# NeighborBean Cafe

**Premium Local Cafe Experience**

**Abstract:**

A System Analysis and Project Management project on enhancing customer engagement through mobile targeting, leveraging geo-fencing and geo-conquesting strategies to deliver customized coupons and premium coffee experiences at NeighborBean Cafe.

Under the assistance of Professor Srinivasan Raghunathan

ANANYA SRINATH ACHAR
axa220264@utdallas.edu

GRASHMA SRIHITA CHAPARLA
gxc220031@utdallas.edu

VAMSHI MAYA
vxm230022@utdallas.edu

Contents

Executive Summary

The "NeighborBean Cafe" initiative uses cutting-edge mobile targeting technologies to change the way coffee businesses interact with their patrons. Conventional marketing strategies frequently fail to draw in and keep customers in the highly competitive coffee shop industry. Personalized marketing methods, artificial intelligence, and real-time location data are used in this initiative to identify and close these gaps. Creating an exceptional and highly personalized customer experience is the aim. The project intends to offer customized recommendations, dynamic pricing, and personalized promotions that take into account variables like weather, consumer behavior, and mood by focusing on clients who frequent the café and it's competitors.

NeighborBean Cafe's suggested solution has a number of cutting-edge features and technology. The cafe will have virtual perimeters created using geo-fencing and geo-conquesting technologies, which will allow the system to notify prospective customers who enter these zones. In order to analyze consumer preferences and behaviors and provide personalized recommendations and dynamic pricing adjustments to maximize profits and control demand, artificial intelligence will be a critical component of the system. The system will also be integrated with well-known food delivery services like UberEats and DoorDash, expanding the cafe's customer base and allowing for notifications. Extra features like mood recognition, health tracking, and social interactions are intended to improve user experience and increase engagement.

The goal of the NeighborBean Cafe project is to greatly increase sales, strengthen customer loyalty, and raise overall customer happiness by correcting the current flaws in coffee shop marketing. NeighborBean Cafe will become the go-to option for coffee enthusiasts because of its extensive mobile targeting system, which guarantees that consumers receive offers that are pertinent and timely. The project intends to establish NeighborBean Cafe as a distinctive choice in the congested coffee shop industry by offering a distinctive and local experience that draws in both new and returning customers. It places a significant emphasis on personalized and technology-driven customer engagement. Incorporating cutting-edge technologies and

customized marketing approaches would not only set NeighborBean Cafe apart from its rivals but also raise the bar for industry standards in customer interaction.

**Problem Statement**

In the competitive coffee shop industry, attracting and retaining customers is crucial and mandatory. With the rise of mobile technology, coffee shops can now use real-time location data to target potential customers more effectively. On the other hand, traditional marketing often misses the mark, fails to engage consumers in a personalized and timely way, and loses opportunities to boost foot traffic and sales.

Every day, many consumers pass by coffee shops without realizing what specials or promotions are happening. There is a lost opportunity, disconnect between coffee shop promotions and potential customers. Additionally, the inability to inform regular customers about exclusive deals and promotions can lead to losing loyal customers. Mobile technology is not effectively utilized by current marketing methods to offer a customized consumer experience.

**Business Need**

There are a lot of cafes and coffee shops like Starbucks and Dutch bros that are popular. Even though they make a lot of profit, we feel they might spend less time pondering about targeting the right audience. So what makes our new system necessary?
1. The fact that more attention should be paid to prospective clients nearby.
2. Currently there is a loss of potential as well as loyal customers due to lack of personalized advertisement.
3. Additionally missed sales opportunities due to lack of knowledge about promotions.
4. There is a need to enhance customer experience with tailored offers.
5. The new system is mostly also required to boost interaction with location-based, real-time promotions.

**Objectives**

Objectives of the proposed mobile targeting system for NeighborBean Cafe are as follows:

- Increase Customer Engagement: Send tailored alerts based on current location information to boost customer engagement.
- Boost Sales: Tell loyal customers about special offers and tailored promotions.
- Enhance Customer Loyalty: Inform regular customers about exclusive deals and personalized offers.
- Optimize Marketing Efforts: For successful and efficient marketing campaigns, make use of mobile technology.
- Improve Customer Experience: Provide a seamless and customized customer experience through advanced features.

**Scope**

The scope of the proposed mobile targeting system includes:

- Implementing Geo-Targeting: Use geo-fencing and geo-conquesting to draw clients from nearby businesses and competitors.
- AI Integration: Make use of AI to provide dynamic pricing and tailored recommendations based on consumer behavior.
- Cross-Platform Integration: To increase reach and send notifications, collaborate with well-known meal delivery services like DoorDash and UberEats.
- Feature Development: To improve user experience, develop features for social interactions, health tracking, and mood detection.
- Contextual Promotions: Send tailored promotions with context by taking into account current variables like the temperature and the preferences of a particular age group.

**Proposed Solution**

The suggested course of action for NeighborBean Cafe is to put in place an extensive mobile targeting system that makes use of cutting-edge technologies and real-time location data to boost sales, improve customer satisfaction, and increase customer engagement. This is a detailed breakdown of the solution:

The name "NeighborBean Cafe" points out a distinctive and local neighborhood experience. The tactic aims to attract newcomers who frequently prefer unique, locally owned businesses to well-known franchises like Starbucks. With this marketing strategy, customers should have a new experience that is local and exciting, making NeighborBean Cafe their first choice.

1. Using geo-fencing, we can target customers who enter the cafe by establishing a virtual perimeter around it. Geo-conquesting focuses on customers close to competitor businesses locations.
   Implementation: When customers are in the vicinity of another cafe or NeighborBean Cafe, the system will promptly notify them via their mobile devices. Personalized offers and promotions will be included in these notifications to persuade customers to visit NeighborBean Cafe rather than a competitor..

2. Artificial Intelligence-Powered and Behavioral Suggestions: Use AI to assess the customers preferences and actions. Implementation: Customers will receive push notifications from the system each time they search for cafes or browse meal delivery apps. The system can make customized recommendations and offers that are more likely to be accepted by the customer by analyzing their behavior.

3. Dynamic Pricing: Adjusting prices based on real-time factors to attract customers.
   Implementation: In order to entice customers to visit NeighborBean Cafe, prices will change depending on how close they are to the cafe. Utilizing this dynamic pricing approach can help us control demand and optimize profits.

4. Enabling Social Features: Enhancing the social aspect of the cafe experience.
   Implementation: Customers will be able to redeem gifts for points and share points with friends through the application. With the help of shared rewards and word-of-mouth, these social features encourage customers to interact with the cafe and attract new customers.

5. Tracking for Health: Healthier decision-making through monitoring and alerts is encouraged by tracking.
   Implementation: The application will keep track of how much coffee is consumed and send out personalized alerts that recommend healthier options, like drinks without caffeine. This feature encourages a balanced lifestyle and appeals to customers who are health-conscious.

6. Mood Detection: Detecting the mood of the customer and customizing offers accordingly.
   Implementation: Push notifications will be sent in accordance with the application's detection of the customers moods. For instance, the system might provide discounts on cakes and pastries to a customer who is having a rough day. Customers will also be able

to select their mood via the app and get personalized drink recommendations like a smiley cookie to get through a difficult day.

7. Age-Based Push Notifications: Utilizing age-based push notifications, marketers can target specific age groups with their messaging.
   Implementation: To appeal to various demographics, the system will send teens vibrant, lively images and graphics along with other age-specific marketing techniques. This guarantees that all age groups will find promotions interesting and relevant.

8. Integration with Famous Food Applications: Integration with Well-Known Food Applications: Increasing exposure via well-known meal delivery services.
   Implementation: The system can notify a larger audience by collaborating with DoorDash and UberEats. Through this integration, NeighborBean Cafe is able to draw in new clients who might not have otherwise visited.

9. Weather-Wise Notifications: SMaking offers in accordance with the weather at the moment.
   Implementation: On rainy days, the system will serve hot chocolate and brownies; on hot summer days, it will serve cool frappes. By aligning promotions with the weather, the cafe can provide more relevant and appealing offers to customers.

**Business Process Model Notation (BPMN)**

The BPMN diagram that shows the complete process flow of the suggested system is shown in the diagram below:
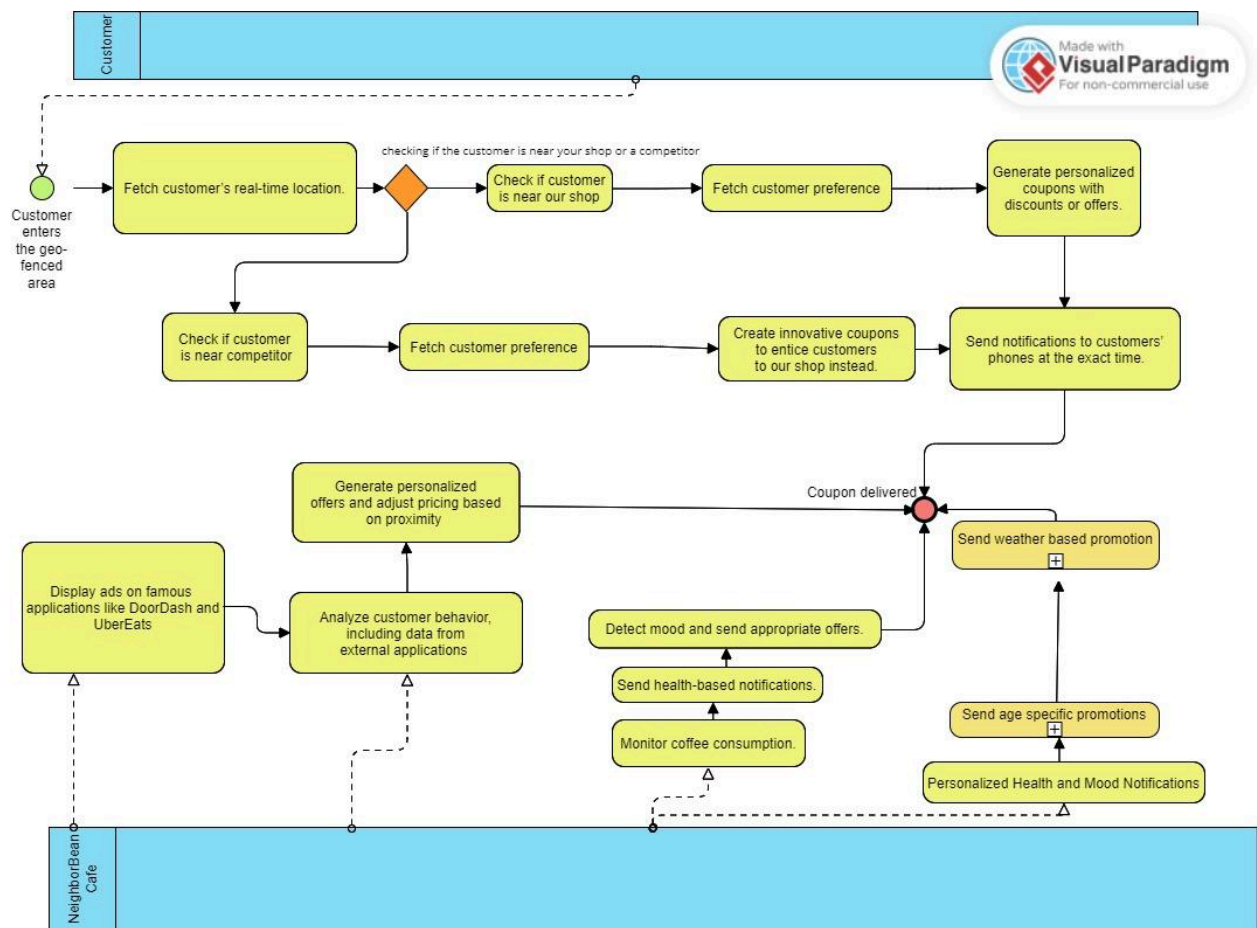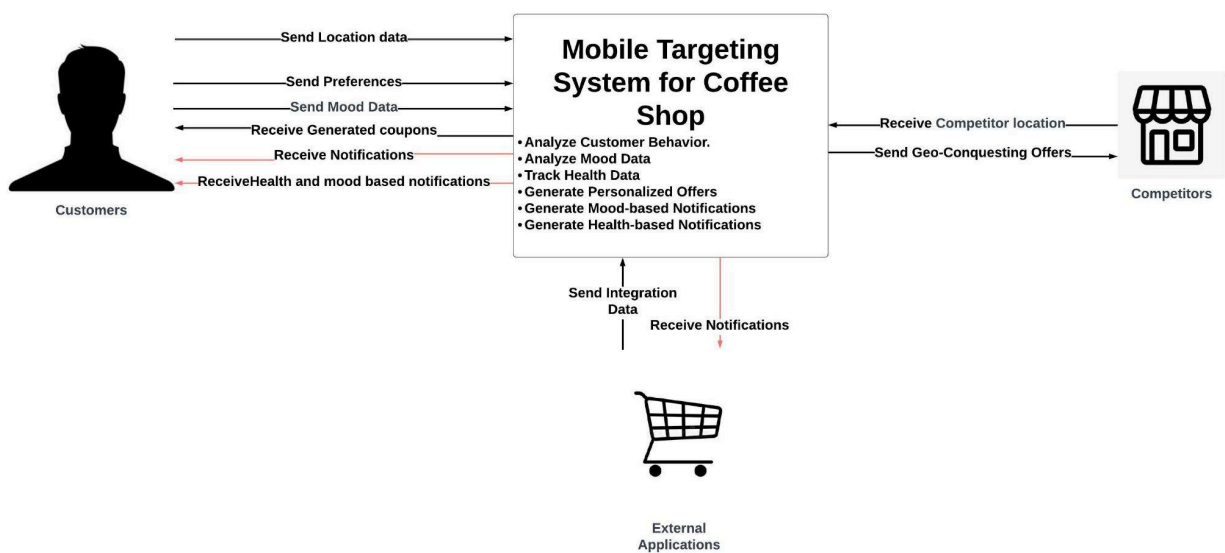
NeighborBean Cafe's mobile targeting system's Business Process Model and Notation (BPMN) diagram shows a comprehensive flow that encapsulates the critical steps and decision points in successfully pursuing customers. When a consumer enters a geofenced area, the system is triggered to retrieve their current location, which starts the process. The system either generates customized coupons with discounts or comes up with creative offers to lure customers away from competitors, depending on whether the customer is near our shop or one of our competitors. The customer's phone receives these offers as notifications at that precise moment, leading up to the coupon's delivery.

In order to improve customer engagement, we also integrated a number of cutting-edge features. These consist of tasks like studying consumer behavior, obtaining information from third-party apps like Uber Eats and DoorDash, and creating customized offers using AI suggestions. Additional features include age- and weather-specific promotions, health and mood tracking to send relevant notifications, and dynamic pricing adjustments based on behavior and proximity. Clear definitions of every task, decision point, and event guarantee a

smooth and effective process that enhances customer satisfaction and increases sales and loyalty for NeighborBean Cafe.

**System Context Diagram**

The context diagram, which provides an overview of the proposed system's functionality, is shown in the diagram below:



The main interactions between the Mobile Targeting System at NeighborBean Cafe and its external entities—Customers, Competitors, and External Applications, are depicted in our context diagram. According to the diagram, customers provide the system with information about their preferences, location, and mood. The system then uses this information to create personalized coupons, alerts, and health and mood-related notifications that depend on the customer's present situation. By doing this, consumers are more likely to interact with the cafe and receive offers that are relevant to them. In order to provide a highly customized client experience, the system also collects health data, analyzes consumer behavior, and creates offers based on AI-driven recommendations.

In order for the system to create geo-conquesting offers meant to entice customers away from competitor sites, competitors must communicate with it by sending location data. The system

receives integration data from external applications like DoorDash and UberEats, which is analyzed to improve the efficacy of personalized offers and provide deeper insights into user behavior. To reach a larger audience, the system notifies these applications. Through timely and appropriate marketing, NeighborBean Cafe may successfully target new consumers, increase sales, and strengthen customer loyalty by utilizing these in-depth interactions.

**Use Case Diagram**

The below diagram represents the use case for the proposed system. The actors of the system are Customer, Competitor, and External Applications.

1. Provide Location Data
2. Submit Preferences
3. Send Mood Data
4. Receive Personalized Coupons
5. Get Notifications
6. Receive Health and Mood-Based Alerts
7. Fetch Competitor Location
8. Create Geo-Conquesting Offers
9. Integrate Data
10. Receive Notifications
11. Analyze customer behavior
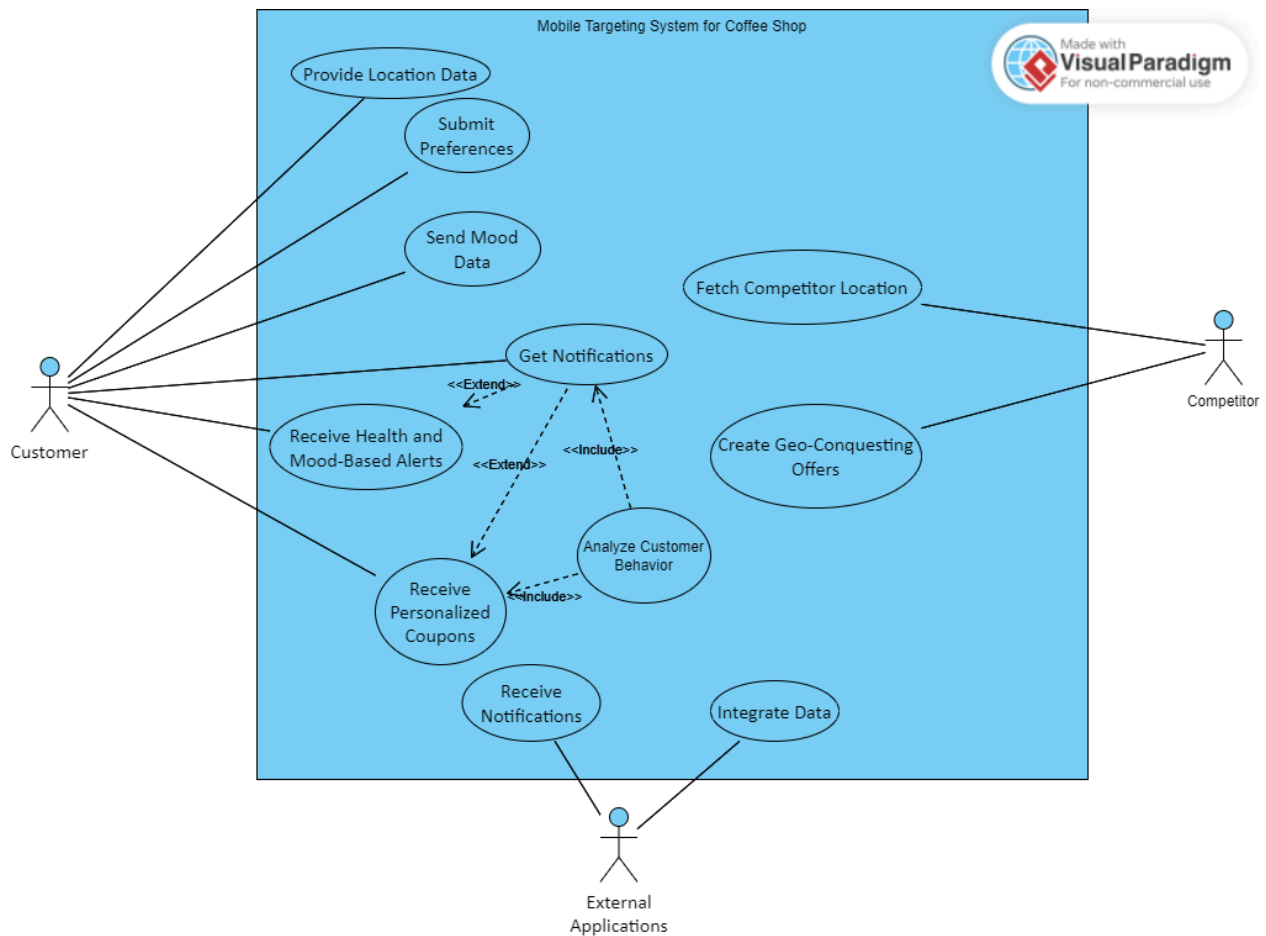
Use Case descriptors -

**Use Case Description 1**

Use Case Name: Provide Location Data
Primary Actor: Customer
Stakeholders: NeighborBean Cafe
Brief Description: Customer provides their current location data to the system.
Trigger: Customer's mobile device enters the geo-fenced area.
Normal flow of events:

1. Customer's device enters the geo-fenced area.
2. The mobile targeting system detects the customer's location.
3. The system updates the location data.

**Use Case Description 2**

Use Case Name: Submit Preferences
Primary Actor: Customer
Stakeholders: NeighborBean Cafe
Brief Description: Customer submits their preferences for personalized offers.
Trigger: Customer logs into the NeighborBean Cafe app and updates their preferences.
Normal flow of events:

1. Customer logs into the NeighborBean Cafe app.
2. Customer navigates to the preferences section.
3. Customers update their preferences.
4. The system stores the updated preferences.

**Use Case Description 3**

Use Case Name: Send Mood Data
Primary Actor: Customer
Stakeholders: NeighborBean Cafe
Brief Description: Customer sends their mood data to the system for personalized offers.
Trigger: Customer interacts with the NeighborBean Cafe app and updates their mood status.
Normal flow of events:

1. Customer logs into the NeighborBean Cafe app.
2. Customers update their mood status.
3. The system stores the mood data.

**Use Case Description 4**

Use Case Name: Receive Generated Coupons
Primary Actor: Customer
Stakeholders: NeighborBean Cafe
Brief Description: Customers receive personalized coupons generated based on their location, preferences, and mood.
Trigger: System detects the customer's location, preferences, and mood.
Normal flow of events:

1. System detects the customer's location, preferences, and mood.
2. System generates personalized coupons.

3. Customers receive the generated coupons.

## Use Case Description 5

Use Case Name: Receive Notifications
Primary Actor: Customer
Stakeholders: NeighborBean Cafe
Brief Description: Customer receives notifications about offers, events, and other relevant information.
Trigger: System generates a notification based on customer data.
Normal flow of events:

1. System generates a notification based on customer data.
2. Customer receives the notification.

## Use Case Description 6

Use Case Name: Receive Health and Mood-Based Notifications
Primary Actor: Customer
Stakeholders: NeighborBean Cafe
Brief Description: Customer receives notifications tailored to their health and mood data.
Trigger: System analyzes customer's health and mood data.
Normal flow of events:

1. System analyzes customer's health and mood data.
2. System generates health and mood-based notifications.
3. Customer receives the notifications.

## Use Case Description 7

Use Case Name: Fetch Competitor Location
Primary Actor: Mobile Targeting System
Stakeholders: NeighborBean Cafe
Brief Description: System fetches the location of competitors' coffee shops.
Trigger: Periodic update or when a customer enters a geo-fenced area.
Normal flow of events:

1. System periodically fetches competitors' locations.
2. System updates the location data.

**Use Case Description 8**

Use Case Name: Create Geo-Conquesting Offers
Primary Actor: Mobile Targeting System
Stakeholders: NeighborBean Cafe
Brief Description: System creates offers to attract customers away from competitors' locations.
Trigger: Customer is near a competitor's location.
Normal flow of events:

1. System detects customers near a competitor's location.
2. System creates geo-conquesting offers.
3. Customers receive the geo-conquesting offers.

**Use Case Description 9**

Use Case Name: Integrate Data
Primary Actor: Mobile Targeting System
Stakeholders: NeighborBean Cafe, External Applications
Brief Description: System integrates data from various external applications.
Trigger: Data synchronization schedule or customer interaction with external apps.
Normal flow of events:

1. System synchronizes data from external applications.
2. System updates integrated data.

**Use Case Description 10**

Use Case Name: Receive Notifications from External Applications
Primary Actor: Customer
Stakeholders: NeighborBean Cafe, External Applications
Brief Description: Customer receives notifications based on data from external applications.
Trigger: System processes integrated data from external applications.
Normal flow of events:

1. System processes integrated data from external applications.
2. System generates notifications based on the data.

3. Customer receives the notifications.

**Data dictionary**

**Use Case 1: Provide Location Data**

- Location Data = LocationID + CustomerID + Latitude + Longitude + Timestamp
- Customer Data = CustomerID + Name + Contact + Email + Preferences + Address

**Use Case 2: Submit Preferences**

- Customer Preferences = CustomerID + PreferenceID + PreferenceDetails
- Customer Data = CustomerID + Name + Contact + Email + Preferences + Address

**Use Case 3: Send Mood Data**

- Mood Data = MoodID + CustomerID + MoodStatus + Timestamp
- Customer Data = CustomerID + Name + Contact + Email + Preferences + Address

**Use Case 4: Receive Generated Coupons**

- Generated Coupons = CouponID + CustomerID + Discount + Expiration + Preferences + Rewards
- Coupon Details = CouponID + Discount + Expiration + Type + Preferences + Rewards
- Sent Coupon Data = SentCouponID + CustomerID + CouponID + SentTimestamp
- Rewards = [Yes | No | Reward Type]
- Customer Data = CustomerID + Name + Contact + Email + Preferences + Address

**Use Case 5: Receive Notifications**

- Notification Data = NotificationID + CustomerID + NotificationType + Message + Timestamp
- Notification Content = NotificationID + Message + Timestamp
- Notification Type = NotificationID + NotificationType
- Customer Data = CustomerID + Name + Contact + Email + Preferences + Address

**Use Case 6: Receive Health and Mood-Based Notifications**

- Health and Mood-Based Notifications = NotificationID + CustomerID + HealthData + MoodData + Message + Timestamp

- Health Data = HealthID + CustomerID + HealthStatus + Timestamp
- Mood Data = MoodID + CustomerID + MoodStatus + Timestamp
- Health and Mood Notification Data = NotificationID + HealthID + MoodID + Message + Timestamp
- Customer Data = CustomerID + Name + Contact + Email + Preferences + Address
- MoodBasedNotification Data = NotificationID + Mood + Criteria + Message

## Use Case 7: Fetch Competitor Location

- Competitor Location Data = CompetitorID + CompetitorName + CompetitorAddress + Latitude + Longitude + Timestamp
- Geofence Data = GeofenceID + ShopLocation + Radius + TargetList
- Customer Geofence Information = CustomerGeofenceID + CustomerID + GeofenceID
- GeoConquest Data = GeoConquestID + CompetitorLoc + Radius + TargetList

## Use Case 8: Create Geo-Conquesting Offers

- Geo-Conquesting Offers = OfferID + CustomerID + CompetitorID + Discount + Expiration + Message + Rewards
- Competitor Information = CompetitorID + CompetitorName + CompetitorLocation
- Geo-Conquesting Offer Details = OfferID + Discount + Expiration + Message + Rewards
- Customer Geo-Conquesting Data = CustomerID + CompetitorID + OfferID + SentTimestamp
- Rewards = [Yes | No | Reward Type]
- GeoConquest Data = GeoConquestID + CompetitorLoc + Radius + TargetList

## Use Case 9: Integrate Data

- Integration Data = IntegrationID + AppName + AppID + Details + Timestamp
- External App Data = AppID + AppName + IntegrationDetails
- Integration Logs = IntegrationID + AppID + Timestamp + Status
- Customer_AppIntegration Data = Customer_AppIntegrationID + CustomerID + IntegrationID + IntegrationDetails

## Use Case 10: Receive Notifications from External Applications

- External Application Notifications = NotificationID + CustomerID + AppID + Message + Timestamp
- AppIntegration Data = IntegrationID + AppName + AppID + Details

**CLASS DIAGRAM - Without Methods**
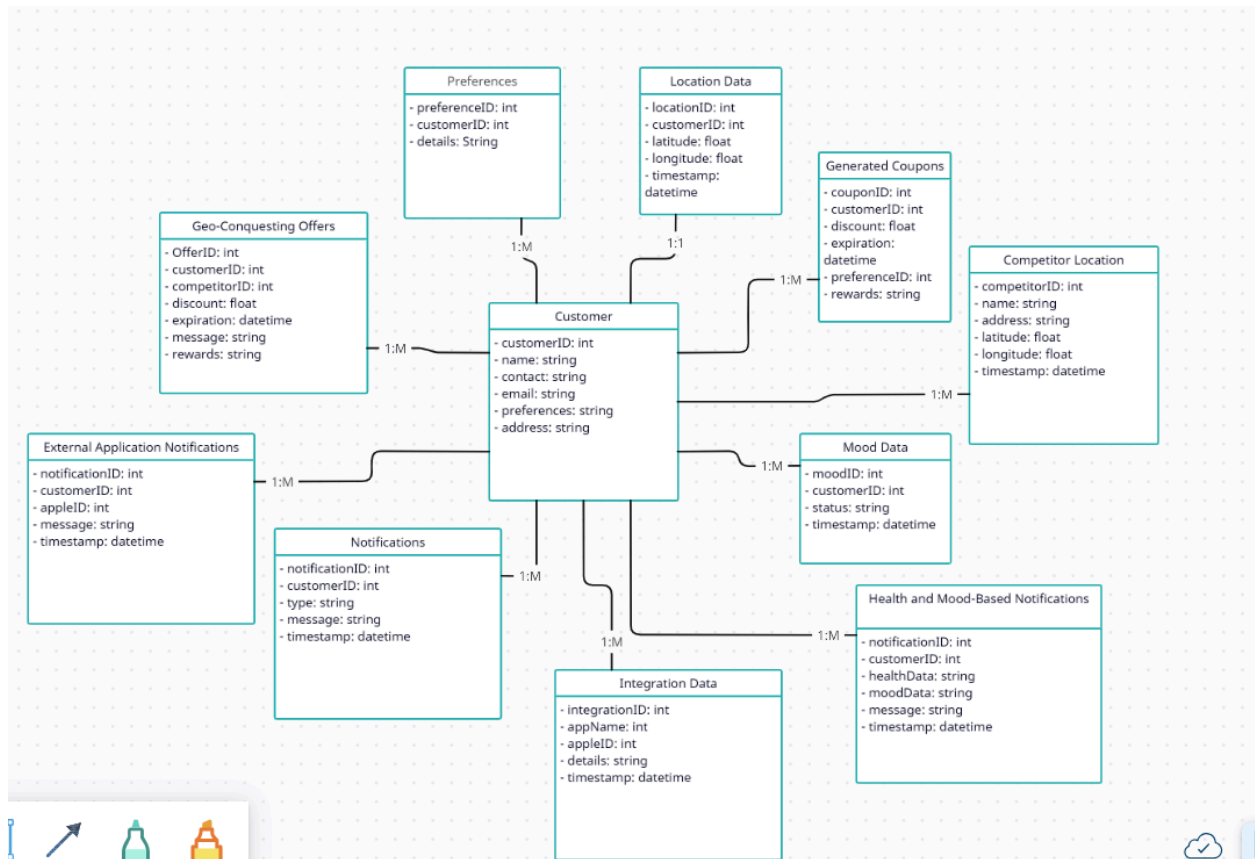


12. **Object behavior model (sequence diagram):**

A sequence diagram is a Unified Modelling Language (UML) diagram that shows us the order of communication of messages between the objects involved.

The objects involved in our sequence diagram include:

1. Customer- The device used by a customer to receive coupons (preferably a mobile device).
2. Geo-fencing module- Recognizes potential customers in the vicinity of our coffee shop.
3. Mobile targeting system- Keeps a track of location data and sends coupons.
4. Notification service- Send the notifications to the mobile devices.
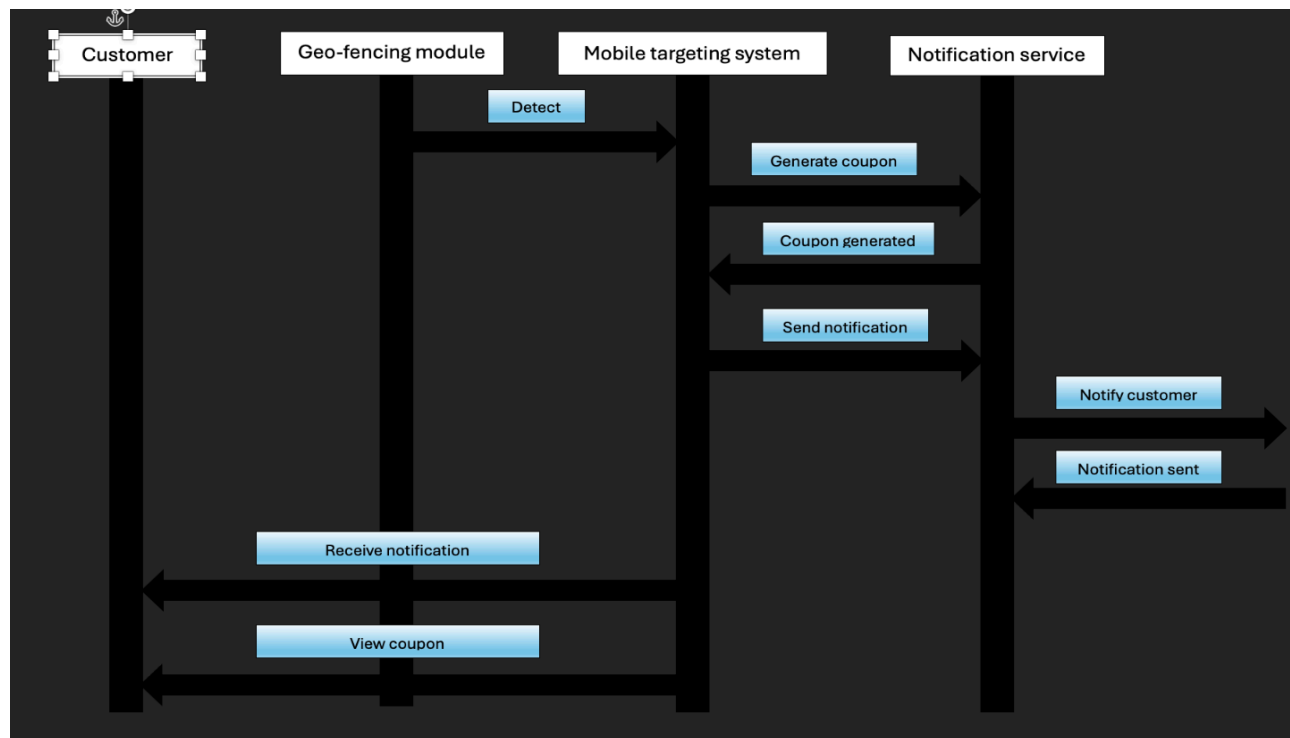
13. Functional specifications for Mobile targeting system:

1. User Login: Consumers can login to our app with a unique username and password to access customized offers.

2. Location detection: Our software detects when potential customers enter the proximity of our coffee shop or a competitor's location using geo-fencing and geo-conquesting technologies.

3. Customer data management: Our system saves and retrieves each customer's preferences to ensure customized coupon creation.

4. Coupon generation: Based on a customer's location and data, personalized and relevant offers are provided.

5. Sending notifications: Customers' mobile devices are sent customized coupons as notifications through a notification service.

6. Performance: Our software is designed to handle 1000 notifications per second and simultaneous location detections.

7. Security: To provide privacy and security, customer data is encrypted.

8. Usability: To make it easy for the customers, our app provides a user-friendly interface that facilitates easier interaction and navigation.

9.    Technical Specification: Our system supports both Android and IOS devices with a cloud based backend using technologies such as React Native, Swift, Kotlin, Node.js, Python, Java, MongoDB, MySQL, Firebase Cloud Messaging (FCM), and Apple Push Notification Service (APNS).

14. Interface design:

Frame 1: This is the first frame that will open when you open the app and right after we click on get started.



Frame 2: This is our login page. Once you login, it will lead you to our home page.
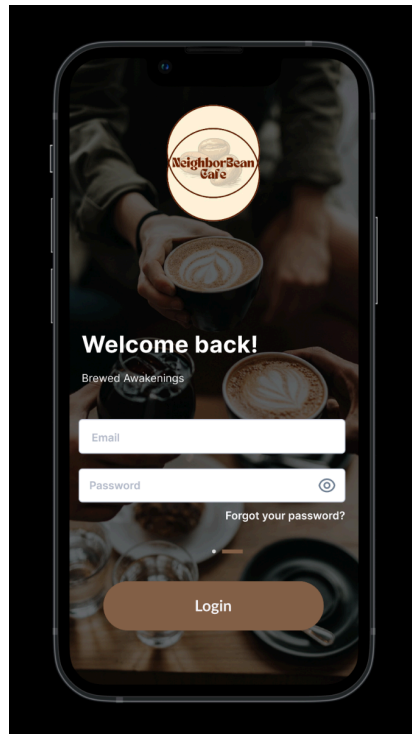
Frame 3: This is our home page which will show you our categories highlighting your favorite drink and the deal of the day which when clicked on will redirect you to our promotions page. The star icon will also do the same. It also gives information about events going on.
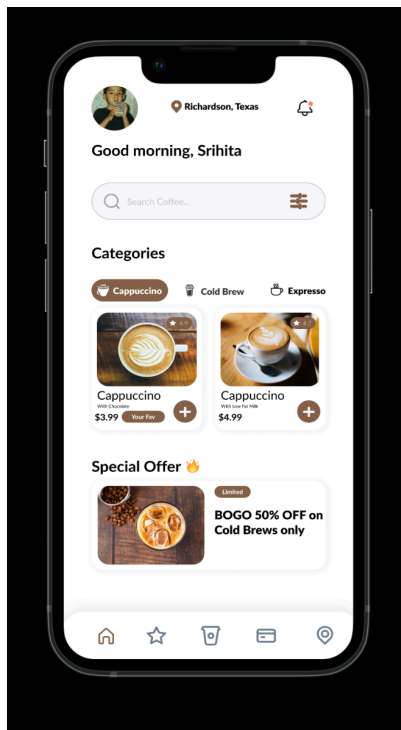
Frame 4: This is the promotions page. It shows the customers deals specifically tailored for them, deals going on that day and the next couple of days and also deals that will stay.

The Bell icon on the top right corner sends out notifications of the coupons exclusively made for that particular customer when they are in the proximity of one of our stores or a competitor's store.

The profile icon shows the customer's information that they have entered and also a questionnaire that they can choose to answer which will help us tailor their drinks according to their preferences.
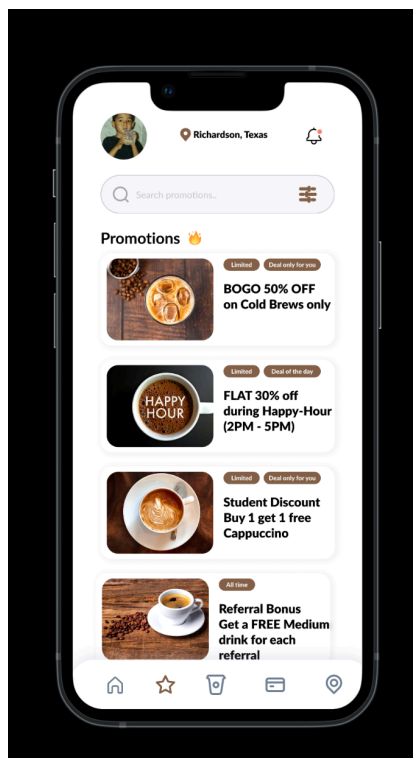


Frame 5: This wallet symbol is our rewards page that gives us information about rewards available now and hidden rewards. The QR code will be scanned for every purchase to add points to the customer's profile.

The coffee cup icon on the left to the wallet icon is your cart where you can checkout.
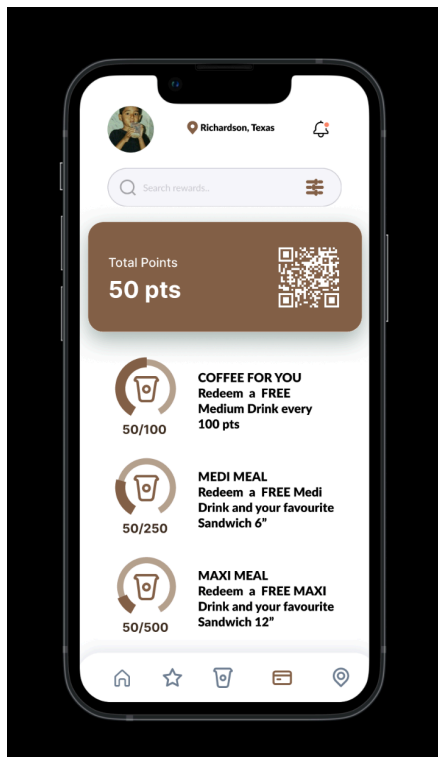
Frame 6: This is the location page. The blue dot on your screen will give us the customer's location and our stores near the customer. The closer stores are represented with bigger icons. Frequently visited stores are marked with an emoji as we also facilitate in-app navigation.
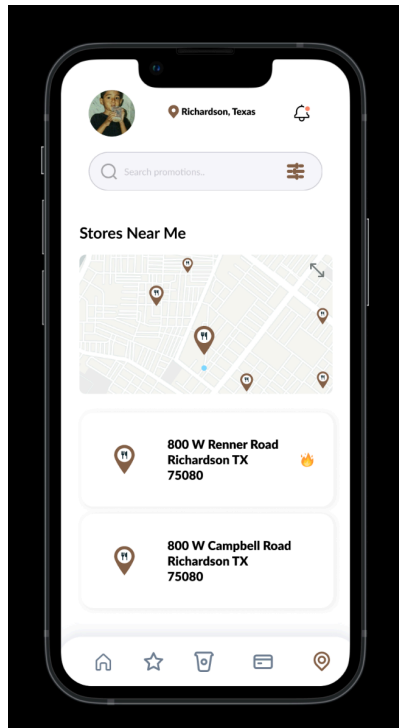
**SOFTWARE DESIGN**

**Signature**

| Method Name: | Class Name: | ID: |
|---|---|---|
| ProvideLocationData() | Location | LocationID |
| Clients (Consumers): | Customers, NeighborBean Cafe | |
| Associated Use Cases: | Provide Location Data | |
| Description of Responsibilities: | Updates the customer's current location data in the system. | |
| Arguments Received: | CustomerID, GeoFenceID, LocationTimestamp | |
| Type of Value Returned: | None | |
| Pre-Conditions: | Customer's mobile device enters the geo-fenced area. | |
| Post-Conditions: | Customer's location data is updated in the system. | |

**Logic**

IF Customer's device enters the geo-fenced area
FETCH `CustomerID` and `GeoFenceID`
UPDATE `LocationTimestamp`

STORE location data in the database

| Method Name: | Class Name: | ID: |
|---|---|---|
| SubmitPreferences() | CustomerPreferences | PreferenceID |
| Clients (Consumers): | Customers, NeighborBean Cafe | |
| Associated Use Cases: | Submit Preferences | |
| Description of Responsibilities: | Stores the customer's preferences for personalized offers. | |
| Arguments Received: | CustomerID, PreferencesData | |
| Type of Value Returned: | None | |
| Pre-Conditions: | Customer logs into the NeighborBean Cafe app. | |
| Post-Conditions: | Customer's preferences are stored in the system. | |

**Logic**

Customer logs into the NeighborBean Cafe app
FETCH `CustomerID`
UPDATE `PreferencesData`
STORE preferences data in the database.

**Signature**

| Method Name: | Class Name: | ID: |
|---|---|---|
| SendMoodData() | CustomerMood | MoodID |
| Clients (Consumers): | Customers, NeighborBean Cafe | |
| Associated Use Cases: | Send Mood Data | |
| Description of Responsibilities: | Stores the customer's mood data for personalized offers. | |
| Arguments Received: | CustomerID, MoodData, MoodTimestamp | |
| Type of Value Returned: | None | |
| Pre-Conditions: | Customer logs into the NeighborBean Cafe app. | |
| Post-Conditions: | Customer's mood data is stored in the system. | |

**Logic**

Customer logs into the NeighborBean Cafe app
FETCH `CustomerID`
UPDATE `MoodData` and `MoodTimestamp`
STORE mood data in the database

**Signature**

| Method Name: | Class Name: | ID: |
|---|---|---|
| ReceiveGeneratedCoupons() | Coupons | CouponID |
| Clients (Consumers): | Customers, NeighborBean Cafe | |
| Associated Use Cases: | Receive Generated Coupons | |
| Description of Responsibilities: | Sends personalized coupons to customers based on their location, preferences, and mood. | |
| Arguments Received: | CustomerID, LocationData, PreferencesData, MoodData | |
| Type of Value Returned: | GeneratedCoupons | |
| Pre-Conditions: | System detects the customer's location, preferences, and mood. | |
| Post-Conditions: | Customer receives the generated coupons. | |

**Logic**

FETCH `LocationData`, `PreferencesData`, `MoodData` for `CustomerID`
GENERATE personalized coupons
SEND `GeneratedCoupons` to the customer

**Signature**

| Method Name: | Class Name: | ID: |
|---|---|---|
| ReceiveNotifications() | Notifications | NotificationID |
| Clients (Consumers): | Customers, NeighborBean Cafe | |
| Associated Use Cases: | Receive Notifications | |
| Description of Responsibilities: | Sends notifications to customers about offers, events, and other relevant information. | |
| Arguments Received: | CustomerID, NotificationData | |
| Type of Value Returned: | None | |
| Pre-Conditions: | System generates a notification based on customer data. | |
| Post-Conditions: | Customer receives the notification. | |

**Logic**

FETCH `NotificationData` for `CustomerID`
SEND notification to the customer

**Signature**

| Method Name: | Class Name: | ID: |
|---|---|---|
| ReceiveHealthAndMoodBasedNotifications() | Notifications | NotificationID |
| Clients (Consumers): | Customers, NeighborBean Cafe | |
| Associated Use Cases: | Receive Health and Mood-Based Notifications | |
| Description of Responsibilities: | Sends notifications tailored to customers' health and mood data. | |
| Arguments Received: | CustomerID, HealthData, MoodData | |
| Type of Value Returned: | None | |
| Pre-Conditions: | System analyzes customer's health and mood data. | |
| Post-Conditions: | Customer receives the notifications. | |

**Logic**

FETCH `HealthData` and `MoodData` for `CustomerID`
ANALYZE data to generate notifications
SEND notifications to the customer

**Signature**

| Method Name: | Class Name: | ID: |
|---|---|---|
| FetchCompetitorLocation() | CompetitorLocation | CompetitorLocationID |
| Clients (Consumers): | Mobile Targeting System, NeighborBean Cafe | |
| Associated Use Cases: | Fetch Competitor Location | |
| Description of Responsibilities: | Fetches the location of competitors' coffee shops. | |
| Arguments Received: | None | |
| Type of Value Returned: | CompetitorLocations | |
| Pre-Conditions: | Periodic update or when customer enters a geo-fenced area. | |
| Post-Conditions: | Competitor locations are updated in the system. | |

**Logic**

PERIODICALLY fetch competitors' locations
UPDATE location data in the database

**Signature**

| Method Name: | Class Name: | ID: |
|---|---|---|
| CreateGeoConquestingOffers() | Offers | OfferID |
| Clients (Consumers): | Mobile Targeting System, NeighborBean Cafe | |
| Associated Use Cases: | Create Geo-Conquesting Offers | |
| Description of Responsibilities: | Creates offers to attract customers away from competitors' locations | |
| Arguments Received: | CustomerID, CompetitorLocation | |
| Type of Value Returned: | GeoConquestingOffers | |
| Pre-Conditions: | Customer is near a competitor's location. | |
| Post-Conditions: | Customer receives the geo-conquesting offers. | |

**Logic**

DETECT customer near a competitor's location
CREATE geo-conquesting offers
SEND offers to the customer

**Signature**

| Method Name: | Class Name: | ID: |
|---|---|---|
| IntegrateData() | DataIntegration | IntegrationID |
| Clients (Consumers): | Mobile Targeting System, NeighborBean Cafe, External Applications | |
| Associated Use Cases: | Integrate Data | |
| Description of Responsibilities: | Integrates data from various external applications. | |
| Arguments Received: | ExternalAppData | |
| Type of Value Returned: | None | |
| Pre-Conditions: | Data synchronization schedule or customer interaction with external apps. | |
| Post-Conditions: | Integrated data is updated in the system. | |

**Logic**

SYNCHRONIZE data from external applications
UPDATE integrated data in the database

**Signature**

| Method Name: | Class Name: | ID: |
|---|---|---|
| ReceiveNotificationsFromExternalApplications() | Notifications | NotificationID |
| Clients (Consumers): | Customers, NeighborBean Cafe, External Applications | |
| Associated Use Cases: | Receive Notifications from External Applications | |
| Description of Responsibilities: | Sends notifications to customers based on data from external applications. | |
| Arguments Received: | CustomerID, ExternalAppData | |
| Type of Value Returned: | None | |
| Pre-Conditions: | System processes integrated data from external applications. | |
| Post-Conditions: | Customer receives the notifications. | |

**Logic**

PROCESS integrated data from external applications
GENERATE notifications based on the data
SEND notifications to the customer

# DATABASE DESIGN
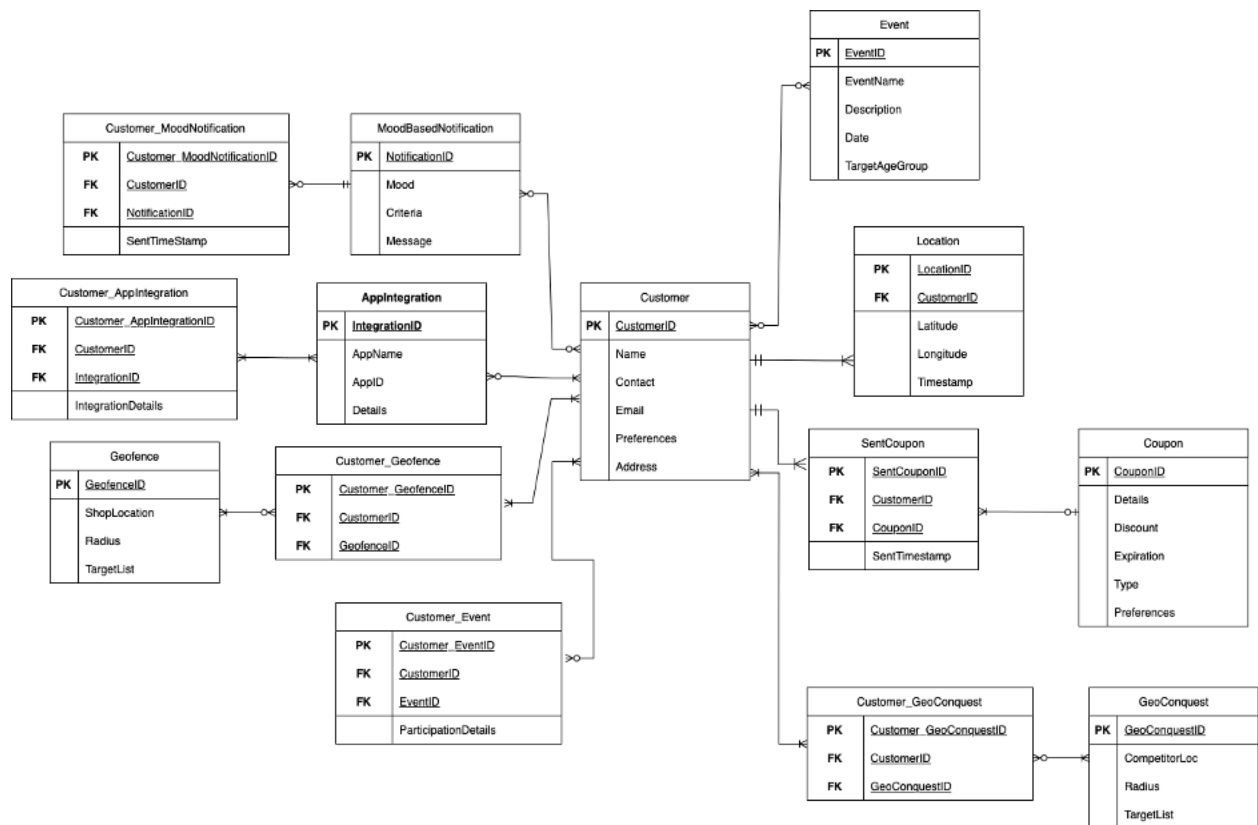


The NeighborBean Cafe's mobile targeting system's database design is set up to efficiently handle and use consumer data in order to present tailored and contextualized ads. An description of the main elements and connections in the database design is provided below:

**Database Constraints**

**Customer Table**

*Constraints:*
Primary Key Constraints/Integrity Constraint
Primary Key `CustomerID` should not be NULL.

*Uniqueness Constraint*
Primary Key `CustomerID` should be unique.

*Domain Specific / Context Specific Constraints*
Attributes `Name`, `Contact`, `Email`, `Preferences`, and `Address` should not have null value.

**AppIntegration Table**

*Constraints:*
Primary Key Constraints/Integrity Constraint
Primary Key `IntegrationID` should not be NULL.

*Uniqueness Constraint*
Primary Key `IntegrationID` should be unique.

*Domain Specific / Context Specific Constraints*
Attributes `AppName` and `AppID` should not have null value.

**Customer_AppIntegration Table**

*Constraints:*
Primary Key Constraints/Integrity Constraint
Primary Key 'Customer_AppIntegrationID' should not be NULL.

*Referential Integrity Constraint*
'CustomerID' should exist as the primary key in the Customer table.
'IntegrationID' should exist as the primary key in the AppIntegration table.

**Geofence Table**

*Constraints:*
Primary Key Constraints/Integrity Constraint
Primary Key `GeofenceID` should not be NULL.

*Uniqueness Constraint*
Primary Key `GeofenceID` should be unique.

*Domain Specific / Context Specific Constraints*
Attributes `ShopLocation`, `Radius`, and `TargetList` should not have null value.

**Customer_Geofence Table**

*Constraints:*
Primary Key Constraints/Integrity Constraint
Primary Key `Customer_GeofenceID` should not be NULL.

*Referential Integrity Constraint*
`CustomerID` should exist as the primary key in the Customer table.
`GeofenceID` should exist as the primary key in the Geofence table.

**Event Table**

*Constraints:*
Primary Key Constraints/Integrity Constraint
Primary Key `EventID` should not be NULL.

*Uniqueness Constraint*
Primary Key `EventID` should be unique.

*Domain Specific / Context Specific Constraints*
Attributes `EventName`, `Description`, `Date`, and `TargetAgeGroup` should not have null value.

**Customer_Event Table**

*Constraints:*
Primary Key Constraints/Integrity Constraint
Primary Key `Customer_EventID` should not be NULL.

Referential Integrity Constraint
`CustomerID` should exist as the primary key in the Customer table.
`EventID` should exist as the primary key in the Event table.

**Location Table**

*Constraints:*
Primary Key Constraints/Integrity Constraint
Primary Key `LocationID` should not be NULL.

*Uniqueness Constraint*
Primary Key `LocationID` should be unique.
Referential Integrity Constraint
`CustomerID` should exist as the primary key in the Customer table.

*Domain Specific / Context Specific Constraints*
Attributes `Latitude`, `Longitude`, and `Timestamp` should not have null value.

**Coupon Table**

*Constraints:*
Primary Key Constraints/Integrity Constraint
Primary Key `CouponID` should not be NULL.

*Uniqueness Constraint*
Primary Key `CouponID` should be unique.

*Domain Specific / Context Specific Constraints*
Attributes `Details`, `Discount`, `Expiration`, `Type`, and `Preferences` should not have null value.

**SentCoupon Table**

*Constraints:*
Primary Key Constraints/Integrity Constraint
Primary Key `SentCouponID` should not be NULL.

*Referential Integrity Constraint*
`CustomerID` should exist as the primary key in the Customer table.
`CouponID` should exist as the primary key in the Coupon table.

**GeoConquest Table**

*Constraints:*
Primary Key Constraints/Integrity Constraint
Primary Key `GeoConquestID` should not be NULL.

*Uniqueness Constraint*
Primary Key `GeoConquestID` should be unique.

*Domain Specific / Context Specific Constraints*
Attributes `CompetitorLoc`, `Radius`, and `TargetList` should not have null value.

**Customer_GeoConquest Table**

*Constraints:*
Primary Key Constraints/Integrity Constraint
Primary Key `Customer_GeoConquestID` should not be NULL.

*Referential Integrity Constraint*
`CustomerID` should exist as the primary key in the Customer table.
`GeoConquestID` should exist as the primary key in the GeoConquest table.

**MoodBasedNotification Table**

*Constraints:*
Primary Key Constraints/Integrity Constraint
Primary Key `NotificationID` should not be NULL.

*Uniqueness Constraint*
Primary Key `NotificationID` should be unique.

*Domain Specific / Context Specific Constraints*
Attributes `Mood`, `Criteria`, and `Message` should not have null value.

**Customer_MoodNotification Table**

*Constraints:*
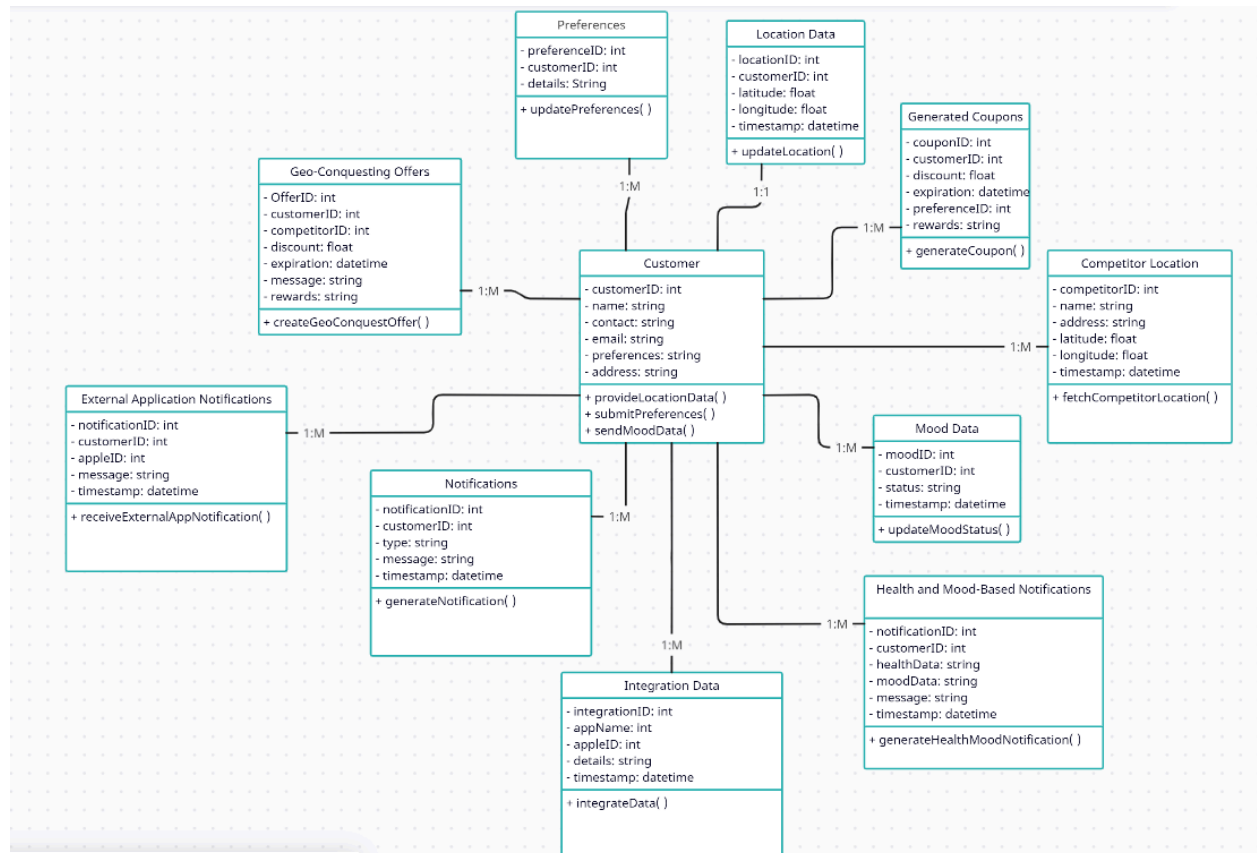Primary Key Constraints/Integrity Constraint
Primary Key `Customer_MoodNotificationID` should not be NULL.

Referential Integrity Constraint
`CustomerID` should exist as the primary key in the Customer table.

`NotificationID` should exist as the primary key in the MoodBasedNotification table.

## COMPLETE CLASS DIAGRAM

| Date | Task |
|---|---|
| 1st June 2024 | Exchanged email ids and phone numbers to ensure easy communication throughout the project. |
| 2nd June 2024 | Got to know the team & started initial project idea brainstorming. |
| 8th June 2024 | Continued brainstorming project ideas, refining and considering feasibility and relevance. |
| 15th June 2024 | Finalized on "NeighborBean Cafe - Mobile Targeting System" and defined project goals and objectives. **Milestone 1: Project Idea finalized.** |
| 22nd June 2024 | Documented the problem and proposed ideas, outlining the business need and objectives. |
| 29th June 2024 | Designed the Context Diagram, identifying external entities interacting with the system and defining system boundaries. |
| 6th July 2024 | Identified actors and processes, designed Use-case Diagram and Process diagram to visually represent interactions and process flow. |
| 13th July 2024 | Wrote Use-case descriptions and developed the Data Dictionary, defining key data entities and their attributes. **Milestone 2: Use-case descriptions and data dictionary completed.** |
| 20th July 2024 | Designed Class Diagrams, defining classes, attributes, and methods, and Sequence Diagrams to detail interactions between objects over time for each use case. |
| 27th July 2024 | Designed Database Diagrams, defining tables, columns, and relationships, and identified Database Constraints to ensure data integrity. **Milestone 3: Database diagrams and constraints identified.** |

| | |
|---|---|
| 3rd August 2024 | Documented methods, detailing the functionality of each method, and developed Pseudo Code to outline logic and flow. |
| 10th August 2024 | Designed Control and Interface, created wireframes and mockups for the user interface, and defined control logic for user interactions. **Milestone 4: Control and Interface design completed.** |
| 17th August 2024 | Compiled all project documents into a cohesive report, proofread for accuracy and clarity, and made the final submission. |
| 24th August 2024 | Prepared presentation slides and content, practiced delivery, and recorded the presentation. **Milestone 5: Report submission and presentation recording completed.** |
| 1st August 2024 | Final day to submit all project-related deliverables. |