Java Review

Software Design Document

Qowiyu Amuda and Jack Pearce

Information Systems Analysis and Design

12/14/23

Table of Contents

Introduction

The Java Review app redefines how coffee enthusiasts discover local coffee shops, emphasizing personalized recommendations, community engagement, and support for local businesses. The SDD serves as a vital blueprint, providing clarity in design, guiding development systematically, fostering collaboration, and ensuring a reference for future updates and maintenance. Its purpose is to give an overview of design choices, facilitate effective team communication, and establish a foundation for the app's long-term growth.

Purpose

The Java Review app addresses the challenges faced by coffee enthusiasts in discovering new and unique local coffee shops, providing personalized recommendations, fostering community engagement, and supporting the visibility and growth of local coffee businesses. It is designed for individuals who share a passion for discovering and enjoying coffee experiences.

Scope

The Java Review app is limited to iOS and Android platforms, focusing on coffee-related features like discovery, recommendations, and community engagement. The initial launch targets a specific geographic region (Decatur, IL), and core functionalities include user accounts, coffee shop data integration, personalized recommendations, and community features. The scope is limited due to resource constraints, aiming for a user-centric and agile development approach. The phased geographic expansion ensures controlled launches, and a focused scope allows thorough testing and quality assurance, enhancing app reliability within available resources and timelines.

Reference Material

Hoffman, J. (2021). What are the pros and cons of Kanban?. WisdomPlexus. https://wisdomplexus.com/blogs/pros-cons-kanban/

Schwaber, K. & Sutherland, J.(2020). The Definitive Guide to Scrum: The Rules of the Game.

Definitions and Acronyms

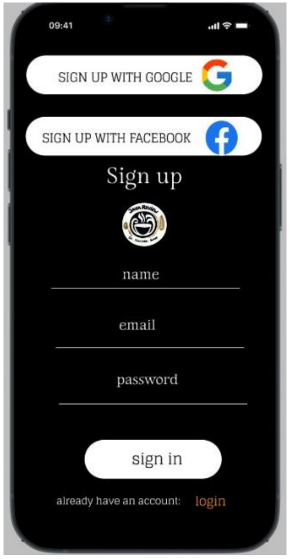
Full Term	Acronym/Abbrev.	Definition
Agile	N/A	A project management methodology which uses cycles of planning, building, deploying, and evaluating to build software or systems in an iterative and incremental fashion. Most used for software development but also used for other types of projects.
Software Design Document	SDD, SWDD, SDS	A design document used to describe the high-level (and some low-level) architecture and design goals & planning of a software project.
Software Development Lifecycle	SDLC	Umbrella Term for methodologies used to make a structured process for producing software systems and applications.

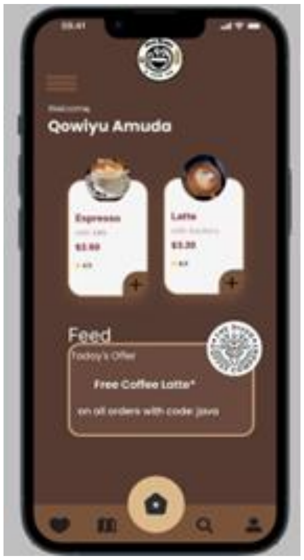
Structured Query Language	SQL	a programming language designed for managing and manipulating relational databases. It serves as a standard means of interacting with databases, enabling users to create, retrieve, update, and delete data. SQL is commonly used for tasks such as querying databases to retrieve specific information, inserting new data, updating existing data, and defining the structure of a database.

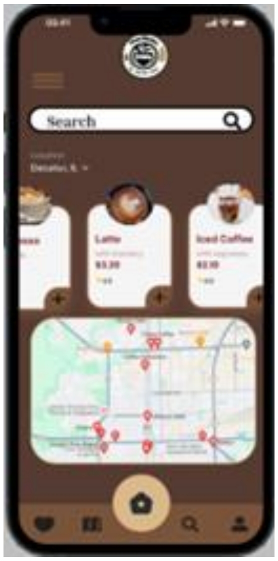
Human Interface Design

The Java Review app offers users a seamless coffee exploration experience with features like a map for discovering local coffee shops, personalized drink recommendations, customizable user profiles, efficient search, and a vibrant community space. There are five main screens available within the app including Home, Search, Maps, Profile, and Like posts. Users praise the app's simplicity and easy to understand, utility on an everyday basis for coffee enthusiasts, and sleek visuals that are pleasing to the eyes.

UI Design Images



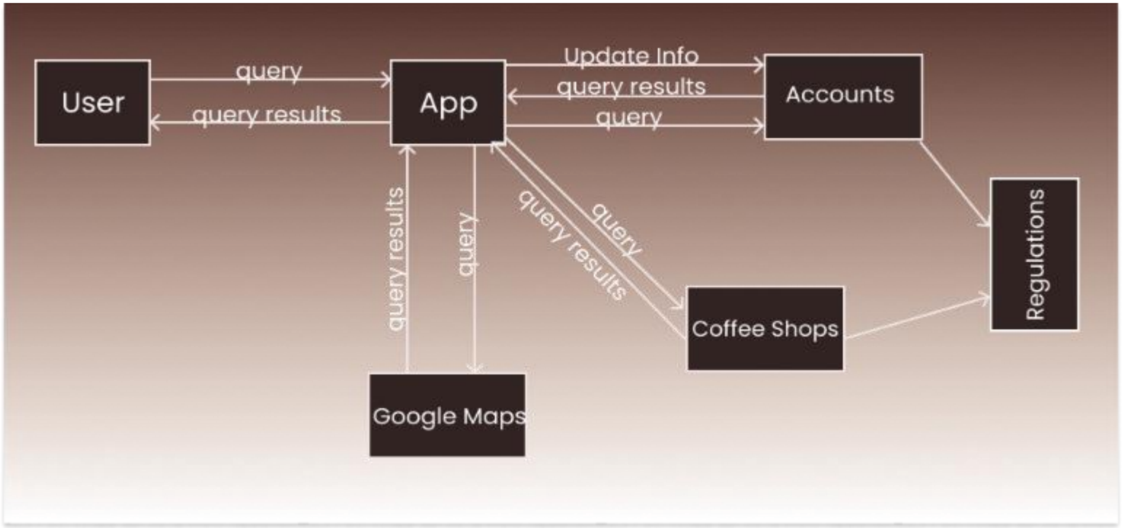




Notes:

- At the bottom of the page is the navigation bar. By clicking each icon, you can navigate to the any of the desired page. When the search icon is clicked, the user is navigated to the interface screenshot on the far right.
- The original design had the search icon/navigator to the far right. However, user feedback suggests that it is best to have the profile icon/navigator to the far right instead. So, on the above user interface, you can clearly see the search icon/navigator switched.

System Architecture



Epics and Use-cases

User Stories:

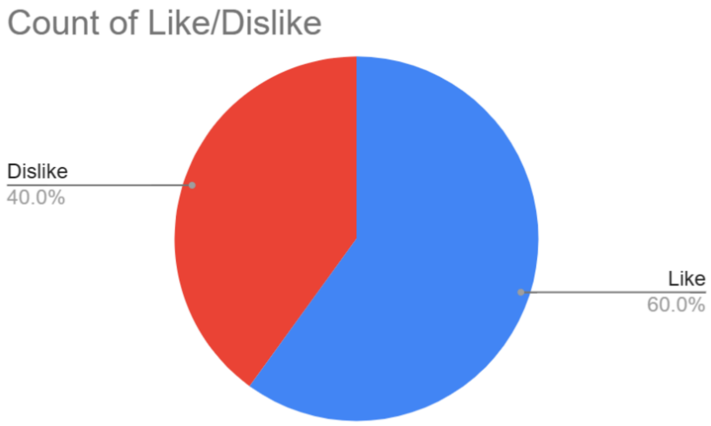
- As a coffee enthusiast, I want to easily discover new local coffee shops in my area, so that I can expand my coffee experience and try different places.
- As a regular user of the Java Review app, I want the app to provide personalized drink recommendations based on my preferences, so that I can quickly find and try drinks that match my taste.
- As a user with specific drink preferences, I want to be able to search for a particular type of drink in my vicinity, so that I can easily find coffee shops that offer my favorite drinks.
- As a regular user of the Java Review app, I want to save my favorite drinks and coffee shops to my profile, so that I can quickly access and share my preferred choices with others.
- As a Java Review app user, I want to leave reviews and ratings for drinks and coffee shops I've visited, so that I can contribute to the community and help others make informed choices.

Epics:

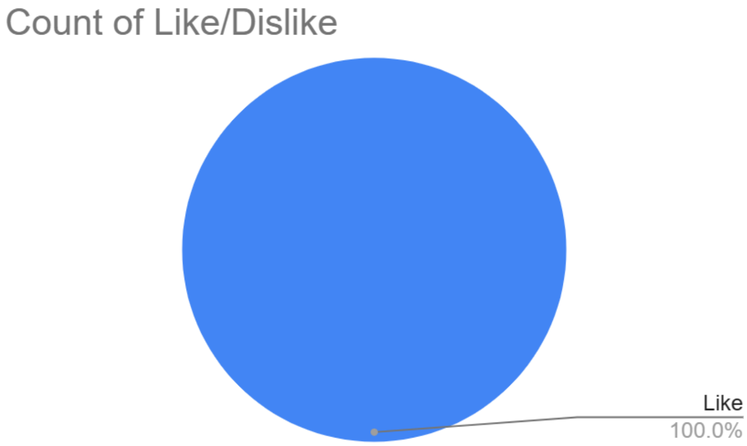
- Coffee Shop Database
- Accounts Database
- User Interface
- Accessibility
- Social Integration

Use-case Feedback Analysis

We did A/B testing for feedback. Difference between A and B is that the profile and the search icon on the main menu were switched. On A, the search icon was right most with profile to the left. On B, the search and profiles icons were flipped.



These were the results for Test A.



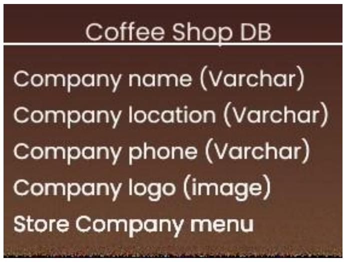
These were the test results for Test B.

Based upon the results, we decided to go with the design of B where the profile icon was right most with search icon to the left.

[Google Maps Integration Architecture]

Integrating Google Maps involves obtaining an API key, utilizing the Google Maps JavaScript API to embed interactive maps in a designated container. Key components include markers for location indication, info windows for additional details, and additional services like Geocoding, Directions, and Places APIs for location-based functionalities. Responsive design considerations and error handling mechanisms enhance the overall user experience.

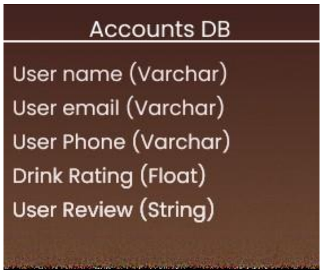
[Coffee Shops Database Data Design] (if data)



Coffee Shop DB

Company name (Varchar)
Company location (Varchar)
Company phone (Varchar)
Company logo (image)
Store Company menu

[Accounts Database Data Design] (if data)



Accounts DB

User name (Varchar)
User email (Varchar)
User Phone (Varchar)
Drink Rating (Float)
User Review (String)

[Google Maps Integration Rationale]

The subsystem design incorporates Google Maps due to its widespread user familiarity, comprehensive functionality, and strong community support. The decision prioritizes an intuitive user experience, efficient development, and reliable performance. Alternative options, including OpenStreetMap, Mapbox, and Bing Maps, were considered, each with tradeoffs related to customization, user familiarity, and global coverage. The chosen design aligns with project goals by emphasizing a seamless user experience while acknowledging tradeoffs in terms of potential costs and privacy concerns.

[Coffee Shops Database Rationale]

The Coffee Shops Database part was picked to organize information about local coffee places efficiently. We chose a standard way called a relational model with an SQL system because it's known to work well for this kind of job. It helps keep information in order, makes sure it's correct, and allows the system to grow smoothly. We thought about other ways, like more flexible databases, but they seemed unnecessary for our structured data about coffee shops. The tradeoffs involve balancing the structured approach with possible flexibility concerns and handling more complexity as the system gets bigger. Overall, our choice aims to make sure the coffee shop data is well-organized and can handle growth in the Java Review app.

[Accounts Database Rationale]

The Accounts Database subsystem was chosen to handle user information effectively in the Java Review app. We went with a familiar approach using a relational database model and an SQL system for reliability and scalability. This design keeps user data in order and makes it easy to manage as more people use the app. While considering other options like more flexible databases, we stuck with what's widely used and understood for organizing user accounts. The tradeoffs involve finding the right balance between structure and flexibility. Our decision aims to ensure user information is well-organized, secure, and ready for growth in the Java Review app.