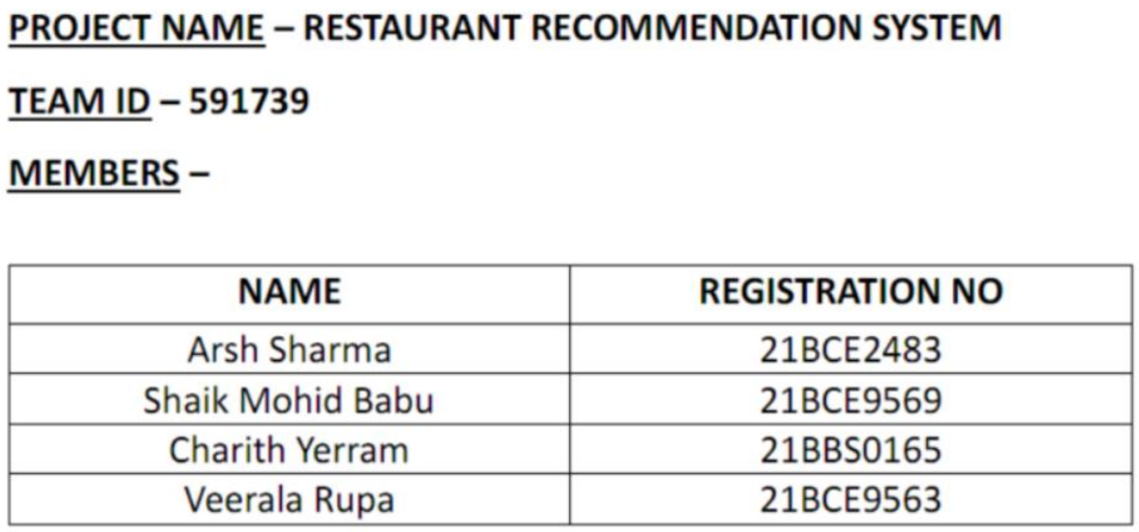
FINAL SMARTINTERZ PROJECT

REQUIREMENT ANALYSIS



PROJECT NAME – RESTAURANT RECOMMENDATION SYSTEM

TEAM ID – 591739

MEMBERS –

NAME	REGISTRATION NO
Arsh Sharma	21BCE2483
Shaik Mohid Babu	21BCE9569
Charith Yerram	21BBS0165
Veerala Rupa	21BCE9563

TWO TYPES OF REQUIREMENTS FUNCTIONAL AND NON FUNCTIONAL

- **Functional Requirements:**

1. **User Registration and Authentication**: Users should be able to create accounts with a valid email address and password.
   The system must provide secure authentication mechanisms.
2. **User Profile Management:** Users can update their profiles with personal information, including dietary preferences and location.
3. **Restaurant Database:** The system should maintain a comprehensive database of restaurants with details such as name, location, cuisine type, price range, ratings, and reviews.
4. **Recommendation Algorithm:** Develop an algorithm that considers user preferences, past behavior, and location for personalized restaurant recommendations.
5. **Search and Filtering:** Users should be able to search for restaurants based on their current location or a specified area.
   Provide filtering options based on cuisine type, price range, ratings, and other relevant criteria.
6. **User Interaction:** Allow users to submit reviews and ratings for restaurants.

Integration with social media for users to share recommendations and reviews.

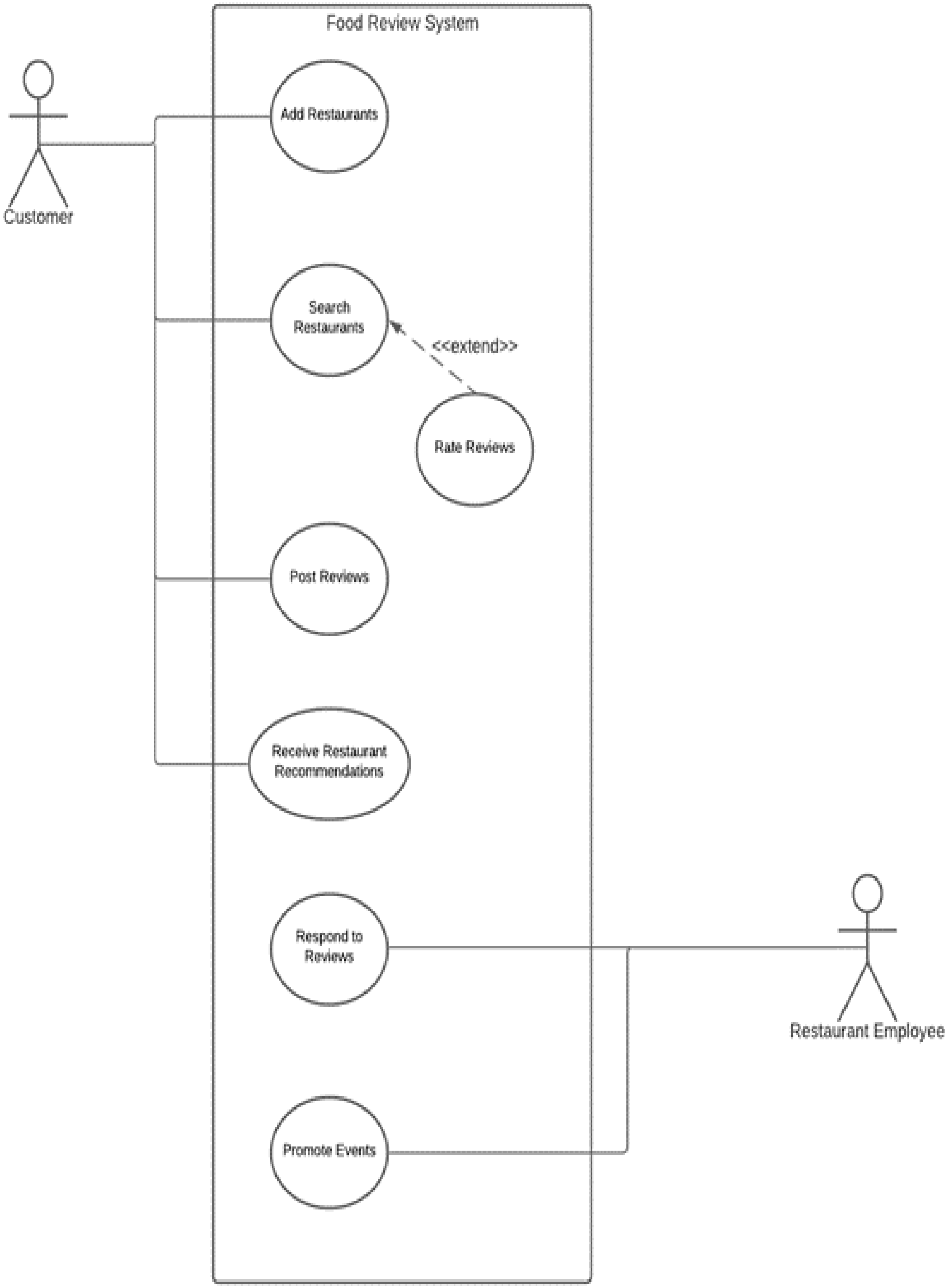
7. **Notifications:** Implement personalized alerts to notify users of new restaurant recommendations, special offers, or events based on their preferences.

   Users should receive feedback alerts to encourage them to provide feedback on recommendations.
8. **Reservation Integration:** Integrate with a reservation system to enable users to book tables directly through the platform.
9. **Mobile-Friendly Interface:** Ensure the system is accessible and user-friendly on various devices, particularly mobile phones.

- **Non-Functional Requirements:**
  1. **Scalability and Performance**: Design the system to handle increased user and restaurant listings without a significant drop in performance. Implement load balancing to distribute traffic efficiently.
  2. **Feedback and Improvement:** Provide a mechanism for users to submit feedback and report issues.

     Regularly update and enhance the recommendation algorithm based on user feedback.
  3. **Security:** Implement data encryption for user data, especially sensitive information like passwords, and ensure secure communication between the user’s device and the system.
  4. **Documentation:** Provide user guides and documentation for both users and administrators.

     Document any APIs used or exposed.
  5. **Testing:** Implement rigorous testing procedures, including unit testing, integration testing, and user acceptance testing.


Food Review System
Customer
Add Restaurants
Search Restaurants
<<extend>>
Rate Reviews
Post Reviews
Receive Restaurant Recommendations
Respond to Reviews
Promote Events
Restaurant Employee