

August 12, 2016

**ADDENDUM NO. 4
FOR
REQUEST FOR PROPOSAL (“RFP”) FOR PAYMENT AND RECEIPT OF
RESIDENTIAL DAILY PARKING PERMIT VIA ELECTRONIC MEANS
SPECIFICATION NO. OCC2016.1**

For which bids are scheduled to open in the Office of the City Clerk, City Hall, 121 N. LaSalle Street Room 107 Chicago, Illinois 60602, at 4:00 p.m., on August 22, 2016

BIDDERS MUST ACKNOWLEDGE RECEIPT OF THIS ADDENDUM

<u>CLARIFICATION TO QUESTIONS</u>	
Question 1	With the goal of 25% MBE and 5% WBE, can the vendor fulfill the entire goal with one subcontractor satisfying both the MBE and WBE requirement, or does this need to be two separate vendors?
Response	<i>No. Separate vendors are needed. If the respondent is certified MBE or WBE, another certified vendor is needed to satisfy the remaining requirement.</i>
Question 2	Fully understanding the limitations of pricing to the customer (citizen), Can the vendor assess development and/or software maintenance and hosting fees to the City? The City Code referenced, does not include fees vendors can apply to development of solutions.
Response	<i>No. Per 6.2.7 Compensation Schedule, Any contractor or vendor shall derive its entire compensation by collecting a fee in accordance with the City of Chicago Municipal Code Section 9-68-021. Please note: Pending Ordinance O2016-5593 is being considered by the City Council to increase the amount the vendor may charge the customer.</i>
Question 3	Are advertising allowed in the app?
Response	<i>No.</i>

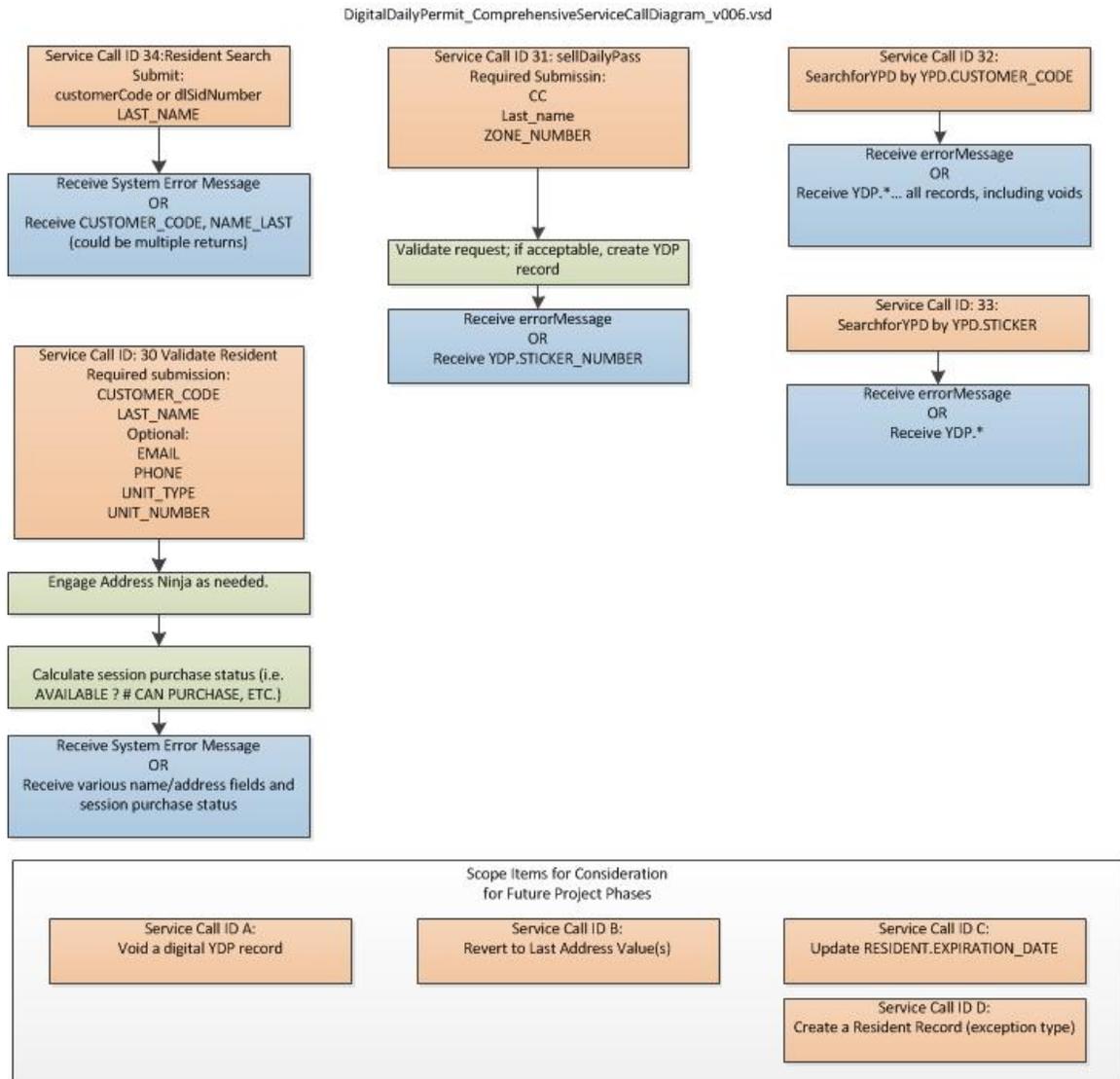
Revision No. 1. The Following section is to be added to “Exhibit 1 (1) a. i. Integration with OCC system” on page 19:

These service call high-level functionalities are outlined hereafter. In summary, there are five service calls for the pilot program:

- Search for Resident
- Validate Resident
- Sell Daily Permit

- Search for Daily Permit by customer code
- Search for Daily Permit by Permit Unique Identifier

The following diagram gives a general overview of these functionalities (orange represents vendor call; green represents internal SAS functional activity; blue represents responses provided by 3PS-API):



Revision No 2. Remove, “Exhibit 1 (1) a. i. 1. Resident management” on page 19 and replace with the following:

1. Search for Resident:

Vendor application makes a call to 3PS-API: find resident record. Vendor application has two options: submit DLSID+LastName or submit CustomerCode+LastName. This is information the customer will already be privy to.

3PS-API responds with an affirmative or negative response. In the case of the affirmative, a validated customer_code value + last name record(s) will be returned. (Notably, the customer_code uniquely identify the record and enable the Vendor application to use this information for down-stream service calls). In the case of the negative, an error message will be returned; for example, the search/update call was made with information that does not match any existing resident record. Please note that more than one record can be returned by this call. For example, client app submitted a DLSID+lastName, and two records were returned: one record which is the customer's record that's linked to the SOS dataset, and another record which was created by an OCC user to allow the customer to purchase at a second property the customer owns.

2. Validate Resident:

Vendor application makes a call to 3PS-API: validate resident. Vendor must submit customer_code and lastName. Vendor application has the option to also submit a subset of Resident attributes to be incorporated into the record, specifically: phone_number, email_address, Unit_type, and unit number.

Upon receiving the call, 3PS-API will perform an internal automated process of updating the record; for example, if for any reason the record currently exists in a state where the address hasn't yet been fully processed through SAS's "address cleanser/parser" (which incorporates an API from the United States Postal Service, as well as a GIS system provided by the City of Chicago's Department of Innovation and Technology), then SAS will automatically process the address to be in a usable state. SAS will also calculate a number of real-time "session" attributes related to the record, for example: is the record in a zone that's eligible to purchase, how many sheets can the record currently be sold (i.e. 0, 1, 2 or 3 based on sales in the last 30 days to the customer/household), and if no sheets can be sold at present time due to recent purchases, what is the soonest date when the person will be able to purchase again.

3PS-API responds with an affirmative or negative response. In the case of the affirmative, various data fields will be returned to the vendor app, reflecting the successful service call (see below for additional detail). In the case of the negative, an error message will be returned; for example, the call was sent with information that does not match any existing records.

The defined database fields that shall be returned include:

database field name	return upon validate resident call:	Rule
EXPIRATION_DATE	YES	
CUSTOMER_CODE	YES	
NAME_FIRST	YES	
NAME_LAST	YES	
NAME_MIDDLE	YES	
CITY	YES	
STATE	YES	
ZIP_CODE	YES	
ADDRESS_TYPE	YES	
ZIP_4	YES	
WARD_2015	YES	
ADDRESS	see rule	if RESIDENT.ADDRESS_TYPE = NONSTANDARD, return this field. If else, do not return this field.
HOUSE_NUMBER	see rule	if RESIDENT.ADDRESS_TYPE = STANDARD, return this field. If else, do not return this field.
DIRECTION	see rule	if RESIDENT.ADDRESS_TYPE = STANDARD, return this field. If else, do not return this field.
STREET_NAME	see rule	if RESIDENT.ADDRESS_TYPE = STANDARD, return this field. If else, do not return this field.
STREET_TYPE	see rule	if RESIDENT.ADDRESS_TYPE = STANDARD, return this field. If else, do not return this field.
UNIT_TYPE	see rule	if RESIDENT.ADDRESS_TYPE = STANDARD, return this field. If else, do not return this field.
UNIT_NUMBER	see rule	if RESIDENT.ADDRESS_TYPE = STANDARD, return this field. If else, do not return this field.
SECOND_ST_DIR	see rule	if RESIDENT.ADDRESS_TYPE = STANDARD, return this field. If else, do not return this field.

Please note that RESIDENT.ADDRESS_TYPE is a value that is automatically calculated by the SAS “Address cleanser/parser). SAS will have extensive validation rules and error messaging system related to the aforementioned functionality. The vendor application will be required to ensure effective customer experience. For example, 3PS-API will error any attempt to update a UNIT_NUMBER / UNIT_TYPE when the existing record presently has an ADDRESS_TYPE = NONSTANDARD.

In addition to the database fields outlined above, SAS shall also provide the following “session fields” aka “attributes”:

Attribute	Description
sosBased	Some resident records were created via the OCC’s data sharing program with the Secretary of State; others were created manually by OCC end users. This attribute field shall distinguish between the two. This is commonly useful information for various stakeholders, including customers.
Status	Indicates a general “ability” to purchase sheets: EXPIRED means the record is expired, and an OCC user needs to update the record in order to sell once again. AVAILABLE means the record can be used to purchase, as long as has zone values (see below) and available sheets (see below)
zoneList	This is the full list of zone numbers for which this record can purchase. Records that do not have zone privilege will have no zoneList values returned. Records that do have zone privilege will have all such zone numbers returned here. Of those records who have a zone offering, the vast majority have only one such value.
AvailablePasses	This is the real-time count of sheets that can be purchased against the record, based on last 30 days purchase history.
rdpNextAvailableDate	When the availablePasses count is zero, this attribute will communicate the next future date when the record can be used to purchase again.

Vendor app (developed by vendor, in collaboration with the OCC, ultimately under the supervision of the OCC) will need to incorporate all of these values into a unified message to customers/endUsers so as to succinctly communicate the current ability to purchase sheets and for what zone.

Revision No 3. “Exhibit 1 (1) a. i. 2. Create Daily Permit Sheet(s)” on page 19 is changed to “Exhibit 1(1) a. i. 3. Create Daily Permit Sheet(s)”

Revision No 4. Remove “Exhibit 1 (1) a. i. 3. Resolve Issues” on page 19.

Revision No 5. The following sections are added to “Exhibit 1 (1) a. i. Integration with OCC systems” on page 19:

4. Search for Daily Permit by Daily Permit Number

Vendor application makes a call to 3PS-API: find yrs daily permit record, using daily permit number and lastName.

3PS-API responds with an affirmative or negative response. In the case of the affirmative, the full daily permit record shall be returned including all relevant database fields, including the last-date that daily passes from the sale can be used by the customer. This information is critical for overall business process management. In the case of the negative, an error message will be returned; for example, the search/update call was made with information that does not match any existing permit record.

Notably, this “retrieve data” call provides all those fields related to the sale which were not returned in the original “create YRS Daily Permit” call; original “create yrs daily permit” call only results in the return of the YRS daily pass unique identifier. The overall activity of creating a record and providing full details about that newly-created daily pass record were bifurcated into two calls for tactical technological reasons.

5. Search for Daily Permit by Customer Code

Vendor application makes a call to 3PS-API: find yrs daily permit record, using customer code and lastName.

3PS-API responds with an affirmative or negative response. In the case of the affirmative, the full list of daily permits (probably limited to last 30 or 60 days) that have been created for this customer code shall be returned. In the case of the negative, an error message will be returned; for example, the search/update call was made with information that does not match any existing permit record.

The purpose of this call is to enable Vendor App to display purchase history to the customer, including both digital daily permit records (sold by vendor app) as well as physical daily permit record (sold by OCCs sellers) ; this information is extremely helpful for enabling the customer to understand those circumstances which resulting in them being blocked from further purchase due to recent purchase history. Daily permit records include a field called PERMIT_TYPE which clearly identifies whether a permit record was physical or digital.

6. Future Scope Considerations

The following functionalities have been identified for further analysis, discussion and possible implementation during the life of the digital daily permit program. These services are not immediately in scope, but could be, based on a state the business could evolve to, in discussion with the awarded vendor.

- (a) *Void a Digital Daily Permit record.* This would allow a digital daily permit record to be voided by awarded vendor's system administrators (not customers / end users), in those cases where the record was created wrongly due to technological error or gross user error. Note that because daily permits are generally speaking not transferrable or refundable, this voiding activity would be premised upon extensive discussion and validation of business practices.
- (b) *Revert to previous address.* As outlined in "Validate Resident," SAS-API will allow customers to update their records. Should the customer/app enter an errant value, it can render a record suddenly useless. For example, if customer/app submit a patently invalid unit number such as '%F&*CV#H*', SAS will often invariably and immediately save the record's address in a state that suddenly makes the record ineligible to purchase (because the address will be in such a gross state that the application can't determine its address-based zone matches). This service call would allow the end user to basically "undo" the errant unit_type / Unit_number submission and thus revert back to the previous (valid) address state.
- (c) *Update Expiration Date, Update Full Address, and Create a Resident Record.* OCC's present business practice is to not allow third party sellers to create resident records, nor update full addresses, nor to allow updating of resident record expiration dates. This means that the duties of receiving evidentiary documentation (i.e., photo of license, scanned image of new lease with new expiration date, etc.... often via email), validating such documentation, and then manually updating the record accordingly... falls on CoC staff. The service calls mentioned here would enable such activity to be performed by third party sellers (specifically, system administrators / call center members).

**ALL REVISIONS INSCRIBED HEREIN ARE HEREBY INCORPORATED INTO THE
BID SPECIFICATION**

END OF ADDENDUM NO. 4

**CITY OF CHICAGO
OFFICE OF THE CITY CLERK**

**SUSANA A. MENDOZA
CITY CLERK**