QuBarz

Orlando Beavers, Andrew Boudousquie, Thomas Clarke, Dave Murray

Why are we here?

• How much is it to get in?

• Is there a specific wait time?

• Are there any specials going on today?



• Is it crowded, or are there any people there?

Scope of System

We will improve the experience of individuals looking to visit bars/restaurants by providing a mobile application that allows them to see various details of such establishments (including wait times, admission fees, etc.) near them through a map view. We will also improve metrics gathering for owners by providing a feature inside of the mobile application that stores such data to be called at any time.



Detailed Class Diagram Overview



Detailed Class Diagram App Controller

The Application Controller is the backbone of our application. It controls what happens when our application is in use. It keeps track of the account that is logged in, whether that be a customer, owner or guest account. It has a method to log the user out. The method *ShowAccountDetails* will provide the user with all their information about their account. Once the account is logged in, into any of our three accounts, there will be a couple of options for the user: *ShowMap, CheckIn, ShowAccountDetails, LogOut*. If the user clicks on *CheckIn* the application will check to make sure the user is at least close to the venue they have chosen and then offer the user to put in a WaitTime. The *Register* method will pass along the user intent to create an account to the RegisterUser class.

| 1 | Application Controller | | |
|---|-----------------------------------|--|--|
| | -Account : Guest, Customer, Owner | | |
| | -LoggedIn : Boolean | | |
| | +ShowMap() : GoogleMap | | |
| | +LogOut() : Void | | |
| | +LogIn() : Void | | |
| | +ShowAccountDetails() : void | | |
| | +CheckIn(int) : void | | |
| | +Register() : void | | |

1

Detailed Class Diagram Main View

Provides an interface for first starting up the application. It provides a way for the user to enter a username and password, or to use a guest account, or to register.

| Main View | |
|---------------------------------|--|
| -Username : String | |
| -Password : String | |
| +Auth(String, parameter) : void | |
| +Guest() : void | |
| +Register() : void | |
| | |

Detailed Class Diagram Register User

The Register Account class will have an attribute called NewAccount, that will hold an Account object. In this class the setters for the account class will be used.

| Register User | Register Users |
|-----------------------|--------------------|
| -NewAccount : Account | will be the Setter |
| | for Account |
| | |
| | |

Detailed Class Diagram Account

The Account class holds information on accounts. It contains a Username, FirstName, LastName, Email, Password, and Location attribute. It will have for each attribute a getter, the setters are in the Register Account class. The Account class will be inherited by the following classes: Guest, Customer, Owner.

| Account | |
|--|--|
| -Username : String -FirstName : String -LastName : String -EMail : String -Password : String | Getters but w ere not included for simplicity sake |
| | |

Detailed Class Diagram Guest

For users who do not wish to create an account, a guest account can be used. This guest account inherits from Account but leaves the Account attributes blank. The Guest class automatically creates a Guest User ID from Google's User Data ID. Source for Google GUID:

https://developer.android.com/training/articles/user-data-ids.html

| Guest |
|--|
| -GUID : String |
| +getGUID() : String +setGUID(String) : void |
| |

Detailed Class Diagram Customer

The customer class are for users who wish to create a profile with our application. It contains a bit more information from the Account class, such as the following: Ethnicity, Sex, DOB, Age. There are getters and setters for each of the new attributes. These attributes will provide data to the Customer Data class.

|--|

Detailed Class Diagram Owner

The Owner class has an extra attribute Venue alongside the Account information. The Venue attribute will keep track of the venues the owner owns. The Venue attribute is a list in case the Owner owns more than one venue. The venue attribute accepts VenueID objects. The owner may want to provide notifications to users about deals for their venue, this is done through pushNotification. Owners can also access customer data about their venue by using the method getCustomerData. If the owner wants to see their venues or a specific venue they can use the getVenues or getVenue respectively.

| Owner | |
|-------------------------------------|--|
| -Venue : List <venueid></venueid> | |
| +pushNotification(String) : void | |
| +newVenue(Venue) : VenueID | |
| +getVenues() : List <venue></venue> | |
| +getVenue(String) : Venue | |
| +getCustomerData() : CustomerData | |
| | |

Detailed Class Diagram Venue

The venue class are the markers on our map. They contain the following attributes: GoogleID, VenueID, Notification, WaitTime, WaitTimes. The GoogleID will be a Place from google. Google Place source: https://developers.google.com/places/android-api/place-details The VenueID will be used to track individual venues in our database. The Notification will be used to set or get a notification. WaitTime, and WaitTimes keep track of the WaitTimes throughout the day.

| Venue | |
|---|--|
| -GoogleID : String | |
| -VenueID : String | |
| -Notification : String | |
| -WaitTime : int | |
| -WaitTimes : List <waittime></waittime> | |
| +getVenueByGID(String) : Venue | |
| +getPlace(String) : Place | |
| +setVenueNotification(String) : void | |
| +getVenueNotification() : String | |
| +addVenueWaitTime(int) : void | |
| +getVenueWaitTime() : int | |
| +getVenueByID(String) : Venue | |
| +getVenueWaitTimes() : List <waittime></waittime> | |

Detailed Class Diagram Map

The Map class will use Google Map as the backbone. And it will use Google's location in the form of LatLng. The method getMap will be called to show the map with the markers or venues shown on the map. The method ColorMap will be used to show the different wait times on the markers.

| Ма | р |
|--|----|
| -Map : GoogleMap | |
| -Location : LatLng | |
| +getMap() : GoogleMa +ColorMap() : void | ар |

Detailed Class Diagram Customer Data

Get demographics to be used when the Owner wants it. It also provides customer data to the Map class.

Customer Data

+getDemographics(): List<Age, Sex, Ethnicity>

The Register Sequence Diagram has three possible courses of action depending on the input of the user:

1a. The User chooses to register an account. The (a) course of action is then followed to completion.

1b. The User chooses to login using an existing account. The (b) course of action is then followed to completion.

1c. The User chooses to login as a guest, using a GUID provided by Google. The (c) course of action is then followed to completion.











What has changed?

- More heavily focused towards Bars and other small venues
- Competing with apps such as NoWait, BuzzTable, Table's Ready is outside of our scope.
- Have updated models to reflect a 1 to N Owner to Venue Relationship

Known Issues

 Need Notification and WaitTime classes to have CreatedDate, CreatedBy, etc. instead of just string/int respectively

Additional Future Work

 Merge past Guest activity with User Account upon creation by storing Guest activity via GUID

Questions?