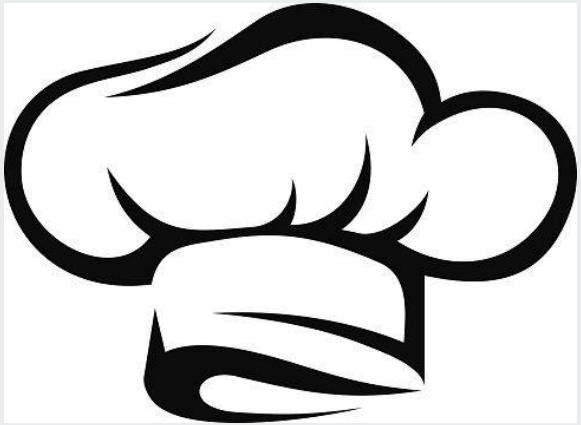
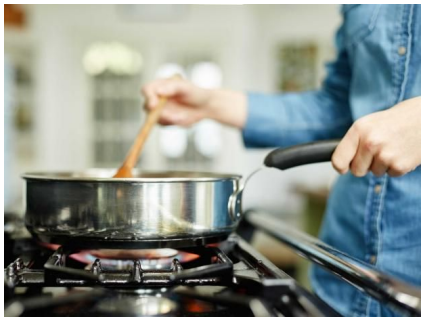
MyChef Sprint 2 Presentation

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

Sprint 2 Goals

- Finish implementing login system using Firebase
- Create a better filtering and searching algorithm for recipes
- Adding favorites to a specific user account
- Compiled shopping list of ingredients, created virtually or physically via PDF

Major Sprint 2 achievements

- Completed Login system using Firebase
- Fleshed out more UI within MyChef
- Added Calendar and Favorites section
- Gained a better understanding of best practices of Kotlin/Android development
- Added search functionality for ease of finding recipes
- Added Details Page with the ability to add recipes to Favorites

Project Backlog

ID	Story	Estimation (hours)	Priority	Sprint When Finished
1	Display a list of recipes	9	1	Sprint 1/ Sprint 2
2	Allow the user to create an account and login	6	1	Sprint 1
3	Add recipes to a calendar for meal planning	4	1	Sprint 2
4	Create shopping list with ingredients from a users selected recipes	8	1	
5	Allow basic filtering and searching of recipes	10	1	Sprint 2
6	Allow user to search for recipes with ingredients and cookware they have	10	1	
7	Allow user to export shopping list	8	2	
8	Calculate average ingredient cost	6	3	
9	Provide accesibility options to the user like text to speech and casting	4	3	
10	Allow user to upload handwritten recipes	8	3	
11	Incorporate a social aspect to the user experience	16	3	
12	Create a details page for the recipes and allow addition to favorites	6	1	Sprint 2

Story Number	Task ID	Task Description	Hours Estimation
1	a	Create UI for serach page	4
1	b	Implement scraping tool	1
1	c	Improve the search algorithm results	3
1	d	Implement measurement conversions for ingredients	1
2	a	UI for login page (connecting to google and facebook)	1
2	b	UI for create account	1
2	c	UI for view/edit account information/ favorites	2
2	d	Backend for User info	3
2	e	Connecting backend to frontend UI	3
3	a	Calendar to add recipes on certain days	3
3	b	Add selector for what weeks/days to aggregate into the shopping list	1
4	a	Compile recipes ingredients from selected weeks/days	2
4	b	Allow the shopping list to be exported to pdf or printed	1
5	a	Add list of filters	2
5	b	Scrape relevant information from recipes for filters	3
5	c	Implement search function	5
12	a	Create UI for details page	3
12	b	Add functionality to add to favorites	3

Sprint Backlog

Story 3: Add recipes to a calendar for meal planning

Story 4: Create shopping list with ingredients from a users selected recipes

Story 5: Allow basic filtering and searching of recipes

Story 7: Allow user to export shopping list

Story 12: Create details page and adding to favorites

Test Cases

- Test case for navigation to favorites

```
@Rule
@JvmField
val rule: ActivityTestRule<MainActivity> = ActivityTestRule(MainActivity::class.java)
@Test
fun useAppContext() {
    // Context of the app under test.
    val appContext = InstrumentationRegistry.getInstrumentation().targetContext
    assertEquals( expected: "com.mychef.mychef", appContext.packageName)
}
@Test
fun checkFavoritesNavigation(){
    onView(withId(R.id.miFavorites)).perform(click())
    onView(withId(R.id.tvFavoritesFragment)).check(matches(withText( text: "Favorites")))
    onView(withId(R.id.tvFavoritesFragment)).check(matches(withText( text: "Lemon Pepper Chicken Breast")))
}
```

Test Cases (cont.)

- Test Case for navigation to Pantry

```
@Test
fun checkPantryNavigation(){
    onView(withId(R.id.miPantry)).perform(click())
    onView(withId(R.id.tvPantryFragment)).check(matches(withText( text: "Pantry")))
}
```

Test Cases (cont.)

- Test Case for navigation to List

```
@Test
fun checkListNavigation(){
    onView(withId(R.id.miList)).perform(click())
    onView(withId(R.id.tvListFragment)).check(matches(withText( text: "List")))
}
```

Not completed

- Add recipes to a calendar
- Creating shopping list with ingredients from user
- Various ease of access capabilities such as text-to-speech
- Filtering the search items

Lessons Learned

- Don’t get caught up in the little details of implementation (get something on paper and go from there)
- More communication and execution of implementations needed in the future
- Be flexible to changing strategies

Live Demo