**PROBLEM KEYWORDS APPLICABLE TO BOTH IMPORTERS:**

* **Warning:** 
  + **Does not stop the importing of the row but just warns that some crucial field(s) did not have data;**
  + **Example: not supplying the gender for the record being created;**
* **Entry error:** 
  + **Stops the importing of the row but the importer will proceed to the next row;**
  + **Displayed in the table at the bottom of the import page;**
  + **Example: not supplying the name of the record being created;**
* **General error:** 
  + **Stops the importing process altogether;**
  + **Displayed at the top of the import page;**
  + **Example: wrong encoding in the csv file.**

**PROSPECTS IMPORTER**

REQUIRED FIELDS

1. A value in the first name and/or middle name and/or last name;
2. Purpose

SUMMARY OF FUNCTIONALITY

1.     Name fields: First Name, Middle Name, Last Name, Suffix

* Leading and trailing whitespace will be removed (left and right);
* All name fields will be capitalized;
* Each word in case of multiple word names will be capitalized;
* Spaces in between multiple word names will be replaced by single space,

E.g.:   ‘   van   der   vert   ‘ becomes ‘Van Der Vert’;

* ‘&’ will be replaced by ‘ - ‘ ;
* Each word in case of multiple word names will be capitalized;

1. The suffix field will only take one of these: 'Jr.', 'Sr.', 'I', 'II', 'III', 'IV';

* If suffix contains the letters(case-insensitive) ‘s’ & ‘r’ we match it to ‘Sr.’;
* If suffix contains the letters(case-insensitive) ‘j’ & ‘r’ we match it to ‘Jr.’;
* If suffix contains hindu-arabic numerals we match it to the roman equivalent (‘1’ => ‘I’, ‘2’ => ‘II’, ‘3’ => ‘III’, ‘4’ => ‘IV’, but also ‘12’ => ‘I’).

2. School

* Can be blank;
* The school the prospect goes to - not the martial arts school.

3. Gender

* Warning displayed if blank;
* ‘female’, ‘f’, and string containing ‘f’  will be matched to ‘Female’, case insensitive;
* ‘male’, ‘m’ and strings containing ‘m’ will be matched to ‘Male’, case insensitive.

4. Birthdate

* Can be blank - no warning;
* Takes different formats: (01/01/12, 01-01-12, ‘January 1ts 2012’);

5. Contact info

* Street, Street 2, City, State, Zip, Country;
* If email is blank a warning will be displayed;
* For the address, we search for an existing address with matching ‘street’ column and tie that address to prospect if such address exists. Otherwise we create a new entry in our addresses table.

6. Lead state

* Can be blank - no warning;
* Default is ‘lead’.

7. Source

* Can be blank - no warning;
* If a source does not exist in school prospect sources table, it will be added.

8. Notes

* Can be blank;

9. Lead campaign id  (‘lc id’)

* A warning will be displayed if a lead campaign with the supplied id was not found;
* A warning will also be displayed if the header is present but the value is absent.

10. Purpose

* Error displayed if the header is not included;
* Error displayed if the field contains anything other than ‘create’ or ‘update’;
* Error displayed if purpose is ‘create’ but the name already exists;
* Error displayed if purpose is ‘update’ but the name was not found.

11. Supported headers. Parsing headers is case-insensitive and non-alphanumeric characters except spaces will be removed. Leading/trailing whitespace will be removed completely, spaces between words will be limited to a single space. Here is the list:

* '  'first name',
* 'middle name',
* 'last name',
* 'suffix',
* 'gender',
* 'street',
* 'street 2',
* 'city',
* 'state',
* 'zip',
* 'country',
* 'birthdate',
* 'phone',
* 'email',
* 'notes',
* 'school',
* 'source',
* 'lead state',
* ‘lc id’

VALIDATIONS

1.     Upon creating a new prospect record, we will search for the current schools’ prospect by the full(er) name (first\_name + middle\_name + last\_name). So if Alex John Smith with or without suffix exists in the database for the school, for instance, we will not be able to create a second Alex John Smith with or without suffix.

**STUDENTS IMPORTER**

**BASIC INFO**

REQUIRED FIELDS: first name, last name, ID, active.

1.     Name fields: First Name, Middle Name, Last Name, Suffix

* Leading and trailing whitespace will be removed (left and right);
* All name fields will be capitalized;
* Each word in case of multiple word names will be capitalized;
* Spaces in between multiple word names will be replaced by single space,

E.g.:   ‘   van   der   vert   ‘ becomes ‘Van Der Vert’;

* ‘&’ will be replaced by ‘ - ‘ ;
* Each word in case of multiple word names will be capitalized;
* The suffix field will only take one of these: 'Jr.', 'Sr.', 'I', 'II', 'III', 'IV';
* If suffix contains the letters(case-insensitive) ‘s’ & ‘r’ we match it to ‘Sr.’;
* If suffix contains the letters(case-insensitive) ‘j’ & ‘r’ we match it to ‘Jr.’;
* If suffix contains hindu-arabic numerals we match it to the roman equivalent (‘1’ => ‘I’, ‘2’ => ‘II’, ‘3’ => ‘III’, ‘4’ => ‘IV’, but also ‘12’ => ‘I’);
* Irish last name prefixes “Mc”, “Mac”, “O’”will be accounted for as far as capitalization.

2. Belt type

* Can be blank;
* Will be sripped of leading/trailing whitespace;
* Not capitalized.

3. Gender

* Warning displayed if blank;
* ‘female’, ‘f’, and string containing ‘f’  will be matched to ‘Female’, case insensitive;
* ‘male’, ‘m’ and strings containing ‘m’ will be matched to ‘Male’, case insensitive.

4. Birthdate

* Can be blank - no warning;
* Takes different formats: (01/01/12, 01-01-12, ‘January 1ts 2012’);

5. Contact info

* Street, Street 2, City, State, Zip, Country;
* If email is blank a warning will be displayed;
* For the address, we search for an existing address with matching ‘street’ column and tie that address to prospect if such address exists. Otherwise we create a new entry in our addresses table.

6. Source

* Can be blank - no warning;
* If a source does not exist in school prospect sources table, it will be added.

7. Notes & medical notes

* Can be blank;

8. School

* Can be blank;
* The school the prospect goes to - not the martial arts school.

9. Supported headers. Parsing headers is case-insensitive and non-alphanumeric characters except spaces will be removed. Leading/trailing whitespace will be removed completely, spaces between words will be limited to a single space. Here is the list:

* ‘id',
* 'active',
* 'name',
* 'first name',
* 'middle name',
* 'last name',
* 'suffix',
* 'gender',
* 'street',
* 'street 2',
* 'city',
* 'state',
* 'zip',
* 'country',
* 'birthdate',
* 'belt size',
* 'belt type',
* 'phone',
* 'email',
* 'notes',
* 'medical notes',
* 'program',
* 'rank',
* 'school',
* 'source',
* 'fm id',
* 'fm name',
* 'fm street',
* 'fm street 2',
* 'fm city',
* 'fm state',
* 'fm zip',
* 'fm country',
* 'fm phone',
* 'fm email',
* 'fm notes',
* 'guardian name 1',
* 'guardian phone 1',
* 'guardian phone type 1',
* 'guardian email 1',
* 'guardian email type 1',
* 'guardian relation 1',
* 'guardian occupation 1',
* 'guardian name 2',
* 'guardian phone 2',
* 'guardian phone type 2',
* 'guardian email 2',
* 'guardian email type 2',
* 'guardian relation 2',
* 'guardian occupation 2',
* 'emergency contact',
* 'emergency phone',
* 'emergency contact 2',
* 'emergency phone 2',
* 'membership program',
* 'membership start date',
* 'membership end date',
* 'national association membership number',
* 'national association membership expiration date',
* 'national association membership start date',
* 'national association membership type’.

**MEMBERSHIPS**

1. Membership program name: error displayed if not found in the school database;
2. Start/end dates: errors displayed if either or both missing.

**PROGRAMS/RANKS**

1. Program name: error displayed if not found in the school database;
2. Rank: error displayed if not found in the school database;
3. If both program and rank were not found, an error just for program will be displayed.

**FAMILY**

1. Headers (parsed same way, case-insensitive, non-alphanumeric characters and removed, leading/trailing whitespace removed, spaces in the middle trimmed to one):

*'fm id',*

*'fm name',*

*'fm street',*

*'fm street 2',*

*'fm city',*

*'fm state',*

*'fm zip',*

*'fm country',*

*'fm phone',*

*'fm email',*

*'fm notes'*

1. ‘fm id’ and ‘fm name’ are REQUIRED fields.
2. ID: will be used in combination with a family name to match the family being imported to a family that might already exist in the database. In case the family with the same name and ID already exists in the database, that family will be updated. If the family name exists but the ID is different, the creation of the new family will fail with an error message saying the name has already been taken. ID can follow ‘fm-1’ pattern but it doesn’t have to.
3. Name: cleaned/formatted the same way as student name fields.
4. Contact info is formatted/cleaned the same way as student contact info. Also:

* Address: the new address will override the old one;
* Emails: will be appended as long as the email address is unique;
* Phone numbers: will be appended as long as the number is unique.

1. Family notes: cleaning limited to leading/trailing whitespace removal.

*Note: it is assumed that basic student data (at least the required fields) will be in the same csv file as the family data that is being imported.*