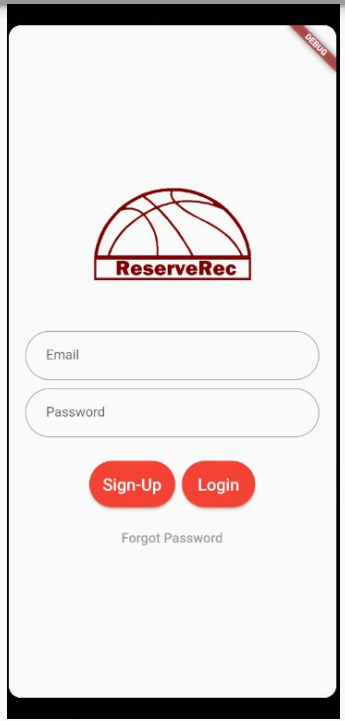
ReserveRec

Ryan Carrigan, Bryant Conquest, Philip Speegle, Zack Withers

Overview: What is ReserveRec?

- Play any sport, any time
- Custom feed where you can join or create events
- Custom user profiles and display statistics



Major Sprint 2 Goals

➢ Authenticate Users via Firebase	4 hours ✓
➢ Link Authenticated Users to Firestore and Create Database	15 hours ✓
➢ Validate User and Location Information	5 hours ✓
➢ Expand User Profiles	10 hours ✓
➢ Expand Post Information	7 hours ✓
➢ Add Dashboard Functionality	10 hours ✓
➢ Create Unit Tests	2 hours ✓

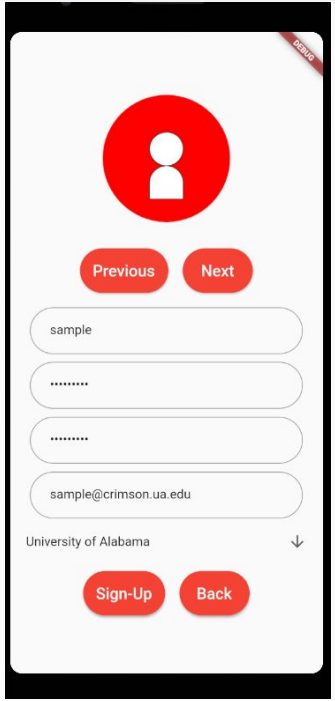
Overarching Theme: Integration via Firestore and Add Functionality

Backend Updates

- Authenticate Users via Cloud Firestore
- Real-time database updates, additions, retrievals, and deletions
- Collections for valid users, posts, emails, and locations

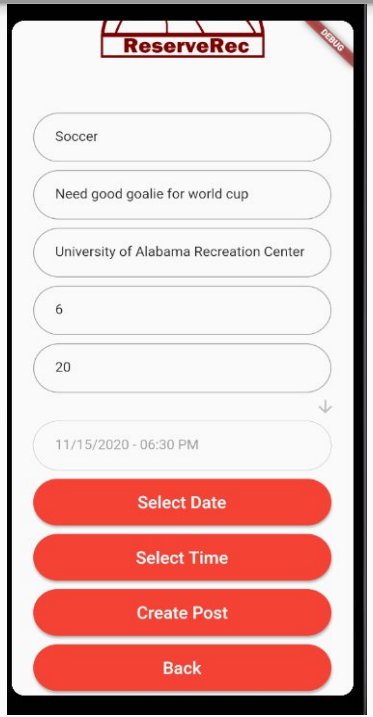
Authentication

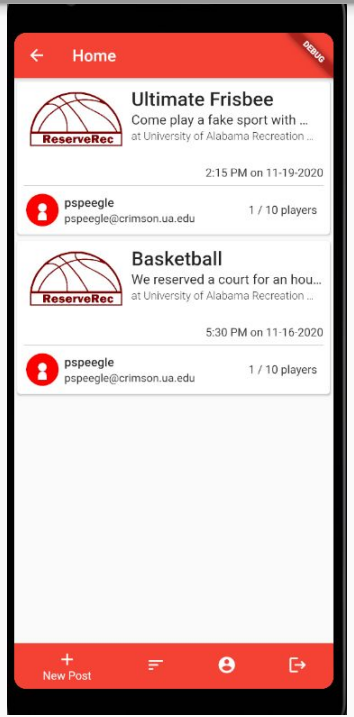
- Requirements for username and password
- Email must be a valid university email
- After sign-up, a user must verify their email with a link in order to login
- “Forgot password” functionality
- All provided through firebase authentication



Feed Updates

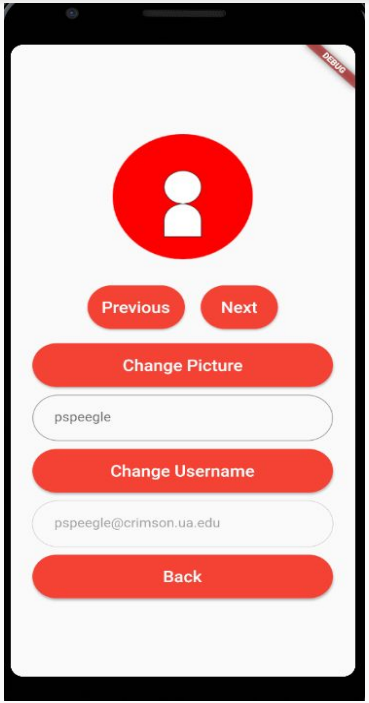
- Sort posts in different ways
- DateTime selector for new posts
- Nicer layout
- Join events others created in real time
- Refresh functionality





Profile

- Separate screen to display user information, change profile picture, and change username
- Future Goals: Add “followers/following” to allow filtering of posts



Challenges

- Dart does not have a mature ecosystem
- No unit tests with firebase
- Lots hidden config things that you don't know until you come across them
  - Linter Options
  - Can't use async functionality in a .ForEach
  - List need to be initialized but it gives no warning

Future Plans

- Create a Manager Profile to be able to create new school and manage posts
- Follower Following for your friends
- Better feedback from App (snackbar)
- Chat for Each Item
- Report Players
- Statistics

Test Cases

```
00:01 +6: Async vs. Sync Asynchronous Check 2
anything
00:01 +18: DateTime Test Date 1
Func: 2000-01-01 00:00:00.000
00:01 +19: DateTime Test Date 2
Func: 2020-12-25 00:00:00.000
00:01 +20: DateTime Test Date 3
Func: 3001-08-10 00:00:00.000
00:01 +21: Testing Assertions Trying DateTime
Func: -0009-08-30 00:00:00.000
00:01 +31: All tests passed!
```

```
test('Converting User From json', () {
  var testMap = Map();
  testMap['user_id'] = 'testid';
  testMap['user_username'] = 'foo';
  testMap['user_password'] = 'bar';
  testMap['user_email'] = 'email@crimson.ua.edu';
  testMap['verified'] = true;
  testMap['school'] = 'University of Alabama';
  testMap['photoURL'] = 'https://i.imgur.com/DfGZewB.png';
  UserC testUser = UserC.fromJson(testMap);
  expect(testUser.userId, 'testid');
  expect(testUser.userUsername, 'foo');
  expect(testUser.userPassword, 'bar');
  expect(testUser.userEmail, 'email@crimson.ua.edu');
  expect(testUser.verified, true);
  expect(testUser.school, 'University of Alabama');
  expect(testUser.photoURL, 'https://i.imgur.com/DfGZewB.png');
});
```

```
group('Username', () {
  test('Validating Short Username', () {
    expect(verifyUsername("t"), false);
  });

  test('Validating Long Username', () {
    expect(verifyUsername("usernameusername"), false);
  });

  test('Empty Name', () {
    expect(verifyUsername(""), false);
  });

  test('Special Character', () {
    expect(verifyUsername("@user"), false);
  });

  test('Numbers', () {
    expect(verifyUsername("123456"), true);
  });
});
```

```
test('Converting Post From json', () {
  var testMap = Map();
  Timestamp testTime =  Timestamp.now();
  testMap["user_id"] = 'testUserId';
  testMap["post_id"] = 0;
  testMap["description"] = 'desc';
  testMap["time_posted"] = testTime;
  testMap["time_set"] = testTime;
  testMap["sport"] ='sportball';
  testMap["location"] = 'testLoc';
  testMap["school"] = 'University of Alabama';
  testMap["max_people"] = 20;
  testMap["min_people"] = 5;
  testMap["cur_people"] = 6;
  Post testPost = Post.fromJson(testMap);
  expect(testPost.postUserId, 'testUserId');
  expect(testPost.postId, 0);
  expect(testPost.postDescription, 'desc');
  expect(testPost.postTimePosted, testTime.toDate());
  expect(testPost.postTimeSet, testTime.toDate());
  expect(testPost.postSport, 'sportball');
  expect(testPost.postLocation, 'testLoc');
  expect(testPost.school, 'University of Alabama');
  expect(testPost.maxPeople, 20);
  expect(testPost.minPeople, 5);
  expect(testPost.curPeople, 6);
});
```

Live Demo

Thank you

Ryan Carrigan: rpcarrigan@crimson.ua.edu
Bryant Conquest: bmconquest@crimson.ua.edu
Philip Speegle: pspeegle@crimson.ua.edu
Zack Withers: wzwithers@crimson.ua.edu