

CHEF GORDY

Making Recipes Easy

Lucia D'Onofrio, Avi Dixit, Robin Oh

Advisor: Steven Weber

MIMS Final Project Report
May 2018

Table of Contents

Acknowledgements	2
Abstract	3
The Challenge	4
The Solution	6
Product Description	6
Product Features	6
Voice User Interface	6
Phone User Interface	7
Value Proposition	7
Market Research	9
Industry Trends	9
Competitive Analysis	10
UX Research	11
Ideation Phase	11
Design Phase	11
Building Phase	12
Design Process	13
Implementation	15
Technical Stack	16
Technical Stack Ramifications	17
Usability Testing	18
The Business	19
Next Steps	20

Acknowledgements

We would like to thank all the people who helped us make this project come true. Thank you to Professor Weber, whose insightful comments and guidance allowed us to concretize our idea and direct us towards what Gordy is today. Thank you to the people who stopped and talked to us outside of grocery stores, giving us comments and tips. Thank you to Alon Salant, who guided us towards finding the target users. We want to thank all our friends who helped with user testing, going through some scrappy user experience, and providing invaluable feedback. Finally, thank you to the MIMS community for the support and fun provided over the year.

Abstract

From our research, it is clear that people would like to cook at home more than they currently do, but are unable to commit to this goal. Why is that? Cooking is burdensome. From meal planning to cleaning the dishes, the process is strenuous. Yet, the desire for people to cook more than they currently do is promising. If we made the cooking process less demanding, people might have an easier time committing to their resolution.

With this in mind, we designed and built Gordy. Utilizing both voice user interface and traditional phone user interface, Gordy minimizes the burden that comes from following recipes in the kitchen. A voice-activated cooking assistant, Gordy recommends recipes and guides the user through ingredients and recipe steps, making it easier to follow instructions, avoiding unnecessary steps, and minimizing multitasking.

The Challenge

The team got together to address one fundamental issue: making cooking a better experience for people. Covering different levels of cooking skills, the team members were well aware that there are still a lot of frictions and pain points with cooking. Lucia is an expert cook who has tried multiple products and services in the food space to help her with her cooking endeavors, and has ultimately been left dissatisfied with what is out there. A food lover and Blue Apron subscriber, Robin is intrigued by anything revolving around food and cooking. Avi, who is not a very mature cook, wishes that he would be able to cook more at home, but feels that the present solutions available in the market do not cater to someone in his circumstances. Looking to create something that not only solves their own problems, but also makes cooking easier for others, the team started brainstorming around this problem statement.

Making cooking a better experience is a broad problem statement. There are many aspects of the cooking process that could be improved: from planning the right amount of groceries to buy, to having all the ingredients for the recipe you want to make; from being bored of the usual five recipes anyone can make, to being overwhelmed when following a recipe and suffering cognitive overload due to multitasking. We wanted to arrive at a more narrowly defined problem statement in a pragmatic way, conducting extensive market research to see what products and services out there are trying to solve.

We found many apps that help users plan for the week ahead, helping users choose recipes and their corresponding groceries, ultimately trying to make users better planners as opposed to day-of shoppers. While there was an opportunity for us to work into that field, we were more intrigued by solutions that would accommodate existing user behavior instead of changing it. With this new interest in mind, we realized there could be an opportunity in the kitchen, where users, who are often feeling overwhelmed from the required multitasking, are unlikely to change their behavior. How might we create a product that accommodates user behavior in the kitchen, minimizing cognitive overload and multitasking?

From our research, it is in fact clear that, when following recipes:

- Users find the steps to be too lengthy, not easy to remember, nor intuitive. Thirty percent of a user's cooking time is in fact spent looking at a screen following instructions.

- Users find it painful to switch between cooking and reading continuously, making their phones greasy or having to constantly wash their hands.
- Users find it hard to involve others, especially kids.

The Solution

Through Gordy, users can forget about multitasking and cognitive overload. Gordy utilizes both voice user interface and traditional phone user interface to support a seamless experience for the user. By guiding users through voice, Gordy allows users to focus 100% on the cooking, making it easier to follow instructions and avoiding unnecessary steps. Through the phone app, Gordy also visually informs the user, providing extra help when needed.

Product Description

Gordy is a voice-activated cooking assistant that recommends recipes and guides the user through ingredients and recipe steps. Gordy is activated through Google Home and is then paired with a phone app, so that, as the user communicates with Gordy, she can follow the steps both through auditory and visual cues. As the voice assistant reads the recipe steps, the visual app provides extra help to the user, showing recipe photos and ingredients, linking with the voice assistant and automatically moving to the next steps. The auditory-visual combination ensures that the user can easily follow the recipe, keeping the experience hands-free, and allowing her to focus on what is important: having fun while cooking.

Product Features

Voice User Interface

The main features within the voice interaction can be divided in two categories: recipe selection, and recipe realization.

- In the recipe selection process, the voice interface provides different categories through which the user can choose her recipe: meal, course, or dietary restriction. The user is then funneled through other layers of categorization to get to the desired type of food. The voice interface then reads the list of recommended recipes to the user.
- In the recipe realization process, the voice interface allows the user to hear each recipe step, go forward or backward in the steps, hear each recipe step again, and hear the ingredients relevant to each step.

Phone User Interface

Mirroring the voice user interface, the main features of the phone app can be divided in two categories: recipe selection, and recipe realization.

- In the recipe selection process, the phone interface provides different categories through which the user can choose her recipe: meal, course, or dietary restriction. The user is then funneled through other layers of categorization to get to the desired type of food. The phone interface then provides the list of recommended recipes to the user.
- In the recipe realization process, the phone interface allows the user to get an overview of the ingredients and the steps necessary, read each step in tandem with seeing its explanatory picture, and go forward or backward in the recipe steps.

Some of the functionalities between the voice user interface and the phone user interface are repeated to ensure that the user can easily follow the flow and choose the medium she prefers to interact with Gordy.

Value Proposition

Gordy provides our users three main values: reducing multitasking and cognitive overload, improving recipe understanding, and making cooking more social.

Gordy allows users to minimize multitasking, and, as a consequence, avoid cognitive overload. From our research, it is clear that users think cooking requires a lot of multitasking, especially when following recipes and having to continuously switch between the cooking task and the reading task. Multitasking leaves the user overwhelmed: cooking and reading are, in fact, both high-intensity skills. Gordy's voice assistant helps the user forget about the reading task. The user can now simply listen to the directions (a low-intensity skill) and follow them, without having to multitask as much.

Gordy improves recipe understanding. Interviewing people who use recipes while cooking, we realized that a lot of time is spent reading and re-reading the same step. Often, recipes are unnecessarily long, requiring users to actively search for the important information, or overly complicated, requiring users to decipher the directions, thus making the reading task even more burdensome. Gordy has been designed to minimize the length of the steps and to decode the

complicated directions for the user. The user can now listen to instructions that are easy to follow and easy to remember.

Gordy also allows users to make cooking more social. Currently, users are too busy multitasking that they have no time to also think about the social aspects of cooking. Users often find it hard to follow a recipe and, at the same time, coordinate and get others involved in the process. There needs to be constant communication between each individual in the kitchen about which step of the recipe they are reading. This becomes even more complicated when kids are involved, as they are often not good readers and need extra help from the adults. Thanks to the voice assistant, users can now better collaborate in the kitchen, following the steps linearly as the voice app suggests and keeping everyone on track.

Market Research

Industry Trends

Analyzing the current industry trends, we believe Gordy comes at the perfect time to be successful and to positively shape the current market. There are two main trends for us to focus on, voice assistant development and cooking industry trends.

Voice technology has been around for many years, but it was not until 2017 that there has been significant user adoption within this space. The market reach of voice-enabled smart speakers in the last three years “is outpacing the adoption rates of smartphones and tablets,”¹ with predictions of a 55% US market reach by 2022.² Even more promising is that 22% of users keep their voice-activated speakers in the kitchen, and that 56% of people who do grocery shopping online use voice-activated speakers to do so.^{3,4} These statistics demonstrate that voice-activated speakers are becoming increasingly popular, and that they are already perceived as a helpful product in the kitchen. Voice technology can liberate us from our screens, and people seem excited about it.

Cooking apps have also been flourishing over the past few years. That is thanks to the trends that have been developing in the cooking and food industry. Millennials cook at home more than baby-boomers, and plan on doing so even more, being “twice as likely as their older counterparts [...] to make this a resolution for 2017.”⁵ Even more promising is that 59% of millennials already cook with the help of their smartphone or tablet.⁶ With a smartphone or tablet in the kitchen, using voice search becomes easy and natural, with 23% of adults using it while cooking. It also appears that millennials like cooking with others: “27% said they were likely to be sharing the

¹ Ong, Thuy. “39 million Americans reportedly own a voice-activated smart speaker.” *The Verge*. Jan 15, 2018.

² Perez, Sarah. “Voice-enabled smart speakers to reach 55% of U.S. households by 2022, says report.” *TechCrunch*. Nov 8, 2017.

³ Kleinberg, Sarah. “5 ways voice assistance is shaping consumer behavior.” *Think with Google*. Jan 2018.

⁴ Jeffs, Mike. “OK Google, Siri, Alexa, Cortana; Can you tell me some stats on voice search?” *Branded3*. Jan 8, 2018.

⁵ “Survey: Millennials Cooking at Home More in 2017.” *Specialty Food Association*. Jan 4, 2017.

⁶ Cooper, Jenny. “Cooking trends among millennials: Welcome to the digital kitchen.” *Think with Google*. June, 2015.

experience with a spouse, friend, or child.”⁷ These trends seem to go hand in hand with our value propositions and our own research, placing Gordy in a booming and promising market.

Competitive Analysis

Within the voice-assisted cooking market, Tasted and Google Home’s embedded cooking features are Gordy’s primary competitors. Tasted, similarly to Gordy, uses a phone app and the voice assistant to help the user in the cooking process. Google Home relies exclusively on voice, without the visual aid.

Google Home clearly has a strong strategic advantage in brand awareness and market strength. However, from our research, it is clear that the lack of visual aid hinders the ability for a user to effectively search recipes without getting frustrated and smoothly going through the recipe steps without getting lost in the overarching flow. A phone app with visual support is clearly a must-have.

Tasted combines both auditory and visual cues, which users seem to find functional and understandable. However, users complain about the often unnecessary use of voice. Especially during recipe selection, touch is often friendlier and more informing than voice, as it provides pictures and the option to quickly glance through things. To interact, Tasted uses voice exclusively, transforming the phone app in a dumb screen that simply transcribes what the voice-assistant says. This reduces functionality and does not take full advantage of the powerful combination of voice user interface and traditional phone user interface.

We have brought functionality and usability to the next level: users can use either voice or touch depending on the convenience of the moment. From our research, it is clear that recipe selection is easier to do through touch, and recipe realization is easier to achieve through voice; we are optimizing for ease of performing an action, thus reducing pain points for the users. Furthermore, we provide “how-to” photos, a key component of our phone app, in each of the recipe steps: we noticed that users tend to look at the screen when they find the instructions through voice confusing. Photos eliminate potential doubts on how to perform a step.

⁷ Cooper, Jenny. “Cooking trends among millennials: Welcome to the digital kitchen.” *Think with Google*. June, 2015.

UX Research

User experience research was fundamental throughout the process, from the ideation phase to the design phase, to the building phase. We used one of the common methods in the field, user interviews, to gain empathy and understand pain points and needs of our users.

Ideation Phase

Within the ideation phase, talking with users was fundamental in understanding what parts of the cooking process, from buying groceries to cleaning up, were most painful to the user. Within this phase, we surveyed people at grocery stores, those who had an ingredients list and those who did not. We chatted with undergrad students at UC Berkeley and co-workers from our previous jobs. We talked with a leading food tech company in the Bay Area to further refine our ideas and narrow down our scope. We gained insights within expat communities with highly celebrated cuisines, specifically Korean, Indian, and Italian. We gathered a lot of information, both from randomly chosen people and segmented groups. Through this initial research, we realized that there were a lot of cooking apps focusing on the grocery and planning phases, which, however, did not seem to be the most painful phases for our users. Through our interviews, we realized that following recipes was incredibly hard. Users' pain points became apparent: people struggled to multitask, had trouble understanding steps, and found it hard to cook with others, making cooking a solitary activity.

Design Phase

We now knew we wanted to focus on the issue of following recipes and what users' pain points were. We started brainstorming what could help us solve some of these pain points and landed on voice-assisted cooking. Within the design phase, talking with users allowed us to refine our thoughts and develop a prototype. We wanted in fact to understand whether voice-assisted cooking was the best option, and whether having a visual assistant could still be helpful. Through these interviews, we came to the conclusion that voice is the best support for the cooking phase, while touch is extremely helpful in the recipe selection phase. While we now knew which type of technology to use for each phase, we still needed to think about the actual flow and look of the

experience. This was the second part of the design phase, which required the development of a user interface to test and interact with, as outlined below.

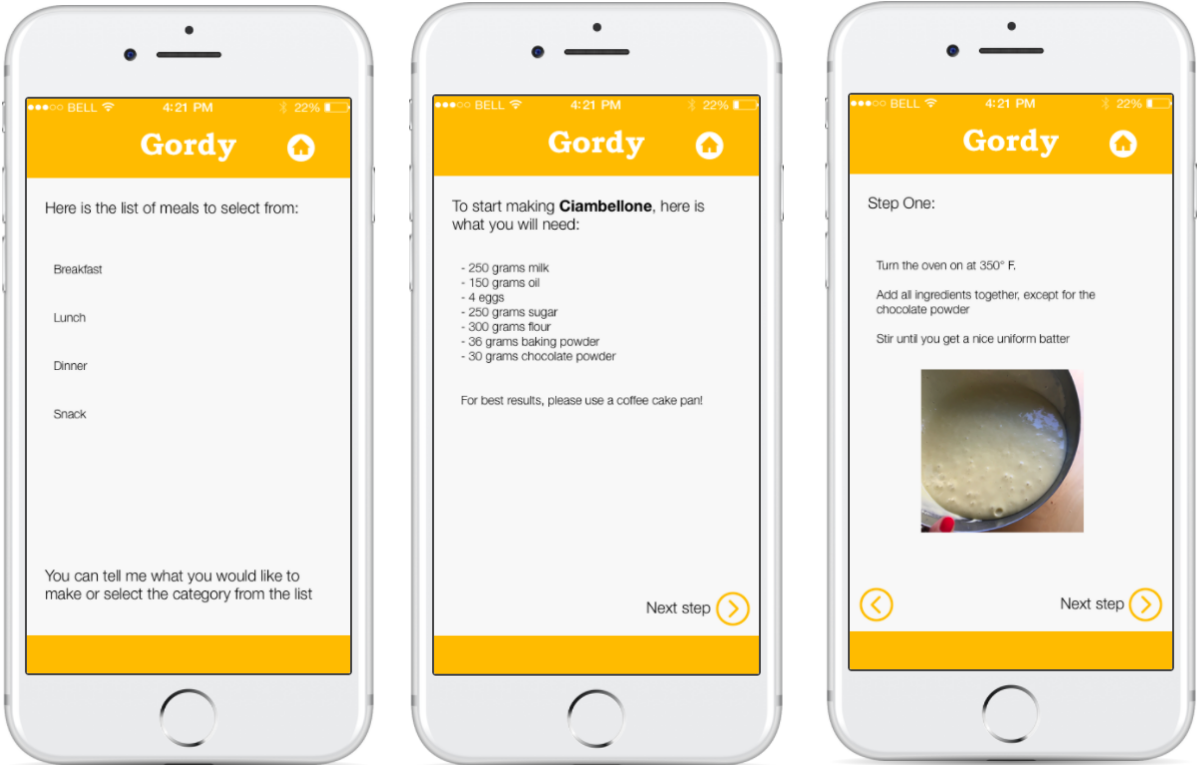
Building Phase

Within the building phase, engineering constraints always come into sight, and the design needs to be adapted to accommodate both the user and the technical limitations. For example, in the original app flow, the recipe steps overview would come together with the ingredients overview after the user said or pressed “start cooking.” However, given how the database was set up, this was technically impossible to do, so we re-designed the flow in a way that would work with our stack, and then showed it to new users to determine whether this flow worked for them as well. Many of these constraints presented themselves throughout the building phase, and required us to go to new users to test the changes we had implemented.

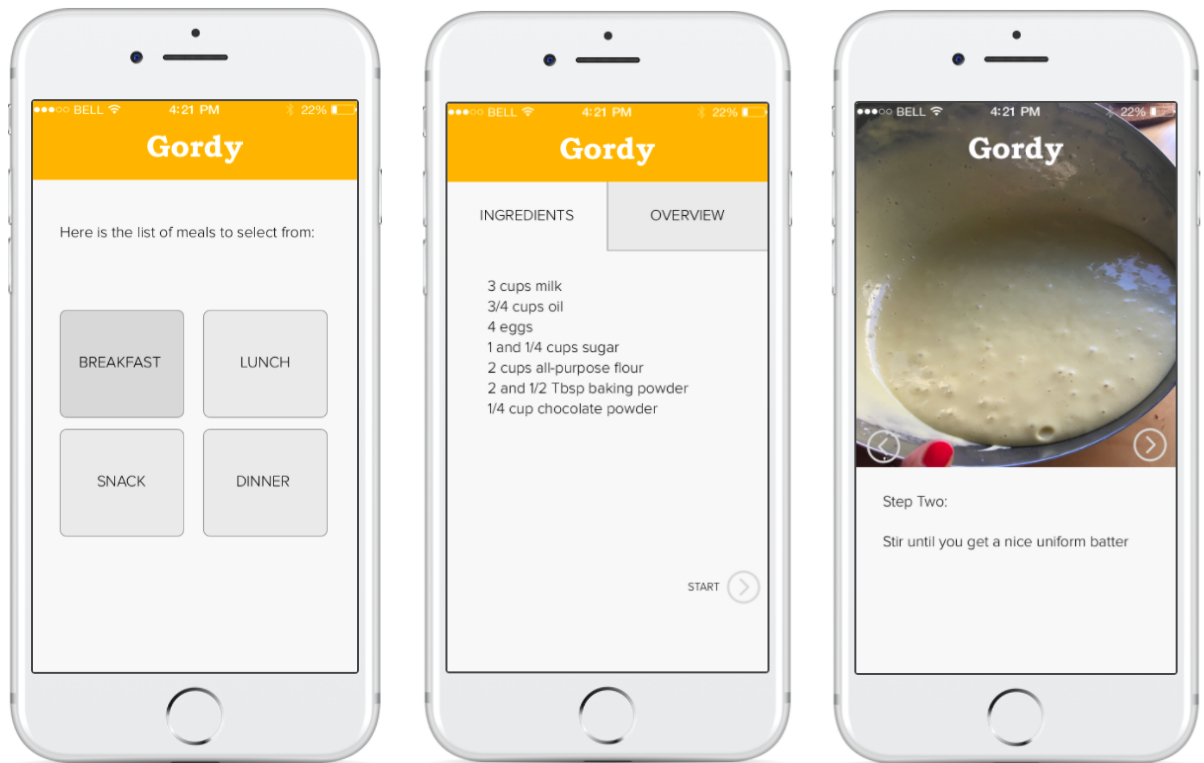
Design Process

As mentioned above, we went through many iterations of the app flow and look, based on the feedback we got from our users. We used Sketch not only to design our phone app, but also to visualize the voice commands and responses. While Sketch was born to create visual design, we found it useful in the voice UI space as well, because it allowed us to link each screen of the phone app with its corresponding voice commands and responses. Below are some examples of the screens.

First iteration



Second iteration



We then used the Marvel app to create interactive prototypes to show the users during our research. Having an interactive prototype was extremely helpful for user testing purposes as it allowed our participants to use our product in the kitchen while making a recipe, and clearly indicate to us where the problems were. The interactive prototype is available at this [link](#).

Implementation

There are two parts of the implementation that are intertwined. The first part is the voice. We use the Dialogflow API, which becomes routed to the Google Action Console. The Google Action Console is the Google Assistant software platform and is deployed in all Android phones and Google Home devices.

The Dialogflow's agent has a prepackaged natural language processing module that is very user friendly. It does not require learning an additional package, library or framework, and is very intuitive. The built in natural language processing allows our team to detect key words (and its synonyms), select phrases, and parse user sentences. It also tags the voice with an already generated set of tags (known as intents) that we can customize as needed. For simple responses, one can even use the web portal. For example, you can use the webportal if you want the Google Assistant to respond "I love you, too." We did create simple responses, but anything that requires complexity is fulfilled by our code hosted in Google Cloud.

In the Google Cloud engine that hosts our node.js code, there is a webhook that connects to Dialogflow. Therefore, any time there is a complex logic, it will be fulfilled by the Google Cloud, not by Dialogflow. Also note that there is a separation of concern as there is no natural language processing in our code in Google Cloud, which is only concerned with the intents, contexts, and actions that we have already tagged using Dialogflow.

The software hosted in Google Cloud connects with a NoSQL database in Firebase (owned by Google), Cloud Firestore, which stores all the recipe ingredients as needed. Images, however, are hosted on Google Cloud Storage (analogous to Amazon S3) and are served by Cloud Firestore as string of the url.

The phone application, created using React Native, makes a connection request to the Dialogflow endpoint to connect. It uses Android voice module and text to speech that is provided in Android, which is less sophisticated than the out-of-the-box Dialogflow module that Google Assistant is embedded with. Since the app goes through Dialogflow, it has the same user flow as the Google Assistant native app (via Android phones and Google Home devices). React Native must also pull separately from the database to allow the users to search through different recipes

and to select them as needed. This connection on Firebase is done asynchronously to speed up the app response time.

Technical Stack

Front End - React Native

The mobile front end was written in React Native, which can be deployed on both Android smartphones and iPhones.

Dialogflow API

We decided to use the Dialogflow API, and therefore the Google ecosystem, rather than using the Amazon Alexa SDK, for three reasons. First, Dialogflow becomes immediately available for all Android phones and Google Home devices, and is available to all Android-run smart devices (TVs, watches, headphones etc), so we felt that we could reach a larger audience in the Google ecosystem. Second, while the Amazon Alexa platform is more mature in its SDK and third party app development, we found Google's documentation to be much friendlier. Amazon Alexa has a number of teaching events and puts in a great effort to bring developers onto the platform, and yet, we felt that Google's onboarding was better (even after attending one of these events). Third, Amazon has a history of copying its independent software vendors within its ecosystem, and we did not want to put ourselves at risk.

Dialog Fulfillment - Node

The fulfillment code was written in Node.js and the backend database connection via Google Cloud device.

DialogFlow SDK & Google Homegraph

Dialogflow Software Development Kit is a hardware vendor friendly SDK that is often used to create dedicated third party vendor devices that run Google Assistant. For example, Bose headphones, Google headphones, and Sonos (in the future) use the Dialogflow SDK. We investigated the SDK and did not feel that this was necessary when we started the project.

Google Homegraph is also an SDK that is used by third party hardware vendors to connect with Google Assistant. For example, Phillips hue light or Nest is connected with Google Assistant via Google Homegraph. We also did not feel that this was necessary when we started the project.

Near the end of the project, however, given that we needed more sophisticated syncing methods between the phone app and Google Assistant, we delved into these SDKs further. Our next steps would be to investigate further whether these APIs solve our syncing needs that are currently unmet in the current iteration of the project.

Technical Stack Ramifications

Our team closely followed Google's recommended database and cloud platform because it integrated best with our hosted voice service. Had we chosen Amazon Alexa, we would have ended up with a technical stack that is best integrated with the Amazon Alexa platform, which would be the following AWS (Amazon Web Services) products:

- Alexa Skills (i.e. Dialogflow API in the Google ecosystem)
- Amazon S3 (i.e. Google Cloud Storage in the Google ecosystem)
- Amazon EC2 (i.e. Google Firestore in the Google ecosystem)

By providing natural language processing, cloud computing capability, database connection, and image store, our team is effectively "locked in" to the Google Ecosystem, and would have been had we chosen to go with Alexa services as well.

Usability Testing

As mentioned above, we did usability testing both in our design phase, through the Marvel app, and in our building phase, through the live version of our phone and voice apps.

In the design phase, we met our users in their kitchens and asked them to cook one of the recipes suggested by Gordy. We used the interactive prototype to recreate the phone app experience. One of us then pretended to be Google Home to recreate the voice experience, following closely the Sketch mockups previously made. While not a perfect simulation, user testing in the design phase was fundamental to understand the pitfalls in both our voice and phone experiences, and address those before we started building.

In the building phase, our product was practically ready to launch. We again met our users in their kitchens and asked them to cook one of Gordy's recipes. This time, both the voice and phone app were fully functional and could be used autonomously by the users. We took notes of users' behaviors and problems to make changes accordingly. User testing in this phase was extremely helpful as a sanity check. Luckily, only minor changes came out of these tests.

The Business

We are planning on launching Gordy on Google Home and the Google Play Store by the end of summer 2018. Our objective is to become the go-to app for people who use or would like to use recipes while cooking. As mentioned above, while there is some competition in the space, Gordy has superior functionalities that provide a better user experience. Our efforts then need to center around a powerful go-to-market strategy to show the quality of our product and acquire users.

Our primary target market is, firstly, consumers who own a voice assistance device, such as Google Home or Amazon Alexa. Our ideal user will also have a good level of technical inclination due to the advanced technological components of our product. In terms of cooking skills, we are targeting beginner and intermediate cooks. Experienced cooks are outside of our target market since they do not feel the pain points that Gordy takes away: experienced cooks do not perceive the stress of multitasking in the kitchen, are great at delegating work while cooking with others, and do not find current recipes hard to understand. This is not true for intermediate and beginner cooks, who have highly praised the benefits of using Gordy in our user interviews.

Our go-to-market strategy will focus on market penetration. Our initial user-acquisition strategy is two-fold. First, we will target users on social media through food tech influencers: we will specifically focus on Instagram, given its many food niches. Second, we will build a credible image within subreddit communities, such as [/r/52weeksofcooking](#), to find our first 1000 true fans; this particular subreddit is a dedicated and active community for its size, and it is the ideal testing ground. Our strategy is fully established in tech communities since we are targeting users who use and enjoy technology.

In terms of revenue streams, our product will follow a subscription business model. There will be free monthly trials for the consumers, followed by monthly fee charges thereafter. This revenue model is highly used in the digital services environment, ensuring that we can establish a relationship with potential paying customers.

Next Steps

As we think about launching Gordy, the most important task at hand is to grow our recipes database. While building Gordy, we decided to develop a limited number of recipes. We in fact only needed a few to test whether our app was functional. However, the more recipes Gordy can help with, the more beneficial it will be to our users. For our first few recipes, we went through a manual process of re-interpreting the recipes to make them accessible via voice: 1) breaking up the steps in more bite-sized pieces; 2) using words that are more understandable via voice; 3) shortening the overall length of the steps as much as possible. This was because existing set of recipe databases that we can easily acquire did not fit our needs for voice. This process is lengthy and would not allow us to scale quickly. We are now brainstorming on how we can build to scale this process (crowdsource, clean existing data sets, etc.).