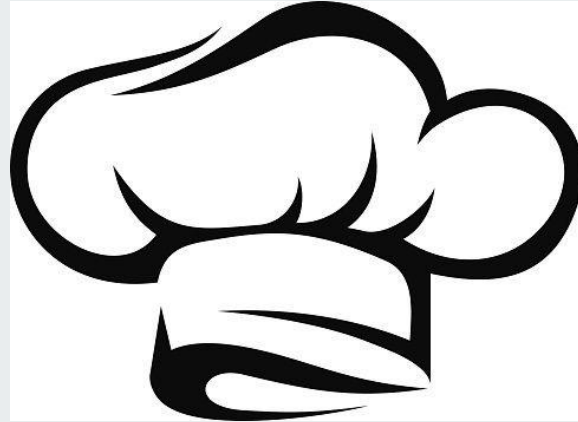




MyChef Sprint 3 Presentation

Erik Mortimer
Marlee Bryant
Andrew Sartor
Kevin Baker



Reminder of the project focus

- Simplify cooking for beginners, college students, and busy families
 - Less waste
 - Save money
 - Saves time



- Features
 - Generation of a shopping list for user (export to PDF or in app)
 - Sorting of recipes based on ingredients that you already have
 - Searching based on diet styles, calories, macronutrients, required cookware, type of dish, time for completion, cuisine, etc.
 - Accessibility options like Text to Speech and casting of recipes to Smart TVs



Major Sprint 3 achievements

- Successfully implemented calendar list functionality for adding ingredients
- Automated deployment pipeline for backend with GitHub
- Increased reliability and speed of backend response times
- Increased communication between Frontend and Backend
- Pantry UI Completed

Useful Test Cases

```
// Tests to make sure when a user selects a new day, calendar binds to the new day
@Test
fun dayBinderIsCalledOnDayChanged() {
    onView(withId(R.id.miList)).perform(click())
    sleep( millis: 2000)

    class DayViewContainer(view: View) : ViewContainer(view)
    val calendarView = homeScreenRule.activity.listFragCalendar
    var boundDay: CalendarDay? = null
    val changedDate = currentMonth.atDay( dayOfMonth: 4)
    homeScreenRule.runOnUiThread {
        calendarView.dayBinder = object : DayBinder<DayViewContainer> {
            override fun create(view: View) = DayViewContainer(view)
            override fun bind(container: DayViewContainer, day: CalendarDay) {
                boundDay = day
            }
        }
    }

    // Allow time to pass for data to propagate
    sleep( millis: 2000)
    homeScreenRule.runOnUiThread {
        calendarView.notifyDateChanged(changedDate)
    }

    // Allow time to pass for data to propagate
    sleep( millis: 2000)
    assertTrue(boundDay?.date == changedDate)
    assertTrue(boundDay?.owner == DayOwner.THIS_MONTH)
}
```

Useful Test Cases (cont.)

```
@RunWith(AndroidJUnit4::class)
class PantryFragmentTest {

    @Rule
    @JvmField
    val rule: ActivityTestRule<MainActivity> = ActivityTestRule(MainActivity::class.java)

    @Test
    fun checkAdd() {
        onView(withId(R.id.imageButton)).perform(click())
        onView(withId(R.id.miFavorites)).perform(click())
        onView(withId(R.id.miPantry)).perform(click())
        onView(withId(R.id.miPantry)).check(matches(withText(text = "Quantity: 1")))
    }

    @Test
    fun checkSub() {
        onView(withId(R.id.minusButton)).perform(click())
        onView(withId(R.id.miFavorites)).perform(click())
        onView(withId(R.id.miPantry)).perform(click())
        onView(withId(R.id.miPantry)).check(matches(withText(text = "Quantity: 0")))
    }
}
```



Not completed

- PDF generation of shopping list items
- Accessibility features (Text to speech, handwriting recognition)
- Social media aspects

Lessons Learned

- Might not be able to implement all of the desired features in a Sprint
- Some of the easiest looking features are deceptively difficult to implement
- Overall, sticking to languages and systems that the whole team has had experience with

Live Demo
