**1. High level Requirement**: Insert CSV file entry into DB.

**2. Broken down requirement into user stories**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S/N | Functionality | As a | I want to | So that | Acceptance criteria and business rules |
| 1 | Create Upload Folder | User | View the upload folder | I can upload csv files | System checks for the existence of the upload folder, if it does not exist in that location, it creates one. |
| 2 | Upload file | User | Upload a file into the upload folder | It can be inserted into the DB | User can use the GUI to upload file into the folder |
| 3 | Discover CSV files in the upload folder | admin | Find the csv files in the upload folder have been updated in the DB | So that I don’t have to do it myself | The system checks the upload folder for .csv files. If it doesn’t exist, nothing happens. |
| 4 | Parse Rows in CSV file | admin | Be aware that the system parses rows in the file | The correct rows are inserted into the db | When the system finds a csv file, it extracts the rows in the file and inserts it into the appropriate tables in the db ignoring the first row. |
| 5 | Create a processed directory | User | Csv files that have been updated to be marked as done | I can differentiate from undone ones | The system checks if the processed directory exists, if not it creates one. Then the system now moves all successfully used csvs into the processed directory. |
| 6 | Check syntax of csv | Admin | Make sure separations in the files are commas | The data is properly used. | The system returns an error if the data is not properly separated. |
|  | Check syntax of csv | user | Know the proper syntax to use | I can properly formulate my csv files | User can clearly see the format to use. |

