

ReserveRec

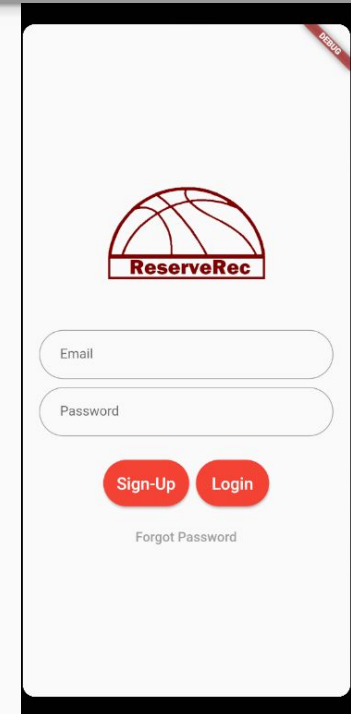
Ryan Carrigan, Bryant Conquest, Philip Speegle, Zack Withers



Overview: What is ReserveRec?



- Play any sport, any time
- Custom feed where you can join or create events
- Custom user profiles
- Notifications for people you follow





Major Sprint 3 Goals

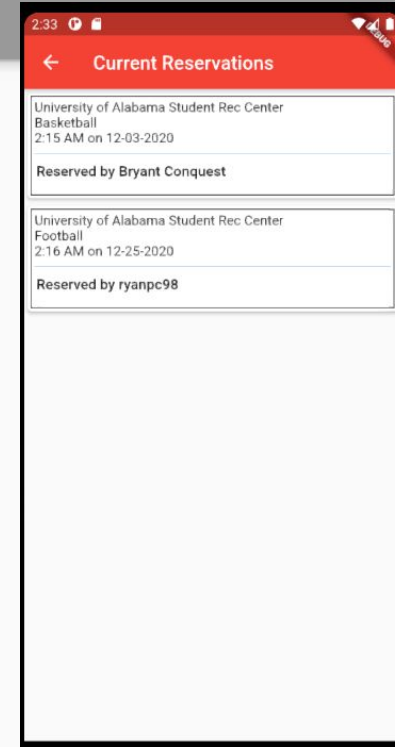
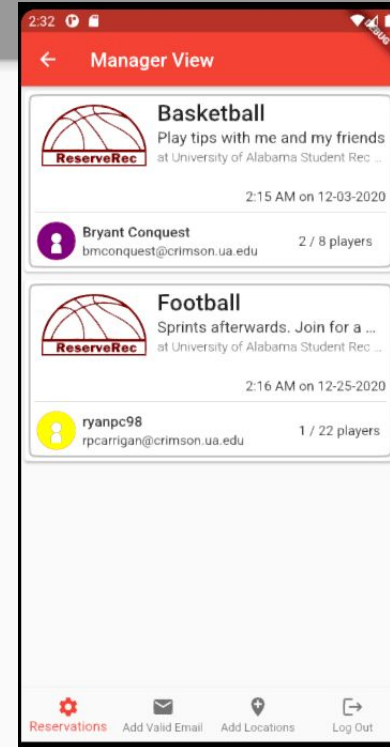
➤ Revamp Dashboard UI	5 hours ✓
➤ Create Entire Separate Section for Managers	25 hours ✓
➤ Implement Chat for Posts	10 hours ✓
➤ Implement Post Notifications and Alerts	12 hours ✓
➤ Add Team Building	8 hours ✓
➤ Add Friends and Followers/Following	15 hours ✓
➤ Create More Unit Tests	2 hours ✓

Overarching Theme: Implementing the Rest of the Core Features

Manager View

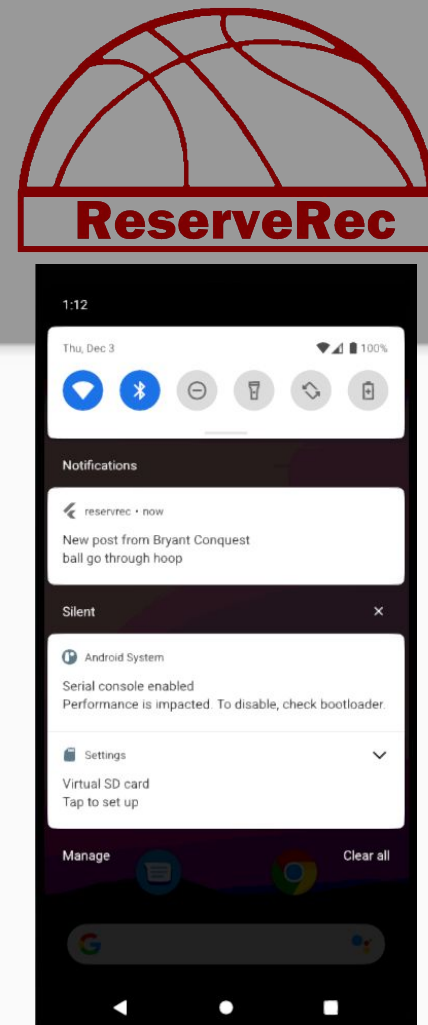
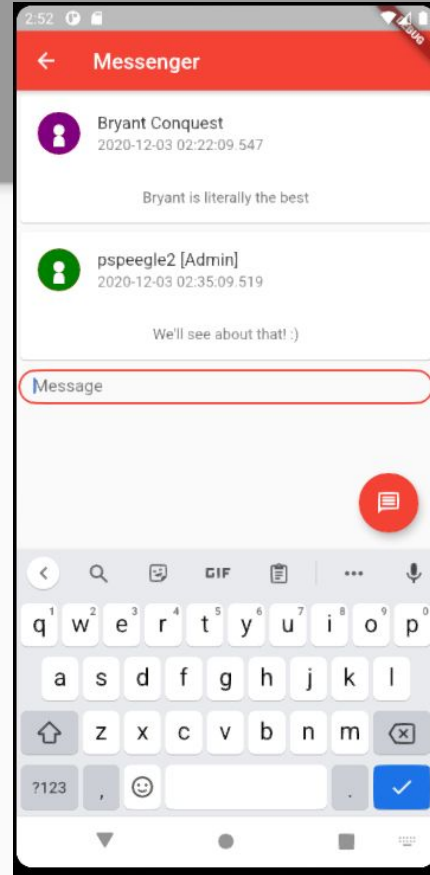


- Reservation List: A simplified version of dashboard that highlights which times are reserved and where
- Managers can add valid emails, locations, and delete any post
- Managers are highlighted with an [Admin] tag
- Map for adding locations



Chat and Notifications

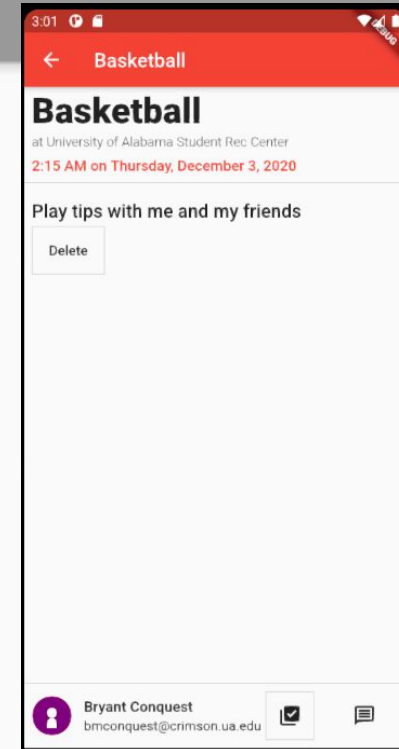
- Each post has its own chat
- Each message is in Firebase
- Followers get notified of posts



UI Revamp



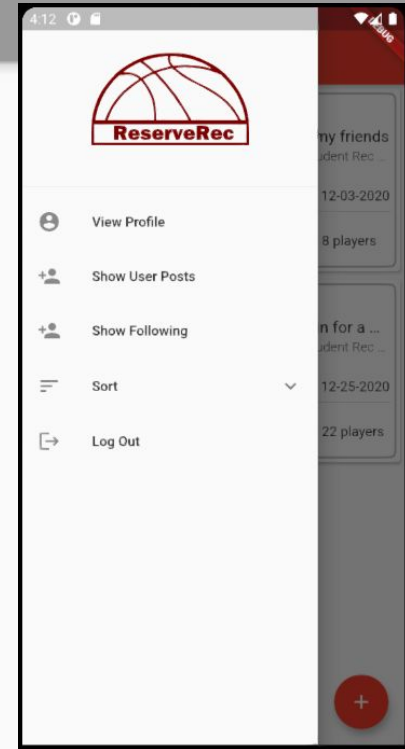
- Join post and chat function added
- Delete post added
- Various style changes to make the app look nicer and more compact
- Profile changes



Followers/Following



- Users now have posts filtered based on whom they are following
- Separate filters for show only user posts and show all
- Drawer layout instead of layers of buttons





Overcoming Challenges

- In Sprint 3, we focused on overcoming the challenges presented in our previous sprints
- We overcame the fact that Dart does not have a mature ecosystem by getting more familiar with the language
- Overhauling the structure of our source code made it more readable and organized
- Working within the “projects” tab allowed us to better assign tasks
- Better communication allowed us to avoid as many merge conflicts

Challenges



- Some of our original widgets were stateless; updating them to stateful was very complex
- Some of the libraries we were using failed to work, so features such as reloading would not work in some modes
- When there are many items in a database, retrieving them slows the application down
- Different code styles made understanding what the other group members wrote harder to understand



Post-Launch Ideas

- Our application is finished and ready to launch. However, there is always room for more:
- More advanced statistics could determine the skill of each player and group them with others of their skill level
- Implement a “Live chat”, allowing users to see chats without refreshing
- Reduce the amount of async calls to improve application speed
- Have a cache to use less data and improve performance of backend

Test Cases (Total: 48)

```
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 +48: All tests passed!
```

- We created 17 new tests, each with new functions made during Sprint 3:

```
group('Is Numeric Test', () {
  {
    test('Integer', () {
      expect(isNumeric("1"), true);
    });

    test('Double', () {
      expect(isNumeric("0.1234"), true);
    });

    test('Negative number', () {
      expect(isNumeric("-17"), true);
    });

    test('Random letters', () {
      expect(isNumeric("this is not a number"), false);
    });

    test('Hash Code', () {
      expect(isNumeric(Sha256("")), false);
    });

    test('Emoji', () {
      expect(isNumeric("👉"), false);
    });
  }
});
```

```
test('Sorting four ways', () {
  assert(sortMethod.length == 4);
});

test('Can set sort index', () {
  setSortIndex(1);
  assert(getSortIndex() == 1);
});

test('Cycling', () {
  setSortIndex(0);
  cycleFunction();
  assert(getSortIndex() == 1);
});

test('Modular Arithmetic !', () {
  setSortIndex(3);
  cycleFunction();
  assert(getSortIndex() == 0);
});
```

```
group('Distance', () {
  {
    test('Zero distance', () {
      expect(calcDistance(0,0,0,0), 0);
    });

    test('Same position', () {
      expect(calcDistance(10,10,10,10), 0);
    });

    test('Random coordinates 1', () {
      expect(calcDistance(13.4,13,12,11), 6595.002306580232);
    });

    test('Random coordinates 2', () {
      expect(calcDistance(-10,3,103,14), 6182.122582136241);
    });

    test('Out of bounds parameters', () {
      expect(() => calcDistance(181,-1000,1000,0), throwsAssertionError);
    });
  }
});
```

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