Demo Project Proposal

**Statement of Purpose**

**Title:** Love Bites (Perfect Plates for perfect Dates)

**Purpose**: Help you find the right restaurant for the right date.

**Runtime Environment**

Mobile App

Cloud services needed

**Information Needed**

* Dietary Preferences (vegan, vegetarian, allergies)
* Cuisine Preferences (Italian, Mexican, American)
* Time Preferences (Breakfast, Lunch, & Dinner)
* Occasion (Casual Weekday, dinner, special occasion)
* Budget
* Location
* Mood/Occasion
* Calendar integration
* User Reviews
* Visuals (Photos of the food/seasonings
* Seasons (Meals being seasonal based on time of year)

**Data to be persisted (data Base)**

* User Profile
* Restaurants
* Date Night preference
* Calendar integration
* Location

**App Concerns**

* Interaction- keeping users engaged or returning to the app
* Screen – fitting all platforms in a custom manner
* Security
* Compatibility with multiple devices
* Performance Optimization

**User Interface Outline**

* Home Screen (Welcome message, quick access buttons)
* User Profile (Profile setup)
* Date Night Preference (Mood and Occasion/Budget)
* Restaurant finder (Search Bar, filters, List)
* Cuisine Type (Mexican, Italian, American)
* Rating (How pristine the establishment)
* Favorites/History
* Notifications
* Settings

**Functional Flow**

1. User Registration/login
2. Home Screen:

* Date night preference
* Restaurant finder
* Cuisine Suggestions
* Favorites and History
* Notifications
* Settings
* Help & Support

**Special Features**

* Ingredient based search
* Left over management

**Classes**

**User Class**

* UserId
* Name
* Email
* PasswordHash
* FavoriteCuisines (List<string>)
* DietaryRestrictions (List<string>)
* Preferences (List<Preference>)
* History (List<History>)

**2. Preference Class**

* PreferenceId
* Mood
* Occasion
* Budget

**3. Restaurant Class**

* RestaurantId
* Name
* Address
* CuisineType
* AverageCost
* MenuItems (List<string>)
* Reviews (List<Review>)

**4. Review Class**

* ReviewId
* UserId
* Rating
* Comment
* Date

**5. History Class**

* HistoryId
* UserId
* RestaurantId
* RecipeId
* Date
* ChoiceType (e.g., “Restaurant” or “Recipe”)

**6. Poll Class**

* PollId
* UserIds (List<int>)
* Options (List<string>)
* Votes (Dictionary<int, string>) // UserId -> Option

**7. Notification Class**

* NotificationId
* UserId
* Message
* Timestamp
* IsRead