Team Name	Let's Code Djibouti
Members	Harsh Deshpande Jeffrey Lucca Kyle Bremont Nasurudin Furi Robert Gomez William Ness
Description	Our product is a recipe sharing web application. The user will have an individual login that upon entrance to their account will allow to them to browse a public collection of recipes submitted by other users. Upon finding a recipe that they like, they can save that recipe to their personal account so that they can find it again with ease in the future. Additionally, the user can upload their own recipes to the public collection. They will be prompted to enter all necessary information including the following: ingredients, recipe directions, and approximate time necessary to complete. Another feature the user will be able to access is to upload recipes that do not need to be added to the public collection and can be saved solely to their personal account or shared to another user's account such that only the privileged users can view the recipe. This product will have the benefit of allowing a one stop collection for any recipe that a user has ever or would ever like to cook straight from the kitchens of others who have already tried them. Additionally, it allows for an online collection of "family recipes" so that if someone had a "secret family recipe" that they only wished to share with a select number of people they could do so without having to rely on being there in person. As a result, a user can have an experience as personal as cooking something from their traditional family background or as diverse and open ended as exploring recipes others have found interesting. In total, the user will be given access to a web application that connects them to recipes others have as well as be able to have a digital cookbook at the click of a button that will only grow as they upload new recipes and explore new recipes from others.
Vision Statement	We want to make a recipe sharing app that lets people be a chef in their kitchen, no matter how bare it might be, and share their recipe inspirations to friends, family, and others around the world.
Motivation	As college students, many of us are having to learn how to cook our own food for the first time. It can be tough to navigate through all of the grocery shopping, utensils, and techniques required to make something healthy and tasty. Our app aims to help simplify that process by documenting straightforward recipes for people to try and share with each other. By having the opportunity to quickly search for their favorite foods, users can enjoy different takes on whatever they might be craving rather than having to buy and leaf through a bulky recipe book. Ingredients are stated clearly for each recipe, giving users a pain-free way to create their shopping lists. In addition, the ability to share personal recipes with others allows users to share their culture, special tricks, or family traditions, growing their passion and skills in cooking in the process.
Risks	1. One risk we could encounter in this project is the difficulty of getting enough data to test our app. Since we'll be loading in recipes, each datapoint (recipe) will need tons of information (ingredients, instructions, etc). 2. Another risk we're probably going to encounter is that our schedules barely line up each week, and different exams and presentations and such will take members out of some meetings. 3. We're particularly weak with experience with frontend technologies.
Risk Mitigation	1. One way we could fix this is once we decide what datapoints we're collecting for each recipe, each member of the team could spend a couple hours getting 20 recipes each, and then we'd have enough for the entire semester. Another option would be to create a simple program to create recipes using random names, ingredients, and instructions. An advantage of this is we could use this to make as many datapoints as we wanted. A

	disadvantage of this is that those recipes wouldn't make any sense when it comes to actual recipes. 2. By using Slack, we can discuss what needs to get worked on and organize submeetings throughout the week with subsections of the team, so that way we can limit our whole-team meetings to short big-idea talk. 3. We've chosen and will continue to choose technologies we're either comfortable with, or we know there'll be a lot of online resources for so we can learn easily.
Version Control	We will be using GitHub for our version control. Our repos can be located at http://www.github.com/WillNess210/SoftwareDev2018 (Source Code), http://www.github.com/WillNess210/SoftwareDev2018Milestones (Milestones), and http://www.github.com/WillNess210/SoftwareDev2018Logs (Meeting Logs)

WillNess210 / **SoftwareDev2018Logs**   👁 Unwatch ▾ 1   ★ Star 0   ⑂ Fork 0

<>Code   ⓘ Issues 0   Pull requests 0   Projects 0   Wiki   Insights   ⚙ Settings

Options	**Collaborators** Push access to the repository
Collaborators	Jeff Lucca — Naby ✕
Branches	Nasuru ✕
Webhooks	robertngomez ✕
Integrations & services	Awaiting contactrmehra's response Copy invite link ▾ Cancel invite
Deploy keys	Awaiting kylebremont's response Copy invite link ▾ Cancel invite
Moderation	Harsh Amol Deshpande — Awaiting harshdesh7's response Copy invite link ▾ Cancel invite
Interaction limits	Awaiting alanparadise's response Copy invite link ▾ Cancel invite

Development Method	We will use Agile to develop our project. We will have a weekly scrum to assign features to be worked on based on priority, difficulty, and time demand. Should someone finish their weekly task early they can progress down the list of features to be finished and begin work on one of those. We will be using an automated kanban board provided by GitHub to plan out the priorities and list what next steps need to be finished. Due to schedule conflicts with each of us, we may have to do separate meetings on some weeks rather than have one big 2 hour meeting, and we will give updates about each of these meetings on Slack and also by updating the kanban board. We will also note down meeting logs in Google Drive and will be added into the repos as well..
Collaboration Tool	We will be using a slack channel for general communication. All of our members are already in the Slack channel, and we will integrate the GitHub app so we can get updates on what's being worked on.
Proposed Architecture	We will use Python with Flask on the backend, PostgreSQL as the database, and Angularjs on the front end. For linking the backend and the integration layer, we will be using a REST api . Our integration level will be using Node.JS possibly with modules like Express to set up the integration layer architecture and Request to communicate with an API to bridge the gap between our frontend and backend. Bootstrap will be used to help create the aesthetic features of the frontend.