Food Truck Location Assignment and Draft System

April 30th, 2015
Siamak Khaledi, Ankit Shah, Matthew Shoaf

DMVFTA
DC, Maryland and Virginia
Food Truck Association

GEORGE MASON UNIVERSITY

Table of Contents

1. Problem Definition and System Overview

1.1. Problem

The introduction of mobile restaurant platforms ("food trucks") into the DC metropolitan area economy has created a problem for the Department of Consumer and Regulatory Affairs (DCRA). There exist a limited number of available parking locations for potential food truck use at any given time. DCRA is presented with the problem of providing food trucks with equitable access to these parking locations.

To answer this problem, DCRA has recently developed a lottery-based monthly assignment mechanism for assigning parking locations in popular food truck zones. Unfortunately, this system has been met with dissatisfaction by the food truck vendors as they have complained about an unfair system that is also not easy to use. DCRA is now concerned with underutilization of assigned parking locations and strategic gaming. There is also a fear of abandonment of the lottery system by the food truck vendors. DC Maryland Virginia Food Truck Association (DMVFTA) is looking to find a solution to assign the parking locations fairly to the vendors.

1.2. Mission Statement

We intend to provide DMVFTA with a prototype of an alternative primary market mechanism that assigns food truck vendors to available locations in an equitable manner. We also intend to demonstrate the "fairness" of our system in a quantifiable way, through a longitudinal analysis of vendor preferences and resulting assignments.

2. System Overview

2.1. Guiding Principles

Following a detailed qualitative and quantitative analysis of DCRA's problem domain, our team determined that weaknesses in the current system fell into three categories: (1) effective equitability, (2) vendor perception of equitability and (3) system usability. These three weaknesses are the guiding principles for our design. These guiding principles are as follows:

2.3.1. Provide an even distribution of access to high-demand food truck parking locations.

2.3.2. Ensure system transparency.

2.3.3. Provide vendors with an easy-to-use system that provides options, rather than dictating outcomes.

2.2. Strategic Approach

Based on the problem statement, analysis of the data and following our guiding principles, we are introducing the concept of a drafting mechanism to give the control back to the vendors. Our research and literature review included sports based drafting algorithm (Snake draft, Ladder draft, NBA draft). Our algorithm takes inspiration from the newly proposed NBA Wheel draft, which is explained in detail in Section 5.1: The FTLADS Model.

3. Requirements

The following system requirements were defined during requirement elicitation:

3.1. Stakeholder Requirements

3.1.1 The system shall receive parking location preferences from food truck vendors.
3.1.2 The system shall match food truck vendors to parking locations based upon selected preferences.
3.1.3 The system shall output location assignments to vendors.
3.1.4 The system shall de-conflict where vendor preferences are greater than location availability.
3.1.5 The system shall match vendors to locations in an equitable way, where “equitability” is considered to be a measure of vendor preference values versus assigned locations.
3.1.6 The system shall not employ auctioneering or bidding to valuate parking spaces monetarily.
3.1.7 The system shall provide web access.

3.2. Derived Requirements

3.2.1. The system shall include a user interface for vendors.
3.2.2. The system shall utilize an algorithm that assigns parking spaces to vendors based on user preferences.
3.2.3. The system shall ensure that preference distributions are not clustered over short intervals.
3.2.4. The system shall utilize a structured query database to store user profile information and process user requests.
3.2.5. The system shall maintain historical location preference data.
3.2.6. The system shall provide secure access.
3.2.7. The system shall be able to process payment transactions.
3.2.8. The system user interface shall be compatible with the following browsers: (1) MS Internet Explorer version 8.0 or later, (2) Mozilla Firefox version 3.0 or later and (3) Google Chrome version 41.0 or later.

4. System Architecture

4.1. Architectural Design

All operations executed by the FTLADS are through five primary sub-systems: (a) the FTLADS draft model, (b) a central database, (c) the system’s user interface, (d) payment processing sub-system and (e) the location trading subsystem.[1] These sub-systems accomplish the following

[1] The secondary trading mechanism was developed by another group of GMU students in Fall 2014. This document provides a design for future integration of this platform with their algorithm.

tasks.

4.1.1. The FTLADS draft model will:

4.1.1.1. Assign food truck vendors draft picks (e.g. 1st through Nth choice) for locations, thus de-conflicting preferences [*Requirements 3.1.2; 3.1.4; 3.1.6 and 3.2.2*]

4.1.1.2. Assign draft picks such that they are evenly distributed over some period of time, T [*Requirement 3.1.5*]

4.1.1.3. Space high-valued draft picks over even intervals [*Requirement 3.2.3*]

4.1.2. The system's central database will:

4.1.2.1. Utilize SQL tables to store user profile data, preferences and current draft assignments [*Requirement 3.2.4*]

4.1.2.2. Maintain historical data relevant to vendor location preferences and draft assignments [*Requirement 3.2.5*]

4.1.2.3. Feature secure access protocols [*Requirement 3.2.6*]

4.1.3. The system's user interface will:

4.1.3.1. Provide food truck vendors with web-based forms to enter profile information, make preference selections and make payments [*Requirements 3.1.1; 3.1.7; 3.2.1 and 3.2.7*]

4.1.3.2. Provide food truck vendors with draft outputs and location assignments [*Requirement 3.1.3*]

4.1.4. The system's payment processing sub-system will:

4.1.4.1. Receive payment inputs from the user interface sub-system and transfer them to external payment processing systems

4.1.4.2. Feature secure access protocols

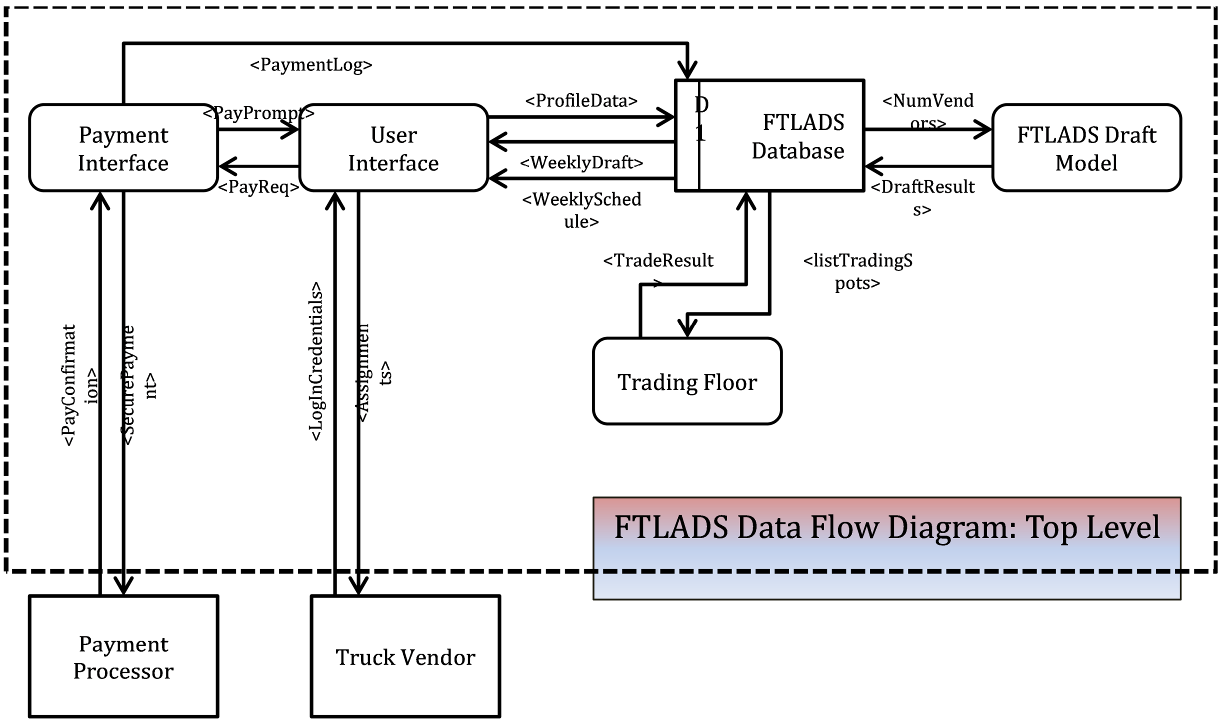
4.1.5. The system's location trading subsystem will:

4.1.5.1. Provide users with the ability to trade assignments with other users via a 1-to-1 swapping pool

These five sub-systems compromise the full range of tasks needed to meet DMVFTA's requirements.

This SDD focuses on subsystems A-C. A location trading sub-system has been developed by another team and is currently being implemented and a functional payment processing capability is already in place. Thus, sub-systems D and E are incorporated into our top-level designs.

The top-level data flow structure of FTLADS provides a visual overview of how the system operates.



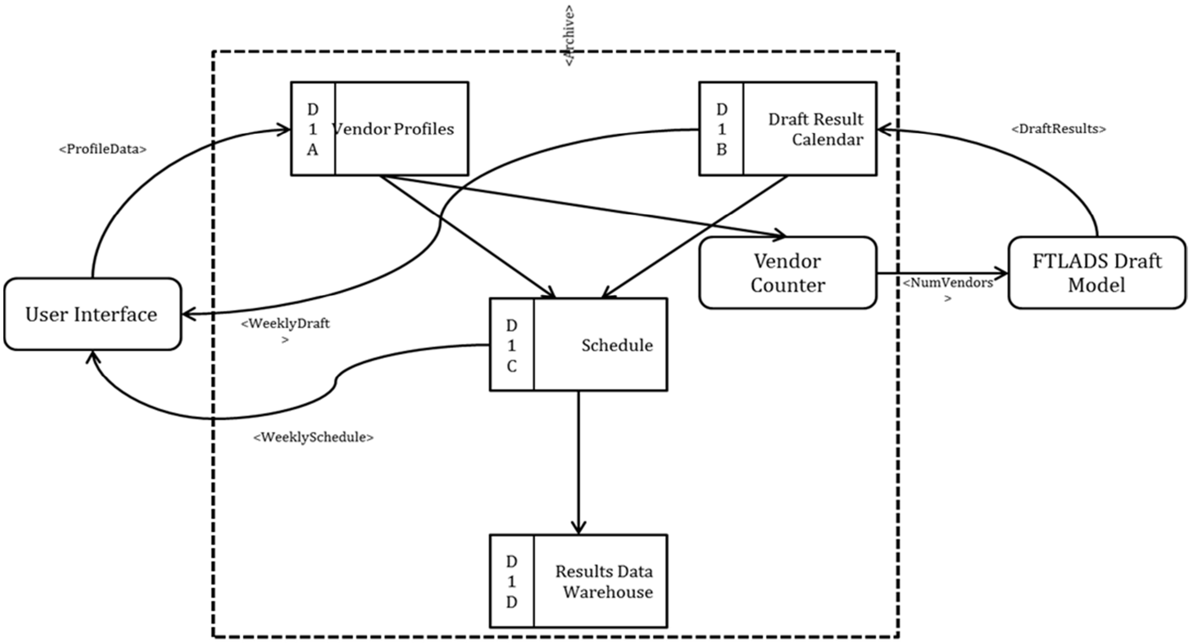
4.2. FTLADS Sub-System Description

4.2.1. The FTLADS Draft Model. The core of the location assignment system is its draft model. Instead of a system defined value assignment of locations, *the FTLADS model dispenses an ordered set of choices*. In other words, instead of matching Vendor A to Location X, the system simply says, "Today Vendor A is fourth in order of priority for his choice.**"**

The FTLADS employs the "Wheel algorithm" to achieve the above. This technique is currently under consideration by NBA, which has spent considerable capital in developing a fair methodology by which basketball teams get to pick new recruits. We have adapted this system to suit the needs of the DC-area food truck stakeholders.

We will provide a mathematical explanation of the model in Section 5.1 below. The following diagram provides a synopsis of the inputs, outputs and data transformation relevant to the FTLADS draft model.

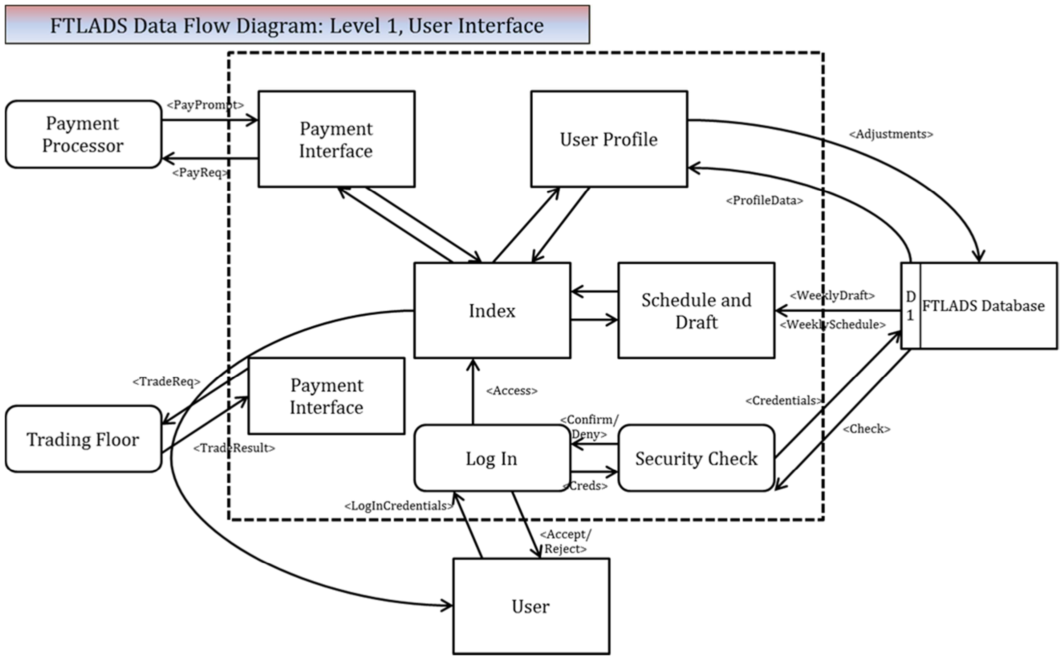
4.2.2. The FTLADS Database. The system database performs three key functions. First, it stores user (vendor) profile information, to include their daily location preferences. Second, the database stores the results of the FTLADS wheel algorithm in the form of daily draft results. Third, the database builds a weekly schedule based on vendor preferences and the draft results.



FTLADS Data Flow Diagram: Level 1, System DB

4.2.3. The FTLADS User Interface. The user interface allows the user to interact with every part of the system, other than the draft mechanism itself. This sub-system will be an HTML-based interface which will be compatible with the following web browsers: (1) MS Internet Explorer version 8.0 and later, (2) Mozilla Firefox version 3.0 and later and (3) Google Chrome version 41.0 and later.

The system's interface will be user friendly. No operation will take more than three screen transitions between action initiation and completion. A more thorough walk-through of user interface interactions is discussed in Section 6.1 of this document. However, for a general understanding of the inputs and outputs of the FTLADS user interface, the following data flow diagram is provided.



Note that for the purposes of readability, the data flow diagram above does not list all the interactions between the user and the various entities within the user interface. Here, all interactions between the user and interface entities (e.g. "User Profile") are routed through the "Index" page. In practice, the "Index" page will allow the user to access other entities and interact with them directly.

4.2.4. The FTLADS Payment Processor. The existing system features a payment processor interface. A full breakdown of this sub-system is not within the scope of this SDD.

4.2.5. The FTLADS Trading Floor. The secondary market or "Trading Floor" subsystem was designed by students from the George Mason, Volgeneau School of Engineering. It has been recently deployed. Please see the Design section for future integration.

4.3 Design Rationale

The central perceived problem that exists within the current food truck parking location assignment methodology is that it does not distribute high-value vending locations in an equitable way. This problem had been voiced by the vendors and DMVFTA. The first step, before we designed the FTLADS system, was to determine whether this problem was supported by empirical evidence, or whether it was purely a function of stakeholder perception. In other words, we needed to determine *whether the system was actually fair*. Thus, the null hypothesis was that the system is equitable while the alternate or "popular" hypothesis was that it was not.

The answer to this question would make for considerably different system designs. If it turned out that the existing system actually did distribute vendor locations equitably, then our focus would be on user interface aspects of the system. Alternately, if our analysis showed that popular sentiment was correct and that the existing system did not produce equitable results, then our design must focus on developing an assignment algorithm does produce fair outcomes.

Developing a method to test for fairness proved to be a task unto itself, and a full discussion of the analysis methodology and results can be found in Appendix A of this SDD. The fairness was assessed by comparing vendor preferences at each level and point in time (e.g. Thursday, 1st Preference is "Farragut Square") against those locations assigned to that vendor and time period (e.g. Thursday, Location assigned is "Metro Center"). Through a one-for-one comparison, we could then convert our results into a binary "yes" or "no" data set. From here, we were able to assess how often a test group of vendors received their preferences versus others. The more even the distribution, the more fair the system.

Our results were definitive; after eight months, the system had failed to evenly distribute high-value locations to vendors within our test sample. With this in mind, we focused our design on the vendor location assignment algorithm itself.

Recall that our guiding principles for development included fairness, transparency and user empowerment. After determining that that the existing system produced uneven results over extended time periods, we did a literature review for potential algorithm options that distribute preferences evenly among users.

The first concept our team considered was to use a bidding system whereby vendors might use either money or digital "tokens" to bid on various locations. Our analysis showed that certain locations were more highly valued by vendors than others. We also found that vendor preferences for locations varied by time. That is, any given location is much more valued on some days of the week than others. For example, Metro Center locations dominated in the middle of the week, while Farragut locations dominated on Fridays. Because of this two-variable gradient, our team assessed that a bidding mechanism might an appropriate choice as the basis for our system's base location assignment process. However, after discussion with our stakeholders, we determined that the second and third order effects of a bidding system were not acceptable. Specifically, stakeholders did not want to create a system that might allow well-financed vendors to drive up prices of highly valued locations such that smaller food truck vendors could not compete. This discussion evolved into a specific requirement (Stakeholder Requirement 3.1.6).

We then started looking at optimization algorithms. These algorithms attempt to apply a value function to the problem at hand and then generate results which optimize for these values. The weakness of such systems is that they imply a pre-defined value system. In other words, our team would have to rely on our own analysis to define which locations were most valuable and apply weights accordingly. Although we felt our analysis was strong enough to build such a system, this type of design is directly in contradiction of our third guiding principle of returning power to vendors. In other words, we wanted to avoid forcing our own valuations (even if justified) onto the system.

From here on, we considered implementing a drafting algorithm. We realized the potential of such a system after we concluded that the problem could be fully understood with one variable, a simple vendor "pick," as opposed to two, vendor preferences and location results. With this realization in hand, it became apparent that we might be able to evenly distribute location matches by simply distributing the order of vendor choices. That is, over a given period of time, each vendor will be able to pick his or her first choice the same number of times as everyone else. Likewise, he or she will be low in choice order an equivalent number of times.

5. Data Design

5.1. In Depth: The FTLADS Model

Based on the strategic approach described earlier in this document, our team started looking at sports based drafting algorithms in order to distribute equal chances to each of the truck vendors to pick their preferences. In our research we came across various algorithms: Snake algorithm, Ladder algorithm and the newly proposed NBA Wheel draft algorithm. There were similarities between how the Wheel draft worked compared to what we were trying to achieve. Hence, we utilized a part of that algorithm where the total draft pick numbers are divided by equally valued prime numbers and create groups. Now, these groups are arranged in an order based on the constraints of the problem and randomly a number is chosen from each group, in that order, without repetition. The string of numbers formed makes the "Wheel".

Our problem is unique as we have to equally distribute these draft numbers to 250 plus trucks while there are only 100 plus spots. This would mean that the truck vendors will have more off days than working days in a week. And we want to distribute these numbers equally on all days of the week for each truck vendor. In that sense, *our problem is harder than that faced by the NBA.*

Based on the data analysis of the DC jurisdiction, there were 3 popular streets identified - L'Enfant Plaza, Farragut Square and Metro Center. The lowest capacity for the above 3 streets is 12. Hence, the draft pick numbers 1 to 12 are *considered equally valued* numbers for creating this draft algorithm. In other words, whether a vendor gets a "1" or a "12" as his or her pick, the functional result is the same: the vendor is guaranteed to get whatever location he or she chooses for that day. Dividing the total draft numbers (as defined by current truck licenses) by 12 gives us 21 groups. The groups are arranged based on the constraints of the problem:

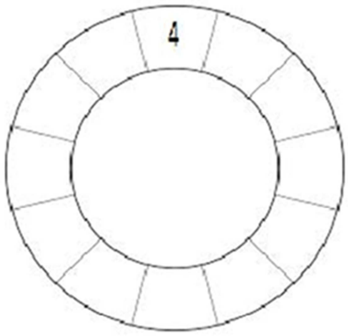
- No consecutive working days.
- The difference between the number of working days between any two trucks cannot be more than 1 over a given week. This is our "Envy Free Factor"
- Prime pick numbers to be distributed across all 5 days of the week for each truck vendor.

- Groups (1-12) and (13-24) are placed at 10 number of groups interval so that a truck vendor has a chance to pick the 1st or 2nd potential preference every 10 days. This is our “Perception Factor.”

With these constraints in mind, our algorithm constructs the “Wheel” using the following methodology. First, ranks are assigned from highest to lowest. These numbers are selected in the order of highest rank from each of the 21 groups to the lowest without repetition.

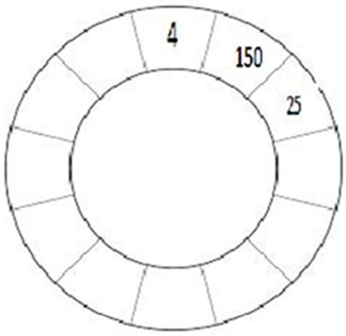
Constructing the Wheel: First Group Number Selection

1	2	3		5	6	7	8	9	10	11	12
145	146	147	148	149	150	151	152	153	154	155	156
97	98	99	100	101	102	103	104	105	106	107	108
25	26	27	28	29	30	31	32	33	34	35	36
181	182	183	184	185	186	187	188	189	190	191	192
37	38	39	40	41	42	43	44	45	46	47	48
133	134	135	136	137	138	139	140	141	142	143	144
193	194	195	196	197	198	199	200	201	202	203	204
49	50	51	52	53	54	55	56	57	58	59	60
217	218	219	220	221	222	223	224	225	226	227	228
229	230	231	232	233	234	235	236	237	238	239	240
13	14	15	16	17	18	19	20	21	22	23	24
109	110	111	112	113	114	115	116	117	118	119	120
241	242	243	244	245	246	247	248	249	250	251	252
85	86	87	88	89	90	91	92	93	94	95	96
157	158	159	160	161	162	163	164	165	166	167	168
73	74	75	76	77	78	79	80	81	82	83	84
121	122	123	124	125	126	127	128	129	130	131	132
169	170	171	172	173	174	175	176	177	178	179	180
61	62	63	64	65	66	67	68	69	70	71	72
205	206	207	208	209	210	211	212	213	214	215	216



Constructing the Wheel: Stepping Through the Draft Pick Groups

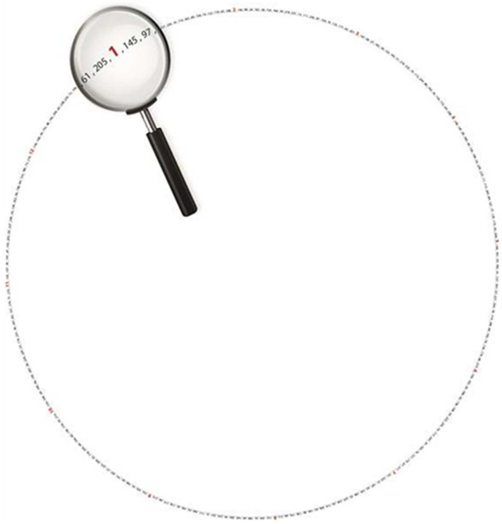
1	2	3		5	6	7	8	9	10	11	12
145	146	147	148	149		151	152	153	154	155	156
97	98	99	100	101	102	103	104	105	106	107	108
	26	27	28	29	30	31	32	33	34	35	36
181	182	183	184	185	186	187	188	189	190	191	192
37	38	39	40	41	42	43	44	45	46	47	48
133	134	135	136	137	138	139	140	141	142	143	144
193	194	195	196	197	198	199	200	201	202	203	204
49	50	51	52	53	54	55	56	57	58	59	60
217	218	219	220	221	222	223	224	225	226	227	228
229	230	231	232	233	234	235	236	237	238	239	240
13	14	15	16	17	18	19	20	21	22	23	24
109	110	111	112	113	114	115	116	117	118	119	120
241	242	243	244	245	246	247	248	249	250	251	252
85	86	87	88	89	90	91	92	93	94	95	96
157	158	159	160	161	162	163	164	165	166	167	168
73	74	75	76	77	78	79	80	81	82	83	84
121	122	123	124	125	126	127	128	129	130	131	132
169	170	171	172	173	174	175	176	177	178	179	180
61	62	63	64	65	66	67	68	69	70	71	72
205	206	207	208	209	210	211	212	213	214	215	216



As an example, Group [1-12] had 1 as the highest rank number, followed by Group [145-156] with 145 as the highest and so on for each of our 21 groups. The algorithm will pick “1,145 ...”

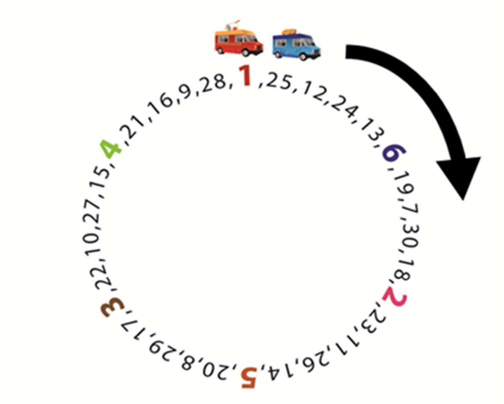
and so on until it has selected a number from all 21 groups. The algorithm will now start over at Group [1-12] and select the next highest ranked number from each of the 21 groups. This process will continue until all numbers from each of the groups are exhausted. The sequence of numbers that would be formed constitutes a "Wheel". Currently, the FTLADS prototype produces a wheel of 252 numbers and is integrated into a functional database.

A 252 Point Wheel is Created



Once the wheel is created and stored in the database, the system randomly pairs each truck vendor to a number on the wheel. This marks a draft cycle's initial state and decides each tuck vendor's picking number for the first day. The next day, each truck moves clockwise and takes the position of the truck that was ahead of it in the wheel. The number on that position becomes the next day's draft pick for that truck. This movement continues until the cycle is complete, at which point a new "wheel" is generated.

Rotating Vendors around the Wheel



By matching the draft wheel with vendor preferences of locations, our solution constructs a weekly assignment schedule. Truck vendors have until a set deadline (set by DCRA administrator) to lock in their street location preferences. After the cut-off time for changes to vendor preferences, the administrator will run the assignment algorithm.[2]

The assignment algorithm requests the street information (identifier, name, capacity) for each street in the database. Next, it requests vendor preferences, arranged on a per truck basis, and retrieves the truck picking order for each of day of the week in question (5 working days of the week) from the database. The algorithm then assigns vending locations based on the truck vendor's preferences ordered by the draft pick numbers for that day. Once the assignment process is completed, the schedule for each truck vendor is itself stored in the database published to the FTLADS interface; it may now be accessed by vendors who log on to the system.

5.2. The FTLADS Database

5.2.7. Database Overview. The FTLADS database accomplishes two purposes. First, it serves as the data link between all other components of FTLADS. Second, it provides the basis for forms which are used by vendors in order to input preferences and receive location and draft assignments. While the user works directly with a web-based interface, all processes are carried out via the database itself.

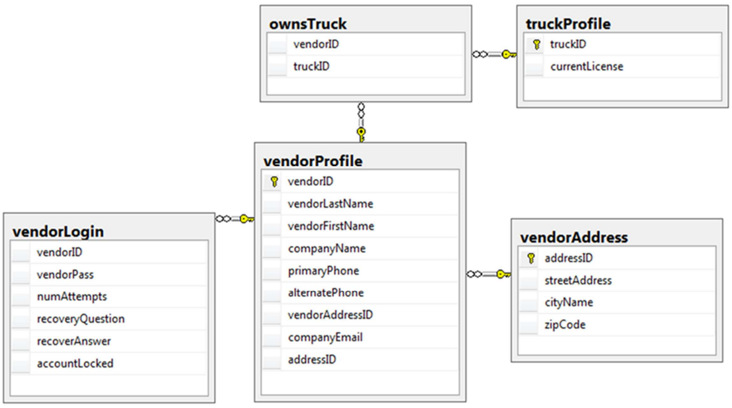
Additionally, the schema is designed so that it can be easily integrated with a data warehouse for pattern analysis purposes in the future. Note that this database is based on components of the existing system which were analyzed through interactions with the current web interface; our team did not have access to diagrams or source code that would more thoroughly describe the existing system.

5.2.8. Entity-Relationship Diagrams. Note that the following schema are built from executable SQL code. These diagrams were built using SQL Server 2012. Because the full schema is somewhat extensive, a single ER diagram for the entire database is difficult to read. For this reason, we have opted to break the ER diagram into components.

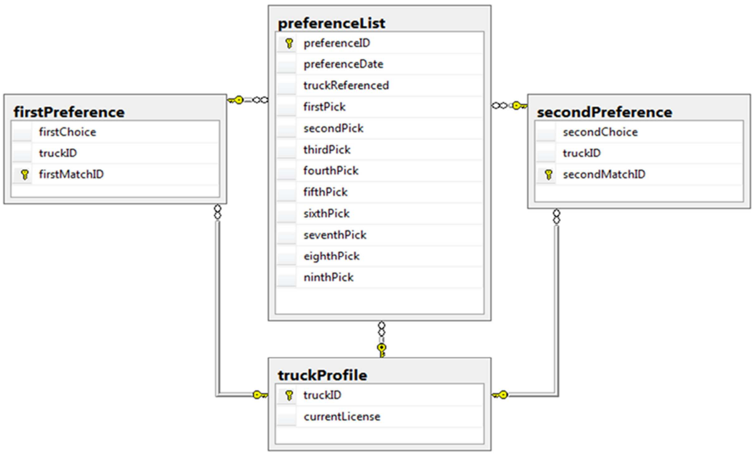
---

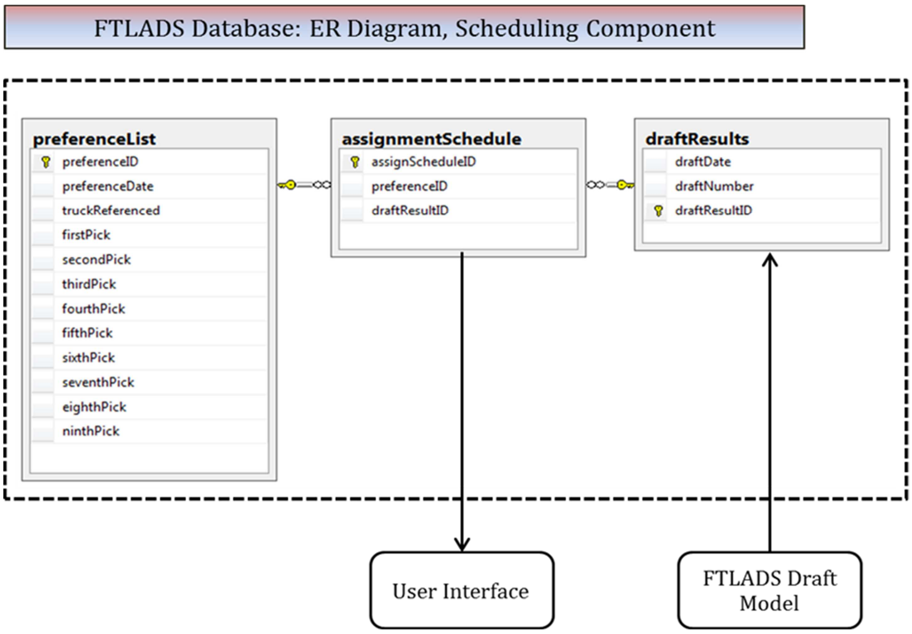
[2] Alternately, this action can be executed using a time based script that runs every week at a particular time. Our data flow diagrams incorporate this capability.

FTLADS Database: ER Diagram, Profiles Component



FTLADS Database: ER Diagram, Vendor Preferences Component





5.2.9. Schema. Please see Annex C (Prototype Database Code) for a thorough explanation of the FTLADS database schema. The code is executable, but is not a stand-alone prototype.

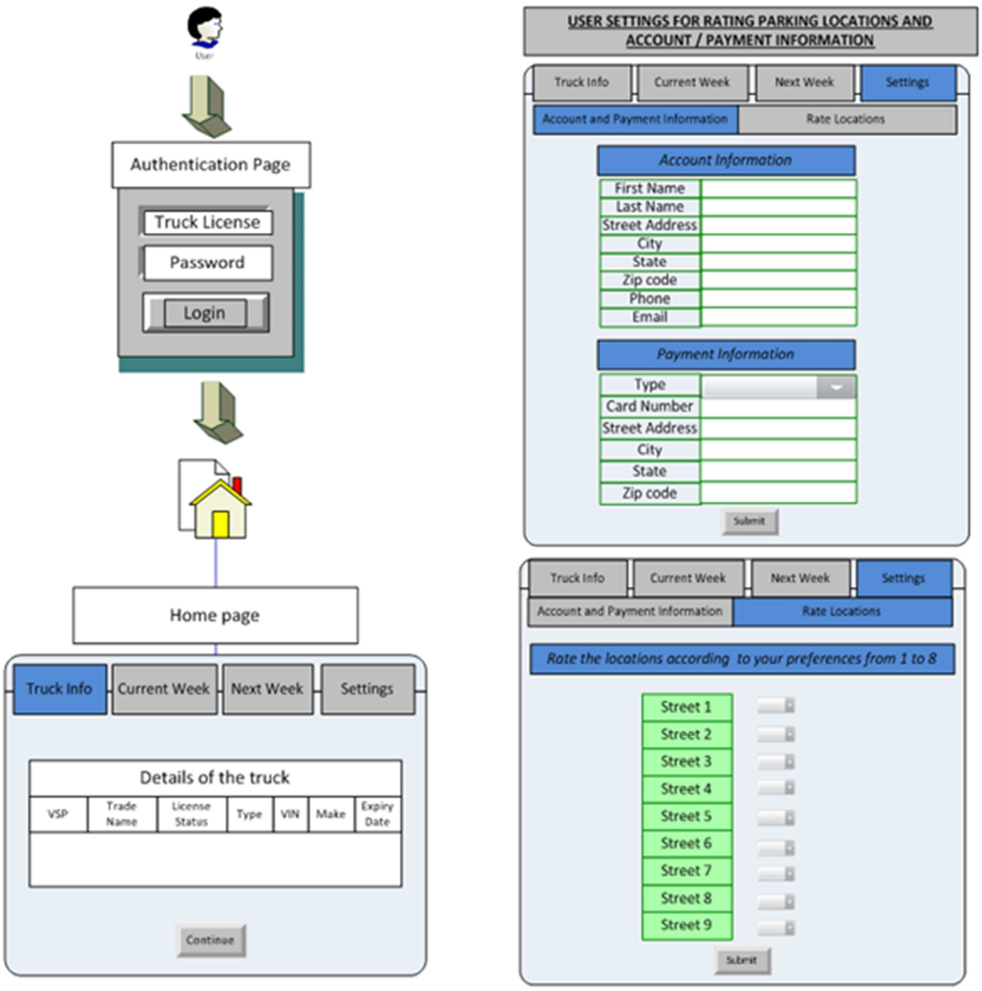
6. Human Interface

6.1. Overview

The proposed modified Web interface is designed to address the problems identified through discussions with stakeholders and observations from the current system interface, as described in detail in the following section.

6.2. Step-Through

The current user interface requests the vendor to input a preference matrix with 45 cells to account for location preferences for each working day. This process must be completed prior to each assignment cycle, and the web interface lacks a feature to save preference data. The modified interface is designed to request default ranking information as part of a profile to be saved for each vendor; thus offering the chance to avoid repeating the input process for vendors should their preferences remain static over time and days of the week, as shown in the following figure.

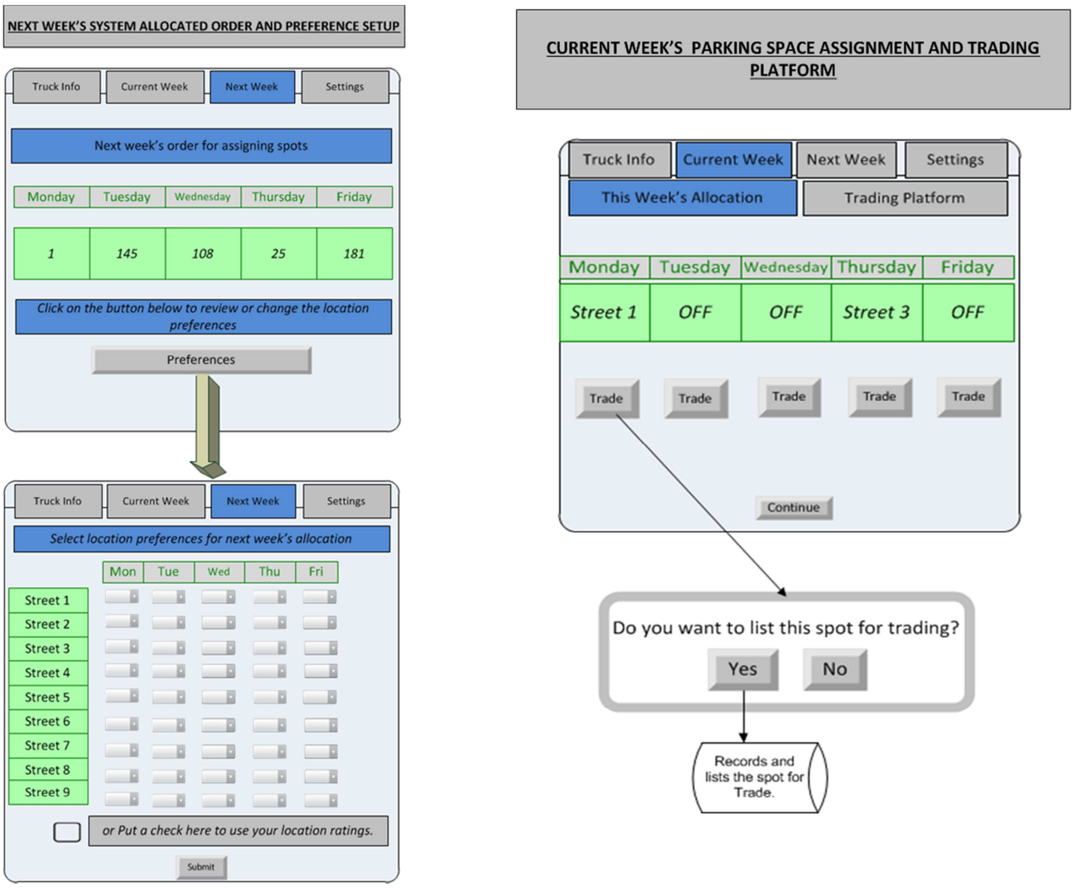


User Interface - Profile

The new system does maintain the current assignment mechanism should vendors intend to make adjustments to their default choices.

Through discussions with stakeholders, we learned that vendors find the lack of transparency in the current system's interface a concern. The current system's lottery assignment merely takes vendor preferences as an input, and outputs a schedule of assignments prior to the start of the month. In order to address this problem, a two-step assignment process is proposed. In the first step, through the drafting algorithm, vendors are assigned the random pick numbers before a request for their preferences is made. This process offers vendors a chance to decide on their location preference for a given day after learning about their chances of getting the requested location. For instance, if the random pick number given to a vendor was 25, the vendor would know that there will be 24 other vendors whose location request would be considered before him. Additionally, given the likelihood that several vendors would share an interest in the locations considered most popular, and the fact that locations have limited capacities, the vendor would anticipate that he will not be assigned the most popular location. This process helps vendors maintain more control over their assignment and aids the perception of fairness from the vendor's point of view. The vendor could then choose a less popular location of his own choice, and his chance of achieving his "new" preferred alternative location is increased. In the second step, the location assignment is announced at a later date based on the location preferences.

Another discussion with stakeholders revealed that a weekly assignment cycle is preferred to the current monthly assignment due to the fact that an out of favor location assignment that a vendor must occupy for a long period of time would increase the likelihood of vendor abandoning the assignment and causing under-utilization. The sign-up fee for vendors to take part in the lottery system is $25 and would not justify the costs to operate the food truck in a less popular location for business owners. The final modification in the interface is to integrate the recently proposed trading mechanism with the new interface, allowing for the trading mechanism to be used following the announcement of weekly schedule. The following figure displays the two-step draft process employed on a modified, weekly cycle.



User Interface – 2 Step Assignment

7. Conclusion

7.1. General Conclusions

In general, our analysis shows that FTLADS produces much more evenly distributed results than does the existing vendor location assignment system. Even better, the system does this in a shorter period of time than does the existing system. FTLADS provides evenly distributed assignments in less than five months, whereas the existing system has uneven results over an eight month period. A more thorough discussion of a comparison between FTLADS and the existing DC food truck vendor location assignment system can be found in Annex A, Analysis and Simulation.

Even distribution of our preference assignments is not the only strength of FTLADS, however. Our algorithm is specifically designed to produce mathematically even results, but also addresses potential vendor perception issues arising from the way preference assignments are distributed. Specifically, our algorithm guarantees that favorable draft results (i.e. low numbers) are interspersed at even intervals vice unfavorable results (i.e. high numbers). Additionally, we believe that the use of a draft system returns power to vendors, and encourages them to interact with the system more regularly. Whereas the current system locks vendors into set monthly schedules, FTLADS operates over two week intervals, with actual location matching changing week over week. This means that even if a vendor perceives that a particular week is not particularly favorable, he is not locked into this particular schedule for long.

Ultimately, we believe our system will meet the needs of all the identified stakeholders: DCRA, DMFTVA, and the food truck vendors themselves. We believe that the improvements our system makes in both usability and functional design makes FTLADS an attractive alternative when compared to the existing system. Specifically, we believe that, should it be implemented, FTLADS will be well received by all parties as it adheres to the principles of fairness, transparency and ease of use.

7.2. Recommendations and Future Efforts

This design document, and its supporting documentation, should offer a proven baseline for the development of a fully functional prototype. Indeed, as of the completion of this document significant progress has been made toward that end. In short, we believe that the next phase of this operation is implementation. As such, we would recommend that this phase of the effort *not* be selected for further efforts relative to an academic project in the fields of Systems Engineering and Operations Research (SEOR). What remain to be completed are coding, testing and deploying; the design aspect of this system is complete.

However, we do believe that there are components of the problem domain that would provide future SEOR students with appropriate work. Specifically, we believe that future student efforts should focus on the following.

7.2.1. Creation of a Data Warehouse. Data currently maintained by the existing system is limited in scope. For example, while it was more than sufficient to complete an analysis of vendor preferences over two variables, location and time, we could not account for seasonality or weather, two variables that are almost certainly relevant to the problem domain. A project dedicated toward improving this design such that it (a) incorporated a data warehouse and (b) integrated algorithms to conduct trend analysis would be exceptionally useful.

7.2.2. Integrate FTLADS with the Newly Deployed Trading Mechanism. This document accounts for previous work conducted by GMU students toward the development of a secondary trading mechanism. However, our primary focus revolved around analyzing the existing primary mechanism and designing an improved capability. Should DCRA decide to implement FTLADS, considerable work would no doubt remain with respect to phasing out the existing system and integrating our system with the secondary trading mechanism, which would remain in place.

In summary, should DCRA decide to implement FTLADS, we believe that this design document will be sufficient for a capable team to develop a prototype, test and implement the new system. While we believe FTLADS would be an excellent alternative to the existing DC-area vending location distribution system, it would certainly benefit from further efforts related to the two areas listed above.