pro

Aplikacija „*Planner Pro*“ sastoji se od tri zaslona i također mogu se dodati događaji, zadaci i bilješke. Ovisno o tome je li riječ o besplatnoj ili plaćenoj verziji aplikacije, neke od mogućnosti su nedostupne za besplatne korisnike, a moguće ih je uključiti mjesečnom pretplatom napredne verzije aplikacije. Slika 2.1 prikazuje sve zabilježene događaje, zadatke i bilješke s dnevnim prikazom obaveza.



Slika 2.1 Planner Pro dnevni pregled, izvor: [1]

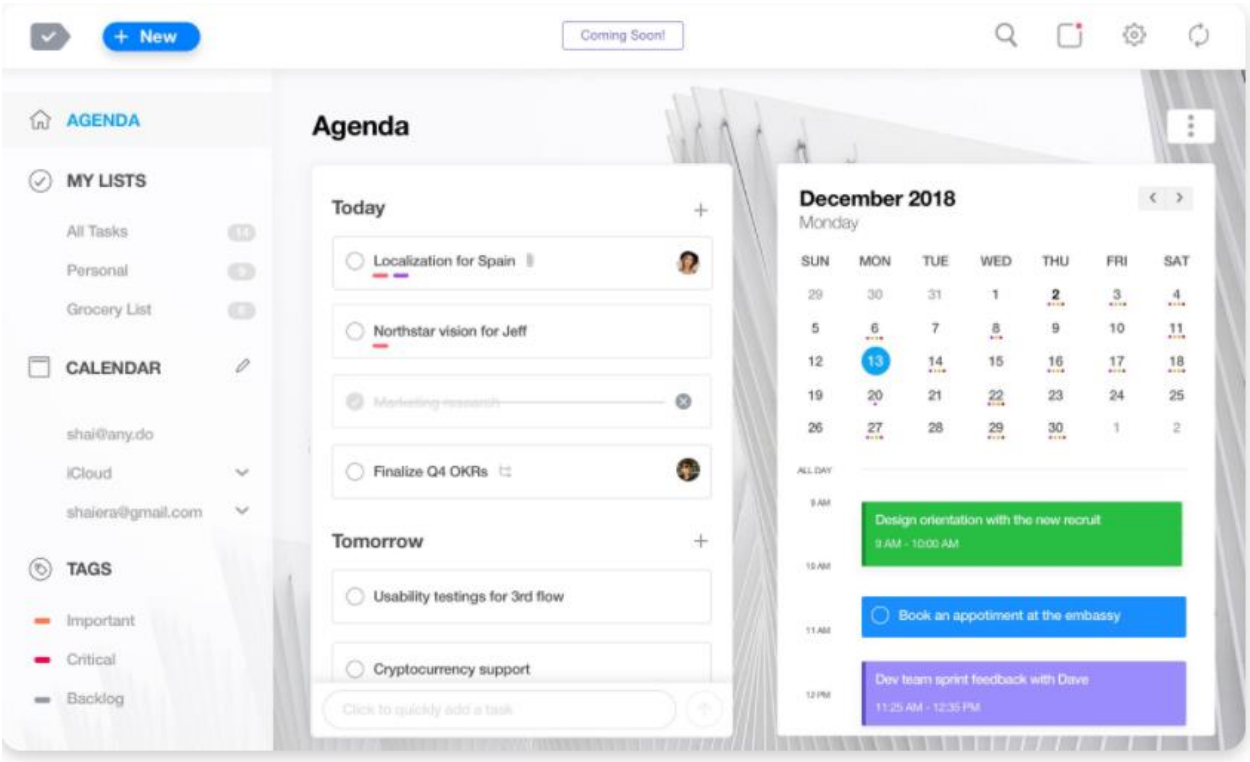
Slika 2.2 prikazuje dodatnu mogućnost zvanu tjedni pregled koji je moguće dobiti jedino ako se korisnik pretplati. Njime vidimo sve događaje, bilješke i zadatke u odabranom tjednu.



Slika 2.2 Planner Pro tjedni prikaz, izvor: [1]

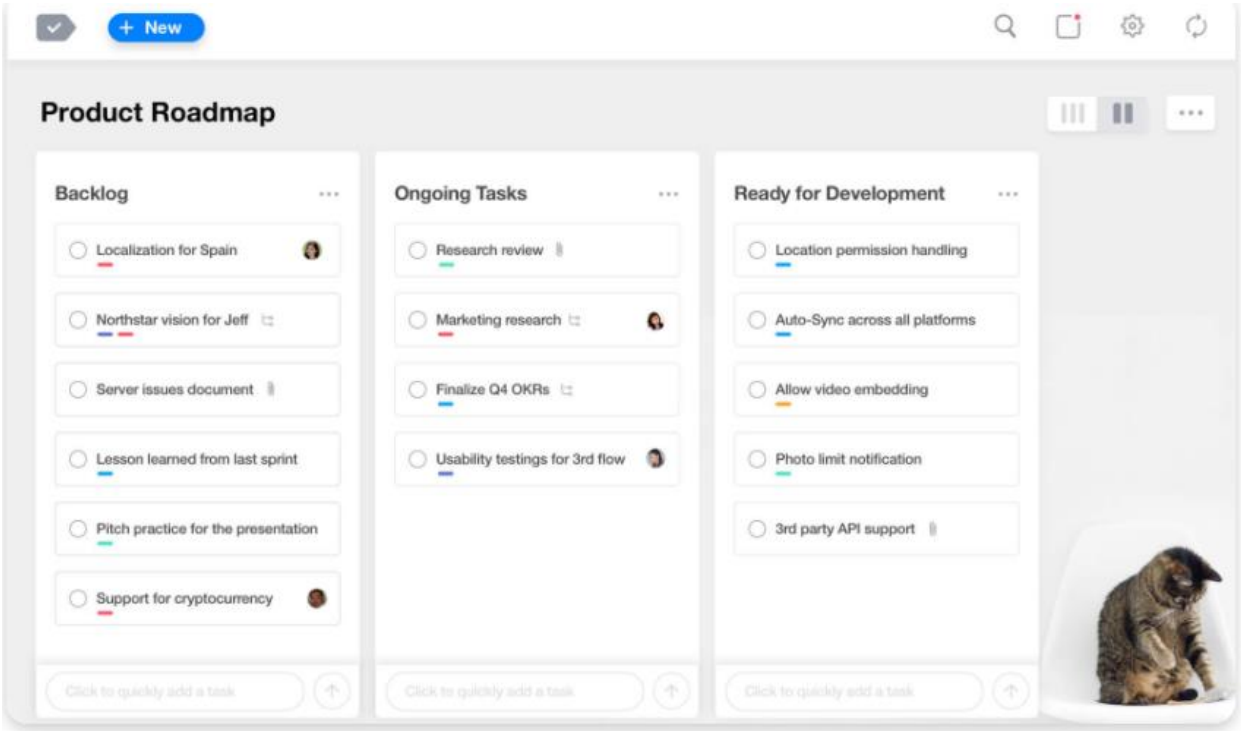
2.1.2. Any.do

Aplikacija „Any.do“ je napravljena kako bi pomogla korisnicima u planiranju vremena. Postoje razni načini dodavanja zadataka i događaja kao što se vidi na slici 2.3 . Aplikacija također omogućava formiranje grupa za lakše upravljanje projektima i raspodjele odgovornosti u timu. U potpunosti se može sinkronizirati s Google kalendarom, dostupna je na iOS i Android platformama, ali i internetskim pretraživačima i pametnim satovima.



Slika 2.3 Any.do aplikacija [2]

Slike 2.3 prikazuje kako aplikacija omogućava korisniku razne mogućnosti organiziranja vremena, pritom pazeći na dizajn kako bi što ljepše i jednostavnije prikazala unesene podatke. Slika 2.4 prikazuje grupiranje zadataka na različite načine. Također je moguće pratiti razvoj zadatka kao što je prikazano na Slici 2.4, gdje se odmah ispod imena zadatka mogu se vidjeti obojene crtice koje govore o važnosti ili o postotku riješenosti zadatka.



Slika 2.4 Any.do podjela zadataka [2]

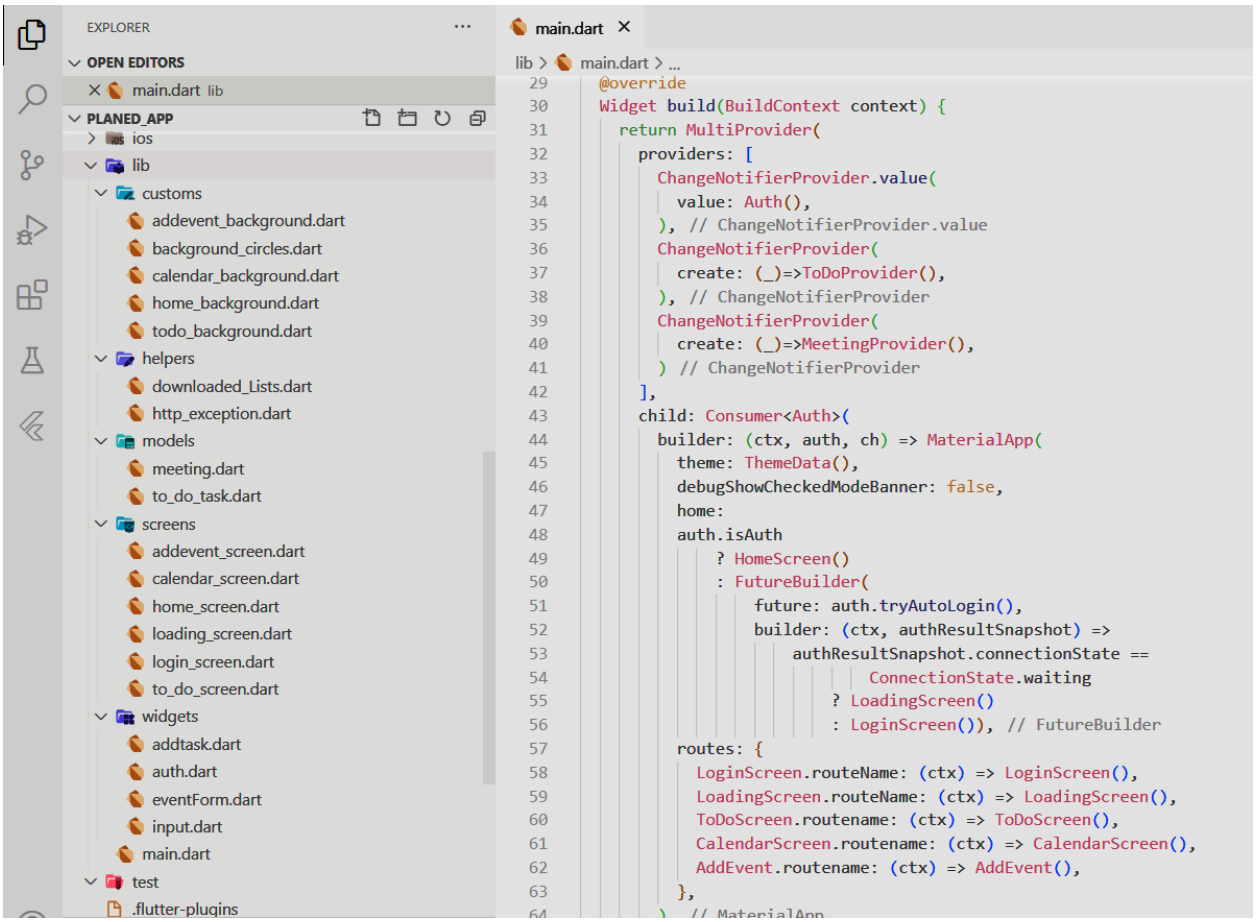
Postoji i nadograđena odnosno plaćena verzija aplikacije koja korisniku daje na raspolaganje još više usluga i mogućnosti pri upravljanju i organiziranju vremena.

3. TEHNOLOGIJE KORIŠTENE U IZRADI RJEŠENJA

Multiplatformska aplikacija zvana „*Planned*“ bit će izrađena u programu zvanom Visual Studio Code, a izvorni kod bit će napisan pomoću programskog jezika Dart. Razvojni okvir Flutter pretvorit će izvorni kod u aplikaciju koja se može koristiti na iOS i Android operacijskim sustavima. Povezivanje aplikacije s bazom podataka će se odvijati preko Firebase platforme kako bi podaci uvijek bili dostupni korisniku, neovisno o korištenom uređaju. Dizajn aplikacije bit će napravljen u besplatnom alatu Adobe XD.

3.1. Visual Studio Code

Visual Studio Code je program otvorenog koda, napravljen od tvrtke Microsoft za tri operacijska sustava: Windows, Linux i macOS. [3] Koristi se prilikom pisanja i uređivanja izvornoga koda. Program olakšava otklanjanje pogrešaka unutar koda, označava kod kako bi se lakše raspoznavali objekte, funkcije itd. Također ima mogućnost automatskog dovršetka riječi te jednostavnog refaktoriranja koda. Podržava pisanje u raznim programskim jezicima poput Jave, JavaScript, C, C++, C#, Python, Ruby. [4] Slika 3.1 prikazuje kako izgleda pokrenuti Visual Studio Code gdje se može primijetiti automatsko dovršavanje riječi zatim različite boje za uvezene biblioteke i varijable.



Slika 3.1 Visual Studio Code

3.2. Dart

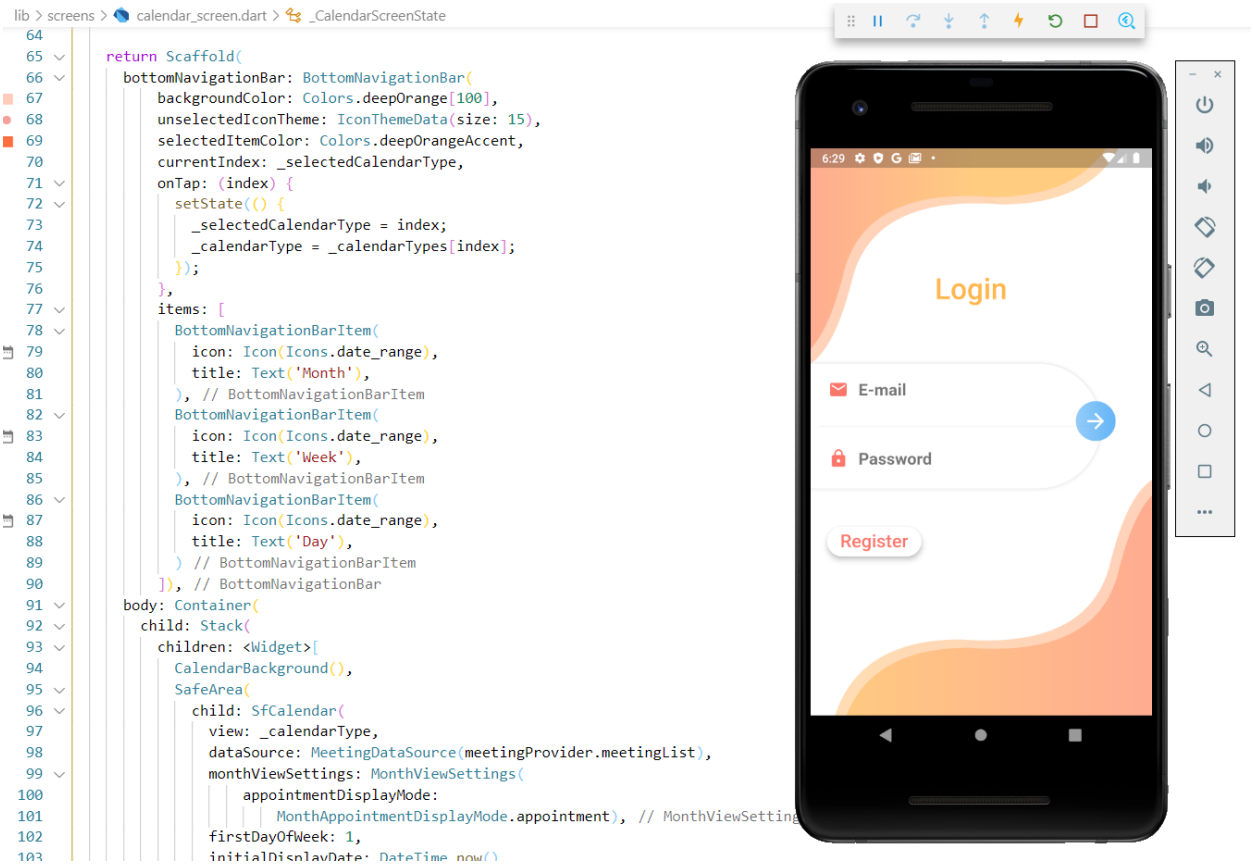
Dart je objektno orijentirani programski jezik sličan C-u. Razvija ga kompanija Google te se pretežito koristi za razvoj mobilnih aplikacija, ali podržava web i desktop aplikacije. Slika 3.2 prikazuje kako se pišu varijable, funkcije i klase da se pokaže sličnost C-u. [5] Za razliku od C-a, Dart ima sakupljanje smeća (engl. *garbage collection*) tako da programer ne mora brinuti o oslobađanju memorije.

```
1
2   //Varijable
3   var name = 'Voyager I';
4   var year = 1977;
5   var flybyObjects = ['Jupiter', 'Saturn', 'Uranus', 'Neptune'];
6   var image = {
7     'tags': ['saturn'],
8     'url': '//path/to/saturn.jpg'
9   };
10
11  //Funkcije
12  int fibonacci(int n) {
13    if (n == 0 || n == 1) return n;
14    return fibonacci(n - 1) + fibonacci(n - 2);
15  }
16
17  var result = fibonacci(20);
18
19  //Klase
20  class Piloted {
21    int astronauts = 1;
22    void describeCrew() {
23      print('Number of astronauts: $astronauts');
24    }
25  }
26
```

Slika 3.2 Dart kod

3.3. Flutter

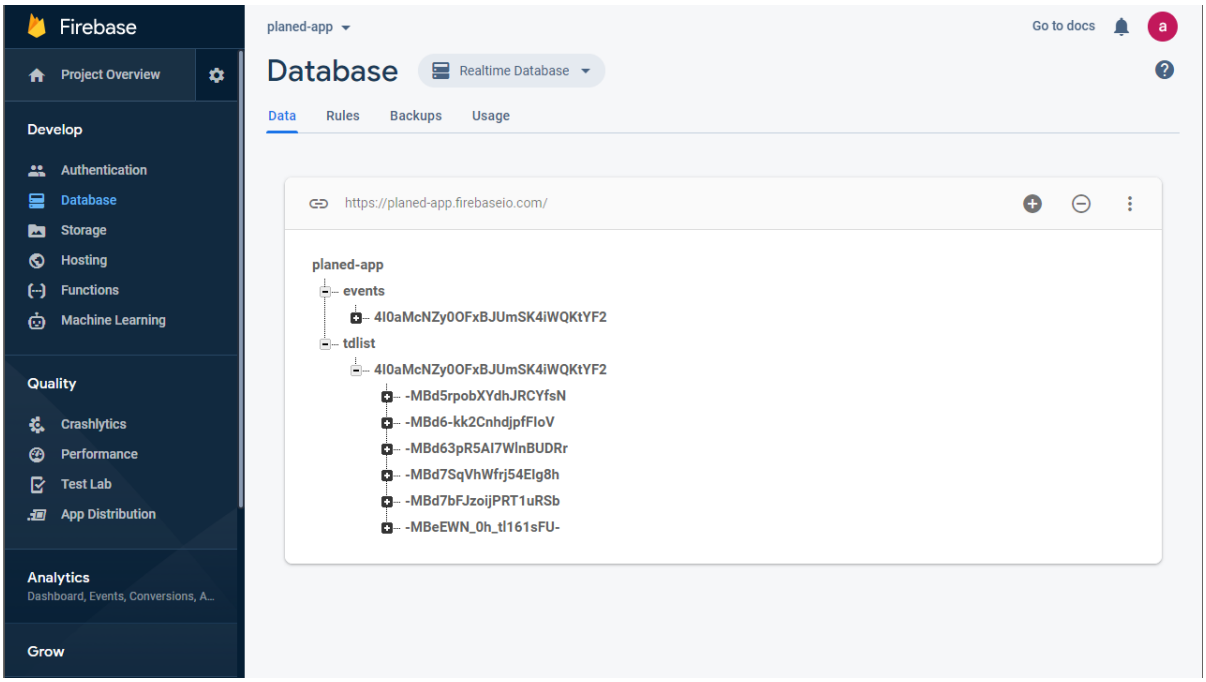
Flutter je relativno novi razvojni okvir kojeg razvija Google od 2017. godine. On je softver otvorenog koda tako da svatko može doprinijeti njegovom razvoju. Koristi se za razvoj aplikacija na različitim platformama poput Android, iOS, Windows, Mac, Linux, Web i Google Fuchsia. Flutter koristi dinamičko prevođenje (engl. *just-in-time compile*) koje omogućava korisniku da vidi sve upisane promjene bez ponovnog pokretanja aplikacije. Tako se sve promjene koje su upisane odmah prikazuju na simulatoru. Slika 3.3 prikazuje pokrenutu simulaciju aplikacije na Androidu, pritiskom na gumb u obliku munje, koji se nalazi u gornjem desnom kutu slike, osvježava aplikaciju i prikazuje sve dodane promjene, bez potrebe za ponovnim pokretanjem simulacije. Veliki broj biblioteka za korištenje olakšava korisniku programiranje jer može iskoristiti već napravljene *widgete*. Takozvanim ponovnim pokretanjem (engl. *Hot reload*) Flutter omogućava programeru da odmah vidi unesene promjene prilikom simulacije aplikacije. [6]



Slika 3.3 Flutter Android simulacija

3.4. Firebase

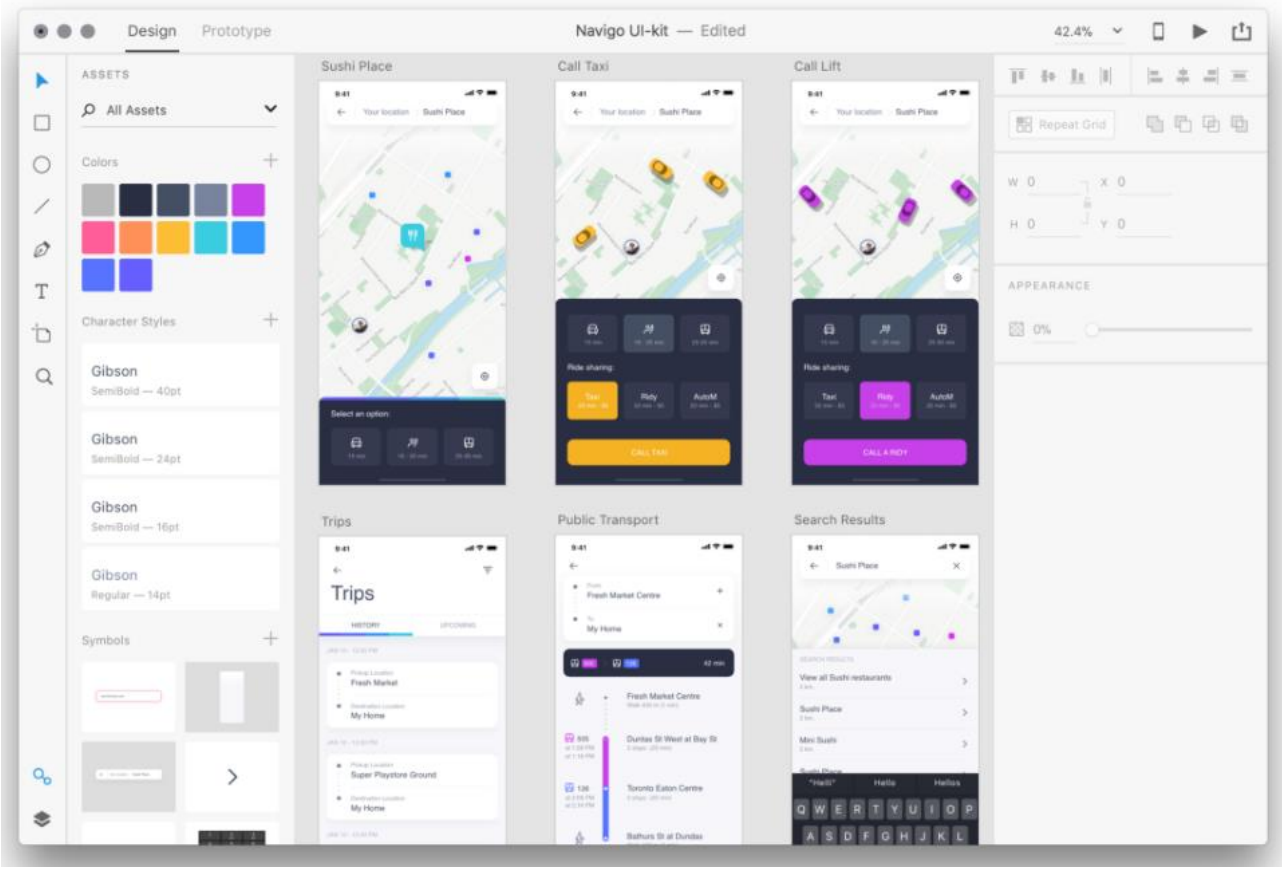
Firebase je razvojna platforma za mobilne i web aplikacije, koja nudi razne servise poput analitike podataka o korištenju aplikacije, autentifikacije korisnika prilikom ulaska u aplikaciju, baze podataka u stvarnom vremenu i razne druge. Slika 3.4 prikazuje bazu podataka u stvarnom vremenu koja sadržava sve događaje korisnika. Firebase nudi razne pogodnosti bez ikakve potrebe za plaćanjem servera ili baze podataka. [7] Naravno da su korisniku besplatne usluge uskraćene neke mogućnosti, ali u osnovnom modelu usluge programeru je dostupno sve što je potrebno.



Slika 3.4 Firebase baza podataka

3.5. Adobe XD

Adobe XD je dizajnerski alat kojeg je razvila Adobe Inc koji ima besplatni početnički paket za upotrebu. [8] Kao što se može vidjeti na slici 3.5, glavni zadatak ovog alata je dizajniranje mobilnih i web aplikacija kako bi grafički dizajneri što bolje predočili programerima izgled aplikacije. Postoje razni dodaci za ovaj alat kako bi pomogli korisnicima lakše korištenje alata i oblikovanje dizajna aplikacije. [9]



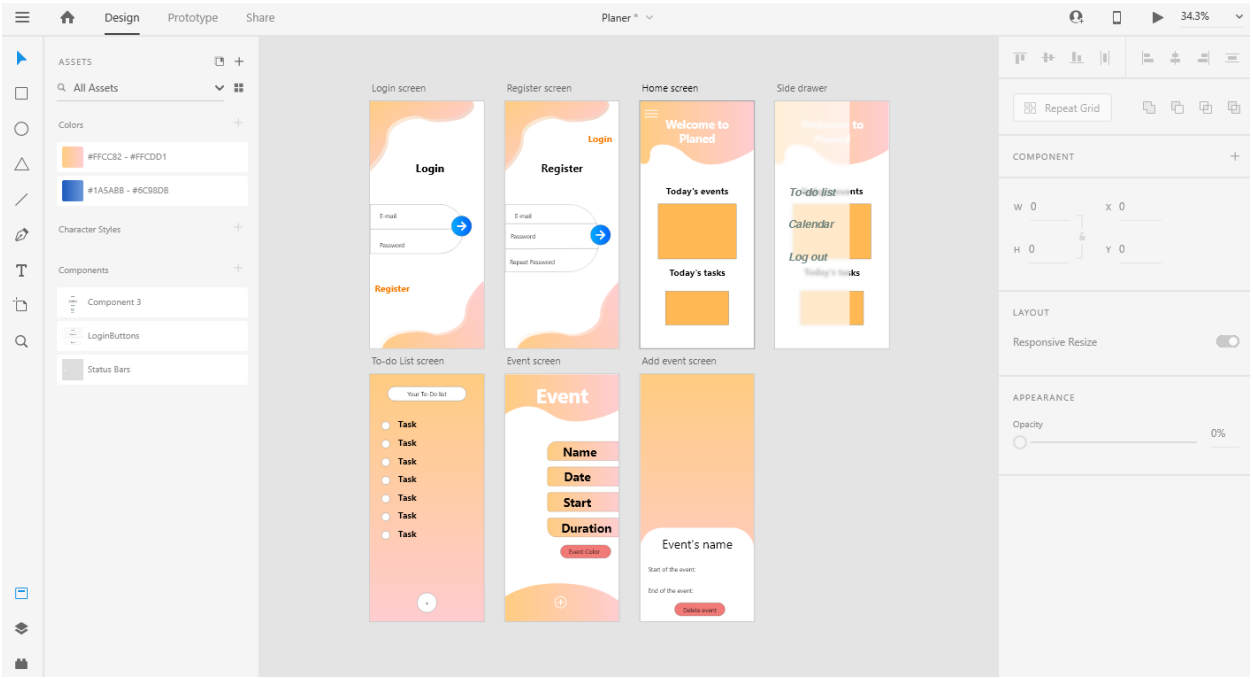
Slika 3.5 Adobe XD alat, izvor: [10]

4. IZRADA APLIKACIJE

U ovome poglavlju bit će ukratko opisan način i slijed izrade aplikacije. Na početku mora postojati ideja koja može nastati prilikom razmišljanja, promatranja potreba društva ili zadatak koji je zadan od strane voditelja ili mentora.

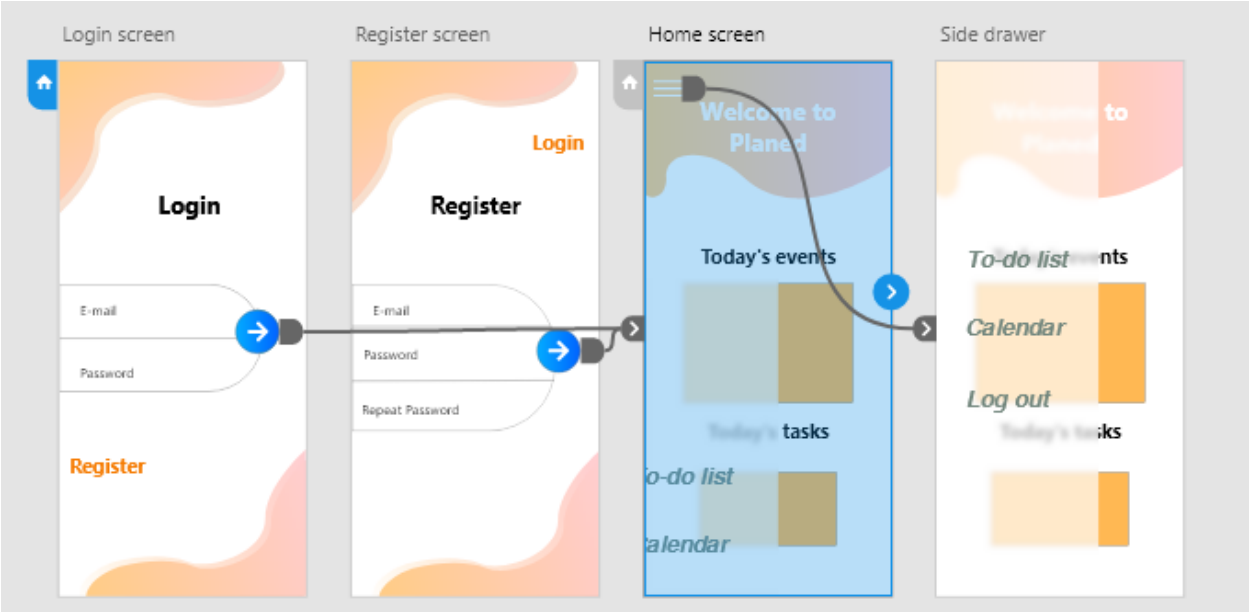
4.1. Dizajniranje aplikacije

Pregledavanjem različitih web stranica i aplikacija lako se može pronaći željeni dizajn aplikacije. Danas mnogi dizajneri javno objavljuju svoje radove, kako bi pokazali svoje znanje u izradi lijepih i oku zanimljivih dizajna s ciljem da netko prepozna njihov talent i iskustvo te zaposli ih. Nakon što je ideja razrađena i odobrena, započinje se s radom. Prvo na redu je dizajniranje izgleda aplikacije u programu Adobe XD. Svaki od zaslona treba biti izrađen prije početka programiranja. Dizajnirani zaslon prikazuje kako bi završni proizvod trebao izgledati.



Slika 4.1 Dizajn aplikacije

Na slici 4.1 nalazi se gotov dizajn aplikacije koju tek treba izraditi i napisati u nekom programskom jeziku. U Adobe XD programu moguće je i izraditi prototip navigacije [11] kako bi se programeru bolje pojasnilo što će koji gumb raditi što se može vidjeti na slici 4.2. gdje sive linije pokazuju mjesto gdje će pritisak gumba odvesti korisnika. Također je moguća simulacija dizajna i navigacije.



Slika 4.2 Povezivanje navigacije

4.2. Izrada projekta

Projektna mapa mora biti izrađena pomoću naredbe „`flutter create ime_aplikacije`“ koja kreira datoteku sa svim potrebnim dodacima za izradu multiplatformske aplikacije. Nakon što je datoteka kreirana, može se početi s programiranjem aplikacije.

4.3. Zasloni

U ovome odjeljku bit će prikazani svih zasloni i njihove glavne funkcije. Pozadinska slika napravljena je kao jedinstven zaslon, tako da se može pozivati zasebno.

4.3.1. Zaslon za prijavu korisnika

Zaslon za prijavu korisnika prvi se prikaže prilikom otvaranja aplikacije kao na slici 5.1. Najbitniji dio toga zaslona je autentifikacija korisnika na server kao što se može vidjeti na slici 4.3 gdje se nalazi dio koda koji komunicira sa serverom, aplikacija šalje serveru elektronsku adresu, lozinku te zahtjev da mu server vrati token koji će korisniku odobriti korištenje aplikacije.

```
Future<void> _authenticate(
    String email, String password, String urlSegment) async {
  try {
    final response = await http.post(
      urlSegment,
      body: json.encode(
        {
          'email': email,
          'password': password,
          'returnSecureToken': true,
        },
      ),
    );
    final responseData = json.decode(response.body);
    if (responseData['error'] != null) {
      throw HttpException(responseData['error']['message']);
    }
    _token = responseData['idToken'];
    _userId = responseData['localId'];
    _expiryDate = DateTime.now().add(
      Duration(
        seconds: int.parse(
          responseData['expiresIn'],
        ),
      ),
    );
    _autoLogout();
    notifyListeners();
    final prefs = await SharedPreferences.getInstance();
    final userData = json.encode({
      'token': _token,
      'userId': _userId,
      'expiryDate': _expiryDate.toIso8601String(),
    });
    prefs.setString('userData', userData);
  } catch (error) {
    throw error;
  }
}
```

Slika 4.3 Dio koda autentikacije

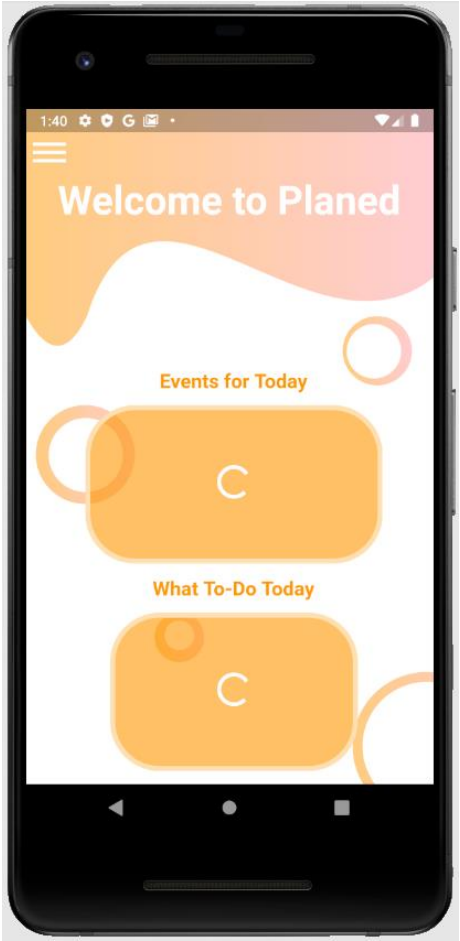
Trajanje tokena je jedan sat, a ako korisniku istekne token, korisnik se vraća na zaslon za prijavu.

4.3.2. Početni zaslon

Početni zaslon sastoji se od dva pokretna popisa gdje jedan prikazuje dnevne događaja a drugi prikazuje dnevne zadatke. Na slici 4.4 nalazi se dio koda koji je zaslužan za prikazivanje svih dnevnih zadataka korisnika, prilikom učitavanja prikazuje se kružna animacija učitavanja kao što se može vidjeti na slici 4.5. Animacija učitavanja je dodana zbog toga kako bi aplikacija imala dovoljno vremena skinuti podatke potrebne za ispis i pohraniti ih na uređaj. Nakon što su podaci dostupni oni se prikazuju samo ako im je datum određen za tadašnji dan.

```
child: FutureBuilder(
  future: Future.delayed(Duration(seconds: 1)),
  builder: (context, snapshot) =>
      snapshot.connectionState ==
          ConnectionState.waiting
        ? Center(
          child: Container(
            width: 30,
            height: 30,
            child: CircularProgressIndicator(
              valueColor:
                  AlwaysStoppedAnimation<Color>(
                      Colors.white), // AlwaysStoppedAnimation
            ), // CircularProgressIndicator
          ), // Container
        ) // Center
        : ListView.builder(
          itemCount:
              toDoProvider.toDoListTodayLenght,
          itemBuilder: (ctx, i) => ListTile(
            title: Center(
              child: Text(toDoProvider
                  .toDoListToday[i].taskName), // Text
            ), // Center
          ), // ListTile
        ), // ListView.builder
), // FutureBuilder
```

Slika 4.4 To-Do dnevna lista zadataka



Slika 4.5 Učitavanje dnevnih zadataka,događaja

4.3.3. Izbornik

Zaslon izbornika služi za navigaciju između zaslona kalendara, to-do liste i odjave iz aplikacije. Slika 4.6 prikazuje kako se dekorira izgled gumba zatim navigaciju koja vodi korisnika na novi zaslon te se prilikom pritiska na gumb izbornik zatvara.

```
Container(
  width: 175,
  height: 55,
  decoration: BoxDecoration(
    color: Colors.white,
    shape: BoxShape.rectangle,
    borderRadius: BorderRadius.all(
      Radius.circular(50),
    ), // BorderRadius.all
  ), // BoxDecoration
  child: FlatButton(
    onPressed: () {
      _scaffoldKey.currentState.openEndDrawer();
      Navigator.of(context).pushNamed(ToDoScreen.routename);
    },
    child: Text('To Do'),
  ), // FlatButton
), // Container
```

Slika 4.6 Izbornik To-Do

4.3.4. Zaslon To-Do liste

Prilikom otvaranja zaslona dolazi do učitavanja svih zadataka upisanih u bazu podataka. Ispis podataka napravljen je pomoću pokretne liste u kojoj su svi elementi kartice. Prilikom brisanja elementa korisnik mora povući karticu u lijevo te potvrditi akciju ako želi obrisati odabranu stavku.

```
direction: DismissDirection.endToStart,
confirmDismiss: (direction) => showDialog(
  context: context,
  child: AlertDialog(
    backgroundColor: Colors.white,
    title: const Text("Confirm"),
    content: const Text(
        "Are you sure you wish to delete this task?"), // Text
    actions: <Widget>[
      FlatButton(
          onPressed: () =>
              Navigator.of(context)
                  .pop(true),
          child: const Text("DELETE")), // FlatButton
      FlatButton(
        onPressed: () =>
            Navigator.of(context)
                .pop(false),
        child: const Text("CANCEL"),
      ), // FlatButton
    ], // <Widget>[]
  ), // AlertDialog
),
```

Slika 4.7 Brisanje To-Do elementa s liste

Na slici 4.7 može se vidjeti kako pomak elementa lijevo poziva na izvršavanje dijalog za upozorenje koji šalje upit korisniku. Također prilikom brisanja element se mora obrisati ne samo u na uređaju nego i na serveru.

```
Future<void> deleteTask(int taskIndex) async {
  final sharedData = await SharedPreferences.getInstance();

  var userData =
      json.decode(sharedData.getString('userData')) as Map<String, Object>;
  var id = userData['userId'];
  var token = userData['token'];
  var url = 'https://planed-app.firebaseio.com/tdlist/$id.json?auth=$token';

  try {
    final response = await http.get(url);
    final downloadedTasks =
        json.decode(response.body) as Map<String, dynamic>;
    var delTaskList = downloadedTasks.keys.toList();

    String delTask = delTaskList[taskIndex];
    url = 'https://planed-app.firebaseio.com/tdlist/$id/$delTask.json';

    final deleteResponse = await http.delete(url);
  } catch (error) {
    throw error;
  }
}
```

Slika 4.8 Brisanje To-Do elementa sa serverske strane

Da bi se element obrisao i na serveru mora se poslati zahtjev za brisanje kao što je prikazano na slici 4.8. Zahtjev koji će sadržavati ključ odnosno šifrirano ime toga elementa.

4.3.5. Donji list za dodavanje To-Do elementa

Dok se korisnik nalazi na To-Do zaslonu pritiskom na gumb za dodavanje otvara se novi donji list prozora u kojem korisnik ispunjava zadane parametre. Na slici 4.9 funkcija za dodavanje zadatka prima klasu tipa ToDoTask, iz koje se onda iščitavaju potrebni podaci i zapisuju u mapu koja će biti poslana serveru da ju obradi i spremi novi element u bazu podataka.

```
Future<void> uploadTask(ToDoTask newTD) async {
  final sharedData = await SharedPreferences.getInstance();
  var userData =
      json.decode(sharedData.getString('userData')) as Map<String, Object>;
  var id = userData['userId'];
  var token = userData['token'];

  var url = 'https://planed-app.firebaseio.com/tdlist/$id.json?auth=$token';
  try {
    final response = await http.post(
      url,
      body: json.encode({'task': newTD.taskName, 'date': newTD.taskDate}),
    );
    setState(() {
      toDoProvider.addTask(newTD);
    });
  } catch (error) {
    throw error;
  }
}
```

Slika 4.9 Dodavanje To-Do elementa na server

4.3.6. Zaslon kalendara

Kalendar je jedan veliki dodatak koji je slobodan za korištenje, ali da bi se kvalitetno iskoristio potrebno je temeljito istražiti koje sve opcije nudi. Radi boljeg korištenja dodatka je bitno dobro poznavati parametre koje prima taj dodatak i koje opcije nudi.

```
child: SfCalendar(
  view: _calendarType,
  dataSource: MeetingDataSource(meetingProvider.meetingList),
  monthViewSettings: MonthViewSettings(
      appointmentDisplayMode:
          MonthAppointmentDisplayMode.appointment), // MonthViewSettings
  firstDayOfWeek: 1,
  initialDisplayDate: DateTime.now(),
  initialSelectedDate: DateTime.now(),
  todayHighlightColor: Colors.redAccent[200],
  headerHeight: 50,
  headerStyle: CalendarHeaderStyle(
    textAlign: TextAlign.center,
    textStyle: TextStyle(
      fontSize: 40,
      color: Colors.black54,
    ), // TextStyle
  ), // CalendarHeaderStyle
  appointmentTextStyle: TextStyle(
    fontSize: 25,
  ), // TextStyle
  selectionDecoration: BoxDecoration(
    color: Colors.transparent,
    border: Border.all(color: Colors.amber, width: 2),
    borderRadius: const BorderRadius.all(Radius.circular(4)),
    shape: BoxShape.rectangle,
  ), // BoxDecoration
  timeSlotViewSettings: TimeSlotViewSettings(
    timeFormat: 'HH:mm',
    timeInterval: Duration(minutes: 30),
    dayFormat: 'EEE',
    timeRulerSize: 50,
    timeTextStyle: TextStyle(
      fontSize: 14,
      color: Colors.black,
    ), // TextStyle
  ), // TimeSlotViewSettings
```

Slika 4.10 Opcije kalendara

Na slici 4.10 mogu se vidjeti razne opcije za prikazivanje vremena, načina na koji se oblikuje prikaz vremena, odabir prvog dana u tjednu te osnovni prikaz događaja. Traka za odabir prikaza događaja nalazi se na samome dnu zaslona. Slika 4.11 prikazuje kako se postavlja donja navigacijska traka te kako se određuje boja i veličina označenog i neoznačenog elementa.

```
bottomNavigationBar: BottomNavigationBar(
    backgroundColor: Colors.deepOrange[100],
    unselectedIconTheme: IconThemeData(size: 15),
    selectedItemColor: Colors.deepOrangeAccent,
    currentIndex: _selectedCalendarType,
    onTap: (index) {
      setState(() {
        _selectedCalendarType = index;
        _calendarType = _calendarTypes[index];
      });
    },
    items: [
      BottomNavigationBarItem(
        icon: Icon(Icons.date_range),
        title: Text('Month'),
      ), // BottomNavigationBarItem
      BottomNavigationBarItem(
        icon: Icon(Icons.date_range),
        title: Text('Week'),
      ), // BottomNavigationBarItem
      BottomNavigationBarItem(
        icon: Icon(Icons.date_range),
        title: Text('Day'),
      ) // BottomNavigationBarItem
    ]), // BottomNavigationBar
```

Slika 4.11 Prikaz kalendara

4.3.7. Zaslon za dodavanje događaja

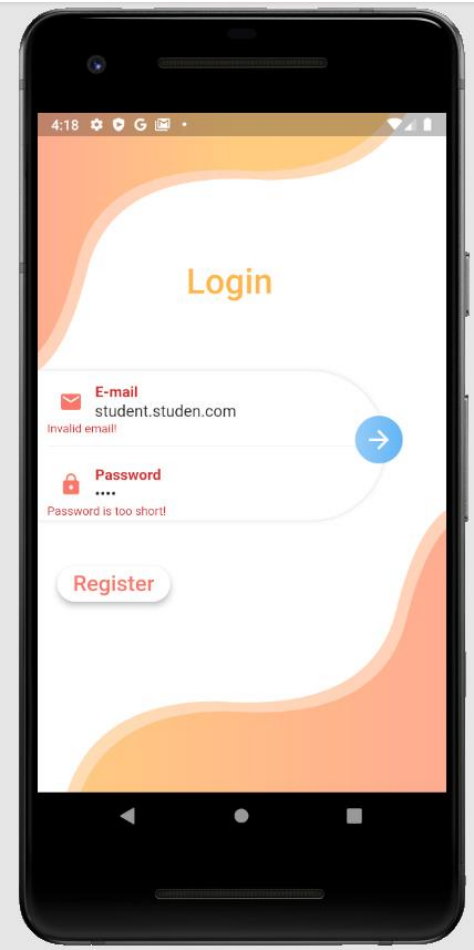
Sastoji se od 4 različita polja za popunjavanje, prvi od njih je tekstualni zatim slijede datum,vremenski odabir te odabir boje. Kao što se vidi na slici 4.12 tekstualno polje se provjerava tako da polje ne smije ostati prazno. Ako korisnik ne upiše ime događaja, a pokuša dodati događaj ispisat će mu se poruka da unese ime događaja.

```
child: TextFormField(
  decoration: InputDecoration(
    errorStyle: TextStyle(height: 6),
    labelText: 'Event name',
    labelStyle: TextStyle(fontSize: 20),
  ), // InputDecoration
  onChanged: (value) {
    _eventName = value;
  },
  onSaved: (value) {
    _eventName = value;
  },
  validator: (value) {
    if (value.isEmpty || value == '') {
      return 'Enter Event name.';
    }
  },
), // TextFormField
```

Slika 4.12 Dodavanje ime događaja

4.4. Testiranje aplikacije

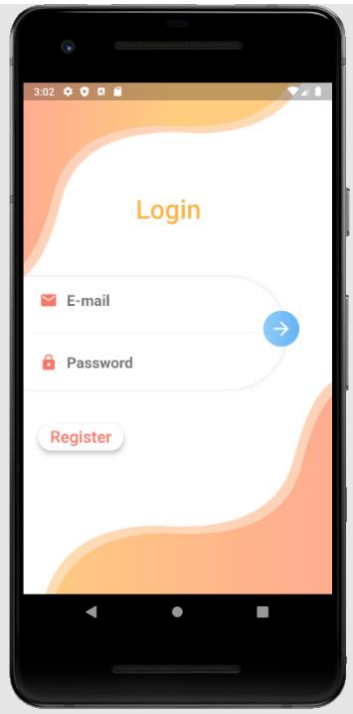
Nakon što je pisanje aplikacije završeno potrebno ju je testirati na moguće pogreške koje se ne uočavaju tako lako dok se ne isprobaju razne kombinacije krivih unosa. Kako bi se utvrdila stabilnost i učinkovitost potrebno je temeljito testirati aplikaciju. Slika 4.13 prikazuje jednostavno testiranje prilikom prijave korisnika, gdje korisnik nije upisao valjanu adresu ni dovoljno dugačku lozinku.



Slika 4.13 Testiranje prijave korisnika

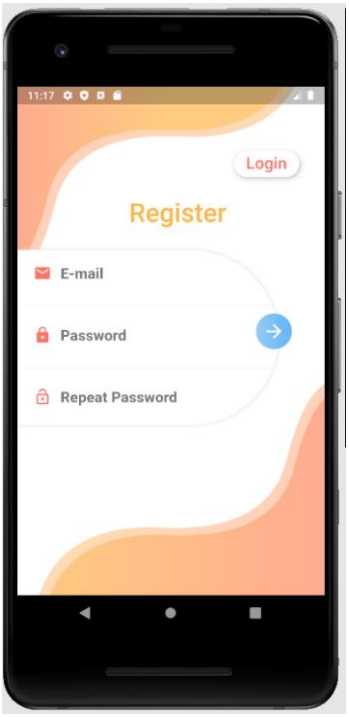
5. FUNKCIONALNOSTI

Sve slike aplikacije i kodovi koje se nalaze u ovom poglavlju preuzete su iz izrađene aplikacije. Aplikacija je koncipirana na način kada ju korisnik prvo uključi prikaže mu se zaslon kao na slici 5.1 gdje se korisnik može se prijaviti pritiskom na polja za unos elektroničke pošte i lozinke te na kraju pritiskom na strelicu kako bi dovršio prijavu. Ako dođe do pogreške zbog netočne lozinke, aplikacija prikazuje skočni prozor (engl. *pop-up window*) na kojemu piše zbog čega je došlo do problema prilikom autentifikacije.



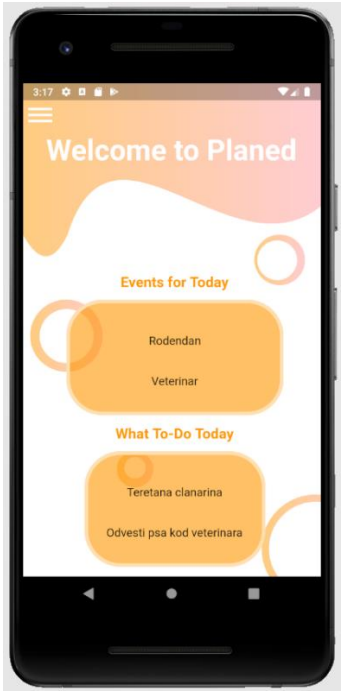
Slika 5.1 Zaslon za prijavu korisnika

Ako korisnik nije registriran, pritiskom na gumb za registraciju korisnika, zaslon će se promijeniti kao na slici 5.2. gdje se vidi povećan broj polja za unos u odnosu na sliku 5.1, kako bi se korisnik registrirao potrebna mu je valjana elektronička adresa zatim lozinka od najmanje šest znakova koju korisnik mora potvrditi ponovnim unosom. Ako dođe do krivog upisa u polje (npr. prekratka lozinka), polje za unos će ispisati razlog zbog kojeg zahtjev odbijen. U slučaju navedenom ranije lozinka koju je korisnik upisao bila je prekratka.



Slika 5.2 Zaslon za registraciju korisnika

Nakon prijave ili registracije korisnik dolazi na naslovnu stranicu ili zaslon, gdje će biti prikazani događaji i zadaci za tadašnji dan.

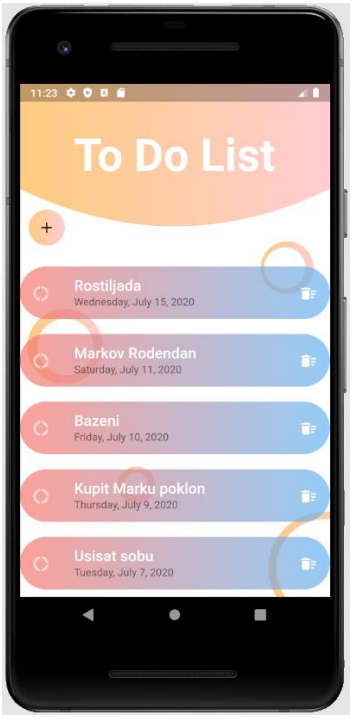


Slika 5.3 Naslovni zaslon

Pritiskom na gumb u gornjem lijevom kutu korisniku se otvara pretinac na zaslonu gdje se nalaze tri gumba, od koji svaki izvršava određenu zadaću te vodi korisnika na novi zaslon.

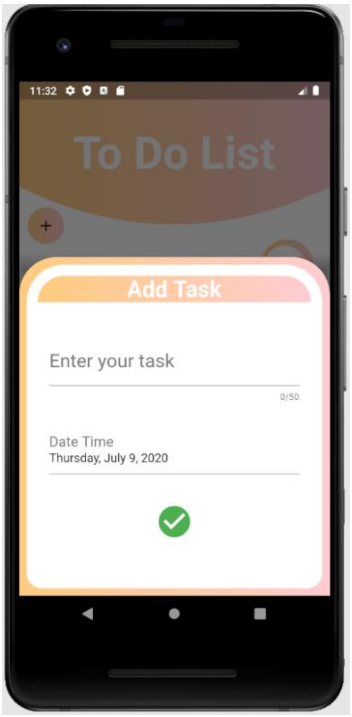
Slika 5.4 Otvoreni pretinac

Pritiskom na gumb „To Do“ korisniku se otvara novi prozor kao na slici 5.5. Svi dotad upisani zadaci sortirani su po datumu kao što se može vidjeti na slici 5.5. Ako postoji više zadataka, a nije ih sve moguće prikazati na zaslonu, popis zadataka postat će pomičan gdje se laganim pomakom prsta mogu prolistati svi upisani zadaci.



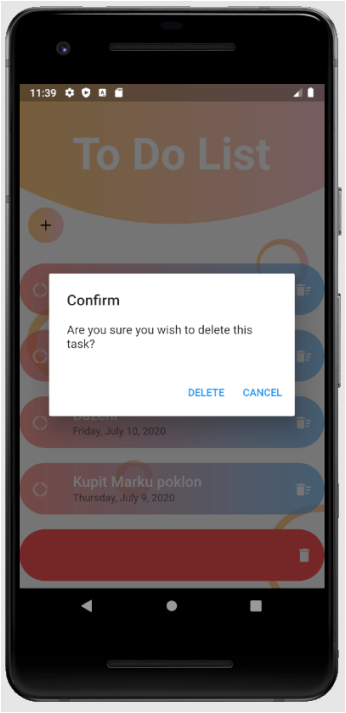
Slika 5.5 Zaslon "To do" liste

Kako bi se dodao novi zadatak korisnik mora pritisnuti okrugli gumb s plusom u sredini koji će otvoriti donji list prozora kao na slici 5.6. na kojemu je moguće upisati ime zadatka u odgovarajuće polje i izabrati datum te pritiskom na zelenu kvačicu potvrditi zadatak koji se sprema u bazu podataka.



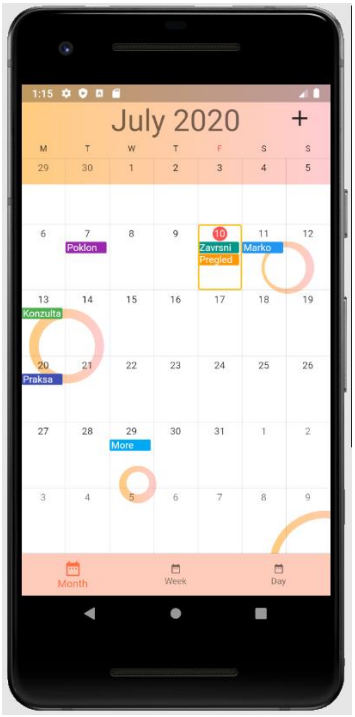
Slika 5.6 Dodavanje zadatka

Ako je korisnik izvršio zadatak te mu više nije potreban moguće ga je obrisati povlačenjem kartice na kojoj se zadatak nalazi u lijevo što će rezultirati upitom je li korisnik siguran da želi obrisati zadanu stavku kao na slici 5.7.



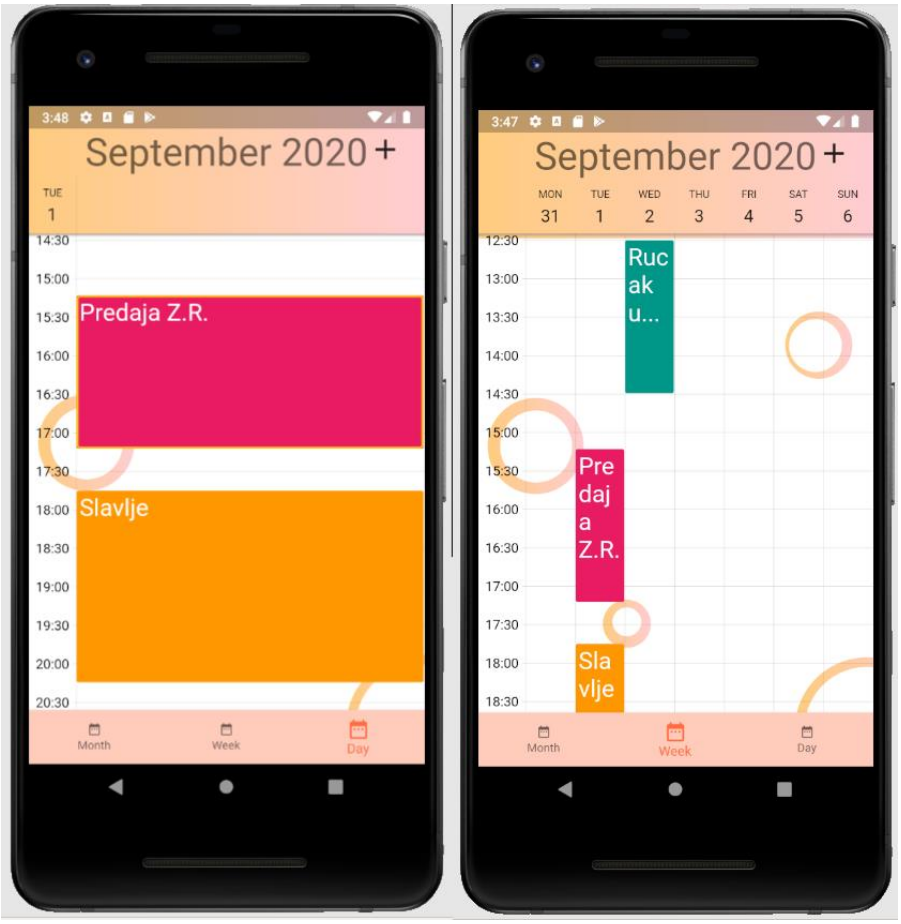
Slika 5.7 Brisanje zadatka

Pritiskom korisnika na potvrdu zadatak se briše iz baze podataka i lista trenutnih zadataka se osvježava. Jednim korakom unazad korisnik se vraća na naslovnu stranicu gdje pritiskom na gumb Kalendar, korisnik odlazi na zaslon svih unesenih događaja prikazan na slici 5.8. Postoje tri vrste prikaza događaja: mjesečni, tjedni i dnevni, a koji su prikazani na slici 5.9.



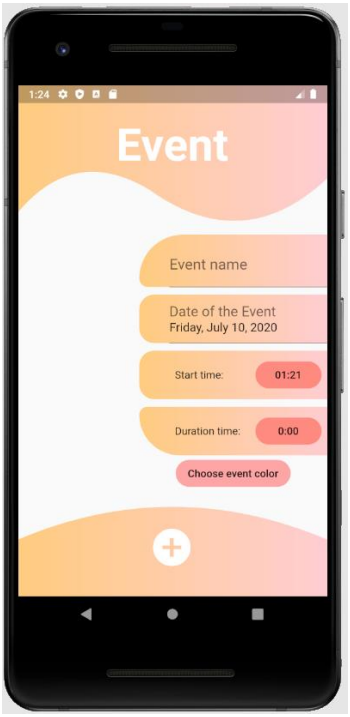
Slika 5.8 Zaslon kalendara

Korisnik odabire prikaz pritiskom na jedan od tri označena mjesta na samome dnu ekrana. Prikaz koji je već odabran bit će u drugoj boji kako bi se lakše raspoznao.



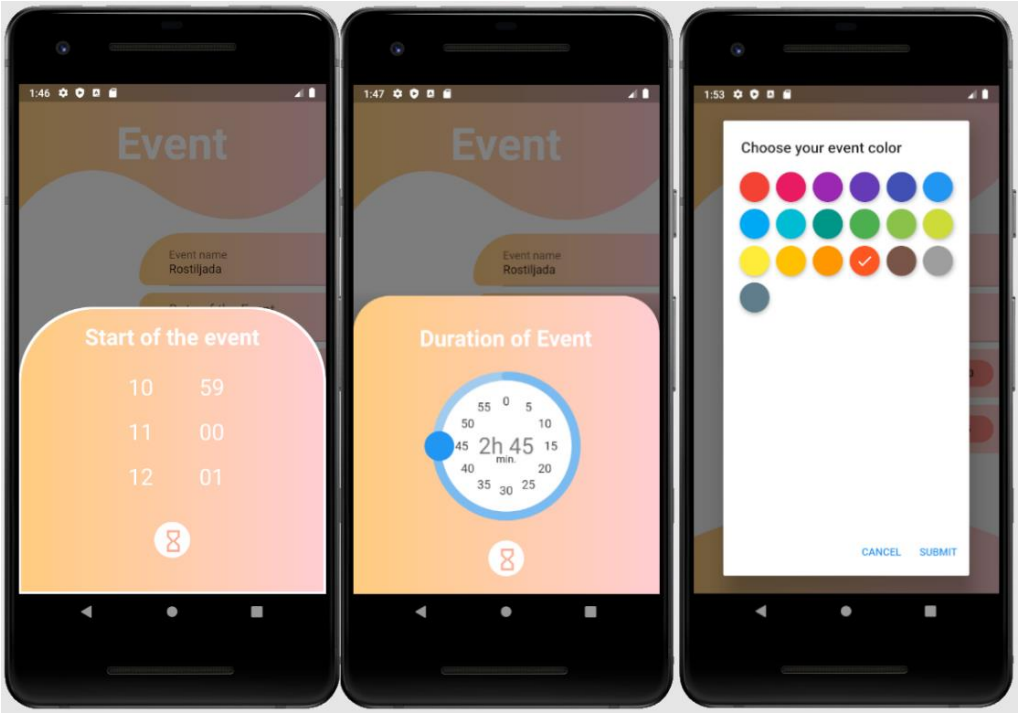
Slika 5.9 Dnevni i tjedni prikaz

Korisnik dodaje novi događaj ako pritisne gumb u obliku plusa u gornjem desnom kutu. Slika 5.10 prikazuje zaslon koji se otvori nakon pritiska korisnik na gumb.



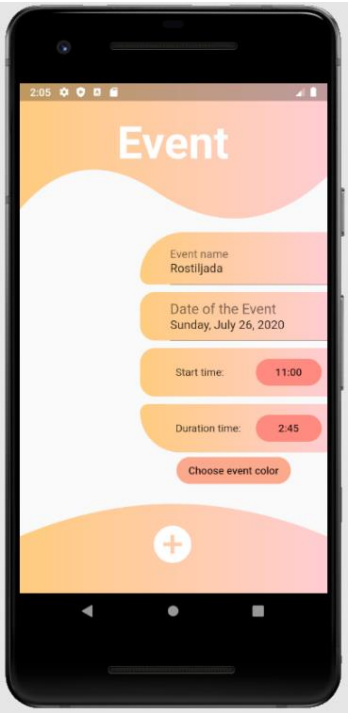
Slika 5.10 Zaslon za dodavanje događaja

Kako bi zahtjev za novi događaj bio uspješno uveden u sustav, korisnik treba popuniti ime, datum održavanja, početak i trajanje događaja. Također korisnik ima mogućnost odabrati boju za događaj koji će biti prikazan na kalendaru. Slika 5.11 prikazuje kako izgleda ispunjavanje početka, trajanja i boje događaja.



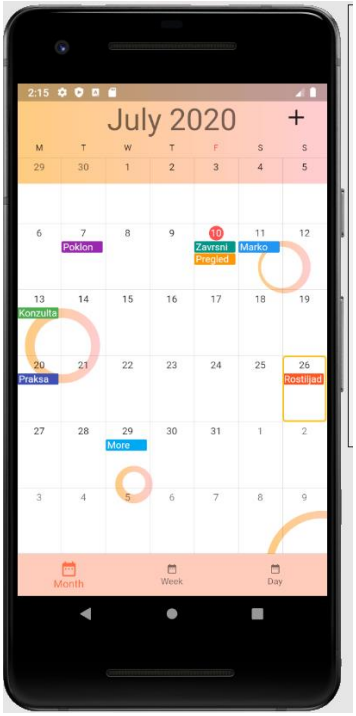
Slika 5.11 Ispunjavanje informacija događaja

Nakon što su ispunjeni svi elementi potrebni za unos novog događaja, događaj se sprema, a aplikacija vraća korisnika na mjesečni pregled koji sada izgleda kao na slici 5.10. Slika 5.12 prikazuje kako izgleda obrazac kada su popunjeni svi elementi koji opisuju događaj.



Slika 5.12 Primjer događaja

Pritiskom na gumb u obliku plusa, ako su ispunjeni svi potrebni podaci, dodaje se događaj u bazu podataka. Slika 5.13 prikazuje kako izgleda kalendar nakon što je unesen novi događaj.



Slika 5.13 Zaslon nakon unosa novog događaja

Slika 5.14 prikazuje što se dogodi ako korisnik pritisne dan, tada on može detaljno pogledati sve događaje koje je unio.

Slika 5.14 Detaljan prikaz događaja

6. PREDNOSTI I NEDOSTACI U ODNOSU NA POSTOJEĆA RJEŠENJA

Aplikacija napravljena od strane studenta teško da se može mjeriti s aplikacijom koju je napravila neka kompanija. Kompanije koje proizvode aplikacije brinu se o raznim aspektima poput dizajna koji privlači korisnike, optimiziranosti za razne uređaje kako bi aplikacija bila dostupna što većem broju korisnika i temeljito testiraju sve dijelove aplikacije kako bi se izbjegle pogreške prilikom korištenja aplikacije. Prednost s obzirom na već postojeće aplikacije je lako i intuitivno korištenje. Sve novije aplikacije koriste neku vrstu spremanja korisničkih podataka na Internet odnosno internetske baze podataka čime osiguravaju korisnicima bezbrižno korištenje aplikacije bez obzira na gubitak ili prelazak na novi mobilni uređaj. Neki od nedostatka su to što aplikacija nije optimizirana za velik broj starih uređaja nego samo za novije modele i postoji veća mogućnost da se dogodi pogreška prilikom nepravilnog unosa podataka u aplikaciju. Izgled dizajna nije posebno razrađen. Dizajn je sačinjen od jednostavnih valovitih ili zaobljenih oblika. Dok je dizajn postojećih rješenja puno bolje razrađen, neke stavke su preglednije zbog boljeg rasporeda elemenata na zaslonu. Većina postojećih rješenja nudi gotovo iste funkcionalnosti kao i aplikacija opisana u radu, najveća razlika je u samome dizajnu i načinu na koji su prikazani podaci. Najveća prednost aplikacije opisane u radu jest ta da se jednom napisanim kodom pokrivaju dvije različite platforme iOS i Android bez potrebe za velikim izmjenama koda.

7. ZAKLJUČAK

U današnje vrijeme ljudi se brinu kako vrijeme sve brže prolazi i sve je teže ostati usredotočen na sadašnjost. Aplikacija je napravljena kako bi pomogla ljudima što bolje organizirati svoje vrijeme, kako ne bi propustili važne događaje svojih bližnjih poput rođendana, putovanja na more i godišnjice braka, ali to mogu biti i mali događaji poput kupovine poklona bližnjoj osobi ili obiteljskog roštilja. Dobra organizacija pomaže čovjeku da se na kraju dana osjeća zadovoljno kada shvati da nije potratio još jedan dan u nizu. Izvorni kod aplikacije napisan je u Dart programskom jeziku kako bi se pomoću Flutter razvojnog okvira aplikacija mogla koristiti na više različitih platformi. Multiplatformski način rada uvelike pomaže programerima prilikom izrađivanja aplikacije jer ne moraju pisati dva različita koda za jednu aplikaciju. Aplikacija koristi Firebase platformu za spremanje podataka u bazu podataka, ali i za autentifikaciju korisnika u sustav.

Sama aplikacija ima još mjesta za razvoj i nadogradnju iako ispunjava sve zadane ciljeve. Postoji mogućnost da bi aplikacija bila još korisnija kada bi svi uneseni događaji unutar aplikacije bili sinkronizirani s ugrađenim kalendarom što bi omogućilo korisniku da unosi događaje s ugrađenog kalendara ili kalendara aplikacije. Bilo bi dobro ponuditi korisnicima više tema kako bi mogli izabrati željeni izgled aplikacije, tako da izaberu temu između nekoliko ponuđenih ili mogućnost da sami sebi naprave svoju temu.

LITERATURA

[1] Beesoft , »Beesoft Apps Planer Pro,« Beesoft , [Mrežno]. Available: https://apps.beesoft.io/PlannerPro/HomePage.html.

[2] Any.do, »Any.do,« Any.do, [Mrežno]. Available: https://www.any.do/teams/.

[3] Visual Studio Code, »Setting up Visual Studio Code,« Microsoft, [Mrežno]. Available: https://code.visualstudio.com/docs. [Pokušaj pristupa 26 August 2020].

[4] Visual Studio Code, »Language Support in Visual Studio Code,« Microsoft, [Mrežno]. Available: https://code.visualstudio.com/docs/languages/overview. [Pokušaj pristupa 26 August 2020].

[5] Dart, »Language samples | Dart,« [Mrežno]. Available: https://dart.dev/samples. [Pokušaj pristupa 26 August 2020].

[6] Flutter, »Hot reload - Flutter,« Google, [Mrežno]. Available: https://flutter.dev/docs/development/tools/hot-reload. [Pokušaj pristupa 26 August 2020].

[7] Firebase, »Firebase Pricing,« Google, [Mrežno]. Available: https://firebase.google.com/pricing. [Pokušaj pristupa 26 August 2020].

[8] Adobe XD, »UI/UX design and collaboration tool | Adobe XD,« Adobe, [Mrežno]. Available: https://www.adobe.com/products/xd.html?promoid=PYPVQ3HN&mv=other. [Pokušaj pristupa 26 August 2020].

[9] L. Valishvili, »Adobe XD Review | Toptal,« [Mrežno]. Available: https://www.toptal.com/designers/adobe/adobe-xd-review. [Pokušaj pristupa 26 August 2020].

[10] J. Pimento, »Managing Assets with Adobe XD,« Medium, 7 April 2018. [Mrežno]. Available: https://medium.com/@Jonthanjosh/design-systems-with-adobe-xd-5b3ac49b7b09. [Pokušaj pristupa 26 August 2020].

[11] Adobe, »Learn how to create interactive prototypes others can use to test, optimize, and perfect the user experience.,« Adobe, [Mrežno]. Available: https://helpx.adobe.com/xd/help/create-prototypes.html. [Pokušaj pristupa 26 August 2020].

POPIS SLIKA

Slika 2.1 Planner Pro dnevni pregled, izvor: [1] 5
Slika 2.2 Planner Pro tjedni prikaz, izvor: [1] 6
Slika 2.3 Any.do aplikacija [2] 7
Slika 2.4 Any.do podjela zadataka [2] 8
Slika 3.1 Visual Studio Code 10
Slika 3.2 Dart kod 11
Slika 3.3 Flutter Android simulacija 12
Slika 3.4 Firebase baza podataka 13
Slika 3.5 Adobe XD alat, izvor: [10] 14
Slika 4.1 Dizajn aplikacije 15
Slika 4.2 Povezivanje navigacije 16
Slika 4.3 Dio koda autentikacije 17
Slika 4.4 To-Do dnevna lista zadataka 18
Slika 4.5 Učitavanje dnevnih zadataka,događaja 18
Slika 4.6 Izbornik To-Do 19
Slika 4.7 Brisanje To-Do elementa s liste 20
Slika 4.8 Brisanje To-Do elementa sa serverske strane 20
Slika 4.9 Dodavanje To-Do elementa na server 21
Slika 4.10 Opcije kalendara 22
Slika 4.11 Prikaz kalendara 23
Slika 4.12 Dodavanje ime događaja 23
Slika 4.13 Testiranje prijave korisnika 24
Slika 5.1 Zaslon za prijavu korisnika 25
Slika 5.2 Zaslon za registraciju korisnika 26
Slika 5.3 Naslovni zaslon 26
Slika 5.4 Otvoreni pretinac 27
Slika 5.5 Zaslon "To do" liste 27
Slika 5.6 Dodavanje zadatka 28
Slika 5.7 Brisanje zadatka 29
Slika 5.8 Zaslon kalendara 29
Slika 5.9 Dnevni i tjedni prikaz 30
Slika 5.10 Zaslon za dodavanje događaja 31

Slika 5.11 Ispunjavanje informacija događaja 31
Slika 5.12 Primjer događaja 32
Slika 5.13 Zaslon nakon unosa novog događaja 32
Slika 5.14 Detaljan prikaz događaja 33

SAŽETAK

Cilj ovog završnog rada bio je napraviti multiplatformsku aplikaciju nazvanu Planned koja pomaže pri organiziranju raznih događaja i zadataka. Svi korisnici aplikacije registracijom osiguravaju da će se njihovi podaci odnosno spremljeni događaji mogu pogledati neovisno o uređaju. Za izradu dizajna aplikacije korišten je Adobe XD alat dok je izvorni kod pisan u programskom jeziku zvanom Dart u programu Visual Studio Code. Flutter razvojno sučelje omogućava distribuciju aplikacije na više platformi. Firebase platforma omogućava aplikaciji jednostavnu autentifikaciju i Internetsku bazu podatka u kojoj su spremljeni svi korisnički podaci. U teorijskom dijelu rada objašnjeno je rukovanje aplikacijom i njezine funkcionalnosti. Također opisane su neke od razlika između postojećih aplikacija i njihovih funkcionalnosti. Funkcionalnosti Planned aplikacije su dodavanje i brisanje događaja i zadataka.

Ključne riječi: Dart, Flutter, Firebase, Planer,

ABSTRACT

Multiplatform application for planning

Goal of this bachelor's thesis was to create a multiplatform application called Planned that helps organize various events and tasks. All users of the application ensure that their data or saved events can be viewed independently of the device simply by registering. The Adobe XD tool was used to design the application, while the source code was written in a programming language called Dart in Visual Studio Code. The Flutter framework allows distribution of application on multiple platforms. The Firebase platform allows the application to easily authenticate and manage online database in which all user data is stored. The theoretical part of the paper explains the handling of the application and its functionality. Some of the differences between existing applications and their functionalities are also described. The functionalities of the Planned application are adding and deleting events and tasks.

Keywords: Dart, Flutter, Firebase, Planner,

ŽIVOTOPIS

Antonio Krušlin je rođen 22. srpnja .1998. godine u Đakovu, Hrvatska. Pohađao je Osnovnu školu Bilje u naselju Bilje. Nakon završetka osnovne škole u OŠ Bilje, 2013. godine upisuje se u prirodoslovno-matematičku gimnaziju odnosno III. Gimnaziju Osijek s dodatnim predmetnom Napredna informatika. Tijekom školovanja savladao je osnove programiranja i robotike. Upisuje se na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija 2017. godine na preddiplomski sveučilišni studij Računarstva.

PRILOZI

Na priloženom DVD disku su sadržani:

- Izvorni kod aplikacije
- Datoteke dizajna