Week 3 | Service Layer Redesign

I. Introduction

The VendorBlendor API delivers reliable and secure access to save and retrieve data to support home based business owners, event coordinators and customers

The following features will be required as the MVP of the Vendor Blendor web portal:

- Account Registration
- Event Creation
- Business Listing Creation
- One-Click Registration of Events
- Search and Filter Events
- Retrieval of data for Businesses
- Retrieval of data for Events
- Event Coordinators approve or deny a Business Event Registration
- Event Coordinators marks a Business Owner as paid for an Event

The following features will be considered stretch features for consideration as future enhancement of the Vendor Blendor web portal:

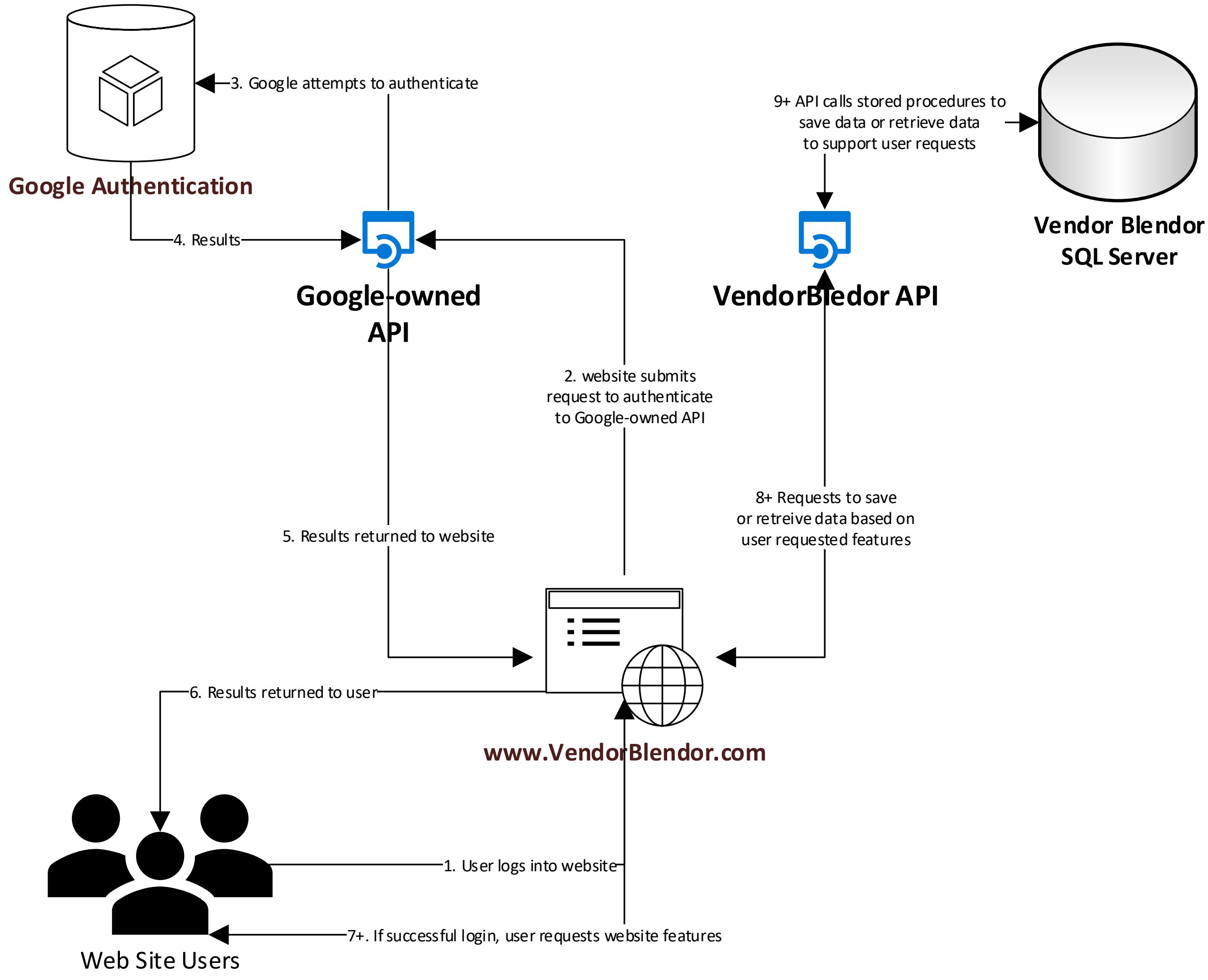
- Rating of Events by customers and business owners
- Rating of Businesses by event coordinators and customers
- My Account screen (personalized)

The API will support REST over JSON messages over HTTP 1.1.

This document will provide the details on the API methods, required parameters, request / response fields and error codes.

II. High-Level REST API Interface Specification

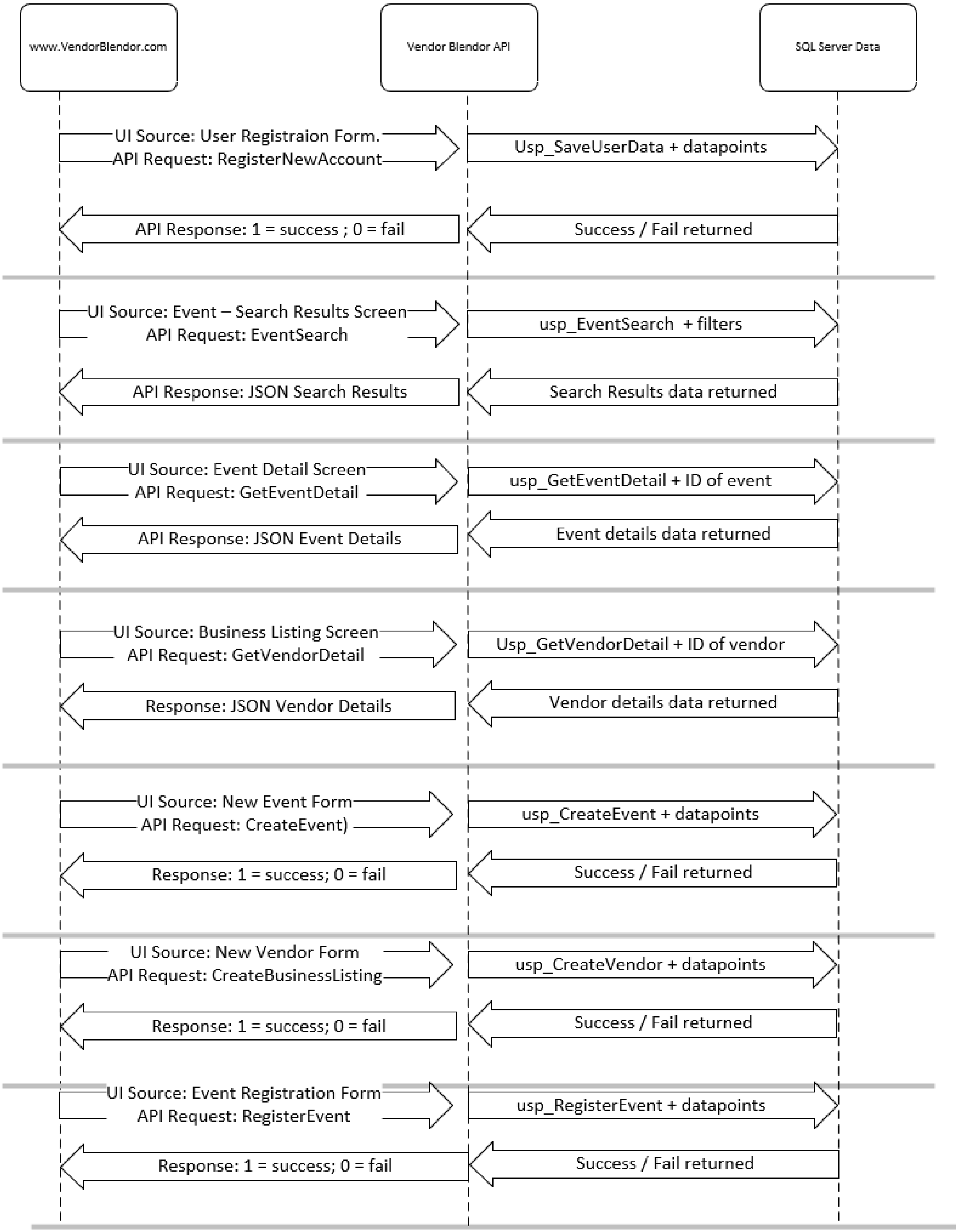
The following diagram gives a high-level view of the components and where the Vendor Blendor API fits into the overall technical architecture.

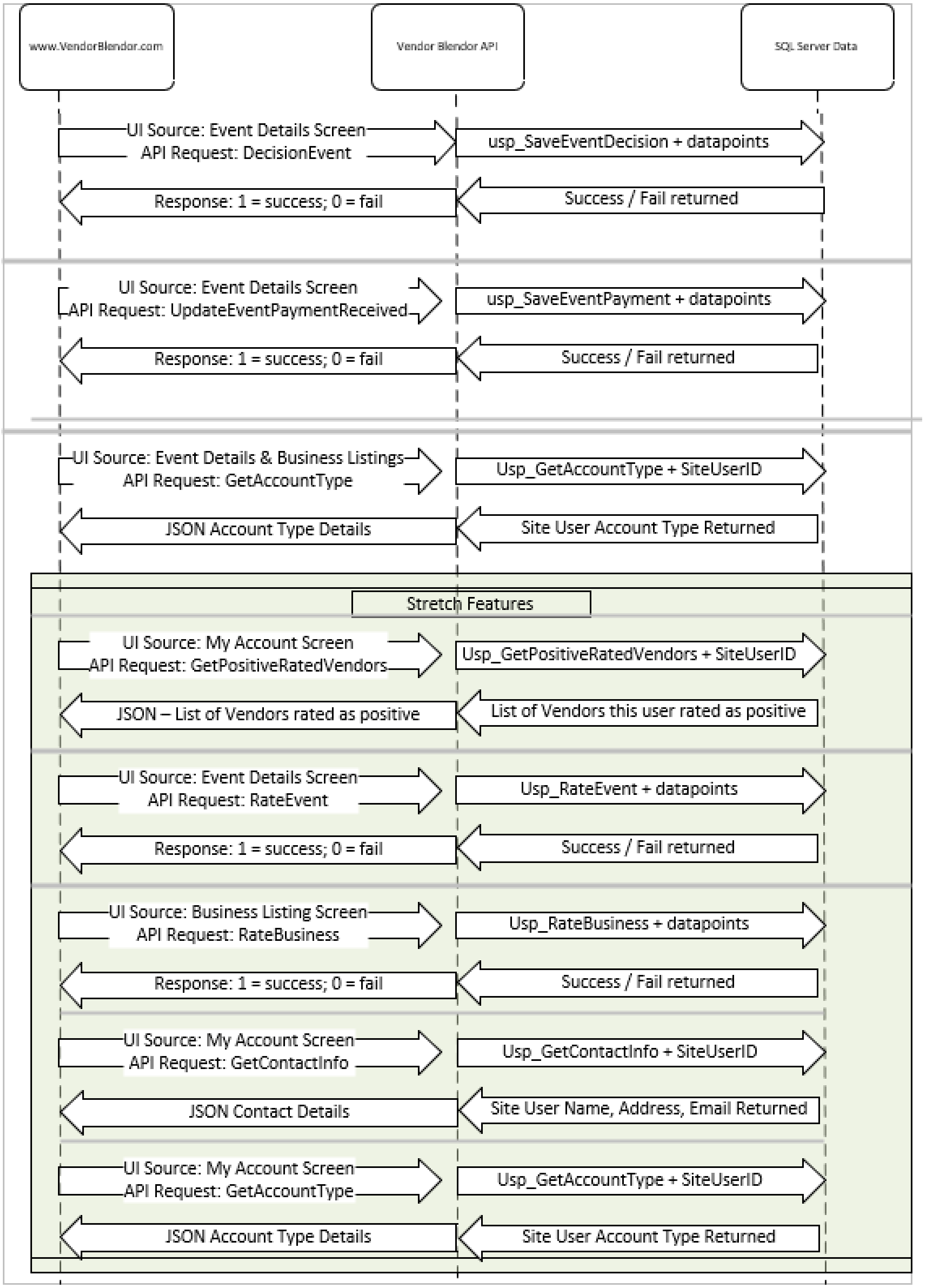


Google Authentication
3. Google attempts to authenticate
4. Results
Google-owned API
9+ API calls stored procedures to save data or retrieve data to support user requests
Vendor Blendor SQL Server
VendorBledor API
2. website submits request to authenticate to Google-owned API
5. Results returned to website
8+ Requests to save or retreive data based on user requested features
6. Results returned to user
www.VendorBlendor.com
1. User logs into website
7+. If successful login, user requests website features
Web Site Users

III. System Interaction Diagram

The following diagram shows interactions between the Web UI, the API and the SQL Server. The items in green are considered stretch features. The remaining items are required for the MVP of the Vendor Blendor web portal.


www.VendorBlendor.com
Vendor Blendor API
SQL Server Data
UI Source: User Registraion Form.
API Request: RegisterNewAccount
Usp_SaveUserData + datapoints
API Response: 1 = success ; 0 = fail
Success / Fail returned
UI Source: Event – Search Results Screen
API Request: EventSearch
usp_EventSearch + filters
API Response: JSON Search Results
Search Results data returned
UI Source: Event Detail Screen
API Request: GetEventDetail
usp_GetEventDetail + ID of event
API Response: JSON Event Details
Event details data returned
UI Source: Business Listing Screen
API Request: GetVendorDetail
Usp_GetVendorDetail + ID of vendor
Response: JSON Vendor Details
Vendor details data returned
UI Source: New Event Form
API Request: CreateEvent)
usp_CreateEvent + datapoints
Response: 1 = success; 0 = fail
Success / Fail returned
UI Source: New Vendor Form
API Request: CreateBusinessListing
usp_CreateVendor + datapoints
Response: 1 = success; 0 = fail
Success / Fail returned
UI Source: Event Registration Form
API Request: RegisterEvent
usp_RegisterEvent + datapoints
Response: 1 = success; 0 = fail
Success / Fail returned

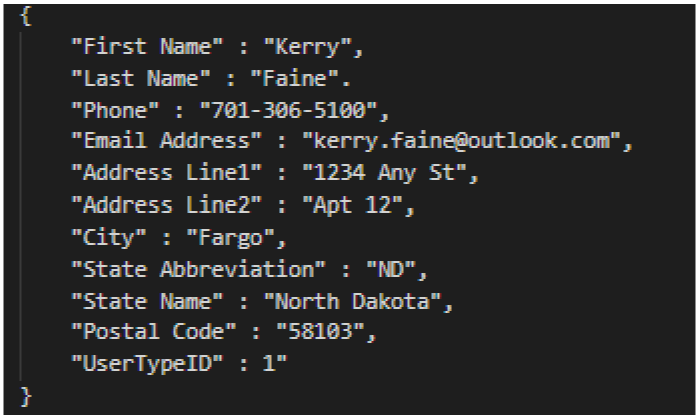

www.VendorBlendor.com
Vendor Blendor API
SQL Server Data
UI Source: Event Details Screen
API Request: DecisionEvent
usp_SaveEventDecision + datapoints
Response: 1 = success; 0 = fail
Success / Fail returned
UI Source: Event Details Screen
API Request: UpdateEventPaymentReceived
usp_SaveEventPayment + datapoints
Response: 1 = success; 0 = fail
Success / Fail returned
UI Source: Event Details & Business Listings
API Request: GetAccountType
Usp_GetAccountType + SiteUserID
JSON Account Type Details
Site User Account Type Returned
Stretch Features
UI Source: My Account Screen
API Request: GetPositiveRatedVendors
Usp_GetPositiveRatedVendors + SiteUserID
JSON – List of Vendors rated as positive
List of Vendors this user rated as positive
UI Source: Event Details Screen
API Request: RateEvent
Usp_RateEvent + datapoints
Response: 1 = success; 0 = fail
Success / Fail returned
UI Source: Business Listing Screen
API Request: RateBusiness
Usp_RateBusiness + datapoints
Response: 1 = success; 0 = fail
Success / Fail returned
UI Source: My Account Screen
API Request: GetContactInfo
Usp_GetContactInfo + SiteUserID
JSON Contact Details
Site User Name, Address, Email Returned
UI Source: My Account Screen
API Request: GetAccountType
Usp_GetAccountType + SiteUserID
JSON Account Type Details
Site User Account Type Returned

2of 2

IV. Method Specifications

The Vendor Blendor API provides the following methods:

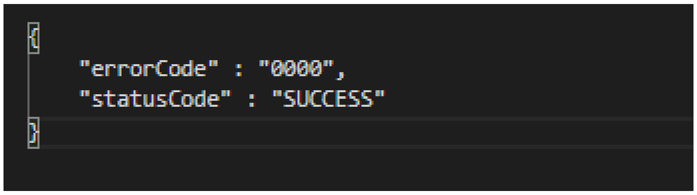
1) RegisterNewAccount
   a) Purpose: Saves the necessary data to create an account in the database.
   b) Target Release: MVP
   c) Stored Procedure: usp_SaveUserData
   d) JSON samples:
      i) Request



```
{
    "First Name" : "Kerry",
    "Last Name" : "Faine".
    "Phone" : "701-306-5100",
    "Email Address" : "kerry.faine@outlook.com",
    "Address Line1" : "1234 Any St",
    "Address Line2" : "Apt 12",
    "City" : "Fargo",
    "State Abbreviation" : "ND",
    "State Name" : "North Dakota",
    "Postal Code" : "58103",
    "UserTypeID" : 1"
}
```

First Name – required : varchar
Last Name – required : varchar
Phone – required : varchar
Email Address – required : varchar
Address Line 1 – required : varchar
Address Line 2 – required : varchar
City – required : varchar
State Abbreviation – required : varchar
State Name – required : varchar
Postal Code – required : varchar
UserTypeID – required : integer

      ii) Response



```
{
    "errorCode" : "0000",
    "statusCode" : "SUCCESS"
}
```

errorCode – integer
statusCode – varchar

2) EventSearch

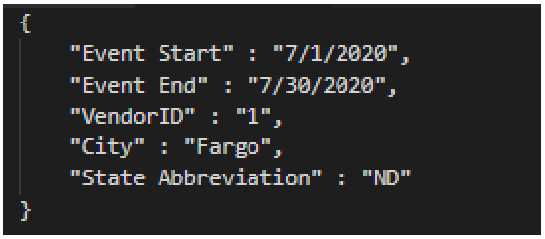
a) Purpose: Performs a search of the events that match the filters chosen by the website user

b) Target Release: MVP

c) Stored Procedure: usp_EventSearch

d) JSON samples :

i) Request



```
{
    "Event Start" : "7/1/2020",
    "Event End" : "7/30/2020",
    "VendorID" : "1",
    "City" : "Fargo",
    "State Abbreviation" : "ND"
}
```

Event Start – optional : Date

Event End - optional : Date

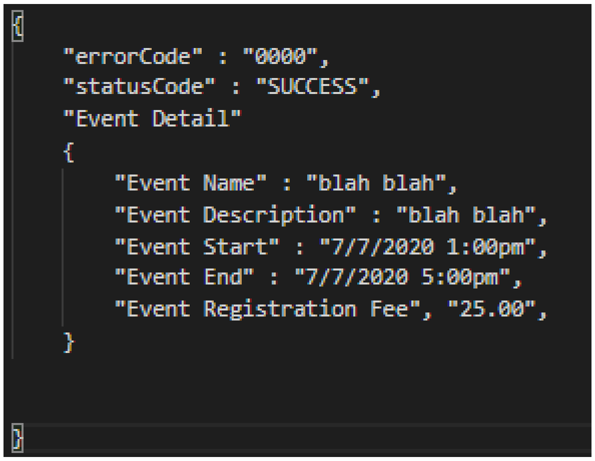
Vendor ID – optional : integer

City – optional : varchar

State Abbreviation – optional : varchar

**Note: if no parameters provided ALL events will be returned.*

ii) Response



```
{
    "errorCode" : "0000",
    "statusCode" : "SUCCESS",
    "Event Detail"
    {
        "Event Name" : "blah blah",
        "Event Description" : "blah blah",
        "Event Start" : "7/7/2020 1:00pm",
        "Event End" : "7/7/2020 5:00pm",
        "Event Registration Fee", "25.00",
    }


}
```

errorCode – integer

statusCode – varchar

Event Name – varchar

Event Description – varchar

Event Start – Datetime

Event End – Datetime

Event Registration Fee – decimal

3) GetEventDetail

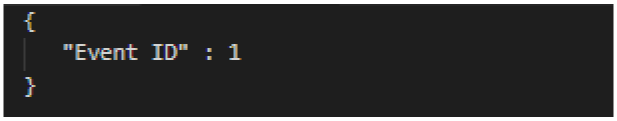
a) Purpose: Retrieves the event data that matches the event ID of the link that the website user clicked.

b) Target Release: MVP

c) Stored Procedure: usp_GetEventDetail

d) JSON samples:

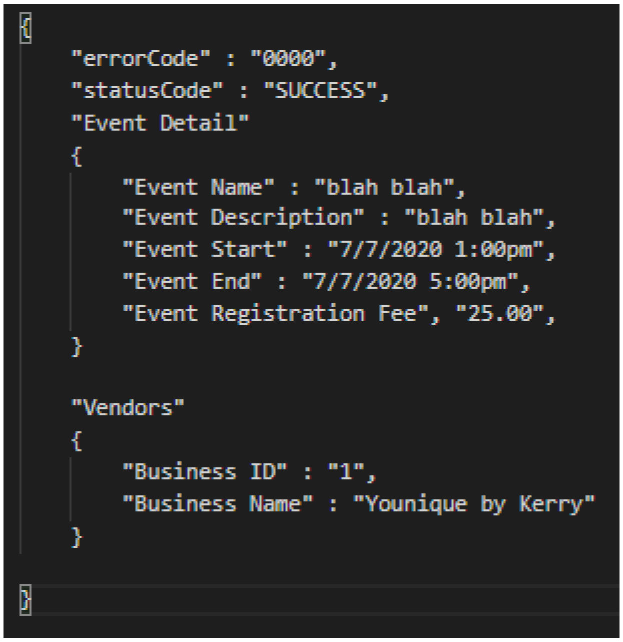
i) Request



```
{
    "Event ID" : 1
}
```

EventID – Required : integer

ii) Response



```
{

    "errorCode" : "0000",
    "statusCode" : "SUCCESS",
    "Event Detail"
    {
        "Event Name" : "blah blah",
        "Event Description" : "blah blah",
        "Event Start" : "7/7/2020 1:00pm",
        "Event End" : "7/7/2020 5:00pm",
        "Event Registration Fee", "25.00",
    }

    "Vendors"
    {
        "Business ID" : "1",
        "Business Name" : "Younique by Kerry"
    }

}
```

errorCode – int
statusCode – varchar
Event Name – varchar
Event Description – varchar
Event Start – datetime
Event End – datetime
Event Registration Fee - decimal
Business ID – integer
Business Name - varchar

4) GetVendorDetail

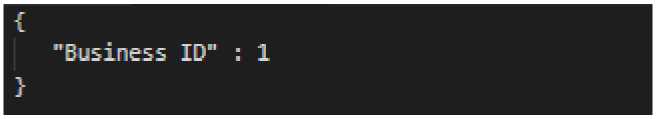
a) Purpose: Retrieves the business data that matches the business ID of the link that the website

b) Target Release: MVP

c) Stored Procedure: usp_GetVendorDetail

d) JSON samples:

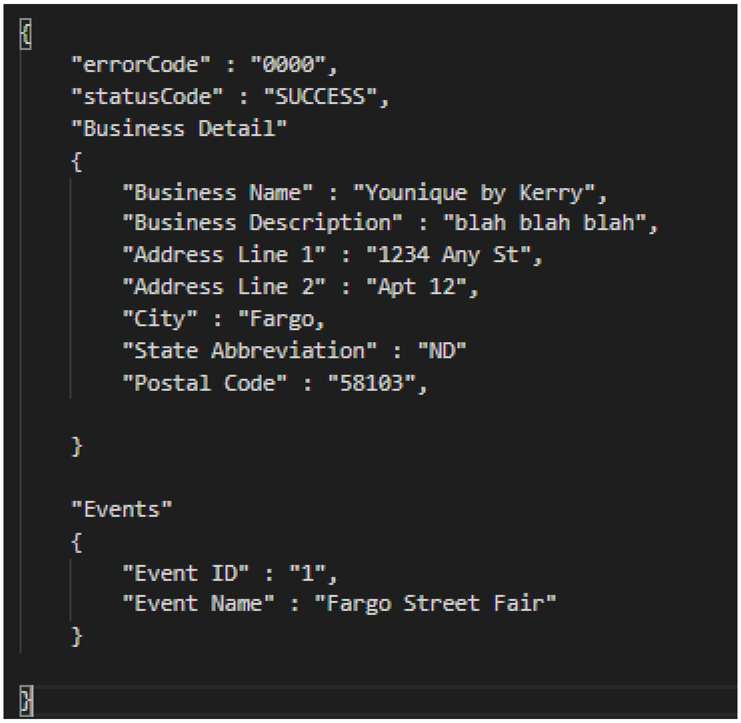
i) Request



```
{
   "Business ID" : 1
}
```

Business ID – required : integer

ii) Response



```
{
    "errorCode" : "0000",
    "statusCode" : "SUCCESS",
    "Business Detail"
    {
        "Business Name" : "Younique by Kerry",
        "Business Description" : "blah blah blah",
        "Address Line 1" : "1234 Any St",
        "Address Line 2" : "Apt 12",
        "City" : "Fargo,
        "State Abbreviation" : "ND"
        "Postal Code" : "58103",

    }

    "Events"
    {
        "Event ID" : "1",
        "Event Name" : "Fargo Street Fair"
    }

}
```

errorCode – integer
statusCode – varchar
Business Name – varchar
Business Description – varchar
Address Line 1 – varchar
Address Line 2 – varchar
City – varchar
State Abbreviation – varchar
Postal Code – varchar
Event ID – integer
Event Name - varchar

5) CreateEvent

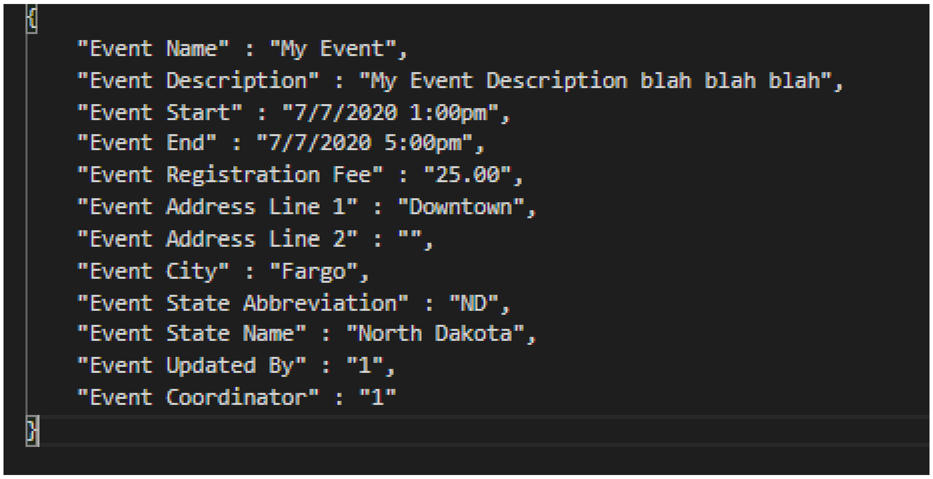
a) Purpose: Saves the necessary data into the database to create an event

b) Target Release: MVP

c) Stored Procedure: usp_CreateEvent

d) JSON samples:

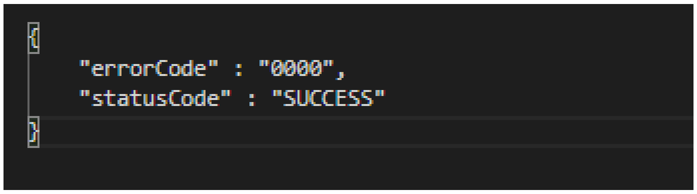
i) Request



```
{
    "Event Name" : "My Event",
    "Event Description" : "My Event Description blah blah blah",
    "Event Start" : "7/7/2020 1:00pm",
    "Event End" : "7/7/2020 5:00pm",
    "Event Registration Fee" : "25.00",
    "Event Address Line 1" : "Downtown",
    "Event Address Line 2" : "",
    "Event City" : "Fargo",
    "Event State Abbreviation" : "ND",
    "Event State Name" : "North Dakota",
    "Event Updated By" : "1",
    "Event Coordinator" : "1"
}
```

Event Name – required : varchar
Event Description – required : varchar
Event Start – required : datetime
Event End – required : datetime
Event Registration Fee – required : decimal
Event Address Line 1 – required : varchar
Event Address Line 2 – required : varchar
Event Address City – required : varchar
Event State Abbreviation – required : varchar
Event State Name – required: varchar
Event Updated By – required : integer
Event Coordinator – required: integer

Response



```
{
    "errorCode" : "0000",
    "statusCode" : "SUCCESS"
}
```

errorCode – int
statusCode – varchar

6) CreateBusinessListing

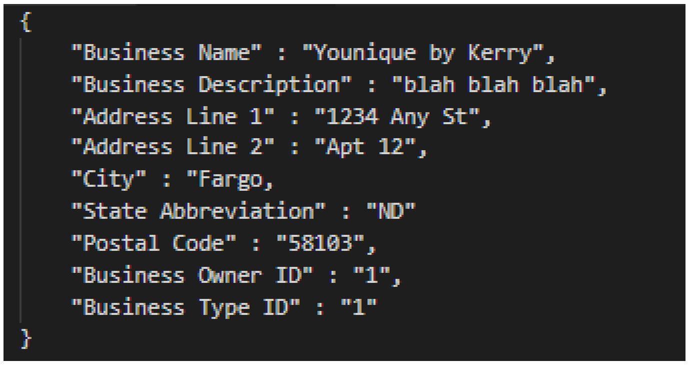
a) Purpose: Creates a new business listing in the database

b) Target Release: MVP

c) Stored Procedure: usp_CreateVendor

d) JSON samples:

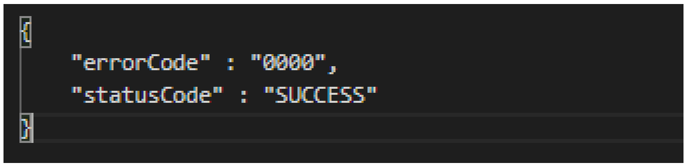
i) Request



```
{
    "Business Name" : "Younique by Kerry",
    "Business Description" : "blah blah blah",
    "Address Line 1" : "1234 Any St",
    "Address Line 2" : "Apt 12",
    "City" : "Fargo,
    "State Abbreviation" : "ND"
    "Postal Code" : "58103",
    "Business Owner ID" : "1",
    "Business Type ID" : "1"
}
```

Business Name – required : varchar
Business Description – required : varchar
Address Line 1 – required: varchar
Address Line 2 – optional : varchar
City – required : varchar
State Abbreviation – required : varchar
Postal Code – required : varchar
Business Owner ID – required: integer
Business Type ID = required: integer

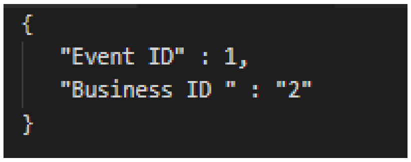
ii) Response



```
{
    "errorCode" : "0000",
    "statusCode" : "SUCCESS"
}
```

errorCode – int
statusCode – varchar

7) RegisterEvent
   a) Purpose: Records a business owner’s registration for an event.
   b) Target Release: MVP
   c) Stored Procedure: usp_RegisterEvent
   d) JSON samples:
      i) Request

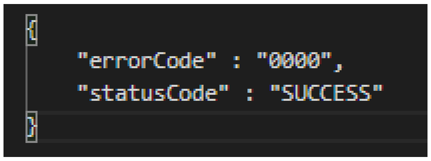


```
{
    "Event ID" : 1,
    "Business ID " : "2"
}
```

Event ID – required : int

Vendor ID – required : int

      ii) Response



```
{
    "errorCode" : "0000",
    "statusCode" : "SUCCESS"
}
```

errorCode – int

statusCode – varchar

8) DecisionEvent

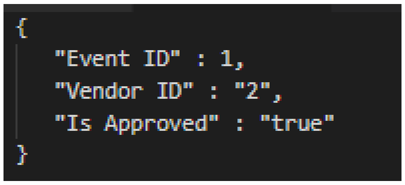
a) Purpose: Allows a coordinator to approve or decline an event registration form

b) Target Release: MVP

c) Stored Procedure: usp_SaveEventDecision

d) JSON samples:

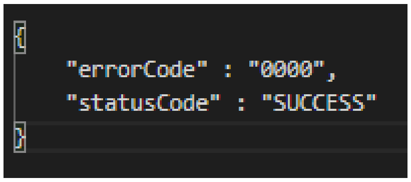
i) Request



```
{
   "Event ID" : 1,
   "Vendor ID" : "2",
   "Is Approved" : "true"
}
```

Event ID – required : int
Vendor ID – required : int
Is Approved – required : bool (true / false)

ii) Response



```
{
    "errorCode" : "0000",
    "statusCode" : "SUCCESS"
}
```

errorCode – int
statusCode – varchar

9) UpdateEventPaymentReceived

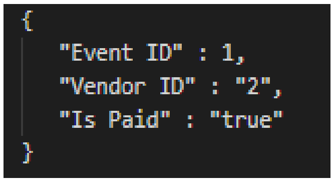
a) Purpose: Allows a coordinator to mark business owner as paid for a given event

b) Target Release: MVP

c) Stored Procedure: usp_SaveEventPayment

d) JSON samples:

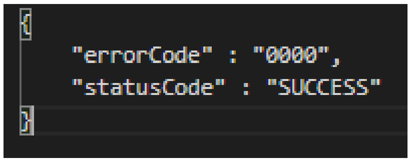
i) Request



```
{
    "Event ID" : 1,
    "Vendor ID" : "2",
    "Is Paid" : "true"
}
```

Event ID – required : int
Vendor ID – required : int
Is Approved – required : bool (true / false)

ii) Response



```
{
    "errorCode" : "0000",
    "statusCode" : "SUCCESS"
}
```

errorCode – int
statusCode – varchar

10) GetAccountType

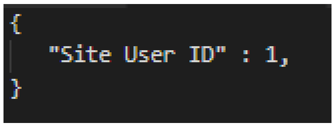
  a) Purpose: Allows for dynamically controlling what controls are visible on the screen or restricting activites to a given role.

  b) Target Release: MVP

  c) Stored Procedure: usp_GetAccountType

  d) JSON samples

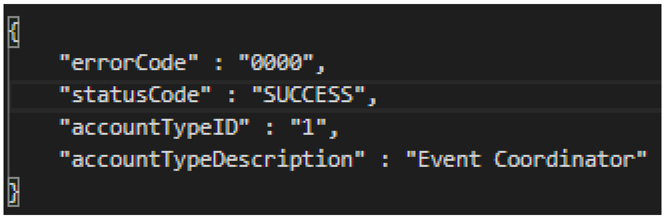
i) Request



```
{
   "Site User ID" : 1,
}
```

Site User ID – required: int

ii) Response



```
{
    "errorCode" : "0000",
    "statusCode" : "SUCCESS",
    "accountTypeID" : "1",
    "accountTypeDescription" : "Event Coordinator"
}
```

errorCode – int

statusCode – varchar

accountTypeID – int

accountTypeDescription - varchar

11) GetPositiveRatedVendors

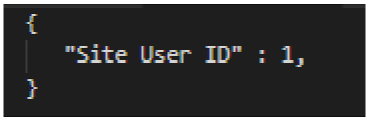
a) Purpose: Allows for pulling a list of the vendors the logged in user has marked with positive results. This will allow a user to stay connected to their favorite vendors from the My Account Screen.

b) Target Release: Stretch

c) Stored Procedure: usp_GetPositiveRatedVendors

d) JSON samples:

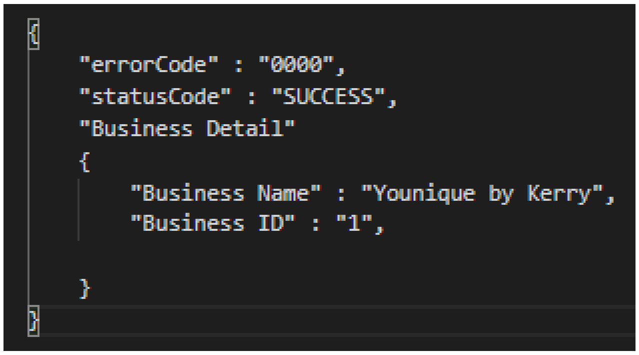
i) Request



```
{
    "Site User ID" : 1,
}
```

Site User ID – required : int

ii) Response



```
{

    "errorCode" : "0000",
    "statusCode" : "SUCCESS",
    "Business Detail"
    {
        "Business Name" : "Younique by Kerry",
        "Business ID" : "1",

    }
}
```

errorCode – int
statusCode – varchar
Business Name – varchar
Business ID – int

12) RateEvent

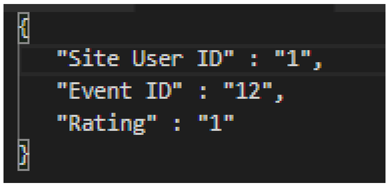
a) Purpose: Allows for the save of user rating activities for events.

b) Target Release: Stretch

c) Stored Procedure: usp_RateEvent

d) JSON Samples

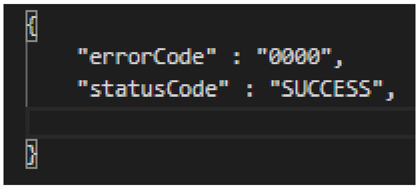
i) Request



```
{
    "Site User ID" : "1",
    "Event ID" : "12",
    "Rating" : "1"
}
```

Site User ID – required : int
Event ID – required : int
Rating – required : int

ii) Response



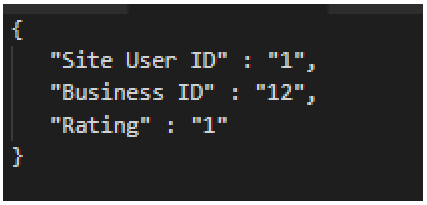
```
{
    "errorCode" : "0000",
    "statusCode" : "SUCCESS",

}
```

errorCode – int
statusCode – varchar

13) RateBusiness

a) Purpose: Allows for the save of user rating activities for businesses.
b) Target Release: Stretch
c) Stored Procedure: usp_RateBusiness
d) JSON Samples

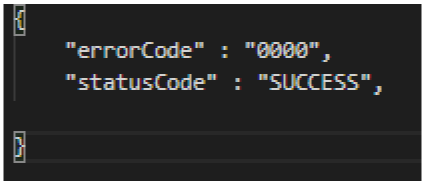
i) Request



```
{
   "Site User ID" : "1",
   "Business ID" : "12",
   "Rating" : "1"
}
```

Site User ID – required : int
Business ID – required : int
Rating – required : int

ii) Response



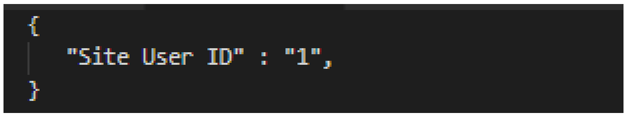
```
{
    "errorCode" : "0000",
    "statusCode" : "SUCCESS",

}
```

errorCode – int
statusCode - varchar

14) GetAccountInfo

a) Purpose: Allows the retrieval of the basic contact information for customers, businesses and event coordinators.
b) Target Release: Stretch
c) Stored Procedure: usp_GetContactInfo
d) JSON Samples

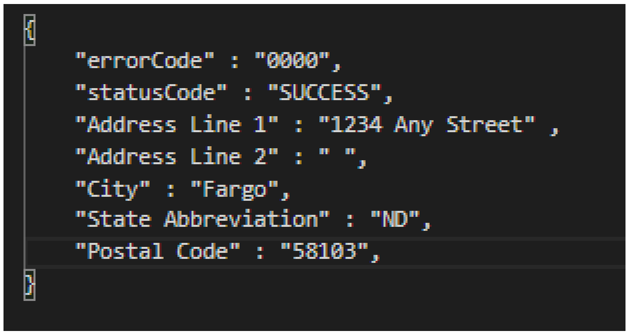
i) Request



```
{
    "Site User ID" : "1",
}
```

Site User ID – required : int

ii) Response



```
{
    "errorCode" : "0000",
    "statusCode" : "SUCCESS",
    "Address Line 1" : "1234 Any Street" ,
    "Address Line 2" : " ",
    "City" : "Fargo",
    "State Abbreviation" : "ND",
    "Postal Code" : "58103",
}
```

errorCode – int
statusCode – varchar
address line 1 – varchar
address line 2 – varchar
City – varchar
State Abbreviation – varchar
Postal Code – varchar

V. Exception Processing

The calling applications must handle the following two type of generated errors:

1) HTTP Response is not 200
2) Vendor Blendor API is not SUCCESS (code = 0000)

HTTP Error Codes include

1) 400 – Bad Request
2) 401 – Unauthorized
3) 404 – Not Found
4) 500 – Internal Server Error

Vendor Blendor API Codes include

1) 0000 – Success
2) 0001 – Fail

All Vendor Blendor failures will be recorded in an error log for troubleshooting purposes.

The calling applications must handle data validation on user forms to ensure proper data is being sent to the API.