INTRODUCING THE RESTAURANT REVIEW

- I built the Restaurant Review System because I wanted to create a review app that's simple, fun, and easy to use. Big apps like Yelp can feel too crowded or complicated, so I made something that focuses only on what users really need.
  - **Simple and Clean**
  - **I didn't want too many features. Just a clean app where users can rate restaurants, write reviews, and see what others think.**
  - **Fun to Use**
  - **I added things like star ratings, mood emojis, flame icons for popular foods, and fun usernames to make the app feel more lively and modern.**
  - **Built with Java and Swing**

- **I created everything myself using Java** and **Java Swing**. This helped me learn how real desktop apps are built, step by step.
- **Inspired by Real Apps**
- **I looked at apps like Google Reviews and Yelp for ideas but I made my own version that works as a smaller, desktop-friendly app**

Classes and OOP Structure Overview

Understanding the Key Components of the System

- **Reviewer Class**

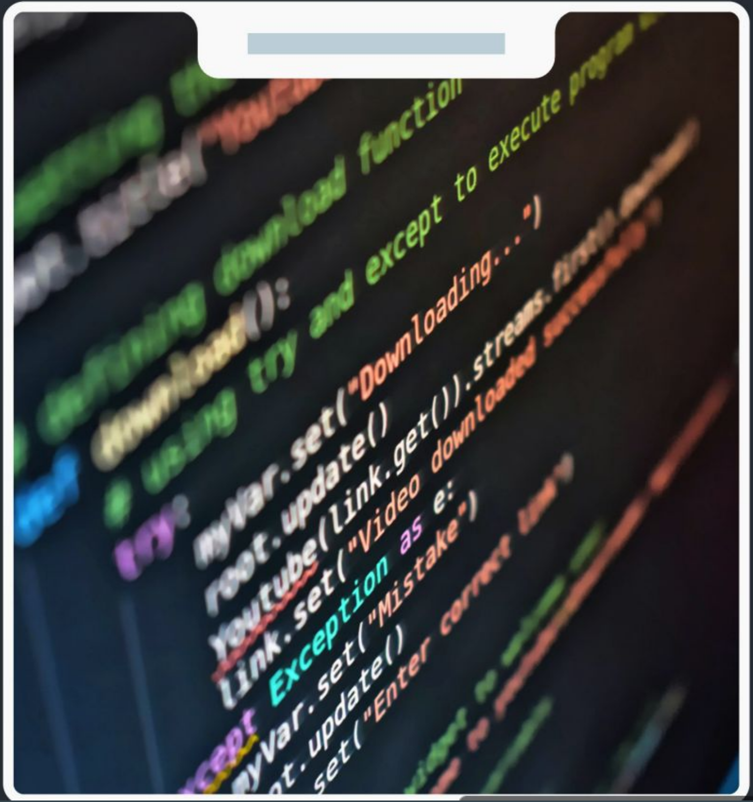
  Extends User class to manage and store reviews submitted by users.

- **Review Class**

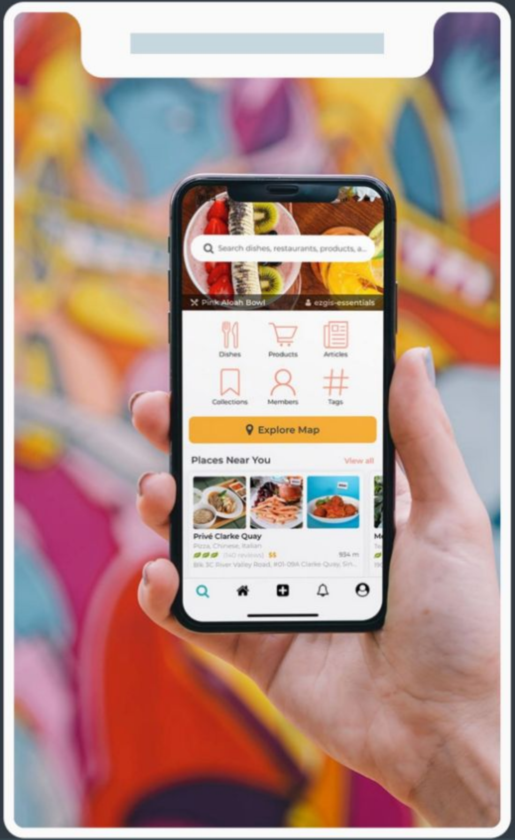
  Contains essential details such as restaurant, rating, and user comments.

- **RestaurantReviewSystem Class**

  Manages the user interface and overall program flow for the application.



🛠️ Features of My Review System



01 **🛠️ Features of My Review System**

This slide highlights the key features that make the review system user-friendly and engaging.

02 **🍴 Restaurant cards with star ratings, names, and emojis**

Each restaurant is presented with a card that includes its name, star rating, and fun emojis to enhance visual appeal.

03 **✏️ Edit button opens a popup window to write reviews**

Users can easily edit their reviews through a convenient popup interface.

04 **💬 Users can type their review and choose a star rating (1–5)**

The system allows users to provide detailed feedback along with a star rating.

05 **😃 Emoji changes based on the rating selected**

The emoji displayed will change according to the star rating chosen by the user, adding a fun element.

06 **🔥 Menu item ratings using flame icons (top 5 items)**

Highlighting the top 5 menu items with flame icons to indicate popularity.

07 **💲 Menu items also show prices**

Users can see the prices of menu items directly on the restaurant cards.

08 **🎨 Fake user reviews appear with colorful usernames**

To create a lively atmosphere, user reviews are displayed with vibrant usernames.

09 **🖤 Dark theme with a clean layout**

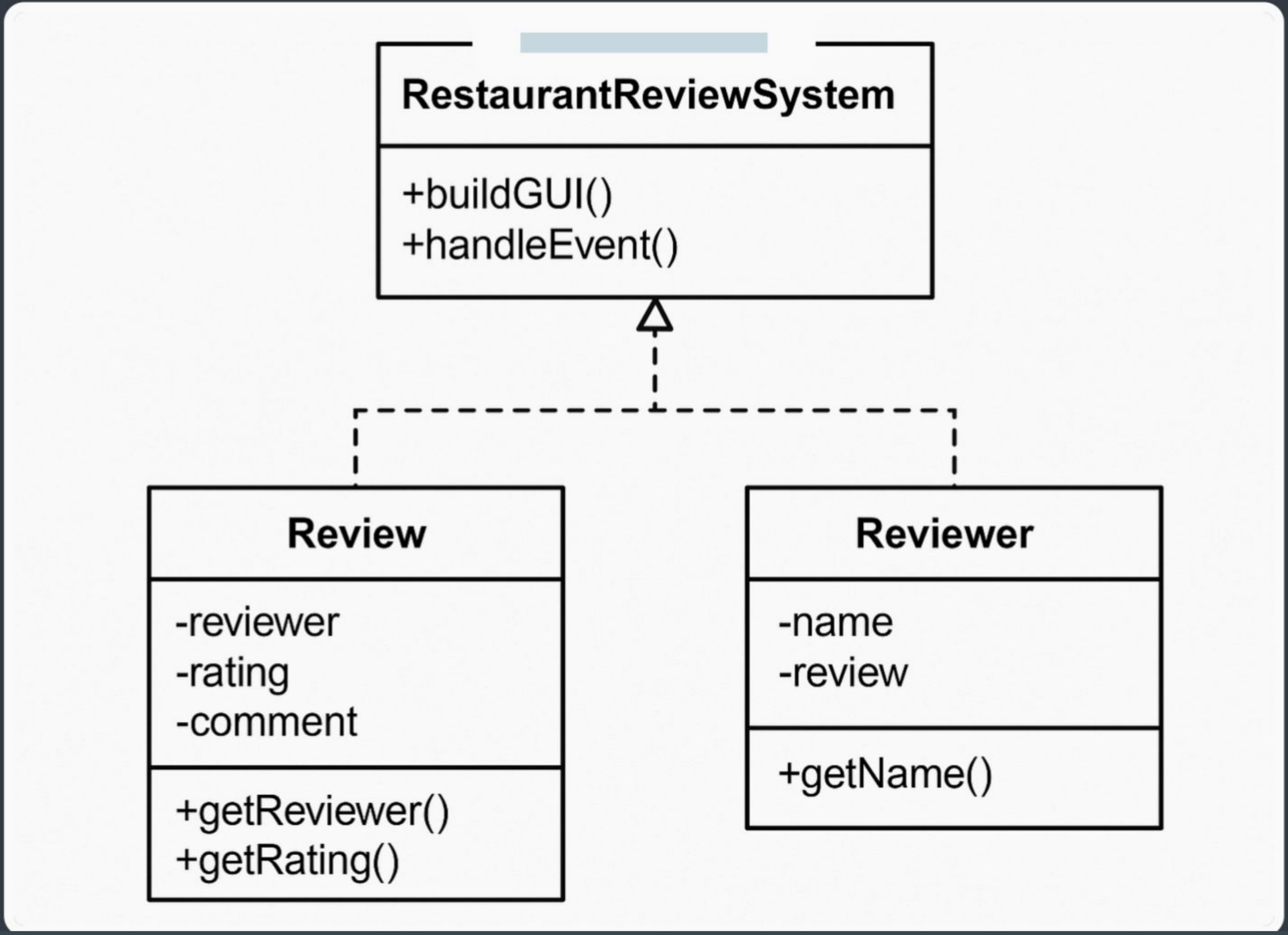
The application features a sleek dark theme that enhances readability and aesthetics.

10 **🔄 Sort dropdown to organize restaurants by name or rating**

Users can easily sort the restaurant list based on their preferences.

11 **👷 Entirely built using Java Swing (AWT + Swing components)**

The application is developed using Java Swing, ensuring a robust and responsive user interface.



UML Class Diagram Overview

Understanding Class Relationships in UML

System Design and UML Classes

- **Understanding Class Relationships in UML:**
- RestaurantReviewSystem → Main class that builds the interface and manages all event
- Review → Stores each review: reviewer name, stars, and comment
- Reviewer → Stores current user info and links to their review
- Other Notes: Data is stored using HashMaps for each restaurant
- Only one editable review per user per restaurant
- Everything is designed using object oriented programming

What I Learned from This Project

- How to create full GUI apps in Java
- How to use Java Swing components (buttons, panels, text boxes)
- How to organize code using OOP (classes and objects)
- How to use HashMaps and ArrayLists to store and update data
- How to handle user actions with event listeners
- How to design a user-friendly interface with good layout and visuals
- How to make apps feel realistic and fun with emojis, color, and interactivity