SW Engineering CSC648-848-05 Fall2023

Project: BiteRate
Team number: Team 5
Team Lead: Alec Nagal | anagal1@mail.sfsu.edu
Frontend Lead: Ali Alsharif
Backend Lead: Gerry Putra
Git Master: Gabriella Arcilla
Database Master: Kenneth Pang
Backend Assistant: Robel Ayelew
Milestone 1
Date: September 21, 2023

Checkpoint #	**Date Submitted**	**Revision #**
Checkpoint #1	09/21/2023	Revision #0

Table of Contents

Executive Summary

***Mission*:** BiteRate is an application designed for the way people discover, share, and enjoy their favorite dishes and restaurants. Dining is a central aspect of our lives, and making informed choices about where and what to eat can be a delightful but difficult task. BiteRate addresses this by providing a platform for users to rate dishes, share their culinary experiences, and receive personalized recommendations.

***Application Overview*:** BiteRate is a comprehensive food-focused application that caters to both seasoned foodies and casual diners. Here's how it works:

***Dish Ratings and Reviews*:** BiteRate allows users to rate and write reviews for individual dishes at restaurants. This micro-level approach ensures that users can share their opinions on specific culinary creations, helping others make informed decisions.

***Restaurant Insights*:** Beyond dish reviews, BiteRate provides valuable insights into restaurants. Users can access ratings for overall dining experiences, service quality, and more, ensuring they have a holistic view before making a reservation.

***Personalized Dish Recommendations*:** BiteRate employs a sophisticated recommendation system that considers a user's past dish ratings and dining preferences. It then suggests new dishes to try, allowing users to broaden their culinary horizons and discover more.

***Culinary Community*:** BiteRate fosters a thriving culinary community. Users can follow fellow food enthusiasts, share their gastronomic adventures through photos and descriptions, and even plan group outings to restaurants that match their taste profiles.

Key Advantages and Novelty
***Dish-Centric*:** BiteRate's focus on individual dishes sets it apart from traditional restaurant review platforms. Users can find the perfect dish, even if they're not familiar with the restaurant itself.

***Comprehensive Insights*:** The application provides a comprehensive view of restaurants, going beyond basic ratings to include detailed insights into various aspects of the dining experience.

***Personalization*:** BiteRate's recommendation system ensures that users receive tailored dish suggestions based on their culinary preferences, making every dining experience enjoyable and unique.

***Community Building*:** The app encourages food enthusiasts to connect, share, and explore together, creating a sense of camaraderie among users who appreciate good food.

***Value*:** BiteRate simplifies the dining decision-making process, promotes culinary exploration, and builds a passionate community of food lovers. By supporting this project, you're championing a platform that elevates dining experiences, supports local eateries, and connects people through their shared love for food.

In summary, BiteRate isn't just an app; it's a culinary adventure that transforms how people discover and enjoy dishes. We want to create a global community of food enthusiasts and enhance the way we savor the world's flavors, one bite at a time.

Main Use Cases

***User-Generated Restaurant Ratings and Reviews*:**

Scenario: A user named Sarah visits a new Italian restaurant in town. After her meal, she logs into the app and leaves a detailed review, including ratings for food quality, service, ambiance, and overall experience.

Use Case: Users can share their dining experiences and opinions by rating and reviewing restaurants they visit, helping others make informed choices about where to dine.

***Taste Compatibility Matching*:**

Scenario: John regularly rates restaurants and has a high taste compatibility score with another user, Emily. The app notifies John about their compatibility and suggests they connect.

Use Case: The app uses its algorithm to match users with similar taste profiles, fostering connections between food enthusiasts who are likely to enjoy dining at the same places.

***Personalized Restaurant Recommendations*:**

Scenario: Michelle logs into the app and sees a list of restaurant recommendations based on her taste compatibility score with other users. She decides to try one of the recommended restaurants for dinner.

Use Case: Users receive restaurant suggestions tailored to their preferences, making it easier for them to discover new dining experiences that align with their tastes.

***Social Dining Planning*:**

Scenario: A group of friends, consisting of Mark, Lisa, and James, all have high taste compatibility scores with each other. They use the app's group planning feature to organize a dinner outing to a restaurant with a high compatibility score for all of them.

Use Case: Users can create and join group outings with people who share similar tastes, making it convenient to plan and enjoy restaurant visits together.

***Foodie Community Building*:**

Scenario: Rachel follows several users with similar taste profiles, and they often share their favorite dishes and restaurant discoveries. Rachel decides to join a forum discussion about the best local brunch spots with her foodie community.

Use Case: Users can build connections with like-minded food enthusiasts, engage in discussions, and share their dining adventures within the app's social community.

***Tracking Dining History*:**

Scenario: Michael uses the app to keep a record of all the restaurants he has visited and the ratings and reviews he has given. He can easily look back at his dining history to remember his favorite places.

Use Case: Users can maintain a personal dining history within the app, allowing them to track their culinary journey and revisit their favorite restaurants.

***Restaurant Owners' Insights*:**

Scenario: The owner of a new restaurant, Bella's Bistro, logs into the app and reads reviews from customers. She notices that many users love her pasta dishes but have mixed feelings about the dessert menu. Using this feedback, she decides to revamp her dessert offerings.

Use Case: Restaurant owners can gain valuable insights from user reviews to improve their menu and overall dining experience.

***Culinary Events and Specials*:**

Scenario: The app features a section where restaurants can promote special events, seasonal menus, and limited-time offers. Users like Alex can explore this section to discover unique dining experiences such as wine tastings, chef's table dinners, and themed food festivals happening in their area.

Use Case: Users can stay informed about exclusive culinary events and promotions hosted by restaurants, enhancing their dining experiences with exciting and diverse options beyond regular menu items. This feature encourages users to explore new dining adventures and support local businesses' initiatives.

List of Main Data Items and Entities

1. User
2. Account
3. Admin
4. Restaurant
5. Friend
6. Point
7. Rating
8. Review
9. Recommendation (Recs)
10. User Activity/History
11. ChatAssistance (chatbot customer service)
12. Address
13. Cities
14. Notifications

Functional Requirements

1. ***User***

1.1. A user shall be able to create many accounts.
1.2. A user shall be authenticated by the system to verify their identity before granting full access to the application.
1.3. A registered user shall be able to rate a restaurant.
1.4. A registered user shall be able to review a restaurant.
1.5. A registered user shall be able to recommend a restaurant to other registered users.
1.6. A registered user shall be able to search for any restaurants.
1.7. A registered user shall be able to search for any other registered users.
1.8. A registered user shall be able to dislike a restaurant.
1.9. A registered user shall be able to dislike a cuisine.
1.10. A registered user who rates a restaurant shall have 'points' accumulated as their 'score'.
1.11. A registered user shall be able to see all their past reviews.
1.12. A registered user shall be able to see all their past restaurant ratings that they rated.
1.13. A registered user shall be able to see all the restaurants that they have visited.
1.14. A registered user shall be able to edit/manage their profile page.
1.15. A registered user shall be able to send a personal message to another registered user.
1.16. A registered user shall be able to add/follow other users.
1.17. A registered user shall be able to bookmark restaurants to their favorites list

2. ***Account***

2.1. An account shall be created by one user.
2.2. An account shall be verified by the user.
2.3. An account shall have a type, 'basic', dedicated for new customers that just join the app.
2.4. An account shall have a type, 'admin', dedicated for employees.
2.5. An account shall have a type, 'restaurant', dedicated for restaurant owners.
2.6. A 'basic' account type user shall receive a 'rank' once the user accumulated a certain amount of points/score.
2.7. A 'basic' account type shall be able to receive rank of 'silver', 'gold', or 'platinum' once the user accumulated a certain amount of points/score.
2.8. An account shall be created as 'restaurant' to be able to post a restaurant.
2.9. A 'restaurant' account shall be able to post many restaurants.
2.10. An account shall need verification upon registering for the first time.

3. ***Admin***

3.1. An account with a type of ‘admin’ shall be able to view all existing users and access to their account information.

3.2. An account with a type of ‘admin’ shall be able to add new restaurants.

3.3. An account with a type of ‘admin’ shall be able to temporarily/permanently remove existing restaurants.

4. ***Restaurant***

4.1. A restaurant shall have information such as brand name, address, location, website.

4.2. A restaurant shall have a rating.

4.3. A restaurant shall declare the type of cuisine they are serving.

4.4. A restaurant shall have its own profile page showcasing their brand image, products, rating, customer reviews and photos.

5. ***Friend/Followers***

5.1. Friends shall be able to see each other’s activities

6. ***Point/Score***

6.1. Point/Score shall be shown as the main information in a user profile page.

6.2. Point/Score of every user in the platform shall be listed by index as ‘leaderboard’.

6.3. Point/Score value shall be assigned to certain activities in the app.

6.4. A certain amount of Points/Score shall be provided to a user who rated a restaurant.

6.5. A certain amount of Points/Score shall be provided to a user who reviewed a restaurant.

6.6. A certain amount of Points/Score shall be provided to a user who recommended a restaurant.

7. ***Rating***

7.1. A user rating shall be uniquely associated with a single restaurant.

7.2. The overall rating of a restaurant shall be determined by averaging the ratings from all individual users.

7.3. A rating shall be able to be modified by the user.

7.4. A rating shall not be able to be deleted.

8. ***Review***

8.1. A review shall be uniquely associated with a single restaurant.

8.2. A review shall allow users to embed images.

8.3. A review shall have a maximum capacity for the amount of characters it can take.

8.4. A review shall be able to be updated or deleted by the author.

9. ***Recommendation (Recs)***

9.1. High rated restaurants shall be recommended to users

10. ***User Activity/History***

10.1. User activities like reviewing, rating, or recommending shall be assigned a dedicated point/score value.

10.2. User activities on each user shall be recorded.

10.3. Certain activities shall be visible to their friends/followers.

11. ***ChatAssistance (chatbot customer service)***

13.1. A chat assistance agent shall be able to serve many users.

13.2. A chat assistance service shall allow users to ask about restaurants and/or the cuisine they like.

13.3. A chat assistance service shall be able to recommend restaurants to users based on cuisine, restaurant name, or location.

12. ***Address***

14.1. An address shall have basic address information such as street name, city name, phone number, zip code, country.

14.2. An address shall have one city.

13. ***Cities***

15.1. A city shall belong to one address.

14. ***Notifications***

14.1 A user shall receive notifications for receiving new messages

14.2 A user shall receive notifications for friend requests

14.3 A user shall receive notifications for restaurant recommendations.

14.4 A user shall receive notifications on updates about their reviews

List of Non-Functional Requirements

1. ***Performance***

1.1 The system shall be able to handle a high amount of concurrent users.
1.2 The system shall have fast response times to user requests and interactions.
1.3 The acceptable response time to user requests shall be within 5 seconds.
1.3 The system shall be able to scale its size to support the growing or shrinking data.

2. ***Security***

2.1 The system shall only allow access to registered and authenticated users.
2.2 The system shall authenticate the user login by password.
2.3 The system shall encrypt user data and all other sensitive information during transmission.
2.4 A password policy shall be enforced and require users to have a password with at least 1 uppercase letter, 1 lowercase letter, 1 digit, and 1 special character.
2.5 Access rights to user data and system configurations are restricted to authorized system administrators.
2.6 The system shall be regularly monitored and tested for any cyber-attacks or vulnerabilities.

3. ***Usability***

3.1 The system shall be easy to use, understand, and navigate.
3.2 There shall be written documentation of the UI design standards that describes the website's layout, color scheme, and overall behavior.
3.3 The system shall follow the documentation of the UI standardized designs to ensure the overall appearance and usability of the website is consistent.
3.4 The system shall be accessible to those with disabilities by including alternative text on all images.
3.5 The system shall be accessible to those with disabilities by including screen reader accessibility.
3.6 The system shall state understandable user errors with hints to quickly fix said errors.

4. ***Maintainability***

4.1 The system shall be easy to maintain, modify, and update if needed without notifying the users.
4.2 The system shall be written in readable and understandable code to quickly update or modify if needed.

5. ***Scalability***

5.1 The system shall be able to handle the existing data size without any crashes or failures.

5.2 The system shall be able to scale up to handle an increase in data size without disrupting performance.

6. ***Storage***

6.1 The system shall be able to handle large amounts of data, including user information, restaurant information, and more.

6.2 The system shall frequently back-up the data.

6.3 The system shall have a recovery storage.

6.4 The system shall encrypt sensitive user data in the database.

7. ***Reliability***

7.1 The system shall be able to perform consistently without any problems.

7.2 The system shall switch to a backup system in case of software or hardware failures.

8. ***Regulatory***

8.1 The system shall comply with data protection regulations.

8.2 The system shall comply with California Consumer Privacy Act (CCPA).

8.3 The system shall comply with net neutrality rules.

8.5 The system shall conduct security audits to ensure ongoing adherence to regulatory requirements.

9. ***Serviceability***

9.1 The system shall be easy to fix when improvements are needed.

9.2 The system shall have good error-handling mechanisms to capture and report errors and exceptions.

9.3 The system shall support debugging tools and diagnostic utilities.

9.4 The system shall provide user support through user assistance resources.

10. ***Data Integrity***

10.1 The system shall contain data that is accurate, complete, and correctly represents reality.

10.2 The system's data shall be retrieved from a dependable source and shall not be altered in any way.

10.3 The system shall validate data to provide reliable and error-free information.

10.4 The system shall enforce all data entries to follow a set of standards to ensure that data will be entered and formatted correctly for accuracy.

10.5 The system shall not omit any data or information that is crucial to the business.

11. ***Privacy***

11.1 The system shall not share user's personal data with other partners.

11.2 The system shall not give anyone access to user's personal data other than the owner.

11.3 The system shall not track users activities.

12. ***Availability***

12.1 The system shall be available consistently without any disruptions.

12.2 The system shall be available during maintenance.

12.3 The system shall be available during backup or restore.

13. ***Capacity***

13.1 The system shall support a minimum of [x] concurrent users during peak usage hours. Number should be determined based on expected user traffic

13.2 The system shall allocate sufficient storage for user profiles, reviews, ratings and related data. Each user shall allow for [x] megabytes of data storage.

14. ***Environmental***

14.1 The system shall be compatible with Windows, macOS and Linux for desktop browsers.

14.2 The system shall be compatible with Android and iOS for mobile browsers

14.3 The applications shall be compatible with a wide range of web browsers, including but not limited to Chrome, Firefox,Safari and Edge.

Competitive Analysis

(Descriptive list in replacement of graph)

1. ***Yelp***
- Allows users to rate and review restaurants.
- Provides information on restaurant hours, locations, and contact details.
- Offers a basic recommendation system based on user reviews and preferences.
- Features a social component where users can follow each other.

2. ***TripAdvisor***
- Focuses on travel-related reviews, including restaurants.
- Allows users to rate and review restaurants.
- Provides travel and accommodation information.
- Offers a basic recommendation system based on user reviews.

3. ***Zomato***
- Specializes in restaurant and food-related content.
- Allows users to rate and review restaurants and dishes.
- Offers detailed restaurant menus and photos.
- Provides a recommendation system based on user preferences.

4. ***OpenTable***
- Primarily a restaurant reservation platform.
- Allows users to reserve tables and view restaurant ratings and reviews.
- Provides a loyalty program for diners.
- Offers basic restaurant information.

5. ***Google Maps***
- Offers restaurant ratings and user reviews.
- Provides location-based restaurant recommendations.
- Includes basic restaurant information and directions.

Comparison Table:

Feature	Yelp	TripAdvisor	Zomato	OpenTable	Google Maps	BiteRate (Planned)
Dish Ratings and Reviews	**No**	**No**	**Yes**	**No**	**No**	**Yes**
Restaurant Insights	**No**	**Yes**	**Yes**	**No**	**No**	**Yes**
Personalized Recommendations	**No**	**No**	**Yes**	**No**	**Yes**	**Yes**
Culinary Community	**Yes**	**No**	**Yes**	**No**	**No**	**Yes**
Comprehensive Dish Details	**No**	**No**	**Yes**	**No**	**No**	**Yes**
Location based Features	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

Summary:

BiteRate distinguishes itself by offering a comprehensive and innovative approach to dining experiences. While existing platforms focus on restaurant ratings and general reviews, BiteRate plans to implement individual dish ratings and reviews, providing users with precise insights into what to order. The application combines the strengths of its competitors, including restaurant insights, personalized recommendations, and a vibrant culinary community. Additionally, BiteRate's unique selling point lies in its emphasis on dish-centric ratings and reviews, allowing users to discover exceptional culinary creations irrespective of the restaurant. This distinctive feature sets BiteRate apart and positions it as a comprehensive and valuable resource for food enthusiasts looking to make the most of their dining experiences.

Checklist

ItemsTeam found a time slot to meet outside of class	Done
Team found a time slot to meet outside of the class.Github Master chosen	Done
Team decided and agreed together on using the listed SW tools and deployment server.	Done
Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing.	Done
Team lead ensured that all team members read the final M1 and agree/ understand it before submission.	Done
Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)	Done