



**University of Arkansas – CSCE Department
Capstone I – Final Report – Fall 2019**

Ubiquitous Cooking

Daniel Perez, Manuel Serna-Aguilera, Jack Scholes, Alejandro Urquieta, Guadalupe Torres, Jean Pierre Roca

Abstract

Cooking a new recipe for the first time takes much more time and effort compared to a familiar recipe. To save more time cooking new recipes, we propose creating an application that aims to make cooking easier using a smartwatch. This application will instruct the user to cook new recipes while minimizing the learning curve to cook new foods. This approach will use a simple and intuitive interface such that the user can focus more on cooking and not rely on looking at videos or reading instructions. Our approach will allow more people to cook easily and not spend so much time overcoming the learning curve of cooking new recipes.

1.0 Problem

Cooking a new recipe takes a significant amount of time, and not everyone, especially those with a limited schedule, may be able to set aside the time to cook. Time could potentially be saved and the obstacle for cooking new recipes could potentially be reduced. Should we not implement any solution, the obstacle for people trying brand new recipes remains where it is and potentially intimidating for some people. People may resort to other, less healthy food options if they do not cook or simply not expand their palate.

2.0 Objective

The objective of this project is to provide an easy-to-use recipe Wear OS smart watch application for people to save time cooking. With this app they can improve their cooking skills and mitigate the learning curve that comes with new recipes.

3.0 Background

3.1 Key Concepts

Wear OS: Google's Android operating system designed for smartwatches and other wearables. Our smartwatch application will be developed as a Wear OS application for Android smartwatches.

Android SDK: Comprehensive set of development tools which include a debugger, libraries, a handset emulator, documentation, sample code, and tutorials. This framework will allow us to develop both the Wear OS and Android Application components of the project.

Database SQL: SQL Database supports storage and manipulation of data. We will employ an internal SQL database on the phone side which will store recipe data and send it to the smartwatch app.

Spring Boot: Server-side Java framework for REST requests. Will be hosted on Heroku, a site that hosts project servers.

Dependency Injection: Ideology that abstracts the configuration of objects and “injects” them where they are needed, configured and ready to go.

JPA: Java Persistence API used for configuring databases.

3.2 Related Work

The capability of using a smartwatch to cook still seems to be something developers have not successfully set in motion yet. There are smartphone applications like the Tasty app that has a step-by-step instructions mode that shows you how to cook your chosen recipe and it keeps your phone awake so you don't have to worry about it falling asleep while you cook[1]. It is almost impossible to stand in one place while you cook, and this app requires your smartphone to stay in one place if you want to use both hands to cook. The smartwatch application could improve the mobility of viewing your recipes. Instead of going to and from your phone, you can view your recipe on your wrist at all times. This could save you time in the kitchen, which is the overall goal for this project.

There is another application called Yum-Yum! that allows you to view recipe steps, use a grocery list and the timer on an Apple Watch[2]. By testing the app with a personal smartwatch, we observed that this app was not loading any of the recipes. Then, we realized that the last update for this app was in 2015. The last update includes the new smartwatch feature, but it doesn't seem like it was a successful launch since it was having trouble connecting its server. For our project, we hope to create a usable app for android smartwatches, not the Apple Watch. The majority of the team has experience with android development, so we decided that it would be easier to implement our project using an android.

4.0 Design

4.1 Requirements and/or Use Cases and/or Design Goals

Application Requirements

- Ability to access recipes from a database
- Ability to retain certain recipes from database for quick access
- Ability for the user to create a recipe checklist

- Executing on a recipe checklist in a clean, UX friendly manner
- Ability to fluently to navigate between each step of the recipes
- Ability to create a timer on certain steps
- Ability for users to upload recipes to the database
- Possible implementation of voice commands

Use Cases: We want our use cases to be universal towards all people who would cook that own a smartwatch. It should be able to guide any user through the cooking process beginning with buying ingredients and ending with an easily prepared meal.

4.2 High Level Architecture

Client - MVVM

The client application will implement a Model-View-ViewModel architecture to implement separation of concerns across the application. This will help assign team members to roles that will allow them to write code without breaking someone else's by decoupling as much as is reasonable.

Model

The Model code will be the low-level details of the application: database management, facilitating network requests, error handling/wrapping.

ViewModel

The ViewModel is the bridge between the Model and the View, that is, for every View file there will be one corresponding ViewModel. It is supposed to abstract the complexity of the Model classes and simply serve data for the View to use and display.

View

The View will be responsible for what is shown to the user and showing them their own recipes, recipes they can download from the server, and the “step-by-step” process of the application.

4.3 Risks

Risk	Risk Reduction
Intellectual Property	Have the user agree to a terms of use, which will include us, the makers, not being responsible for any recipes uploaded. Users upload recipes at their own risk.
Damage to smartwatch	Disclaimer that we are not responsible for personal damage while the user cooks.

4.4 Tasks

1. Create user stories and then assign roles to each team member.
2. Implement MVVM architecture on the client quickly so people can begin assume their roles fast and without breaking everyone else's code.
3. Understand and design the schema we need. With Spring and Wear OS, leverage the fact that we are using Java on both the client and server for JSON serialization/deserialization.
4. Implement server code using Spring and JPA for database connections.
5. Understand how to create and use a local database and abstract its complexity in a way that makes sense for the client architecture.
6. Understand how to make network requests to the server and abstract their complexity in a way that makes sense for the client architecture.
7. Feature Freeze, clean everything and refine the User Interface and User Experience and clean any lingering non-reusable, non-scalable code.
8. Test the application and try to make everything that can go wrong, go wrong. Refine accordingly.
9. Write unit tests for our code using JUnit to make sure that our code works as intended.
10. Create a wiki for the codebase explaining what each package does and how it relates to the greater problem (reduce the learning curve to cook new recipes).

4.5 Schedule

Tasks	Dates
1) <ol style="list-style-type: none"> a. Determine the services needed by a server. b. Create user stories, Discuss/implement project architecture. c. Start documenting process. 	1/20-2/3
2) <ol style="list-style-type: none"> a. Create database schema b. Set up architecture for smart watch application. General application configuration. 	2/3-2/17
3) <ol style="list-style-type: none"> a. Create RESTful services on the server (Create, Read, Update, Delete). 	2/18-3/1

b. UI/UX begins implementation, “backend” of app begins to abstract work and data processing for the View to display.	
4) a. UI/UX sets up more screens and refines the user experience. Everything should be set up to plug in networking next sprint with the view code having no knowledge of the origin of the data (network or local database).	3/2-3/16
5) a. Spring Break, refine and clean any “hacky” code that has accrued, if needed.	3/17-3/29
6) a. Connect client and server. Write unit tests against critical code for the app. Stress test every worst-case scenario possible and clean up crashes.	3/30-4/13
9) a. Refine against focus groups and see where improvements can be made for users.	4/13-4/27
10) a. Finalize documentation, refine any rough edges.	4/27-5/4

4.6 Deliverables

- **Preliminary report:** This report will contain our initial thoughts as to what we envision the project to look like. This document will also serve as a reference document so we can look at our schedule, project components, proposed test cases, etc.
- **Database schema:** The server that will contain starting recipes and store recipes for each client. This database schema is java code for a Spring server which will be housed in Heroku. This schema will be stored in a .jar file.
- **Server code:** This will be a Spring server implemented with Java.

- **Android project:** This Java code will implement the Model-View-ViewModel design of our project. This project will run on a Wear OS watch and an Android phone.
- **Final report:** A final evaluation of our completed project.

5.0 Key Personnel

Daniel Perez – Perez is a senior Computer Science major at the University of Arkansas. He was a Walmart Cybersecurity Intern in Summer 2019. He is taking Mobile Programming which uses the Android SDK. He is responsible for the application architecture and watch implementation along with Serna-Aguilera.

Manuel Serna-Aguilera – Serna-Aguilera is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Programming Paradigms and Software Engineering. He was a software engineering intern at LiveRamp in the summer of 2019. He is working alongside Perez with the architecture.

Jack Scholes – Scholes is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Programming Paradigms and Software Engineering. He will be responsible for database implementation and connections.

Alejandro Urquieta – Urquieta is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas and is also pursuing a Mathematics minor. He has completed Programming Paradigms, Software Engineering, and Mobile Programming. He was a software engineering intern at PricewaterhouseCoopers LLP. He will be responsible for the front-end implementations shown on the UI along with Torres.

Guadalupe Torres – Torres is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed Paradigms, Software Engineering, and Mobile Programming. She has intern experience with Tyson Foods Inc. and Phillips 66 in IT. Torres will be responsible for the front-end implementations shown on the UI.

Jean Pierre Roca – Roca is a senior Computer Science major and Mathematics minor in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed all major courses required for his degree and has had internship experience in software engineering from ABC Financial and J.B.Hunt. Having experience with backend, Spring, and databases, Roca will be responsible for the server side code and database connections.

6.0 Facilities and Equipment

Android Compatible Smartwatch- will be used throughout the development of the smartwatch component of the app required for downloading and viewing recipes while cooking.

Android Smartphone- will be used for testing throughout the development of the smartphone component of the app which will have all the functionality of the smartwatch app, but also include the capability to create and submit recipes for smartwatch users.

Heroku Account- will be used to host our server and database when we finalize local development and are ready to deploy our application.

7.0 References

[1] Tasty on the App Store, <https://apps.apple.com/us/app/tasty/id1217456898>

[2] Yum-Yum! on the App Store,
<https://apps.apple.com/us/app/yum-yum-1000-free-recipes-step-by-step-photos-grocery/id962021085/?platform=iphone>