

Skillet

Final Report

Caillin C.

Max K.

Ian J.

CJ K.

Introduction

Value Statement:

Become the Top Chef

Mission Statement:

Revolutionizing cooking education by providing a competitive and fun atmosphere for the Top Chef to thrive.

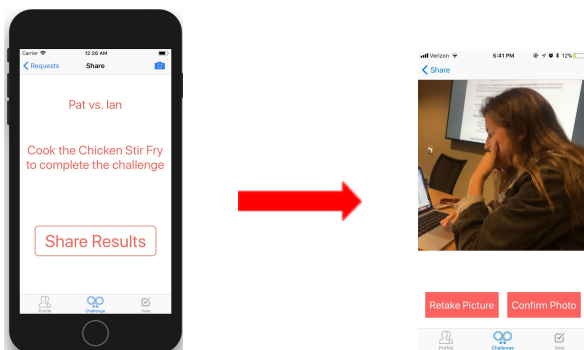
Problem and Solution:

Cooking can be unengaging, socially isolating, and challenging to many. Skillet is our solution to allow people a more fun way to engage in cooking with friends through friendly competition.

Tasks & Final Interface Scenarios:

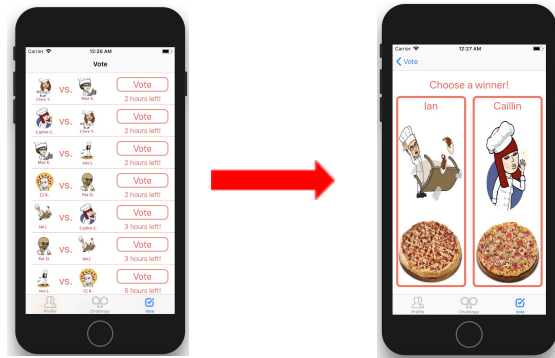
Simple Task: Sharing Results

In our original needfinding, we learned that beginning chefs were often quite proud of their completed recipes and enjoyed showing their successes to friends. As a result, we incorporated the ability to share pictures of completed meals.



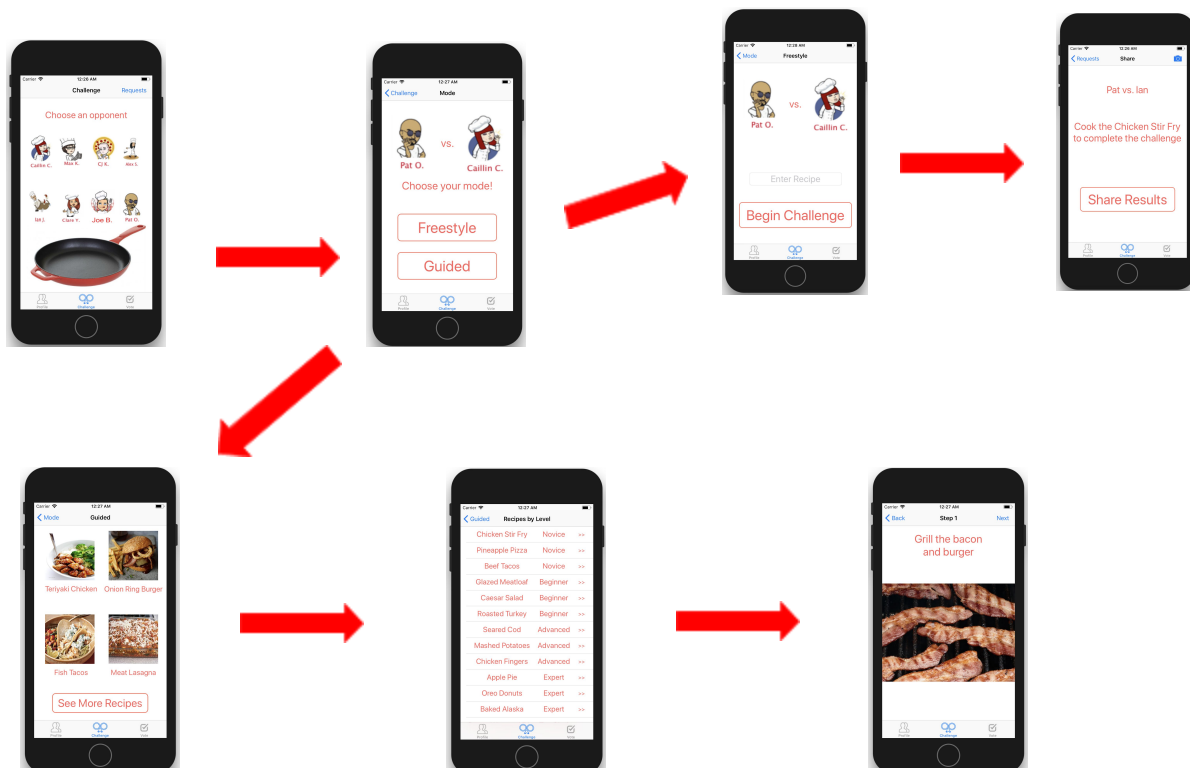
Medium Task: Determining a Winner

In speaking to a number of young people and those lacking experience in cooking, we found that learners often lacked motivation to become better chefs. To address this, we hoped to encourage people through friendly competition by allowing friends to vote on finished products after two users had completed a cooking challenge!

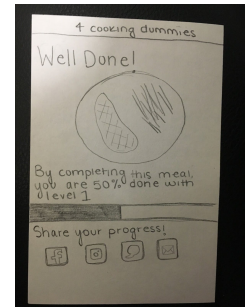
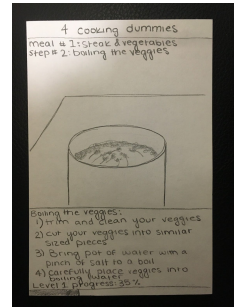
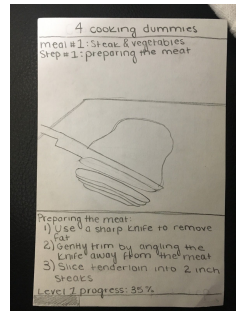
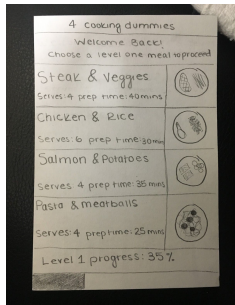


Complex Task: Challenging a Friend and Cooking to Meal

Our complex task was the full process of challenging a friend, and then cooking a full meal. Firstly, we allowed a user to select the bitmoji of the friend that they wanted to challenge. After selecting a friend, the user had two options to begin the challenge. For beginners and those still hoping to follow a recipe, the user can choose the “guided” option. From there, the user can choose from a number of recipes, and will be guided through every step until completion. On the other hand, expert users can choose the “freestyle” mode, and create their own meal without following a set recipe.

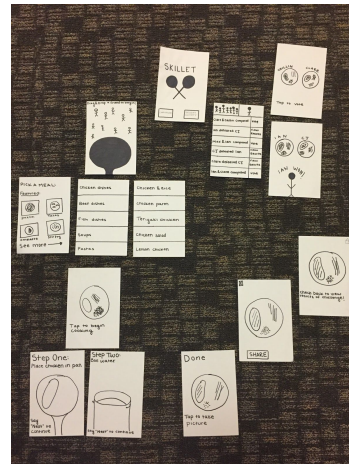
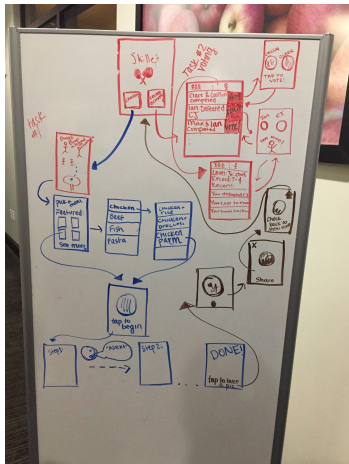


Design Evolution:



In our initial sketches, we planned an app that would guide beginning cooks through an educational step-by-step recipe, and we wanted this app to be social in nature, so we allowed user to share their progress on existing platforms such as facebook, instagram, and iMessaging. However, we were interested to hear from the Stanford Athletics Nutritionist that students were often most motivated to learn to cook when they had a bit of competition with their friends. With that in mind, we decided to shift our focus towards an app that would allow users to challenge and learn alongside their friends.

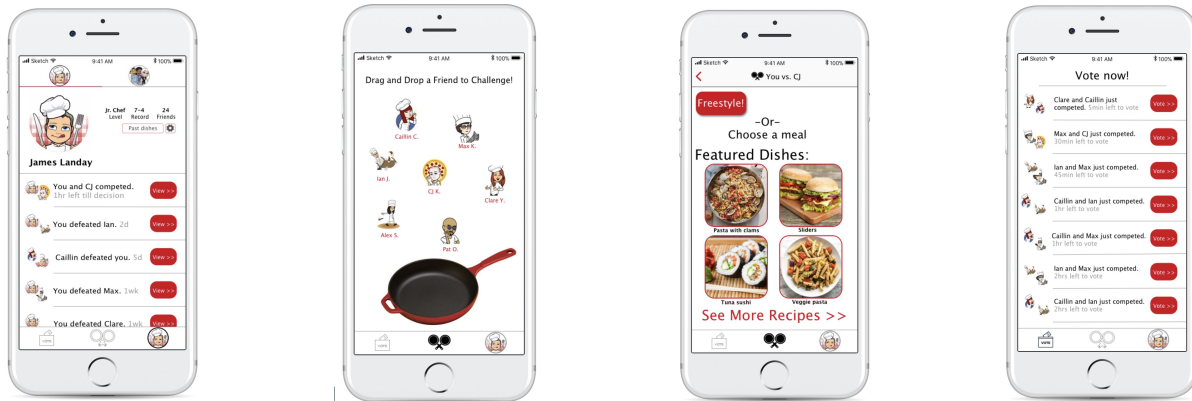
We began designing our competition based cooking app by selecting out our three tasks and developing a task flow to determine the structure of our app.



From our task flow, we designed our low-fi prototype. Our home screen featured two buttons, a “challenge” and a “recent activity” button. From the challenge button, one could complete the task of challenging a friend (shown in blue on the task flow) and share their finished product (in brown). From the recent activity button, a user could vote on prior competitions (shown in red). We then tested our low fi prototype with a number of potential users. We were pleased that they seemed excited about this concept, but this biggest feedback that we got was that the “recent activity” button did not seem like the best way of describing the voting page. In addition, one participant mentioned that expert users might not want to follow a strict recipe, and would rather cook a meal without guidance.

Taking this feedback into consideration, we developed a medium fi prototype which used a navigation bar to display a challenge page, profile page, and voting page, rather than the

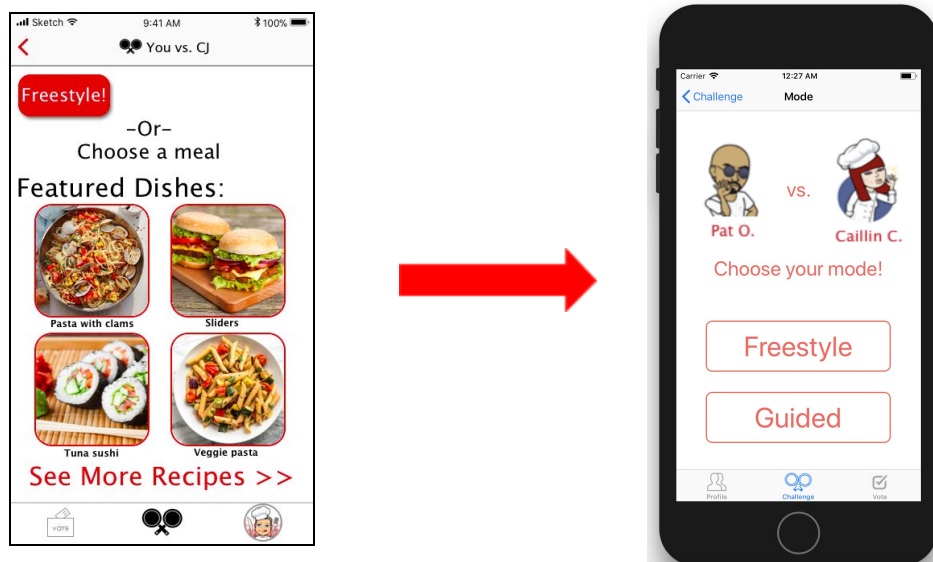
“challenge” and “recent activity” buttons we displayed in our low fi prototype. Additionally, we developed a freestyle mode for users who wanted to challenge a friend without following a set recipe.



After developing our medium fi prototype, we used the feedback from our heuristic evaluations to address our major problems and develop our high fi prototype.

Major Usability Problems Addressed:

- Cluttered Freestyle/Regular battle screen
 - In our medium-fi prototype, we allowed the user to choose between three buttons: ‘Freestyle’, ‘Featured Dishes’, and ‘See More Recipes’. As can be seen from the figure, this was extremely cluttered.
 - We added an intermediate screen that allows you to choose ‘Freestyle’ or ‘Standard’ battle

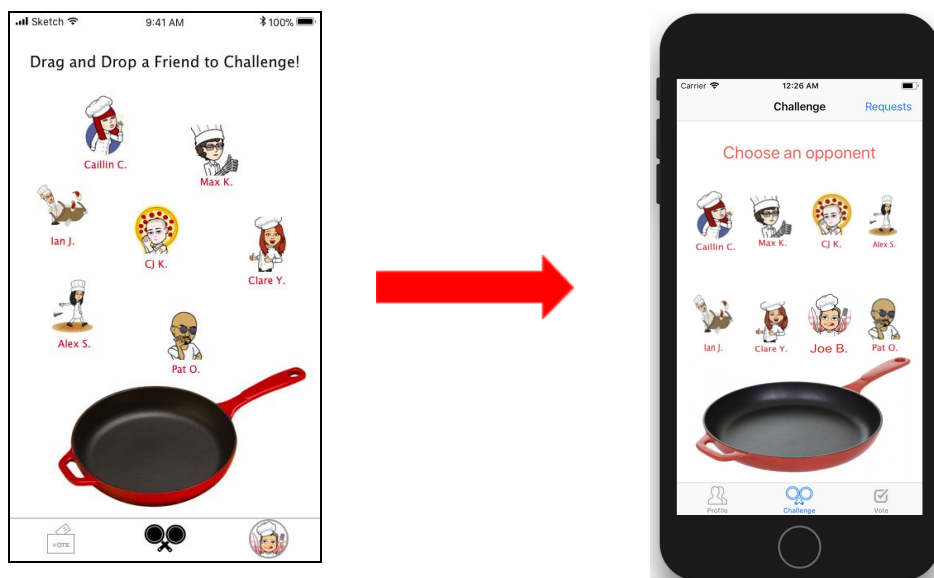


- Back button when completing a recipe
 - We were asked to add a “Back” button that mirrored the “Next” button when completing a recipe.
 - We felt that although this was rated as a level 3 heuristic error, to us the back

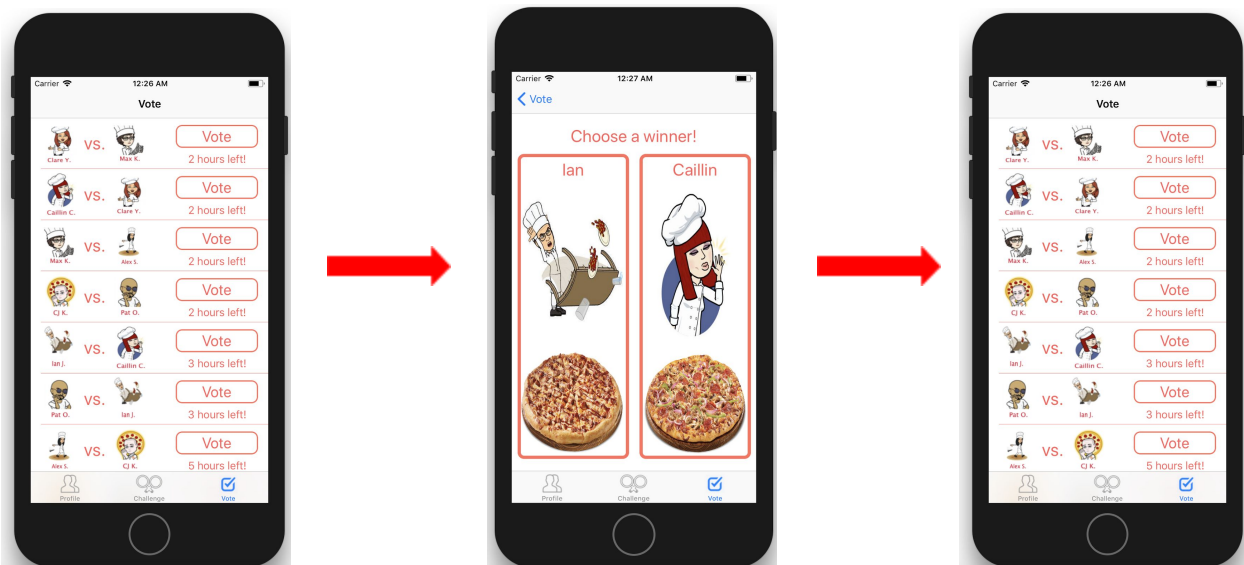
- button intuitively means “back” to the previous page that you just came from.
- This is also a built in function of Xcode so if users are familiar with the upper left hand back error they know that this arrow always returns them to the previous page
- We decided to keep the back arrow in the upper left hand corner of the screen and move the Next button to the top right corner to mirror the back arrow



- Bitmoji screen cluttered
 - The evaluation stated that, if the user had too many friends then this screen would quickly become cluttered.
 - However, we would add functionality so that if your list of friends grew, the Bitmoji screen would have the option of zooming in and out and swiping to navigate around in order to find the bitmoji friend that you are looking for.
 - The pan would stay the same size and remain on the bottom of the screen when you are navigating around the bitmoji world.
 - Realistically, this seems to be an app where you would have only a few friends that you would consistently challenge, but we could add a search bar if this changed.



- Back button when viewing past competitions
 - This was just a slight error we made on the medium fi prototype when uploading to Marvel.
 - We added the wrong header to the specific page that the evaluation referred to.
- Unclear what happens after challenging someone
 - This would be cleared up in a tutorial that the user would optionally follow the first time they logged in
 - Once you challenge someone you don't have to wait for confirmation, you can just do your side of the challenge and then you have to wait for them to complete their side.
- Notifications of battles that you've voted on
 - We had to make a decision here about whether or not the user would like to receive notifications about every battle that they have voted on.
 - We decided that the 'vote' function is similar to 'liking' on Facebook or Instagram; theoretically you could be 'voting' on tens to hundreds of battles every day and we think that the user wouldn't like to receive notifications for all these battles.
 - The Friends feed is there so that you can go back and check on battles that you've voted on.
 - We could add an option in settings if the user really wants the notifications.
- Freestyle search ambiguous
 - Evaluators were confused if the user had to enter a specific recipe name in the search bar that would have to be confirmed, or if they could enter any word or phrase.
 - We decided that the user would be allowed to enter any word or phrase so that they could create their own recipe names.
- Change vote
 - The evaluation stated that it was problematic that the user can't change their vote once they've submitted it in the medium-fi prototype.
 - We added a back arrow that allows you to return to the vote page and re-vote if you accidentally pressed the wrong person.



Prototype Implementation:

We built our prototype in xcode using swift. No one on our team had any prior experience using swift so we had to learn through the use of online videos and tutorials. We layed out our implementation into three different parts. The profile page contains information about the user as well as past matches that the user has won or lost. The challenges page allows users to initiate and accept challenges. The user can initiate a freestyle challenge and choose his/her own recipe, or the guided mode that provides recipes and instructions on how to perform them. The vote page allows users to vote on recent challenges that their friends have completed. Since our application was broken up into these three parts, we utilized a tab bar controller to maintain the three states at all times. We also used many table view controllers to organize data in the application and allow users to quickly and easily see the information on the screen. Lastly, the navigation bar helped us maintain titles and control the flow of the application if the user needed to go back. We did not utilize other xcode features like the grid layout or ARKit. We did not use any of the wizard of Oz techniques in our app since most of the platform is implemented through the mobile app. We hard-coded in all of the users into our system along with recipes, recent matches, voting results, and requests. We hard-coded these into our table views so that the user could get a feel of how the application would look and feel after using it with friends. We also hard-coded in the instructions for our recipes. We provide four different recipes in our guided mode, and each one of them has a set of instructions to allow the user to see how the task would flow. We would add a couple of new features with more time. One would be to make the app completely functional as a social network. This includes creating new user accounts, sending and accepting challenges, and having a messaging system between users. Furthermore, we would add another section of the app that mediates an in-person competition complete with recipes, timing, and judges. Lastly, we would add a more advanced voting system that accurately tracks which votes have been accounted for, which user wins the match, and additional pictures of the results. Overall, we are happy with how much we implemented, and think a couple of further additions could make the app fully functional.